Fall 1996

Reaction kinetics of methanol and MTBE oxidation and pyrolysis

Wen-Chiun Ing
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

UMI Number: 9618588

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

ABSTRACT

REACTION KINETICS OF METHANOL AND MTBE OXIDATION AND PYROLYSIS

by
Wen-chiun Ing

This study presents experimental data on the decomposition of methanol in several different reaction environments - fuel lean to stoichiometric at a temperature range of 873 and 1073 K and a pressure range of 1 and 5 atm. Methane fuel is also added in several of the systems studied in order to provide experimental data to understand the methanol addition effect on the methane oxidation.

Computer codes: ThermCal, ThermSrt and ThermCvt have been developed for the thermal property calculations of stable molecules by the Benson group additivity method and of radicals by the NJIT hydrogen bond increment method.

Pressure dependent rate coefficients have been expressed using Chebyshev polynomials adopted for complex chemical activated reaction systems in this study, as well as unimolecular decomposition reactions. This method has also been tested and shows significant improvement over two convention methods, Troe's and SRI. The Levenberg-Marquardt algorithm has been incorporated with the QRRK code, CHEMACT, for the fitting of Chebyshev polynomials.

A pressure dependent mechanism which consists 147 species and 448 elementary reactions, based on thermochemical kinetic principals has been developed and calibrated by the experimental data. The reaction mechanisms (models) include pathways for formation of higher molecular weight products, such as the formation of methyl ethers. This accurate model based on principles of thermochemical kinetics and statistical mechanics will not only provide fundamental understanding, but can be used to suggest directions toward process optimization for experimental testing.

The CHEMKIN interpreter has been modified in this study to take the flexible matrix size of Chebyshev polynomials and to generate the appropriate link file for further processing. Subroutines involving kinetic rate coefficient calculation and array size initialization are also modified to incorporate the Chebyshev polynomials expression. Existing CHEMKIN drivers can simply link with the modified CHEMKIN library to take temperature and pressure dependent Chebyshev polynomials without modification. A couple of drivers have been linked with the new CHEMKIN library and tested. These include a shock tube driver, a constant temperature and pressure driver, and an adiabatic constant pressure driver.

The mechanism is validated with methanol oxidation and pyrolysis experimental data and serves as a basis to build upon during the subsequent efforts on higher molecular weight oxygenated hydrocarbon (MTBE in this study). The methanol addition shows dramatic acceleration effect on the methane oxidation experimentally and predicted by the model.

REACTION KINETICS OF METHANOL AND MTBE OXIDATION AND PYROLYSIS

by
Wen-chiun Ing

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Chemical Engineering,
Chemistry, and Environmental Science

January 1996

APPROVAL PAGE

REACTION KINETICS OF METHANOL AND MTBE OXIDATION AND PYROLYSIS

Wen-chiun Ing

Dr. Joseph W. Bozzelli, Dissertation Advisor — Date
Distinguished Professor of Chemistry, NJIT

Dr. Robert B. Barat, Committee Member — Date
Associate Professor of Chemical Engineering, NJIT

Dr. Dana E. Knox, Committee Member — Date
Associate Professor of Chemical Engineering, NJIT

Dr. Lev N. Krasnoperov, Committee Member — Date
Associate Professor of Chemistry, NJIT

Dr. Michael Booty, Committee Member — Date
Associate Professor of Mathematics, NJIT

BIOGRAPHICAL SKETCH

Author: Wen-chiun Ing

Degree: Doctor of Philosophy in Chemical Engineering

Date: January 1996

Undergraduate and Graduate Education:

- Doctor of Philosophy in Chemical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1996
- Master of Science in Environmental Science,
  New Jersey Institute of Technology, Newark, NJ, 1990
- Diploma in Chemical Engineering,
  National Taipei Institute of Technology, Taipei, Taiwan, R.O.C., 1983

Major: Chemical Engineering

Presentations and Publications:

Ing, Wen-chiun; Guo, Cheng; Krasnoperov, Lev N.; Bozzelli, Joseph W. "Pyrolysis and Oxidation of Methanol/Methane Mixtures: Experimental and Detailed Reaction Model", *Clean Fuels Processing section, AICHE annual meeting*, San Francisco, California, USA, Nov 13-18, **1994**.

Ing, Wen-chiun; Chen, Chiung-Ju; Bozzelli, Joseph W. "Pyrolysis and Oxidation of MTBE (Methyl Tert-Butyl Ether) and Methanol: Experimental and Detailed Reaction Model", *Combustion Reaction Engineering section, AICHE annual meeting*, San Francisco, California, USA, Nov 13-18, **1994**.

Ing, Wen-chiun; Guo, Cheng; Krasnoperov, Lev N.; Bozzelli, Joseph W. "Pyrolysis and Oxidation of Methanol/Methane Mixtures: Experimental and Detailed Reaction Model", *13th International Symposium on Chemical Reaction Engineering*, Baltimore, Maryland, USA, Sep. 25-28, **1994**.

Krasnoperov, Lev N.; Ing, Wen-chiun; Dean, Anthony M.; Bozzelli, Joseph W. "Study on the $CH_2OH + O_2$ and $CH_2O + HO_2$ Reaction system: Potential Energy

Diagram and Branching Rations by Master Equation", *13th Internation Symposium on Gas Kinetics*, Dublin, Ireland, Sep. 11-16, **1994**.

Ing, Wen-chiun; Guo, Cheng; Krasnoperov, Lev N.; Bozzelli, Joseph W. "Pyrolysis and Oxidation of Methanol/Methane Mixtures: Experimental and Detailed Reaction Model", *208th National Meeting of the American Chemical Society*, Washington DC, USA, Aug., **1994**.

Krasnoperov, Lev N.; Ing, Wen-chiun; Dean, Anthony M.; Bozzelli, Joseph W. "Study on the $CH_2OH + O_2$ and $CH_2O + HO_2$ Reaction system: Potential Energy Diagram and Branching Rations by Master Equation", *208th National Meeting of the American Chemical Society*, Washington DC, USA, Aug., **1994**.

Ing, Wen-chiun; Chen, Chiung-Ju; Krasnoperov, Lev N.; Bozzelli, Joseph W. "Pyrolysis and Oxidation of Methanol/Methane Mixtures: Experimental and Detailed Reaction Model", *The Twenty-Fifth International Symposium on Combustion*, Irvine, California, USA, July 31-Aug. 5, **1994**.

Krasnoperov, Lev N.; Ing, Wen-chiun; Dean, Anthony M.; Bozzelli, Joseph W. "Study on the $CH_2OH + O_2$ and $CH_2O + HO_2$ Reaction system: Potential Energy Diagram and Branching Rations by Master Equation", *The Twenty-Fifth International Symposium on Combustion*, Irvine, California, USA, July 31-Aug. 5, **1994**.

Ing, Wen-chiun; Chen, Chiung-Ju; Guo, Cheng; Krasnoperov, Lev N.; Bozzelli, Joseph W. "Fundamental Study of the High Temperature Chemistry of Oxygenated Hydrocarbons as Alternate Motor Fuels and Additives", *Symposium on Prioritizing Hazardous Air Pollutants*, Cambridge, Massachusetts, USA, July 5-8, **1994**.

Ing, Wen-chiun; Ho, Wen-pin; Krasnoperov. Lev N.; Bozzelli, Joseph W. "Study on Ethylene and Hydroxyl-Ethyl Radical Oxidation in the Atmosphere: Ethylene + OH and $C.H_2CH_2OH$, $CH_3C.HOH$ + Oxygen", *7th BOC Priestley Conference*, Lewisburg, Pennsylvania, USA, Jun. 24-27, **1994**.

Ing, Wen-chiun; Guo, Cheng; Krasnoperov, Lev N.; Dean, Anthoy M.; Bozzelli, Joseph W. "Pyrolysis and Oxidation of Methane/Methanol Mixtures", *Internation Symposium on Gas Conversion and Utilization*, New York, USA, May 9-11, **1994**.

This dissertation is dedicated to
my wife, 蘇秋慧

ACKNOWLEDGMENT

I wish to express my appreciation to Prof. Joseph W. Bozzelli, my advisor, not only for his professional advice but also his encouragement, patience, and kindness. I am deeply indebted to him for the opportunities which he made available to me.

I would also like to thank to my dissertation committee members, Dr. Dana E. Knox, Dr. Robert B. Barat, Dr. Lev N. Krasnoperov and Dr. Michael Booty for their helpful corrections and productive comments.

It is my pleasure to thank Dr. Yo Ping Wu, Dr. Wen Pin Ho, who shared their experiences with me and helped me with GC/MS analysis. In addition, I like to thank my coworkers at NJIT, Dr. Hong-Ming Chiang, Xian Zhong and Chiung-Ju Chen, for having dealt with me as a colleague, which has made my time at NJIT much more enjoyable and productive.

For love and inspiration I shall be eternally grateful to my wife, 蘇秋慧 and my parents, 應信岳, 鄭春. Without their constant support and encouragement, I truly believe all of this would not have been possible.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Overview

Alcohols, such as methanol and ethanol, are under consideration for widespread use as alternative motor fuels (1). Ethers, such as methyl tert-butyl ether (MTBE), are already in limited use as anti-knock components in gasolines (2).

A key factor which will determine the utility of oxygenated hydrocarbons (OHC's) as dedicated fuels, blend components, or additives is their environmental impact, especially on urban air pollutants such as ozone and peroxyacetyl nitrates (PAN's). Methanol is of interest as an additive and an alternative to conventional transportation fuels, because of potential reductions in pollutant emissions. Specifically, methanol is indicated to aid in reduction of unburned hydrocarbons, which enhance ozone production and photo-chemical smog formation, but enhanced emissions of formaldehyde from methanol oxidation might actually contribute to urban smog (3). The usefulness of OHC's will also depend on their engine performance characteristics. For example, laminar burning velocities of iso-octane / methanol blends in air are lower than the velocity of either component alone (4).

Methanol is easily produced from petroleum and non-petroleum resources such as coal, wood etc. It is also an important early intermediate formed in the combustion of the high-octane fuel additive MTBE (methyl tert-butyl ether) (5).

During the 1990s, methyl tert-butyl ether (MTBE) will be one of the fastest growing chemicals in the world. Between 1980, the first year in which the U. S. International Trade Commission began reporting separate data on MTBE, and 1984, production has grown at an annual compound rate of 20% (6). U. S. consumption of MTBE has been reported to be growing at a rate of 40% per year in 1986 (7). About forty plants are running and thirty more are under construction or planned (8) including three plants which will be built in Canada with a total capacity of 1.5 million metric tons (9-10}.

If all projects planned in 1987 are built, MTBE production will reach about 20 billion lb/yr which is about the double of the production in 1987 (11). MTBE demand for methanol will triple to 1.2 billion gal, up from 400 million gal in 1989 and cause the global demand for methanol to grow more than 18% (12).

There are several factors causing the demand for MTBE: the lead phasedown in gasoline in both the U. S. and in Europe, the failure of other nonleaded oxygenates to significantly improve octane ratings and therefore to serve as replacements for tetra ethyl lead in gasoline, and the restrictions of U. S. Environmental Protection Agency on gasoline RVP (Reid Vapor Pressure) in order to decrease tropospheric ozone concentrations (13).

All the refiners in United States and most of those in Europe have accepted MTBE as an octane enhancer. It blends like a hydrocarbon and does not have the handling problems that alcohols do. It can also solve the octane problems of refiners without requiring large additional capital investment. Many processes have been

developed to reduce the cost of MTBE production (14-19), such as : a new process using sulfuric acid catalyst (14), an adsorption process reducing control problems (15), adding a MTBE unit ahead of alkylation in refinery processing (16), a bifunctional catalyst process (17)... etc.(18-19)

1.2 Objectives of this Study

The goals of this study are :

- Development and advancement of the understanding of oxidation processes on oxygenated fuels under atmospheric and high pressure conditions.
- Identification of important reactions and key species in the high temperature chemistry of OHC's.
- Acquisition of experimental data from lab experiments and from the archival literature on oxidation of the oxygenated fuels, i.e. methanol and MTBE mixed with hydrocarbons.
- Developmentj of a pressure and temperature dependent kinetic model, based on thermochemical kinetic principles and calibrated by experimental data, which will allow computer experiments to suggest trends for future experimental testing and preferred fuel compositions that reduce hydrocarbon emissions while hopefully maintaining or improving engine performance.
- Development of computer tools for the above objectives. These incorporate a pressure dependent mechanism into kinetic modeling and are used in conjunction with a modified Sandia National Lab CHEMKIN general kinetic library. The modification is included in this study.

This study presents experimental data on the decomposition of methanol in several different reaction environments - fuel lean to stoichiometric at a temperature range from 873 to 1073 K and a pressure range from 1 to 5 atm. Methane fuel is also added in several of the systems studied in order to provide experimental data on the use of methanol as an additive in motor vehicle fuels. This data will be a basis, along with that in the literature, for validation of a model of methanol oxidation and pyrolysis which can be used for evaluation and simulation of methanol combustion under atmospheric conditions as well as in compression engines or turbines, where pressures are very different from atmospheric.

Methanol was studied as a base case. During oxidation, all higher OHC's can be considered to pass through the same or similar intermediates as methanol. Methanol is also a key intermediate in the oxidation of MTBE. Some experimental data on methanol oxidation already exists, however, modeling based on fundamental principles is lacking. For example, pressure dependencies of reaction rate constants, which are critical to describe oxidation in internal combustion engines correctly, were not considered. The proposed study was therefore begun with limited available experimental data on methanol pyrolysis and oxidation. A fundamental elementary reaction model has therefore been generated to describe data from both the literature and our own experimental data. These studies on methanol will provide experimental and modeling bases upon which to build during subsequent research on OHC's with higher molecular weight than MTBE.

An important issue here is the impact of OHC's on the formation of higher molecular weight hydrocarbons and soot in diesel engines. This is likely to occur in fuel rich, pyrolytic zones. Addition of an appropriate OHC to the fuel has the potential to limit molecular weight growth while not adversely affecting engine performance (20-21). Ethane is chosen as a model compound fuel. Its pyrolytic chemistry is presented as a function of OHC additive.

1.3 Literature Review

1.3.1 Methanol

The oxidation and pyrolysis of gas phase methanol has been studied widely over the past half century by several different methods: diffusion flame, shock tube and static or flow reactors. Study on the pyrolysis and oxidation of methanol by diffusion flames was reported by Smith and Gordon (22) in 1956. They used a quartz probe sampling technique to extract stable species from the flame regions and the samples were analyzed with a consolidated analytical mass spectrometer. The temperature measured in the flames varied from about 200 C at the wick to about 1400 C at the tip and edge of the flame. From the analysis of these products, they indicated that the mechanism of burning involves pyrolysis of the alcohol followed by oxidation of the pyrolysis products. The pyrolysis of methanol was free radicals induced by the small percentage of O2 which diffuses in near the base of the burner, and H or OH radicals which come from the reaction of O2 an H2.

Earlier studies were reported by Fletcher (23) in 1934 and by Someno (24) in 1942. Fletcher studied the pyrolysis of methanol in a static system at pressure less than

atmospheric and temperatures between 899 and 1003 K. He proposed that the pyrolysis occurs in two stages; the first stage gives formaldehyde and hydrogen, and in the second stage, formaldehyde is pyrolyzed to CO and H2.

Someno studied the thermal decomposition and slow combustion of monohydric alcohols. He worked in quartz vessels at sub-atmospheric pressure and temperatures up to 923 K. The identification of products was made spectrographically, by IR and visible spectroscopy, and Ketones, aldehydes, ketenes and unsaturated hydrocarbons were found in the products. Formaldehyde was found in the products only under conditions of slow oxidation.

A shock tube study on the ignition of methanol with oxygen was made by Cooke and co-workers (25). The experiment was performed using stoichiometric mixtures of methanol and oxygen diluted with 95% argon at shock generated temperatures from 1573 to 1873 K, pressure range from 200 to 300 Torr. They found that methanol was less reactive than ethanol. In high temperature conditions, the reactions of $CH_3OH \rightarrow CH_3 + OH$ and $CH_3OH \rightarrow CH_2OH + H$ were reported to evolve with fission of the O-H bond less likely to occur.

Bowman studied the high temperature oxidation of methanol behind reflected shock waves in the temperature range from 1545 to 2180 K (26). The reaction of methanol-oxygen mixtures diluted with argon was initiated by reflected shock waves. The concentrations of O, OH, H_2O and CO were measured by various spectroscopic techniques. Following shock-heating, oxidation appears to proceed via two distinct phases: an induction period, in which the concentrations of radical species and water

increase rapidly with little change in temperature; followed by exothermic reaction with the concentration of radical species and water slowly approaching equilibrium values. He proposed a small reaction mechanism and indicated that the significant reactions include thermal decomposition of methanol, attack of radicals O, OH and H on methanol and thermal decomposition of the important radical intermediate CH_2OH.

Aronowitz et al. (27) reported methanol oxidation results ($CH_3OH/O_2/N_2$) in an adiabatic, turbulent flow reactor, having plug flow characteristics which allowed for chemical sampling with high spatial resolution. Experiments were performed at atmospheric pressure, in the temperature and equivalence ratio ranges from 680 to 880 °C and 0.03 to 3.16, respectively. Major products included CO, CO_2 and H_2O, with small amounts of H_2 and CH_2O. Trace quantities of hydrocarbon products were observed at more fuel rich equivalence ratios. They presented a two-step overall or global model to describe the oxidation of methanol to carbon dioxide. They specifically emphasized the importance of HO_2 chemistry in the oxidation of methanol over this temperature region at atmospheric pressure, while Westbrook et al. (28) had previously indicated that hydroperoxyl radicals can be important in high temperature chemistry.

A study of methanol combustion in laminar flames was made by Vandooren and Van Tiggelen (29) in 1981. A detailed mechanism and rate constants of elementary steps for lean methanol flames burning at 40 Torr were investigated by using molecular beam sampling coupled with mass spectrometric analysis. They reported that "about 70% of the fuel molecules were consumed by reaction with hydroxyl radical and 30% by reaction with hydrogen atom". The main initial product was CH_2OH which then

reacts mostly with molecular oxygen. The occurrence of electronically excited CH_2O was detected, and attributed to highly exothermic reactions between CH_2OH and O, H or OH radicals. They indicated that the formaldehyde conversion occurred via reactions with radicals, mainly OH, or by unimolecular (low pressure limit) decomposition processes.

Important elementary reactions of CH_2OH with molecular and atomic oxygen in the methanol oxidation system were studied by Grotheer et al (30) using a direct discharge flow reactor over temperature range from 25 to 400 °C at pressures around 1 mbar. Radical profiles were monitored by a low-energy electron impact ionization mass spectrometer. The rate coefficient was measured for the CH_2OH + O reaction. A strong non-Arrhenius behavior for $CH_2OH + O_2$ reaction was reported.

Recent studies on methanol oxidation and pyrolysis have been reported by Norton and Dryer (31-32). In 1989, they performed methanol oxidation experiments using a turbulent flow reactor at equivalence ratios in the range from 0.6 to 1.6 and initial temperatures from 1025 to 1090 K at atmospheric pressure. They indicate that the existing and widely-used mechanism (33) for methanol oxidation is in error and should be updated.

Norton and Dryer present a kinetic mechanism for methanol pyrolysis which matches multiple sets of experimental data from static, flow, and shock tube reactors, covering temperatures of 973 to 1993 K and pressures of 0.3 to 1 atmosphere. They indicated that the fuel decomposition reaction $CH_3OH \rightarrow CH_2OH + H$, previously been included only in mechanisms for high temperature conditions, also has a significant

effect at low temperatures through radical combination reaction. They reported that the reaction $CH_3O + CO \rightarrow CH_3 + CO_2$ rather than $CH_3OH + H \rightarrow CH_3 + H_2O$, was the major source of CH_3 at low temperatures and the reverse of $CH_3 + OH \rightarrow CH_2OH + H$ was important to CH_3 production at high temperatures.

One of the principle problems with the Norton Dryer mechanism is that it does not consider pressure effects of the important unimolecular and bimolecular reactions. For example, the pressure dependence of the decomposition steps below are not fully included or analyzed for different systems:

$CH_3OH \rightarrow CH_3 + OH$

$CH_3OH \rightarrow {}^1CH_2 + H_2O$

$CH_3OH \rightarrow {}^1HCOH + H_2$

$CH_3OH \rightarrow CH_2OH + H.$

More importantly, the correct path way analysis for :

$CH_2OH + O_2 \rightarrow$ [.OOCH2OH]# $\rightarrow$ $.OCH_2OH + O$

$\rightarrow$ [HOOC.HOH]# $\rightarrow$ $HCO_2 + OH$

$\rightarrow$ [CH2O...HOO]# $\rightarrow$ $CH_2O + HO_2$

$\rightarrow$ [HOOCH2O.]# $\rightarrow$ $CH_2O + HO_2$

$\rightarrow$ $CH_2OH + O_2$ (Reverse - No Rxn)

$\rightarrow$ $.OOCH_2OH$ (Stabilization)

as well as reactions of $CH_3O + O_2$ to form $[CH_3OOO]$# and its further reactions were not included.

We note that the Norton-Dryer mechanism does not properly incorporate the negative temperature dependence, pressure effects and correct reaction pathways which account for the observations by Grotheer.

These pressure dependent reactions are very important in the formation of CO and CO_2. In particular, it is important to evaluate properly the CO_2 formation path from HCO_2 formation in the reaction of $CH_2OH + O_2$ listed above. This reaction proceeds through a chemically activated intermediate, which can be stabilized, dissociate back to reactants (no reaction) or react via isomerization to either HOOC.HOH or $HOOCH_2O$. isomers; both of which will then immediately dissociate to lower energy products.

Further items missing from present mechanisms include pathways for the formation of higher molecular weight products, such as the formation of methyl ethers, which we observe in this study and report for the first time.

1.3.2 MTBE (Methyl Tert-butyl Ether)

T. J. Wallington (34) conducted a flash photolysis resonance fluorescence study on the kinetics of the hydroxyl radical (OH) reaction with MTBE over the temperature range from 240 to 440 K at total pressures between 25 and 50 Torr. The Arrhenius expression for the overall rate constant was determined as follows :

$C_3COC + OH \rightarrow$ Products

$k_1 = (5.1 \pm 1.6) \times 10^{-12} \exp[-(155 \pm 100)/T]$ cm^3/mole sec

We note that this rate constant includes abstraction of both the methyl ether hydrogens and the primary methyl tert-butyl hydrogens.

A reaction study was made earlier by Cox and Goldstone, but only at room temperature and it was a relative-rate investigation, yielding on overall rate constant at 298 K (35).

There have been three previous experimental kinetic studies of rate constant determinations on the thermal decomposition kinetics of MTBE up to temperatures of 1160 K, and they conclude that the reaction is a four-center molecular elimination,

$(CH_3)_3COCH_3 \rightarrow (CH_3)_2C{=}CH_2 + CH_3OH.$

Daly and Wentrup (36) report that the decomposition reaction of MTBE is unimolecular, and they determine a rate constant, of $10^{14.38}$ exp(-30970/T) s^{-1} over the temperature range from 706 to 768 K. Choo et al. (37) report a value of $k = 10^{13.9}$ exp(-29700/T) s^{-1} in a VLPP (Very Low-Pressure Pyrolysis) reactor from 890 to 1160 K, while Brocard and Baronnet (38) obtained $k = 10^{14.0}$ exp(-29960/T) s^{-1} in a Pyrex reactor at 725 to 761 K. The latter workers also find that the addition of propene or toluene does not modify the rate of formation of the major products, methanol, and they conclude that a four-center unimolecular elimination path accounts for the homogeneous thermal decomposition of MTBE over the temperature and pressure ranges of their study.

Other studies on the high temperature oxidation of MTBE in an argon diluent have been reported by Dunphy and Simmie (39) in reflected shock waves over the temperature range from 1024 to 1850 K and at a pressures of 3.5 bar. The mixture compositions varied widely, with equivalence ratios varying from 0.25 for a fuel-lean, through 1.0 for a stoichiometric, to 2.5 for a fuel-rich mix. Measurements of the

ignition delay times, characterized by chemiluminescence and pressure rise, can be concluded that the high temperature oxidation of MTBE is in essence that of methanol + iso-butene.

Norton and Dryer (40) presented the experimental results for flow reactor oxidation of MTBE at equivalence ratio 0.96, initial temperature of 1024 K and atmospheric pressure. Gas samples extracted at fifteen positions along the reactor duct centerline were quenched in the hot-water-cooled probe and stored at 343 K for later gas chromatographic analysis. The purity of the MTBE was 97%.

The main products they observe are isobutene and methanol. At 1024 K, 50% MTBE is converted in a time of about 5 msec, and 90% of conversion is achieved at about 40 msec. Isobutene and methanol are generated at up to 75% and 40% of initial MTBE concentration, and decay slowly compared to the MTBE decomposition rate. An increasing amount of carbon monoxide is produced with MTBE conversion. Methane and propene are observed at about 10% of MTBE initial concentration. Other, lower concentration, products (less than 10% of MTBE initial concentration) such as CO_2, C2, C3, and C5, are also produced.

Norton and Dryer describe the result of MTBE decay by the unimolecular elimination reaction only, which occurs through a four-center activated complex. They attribute the final products observed to faster MTBE oxidation rather than to either isobutene or methanol oxidation.

CHAPTER 2

EXPERIMENTAL

The oxidation and pyrolysis of methanol with and without methane is analyzed by varying temperature, pressure, residence time and reactant ratios. A temperature range from 873 to 1073 K, pressure range from 1 to 5 atm and residence time range from 0.1 to 2.0 seconds is studied in two reaction ratios (stoichiometric and fuel lean) using a high temperature and high pressure tubular reactor for methanol oxidation. Methanol pyrolysis is studied over a pressure range of 1 to 10 atm at 1073 K. Oxidation of Methanol and methane mixtures is studied with an overall stoichiometric reactant ratio at selected temperatures and pressures. Methane oxidation under stoichiometric conditions, with pressure range from 1 to 5 atm and temperatures from 1023 to 1073 K is studied for comparison with methanol and methanol/methane oxidation and for further validation of kinetic models.

2.1 Experimental Conditions

2.1.1 Methanol Pyrolysis

Methanol pyrolysis is studied over a pressure range from 1 to 10 atm at 1073 K under isothermal reactor conditions. The reactant concentrations at different pressures are:

1 atm	CH_3OH : Ar = 0.0395 : 0.9605
3 atm	CH_3OH : Ar = 0.01317 : 0.98683
5 atm	CH_3OH : Ar = 0.0079 : 0.9921
8 atm	CH_3OH : Ar = 0.00494 : 0.99506
8 atm	CH_3OH : Ar = 0.0025 : 0.9975
10 atm	CH_3OH : Ar = 0.00395 : 0.99605

2.1.2 Methanol Oxidation

Methanol oxidation experiments are performed with two reactant ratio sets which represent stoichiometric and fuel lean conditions over a temperature range from 873 to 1073 K and a pressure range from 1 to 5 atm. The reactant concentration ratios are:

Fuel Lean	$\phi = 0.75$	$CH_3OH : O_2 : Ar = 0.0078 : 0.0156 : 0.9766$
Stoichiometric	$\phi = 1.0$	$CH_3OH : O_2 : Ar = 0.0078 : 0.0117 : 0.9805$.

2.1.3 Methanol/Methane Oxidation

Methanol/methane oxidation is carried out at overall stoichiometric conditions at selected temperature and pressure in order to understand the effects of methanol addition on methane oxidation and the effects of methane on methanol oxidation. The total concentration and ratio of methanol and methane at the indicated temperature and pressure are:

Temperature	Pressure	$X_{o,CH4+CH3OH}$	$CH_3OH : CH_4$
873 K	5 atm	0.0156	0.0 : 2.0
873 K	5 atm	0.0156	0.2 : 1.8
873 K	5 atm	0.0156	0.3 : 1.7
873 K	5 atm	0.0156	0.5 : 1.5
873 K	5 atm	0.0156	0.7 : 1.3
873 K	5 atm	0.0156	1.0 : 1.0
873 K	5 atm	0.0078	2.0 : 0.0
873 K	5 atm	0.0078	1.5 : 0.5
873 K	5 atm	0.0078	1.3 : 0.7
873 K	5 atm	0.0078	1.0 : 1.0

Temperature	Pressure	$X_{o,CH4+CH3OH}$	$CH_3OH : CH_4$
1023 K	1 atm	0.0156	1.0 : 1.0
1073 K	1 atm	0.0156	2.0 : 0.0
1073 K	1 atm	0.0156	1.0 : 1.0
1073 K	1 atm	0.0156	0.0 : 2.0

2.2 Experimental Apparatus

A diagram of the experimental apparatus is shown in Figure B.1. Reactants are reagent grade supplied by Aldrich Co. Methane and argon gases are filtered for O_2, H_2O, and hydrocarbon impurities before entering the reactor system. The carrier gas (argon) is passed through a saturation bubbler which contained liquid methanol held at 273 K

using an ice bath. A second argon flow stream (after the bubbler) is used as make-up gas in order to achieve the desired reactant ratio. Methane and oxygen are then added to the CH_3OH/Ar flow as required. Those four gas streams (argon, make-up argon, oxygen and methane) are controlled by Union Carbide LINDE® Model FM-4550 mass flowmeter-flowcontroller with four mass flow control modules. Four channels are calibrated to indicate 0-100% range of the desired gases. The calibration curves are plotted as Figures B.2 to B.5 of Appendix B.

A quartz tube of 6 mm ID and 12 mm OD is employed as the reactor, which is housed within 75 cm length of three-zone Chemshell 1.25" ID electric tube furnace equipped with three independent Omega Model CN-310 digital temperature controllers. A Neon Controls BPS 26G2501, 200 psi back pressure regulator is used to maintain the desired pressure within the reactor.

The mixed reactants (feed mixture) are preheated to about 373 K to prevent condensation and to improve reactor temperature control. The reactants can either flow through the reactor or flow directly to a GC sampling valve via a bypass line. The bypass is used to determine the initial concentration of reactants without going through the high temperature reactor.

Gas samples are drawn through the sampling line by means of a mechanical vacuum pump with a constant flow rate of 30 cm^3/min. A HP-5890 Series II gas chromatograph with two flame ionization detectors is used on-line for analysis. The bulk of the effluent is passed through a sodium bicarbonate ($NaHCO_3$) flask for neutralization before being released to a fume hood.

2.3 Temperature Control and Measurement

Temperature profiles are obtained at each flow using a type K thermocouple probe moved axially within the 75 cm length reactor. Thermocouple error caused by furnace wall radiation is minimized by using a grounded junction sheath and with a representative flow of inert. The darkened outside surface of the quartz tube reactor also served as a second rdiation shield. Tight temperature control resulted in temperature profiles isothermal to within ± 5 K over 80 - 85% of the furnace length for each temperature. Steep temperature gradients of 500 K in 5 cm occurr at the inlet and outlet of the reactor.

Uncertainty in absolute temperature measurements is estimated to be ± 1% (i.e. ± 8-12 K) but relative temperatures are measured to within ± 5 K. The temperature profiles are shown as Figures B.6 and B.7 with and without reactions present.

2.4 Qualitative and Quantitative Analysis

A HP-5890 Series II gas chromatograph with two flame ionization detectors is used on-line to determine the concentration of reactants and products. A ten-port VALCO gas sampling valve is employed to introduce the gas samples into the GC columns. Gas samples are passed through two sampling loops, 1.0 cm^3 and 0.25 cm^3, at a constant flow rate of 30 cm^3 per minute, and then injected into a packed column and a capillary column, respectively.

Two columns, one packed and one capillary, are used to perform separations. The 6' × 1/8" feet stainless steel column is packed with 50% 80/100 Poropak T and 50% 80/100 Poropak Q for the separation of CO, CO_2 and light hydrocarbons. In order

to increase the accuracy of quantitative analysis, a catalytic converter connected in series after the packed column with 5% of 80/100 ruthenium on alumina catalyst is used to convert the CO and CO_2 to methane after separation. The 90 m × 0.53 mm Hewlett Packard fused silica capillary column is used for heavier hydrocarbon and oxy-hydrocarbon separation. The chromatogram peaks are analyzed with two HP 3396A integrators.

Calibration for obtaining appropriate molar response factors and retention times of relevant compounds is performed by injecting known concentrations of standard gases and known quantities of liquid samples. The average retention times and relative response factors are shown in Tables A.1 and A.2.

Product identifications are also verified by HP 5899A GC/Mass Spectrometry, with a HP 90 m × 0.53 mm fused silica capillary columns (same as the one used in the on-line GC system), on batch samples of reactor gas drawn, from the reactor exit into evacuated 25 cm^3 stainless steel sample cylinders for later analysis.

2.5 Summary of Experimental Results

2.5.1 Methanol Pyrolysis

Experimental results on methanol pyrolysis at 1073 K over a pressure range from 1 to 10 atm are displayed in Figures B.8 to B.13. The decay rate of methanol increases with increasing pressure, however, the differences between 5, 8 and 10 atmospheres are small.

Methanol decays and forms the intermediate stable product formaldehyde, which then decomposes to carbon monoxide. Normalized concentrations (C/C_o) of

methane formation are about the same at different pressures. Ethylene, which results from methyl radical combination through ethane, is observed at ppm levels with 3.95 mole % initial methanol concentration at 1 atm total pressure.

2.5.2 Methanol Oxidation

Results of methanol oxidation are shown in Figures B.14 to B.61, which plot the mole fraction as a function of reaction time and temperature, at temperatures ranging from of 873 and 1073 K, 3 pressures (1, 3, 5 atm) and two equivalence ratios of 1.0 and 0.75.

Changes in methanol conversion, intermediates and products under conditions of excess oxygen (equivalence ratio = 0.75) are small relative to reaction at stoichiometric conditions for all temperatures and pressures of this study. Methanol starts to decay at 923 K and 100% conversion is observed at 1073 K, at 1 atm and reaction time of 0.15 second. At 3 atm and a reaction time of 0.6 second, methanol decay begins at 873 K and is 95% converted at 923 K. At 5 atm and a reaction time of 0.6 second, methanol conversion is 5% at 873 K and 100% at 923 K.

At increased temperatures, formaldehyde levels decrease and this intermediate only exists for a short residence time (ca. 0.2 seconds). CO increases rapidly with the conversion of methanol. The formation of CO_2 starts to occur when significant amounts of CO produced. At higher temperature, oxidation of CO occurs rapidly to the final product CO_2.

Methane is observed as a minor product. Hydrocarbon molecular weight growth species are not detected (or are below instrument detection limits).

2.5.3 Methane / Methanol Oxidation

Experimental results on the oxidation of methane / methanol mixtures are shown in Figures B.62 to B.73. Methanol is observed to have a significant acceleration effect on methane oxidation. A uniform trend is observed over the temperature and pressure ranges studied by changing the initial composition of methane / methanol mixtures. This acceleration by methanol on methane oxidation will be discussed and compared with modeling results in Chapter 6.

CHAPTER 3

KINETIC MODELING

A mechanism which consists 147 species and 448 elementary reactions, based on thermochemical kinetic principles has been developed and calibrated by the experimental data.

Specific emphasis has been placed on understanding and properly treating the pressure dependent reactions and identifying important reactions and key species in the high temperature chemistry so the model can be applied to both atmospheric and internal engine oxidation / combustion. The reaction mechanisms (models) include pathways for formation of higher molecular weight products, such as the formation of methyl ethers, which are observed in the experiment and reported here for the first time. An accurate model based on principles of thermochemical kinetics and statistical mechanics will not only provide fundamental understanding, but will suggest directions toward process optimization for experimental testing. It will also have a higher probability of successful application outside the range of calibration.

The reaction mechanism is based upon principles of thermochemical kinetics including Transition State Theory (TST) and accurate molecular thermodynamic properties. The mechanism consists of elementary reactions, with each reaction based on literature evaluation or, if it is estimated, on thermochemical and kinetic principles.

3.1 Thermodynamic Properties

The addition reactions and subsequent unimolecular isomerization or dissociation reactions are analyzed by construction of potential energy diagrams of the systems based on existing experimental data, theoretical data and on group additivity estimation techniques.

Thermodynamic parameters - ΔH_{298}, S_{298} and $C_p(300)$ to $C_p(\infty)$ for species in the reaction schemes are listed in Table 1 along with appropriate references. Enthalpies of radicals are from evaluated literature on C-H bond energies and ΔH_f of the stable molecule which corresponds to the radical with a H atom at the radical site. Entropies and $C_p(T)$ values are from use of Hydrogen Bond Increment (HBI) (41). The HBI group technique is based on known thermodynamic properties of the parent molecule and calculated changes that occur upon formation of a radical via loss of a H atom. The HBI group incorporates evaluated carbon hydrogen (C-H) bond energies, for ΔH_{298} of the respective radical, and changes that result from loss or changes in vibrational frequencies, internal rotation symmetry, and spin degeneracy when a hydrogen atom is removed from the specific carbon site. HBI groups, are described fully in Ref. (42, 43).

3.2 Determination of Rate Coefficients

3.2.1 Abstraction Reactions

Abstraction reaction rate constants are not pressure dependent and usually taken from evaluated literature when available. If estimation is required, a generic reaction is utilized as a model and adjusted for steric effects. Evans Polanyi analysis is used on the reaction in the exothermic direction to estimate the energy of activation (E_a) for rate constant. An Evan Polanyi plot, E_a versus ΔH_{rxn}, allows use of a known ΔH_{rxn} to obtain

E_a for these reactions. The abstraction reaction in an endothermic reaction must incorporate the ΔH_{rxn} or thermodynamics will be violated.

3.2.2 Addition Reactions

Addition reactions are treated with the quantum RRK formalism for k(E) and modified strong collision theory of Gilbert et al. for falloff. This will be described in the following section. The reaction involves addition of an atom or radical to an unsaturated species. Addition reactions typically form an energized adduct with ca. 20 to 50 kcal/mol of energy above the ground state. This energy is often sufficient to allow the adduct to react to other reaction products (lower energy) before stabilization occurs.

3.2.3 Elimination Reactions (Beta Scission)

These reactions are the reverse of addition reactions and utilize the quantum RRK formalism for k(E) and modified strong collision theory of Gilbert et al. for falloff and are treated in two ways. In the first way, a unimolecular quantum RRK formalism for k(E) and modified strong collision theory of Gilbert et al. for falloff is employed where reverse high-pressure reaction (addition) parameters are determined. The corresponding high-pressure unimolecular beta scission rate constants using microscopic reversibility are then calculated. These high pressure unimolecular elimination parameters are then input to the QRRK formalism to calculate the rate constants at the appropriate pressure. An alternate method is to use the reverse rate constants from the QRRK - modified strong collision reaction calculations for the corresponding addition reaction.

3.2.4 Dissociation Reactions (Simple Unimolecular)

Simple unimolecular (dissociation) rate constants are determined by two methods similar to beta scission reactions. The reverse high-pressure reaction (combination) parameters are determined. The corresponding high-pressure unimolecular dissociation rate constants are then calculated using microscopic reversibility. These high pressure unimolecular dissociation parameters are then input to the QRRK formalism to calculate the rate constants at the appropriate pressure and temperature. The reverse rate constant from the QRRK combination reaction calculation can also be used for the respective reaction.

3.2.5 Combination Reactions

These reactions involve the combination of two radicals or an atom with a radical. The energy of the adduct formed before stabilization is equal to the bond energy of the new bond(s) formed and this is typically on the order of 80 to 120 kcal/mol. This is often sufficient under combustion conditions for an adduct, with this initial energy above its ground state energy, to react to lower energy products before stabilization occurs. The high-pressure limit rate constant for combination is obtained from literature or estimated from known generic combination reactions. The QRRK chemical activation formalism is then used to calculate the rate constants at the appropriate pressure and temperature to all the recognized available channels.

Reaction to other new product channels as well as isomerization, stabilization and reverse reaction are included in this calculation. This an important aspect of the reaction analysis for both combination as well as addition reactions that other modelers do not usually incorporate. This leads to a more correct treatment of fall-off and

pressure dependence for these non-elementary reaction systems. Rate constants inclusion in for the model are obtained which incorporate the calculated pressure dependency and therefore make the model more fundamentally correct.

3.3 Quantum RRK Treatment

Branching ratios of the adduct formed from combination, addition or insertion reactions to various product channels are calculated using a quantum version of RRK theory (QRRK) to evaluate energy dependent rate constants, k(E), of the adduct to the various channels. QRRK analysis, as initially presented by Dean (44, 45, 46), combined with the "modified strong collision approach" of Gilbert et al. (47) is used to compute rate constants for both chemical activation and unimolecular reactions over a range of temperature and pressure. For the chemical activation reactions, stabilization is also included in the calculations.

A significant number of modifications have been made since the initially descriptions of the quantum RRK and fall-off calculations are published (44, 45, 46). These modifications include (46);

- The use of a manifold of 3 vibration frequencies and respective degeneracies, plus incorporation of energies from 1 external rotation mode for the calculation of the ratio of the density of states to the partition coefficient, $\rho(E)/Q(T)$.
- The 3 vibrational frequencies and degeneracies are also used in calculation of k(E) and of F(E).
- The FE factor is now explicitly calculated for use in determining the collision efficiency βc (47), in place of the previously assigned 1.15 value.

- β_c is now calculated by :

$$\beta_c = [\alpha_c /(\alpha_c + F_E*k*T)]^2/\Delta$$

from Gilbert et. al. Eqn. 4.7 (47),

$$\Delta = \Delta_1 - (F_E*k*T)/(\alpha_c + F_E*k*T)*\Delta_2.$$

Where Δ_1 and Δ_2 are temperature-dependent integrals involving the density of states, and α_c is the average energy of down-collisions.

- The Lennard-Jones collision frequency Z_{LJ} is now calculated by $Z_{LJ} \equiv Z$ $\Omega^{(2,2)}$ integral. (48, 49, 50), where Ω is obtained from fit of Reid et al. (50)

The QRRK analysis with the modified strong collision approach and constant FE for fall-off has been used to analyze a variety of chemical activation reaction systems , Westmoreland et al. (51, 52, 53), Dean et al. (52, 54, 55, 56, 57), Bozzelli et al. (54-57, 58, 59, 60) It is shown to yield reasonable results in these applications, and provides a mechanism by which the effects of temperature and pressure can be both evaluated and included in the kinetics.

Limitations affected by the assumptions in the QRRK and fall off calculations are often over shadowed by uncertainties in high pressure limit rate constants and thermodynamic properties for species and transition state structures in the chemical systems.

Input data requirements for the QRRK calculations :

- Pre-exponential factors (Arrhenius A factors) in the high pressure limits, which are obtained from the literature using the methods of Benson (61) and of Dean (44).

- Activation energies come from endothermicity of reaction ΔU_{rxn} and from analogy to similar reactions with known energetics.
- The Three vibration frequencies and their associated degeneracies are computed from fits to heat capacity data, as described by Ritter.(62) These have been shown by Ritter to accurately reproduce molecular heat capacities, Cp(T), and by Bozzelli et al.(63) to yield accurate density of states $\rho(E)$ to partition coefficient (Q) ratios. Frequency sets for the adducts are listed in Table 1.
- Lennard-Jones parameters (σ, e/k) are obtained from tabulations(50) and from a calculation method based on molar volumes and compressibility (64).
- Arrhenius A factors for bimolecular combination of radicals at the high pressure limit are obtained from literature, and from trends in homologous series of these type reactions.

It is important to have accurate input data for the Calculations, without this aspect the accuracy or assumptions in the calculations on fall off or chemical activation rate constants are of less value. Several of the input parameters for these calculations, e.g., thermodynamic properties and kinetic parameters, need to be estimated. This is done in a consistent and uniform manner with reference to literature experiment and / or theoretical calculations in all cases. The accumulation and referencing of these data comprise a major component of this work. The input parameters and references are of

utmost importance; they are assembled as indicated in the tables associated with each calculation in Appendix C.

3.4 Expression of Rate Coefficients: Chebyshev Polynomials

Detailed reaction modeling of gas-phase kinetics is a powerful tool in the studies of combustion and atmospheric chemistry. Rate coefficients for dissociation, recombination and chemically-activated reactions can be estimated by statistical theories such as QRRK (44, 51), RRKM (65, 66, 67) and SACM (68, 69, 70, 71). Temperature and pressure-dependent rate coefficients of elementary reactions can be evaluated using one of these theories. However, calculation of rate coefficients is computationally demanding. Empirical or semiempirical approximate formulas, therefore, are generally practiced.

Approximations of fall-off surfaces of simple unimolecular reactions using empirical expressions has been extensively studied such as Troe's F_{cent} method and method of Stewart and coworkers (SRI method hereafter) (47, 72). Less attention has been taken for complex chemically-activated systems. Recently, Kazakov et al. (73) gave a parametrization formula called generalized-mean-limits formula (GML) based on Gardiner's (74) treatment. They report good agreement for "isothermal" fitting over pressure domains. Venkatesh et al. (75) directly approximate the rate coefficients via Chebyshev polynomials and report good agreement over broad temperature and pressure range for a number of different types of reaction systems. In light of the accuracy using Chebyshev polynomials for complex chemical activated reaction

systems, this method has been tested, employed and compared with two convention methods, Troe's and SRI, in this study.

3.4.1 Chebyshev Approximants

Chebyshev series (76, 77) in the inverse temperature and logarithm of pressure are used as the approximation of the logarithm of the rate coefficients. Temperature and pressure are constrained by predefined minimum and maximum values,

$$T_{min} \leq T \leq T_{max},$$

$$P_{min} \leq P \leq P_{max}.$$

The Chebyshev polynomial of degree *(i - 1)* is given by

$$\varphi_i(x) = \cos((i-1)\arccos(x)) \; ; \; i = 1, 2, \ldots,$$

where x is the variable of interest defined to be in the close interval [-1, +1]. The transformations to map the temperature and pressure domain onto the unit square are

$$\frac{2T^{-1} - T_{min}^{-1} - T_{max}^{-1}}{T_{max}^{-1} - T_{min}^{-1}} \rightarrow \tilde{T},$$

and

$$\frac{2\log P - \log P_{min} - P_{max}}{\log P_{max} - \log P_{min}} \rightarrow \tilde{P}.$$

$k(T,P)$ is then mapped to $k(\tilde{T},\tilde{P})$,

$$k(T,P) \rightarrow k(\tilde{T},\tilde{P}).$$

The logarithm of the rate coefficient is thus approximated as

$$\log k(\tilde{T},\tilde{P}) = \sum_{i=1}^{N}\sum_{j=1}^{M} a_{ij}\varphi_i(\tilde{T})\varphi_j(\tilde{P}),$$

where N and M denote temperature and pressure axis respectively and are predefined for different orders of accuracy.

The data points are taken as $d \times d$ *Gauss-Chebyshev grid* (76, 77) which is given by

$$\tilde{T}_i = \cos\left[\frac{2i-1}{2d}\pi\right], \quad \left(-1 \le \tilde{T}_i \le 1\right),$$

$$\tilde{P}_i = \cos\left[\frac{2i-1}{2d}\pi\right], \quad \left(-1 \le \tilde{P}_i \le 1\right),$$

where $\le i \le d$.

A 50 x 50 Gauss-Chebyshev grid is taken to fit N by M Chebyshev polynomials using Levenberg-Marquardt algorithm in the next section for the reaction systems of interest.

3.4.2 Comparison of Modified Troe and SRI Methods and Chebyshev Polynomials

The comparison of average and maximum errors of reactions tested for modified Troe, modified SRI , 7×3 and 9×5 Chebyshev approximants are shown as Table 3.1. The modified Troe and SRI methods are taken as Venkatesh et al. (75). First, the low- and high-pressure limit rate coefficients are enhanced and expressed as

$$k_0(T) = \left(a_0 + \frac{a_1}{\sqrt{T}} + \frac{a_2}{T} + \frac{a_3}{T\sqrt{T}} + \frac{a_4}{T^2}\right) \exp\left(-\frac{E_0}{RT}\right)$$

and

$$k_\infty(T) = \left(b_0 + \frac{b_1}{\sqrt{T}} + \frac{b_2}{T} + \frac{b_3}{T\sqrt{T}} + \frac{b_4}{T^2}\right) \exp\left(-\frac{E_\infty}{RT}\right),$$

where $a_0, a_1, a_2, a_3, a_4, E_0, b_0, b_1, b_2, b_3, b_4$, and E_∞ are adjustable parameters. Another enhancement is to extend the original single well treatment by explicitly adding the number of wells for different product channels. These modified methods are tested and showed better performance for multi-well chemical-activated reactions than the original Troe and SRI methods.

C_2H_6 decomposed to form $C_2H_5 + H$ and $CH_3 + CH_3$ is a typical unimolecular dissociation reaction system. The rate coefficients of later channel are displayed as Figure B.74 which shows typical high- and low-pressure limit and fall-off curves of the unimolecular reaction through the isothermal slices. From Table 3.1, the average errors of all of methods are less than 10% for C_2H_6 decomposition and maximum errors are less than 32%. The performances of modified Troe, modified SRI and Chebyshev polynomials of 7 by 3 are similar. The Chebyshev polynomials of 9 by 5 give excellent agreement - less than 2% average errors and 5% maximum errors.

The results of single well chemical-activation reactions of $CH_3 + O_2$ are similar to the C_2H_6 reactions. Chebyshev polynomials of 9 by 5 still yields best performance - less than 1% average errors and only 4% maximum errors.

For multi-wells chemical activation reactions of $CH_2OH + O_2$, the results are dramatically different with unimolecular dissociation and single well chemical activation reactions. Rate coefficients as a function of temperature and pressure for first two channels are plotted as Figure B.75 (a) and (b). It is obviously hard to fit results of rate coefficients for multi channels with traditional single well semi-empirical expressions, even extended version of them. The modified Troe method yields about 150%

maximum errors for first the two channels and the modified SRI method yields about 700% maximum errors. The results of two well reaction of of $CH_2OH + O_2 \leftrightarrow CH_2O + HO_2$ are the worst than the other one well reactions. The average and maximum errors of Chebyshev 7 by 3 polynomials are about 10% which are still in an acceptable range for kinetic modeling. Chebyshev 9 by 5 polynomials can give excellent performance if more accurate expression is required. The price here is more parameters (45, about double the number for 7 by 3) and much longer CPU time for the fitting (more than four times of CPU time for 7 by 3 fitting). In some complicated systems, the price of CPU time is high for the speed of workstations or midrange mainframes. Therefore, a balanced choice is 7 × 3 Chebyshev polynomials. This is also adopted in the kinetic modeling of this study.

The absolute relative errors of these three channels for 7 × 3 and 9 × 5 Chebyshev polynomials are displayed as Figure B.76, B.77 and B.78. The dramatic improvement using 9 × 5 parameters can be seen from the different scale of error axis.

Table 3.1 Comparisons of Average/Maximum Absolute Relative Errors in the Rate Coefficients

Reaction	modified Troe	modified SRI	Chebyshev Approximants	
			7 x 3	9 x 5
$C_2H_6 \leftrightarrow C_2CH_5 + H$	0.05 / 0.15	0.04 / 0.09	0.08 / 0.24	0.02 / 0.05
$C_2H_6 \leftrightarrow CH_3 + CH_3$	0.10 / 0.29	0.10 / 0.32	0.08 / 0.27	0.01 / 0.02
$CH_3 + O_2 \leftrightarrow CH_3(OO)$	0.08 / 0.19	0.05 / 0.19	0.07 / 0.21	0.01 / 0.04
$CH_3 + O_2 \leftrightarrow CH_2O + OH$	0.05 / 0.33	0.04 / 0.24	0.08 / 0.30	0.01 / 0.02
$CH_2OH + O_2 \leftrightarrow HOC(OO)H_2$	0.68 / 1.53	0.79 / 6.87	0.10 / 0.51	0.02 / 0.08
$CH_2OH + O_2 \leftrightarrow CH_2O + HO_2$	0.62 / 1.45	0.88 / 7.11	0.15 / 0.52	0.02 / 0.06
$CH_2OH + O_2 \leftrightarrow HOC(O)H_2 + O$	0.39 / 0.80	0.44 / 3.81	0.07 / 0.27	0.02 / 0.05

3.5 Computer Codes Used / Developed for Kinetic Modeling

3.5.1 ThermCal (developed in this study)

The thermodynamic properties related to this reaction system are evaluated from the literature. When no literature data are available, the values are estimated using ThermCal computer code which is developed in this study and based on THERM (78) computer code.

ThermCal can be used to calculate thermodynamic property data for gas phase radicals and molecules using Benson' Group Additivity method (79). All group contributions considered for a species are recorded and thermodynamic properties are generated in NASA polynomial format (for compatibility with CHEMKIN (80)) in addition to a listing which are more convenient for thermodynamic, kinetic and equilibrium calculation.

ThermCal is a "batch" process by contrast with the "interactive" THERM computer code. The advantages of ThermCal are use of much less memory, much less storage for equivalent documant file, and the batch process which makes it portable to other computer platforms. An updated NASA format compatible thermodynamic property file can be easily obtained by a recalculation of the input file by running ThermCal. This, for example, is required whenever the thermodynamic groups been updated. A thermodynamic data base up to C6 for the species with C/H/O elements is developed at NJIT and used for modeling the kinetic scheme of elementary reactions input to the program.

3.5.2 ThermCvt / ThermSrt (developed in this study)

ThermCvt is programmed to convert therm document file (*.doc) to ThermCal type input file. ThermCvt serves to avoid from human key-in error and save a great deal of time. ThermSrt is a sorting program to sort the ThermCal input file by the order specified in therm.cfg configuration file.

3.5.3 RadiCalc (used in this study)

RadiCalc (81) is a computer code that estimate the entropies and heat capacities of radical species and transition structures. The calculation is based upon the properties of these parent molecules. It calculates the effect of loss of vibrational modes (including inversion) and/or changes in barriers to internal rotation and moments of inertia. Databases are complied from literature data on generic vibrational frequencies, moments of inertia and changes in the barrier to rotation.

3.5.4 ThermFit (used in this study)

ThermFit (82, 83) is a computer code that determine geometric mean frequency as well as a three frequency set. It accepts input in the form of heat capacities versus temperature to 1000 K in addition to the number of vibrational modes and the number of internal rotors in the molecule. This code fits the heat capacity data in the above range to a five parameter harmonic oscillator model and extends the temperature range to 5000 K. An additional method of estimating the vibrational frequencies of a radical is to use MOPAC.

3.5.5 CHEMACT (modified in this study for fitting of Chebyshev Polynomials)

CHEMACT (84) is a computer code that uses the quantum version of RRK theory (QRRK) to evaluate k(E) and modified strong collision theory of Gilbert et al. for falloff as functions of temperature for various channels that can result in addition, combination and insertion reactions as stated in the previous section. Since the rate constants depend on both pressure and temperature of interest in the modeling calculation, a fitting optional of flexible a N × M Chebyshev polynomials expression as discussed previously is now available with user definable data grid. An output file that is CHEMKIN input compatible, is generated for pasting to the input file of modified CHEMKIN drivers for kinetic modeling.

3.5.6 CHEMKIN Interpreter and Library (modified in this study)

The CHEMKIN interpreter has been modified in this study to take the flexible matrix size of Chebyshev polynomials and to generate the appropriate link file for further processing. Subroutines involving kinetic rate coefficient calculation and array size

initialization are also modified to incorporate the Chebyshev polynomials expression. The advantage of modifying the CHEMKIN library instead of involving modification of drivers is any existing drivers can just link with the modified CHEMKIN library to take temperature and pressure dependent Chebyshev polynomials without modification. A couple of drivers have been linked with the new CHEMKIN library and tested. These include a shock tube driver, a constant temperature and pressure driver, and an adiabatic constant pressure driver.

CHAPTER 4

$CH_2OH + O_2$ AND $CH_2O + HO_2$ REACTION SYSTEMS

4.1 Introduction

The $CH_2OH + O_2$ reaction is important in the methanol oxidation reaction system. CH_2OH is formed from the abstraction of H from methanol by radicals. The reaction system is also important in evaluating kinetic parameters for O_2 reactions with alkyl hydroxyl radicals, that result from OH addition to olefins and aromatic compounds. The second reaction is responsible for creation of hydroxyl peroxy radicals which can further react to form CO_2 and acids.

Dr. Lev N. Krasnoperov initials the idea of hydrogen bonded complex for this reaction systems and helps me with the kinetic and thermodynamic calculations on this reaction systems. Without his help, this work can not be done.

Radford (85) measured a rate constant of $CH_2OH + O_2$ reaction of 1.2×10^{12} $cm^3\ mole^{-1}\ s^{-1}$ at 300 K and 0.5 Torr from relative HO_2 formation using laser magnetic resonance (LMR) to monitor HO_2 and by varying O_2 concentration for a fixed residence time. The rate constant measured is similar to a value of 8.4×10^{11} $cm^3\ mole^{-1}$ s^{-1} at 298 K and 1.5 Torr obtained by Wang et al. (86) who detected HO_2 from OH photofragment emission. Those two experiments are similar in that they use a discharge flow reactor, but the HO_2 measurement are different. Both of the rate constant obtained are lower than other measurements.

A rate constant of 5.7×10^{12} cm^3 $mole^{-1}$ s^{-1} at 298 K and 0.3 to 0.9 Torr is obtained by Grotheer et al. (87) from CH2OH profiles on a molecular beam sampling mass spectrometer. This reaction forms CH_2O as the major channel as concluded from simultaneous CH_2O measurements. The MS method was also used by Payne et al. (88) and a rate constant of 5.2×10^{12} cm^3 $mole^{-1}$ s^{-1} at 298 K and 1.0 Torr was obtained.

Dobe et al. (89) monitored the CH2OH decay by LMR in a isothermal flow system and obtained rate constant of 6.4×10^{12} cm^3 $mole^{-1}$ s^{-1} at 296 K and 0.52 to 4.88 Torr. No pressure dependent in the pressure range of above was observed.

Pagsberg et al. (90) detected the CH_2OH radical by UV absorption in a static reactor and reported a rate constant of 5.3×10^{12} cm^3 $mole^{-1}$ s^{-1} at 298 K and 760 Torr

A rate experiment using a discharge flow system at 215 to 300 K and 1.0 Torr was performed by Nesbitt et al. (91) Rate constant was determined by monitoring the decay of CH2OH using mass spectrometry. However, large intercepts in the k vs. O2 plots may indicate a wall problem.

Miyoshi et al. (92) reported a rate constant of 7.0×10^{12} cm^3 $mole^{-1}$ s^{-1} at 296 K and 1.3 to 5.5 Torr by laser flash photolysis-photoionization mass spectrometry.

For the reaction of $CH_2O + HO_2 \rightarrow H_2(OH)COO$ which is the reverse of $CH_2OH + O_2$, the forward and reverse rate constants are measured by Veyret et al. (93) using the technique of flash photolysis kinetic spectroscopy over a temperature range of 275 to 333 K and a pressure range of 85 to 170 Torr. Same experiment but at 298 K and 55 to 265 Torr was also performed by Veyret et al. (94)

Evleth et al. (95) characterized the $CH_2O + HO_2$ reaction at ab initio levels and reported the pathway via hydrogen-bonded complex. The experimental kinetic data can not be explained unless the hydrogen-bonded complex $CH_2O..HOO\bullet$ is incorporated into the potential energy diagram and reaction system.

4.2 Calculations and Input Parameters

Branching ratios at different temperatures and pressures are calculated using computer code "CHEMACT" discussed in previous chapter. A quantum version of RRK theory (QRRK) was used to evaluate energy dependent rate constants, k(E), for different channels. All input information required for QRRK calculations (set of frequencies or Cp(T)) was obtained from literature data as well as by Group Additivity.

The potential energy diagram is shown as Figure B.79 and the input parameters for QRRK calculations are summarized in Table 4.1 and 4.2. The potential energy diagram for the reaction system includes $H(OH)C=O + OH$, $H_2C=O + HO_2$, $H(OOH)C=O + H$, $H_2(OH)CO\bullet$ product channels with $H_2(OH)COO\bullet$, $C\bullet H(OOH)OH$, $CH_2(OOH)O\bullet$ and $H_2CO..HOO\bullet$ intermediates. This is based on existing experimental and theoretical data and on Group Additivity estimation techniques. Analysis indicates that the experimental kinetic data can not be explained unless the hydrogen-bonded complex $CH_2O..HOO\bullet$ is incorporated into the potential energy diagram and reaction system.

Table 4.1 QRRK Input Parameters for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Products

	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	E_a (kcal/mol)
1	$CH_2OH + O_2 \rightarrow H_2(OH)COO\bullet$	6.00E+12	0.0
-1	$H_2(OH)COO\bullet \rightarrow CH_2OH + O_2$	$2.93E+16\ T^{-0.54}e^{-0.00118T}$	34.4
2	$H_2(OH)COO\bullet \rightarrow H_2(OH)CO\bullet + O$	7.55E+14	57.5
3	$H_2(OH)COO\bullet \rightarrow H_2CO..HOO\bullet$	$4.96E+6\ T^{2.11}e^{-0.00069T}$	8.6
-3	$H_2CO..HOO\bullet \rightarrow H_2(OH)COO\bullet$	$3.76E+7\ T^{1.0}$	1.3
4	$H_2CO..HOO\bullet \rightarrow H_2C{=}O + HO_2$ (I)	$5.87E+17\ T^{-2.68}e^{-0.00007T}$	7.2
5	$H_2(OH)COO\bullet \rightarrow CH_2(OOH)O\bullet$	$6.86E+8\ T^{1.0}$	21.8
-5	$CH_2(OOH)O\bullet \rightarrow H_2(OH)COO\bullet$	$5.56E+8\ T^{0.84}e^{0.00042T}$	7.7
6	$CH_2(OOH)O\bullet \rightarrow H_2C{=}O + HO_2$ (II)	1.40E+14	5.0
7	$H_2(OH)COO\bullet \rightarrow C\bullet H(OOH)OH$	$5.90E+9\ T^{1.0}$	37.1
-7	$C\bullet H(OOH)OH \rightarrow H_2(OH)COO\bullet$	$4.19E+9\ T^{0.8}e^{0.00064T}$	31.6
8	$C\bullet H(OOH)OH \rightarrow H(OH)C{=}O + OH$	3.31E+13	1.0
9	$C\bullet H(OOH)OH \rightarrow H(OOH)C{=}O + H$	3.24E+13	22.5

frequencies/degenercies (from CPFIT): 415 cm^{-1}/6.178, 1506 cm^{-1}/5.998, 3198 cm^{-1}/2.824

Lennard-Jones parameters: σ = 4.83 Å, ϵ/k = 488 K

$\langle \Delta E \rangle_{down}$ = 800 cal/mol, Bath gas = N_2

k_1 A_1 taken as double of $CC\cdot + O_2$ addition from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 A_2 from MR with A_{-2} = 2.00E+13 based on addition of O to CH_3

k_3 $A_3 = (10^{13.55})(10^{\Delta S^{\#}/4.6})$, $\Delta S^{\#}$ = -6.4; $Ea_3 = \Delta H_3$ + 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_{-3} A_{-3} = (ekT/h)[exp($\Delta S^{\#}$/R)], $\Delta S^{\#}$ = -14.4; Ea_{-3} = 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_4 Based on the study of $CH_2O + HO_2$ system in this paper.

k_5 A_5 = (ekT/h)[exp($\Delta S^{\#}$/R)], ΔS = -8.6 (loss of 2 rotors); $Ea_5 = \Delta H_5$ + ring strain (4.7) + $E_{abstraction}$ (3.0), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-5} MR with the temperature range of 298 K and 2000 K.

k_6 A_6 from MR with A_{-6} = 6.05E+11 and $Ea_{-6} = \Delta H_{-6}$ + 5.0, based on addition of HO_2 to C_2H_4 by Tsang's recommendation, J. Phys. Chem. Ref. Data, 1987.

k_7 $A_7 = (ekT/h)[\exp(\triangle S^{\#}/R)](2)$, $\triangle S = -4.3$ (loss of 1 rotors); $Ea_7 = \triangle H_7$ + ring strain (23) + $E_{abstraction}$ (8.6), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-7} MR with the temperature range of 298 K and 2000 K.

k_8 A_8 from MR with A_{-8} = 2.7E+12, which is one half of the rate constant for addition of OH to C_2H_4, Atkinson *et al.*, J. Phys. Chem., 1989; Ea_8 from Soto and Page, Chem. Phys. 1991.

k_9 A_9 from MR with A_{-9} = 1.46E+13, which is one half of the rate constant for addition of H to C_2H_4 and $Ea_9 = \triangle H_9 + 2.7$, based on NIST fitting.

Table 4.2 QRRK Input Parameters for $H_2C=O + HO_2 \leftrightarrow [H_2CO..HOO\bullet]^* \rightarrow$ Products

	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	E_a (kcal/mol)
1	$H_2C=O + HO_2 \rightarrow H_2CO..HOO\bullet$	5.00E+10	0.0
-1	$H_2CO..HOO\bullet \rightarrow H_2C=O + HO_2$	$5.87E+17\ T^{-2.68}e^{-0.00007T}$	7.2
2	$H_2CO..HOO\bullet \rightarrow H_2(OH)COO\bullet$	$3.76E+7\ T^{1.0}$	1.3
-2	$H_2(OH)COO\bullet \rightarrow H_2CO..HOO\bullet$	$4.96E+6\ T^{2.11}e^{-0.00069T}$	8.6
3	$H_2(OH)COO\bullet \rightarrow CH_2OH + O_2$	9.00E+14	34.4
4	$H_2(OH)COO\bullet \rightarrow H_2(OH)CO\bullet + O$	7.55E+14	57.5
5	$H_2(OH)COO\bullet \rightarrow CH_2(OOH)O\bullet$	$6.86E+8\ T^{1.0}$	21.8
-5	$CH_2(OOH)O\bullet \rightarrow H_2(OH)COO\bullet$	$5.56E+8\ T^{0.84}e^{0.00042T}$	7.7
6	$CH_2(OOH)O\bullet \rightarrow CH_2O + HO_2$	1.40E+14	5.0
7	$H_2(OH)COO\bullet \rightarrow C\bullet H(OOH)OH$	$5.90E+9\ T^{1.0}$	37.1
-7	$C\bullet H(OOH)OH \rightarrow H_2(OH)COO\bullet$	$4.19E+9\ T^{0.8}e^{0.00064T}$	31.6
8	$C\bullet H(OOH)OH \rightarrow H(OH)C=O + OH$	3.31E+13	1.0
9	$C\bullet H(OOH)OH \rightarrow H(OOH)C=O + H$	3.24E+13	22.5

frequencies/degenercies (from CPFIT): 188 cm^{-1}/5.857, 1308 cm^{-1}/5.714, 2952 cm^{-1}/3.430

Lennard-Jones parameters: σ = 4.83 Å, ϵ/k = 488 K

$\langle \Delta E \rangle_{down}$ = 800 cal/mol, Bath gas = N_2

k_1 This study, estimated from the reaction of $CH_2O+HO_2 = CQ \cdot H_2OH$, Veyret *et al.*, J. Phys. Chem., 1989.

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 $A_2 = (ekT/h)[\exp(\Delta S^{\#}/R)]$, $\Delta S^{\#}$ = -14.4; Ea_2 = 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_{-2} MR with the temperature range of 298 K and 2000 K.

k_3 A_3 from MR with A_{-3} taken as double of $CC \cdot + O_2$ addition from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_4 A_4 from MR with A_{-4} = 2.00E+13 based on addition of O to CH_3.

k_5 $A_5 = (ekT/h)[\exp(\Delta S^{\#}/R)]$, ΔS = -8.6 (loss of 2 rotors); $Ea_5 = \Delta H_5$ + ring strain (4.7) + $E_{abstraction}$ (3.0), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-5} MR with the temperature range of 298 K and 2000 K.

k_6 A_6 from MR with A_{-4} = 6.05E+11 and $Ea_{-6} = \Delta H_{-4}$ + 5.0, based on addition of HO_2 to C_2H_4 by Tsang's recommendation, J. Phys. Chem. Ref. Data, 1987.

k_7 $A_7 = (ekT/h)[\exp(\triangle S^{\#}/R)](2)$, $\triangle S = -4.3$ (loss of 1 rotors); $Ea_7 = \triangle H_7$ + ring strain (23) + $E_{abstraction}$ (8.6), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-7} MR with the temperature range of 298 K and 2000 K.

k_8 A_8 from MR with A_{-8} = 2.7E+12, which is one half of the rate constant for addition of OH to C_2H_4, Atkinson *et al.*, J. Phys. Chem., 1989; Ea_8 from Soto and Page, Chem. Phys. 1991.

k_9 A_9 from MR with A_{-9} = 1.46E+13, which is one half of the rate constant for addition of H to C_2H_4 and $Ea_7 = \triangle H_7 + 2.7$, based on NIST fitting.

The input parameters required for QRRK analysis are discussed in previous chapter except the estimation of high pressure rate constants. Elimination or dissociation rate constants are calculated thermochemically from the reverse (e.g. addition, combination) reactions and microscopic reversibility if literature data is not available. The addition or combination rate constants are taken from literature or estimated from the trends in homologous series of this type of reactions or analogous reaction. Isomerization rate constants are analyzed via Transition-State-Theory (TST) and thermochemical kinetic methods of Benson. The high pressure A factor is calculated as $A = (ekT/h)[\exp(\triangle S^{\#}/R)]$, where $\triangle S^{\#}$ is the entropy difference from reactant to transition state. The activation energy is calculated as the sum of abstraction barrier, ring strain, plus enthalpy of isomerization where the reaction is endothermic.

4.3 Results and Discussion

The results of $CH_2OH + O_2$ calculations are displayed in Figure B.80 to B.84 and the comparison with experimental data is in Figure B.85. There are two pathways to form the $CH_2O + HO_2$ products. One is via $H_2C(OO\bullet)OH$ isomerization to $CH_2(OOH)O\bullet$, which then decomposes to $CH_2O + HO_2$. The rate constant via this pathway is about two orders of magnitude smaller than the pathway via hydrogen-bonded complex and reported experimental data at low pressure and temperature. The dominance of hydrogen-bonded channel is why the experimental kinetic data can not be explained unless the hydrogen-bonded complex $CH_2O..HOO\bullet$ is incorporated into the potential energy diagram and reaction system. The formation of $CH_2O + HO_2$ adduct is the

dominate channel for pressures < 1 atm and temperature < 2000 K. The stabilization adduct of $H_2C(OO\bullet)OH$ formation is important for pressures > 3 atm at 298 K and for p > 25 atm at 900 K. The Acid formation adduct is important for T > 2000 K.

Figure B.86 to B.89 show the results of $CH_2O + HO_2$ reaction. A comparison of $CH_2O + HO_2 \leftrightarrow H_2(OH)COO\bullet$ with forward and reverse experimental data is plotted in Figure B.90 which shows good agreement for both directions. Adduct formation is important for pressures above 0.01 atm at 298 and p above 0.15 atm at 900 K. Here, formation of $CH_2OH + O_2$ and $HCO_2H + H$ surpasses stabilization in importance at 0.005 atm and 900 K, and increase in importance with increasing temperature.

Results of this study confirm the importance of hydrogen bonded complex for this reaction system and suggest the possible importance of such a complexes in other, similar systems. The reaction description is based entirely on "complex formation" mechanism without "direct reaction". Results of these calculations, which are based on limited, low-pressure and narrow temperature range experimental data, can be used in combustion models over a wide temperature and pressure range.

85 Radford, H.E. *Chem. Phys. Lett.* **1980**, 71, 195.

86 Wang, W. C.; Suto, M.; Lee, L. C. *J. Chem. Phys.* **1984**, 81, 3122.

87 Grotheer, H. H.; Riekert, G.; Meier, U.; Just, Th. *Ber. Bunsen-Ges. Phys. Chem.* **1985**, 89, 187.

88 Payne, W. A.; Bruuing, J.; Mitchell, M. B.; Stief, L. J. *Int. J. Chem. Kinet.* **1988**, 20, 63.

89 Dobe, J. E.; Warnatz, J. *Ber. Bunsen-Ges Phys. Chem.* **1983**, 87, 1040.

90 Pagsberg, P.; Munk, J. *J. Phys. Chem.* **1989**, 93, 5162.

91 Nesbitt, F. L.; Payne, W. A.; Stief, L. J. *J. Phys. Chem.* **1988**, 92, 4030.

92 Miyoshi, A.; Matsui, H.; Washida, N. *J. Phys. Chem.* **1990**, 94, 3016.

93 Veyret, B.; Lesclaux, R.; Rayez, M.; Rayez, J. *J. Phys. Chem.* **1989**, 93, 2368.

94 Veyret, B.; Rayez, J.; Lesclaux, R. *J. Phys. Chem.* **1982**, 86, 3424.

95 Evleth, E. M.; Melius, C. F.; Rayez, J. C.; Rayez, W.; Forst, W. *J. Phys. Chem.* **1993**, 97, 5040.

CHAPTER 5

METHANOL PYROLYSIS AND OXIDATION

5.1 Methanol Pyrolysis

5.1.1 Experimental Results

Methanol pyrolysis is studied over a pressure range from 1 to 10 atm at 1073 K under isothermal reactor conditions. The reactant concentrations at different pressures are listed in Table 5.1.

Table 5.1 Experimental conditions of methanol pyrolysis

Pressure	Molar Fraction	
	Methanol	Argon
1 atm	0.03950	0.96050
3 atm	0.01317	0.98683
5 atm	0.00790	0.99210
8 atm	0.00494	0.99506
8 atm	0.00250	0.99750
10 atm	0.00395	0.99605

Experimental results on methanol pyrolysis at 1073 K and a pressure range from 1 to 10 atm are displayed in Appendix B, Figure B-7 to B-12. Methanol decay curves at different pressures are displayed as normalized concentrations (C/Co) versus reaction time in Figure B.91.

Methanol decays and forms the intermediate stable products: formaldehyde which then decomposes to carbon monoxide. The decay rate of methanol increases with increasing pressure, however, the differences between 5, 8 and 10 atmospheres are

less than 5%. Normalized concentrations (C/Co) of methane formation are about the same for different pressures.

5.1.2 Kinetic Modeling

The initial decomposition steps of methanol below are analyzed by QRRK formalism for k(E) and modified strong collision theory of Gilbert et al. for falloff. The potential energy diagram is shown in Figure B.92 and the input parameters for QRRK analysis are in Appendix C. The barriers of $HCOH + H_2$, $^1CH_2 + H_2O$ and $CH_2O + H_2$ channels are taken from the study of Walch (96) using complete-active-space self-consistent-field (CASSCF) / internally contracted configuration-interaction (CCI), ab initio calculations. The reverse of $^1CH_2 + H_2O$ channel is found to have no barrier. The barriers of the reverse of $HCOH + H_2$ and $CH_2O + H_2$ channels are calculated as -5.2 and 1.7 kcal/mol with respect to the energy level of $CH_3 + OH$ channel. The barriers of $HCOH + H_2$, $^1CH_2 + H_2O$ and $CH_2O + H_2$ channels are then can be derived as 87.4, 90.9 and 94.3 kcal/mol respectively. The unimolecular dissociations show that $CH_3 + OH$ channel is dominant at moderate temperature (i.e. about 1000 K) and $HCOH + H_2$ channel becomes important when temperature increases.

CH_3OH	$\rightarrow$	$CH_3 + OH$	$\Delta H_{rxn} = 92$ kcal/mol
	$\rightarrow$	$HCOH + H_2$	$\Delta H_{rxn} = 71$ kcal/mol
	$\rightarrow$	$^1CH_2 + H_2O$	$\Delta H_{rxn} = 92$ kcal/mol
	$\rightarrow$	$CH_2O + H_2$	$\Delta H_{rxn} = 74$ kcal/mol
	$\rightarrow$	$CH_2OH + H$	$\Delta H_{rxn} = 97$ kcal/mol
	$\rightarrow$	$CH_3O + H$	$\Delta H_{rxn} = 104$ kcal/mol

Abstraction reactions: two H atom abstraction reactions are found to be the most significant after the initial decomposition of methanol to form CH_2OH and CH_3O radicals.

$$CH_3OH + H \rightarrow CH_2OH + H_2$$

$$CH_3OH + H \rightarrow CH_3O + H_2$$

There are no well accepted rate coefficients for CH_3OH + H abstractions available. Warantz's data are adopted in Norton's mechanism and this study. The ratio for forming CH_2OH and CH_3O is 4 to 1. The CH_2OH and CH_3O radicals decompose beta scission to form formaldehyde:

$$CH_2OH + M \rightarrow CH_2O + H + M$$

$$CH_3O + M \rightarrow CH_2O + H + M.$$

These two unimolecular decomposition reactions are pressure dependent and are analyzed by QRRK analysis.

Formaldehyde, important intermediate decays mostly by H abstraction reaction : $CH_2O + H \rightarrow HCO + H_2$.

The decomposition reaction of $HCO + M \rightarrow CO + H + M$ is found to be the dominant channel to form the pyrolysis final product of carbon monoxide.

Comparisons of modeling and experimental results at 1 and 3 atm are illustrated in Figure B.93 and B.94 and show good agreement for both reactant decay and main product formations at 1 atm. At 3 atm, the modeling result is a little faster than experimental data. Methane formations are under predicted at both conditions. The model predicts even faster decay of methanol than experimental at higher pressure.

5.2 Methanol Oxidation

5.2.1 Experimental Results

Methanol oxidation experiments are performed with two reactant ratio sets which represent stoichiometric and fuel lean conditions over temperature range of 873 to 1073 K and a pressure range of 1 to 5 atm. The reactant concentration ratios are:

Fuel Lean $\phi = 0.75$ $CH_3OH : O_2 : AR = 0.0078 : 0.0156 : 0.9766$

Stoichiometric $\phi = 1.0$ $CH_3OH : O_2 : AR = 0.0078 : 0.0117 : 0.9805$.

Results of methanol oxidation are shown in Figures B-13 to B-60, which plot the mole fraction as a function of reaction time and temperature, at conditions ranging from of 873 to 1073 K, pressures of 1, 3 and 5 atm and two equivalence ratios of 1.0 and 0.75.

Changes in methanol conversion, intermediates and products profiles under conditions of excess oxygen (equivalence ratio = 0.75) are small relative to reaction at stoichiometric conditions for all temperatures and pressures studied. Methanol starts to decay at 923 K and 100% conversion is observed at 1073 K, at 1 atm and reaction time of 0.15 second. For 3 atm and reaction time of 0.6 second, methanol decay starts at 873 K and is 95% converted at 923 K. For 5 atm and reaction time of 0.6 second, methanol conversion is 5% at 873 K and 99% at 923 K.

At increased temperatures, maximum formaldehyde levels decrease and this intermediate only exists for a short residence time (ca. 0.2 seconds). CO increases rapidly with the conversion of methanol. The formation of CO_2 starts to occur when

significant amounts of CO produced. At higher temperature, oxidation of CO occurs rapidly to the final product CO_2.

Methane is observed as a minor product. Impurity, methane, has been found in the supply gas at a level of few ppm. The experimental result of methane is calibrated by the blank (by-pass) correction. Due to the low concentration of methane, hydrocarbon molecular weight growth species are not detected (probably below few ppm, instrument detection limit).

5.2.2 Kinetic Modeling

After methanol decomposition occurs, the following reactions play important roles for the methanol decay in the oxidation condition and intermediate temperature. In contrast to pyrolysis where H atom is important, $CH_3OH + OH$ is now the most responsible reaction after unimolecular decomposition for the methanol decay.

$$CH_3OH + OH \rightarrow CH_2OH / CH_3O + H_2O_2$$

$$CH_3OH + HO_2 \rightarrow CH_2OH / CH_3O + H_2O$$

$$CH_3OH + H \rightarrow CH_2OH / CH_3O + H_2$$

The reaction of CH_2OH radical with O_2 to form formaldehyde and hydroperoxy radical is the most important reaction for the production of important intermediate, formaldehyde.

$$CH_2OH + O_2 \rightarrow [.OOCH_2OH]\# \rightarrow .OCH_2OH + O$$

$$\rightarrow [HOOC.HOH]\# \rightarrow HCO_2 + OH$$

$$\rightarrow [CH_2O...HOO]\# \rightarrow CH_2O + HO_2$$

$$\rightarrow [HOOCH_2O.]\# \rightarrow CH_2O + HO_2$$

$\rightarrow CH_2OH + O_2$ (Reverse - No Rxn)

$\rightarrow .OOCH_2OH$ (Stabilization)

A more detailed discussion of the $CH_2OH + O_2$ reaction is described in Chapter 4. There are two pathways to form $CH_2O + HO_2$. One through the isomerization of $[.OOCH_2OH]^{\#}$ to $[HOOCH_2O.]^{\#}$ then forms $CH_2O + HO_2$ when another through the isomerization of $.OOCH_2OH$ to $[CH_2O...HOO]^{\#}$, hydrogen-bonded complex. The experimental kinetic data can not be explained through the first route unless the hydrogen-bonded complex $CH_2O..HOO\bullet$ is characterized and incorporated into the potential energy diagram and reaction scheme of Evleth et al.

Hydroperoxy radical reacts with hydrogen atom to form hydroxyl radical. This is found to be an important source of hydroxyl radical for the $CH_3OH + OH$ abstraction reaction.

Formaldehyde then decays mostly by the reaction of $CH_2O + OH \rightarrow HCO + H_2O$. The decomposition reaction of $HCO + M \rightarrow CO + H + M$ is found to be the dominant channel to form carbon monoxide and $CO + O + M \rightarrow CO_2 + M$ to form final product of carbon dioxide.

Comparisons of the model and the experimental results at 5 atm and two equivalence ratios: stoichimatric ($\phi = 1$) and fuel lean ($\phi = 0.75$), are in Figures B.95 and B.96. These show reasonable agreement for reactant decay and main product formation. . Reaction changes in methanol conversion and the intermediates or products profiles under conditions of excess oxygen (equivalence ratio = 0.75) are small relative to those illustrated at stoichiometric conditions throughout all

temperatures and pressures studied, in both the experimental data and model predictions.

Figures B.97 to B.99 shows the comparisons of modeling and experimental results for reactant conversion and product distribution versus temperature at 5 atm, average residence times of 0.8, 1.0 and 1.5 seconds respectively. The reactant decay and major product distributions are in reasonable agreement but the model predicts faster CO oxidation at higher temperature (ca. > 1000 K). The methane impurity is possibly why the methane been observed even at more than 99% of methanol been convert to CO_2.

At 3 atmospherics pressure, modeling result is a little slower than the observed experimental data. Figure B.100 shows a time delay of 0.15 second for modeling result relative to experimental data. In general, there is a longer time delay for the model compared with experimental data at lower pressures. At lower pressure and temperature, decay will occur at longer residence times which are unfortunately, out of the flow range of the flow controller used in this study.

5.3 Summary

The pyrolysis and oxidation of methanol follow the decomposition order of $CH_3OH \rightarrow CH_2O \rightarrow CO \rightarrow CO_2$. CO_2 is observed in the oxidation studies only. The important reactions can be summarized as below.

CHAPTER 6

EFFECTS OF METHANOL ADDITION TO METHANE OXIDATION

6.1 Experimental Results

Experiments and model calculations were performed to determine the effects of methanol on methane oxidation. The total mole fraction of CH_4 + CH_3OH was hold constant at 1.56%. Experiments on methane oxidation for pure methane and for mixture of CH_4 / CH_3OH at 100% increments of CH_3OH up to equal molar were conducted. The results show that addition of CH_3OH to CH_4 oxidation at a near constant fuel equivalence ratio, dramatically increased CH_4 conversions.

Experimental results on oxidation of methane / methanol mixtures are shown as Figures B-61 to B-72. The presence of a small quantity of methanol increases the rate of CH_3OH decay. Methanol is observed to have a significant acceleration effect on methane oxidation. The experimental results are obvious and a uniform trend is observed over the temperature, pressure and relative concentration range of the methane / methanol mixtures.

The ratio of methane / methanol is varied with a constant total mixture concentration and equivalence ratio to understand and verify the methanol acceleration effect on methane oxidation. At equal molar mixtures conversion of methane is enhanced by 50% at 1.5 seconds reaction time and further enhancement is slight.

Molecular weight growth species: ethane and ethylene are also observed which result from CH_3 + CH_3 recombination reactions.

6.1 QRRK analysis of $CH_3 + O_2$ reactions

CH_3 is the initial radical intermediate in CH_4 oxidation. The major reaction of CH_3 at low CH_3 concentration is reaction with O_2. A potential energy level diagram for CH_3 reaction with O_2 is shown in Figure B.101 and a reaction scheme is as below.

$$\begin{array}{ccccccc} & & CH_3O+O & & & & \\ & & \uparrow & & & & \\ CH_3+O_2 & \Leftrightarrow & [CH_3OO]^* & \Leftrightarrow & [CH_2OOH]^* & \rightarrow & CH_2O+OH \\ & & \downarrow & & \downarrow & & \\ & & CH_3OO & & CH_2OOH & & \end{array}$$

CH_3 reaction with O_2 forms $[CH_3OO]^*$ complex. The $[CH_3OO]^*$ complex can either stabilize, form $CH_3O + O$, isomize to $[CH_2OOH]^*$ or go back to reactants. The $[CH_2OOH]^*$ isomer can also isomerize back to $[CH_3OO]^*$, stabilize or form the products: $CH_2O + OH$.

High-pressure-limit input parameters for the $CH_3 + O_2$ combination to form the CH_3OO complex, are taken from Cobos et al. (97) The input parameters for dissociation of the complex back to reactants are calculated by microscopic reversibility with the temperature range 298 to 2000 K. Parameters for the $CH_3O + O$ product channel are obtained from an estimate of 5.0×10^{13} $cm^3mol^{-1}s^{-1}$ for the high-pressure recombination rate constant via microscopic reversibility. The A factor of isomerization is taken as Transition State Theory loss of one rotor ($\Delta S^{\ddagger}$ = -4.3 cal/mol) and the degeneracy of 3. The activation energy is estimated as a sum of reaction enthalpy, ring strain (26 kcal/mol) and H abstraction (6 kcal/mol). The A factor of exit channel, $CH_2O + OH$, is calculated from microscopic reversibility with reverse taken as

the addition of OH to CH_3CHO by Semmes et al. (98) with Ea is estimated as 2 kcal/mol from the intrinsic activation energy expected for OH addition.

Comparisons of experimental and QRRK results for $CH_3 + O_2 \rightarrow CH_3OO$ in argon and nitrogen bath gas are displayed in Figure B.102 and $CH_3O + O$ and $CH_2O +$ OH channels are in Figure B.103. Results for $CH_3 + O_2$ to all channels at atmospheric pressure are displayed in Figure B.104.

Here one can see a dramatic difference from the $CH_2OH + O_2$ reaction system; there is no low energy exit channel such as the $CH_2O + HO_2$ product set in $CH_3 + O_2$.

6.2 Comparison of Methane oxidation versus methanol oxidation

The comparison of methane and methanol oxidation can be described in three stages:

6.2.1 Initiation

Several channels of CH_3OH unimolecular decomposition are much lower in energy than CH_4 and therefore faster than CH_4 decomposition.

CH_3OH	$\rightarrow$	$CH_3 + OH$	$\Delta H_{rxn} = 92$ kcal/mol
	$\rightarrow$	$HCOH + H_2$	$\Delta H_{rxn} = 71$ kcal/mol
	$\rightarrow$	$^1CH_2 + H_2O$	$\Delta H_{rxn} = 92$ kcal/mol
	$\rightarrow$	$CH_2O + H_2$	$\Delta H_{rxn} = 74$ kcal/mol
CH_4	$\rightarrow$	$CH_3 + H$	$\Delta H_{rxn} = 103$ kcal/mol

6.2.2 Propagation

$H–CH_2OH$ has a lower bond energy (97 kcal/mol) than $H–CH_3$ (105 kcal/mol). CH_2OH forms easier than CH_3 (from CH_4) by decomposition as well as through

abstraction reactions. CH_2OH also has a much more rapid unimolecular decomposition channel relative CH_3.

$$CH_3 \rightarrow CH_2 + H \qquad \Delta H_{rxn} = 108 \text{ kcal/mol}$$

$$CH_2OH \rightarrow CH_2O + H \qquad \Delta H_{rxn} = 30 \text{ kcal/mol}$$

Therefore, propagation reactions of CH_3OH oxidation are faster than CH_4 oxidation too.

6.2.3 Oxidation

Methane oxidation, $CH_3 + O_2$ which will be discussed in next section is much slower than that of $CH_2OH + O_2$. The difference between $CH_2OH + O_2$ and $CH_3 + O_2$ reaction system is that $CH_2OH + O_2$ has a low energy exit channel. Barrier is below the energy level of initial $CH_2OH + O_2$ reactants to products, $CH_2O + HO_2$. This explains the faster $CH_2OH + O_2$ reaction.

6.3 Acceleration of Methane oxidationby Added Methanol

Comparisons of modeling and experimental results for oxidation of methane /methanol mixtures with equal initial concentration at 873 K and 5 atm in stoichiometric reaction condition are shown in Figure B.105 and B.106. Methane conversion is more than 50% at 1.5 seconds, when there is almost no conversion at all, without methanol under the same conditions.

Methanol decomposes faster than methane as stated in previous section. The important intermediate, CH_2OH, then reacts with O_2 and undergoes unimolecular decay to accelerate the radical pool production. The formation of OH and HO_2 radicals accelerated and the OH initiates methane abstraction reactions.

Figure B.107 shows the methanol acceleration effect on methane oxidation at varied ratios of methane/methanol (100/0, 90/10, 85/15, 75/25, 50/50) but constant total initial mixture concentration and equivalence ratio for both the experimental and modeling results. The more methanol added, the faster is methane conversion. 50% methane conversion is reached at about 1.5 seconds reaction time with equal initial concentration of methane and methanol, when there is no decay of methane with no methanol added.

Figure B.108 shows the effect of methane added to methanol oxidation at 873 K and 5 atm in stoichiometic (overall) condition is small. Comparisons of methanol decay for 0.78% methanol (neat methanol) and a mixture of 0.39% methane and 0.39% methanol, initial concentration, shows very small difference in conversion results. Comparing the 0.39% and 0.78% methanol oxidation in Figures B.108 and B.109, we observe effectively the same CH3OH conversion. There is very little effect of adding methane to methanol oxidation.

CHAPTER 7

MTBE (METHYL TERT-BUTYL ETHER) OXIDATION

7.1 Pyrolysis of MTBE

7.1.1 Experimental Results

MTBE pyrolysis experiments have been performed in collaboration with Chiung-Ju Chen (NJIT) at atmospheric pressure and over a temperature range of 873 and 948 K with a 0.5% constant initial concentration using argon as bath gas. The pyrolysis of MTBE is dominated by the unimolecular decomposition of MTBE at low to intermediate temperatures of this study. That this $C_3COC \rightarrow C_2C{=}C + CH_3OH$ reaction dominates, is verified in this study by a combination of QRRK analysis and mechanism sensitivity analysis.

The experimental data of MTBE decay is plotted as -ln (C/C_o) versus reaction time in Figure B.110, in order to get the overall reaction rate constant of MTBE pyrolysis. The regression results are in Table 7.1.

Table 7.1 Regression results of overall rate constant for MTBE pyrolysis experimental data

Temperature / K	$k_{MTBE\ pyrolysis}$ / s^{-1}
873	0.167
898	0.493
923	1.331
948	2.936

7.1.2 Unimolecular Dissociation of C_3COC

The initiation reactions are unimolecular decompositions of MTBE :

C_3COC	$\xrightarrow{k_1}$	$C_2C{=}C + CH_3OH$	$\Delta H_{rxn} = 19$ kcal/mol
	$\xrightarrow{k_2}$	$C_3C\cdot + CH_3O$	$\Delta H_{rxn} = 86$ kcal/mol
	$\xrightarrow{k_3}$	$C_3CO\cdot + CH_3$	$\Delta H_{rxn} = 82$ kcal/mol
	$\xrightarrow{k_4}$	$C_2C\cdot OC + CH_3$	$\Delta H_{rxn} = 89$ kcal/mol
	$\xrightarrow{k_6}$	$C_3COC\cdot + H$	$\Delta H_{rxn} = 97$ kcal/mol
	$\xrightarrow{k_5}$	$C_3\cdot COC + H$	$\Delta H_{rxn} = 101$ kcal/mol

The MTBE dissociation reaction is the important reaction in the both of the MTBE pyrolysis and oxidation reaction systems. The energy level diagram and input parameters for the chemical activation calculations are shown in Figure B.111 and Appendix C.

The channels involving loss of a H atom are of no importance due to the higher energy barrier of breaking the C--H bond compared with C--O and C--C bonds. The relative A factors for C--H bond cleavage are also lower, further reducing the relative probability for these reactions. The calculation results indicate that the dominant channels are the dissociations to $C_2C{=}C + CH_3OH$ and $C_3C\cdot + CH_3O$. The apparent rate constants of both reactions are consistant with their high pressure limits when the temperature is below than 800 K and the pressure higher than 0.001 atm.

The unimolecular elimination, channel (1), is dominant when the temperature is lower than ~900 K. This is consistent with Daly et al. (1) and Choo et al. (2) conclusion's that the decomposition reaction of MTBE is a four-center molecular elimination at temperatures ~700 K.

The effect of temperature on the falloff behavior of dissociation reactions is dramatic. Increasing the temperature from 800 to 1200 K for $C_3COC \rightarrow C_3C\cdot + CH_3O$ shifts the falloff curve by 4 orders of magnitude toward higher pressure. The result indicates the dramatic differences between application of the present approach and use of the assumption that the dissociation is always at the high pressure limit. The difference between the actual dissociation rate constant and the high pressure limit increases when the temperature is above 1000 K at the atmospheric pressure.

The apparent rate constant of $C_3COC \rightarrow C_2C{=}C + CH_3OH$ reaction is not in the high pressure limit regime under temperature range from 873 to 948 K at 1 atm. Therefore, a fitting process is required to get the apparent rate constants and the input high pressure limit parameters for the conditions of interested. The results of this optimization are shown below:

$$C_3COC \rightarrow C_2C{=}C + CH_3OH \qquad k_{app} = 1.55\times10^{15}\, T^{-0.287} \exp(-60186 / RT)$$

$$k_{\infty} = 2.2\times10^{14} \exp(-60{,}000 / RT).$$

Note that the apparent rate constants are fitted between temperatures of 873 and 973 K at atmospheric pressure.

7.1.3 Comparison of Modeling and Experimental Results

A comparison of modeling results and experimental data is displayed in Figure B.112 for validation. Good agreement between the model and experimental results is shown for the pyrolysis of MTBE in different temperatures.

7.2 Oxidation of MTBE

7.2.1 Kinetic Modeling

The MTBE dissociation reaction is the initial reaction in the MTBE pyrolysis and also in this oxidation reaction system and discussed in detail in previous section. Initial abstraction reactions of MTBE are mainly by radical species: H, O, OH, HO2, CH3O. and C3C.

$$C_3COC + H \rightarrow C_3COC\cdot + H_2$$

$$\rightarrow C_3\cdot COC + H_2$$

$$C_3COC + O \rightarrow C_3COC\cdot + OH$$

$$\rightarrow C_3\cdot COC + OH$$

$$C_3COC + OH \rightarrow C_3COC\cdot + H_2O$$

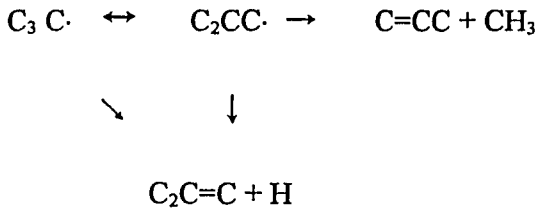
$$\rightarrow C_3\cdot COC + H_2O$$

$$C_3COC + HO_2 \rightarrow C_3COC\cdot + H_2O_2$$

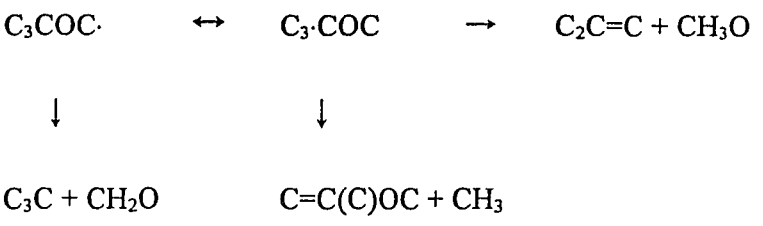
$$\rightarrow C_3\cdot COC + H_2O_2$$

The elementary reaction rate parameters for abstraction reactions are based upon literature survey, thermodynamics and generic A factors with Evans Polanyi plots for Ea's.

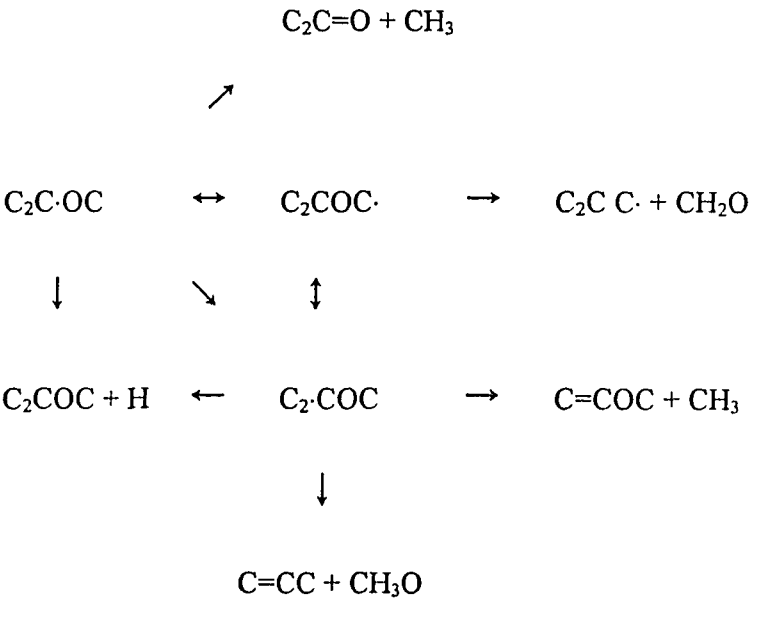
The thermal decompositions (Beta scissions) of $C_3COC\cdot$, $C_3\cdot COC$, $C_3C\cdot$, and $C_2C\cdot OC$ radicals and abstractions by other radicals are the important reactions for the MTBE oxidation system. These radical decomposition rate constants are determined from the QRRK calculation as described in the previous section.



$C_3COC\cdot$ and $C_3\cdot COC$ decomposition reactions are also important in MTBE reaction system.



The corresponding radicals of iso-propyl methyl ether, second radical, are also the important species in MTBE oxidation system. The following reactions contribute to the formation of acetone and propene.



CH_3 and CH_3O abstract other species to form CH_4 and CH_3OH or recombine with O, H, and OH. CH_3 also reacts with O_2 to CH_3OO and $CH_2{\cdot}OOH$. Higher temperatures or oxygen rich conditions accelerate radical generation reactions and the formation of CO and CO_2.

7.2.2 QRRK Analysis

A QRRK analysis of the chemically activated system, using generic estimates or literature values for high pressure rate constants and species thermodynamic properties for the enthalpies of reaction, yields thermodynamically and kinetically plausible apparent rate constants. The input rate parameters used in these calculations and results from the calculation are summarized in Appendix C.

Unimolecular Dissociation of $C_3C{\cdot}$

The potential energy diagram and input parameters are shown in Appendix C.

$$C_3C{\cdot} \xrightarrow{k_1} C_2C{=}C + H \qquad \Delta H_{rxn} = 97 \text{ kcal/mol}$$

$$\xleftarrow{k_2} C_2CC{\cdot} \xrightarrow{k_3} C{=}CC + CH_3$$

$$\Delta H_{rxn}=97\text{kcal/mol}$$

There are two channels for $C_3C{\cdot}$ decomposition breaking the C--H bond to form iso-butene and intramolecular isomerization (H atom shift) through a 3 member cyclic intermediate, to $C_2C{=}C{\cdot}$ Reaction (1) is about ten times faster than reaction (2) for pressures above 1.0 atm, and more than ten times faster for pressures below 1.0 atm. That is due to high energy barrier and tight transition state (low A factor) to isomerization, although reaction (2) goes to a lower final energy level.

Unimolecular Dissociation of C_3·COC

The potential energy diagram and input parameters are shown in Appendix C.

C_3·COC	$\xrightarrow{k_1}$	C_3COC·	ΔH_{rxn} = 97 kcal/mol
	$\xrightarrow{k_2}$	C=C(C)OC + CH_3	ΔH_{rxn} = 97 kcal/mol
	$\xrightarrow{k_3}$	C_3C=C + CH_3O	ΔH_{rxn} = 97 kcal/mol

The rate constants of these reactions are below their high pressure limit at atmospheric pressure and 1000 K. Reaction (3) of C_3·COC dissociation to C=C(C)OC (iso-butene) and CH_3 is the dominant channel. Reaction (1) which undergoes a intramolecular isomerization, 5 member cyclic intermediate, to C_3COC·, is about ten times slower than reaction (3) due to the high energy barrier, although it goes to a lower final energy level. Reaction (2) is also limited by a lower A factor than Reaction (3) even though it is slightly more thermodynamically favorable.

Unimolecular Dissociation of C_3COC·

The potential energy diagram and input parameters are shown in Appendis C.

C_3COC·	$\xrightarrow{k_1}$	C_3C· + CH_2O	ΔH_{rxn} = 97 kcal/mol
	$\xrightarrow{k_2}$	C_3·COC	ΔH_{rxn} = 97 kcal/mol

The rate constants of these reactions are below their high pressure limit at atmospheric pressure and 1000 K. Channel (1) is dominant. Channel (2) which undergoes a intramolecular isomerization, 5 member cyclic intermediate, to C_3·COC, does not occur due to the high energy barrier.

Unimolecular Dissociation of $C_2C{\cdot}OC$

The potential energy diagram and input parameters are shown in Appendix C.

$C_2C{\cdot}OC$	$\xrightarrow{k_1}$	$C_2C{=}O + CH_3$	$\Delta H_{rxn} = 97$ kcal/mol
	$\xrightarrow{k_2}$	$C{=}C(C)OC + H$	$\Delta H_{rxn} = 97$ kcal/mol
	$\xrightarrow{k_3}$	$C_2COC{\cdot}$	$\Delta H_{rxn} = 97$ kcal/mol
	$\xrightarrow{k_4}$	$C_2{\cdot}COC$	$\Delta H_{rxn} = 97$ kcal/mol

The rate constants of these reactions are below their high pressure limit at atmospheric pressure and 1000 K. Reaction (1) is dominant. Reaction (2), (3), and (4) are not important due to the higher energy barriers.

Unimolecular Dissociation of $C_2COC{\cdot}$

The potential energy diagram and input parameters are shown in Appendix C.

$C_2COC{\cdot}$	$\xrightarrow{k_1}$	$C_2C{\cdot}OC$
	$\xrightarrow{k_2}$	$C_2C{\cdot} + CH_2O$
	$\xrightarrow{k_3}$	$C_2{\cdot}COC$

The rate constants of these reactions are below their high pressure limit at atmospheric pressure and 1000 K. Reaction (2) is dominant. Reaction (1) and (3) which undergo the intramolecular isomerizations, 4 and 5 member cyclic intermediate respectively, to $C_2C{\cdot}OC$ and $C_2{\cdot}COC$, do not occur to a significant degree due to the high energy barriers, although they go to a lower final energy levels.

Unimolecular Dissociation of $C_2{\cdot}COC$

The potential energy diagram and input parameters are shown in Appendix C.

$C_2{\cdot}COC \xrightarrow{k_1} C_2COC{\cdot} \rightarrow C_2C{\cdot} + CH_2O$ $\Delta H_{rxn} = 97$ kcal/mol

$\xrightarrow{k_2} C{=}COC + CH_3$ $\Delta H_{rxn} = 97$ kcal/mol

$\xrightarrow{k_3} C{=}C(C)OC + H$ $\Delta H_{rxn} = 97$ kcal/mol

$\xrightarrow{k_4} C{=}CC + CH_3O$ $\Delta H_{rxn} = 97$ kcal/mol

$\xrightarrow{k_5} C_2C{\cdot}OC$ $\Delta H_{rxn} = 97$ kcal/mol

The rate constants of these reactions are below their high pressure limit at atmospheric pressure and 1000 K. The rate constants for all channels are close to their high pressure limit when the pressure reaches 10 atm. Reaction (1) and (2) are dominant.

7.2.3 Mechanism Validation

A detailed reaction kinetic mechanism which is listed in Table A.4, was developed to describe the MTBE oxidation reaction system of reactions studied based on methanol oxidation reaction system studied in previous chapters. Elementary reaction rate parameters for abstraction reactions are based upon evaluated literature comparision, thermodynamics, Transition State Theory determination of Arrhenius A factor and energies of activation when literature data was not available. QRRK calculation, as described in previous section, were used to estimate apparent rate parameters for dissociation and combination reactions reported for a temperature range of 300 to 2500 K and at pressure of 0.001 to 50 atm for N_2 as bath gas.

Model predictions and experimental data of Norton et al. are shown in Fig B.113 which shows reasonable agreement between calculated and experimental results for MTBE decay and methanol formation. An offset of 10 msec has been taken for the model result to compensate the reagent injection and mixing issues with their experimental setup. Model result of Iso-butene formation is consistent with experimental dat in the initial stage, but over predicted for propagation which is probably due to the complexity of C4 oxidation. Even the mechanism of 448 reactions is not complete enough for the iso-butene oxidation. For the decomposition order of C4 $\rightarrow$ C3 $\rightarrow$ C2 $\rightarrow$ C1, C3 and C2 are therefore under predicted.

CHAPTER 8

CONCLUSIONS

This study presents experimental data on the decomposition of methanol in several different reaction environments - fuel lean to stoichiometric at a temperature range of 873 and 1073 K and a pressure range of 1 and 5 atm. Methane fuel is also added in several of the systems studied in order to provide experimental data to understand the methanol addition effect on the methane oxidation.

Computer codes: ThermCal, ThermSrt and ThermCvt have been developed for the thermal property calculations of stable molecules by the Benson group additivity method and of radicals by the NJIT hydrogen bond increment method.

Pressure dependent rate coefficients have been expressed using Chebyshev polynomials adopted for complex chemical activated reaction systems in this study, as well as unimolecular decomposition reactions. This method has also been tested and shows significant improvement over two convention methods, Troe's and SRI. The Levenberg-Marquardt algorithm has been incorporated with the QRRK code, CHEMACT, for the fitting of Chebyshev polynomials.

A pressure dependent mechanism which consists 147 species and 448 elementary reactions, based on thermochemical kinetic principals has been developed and calibrated by the experimental data. The reaction mechanisms (models) include pathways for formation of higher molecular weight products, such as the formation of methyl ethers. This accurate model based on principles of thermochemical kinetics and

statistical mechanics will not only provide fundamental understanding, but can be used to suggest directions toward process optimization for experimental testing.

The CHEMKIN interpreter has been modified in this study to take the flexible matrix size of Chebyshev polynomials and to generate the appropriate link file for further processing. Subroutines involving kinetic rate coefficient calculation and array size initialization are also modified to incorporate the Chebyshev polynomials expression. Existing CHEMKIN drivers can simply link with the modified CHEMKIN library to take temperature and pressure dependent Chebyshev polynomials without modification. A couple of drivers have been linked with the new CHEMKIN library and tested. These include a shock tube driver, a constant temperature and pressure driver, and an adiabatic constant pressure driver.

The mechanism is validated with methanol oxidation and pyrolysis experimental data and serves as a basis to build upon during the subsequent efforts on higher molecular weight oxygenated hydrocarbon (MTBE in this study). The methanol addition shows dramatic acceleration effect on the methane oxidation experimentally and predicted by the model.

APPENDIX A

TABLES

Table A.1 Average Retention Times and Relative Response Factors for Column A, 6' × 1/8" Feet Stainless Steel Column Packed with 50% 80/100 Poropak T and 50% 80/100 Poropak Q

Compounds	Rentention Time (min)	Relative Responce Factor (RRF)
CO	0.9	0.976
CH_4	1.3	1.000
CO_2	3.2	0.899
$CH_2{=}CH_2$	4.3	1.759
CH_3CH_3	5.5	1.981
$CH{\equiv}CH$	6.8	0.811
CH_2O	10.6	1.114
$CH_2{=}CHCH_3$	11.2	2.744
$CH_3CH_2CH_3$	11.5	2.960
$CH{\equiv}CCH3$	12.8	1.070
CH_3OCH_3	13.0	1.668
CH_3OH	14.4	1.088
$CH_3CH{=}O$	14.9	---
$CH_2{=}CHCH_2CH_3$	15.6	3.458
$CH_3CH_2CH_2CH_3$	15.9	3.963
$CH_2{=}CHCH_2CH_2CH_3$	20.7	4.390
$CH_3CH_2CH_2CH_2CH_3$	21.2	4.901
$CH_3OCH_2OCH_3$	22.1	2.492
$CH_2{=}CHCH_2CH_2CH_2CH_3$	32.9	5.541
$CH_3CH_2CH_2CH_2CH_2CH_3$	33.8	5.748

Table A.2 Average Retention Times and Relative Response Factors for Column B, 90 m × 0.53 mm Hewlett Packard Fused Silica Capillary Column

Compounds	Rentention Time (min)	Relative Responce Factor (RRF)
CH_4	3.50	1.000
$CH_2{\equiv}CH_2$	3.58	1.618
$CH_2{=}CH_2$	3.58	1.735
CH_3CH_3	3.62	1.861
CH_2O	3.8	< 0.01
$CH_2{=}CHCH_3$	3.9	2.655
$CH_3CH_2CH_3$	4.1	2.835
CH_3OCH_3	4.2	1.067
CH_3OH	4.5	0.729
$CH_2{=}CHCH_2CH_3$	4.6	3.437
$CH_3CH_2CH_2CH_3$	4.7	3.842
$CH_3CH{=}O$	4.7	---
$CH_2{=}CHCH_2CH_2CH_3$	6.3	4.414
$CH_3CH_2CH_2CH_2CH_3$	6.4	4.726
$CH_3OCH_2OCH_3$	7.2	1.626
CH_2O Trimer	8.1	---
$CH_2{=}CHCH_2CH_2CH_2CH_3$	8.9	5.507
$CH_3CH_2CH_2CH_2CH_2CH_3$	9.1	5.551

Table A.3 Thermodynamic Properties

SPECIES	Hf	S	Cp 300	400	500	600	800	1000	1500	DATE		ELEMENTS							
AR	.00	36.98	4.97	4.97	4.97	4.97	4.97	4.97	4.97	7/12/93			0		0		0 AR	1	G 0
N2	.00	45.77	6.96	6.99	7.08	7.20	7.50	7.81	8.31			N	2		0		0	0	G 0
O	59.50	38.40	5.00	5.00	5.00	5.00	5.00	5.00	5.00		120186	O	1		0		0	0	G 0
O2	.00	49.00	7.00	7.20	7.40	7.60	8.00	8.30	8.70		121386	O	2		0		0	0	G 0
H	52.10	27.30	4.90	4.90	4.90	4.90	4.90	4.90	4.90			H	1		0		0	0	G 0
OH	9.30	43.80	7.10	7.00	7.00	7.00	7.10	7.30	7.80		121286	H	1	O	1		0	0	G 0
HO2	2.50	54.70	8.30	8.90	9.40	9.90	10.70	11.30	12.40		20387	H	1	O	2		0	0	G 0
H2	.00	31.20	7.10	7.10	7.10	7.20	7.30	7.40	7.90			H	2		0		0	0	G 0
H2O	-57.80	45.10	8.20	8.40	8.60	8.90	9.50	10.10	11.50		20387	H	2	O	1		0	0	G 0
H2O2	-32.50	55.60	10.40	11.40	12.30	13.10	14.30	15.20	16.80		120186	H	2	O	2		0	0	G 0
C(S)	.00	21.83	2.06	2.86	3.50	4.02	4.73	5.16	5.65		J 3/61	C	1		0		0	0	S 0
C	170.88	38.31	4.98	4.97	4.97	4.97	4.97	4.97	4.97		J 3/61	C	1		0		0	0	G 0
CO	-26.40	47.20	6.90	7.00	7.10	7.20	7.60	7.90	8.40		121286	C	1	O	1		0	0	G 0
CO2	-94.00	51.00	8.90	9.80	10.60	11.30	12.30	13.00	13.90		121286	C	1	O	2		0	0	G 0
CH	142.00	43.72	6.97	6.97	7.03	7.12	7.41	7.77	8.74		J12/67	C	1	H	1		0	0	G 0
HCO	10.40	53.60	8.20	8.70	9.20	9.70	10.70	11.50	12.50		121286	C	1	H	1	O	1	0	G 0
CHO	10.40	53.60	8.20	8.70	9.20	9.70	10.70	11.50	12.50		121286	C	1	H	1	O	1	0	G 0
HC.*O	10.90	53.42	7.64	7.95	8.50	9.10	10.21	11.08	.00	10/12/95	ThmCal	C	1	H	1	O	1	0	G 0
HCO2.	-38.30	56.71	8.54	10.08	11.42	12.45	14.43	15.77	.00	7/12/93		C	1	H	1	O	2	0	G 0
CYC.OO	43.93	59.07	12.02	13.18	14.60	15.77	17.37	19.01	.00	10/12/95	ThmCal	C	1	H	1	O	2	0	G 0
HCO3	-31.30	74.08	13.22	14.83	16.12	17.17	19.11	20.51	.00	10/12/95	ThmCal	C	1	H	1	O	3	0	G 1
HCQ.*O	-31.30	74.08	13.22	14.83	16.12	17.17	19.11	20.51	.00	10/12/95	ThmCal	C	1	H	1	O	3	0	G 1
C.Q*O	-30.50	74.98	14.44	16.24	17.71	18.84	20.67	21.75	.00	10/12/95	ThmCal	C	1	H	1	O	3	0	G 2
CH2	92.35	46.32	8.28	8.62	8.99	9.37	10.15	10.88	12.22		J12/72	C	1	H	2		0	0	G 0
CH2S	101.45	44.15	8.33	8.77	9.21	9.64	10.45	11.16	12.47	K		C	1	H	2		0	0	G 0
CH2	92.36	46.32	8.33	8.77	9.21	9.64	10.45	11.16	12.47	K		C	1	H	2		0	0	G 0
HCOHT	48.36	60.95	10.79	11.97	12.81	13.58	14.84	15.81	17.26	10/ 9/95		C	1	H	2	O	1	0	G 1
HCOH	22.56	58.19	10.79	11.97	12.81	13.58	14.84	15.81	17.26	10/ 9/95		C	1	H	2	O	1	0	G 1
CH2O	-26.00	50.92	8.47	9.38	10.46	11.52	13.37	14.81	.00	10/12/95	ThmCal	C	1	H	2	O	1	0	G 0
HCO2H	-90.20	64.25	10.83	12.87	14.62	15.98	18.40	20.00	.00	10/12/95	ThmCal	C	1	H	2	O	2	0	G 1
O.HC.OOHT	4.86	76.52	18.58	20.11	21.46	22.65	24.57	25.96	27.80	7/01/94		C	1	H	2	O	3	0	G 2
O.HC.OOH	17.40	73.14	18.26	19.88	21.21	22.52	24.51	26.17	.00	10/ 9/95		C	1	H	2	O	3	0	G 2
HOC.HOO.	1.54	74.82	17.19	18.34	19.27	20.32	22.17	24.00	.00	10/ 9/95		C	1	H	2	O	3	0	G 2
HOC.HOO.T	-13.00	77.58	17.19	18.34	19.27	20.32	22.17	24.00	.00	10/ 9/95		C	1	H	2	O	3	0	G 2
HCQ*O	-67.40	73.86	15.27	17.67	19.67	21.26	23.83	25.48	.00	10/12/95	ThmCal	C	1	H	2	O	3	0	G 2
O.HC.OOH	17.40	75.90	18.26	19.88	21.21	22.52	24.51	26.17	.00	10/12/95	ThmCal	C	1	H	2	O	3	0	G 2
HOCHO.O.	-31.01	72.22	15.48	16.90	18.41	19.96	22.61	24.28	.00	10/12/95	ThmCal	C	1	H	2	O	3	0	G 1
CH3	34.80	46.30	9.20	10.00	10.70	11.50	12.80	14.00	16.20		J 6/69	C	1	H	3		0	0	G 0
CH3	35.12	46.38	9.26	10.05	10.81	11.54	12.90	14.09	16.26		J 6/69	C	1	H	3		0	0	G 0
CH3O	3.96	55.84	9.51	11.04	12.61	14.13	16.67	18.58	21.43	10/12/95	ThmCal	C	1	H	3	O	1	0	G 0
CH2OH	-3.60	59.03	11.77	13.27	14.42	15.47	17.22	18.61	20.85	10/12/95	ThmCal	C	1	H	3	O	1	0	G 1
CH3O.	3.96	55.84	9.51	11.04	12.61	14.13	16.67	18.58	21.43	10/12/95	ThmCal	C	1	H	3	O	1	0	G 0
C.H2OH	-3.60	59.03	11.77	13.27	14.42	15.47	17.22	18.61	20.85	10/12/95	ThmCal	C	1	H	3	O	1	0	G 1

CH3OO	4.50	66.54	13.09	15.15	17.04	18.79	21.32	23.32	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 1
CH2OOH	14.80	68.23	15.76	18.08	20.06	21.79	24.03	25.54	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 2
CH3OO	4.50	66.54	13.09	15.15	17.04	18.79	21.32	23.32	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 1
CH2OOH	14.80	68.23	15.76	18.08	20.06	21.79	24.03	25.54	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 2
CH2O.OH	-39.07	66.10	13.12	14.65	16.28	17.92	20.73	22.76	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 1
CO.H2OH	-39.07	66.10	13.12	14.65	16.28	17.92	20.73	22.76	.00	10/12/95	ThmCal	C	1	H	3	O	2	0	G 1
CH2OHOO	-30.30	81.70	19.68	21.49	23.31	24.97	27.72	29.82	33.11	rayez,lev	11/93	C	1	H	3	O	3	0	G 2
CQ.H2OH	-40.36	76.80	16.86	18.64	20.29	21.99	24.84	27.39	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 2
CQH2O.	-24.50	75.12	17.93	20.18	22.23	24.19	27.18	29.56	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 2
C.QOH	-32.56	80.40	19.05	20.64	22.18	23.78	26.38	28.57	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 3
HOC.HQ	-34.56	77.36	19.24	21.18	22.82	24.41	26.89	28.97	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 3
CQ.H2OH	-40.36	76.80	16.86	18.64	20.29	21.99	24.84	27.39	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 2
CQH2O.	-24.50	75.12	17.93	20.18	22.23	24.19	27.18	29.56	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 2
C.QOH	-32.56	80.40	19.05	20.64	22.18	23.78	26.38	28.57	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 3
HOC.HQ	-34.56	77.36	19.24	21.18	22.82	24.41	26.89	28.97	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 3
HC(OH)2O.	-82.97	73.68	16.46	18.20	20.02	21.85	24.99	27.08	.00	10/12/95	ThmCal	C	1	H	3	O	3	0	G 2
CH4	-17.90	44.40	8.50	9.80	11.10	12.30	14.80	16.90	20.40		J 3/61	C	1	H	4		0	0	G 0
CH3OH	-48.00	57.30	10.49	12.34	14.22	16.02	19.05	21.38	25.02	10/12/95	ThmCal	C	1	H	4	O	1	0	G 1
CH3OOH	-31.60	66.32	15.14	17.99	20.59	22.88	26.04	28.29	.00	10/12/95	ThmCal	C	1	H	4	O	2	0	G 2
HOCH2OH	-91.03	67.56	14.10	15.95	17.89	19.81	23.11	25.56	.00	10/12/95	ThmCal	C	1	H	4	O	2	0	G 2
CH2(OH)2	-91.03	66.18	14.10	15.95	17.89	19.81	23.11	25.56	.00	10/12/95	ThmCal	C	1	H	4	O	2	0	G 1
CQH2OH	-76.46	76.58	18.91	21.48	23.84	26.08	29.56	32.36	.00	10/12/95	ThmCal	C	1	H	4	O	3	0	G 3
HOCH2Q	-76.46	76.58	18.91	21.48	23.84	26.08	29.56	32.36	.00	10/12/95	ThmCal	C	1	H	4	O	3	0	G 3
HC(OH)3	-134.93	75.14	17.44	19.50	21.63	23.74	27.37	29.88	.00	10/12/95	ThmCal	C	1	H	4	O	3	0	G 3
HC#C.	134.46	51.51	10.05	10.42	10.71	10.96	11.47	11.95	12.92	10/12/95	ThmCal	C	2	H	1		0	0	G 0
C2H	134.46	51.51	10.05	10.42	10.71	10.96	11.47	11.95	12.92	10/12/95	ThmCal	C	2	H	1		0	0	G 0
C.*C*O	41.36	60.49	12.25	13.60	15.09	15.65	16.52	17.38	18.45	10/12/95	ThmCal	C	2	H	1	O	1	0	G 0
HCCO	41.36	60.49	12.25	13.60	15.09	15.65	16.52	17.38	18.45	10/12/95	ThmCal	C	2	H	1	O	1	0	G 0
CYC.*CO	80.99	54.82	13.57	15.37	16.34	16.85	17.69	18.23	19.18	10/12/95	ThmCal	C	2	H	1	O	1	0	G 0
C#COO	85.13	69.83	15.98	17.57	18.56	19.91	20.88	22.13	.00	10/12/95	ThmCal	C	2	H	1	O	2	0	G 1
C.#COOH	129.63	71.72	17.52	18.85	19.84	21.22	22.13	23.13	.00	10/12/95	ThmCal	C	2	H	1	O	2	0	G 2
C.*OCHO	-17.10	62.99	13.36	15.72	17.56	19.08	21.53	22.66	.00	10/12/95	ThmCal	C	2	H	1	O	2	0	G 1
CQ.*C*O	5.47	77.33	19.01	21.50	23.43	24.37	25.69	26.80	.00	10/12/95	ThmCal	C	2	H	1	O	3	0	G 1
C.Q*C*O	23.77	78.97	21.10	23.61	25.53	26.42	27.48	28.19	.00	10/12/95	ThmCal	C	2	H	1	O	3	0	G 2
CHOCO2	-60.74	75.90	15.73	18.79	21.18	22.99	25.96	27.67	.00	10/12/95	ThmCal	C	2	H	1	O	3	0	G 1
C.*OCO2H	-70.40	73.83	16.41	19.69	22.01	23.65	26.22	27.62	.00	10/12/95	ThmCal	C	2	H	1	O	3	0	G 2
C2H2	54.19	48.01	10.60	11.97	13.08	13.97	15.31	16.29	18.31		J 3/61	C	2	H	2		0	0	G 0
C#C	53.86	48.02	10.56	11.98	12.98	13.74	14.94	15.92	17.70	10/12/95	ThmCal	C	2	H	2		0	0	G 0
HC#COH	20.43	58.71	13.22	14.78	16.16	17.73	19.15	20.30	.00	10/12/95	ThmCal	C	2	H	2	O	1	0	G 1
C*C*O	-11.74	57.82	12.70	14.65	16.73	17.80	19.50	20.98	23.03	10/12/95	ThmCal	C	2	H	2	O	1	0	G 0
CYC*CO	24.09	51.63	13.91	16.58	18.28	19.37	21.03	22.14	23.94	10/12/95	ThmCal	C	2	H	2	O	1	0	G 0
C#COOH	49.03	69.61	18.03	20.41	22.11	24.00	25.60	27.10	.00	10/12/95	ThmCal	C	2	H	2	O	2	0	G 2
CHOCHO	-50.60	65.42	14.90	17.54	19.64	21.40	24.28	25.80	.00	10/12/95	ThmCal	C	2	H	2	O	2	0	G 1
O*CC*O	-50.60	65.42	14.90	17.54	19.64	21.40	24.28	25.80	.00	10/12/95	ThmCal	C	2	H	2	O	2	0	G 1
CO.C.*O	15.36	73.23	15.72	17.34	19.77	20.10	22.78	24.48	.00	10/12/95	ThmCal	C	2	H	2	O	2	0	G 1
C.CO2.	-9.08	67.06	14.48	17.27	19.50	21.19	23.69	25.34	.00	10/12/95	ThmCal	C	2	H	2	O	3	0	G 1
CQ*C*O	-30.63	77.11	21.06	24.34	26.38	28.46	30.41	31.77	.00	10/12/95	ThmCal	C	2	H	2	O	3	0	G 2
CHOCO2H	-112.70	77.36	16.71	20.09	22.79	24.88	28.34	30.47	.00	10/12/95	ThmCal	C	2	H	2	O	3	0	G 2

C2H3	71.62	56.61	10.01	11.97	13.66	15.08	17.32	19.05	21.85	10/12/95	ThmCal	C	2	H	3		0	0	G 0
CC.*O	-2.28	64.25	12.39	14.28	16.26	18.05	21.06	23.24	.00	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
C.CHO	3.12	60.40	12.92	15.31	17.44	19.24	22.10	24.12	.00	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
CH3C.*O	-2.28	64.25	12.39	14.28	16.26	18.05	21.06	23.24	.00	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
C.C*O	3.12	60.40	12.92	15.31	17.44	19.24	22.10	24.12	.00	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
CH2*C.OH	24.79	64.77	14.19	16.59	18.52	20.04	22.26	23.86	26.51	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
C*C.OH	24.79	64.77	14.19	16.59	18.52	20.04	22.26	23.86	26.51	10/12/95	ThmCal	C	2	H	3	O	1	0	G 1
C.*COH	29.49	64.30	13.96	16.57	18.61	20.16	22.37	23.95	26.56	10/12/95		C	2	H	3	O	1	0	G 1
C*CO.	13.79	61.80	12.81	15.33	17.49	19.29	22.06	24.11	27.30	10/12/95		C	2	H	3	O	1	0	G 0
CYC.CO	29.32	60.67	11.45	14.42	16.94	18.96	21.96	24.11	.00	10/12/95	ThmCal	C	2	H	3	O	1	0	G 0
C*COO.	29.73	73.35	16.51	19.57	21.72	23.57	26.33	28.36	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 1
C.*COOH	52.73	74.52	18.37	21.66	23.91	25.74	28.23	29.84	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
C*C.OOH	50.53	74.94	18.22	21.20	23.33	25.14	27.71	29.42	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
O*CCO.	-21.54	72.11	16.55	18.77	20.73	22.52	25.94	28.21	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 1
CCO2	-51.38	68.41	14.78	17.67	20.28	22.42	25.81	28.19	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 1
C.CO2H	-60.98	64.89	15.79	19.14	21.82	23.88	26.97	29.08	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
COHC.*O	-36.60	74.69	16.70	18.64	20.38	21.99	25.16	27.28	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
C.OHCHO	-33.70	71.20	16.17	18.50	20.61	22.52	25.66	27.90	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
CO.CHO	-21.54	72.11	16.55	18.77	20.73	22.52	25.94	28.21	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 1
CYCOCO.	-6.42	65.26	15.16	18.54	21.97	24.31	27.62	30.13	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 0
C.OCHO	-39.99	75.05	17.65	19.94	22.02	23.86	27.05	29.50	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
COC.*O	-47.49	72.65	16.30	18.59	20.86	22.89	26.39	29.04	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
CYCOOC.	41.61	67.33	14.19	17.84	21.28	24.01	27.89	31.11	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 0
C.CYCOO	37.56	69.27	16.67	19.89	23.45	25.81	28.97	32.02	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 1
CYCOCO.	-6.42	65.26	15.16	18.54	21.97	24.31	27.62	30.13	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 0
C.OCHO	-39.99	75.05	17.65	19.94	22.02	23.86	27.05	29.50	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
COC.*O	-47.49	72.65	16.30	18.59	20.86	22.89	26.39	29.04	.00	10/12/95	ThmCal	C	2	H	3	O	2	0	G 2
HC*OC.OOH	-19.13	80.22	20.98	24.03	26.56	28.79	32.11	34.70	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 3
HC*OCOO.	-22.83	82.81	20.29	22.76	24.74	26.59	30.05	32.84	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 0
C.*OCOOH	-22.03	83.71	21.51	24.17	26.33	28.26	31.61	34.08	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 0
O*COO.C	-44.38	77.45	18.48	21.50	24.10	26.30	29.80	32.44	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 2
C.CO3H	-38.18	75.88	20.23	23.94	26.87	29.16	32.40	34.56	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 4
CCO3	-44.38	77.45	18.48	21.50	24.10	26.30	29.80	32.44	.00	10/12/95	ThmCal	C	2	H	3	O	3	0	G 3
C2H4	12.52	52.47	10.20	12.72	15.02	17.00	20.14	22.54	26.38	10/12/95	ThmCal	C	2	H	4		0	0	G 0
CCHO	-39.18	63.13	13.22	15.71	18.22	20.47	24.22	26.97	.00	10/12/95	ThmCal	C	2	H	4	O	1	0	G 1
CH3CHO	-39.18	63.13	13.22	15.71	18.22	20.47	24.22	26.97	.00	10/12/95	ThmCal	C	2	H	4	O	1	0	G 1
CH2*CHOH	-29.61	62.91	14.15	17.32	19.97	22.08	25.19	27.44	31.09	10/12/95	ThmCal	C	2	H	4	O	1	0	G 1
C*COH	-29.61	62.91	14.15	17.32	19.97	22.08	25.19	27.44	31.09	10/12/95	ThmCal	C	2	H	4	O	1	0	G 1
CYCCO	-12.58	58.14	11.58	15.09	18.25	20.85	24.78	27.60	.00	10/12/95	ThmCal	C	2	H	4	O	1	0	G 0
C*COOH	-6.37	73.13	18.56	22.41	25.27	27.66	31.05	33.33	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2
O*CCOH	-73.50	73.57	17.53	20.07	22.34	24.41	28.32	31.01	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2
CCO2H	-103.28	67.62	16.09	19.54	22.60	25.11	29.09	31.93	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2
COHCHO	-73.50	73.57	17.53	20.07	22.34	24.41	28.32	31.01	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2
COCHO	-84.39	71.53	17.13	20.02	22.82	25.31	29.55	32.77	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2
CYC2O2	-2.29	60.76	14.05	18.69	22.94	26.31	31.07	34.90	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 0
CCYCOO	-11.44	64.48	17.44	21.25	25.36	28.21	32.13	35.76	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 1
CYCOCOH	-58.38	66.72	16.14	19.84	23.58	26.20	30.00	32.93	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 1
COCHO	-84.39	71.53	17.13	20.02	22.82	25.31	29.55	32.77	.00	10/12/95	ThmCal	C	2	H	4	O	2	0	G 2

HC*OCOOH	-58.93	82.59	22.34	25.60	28.29	30.68	34.77	37.81	.00	10/12/95	ThmCal	C	2	H	4	O	3	0	G	3
O*COOHC	-80.48	77.23	20.53	24.34	27.65	30.39	34.52	37.41	.00	10/12/95	ThmCal	C	2	H	4	O	3	0	G	3
O*CCQ	-58.93	82.59	22.34	25.60	28.29	30.68	34.77	37.81	.00	10/12/95	ThmCal	C	2	H	4	O	3	0	G	3
CCO3H	-80.48	77.23	20.53	24.34	27.65	30.39	34.52	37.41	.00	10/12/95	ThmCal	C	2	H	4	O	3	0	G	4
CC.	28.60	59.87	11.61	14.32	16.89	19.18	22.88	25.80	30.50	10/12/95	ThmCal	C	2	H	5		0	0	G	1
C2H5	28.60	59.87	11.61	14.32	16.89	19.18	22.88	25.80	30.50	10/12/95	ThmCal	C	2	H	5		0	0	G	1
CCO.	-4.24	65.64	14.50	17.89	20.91	23.56	27.77	30.91	.00	10/12/95	ThmCal	C	2	H	5	O	1	0	G	1
CC.OH	-14.30	67.88	15.81	18.89	21.50	23.78	27.48	30.32	.00	10/12/95	ThmCal	C	2	H	5	O	1	0	G	2
C.COH	-7.20	70.51	14.71	17.83	20.61	23.05	26.99	29.97	.00	10/12/95	ThmCal	C	2	H	5	O	1	0	G	2
COC.	1.00	67.28	16.30	19.30	21.70	23.93	27.96	30.87	.00	10/12/95	ThmCal	C	2	H	5	O	1	0	G	2
C.COOH	6.87	79.53	19.52	23.36	26.56	29.32	33.44	36.77	.00	10/12/95		C	2	H	5	O	2	0	G	3
CCOO	-5.53	76.34	18.24	21.88	24.92	27.63	31.88	35.54	.00	10/12/95	ThmCal	C	2	H	5	O	2	0	G	2
CC.OOH	2.27	79.94	20.43	23.88	26.81	29.42	33.42	36.72	.00	10/12/95	ThmCal	C	2	H	5	O	2	0	G	3
CCO.OH	-50.04	72.84	19.06	22.64	26.24	28.91	32.99	36.24	.00	10/12/95	ThmCal	C	2	H	5	O	2	0	G	2
C.C(OH)2	-54.50	79.09	19.27	22.58	25.94	28.40	32.21	35.30	.00	10/12/95		C	2	H	5	O	2	0	G	3
CC.(OH)2	-62.80	78.77	19.27	22.01	25.03	27.39	31.25	34.51	.00	10/12/95		C	2	H	5	O	2	0	G	3
COC.OH	-44.53	76.18	19.72	22.69	25.15	27.50	31.85	34.93	.00	10/12/95		C	2	H	5	O	2	0	G	3
COCO.	-34.47	73.94	18.41	21.69	24.56	27.28	32.14	35.52	.00	10/12/95	ThmCal	C	2	H	5	O	2	0	G	2
C.OCOH	-42.03	77.54	19.91	22.91	25.37	27.72	32.02	35.05	.00	10/12/95	ThmCal	C	2	H	5	O	2	0	G	3
COHCOO	-41.33	86.98	21.34	25.39	28.64	31.50	35.99	39.71	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	2
COHCQ.	-41.33	86.98	21.34	25.39	28.64	31.50	35.99	39.71	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	2
CO.COOH	-25.47	85.30	22.41	26.93	30.58	33.70	38.33	41.88	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	2
CO.CQ	-25.47	85.30	22.41	26.93	30.58	33.70	38.33	41.88	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	2
C.OHCOOH	-35.53	87.54	23.72	27.93	31.17	33.92	38.04	41.29	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	3
C.OHCQ	-35.53	87.54	23.72	27.93	31.17	33.92	38.04	41.29	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	3
COHC.OOH	-33.53	90.58	23.53	27.39	30.53	33.29	37.53	40.89	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	3
C.CQOH	-38.43	88.11	24.08	28.11	31.89	34.67	38.66	42.10	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	4
CCQ.OH	-51.33	83.54	22.80	26.63	30.25	32.98	37.10	40.87	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	3
CCQO.	-35.47	81.86	23.87	28.17	32.19	35.18	39.44	43.04	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	3
CC.QOH	-46.73	89.97	24.08	27.54	30.98	33.66	37.70	41.31	.00	10/12/95	ThmCal	C	2	H	5	O	3	0	G	4
HOOCCOO.	-26.76	93.25	26.15	30.92	34.59	37.77	42.44	46.51	.00	10/12/95	ThmCal	C	2	H	5	O	4	0	G	4
HOOCC.OOH	-18.96	98.22	28.34	32.92	36.48	39.56	43.98	47.69	.00	10/12/95	ThmCal	C	2	H	5	O	4	0	G	5
C2H6	-20.24	54.85	12.58	15.77	18.68	21.31	25.80	29.33	34.91		L 5/72	C	2	H	6		0	0	G	0
CC	-20.40	55.08	12.38	15.68	18.80	21.58	26.04	29.54	35.16	10/12/95	ThmCal	C	2	H	6		0	0	G	1
C2H6	-20.40	55.08	12.38	15.68	18.80	21.58	26.04	29.54	35.16	10/12/95	ThmCal	C	2	H	6		0	0	G	1
CCOH	-56.20	67.10	15.48	19.19	22.52	25.45	30.15	33.71	.00	10/12/95	ThmCal	C	2	H	6	O	1	0	G	2
COC	-43.40	63.76	15.78	19.38	22.50	25.38	30.46	34.14	.00	10/12/95	ThmCal	C	2	H	6	O	1	0	G	2
CCOOH	-41.63	76.12	20.29	24.72	28.47	31.72	36.60	40.51	.00	10/12/95	ThmCal	C	2	H	6	O	2	0	G	3
CC(OH)2	-102.00	74.30	20.04	23.94	27.85	30.80	35.37	39.04	.00	10/12/95	ThmCal	C	2	H	6	O	2	0	G	3
COCOH	-86.43	75.40	19.39	22.99	26.17	29.17	34.52	38.32	.00	10/12/95	ThmCal	C	2	H	6	O	2	0	G	3
COHCOOH	-77.43	86.76	23.39	28.23	32.19	35.59	40.71	44.68	.00	10/12/95	ThmCal	C	2	H	6	O	3	0	G	3
CCQOH	-87.43	83.32	24.85	29.47	33.80	37.07	41.82	45.84	.00	10/12/95	ThmCal	C	2	H	6	O	3	0	G	4
HOOCCOOH	-62.86	94.40	28.20	33.76	38.14	41.86	47.16	51.48	.00	10/12/95	ThmCal	C	2	H	6	O	4	0	G	5
C#CCBR	50.43	71.35	17.75	20.67	23.12	24.96	27.77	29.79	.00	10/12/95	ThmCal	C	3	H	3	BR	1	0	G	1
C#CC.	81.58	59.57	13.76	16.14	18.14	19.80	22.39	24.34	27.47	10/12/95	ThmCal	C	3	H	3		0	0	G	1
C*C*C.	81.62	62.51	13.65	16.07	18.08	19.85	22.46	24.44	27.50	10/12/95	ThmCal	C	3	H	3		0	0	G	0
C#C.C*C	103.13	60.69	11.87	14.81	17.28	19.25	22.17	24.25	27.42	10/12/95	ThmCal	C	3	H	3		0	0	G	0
C#CCCL	39.17	68.78	17.25	20.21	22.48	24.43	27.22	29.45	.00	10/12/95	ThmCal	C	3	H	3	CL	1	0	G	1

C*C.CHOA	36.58	69.20	16.67	19.71	22.14	24.10	27.22	29.37	.00	10/12/95	ThmCal	C	3	H	3	O	1	0	G 1
C*CC.*O	21.58	68.54	16.88	20.25	22.77	24.73	27.74	29.79	.00	10/12/95	ThmCal	C	3	H	3	O	1	0	G 1
C*C.CHOB	16.98	66.88	16.17	19.75	22.52	24.67	27.86	29.97	.00	10/12/95	ThmCal	C	3	H	3	O	1	0	G 1
C*C*CO.	40.49	67.80	16.71	19.73	22.19	24.29	27.36	29.61	33.00	10/12/95	ThmCal	C	3	H	3	O	1	0	G 0
C*C.C*O	38.99	70.21	16.47	19.48	22.00	24.15	27.28	29.53	32.98	10/12/95	ThmCal	C	3	H	3	O	1	0	G 1
C#CCO.	61.78	68.46	16.85	19.53	21.83	23.79	26.93	29.17	33.45	10/12/95	ThmCal	C	3	H	3	O	1	0	G 1
C#CC.OH	44.72	69.47	17.24	19.63	21.69	23.49	26.40	28.48	31.55	10/12/95	ThmCal	C	3	H	3	O	1	0	G 2
C#CCOO	61.49	78.81	20.59	23.52	25.84	27.86	31.04	33.80	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 2
C.#CCOOH	105.99	80.70	22.13	24.80	27.12	29.17	32.29	34.80	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 3
C#CC.OOH	58.89	74.78	21.10	24.54	27.31	29.63	33.01	35.63	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 3
C*C*COO	63.93	79.35	20.41	23.97	26.42	28.57	31.63	33.86	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 1
C.*C*COOH	61.50	79.78	21.85	25.36	28.15	30.43	33.56	35.64	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 2
C.#COOC	129.80	80.66	22.40	25.28	27.67	30.19	33.33	36.09	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 3
C#COOC.	93.60	80.69	23.43	26.76	29.14	31.52	34.30	36.79	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 3
C*CO.CHO	-5.45	74.05	20.27	23.70	26.47	28.72	32.35	34.60	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 1
C.C*OCHO	-23.01	76.81	19.67	22.76	25.41	27.68	31.52	33.99	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 2
C.C*OC*O	-22.18	74.88	18.80	22.53	25.61	28.14	32.04	34.59	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 2
CC*OC.*O	-24.68	73.86	17.74	21.36	24.66	27.48	31.50	34.33	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 2
CYC*C.COO	61.07	71.26	16.38	20.43	24.11	27.21	31.82	35.15	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 0
C.CYC*COO	63.49	67.45	19.16	24.29	27.84	30.66	34.70	37.02	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 1
CCYC*C.OO	81.79	71.07	19.82	24.12	27.17	29.74	33.61	35.90	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 1
CYC*CC.OO	37.67	65.64	15.18	19.82	23.97	27.41	32.41	35.92	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 0
CYC.*CCOO	58.57	71.31	16.76	20.91	24.60	27.69	32.23	35.48	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 0
C.*CYCCOO	83.21	73.35	17.29	21.57	25.20	28.13	32.57	35.66	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 0
C*CYCC.OO	57.61	68.15	15.94	20.50	24.48	27.73	32.64	36.01	.00	10/12/95	ThmCal	C	3	H	3	O	2	0	G 0
C*C*C	45.90	58.30	14.30	17.00	19.60	22.00	25.70	27.70	32.00		API53	C	3	H	4		0	0	G 0
C#CC	44.28	59.28	14.60	17.31	19.70	21.75	25.09	27.65	32.78	10/12/95	ThmCal	C	3	H	4		0	0	G 1
C*C*C	46.72	58.47	14.10	17.12	19.72	22.00	25.44	28.04	32.08	10/12/95	ThmCal	C	3	H	4		0	0	G 0
CYCC*C	66.23	58.19	12.70	16.24	19.24	21.67	25.33	27.98	32.06	10/12/95	ThmCal	C	3	H	4		0	0	G 0
C*CCHO	-20.32	67.39	17.01	20.92	24.08	26.62	30.56	33.28	.00	10/12/95	ThmCal	C	3	H	4	O	1	0	G 1
C*C*COH	4.59	68.91	18.05	21.72	24.67	27.08	30.49	32.94	36.79	10/12/95	ThmCal	C	3	H	4	O	1	0	G 1
C#CCOH	9.82	69.92	17.83	20.83	23.44	25.68	29.31	31.97	37.04	10/12/95	ThmCal	C	3	H	4	O	1	0	G 2
C#CCOOH	25.39	78.59	22.64	26.36	29.39	31.95	35.76	38.77	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 3
C*C*COOH	27.83	79.13	22.46	26.81	29.97	32.66	36.35	38.83	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 2
C#COOC	48.20	78.90	22.91	26.84	29.94	32.97	36.80	40.06	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 3
C*COHCHO	-57.41	75.51	21.25	25.00	28.08	30.61	34.73	37.40	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 2
CC*OC*O	-64.48	76.23	19.10	22.93	26.39	29.37	34.16	37.44	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 2
CC*OC*O	-64.48	76.23	19.10	22.93	26.39	29.37	34.16	37.44	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 2
CYC*CCOO	4.17	69.45	16.72	21.64	26.05	29.73	35.16	39.06	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 0
CCYC*COO	27.39	69.21	19.78	24.85	28.62	31.78	36.54	39.48	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 1
C*CYCCOO	24.11	71.96	17.48	22.32	26.56	30.05	35.39	39.15	.00	10/12/95	ThmCal	C	3	H	4	O	2	0	G 1
C*CCBR	10.80	75.88	18.60	22.59	26.14	29.00	33.42	36.63	.00	10/12/95	ThmCal	C	3	H	5	BR	1	0	G 1
C.*CC	63.75	65.20	15.26	18.48	21.36	23.87	27.92	31.00	35.86	10/12/95	ThmCal	C	3	H	5		0	0	G 1
C*C.C	61.55	65.62	15.11	18.02	20.78	23.27	27.40	30.58	35.63	10/12/95	ThmCal	C	3	H	5		0	0	G 1
C*CC.	40.75	62.05	14.83	18.67	21.94	24.67	28.90	32.03	36.90	10/12/95	ThmCal	C	3	H	5		0	0	G 1
CC*C.	63.75	65.20	15.26	18.48	21.36	23.87	27.92	31.00	35.86	10/12/95	ThmCal	C	3	H	5		0	0	G 1
CC.*C	61.55	65.62	15.11	18.02	20.78	23.27	27.40	30.58	35.63	10/12/95	ThmCal	C	3	H	5		0	0	G 1
C.C*C	40.75	62.05	14.83	18.67	21.94	24.67	28.90	32.03	36.90	10/12/95	ThmCal	C	3	H	5		0	0	G 1

C*CCCL	-.46	73.31	18.10	22.13	25.50	28.47	32.87	36.29	.00	10/12/95	ThmCal	C	3	H	5	CL	1	0 G 1
CCC.*O	-7.60	73.85	18.59	21.98	24.96	27.55	32.16	35.44	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
CC.CHO	-4.70	70.36	18.06	21.84	25.19	28.08	32.66	36.06	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.CCHO	4.50	76.14	18.65	22.05	25.01	27.57	32.16	35.43	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C2.C*O	-9.26	71.11	18.29	22.19	25.74	28.77	33.50	36.93	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C*CCO.	22.15	72.99	17.70	21.45	24.85	27.83	32.58	36.01	41.06	10/12/95	ThmCal	C	3	H	5	O	1	0 G 1
C*CC.OH	3.69	70.64	17.14	20.93	24.38	27.40	32.21	35.67	40.80	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.*CCOH	29.29	75.84	18.49	22.00	25.10	27.80	32.14	35.32	40.12	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C*C.COH	24.59	76.31	18.72	22.02	25.01	27.68	32.03	35.23	40.07	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C*CO.C	13.15	71.16	18.20	21.96	25.16	27.93	32.57	35.92	40.78	10/12/95	ThmCal	C	3	H	5	O	1	0 G 1
C*COHC.	-2.71	70.86	18.56	22.70	25.99	28.70	33.11	36.26	40.88	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.*COHC	20.29	74.01	18.99	22.51	25.41	27.90	32.13	35.23	39.84	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
CC*C.OH	16.92	73.36	19.44	23.10	26.22	28.83	32.86	35.81	40.52	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
CC.*COH	19.42	73.31	19.06	22.62	25.73	28.35	32.45	35.48	40.34	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.C*COH	-1.38	69.74	18.78	23.27	26.89	29.75	33.95	36.93	41.61	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.*CCOH	29.29	75.84	18.49	22.00	25.10	27.80	32.14	35.32	40.12	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
C.CYC2O	27.12	70.36	16.81	21.36	25.54	28.54	33.34	36.70	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 1
CCYC.CO	20.02	68.10	17.45	22.05	26.14	29.05	33.68	36.95	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 1
CYCC.CO	27.99	68.98	13.65	18.38	22.59	26.39	32.15	35.79	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 0
CYC.CCO	23.54	65.69	15.02	20.04	24.38	27.89	33.08	36.75	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 0
C*COC.	18.86	73.88	20.47	24.89	28.35	31.03	35.02	38.13	.00	10/12/95	ThmCal	C	3	H	5	O	1	0 G 2
CO.CCHO	-28.34	81.91	21.54	25.62	29.03	31.95	37.05	40.54	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C*CCOO.	20.86	83.69	21.44	25.44	28.86	31.90	36.69	40.64	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 2
C*CC.OOH	18.26	79.66	21.95	26.46	30.33	33.67	38.66	42.47	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 2
C.*CCOOH	43.17	85.30	23.91	27.99	31.28	34.32	38.27	41.62	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C*COOC.	40.20	83.98	24.06	28.93	32.57	35.54	40.24	43.54	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 2
C*C.COOH	41.66	85.28	23.15	27.07	30.47	33.47	38.07	41.70	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C.*COOHC	43.53	84.23	23.40	27.60	30.71	33.48	37.99	41.12	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 4
C*COOHC.	20.53	81.08	22.97	27.79	31.29	34.28	38.97	42.15	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C*COO.C	20.53	85.24	21.54	25.51	28.52	31.31	36.09	39.64	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
CC*C.OOH	40.16	83.58	23.85	28.19	31.52	34.41	38.72	41.70	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 4
CC.*COOH	42.66	83.53	23.47	27.71	31.03	33.93	38.31	41.37	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C.C*COOH	21.86	79.96	23.19	28.36	32.19	35.33	39.81	42.82	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
CC*COO.	21.86	81.94	21.76	26.08	29.42	32.36	36.93	40.31	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 2
CCO.C*O	-31.24	79.64	21.01	25.73	29.88	32.98	38.10	41.58	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 2
C.COHC*O	-34.20	85.89	21.22	25.67	29.58	32.47	37.32	40.64	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
CCOHC.*O	-46.30	82.22	21.16	25.60	29.53	32.45	37.32	40.65	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C.C*OCOH	-43.58	80.56	21.98	25.96	29.23	32.05	36.91	40.34	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
CC*OC.OH	-46.08	79.54	20.92	24.79	28.28	31.39	36.37	40.08	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 3
C.*CCOOH	44.86	84.51	23.30	27.53	31.05	34.07	38.59	42.12	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 1
CYC.CCOO	16.71	74.36	19.57	25.01	30.25	34.36	39.86	44.43	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 0
CYCC.COO	19.16	74.98	17.93	23.52	28.81	33.27	39.29	43.77	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 0
C.CYCCOO	37.41	75.73	19.28	24.95	30.23	34.00	39.63	44.00	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 1
CCYC.COO	29.91	78.30	20.27	25.00	29.69	33.19	38.66	43.12	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 1
CCYCC.OO	32.31	76.14	20.19	25.47	30.48	34.10	39.61	43.95	.00	10/12/95	ThmCal	C	3	H	5	O	2	0 G 1
CC.QC*O	-25.73	90.52	26.67	31.89	36.13	39.25	44.11	47.69	.00	10/12/95	ThmCal	C	3	H	5	O	3	0 G 4
CC*OCQ.	-35.21	91.15	25.04	29.05	32.41	35.46	40.76	45.02	.00	10/12/95	ThmCal	C	3	H	5	O	3	0 G 3
C.C*OCQ	-29.01	89.58	26.79	31.49	35.18	38.32	43.36	47.14	.00	10/12/95	ThmCal	C	3	H	5	O	3	0 G 4

CC*OC.OOH	-34.01	90.42	26.25	30.72	34.40	37.60	42.78	46.68	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
CCQ.C*O	-31.53	89.59	24.75	29.72	33.89	37.05	42.21	46.21	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 3
C.CQC*O	-18.63	94.16	26.03	31.20	35.53	38.74	43.77	47.44	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
CCQC.*O	-30.73	90.49	25.97	31.13	35.48	38.72	43.77	47.45	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
CQC.C*O	-25.93	90.02	25.97	30.88	34.86	38.22	43.22	47.03	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
CQCC.*O	-28.83	93.51	26.50	31.02	34.63	37.69	42.72	46.41	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
C.QCC*O	-19.33	93.50	27.95	32.54	36.06	39.02	43.87	47.39	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
CQ.CC*O	-29.63	92.61	25.28	29.61	33.04	36.02	41.16	45.17	.00	10/12/95	ThmCal	C	3	H	5	O	3	0	G 4
C*OCQCO.	-52.67	98.65	29.25	34.59	39.20	42.68	48.23	52.15	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 4
C*OCQ.COH	-68.53	100.33	28.18	33.05	37.26	40.48	45.89	49.98	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 4
C.*OCQCOH	-67.73	101.23	29.40	34.46	38.85	42.15	47.45	51.22	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 4
CYCC.OOCQ	-7.89	93.05	27.98	34.73	40.87	45.24	51.07	55.90	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 2
CYCCOOCQ.	-15.69	89.45	25.79	32.73	38.98	43.45	49.53	54.72	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 1
CYC.COOCQ	-5.44	92.30	27.23	34.02	40.25	44.73	50.71	55.64	.00	10/12/95	ThmCal	C	3	H	5	O	4	0	G 2
C*CC	4.65	63.81	15.45	19.23	22.72	25.79	30.74	34.49	40.39	10/12/95	ThmCal	C	3	H	6		0	0	G 1
CC*C	4.65	63.81	15.45	19.23	22.72	25.79	30.74	34.49	40.39	10/12/95	ThmCal	C	3	H	6		0	0	G 1
CCCHO	-44.50	72.73	19.42	23.41	26.92	29.97	35.32	39.17	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C2C*O	-51.56	70.09	17.97	22.00	25.89	29.34	34.93	39.15	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C.CO.C	35.46	76.48	19.73	24.16	28.20	31.25	36.33	40.01	45.86	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
CC.O.C	28.36	73.84	20.83	25.22	29.09	31.98	36.82	40.36	46.03	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
CC.CO.	37.18	79.50	18.50	22.51	26.06	29.52	35.09	38.80	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C2.CHO.	35.46	75.67	19.73	24.16	28.20	31.25	36.33	40.01	45.86	10/12/95	ThmCal	C	3	H	6	O	1	0	G 3
CCCHO	-44.50	72.73	19.42	23.41	26.92	29.97	35.32	39.17	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C*CCOH	-29.81	74.45	18.68	22.75	26.46	29.72	34.96	38.81	44.65	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C*COHC	-38.81	72.62	19.18	23.26	26.77	29.82	34.95	38.72	44.37	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
CC*COH	-37.48	71.50	19.40	23.83	27.67	30.87	35.79	39.39	45.10	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C*COC	-24.54	53.22	18.79	23.07	26.55	29.43	34.26	37.91	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
C*COHC	-38.81	72.62	19.18	23.26	26.77	29.82	34.95	38.72	44.37	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
CCYC2O	-21.88	66.95	17.58	22.72	27.45	30.94	36.50	40.44	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 1
CYCCCO	-18.36	64.54	15.15	20.71	25.69	29.78	35.90	40.24	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 0
C*COC	-25.54	71.74	19.95	24.97	29.15	32.48	37.52	41.40	.00	10/12/95	ThmCal	C	3	H	6	O	1	0	G 2
COHCCHO	-80.30	83.37	22.52	26.92	30.64	33.84	39.43	43.34	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 3
C*CCOOH	-15.24	83.47	23.49	28.28	32.41	35.99	41.41	45.61	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 3
C*COOC	-7.20	82.42	23.44	28.84	33.10	36.63	42.25	46.29	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 2
C*COOHC	-15.57	82.84	23.59	28.35	32.07	35.40	40.81	44.61	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 4
CC*COOH	-14.24	81.72	23.81	28.92	32.97	36.45	41.65	45.28	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 4
CCOHC*O	-83.20	81.10	21.99	27.03	31.49	34.87	40.48	44.38	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 3
C*CMOOH	-15.57	82.84	23.59	28.35	32.07	35.40	40.81	44.61	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 3
CC*OCO.	-33.92	80.45	21.30	25.06	28.40	31.39	36.65	40.39	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 2
C(OH)CCHO	-80.30	83.37	22.52	26.92	30.64	33.84	39.43	43.34	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 3
CYC3O2	-27.19	70.54	19.43	25.85	31.91	36.66	43.04	48.22	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 0
CCYCCOO	-11.59	72.32	20.05	26.31	32.14	36.40	42.79	47.74	.00	10/12/95	ThmCal	C	3	H	6	O	2	0	G 1
CC*OCQ	-71.31	90.93	27.09	31.89	35.96	39.55	45.48	49.99	.00	10/12/95	ThmCal	C	3	H	6	O	3	0	G 4
CCQC*O	-67.63	89.37	26.80	32.56	37.44	41.14	46.93	51.18	.00	10/12/95	ThmCal	C	3	H	6	O	3	0	G 4
CQCC*O	-65.46	92.21	26.63	31.70	36.14	39.96	45.85	50.28	.00	10/12/95	ThmCal	C	3	H	6	O	3	0	G 4
C*OCQCOH	-104.63	100.11	30.23	35.89	40.81	44.57	50.61	54.95	.00	10/12/95	ThmCal	C	3	H	6	O	4	0	G 4
CYCCOOCQ	-52.16	89.63	26.79	34.47	41.24	46.25	53.46	59.06	.00	10/12/95	ThmCal	C	3	H	6	O	4	0	G 2
CCC.	23.67	69.29	17.11	21.27	25.14	28.53	33.95	38.14	44.70	10/12/95	ThmCal	C	3	H	7		0	0	G 2

CC.C	21.02	68.94	16.38	20.30	23.95	27.54	33.36	37.43	44.16	10/12/95	ThmCal	C	3	H	7		0	0 G 2
CCCO.	-9.17	75.06	20.00	24.84	29.16	32.91	38.84	43.25	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 2
C2.COH	-16.50	77.94	20.71	25.46	29.81	33.14	38.71	42.81	49.45	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
C2C.OH	-23.60	73.93	21.81	26.52	30.70	33.87	39.20	43.16	49.62	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
C2CO.	-13.54	73.06	20.50	25.52	30.11	33.65	39.49	43.75	50.52	10/12/95	ThmCal	C	3	H	7	O	1	0 G 2
C.CCOH	-12.13	79.93	20.21	24.78	28.86	32.40	38.06	42.31	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
CC.COH	-14.78	80.96	19.48	23.81	27.67	31.41	37.47	41.60	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
CCC.OH	-19.23	77.30	21.31	25.84	29.75	33.13	38.55	42.66	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
C.OCC	-7.20	77.08	21.29	26.15	30.00	33.36	39.06	43.20	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
COCC.	-2.60	78.35	20.00	24.87	28.89	32.41	38.40	42.73	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
COC.C	-9.70	76.08	20.64	25.56	29.49	32.92	38.74	42.98	.00	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
CCOHC.	-16.50	77.94	20.71	25.46	29.81	33.14	38.71	42.81	49.45	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
CC.OHC	-23.60	73.93	21.81	26.52	30.70	33.87	39.20	43.16	49.62	10/12/95	ThmCal	C	3	H	7	O	1	0 G 3
CCO.C	-13.54	73.06	20.50	25.52	30.11	33.65	39.49	43.75	50.52	10/12/95	ThmCal	C	3	H	7	O	1	0 G 2
CC.COOH	-.21	89.98	24.29	29.34	33.62	37.68	43.92	48.40	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
CCC.OOH	-2.66	89.36	25.93	30.83	35.06	38.77	44.49	49.06	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
CCCOO	-10.46	85.76	23.74	28.83	33.17	36.98	42.95	47.88	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 3
C.CCOOH	2.44	88.95	25.02	30.31	34.81	38.67	44.51	49.11	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
C2COO	-13.83	83.01	24.24	29.51	34.12	37.72	43.60	48.38	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 3
C2C.OOH	-8.43	88.77	26.51	31.04	35.22	38.60	44.19	48.73	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 3
C2.COOH	-.93	87.59	25.52	30.99	35.76	39.41	45.16	49.61	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 3
CCOOC.	3.94	88.70	25.79	31.24	35.77	39.60	45.79	50.72	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
C.COOC	6.54	90.20	24.40	29.79	34.39	38.29	44.64	49.73	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
CC.OOC	3.44	87.72	25.31	30.31	34.64	38.39	44.62	49.68	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
C.COOHC	-.93	87.59	25.52	30.99	35.76	39.41	45.16	49.61	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
CC.OOHC	-8.43	88.77	26.51	31.04	35.22	38.60	44.19	48.73	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
COCOC.	-37.43	85.38	25.20	29.95	33.65	37.08	43.43	47.81	.00	10/12/95	ThmCal	C	3	H	7	O	2	0 G 4
COC.OC	-39.93	83.01	24.55	29.36	33.14	36.64	43.11	47.59	.00	10/12/95		C	3	H	7	O	2	0 G 4
C.CQCOH	-36.73	98.23	28.62	34.50	39.48	43.28	49.27	53.78	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CCQCO.	-33.77	91.98	28.41	34.56	39.78	43.79	50.05	54.72	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CCQ.COH	-49.63	93.66	27.34	33.02	37.84	41.59	47.71	52.55	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CCQC.OH	-43.83	94.22	29.72	35.56	40.37	44.01	49.76	54.13	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CC.QCOH	-44.23	99.42	29.61	34.55	38.94	42.47	48.30	52.90	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
C.COHCQ	-37.73	98.98	28.62	34.50	39.48	43.28	49.27	53.78	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CCO.CQ	-34.77	92.73	28.41	34.56	39.78	43.79	50.05	54.72	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CCOHCQ.	-50.63	94.41	27.34	33.02	37.84	41.59	47.71	52.55	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CCOHC.Q	-42.83	98.01	29.53	35.02	39.73	43.38	49.25	53.73	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CQCCO.	-30.40	94.72	27.91	33.88	38.83	43.05	49.40	54.22	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CQ.CCOH	-46.26	96.40	26.84	32.34	36.89	40.85	47.06	52.05	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
C.QCCOH	-38.46	100.00	29.03	34.34	38.78	42.64	48.60	53.23	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CQC.COH	-36.01	100.62	27.39	32.85	37.34	41.55	48.03	52.57	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
CQCC.OH	-40.46	96.96	29.22	34.88	39.42	43.27	49.11	53.63	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
COCOCOH	-124.86	93.50	28.29	33.64	38.12	42.32	49.99	55.26	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 5
COCOCO.	-72.90	92.04	27.31	32.34	36.51	40.43	47.61	52.46	.00	10/12/95	ThmCal	C	3	H	7	O	3	0 G 4
CCQ.CQ	-34.06	102.73	32.15	38.55	43.79	47.86	54.16	59.35	.00	10/12/95	ThmCal	C	3	H	7	O	4	0 G 5
CCQCQ.	-34.06	102.73	32.15	38.55	43.79	47.86	54.16	59.35	.00	10/12/95	ThmCal	C	3	H	7	O	4	0 G 5
C.CQCQ	-21.16	107.30	33.43	40.03	45.43	49.55	55.72	60.58	.00	10/12/95	ThmCal	C	3	H	7	O	4	0 G 6
CC.QCQ	-28.66	108.49	34.42	40.08	44.89	48.74	54.75	59.70	.00	10/12/95	ThmCal	C	3	H	7	O	4	0 G 6

CCQC.Q	-26.26	106.33	34.34	40.55	45.68	49.65	55.70	60.53	.00	10/12/95	ThmCal	C	3	H	7	O	4	0	G 6
QCC.CQ	-21.44	109.64	32.20	38.38	43.29	47.82	54.48	59.37	.00	10/12/95	ThmCal	C	3	H	7	O	4	0	G 6
Q.CCCQ	-31.69	105.42	31.65	37.87	42.84	47.12	53.51	58.85	.00	10/12/95	ThmCal	C	3	H	7	O	4	0	G 5
CQCCQ.	-31.69	105.42	31.65	37.87	42.84	47.12	53.51	58.85	.00	10/12/95	ThmCal	C	3	H	7	O	4	0	G 5
QCCC.Q	-21.89	108.32	33.84	39.87	44.73	48.91	55.05	60.03	.00	10/12/95	ThmCal	C	3	H	7	O	4	0	G 6
CCC	-25.33	64.50	17.88	22.63	27.05	30.93	37.11	41.88	49.36	10/12/95	ThmCal	C	3	H	8		0	0	G 2
C2COH	-65.50	74.52	21.48	26.82	31.72	35.54	41.87	46.55	54.11	10/12/95	ThmCal	C	3	H	8	O	1	0	G 3
CCCOH	-61.13	76.52	20.98	26.14	30.77	34.80	41.22	46.05	.00	10/12/95	ThmCal	C	3	H	8	O	1	0	G 3
COCC	-51.60	74.93	20.77	26.23	30.80	34.81	41.56	46.47	.00	10/12/95	ThmCal	C	3	H	8	O	1	0	G 3
CCOHC	-65.50	74.52	21.48	26.82	31.72	35.54	41.87	46.55	54.11	10/12/95	ThmCal	C	3	H	8	O	1	0	G 3
CCOC	-51.60	74.93	20.77	26.23	30.80	34.81	41.56	46.47	.00	10/12/95	ThmCal	C	3	H	8	O	1	0	G 3
CCCOOH	-46.56	85.54	25.79	31.67	36.72	41.07	47.67	52.85	.00	10/12/95	ThmCal	C	3	H	8	O	2	0	G 4
C2COOH	-49.93	82.79	26.29	32.35	37.67	41.81	48.32	53.35	.00	10/12/95	ThmCal	C	3	H	8	O	2	0	G 4
CCOOC	-42.46	85.40	25.17	31.15	36.30	40.69	47.80	53.47	.00	10/12/95	ThmCal	C	3	H	8	O	2	0	G 4
CCOOHC	-49.93	82.79	26.29	32.35	37.67	41.81	48.32	53.35	.00	10/12/95	ThmCal	C	3	H	8	O	2	0	G 4
COCOC	-81.83	81.86	24.68	30.03	34.45	38.53	45.93	51.08	.00	10/12/95	ThmCal	C	3	H	8	O	2	0	G 4
CCQCOH	-85.73	93.44	29.39	35.86	41.39	45.68	52.43	57.52	.00	10/12/95	ThmCal	C	3	H	8	O	3	0	G 5
CCOHCQ	-86.73	94.19	29.39	35.86	41.39	45.68	52.43	57.52	.00	10/12/95	ThmCal	C	3	H	8	O	3	0	G 5
CQCCOH	-82.36	96.18	28.89	35.18	40.44	44.94	51.78	57.02	.00	10/12/95	ThmCal	C	3	H	8	O	3	0	G 5
CCQCQ	-70.16	102.51	34.20	41.39	47.34	51.95	58.88	64.32	.00	10/12/95	ThmCal	C	3	H	8	O	4	0	G 6
QCCCQ	-65.79	104.50	33.70	40.71	46.39	51.21	58.23	63.82	.00	10/12/95	ThmCal	C	3	H	8	O	4	0	G 6
C#CC#C.	185.66	63.27	17.13	18.54	19.51	20.24	21.47	22.41	24.06	10/12/95	ThmCal	C	4	H	1		0	0	G 1
C#CC#C	105.06	59.78	17.64	20.10	21.78	23.02	24.94	26.38	28.84	10/12/95	ThmCal	C	4	H	2		0	0	G 1
C#CC*C.	127.27	68.16	17.22	20.93	22.89	25.79	28.41	30.35	33.40	10/12/95	ThmCal	C	4	H	3		0	0	G 1
C#CC.*C	117.57	67.48	17.60	20.92	22.74	25.70	28.53	30.67	34.07	10/12/95	ThmCal	C	4	H	3		0	0	G 1
C#CC*C	68.17	66.77	17.41	21.68	24.25	27.71	31.23	33.84	37.93	10/12/95	ThmCal	C	4	H	4		0	0	G 1
C.*CC*C.	144.28	69.38	18.74	22.80	25.80	28.00	31.20	33.78	37.50	10/12/95	ThmCal	C	4	H	4		0	0	G 1
C*C.C*C.	142.08	71.18	18.59	22.34	25.22	27.40	30.68	33.36	37.27	10/12/95	ThmCal	C	4	H	4		0	0	G 1
C*C*C*C	80.92	64.47	18.00	21.52	24.42	27.00	30.74	33.54	37.78	10/12/95	ThmCal	C	4	H	4		0	0	G 0
C#CC*C	68.17	66.77	17.41	21.68	24.25	27.71	31.23	33.84	37.93	10/12/95	ThmCal	C	4	H	4		0	0	G 1
C*CC.*C	75.48	68.69	19.31	23.54	27.01	29.83	34.14	37.59	42.70	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*C*CC.	74.95	68.05	18.73	23.07	26.64	29.67	34.20	37.53	42.60	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C#CCC.	88.55	72.99	18.78	22.51	25.72	28.43	32.79	36.10	42.32	10/12/95	ThmCal	C	4	H	5		0	0	G 2
C#CC.C	74.45	69.13	18.96	22.67	25.88	28.64	33.04	36.35	41.49	10/12/95	ThmCal	C	4	H	5		0	0	G 1
CC#CC.	72.00	69.45	17.80	21.47	24.86	27.81	32.54	36.07	42.55	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*C*CCJ	74.95	66.68	18.73	23.07	26.64	29.67	34.20	37.53	42.60	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*C.C*C	75.48	68.69	19.31	23.54	27.01	29.83	34.14	37.59	42.70	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C.*C*CC	73.75	71.10	18.90	22.58	25.78	28.64	33.06	36.39	41.51	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*C*C.C	73.75	71.10	18.90	22.58	25.78	28.64	33.06	36.39	41.51	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*CC*C.	85.18	69.37	18.93	23.55	27.16	29.92	34.02	37.27	42.03	10/12/95	ThmCal	C	4	H	5		0	0	G 1
C*CC.*C	75.48	68.69	19.31	23.54	27.01	29.83	34.14	37.59	42.70	10/12/95	ThmCal	C	4	H	5		0	0	G 1
CC*CC.*O	2.71	74.59	22.07	26.58	30.19	33.20	37.95	41.34	.00	10/12/95	ThmCal	C	4	H	5	O	1	0	G 1
C.C*CCHO	7.91	74.22	21.64	26.87	31.00	34.29	39.32	42.77	.00	10/12/95	ThmCal	C	4	H	5	O	1	0	G 1
C*C*CC	38.85	69.81	19.35	23.63	27.42	30.79	36.04	39.99	46.09	10/12/95	ThmCal	C	4	H	6		0	0	G 1
C*CC*C	26.08	66.60	19.12	24.30	28.52	31.84	36.84	40.76	46.56	10/12/95	ThmCal	C	4	H	6		0	0	G 1
C#CCC	39.55	69.58	19.55	23.87	27.63	30.83	35.95	39.84	46.98	10/12/95	ThmCal	C	4	H	6		0	0	G 2
CC#CC	34.70	67.78	18.64	22.64	26.42	29.76	35.24	39.38	47.86	10/12/95	ThmCal	C	4	H	6		0	0	G 2
C*C*CC	38.85	69.81	19.35	23.63	27.42	30.79	36.04	39.99	46.09	10/12/95	ThmCal	C	4	H	6		0	0	G 1

C*C*CC	38.85	69.81	19.35	23.63	27.42	30.79	36.04	39.99	46.09	10/12/95	ThmCal	C	4	H	6		0	0	G 1
C*CC*C	26.08	66.60	19.12	24.30	28.52	31.84	36.84	40.76	46.56	10/12/95	ThmCal	C	4	H	6		0	0	G 1
CC*CCHO	-28.19	75.98	22.26	27.43	31.78	35.41	41.16	45.23	.00	10/12/95	ThmCal	C	4	H	6	O	1	0	G 1
C*CCCHO	-18.54	80.68	20.99	26.06	30.54	34.28	40.22	44.52	.00	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C*CC2*O	-27.34	74.64	23.14	28.34	32.66	36.17	41.73	45.56	.00	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C2C*C*O	-28.06	73.97	24.08	28.58	33.01	36.14	41.27	45.33	51.36	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C2C*C*O	-28.06	73.97	24.08	28.58	33.01	36.14	41.27	45.33	51.36	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C*CICC*O	-27.34	74.64	23.14	28.34	32.66	36.17	41.73	45.56	.00	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C*CCC *O	-30.40	76.82	21.21	26.31	30.83	34.59	40.44	44.92	.00	10/12/95	ThmCal	C	4	H	6	O	1	0	G 2
C*CC2.	32.30	69.61	20.96	26.09	30.52	34.22	40.07	44.43	51.22	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C2.C*C	32.30	69.61	20.96	26.09	30.52	34.22	40.07	44.43	51.22	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C*CC.C	33.39	69.80	19.03	24.27	28.96	32.96	39.21	43.83	50.90	10/12/95	ThmCal	C	4	H	7		0	0	G 1
CC*CC.	32.88	70.64	20.08	25.18	29.64	33.46	39.50	43.98	50.91	10/12/95	ThmCal	C	4	H	7		0	0	G 2
CC.*CC	53.68	74.20	20.36	24.53	28.48	32.06	38.00	42.53	49.64	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C*C.CC	56.79	75.42	20.23	24.88	29.10	32.76	38.62	43.06	49.99	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C*CCC.	48.89	77.02	19.80	24.73	29.13	32.88	38.80	43.23	50.09	10/12/95	ThmCal	C	4	H	7		0	0	G 2
CC.*CC	53.68	74.20	20.36	24.53	28.48	32.06	38.00	42.53	49.64	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C.*CCC	58.99	75.00	20.38	25.34	29.68	33.36	39.14	43.48	50.22	10/12/95	ThmCal	C	4	H	7		0	0	G 2
C2C*C.	55.30	72.75	21.39	25.90	29.94	33.42	39.09	43.40	50.18	10/12/95	ThmCal	C	4	H	7		0	0	G 2
CC.*CC	53.68	74.20	20.36	24.53	28.48	32.06	38.00	42.53	49.64	10/12/95	ThmCal	C	4	H	7		0	0	G 2
CC*CCO.	14.28	81.58	22.95	27.96	32.55	36.62	43.18	47.96	55.07	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C*CCCO.	16.05	82.79	22.69	28.30	33.15	37.26	43.69	48.34	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C*CCO.C	12.71	80.32	23.24	29.25	34.38	38.30	44.62	49.10	55.26	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C*CC2O.	13.70	80.55	23.83	28.87	33.43	37.38	43.75	48.41	55.38	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C2CC.*O	-14.30	80.77	22.74	28.03	33.00	37.03	43.54	48.20	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C2C.C*O	-13.50	77.94	24.19	29.26	33.73	37.63	43.83	48.46	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C2C*CO.	6.03	77.59	24.55	29.95	34.64	38.53	44.58	48.99	55.83	10/12/95	ThmCal	C	4	H	7	O	1	0	G 0
C*CC2O.	13.70	80.55	23.83	28.87	33.43	37.38	43.75	48.41	55.38	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C*C(C.)OC	1.36	79.69	24.36	30.35	35.17	39.10	45.44	50.22	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 2
C*C(C)OC.	9.66	83.59	25.50	30.83	35.15	38.77	44.78	49.41	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 3
C*CCOOC.	35.97	95.23	28.81	34.67	39.54	43.83	49.52	53.56	.00	10/12/95	ThmCal	C	4	H	7	O	1	0	G 3
C*CCCOO	14.76	93.49	26.43	32.29	37.16	41.33	47.80	52.97	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
C.*CCCOOH	37.76	94.66	28.29	34.38	39.35	43.50	49.70	54.45	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C*CC.COOH	12.16	89.46	26.94	33.31	38.63	43.10	49.77	54.80	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
C*CCC.OOH	22.56	97.09	28.62	34.29	39.05	43.12	49.34	54.15	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C*CC2OO.	11.72	91.69	28.18	33.32	37.67	41.70	47.54	52.54	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
C*CIC.CQ	11.72	89.71	29.61	35.60	40.44	44.67	50.42	55.05	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C*CICC.Q	9.12	87.66	28.69	34.34	39.14	43.47	49.51	54.37	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C.*CC2OOH	34.72	92.86	30.04	35.41	39.86	43.87	49.44	54.02	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
CC*CC.Q	9.70	88.69	27.81	33.43	38.26	42.71	48.94	53.92	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
CC*CCOO.	12.30	92.72	27.30	32.41	36.79	40.94	46.97	52.09	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C*CCOO.C	12.42	90.27	26.98	33.24	38.39	42.37	48.73	53.73	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C*CCQC.	25.32	93.46	28.26	34.72	40.03	44.06	50.29	54.96	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
C.*CCQC	35.42	91.44	28.84	35.33	40.58	44.54	50.63	55.21	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4
C*CC.QC	7.62	86.36	27.24	33.70	39.20	43.49	50.17	55.15	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
C.C*CCQ	12.30	90.74	28.73	34.69	39.56	43.91	49.85	54.60	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
CC*CCOO	12.30	92.72	27.30	32.41	36.79	40.94	46.97	52.09	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 3
CC.*CCQ	33.10	94.31	29.01	34.04	38.40	42.51	48.35	53.15	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G 4

CC*C.CQ	33.10	94.31	29.01	34.04	38.40	42.51	48.35	53.15	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CC*OCCO.	-40.72	90.25	26.29	31.91	36.70	40.82	47.76	52.72	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
CC*OC.COH	-52.88	89.34	25.91	31.64	36.58	40.82	47.48	52.41	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C.C*OCCOH	-50.38	88.98	26.97	32.81	37.53	41.48	48.02	52.67	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CC*OCC.OH	-50.78	92.49	27.60	32.91	37.29	41.04	47.47	52.13	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
O*CCC2O.	-35.04	88.84	25.69	31.67	37.07	41.43	48.43	53.30	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
O*CCM.COH	-38.00	93.71	25.90	31.61	36.77	40.92	47.65	52.36	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
O*C.CC2OH	-50.10	91.42	25.84	31.54	36.72	40.90	47.65	52.37	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CCO.CCHO	-37.64	89.34	27.54	33.25	38.23	42.04	48.77	53.38	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
CCOHCC.*O	-52.70	91.92	27.69	33.12	37.88	41.51	47.99	52.45	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
O*C.CCOHC	-52.70	91.92	27.69	33.12	37.88	41.51	47.99	52.45	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
O*CC.COHC	-49.80	88.43	27.16	32.98	38.11	42.04	48.49	53.07	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
O*CCCO.C	-37.64	89.34	27.54	33.25	38.23	42.04	48.77	53.38	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
O*CCCOHC.	-40.60	94.21	27.75	33.19	37.93	41.53	47.99	52.44	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CCICO.C*O	-35.04	88.84	25.69	31.67	37.07	41.43	48.43	53.30	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
C*CICOOC.	30.00	95.42	29.09	34.87	39.37	43.28	50.00	54.82	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
C*CC2.OOH	9.12	88.46	28.69	34.34	39.14	43.47	49.51	54.37	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C*CC2.Q	9.12	87.66	28.69	34.34	39.14	43.47	49.51	54.37	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	2
C*CCOO.C	12.42	90.27	26.98	33.24	38.39	42.37	48.73	53.73	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C.*CCQC	35.42	91.44	28.84	35.33	40.58	44.54	50.63	55.21	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C*CC.QC	7.62	86.36	27.24	33.70	39.20	43.49	50.17	55.15	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CC*OCCO.	-40.72	90.25	26.29	31.91	36.70	40.82	47.76	52.72	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	3
CC*OCC.OH	-48.28	92.64	28.55	34.14	38.51	42.16	48.31	52.75	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C.*CCCOOH	38.76	94.31	28.29	34.38	39.35	43.50	49.70	54.45	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
C.*CC2OOH	36.41	92.07	29.43	34.95	39.63	43.62	49.76	54.52	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	2
C.*CCQC	35.42	91.84	28.84	35.33	40.58	44.54	50.63	55.21	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	5
CC.*CCQ	34.79	93.52	28.40	33.58	38.17	42.26	48.67	53.65	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CC*C.CQ	34.79	93.52	28.40	33.58	38.17	42.26	48.67	53.65	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	4
CYCC.CCOO	8.43	79.37	22.16	29.60	36.54	42.37	50.48	56.45	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	0
CYCCCC.OO	5.98	78.75	23.80	31.09	37.98	43.46	51.05	57.11	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	0
C.CYCCOOC	14.64	82.06	23.89	31.38	38.32	43.75	51.14	56.96	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
CCYCC.OOC	7.14	84.63	24.88	31.43	37.78	42.94	50.17	56.08	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
CCYC.COOC	10.04	83.89	23.88	30.26	36.68	42.00	49.55	55.68	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
CCYCC.COO	9.86	83.78	23.93	31.15	38.01	43.36	51.01	56.61	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
CCYCCC.OO	7.41	83.16	25.57	32.64	39.45	44.45	51.58	57.27	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
CCYC.CCOO	5.01	85.32	25.65	32.17	38.66	43.54	50.63	56.44	.00	10/12/95	ThmCal	C	4	H	7	O	2	0	G	1
C2CQ.C*O.	-39.73	93.49	30.46	36.78	42.27	46.93	53.91	59.14	.00	10/12/95	ThmCal	C	4	H	7	O	3	0	G	4
C2CQC.*O	-38.93	94.39	31.68	38.19	43.86	48.60	55.47	60.38	.00	10/12/95	ThmCal	C	4	H	7	O	3	0	G	4
C2C.OOC*O	-43.23	105.60	31.36	37.27	42.55	46.54	53.72	58.99	.00	10/12/95	ThmCal	C	4	H	7	O	3	0	G	4
C2.CQC*O	-26.43	98.92	31.86	38.34	43.94	48.64	55.47	60.36	.00	10/12/95	ThmCal	C	4	H	7	O	3	0	G	5
C*CICCO.Q	-17.87	96.02	33.07	39.14	44.69	48.94	55.31	60.39	.00	10/12/95	ThmCal	C	4	H	7	O	3	0	G	4
CCYQCOOC	-25.66	92.76	30.46	39.02	46.24	51.92	60.15	66.70	.00	10/12/95	ThmCal	C	4	H	7	O	4	0	G	2
C.CYQCOOC	-12.76	95.96	31.74	40.50	47.88	53.61	61.71	67.93	.00	10/12/95	ThmCal	C	4	H	7	O	4	0	G	3
CCYQC.OOC	-19.86	95.07	32.38	41.19	48.48	54.12	62.05	68.18	.00	10/12/95	ThmCal	C	4	H	7	O	4	0	G	3
C*CC2	-3.80	69.99	21.58	26.65	31.30	35.34	41.91	46.89	54.71	10/12/95	ThmCal	C	4	H	8		0	0	G	2
C2C*C	-3.80	69.99	21.58	26.65	31.30	35.34	41.91	46.89	54.71	10/12/95	ThmCal	C	4	H	8		0	0	G	2
C*CCC	-.11	73.61	20.57	26.09	31.04	35.28	41.96	46.97	54.75	10/12/95	ThmCal	C	4	H	8		0	0	G	2
CC*CC	-3.22	71.02	20.70	25.74	30.42	34.58	41.34	46.44	54.40	10/12/95	ThmCal	C	4	H	8		0	0	G	2

C*CCC	-.11	73.61	20.57	26.09	31.04	35.28	41.96	46.97	54.75	10/12/95	ThmCal	C	4	H	8		0	0	G 2
CC*CC	-3.22	71.02	20.70	25.74	30.42	34.58	41.34	46.44	54.40	10/12/95	ThmCal	C	4	H	8		0	0	G 2
C2C*C	-3.80	69.99	21.58	26.65	31.30	35.34	41.91	46.89	54.71	10/12/95	ThmCal	C	4	H	8		0	0	G 2
C*CMOC	-34.74	81.44	24.98	30.91	35.95	40.22	47.28	52.68	.00	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
CC*CCOH	-37.68	83.04	23.93	29.26	34.16	38.51	45.56	50.76	58.66	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C*CCCOH	-35.91	84.25	23.67	29.60	34.76	39.15	46.07	51.14	.00	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C*CCOHC	-39.25	81.78	24.22	30.55	35.99	40.19	47.00	51.90	58.85	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C*CC2OH	-38.26	82.01	24.81	30.17	35.04	39.27	46.13	51.21	58.97	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C2CC*O	-51.20	79.65	23.57	29.46	34.96	39.45	46.70	51.93	.00	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C2C*COH	-45.93	79.05	25.53	31.25	36.25	40.42	46.96	51.79	59.42	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C2CYC2O	-31.48	72.61	23.30	30.11	36.00	40.70	47.91	53.05	.00	10/12/95	ThmCal	C	4	H	8	O	1	0	G 2
C*C(C)OC	-34.74	81.44	24.98	30.91	35.95	40.22	47.28	52.68	.00	10/12/95	ThmCal	C	4	H	8	O	1	0	G 3
C*CCOOC	-20.13	111.45	35.83	43.05	49.10	54.55	61.14	65.65	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
C*CCCOOH	-21.34	93.27	28.48	35.13	40.71	45.42	52.52	57.94	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CCOHCCHO	-89.60	90.80	28.52	34.55	39.84	43.93	51.15	56.18	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
O*CCCOHC	-89.60	90.80	28.52	34.55	39.84	43.93	51.15	56.18	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
O*CCC2OH	-87.00	90.30	26.67	32.97	38.68	43.32	50.81	56.10	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CC*CCQ	-23.80	92.50	29.35	35.25	40.34	45.03	51.69	57.06	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CC*OCCOH	-92.68	91.71	27.27	33.21	38.31	42.71	50.14	55.52	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
C*CC2OOH	-24.38	91.47	30.23	36.16	41.22	45.79	52.26	57.51	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
C*CCQC	-23.68	90.05	29.03	36.08	41.94	46.46	53.45	58.70	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CC*OCCOH	-92.68	91.71	27.27	33.21	38.31	42.71	50.14	55.52	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CCICOCOH	-87.00	90.30	26.67	32.97	38.68	43.32	50.81	56.10	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CCICOC*O	-87.00	90.30	26.67	32.97	38.68	43.32	50.81	56.10	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
C*CICOOC	-16.40	92.12	28.47	34.78	39.90	44.37	52.01	57.57	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
C2C*CQ	-23.92	88.63	29.78	35.94	41.37	45.92	53.01	58.09	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CC*OCCOH	-92.68	91.71	27.27	33.21	38.31	42.71	50.14	55.52	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 4
CYCCCCOO	-37.92	73.55	23.66	31.93	39.64	45.76	54.23	60.90	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 0
CCYCCOOC	-34.36	78.65	24.66	32.74	40.23	46.15	54.30	60.70	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 1
CCYCCCOO	-36.49	79.34	25.43	33.48	41.11	46.75	54.76	61.06	.00	10/12/95	ThmCal	C	4	H	8	O	2	0	G 1
C2CQC*O	-75.43	93.97	32.51	39.62	45.82	51.02	58.63	64.11	.00	10/12/95	ThmCal	C	4	H	8	O	3	0	G 5
C2COOC*O	-84.73	100.02	31.14	38.58	45.00	49.75	57.85	63.61	.00	10/12/95	ThmCal	C	4	H	8	O	3	0	G 5
C*CICCOHQ	-69.83	97.48	34.05	40.44	46.30	50.83	57.69	63.19	.00	10/12/95	ThmCal	C	4	H	8	O	3	0	G 5
CCYQCCOOC	-61.76	92.54	32.51	41.86	49.79	56.01	64.87	71.67	.00	10/12/95	ThmCal	C	4	H	8	O	4	0	G 3
C3C.	11.90	74.29	22.33	27.04	31.82	36.27	43.62	49.34	58.53	10/12/95	ThmCal	C	4	H	9		0	0	G 3
C3.CH	16.50	77.40	22.34	28.16	33.46	38.02	45.21	50.62	59.26	10/12/95	ThmCal	C	4	H	9		0	0	G 3
C2CC.	16.50	77.40	22.34	28.16	33.46	38.02	45.21	50.62	59.26	10/12/95	ThmCal	C	4	H	9		0	0	G 3
CCCC.	18.74	78.71	22.61	28.22	33.39	37.88	45.02	50.48	58.90	10/12/95	ThmCal	C	4	H	9		0	0	G 3
CC.CC	16.09	79.73	21.88	27.25	32.20	36.89	44.43	49.77	58.36	10/12/95	ThmCal	C	4	H	9		0	0	G 3
CCC.C	16.09	79.73	21.88	27.25	32.20	36.89	44.43	49.77	58.36	10/12/95	ThmCal	C	4	H	9		0	0	G 3
C3.COH	-26.10	84.98	26.43	32.85	38.36	42.90	50.12	55.42	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
CCCCO	-14.10	84.48	25.50	31.79	37.41	42.26	49.91	55.59	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
CCO.CC	-18.47	82.48	26.00	32.47	38.36	43.00	50.56	56.09	64.72	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
C2CCO	-16.34	81.79	25.23	31.73	37.48	42.40	50.10	55.73	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
C3CO	-23.14	76.55	26.22	32.91	38.66	43.41	50.90	56.36	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
C2CCO.	-16.34	81.79	25.23	31.73	37.48	42.40	50.10	55.73	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
C3.COH	-26.10	84.98	26.43	32.85	38.36	42.90	50.12	55.42	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
C2C.COH	-23.90	88.49	25.43	30.55	35.54	40.14	47.73	53.51	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4

C3.COH	-26.10	83.61	26.43	32.85	38.36	42.90	50.12	55.42	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
C3CO.	-23.14	76.55	26.22	32.91	38.66	43.41	50.90	56.36	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 3
C2C.OC	-16.50	83.70	27.29	33.78	39.20	43.45	50.78	56.04	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
C2COC.	-16.50	84.51	27.29	33.78	39.20	43.45	50.78	56.04	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
C2.COC	-11.90	85.78	26.00	32.50	38.09	42.50	50.12	55.57	.00	10/12/95	ThmCal	C	4	H	9	O	1	0	G 4
C3.COOH	-10.93	92.55	31.24	38.38	44.31	49.17	56.57	62.22	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
C.COOCC	3.12	100.05	28.81	35.47	41.48	46.87	54.39	59.74	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C3COO	-23.83	85.80	29.96	36.90	42.67	47.48	55.01	60.99	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
CCCCOO	-15.39	95.18	29.24	35.78	41.42	46.33	54.02	60.22	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
CCCC.OOH	-7.59	98.78	31.43	37.78	43.31	48.12	55.56	61.40	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
CCC.COOH	-5.14	99.40	29.79	36.29	41.87	47.03	54.99	60.74	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
CC.CCOOH	-5.14	99.40	29.79	36.29	41.87	47.03	54.99	60.74	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C.CCCOOH	-2.49	99.75	30.52	37.26	43.06	48.02	55.58	61.45	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C2CCOO	-17.63	92.49	28.97	35.72	41.49	46.47	54.21	60.36	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
C2CC.OOH	-9.83	96.09	31.16	37.72	43.38	48.26	55.75	61.54	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C2.CCOOH	-4.73	97.07	30.25	37.20	43.13	48.16	55.77	61.59	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C2C.COOH	-9.33	97.51	30.24	36.08	41.49	46.41	54.18	60.31	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
CCOO.CC	-18.76	92.43	29.74	36.46	42.37	47.07	54.67	60.72	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
CCOOHCC.	-5.86	97.01	31.02	37.94	44.01	48.76	56.23	61.95	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
CCOOHC.C	-8.51	96.65	30.29	36.97	42.82	47.77	55.64	61.24	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
CC.OOHCC	-13.36	98.19	32.01	37.99	43.47	47.95	55.26	61.07	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C.COOHCC	-5.86	97.01	31.02	37.94	44.01	48.76	56.23	61.95	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C2C.CQ	-9.53	96.86	30.53	36.44	41.82	46.68	54.40	60.46	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 5
C2CC.Q	-10.93	91.72	31.53	38.50	44.22	48.97	56.26	61.92	.00	10/12/95	ThmCal	C	4	H	9	O	2	0	G 4
C2CO.CQ	-44.37	98.39	34.13	41.95	48.33	53.55	61.46	67.33	.00	10/12/95	ThmCal	C	4	H	9	O	3	0	G 5
C2CQ.COH	-59.23	99.32	33.06	40.41	46.39	51.35	59.12	65.16	.00	10/12/95	ThmCal	C	4	H	9	O	3	0	G 5
C2CQCO.	-43.37	97.64	34.13	41.95	48.33	53.55	61.46	67.33	.00	10/12/95	ThmCal	C	4	H	9	O	3	0	G 5
C2CO.CQ	-44.37	98.39	34.13	41.95	48.33	53.55	61.46	67.33	.00	10/12/95	ThmCal	C	4	H	9	O	3	0	G 5
C2COHCQ.	-60.23	100.07	33.06	40.41	46.39	51.35	59.12	65.16	.00	10/12/95	ThmCal	C	4	H	9	O	3	0	G 5
C3CH	-32.50	70.43	23.11	29.52	35.37	40.42	48.37	54.36	63.92	10/12/95	ThmCal	C	4	H	10		0	0	G 3
C3C	-32.50	70.43	23.11	29.52	35.37	40.42	48.37	54.36	63.92	10/12/95	ThmCal	C	4	H	10		0	0	G 3
CCCC	-30.26	73.92	23.38	29.58	35.30	40.28	48.18	54.22	63.56	10/12/95	ThmCal	C	4	H	10		0	0	G 3
C2COC	-60.90	82.36	26.77	33.86	40.00	44.90	53.28	59.31	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
CCCCOH	-66.06	85.94	26.48	33.09	39.02	44.15	52.29	58.39	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
CCOHCC	-70.43	83.94	26.98	33.77	39.97	44.89	52.94	58.89	68.31	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
C3COH	-75.10	78.01	27.20	34.21	40.27	45.30	53.28	59.16	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
C2CCOH	-68.30	83.25	26.21	33.03	39.09	44.29	52.48	58.53	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
C3COH	-75.10	78.01	27.20	34.21	40.27	45.30	53.28	59.16	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
C2COC	-60.90	82.36	26.77	33.86	40.00	44.90	53.28	59.31	.00	10/12/95	ThmCal	C	4	H	10	O	1	0	G 4
C3COOH	-59.93	85.58	32.01	39.74	46.22	51.57	59.73	65.96	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
CCOOCC	-45.88	95.25	29.58	36.83	43.39	49.27	57.55	63.48	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
CCCCOOH	-51.49	94.96	31.29	38.62	44.97	50.42	58.74	65.19	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
CCOOHCC	-54.86	93.58	31.79	39.30	45.92	51.16	59.39	65.69	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
C2CCOOH	-53.73	92.27	31.02	38.56	45.04	50.56	58.93	65.33	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
C2CCQ	-53.73	92.27	31.02	38.56	45.04	50.56	58.93	65.33	.00	10/12/95	ThmCal	C	4	H	10	O	2	0	G 5
C2CQCOH	-95.33	99.10	35.11	43.25	49.94	55.44	63.84	70.13	.00	10/12/95	ThmCal	C	4	H	10	O	3	0	G 6
C2COHCQ	-96.33	99.85	35.11	43.25	49.94	55.44	63.84	70.13	.00	10/12/95	ThmCal	C	4	H	10	O	3	0	G 6
C#CC.C#C	123.06	69.23	19.09	23.15	26.33	28.84	32.59	35.20	41.34	10/12/95	ThmCal	C	5	H	3		0	0	G 1

C#CC#CC	95.48	71.04	21.68	25.43	28.50	31.03	35.09	38.11	43.92	10/12/95	ThmCal	C	5	H	4		0	0	G 2
C#CC*C*C	102.37	72.77	21.31	26.08	28.95	32.71	36.53	39.34	43.63	10/12/95	ThmCal	C	5	H	4		0	0	G 1
C#CC#C	99.16	71.90	21.22	25.11	28.21	30.73	34.79	37.80	44.55	10/12/95	ThmCal	C	5	H	4		0	0	G 2
C#CC.C*C	88.97	73.28	20.07	25.37	29.74	33.34	38.60	42.33	49.11	10/12/95	ThmCal	C	5	H	5		0	0	G 2
C#CCC.*C	121.50	78.74	22.28	26.21	29.61	32.52	37.31	40.88	47.45	10/12/95	ThmCal	C	5	H	5		0	0	G 2
C#CCC*C.	123.70	78.32	22.43	26.67	30.19	33.12	37.83	41.30	47.68	10/12/95	ThmCal	C	5	H	5		0	0	G 2
CY13PD5	57.16	63.57	18.43	24.69	29.65	33.48	38.95	42.69	48.22	10/12/95	ThmCal	C	5	H	5		0	0	G 0
C#CC*CC	60.30	75.36	22.66	28.19	31.95	36.50	41.83	45.79	51.94	10/12/95	ThmCal	C	5	H	6		0	0	G 2
C#CC*C	65.07	77.33	22.20	27.33	31.62	35.23	40.80	44.93	52.32	10/12/95	ThmCal	C	5	H	6		0	0	G 2
C#CCC*C	64.60	76.93	22.62	27.42	31.55	35.04	40.65	44.79	52.21	10/12/95	ThmCal	C	5	H	6		0	0	G 2
CY13PD	31.26	65.49	18.14	24.69	30.22	34.66	41.21	45.78	52.56	10/12/95	ThmCal	C	5	H	6		0	0	G 0
C*CCJC*C	49.31	75.93	21.09	27.62	33.16	37.71	44.54	49.44	56.81	10/12/95	ThmCal	C	5	H	7		0	0	G 2
C*CCC*C.	84.51	82.75	23.03	28.83	33.68	37.68	43.92	48.55	55.49	10/12/95	ThmCal	C	5	H	7		0	0	G 2
C*CCC.*C	82.31	83.17	22.88	28.37	33.10	37.08	43.40	48.13	55.26	10/12/95	ThmCal	C	5	H	7		0	0	G 2
C*CC.C*C	49.31	75.93	21.09	27.62	33.16	37.71	44.54	49.44	56.81	10/12/95	ThmCal	C	5	H	7		0	0	G 2
C*CCC*C	25.41	78.61	23.22	29.58	35.04	39.60	46.74	52.04	60.02	10/12/95	ThmCal	C	5	H	8		0	0	G 2
C*CCC*C	25.41	79.98	23.22	29.58	35.04	39.60	46.74	52.04	60.02	10/12/95	ThmCal	C	5	H	8		0	0	G 2
C5DE23	30.98	77.02	24.60	30.14	35.12	39.58	46.64	51.94	60.10	10/12/95	ThmCal	C	5	H	8		0	0	G 1
CYC5E	7.82	69.24	19.56	26.79	33.22	38.57	46.73	52.57	61.31	10/12/95	ThmCal	C	5	H	8		0	0	G 0
C5DE14	25.41	79.98	23.22	29.58	35.04	39.60	46.74	52.04	60.02	10/12/95	ThmCal	C	5	H	8		0	0	G 2
C*CC.CC	28.46	79.22	24.53	31.22	37.21	42.31	50.28	56.17	65.10	10/12/95	ThmCal	C	5	H	9		0	0	G 3
C*CCCC.	43.96	86.44	25.30	31.68	37.38	42.23	49.87	55.57	64.29	10/12/95	ThmCal	C	5	H	9		0	0	G 3
C*CCC.C	41.31	87.47	24.57	30.71	36.19	41.24	49.28	54.86	63.75	10/12/95	ThmCal	C	5	H	9		0	0	G 3
C*CC.CC	28.46	79.22	24.53	31.22	37.21	42.31	50.28	56.17	65.10	10/12/95	ThmCal	C	5	H	9		0	0	G 2
C*C.CCC	54.06	84.42	25.88	32.29	37.93	42.71	50.21	55.82	64.42	10/12/95	ThmCal	C	5	H	9		0	0	G 3
C.*CCCC	54.06	84.42	25.88	32.29	37.93	42.71	50.21	55.82	64.42	10/12/95	ThmCal	C	5	H	9		0	0	G 3
C*C.CCC	51.86	84.84	25.73	31.83	37.35	42.11	49.69	55.40	64.19	10/12/95	ThmCal	C	5	H	9		0	0	G 2
C.*CCCC	54.06	84.42	25.88	32.29	37.93	42.71	50.21	55.82	64.42	10/12/95	ThmCal	C	5	H	9		0	0	G 2
C*CCCCQ.	9.83	102.91	31.93	39.24	45.41	50.68	58.87	65.31	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 4
C*CCCC.Q	17.63	106.51	34.12	41.24	47.30	52.47	60.41	66.49	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C*CCC.CQ	20.08	107.13	32.48	39.75	45.86	51.38	59.84	65.83	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C*CC.CCQ	7.23	98.88	32.44	40.26	46.88	52.45	60.84	67.14	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 4
C.*CCCCQ	32.83	104.08	33.79	41.33	47.60	52.85	60.77	66.79	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C*C.CCCQ	30.63	104.50	33.64	40.87	47.02	52.25	60.25	66.37	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C*CCQ.CC	7.49	99.69	32.48	40.19	46.64	51.72	59.80	66.07	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 4
C*CCQCC.	20.39	102.88	33.76	41.67	48.28	53.41	61.36	67.30	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C*CCQC.C	17.74	103.91	33.03	40.70	47.09	52.42	60.77	66.59	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C.*CCQCC	30.49	100.86	34.34	42.28	48.83	53.89	61.70	67.55	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C.*CCCCQ	33.83	103.73	33.79	41.33	47.60	52.85	60.77	66.79	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 2
C*C.CCCQ	30.63	104.50	33.64	40.87	47.02	52.25	60.25	66.37	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 5
C.*CCQCC	30.49	101.26	34.34	42.28	48.83	53.89	61.70	67.55	.00	10/12/95	ThmCal	C	5	H	9	O	2	0	G 3
2ME1BUTE	-8.56	81.16	26.70	33.51	39.62	44.83	53.13	59.37	69.07	10/12/95	ThmCal	C	5	H	10		0	0	G 3
2ME2BUTE	-11.67	79.95	26.83	33.16	39.00	44.13	52.51	58.84	68.72	10/12/95	ThmCal	C	5	H	10		0	0	G 3
3ME1BUTE	-7.03	80.34	25.80	32.98	39.46	44.77	53.22	59.45	69.25	10/12/95	ThmCal	C	5	H	10		0	0	G 3
C*CCCC	-5.04	83.03	26.07	33.04	39.29	44.63	53.03	59.31	68.95	10/12/95	ThmCal	C	5	H	10		0	0	G 3
C*CCCC	-5.04	83.03	26.07	33.04	39.29	44.63	53.03	59.31	68.95	10/12/95	ThmCal	C	5	H	10		0	0	G 3
C*CCCCQ	-26.27	102.69	33.98	42.08	48.96	54.77	63.59	70.28	.00	10/12/95	ThmCal	C	5	H	10	O	2	0	G 5
C*CCQCC	-28.61	98.45	34.53	43.03	50.19	55.81	64.52	71.04	.00	10/12/95	ThmCal	C	5	H	10	O	2	0	G 3

C3CC.	8.70	79.04	28.36	36.13	43.05	48.88	57.69	64.10	73.78	10/12/95	ThmCal	C	5	H	11		0	0 G 4
CCCCC.	13.81	88.13	28.11	35.17	41.64	47.23	56.09	62.82	73.10	10/12/95	ThmCal	C	5	H	11		0	0 G 4
CCCC.C	11.16	89.15	27.38	34.20	40.45	46.24	55.50	62.11	72.56	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C.CCCC	13.81	88.13	28.11	35.17	41.64	47.23	56.09	62.82	73.10	10/12/95	ThmCal	C	5	H	11		0	0 G 4
CC.CCC	11.16	89.15	27.38	34.20	40.45	46.24	55.50	62.11	72.56	10/12/95	ThmCal	C	5	H	11		0	0 G 4
CCC.CC	11.16	87.78	27.38	34.20	40.45	46.24	55.50	62.11	72.56	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C2.CCC	11.57	85.45	27.84	35.11	41.71	47.37	56.28	62.96	73.46	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C2C.CC	6.97	85.89	27.83	33.99	40.07	45.62	54.69	61.68	72.73	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C2CC.C	8.92	86.47	27.11	34.14	40.52	46.38	55.69	62.25	72.92	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C2CCC.	11.57	85.45	27.84	35.11	41.71	47.37	56.28	62.96	73.46	10/12/95	ThmCal	C	5	H	11		0	0 G 4
CC4.	8.70	79.04	28.36	36.13	43.05	48.88	57.69	64.10	73.78	10/12/95	ThmCal	C	5	H	11		0	0 G 4
C3CCO.	-24.14	84.81	31.25	39.70	47.07	53.26	62.58	69.21	.00	10/12/95	ThmCal	C	5	H	11	O	1	0 G 4
C3.CCOH	-27.10	93.24	31.46	39.64	46.77	52.75	61.80	68.27	.00	10/12/95	ThmCal	C	5	H	11	O	1	0 G 5
C3.COC	-21.50	91.44	31.72	39.89	46.64	52.26	61.53	68.18	.00	10/12/95	ThmCal	C	5	H	11	O	1	0 G 5
C3COC.	-26.10	87.99	33.01	41.17	47.75	53.21	62.19	68.65	.00	10/12/95	ThmCal	C	5	H	11	O	1	0 G 5
C3.COOH	-12.53	100.89	36.27	45.17	52.72	59.02	68.25	75.07	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C3CCOO	-25.43	95.51	34.99	43.69	51.08	57.33	66.69	73.84	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
C2Q.CCC	-22.56	101.91	34.47	42.67	49.74	55.82	65.28	72.70	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
CQCIC.CC	-9.66	105.11	35.75	44.15	51.38	57.51	66.84	73.93	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2QC.CC	-14.26	106.93	35.74	43.03	49.74	55.76	65.25	72.65	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2QCC.C	-12.31	106.13	35.02	43.18	50.19	56.52	66.25	73.22	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2QCCC.	-9.66	105.11	35.75	44.15	51.38	57.51	66.84	73.93	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CQ.CC	-28.36	98.10	35.46	43.85	50.92	56.83	66.08	73.33	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
C2.CQCC	-15.46	101.30	36.74	45.33	52.56	58.52	67.64	74.56	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CQC.C	-18.11	102.32	36.01	44.36	51.37	57.53	67.05	73.85	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CQCC.	-15.46	101.30	36.74	45.33	52.56	58.52	67.64	74.56	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CCQ.C	-25.93	99.17	34.97	43.35	50.69	56.56	65.93	73.20	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
C2.CCQC	-13.03	102.37	36.25	44.83	52.33	58.25	67.49	74.43	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2C.CQC	-17.63	104.19	36.24	43.71	50.69	56.50	65.90	73.15	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CCQC.	-13.03	102.37	36.25	44.83	52.33	58.25	67.49	74.43	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CCCQ.	-22.56	101.91	34.47	42.67	49.74	55.82	65.28	72.70	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
C2.CCCQ	-9.66	105.11	35.75	44.15	51.38	57.51	66.84	73.93	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2C.CCQ	-14.26	106.93	35.74	43.03	49.74	55.76	65.25	72.65	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C2CC.CQ	-12.31	106.13	35.02	43.18	50.19	56.52	66.25	73.22	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
CC4Q.	-25.43	95.51	34.99	43.69	51.08	57.33	66.69	73.84	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 5
CC4.Q	-12.53	100.89	36.27	45.17	52.72	59.02	68.25	75.07	.00	10/12/95	ThmCal	C	5	H	11	O	2	0 G 6
C4C	-40.30	72.87	29.13	37.49	44.96	51.28	60.85	67.84	78.44	10/12/95	ThmCal	C	5	H	12		0	0 G 4
CCCCC	-35.19	83.34	28.88	36.53	43.55	49.63	59.25	66.56	77.76	10/12/95	ThmCal	C	5	H	12		0	0 G 4
CCICCC	-35.83	82.03	28.61	36.47	43.62	49.77	59.44	66.70	.00	10/12/95	ThmCal	C	5	H	12		0	0 G 4
C2CCC	-37.43	82.03	28.61	36.47	43.62	49.77	59.44	66.70	78.12	10/12/95	ThmCal	C	5	H	12		0	0 G 4
CC4	-40.30	72.87	29.13	37.49	44.96	51.28	60.85	67.84	78.44	10/12/95	ThmCal	C	5	H	12		0	0 G 4
C3CCOH	-76.10	86.27	32.23	41.00	48.68	55.15	64.96	72.01	.00	10/12/95	ThmCal	C	5	H	12	O	1	0 G 5
C3COC	-70.50	85.84	32.49	41.25	48.55	54.66	64.69	71.92	.00	10/12/95	ThmCal	C	5	H	12	O	1	0 G 5
C3CCOOH	-61.53	95.29	37.04	46.53	54.63	61.42	71.41	78.81	.00	10/12/95	ThmCal	C	5	H	12	O	2	0 G 6
C2QCCC	-58.66	101.69	36.52	45.51	53.29	59.91	70.00	77.67	.00	10/12/95	ThmCal	C	5	H	12	O	2	0 G 6
C2CQCC	-65.46	98.63	37.51	46.69	54.47	60.92	70.80	78.30	.00	10/12/95	ThmCal	C	5	H	12	O	2	0 G 6
C2CCQC	-62.03	98.95	37.02	46.19	54.24	60.65	70.65	78.17	.00	10/12/95	ThmCal	C	5	H	12	O	2	0 G 6
C2CCCQ	-58.66	101.69	36.52	45.51	53.29	59.91	70.00	77.67	.00	10/12/95	ThmCal	C	5	H	12	O	2	0 G 6

CC4Q	-61.53	95.29	37.04	46.53	54.63	61.42	71.41	78.81	.00	10/12/95	ThmCal	C	5	H	12	O	2		0	G	6
C#CC#CC#C	156.26	71.54	24.72	28.22	30.58	32.30	34.94	36.84	39.98	10/12/95	ThmCal	C	6	H	2		0		0	G	2
C#CC#CC*C	119.37	78.53	24.49	29.80	33.05	36.99	41.23	44.30	49.07	10/12/95	ThmCal	C	6	H	4		0		0	G	2
ORTACLO.	4.69	84.87	26.75	33.67	39.35	43.81	50.21	54.42	.00	10/12/95	ThmCal	C	6	H	4	O	1	CL	1	G	0
C#CC.CC#C	134.40	82.25	26.13	30.86	34.71	37.89	42.95	46.65	53.31	10/12/95	ThmCal	C	6	H	5		0		0	G	3
LINC6H5P.	140.83	80.92	26.14	32.51	36.39	40.63	45.11	48.57	53.58	10/12/95	ThmCal	C	6	H	5		0		0	G	3
C6H5.	80.70	69.28	19.03	25.46	30.83	35.11	41.49	45.98	53.14	10/12/95	ThmCal	C	6	H	5		0		0	G	0
C6H5CL	12.39	74.82	23.37	30.55	36.46	41.21	48.04	52.73	.00	10/12/95	ThmCal	C	6	H	5	CL	1		0	G	0
PHENOXY	12.10	76.47	22.82	29.76	35.65	40.40	47.41	52.15	59.35	10/12/95	ThmCal	C	6	H	5	O	1		0	G	0
CYC6.CLO.	51.61	86.20	28.18	36.62	43.19	48.15	54.85	59.29	.00	10/12/95	ThmCal	C	6	H	5	O	1	CL	1	G	0
CL.CYC6O.	53.74	87.93	27.45	35.59	41.58	46.13	54.50	59.05	.00	10/12/95	ThmCal	C	6	H	5	O	1	CL	1	G	0
ORTACLOH	-29.71	83.57	28.33	35.91	42.02	46.74	53.42	57.83	.00	10/12/95	ThmCal	C	6	H	5	O	1	CL	1	G	1
FULVENE	42.79	63.22	23.84	31.90	38.08	42.50	48.58	52.93	59.35	10/12/95	ThmCal	C	6	H	6		0		0	G	0
FULVENE	42.79	63.22	23.84	31.90	38.08	42.50	48.58	52.93	59.35	10/12/95	ThmCal	C	6	H	6		0		0	G	0
C#CCCC#C	99.50	81.32	26.72	32.06	36.46	40.08	45.86	50.14	58.80	10/12/95	ThmCal	C	6	H	6		0		0	G	3
C*C*CC*C*	98.10	81.78	26.32	31.58	36.04	40.00	46.04	50.44	57.02	10/12/95	ThmCal	C	6	H	6		0		0	G	1
C#CC#CCC	90.75	81.34	26.63	31.99	36.43	40.11	45.95	50.30	58.12	10/12/95	ThmCal	C	6	H	6		0		0	G	3
C#CC*C*CC	94.50	81.36	26.56	32.59	36.65	41.50	47.13	51.29	57.64	10/12/95	ThmCal	C	6	H	6		0		0	G	2
C#CC*CC*C	81.73	79.53	26.33	33.26	37.75	42.55	47.93	52.06	58.11	10/12/95	ThmCal	C	6	H	6		0		0	G	3
CYC6H6	19.80	64.24	19.44	26.64	32.76	37.80	45.24	50.46	58.38	10/12/95	ThmCal	C	6	H	6		0		0	G	0
C6H5OH	-22.30	75.17	24.40	32.00	38.32	43.33	50.62	55.56	63.16	10/12/95	ThmCal	C	6	H	6	O	1		0	G	1
PHENOL	-22.30	75.17	24.40	32.00	38.32	43.33	50.62	55.56	63.16	10/12/95	ThmCal	C	6	H	6	O	1		0	G	1
CYHXE.O.	62.75	79.90	24.53	32.64	39.33	44.28	51.75	56.72	63.21	10/12/95	ThmCal	C	6	H	6	O	1		0	G	0
CYC6.CLOH	-.35	87.66	29.16	37.92	44.80	50.04	57.23	62.09	.00	10/12/95	ThmCal	C	6	H	6	O	1	CL	1	G	1
CYC6CL.OH	1.78	89.39	28.43	36.89	43.19	48.02	56.88	61.85	.00	10/12/95	ThmCal	C	6	H	6	O	1	CL	1	G	1
C#CC.CC*C	94.74	86.28	27.15	33.08	38.12	42.34	48.96	53.78	61.08	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCC.C*C	93.34	82.92	26.20	32.46	37.79	42.21	49.12	54.13	62.72	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCCC.*C	116.74	88.54	27.40	33.07	37.93	42.01	48.53	53.36	61.81	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCC.C*C	93.34	82.92	26.20	32.46	37.79	42.21	49.12	54.13	62.72	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCCC*C.	118.94	88.12	27.55	33.53	38.51	42.61	49.05	53.78	62.04	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCC.*CC	114.10	87.73	27.11	32.63	37.38	41.50	48.06	52.97	61.57	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCC*C.C	114.10	87.73	27.11	32.63	37.38	41.50	48.06	52.97	61.57	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CCC*CC.	93.30	84.16	26.83	33.28	38.54	42.90	49.56	54.42	62.84	10/12/95	ThmCal	C	6	H	7		0		0	G	3
C#CC.C*CC	86.10	82.37	26.36	32.22	37.31	41.39	48.33	53.40	62.21	10/12/95	ThmCal	C	6	H	7		0		0	G	2
CYHEXE.OH	10.79	81.36	25.51	33.94	40.94	46.17	54.13	59.52	66.80	10/12/95	ThmCal	C	6	H	7	O	1		0	G	1
CYC6CLOH1	-24.87	88.34	29.96	39.16	46.57	52.34	60.41	65.94	.00	10/12/95	ThmCal	C	6	H	7	O	1	CL	1	G	1
CYC6CLOH2	-22.74	90.07	29.23	38.13	44.96	50.32	60.06	65.70	.00	10/12/95	ThmCal	C	6	H	7	O	1	CL	1	G	1
C#CCCC*C	59.84	86.73	27.74	34.28	39.87	44.53	51.87	57.27	66.57	10/12/95	ThmCal	C	6	H	8		0		0	G	3
C.*CC*CC.	106.05	83.95	27.76	35.10	41.10	45.88	53.09	58.56	66.55	10/12/95	ThmCal	C	6	H	8		0		0	G	1
C#CCCC*C	59.84	86.73	27.74	34.28	39.87	44.53	51.87	57.27	66.57	10/12/95	ThmCal	C	6	H	8		0		0	G	3
C#CCC*CC	57.20	85.92	27.45	33.84	39.32	44.02	51.40	56.88	66.33	10/12/95	ThmCal	C	6	H	8		0		0	G	3
CY13HD	25.41	73.86	22.66	30.72	37.76	43.56	52.27	58.44	.00	10/12/95	ThmCal	C	6	H	8		0		0	G	0
CYC6H7OH	-13.73	82.04	26.31	35.18	42.71	48.47	57.31	63.37	71.62	10/12/95	ThmCal	C	6	H	8	O	1		0	G	1
C*CC*CC.C	41.35	84.82	27.66	35.81	42.56	48.13	56.36	62.69	71.78	10/12/95	ThmCal	C	6	H	9		0		0	G	2
C*CCCC*C.	79.28	92.15	28.57	35.75	41.92	47.06	55.06	60.91	69.81	10/12/95	ThmCal	C	6	H	9		0		0	G	3
C*CCCC.*C	77.08	92.57	28.42	35.29	41.34	46.46	54.54	60.49	69.58	10/12/95	ThmCal	C	6	H	9		0		0	G	3
C*CCC.C*C	53.68	86.95	27.22	34.68	41.20	46.66	55.13	61.26	70.49	10/12/95	ThmCal	C	6	H	9		0		0	G	3
C*CC*CCC	13.45	86.37	29.49	37.67	44.54	50.12	58.66	65.19	74.93	10/12/95	ThmCal	C	6	H	10		0		0	G	3

CC*CC*CC	10.34	85.15	29.62	37.32	43.92	49.42	58.04	64.66	74.58	10/12/95	ThmCal	C	6	H	10		0	0	G 3
C*CC*CCC	13.45	86.37	29.49	37.67	44.54	50.12	58.66	65.19	74.93	10/12/95	ThmCal	C	6	H	10		0	0	G 3
C*CCCC*C	20.18	89.38	28.76	36.50	43.28	48.98	57.88	64.40	74.34	10/12/95	ThmCal	C	6	H	10		0	0	G 3
C*CCC.CC	36.38	96.89	30.07	37.66	44.44	50.59	60.35	67.20	77.95	10/12/95	ThmCal	C	6	H	11		0	0	G 5
C*CC.CCC	23.53	88.64	30.03	38.17	45.46	51.66	61.35	68.51	79.30	10/12/95	ThmCal	C	6	H	11		0	0	G 4
C*CCCCC.	39.03	95.86	30.80	38.63	45.63	51.58	60.94	67.91	78.49	10/12/95	ThmCal	C	6	H	11		0	0	G 5
C.C*CCCC	23.19	89.86	30.70	38.99	46.21	52.30	61.79	68.80	79.47	10/12/95	ThmCal	C	6	H	11		0	0	G 4
CC*CCCC.	36.09	95.03	30.55	38.19	45.08	51.02	60.47	67.52	78.30	10/12/95	ThmCal	C	6	H	11		0	0	G 4
CC*CC.CC	20.59	87.80	29.78	37.73	44.91	51.10	60.88	68.12	79.11	10/12/95	ThmCal	C	6	H	11		0	0	G 3
CC*CCC.C	33.44	96.05	29.82	37.22	43.89	50.03	59.88	66.81	77.76	10/12/95	ThmCal	C	6	H	11		0	0	G 4
C*CCQ.CCC	2.56	106.92	37.98	47.14	54.89	61.07	70.87	78.41	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C*CCQC.CC	12.81	111.14	38.53	47.65	55.34	61.77	71.84	78.93	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C*CCQCC.C	12.81	111.14	38.53	47.65	55.34	61.77	71.84	78.93	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C*CCQCCC.	15.46	110.12	39.26	48.62	56.53	62.76	72.43	79.64	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C.*CCQCCC	25.56	110.28	39.84	49.23	57.08	63.24	72.77	79.89	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C*C.CQCCC	23.36	110.70	39.69	48.77	56.50	62.64	72.25	79.47	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
C*CC.QCCC	-2.24	105.20	38.24	47.60	55.70	62.19	72.31	79.83	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
C*CCQCCC.	15.46	112.30	39.26	48.62	56.53	62.76	72.43	79.64	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
Q.CC*CCCC	2.61	111.94	37.92	46.22	53.36	59.78	69.26	76.91	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
QC.C*CCCC	.01	107.91	38.43	47.24	54.83	61.55	71.23	78.74	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
QCC.*CCCC	32.00	121.50	43.79	52.88	60.78	67.85	78.29	86.42	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
QCC*CC.CC	.01	107.91	38.43	47.24	54.83	61.55	71.23	78.74	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
QCC*CCC.C	12.86	116.16	38.47	46.73	53.81	60.48	70.23	77.43	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
QCC*CCCC.	15.51	116.51	39.20	47.70	55.00	61.47	70.82	78.14	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CC*CCQ.CC	-.38	108.27	37.73	46.70	54.34	60.51	70.40	78.02	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
C.C*CCQCC	-.38	107.68	39.16	48.98	57.11	63.48	73.28	80.53	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CC.*CCQCC	20.42	109.86	39.44	48.33	55.95	62.08	71.78	79.08	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CC*C.CQCC	20.42	109.86	39.44	48.33	55.95	62.08	71.78	79.08	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CC*CC.QCC	-5.18	104.36	37.99	47.16	55.15	61.63	71.84	79.44	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CC*CCQC.C	9.87	112.49	38.28	47.21	54.79	61.21	71.37	78.54	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CC*CCQCC.	12.52	112.85	39.01	48.18	55.98	62.20	71.96	79.25	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CCQ.C*CCC	-.21	108.65	37.35	46.61	54.41	60.65	70.55	78.16	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
C.CQC*CCC	12.69	113.23	38.63	48.09	56.05	62.34	72.11	79.39	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CCQC*CCC.	12.69	113.23	38.63	48.09	56.05	62.34	72.11	79.39	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CC.QC*CCC	-5.01	104.74	37.61	47.07	55.22	61.77	71.99	79.58	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CCQC.*CCC	20.59	110.24	39.06	48.24	56.02	62.22	71.93	79.22	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CCQC*C.CC	20.59	110.24	39.06	48.24	56.02	62.22	71.93	79.22	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 6
CCQC*CC.C	-2.81	104.62	37.86	47.63	55.88	62.42	72.52	79.99	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
C*C.CQCCC	23.36	110.70	39.69	48.77	56.50	62.64	72.25	79.47	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
QCC*C.CCC	24.10	113.09	39.02	47.39	54.74	61.10	70.96	78.47	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
QCC.*CCCC	24.10	113.09	39.02	47.39	54.74	61.10	70.96	78.47	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CC.*CCQCC	20.42	109.86	39.44	48.33	55.95	62.08	71.78	79.08	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CC*C.CQCC	20.42	109.86	39.44	48.33	55.95	62.08	71.78	79.08	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CCQC.*CCC	20.59	110.24	39.06	48.24	56.02	62.22	71.93	79.22	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
CCQC*C.CC	20.59	110.24	39.06	48.24	56.02	62.22	71.93	79.22	.00	10/12/95	ThmCal	C	6	H	11	O	2	0	G 5
C*CCCCC	-9.97	92.45	31.57	39.99	47.54	53.98	64.10	71.65	83.15	10/12/95	ThmCal	C	6	H	12		0	0	G 4
CC*CCCC	-12.91	91.61	31.32	39.55	46.99	53.42	63.63	71.26	82.96	10/12/95	ThmCal	C	6	H	12		0	0	G 4
C*CCQCCC	-33.54	106.70	40.03	49.98	58.44	65.16	75.59	83.38	.00	10/12/95	ThmCal	C	6	H	12	O	2	0	G 6

QCC*CCCC	-33.49	111.72	39.97	49.06	56.91	63.87	73.98	81.88	.00	10/12/95	ThmCal	C	6	H	12	O	2	0	G	6
CC*CCQCC	-36.48	108.05	39.78	49.54	57.89	64.60	75.12	82.99	.00	10/12/95	ThmCal	C	6	H	12	O	2	0	G	6
CCQC*CCC	-36.31	108.43	39.40	49.45	57.96	64.74	75.27	83.13	.00	10/12/95	ThmCal	C	6	H	12	O	2	0	G	6
CYC6H5C.	48.71	75.63	25.81	33.78	40.51	46.02	54.27	60.08	68.84	10/12/95	ThmCal	C	7	H	7		0	0	G	1
C6H5CH3	11.81	76.81	25.06	33.18	40.38	46.44	55.68	62.26	72.21	10/12/95	ThmCal	C	7	H	8		0	0	G	1
CYC6H5C	11.81	76.81	25.06	33.18	40.38	46.44	55.68	62.26	72.21	10/12/95	ThmCal	C	7	H	8		0	0	G	1
LINC7H9.	90.87	90.72	33.10	41.64	48.36	53.55	61.32	67.44	76.22	10/12/95	ThmCal	C	7	H	9		0	0	G	3
LINC7H9VS	88.67	91.14	32.95	41.18	47.78	52.95	60.80	67.02	75.99	10/12/95	ThmCal	C	7	H	9		0	0	G	3
LINC7H10	31.77	89.33	33.29	42.39	49.72	55.47	64.14	70.93	80.75	10/12/95	ThmCal	C	7	H	10		0	0	G	3
C*CC2CC*C	11.73	96.94	34.89	43.92	51.86	58.53	69.05	76.80	88.66	10/12/95	ThmCal	C	7	H	12		0	0	G	4
C*CCICCC*	13.26	96.12	33.99	43.39	51.70	58.47	69.14	76.88	88.84	10/12/95	ThmCal	C	7	H	12		0	0	G	4
C*CC2CC*C	11.73	98.32	34.89	43.92	51.86	58.53	69.05	76.80	88.66	10/12/95	ThmCal	C	7	H	12		0	0	G	5
C*CCCCCC.	34.10	105.28	36.30	45.58	53.88	60.93	72.01	80.25	92.69	10/12/95	ThmCal	C	7	H	13		0	0	G	5
C*CCCCC.C	31.45	106.31	35.57	44.61	52.69	59.94	71.42	79.54	92.15	10/12/95	ThmCal	C	7	H	13		0	0	G	5
C*CC.CCCC	18.60	98.06	35.53	45.12	53.71	61.01	72.42	80.85	93.50	10/12/95	ThmCal	C	7	H	13		0	0	G	5
C*CCCCCC	-14.90	101.87	37.07	46.94	55.79	63.33	75.17	83.99	97.35	10/12/95	ThmCal	C	7	H	14		0	0	G	5
C3CCIC*C	-22.52	90.91	37.95	48.37	57.53	65.18	76.87	85.39	98.10	10/12/95	ThmCal	C	7	H	14		0	0	G	5
C*CYHEPE.	51.65	85.68	32.39	43.56	53.01	60.67	72.24	80.48	92.78	10/12/95	ThmCal	C	8	H	11		0	0	G	0
C*CYHEPE	19.92	85.62	32.75	44.33	54.31	62.53	75.14	84.18	97.59	10/12/95	ThmCal	C	8	H	12		0	0	G	0
DIC2.C*C	3.28	105.87	41.02	51.34	60.44	68.08	80.22	89.20	102.98	10/12/95	ThmCal	C	8	H	14		0	0	G	5
C3.CCC3	-4.80	101.20	45.11	57.94	69.21	78.58	92.50	102.40	.00	10/12/95	ThmCal	C	8	H	17		0	0	G	6
C3.CCC3	-4.80	101.20	45.11	57.94	69.21	78.58	92.50	102.40	.00	10/12/95	ThmCal	C	8	H	17		0	0	G	6
C2C.COCC3	-46.40	115.68	47.43	59.46	69.87	78.78	93.37	104.05	.00	10/12/95	ThmCal	C	8	H	17	O	1	0	G	7
C3CCC3Q.	-45.33	116.86	51.74	65.50	77.24	87.03	101.50	112.14	.00	10/12/95	ThmCal	C	8	H	17	O	2	0	G	8
C2C.COOTB	-35.26	125.45	51.83	64.38	75.37	84.66	99.61	111.05	.00	10/12/95	ThmCal	C	8	H	17	O	2	0	G	8
C3.COOCC3	-37.46	121.92	52.83	66.68	78.19	87.42	102.00	112.96	.00	10/12/95	ThmCal	C	8	H	17	O	2	0	G	8
C3CCC3	-53.80	95.60	45.88	59.30	71.12	80.98	95.66	106.14	.00	10/12/95	ThmCal	C	8	H	18		0	0	G	7
C3CCC3	-53.80	95.60	45.88	59.30	71.12	80.98	95.66	106.14	.00	10/12/95	ThmCal	C	8	H	18		0	0	G	9
C3CCCICC	-54.13	101.20	45.36	58.28	69.78	79.47	94.25	105.00	.00	10/12/95	ThmCal	C	8	H	18		0	0	G	7
C2CCOCC3	-90.80	111.80	48.21	61.94	73.42	82.93	98.12	109.07	.00	10/12/95	ThmCal	C	8	H	18	O	1	0	G	7
C2CCOOCC3	-79.66	121.57	52.61	66.86	78.92	88.81	104.36	116.07	.00	10/12/95	ThmCal	C	8	H	18	O	2	0	G	8
C3COOCC3	-86.46	114.95	53.60	68.04	80.10	89.82	105.16	116.70	.00	10/12/95	ThmCal	C	8	H	18	O	2	0	G	8
INDENE.	71.55	82.81	28.80	38.03	46.18	52.84	62.63	69.09	78.65	10/12/95	ThmCal	C	9	H	7		0	0	G	0
INDENE	39.82	82.75	29.16	38.80	47.48	54.70	65.53	72.79	83.46	10/12/95	ThmCal	C	9	H	8		0	0	G	0
INDANEAL.	46.39	84.56	30.62	41.50	50.82	58.25	69.15	76.61	86.94	10/12/95	ThmCal	C	9	H	9		0	0	G	0
CC7H913V.	88.46	51.38	38.06	49.56	59.50	67.08	78.33	86.45	98.49	10/12/95	ThmCal		9		9		0	0	G	1
INDANE	14.66	83.12	30.98	42.27	52.12	60.11	72.05	80.31	91.75	10/12/95	ThmCal	C	9	H	10		0	0	G	0
CC7H1013V	31.56	89.52	38.40	50.77	61.44	69.60	81.67	90.36	103.25	10/12/95	ThmCal	C	9	H	10		0	0	G	1
PHPH.	103.82	98.12	38.65	51.66	62.45	70.85	83.17	91.52	104.16	10/12/95	ThmCal	C	12	H	9		0	0	G	1
CLBIPHENY	35.51	103.66	42.99	56.75	68.08	76.95	89.72	98.27	.00	10/12/95	ThmCal	C	12	H	9	CL	1	0	G	1
BIPHENYL	42.92	93.89	39.06	52.84	64.38	73.54	86.92	96.00	109.40	10/12/95	ThmCal	C	12	H	10		0	0	G	1

Table A.4 Detailed Mechanism for Methanol and MTBE Oxidation

```
ELEMENTS
H C O N AR
END
SPECIES
CH4                 CH3OH               CO                  CO2                 CH2O
C2H4                C2H6                COC                 COCOC               C2H2
C3COC C2C*C C*CC C#CC
H2O                 O2                  H2                  HO2                 H2O2
AR                  N2                  OH                   O                  H
CH2OH               CH3O                CH3                 CH2                 CH2S
CH                  HCO                 HCOH                C
COC.                COCOC.              COCOH               COCO.               C.OCOH
CQ.H2OH             CO.H2OH             CH2OHOO             CQH2O.              C.QOH
C2H5                C2H                 C2H3                CC.
HCQ*O               HOC.HOO.T           HCO2H               O.HC.OOHT           HOCHO.O.
HCCO                C.COH               CCO.                CCO2                CCO3
CC.OH               CH3OO               CCOO                CH3OOH              C*COOH
CCOH                CCOOH               C.COOH              CCHO                HCO3
HCO2.               CCC.                CCC                 C*C*O               CC.*O
CCQO.               COHCQ.              C.OHCQ              CO.CQ               CCQ.OH
C*COH               CCO3H               CYC.CO              C.CHO               CYCCO
COC.OC              COC.OH              CH2OOH              HCQ.*O              C.Q*O
C*COO.              C.*COOH             CYCOOC.             O*CCO.              O*CC*O
C.*CC               C*CC.               C*C.C               C*C*C
C.*COH              HC#COH              C*CO.               C.C*O               C*CC
CC.OOH              CH3CHO              COHC.OOH            O*CCOH              C.CQOH
CC.QOH              O*COOHC             CCO2H               CC.C                O*CCQ
C3C C3C. C2CC.  C*COC
C*C(C)OC C2.C*C CC*C. C#CC.
C2C*O C2COC. C2.COC C2C.OC
C3COC. C3.COC C3COO C3CO. C3.COOH
END
REACTIONS
O + O + M           <=> O2 + M              1.88E+13    0.0     -1788. !86 TSANG
      AR/1.0/
H + O + M           <=> OH + M              4.71E+18   -1.0        0. !86 TSANG
H + O2              <=> O + OH              1.99E+14    0.0    16802. !92 BAULCH
H + O2 + M          <=> HO2 + M             6.16E+17   -0.8        0. !92 BAULCH
      AR/1.0/ H2/3.41/ N2/2.29/ H2O/2.53/
H + H + M           <=> H2 + M              6.52E+17   -1.0        0. !92 BAULCH
      AR/1.0/ N2/1.53/
H2 + O              <=> H + OH              5.12E+04    2.67     6285. !92 BAULCH
OH + H + M          <=> H2O + M             8.34E+21   -2.0        0. !92 BAULCH
      AR/1.0/ N2/2.65/ H2O/16.96/
OH + OH             <=> H2O + O             1.51E+09    1.14       99. !92 BAULCH
OH + H2             <=> H + H2O             1.02E+08    1.60     3298. !92 BAULCH
HO2 + O             <=> OH + O2             3.25E+13    0.0         0. !92 BAULCH
HO2 + H             <=> OH + OH             1.69E+14    0.0       874. !92 BAULCH
HO2 + H             <=> H2 + O2             4.28E+13    0.0      1411. !92 BAULCH
HO2 + H             <=> H2O + O             3.01E+13    0.0      1721. !92 BAULCH
HO2 + OH            <=> H2O + O2            2.89E+13    0.0      -497. !92 BAULCH
HO2 + HO2           <=> H2O2 + O2           1.87E+12    0.0      1540. !92BAULCH/500-
1250K
H2O2 + M            <=> OH + OH + M         1.21E+17    0.0     45507. !92 BAULCH
H2O2 + H            <=> H2O + OH            1.02E+13    0.0      3577. !92 BAULCH
H2O2 + H            <=> H2 + HO2            1.69E+12    0.0      3756. !92 BAULCH
H2O2 + O            <=> OH + HO2            9.63E+06    2.0      3974. !92 BAULCH
H2O2 + OH           <=> H2O + HO2           7.83E+12    0.0      1311. !92 BAULCH
C + O2              <=> O + CO              5.80E+13    .000      576.
C + OH              <=> H + CO              5.00E+13    .000        0.
C + CH2             <=> H + C2H             5.00E+13    .000        0.
C + CH3             <=> H + C2H2            5.00E+13    .000        0.
CO + O + M          <=> CO2 + M             6.17E+14    0.0      3001. !86 TSANG
      H2/2.0/ O2/6.0/ H2O/6.0/ CH4/2.0/ CO/1.5/ CO2/3.5/ C2H6/3.0/ AR/0.5/
CO + OH             <=> CO2 + H             6.32E+06    1.50     -497. !92 BAULCH
CO + O2             <=> CO2 + O             2.53E+12    0.00    47693. !86 TSANG
CO + HO2            <=> CO2 + OH            1.51E+14    0.00    23648. !86 TSANG
CH + O              <=> H + CO              5.70E+13    .000        0.
CH + O2             <=> CO + OH             3.31E+13    0.0         0. !92 BAULCH
CH + O2             <=> HCO + O             3.31E+13    0.0         0. !92 BAULCH
```

```
CH + H                <=> C + H2                1.10E+14    .000        0.
CH + OH               <=> H + HCO               3.00E+13    .000        0.
CH + H2               <=> CH3                   1.45E+14    0.00     3497. !92 BAULCH
CH + H2O              <=> H + CH2O              1.71E+13    .000     -755.
CH + CH2              <=> H + C2H2              4.00E+13    .000        0.
CH + CH3              <=> H + C2H3              3.00E+13    .000        0.
CH + CH4              <=> H + C2H4              6.00E+13    .000        0.
CH + CO (+M)          <=> HCCO (+M)             5.00E+13    .000        0.
     LOW  /  2.690E+28   -3.740   1936.00/
     TROE/   .5757  237.00  1652.00  5069.00 /
     H2/2.00/ H2O/6.00/ CH4/2.00/ CO/1.50/ CO2/2.00/ C2H6/3.00/ AR/ .70/
CH + CO2              <=> HCO + CO              3.40E+12    .000      690.
CH + CH2O             <=> H + C*C*O             9.46E+13    .000     -515.
CH + HCCO             <=> CO + C2H2             5.00E+13    .000        0.
HCO + M               <=> CO + H + M            1.87E+17   -1.00    17000. !87TIMONEN
    H2/2.0/ H2O/6.0/ CH4/2.0/ CO/1.5/ CO2/2.0/ C2H6/3.0/ AR/0.7/ N2/1.0/
HCO + O               <=> CO + OH               3.01E+13    0.00        0. !92BAULCH/86TSANG
HCO + O               <=> H + CO2               3.00E+13    0.00        0. !SRI2
HCO + H               <=> CO + H2               9.04E+13    0.00        0. !92 BAULCH
HCO + OH              <=> CO + H2O              3.00E+13    0.00        0. !92 BAULCH
HCO + HO2             <=> CO2 + OH + H          3.00E+13    0.00        0. !92 BAULCH
HCO + O2 (+M)         <=> HCQ.*O (+M)           1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.0438E+01  2.1076E+00 -2.1625E-01 -1.5689E+00/
    CHEB/  3.9500E-01  2.0033E-01 -6.9226E-01  2.4277E-02  2.9829E-02/
    CHEB/ -2.8569E-01 -1.9522E-02 -8.6222E-03 -1.0518E-01 -1.1326E-02/
    CHEB/ -7.5687E-03 -3.2695E-02 -3.4330E-03 -2.8109E-03 -6.8658E-03/
    CHEB/ -4.4532E-04 -5.6770E-04/
HCO + O2 (+M)         <=> HCO2. + O (+M)        1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       7.1338E+00 -4.2012E-02 -3.0570E-02  3.6623E+00/
    CHEB/  5.3923E-02  3.8997E-02  3.0405E-02 -8.6167E-03 -5.8740E-03/
    CHEB/ -5.2534E-02 -4.4954E-03 -3.4104E-03  4.9426E-03  4.4950E-04/
    CHEB/  2.8644E-04  2.2599E-02  1.0334E-03  7.6782E-04 -1.1284E-02/
    CHEB/ -3.1293E-04 -2.1614E-04/
HCO + O2 (+M)         <=> C.Q*O (+M)            1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       8.3954E+00  2.1127E+00 -2.1721E-01 -1.0511E+00/
    CHEB/  4.1760E-01  2.2024E-01 -3.2671E-01  4.7939E-03  1.7720E-02/
    CHEB/ -6.8985E-02 -2.6980E-02 -1.4281E-02 -7.9502E-03 -1.1458E-02/
    CHEB/ -8.1398E-03 -3.2210E-03 -2.0381E-03 -1.9929E-03 -4.2591E-03/
    CHEB/  4.6441E-04  6.7923E-05/
HCO + O2 (+M)         <=> CO + HO2 (+M)         1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.1936E+01 -3.7995E-01 -2.1314E-01 -2.5373E-01/
    CHEB/  4.1696E-01  2.2073E-01 -2.1752E-01  6.5189E-05  1.4890E-02/
    CHEB/ -9.2033E-02 -2.8998E-02 -1.5821E-02 -2.8060E-02 -1.1417E-02/
    CHEB/ -8.2623E-03 -5.5673E-03 -1.5507E-03 -1.6999E-03 -6.0312E-05/
    CHEB/  7.7765E-04  2.8994E-04/
HCO + O2 (+M)         <=> CO2 + OH (+M)         1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.0877E+01 -3.7995E-01 -2.1314E-01 -2.5373E-01/
    CHEB/  4.1696E-01  2.2073E-01 -2.1752E-01  6.5186E-05  1.4890E-02/
    CHEB/ -9.2033E-02 -2.8998E-02 -1.5821E-02 -2.8060E-02 -1.1417E-02/
    CHEB/ -8.2623E-03 -5.5673E-03 -1.5506E-03 -1.6999E-03 -6.0260E-05/
    CHEB/  7.7761E-04  2.8962E-04/
HCO + CH3 = CH4 + CO              1.21E+14     0.00       0. !86 TSANG
HCQ.*O (+M)           <=> CO + HO2 (+M)         1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.6570E+00  1.2212E+00 -3.4313E-01  9.9764E+00/
    CHEB/  8.6302E-01  1.0213E-01 -6.5641E-01  2.0664E-01  8.7259E-02/
    CHEB/ -3.0727E-01  2.5441E-02  3.3186E-02 -1.3294E-01 -1.1843E-02/
    CHEB/  4.7664E-03 -5.0903E-02 -1.1049E-02 -2.6670E-03 -1.6015E-02/
    CHEB/ -5.4741E-03 -2.4831E-03/
HCQ.*O (+M)           <=> CO2 + OH (+M)         1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -3.7162E+00  1.2212E+00 -3.4313E-01  9.9764E+00/
    CHEB/  8.6302E-01  1.0213E-01 -6.5641E-01  2.0664E-01  8.7259E-02/
    CHEB/ -3.0727E-01  2.5441E-02  3.3186E-02 -1.3294E-01 -1.1843E-02/
    CHEB/  4.7664E-03 -5.0903E-02 -1.1049E-02 -2.6667E-03 -1.6015E-02/
    CHEB/ -5.4742E-03 -2.4833E-03/
C.Q*O (+M)            <=> CO + HO2 (+M)         1.00E+00     .000       0. ! ING341   10/95
    LOW / 1.0 0.0 0.0 /
```

```
     CHEB/  7   3       5.9464E+00  2.1452E+00 -1.9621E-01  2.1867E+00/
     CHEB/  3.8262E-01  1.9660E-01 -3.9233E-01 -8.0541E-03  1.1095E-02/
     CHEB/ -6.0977E-02 -2.7320E-02 -1.5235E-02  9.7130E-03 -5.4188E-03/
     CHEB/ -4.9042E-03  8.2732E-03  2.2483E-03  8.2242E-04  8.4029E-04/
     CHEB/  1.4555E-03  9.8956E-04/
C.Q*O (+M)            <=> CO2 + OH (+M)         1.00E+00     .000        0. ! ING341   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       4.8872E+00  2.1452E+00 -1.9621E-01  2.1867E+00/
     CHEB/  3.8262E-01  1.9660E-01 -3.9233E-01 -8.0542E-03  1.1095E-02/
     CHEB/ -6.0977E-02 -2.7320E-02 -1.5235E-02  9.7130E-03 -5.4187E-03/
     CHEB/ -4.9043E-03  8.2732E-03  2.2484E-03  8.2201E-04  8.4026E-04/
     CHEB/  1.4555E-03  9.8937E-04/
CH2S + O              <=> H2 + CO               1.50E+13     .000        0.
CH2S + O              <=> H + HCO               1.50E+13     .000        0.
CH2S + H              <=> CH + H2               3.00E+13     .000        0.
CH2S + O2             <=> CO + H2O              1.20E+13     .000        0.
CH2S + OH             <=> CH2O + H              3.00E+13     .000        0.
CH2S + H2             <=> CH3 + H               7.23E+13     .000        0.
CH2S + H2O            <=> CH2 + H2O             3.00E+13     .000        0.
CH2S + CH3            <=> H + C2H4              1.20E+13     .000     -570.
CH2S + CO             <=> CH2 + CO              9.00E+12     .000        0.
CH2S + CO2            <=> CH2 + CO2             7.00E+12     .000        0.
CH2S + CO2            <=> CO + CH2O             1.40E+13     .000        0.
CH2S + M              <=> CH2 + M               1.00E+13     0.0         0. !MILLER
CH2S + CH4            <=> 2CH3                  4.00E+13     0.0         0. !MILLER
CH2S + C2H6           <=> CH3 + C2H5            1.20E+14     0.0         0. !MILLER
CH2S + O2             <=> CO + OH + H           3.00E+13     0.0         0. !MILLER
CH2S + H              <=> CH2 + H               2.00E+14     0.0         0. !MILLER
CH2 + H (+M)          <=> CH3 (+M)              2.50E+16     -.80        0.
     LOW  /  3.200E+27   -3.140   1230.00/
     TROE/   .6800   78.00  1995.00  5590.00 /
     H2/2.00/ H2O/6.00/ CH4/2.00/ CO/1.50/ CO2/2.00/ C2H6/3.00/ AR/ .70/
CH2 + H2              <=> H + CH3               5.00E+05    2.00     7230.
2CH2                  <=> H2 + C2H2             3.200E+13   .000        0.
CH2 + CH3             <=> H + C2H4              4.000E+13   .000        0.
CH2 + CH4             <=> 2CH3                  2.460E+06   2.00     8270.
CH2 + CO (+M)         <=> C*C*O (+M)            8.100E+11    .50     4510.
     LOW  /  2.690E+33   -5.110   7095.00/
     TROE/   .5907  275.00  1226.00  5185.00 /
     H2/2.00/ H2O/6.00/ CH4/2.00/ CO/1.50/ CO2/2.00/ C2H6/3.00/ AR/ .70/
CH2 + HCCO            <=> C2H3 + CO             3.00E+13    .000        0.
CH2 + O               <=> CO + H + H            5.00E+13    0.00        0. !MILLER
CH2 + O2              <=> CO2 + H + H           1.60E+12    0.00     1000. !MILLER
CH2 + O2              <=> CH2O + O              5.00E+13    0.00     9011. !MILLER
CH2 + O2              <=> CO2 + H2              6.90E+11    0.00      502. !MILLER
CH2 + O2              <=> CO + OH + H           8.60E+10    0.00     -502. !MILLER
CH2 + O2              <=> HCO + OH              4.30E+10    0.00     -502. !MILLER
CH2 + H               <=> CH + H2               4.00E+13    0.0         0. !MILLER
CH2 + OH              <=> CH + H2O              1.13E+07    2.0      3011. !MILLER
CH2 + OH              <=> CH2O + H              2.50E+13    0.0         0. !MILLER
HCOH (+M)             <=> CH2O (+M)             1.00E+00    .000        0. ! ING111   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       6.1562E+00  2.0857E+00 -2.0733E-01  1.8817E+00/
     CHEB/  4.3977E-01  2.0117E-01 -3.5772E-01 -7.3874E-03  1.4671E-02/
     CHEB/ -5.0220E-02 -2.7614E-02 -1.3531E-02  8.7047E-03 -6.1176E-03/
     CHEB/ -4.9343E-03  6.9705E-03  1.3861E-03  1.6764E-04  9.3469E-04/
     CHEB/  1.1733E-03  6.7250E-04/
HCOH + O2 (+M)        <=> HOC.HOO.T (+M)        1.00E+00     .000        0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       9.8754E+00  2.4000E+00 -6.9109E-02 -1.3323E+00/
     CHEB/  1.2612E-01  8.5682E-02 -3.9071E-01 -2.0950E-02 -1.2240E-02/
     CHEB/ -7.1692E-02 -8.4939E-03 -6.5982E-03  5.6278E-03  4.6319E-04/
     CHEB/  9.7938E-05  7.1146E-03  1.3651E-03  1.0155E-03 -1.4034E-03/
     CHEB/  4.1476E-04  3.4317E-04/
HCOH + O2 (+M)        <=> HCO2H + O (+M)        1.00E+00     .000        0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.2574E+01 -9.4583E-02 -6.5484E-02 -1.3858E-01/
     CHEB/  1.2220E-01  8.3300E-02 -1.2444E-01 -2.2717E-02 -1.3643E-02/
     CHEB/ -5.6074E-02 -7.9480E-03 -6.2684E-03 -2.0959E-02  9.0130E-04/
     CHEB/  4.3887E-04 -7.1897E-03  1.3559E-03  1.0246E-03 -2.3980E-03/
     CHEB/  3.0783E-04  2.6385E-04/
HCOH + O2 (+M)        <=> O.HC.OOHT (+M)        1.00E+00     .000        0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
```

```
     CHEB/  7   3       6.3550E+00  2.3607E+00 -9.7302E-02 -4.0759E-01/
     CHEB/  1.8126E-01  1.2489E-01 -5.8387E-02 -4.1168E-02 -2.6111E-02/
     CHEB/ -4.8702E-03 -6.6320E-03 -5.6742E-03 -1.7010E-02  2.1535E-03/
     CHEB/  1.3989E-03 -1.6592E-02  1.2474E-03  9.7091E-04 -8.8423E-03/
     CHEB/  1.8141E-04  1.6455E-04/
 HCOH + O2 (+M)       <=> HCO + HO2 (+M)       1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       8.7604E+00 -1.0537E-01 -7.4657E-02  9.4673E-01/
     CHEB/  1.4115E-01  9.8933E-02  6.6641E-02 -3.6850E-02 -2.4383E-02/
     CHEB/ -2.1516E-03 -3.5706E-03 -3.3300E-03 -6.1249E-03  2.1393E-03/
     CHEB/  1.5085E-03 -4.1403E-03  8.8101E-04  7.0049E-04 -2.0348E-03/
     CHEB/  1.7186E-05  2.8953E-05/
 HCOH + O2 (+M)       <=> HOCHO.O. (+M)        1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       7.5693E+00  2.4055E+00 -6.5473E-02 -4.6626E-01/
     CHEB/  1.2204E-01  8.3276E-02 -9.1759E-02 -2.3037E-02 -1.3900E-02/
     CHEB/ -3.7390E-02 -7.8243E-03 -6.1866E-03 -2.3282E-02  9.6387E-04/
     CHEB/  4.8882E-04 -1.3562E-02  1.3487E-03  1.0215E-03 -6.7901E-03/
     CHEB/  2.9458E-04  2.5380E-04/
 HCOH + O2 (+M)       <=> HCO2. + OH (+M)      1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.2463E+01 -9.4112E-02 -6.5226E-02 -6.5963E-02/
     CHEB/  1.2179E-01  8.3120E-02 -1.0078E-01 -2.3167E-02 -1.4001E-02/
     CHEB/ -4.8667E-02 -7.7882E-03 -6.1650E-03 -1.8865E-02  9.9518E-04/
     CHEB/  5.1314E-04 -6.7783E-03  1.3486E-03  1.0226E-03 -2.4298E-03/
     CHEB/  2.8742E-04  2.4861E-04/
 HCOH + O2 (+M)       <=> CO2 + H2O (+M)       1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.0967E+01 -9.3770E-02 -6.4995E-02 -3.2750E-02/
     CHEB/  1.2155E-01  8.2975E-02 -8.5631E-02 -2.3291E-02 -1.4099E-02/
     CHEB/ -4.2054E-02 -7.7562E-03 -6.1459E-03 -1.6095E-02  1.0262E-03/
     CHEB/  5.3718E-04 -5.6744E-03  1.3493E-03  1.0240E-03 -2.0120E-03/
     CHEB/  2.8023E-04  2.4296E-04/
 CH2O (+M)            <=> HCO + H (+M)         1.00E+00     .000       0. ! ING431   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -2.4865E+01  2.4983E+00 -1.2744E-03  2.7176E+01/
     CHEB/  2.0398E-03  1.5344E-03 -4.0470E-01 -3.1637E-04 -2.3747E-04/
     CHEB/ -1.7801E-01 -4.0163E-05 -3.0210E-05 -7.3725E-02 -7.1875E-06/
     CHEB/ -5.4854E-06 -2.9768E-02 -1.2684E-06 -9.4980E-07 -1.2275E-02/
     CHEB/ -1.7169E-07 -3.1177E-07/
 CH2O (+M)            <=> CO + H2 (+M)         1.00E+00     .000       0. ! ING431   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -1.6268E+01  2.4663E+00 -2.4732E-02  2.1289E+01/
     CHEB/  3.9566E-02  2.8837E-02 -4.6831E-01 -4.8873E-03 -3.3334E-03/
     CHEB/ -2.1142E-01 -1.2857E-03 -1.0002E-03 -9.2226E-02 -1.7399E-04/
     CHEB/ -1.4019E-04 -3.9322E-02 -1.6145E-06 -1.8082E-06 -1.6707E-02/
     CHEB/  1.0825E-05  8.5672E-06/
 CH2O + O             <=> HCO + OH             3.50E+13    0.0      3513. !HUN 94
 CH2O + H             <=> HCO + H2             2.19E+08    1.77     3000. !86 TSANG
 CH2O + OH            <=> HCO + H2O            3.44E+09    1.18     -447. !92 BAULCH/86
 TSANG
 CH2O + HO2           <=> HCO + H2O2           3.01E+12    0.0     13076. !92 BAULCH
 CH2O + CH3           <=> HCO + CH4            5.54E+03    2.81     5862. !86 TSANG
 CH2O + HO2 (+M)      <=> CQ.H2OH (+M)         1.00E+00     .000       0. ! ING021   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       8.1389E+00  4.4938E-01 -4.3163E-01 -1.5342E+00/
     CHEB/  5.2669E-01  1.3430E-01 -5.0038E-01 -3.2560E-03 -3.9525E-02/
     CHEB/ -1.0076E-01  4.0030E-02 -8.2585E-03 -3.4191E-02  4.9415E-02/
     CHEB/  1.6386E-02 -3.7488E-02  2.7022E-02  1.0633E-02 -3.3175E-02/
     CHEB/  1.1866E-02  3.2254E-03/
 HOC.HOO.T (+M)       <=> HCO + HO2 (+M)       1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -2.1521E+00  2.3515E+00 -1.0373E-01  6.7230E+00/
     CHEB/  1.8812E-01  1.2943E-01 -3.7538E-01 -3.8917E-02 -2.4401E-02/
     CHEB/ -6.6129E-02 -7.7131E-03 -6.3515E-03  6.2980E-03  1.9514E-03/
     CHEB/  1.2115E-03  6.4857E-03  1.3587E-03  1.0405E-03 -1.8611E-03/
     CHEB/  2.7897E-04  2.3868E-04/
 HOC.HOO.T (+M)       <=> CO2 + H2O (+M)       1.00E+00     .000       0. ! ING121   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       2.7856E+00  2.0202E+00 -2.2705E-01  3.4407E+00/
     CHEB/  5.2878E-01  2.2434E-01 -4.9591E-01 -1.0532E-02  2.3468E-02/
     CHEB/ -1.1532E-01 -4.2092E-02 -2.0364E-02 -8.3447E-03 -1.1935E-02/
     CHEB/ -9.6433E-03  4.5276E-03  2.2807E-03 -4.5572E-05 -1.5180E-03/
```

```
    CHEB/  3.2669E-03  1.8814E-03/
O.HC.OOHT (+M)        <=> HCO + HO2 (+M)        1.00E+00     .000        0. ! ING121   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.0115E+00  2.4367E+00 -4.4686E-02  1.8625E+00/
    CHEB/  8.1362E-02  5.6689E-02 -1.5311E-01 -2.0381E-02 -1.3303E-02/
    CHEB/ -5.1438E-03 -8.8841E-04 -1.0796E-03 -6.7028E-03  1.5635E-03/
    CHEB/  1.1257E-03 -9.1339E-03  1.0021E-04  1.1286E-04 -5.0691E-03/
    CHEB/ -1.6477E-04 -1.2066E-04/
O.HC.OOHT (+M)        <=> CO2 + H2O (+M)        1.00E+00     .000        0. ! ING121   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        4.9603E+00  2.3986E+00 -6.0571E-02  8.3622E-01/
    CHEB/  8.3202E-02  4.1942E-02 -1.6474E-01  3.9956E-02  3.3133E-02/
    CHEB/ -5.4621E-02 -2.1324E-02 -1.2735E-02 -3.3022E-02 -9.8959E-03/
    CHEB/ -8.7100E-03 -1.5086E-02  4.1572E-03  2.6470E-03 -6.4273E-03/
    CHEB/  2.6020E-03  2.2528E-03/
HOCHO.O. (+M)         <=> HCO + HO2 (+M)        1.00E+00     .000        0. ! ING121   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.2500E+01  2.3612E+00 -9.7133E-02  1.4780E+01/
    CHEB/  1.8024E-01  1.2441E-01 -8.8125E-02 -4.0654E-02 -2.5884E-02/
    CHEB/ -5.0792E-02 -6.8133E-03 -5.7570E-03 -2.8858E-02  2.1948E-03/
    CHEB/  1.4212E-03 -1.5079E-02  1.2623E-03  9.8156E-04 -7.0613E-03/
    CHEB/  1.8265E-04  1.6573E-04/
HOCHO.O. (+M)         <=> CO2 + H2O (+M)        1.00E+00     .000        0. ! ING121   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        5.4464E+00  2.4974E+00 -1.9593E-03  1.4572E+00/
    CHEB/  2.8658E-03  2.1535E-03 -8.1607E-02 -5.9869E-04 -4.4854E-04/
    CHEB/ -8.2366E-03  3.1303E-05  2.3178E-05 -5.2053E-03  1.8335E-05/
    CHEB/  1.3961E-05 -4.8180E-03  2.7942E-06  2.1542E-06 -3.1329E-03/
    CHEB/  5.6376E-07  4.9922E-07/
CH3 + O               <=> CH2O + H              1.34E+14    -0.08       79. !CHEM221
CH3 + O               <=> CH2OH                 6.57E+12     0.04      -24. !CHEM221
CH3 + HO2             <=> CH3O + OH             1.81E+13     0.0         0. !92 BAULCH
CH3 + O2 (+M)         <=> CH3OO (+M)            1.00E+00     .000        0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.0700E+01  1.6297E+00 -3.2527E-01 -1.0852E+00/
    CHEB/  6.5665E-01  1.9239E-01 -5.8635E-01  1.7435E-01  8.8831E-02/
    CHEB/ -2.3815E-01  8.7690E-03  1.8628E-02 -6.6460E-02 -2.2693E-02/
    CHEB/ -5.3810E-03 -1.4917E-03 -1.5706E-02 -7.1318E-03  1.3324E-02/
    CHEB/ -5.6522E-03 -3.5378E-03/
CH3 + O2 (+M)         <=> CH3O + O (+M)         1.00E+00     .000        0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        2.1061E+00 -3.0803E-02 -2.2715E-02  9.6320E+00/
    CHEB/  3.3066E-02  2.4264E-02  2.0151E-01  1.3671E-03  1.1445E-03/
    CHEB/ -4.9486E-02 -3.0499E-03 -2.2430E-03 -8.0457E-02 -1.7591E-03/
    CHEB/ -1.3204E-03  5.1917E-03  1.5682E-04  1.0281E-04  2.5893E-02/
    CHEB/  5.8059E-04  4.2877E-04/
CH3 + O2 (+M)         <=> CH2OOH (+M)           1.00E+00     .000        0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.4413E+00  2.3493E+00 -1.0133E-01  4.2581E+00/
    CHEB/  1.6322E-01  1.0716E-01  1.0478E-01  4.9253E-03  6.1276E-03/
    CHEB/ -2.2086E-03 -1.4627E-02 -9.6212E-03 -1.8007E-02 -6.5285E-03/
    CHEB/ -4.8748E-03 -1.3336E-02 -4.0632E-04 -5.1982E-04 -8.1366E-03/
    CHEB/  1.1191E-03  7.4071E-04/
CH3 + O2 (+M)         <=> CH2O + OH (+M)        1.00E+00     .000        0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.0758E+00 -1.5011E-01 -1.0091E-01  4.5254E+00/
    CHEB/  1.6313E-01  1.0713E-01  1.2203E-01  4.4568E-03  5.8088E-03/
    CHEB/  1.4095E-02 -1.4789E-02 -9.7477E-03 -6.4238E-03 -6.4592E-03/
    CHEB/ -4.8337E-03 -8.0373E-03 -3.1334E-04 -4.5406E-04 -6.3695E-03/
    CHEB/  1.1546E-03  7.6868E-04/
CH3 + OH (+M)         <=> CH2 + H2O (+M)        1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.1887E+01 -8.8062E-01 -2.9693E-01  5.0743E-01/
    CHEB/  8.3088E-01  2.1347E-01  1.2722E-02  1.3365E-01  1.0429E-01/
    CHEB/ -4.5277E-02 -3.3395E-02  7.2118E-03 -2.8381E-02 -3.8573E-02/
    CHEB/ -1.7284E-02 -1.0654E-02 -1.7214E-02 -1.1925E-02 -1.8788E-03/
    CHEB/ -3.5368E-03 -3.8688E-03/
CH3 + OH (+M)         <=> HCOH + H2 (+M)        1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.2067E+01 -9.2229E-01 -3.0514E-01  2.6578E-01/
    CHEB/  8.2957E-01  2.0333E-01 -6.6929E-02  1.6023E-01  1.0965E-01/
    CHEB/ -7.6687E-02 -1.9150E-02  1.5070E-02 -4.3600E-02 -3.6100E-02/
    CHEB/ -1.3614E-02 -1.7643E-02 -1.8662E-02 -1.1509E-02 -4.3850E-03/
```

```
    CHEB/ -5.1499E-03 -4.5256E-03/
CH3 + OH (+M)          <=> CH2O + H2 (+M)          1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        9.9084E+00 -8.4912E-01 -2.8989E-01  6.9564E-01/
    CHEB/  8.3128E-01  2.1827E-01  6.5480E-02  1.0585E-01  9.8533E-02/
    CHEB/ -3.4859E-02 -4.2790E-02  1.5555E-03 -2.3900E-02 -3.6199E-02/
    CHEB/ -1.8696E-02 -8.3972E-03 -1.4302E-02 -1.1215E-02 -1.2164E-03/
    CHEB/ -2.2641E-03 -3.0659E-03/
CH3 + OH (+M)          <=> CH2OH + H (+M)          1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.0895E+01 -6.5661E-01 -2.8393E-01  1.4448E+00/
    CHEB/  6.5636E-01  2.4261E-01  4.9751E-02  7.4212E-02  7.3723E-02/
    CHEB/ -2.2894E-02 -3.8550E-02 -7.9141E-03 -1.7743E-02 -3.0216E-02/
    CHEB/ -1.7926E-02 -6.0229E-03 -1.0265E-02 -8.6587E-03 -5.0491E-04/
    CHEB/ -5.6599E-04 -1.5797E-03/
CH3 + OH (+M)          <=> CH3O + H (+M)           1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.8861E+00 -3.0043E-01 -1.7831E-01  4.0315E+00/
    CHEB/  3.3324E-01  1.8706E-01 -1.5833E-02  7.3848E-04  1.3838E-02/
    CHEB/ -1.9350E-02 -2.4573E-02 -1.4778E-02 -6.5900E-03 -1.0270E-02/
    CHEB/ -8.1060E-03 -6.7737E-04 -1.3607E-03 -1.5787E-03  6.4467E-04/
    CHEB/  9.6099E-04  5.5283E-04/
CH3 + CH3 (+M)         <=> C2H6 (+M)               1.00E+00     .000        0. ! ING231   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.2569E+01  8.1070E-01 -1.8575E-01 -9.7959E-01/
    CHEB/  9.6455E-01 -5.3653E-02 -7.9695E-01  4.0332E-01  6.7579E-02/
    CHEB/ -4.3713E-01  1.1367E-01  5.7628E-02 -2.2321E-01  9.0085E-03/
    CHEB/  2.2893E-02 -1.0764E-01 -1.3425E-02  2.7864E-03 -4.8751E-02/
    CHEB/ -1.0925E-02 -2.8739E-03/
CH3 + CH3 (+M)         <=> C2H5 + H (+M)           1.00E+00     .000        0. ! ING231   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        8.3715E+00 -7.8063E-01 -2.9278E-01  3.8248E+00/
    CHEB/  7.7511E-01  2.3844E-01  6.1788E-02  1.1063E-01  9.3773E-02/
    CHEB/ -4.0584E-02 -5.2498E-02 -5.0241E-03 -2.8767E-02 -4.5633E-02/
    CHEB/ -2.3854E-02 -8.0692E-03 -1.4841E-02 -1.2738E-02  7.3971E-04/
    CHEB/  5.4986E-04 -2.2629E-03/
CH3 + CH3              <=> C2H4 + H2               1.00E+16     0.0     32028. !84 warantz
CH3O (+M)              <=> CH2O + H (+M)           1.00E+00     .000        0. ! ING071   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -5.7799E-01  1.7078E+00 -3.3987E-01  7.9829E+00/
    CHEB/  5.9444E-01  1.8894E-01 -4.8962E-01  1.4600E-01  9.3075E-02/
    CHEB/ -2.3170E-01  2.0974E-02  2.6894E-02 -9.2546E-02 -1.0394E-02/
    CHEB/  4.6192E-04 -2.9863E-02 -1.2060E-02 -5.7632E-03 -5.7033E-03/
    CHEB/ -7.0779E-03 -4.6915E-03/
CH3O + O2              <=> CH2O+HO2                3.61E+10     0.00     2126. !92 BAULCH
CH3O + O               <=> CH2O + OH               6.02E+12     0.00        0. !NORTON
CH3O + H               <=> CH2O + H2               1.99E+13     0.00        0. !NORTON
CH3O + OH              <=> CH2O + H2O              1.81E+13     0.00        0. !NORTON
CH3O + HO2             <=> CH2O + H2O2             3.01E+11     0.00        0. !NORTON
CH3O + CO              <=> CH3 + CO2               1.57E+13     0.00    11797. !NORTON
CH3O + CH2O            <=> CH3OH + HCO             1.02E+11     0.00     2981. !86 TSANG
CH3O + C2H6            <=> C2H5 + CH3OH            2.41E+11     0.00     7094. !86 TSANG
CH3O + CH3             <=> CH2O + CH4              2.41E+13     0.00        0. !86 TSANG
CH3O + CH3 (+M)        <=> COC (+M)                1.00E+00     .000        0. ! ING131   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.2114E+01  6.9341E-01 -1.0286E-01 -1.6652E+00/
    CHEB/  1.0564E+00 -8.2152E-02 -1.1477E+00  5.2850E-01  4.1011E-02/
    CHEB/ -6.7729E-01  1.6105E-01  6.2820E-02 -3.5610E-01  2.4959E-03/
    CHEB/  3.2070E-02 -1.6871E-01 -3.3499E-02  4.5106E-03 -7.0487E-02/
    CHEB/ -2.5548E-02 -5.2867E-03/
CH3O + CH3 (+M)        <=> COC. + H (+M)           1.00E+00     .000        0. ! ING131   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        4.4340E+00 -7.7128E-01 -2.8202E-01  4.9374E+00/
    CHEB/  8.2139E-01  2.5007E-01 -2.0666E-01  8.1549E-02  8.9885E-02/
    CHEB/ -1.7759E-01 -8.5995E-02 -2.1477E-02 -8.9828E-02 -5.5854E-02/
    CHEB/ -3.4125E-02 -3.1076E-02 -8.6274E-03 -1.2427E-02 -7.4614E-03/
    CHEB/  9.4737E-03  2.6644E-03/
CH3O + CH3 (+M)        <=> CH2S + CH3OH (+M)       1.00E+00     .000        0. ! ING131   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        5.5054E+00 -7.7401E-01 -2.8258E-01  4.9065E+00/
    CHEB/  8.2208E-01  2.4960E-01 -2.1453E-01  8.3935E-02  9.0785E-02/
    CHEB/ -1.7833E-01 -8.5071E-02 -2.0849E-02 -8.9554E-02 -5.6577E-02/
    CHEB/ -3.4248E-02 -3.1159E-02 -9.3621E-03 -1.2772E-02 -7.5350E-03/
```

```
    CHEB/  9.3834E-03  2.5130E-03/
CH3O + CH2O (+M)      <=> COCO. (+M)            1.00E+00     .000        0. ! ING171   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        4.8507E+00  2.4880E+00 -8.9090E-03  1.0885E+00/
    CHEB/  8.4556E-03  6.3254E-03 -3.8968E-01  4.2288E-03  3.1392E-03/
    CHEB/ -4.9090E-03 -4.9612E-04 -3.7005E-04  8.4576E-02 -8.9232E-04/
    CHEB/ -6.6124E-04  6.0023E-02 -1.4535E-04 -1.0935E-04  2.1901E-02/
    CHEB/  1.8052E-04  1.3041E-04/
CH3O + CH2O (+M)      <=> COC. + O (+M)         1.00E+00     .000        0. ! ING171   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -2.4111E+01 -9.9228E-06 -7.6427E-06  2.9025E+01/
    CHEB/ -6.2432E-06 -4.3240E-06  2.9663E-01 -4.6701E-06 -3.5965E-06/
    CHEB/  9.0999E-02 -4.5005E-06 -3.0985E-06  3.1464E-02 -3.6235E-06/
    CHEB/ -2.7602E-06  1.1995E-02 -2.1099E-06 -1.6468E-06  4.8342E-03/
    CHEB/ -1.1241E-06 -8.9420E-07/
CH3O + CH2O (+M)      <=> C.OCOH (+M)           1.00E+00     .000        0. ! ING171   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        3.7598E+00  1.3138E+00 -2.8299E-01  2.3923E+00/
    CHEB/  1.0212E+00  1.2356E-01 -2.0388E-01  2.3200E-01  1.2998E-01/
    CHEB/ -5.2722E-02 -3.2222E-02  3.3932E-02  3.5542E-02 -6.0108E-02/
    CHEB/ -1.3784E-02  4.5057E-02 -2.6695E-02 -1.7156E-02  2.4618E-02/
    CHEB/ -5.3400E-04 -6.4646E-03/
CH2OH (+M)            <=> CH2O + H (+M)         1.00E+00     .000        0. ! ING092   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -2.7083E+00  1.8020E+00 -3.0931E-01  9.9291E+00/
    CHEB/  5.9467E-01  2.1337E-01 -6.2201E-01  1.0554E-01  7.9454E-02/
    CHEB/ -2.7808E-01 -3.9170E-03  1.0325E-02 -1.1301E-01 -1.6324E-02/
    CHEB/ -6.6713E-03 -3.9890E-02 -9.5562E-03 -5.9374E-03 -1.0784E-02/
    CHEB/ -3.5702E-03 -2.7335E-03/
CH2OH (+M)            <=> CH3O (+M)             1.00E+00     .000        0. ! ING092   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -5.2526E+00  2.1026E+00 -2.2272E-01  1.1233E+01/
    CHEB/  3.9362E-01  2.0200E-01 -5.8249E-01  3.3501E-02  3.6550E-02/
    CHEB/ -2.4821E-01 -1.7730E-02 -7.0233E-03 -9.4377E-02 -1.3182E-02/
    CHEB/ -8.9032E-03 -3.0028E-02 -4.8613E-03 -3.9840E-03 -6.3238E-03/
    CHEB/ -7.8881E-04 -9.0552E-04/
CH2OH + H             <=> CH2O + H2             6.03E+12    0.0          0. !87 TSANG
CH2OH + O             <=> CH2O + OH             4.22E+13    0.0          0. !87 TSANG
CH2OH + O2            = CH2O + HO2             2.41E+14    0.0       5000. !NORTON
CH2OH + OH            <=> CH2O + H2O            2.41E+13    0.0          0. !87 TSANG
CH2OH + CH2O          <=> CH3OH + HCO           5.49E+03    2.8       5872. !87 TSANG
CH2OH + CH3O          <=> CH3OH + CH2O          2.41E+13    0.0          0. !87 TSANG
CH2OH + CH2OH         <=> CH3OH + CH2O          4.82E+12    0.0          0. !87 TSANG
CH2OH + CH3OH         <=> CH3OH + CH3O          7.83E+09    0.0      12062. !87 TSANG
CH2OH + CH3           <=> CH2O + CH4            2.41E+12    0.00         0. !87 TSANG
CH2OH + CH3           <=> CCOH                  1.21E+13    0.0          0. !87 TSANG
CH2OH + O2 (+M)       <=> CQ.H2OH (+M)          1.00E+00     .000        0. ! ING011   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        9.6614E+00  2.0073E+00 -1.6901E-01 -1.5420E+00/
    CHEB/ -1.9434E-01  3.9587E-04 -4.6551E-01 -2.2952E-02 -3.7842E-02/
    CHEB/ -1.6188E-01  9.2704E-02  8.8166E-03 -7.5633E-02  6.0138E-02/
    CHEB/  8.9994E-03 -4.6184E-02  2.3746E-02  2.6229E-03 -2.7782E-02/
    CHEB/  8.0334E-03  1.2223E-03/
CH2OH + O2 (+M)       <=> CH2OHOO (+M)          1.00E+00     .000        0. ! ING011   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.1535E+01  1.4996E+00 -4.6745E-01  1.1082E-01/
    CHEB/  2.9235E-02  9.3391E-02 -2.0725E-01  8.0097E-02  8.9641E-03/
    CHEB/ -1.3214E-01  1.1602E-01  2.3387E-02 -6.9324E-02  6.4922E-02/
    CHEB/  1.4003E-02 -3.2258E-02  2.4112E-02  4.5022E-03 -1.2958E-02/
    CHEB/  7.0698E-03  1.8530E-03/
CH2OH + O2 (+M)       <=> CH2O + HO2 (+M)       1.00E+00     .000        0. ! ING011   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.1807E+01 -9.7905E-01 -4.6411E-01 -6.0240E-01/
    CHEB/  5.9595E-02  9.7410E-02 -3.6469E-01  9.4968E-02  1.1398E-02/
    CHEB/ -1.6631E-01  1.1974E-01  2.4759E-02 -7.6713E-02  6.3188E-02/
    CHEB/  1.4542E-02 -3.3665E-02  2.1201E-02  4.6420E-03 -1.2829E-02/
    CHEB/  4.8514E-03  1.8703E-03/
CH2OH + O2 (+M)       <=> HCO2H + OH (+M)       1.00E+00     .000        0. ! ING011   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        7.8612E+00 -3.3636E-01 -1.4096E-01  1.8184E+00/
    CHEB/ -3.7065E-02  1.7599E-02  1.0557E-01  2.6997E-02 -2.3965E-02/
    CHEB/ -6.7768E-02  7.8383E-02  1.4614E-02 -5.4558E-02  3.1950E-02/
    CHEB/  9.8515E-03 -2.2362E-02  4.0035E-03  2.5234E-03 -7.2599E-03/
```

```
    CHEB/ -1.6108E-03  9.8008E-04/
CH2OH + CH3O (+M)     <=> COCOH (+M)            1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.2765E+01  3.1585E-01 -5.3598E-02 -5.8913E-01/
    CHEB/  5.7115E-01 -8.6727E-02 -4.7133E-01  4.2284E-01 -4.1197E-02/
    CHEB/ -3.2940E-01  2.5509E-01  2.4025E-04 -2.0398E-01  1.2181E-01/
    CHEB/  2.0563E-02 -1.1346E-01  4.0965E-02  2.1313E-02 -5.7199E-02/
    CHEB/  3.5239E-03  1.3004E-02/
CH2OH + CH3O (+M)     <=> COC. + OH (+M)        1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       8.4362E+00 -1.8074E+00 -9.7130E-02  4.1392E+00/
    CHEB/  1.1160E+00 -1.0038E-01  5.1088E-01  5.9278E-01  3.7423E-02/
    CHEB/ -2.9236E-02  1.7853E-01  8.6223E-02 -9.0916E-02 -1.4344E-02/
    CHEB/  5.7277E-02 -5.9203E-02 -5.5903E-02  1.4795E-02 -2.6655E-02/
    CHEB/ -4.0148E-02 -7.7983E-03/
CH2OH + CH3O (+M)     <=> COCO. + H (+M)        1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       5.0405E+00 -1.6906E+00 -1.3637E-01  5.9746E+00/
    CHEB/  1.1660E+00 -7.9029E-02  3.6138E-01  5.1796E-01  8.7344E-02/
    CHEB/  8.1062E-03  1.1445E-01  9.7251E-02 -7.5612E-02 -3.5707E-02/
    CHEB/  4.2447E-02 -5.0719E-02 -5.5418E-02  6.1221E-04 -1.8706E-02/
    CHEB/ -3.4559E-02 -1.3994E-02/
CH2OH + CH3O (+M)     <=> COC.OH + H (+M)       1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       8.1247E+00 -1.8315E+00 -9.0990E-02  3.7136E+00/
    CHEB/  1.0962E+00 -1.0180E-01  5.3473E-01  6.0448E-01  2.7028E-02/
    CHEB/ -3.6833E-02  1.9585E-01  8.0984E-02 -9.6025E-02 -5.3785E-03/
    CHEB/  5.8977E-02 -6.2474E-02 -5.4119E-02  1.8236E-02 -2.8035E-02/
    CHEB/ -4.1083E-02 -5.7476E-03/
CH2OH + CH2O (+M)     <=> C.OCOH (+M)           1.00E+00     .000       0. ! ING181   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       6.8994E+00  1.1028E+00 -2.0010E-01  9.1185E-01/
    CHEB/  1.1577E+00  2.3886E-02 -6.6539E-01  3.1431E-01  1.2276E-01/
    CHEB/ -2.0079E-01 -1.7867E-02  5.2954E-02 -9.5693E-03 -6.9152E-02/
    CHEB/ -4.4478E-03  3.5394E-02 -3.6961E-02 -1.7204E-02  2.5521E-02/
    CHEB/ -5.9296E-03 -9.3026E-03/
CH2OH + CH2O (+M)     <=> COCO. (+M)            1.00E+00     .000       0. ! ING181   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.7013E-01  1.3674E+00 -2.8718E-01  2.7185E+00/
    CHEB/  1.0031E+00  1.4815E-01 -3.9961E-01  2.2583E-01  1.3106E-01/
    CHEB/ -9.7280E-02 -3.8185E-02  2.8726E-02  1.7643E-02 -6.5749E-02/
    CHEB/ -1.8485E-02  3.3334E-02 -2.9647E-02 -1.9569E-02  1.7779E-02/
    CHEB/ -1.2999E-03 -7.0409E-03/
CH2OH + CH2O (+M)     <=> CH3O + CH2O (+M)      1.00E+00     .000       0. ! ING181   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       4.2929E+00 -9.7875E-01 -2.7206E-01  3.6774E+00/
    CHEB/  1.0033E+00  1.8913E-01 -1.4265E-01  1.0708E-01  1.1757E-01/
    CHEB/ -1.4623E-01 -9.4929E-02 -4.5817E-03 -7.0153E-02 -6.1151E-02/
    CHEB/ -3.3393E-02 -1.9392E-02 -9.5076E-03 -1.5869E-02 -8.9030E-04/
    CHEB/  1.0714E-02  1.0726E-03/
CH2OH + CH2O (+M)     <=> COC. + O (+M)         1.00E+00     .000       0. ! ING181   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.7707E+01 -7.1296E-02 -5.0802E-02  2.9566E+01/
    CHEB/  8.7206E-02  6.1354E-02  4.1510E-02 -1.2480E-02 -7.7945E-03/
    CHEB/ -4.1169E-03 -6.7633E-03 -5.1554E-03 -6.5663E-03 -7.3466E-04/
    CHEB/ -6.6434E-04 -2.3978E-03  8.2975E-04  6.0861E-04  3.0678E-04/
    CHEB/  5.5392E-04  4.2834E-04/
CH2OOH (+M)           <=> CH3O + O (+M)         1.00E+00     .000       0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.1900E+01  2.4713E+00 -2.1186E-02  1.5646E+01/
    CHEB/  3.1595E-02  2.3228E-02  1.0411E-01 -7.4855E-04 -4.2589E-04/
    CHEB/  3.3734E-03 -2.4250E-03 -1.7963E-03 -2.7645E-02 -8.4948E-04/
    CHEB/ -6.4679E-04 -2.3155E-02  6.4128E-06 -1.7254E-06 -1.2591E-02/
    CHEB/  1.6106E-04  1.2022E-04/
CH2OOH (+M)           <=> CH2O + OH (+M)        1.00E+00     .000       0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       8.6405E+00  2.4963E+00 -2.8001E-03  1.1638E-01/
    CHEB/  3.5304E-03  2.6505E-03 -1.0409E-01 -3.2234E-04 -2.4017E-04/
    CHEB/ -2.1004E-02 -1.2670E-05 -9.6422E-06 -7.8210E-03 -1.7268E-05/
    CHEB/ -1.2727E-05 -3.3591E-03 -6.0558E-06 -4.4509E-06 -9.3367E-04/
    CHEB/  4.2036E-07  3.8397E-07/
CH3OO (+M)            <=> CH3O + O (+M)         1.00E+00     .000       0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
```

```
    CHEB/  7   3      -1.2026E+01  2.4705E+00 -2.1734E-02  1.7806E+01/
    CHEB/  3.1539E-02  2.3153E-02 -3.8796E-01  1.9751E-04  2.7895E-04/
    CHEB/ -1.3528E-01 -2.1673E-03 -1.5986E-03 -3.2523E-02 -1.0010E-03/
    CHEB/ -7.5705E-04  1.8921E-03 -1.6456E-04 -1.2970E-04  7.9337E-03/
    CHEB/  9.9761E-05  7.3139E-05/
CH3OO (+M)           <=> CH2OOH (+M)          1.00E+00     .000       0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.1997E+01  4.8377E+00 -1.0901E-01  1.2247E+01/
    CHEB/  1.6314E-01  1.0652E-01 -4.8526E-01  1.3378E-02  1.1820E-02/
    CHEB/ -1.7400E-01 -1.0983E-02 -6.8639E-03 -4.2909E-02 -7.2866E-03/
    CHEB/ -5.2492E-03  2.7689E-03 -1.9804E-03 -1.6169E-03  1.0826E-02/
    CHEB/  3.5005E-04  1.5337E-04/
CH3OO (+M)           <=> CH2O + OH (+M)       1.00E+00     .000       0. ! ING061   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -7.3700E+00  2.3383E+00 -1.0863E-01  1.2506E+01/
    CHEB/  1.6321E-01  1.0661E-01 -4.7067E-01  1.3049E-02  1.1600E-02/
    CHEB/ -1.5754E-01 -1.1172E-02 -7.0045E-03 -3.0539E-02 -7.2980E-03/
    CHEB/ -5.2650E-03  8.6286E-03 -1.9291E-03 -1.5821E-03  1.2828E-02/
    CHEB/  3.9036E-04  1.8339E-04/
CH3OO + CH2O         <=> CH3OOH + HCO         5.60E+12    0.00    13600.
CH3OO + C2H6         <=> C2H5 + CH3OOH        1.70E+13    0.00    20460.
CH3OO + CH3OH        <=> CH2OH + CH3OOH       6.30E+12    0.00    21360.
CH3OO + CH3          <=> CH3O + CH3O          3.80E+12    0.00    -1200.
CH3OO + C2H5         <=> CH3O + CCO.          3.80E+12    0.00    -1200.
CH3OO + CH3OO        <=> CH2O + CH3OH + O2    3.00E+10    0.00     -830.
CH3OO + CH3OO        <=> CH3O + CH3O + O2     3.00E+10    0.00     -830.
CH3OO + C2H4         <=> CYCCO + CH3O         2.82E+11    0.00    17110.
CH4 (+M)             <=> CH3 + H (+M)         1.00E+00     .000       0. ! ING221   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.8596E+01  1.9137E+00 -2.9631E-01  3.2580E+01/
    CHEB/  4.5476E-01  1.9218E-01 -5.0368E-01  9.2796E-02  6.6035E-02/
    CHEB/ -2.5914E-01  8.6217E-03  1.2947E-02 -1.2334E-01 -6.4856E-03/
    CHEB/ -1.6143E-03 -5.5482E-02 -5.1440E-03 -2.9219E-03 -2.4031E-02/
    CHEB/ -2.3138E-03 -1.6043E-03/
CH4 + H              <=> CH3 + H2             1.33E+04    3.00     8038. !92 BAULCH
CH4 + OH             <=> CH3 + H2O            2.36E+07    1.83     2782. !1.5*92 BAULCH
CH4 + O              <=> CH3 + OH             2.13E+06    2.21     6480. !83 MICHAEL
CH4 + HO2            = CH3 + H2O2             9.04E+12    0.00    24641. !92 BAULCH
CH3OH (+M)           <=> CH3 + OH (+M)        1.00E+00     .000       0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.3286E+01  1.4988E+00 -3.4145E-01  2.8377E+01/
    CHEB/  7.4476E-01  1.5818E-01 -6.6933E-01  1.7894E-01  1.0433E-01/
    CHEB/ -3.3830E-01  2.2095E-02  3.1823E-02 -1.5831E-01 -9.8090E-03/
    CHEB/  2.3352E-03 -7.0255E-02 -9.3394E-03 -3.8564E-03 -3.0158E-02/
    CHEB/ -4.5737E-03 -2.8842E-03/
CH3OH (+M)           <=> CH2S + H2O (+M)      1.00E+00     .000       0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.3346E+01  1.1983E+00 -3.1503E-01  2.7778E+01/
    CHEB/  8.3822E-01  6.6408E-02 -6.7314E-01  2.4388E-01  9.8986E-02/
    CHEB/ -3.4435E-01  4.7938E-02  4.2367E-02 -1.6266E-01 -1.7900E-03/
    CHEB/  9.2062E-03 -7.2982E-02 -7.6787E-03 -1.1752E-03 -3.1699E-02/
    CHEB/ -4.6260E-03 -2.1629E-03/
CH3OH (+M)           <=> HCOH + H2 (+M)       1.00E+00     .000       0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.1487E+01  6.7741E-01 -2.0500E-01  2.6451E+01/
    CHEB/  7.9051E-01 -1.0871E-01 -5.9844E-01  3.2619E-01  2.9857E-02/
    CHEB/ -3.2848E-01  1.0064E-01  3.9291E-02 -1.6256E-01  1.9873E-02/
    CHEB/  1.7832E-02 -7.5758E-02 -1.0739E-03  3.7131E-03 -3.4040E-02/
    CHEB/ -3.3077E-03 -7.8687E-04/
CH3OH (+M)           <=> CH2O + H2 (+M)       1.00E+00     .000       0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.6005E+01  1.4993E+00 -3.4108E-01  2.8380E+01/
    CHEB/  7.4579E-01  1.5872E-01 -6.6772E-01  1.7918E-01  1.0446E-01/
    CHEB/ -3.3763E-01  2.1758E-02  3.1710E-02 -1.5810E-01 -1.0117E-02/
    CHEB/  2.1883E-03 -7.0160E-02 -9.4609E-03 -3.9394E-03 -3.0098E-02/
    CHEB/ -4.5914E-03 -2.9090E-03/
CH3OH (+M)           <=> CH2OH + H (+M)       1.00E+00     .000       0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.5560E+01  1.7769E+00 -3.1652E-01  2.9312E+01/
    CHEB/  6.1064E-01  2.1113E-01 -6.6091E-01  1.1144E-01  8.4885E-02/
    CHEB/ -3.3019E-01 -5.6906E-04  1.4088E-02 -1.5270E-01 -1.4540E-02/
    CHEB/ -5.1198E-03 -6.6634E-02 -8.7657E-03 -5.5591E-03 -2.8072E-02/
    CHEB/ -3.3272E-03 -2.7167E-03/
```

```
CH3OH (+M)          <=> CH3O + H (+M)        1.00E+00     .000        0. ! ING101   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -3.0297E+01  2.1580E+00 -1.9965E-01  3.1833E+01/
    CHEB/  3.5152E-01  1.9075E-01 -6.3700E-01  1.8231E-02  2.5620E-02/
    CHEB/ -3.0396E-01 -1.8791E-02 -9.4541E-03 -1.3468E-01 -1.0431E-02/
    CHEB/ -7.6315E-03 -5.6291E-02 -2.6550E-03 -2.4745E-03 -2.2965E-02/
    CHEB/  5.4195E-05 -1.8991E-04/
CH3OH + O           <=> CH2OH + OH           2.91E+05    2.50     3080. !86 TSANG
CH3OH + O           <=> CH3O + OH            9.70E+04    2.50     3080. !86 TSANG
CH3OH + H           <=> CH2OH + H2           3.20E+13    0.0      6095. !84 WARANTZ
CH3OH + H           <=> CH3O + H2            8.00E+12    0.0      6095. !84 WARANTZ
CH3OH + OH          <=> CH2OH + H2O          9.96E+04    2.5      -960. !86TSANG
CH3OH + OH          <=> CH3O + H2O           1.32E+05    2.5      -960. !86TSANG
CH3OH + HO2         <=> CH2OH + H2O2         9.64E+10    0.0     12579. !86 TSANG
CH3OH + CH3         <=> CH2OH + CH4          3.19E+01    3.20     7170. !86 TSANG
CH3OH + CH3         <=> CH3O + CH4           1.44E+01    3.10     6935. !86 TSANG
CH3OOH              <=> CH3O + OH            6.46E+14    0.00    43000.
CH3OOH + O2         <=> CH3OO + HO2          3.00E+12    0.00    39000.
CQ.H2OH (+M)        <=> CH2OHOO (+M)         1.00E+00     .000        0. ! ING011   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       6.5764E+00  2.2108E+00 -2.8065E-01  1.4668E+00/
    CHEB/  8.4505E-01  1.4580E-01 -5.6864E-01  1.2051E-01 -6.4791E-02/
    CHEB/ -1.0833E-01  2.9421E-02 -7.0577E-02  5.9820E-03  3.7417E-02/
    CHEB/ -1.5349E-02  2.9306E-03  2.5798E-02  9.5806E-03 -9.2093E-03/
    CHEB/  8.0294E-03  9.1385E-03/
C2H + O2            <=> HCO + CO             9.00E+11    0.0         0. !JWB NCA
C2H + O2            <=> HCCO + O             9.00E+11    0.0         0. !JWB NCA
C2H + O             <=> CH + CO              5.00E+13    .000        0.
C2H + OH            <=> H + HCCO             2.00E+13    .000        0.
HCCO + O            <=> HCO + CO             3.40E+13    0.00      500.
HCCO + O            <=> H + 2CO              1.00E+14    .000        0.
HCCO + O2           <=> HCO + CO2            3.00E+12    0.0         0.
HCCO + H            <=> CH2S + CO            9.00E+13    0.0         0.
C2H2 + O            <=> OH + C2H             4.60E+19   -1.41    28950.
C2H2 + O2           <=> C*C*O + O            4.00E+12    0.0     28000. !JWB
C2H2 + H (+M)       <=> C2H3 (+M)            1.00E+00     .000        0. ! ING371   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       9.9505E+00  1.8486E+00 -3.0648E-01 -4.0050E-01/
    CHEB/  5.3507E-01  2.0668E-01 -5.6379E-01  1.0741E-01  7.7576E-02/
    CHEB/ -2.7037E-01  5.1829E-03  1.4241E-02 -1.2008E-01 -1.2664E-02/
    CHEB/ -4.2738E-03 -4.7119E-02 -9.5902E-03 -5.7234E-03 -1.4750E-02/
    CHEB/ -4.5560E-03 -3.3187E-03/
C2H2 + H            <=> C2H + H2             6.03E+13    0.0     27821. !92 BAULCH
C2H2 + OH           <=> C2H + H2O            1.45E+04    2.68    12042. !92 JPCRD
C2H2 + O (+M)       <=> HCCO + H (+M)        1.00E+00     .000        0. ! ING321   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.1729E+01 -1.3629E-02 -1.0150E-02  4.8017E-01/
    CHEB/  1.7020E-02  1.2643E-02  7.9983E-02 -2.8096E-03 -2.0439E-03/
    CHEB/  3.3869E-02 -7.5738E-04 -5.8019E-04  1.1376E-02 -8.9164E-05/
    CHEB/ -6.9103E-05  2.7879E-03  5.1650E-05  3.8866E-05  2.4160E-04/
    CHEB/  4.4238E-05  3.3464E-05/
C2H2 + O (+M)       <=> CO + CH2 (+M)        1.00E+00     .000        0. ! ING321   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.2321E+01 -2.1016E-02 -1.5683E-02  2.8691E-01/
    CHEB/  2.6106E-02  1.9440E-02 -3.3189E-02 -4.3143E-03 -3.1579E-03/
    CHEB/ -1.7547E-02 -1.0922E-03 -8.3533E-04 -8.5775E-03 -1.3660E-04/
    CHEB/ -1.0512E-04 -3.9687E-03  6.8281E-05  5.1428E-05 -1.7689E-03/
    CHEB/  6.2862E-05  4.7500E-05/
C2H2 + OH (+M)      <=> CH3 + CO (+M)        1.00E+00     .000        0. ! ING331   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       9.5811E+00 -1.4917E+00 -5.8112E-01  1.1463E+00/
    CHEB/  1.2749E+00  3.6436E-01  3.2797E-03  1.6788E-01  1.3895E-01/
    CHEB/ -6.3254E-02 -4.6408E-02  8.2158E-03 -3.1512E-02 -3.5754E-02/
    CHEB/ -1.5537E-02 -8.4663E-03 -7.9215E-03 -7.4338E-03 -1.0756E-03/
    CHEB/  2.0876E-03 -6.9570E-04/
C2H3 + O            <=> H + C*C*O            3.00E+13    .000        0.
C2H3 + H            <=> C2H2 + H2            1.21E+13    0.0         0. !92 BAULCH
C2H3 + OH           <=> H2O + C2H2           5.00E+12    .000        0.
C2H3 + H (+M)       <=> C2H4 (+M)            6.08E+12    .270      280. !GRI 2.0
     LOW  /  1.400E+30   -3.860   3320.00/
     TROE/   .7820  207.50  2663.00  6095.00 /
     H2/2.00/ H2O/6.00/ CH4/2.00/ CO/1.50/ CO2/2.00/ C2H6/3.00/ AR/ .70/
C2H3 + O2 (+M)      <=> C*COO. (+M)          1.00E+00     .000        0. ! ING271   10/95
```

```
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.0530E+01  2.0361E+00 -2.4110E-01 -1.5695E+00/
     CHEB/  4.8036E-01  2.2567E-01 -6.7872E-01  3.1429E-02  4.1055E-02/
     CHEB/ -2.3864E-01 -3.7278E-02 -1.6239E-02 -4.8798E-02 -2.1886E-02/
     CHEB/ -1.5217E-02  1.0779E-02 -3.9658E-03 -4.4667E-03  1.5762E-02/
     CHEB/  2.9015E-03  1.2123E-03/
C2H3 + O2 (+M)        <=> C2H2 + HO2 (+M)       1.00E+00     .000       0. ! ING271   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.0586E+01 -8.0932E-01 -4.2621E-01  7.2196E-01/
     CHEB/  9.1419E-01  4.4058E-01 -6.1103E-02 -4.3690E-02  2.3182E-02/
     CHEB/ -8.9611E-02 -8.5003E-02 -4.8252E-02 -2.7587E-02 -1.6413E-02/
     CHEB/ -1.6541E-02 -8.0614E-04  8.7669E-03  3.5583E-03  2.9001E-03/
     CHEB/  7.0740E-03  5.0142E-03/
C2H3 + O2 (+M)        <=> CH2O + HCO (+M)       1.00E+00     .000       0. ! ING271   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.2134E+01 -4.4216E-01 -2.3095E-01 -1.4641E-01/
     CHEB/  4.8344E-01  2.3182E-01 -2.4621E-01  1.4188E-02  3.1877E-02/
     CHEB/ -1.4138E-01 -4.4168E-02 -2.1909E-02 -5.6157E-02 -2.0436E-02/
     CHEB/ -1.5275E-02 -1.4680E-02 -1.1636E-03 -2.6773E-03 -9.0923E-04/
     CHEB/  4.1452E-03  2.3018E-03/
C2H3 + O2 (+M)        <=> O*CC*O + H (+M)       1.00E+00     .000       0. ! ING271   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.0813E+01 -4.3764E-01 -2.2867E-01 -2.9839E-02/
     CHEB/  4.8351E-01  2.3258E-01 -1.8651E-01  1.0836E-02  3.0051E-02/
     CHEB/ -1.1166E-01 -4.5631E-02 -2.3100E-02 -4.1386E-02 -2.0143E-02/
     CHEB/ -1.5271E-02 -7.7431E-03 -5.4426E-04 -2.2714E-03  2.0231E-03/
     CHEB/  4.4232E-03  2.5410E-03/
C2H3 + CH3            <=> CH4 + C2H2            3.92E+11     0.0         0. !86 TSANG
C2H3 + CH3OOH         <=> C2H4 + CH3OO          7.00E+11     0.0      1000. !CH3
C2H3 + CCOOH          <=> C2H4 + CCOO           7.00E+11     0.0      1000. !CH3
C2H3 + HO2            <=> C.CHO + OH            1.00E+13     0.0         0. !JWB
C2H3 + HO2            <=> C2H4 + O2             1.00E+12     0.0         0. !JWB
C2H3 + H2O2           <=> C2H4 + HO2            1.00E+12     0.0         1. !JWB EST
CCO2                  <=> CH3+CO2               4.40E+15     0.00    10500.
CCO3                  <=> CC.*O + O2            5.00E+14     0.0     37300.
CCO3 + CH3OO          <=> CCO2 + CH3O + O2      6.00E+10     0.00     -830.
CCO3H + O2            <=> CCO3 + HO2            3.00E+12     0.00    39000.
CCO3H + CH3           <=> CH4 + CCO3            7.43E+11     0.00     5500.
CCO3H + HO2           <=> H2O2 + CCO3           2.40E+12     0.00    10000.
CCO3H + C2H5          <=> C2H6 + CCO3           5.00E+11     0.00     6500.
CCO3H + HCO           <=> CH2O + CCO3           8.00E+11     0.00    10000.
C2H4                  <=> C2H2 + H2             9.33E+13     0.0     77200. !NDISSOC HPL
C2H4 + H (+M)         <=> C2H5 (+M)             1.00E+00     .000       0. ! ING361   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.1661E+01  1.0359E+00 -2.6912E-01 -3.7752E-01/
     CHEB/  9.4025E-01  2.0266E-02 -6.8299E-01  3.5172E-01  1.0581E-01/
     CHEB/ -3.4527E-01  8.8351E-02  6.4621E-02 -1.5108E-01 -5.2684E-03/
     CHEB/  2.0222E-02 -5.2057E-02 -2.4334E-02 -1.6496E-03 -9.0972E-03/
     CHEB/ -1.8380E-02 -6.8157E-03/
     H2/2.00/ H2O/6.00/ CH4/2.00/ CO/1.50/ CO2/2.00/ C2H6/3.00/ AR/ .70/
C2H4 + H              <=> C2H3 + H2             5.42E+14     0.0     14904. !92 BAULCH
C2H4 + CH3            <=> C2H3 + CH4            4.16E+12     0.0     11128. !92 BAULCH
C2H4 + OH             <=> C2H3 + H2O            2.05E+13     0.0      5942. !92 BAULCH
C2H4 + CO             <=> C2H3 + HCO            1.51E+14     0.0     90561. !86 TSANG
C2H4 + O (+M)         <=> C.CHO+ H (+M)         1.00E+00     .000       0. ! ING301   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.2123E+01 -5.5629E-03 -4.1734E-03  4.7606E-01/
     CHEB/  6.4949E-03  4.8663E-03 -3.0064E-02 -1.4580E-03 -1.0861E-03/
     CHEB/ -1.8526E-02 -4.7484E-05 -3.8379E-05 -1.0411E-02  6.5035E-05/
     CHEB/  4.9246E-05 -5.3067E-03  2.1670E-05  1.6592E-05 -2.4514E-03/
     CHEB/  5.4526E-06  4.2551E-06/
C2H4 + O (+M)         <=> CH2O + CH2 (+M)       1.00E+00     .000       0. ! ING301   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.0185E+01 -5.4288E-03 -4.0726E-03  1.3237E+00/
     CHEB/  6.5347E-03  4.8965E-03  3.4442E-01 -1.5248E-03 -1.1365E-03/
     CHEB/  8.4879E-02 -8.7966E-05 -6.8893E-05  4.7002E-04  6.5149E-05/
     CHEB/  4.9201E-05 -1.0173E-02  2.7381E-05  2.0709E-05 -4.8293E-03/
     CHEB/  6.0790E-06  4.7285E-06/
C2H4 + O (+M)         <=> CCHO (+M)             1.00E+00     .000       0. ! ING301   10/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       7.4090E+00  2.2536E+00 -1.6416E-01 -6.2953E-01/
     CHEB/  2.7976E-01  1.8152E-01 -5.8322E-01 -1.3800E-02 -3.4383E-03/
     CHEB/ -2.9128E-01 -1.9412E-02 -1.3279E-02 -1.3435E-01 -6.2285E-03/
```

```
    CHEB/ -4.9381E-03 -5.7490E-02 -1.2264E-04 -3.4308E-04 -2.2987E-02/
    CHEB/  9.7642E-04  6.6584E-04/
C2H4 + O (+M)         <=> CC.*O + H (+M)        1.00E+00     .000        0. ! ING301   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.0149E+01 -2.3605E-01 -1.5735E-01  1.0651E+00/
    CHEB/  2.7994E-01  1.8238E-01  1.4277E-01 -2.1840E-02 -8.9609E-03/
    CHEB/  2.3446E-02 -2.3306E-02 -1.6290E-02 -6.6407E-03 -5.5669E-03/
    CHEB/ -4.6163E-03 -8.7191E-03  1.5801E-03  8.7975E-04 -5.0566E-03/
    CHEB/  1.8518E-03  1.3441E-03/
C2H4 + O (+M)         <=> HCO + CH3 (+M)        1.00E+00     .000        0. ! ING301   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        7.4689E+00 -2.3627E-01 -1.5741E-01  1.0919E+00/
    CHEB/  2.8078E-01  1.8284E-01  1.6253E-01 -2.2322E-02 -9.2605E-03/
    CHEB/  3.0087E-02 -2.3644E-02 -1.6518E-02 -5.1033E-03 -5.4560E-03/
    CHEB/ -4.5536E-03 -8.3850E-03  1.7168E-03  9.7377E-04 -4.9885E-03/
    CHEB/  1.8761E-03  1.3657E-03/
C2H4 + OH (+M)        <=> C.COH (+M)            1.00E+00     .000        0. ! ING311   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.1404E+01  1.2697E+00 -2.6561E-01 -1.5697E+00/
    CHEB/  9.9945E-01  1.0505E-01 -7.7960E-01  2.5859E-01  1.2033E-01/
    CHEB/ -3.2208E-01  2.8545E-03  3.8799E-02 -9.8984E-02 -4.2643E-02/
    CHEB/ -4.5785E-03 -8.5909E-03 -2.8262E-02 -1.2254E-02  1.6777E-02/
    CHEB/ -9.8758E-03 -7.1983E-03/
C2H4 + OH (+M)        <=> C*COH + H (+M)        1.00E+00     .000        0. ! ING311   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.0602E+01 -8.1901E-01 -2.7411E-01  1.2797E+00/
    CHEB/  8.6331E-01  2.2662E-01  1.5478E-02  6.9200E-02  9.1985E-02/
    CHEB/ -5.4971E-02 -7.3546E-02 -1.2791E-02 -2.6334E-02 -4.4509E-02/
    CHEB/ -2.7166E-02 -4.4020E-03 -1.0157E-02 -1.1927E-02  3.1007E-03/
    CHEB/  4.0671E-03 -2.2874E-04/
C2H4 + OH (+M)        <=> CCO. (+M)             1.00E+00     .000        0. ! ING311   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        3.3398E+00  1.9388E+00 -3.0608E-01  2.0686E+00/
    CHEB/  6.0856E-01  3.1033E-01 -4.8489E-01  1.8517E-02  3.3886E-02/
    CHEB/ -1.4075E-01 -4.6626E-02 -2.2520E-02 -2.2978E-03 -2.4657E-02/
    CHEB/ -1.7092E-02  2.9409E-02 -4.9817E-03 -5.2428E-03  2.1908E-02/
    CHEB/  2.4699E-03  8.0522E-04/
C2H4 + OH (+M)        <=> CH2O + CH3 (+M)       1.00E+00     .000        0. ! ING311   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.4834E+00 -5.3567E-01 -2.9302E-01  3.5421E+00/
    CHEB/  6.0454E-01  3.1124E-01 -2.0973E-02 -5.2460E-04  2.3426E-02/
    CHEB/ -3.6225E-02 -5.3640E-02 -2.8138E-02 -1.3059E-02 -2.2729E-02/
    CHEB/ -1.6861E-02 -4.8853E-04 -1.5579E-03 -3.1417E-03  2.4722E-03/
    CHEB/  4.2520E-03  2.2396E-03/
C2H5 + O              <=> CCHO + H              8.02E+13     0.0         0. !86 TSANG
C2H5 + O              <=> CH2O + CH3            1.61E+13     0.0         0. !86 TSANG
C2H5 + H              <=> H2 + C2H4             2.00E+12     .00         0. ! GRI 2.0
C2H5 + OH             <=> C2H4 + H2O            2.41E+13     0.0         0. !86TSANG
C2H5 + OH             <=> CCOH                  3.70E+13     0.0         0.
C2H5 + HO2            <=> CCO. + OH             3.00E+13     0.00        0. !90 BOZZELLI
C2H5 + O2 (+M)        <=> CCOO (+M)             1.00E+00      .000       0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.0735E+01  1.3744E+00 -2.3621E-01 -2.1377E+00/
    CHEB/  1.0402E+00  1.3597E-01 -1.0038E+00  1.8687E-01  1.0776E-01/
    CHEB/ -3.4977E-01 -4.9674E-02  1.2454E-02 -6.2474E-02 -5.3407E-02/
    CHEB/ -1.6309E-02  2.5384E-02 -1.9260E-02 -1.0816E-02  3.1201E-02/
    CHEB/ -3.9837E-04 -2.6200E-03/
C2H5 + O2 (+M)        <=> CCO. + O (+M)         1.00E+00     .000        0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        8.6395E-01 -3.5874E-02 -2.6224E-02  9.0285E+00/
    CHEB/  4.7385E-02  3.4471E-02  1.3177E-01 -8.4185E-03 -5.8493E-03/
    CHEB/ -9.1029E-02 -4.2843E-03 -3.2656E-03 -7.6511E-02 -2.2442E-04/
    CHEB/ -1.9576E-04  5.4170E-02  1.4295E-03  1.0634E-03  4.4774E-02/
    CHEB/  5.4436E-04  4.1403E-04/
C2H5 + O2 (+M)        <=> C.COOH (+M)           1.00E+00     .000        0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        8.1555E+00  7.5101E-01 -4.7089E-01 -1.1449E+00/
    CHEB/  1.7970E+00  3.7681E-01 -7.0573E-01  1.3593E-01  1.3922E-01/
    CHEB/ -2.0334E-01 -1.3805E-01 -2.3025E-02  4.3746E-03 -7.0020E-02/
    CHEB/ -3.1185E-02  3.2871E-02 -9.8451E-03 -8.8708E-03  1.3784E-02/
    CHEB/  7.2463E-03  1.3034E-03/
C2H5 + O2 (+M)        <=> C2H4 + HO2 (+M)       1.00E+00     .000        0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
```

```
    CHEB/  7   3       9.4234E+00 -1.4042E+00 -4.3255E-01  7.3608E-01/
    CHEB/  1.6443E+00  4.1830E-01 -2.3213E-01 -7.0142E-02  8.9408E-02/
    CHEB/ -1.1704E-01 -1.6800E-01 -5.6430E-02 -9.7358E-03 -4.0106E-02/
    CHEB/ -3.3267E-02  1.3130E-02  1.2841E-02 -1.7757E-03  8.2939E-03/
    CHEB/  1.4444E-02  6.4434E-03/
C2H5 + O2 (+M)        <=> CC.OOH (+M)          1.00E+00     .000       0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       1.6036E+00  2.1727E+00 -1.7892E-01  2.4364E+00/
    CHEB/  4.1352E-01  2.1530E-01 -1.3140E-01 -4.7130E-02 -8.5493E-03/
    CHEB/ -6.5423E-02 -4.6386E-02 -2.9449E-02 -9.5444E-03 -2.7558E-03/
    CHEB/ -5.1087E-03  5.8820E-03  7.4744E-03  4.2414E-03  2.5381E-03/
    CHEB/  3.4470E-03  2.7768E-03/
C2H5 + O2 (+M)        <=> CH3CHO + OH (+M)     1.00E+00     .000       0. ! ING261   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       6.7877E+00 -3.2641E-01 -1.7846E-01  2.6966E+00/
    CHEB/  4.1301E-01  2.1512E-01 -4.9962E-02 -4.7832E-02 -8.9761E-03/
    CHEB/ -2.5781E-02 -4.6276E-02 -2.9445E-02  2.3385E-04 -2.5214E-03/
    CHEB/ -4.9647E-03  4.2630E-03  7.5230E-03  4.2938E-03  1.5240E-03/
    CHEB/  3.4059E-03  2.7582E-03/
C2H5 + CH3            <=> CH4 + C2H4           1.15E+12    0.0         0. !92 BAULCH
C2H5 + C2H3           <=> C2H6 + C2H2          4.80E+11   0.0         0.
C2H5 + C2H3           <=> C2H4 + C2H4          3.00E+12    0.00        0.
C2H5 + C2H4           <=> C2H6 + C2H3          6.30E+02   3.13   18000.
C2H5 + C2H5           <=> C2H6 + C2H4          1.45E+12    0.00        0. !92 BAULCH
C2H6 (+M)             <=> C2H5 + H (+M)        1.00E+00     .000       0. ! ING231   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.5461E+01  1.6191E+00 -3.2749E-01  3.0034E+01/
    CHEB/  7.2967E-01  2.0052E-01 -8.3003E-01  1.7291E-01  1.0877E-01/
    CHEB/ -4.2963E-01  7.1637E-03  2.5984E-02 -2.0513E-01 -2.3606E-02/
    CHEB/ -5.3012E-03 -9.1235E-02 -1.6790E-02 -9.1525E-03 -3.7614E-02/
    CHEB/ -7.0179E-03 -5.2955E-03/
C2H6 + O2             <=> C2H5 + HO2           6.03E+13    0.00    51866. !92 BAULCH
C2H6 + HO2            <=> C2H5 + H2O2          1.70E+13    0.00    20460. !86 BALDWIN
C2H6 + CH3            <=> C2H5 + CH4           1.51E-07    6.00     6047. !92 BAULCH
C2H6 + H              <=> C2H5 + H2            1.44E+09    1.50     7412. !92 BAULCH
C2H6 + OH             <=> C2H5 + H2O           7.22E+06    2.00      864. !92 BAULCH
C2H6 + O              <=> C2H5 + OH            9.99E+08    1.50     5803. !92 BAULCH
CCOH + O2             <=> C.COH +HO2           2.00E+13    0.0     53800.
CCOH + O2             <=> CC.OH+HO2            1.33E+13    0.0     48400.
CCOH + OH             <=> C.COH + H2O          7.55E+09    0.97     1586.
CCOH + OH             <=> CC.OH + H2O          6.78E+07    1.61       35.
CCOH + OH             <=> CCO. + H2O           5.00E+09    0.97     2586.
CCOH + H              <=> C.COH + H2           9.60E+06    2.0      7700.
CCOH + H              <=> CC.OH + H2           2.52E+14    0.0      7300.
CCOH + H              <=> CCO. + H2            6.40E+06    0.0     10700.
CCOH + O              <=> C.COH + OH           2.25E+13    0.0      7700.
CCOH + O              <=> CC.OH + OH           3.07E+13    0.0      5200.
CCOH + CH3            <=> C.COH + CH4          3.19E+01    3.17     7172.
CCOH + CH3            <=> CC.OH + CH4          1.74E+00    3.46     5480.
CCOH + C2H5           <=> C.COH + C2H6         5.00E+10    0.0     13400.
CCOH + C2H5           <=> CC.OH + C2H6         3.30E+10    0.0     10400.
CCOH + HO2            <=> C.COH + H2O2         8.40E+12    0.0     20430.
CCOH + HO2            <=> CC.OH + H2O2         5.60E+12    0.0     17700.
CCOH + C2H3           <=> C.COH + C2H4         5.00E+11    0.0     10400.
CCOH + C2H3           <=> CC.OH + C2H4         3.30E+11    0.0      5200.
COC (+M)              <=> COC. + H (+M)        1.00E+00     .000       0. ! ING131   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.6507E+01  1.6551E+00 -3.0699E-01  2.9143E+01/
    CHEB/  8.0116E-01  2.2723E-01 -1.2081E+00  1.4867E-01  1.1258E-01/
    CHEB/ -6.0669E-01 -3.5496E-02  9.3434E-03 -2.8167E-01 -4.9669E-02/
    CHEB/ -2.2572E-02 -1.1780E-01 -2.4160E-02 -1.7769E-02 -4.2397E-02/
    CHEB/ -4.6646E-03 -6.4519E-03/
COC (+M)              <=> CH2S + CH3OH (+M)    1.00E+00     .000       0. ! ING131   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.5427E+01  1.6525E+00 -3.0760E-01  2.9118E+01/
    CHEB/  8.0116E-01  2.2643E-01 -1.2159E+00  1.5036E-01  1.1317E-01/
    CHEB/ -6.0918E-01 -3.4008E-02  1.0202E-02 -2.8231E-01 -4.9312E-02/
    CHEB/ -2.2196E-02 -1.1812E-01 -2.4555E-02 -1.7855E-02 -4.2648E-02/
    CHEB/ -5.0827E-03 -6.6719E-03/
COC + O               <=> COC. + OH            3.24E+12    0.0      2623. !90 LIU
COC + H               <=> COC. + H2            1.90E+13    0.0      5167. !79 FAUBEL
COC + OH              <=> COC. + H2O           6.27E+12    0.0       739. !87 TULLY
COC + CH3             <=> COC. + CH4           3.55E+12    0.0     11800. !82 BATT
```

```
CCOOH                 <=> CCO. + OH             6.46E+14    0.00    43000.
CCOOH + O2            <=> CCOO + HO2            3.00E+12    0.00    39000.
COCOH (+M)            <=> COCO. + H (+M)        1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.6877E+01  5.8798E-01 -1.3647E-01  3.1295E+01/
    CHEB/  8.7664E-01 -1.1799E-01 -6.6639E-01  4.9848E-01  8.5840E-03/
    CHEB/ -3.8359E-01  2.1884E-01  5.4383E-02 -2.0767E-01  6.7834E-02/
    CHEB/  4.5988E-02 -1.0438E-01  4.4182E-03  2.3454E-02 -4.8968E-02/
    CHEB/ -1.4331E-02  6.6350E-03/
COCOH (+M)            <=> COC.OH + H (+M)       1.00E+00     .000       0. ! ING151   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.2596E+01  3.6088E-01 -6.4998E-02  2.8374E+01/
    CHEB/  6.4162E-01 -1.0043E-01 -5.4112E-01  4.5606E-01 -4.0522E-02/
    CHEB/ -3.4782E-01  2.5983E-01  7.7360E-03 -2.0505E-01  1.1514E-01/
    CHEB/  2.6526E-02 -1.0973E-01  3.3948E-02  2.3184E-02 -5.4873E-02/
    CHEB/ -5.7950E-04  1.2342E-02/
C#CC + H              <=> C#CC. + H2            5.00E+13   0.0     5000.
C#CC + H              <=> CC*C.                 2.59E+34  -6.587  12401.
C#CC + CH3            <=> C#CC. + CH4           3.80E+11   0.0     9000.
C*C*C + H             <=> CC*C.                 6.38E+25  -4.047  14693.
C*C*C + H             <=> C#CC. + H2            5.00E+13   0.0     5000.
C*C*C + CH3           <=> C2.C*C                6.60E+10   0.0     7990.
C*C*C + CH3           <=> C#CC. + CH4           3.80E+11   0.0     9000.
C*C*C + H (+M)        <=> C#CC + H (+M)         1.00E+00     .000       0. ! ING391   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       6.0892E+00 -2.6850E+00 -3.7644E-01  3.9672E+00/
    CHEB/  2.1188E+00  8.8175E-02  3.9106E-01  5.9831E-01  1.5674E-01/
    CHEB/ -9.3986E-02  4.5723E-02  7.5101E-02 -8.9163E-02 -6.8816E-02/
    CHEB/  2.5153E-02 -3.7449E-02 -5.2734E-02  1.3932E-03 -1.0298E-02/
    CHEB/ -2.6410E-02 -7.2624E-03/
C*CC. + H             <=> C*C*C + H2            3.60E+12   0.0        0.
C*CC. + CH3           <=> C*C*C + CH4           8.92E+11   0.0    -1571.
C*CC. + CH3O          <=> C*C*C + CH3OH         1.00E+13   0.0        0.
C*CC. + C2H5          <=> C*C*C + C2H6          2.40E+12   0.0        0.
C*CC. + C2H5          <=> C*CC + C2H4           1.20E+12   0.0        0.
C*CC. + C*CC.         <=> C*CC + C*C*C          1.00E+12   0.0        0.
C.*CC (+M)            <=> C#CC + H (+M)         1.00E+00     .000       0. ! ING381   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -4.5209E+00  1.0416E+00 -2.4309E-01  1.0338E+01/
    CHEB/  1.0137E+00  2.3092E-02 -6.9760E-01  3.4655E-01  1.1362E-01/
    CHEB/ -2.8827E-01  4.8651E-02  6.0548E-02 -9.1424E-02 -3.7335E-02/
    CHEB/  7.7984E-03 -9.6363E-03 -3.6696E-02 -1.2765E-02  1.5280E-02/
    CHEB/ -1.6660E-02 -1.2175E-02/
C.*CC (+M)            <=> C*CC. (+M)            1.00E+00     .000       0. ! ING381   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.8051E+00  1.1978E+00 -2.8334E-01  1.0063E+01/
    CHEB/  1.2971E+00 -4.3697E-02 -9.0963E-01  5.5930E-01  7.6091E-02/
    CHEB/ -4.3400E-01  1.8084E-01  5.0334E-02 -1.8209E-01  3.0332E-02/
    CHEB/  1.3010E-02 -6.0826E-02 -9.5896E-03 -3.3478E-03 -1.0956E-02/
    CHEB/ -1.0159E-02 -4.6454E-03/
C.*CC (+M)            <=> C*C*C + H (+M)        1.00E+00     .000       0. ! ING381   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -6.6715E+00 -5.4142E-01 -3.6237E-01  1.2971E+01/
    CHEB/  1.9400E+00  2.6823E-02 -2.4068E-01  7.0789E-01  1.1400E-01/
    CHEB/ -2.4635E-01  1.7081E-01  5.6233E-02 -1.1411E-01 -7.9385E-04/
    CHEB/  1.9063E-02 -2.8871E-02 -3.1066E-02  7.5524E-03  3.2105E-03/
    CHEB/ -2.2730E-02  4.0548E-03/
C*CC. (+M)            <=> C#CC + H (+M)         1.00E+00     .000       0. ! ING381   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.4761E+01 -7.5547E-01 -3.2468E-01  1.9626E+01/
    CHEB/  1.8941E+00 -2.5941E-02 -3.9897E-01  7.0383E-01  1.0431E-01/
    CHEB/ -3.8665E-01  1.8770E-01  4.9141E-02 -2.1272E-01  1.9231E-02/
    CHEB/  1.1641E-02 -9.1182E-02 -1.7357E-02  3.6735E-03 -2.8652E-02/
    CHEB/ -1.5285E-02  3.4793E-03/
C*CC. (+M)            <=> C*C*C + H (+M)        1.00E+00     .000       0. ! ING381   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.2521E+01  4.5196E-01 -9.2834E-02  1.8954E+01/
    CHEB/  7.7481E-01 -1.3182E-01 -5.3330E-01  5.0514E-01 -3.7759E-02/
    CHEB/ -3.2329E-01  2.5226E-01  2.4120E-02 -1.6529E-01  8.9490E-02/
    CHEB/  3.8772E-02 -7.3952E-02  1.1641E-02  2.7438E-02 -2.8424E-02/
    CHEB/ -1.4114E-02  1.1927E-02/
C*CC + CH3O           <=> C*CC. + CH3OH         1.00E+13   0.0     8000.
C*CC + CH2OH          <=> C*CC. + CH3OH         1.00E+11   0.0     9000.
```

```
C*CC + H              <=> C*CC. + H2           5.00E+13   0.0      5000.
C*CC + CH3            <=> C*CC. + CH4          3.80E+11   0.0      9000.
C*CC + C2H5           <=> C*CC. + C2H6         3.80E+11   0.0      9000.
C*CC + H (+M)         <=> CC.C (+M)            1.00E+00     .000       0. ! ING411   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.1711E+01  6.6497E-01 -1.0609E-01 -7.9259E-01/
    CHEB/  1.0395E+00 -9.8903E-02 -7.6574E-01  5.4787E-01  3.0518E-02/
    CHEB/ -3.9857E-01  1.7991E-01  6.7713E-02 -1.6231E-01  4.4191E-03/
    CHEB/  4.2964E-02 -4.1519E-02 -4.3125E-02  1.0836E-02  6.6360E-03/
    CHEB/ -3.6299E-02 -5.7006E-03/
C*CC + CH3O (+M)      <=> C*C(C)OC + H (+M)    1.00E+00     .000       0. ! 5        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        4.1070E+00 -8.7270E-01 -2.7914E-01  4.5988E+00/
    CHEB/  9.3946E-01  2.1813E-01  1.8532E-02  4.6188E-02  1.0577E-01/
    CHEB/ -1.7723E-01 -9.9835E-02 -2.0923E-02 -6.3218E-02 -4.0947E-02/
    CHEB/ -3.3002E-02  1.8477E-03  6.1527E-03 -6.6753E-03  9.2912E-03/
    CHEB/  1.4211E-02  7.2422E-03/
C*CC + CH3O (+M)      <=> C*COC + CH3 (+M)     1.00E+00     .000       0. ! 5        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        4.7760E+00 -8.3729E-01 -2.7903E-01  5.0813E+00/
    CHEB/  9.1246E-01  2.2772E-01  6.0932E-02  3.4263E-02  9.8536E-02/
    CHEB/ -1.6683E-01 -9.8455E-02 -2.5086E-02 -5.6065E-02 -3.8501E-02/
    CHEB/ -3.2228E-02  4.1200E-03  6.9839E-03 -5.2385E-03  8.7779E-03/
    CHEB/  1.4083E-02  7.6527E-03/
C*CC + CH3O (+M)      <=> CC.C + CH2O (+M)     1.00E+00     .000       0. ! 5        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        5.8536E+00 -1.1020E+00 -3.8008E-01  3.4134E+00/
    CHEB/  1.1790E+00  3.2758E-01 -2.8111E-01  4.1923E-02  8.6319E-02/
    CHEB/ -2.2608E-01 -1.0900E-01 -1.6366E-02 -8.3448E-02 -4.1817E-02/
    CHEB/ -2.5626E-02 -1.0085E-02  6.5101E-03 -8.0557E-03  7.8720E-03/
    CHEB/  1.4197E-02  4.3569E-03/
C*CC + CH3O (+M)      <=> C2C*O + CH3 (+M)     1.00E+00     .000       0. ! 5        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.2559E+00 -7.2924E-01 -2.9485E-01  6.7868E+00/
    CHEB/  8.1702E-01  2.7977E-01  6.0883E-02  1.0433E-02  6.2996E-02/
    CHEB/ -1.1593E-01 -9.4839E-02 -3.3255E-02 -4.0637E-02 -3.1763E-02/
    CHEB/ -2.7610E-02  5.2094E-03  1.0716E-02 -1.0344E-03  7.4375E-03/
    CHEB/  1.3707E-02  8.4128E-03/
C2C*O + CH3 (+M)      <=> C*C(C)OC + H (+M)    1.00E+00     .000       0. ! 6        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -2.8661E+00 -4.5725E-02 -3.2991E-02  1.1052E+01/
    CHEB/  6.3955E-02  4.5825E-02 -9.6147E-03 -1.9165E-02 -1.3255E-02/
    CHEB/ -1.5945E-03 -2.2811E-03 -1.9898E-03  7.9706E-04  2.2525E-03/
    CHEB/  1.6491E-03  4.6627E-05  6.4720E-04  5.2272E-04 -3.4303E-04/
    CHEB/ -2.5841E-04 -1.8839E-04/
C2C*O + CH3 (+M)      <=> C*COC + CH3 (+M)     1.00E+00     .000       0. ! 6        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -7.4005E+00 -1.1407E+00 -2.3131E-01  1.4862E+01/
    CHEB/  1.2414E+00  1.4957E-01 -3.4745E-02  9.4026E-02  1.3482E-01/
    CHEB/ -1.2057E-01 -1.5328E-01 -5.4794E-03 -4.0224E-02 -7.8482E-02/
    CHEB/ -4.8492E-02  7.5231E-03  1.2346E-03 -1.9777E-02  1.5064E-02/
    CHEB/  2.3985E-02  6.9270E-03/
CCC.                  <=> C*CC + H             1.20E+13   0.0    38500.
CCC. + CH3O           <=> C*CC + CH3OH         5.00E+12   0.0        0.
CC.C + CH3            <=> C3C                  1.03E+15  -0.645    -83.
CC.C + CH2O (+M)      <=> C2C*O + CH3 (+M)     1.00E+00     .000       0. ! 4        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        5.3548E+00 -7.5902E-01 -2.2464E-01  2.6527E+00/
    CHEB/  9.7975E-01  2.4420E-01 -1.8458E-01 -1.3263E-01  3.4988E-02/
    CHEB/ -6.3400E-02 -1.1934E-01 -5.3680E-02  1.1562E-02  2.3418E-03/
    CHEB/ -1.7761E-02  1.5364E-02  2.6561E-02  1.0628E-02  3.0888E-03/
    CHEB/  8.6018E-03  8.5205E-03/
CC.C + CH2O (+M)      <=> C*C(C)OC + H (+M)    1.00E+00     .000       0. ! 4        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        5.2475E-02 -5.4446E-01 -2.0545E-01  4.9211E+00/
    CHEB/  7.6468E-01  2.5996E-01  4.6232E-01 -1.9770E-01 -1.9121E-02/
    CHEB/  7.1296E-02 -6.0495E-02 -5.1725E-02  2.3408E-02  2.7889E-02/
    CHEB/  4.8819E-03  1.4732E-02  1.6268E-02  1.3543E-02  6.2930E-03/
    CHEB/ -2.7417E-03  9.9976E-04/
CC.C + CH2O (+M)      <=> C*COC + CH3 (+M)     1.00E+00     .000       0. ! 4        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -4.2305E-02 -1.0913E+00 -3.7797E-01  7.9163E+00/
    CHEB/  1.1737E+00  3.2852E-01 -2.8868E-01  3.5882E-02  8.3657E-02/
```

```
    CHEB/ -2.3529E-01 -1.1078E-01 -1.7248E-02 -8.5460E-02 -3.9351E-02/
    CHEB/ -2.5191E-02 -2.5396E-03  8.1154E-03 -7.4566E-03  1.4640E-02/
    CHEB/  1.3930E-02  4.5836E-03/
CCC + H              <=> CC.C + H2            1.30E+06   2.40    4470.
CCC + H              <=> CCC. + H2            1.30E+06   2.54    6760.
CCC + CH3            <=> CC.C + CH4           9.00E-02   4.00    6285.
CCC + CH3            <=> CCC. + CH4           2.70E-01   4.00    8285.
COCOC (+M)           <=> COC.OC + H (+M)      1.00E+00     .000       0. ! ING201   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.3533E+01  4.8162E-01 -7.4361E-02  2.9241E+01/
    CHEB/  8.5635E-01 -1.1302E-01 -7.5928E-01  5.9877E-01 -3.6074E-02/
    CHEB/ -4.8983E-01  3.1714E-01  2.8559E-02 -2.7568E-01  1.0828E-01/
    CHEB/  5.1681E-02 -1.2534E-01 -1.6503E-03  4.0270E-02 -4.1010E-02/
    CHEB/ -3.7194E-02  1.7336E-02/
COCOC (+M)           <=> COCOC. + H (+M)      1.00E+00     .000       0. ! ING201   10/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.4053E+01  5.2176E-01 -8.1832E-02  3.0173E+01/
    CHEB/  9.1282E-01 -1.1757E-01 -7.8858E-01  6.1075E-01 -2.5095E-02/
    CHEB/ -4.9860E-01  2.9892E-01  4.3968E-02 -2.7124E-01  8.4611E-02/
    CHEB/  6.0297E-02 -1.1800E-01 -1.6735E-02  4.0269E-02 -3.5791E-02/
    CHEB/ -4.2592E-02  1.3226E-02/
C2.C*C + C*CC.       <=> C2C*C + C*C*C        2.40E+12   0.0       0.
C2.C*C (+M)          <=> C*C*C + CH3 (+M)     1.00E+00     .000       0. ! 9       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -9.7005E+00  3.6660E-01 -6.8241E-02  1.7178E+01/
    CHEB/  6.6726E-01 -1.1482E-01 -4.6016E-01  4.9957E-01 -6.3115E-02/
    CHEB/ -3.0647E-01  2.9934E-01 -9.9982E-03 -1.7440E-01  1.3168E-01/
    CHEB/  2.2047E-02 -7.8060E-02  2.7695E-02  2.8555E-02 -2.1900E-02/
    CHEB/ -1.6864E-02  1.9223E-02/
C2C*C + CH3O         <=> C2.C*C + CH3OH       2.00E+13   0.0     8000.
C2C*C + CH2OH        <=> C2.C*C + CH3OH       2.00E+11   0.0     9000.
C2C*C                <=> C2.C*C + H           3.98E+15   0.0    80160.
C2C*C + H            <=> C2.C*C + H2          1.30E+14   0.0     5000.
C2C*C + H (+M)       <=> C*CC + CH3 (+M)      1.00E+00     .000       0. ! 1       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       7.4482E+00 -2.6228E+00 -3.4959E-01  5.0054E+00/
    CHEB/  2.4747E+00  8.9299E-02  7.7543E-02  5.0870E-01  2.5244E-01/
    CHEB/ -2.6042E-01 -1.7835E-01  7.4616E-02 -1.2890E-01 -1.9486E-01/
    CHEB/ -3.9645E-02 -2.0496E-02 -6.6602E-02 -4.6045E-02  1.8015E-02/
    CHEB/  1.1285E-02 -1.6693E-02/
C2CC.                <=> C*CC + CH3           4.00E+13   0.0    30240.
C2CC.                <=> C2C*C + H            6.00E+12   0.0    33800.
C2CC. + H            <=> C2C*C + H2           1.81E+12   0.0        0.
C2CC. + OH           <=> C2C*C + H2O          2.41E+13   0.0        0.
C2CC. + HO2          <=> C2C*C + H2O2         3.01E+11   0.0        0.
C2CC. + CH3          <=> C2C*C + CH4          1.13E+12   0.50       0.
C2CC. + CH3O         <=> C2C*C + CH3OH        2.40E+12   0.50       0.
C2CC. + CH2OH        <=> C2C*C + CH3OH        2.40E+11   0.0        0.
C3C. (+M)            <=> C2C*C + H (+M)       1.00E+00     .000       0. ! 1       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.6283E+00  6.2949E-01 -7.7022E-02  1.1093E+01/
    CHEB/  1.0551E+00 -9.5476E-02 -7.8101E-01  6.1124E-01  7.8104E-03/
    CHEB/ -3.9481E-01  2.0999E-01  5.9686E-02 -1.3751E-01 -7.8303E-03/
    CHEB/  4.7236E-02 -1.3134E-02 -6.6442E-02  1.3405E-02  2.3964E-02/
    CHEB/ -4.8844E-02 -7.5420E-03/
C3C. (+M)            <=> C2CC. (+M)           1.00E+00     .000       0. ! 1       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -5.3751E+00  1.5984E+00 -3.0910E-01  1.1247E+01/
    CHEB/  1.9855E+00 -9.5344E-02 -1.2854E+00  8.0599E-01  1.2235E-01/
    CHEB/ -5.4002E-01  1.6689E-01  1.0033E-01 -1.5079E-01 -4.8741E-02/
    CHEB/  3.1704E-02 -6.1930E-03 -6.1542E-02 -5.0033E-03  1.6784E-02/
    CHEB/ -2.0795E-02 -1.0257E-02/
C3C. (+M)            <=> C*CC + CH3 (+M)      1.00E+00     .000       0. ! 1       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -5.5756E+00 -5.1843E-01 -3.1783E-01  1.3428E+01/
    CHEB/  2.3496E+00 -1.8812E-02 -7.3214E-01  7.7336E-01  1.9302E-01/
    CHEB/ -4.6072E-01  2.6103E-02  1.0759E-01 -1.5634E-01 -1.4630E-01/
    CHEB/  7.2685E-03  1.0183E-03 -9.9687E-02 -2.5407E-02  4.3032E-02/
    CHEB/ -2.7567E-02 -1.7409E-02/
C2CC. (+M)           <=> C2C*C + H (+M)       1.00E+00     .000       0. ! 1       11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -6.9005E+00  1.2208E+00  4.0735E-01  7.3612E+00/
    CHEB/ -1.0856E+00  2.0757E-02 -1.8529E+00 -1.0278E-01 -1.8588E-01/
```

```
    CHEB/ -7.6064E-01  5.7582E-02 -2.1981E-02 -2.8726E-01  5.6988E-02/
    CHEB/  5.0157E-02 -1.2141E-01  3.2389E-02  5.2729E-02 -6.5146E-02/
    CHEB/  8.5921E-03  3.1326E-02/
C2CC. (+M)          <=> C*CC + CH3 (+M)       1.00E+00     .000       0. ! 1        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -3.2507E-01  3.2128E-01 -1.6139E-01  6.8265E+00/
    CHEB/  4.4189E-01 -2.0771E-01 -1.7419E+00  1.3669E-01 -5.0554E-02/
    CHEB/ -8.2097E-01 -2.9636E-02  1.8200E-02 -3.1557E-01 -6.4498E-02/
    CHEB/  2.2561E-02 -1.1320E-01 -4.4414E-02  1.0452E-02 -4.7971E-02/
    CHEB/ -1.5760E-02  1.1400E-03/
C3C. + H            <=> C2C*C + H2            5.43E+12   0.0         0.
C3C. + OH           <=> C2C*C + H2O           7.23E+13   0.0         0.
C3C. + HO2          <=> C2C*C + H2O2          9.03E+11   0.0         0.
C3C. + CH3          <=> C2C*C + CH4           3.39E+12   0.50        0.
C3C. + CH3O         <=> C2C*C + CH3OH         7.20E+12   0.50        0.
C3C. + CH2OH        <=> C2C*C + CH3OH         7.21E+11   0.0         0.
C3C. + O2           <=> C3COO                 4.46E+07   0.0    -15700.
C3C. + O2           <=> C3CO. + O             9.84E+10   0.0     25600.
C3C. + O2           <=> C3.COOH               3.69E+03   0.0    -12700.
C3C. + O2           <=> C2C*C + HO2           1.77E+09   0.0     -4520.
C3C. + CH2O (+M)    <=> C3COC. (+M)           1.00E+00     .000       0. ! 3        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.3725E+00  1.1225E+00 -1.1608E-01  1.5442E-01/
    CHEB/  1.3708E+00  9.0966E-03 -8.5187E-01  2.6187E-01  1.1863E-01/
    CHEB/ -1.2650E-01 -1.6201E-01  3.2708E-02  7.3823E-02 -1.2332E-01/
    CHEB/ -3.0450E-02  3.8111E-02 -1.1710E-02 -2.3122E-02 -1.9656E-02/
    CHEB/  2.8522E-02 -4.7857E-04/
C3C. + CH2O (+M)    <=> C3.COC (+M)           1.00E+00     .000       0. ! 3        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        2.9332E+00 -1.4202E-01 -2.7932E-01  2.4383E+00/
    CHEB/  2.4103E+00  9.4274E-02 -1.2284E+00  4.6979E-01  1.7090E-01/
    CHEB/ -5.4431E-01 -1.1259E-01  3.5014E-02 -1.0157E-01 -1.0471E-01/
    CHEB/ -1.1598E-02  1.7296E-02 -1.8557E-02 -7.7000E-03  1.1683E-02/
    CHEB/  9.4516E-03 -2.6576E-03/
C3C. + CH2O (+M)    <=> C2C*C + CH3O (+M)     1.00E+00     .000       0. ! 3        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.7836E+00 -1.7691E+00 -3.7335E-01  7.0454E+00/
    CHEB/  2.0113E+00  3.1086E-01 -6.0663E-01  3.6975E-02  1.3987E-01/
    CHEB/ -2.9980E-01 -2.5862E-01 -3.5617E-02 -1.3844E-02 -7.9999E-02/
    CHEB/ -4.9952E-02  4.8473E-02  2.9652E-02 -8.7190E-03  2.4056E-02/
    CHEB/  3.2185E-02  1.0929E-02/
C3C. + CH2O (+M)    <=> C*C(C)OC + CH3 (+M)   1.00E+00     .000       0. ! 3        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        2.6057E-01 -1.6804E+00 -3.8358E-01  7.4249E+00/
    CHEB/  1.9314E+00  3.3366E-01 -5.8145E-01  1.3857E-02  1.3177E-01/
    CHEB/ -2.7612E-01 -2.5399E-01 -4.3704E-02 -1.1913E-02 -7.1024E-02/
    CHEB/ -4.9914E-02  4.4586E-02  3.2856E-02 -5.6095E-03  2.0938E-02/
    CHEB/  3.1295E-02  1.2425E-02/
C3C. + CH3O (+M)    <=> C2C*C + CH3OH (+M)    1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        9.0843E+00 -1.8190E+00 -6.0258E-02  2.4045E+00/
    CHEB/  1.1597E+00 -7.4971E-02  7.4690E-02  6.8454E-01  1.6506E-02/
    CHEB/ -4.5029E-01  2.0979E-01  7.3006E-02 -3.1521E-01 -6.8790E-02/
    CHEB/  6.2197E-02 -1.1790E-01 -1.3324E-01  1.5860E-02 -8.3348E-03/
    CHEB/ -8.1496E-02 -1.9071E-02/
C3C. + CH3O (+M)    <=> C3CO. + CH3 (+M)      1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        6.4939E+00 -1.7184E+00 -6.8081E-02  5.0947E+00/
    CHEB/  1.2727E+00 -6.8076E-02  3.0436E-01  6.3616E-01  5.0957E-02/
    CHEB/ -3.7812E-01  8.5940E-02  9.4162E-02 -2.2458E-01 -1.5143E-01/
    CHEB/  4.6869E-02 -5.9132E-02 -1.3961E-01 -1.5761E-02  1.2551E-02/
    CHEB/ -4.8169E-02 -3.7206E-02/
C3C. + CH3O (+M)    <=> C2C.OC + CH3 (+M)     1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        7.3459E+00 -1.7128E+00 -6.8798E-02  5.2696E+00/
    CHEB/  1.2780E+00 -6.7712E-02  3.0351E-01  6.3175E-01  5.3147E-02/
    CHEB/ -3.7233E-01  7.8775E-02  9.5034E-02 -2.1989E-01 -1.5457E-01/
    CHEB/  4.5413E-02 -5.6800E-02 -1.3842E-01 -1.7599E-02  1.2962E-02/
    CHEB/ -4.5829E-02 -3.7728E-02/
C3C. + CH3O (+M)    <=> C3.COC + H (+M)       1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.2759E+00 -1.5934E+00 -9.1806E-02  8.7917E+00/
    CHEB/  1.3577E+00 -5.4265E-02  1.8368E-01  5.1207E-01  1.0377E-01/
```

```
     CHEB/ -2.5273E-01 -4.9276E-02  9.9577E-02 -1.4479E-01 -1.7943E-01/
     CHEB/  6.0428E-03 -2.9015E-02 -9.7163E-02 -4.6495E-02  1.4702E-02/
     CHEB/ -6.1343E-03 -3.4947E-02/
C3C. + CH3O (+M)      <=> C3COC. + H (+M)       1.00E+00     .000        0. ! 0        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       1.8420E+00 -1.6478E+00 -7.9528E-02  7.2695E+00/
     CHEB/  1.3293E+00 -6.2204E-02  2.4576E-01  5.7127E-01  8.0170E-02/
     CHEB/ -3.0317E-01  2.0511E-03  1.0106E-01 -1.7282E-01 -1.7700E-01/
     CHEB/  2.5238E-02 -3.7867E-02 -1.1785E-01 -3.6253E-02  1.4987E-02/
     CHEB/ -2.1126E-02 -3.9070E-02/
C2COC. (+M)           <=> CC.C + CH2O (+M)      1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       5.3069E+00  1.4971E+00 -2.0475E-01  3.3677E+00/
     CHEB/  1.1138E+00  1.6101E-01 -7.7226E-01  5.7192E-02  9.3844E-02/
     CHEB/ -6.3335E-02 -1.4002E-01 -2.1929E-02  8.6012E-02 -5.6832E-02/
     CHEB/ -3.1740E-02  4.1528E-02  9.0669E-03 -6.0416E-03 -1.0724E-02/
     CHEB/  1.7593E-02  5.9315E-03/
C2COC. (+M)           <=> C2C.OC (+M)           1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       2.5298E-01  3.6211E+00 -1.0178E-01  1.8813E+00/
     CHEB/  1.2995E+00  9.9399E-02 -1.0650E+00  2.5313E-01  3.1540E-02/
     CHEB/ -9.7810E-02 -6.2778E-02 -2.6742E-02  8.8201E-02 -7.6076E-02/
     CHEB/ -2.0871E-02  3.5573E-02 -3.6810E-02  4.2775E-03 -1.8468E-02/
     CHEB/ -1.2472E-02  1.2429E-02/
C2COC. (+M)           <=> C2C*O + CH3 (+M)      1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       3.6423E+00  1.1956E+00 -1.0773E-01  2.9636E+00/
     CHEB/  1.3190E+00  1.0407E-01 -8.9537E-01  1.9075E-01  3.2473E-02/
     CHEB/ -1.0707E-01 -1.1169E-01 -2.7155E-02  9.8550E-02 -8.9080E-02/
     CHEB/ -1.7438E-02  6.7326E-02 -2.8232E-02  7.3237E-03  3.7905E-03/
     CHEB/  1.7423E-03  1.1465E-02/
C2COC. (+M)           <=> C*C(C)OC + H (+M)     1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -2.4080E+00  1.5974E+00 -1.6070E-01  5.3472E+00/
     CHEB/  1.1671E+00  1.6228E-01  7.4148E-02 -1.4866E-01  4.4650E-02/
     CHEB/  2.0751E-01 -1.6052E-01 -4.1812E-02  1.2170E-01  5.3527E-03/
     CHEB/ -1.8870E-02  1.1962E-02  3.9448E-02  7.8711E-03 -3.1942E-02/
     CHEB/  1.0701E-02  8.1009E-03/
C2COC. (+M)           <=> C2.COC (+M)           1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       8.2026E-01  2.4745E+00 -3.5980E-01  3.3340E+00/
     CHEB/  1.8422E+00  1.4714E-01 -1.3411E+00  4.8964E-01  7.8582E-02/
     CHEB/ -3.5961E-01  4.2635E-02  2.0192E-02 -5.5213E-02 -1.8339E-02/
     CHEB/  1.0604E-02 -2.7389E-02  5.1004E-04  8.6431E-03 -4.6910E-02/
     CHEB/  6.6947E-03  3.8870E-03/
C2COC. (+M)           <=> C*COC + CH3 (+M)      1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -3.5788E+00  1.3374E+00 -3.9944E-01  9.5994E+00/
     CHEB/  1.1708E+00  3.1544E-01 -6.0803E-01  8.8833E-02  9.4696E-02/
     CHEB/ -1.5892E-01 -9.4768E-02 -4.7299E-03 -3.5494E-02 -4.3228E-02/
     CHEB/ -1.9630E-02 -2.5062E-02  5.4627E-03 -7.3566E-03 -2.8378E-02/
     CHEB/  1.5223E-02  3.1726E-03/
C2COC. (+M)           <=> C*CC + CH3O (+M)      1.00E+00     .000        0. ! 4        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -4.9675E+00  1.3352E+00 -3.9999E-01  9.5763E+00/
     CHEB/  1.1712E+00  3.1538E-01 -6.1261E-01  9.0998E-02  9.5034E-02/
     CHEB/ -1.5845E-01 -9.4922E-02 -4.5710E-03 -3.5584E-02 -4.3657E-02/
     CHEB/ -1.9556E-02 -2.5255E-02  5.5015E-03 -7.3661E-03 -2.8422E-02/
     CHEB/  1.5314E-02  3.1229E-03/
C2C.OC (+M)           <=> C2C*O + CH3 (+M)      1.00E+00     .000        0. ! 6        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3       6.6745E+00  1.7933E+00 -2.3385E-01  1.7989E+00/
     CHEB/  8.4559E-01  2.4155E-01 -5.1509E-01 -3.5117E-02  4.2013E-02/
     CHEB/  4.2810E-02 -1.1023E-01 -4.1459E-02  8.0660E-02 -1.9448E-02/
     CHEB/ -1.9458E-02  8.5751E-03  1.7378E-02  4.3481E-03 -2.6102E-02/
     CHEB/  1.0530E-02  6.3998E-03/
C2C.OC (+M)           <=> C*C(C)OC + H (+M)     1.00E+00     .000        0. ! 6        11/95
     LOW / 1.0 0.0 0.0 /
     CHEB/  7   3      -5.8902E+00  2.4528E+00 -3.4062E-02  1.1266E+01/
     CHEB/  6.5329E-02  4.6767E-02 -1.4221E-01 -1.8744E-02 -1.2917E-02/
     CHEB/  9.6299E-02 -2.6804E-03 -2.2771E-03  3.5243E-02  2.1708E-03/
     CHEB/  1.5797E-03 -2.8431E-02  7.5533E-04  6.0231E-04 -3.6203E-02/
     CHEB/ -2.1572E-04 -1.5418E-04/
```

```
C2C.OC (+M)          <=> CC.C + CH2O (+M)      1.00E+00     .000        0. ! 6         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -8.9906E+00  1.6998E+00 -2.2864E-01  1.3121E+01/
    CHEB/  1.0095E+00  2.3870E-01 -2.8910E-01 -1.0803E-01  4.5748E-02/
    CHEB/  2.6940E-02 -1.2561E-01 -5.0046E-02  3.9576E-02 -6.7797E-03/
    CHEB/ -2.1655E-02 -1.3200E-02  2.5360E-02  7.9157E-03 -3.1057E-02/
    CHEB/  1.1049E-02  8.9785E-03/
C2C.OC (+M)          <=> C2.COC (+M)           1.00E+00     .000        0. ! 6         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.0900E+01  3.6517E+00 -2.3347E-01  1.2421E+01/
    CHEB/  1.2295E+00  8.4460E-02 -8.9992E-01  2.7580E-01  1.3447E-01/
    CHEB/ -1.2210E-01 -6.9818E-02  4.6622E-02  3.0903E-02 -9.7242E-02/
    CHEB/ -1.9744E-02  7.2233E-03 -3.6372E-02 -2.6741E-02 -2.4258E-02/
    CHEB/  7.6941E-03 -8.7338E-03/
C2C.OC (+M)          <=> C*COC + CH3 (+M)      1.00E+00     .000        0. ! 6         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.0665E+01  1.3251E+00 -2.3196E-01  1.5220E+01/
    CHEB/  1.2473E+00  1.3918E-01 -1.0869E-01  1.2377E-01  1.3787E-01/
    CHEB/ -1.7196E-02 -1.4455E-01  3.7351E-03 -1.3912E-02 -8.4940E-02/
    CHEB/ -4.5866E-02 -2.4905E-02 -4.9105E-03 -2.2798E-02 -2.0771E-02/
    CHEB/  2.2319E-02  4.2193E-03/
C2C.OC (+M)          <=> C*CC + CH3O (+M)      1.00E+00     .000        0. ! 6         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.2067E+01  1.3237E+00 -2.3172E-01  1.5208E+01/
    CHEB/  1.2480E+00  1.3857E-01 -1.1139E-01  1.2451E-01  1.3807E-01/
    CHEB/ -1.7873E-02 -1.4429E-01  3.9787E-03 -1.3948E-02 -8.5198E-02/
    CHEB/ -4.5839E-02 -2.4980E-02 -5.1665E-03 -2.2874E-02 -2.0821E-02/
    CHEB/  2.2329E-02  4.1543E-03/
C2.COC (+M)          <=> C*CC + CH3O (+M)      1.00E+00     .000        0. ! 5         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.0201E+00  1.4891E+00 -2.9869E-01  8.2208E+00/
    CHEB/  9.5116E-01  1.7582E-01 -5.7866E-01  1.4707E-01  1.2348E-01/
    CHEB/ -1.5002E-01 -7.3512E-02  1.2014E-02 -3.7954E-02 -5.1946E-02/
    CHEB/ -2.3542E-02 -2.4781E-02 -1.7982E-04 -1.1511E-02 -2.6391E-02/
    CHEB/  1.6192E-02  3.2749E-03/
C2.COC (+M)          <=> C*C(C)OC + H (+M)     1.00E+00     .000        0. ! 5         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        1.6820E-01  1.3821E+00 -2.8240E-01  7.3183E+00/
    CHEB/  1.0319E+00  1.3430E-01 -5.8261E-01  1.7461E-01  1.3469E-01/
    CHEB/ -1.6075E-01 -7.4167E-02  2.4160E-02 -4.3695E-02 -5.6677E-02/
    CHEB/ -2.1442E-02 -2.7589E-02 -1.8157E-03 -1.3722E-02 -2.7700E-02/
    CHEB/  1.6880E-02  1.7293E-03/
C2.COC (+M)          <=> C*COC + CH3 (+M)      1.00E+00     .000        0. ! 5         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        3.6777E-01  1.4908E+00 -2.9847E-01  8.2444E+00/
    CHEB/  9.5137E-01  1.7617E-01 -5.7351E-01  1.4524E-01  1.2340E-01/
    CHEB/ -1.5074E-01 -7.3809E-02  1.1626E-02 -3.7972E-02 -5.1539E-02/
    CHEB/ -2.3668E-02 -2.4520E-02 -6.3706E-05 -1.1424E-02 -2.6320E-02/
    CHEB/  1.6113E-02  3.3433E-03/
C2.COC (+M)          <=> CC.C + CH2O (+M)      1.00E+00     .000        0. ! 5         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3        3.9011E+00  3.4704E-01 -3.5131E-01  4.3494E+00/
    CHEB/  1.9699E+00  1.7697E-01 -7.9715E-01  3.1081E-01  9.8857E-02/
    CHEB/ -2.7799E-01 -1.0968E-01  3.4223E-02 -4.4858E-02 -8.5975E-02/
    CHEB/  1.0735E-02 -9.6710E-03 -1.3118E-02 -1.4826E-03 -2.4903E-02/
    CHEB/  1.5151E-02 -4.6144E-03/
C2.COC (+M)          <=> C2C*O + CH3 (+M)      1.00E+00     .000        0. ! 5         11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -4.0600E+00  1.6800E+00 -3.2039E-01  1.0687E+01/
    CHEB/  8.3116E-01  2.6023E-01 -5.3066E-01  7.1927E-02  8.4760E-02/
    CHEB/ -1.2511E-01 -7.5462E-02 -1.2841E-02 -1.8957E-02 -3.9976E-02/
    CHEB/ -2.4678E-02 -1.5422E-02  4.5117E-03 -5.6219E-03 -2.4210E-02/
    CHEB/  1.4756E-02  6.1401E-03/
C3COC.               <=> C3.COC                2.84E+08   0.0    10800.
C3.COC               <=> C2C*C + CH3O          1.09E+12   0.0    17900.
C3.COC               <=> C*C(C)OC + CH3        8.66E+10   0.0    17900.
C3COC (+M)           <=> C3C. + CH3O (+M)      1.00E+00     .000        0. ! ing503    11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3       -1.5174E+01  5.2880E-01 -6.8196E-02  2.4058E+01/
    CHEB/  9.3515E-01 -1.0267E-01 -8.6242E-01  6.3401E-01 -2.7739E-02/
    CHEB/ -5.3347E-01  2.9944E-01  3.6503E-02 -2.6727E-01  5.5650E-02/
    CHEB/  5.7352E-02 -9.0570E-02 -5.5571E-02  3.9732E-02 -9.1985E-03/
    CHEB/ -6.5933E-02  1.0150E-02/
```

```
C3COC (+M)          <=> C2C*C + CH3OH (+M)   1.00E+00     .000       0. ! ing503   11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.0505E+01  3.5936E-01 -4.9706E-02  1.8346E+01/
    CHEB/  6.6158E-01 -8.4821E-02 -6.2305E-01  5.1146E-01 -4.8453E-02/
    CHEB/ -4.4296E-01  3.2107E-01 -7.7699E-03 -2.7830E-01  1.4827E-01/
    CHEB/  2.0405E-02 -1.4011E-01  3.0971E-02  2.8790E-02 -5.2949E-02/
    CHEB/ -2.3192E-02  2.1295E-02/
C3COC (+M)          <=> C2C.OC + CH3 (+M)    1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -1.6586E+01  5.4074E-01 -6.4946E-02  2.4950E+01/
    CHEB/  9.5329E-01 -9.6115E-02 -9.1300E-01  6.3997E-01 -2.2248E-02/
    CHEB/ -5.6121E-01  2.9560E-01  3.8974E-02 -2.7743E-01  4.9064E-02/
    CHEB/  5.6143E-02 -9.1318E-02 -5.9425E-02  3.6400E-02 -6.9188E-03/
    CHEB/ -6.5926E-02  7.1576E-03/
C3COC (+M)          <=> C3.COC + H (+M)      1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.4625E+01  7.2899E-01 -8.9933E-02  2.9891E+01/
    CHEB/  1.1807E+00 -9.0678E-02 -1.0774E+00  6.1896E-01  4.3523E-02/
    CHEB/ -5.3839E-01  1.4427E-01  9.3810E-02 -1.9529E-01 -7.9777E-02/
    CHEB/  5.7126E-02 -2.2406E-02 -1.0627E-01  2.1902E-03  2.7899E-02/
    CHEB/ -5.4542E-02 -2.3858E-02/
C3COC (+M)          <=> C3COC. + H (+M)      1.00E+00     .000       0. ! 0        11/95
    LOW / 1.0 0.0 0.0 /
    CHEB/  7   3      -2.3371E+01  6.4852E-01 -7.7143E-02  2.7920E+01/
    CHEB/  1.0974E+00 -9.6404E-02 -1.0217E+00  6.4869E-01  1.0808E-02/
    CHEB/ -5.6215E-01  2.1716E-01  7.3552E-02 -2.3168E-01 -3.2134E-02/
    CHEB/  6.4607E-02 -4.7682E-02 -9.7933E-02  2.1374E-02  1.7004E-02/
    CHEB/ -6.6067E-02 -1.1639E-02/
C3COC               <=> C3CO. + CH3          3.96E+16   0.0    83000.
C3COC               <=> C2C.OC + CH3         2.26E+17   0.0    83700.
C3COC               <=> C3.COC + H           1.72E+16   0.0    99700.
C3COC               <=> C3COC. + H           6.13E+14   0.0    93400.
C3COC + O           <=> C3.COC + OH          4.53E+13   0.0     5800.
C3COC + O           <=> C3COC. + OH          1.86E+13   0.0     3306.
C3COC + H           <=> C3.COC + H2          1.89E+13   0.0     8700.
C3COC + H           <=> C3COC. + H2          8.00E+12   0.0     5951.
C3COC + O2          <=> C3.COC + HO2         6.06E+13   0.0    50867.
C3COC + O2          <=> C3COC. + HO2         1.05E+13   0.0    44900.
C3COC + OH          <=> C3.COC + H2O         1.53E+13   0.0     2216.
C3COC + OH          <=> C3COC. + H2O         7.76E+12   0.0      769.
C3COC + HO2         <=> C3.COC + H2O2        2.52E+11   0.0    12000.
C3COC + HO2         <=> C3COC. + H2O2        8.50E+10   0.0    13000.
C3COC + CH3         <=> C3.COC + CH4         1.19E+11   0.0    11600.
C3COC + CH3         <=> C3COC. + CH4         2.46E+12   0.0    12073.
C3COC + CH3O        <=> C3.COC + CH3OH       3.62E+11   0.0     7094.
C3COC + CH3O        <=> C3COC. + CH3OH       1.20E+11   0.0     3500.
C3COC + CH2OH       <=> C3.COC + CH3OH       3.62E+11   0.0     7094.
C3COC + CH2OH       <=> C3COC. + CH3OH       1.20E+11   0.0     3500.
C3COC + HCO         <=> C3.COC + CH2O        7.04E+04   2.72   18233.
C3COC + HCO         <=> C3COC. + CH2O        2.68E+04   2.72   18233.
C3COC + C2H5        <=> C3.COC + C2H6        4.74E+11   0.0    12300.
C3COC + C2H5        <=> C3COC. + C2H6        1.58E+11   0.0    12300.
C3COC + C3C.        <=> C3.COC + C3C         1.50E+11   0.0    12900.
C3COC + C3C.        <=> C3COC. + C3C         5.01E+10   0.0    12900.
C3COC + CC.C        <=> C3.COC + CCC         1.50E+11   0.0    12900.
C3COC + CC.C        <=> C3COC. + CCC         5.01E+10   0.0    12900.
END
```

APPENDIX B

FIGURES

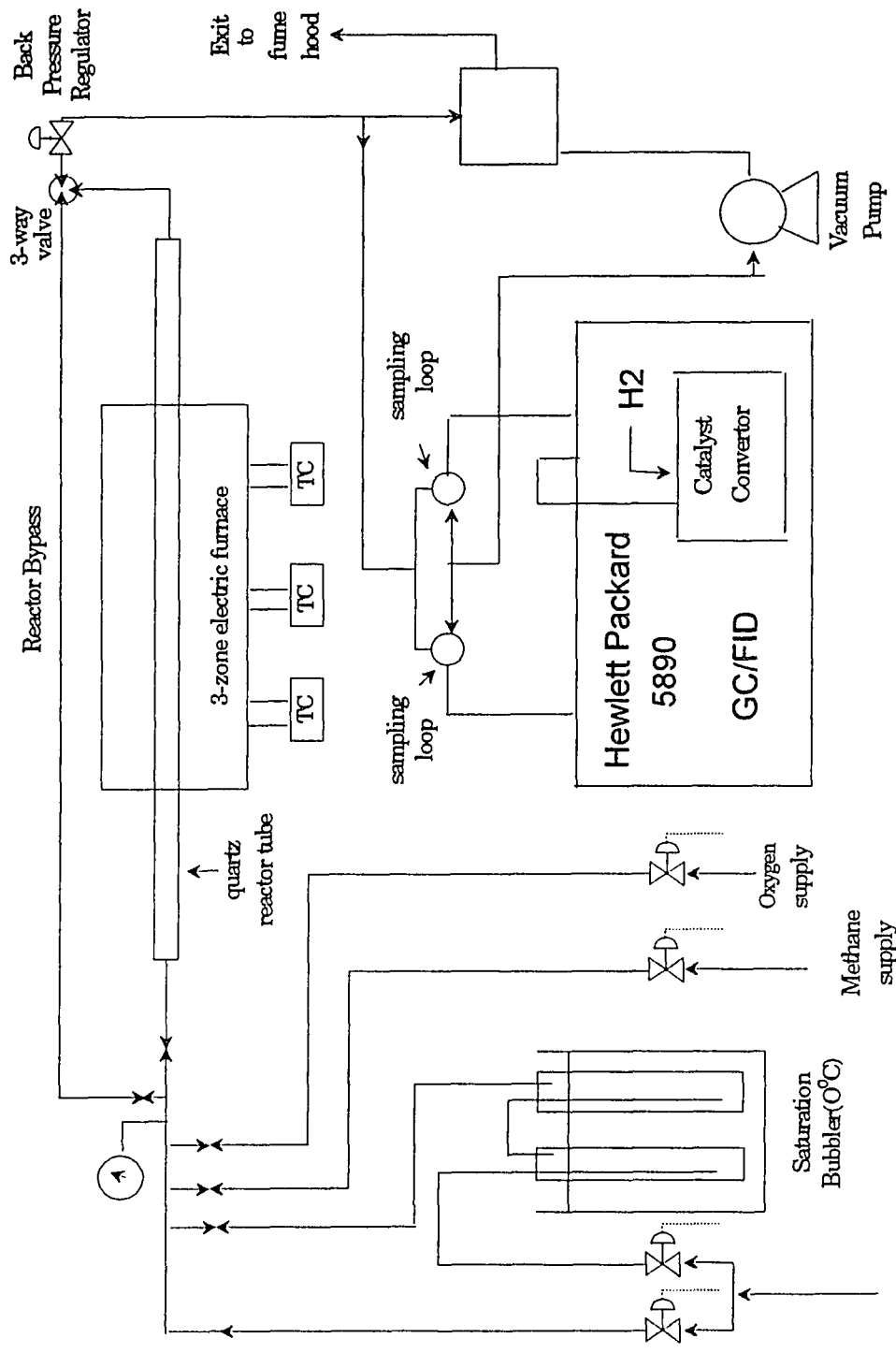


Figure B.1 Experimental Apparatus

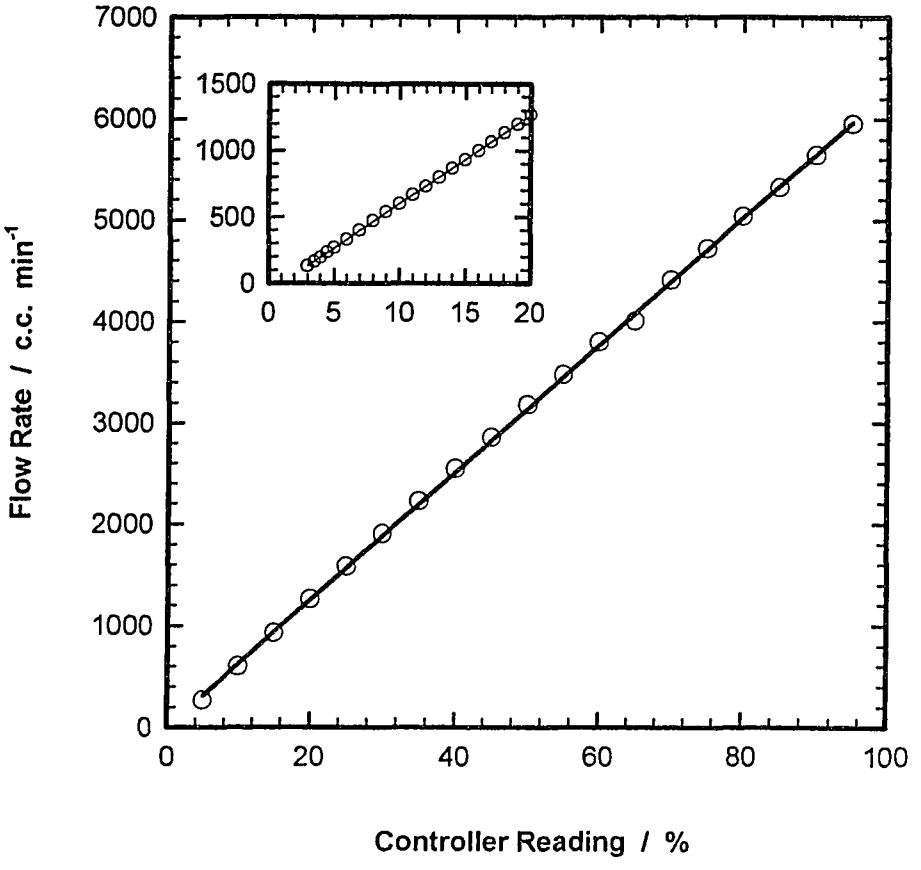


Figure B.2 Calibration curve of mass flow controller 1 for argon with inlet pressure of 200 psig.

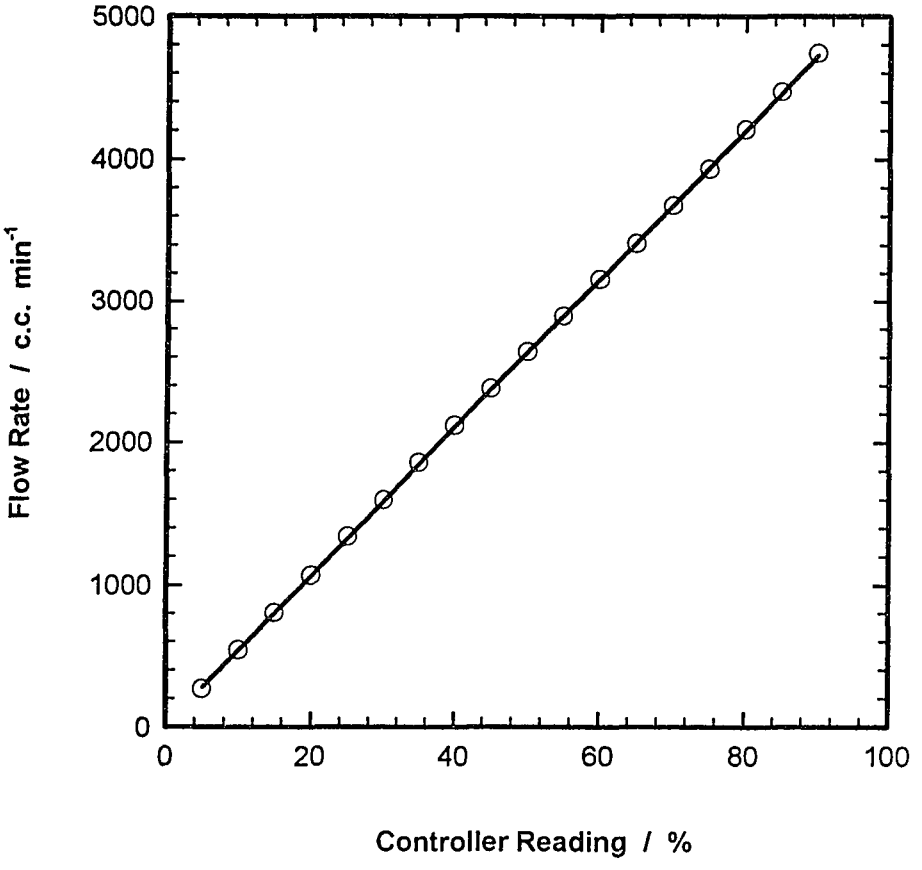


Figure B.3 Calibration curve of mass flow controller 2 for Argon with inlet pressure of 200 psig.

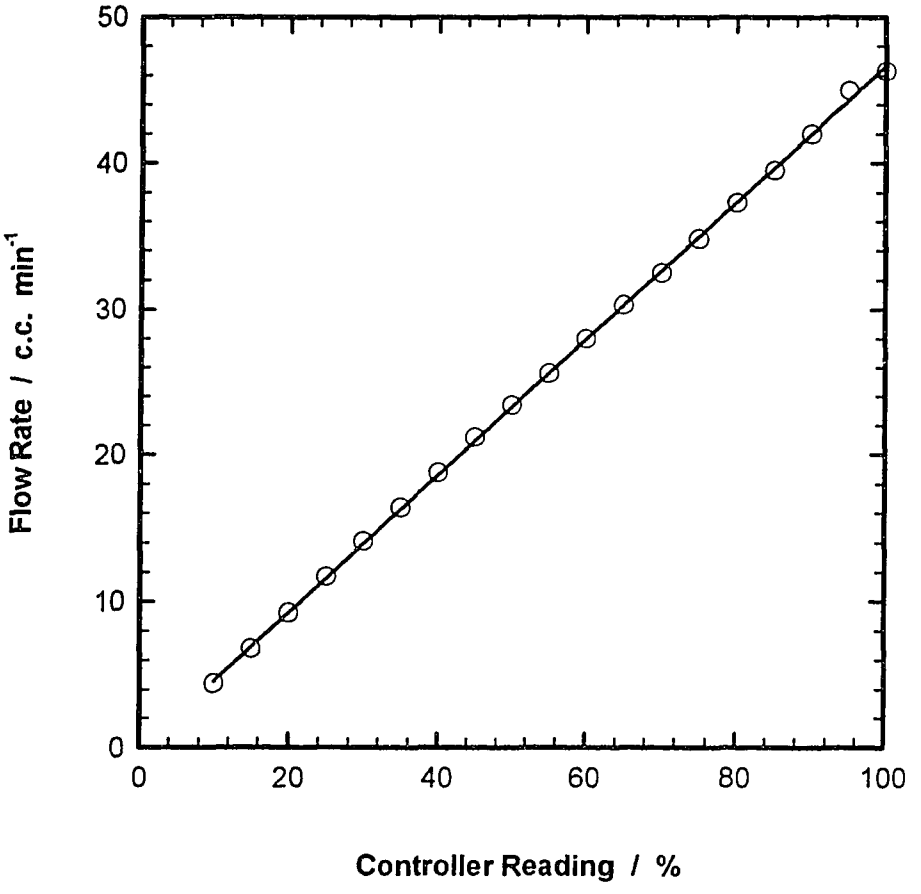


Figure B.4 Calibration curve of mass flow controller 3 for methane with inlet pressure of 200 psig.

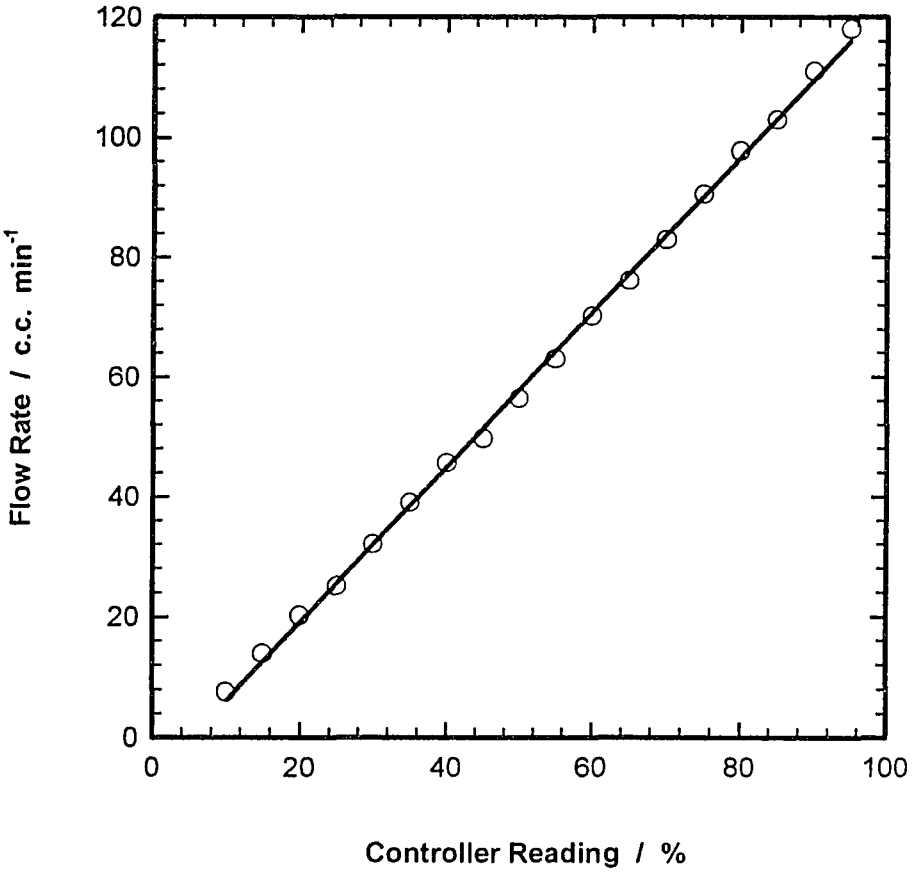


Figure B.5 Calibration curve of mass flow controller 4 for oxygen with inlet pressure of 200 psig.

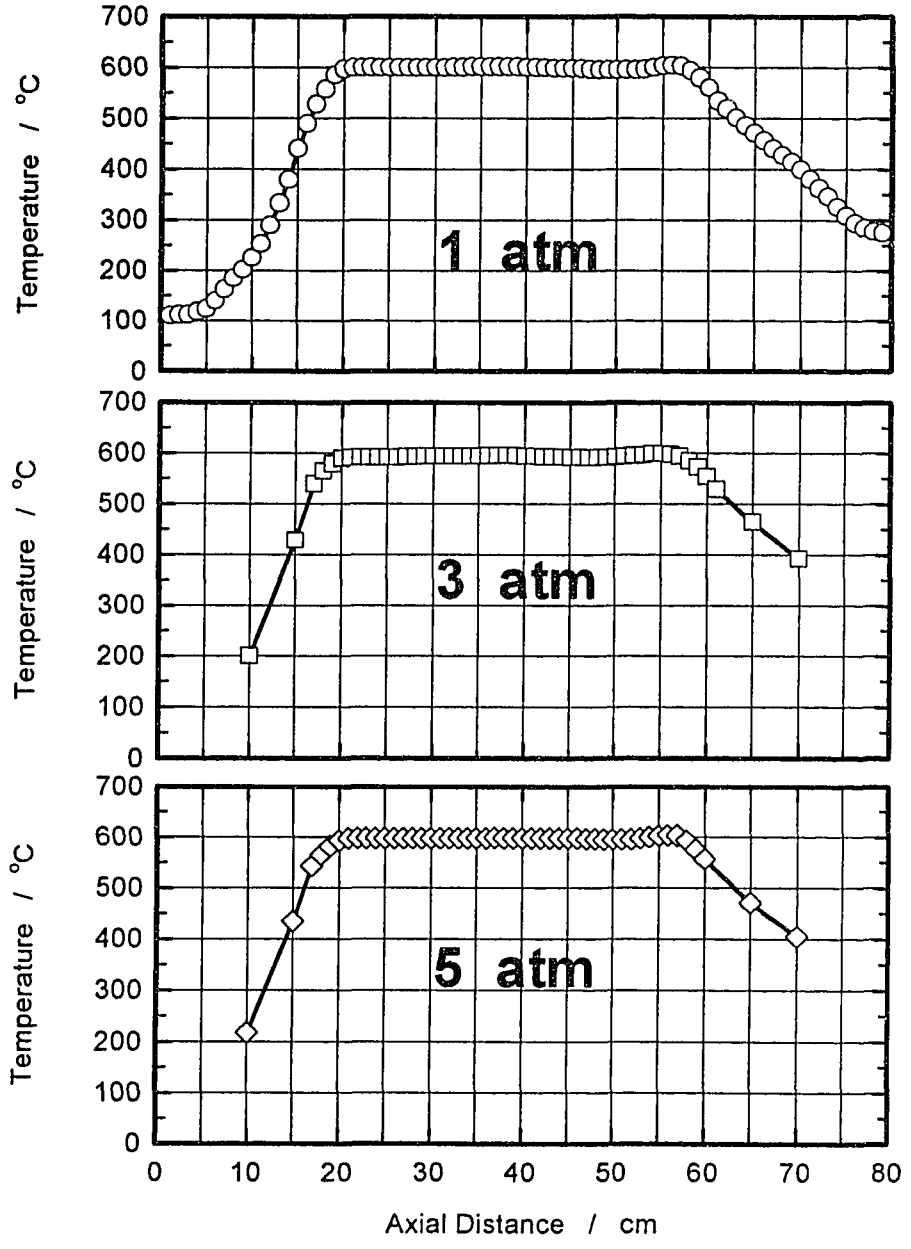


Figure B.6 Reactor Temperature Profiles at 1.3 gm Ar / min

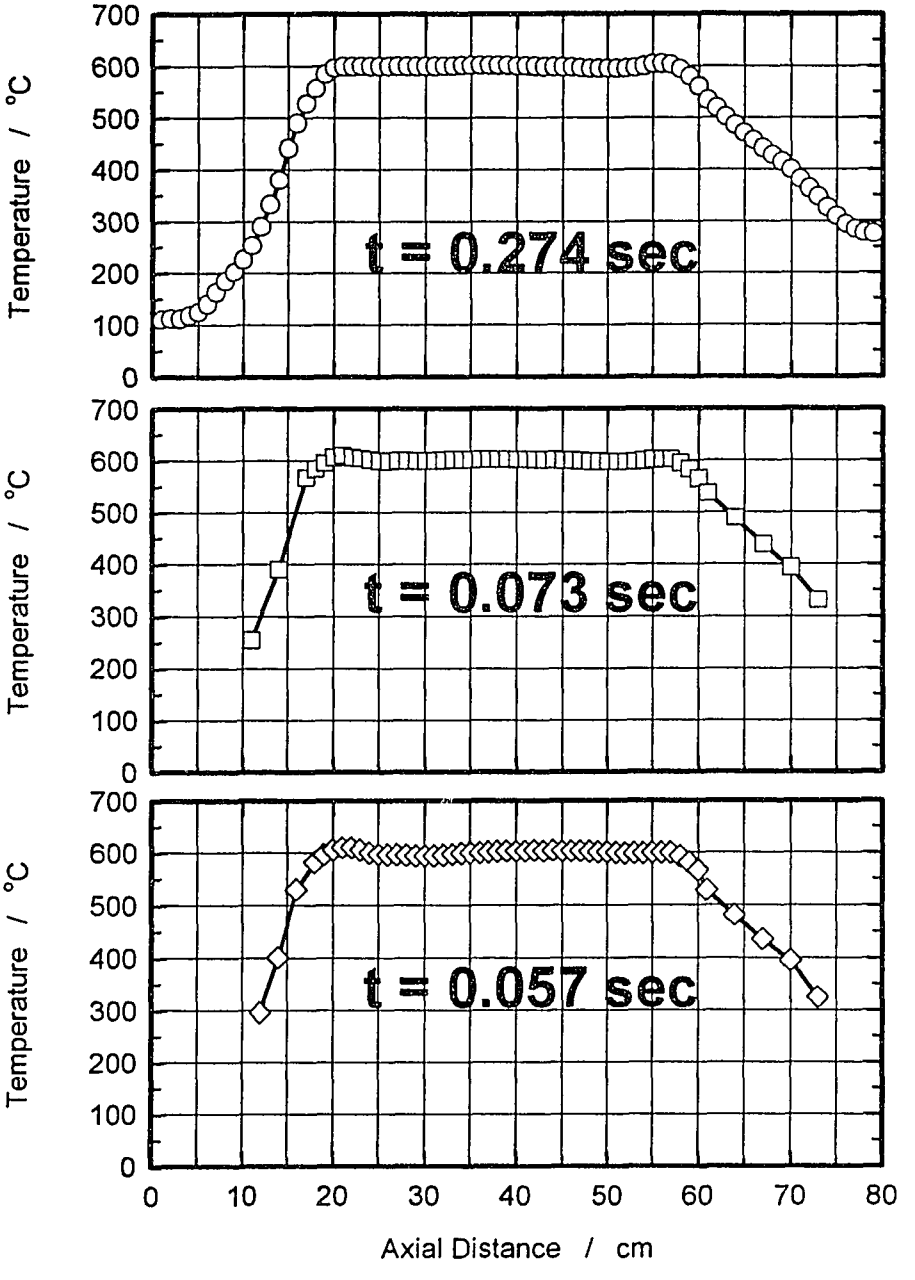


Figure B.7 Reactor Temperature Profiles at 1 atm

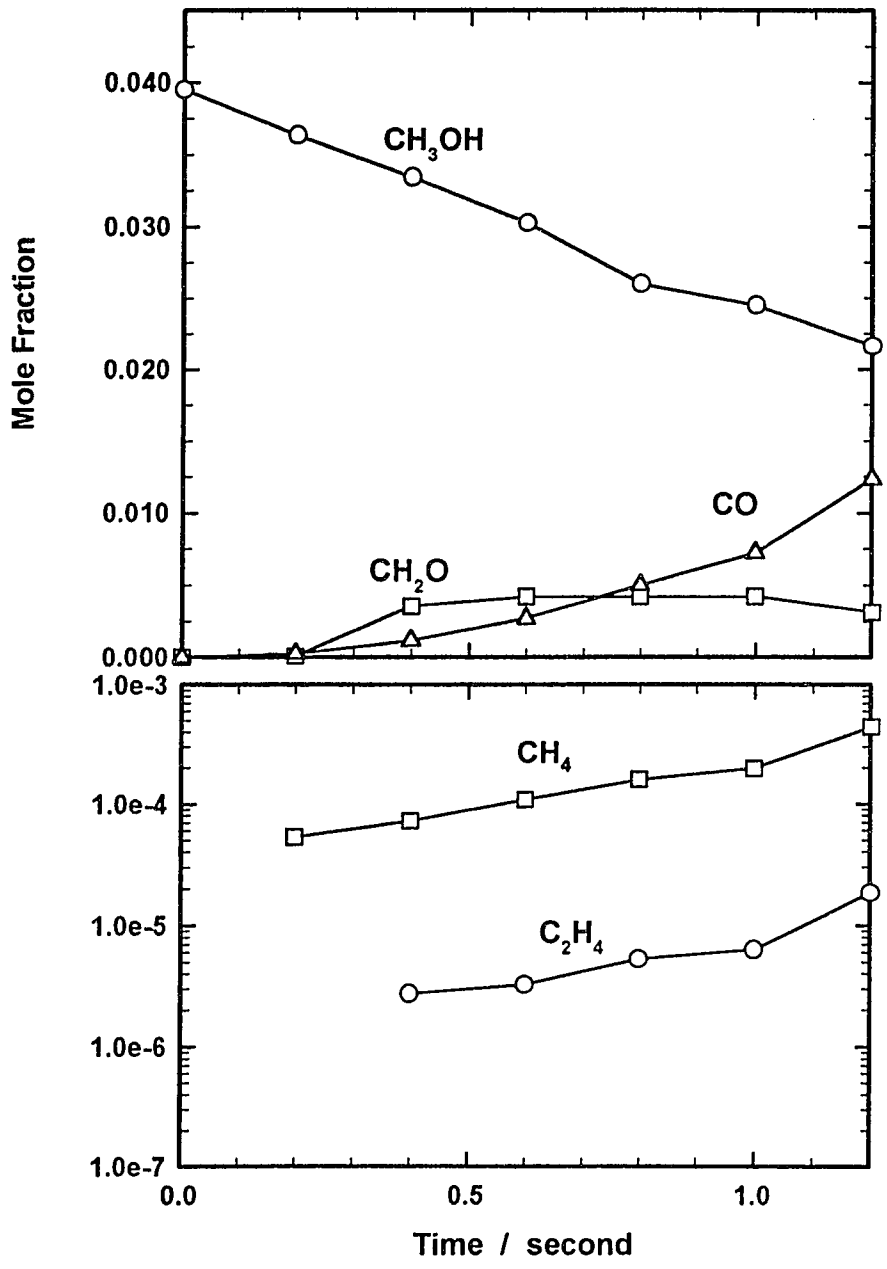


Figure B.8 Experimental result of methanol pyrolysis: T = 1073 K; p = 1 atm; $X_{o,CH3OH}$ = 0.0395

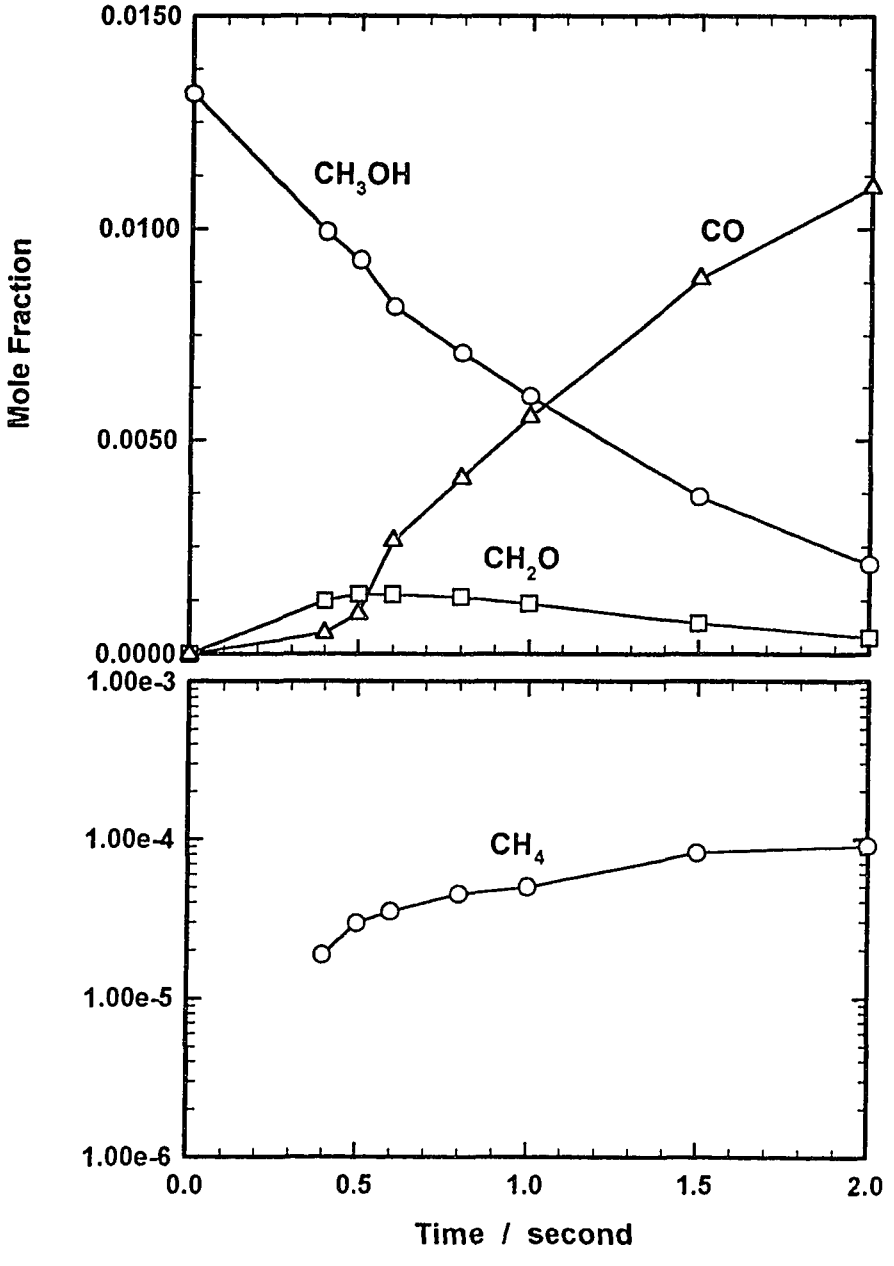


Figure B.9 Experimental result of methanol pyrolysis: $T = 1073$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.01317$

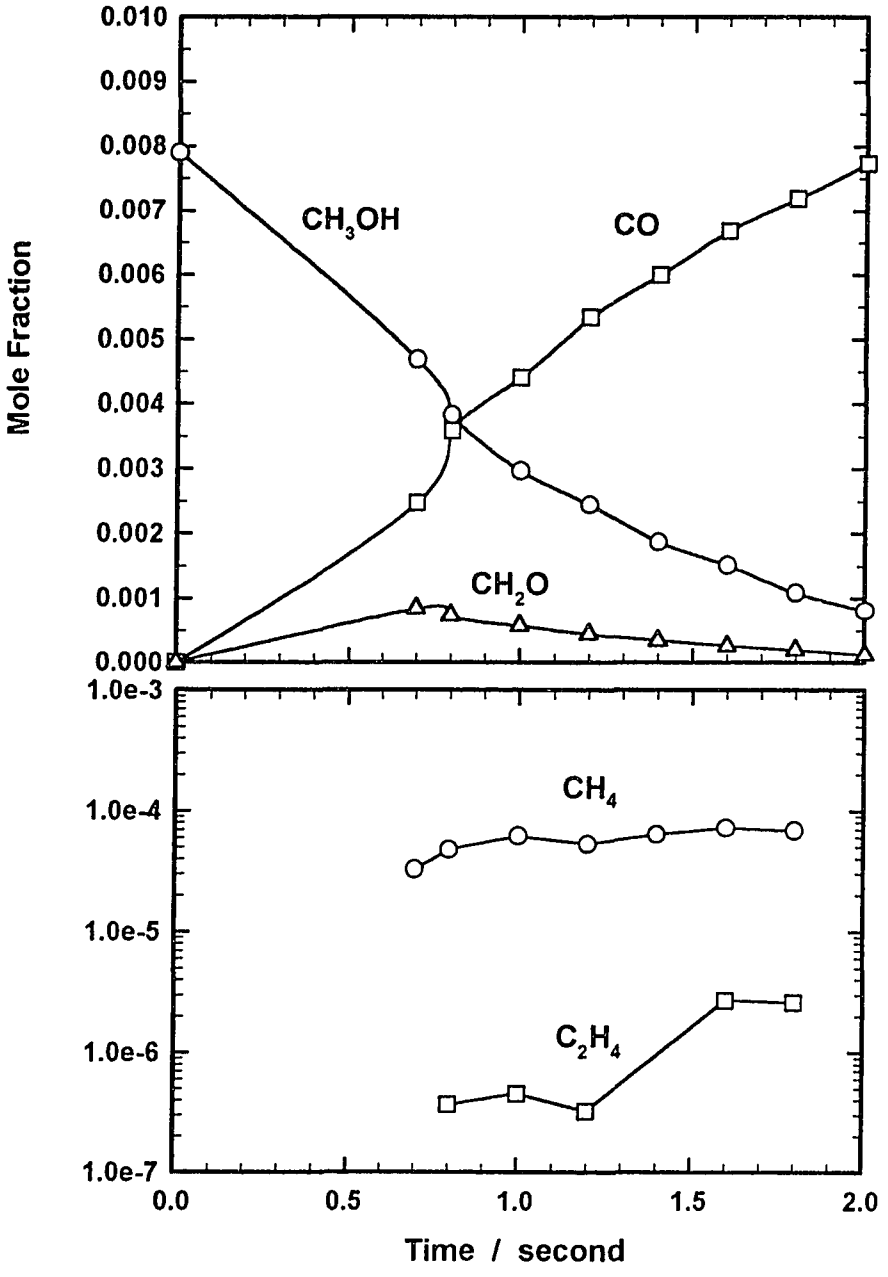


Figure B.10 Experimental result of methanol pyrolysis: $T = 1073$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0079$

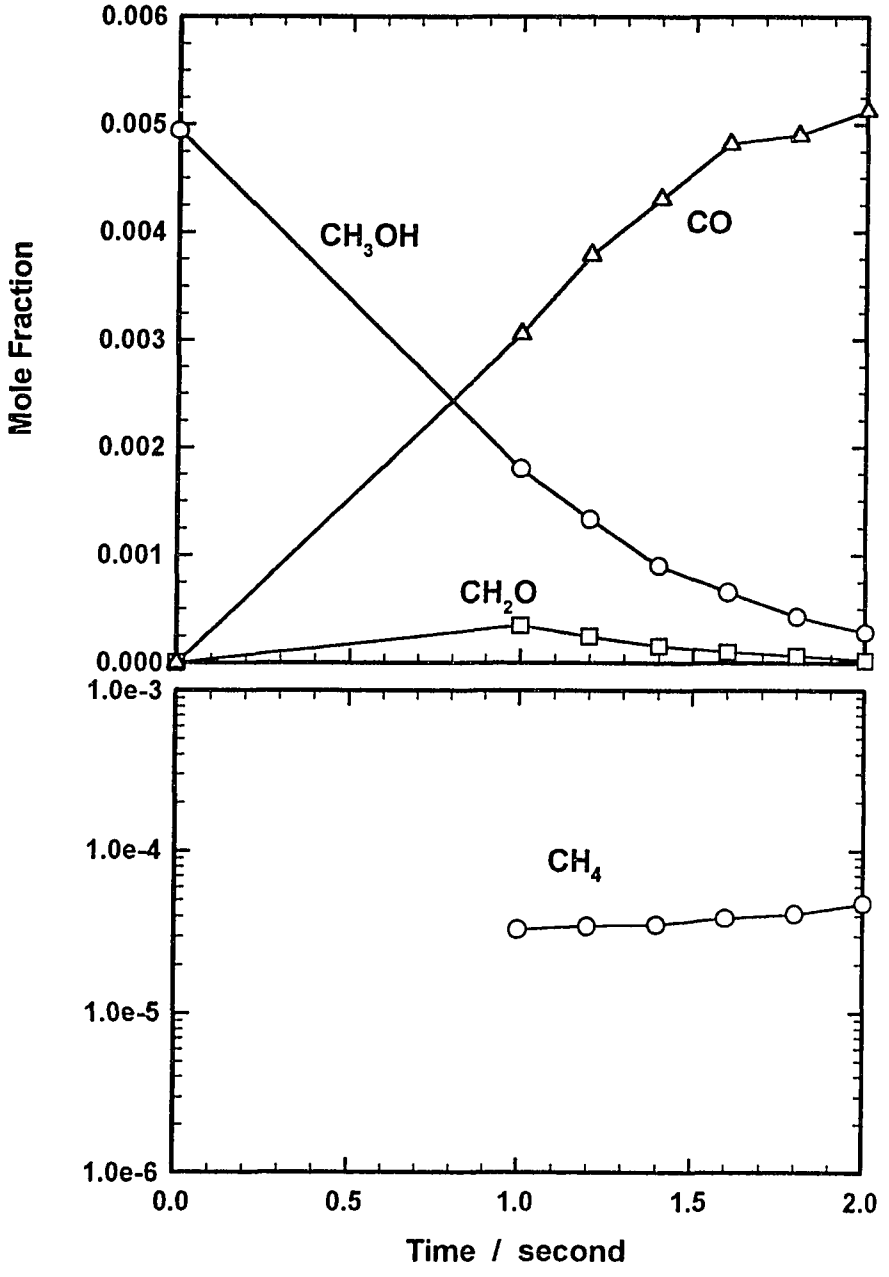


Figure B.11 Experimental result of methanol pyrolysis: $T = 1073$ K; $p = 8$ atm; $X_{o,CH3OH} = 0.00494$

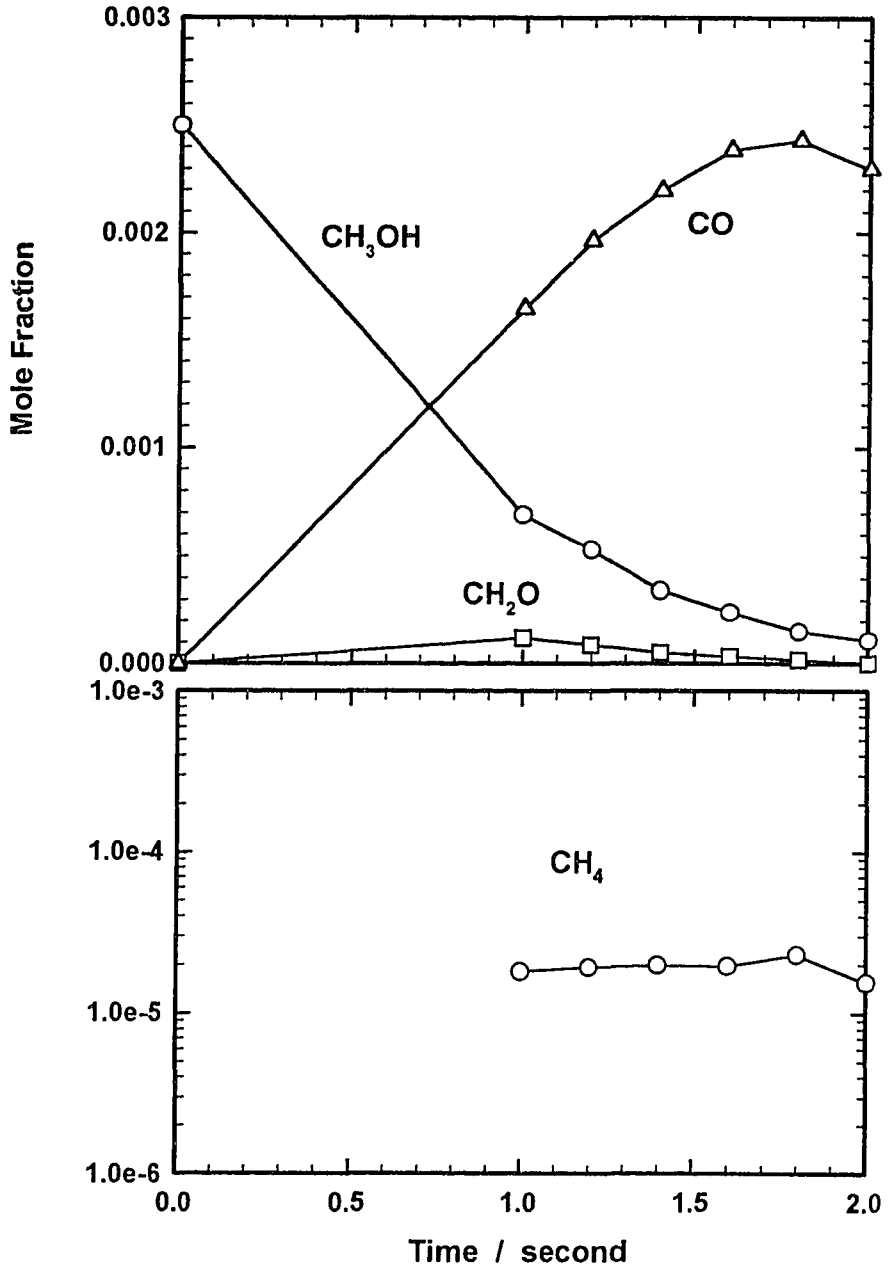


Figure B.12 Experimental result of methanol pyrolysis: $T = 1073$ K; $p = 8$ atm; $X_{o,CH3OH} = 0.0025$

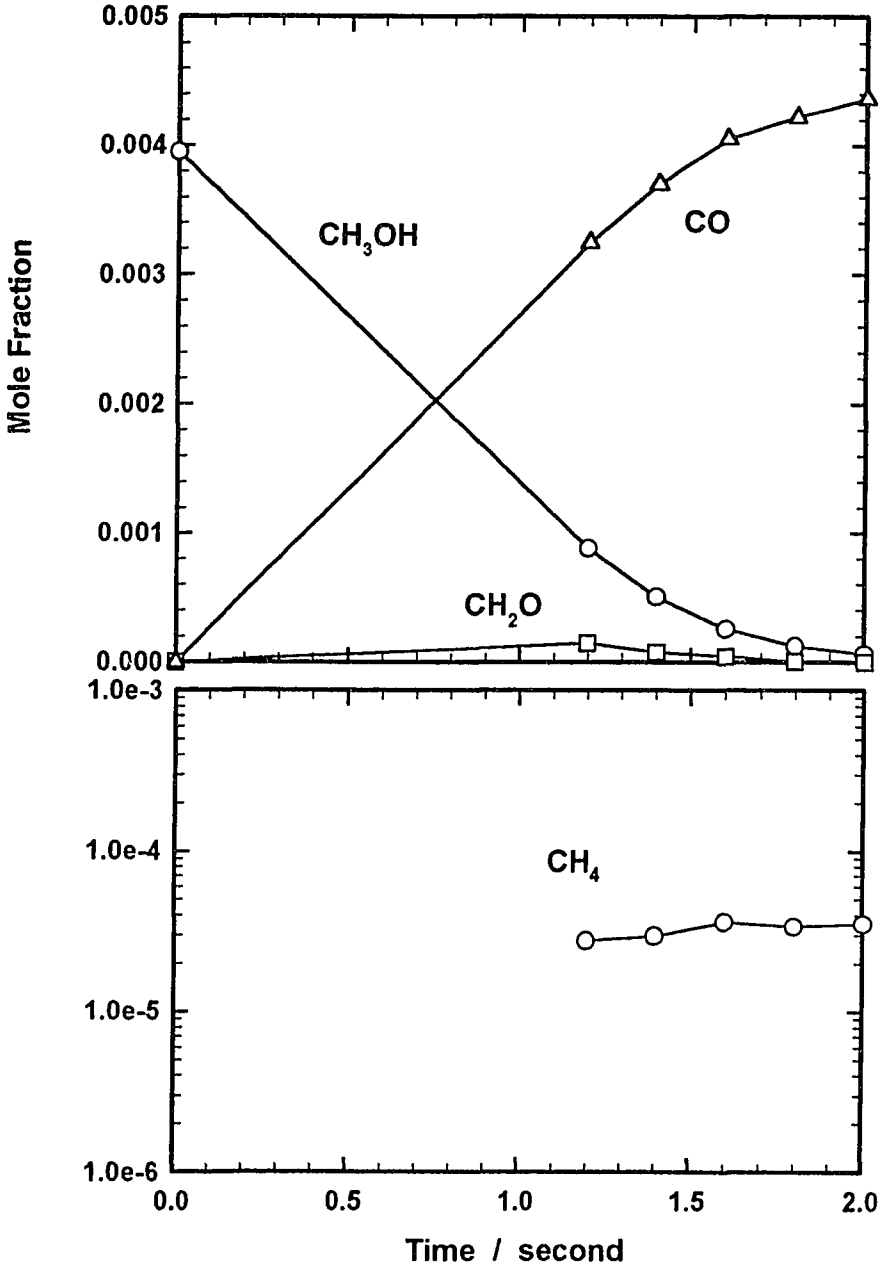


Figure B.13 Experimental result of methanol pyrolysis: T = 1073 K; p = 10 atm; $X_{o,CH3OH}$ = 0.00395

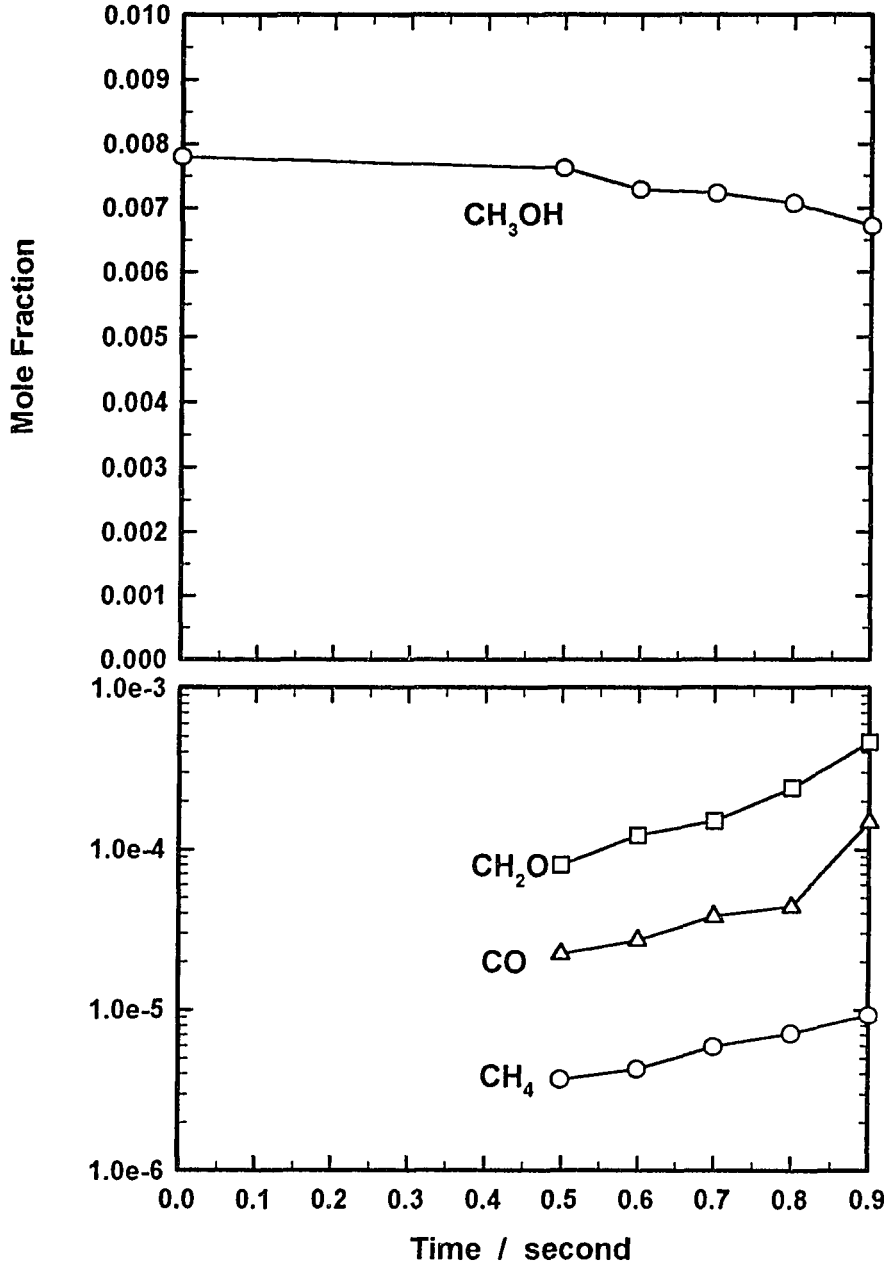


Figure B.14 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

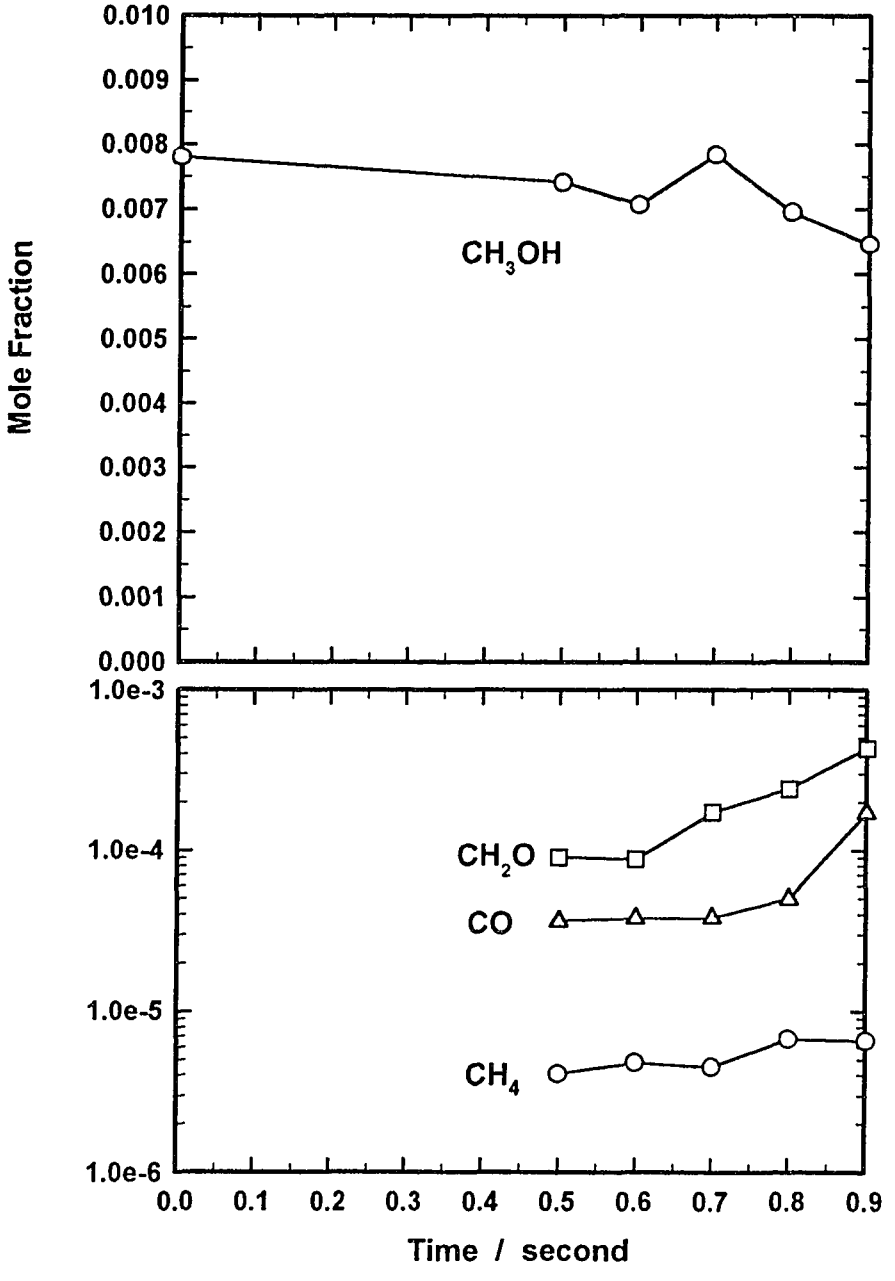


Figure B.15 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 873$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

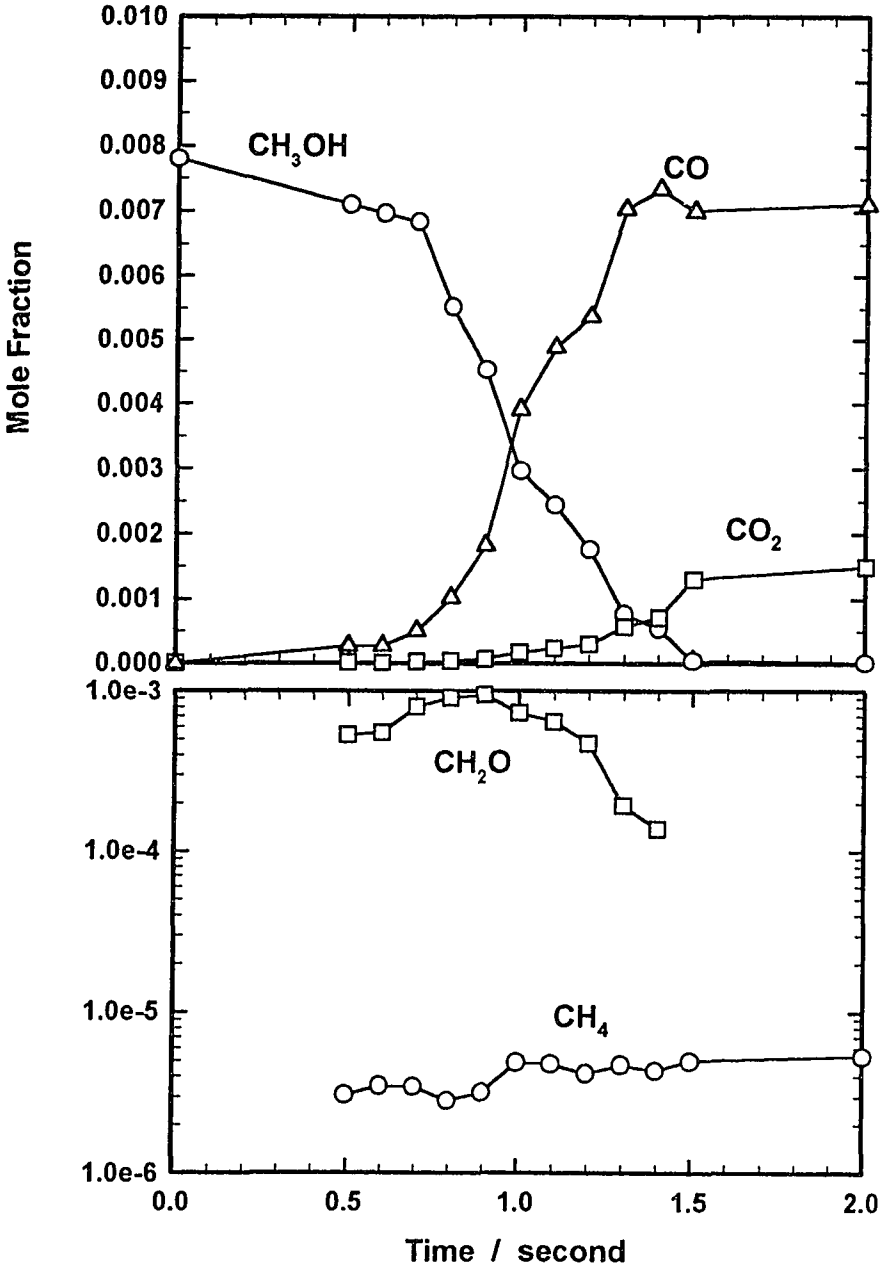


Figure B.16 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

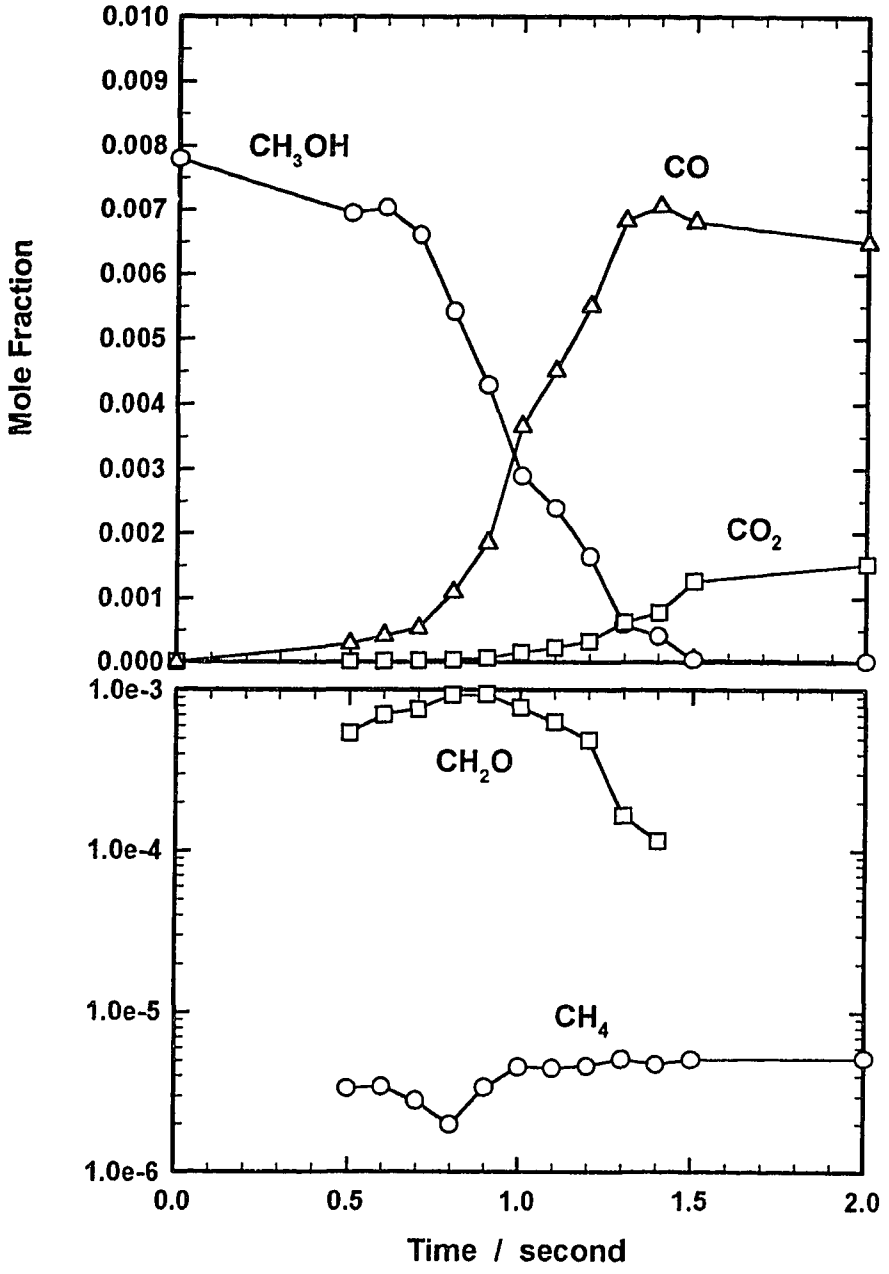


Figure B.17 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 873$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

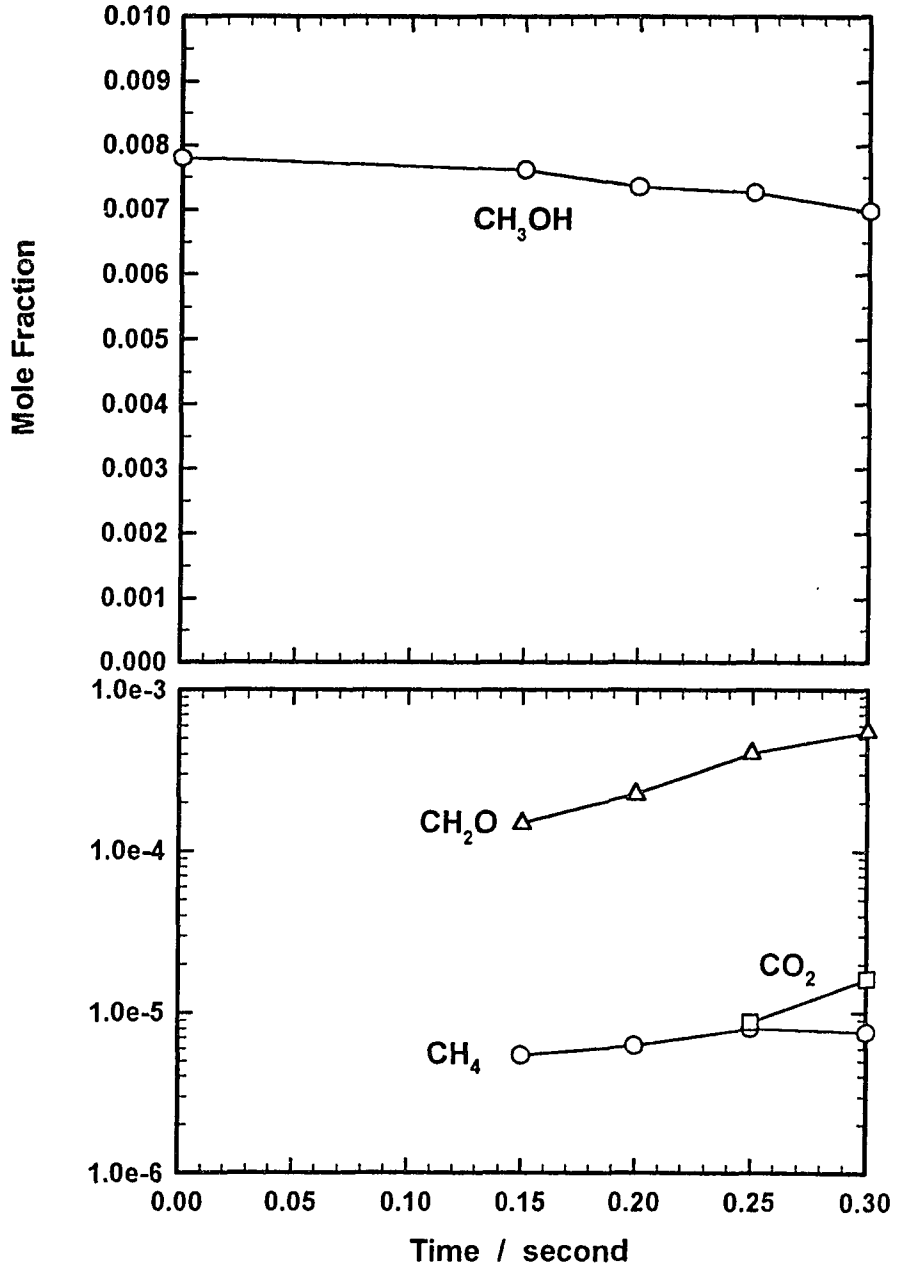


Figure B.18 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 923$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

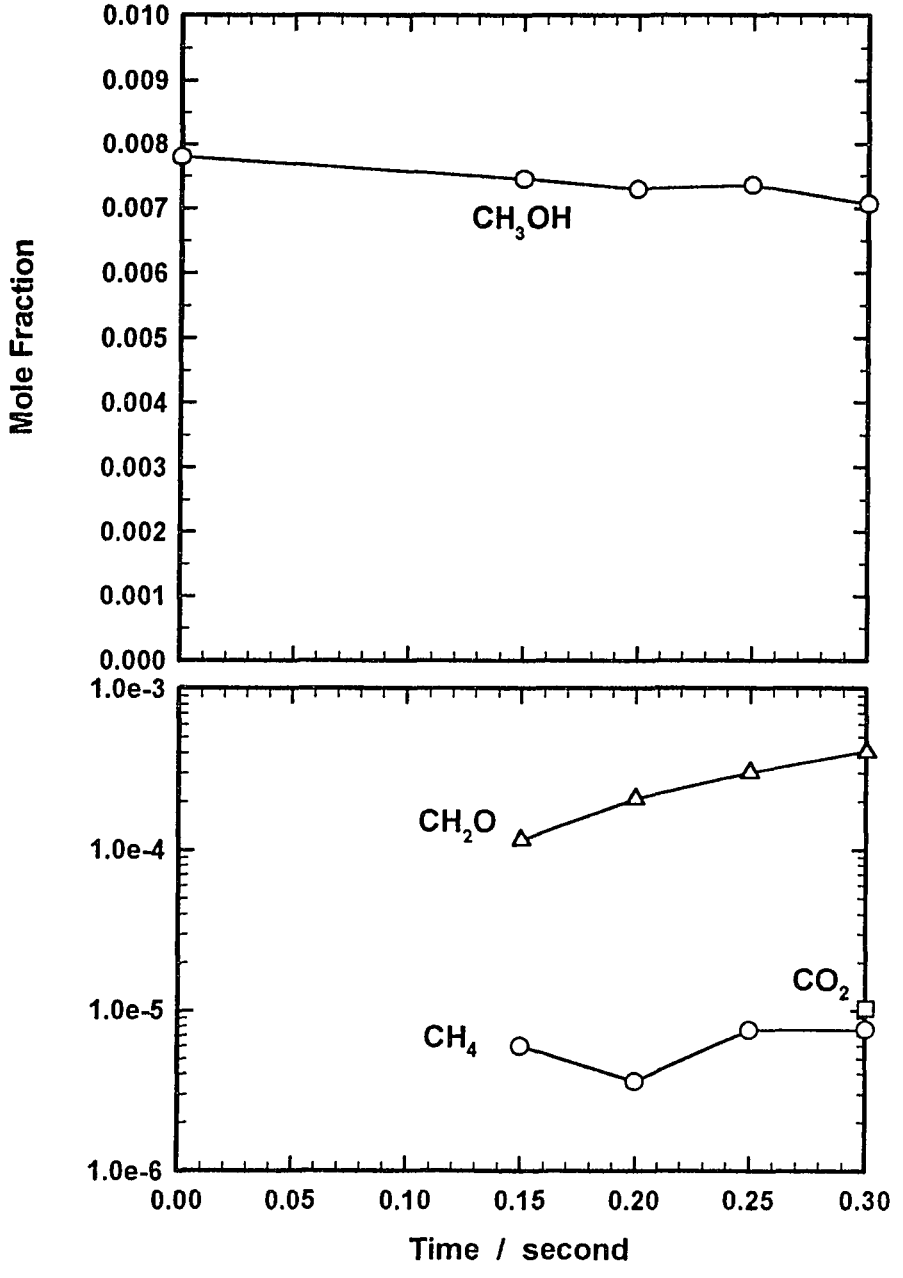


Figure B.19 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 923$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

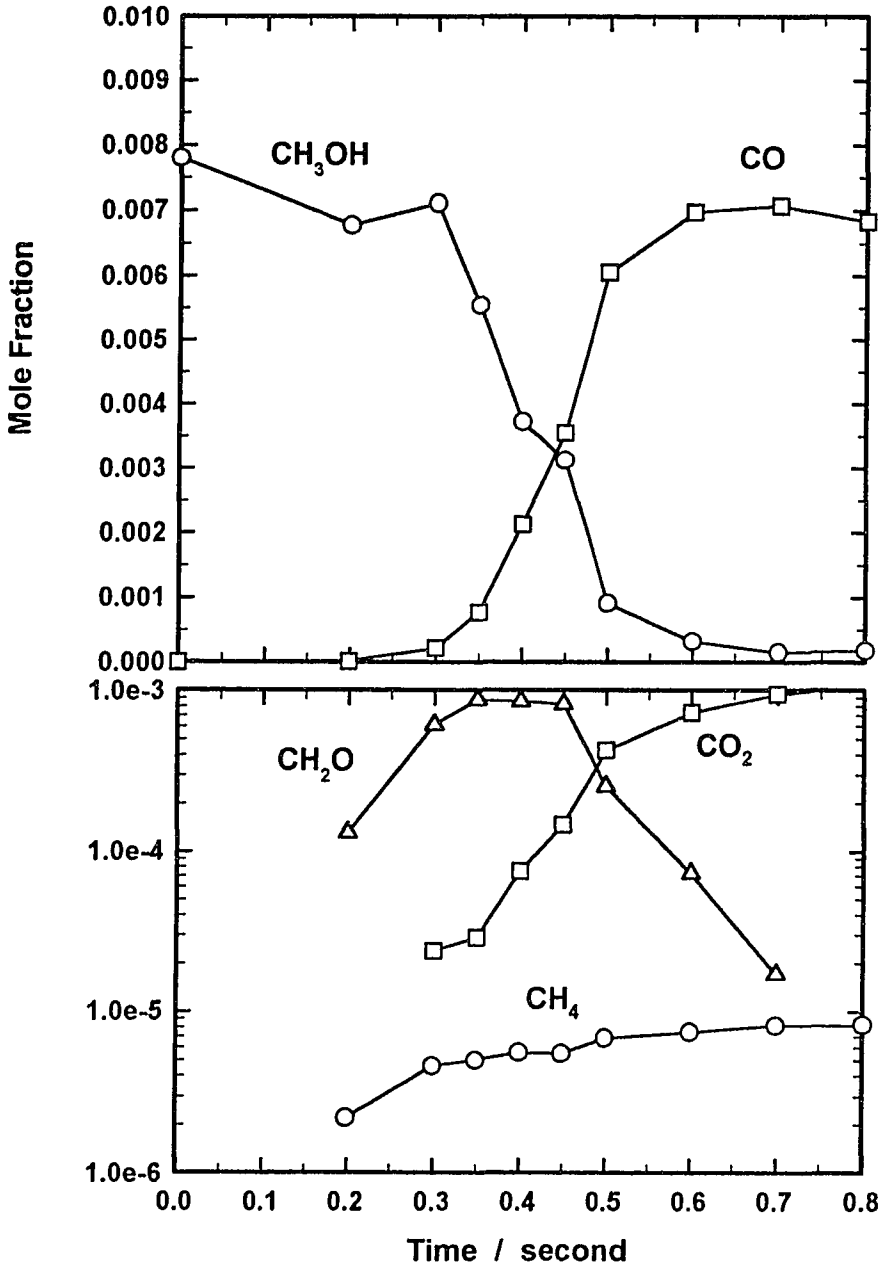


Figure B.20 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 923$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

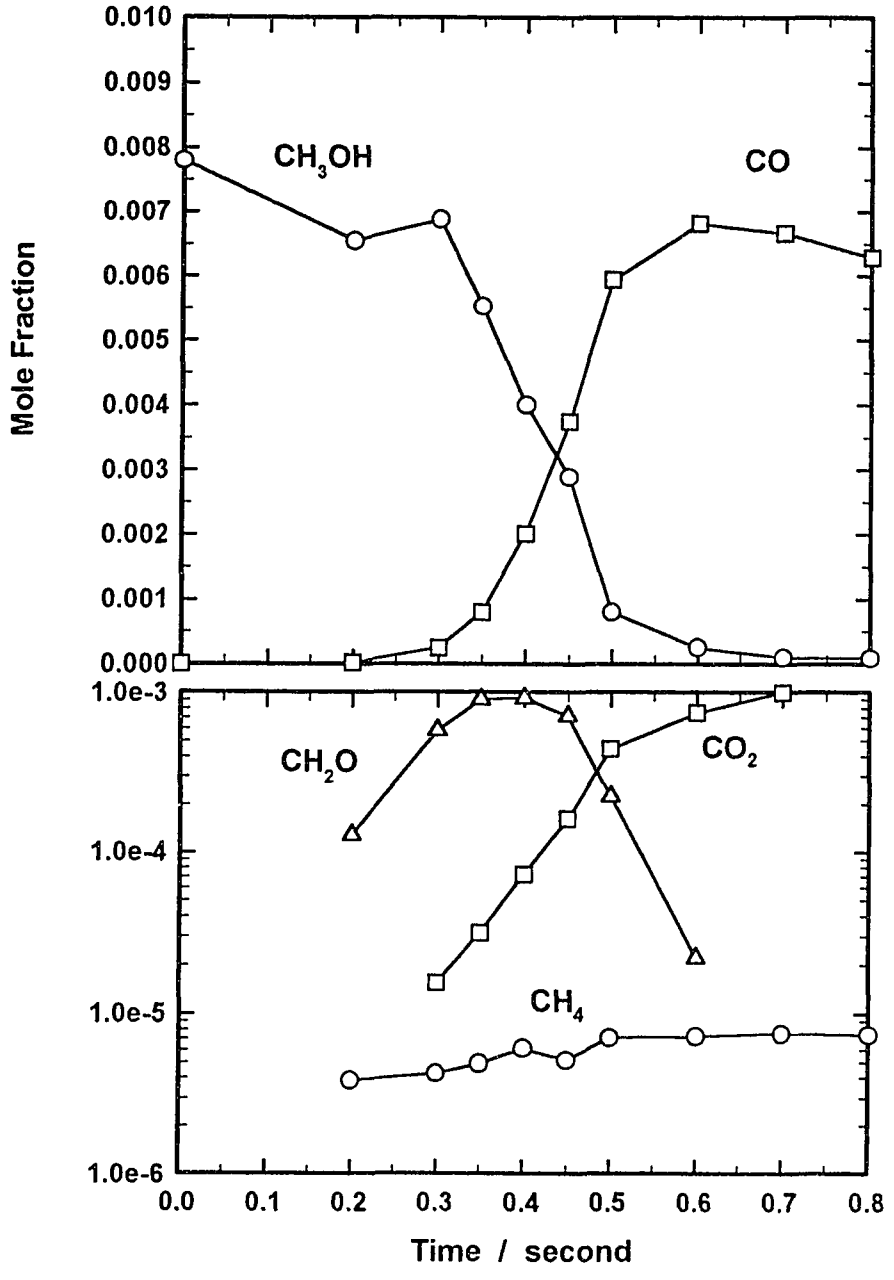


Figure B.21 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 923$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

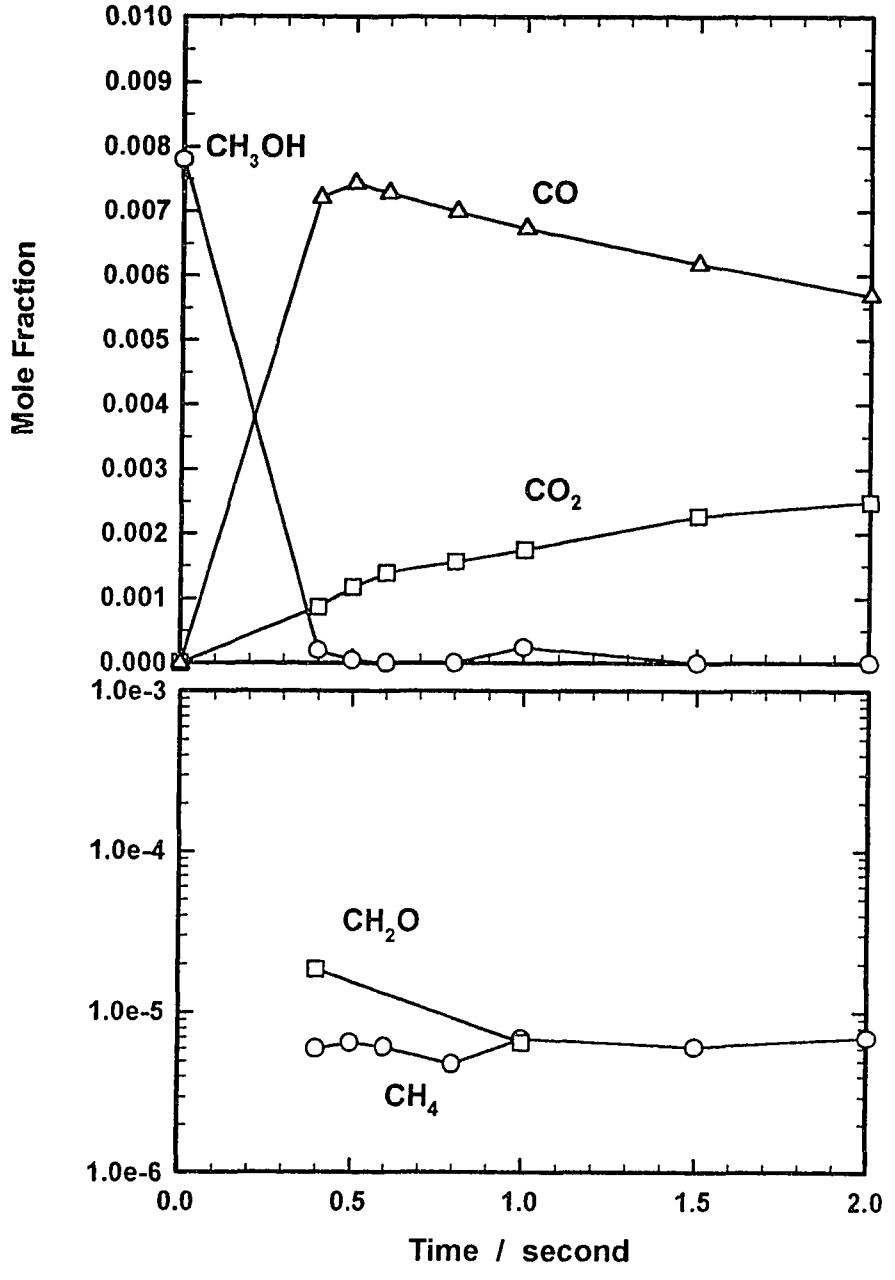


Figure B.22 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 923$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

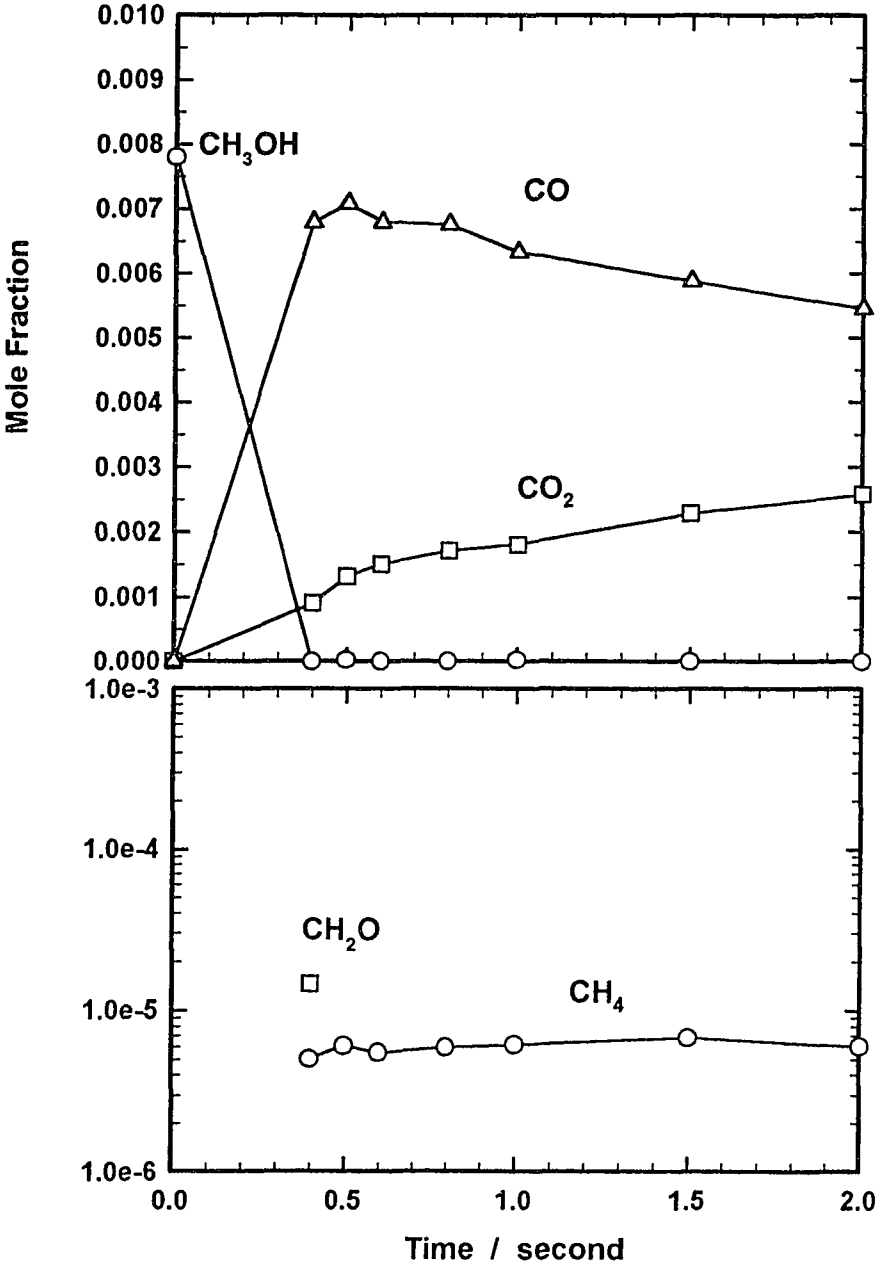


Figure B.23 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 923$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

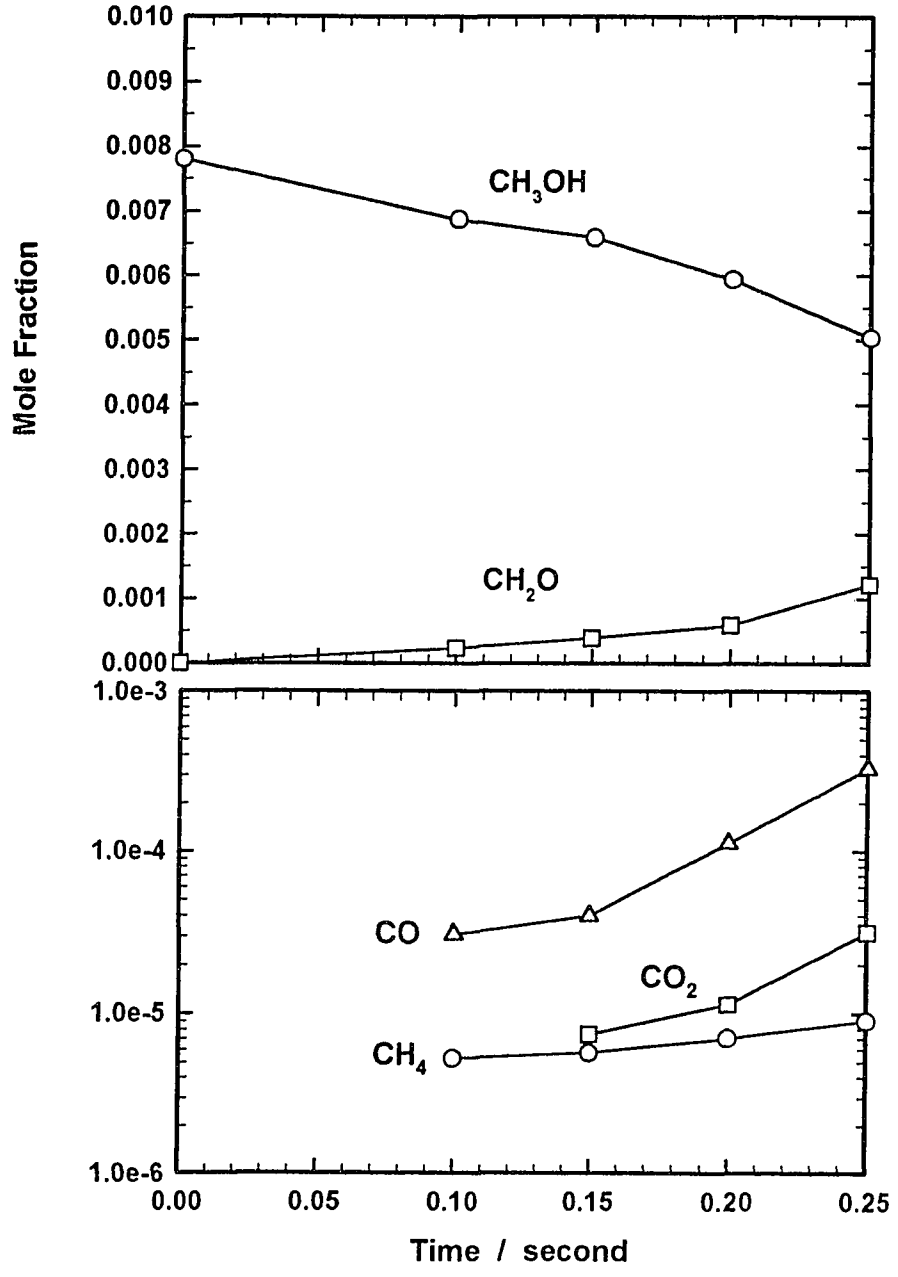


Figure B.24 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 973$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

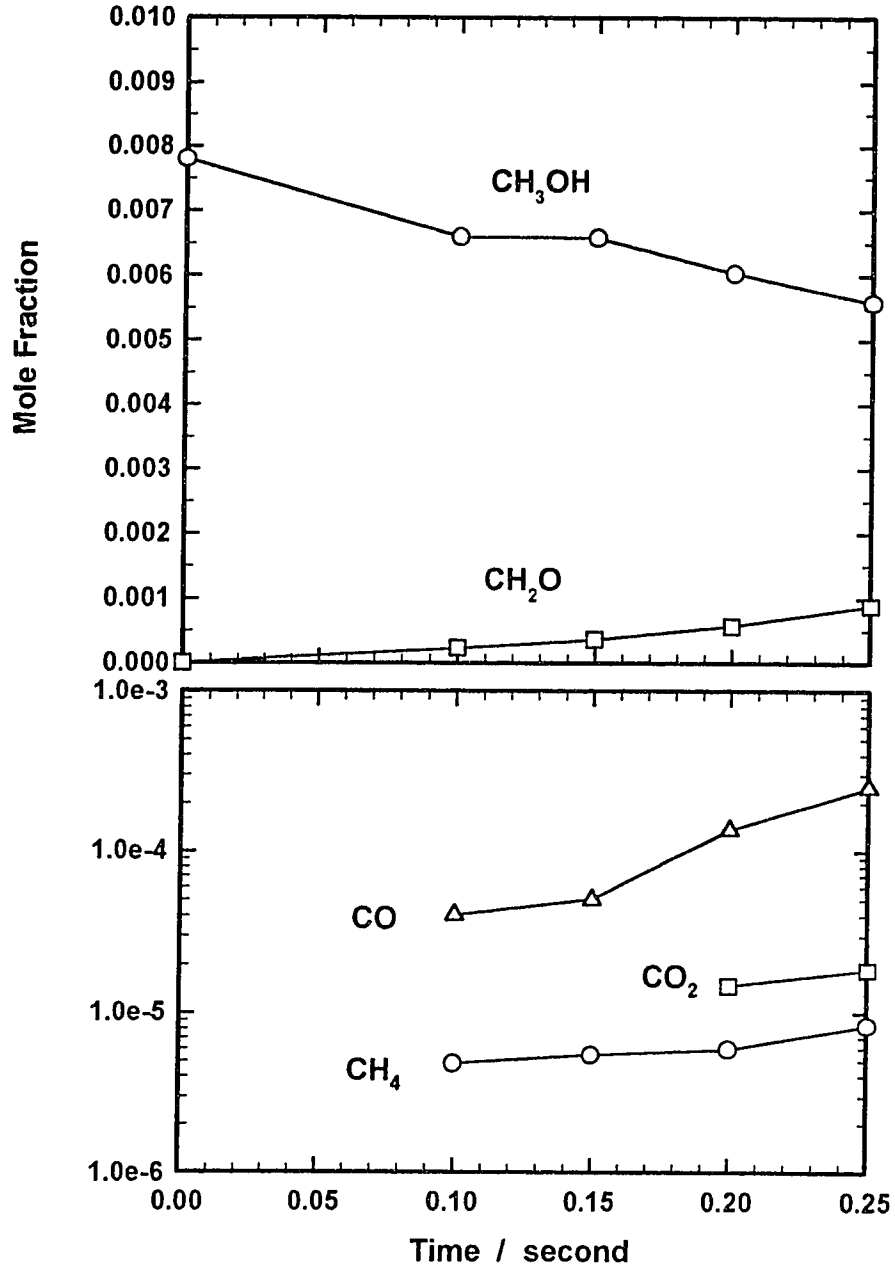


Figure B.25 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 973$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

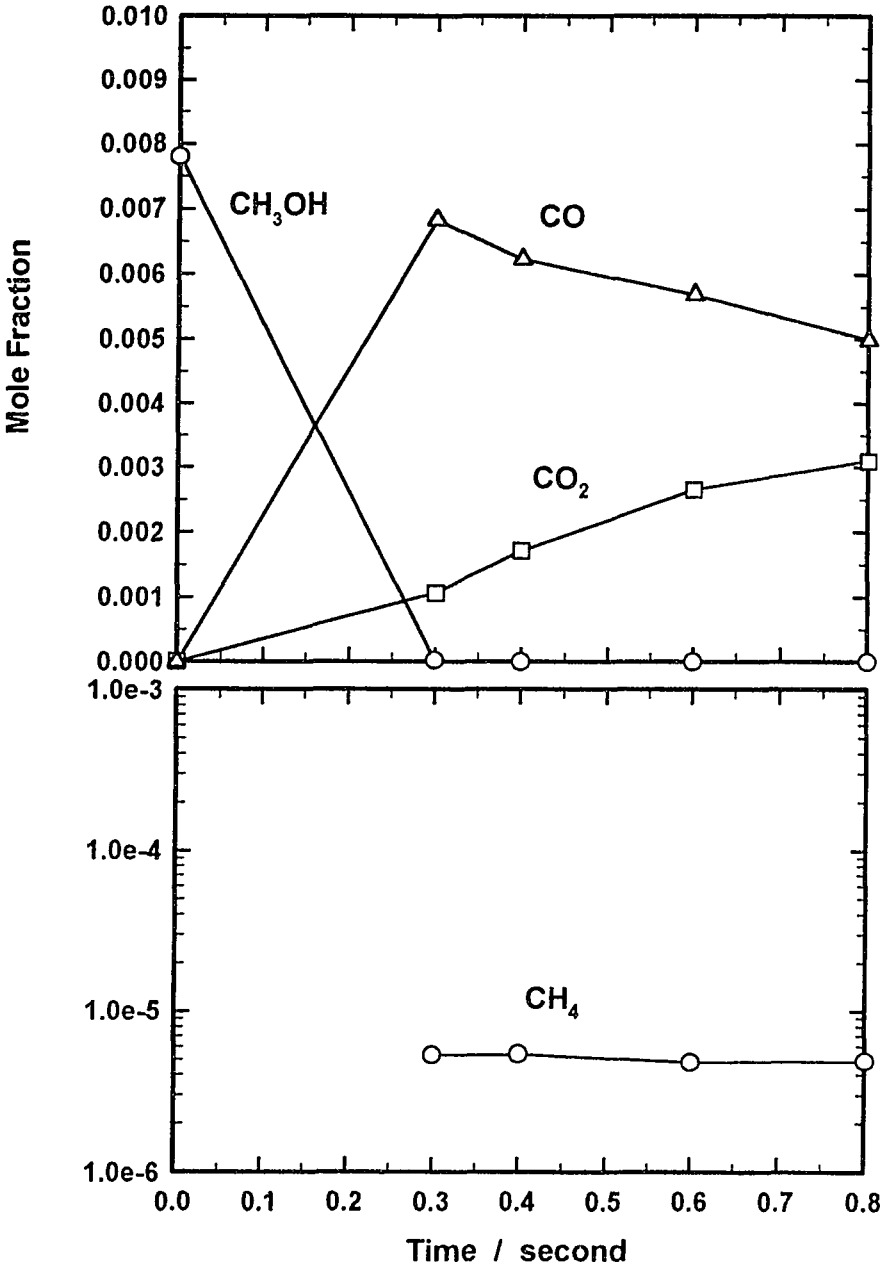


Figure B.26 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 973$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

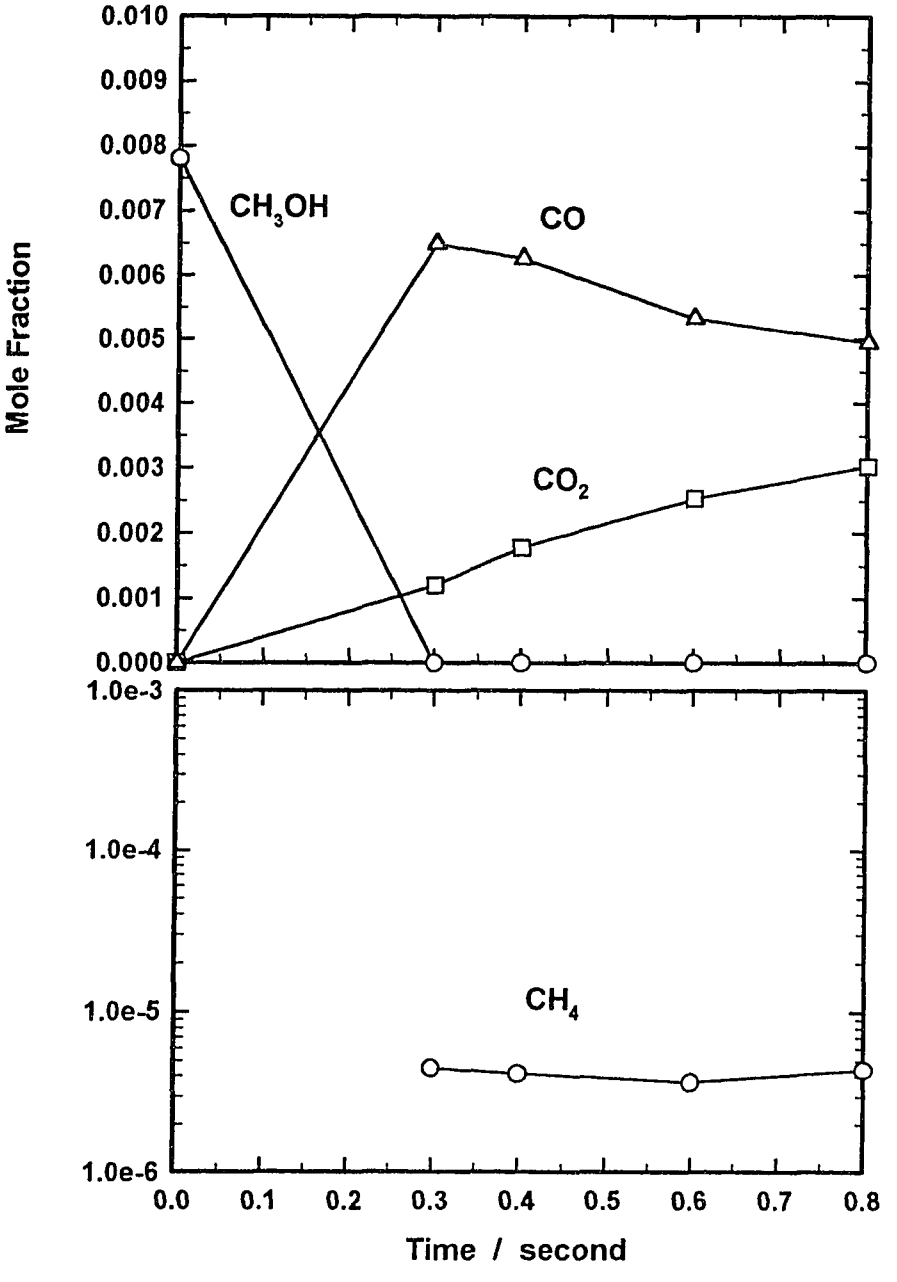


Figure B.27 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 973$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

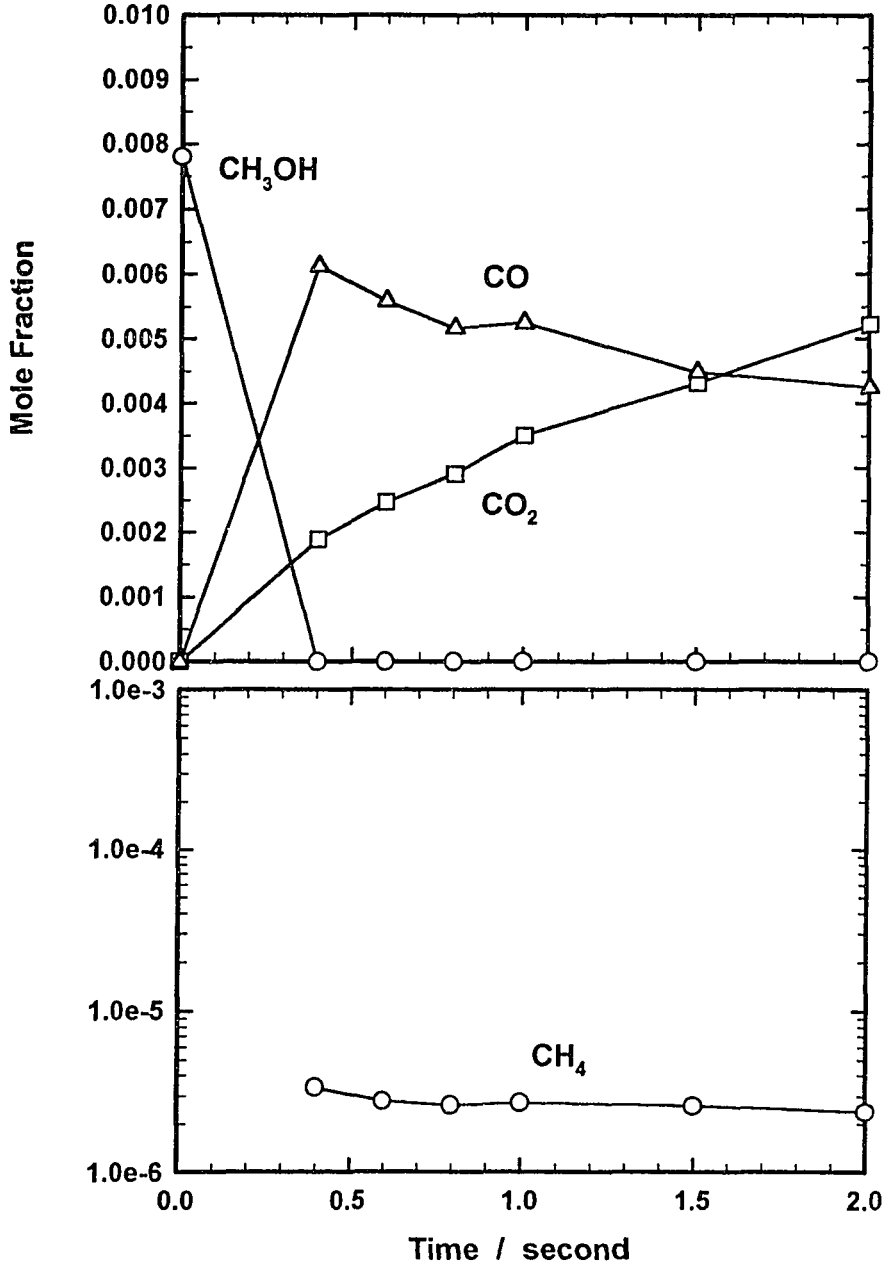


Figure B.28 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 973$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

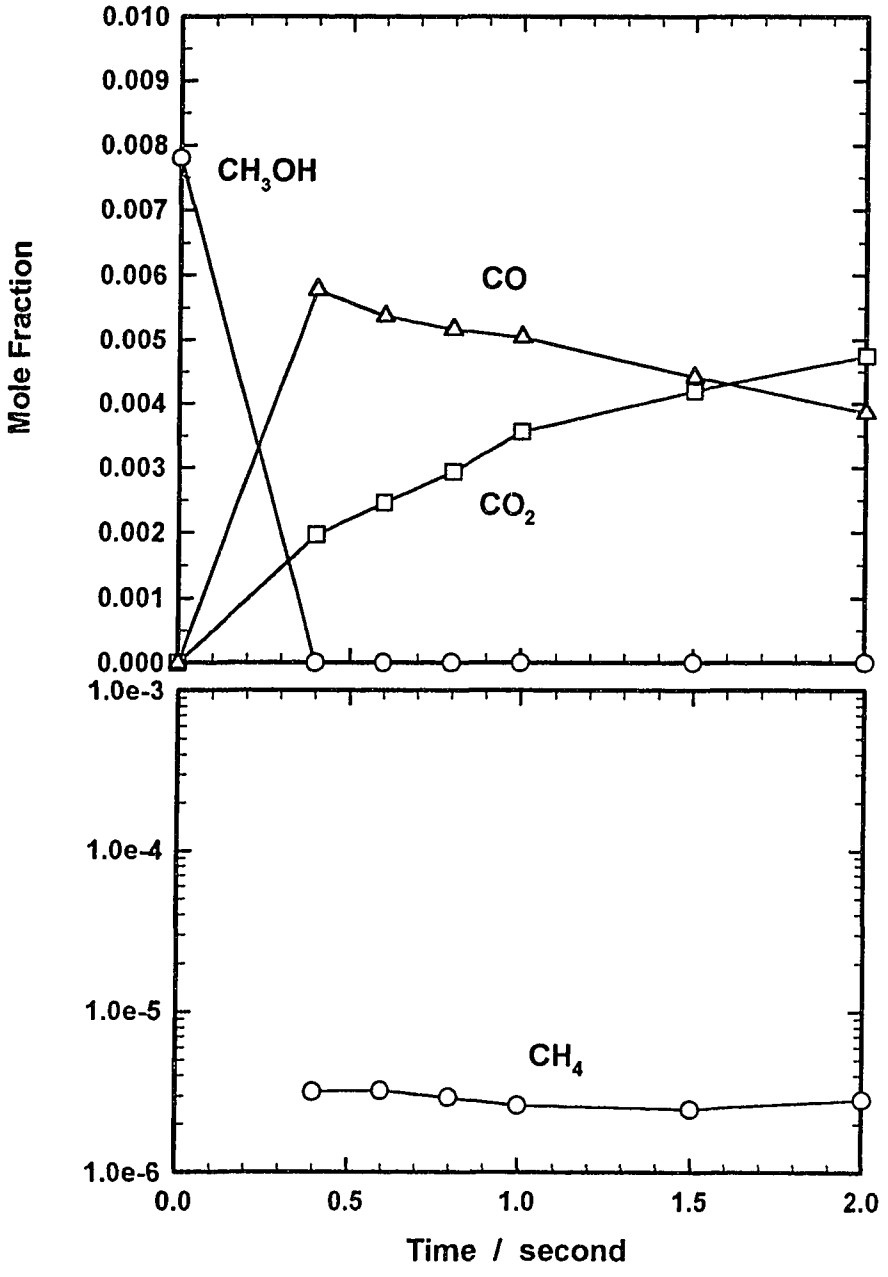


Figure B.29 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 973$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

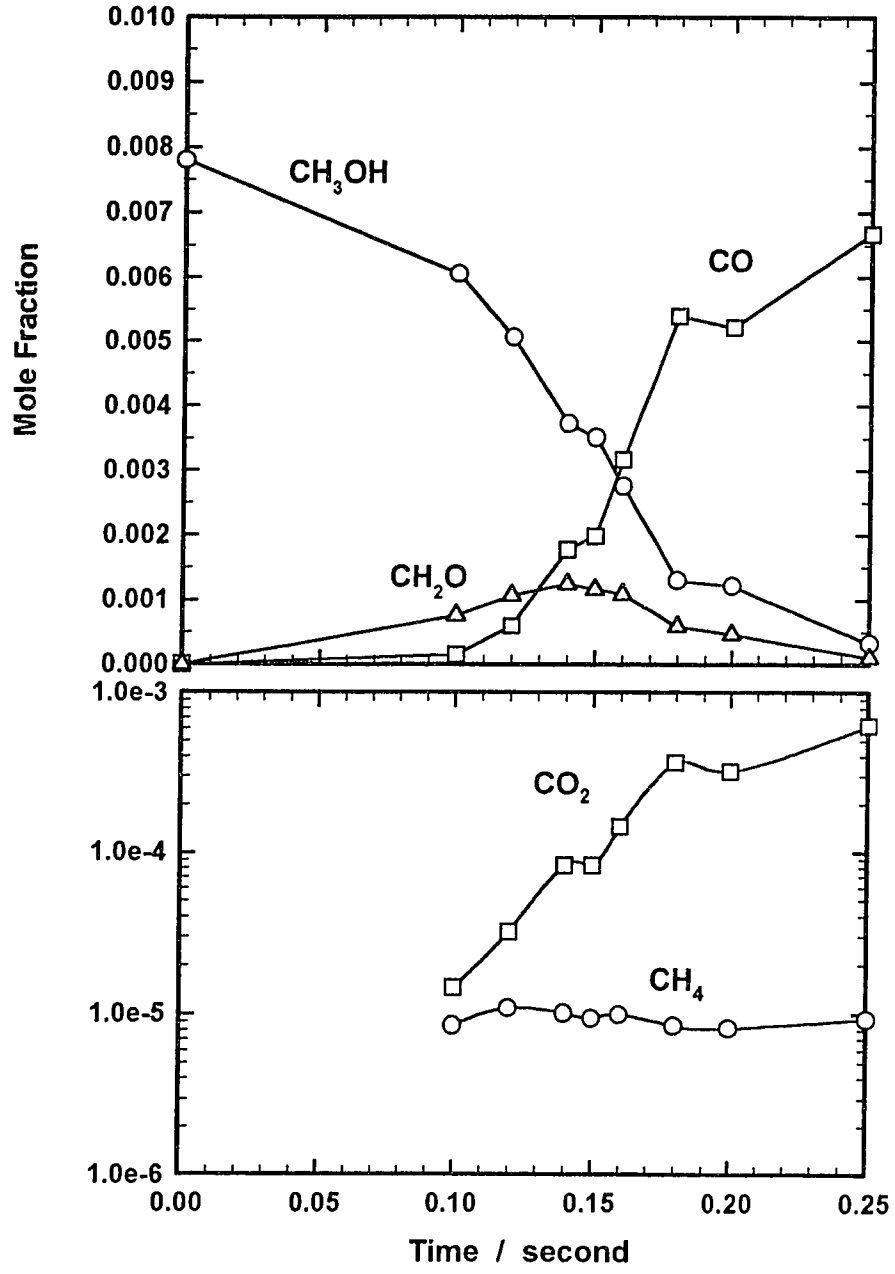


Figure B.30 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1023$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

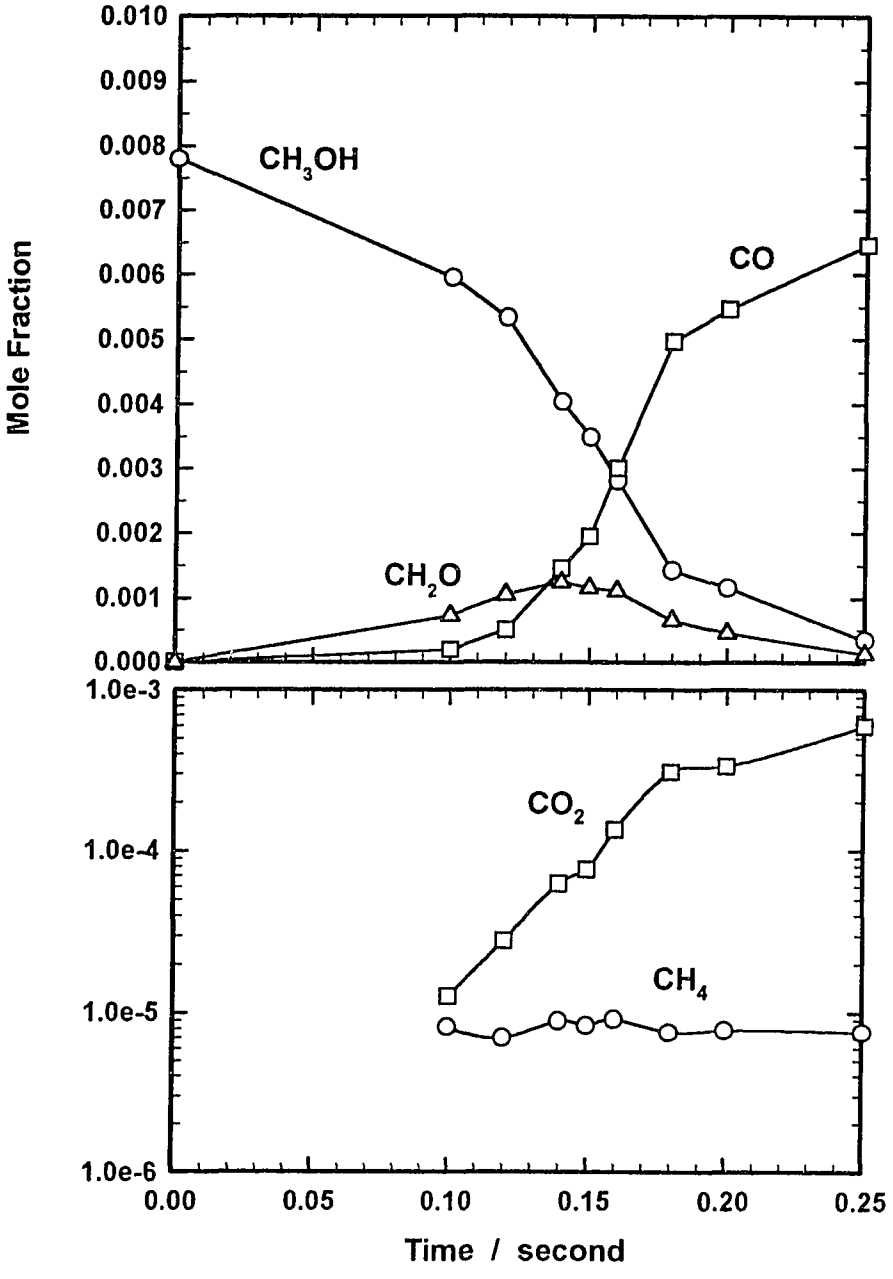


Figure B.31 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1023$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

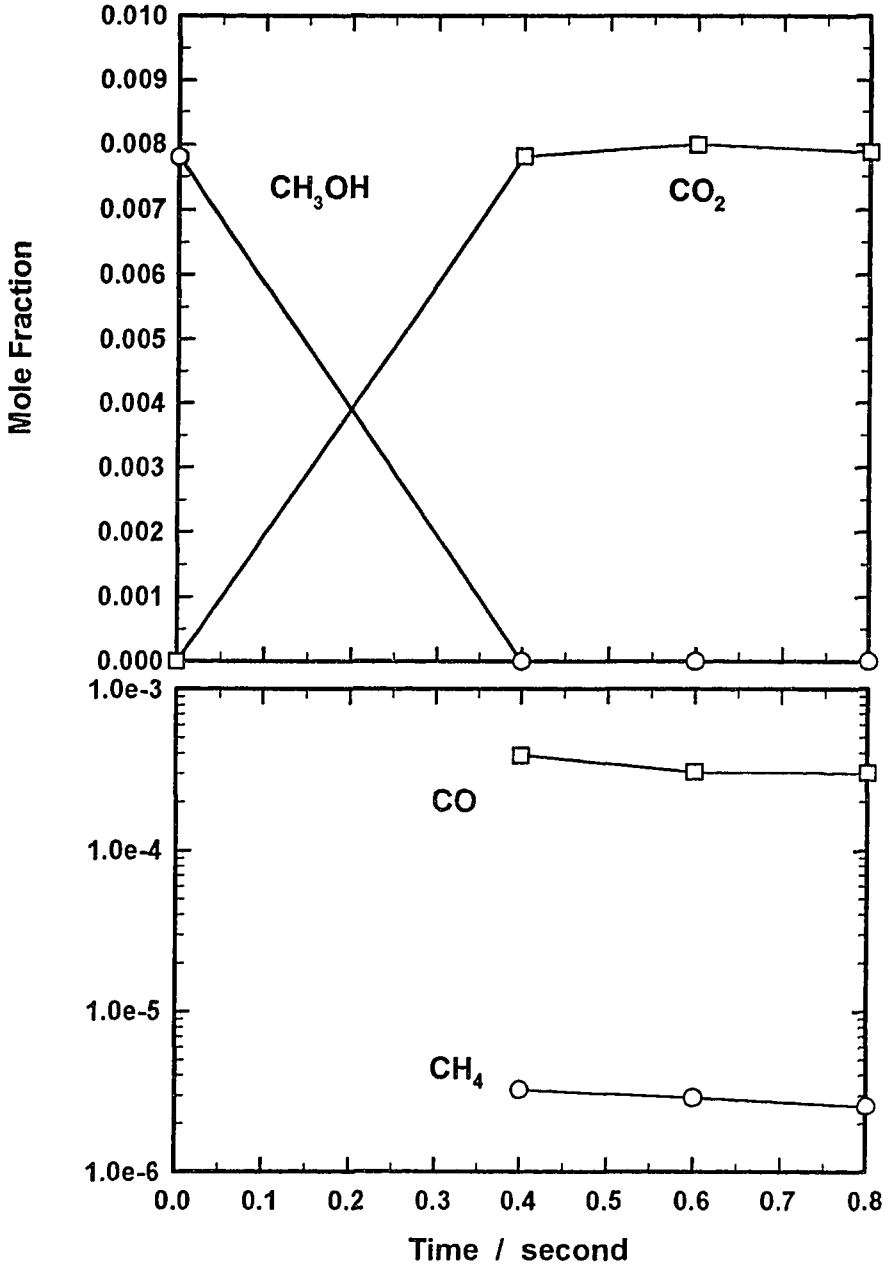


Figure B.32 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1023$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

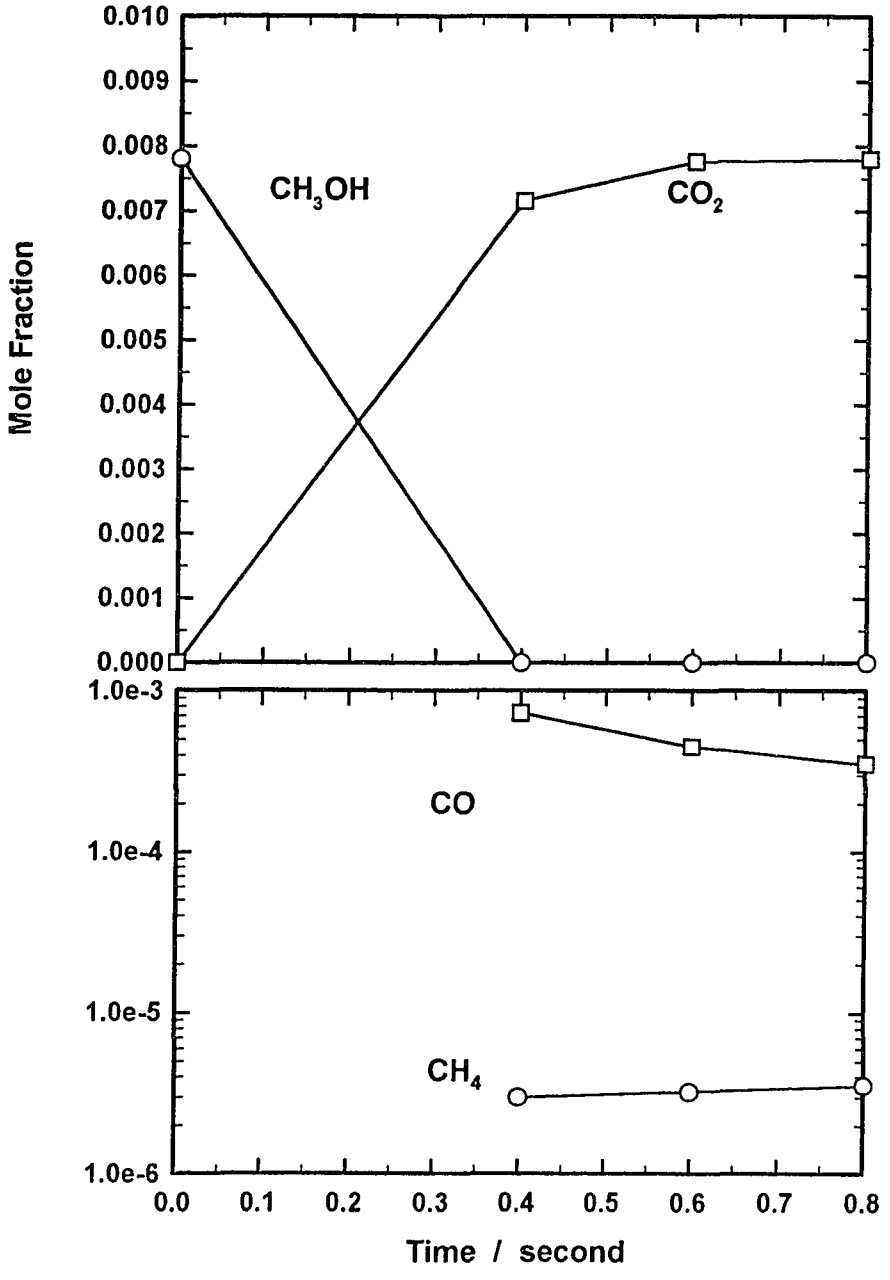


Figure B.33 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1023$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

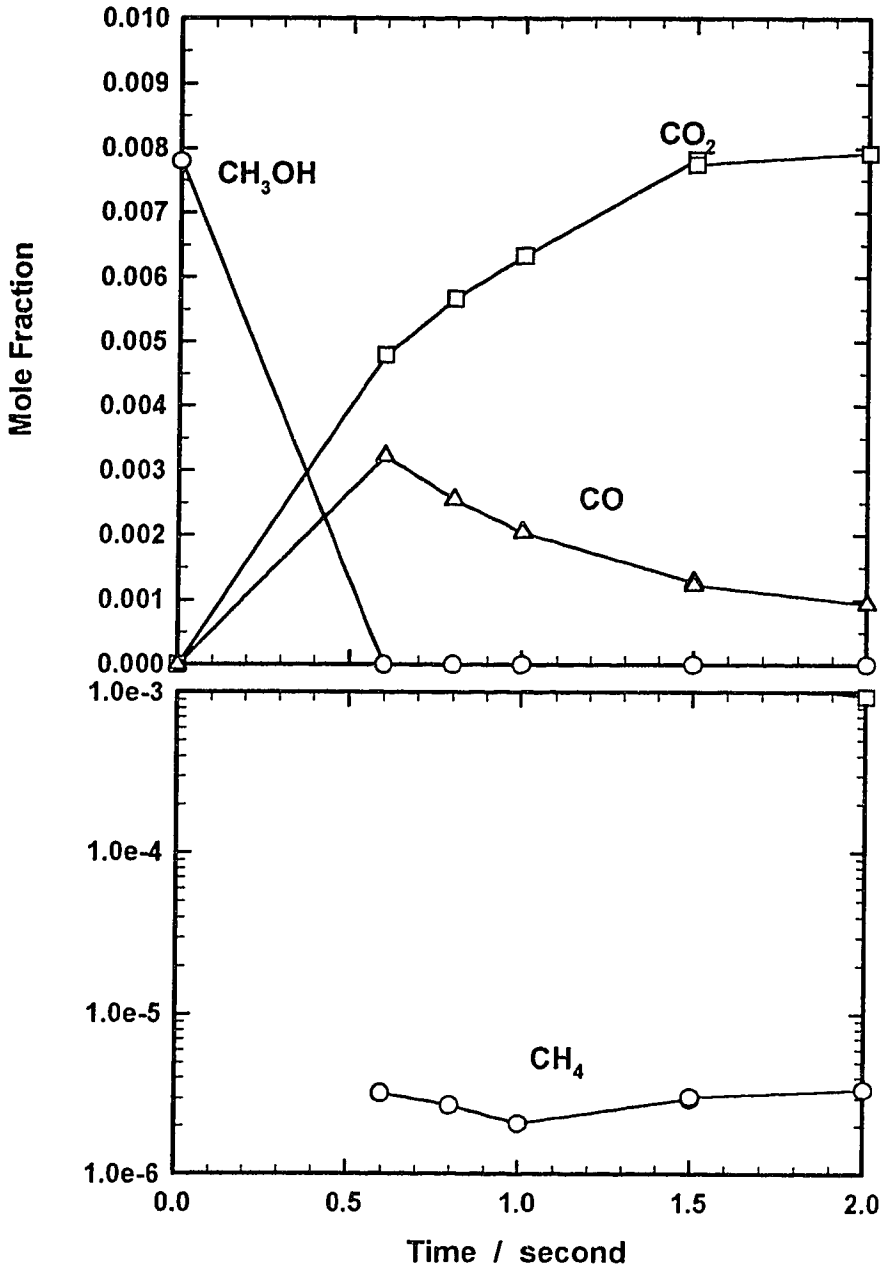


Figure B.34 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1023$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

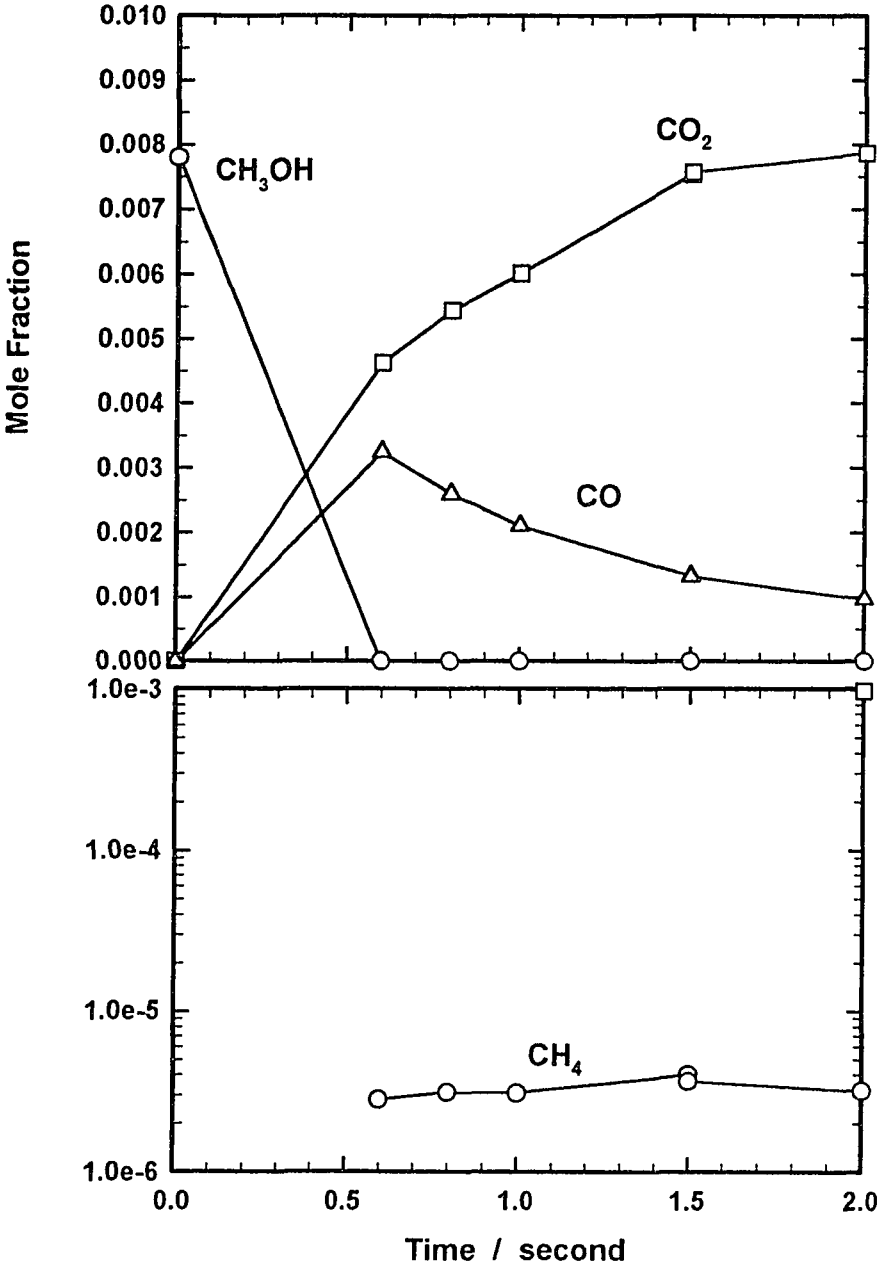


Figure B.35 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1023$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

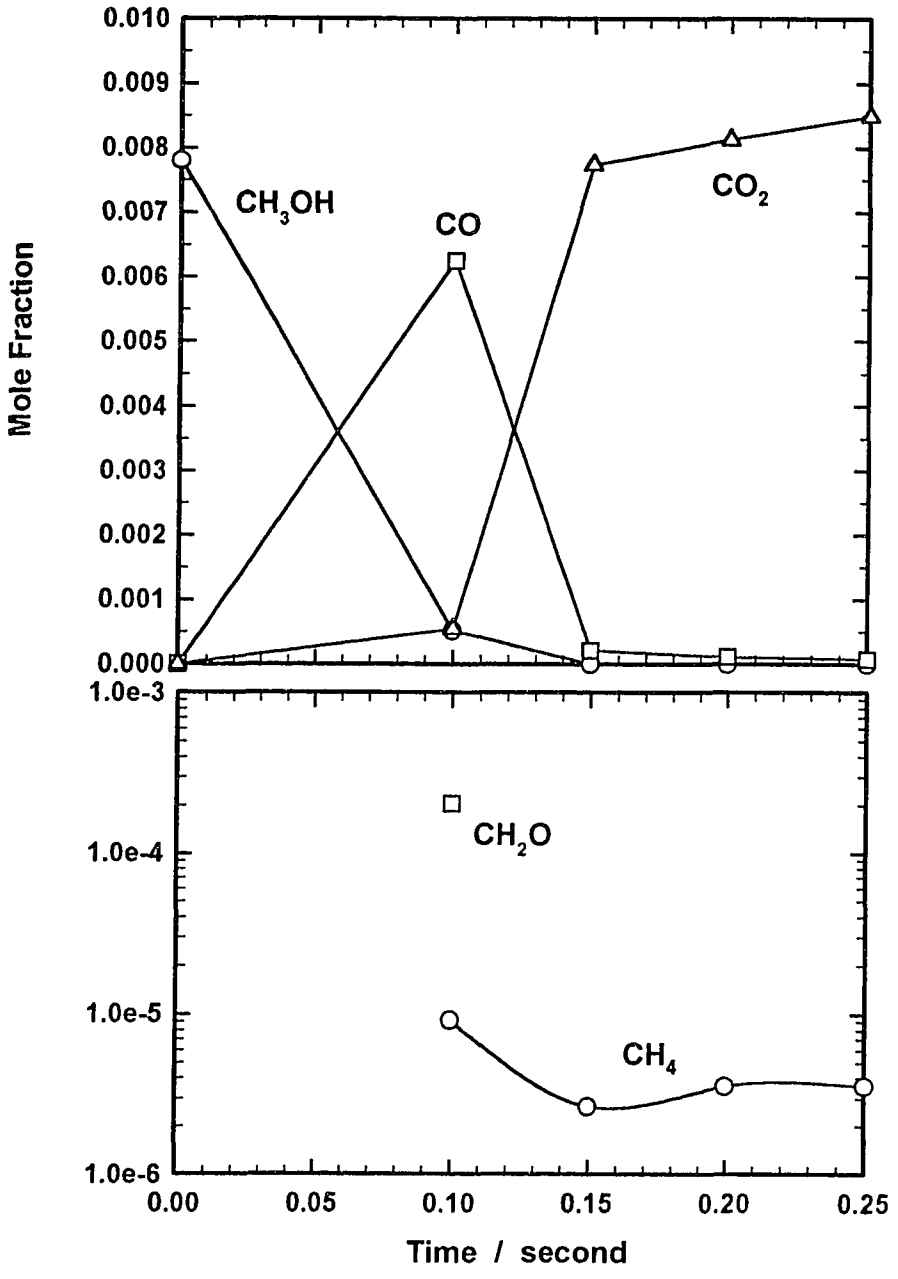


Figure B.36 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

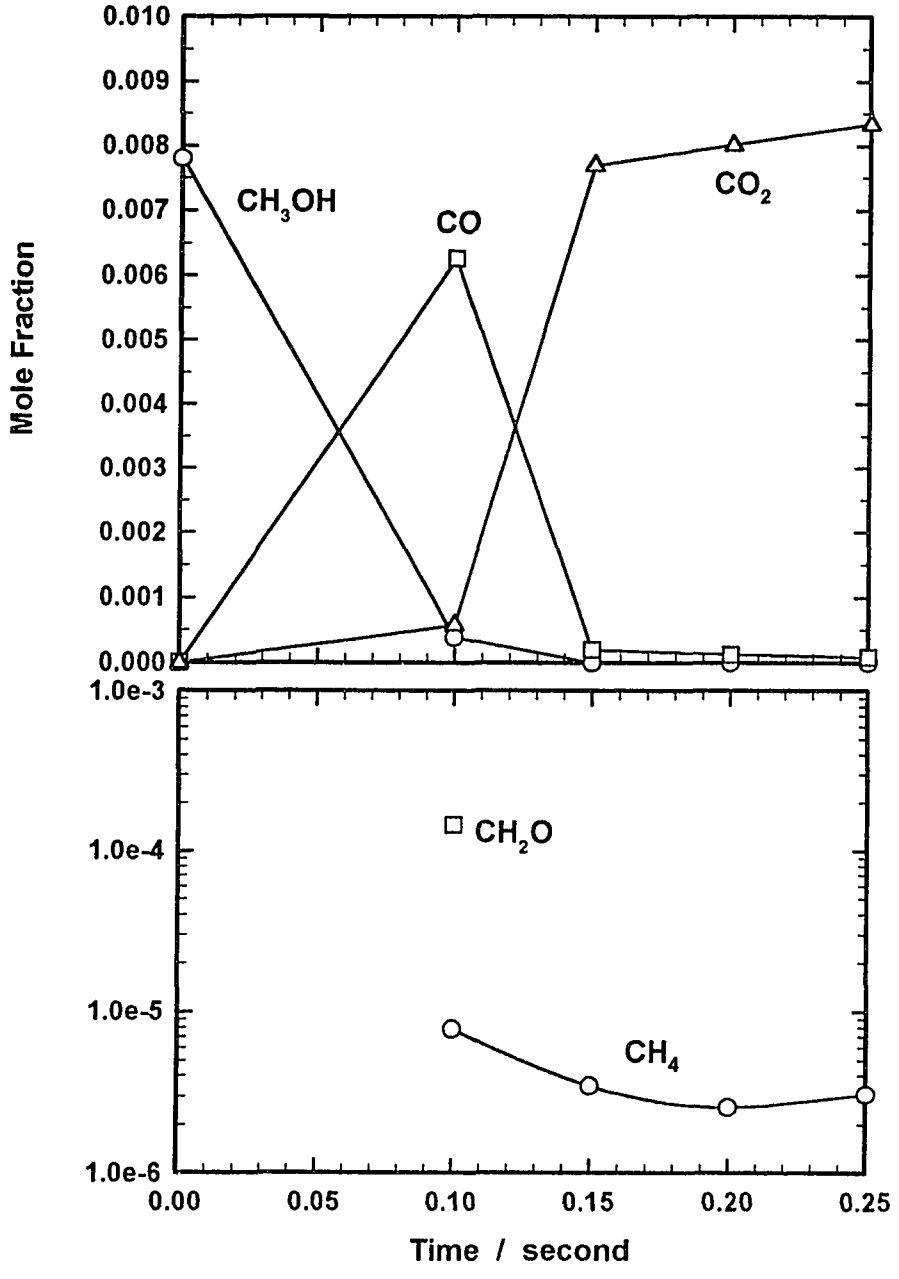


Figure B.37 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1073$ K; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

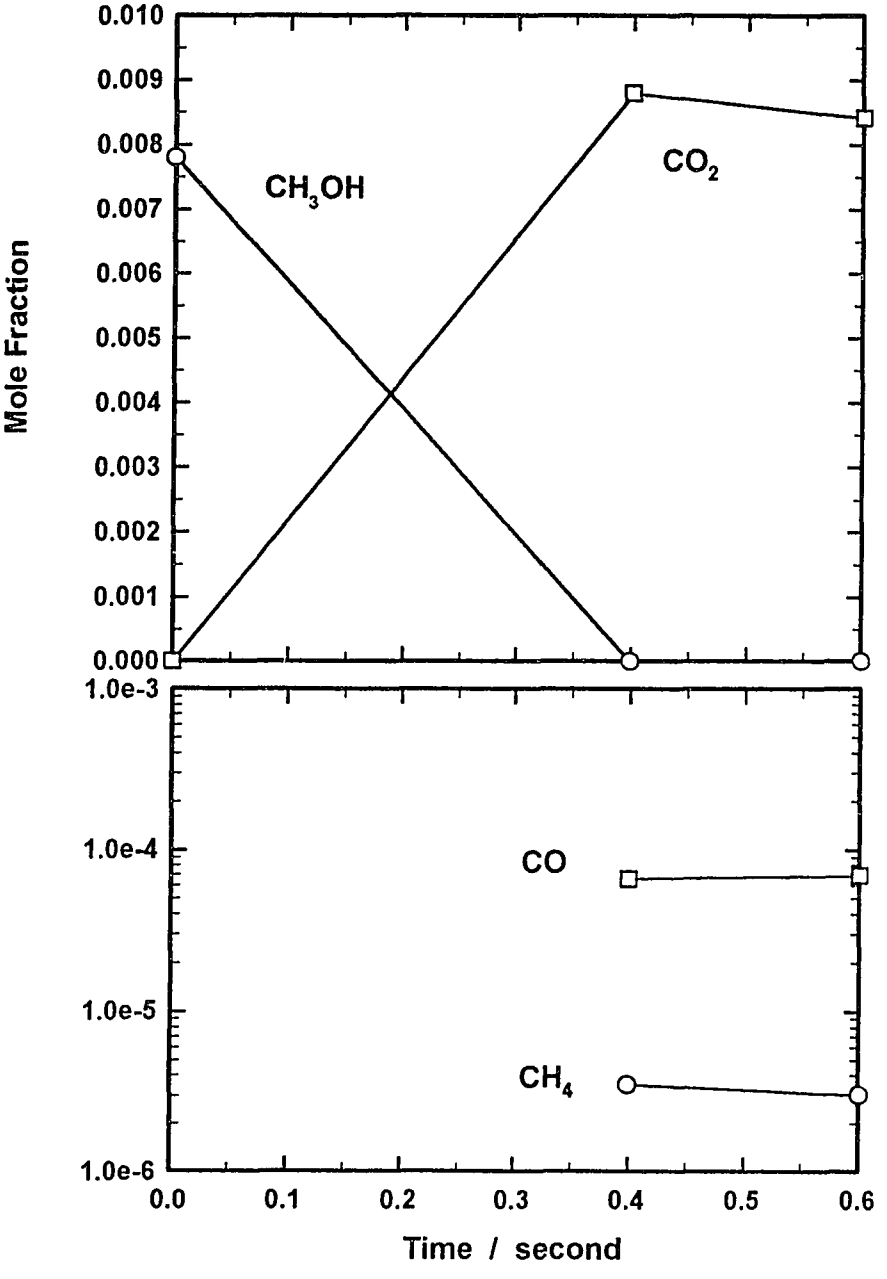


Figure B.38 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

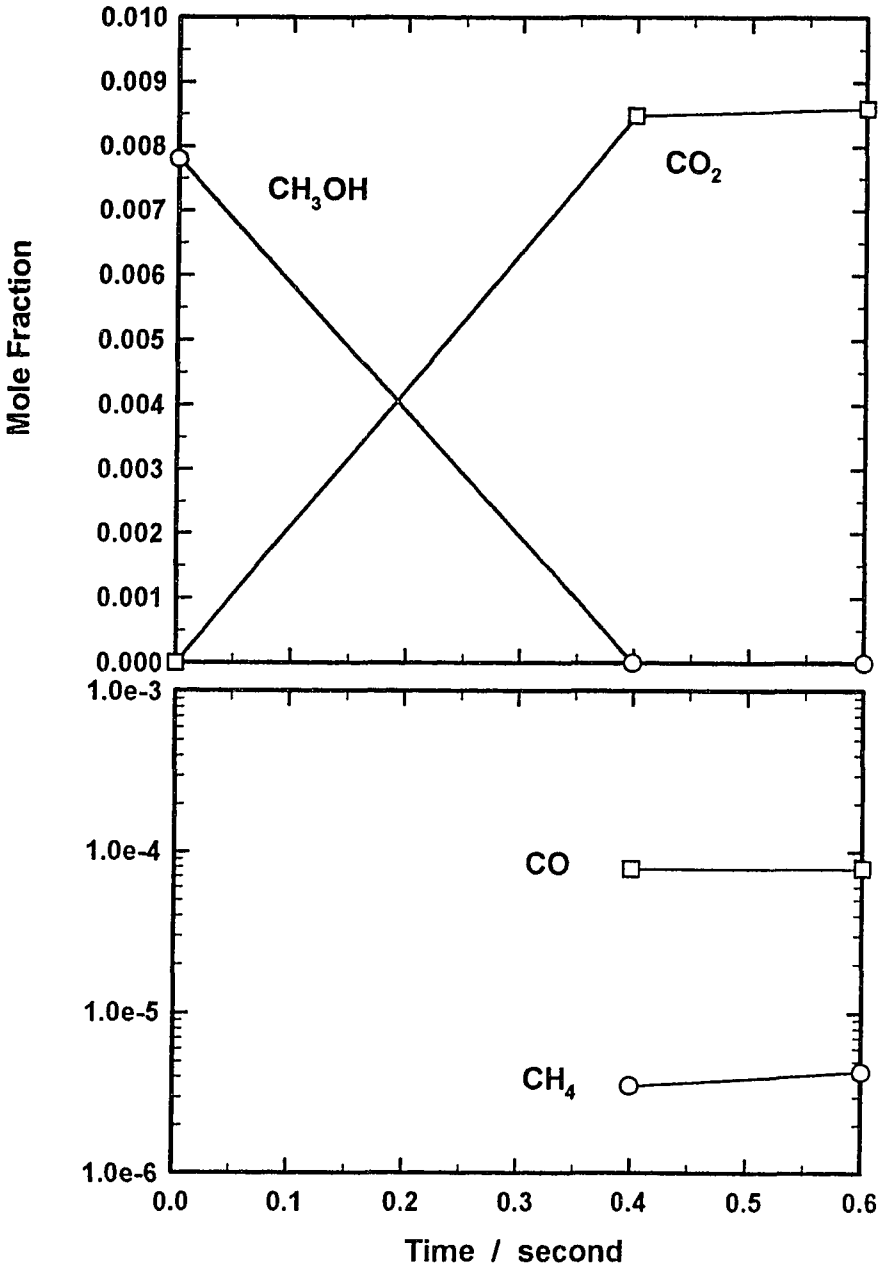


Figure B.39 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1073$ K; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

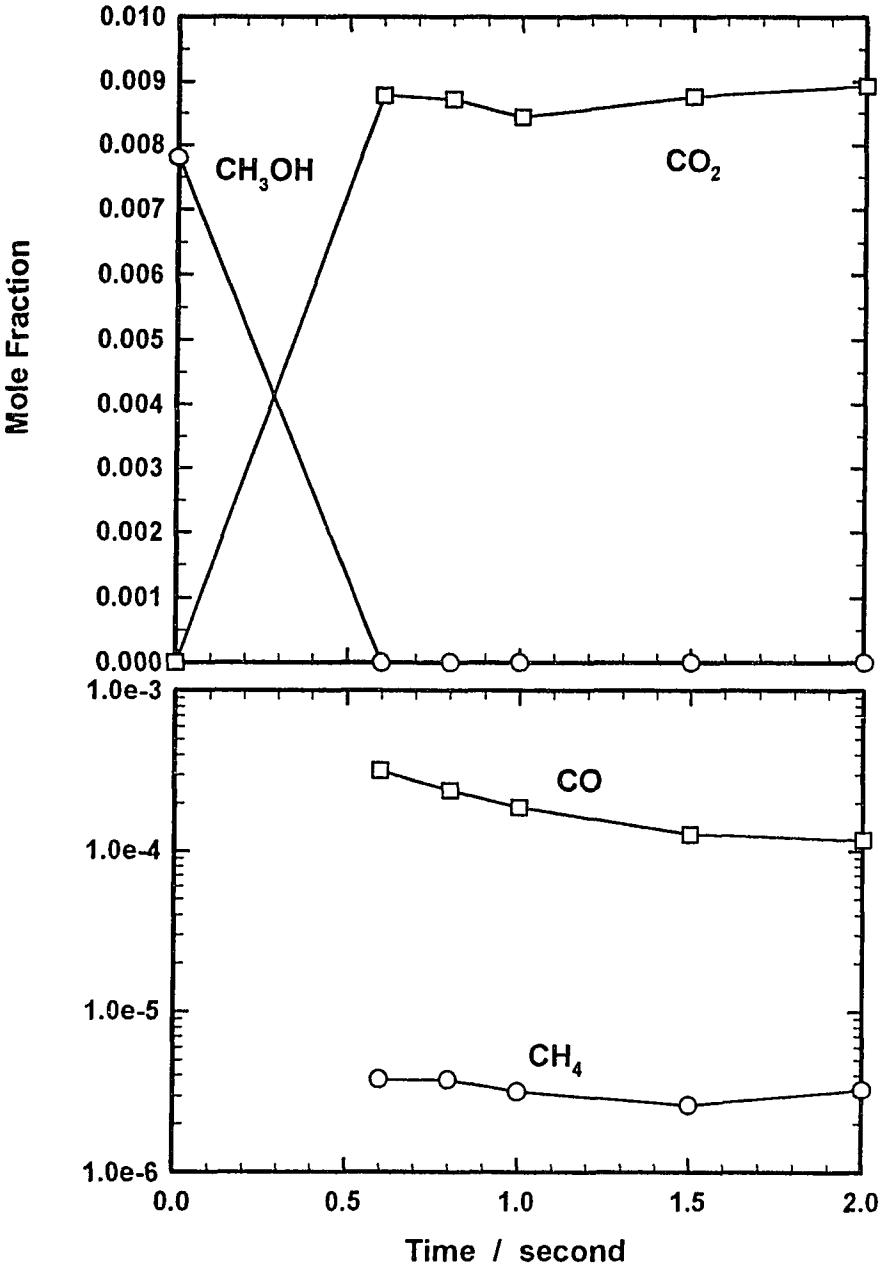


Figure B.40 Experimental result of methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

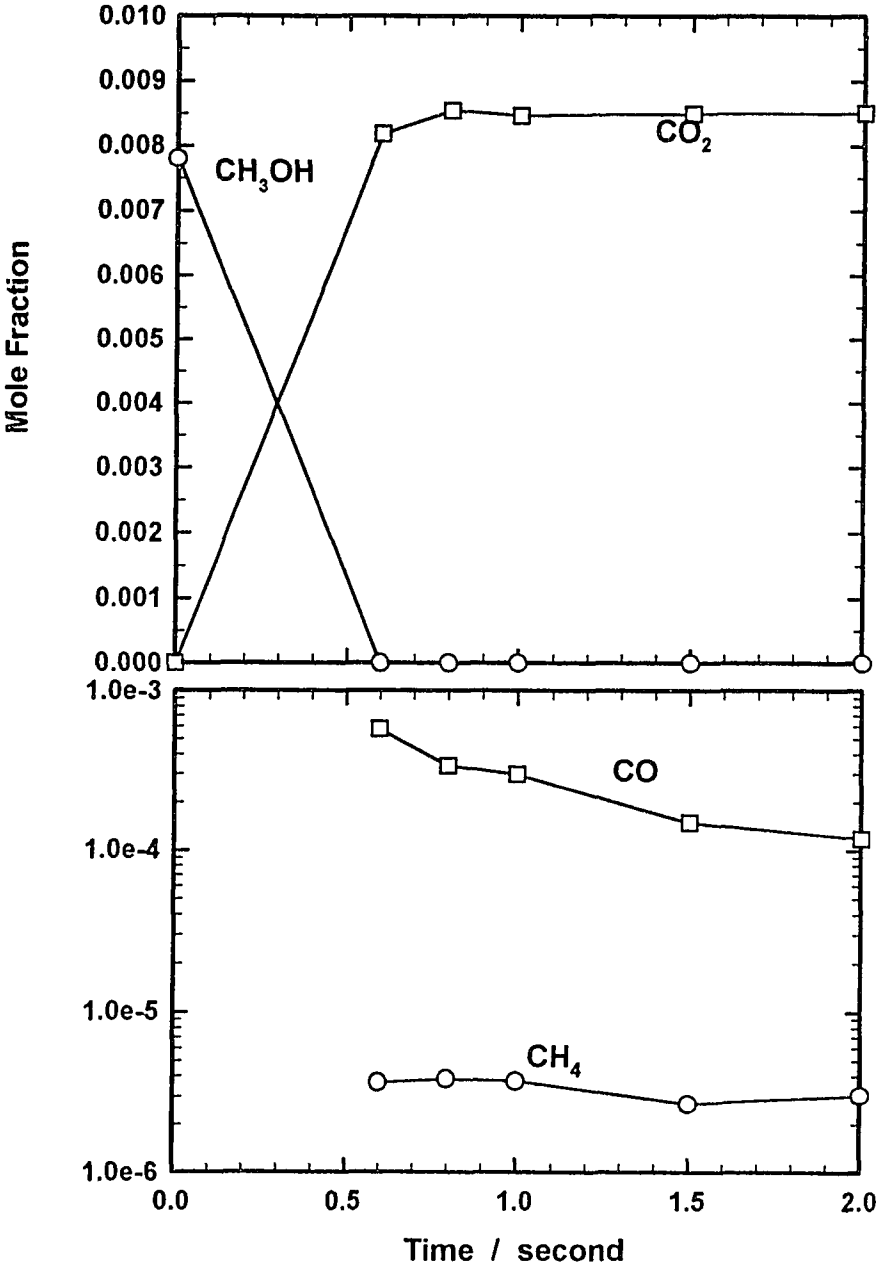


Figure B.41 Experimental result of methanol oxidation: $\varphi = 0.75$; $T = 1073$ K; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

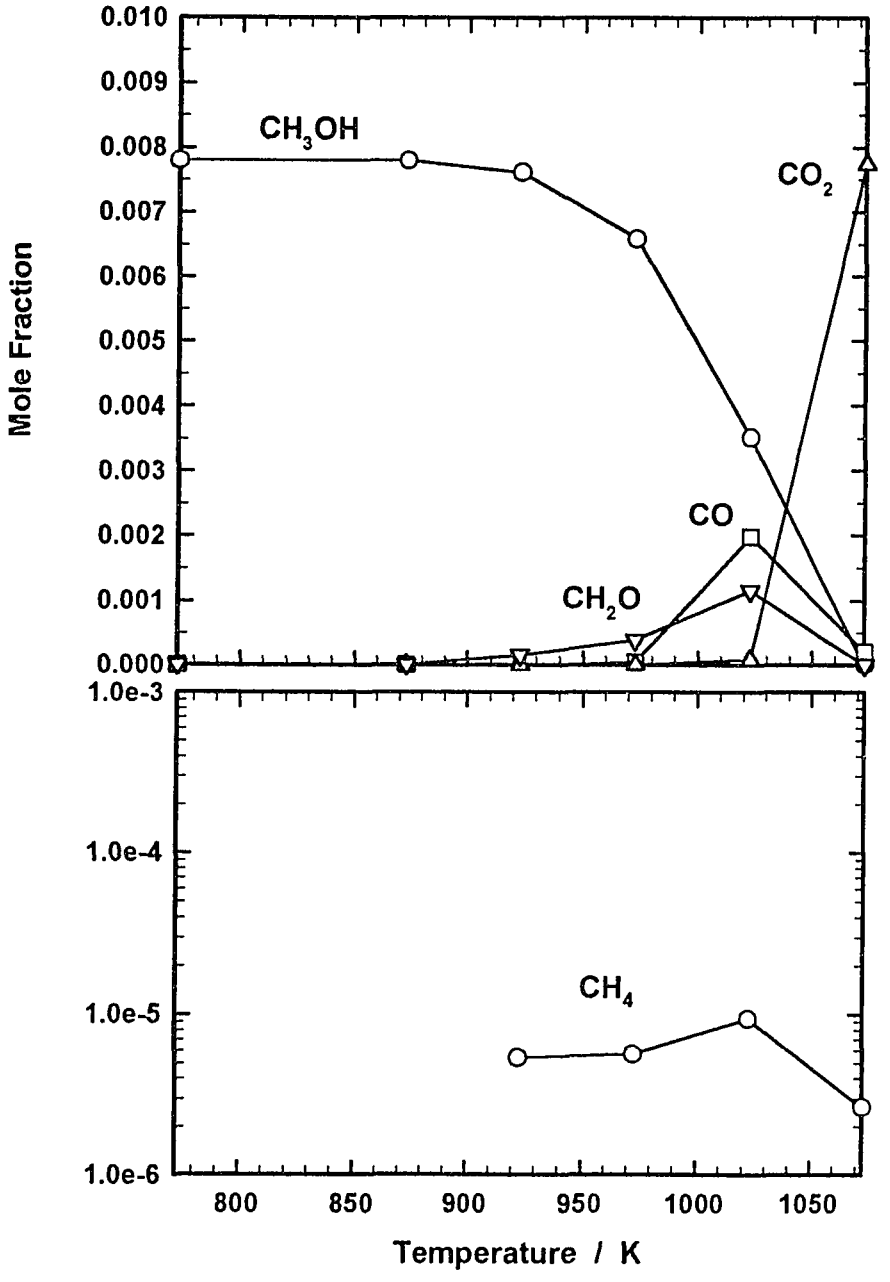


Figure B.42 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.15$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

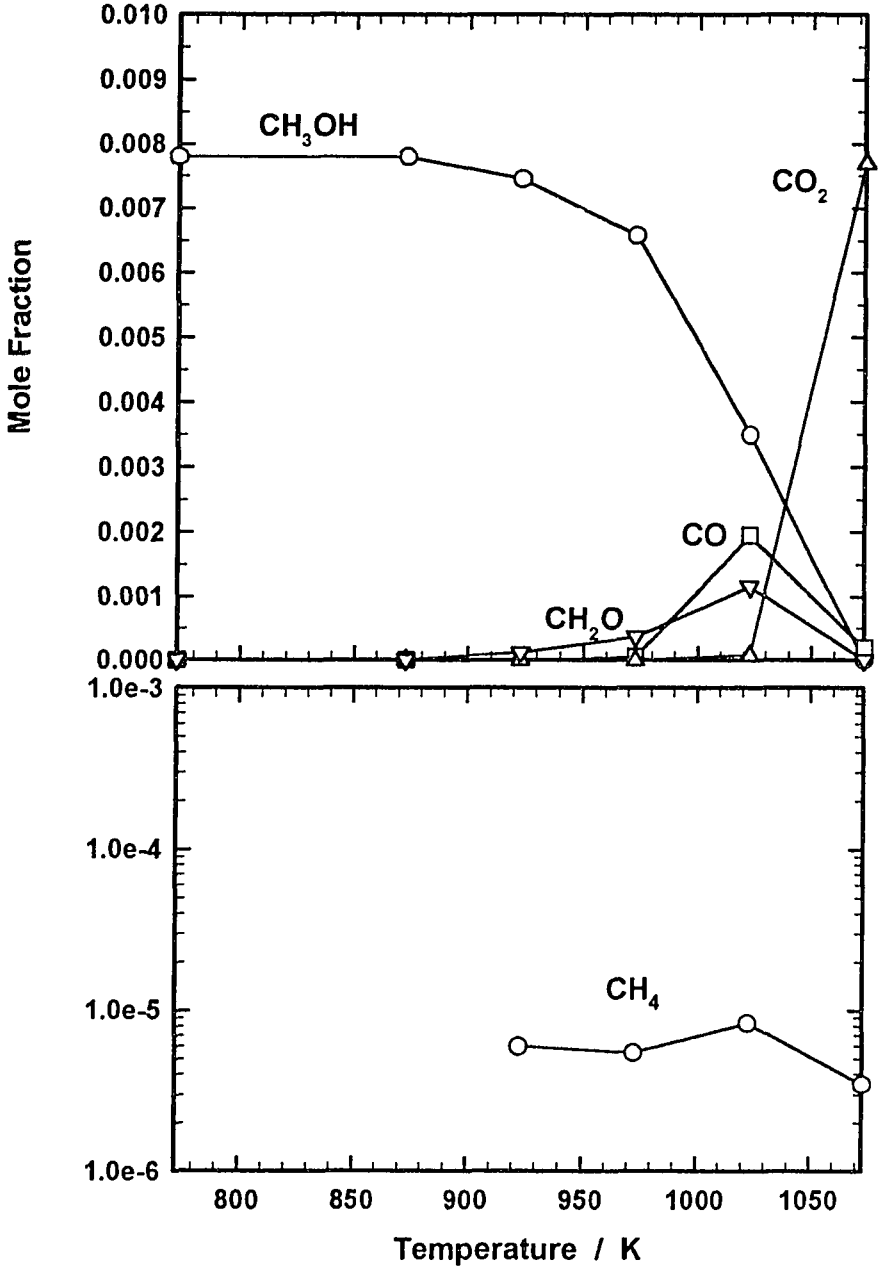


Figure B.43 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.15$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

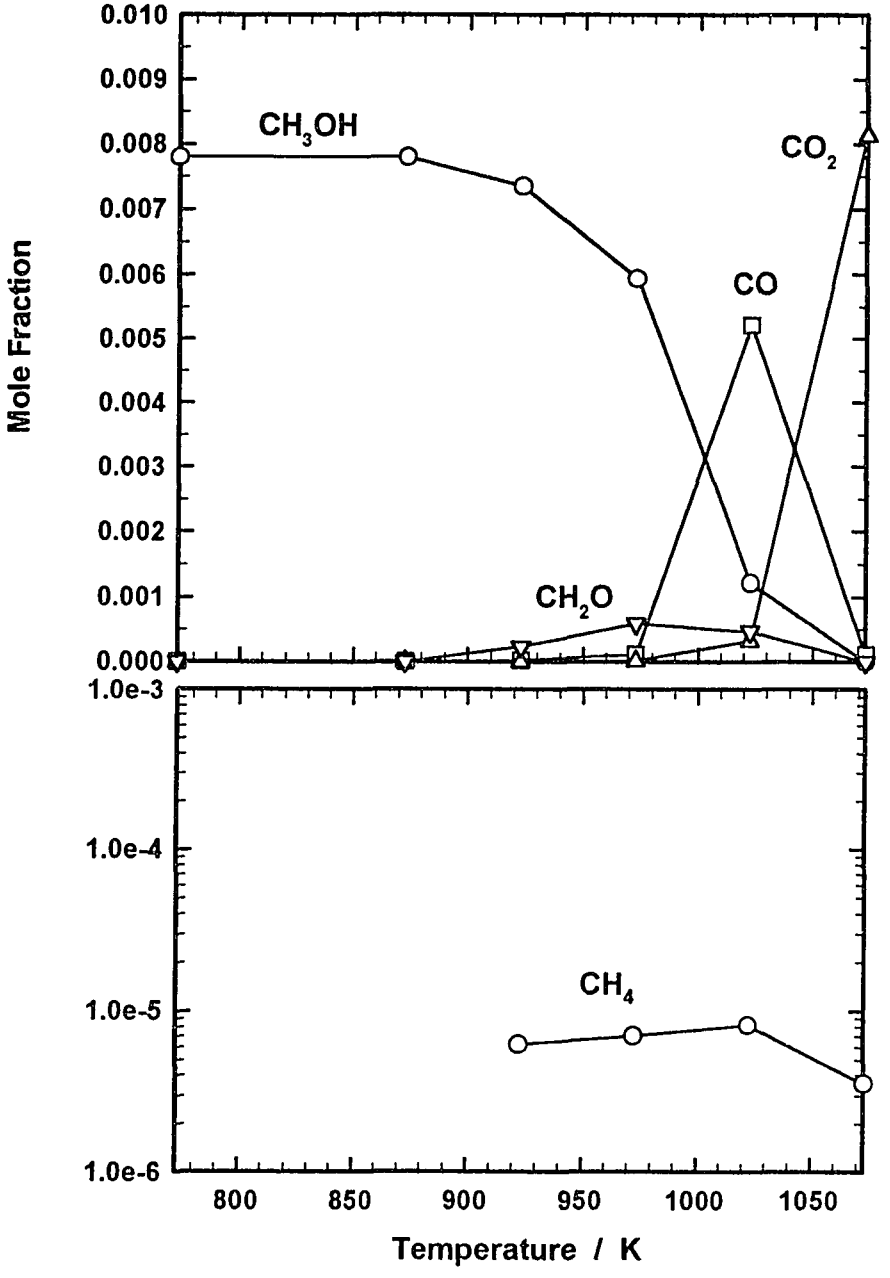


Figure B.44 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.2$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

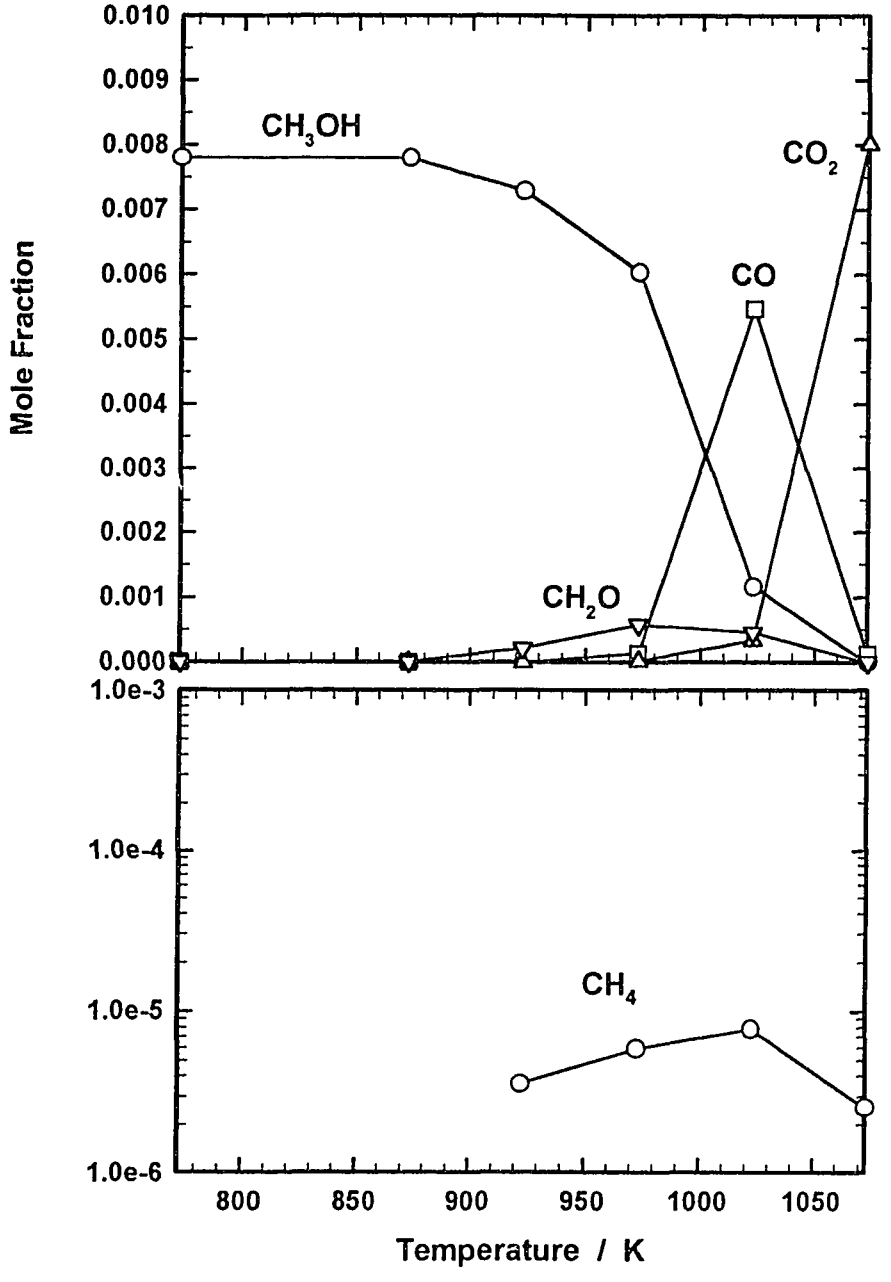


Figure B.45 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.2$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

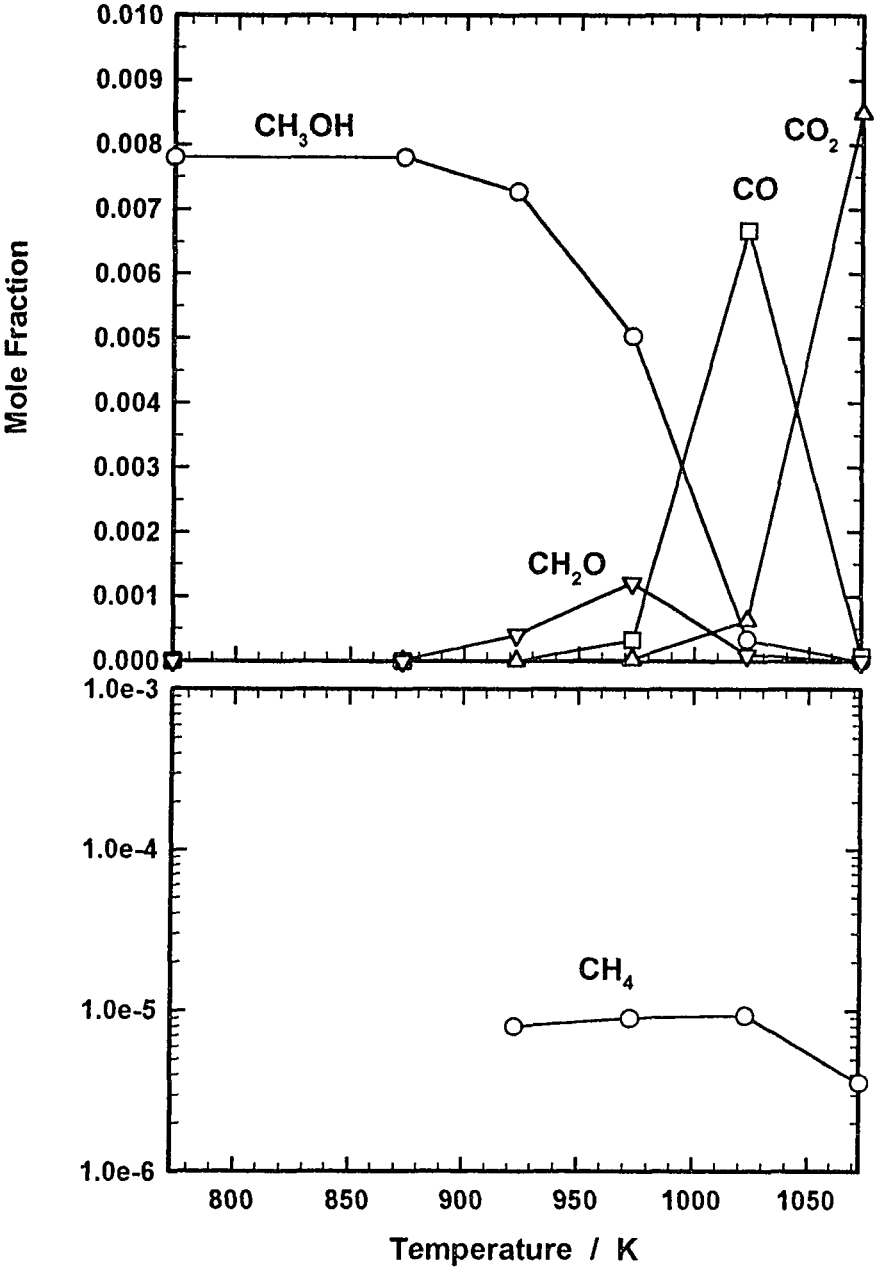


Figure B.46 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.25$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

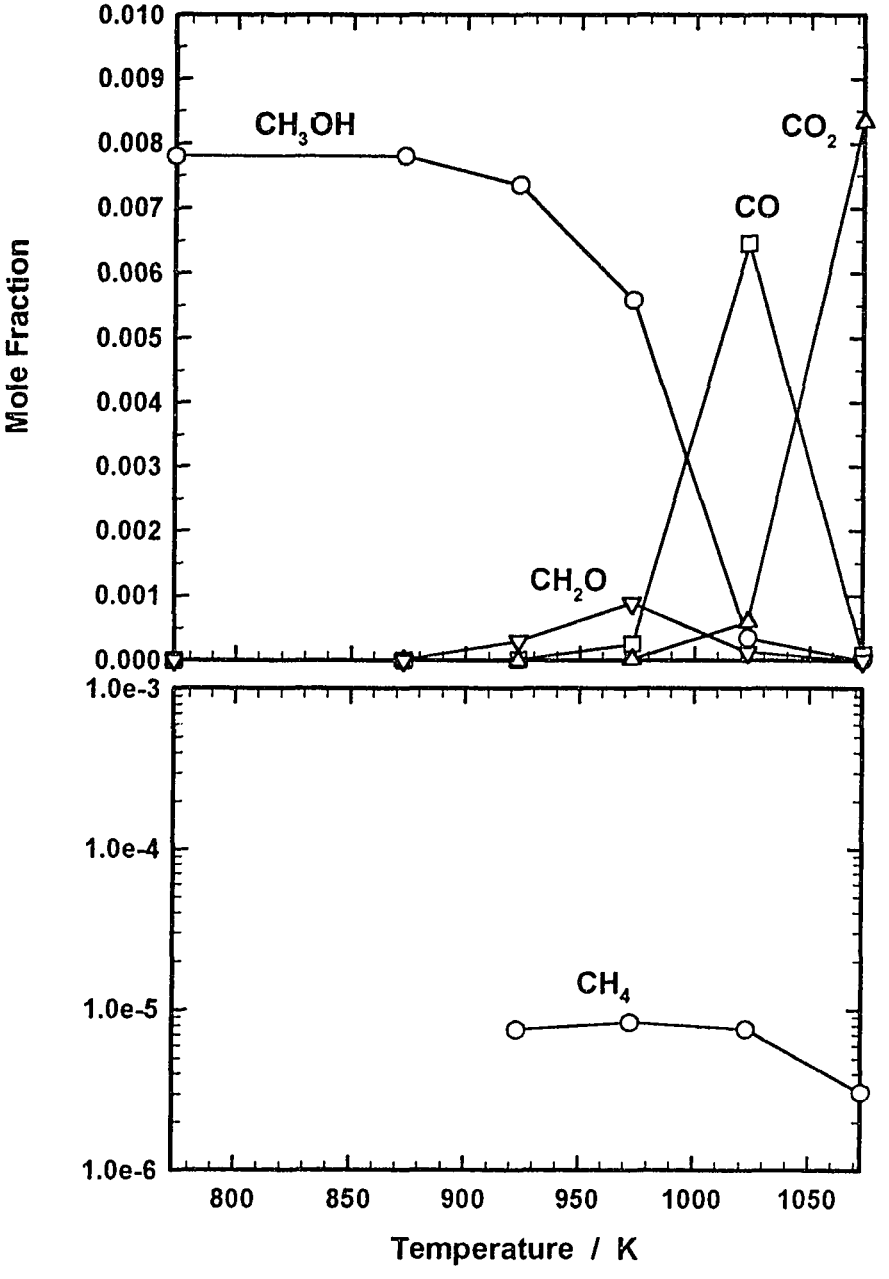


Figure B.47 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.25$ s; $p = 1$ atm; $X_{o,CH3OH} = 0.0078$.

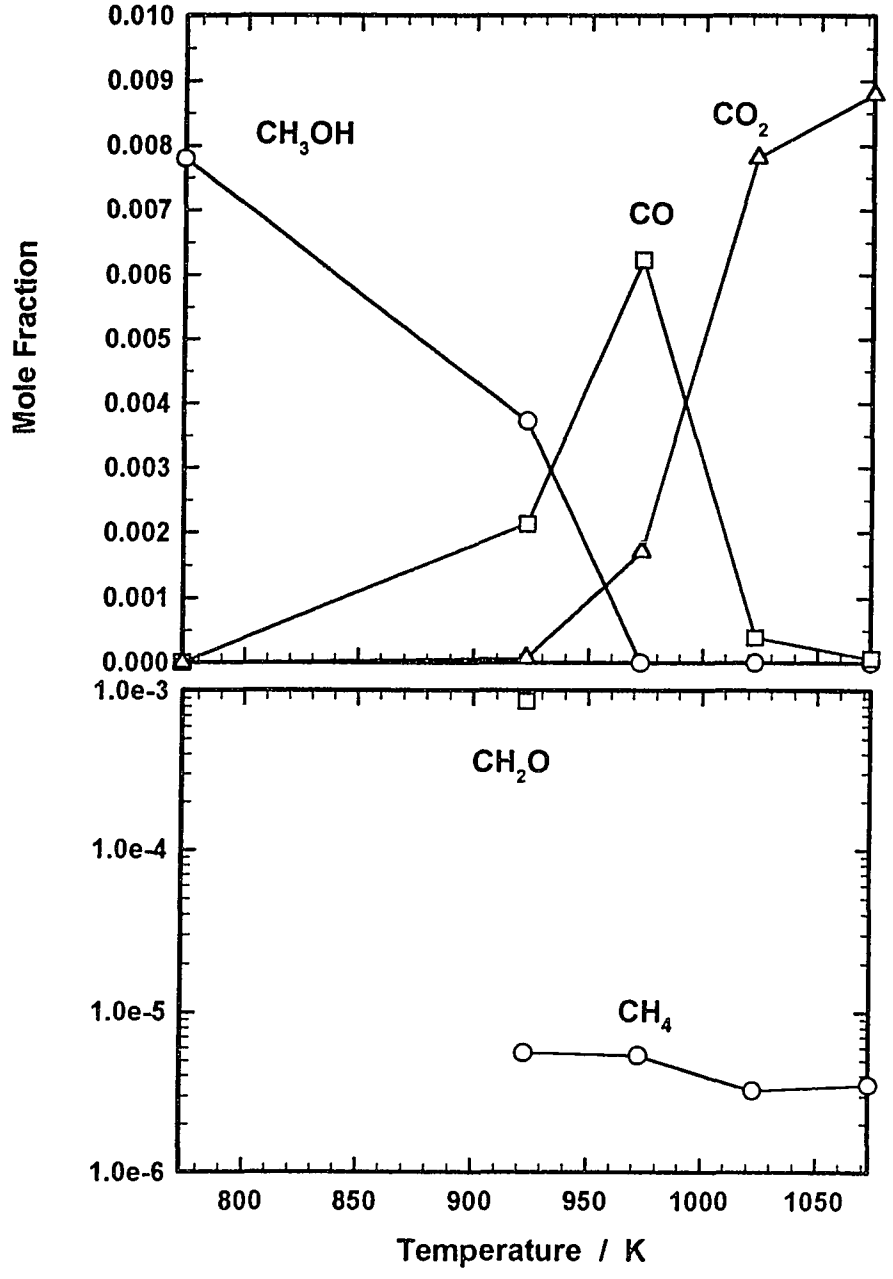


Figure B.48 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.4$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

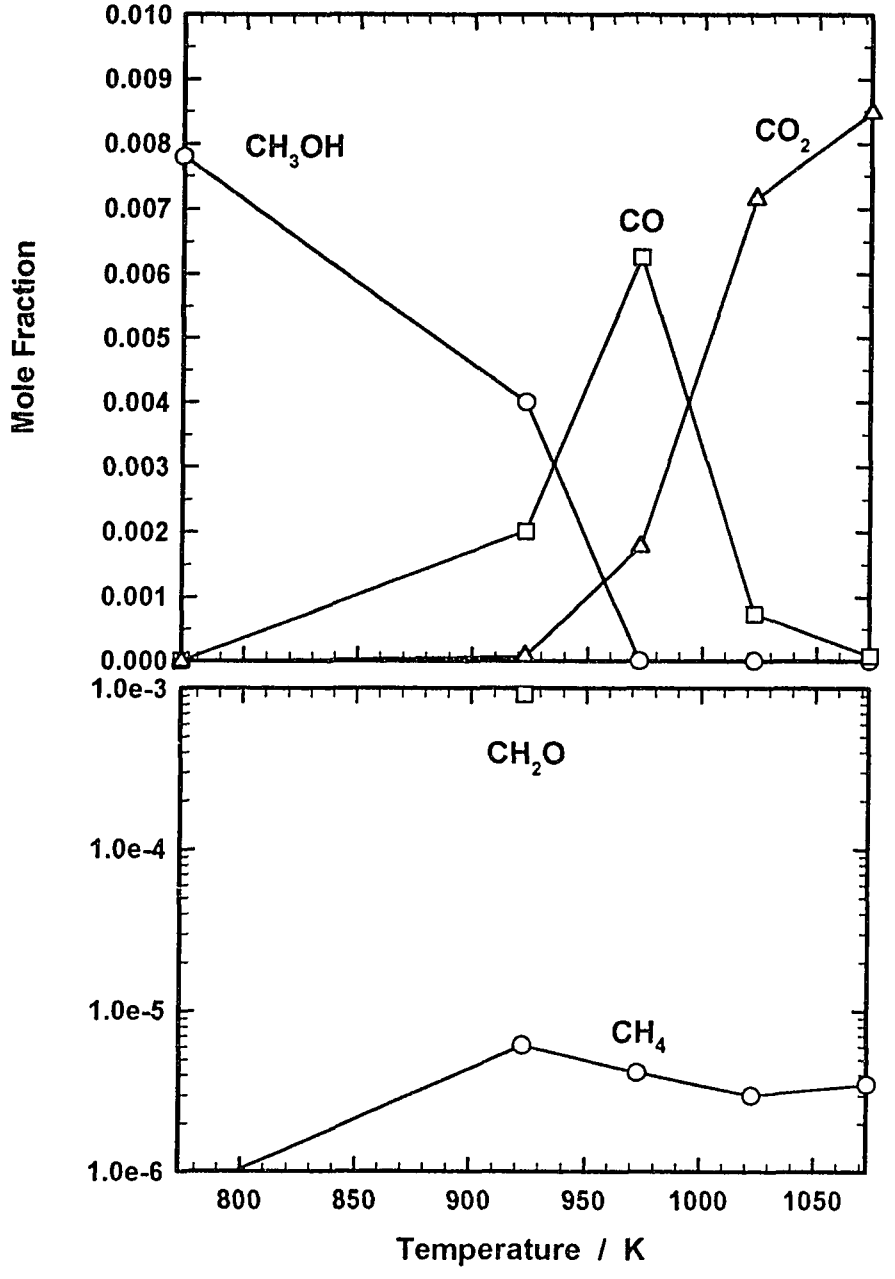


Figure B.49 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.4$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

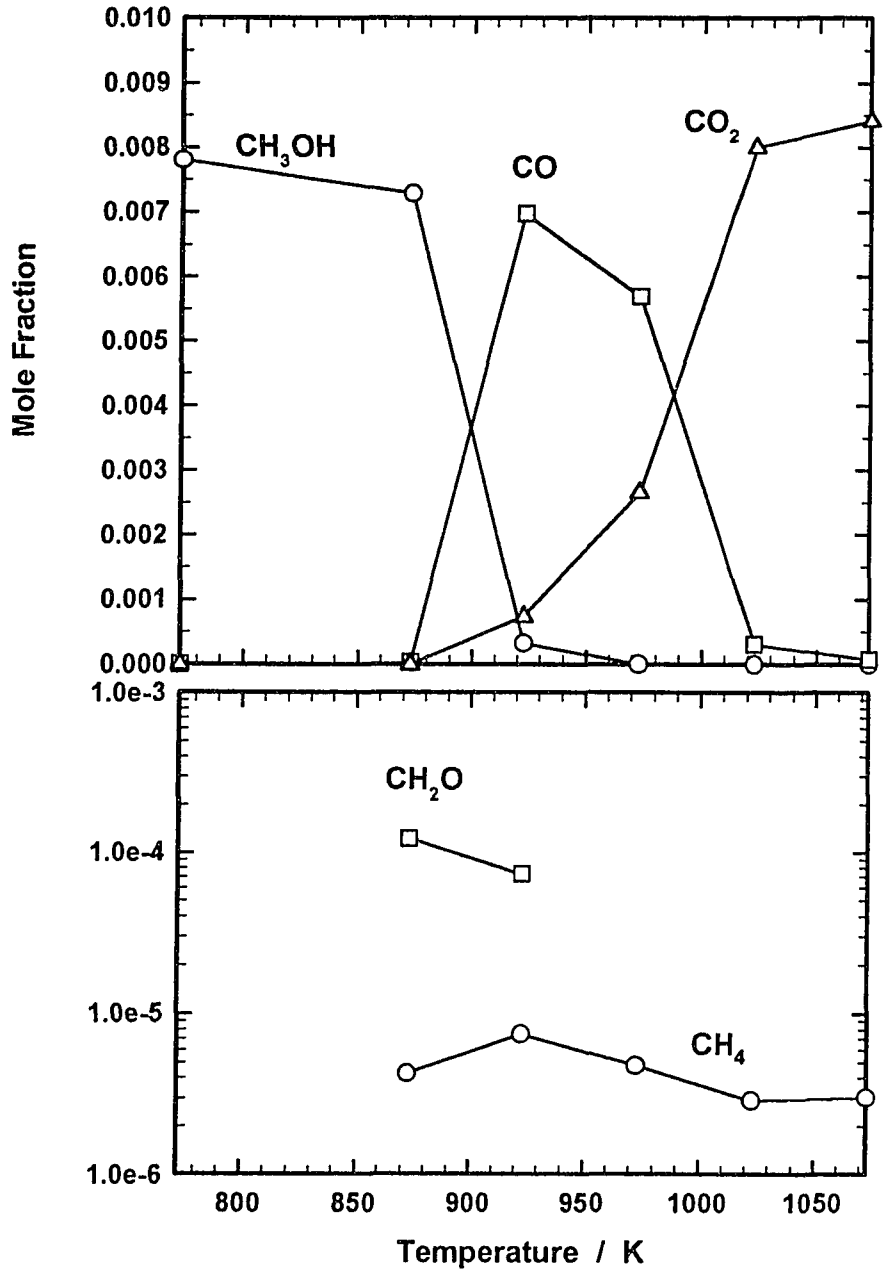


Figure B.50 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.6$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

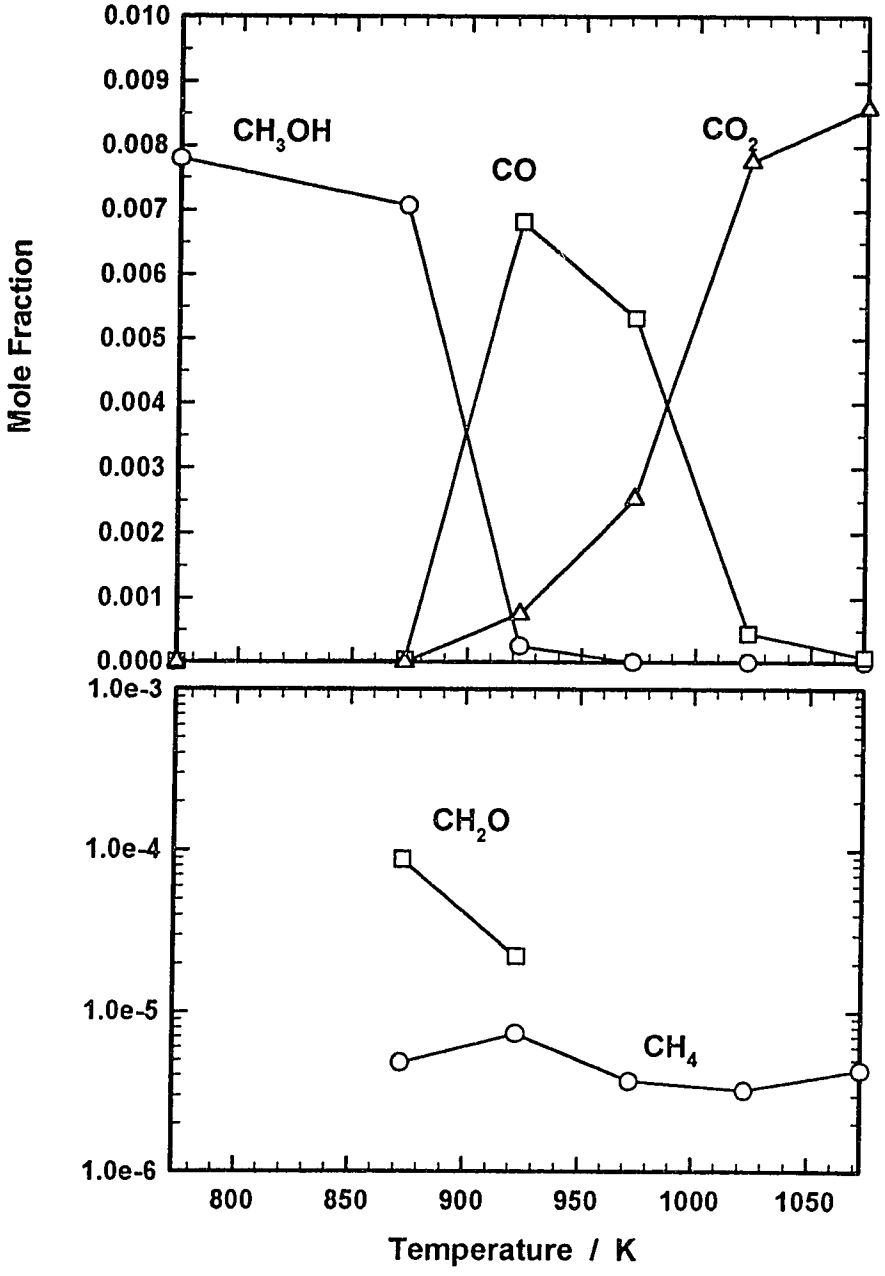


Figure B.51 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.6$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

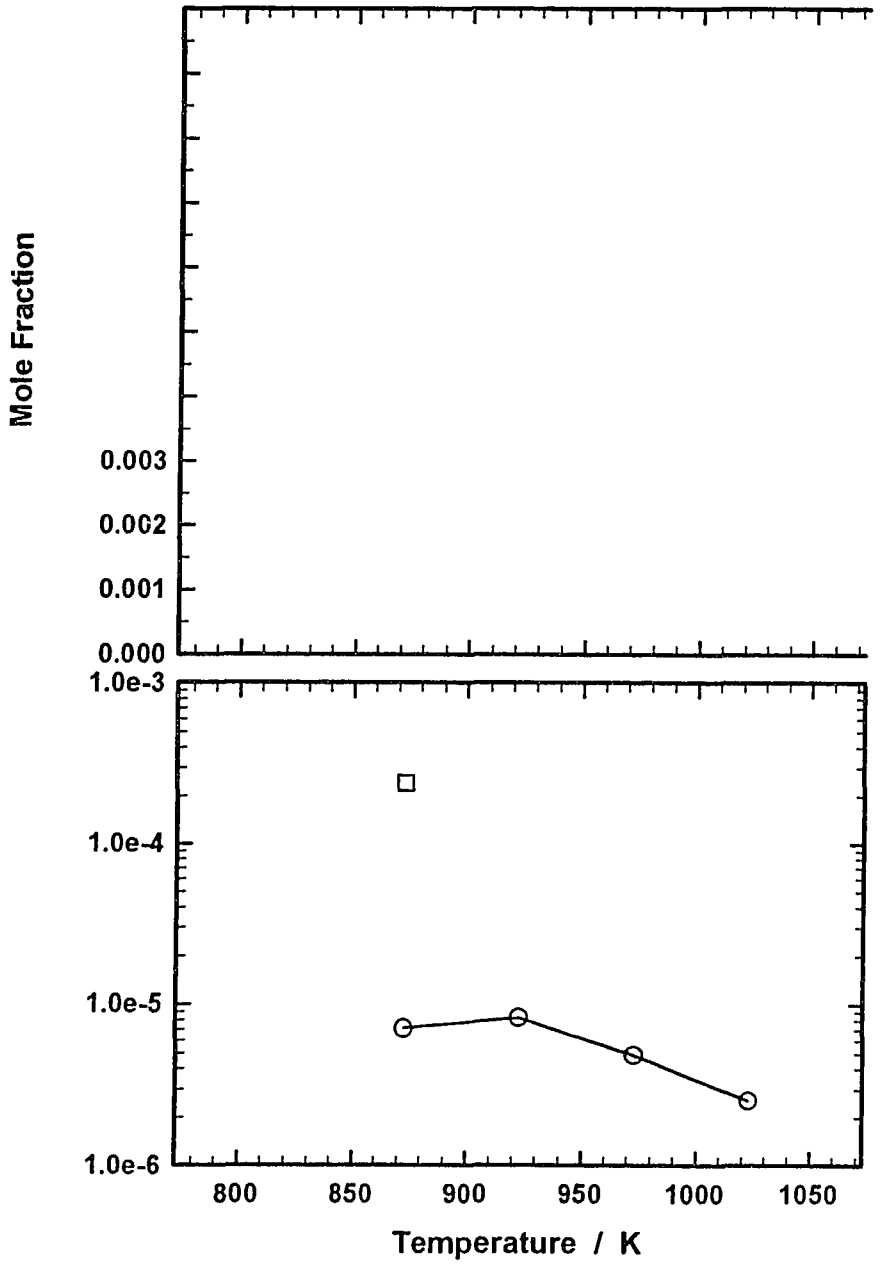


Figure B.52 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.8$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

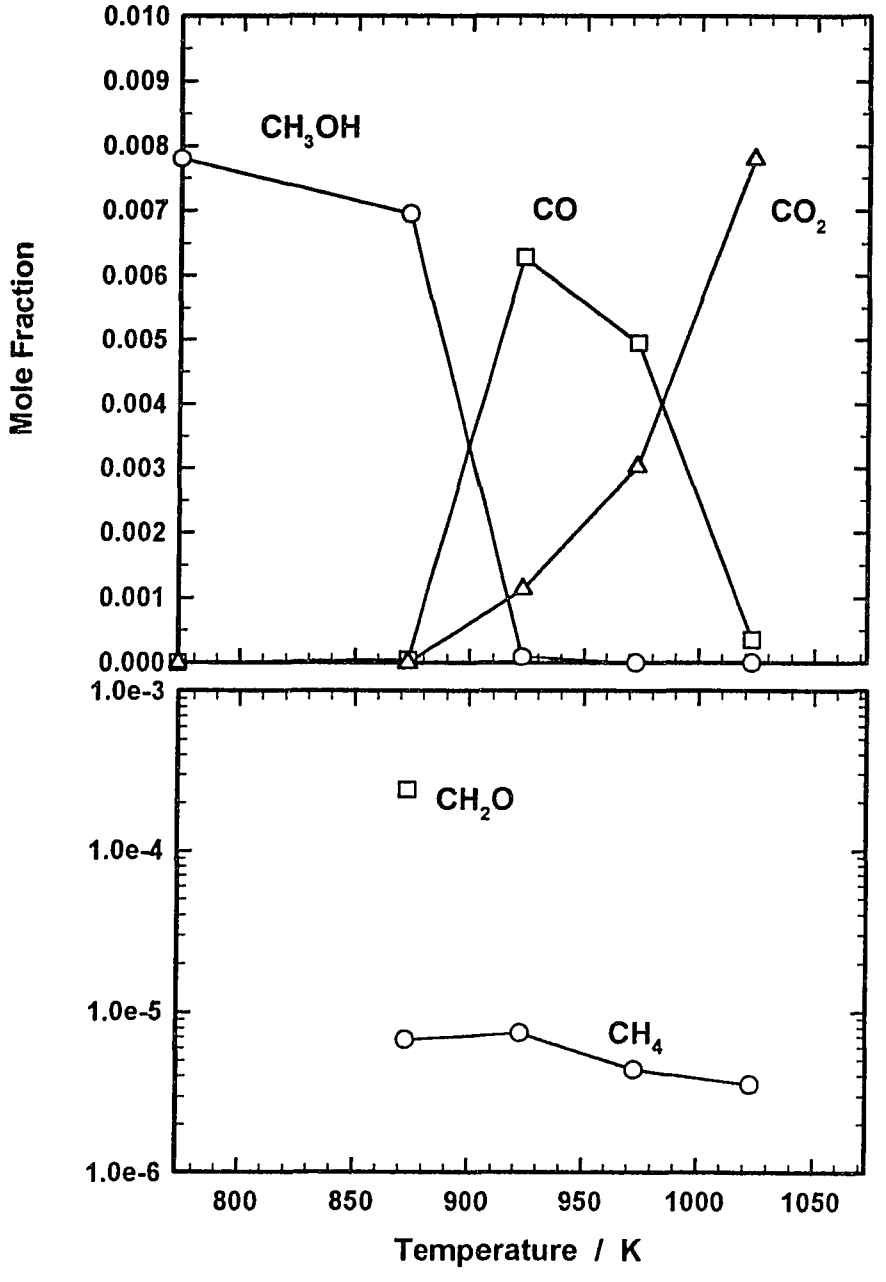


Figure B.53 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.8$ s; $p = 3$ atm; $X_{o,CH3OH} = 0.0078$.

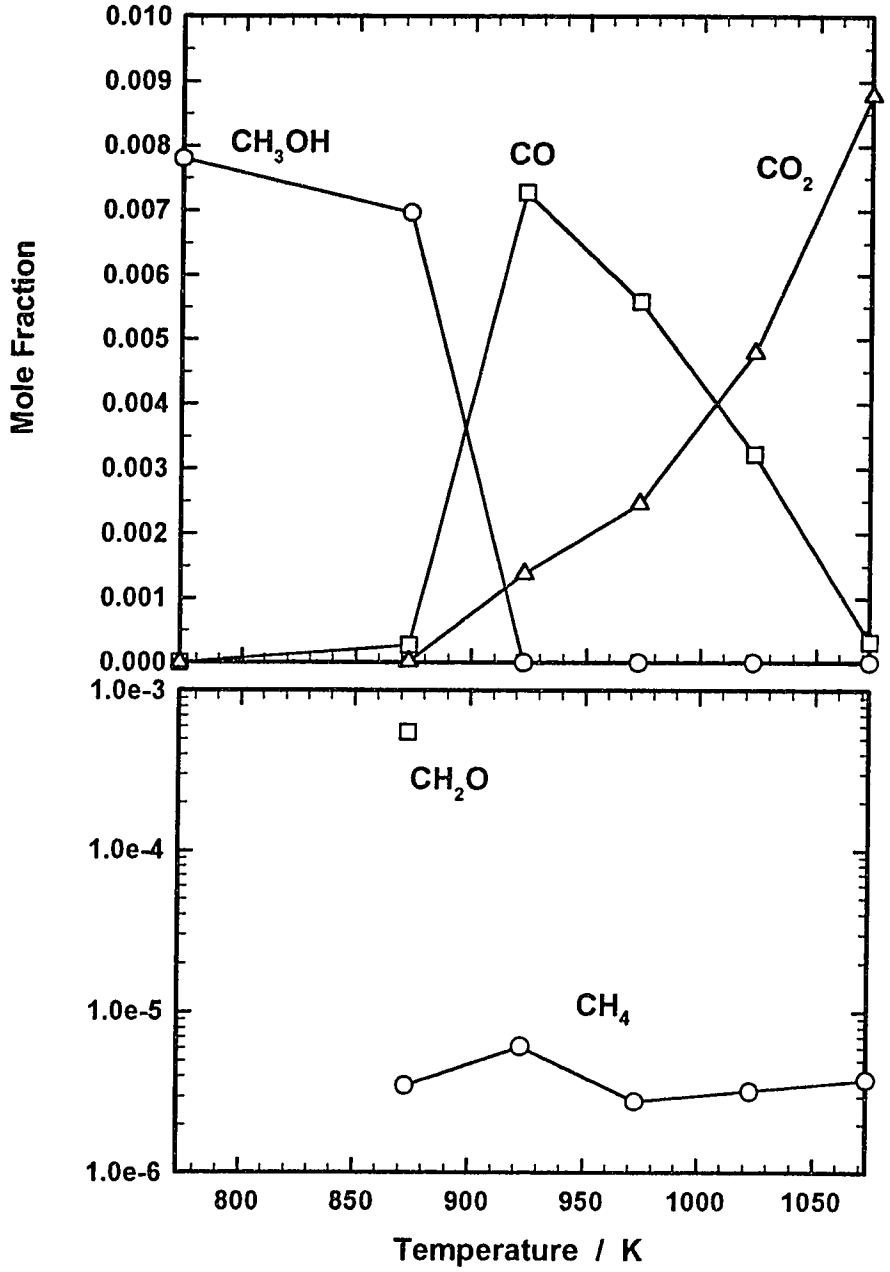


Figure B.54 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.6$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

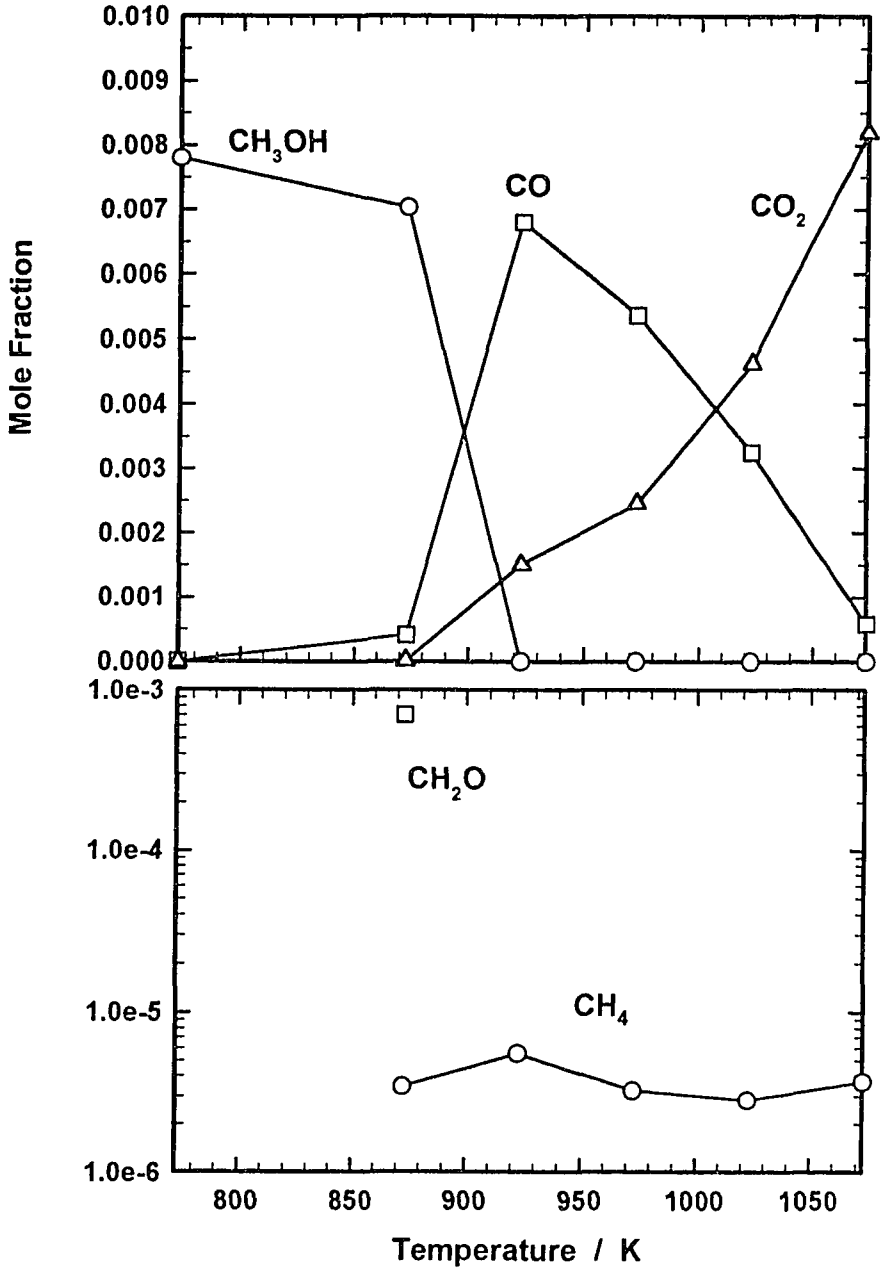


Figure B.55 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.6$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

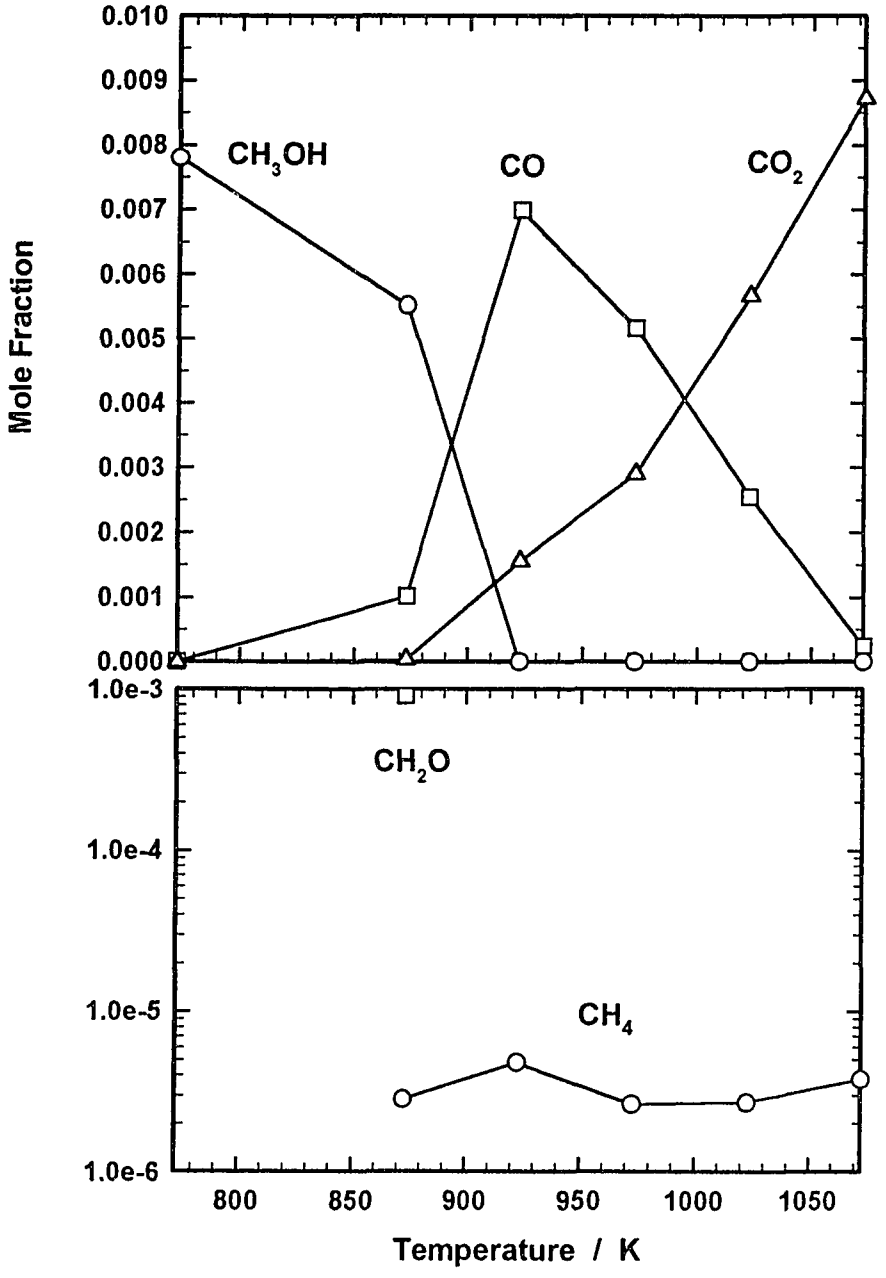


Figure B.56 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 0.8$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

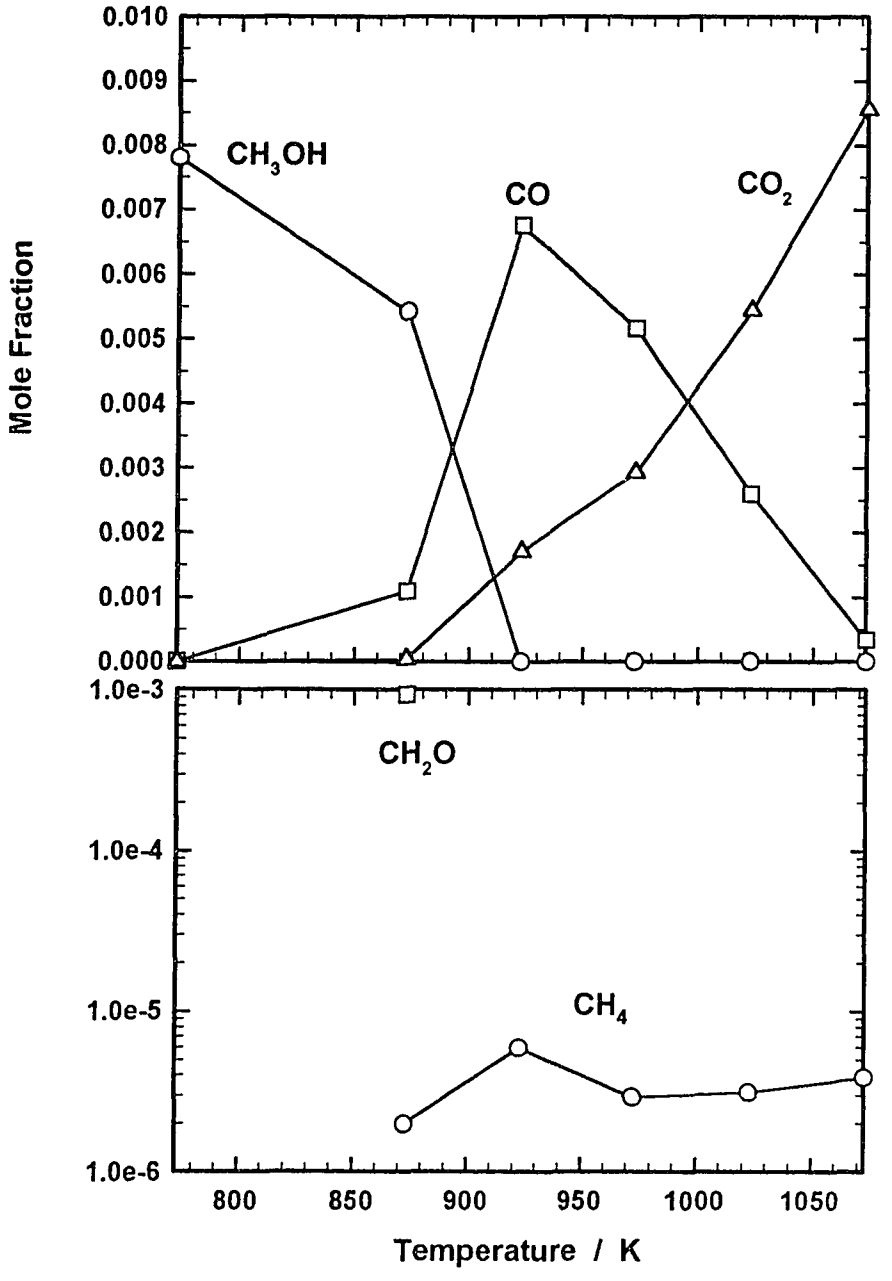


Figure B.57 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 0.8$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

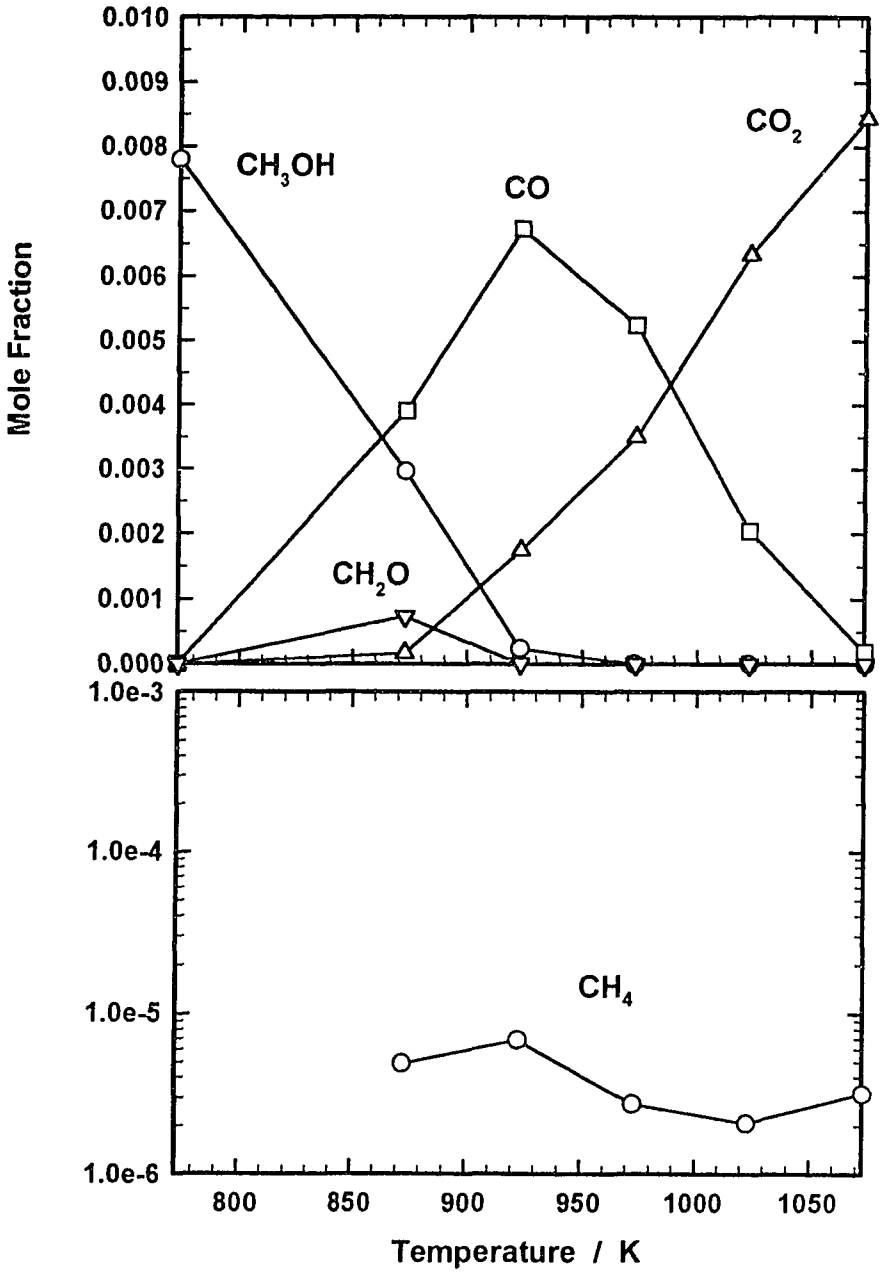


Figure B.58 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 1.0$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

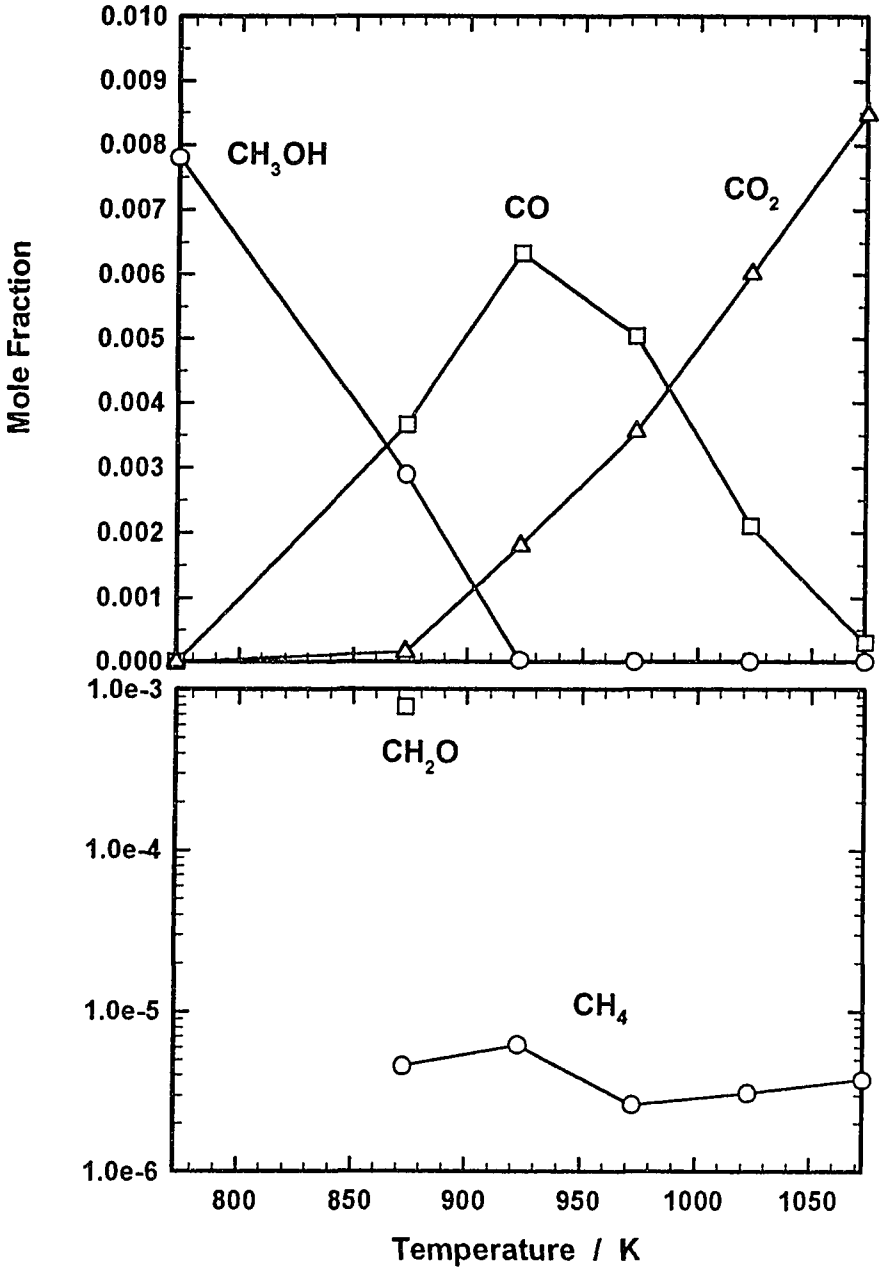


Figure B.59 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 1.0$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

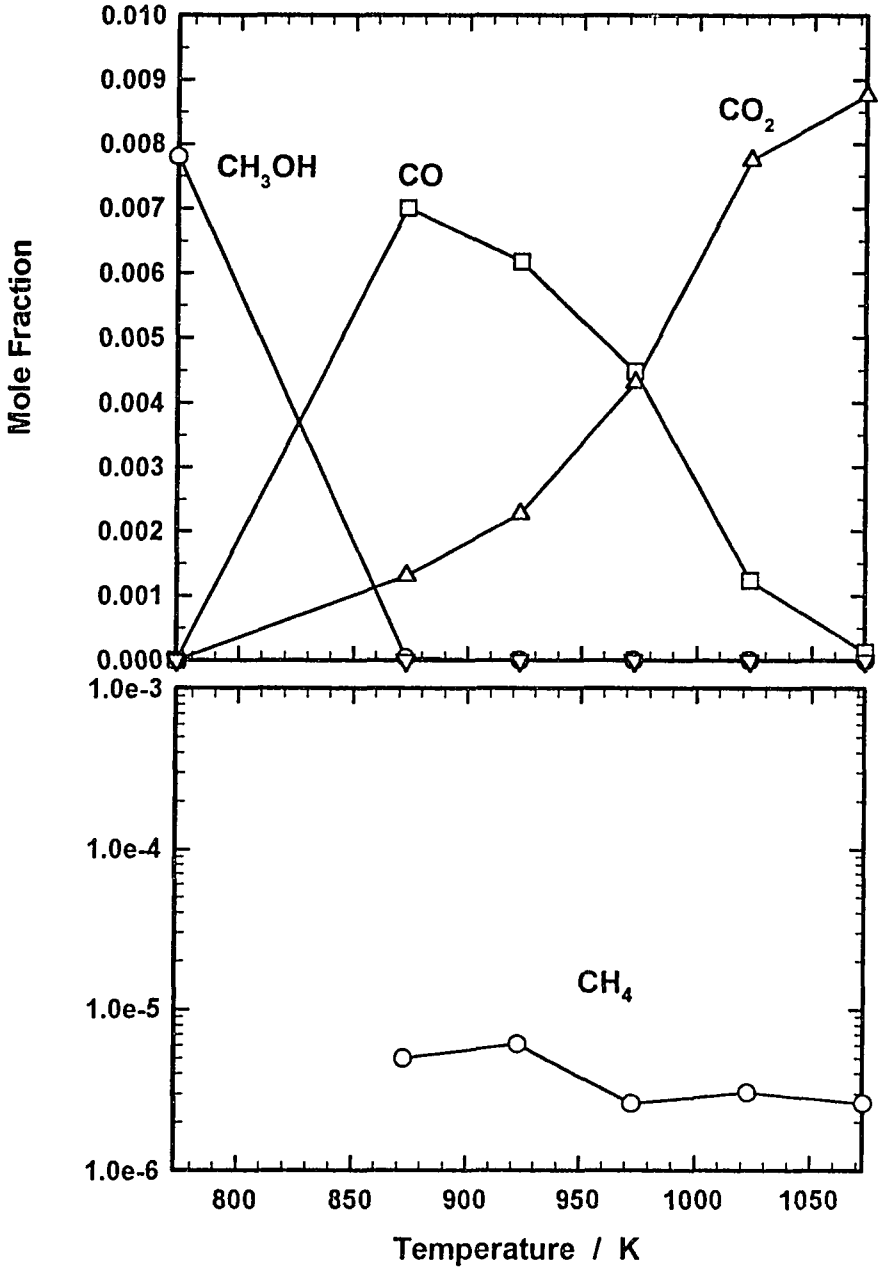


Figure B.60 Experimental result of methanol oxidation: $\varphi = 1.0$; $t = 1.5$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

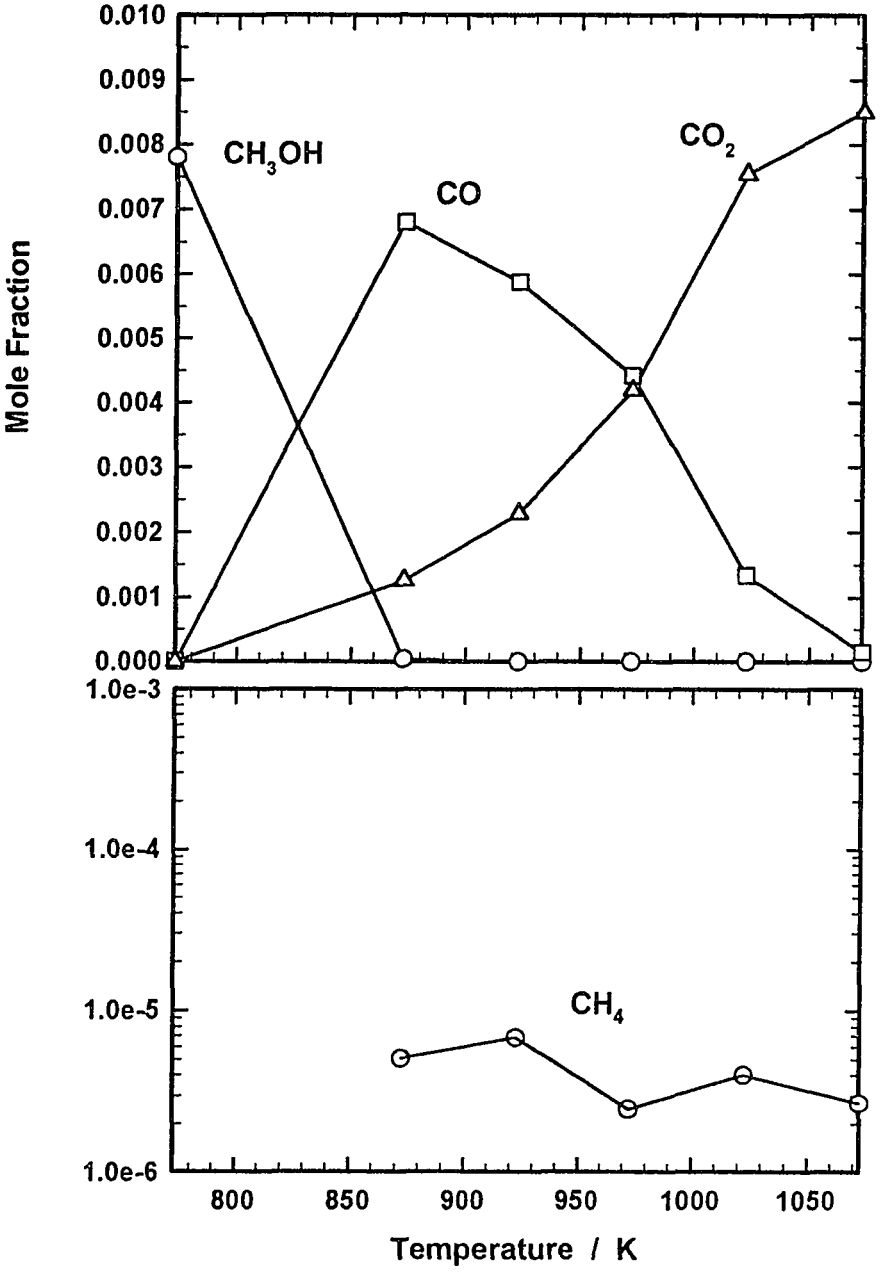


Figure B.61 Experimental result of methanol oxidation: $\varphi = 0.75$; $t = 1.5$ s; $p = 5$ atm; $X_{o,CH3OH} = 0.0078$.

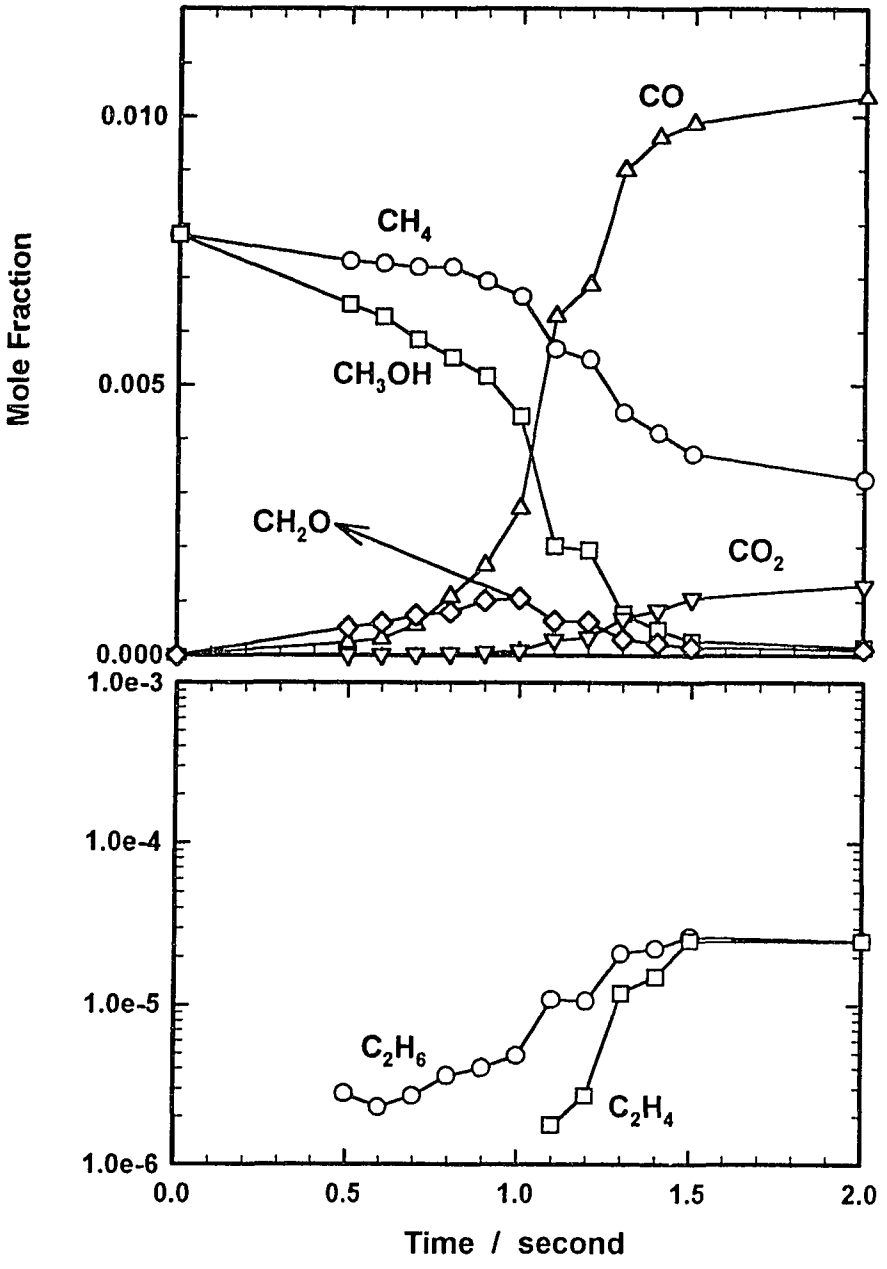


Figure B.62 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.0078$; $X_{o,CH3OH} = 0.0078$.

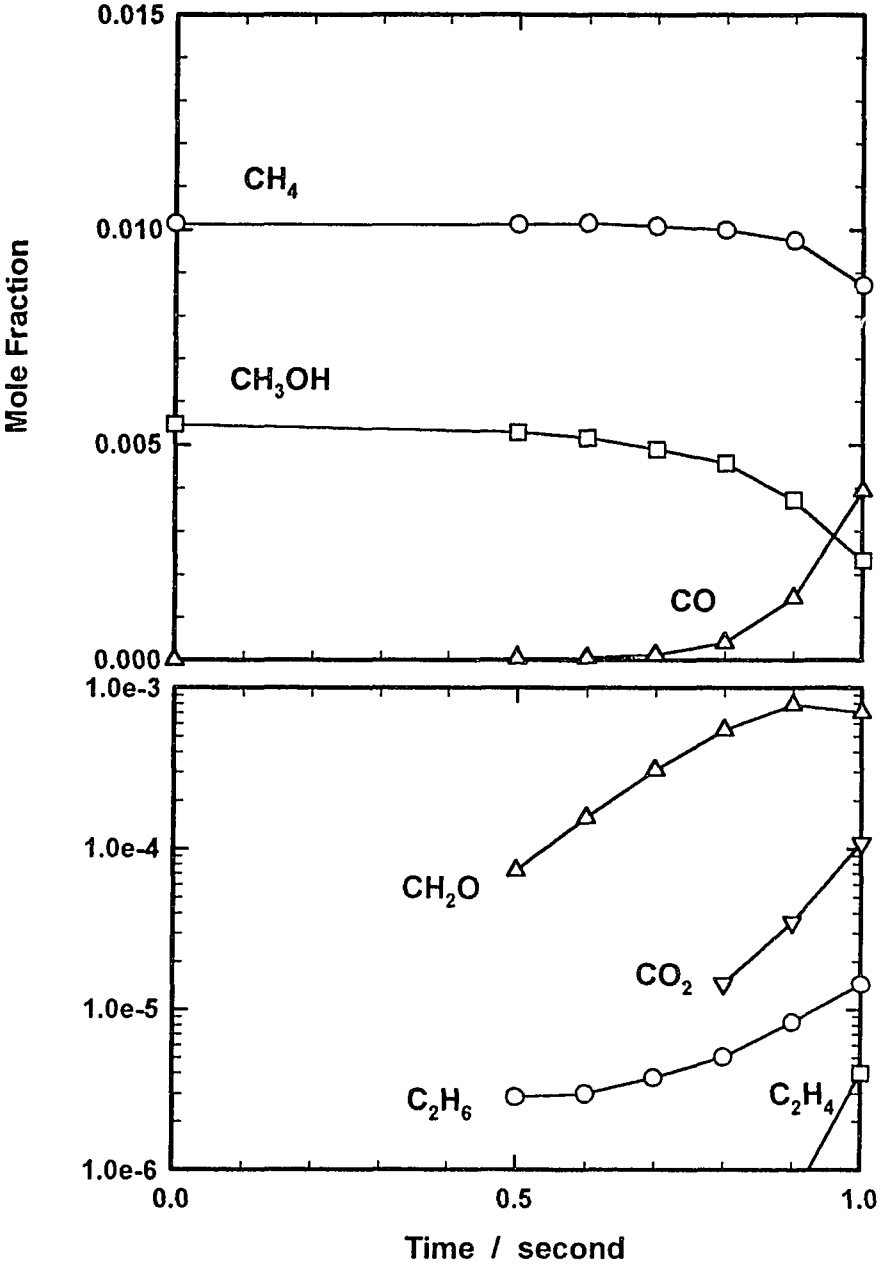


Figure B.63 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.01014$; $X_{o,CH3OH} = 0.00546$.

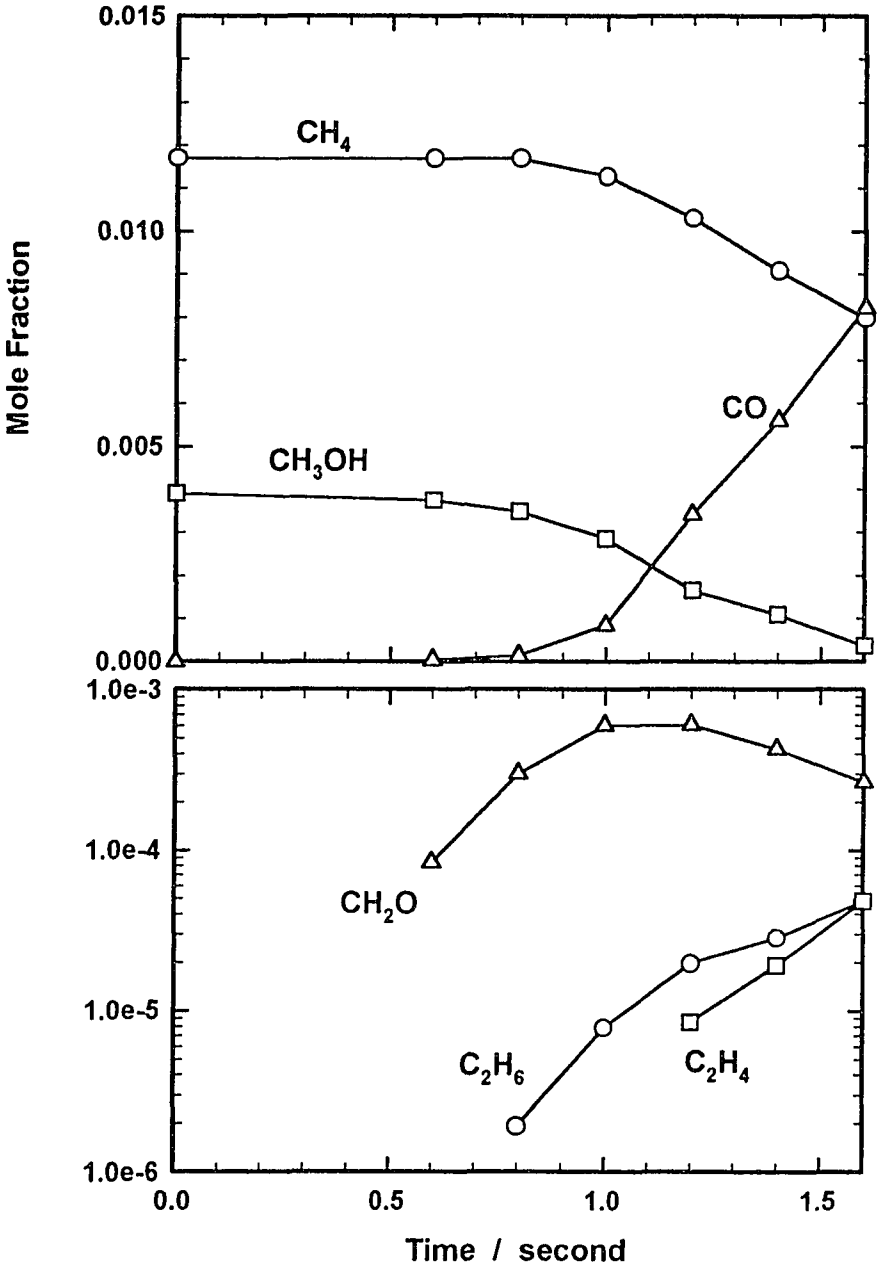


Figure B.64 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.0117$; $X_{o,CH3OH} = 0.0039$.

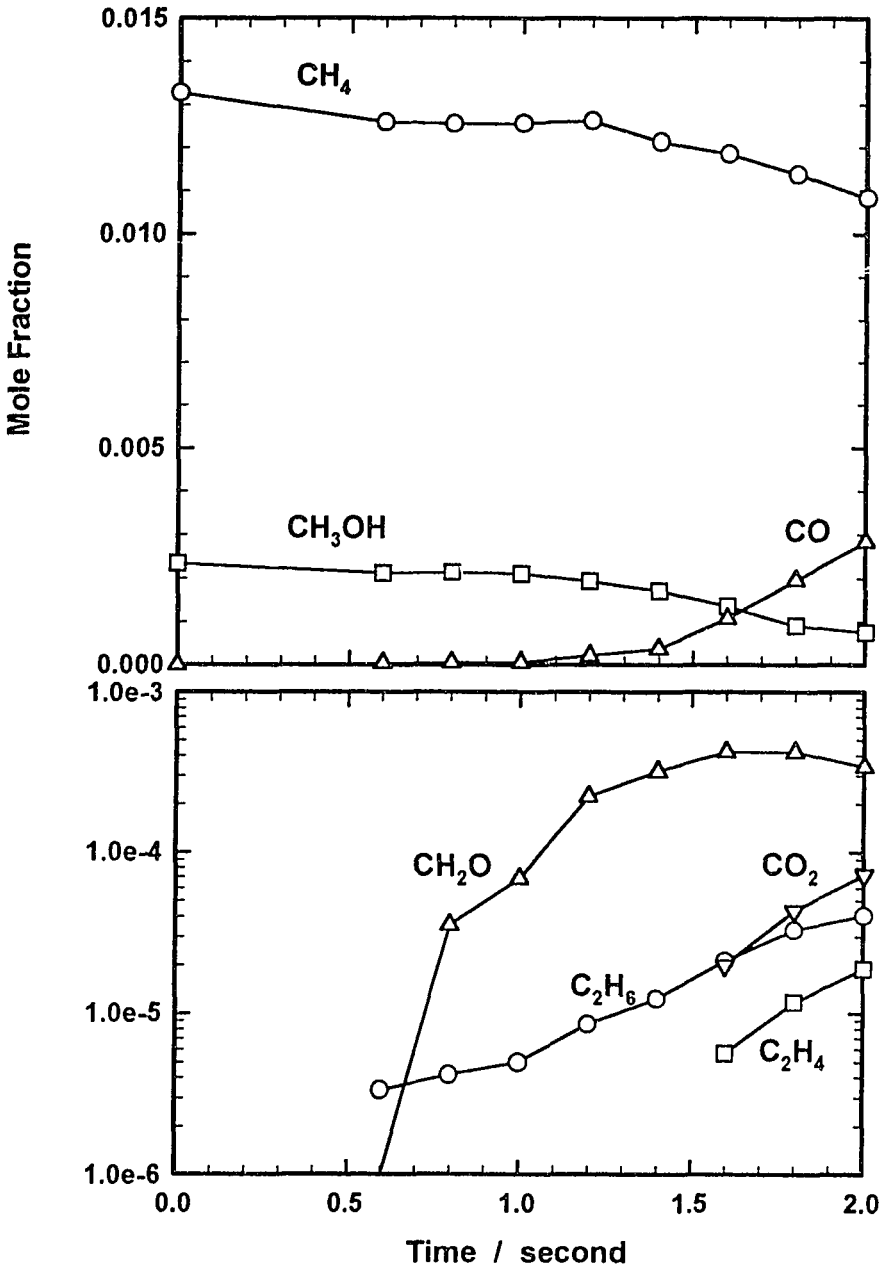


Figure B.65 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.01326$; $X_{o,CH3OH} = 0.00234$.

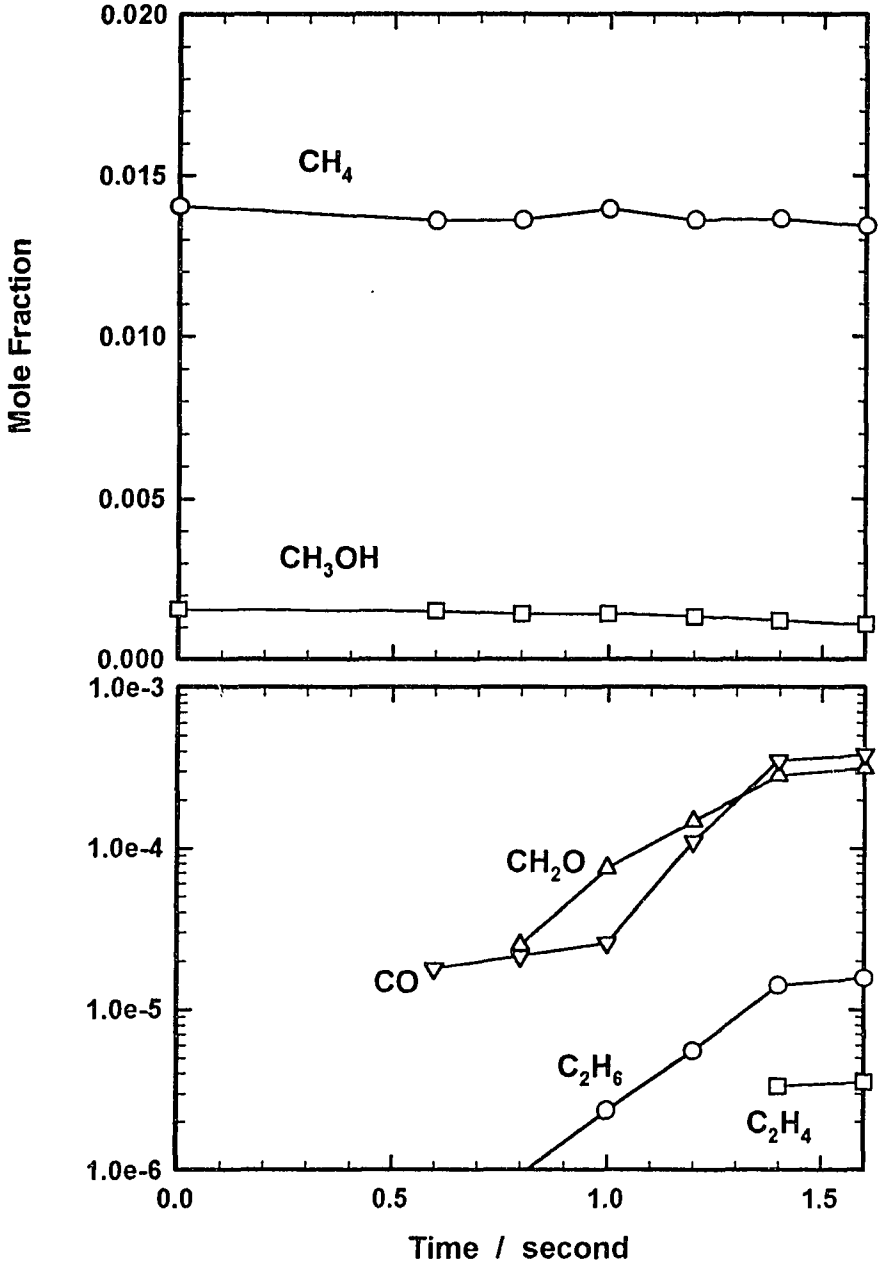


Figure B.66 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.01404$; $X_{o,CH3OH} = 0.00156$.

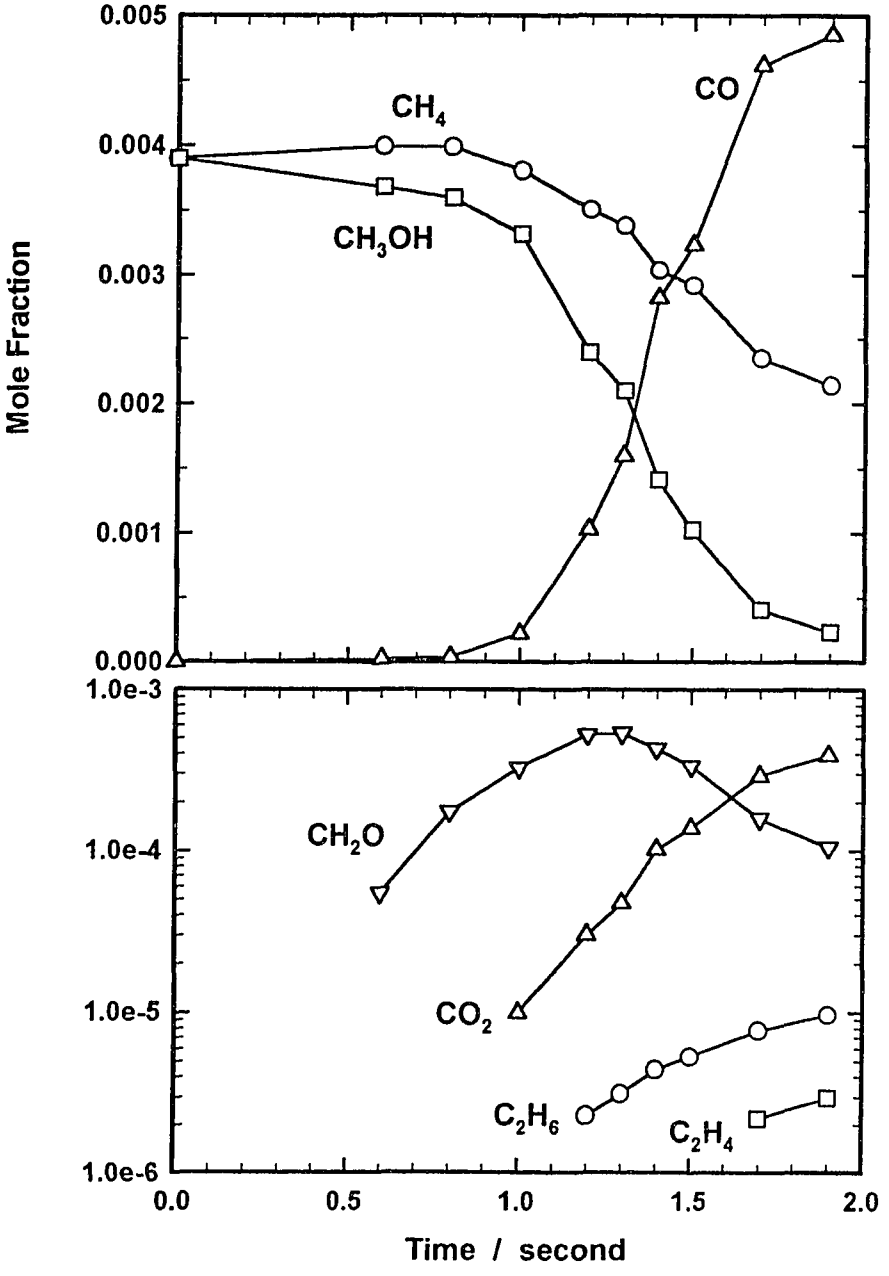


Figure B.67 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.0039$; $X_{o,CH3OH} = 0.0039$.

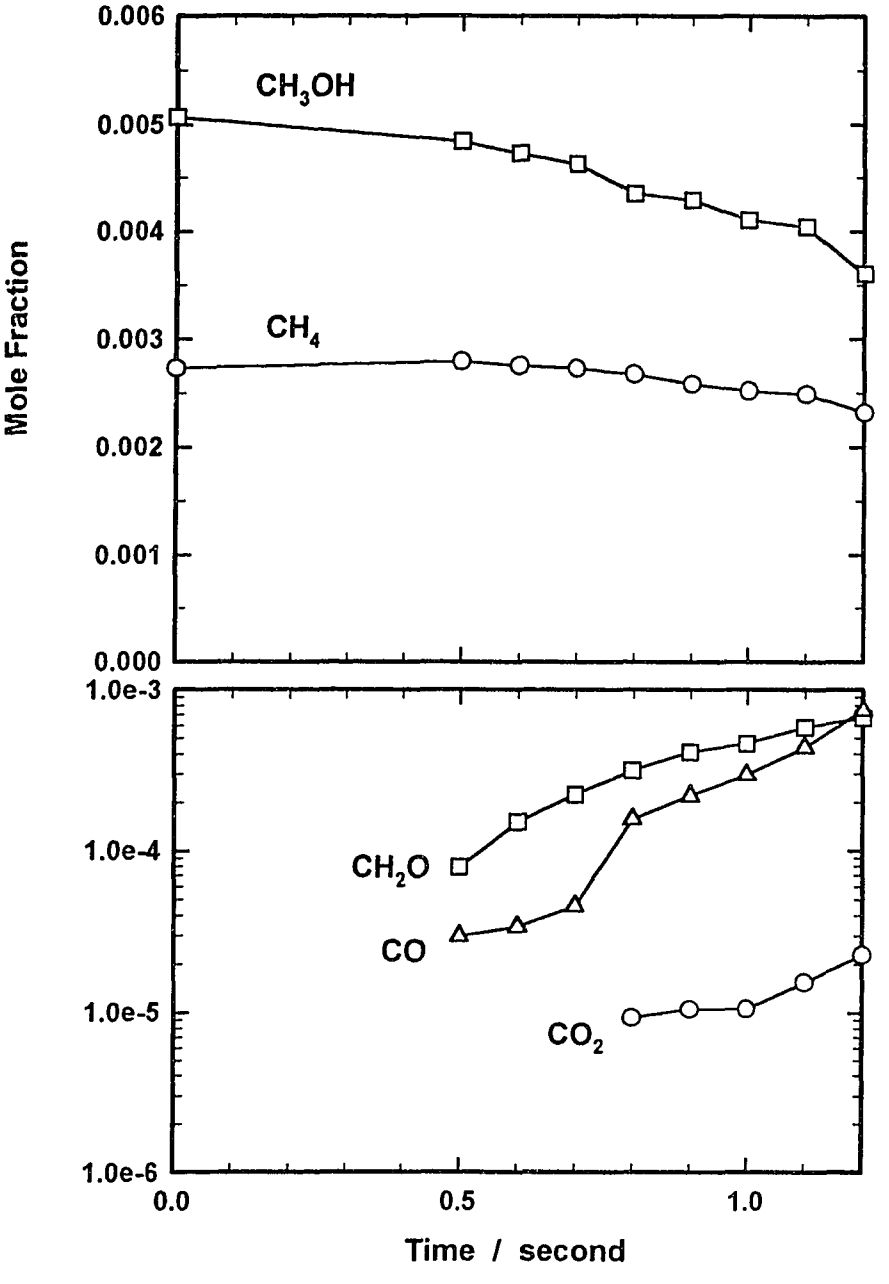


Figure B.68 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.00273$; $X_{o,CH3OH} = 0.00507$.

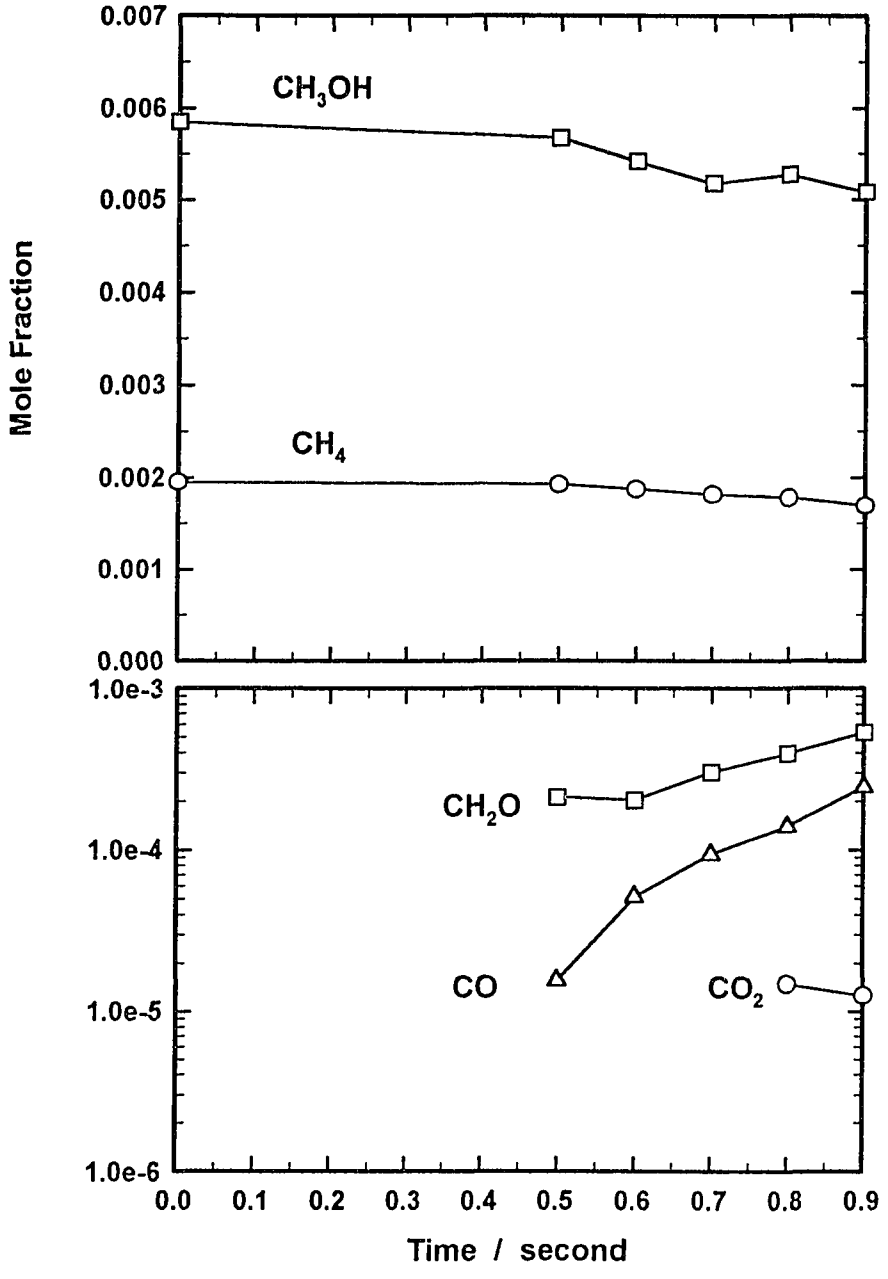


Figure B.69 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 873$ K; $p = 5$ atm; $X_{o,CH4} = 0.00195$; $X_{o,CH3OH} = 0.00585$.

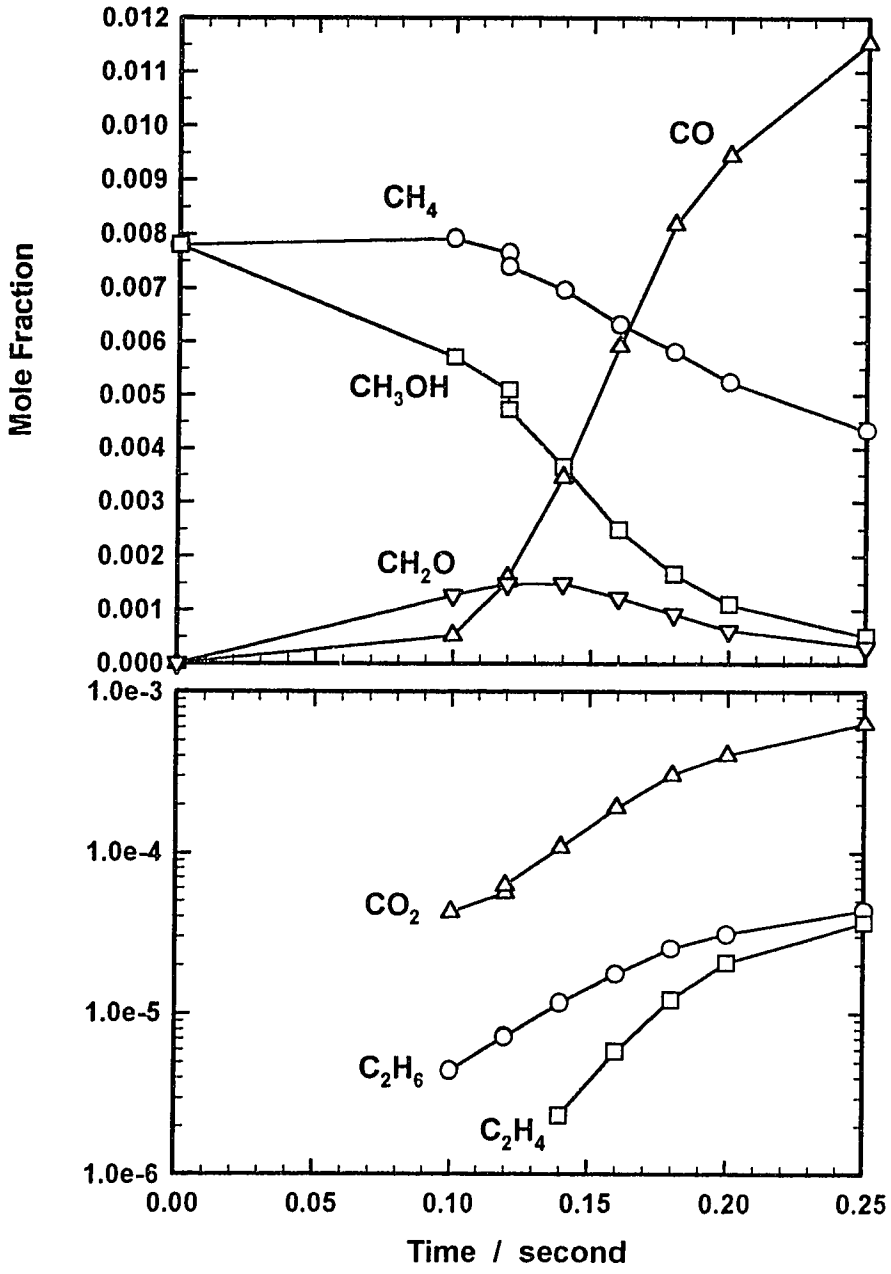


Figure B.70 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 1023$ K; $p = 1$ atm; $X_{o,CH4} = 0.0078$; $X_{o,CH3OH} = 0.0078$.

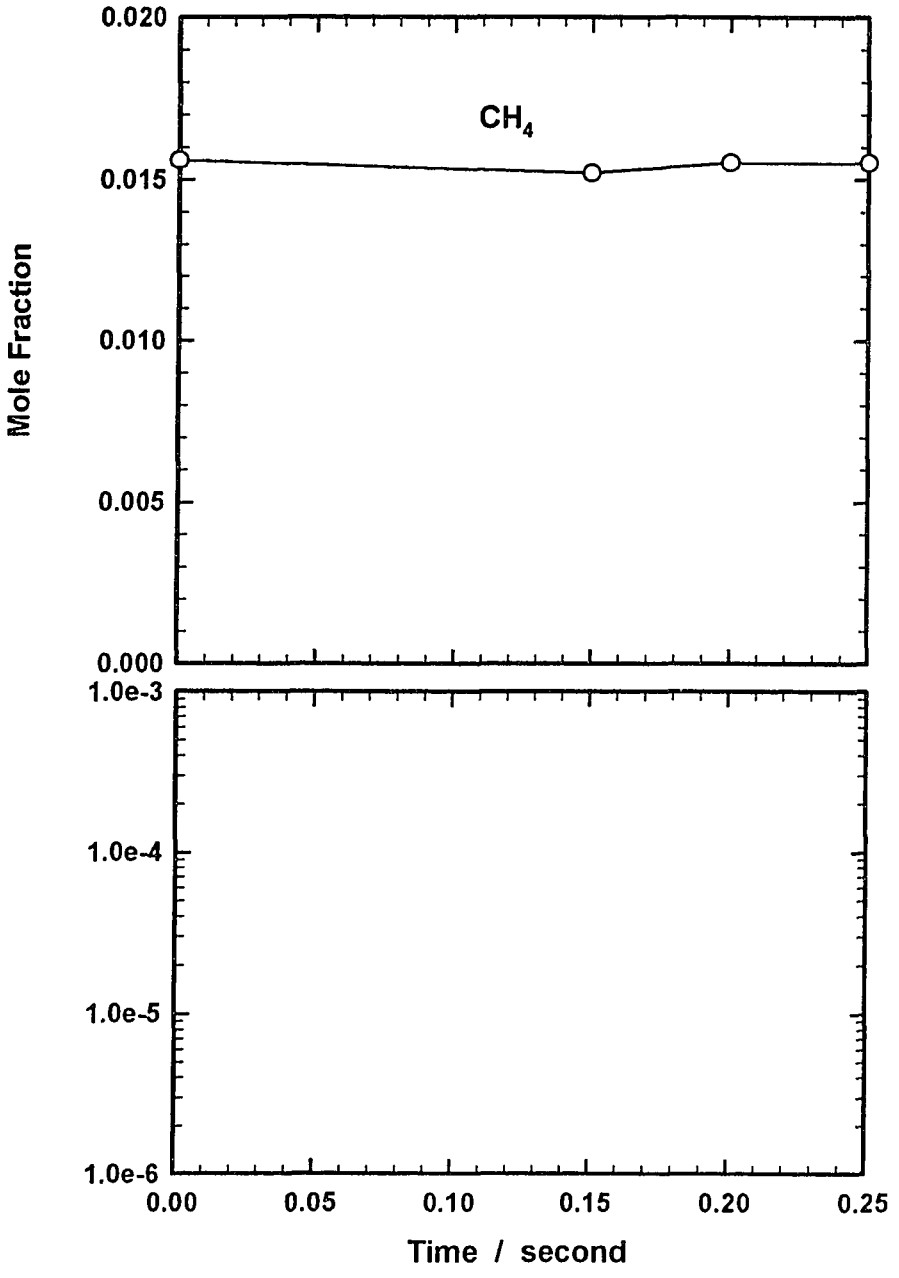


Figure B.71 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 1$ atm; $X_{o,CH4} = 0.0156$; $X_{o,CH3OH} = 0.0$.

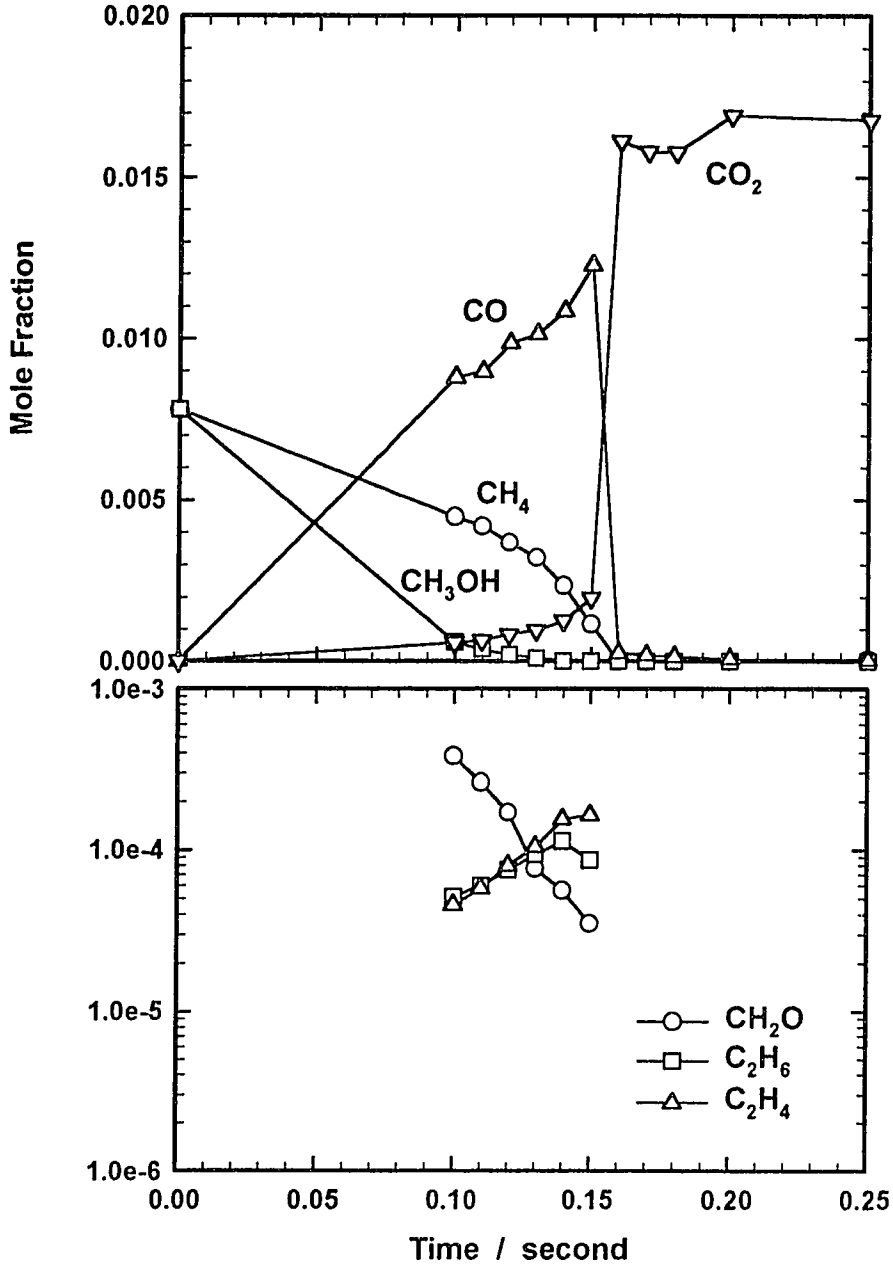


Figure B.72 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 1$ atm; $X_{o,CH4} = 0.0078$; $X_{o,CH3OH} = 0.0078$.

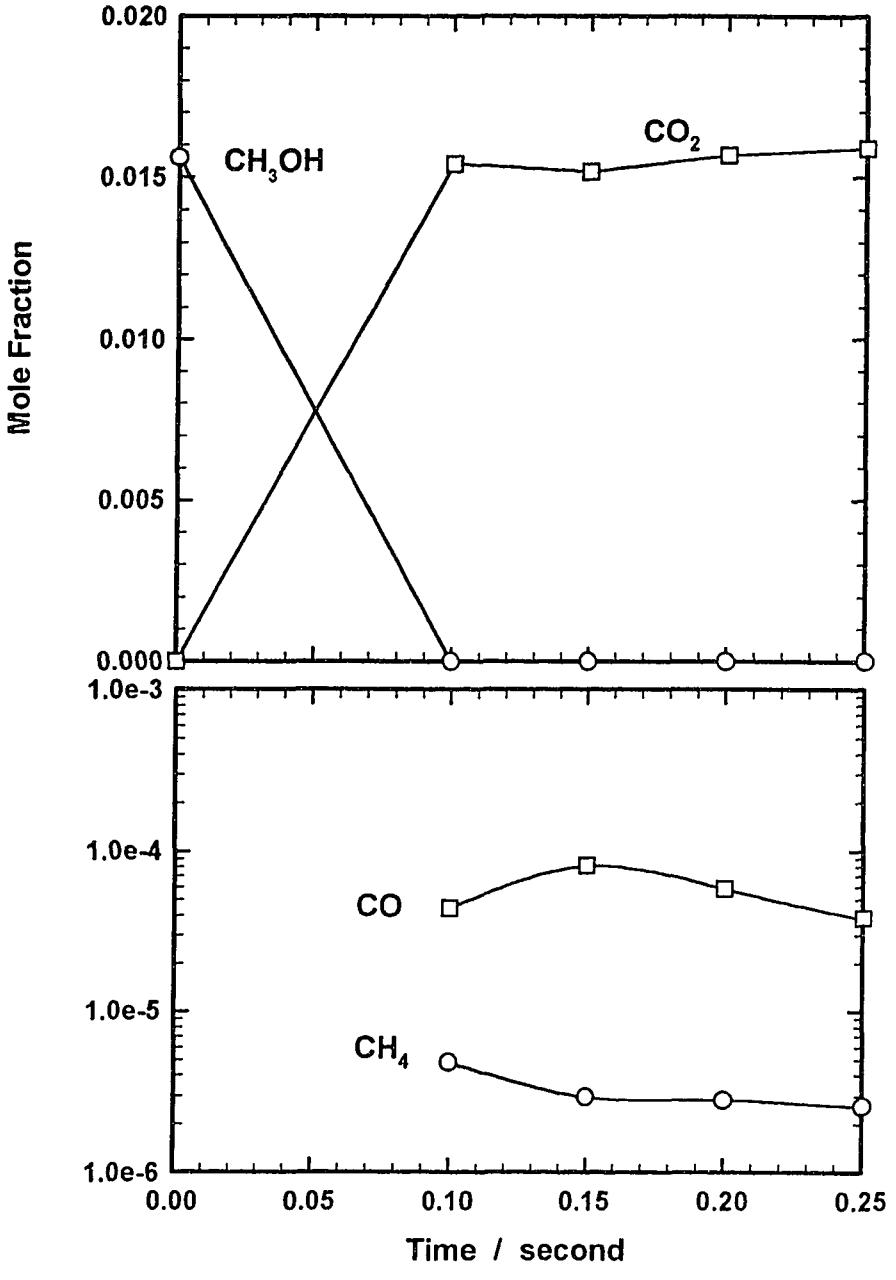


Figure B.73 Experimental result of methane / methanol oxidation: $\varphi = 1.0$; $T = 1073$ K; $p = 1$ atm; $X_{o,CH4} = 0.0$; $X_{o,CH3OH} = 0.0156$.

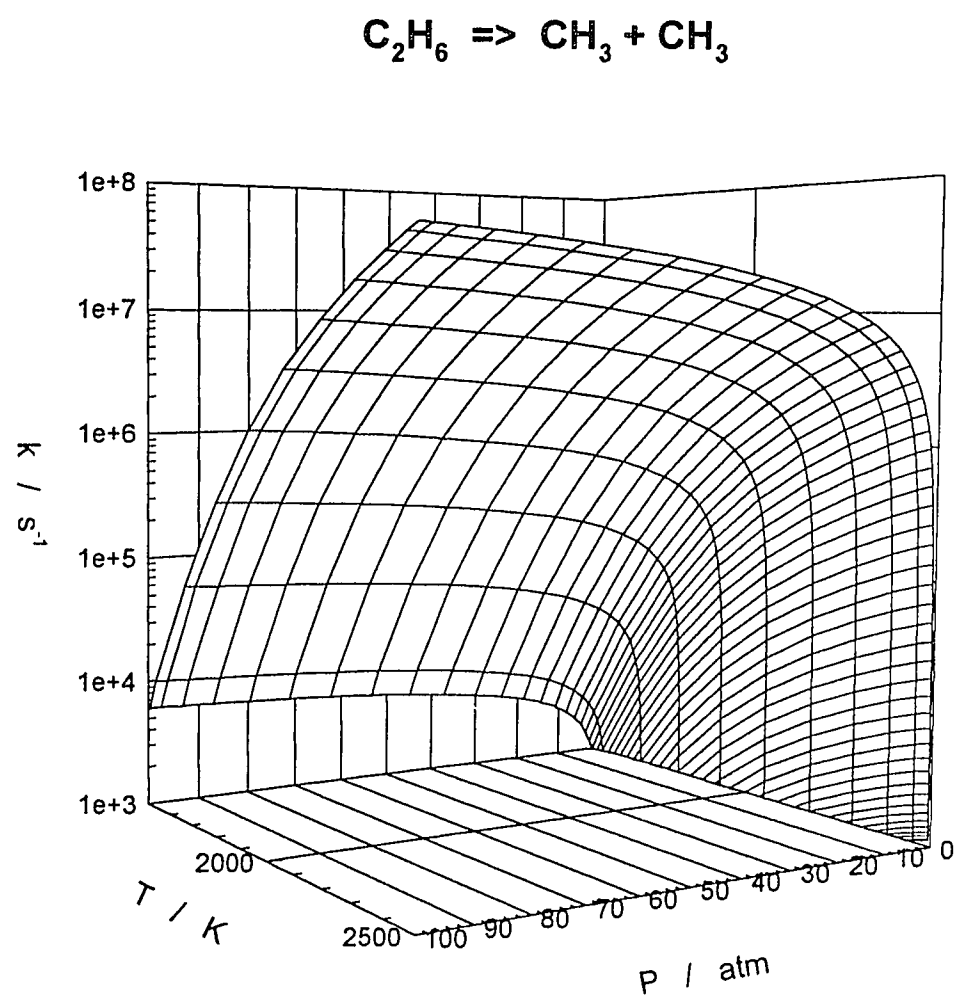


Figure B.74 Rate coefficients for $C_2H_6 \rightarrow CH_3 + CH_3$

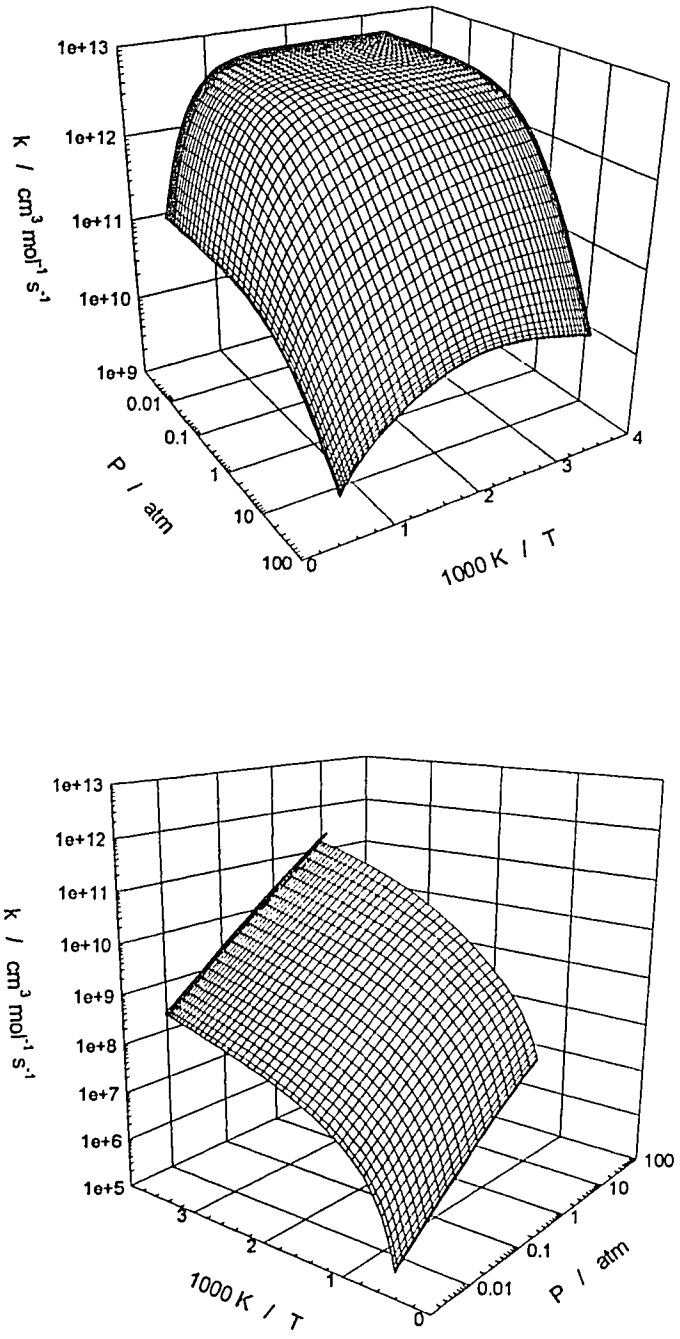


Figure B.75 Rate coefficients for (a) $CH_2OH + O_2 \rightarrow CH_2O + HO_2$ (b) $CH_2OH + O_2 \rightarrow HOC(OO)H_2$

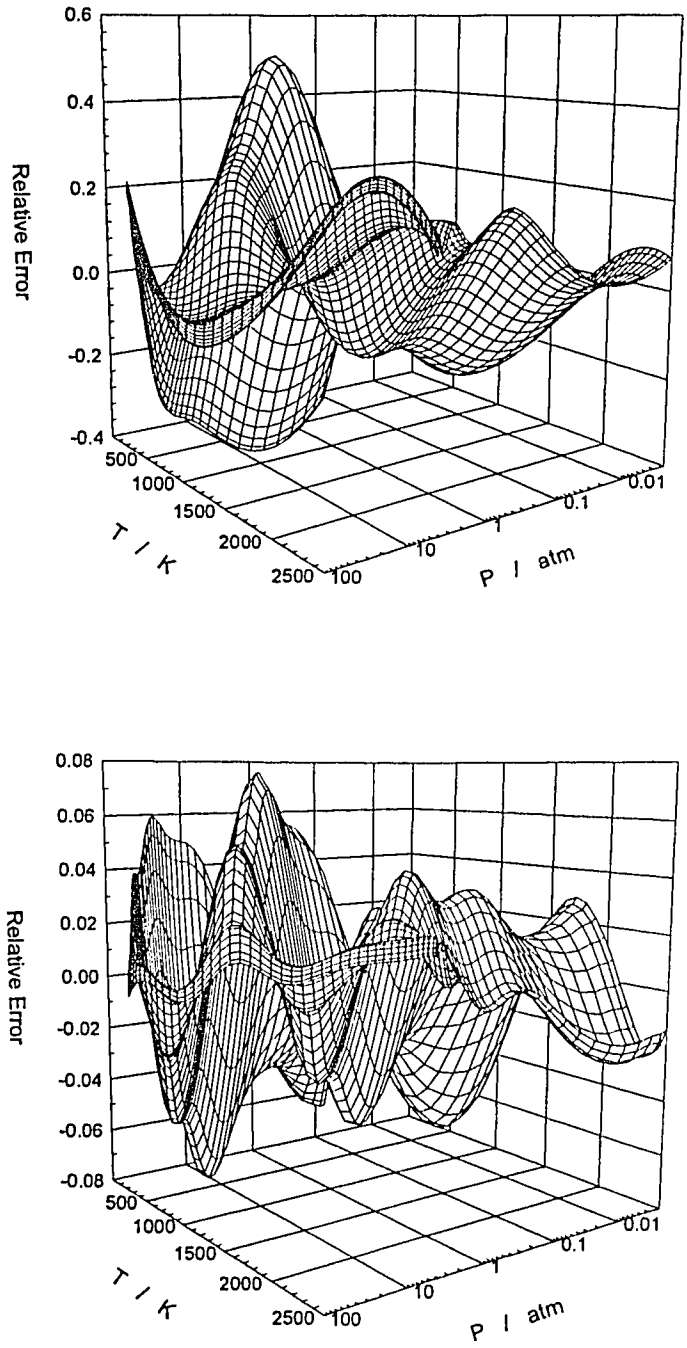


Figure B.76 Relative errors for $CH_2OH + O_2 \rightarrow HOC(OO)H_2$
(a) Chebyshev 7×3 (b) Chebyshev 9×5

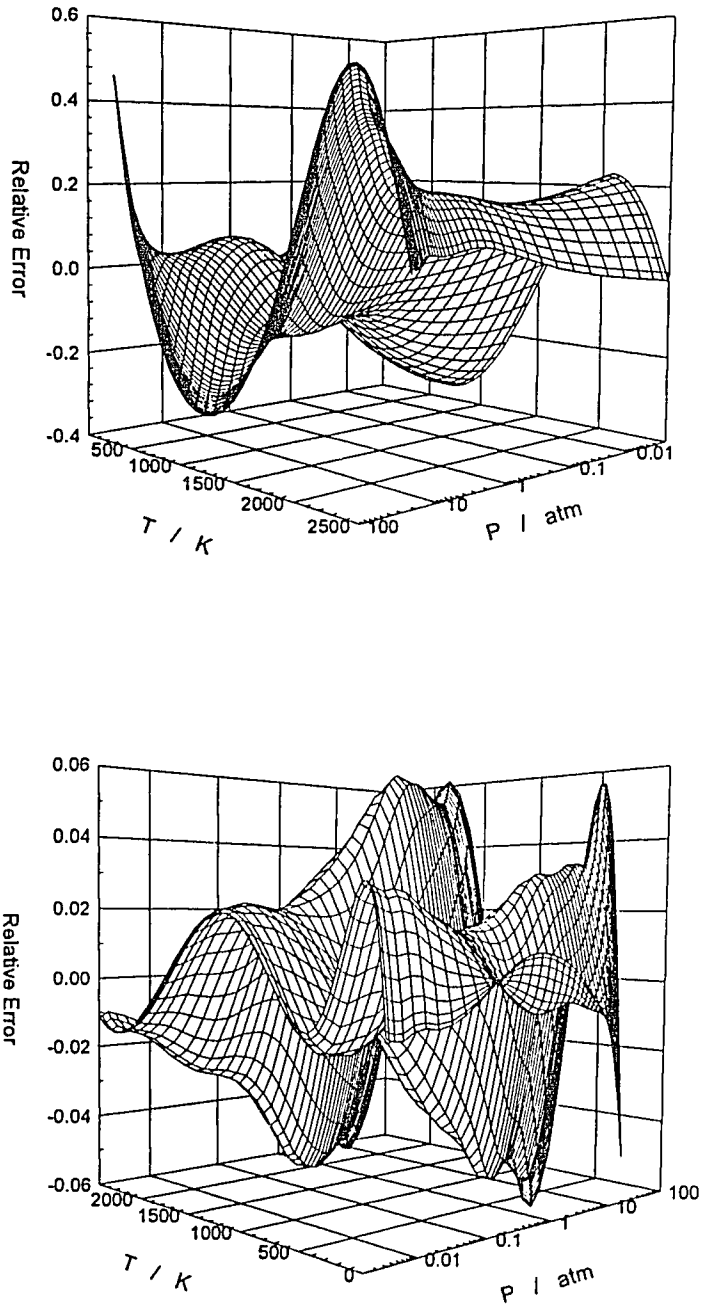


Figure B.77 Relative errors for $CH_2OH + O_2 \rightarrow CH_2O + HO_2$
(a) Chebyshev 7×3 (b) Chebyshev 9×5

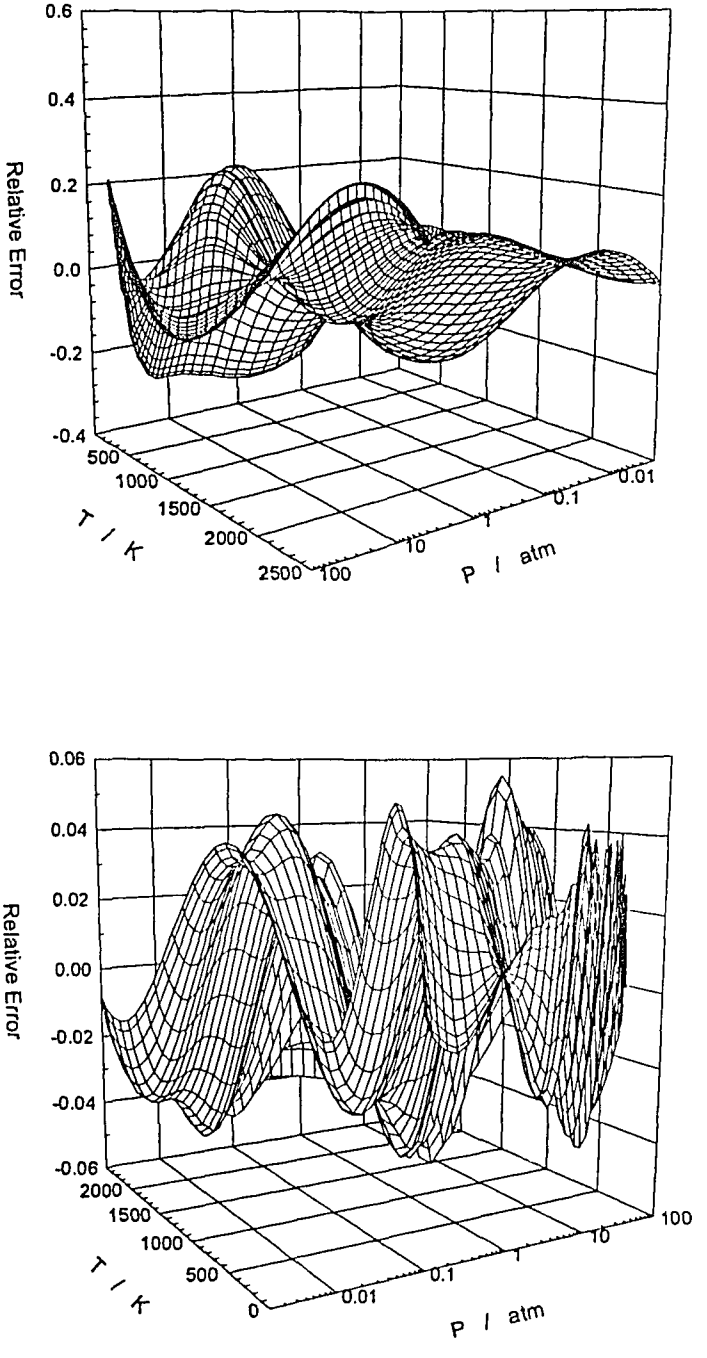


Figure B.78 Relative errors for $CH_2OH + O_2 \rightarrow HOC(O)H_2$
(a) Chebyshev 7×3 (b) Chebyshev 9×5

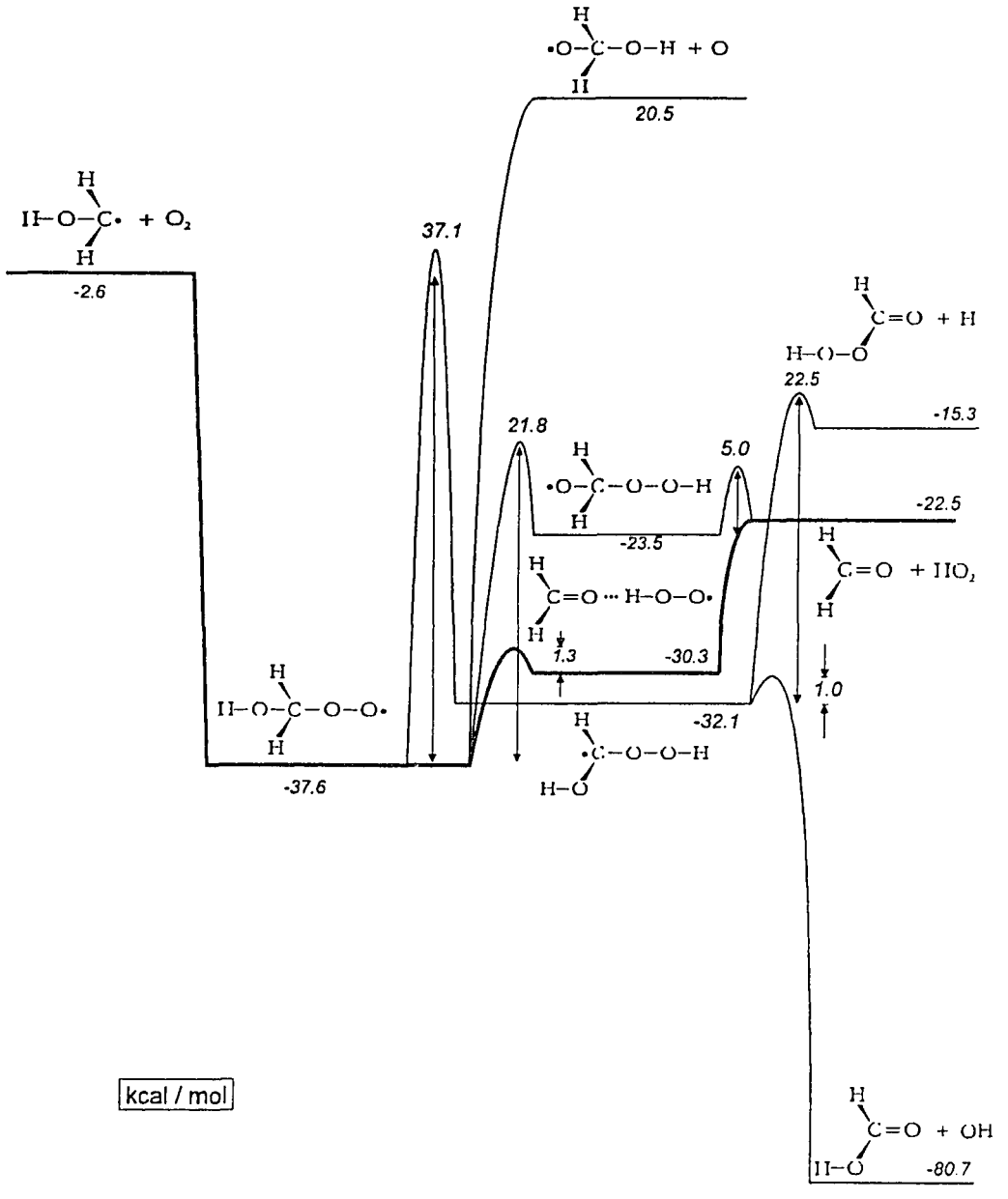


Figure B.79 Potential Energy Diagram of $CH_2OH + O_2$and $H_2C=O + HO_2$ Reaction Systems

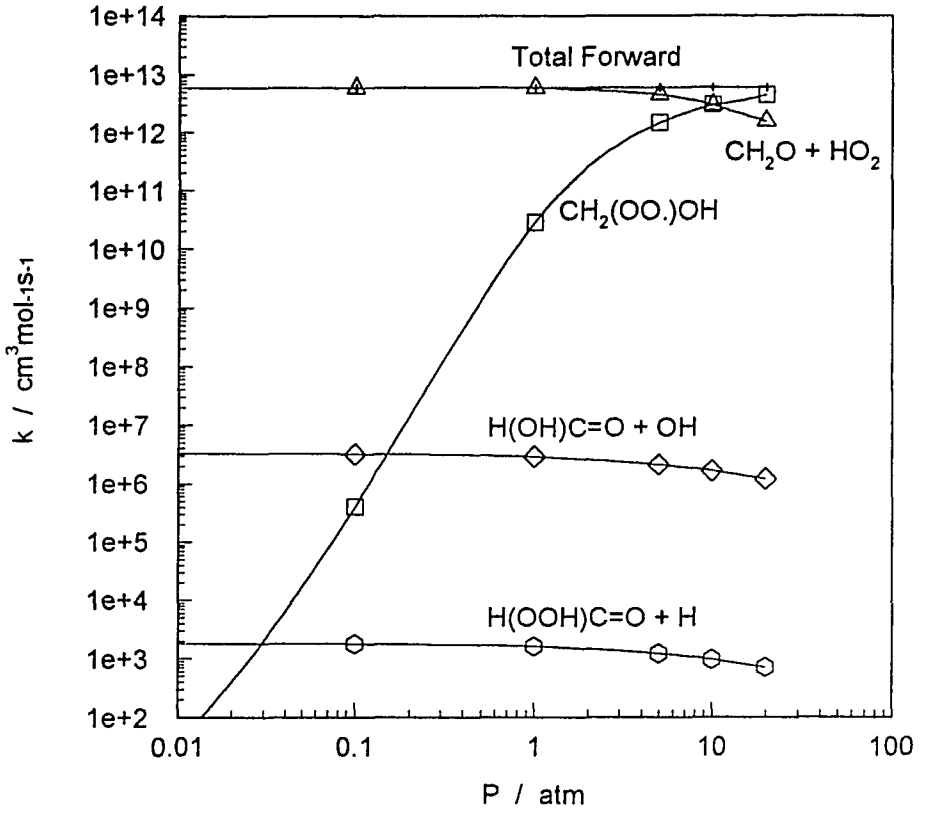


Figure B.80 Results of QRRK Analysis for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product at 300K

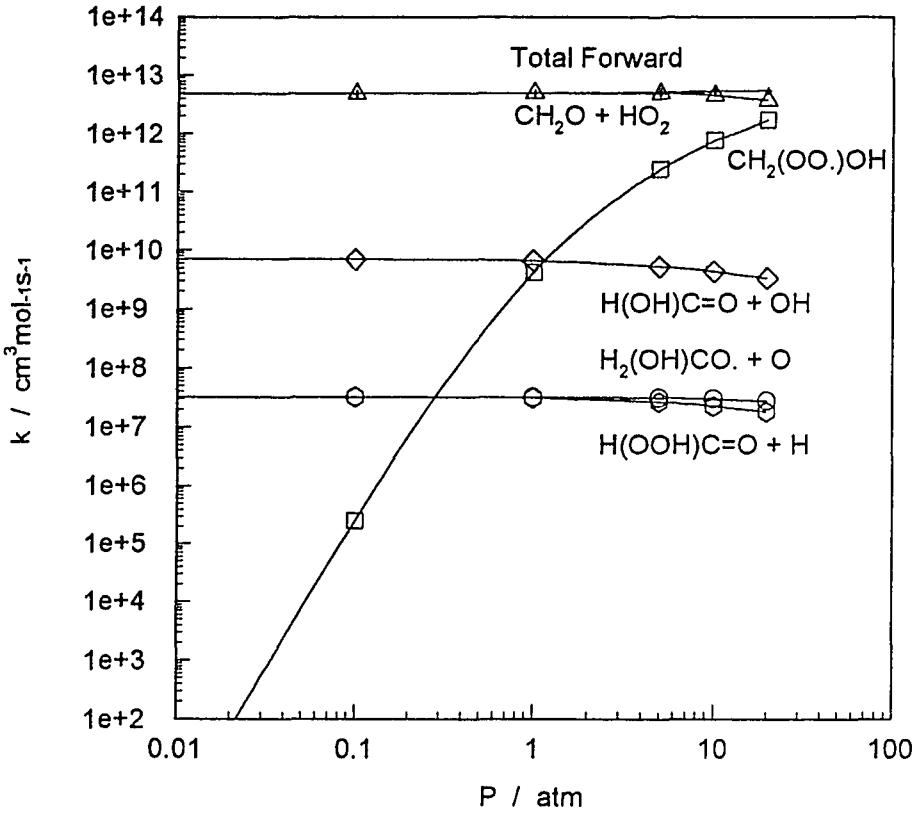


Figure B.81 Results of QRRK Analysis for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product at 900K

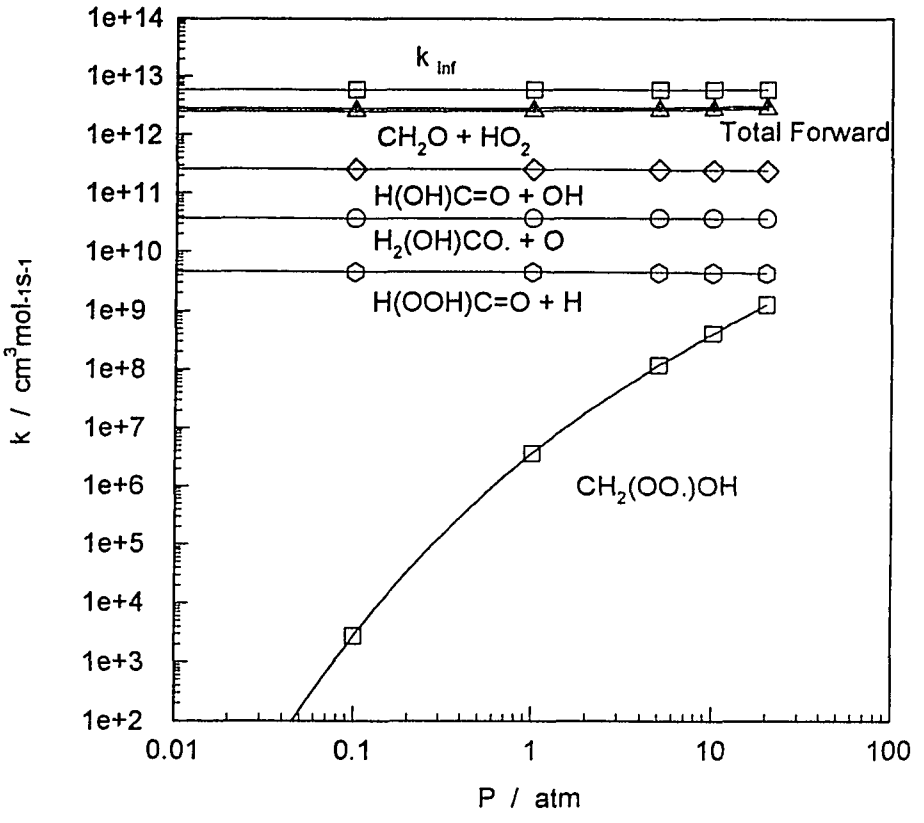


Figure B.82 Results of QRRK Analysis for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product at 1800K

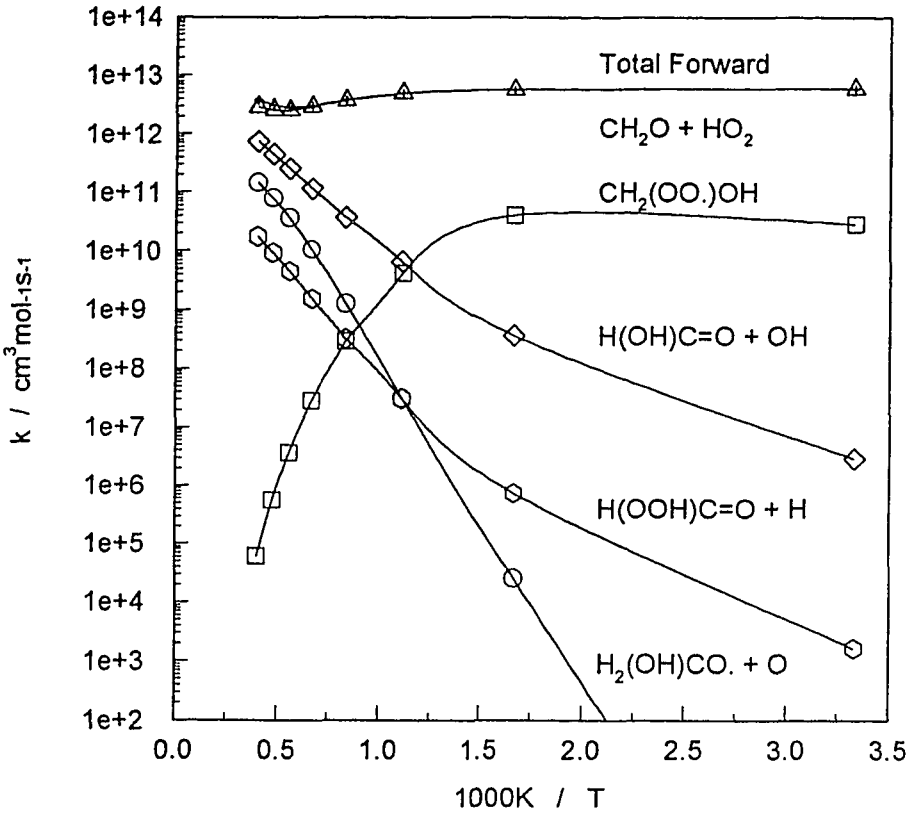


Figure B.83 Results of QRRK Analysis for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product at 1 atm

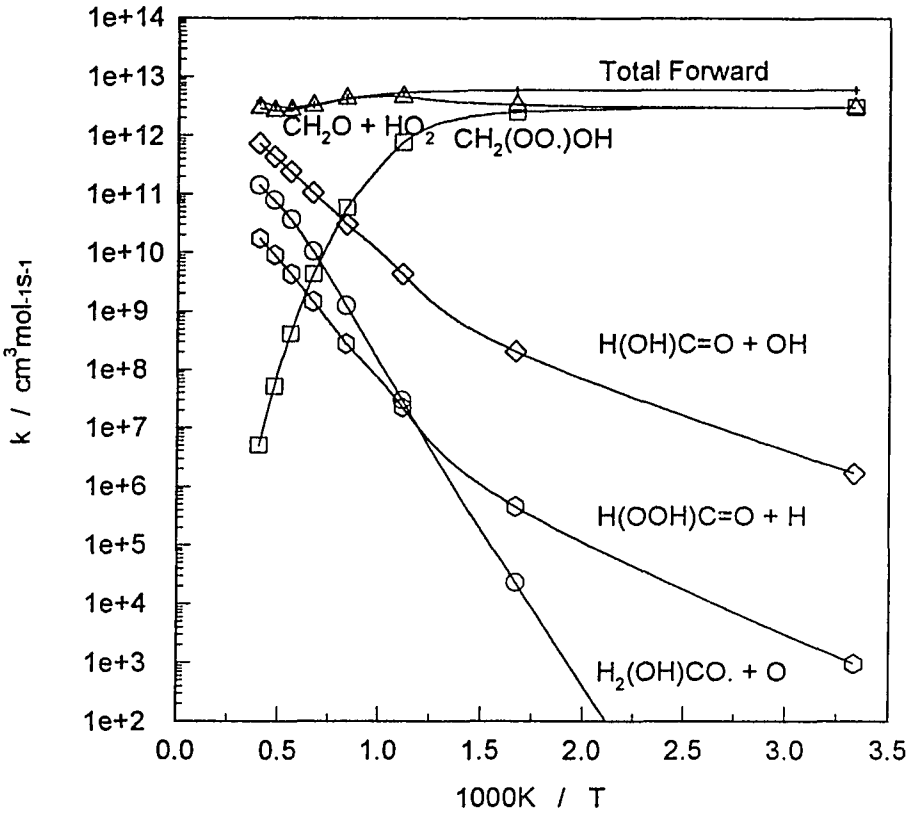


Figure B.84 Results of QRRK Analysis for $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product at 10 atm

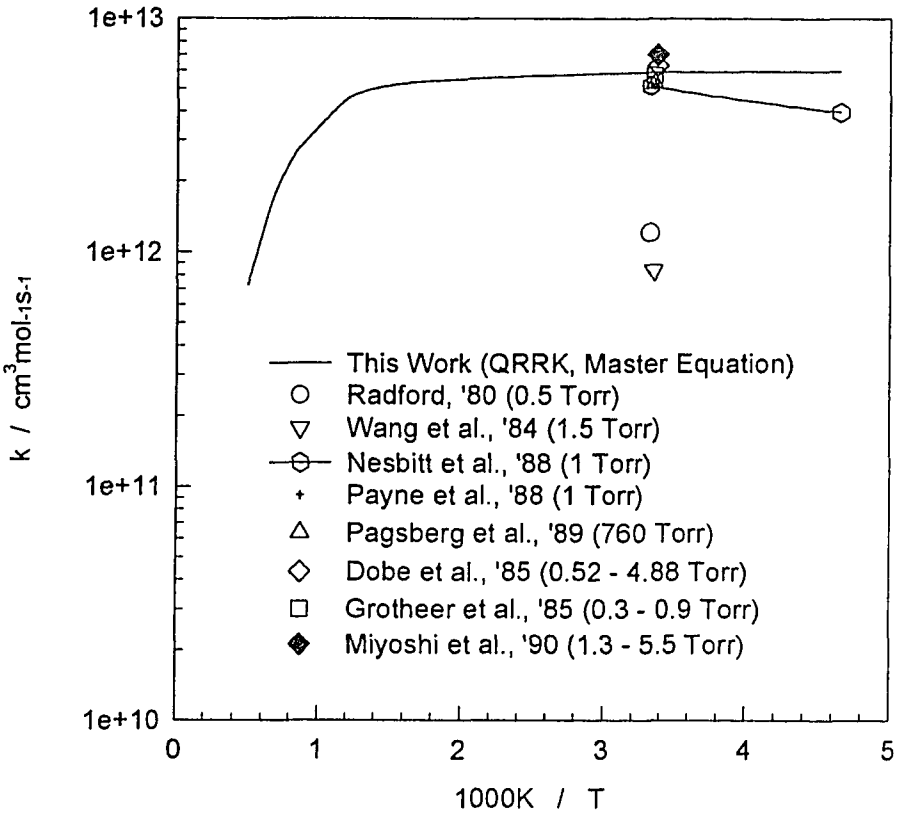


Figure B.85 Comparison of QRRK Analysis and Experimaenal Data for $CH_2OH + O_2 \rightarrow H_2C{=}O + HO_2$

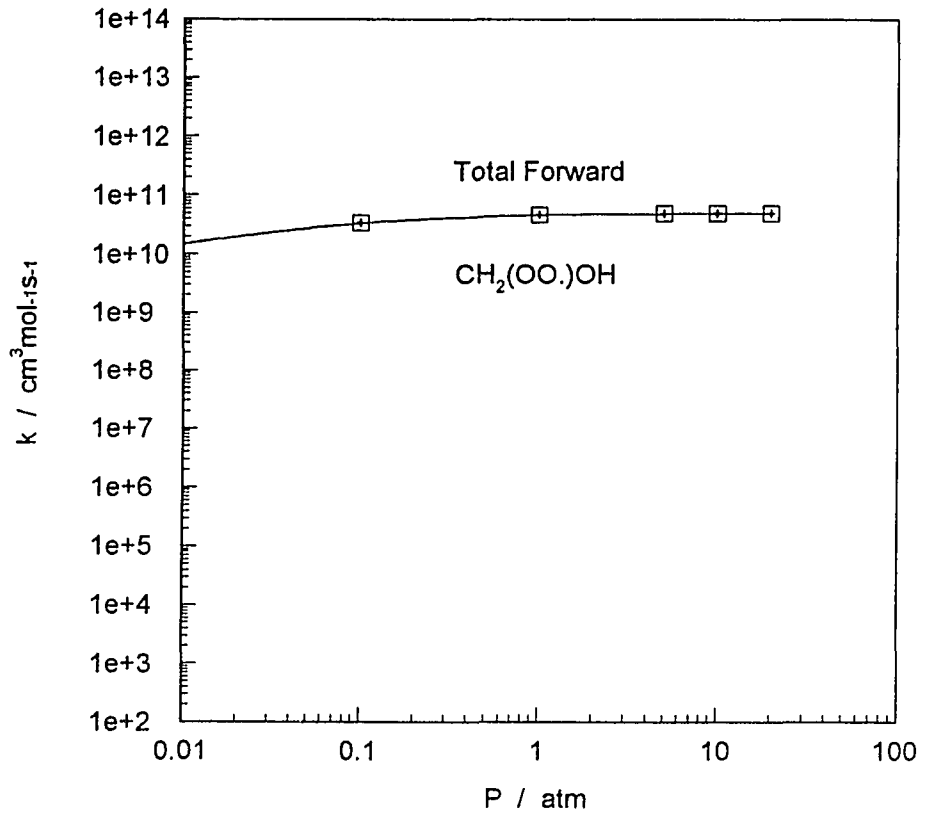


Figure B.86 Results of QRRK Analysis for $H_2C{=}O + HO_2 \leftrightarrow [H_2CO..HOO\bullet]^* \rightarrow$ Product at 300 K

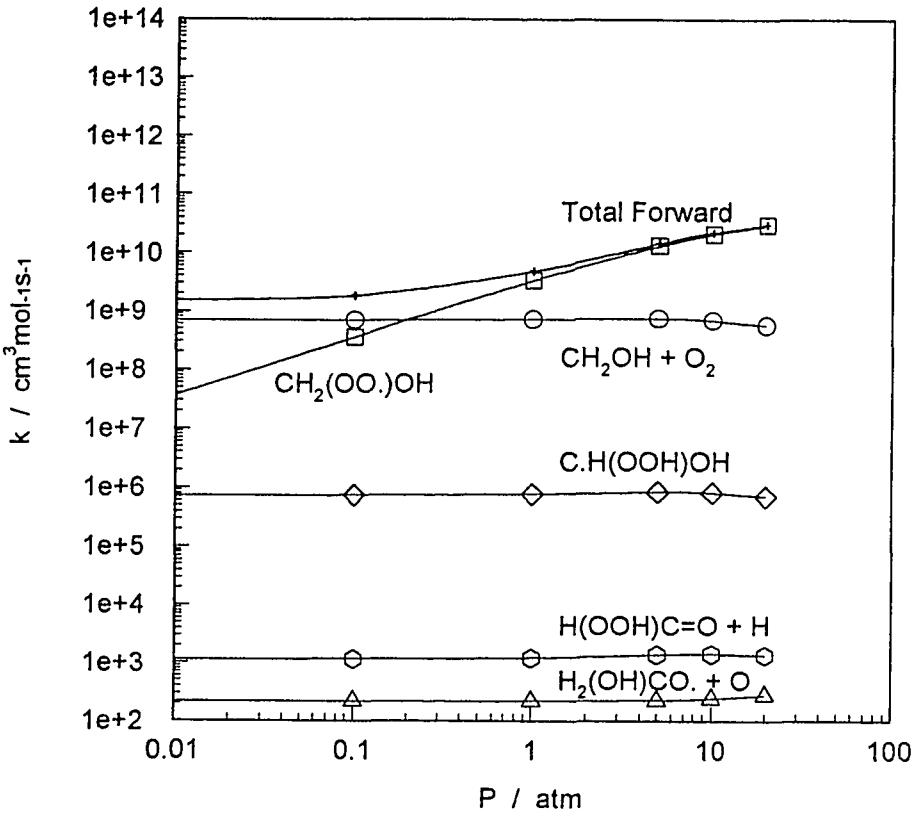


Figure B.87 Results of QRRK Analysis for $H_2C{=}O + HO_2 \leftrightarrow [H_2CO..HOO\bullet]^* \rightarrow$ Product at 900 K

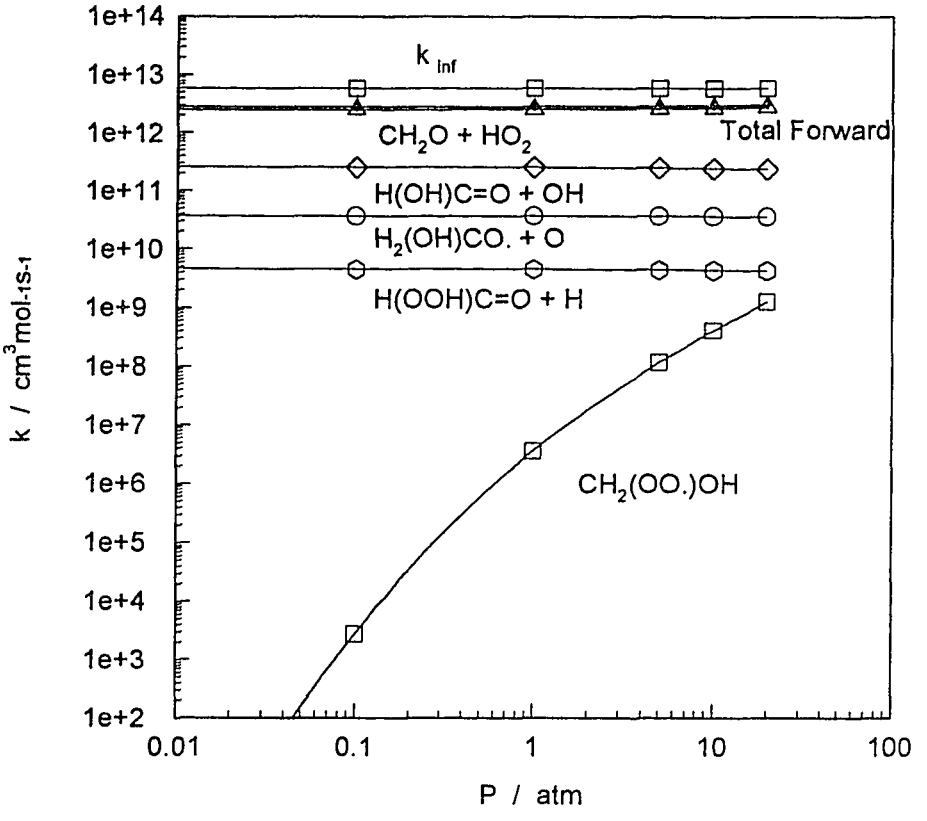


Figure B.88 Results of QRRK Analysis for $H_2C{=}O + HO_2 \leftrightarrow [H_2CO..HOO\bullet]^* \rightarrow$ Product at 1800 K

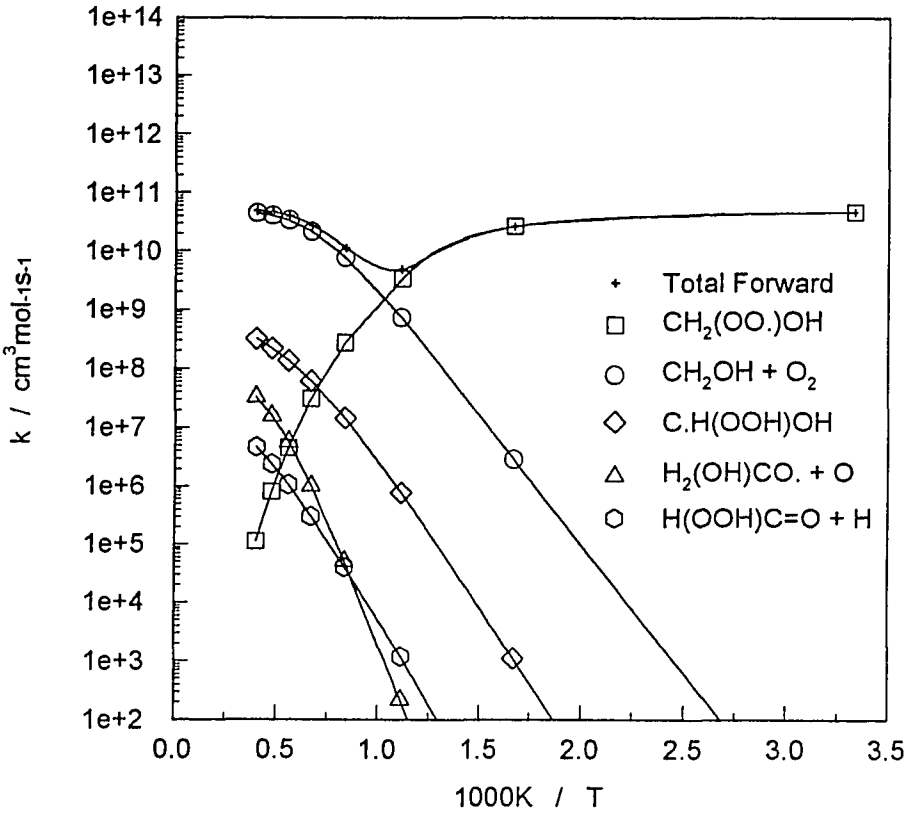


Figure B.89 Results of QRRK Analysis for $H_2C{=}O + HO_2 \leftrightarrow [H_2CO..HOO\bullet]^* \rightarrow$ Product at 1 atm

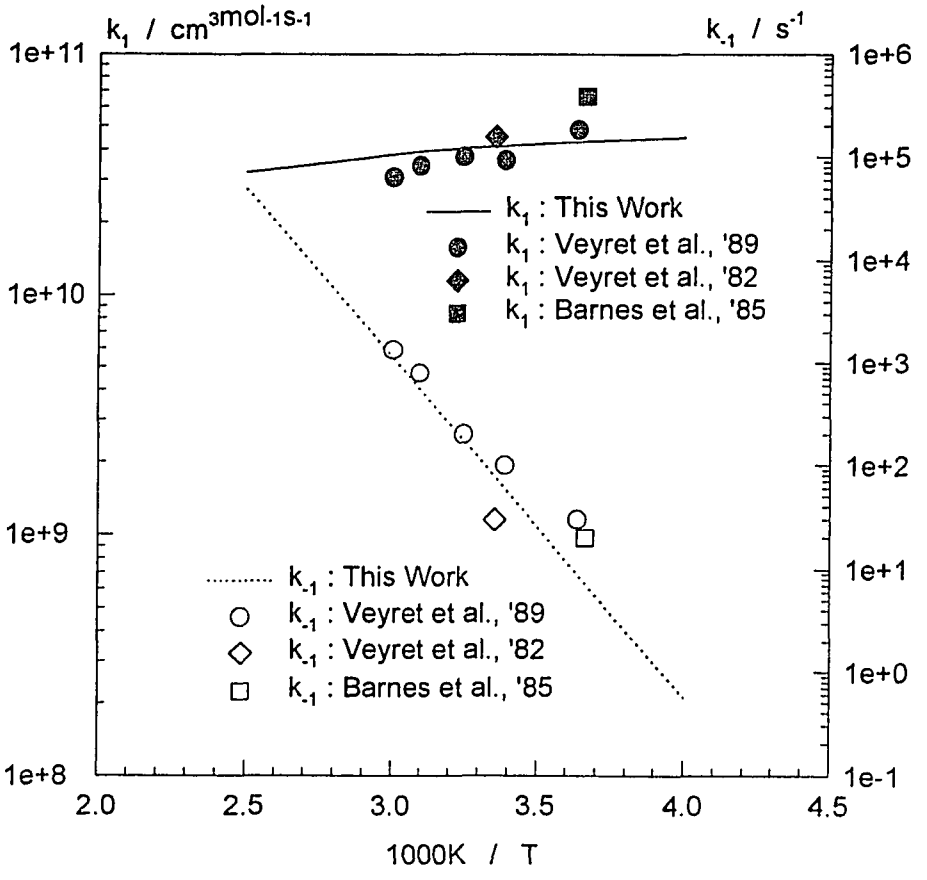


Figure B.90 Comparison of QRRK Analysis and Experimaenal Data for $H_2C{=}O + HO_2 \leftrightarrow H_2(OH)COO\bullet$

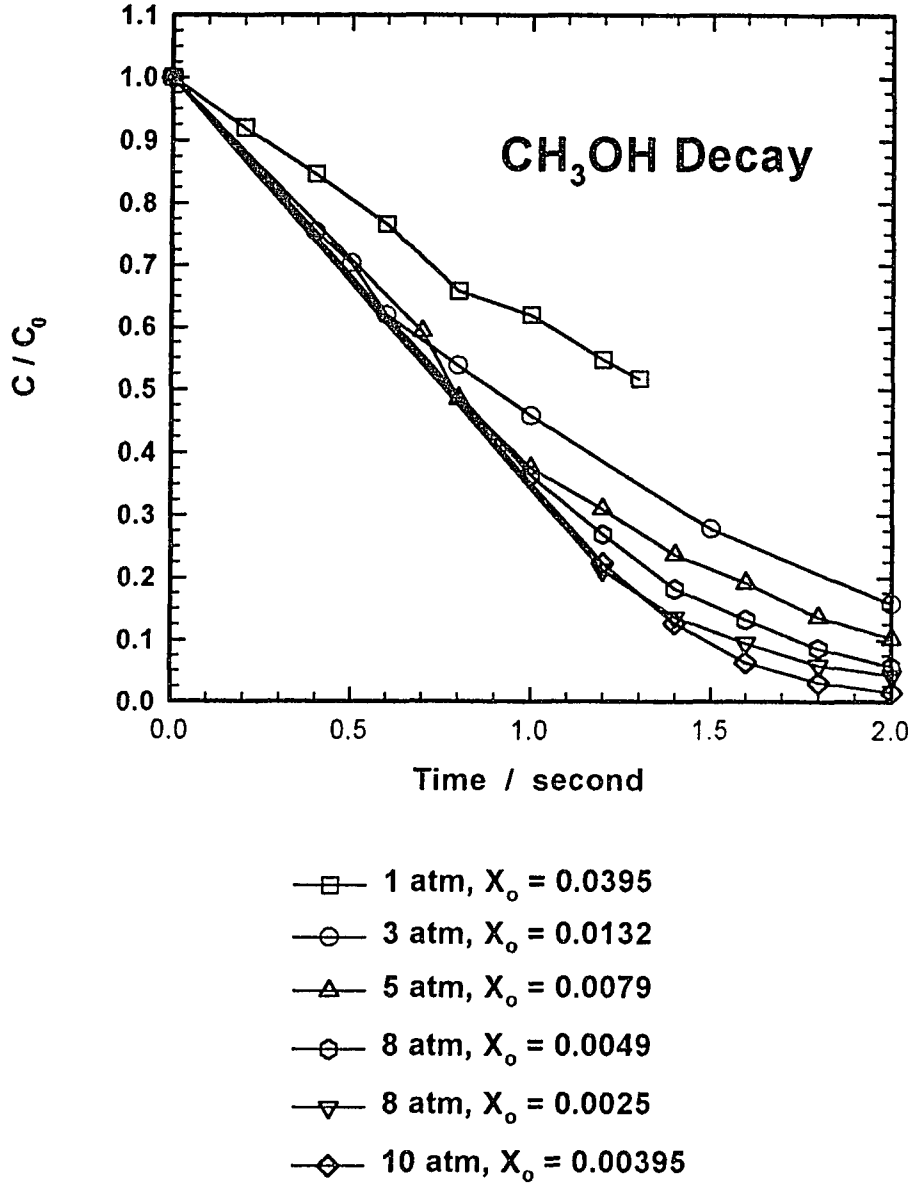


Figure B.91 Experimental Results of Methanol Pyrolysis at 873 K

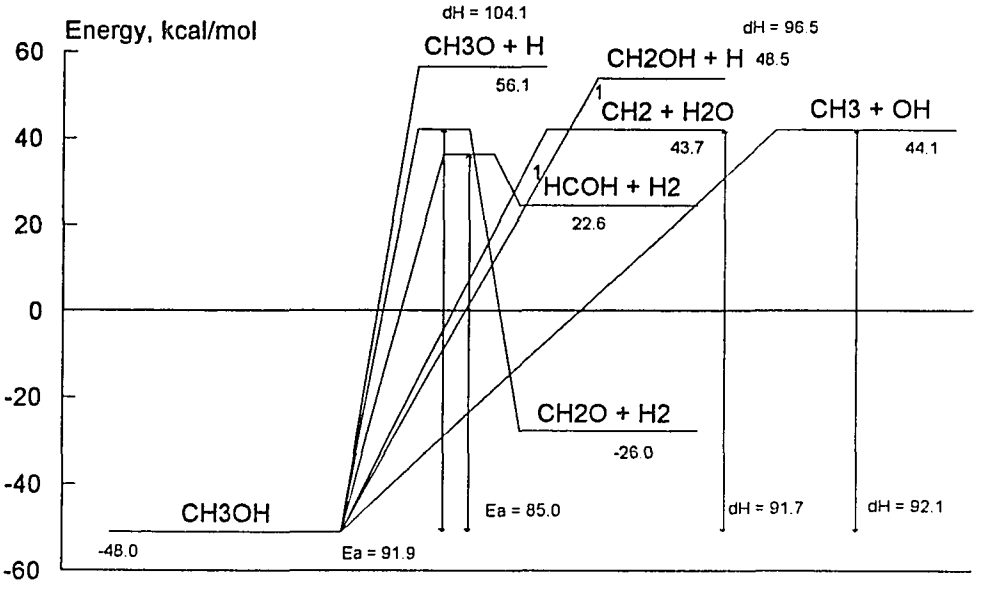


Figure B.92 Potential Energy Diagram for $CH_3OH \rightarrow$ Products

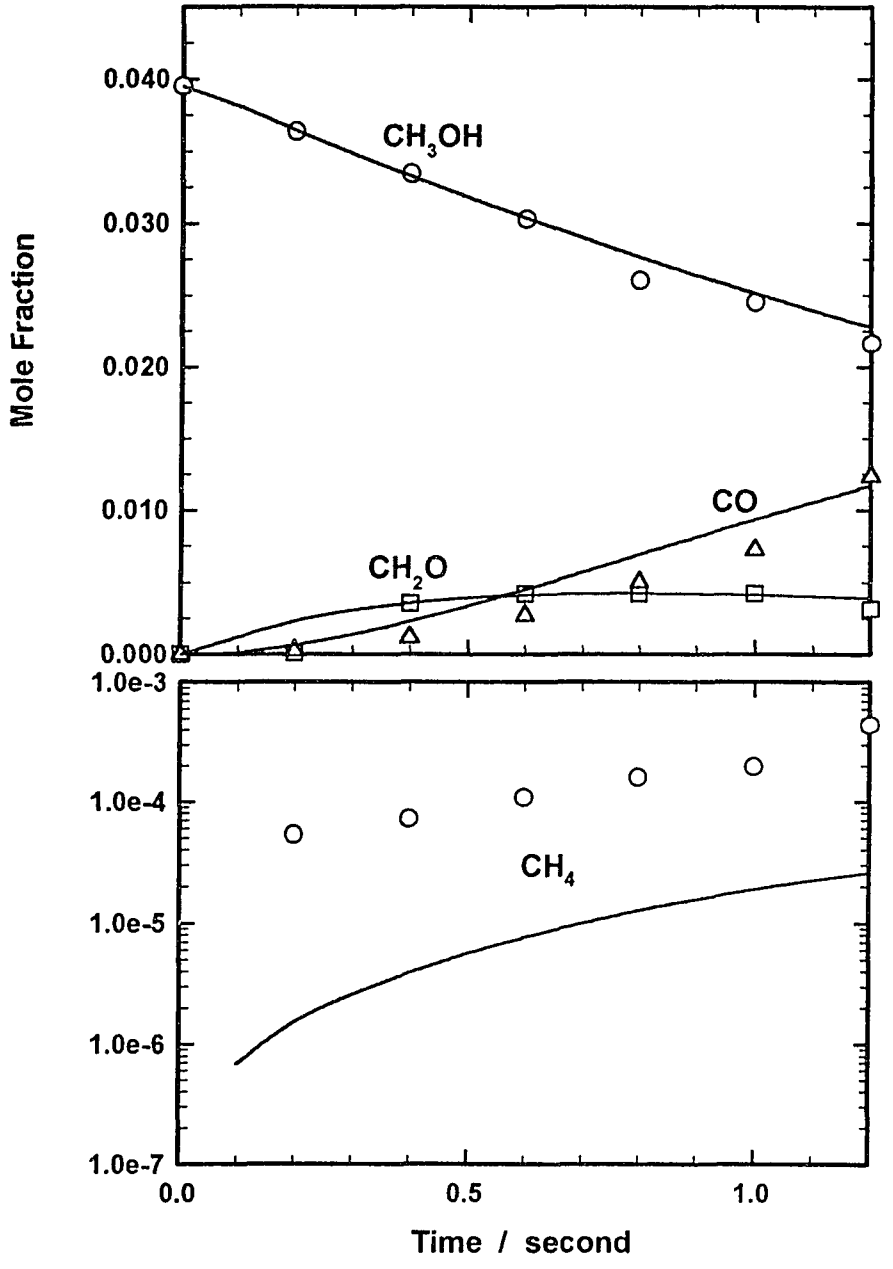


Figure B.93 Comparison of Model and Experimental Results for Methanol Pyrolysis at 1073 K, 1 atm, $X_o = 3.95\%$

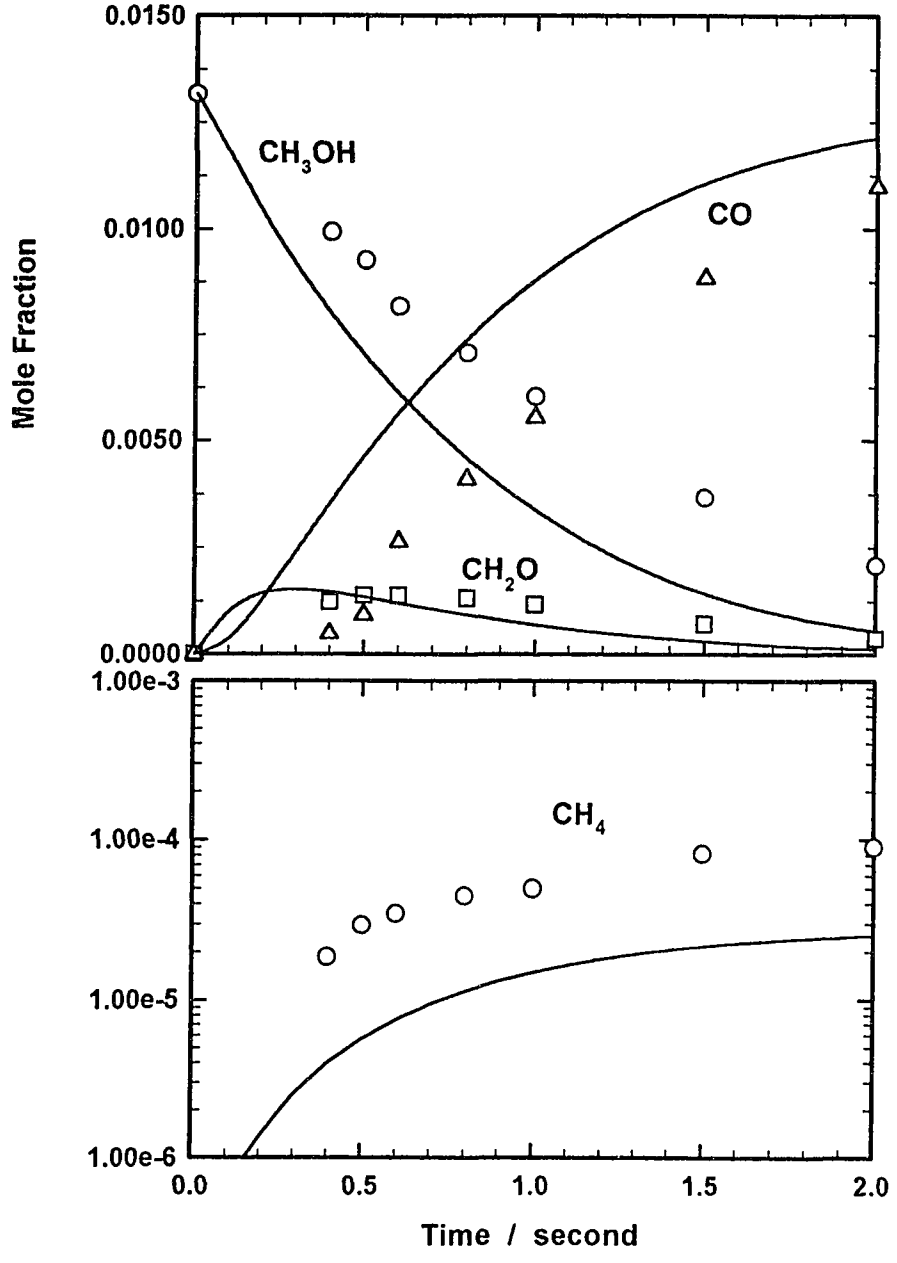


Figure B.94 Comparison of Model and Experimental Results for Methanol Pyrolysis at 1073 K, 3 atm, $X_o = 1.317\%$

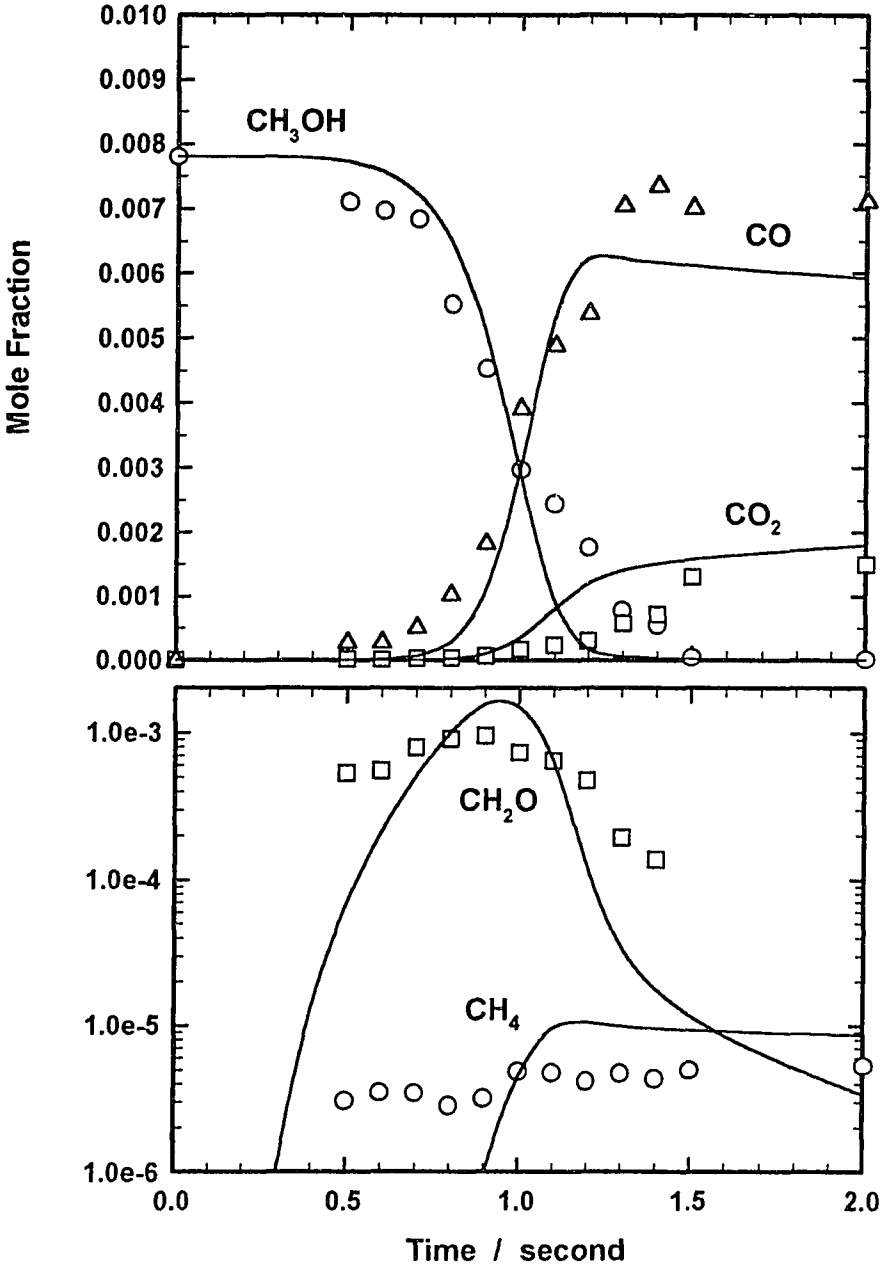


Figure B.95 Comparison of Model and Experimental Results for Methanol Oxidation at 873 K, 5 atm, $X_o = 0.78\%$, $\phi = 1.0$

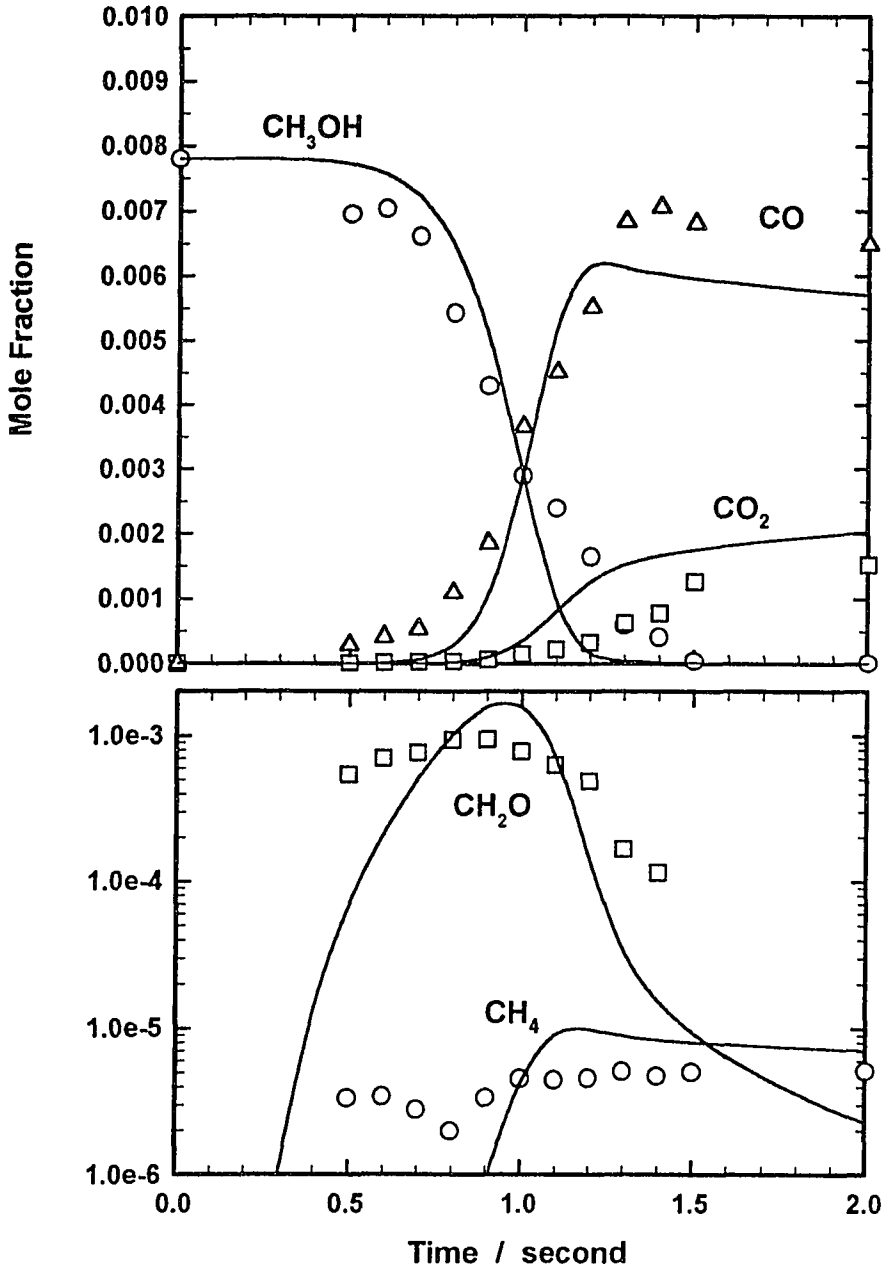


Figure B.96 Comparison of Model and Experimental Results for Methanol Oxidation at 873 K, 5 atm, $X_o = 0.78\%$, $\phi = 0.75$

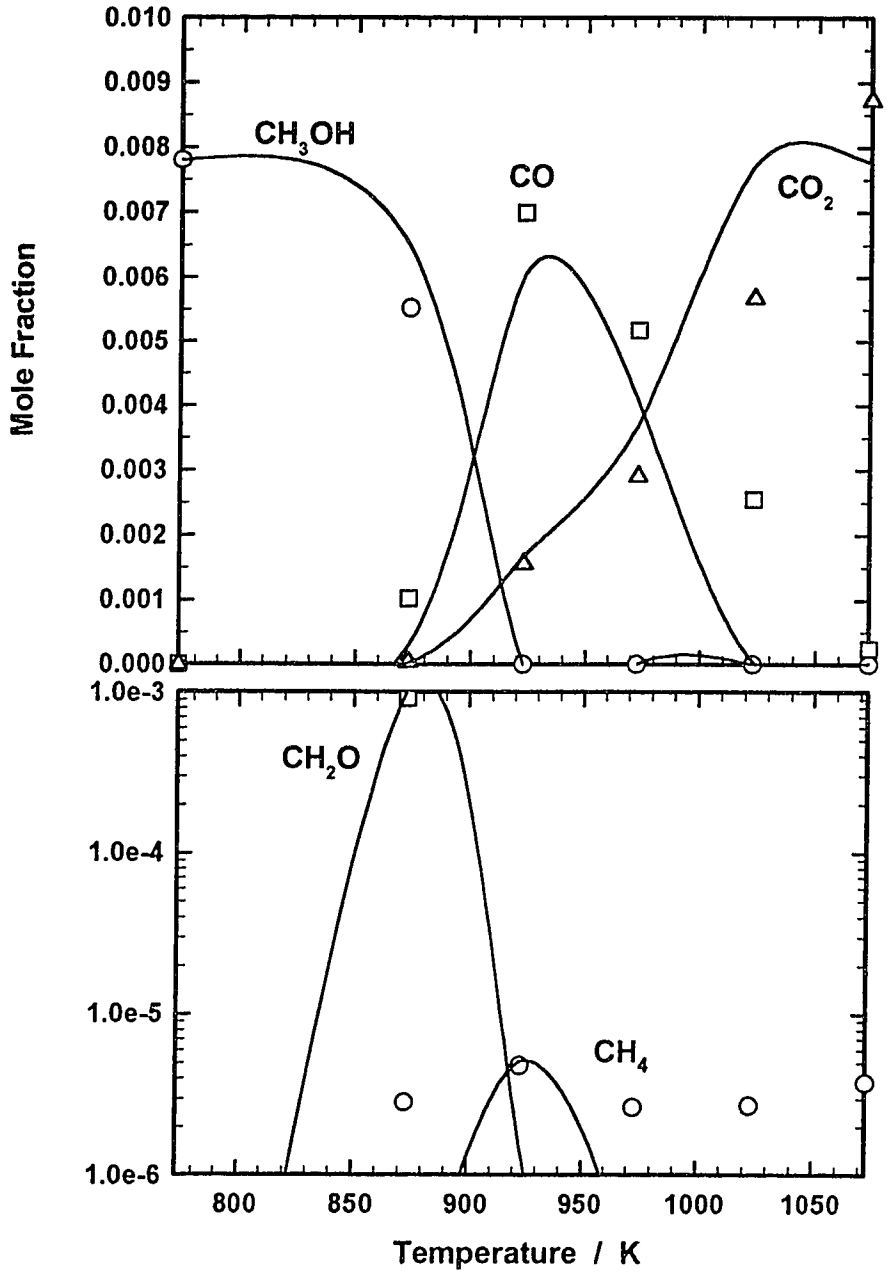


Figure B.97 Comparison of Model and Experimental Results for Methanol Oxidation at Residence Time = 0.8 second, 5 atm, X_o = 0.78%, ϕ = 1.0

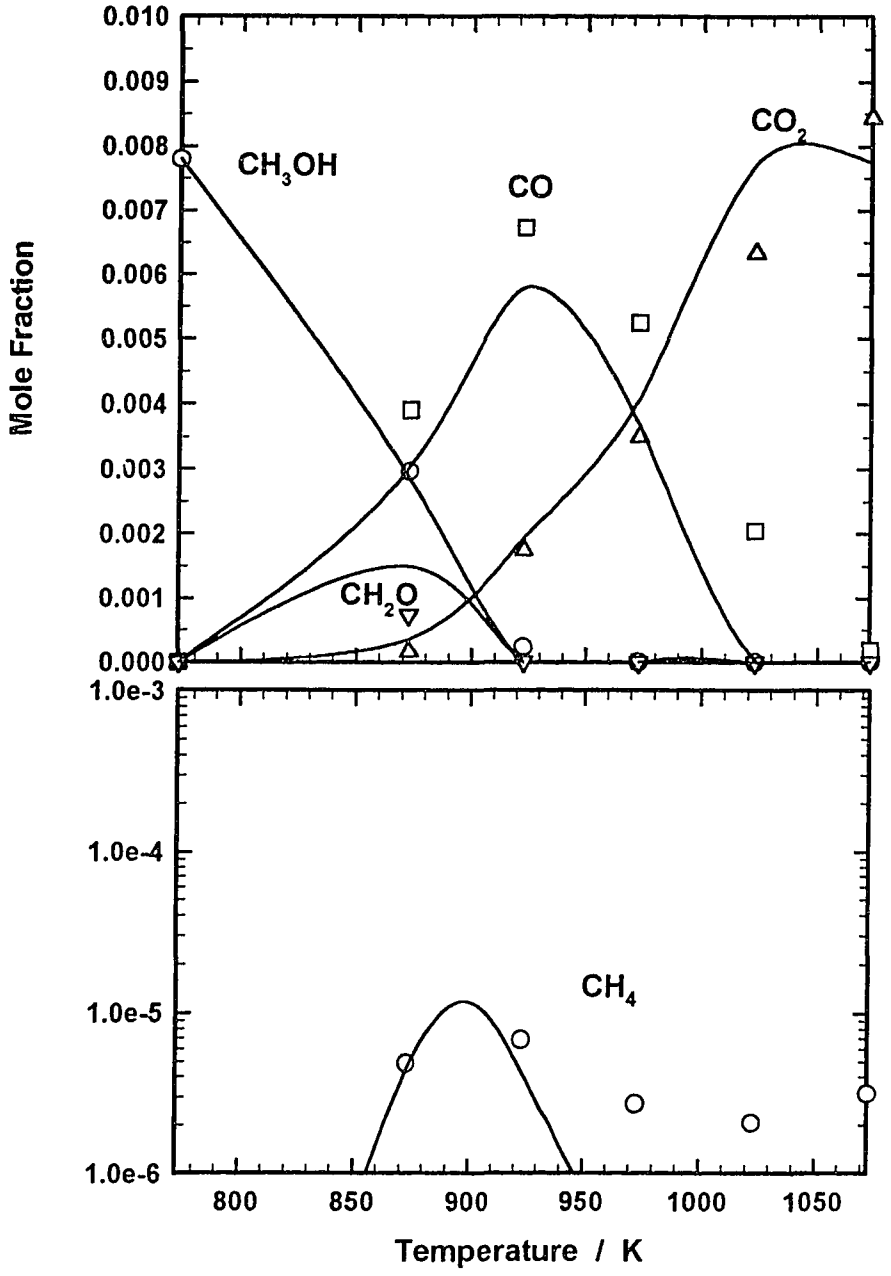


Figure B.98 Comparison of Model and Experimental Results for Methanol Oxidation at Residence Time = 1.0 second, 5 atm, $X_o = 0.78\%$, $\phi = 1.0$

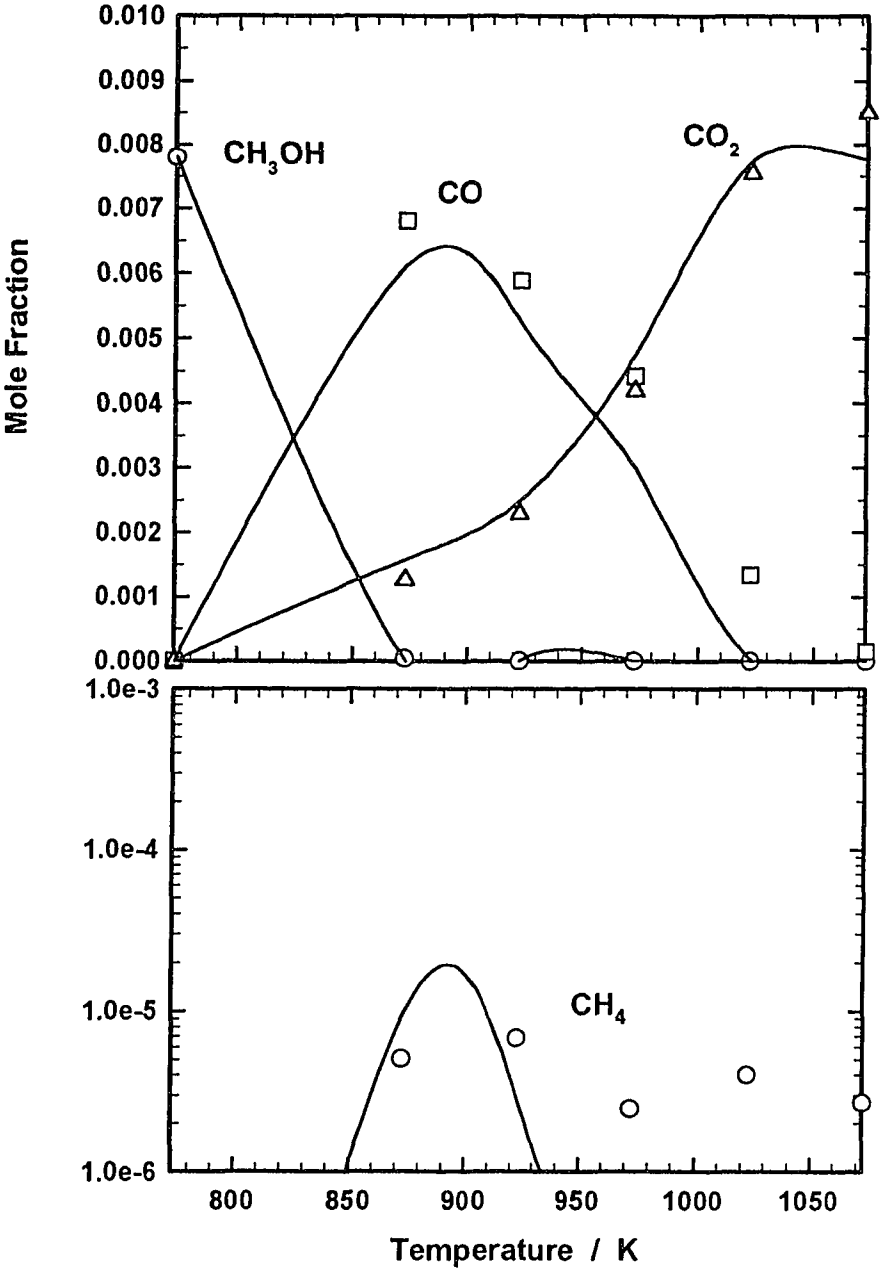


Figure B.99 Comparison of Model and Experimental Results for Methanol Oxidation at Residence Time = 1.5 second, 5 atm, X_o = 0.78%, ϕ = 1.0

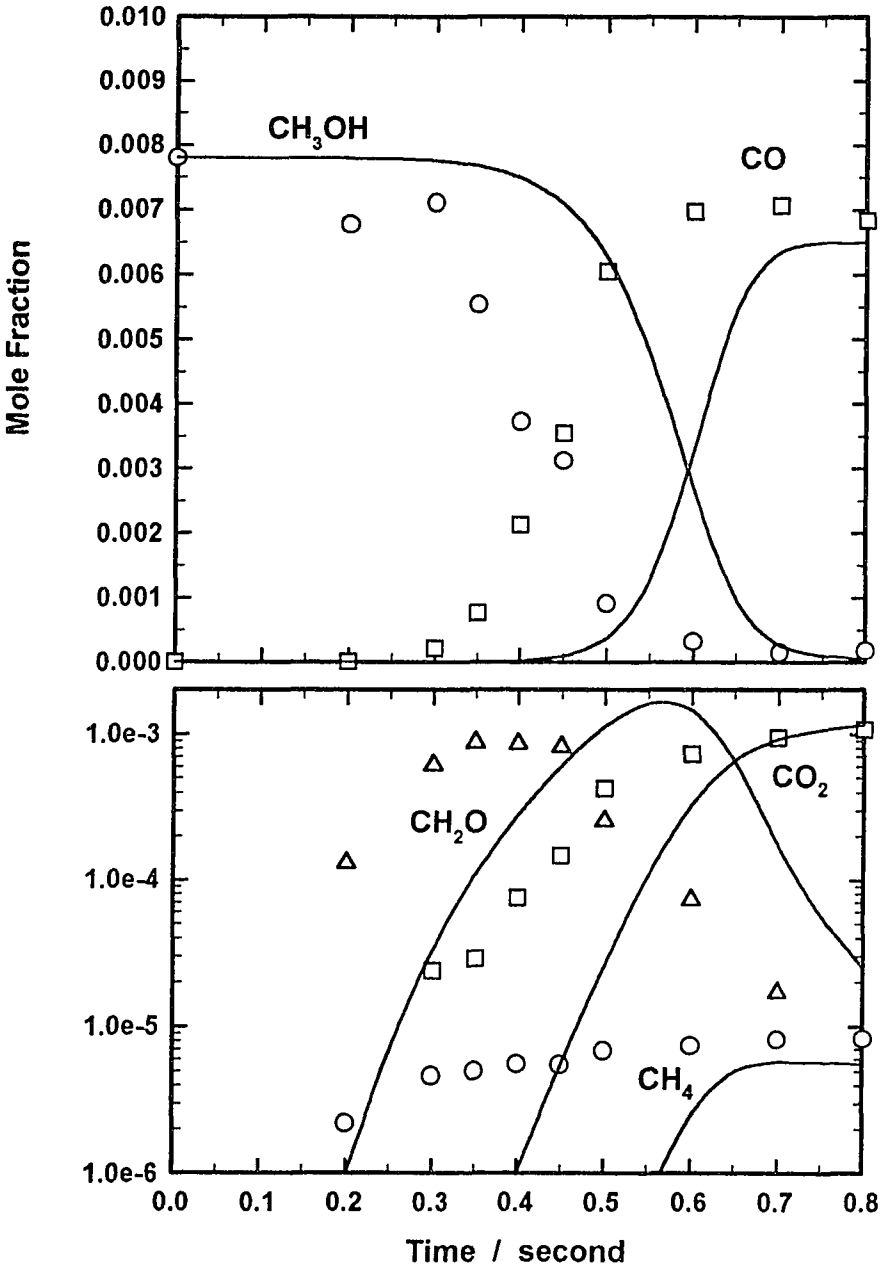


Figure B.100 Comparison of Model and Experimental Results for Methanol Oxidation at 923 K, 3 atm, $X_o = 0.78\%$, $\phi = 1.0$

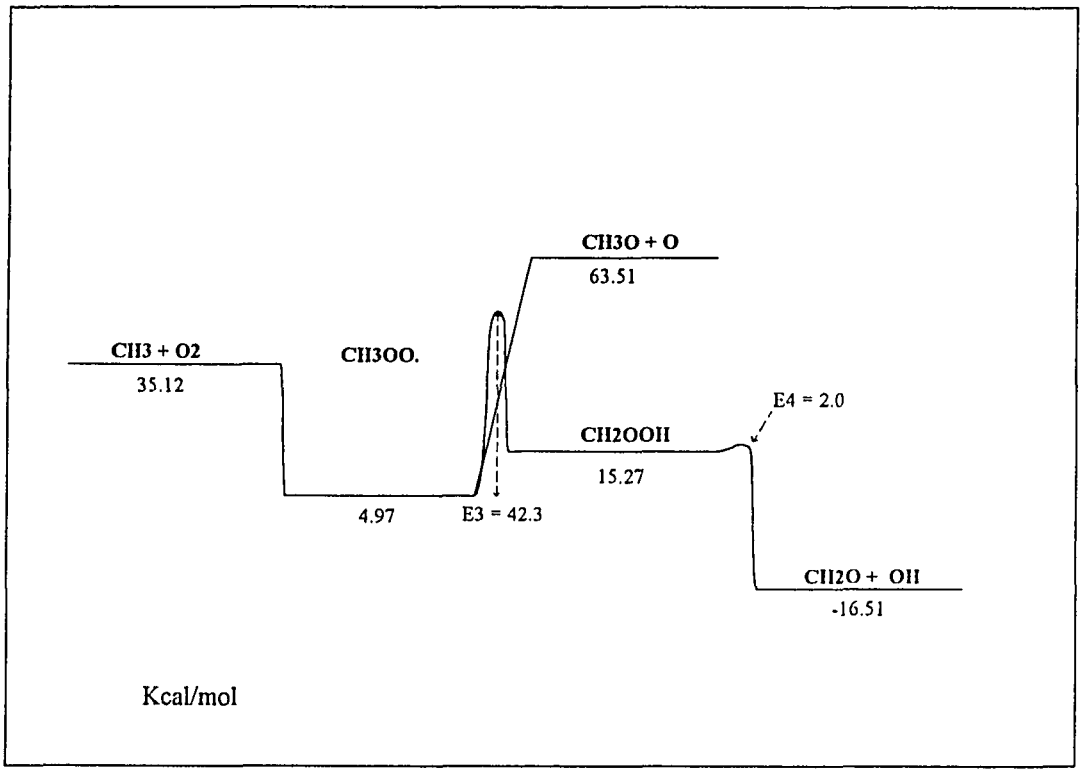
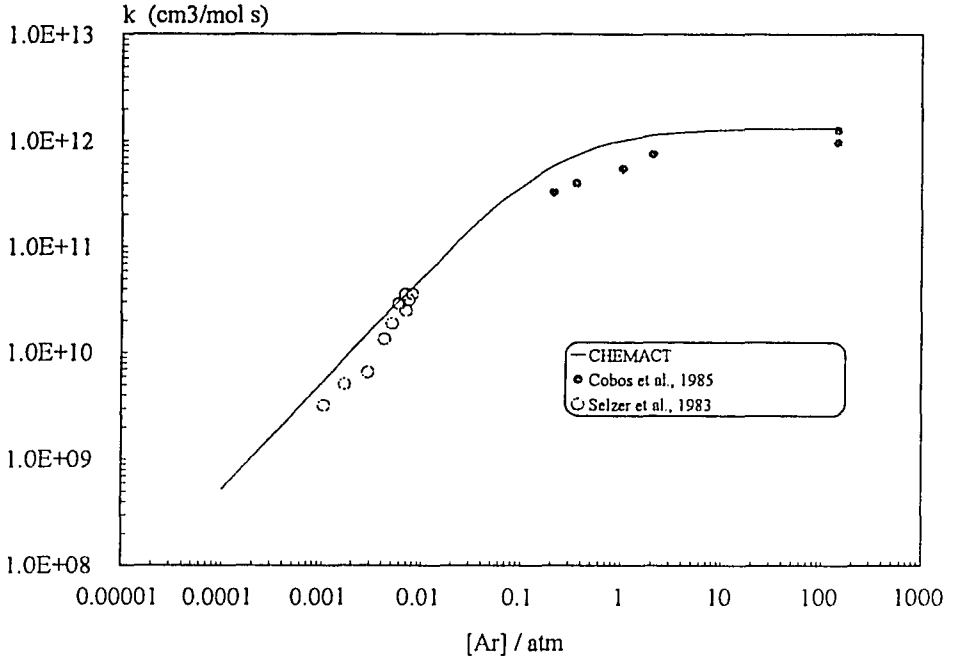


Figure B.101 Potential Energy Diagram for $CH_3 + O_2 \leftrightarrow [CH_3OO]^* \rightarrow$ Products



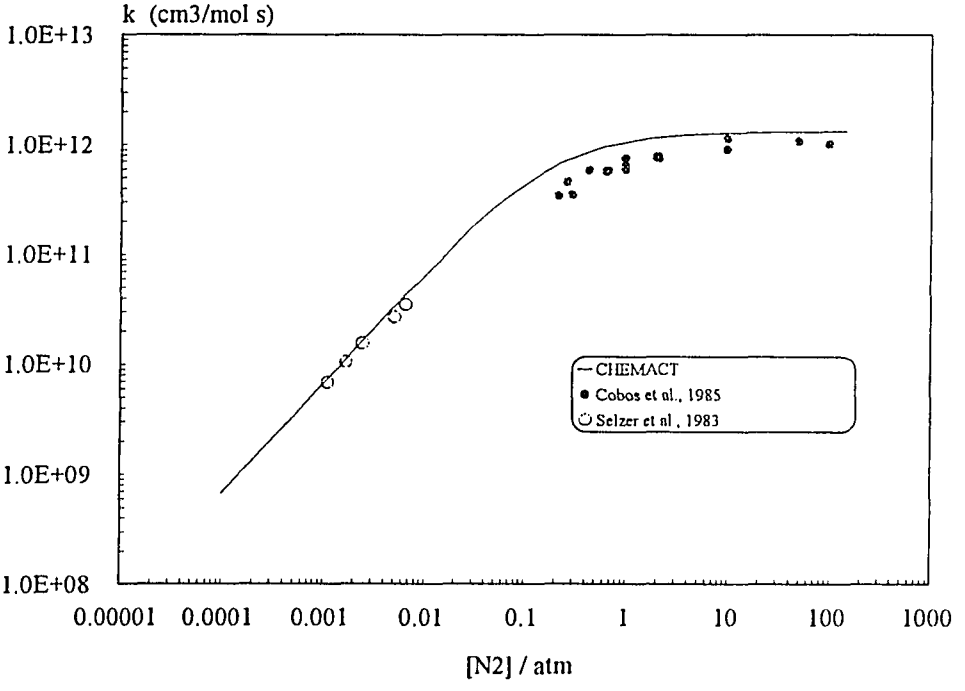


Figure B.102 Comparisons of QRRK Analysis Results and Experimental Data for $CH_3 + O_2 \rightarrow CH_3OO$ (a) Bath Gas = Ar (b) Bath Gas = N_2

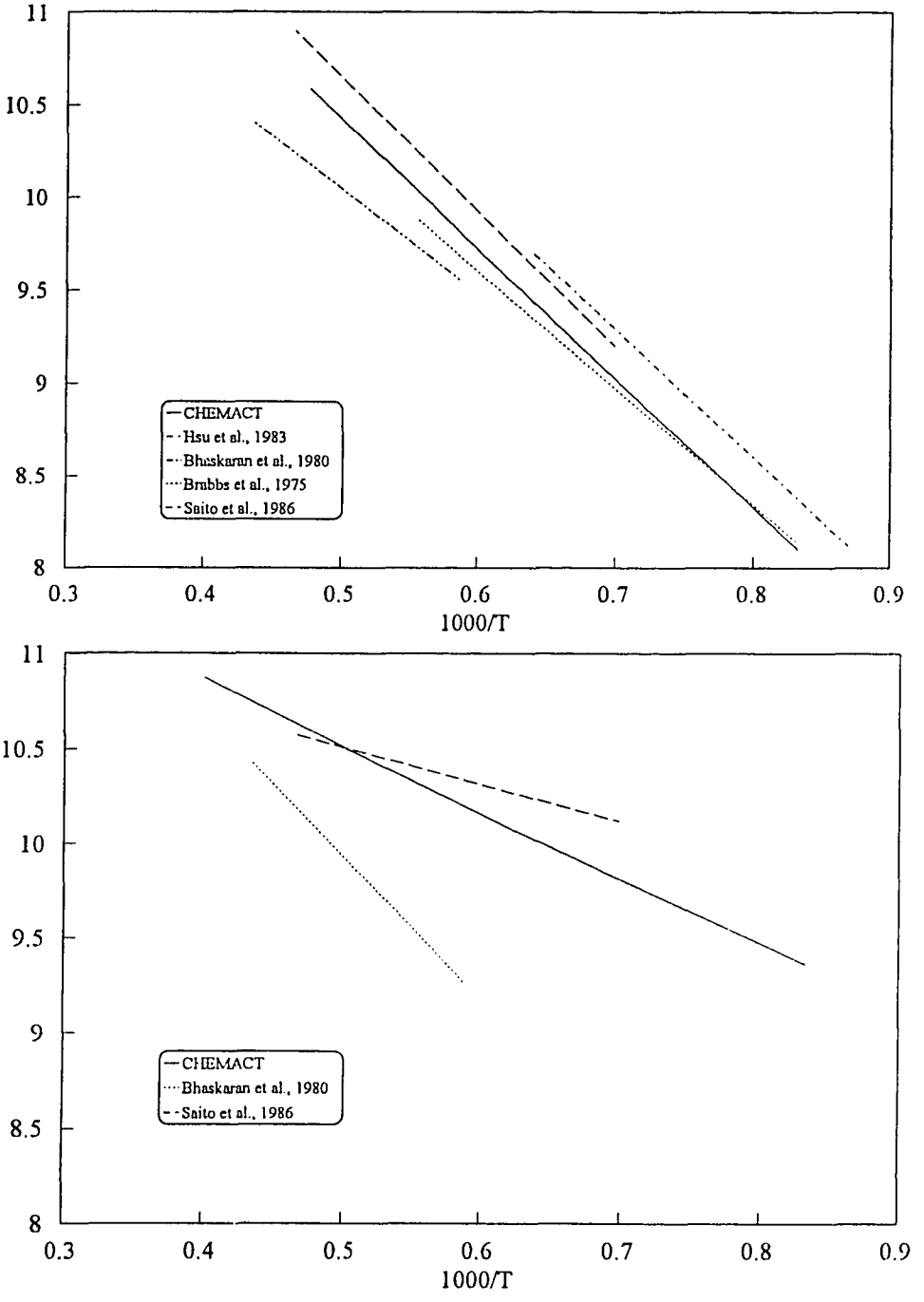


Figure B.103 Comparisons of QRRK Analysis Results and Experimental Data for (a) $CH_3 + O_2 \rightarrow CH_3O + O$ (b) $CH_3 + O_2 \rightarrow CH_2O + OH$

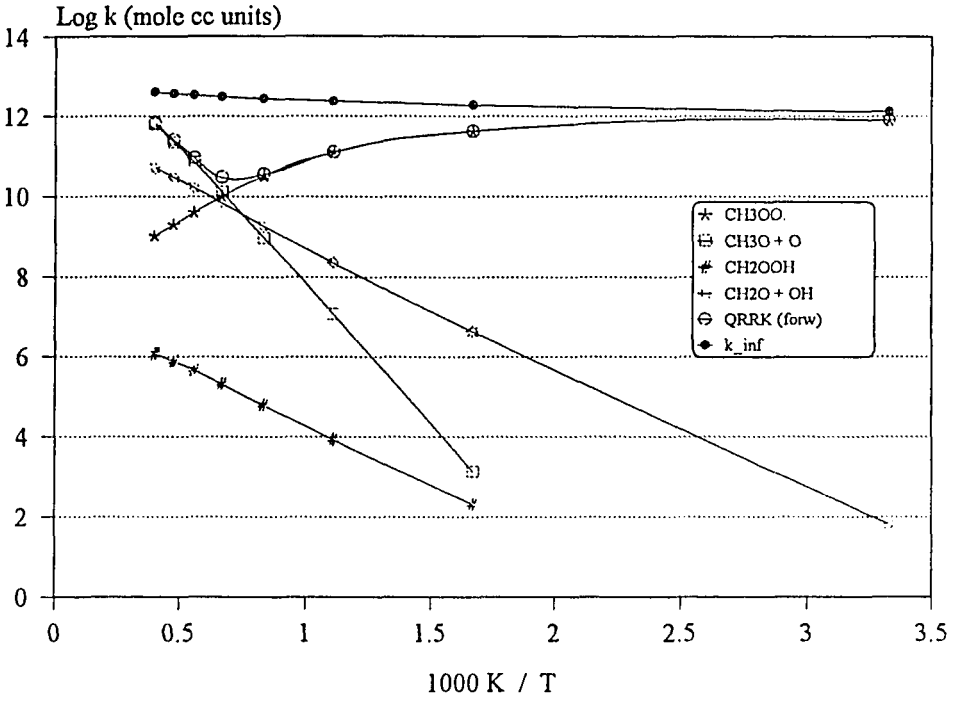


Figure B.104 Results of QRRK Analysis for $CH_3 + O_2 \rightarrow [CH_3OO]^* \rightarrow$ Products

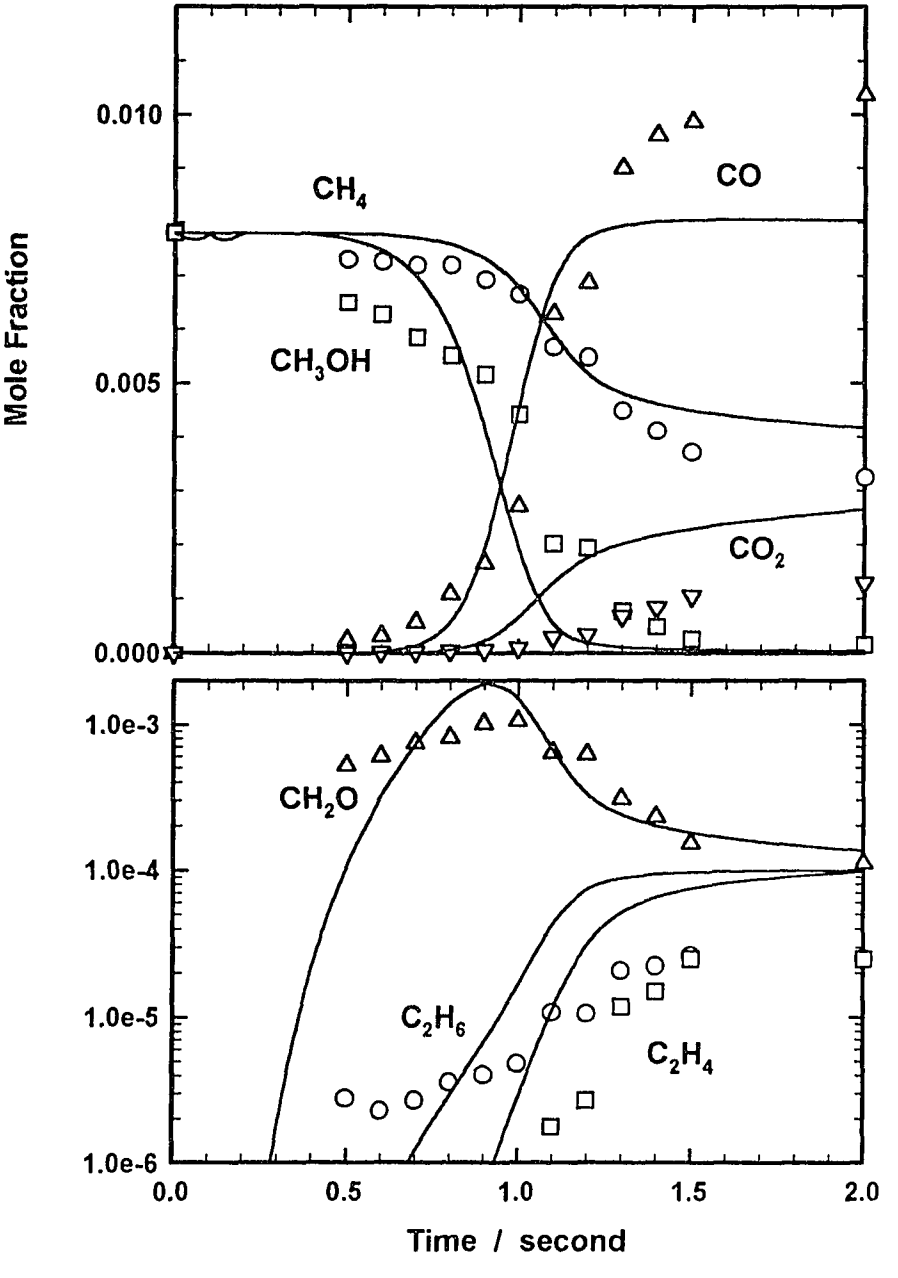


Figure B.105 Comparison of Model and Experimental Results for Methane/Methanol Oxidation at 873 K, 1 atm, $X_o(CH_4) = 0.78\%$, $X_o(CH_3OH) = 0.78\%$, $\phi = 1.0$

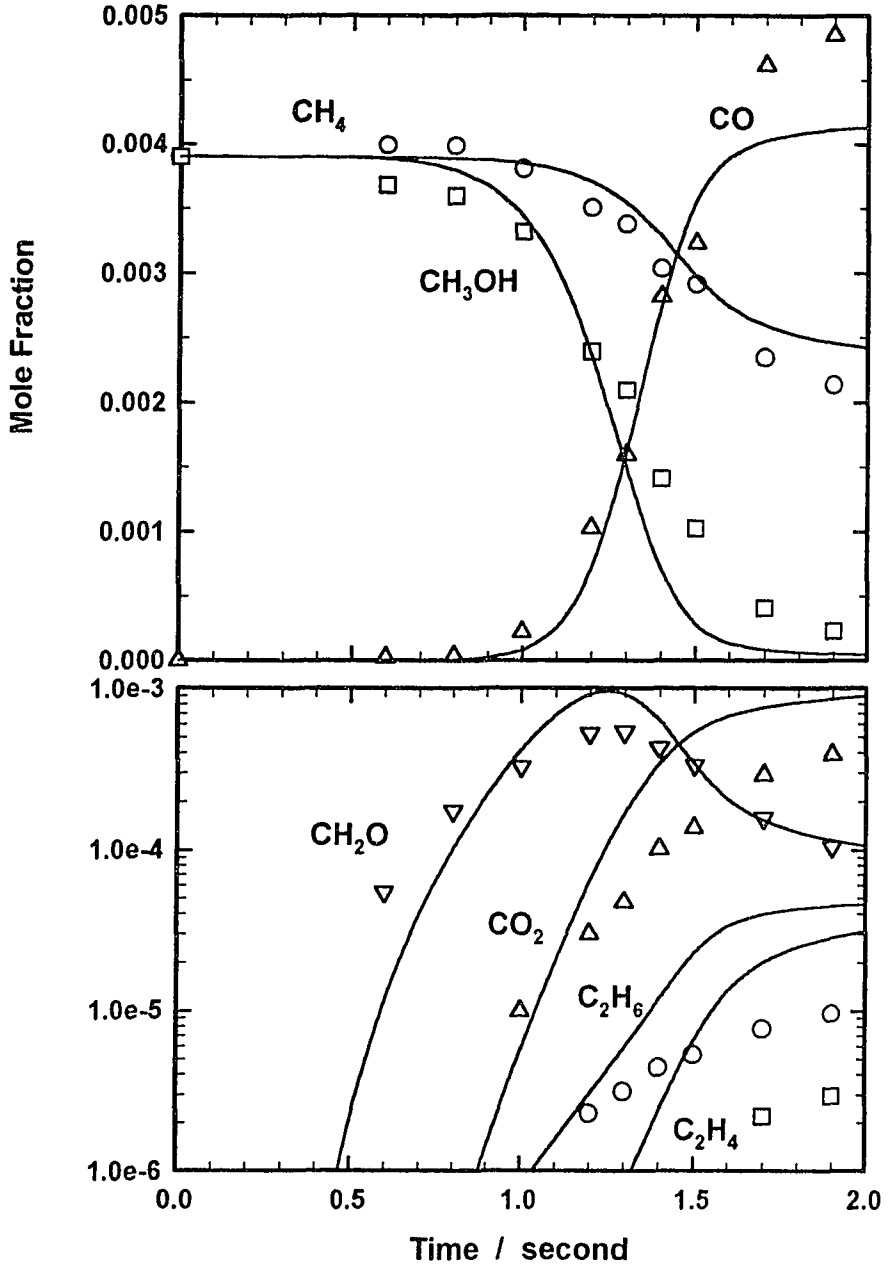


Figure B.106 Comparison of Model and Experimental Results for Methane/Methanol Oxidation at 873 K, 1 atm, $X_o(CH_4) = 0.39\%$, $X_o(CH_3OH) = 0.39\%$, $\phi = 1.0$

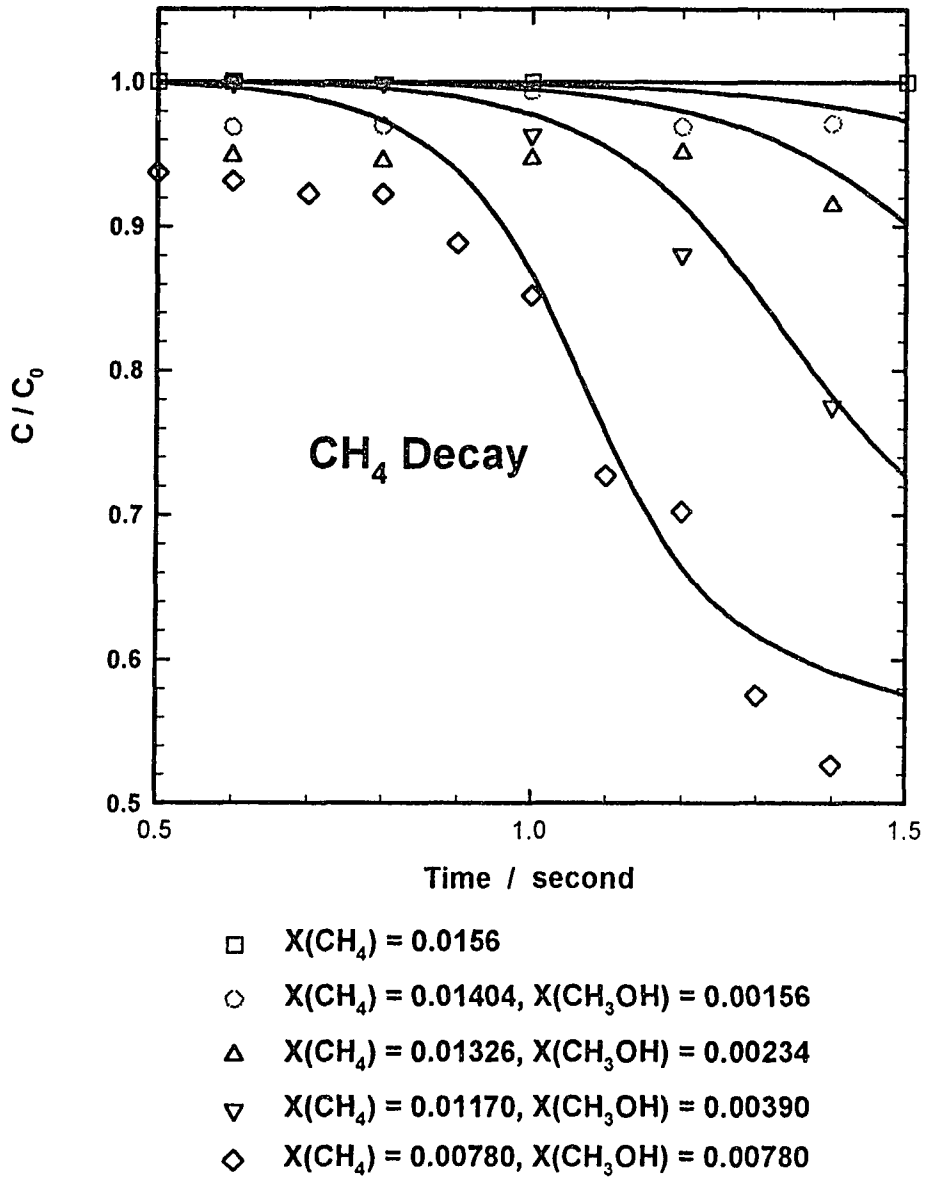


Figure B.107 Comparison of Methane Decay of Model and Experimental Results for Methane/Methanol Oxidation at 873 K, 1 atm, $X_{o,TOTAL} = 1.56$, $\phi = 1.0$

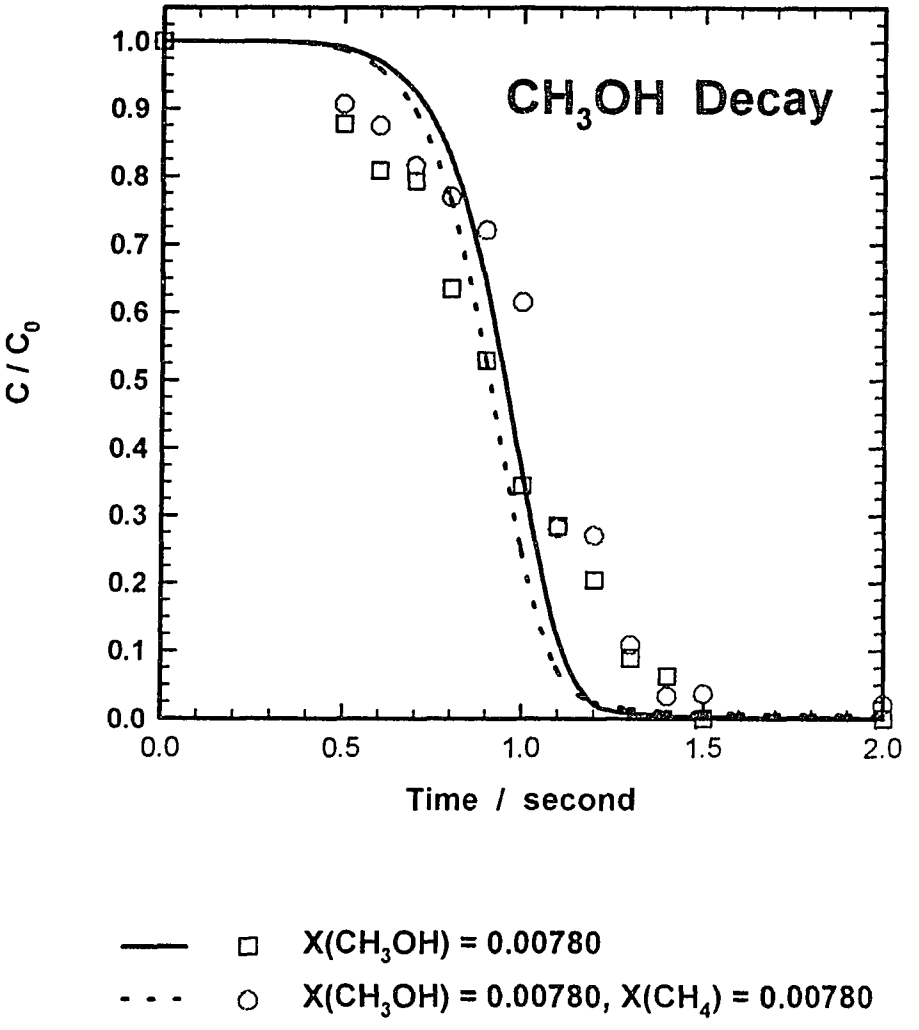


Figure B.108 Comparison of Methanol Decay of Model and Experimental Results for Methanol and Methane/Methanol Oxidation at 873 K, 5 atm, $\phi = 1.0$

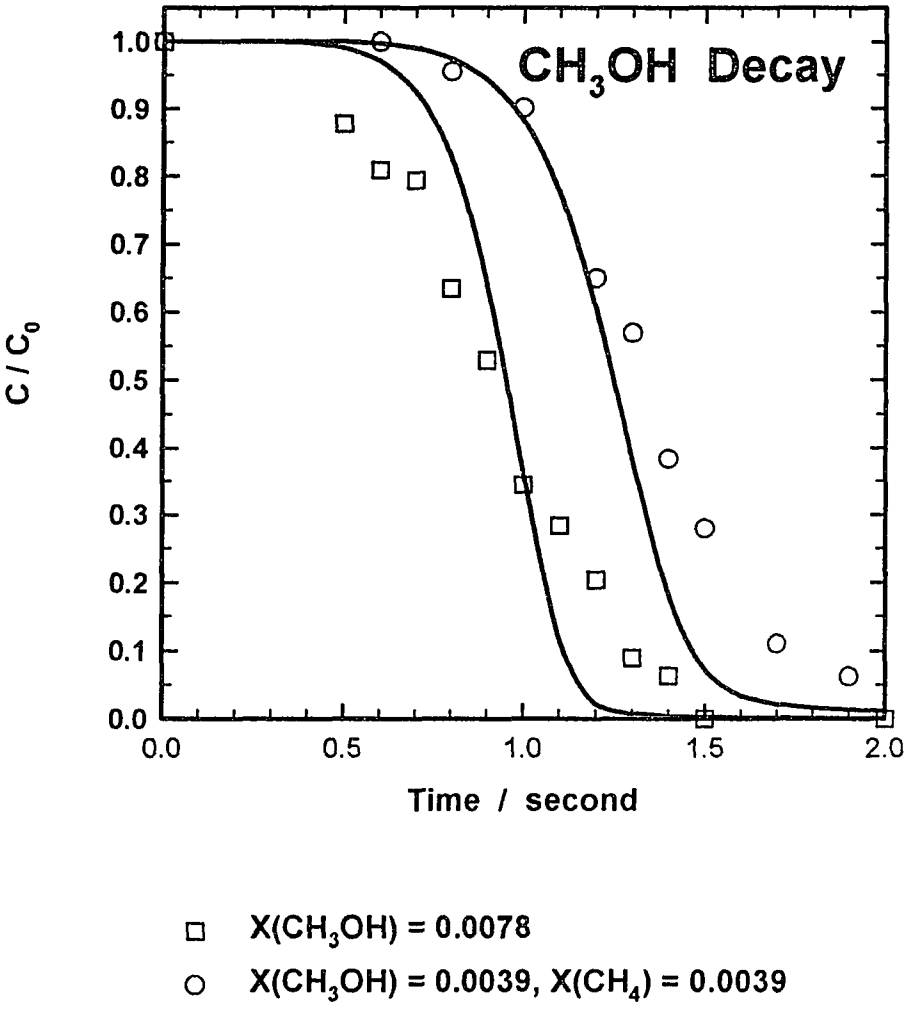


Figure B.109 Comparison of Methanol Decay of Model and Experimental Results for Methanol and Methane/Methanol Oxidation at 873 K, $X_{o,TOTAL}$ = 0.78%, 5 atm, ϕ = 1.0

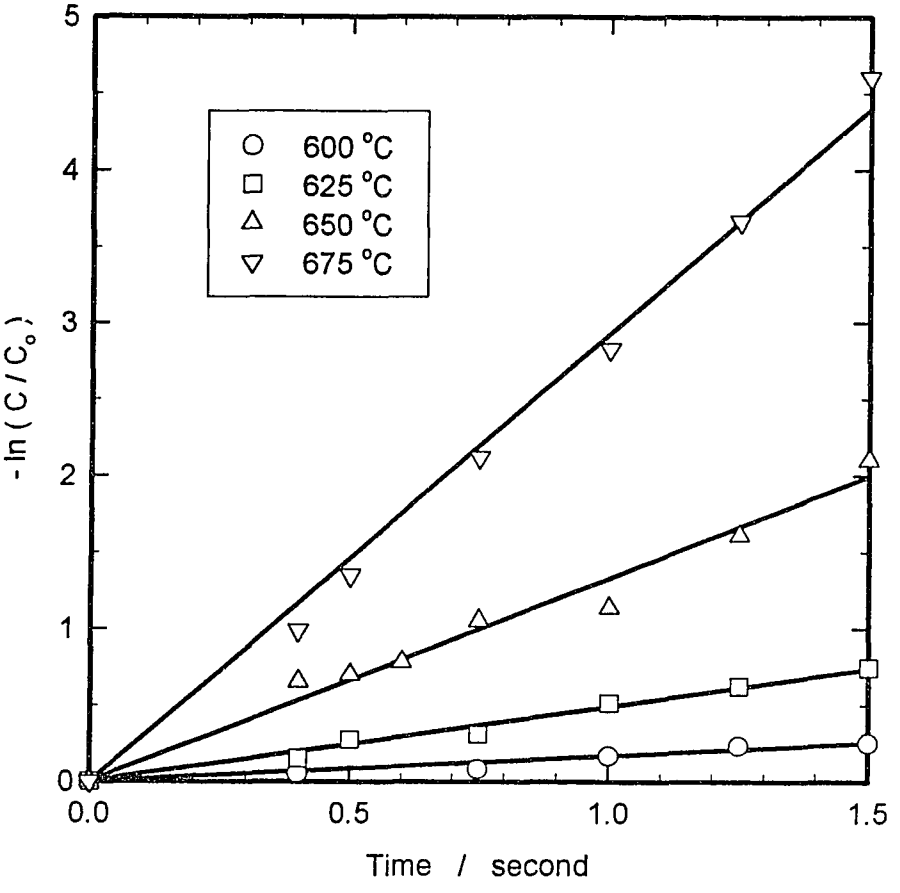


Figure B.110 MTBE Pyrolysis, C_o = 0.5%, -ln(C/C_o) versus Retention Time

Potential Energy Diagram for $(CH_3)_3COCH_3$ Dissociation

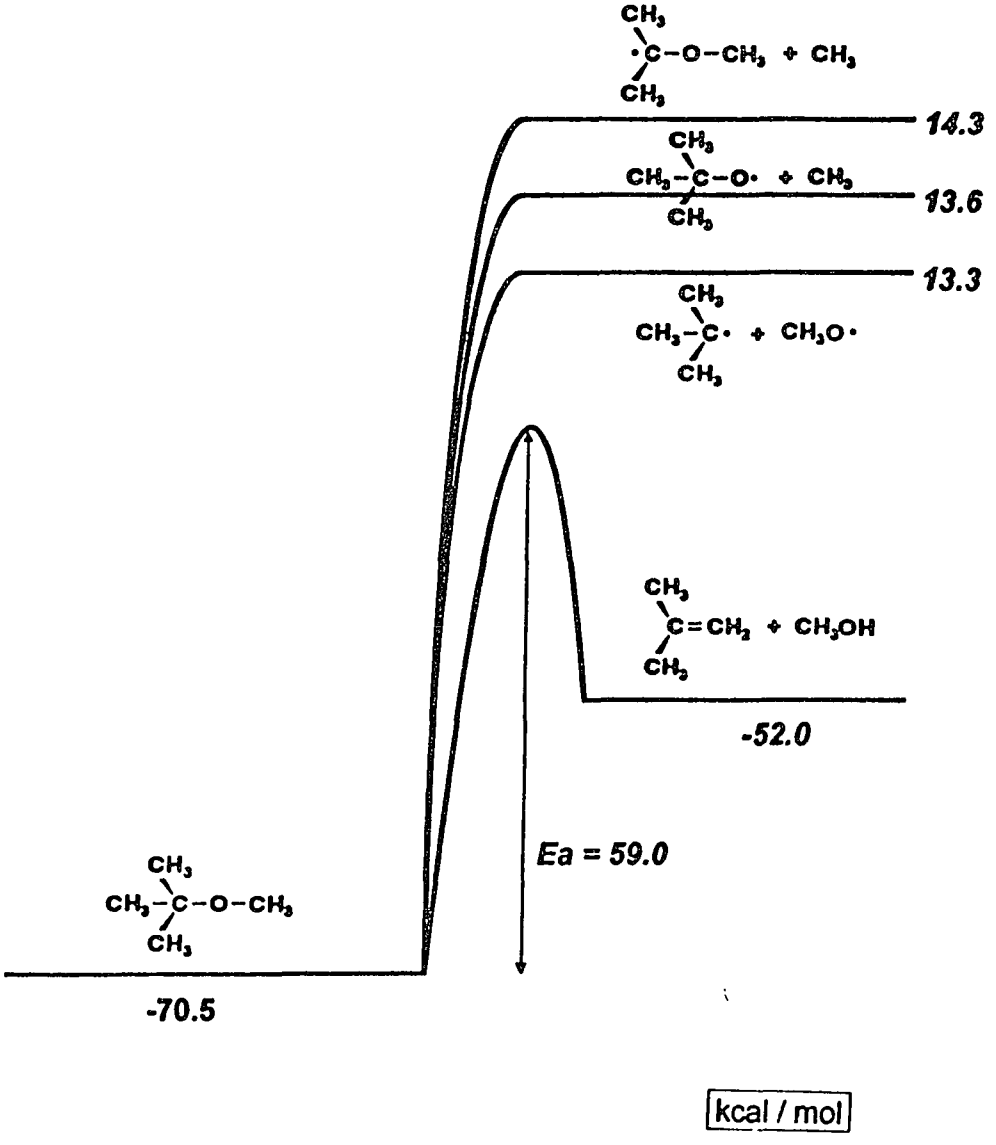


Figure B.111 Potential Energy Diagram for MTBE → Products

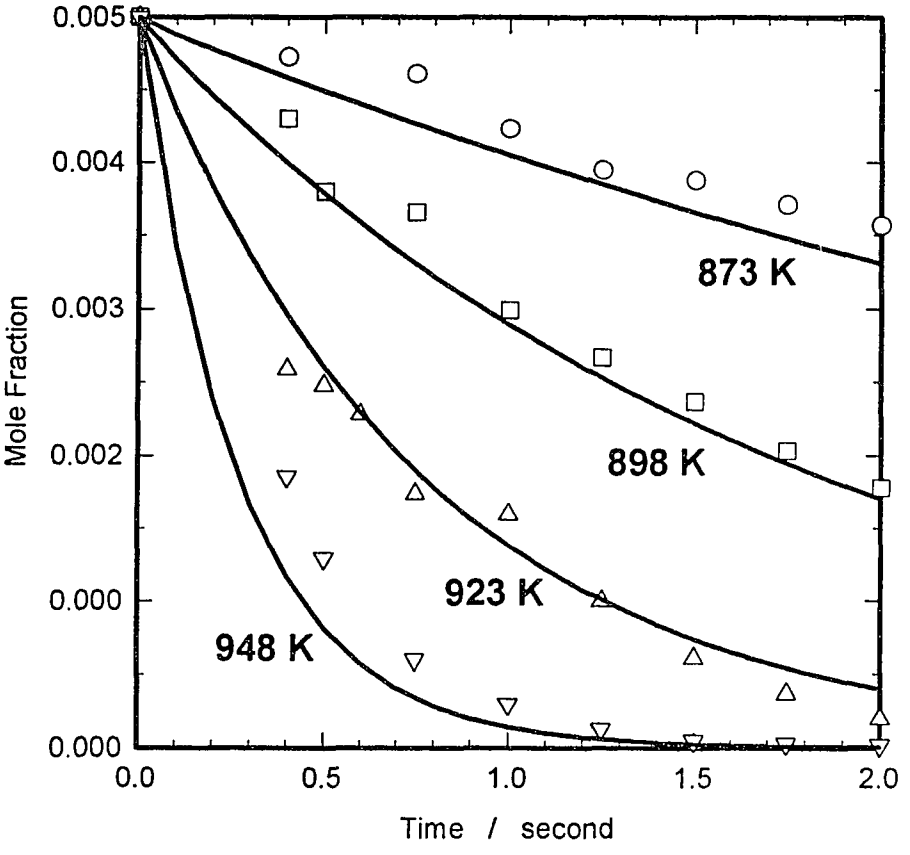


Figure B.112 Comparisons of Experimental and Model Results for MTBE Pyrolysis, $C_o = 0.5\%$

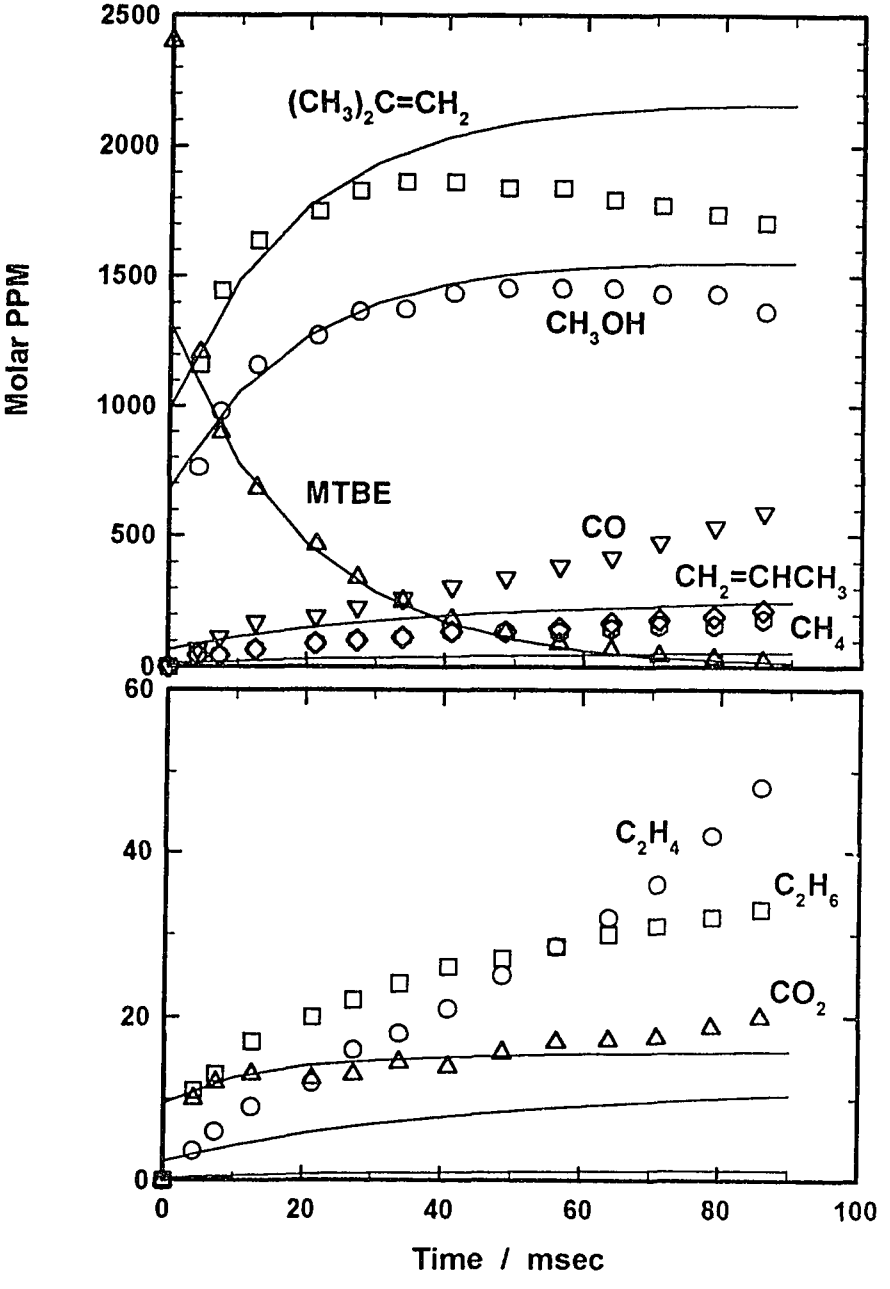
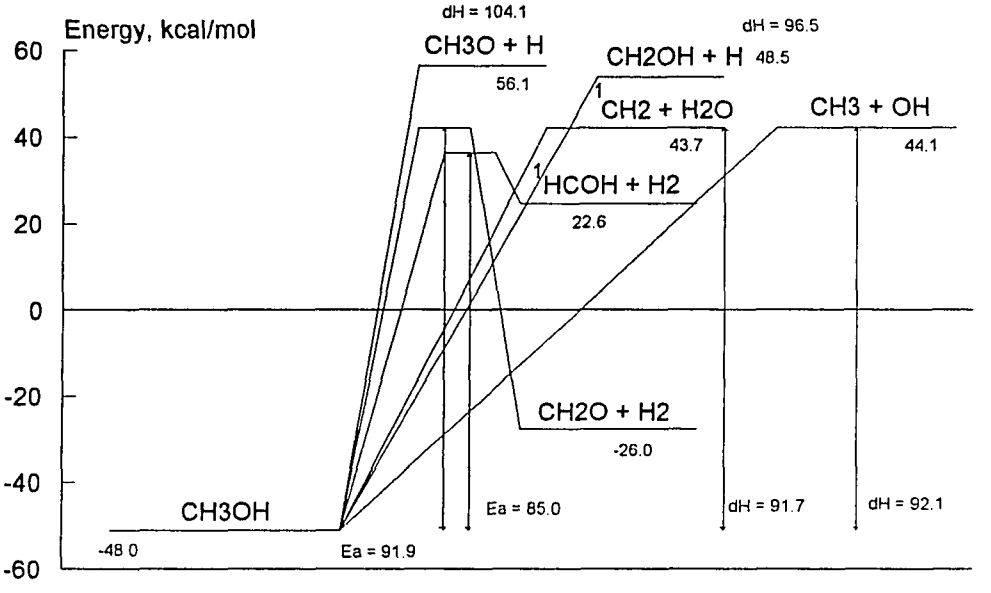


Figure B.113 Model Comparison with Norton's Experimental Data for MTBE Oxidation at 1024 K, 1 atm, $\phi = 0.96$ (Model Result 10 msec offset)

APPENDIX C

QRRK INPUT PARAMETERS

C.1 $CH_3OH \rightarrow$ Products



	reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_3 + OH \rightarrow CH_3OH$	2.62E+13	0.0
-1	$CH_3OH \rightarrow CH_3 + OH$	6.30E+15	91.8
2	$CH_3OH \rightarrow {}^1HCOH + H_2$	1.09E+14	85.0
3	$CH_3OH \rightarrow {}^1CH_2 + H_2O$	4.40E+14	89.8
4	$CH_3OH \rightarrow CH_2O + H_2$	1.27E+13	91.9
5	$CH_3OH \rightarrow CH_2OH + H$	1.30E+15	95.0
6	$CH_3OH \rightarrow CH_3O + H$	2.39E+14	103.4

frequencies/degenercies (from CPFIT): 738 cm^{-1}/3.049,
1729 cm^{-1}/5.932, 3695 cm^{-1}/2.518
Lennard-Jones parameters: σ = 3.78 Å, ϵ/k = 377 K

k_1 Microscopic Reversibility (MR).

k_{-1} A_{-1} from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63. Ea_1 from R. Humpfer et al. Twenty-fiftth Sumposium on Combustion, 1994, 721.

k_2 A_2 from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63. Ea_2 from R. Humpfer et al. Twenty-fiftth Sumposium on Combustion, 1994, 721.

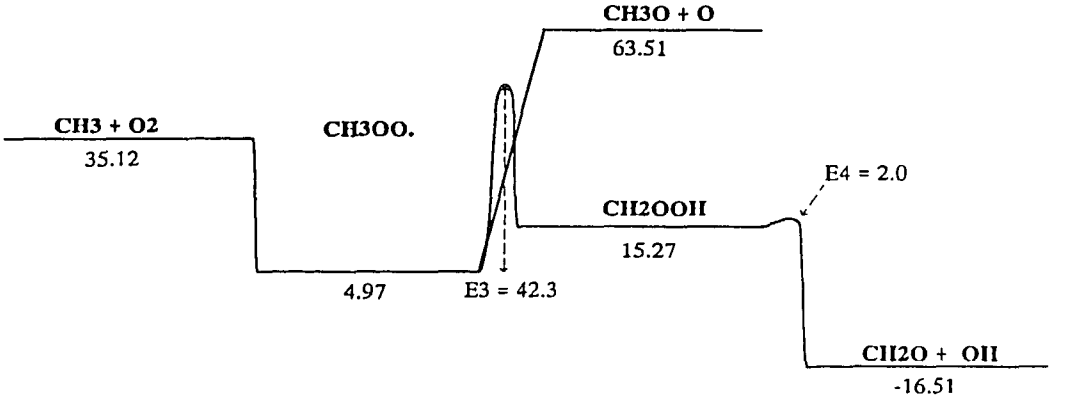
k_3 A_3 from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63. Ea_3 from R. Humpfer et al. Twenty-fiftth Sumposium on Combustion, 1994, 721.

k_4 A_4 from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63. Ea_4 from R. Humpfer et al. Twenty-fiftth Sumposium on Combustion, 1994, 721.

k_5 from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63.

k_6 from A. M. Dean, J. W. Bozzelli and E. R. Ritter (1991), Combust. Sci. Technol. 80, 63.

C.2 $CH_3 + O_2 \rightarrow [CH_3OO\bullet]^* \rightarrow$ Products



Kcal/mol

	reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_3 + O_2 \rightarrow CH_3OO\cdot$	$6.44E+10\ T^{0.53}$	0.0
-1	$CH_3OO\cdot \rightarrow CH_3 + O_2$	$3.09E+15\ T^{-0.33}e^{-.00207T}$	30.2
2	$CH_3OO\cdot \rightarrow CH_3O + O$	3.10E+15	58.5
3	$CH_3OO\cdot \rightarrow CH_2OOH$	6.00E+12	42.3
-3	$CH_2OOH \rightarrow CH_3OO\cdot$	$1.60E+18\ T^{-2.49}e^{+.00207T}$	32.0
4	$CH_2OOH \rightarrow CH_2O + OH$	5.50E+13	2.0

frequencies/degen (from CPFIT): 525 cm^{-1}/3.204, 801 cm^{-1}/2.752, 1912 cm^{-1}/5.544
Lennard-Jones parameters: σ = 4.36 Å, ϵ/k = 471 K

k_1 Taken from Cobos et al., *J. Phys. Chem.*, **89**, 4332 (1985).

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 1500 K.

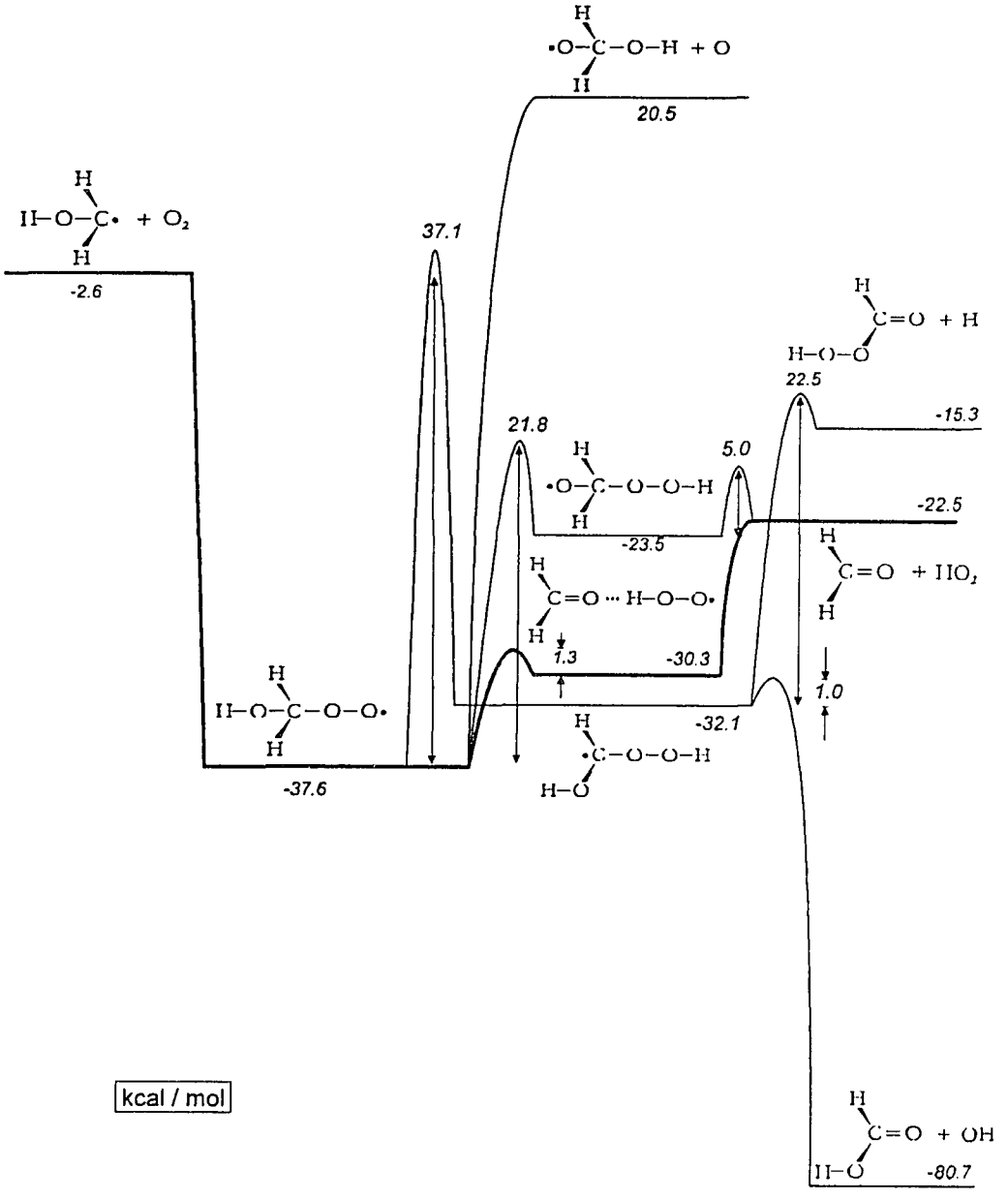
k_2 A_2 from MR with A_{-3} = 5.00E+13 for the high-pressure recombination rate constant.

k_3 A_3 = 6.00E+12 as Benson's for H-transfer, $Ea_3 = \triangle H_3$ + ring strain (26.0) + $E_{abstraction}$ (6.0)

k_{-3} MR with the temperature range of 298 K and 2000 K.

k_4 A_4 from MR with reverse taken as the addition of OH to CH_3CHO, Semmes et al.; Ea_4 = 2.0 from the intrinsic activation energy expected for OH addition.

C.3 $CH_2OH + O_2 \leftrightarrow [H_2(OH)COO\bullet]^* \rightarrow$ Product



	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	E_a (kcal/mol)
1	$CH_2OH + O_2 \rightarrow H_2(OH)COO\bullet$	6.00E+12	0.0
-1	$H_2(OH)COO\bullet \rightarrow CH_2OH + O_2$	$2.93E+16\ T^{-0.54}e^{-0.00118T}$	34.4
2	$H_2(OH)COO\bullet \rightarrow H_2(OH)CO\bullet + O$	7.55E+14	57.5
3	$H_2(OH)COO\bullet \rightarrow H_2CO..HOO\bullet$	$4.96E+6\ T^{2.11}e^{-0.00069T}$	8.6
-3	$H_2CO..HOO\bullet \rightarrow H_2(OH)COO\bullet$	$3.76E+7\ T^{1.0}$	1.3
4	$H_2CO..HOO\bullet \rightarrow H_2C{=}O + HO_2$ (I)	$5.87E+17\ T^{-2.68}e^{-0.00007T}$	7.2
5	$H_2(OH)COO\bullet \rightarrow CH_2(OOH)O\bullet$	$6.86E+8\ T^{1.0}$	21.8
-5	$CH_2(OOH)O\bullet \rightarrow H_2(OH)COO\bullet$	$5.56E+8\ T^{0.84}e^{0.00042T}$	7.7
6	$CH_2(OOH)O\bullet \rightarrow H_2C{=}O + HO_2$ (II)	1.40E+14	5.0
7	$H_2(OH)COO\bullet \rightarrow C\bullet H(OOH)OH$	$5.90E+9\ T^{1.0}$	37.1
-7	$C\bullet H(OOH)OH \rightarrow H_2(OH)COO\bullet$	$4.19E+9\ T^{0.8}e^{0.00064T}$	31.6
8	$C\bullet H(OOH)OH \rightarrow H(OH)C{=}O + OH$	3.31E+13	1.0
9	$C\bullet H(OOH)OH \rightarrow H(OOH)C{=}O + H$	3.24E+13	22.5

frequencies/degenercies (from CPFIT): 415 cm^{-1}/6.178, 1506 cm^{-1}/5.998, 3198 cm^{-1}/2.824
Lennard-Jones parameters: σ = 4.83 Å, ϵ/k = 488 K
$\langle \Delta E \rangle_{down}$ = 800 cal/mol, Bath gas = N_2

k_1 A_1 taken as double of $CC\cdot + O_2$ addition from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 A_2 from MR with A_{-2} = 2.00E+13 based on addition of O to CH_3

k_3 $A_3 = (10^{13.55})(10^{\Delta S\#/4.6})$, $\Delta S^{\#}$ = -6.4; $Ea_3 = \Delta H_3$ + 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_{-3} A_{-3} = (ekT/h)[exp($\Delta S^{\#}$/R)], $\Delta S^{\#}$ = -14.4; Ea_{-3} = 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_4 Based on the study of $CH_2O + HO_2$ system in this paper.

k_5 A_5 = (ekT/h)[exp($\Delta S^{\#}$/R)], ΔS = -8.6 (loss of 2 rotors); $Ea_5 = \Delta H_5$ + ring strain (4.7) + $E_{abstraction}$ (3.0), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-5} MR with the temperature range of 298 K and 2000 K.

k_6 A_6 from MR with A_{-6} = 6.05E+11 and $Ea_{-6} = \Delta H_{-6}$ + 5.0, based on addition of HO_2 to C_2H_4 by Tsang's recommendation, J. Phys. Chem. Ref. Data, 1987.

k_7 A_7 = (ekT/h)[exp($\Delta S^{\#}$/R)](2), ΔS = -4.3 (loss of 1 rotors); $Ea_7 = \Delta H_7$ + ring strain (23) + $E_{abstraction}$ (8.6), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-7} MR with the temperature range of 298 K and 2000 K.

k_8 A_8 from MR with A_{-8} = 2.7E+12, which is one half of the rate constant for addition of OH to C_2H_4, Atkinson *et al.*, J. Phys. Chem., 1989; Ea_8 from Soto and Page, Chem. Phys. 1991.

k_9 A_9 from MR with A_{-9} = 1.46E+13, which is one half of the rate constant for addition of H to C_2H_4 and $Ea_9 = \triangle H_9 + 2.7$, based on NIST fitting.

C.4 $CH_2O + HO_2 \leftrightarrow [CH_2O...HOO\bullet]^* \rightarrow$ Product

	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	E_a (kcal/mol)
1	$H_2C=O + HO_2 \rightarrow H_2CO..HOO\bullet$	5.00E+10	0.0
-1	$H_2CO..HOO\bullet \rightarrow H_2C=O + HO_2$	$5.87E+17\ T^{-2.68}e^{-0.00007T}$	7.2
2	$H_2CO..HOO\bullet \rightarrow H_2(OH)COO\bullet$	$3.76E+7\ T^{1.0}$	1.3
-2	$H_2(OH)COO\bullet \rightarrow H_2CO..HOO\bullet$	$4.96E+6\ T^{2.11}e^{-0.00069T}$	8.6
3	$H_2(OH)COO\bullet \rightarrow CH_2OH + O_2$	9.00E+14	34.4
4	$H_2(OH)COO\bullet \rightarrow H_2(OH)CO\bullet + O$	7.55E+14	57.5
5	$H_2(OH)COO\bullet \rightarrow CH_2(OOH)O\bullet$	$6.86E+8\ T^{1.0}$	21.8
-5	$CH_2(OOH)O\bullet \rightarrow H_2(OH)COO\bullet$	$5.56E+8\ T^{0.84}e^{0.00042T}$	7.7
6	$CH_2(OOH)O\bullet \rightarrow CH_2O + HO_2$	1.40E+14	5.0
7	$H_2(OH)COO\bullet \rightarrow C\bullet H(OOH)OH$	$5.90E+9\ T^{1.0}$	37.1
-7	$C\bullet H(OOH)OH \rightarrow H_2(OH)COO\bullet$	$4.19E+9\ T^{0.8}e^{0.00064T}$	31.6
8	$C\bullet H(OOH)OH \rightarrow H(OH)C=O + OH$	3.31E+13	1.0
9	$C\bullet H(OOH)OH \rightarrow H(OOH)C=O + H$	3.24E+13	22.5

frequencies/degenercies (from CPFIT): 188 cm^{-1}/5.857, 1308 cm^{-1}/5.714, 2952 cm^{-1}/3.430
Lennard-Jones parameters: σ = 4.83 Å, ϵ/k = 488 K
$\langle \Delta E \rangle_{down}$ = 800 cal/mol, Bath gas = N_2

k_1 This study, estimated from the reaction of $CH_2O+HO_2 = CQ \cdot H_2OH$, Veyret *et al.*, J. Phys. Chem., 1989.

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 A_2 = (ekT/h)[exp($\triangle S^{\#}$/R)], $\triangle S^{\#}$ = -14.4; Ea_2 = 2.0 by transition state calculation, transition state thermo properties based on Evleth *et al.*, J. Phys. Chem., 1993.

k_{-2} MR with the temperature range of 298 K and 2000 K.

k_3 A_3 from MR with A_{-3} taken as double of $CC \cdot + O_2$ addition from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_4 A_4 from MR with A_{-4} = 2.00E+13 based on addition of O to CH_3.

k_5 A_5 = (ekT/h)[exp($\triangle S^{\#}$/R)], $\triangle S$ = -8.6 (loss of 2 rotors); $Ea_5 = \triangle H_5$ + ring strain (4.7) + $E_{abstraction}$ (3.0), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-5} MR with the temperature range of 298 K and 2000 K.

k_6 A_6 from MR with A_{-4} = 6.05E+11 and $Ea_{-6} = \triangle H_{-4}$ + 5.0, based on addition of HO_2 to C_2H_4 by Tsang's recommendation, J. Phys. Chem. Ref. Data, 1987.

k_7 $A_7 = (ekT/h)[\exp(\triangle S^{\#}/R)](2)$, $\triangle S = -4.3$ (loss of 1 rotors); $Ea_7 = \triangle H_7$ + ring strain (23) + $E_{abstraction}$ (8.6), ring strain from J. W. Bozzelli and A. M. Dean, J. Phys. Chem., 1990.

k_{-7} MR with the temperature range of 298 K and 2000 K.

k_8 A_8 from MR with A_{-8} = 2.7E+12, which is one half of the rate constant for addition of OH to C_2H_4, Atkinson *et al.*, J. Phys. Chem., 1989; Ea_8 from Soto and Page, Chem. Phys. 1991.

k_9 A_9 from MR with A_{-9} = 1.46E+13, which is one half of the rate constant for addition of H to C_2H_4 and $Ea_7 = \triangle H_7 + 2.7$, based on NIST fitting.

C.5 $CH_2OH + HO_2 \leftrightarrow [CQH_2OH]^* \rightarrow$ Product

	reation	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_2OH + HO_2 \rightarrow CQH_2OH$	2.20E+13	0.0
-1	$CQH_2OH \rightarrow CH_2OH + HO_2$	3.56E+16	76.1
2	$CQH_2OH \rightarrow CH_2OHO\cdot + OH$	2.54E+15	45.7
3	$CQH_2OH \rightarrow CQ\cdot H_2OH + H$	3.91E+13	88.2

frequencies (from CPFIT): 1693.6 cm^{-1}
Lennard-Jones parameters: σ = 4.49 Å, ϵ/k = 217.1 K

k_1 A_1 taken as the average of $C\cdot$ and $CCC\cdot + HO_2$ addition from Tsang 86'.

k_{-1} MR

k_2 A_2 from MR with A_{-2} = 2.41E+13, which is the rate constant for addition of HO_2 to $CCC\cdot$, Tsang 88', $Ea_2 = \triangle H_2$

k_3 A_3 from MR with A_{-3} = 4.0E+13, based on H addition to C_2H_5, NIST, and $Ea_3 = \triangle H_3$.

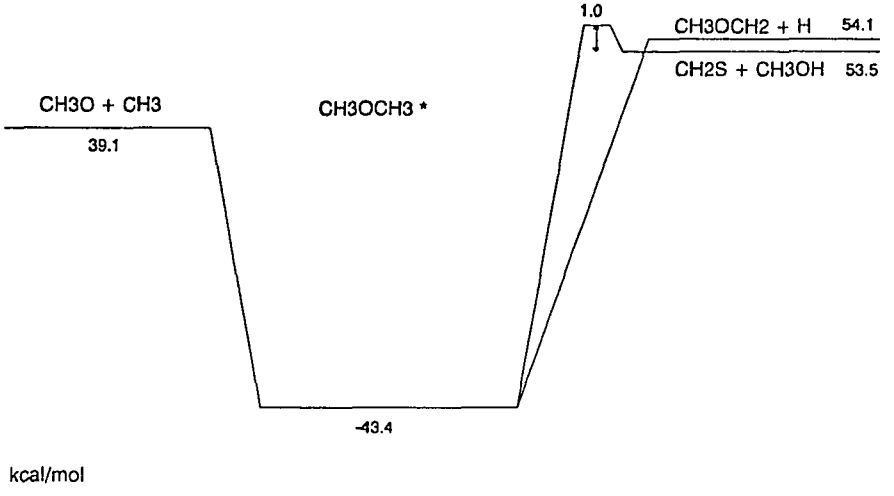
C.6 $CH_2O \rightarrow$ Product

	reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_2OH \rightarrow CH_2O + H$	2.63E+13	33.480

frequencies (from CPFIT): 1414.5 cm^{-1}
Lennard-Jones parameters: σ = 4.27 Å, ϵ/k = 454.0 K

k_1 A_1 from MR with A_{-1} as 1/2 of H addition to C=C and $Ea_1 = \triangle H + 2.5$ kcal/mol, Dean's suggestion for high pressure limit

C.7 $CH_3O + CH_3 \leftrightarrow [CH_3OCH_3]^* \rightarrow$ Product



	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_3O + CH_3 \rightarrow CH_3OCH_3$	1.21E+13	0.0
-1	$CH_3OCH_3 \rightarrow CH_3O + CH_3$	$1.05E+17\ T^{-0.19}e^{-0.00144T}$	82.5
2	$CH_3OCH_3 \rightarrow CH_3OCH_2 + H$	2.04E+15	97.5
3	$CH_3OCH_3 \rightarrow {}^1CH_2 + CH_3OH$	2.11E+16	97.3

frequencies/degen (from CPFIT): 622 cm^{-1}/7.659, 1789 cm^{-1}/9.169, 3598 cm^{-1}/4.172
Lennard-Jones parameters: σ = 4.31 Å, ϵ/k = 395 K

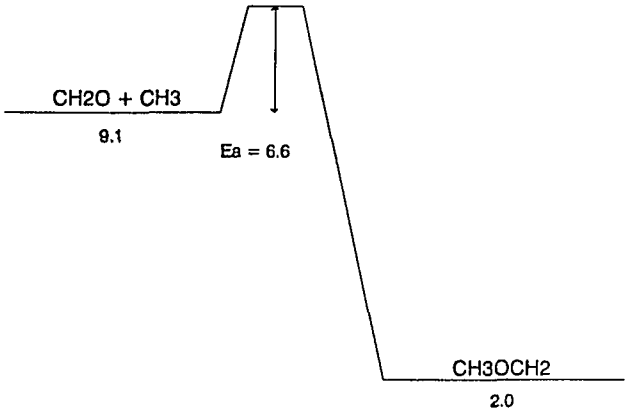
k_1 from Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 based on k_{-2} = 6.40E+13.

k_3 based on A_{-3} = (1.00E+14)*1/4 and Ea_{-3} = 1.0 kcal/mol.

C.8 $CH_2O + CH_3 \leftrightarrow [CH_3OCH_2]^* \rightarrow$ Product



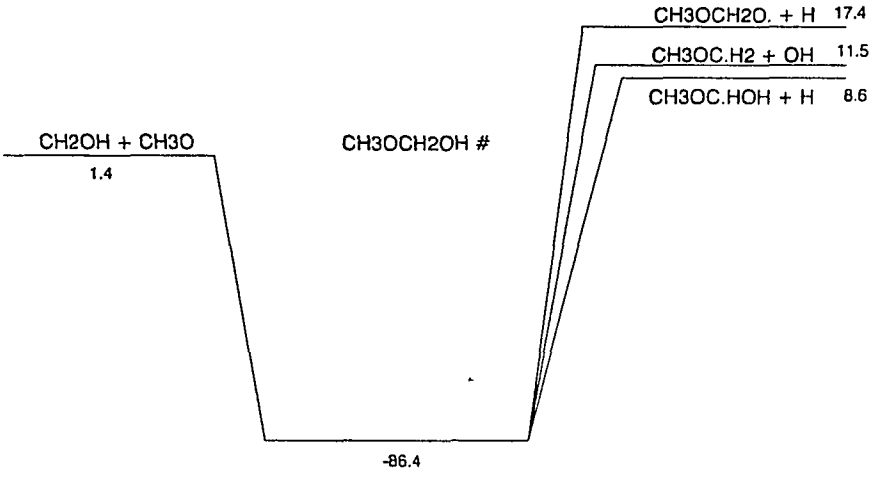
	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_2O + CH_3 \rightarrow CH_3OCH_2$	5.19E+11	6.6
-1	$CH_3OCH_2 \rightarrow CH_2O + CH_3$	$8.69E+17\ T^{-1.46}e^{-0.00164T}$	13.7

frequencies/degen (from CPFIT): 631 cm^{-1}/7.878, 1899 cm^{-1}/7.530, 3593 cm^{-1}/2.592
Lennard-Jones parameters: σ = 4.31 Å, ϵ/k = 395 K

k_1 based on CH_3 + CO addition from Anastasi et al., *J. Chem. Soc. Faraday Trans. 1*, **878**, 2423 (1982).

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

C.9 $CH_2OH + CH_3O \leftrightarrow [CH_3OCH_2OH]^* \rightarrow$ Product

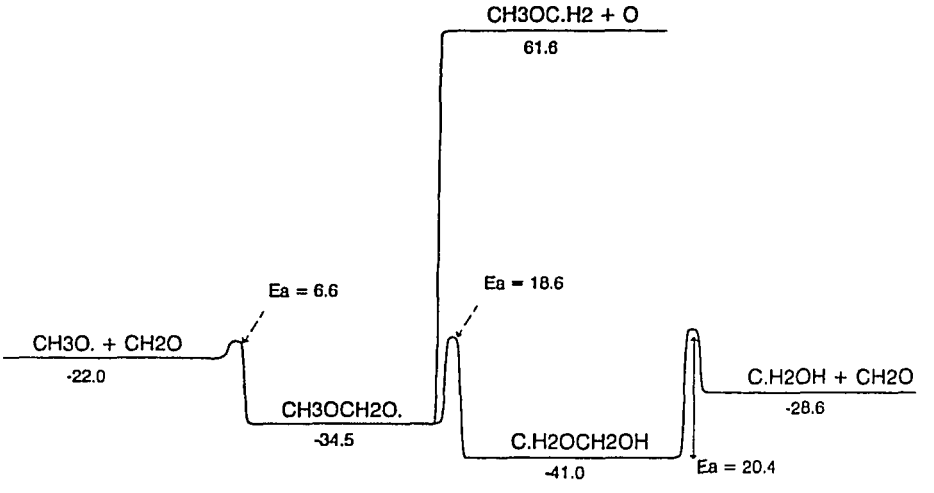


	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_2OH + CH_3O \rightarrow CH_3OCH_2OH$	1.21E+13	0.0
-1	$CH_3OCH_2OH \rightarrow CH_2OH + CH_3O$	$6.01E+17\ T^{-0.66}e^{-0.00077T}$	87.8
2	$CH_3OCH_2OH \rightarrow CH_3OC \cdot H(OH) + H$	5.70E+14	90.6
3	$CH_3OCH_2OH \rightarrow CH_3OC \cdot H_2 + OH$	4.84E+15	92.8
4	$CH_3OCH_2OH \rightarrow CH_3OCO \cdot + H$	5.34E+14	100.5

frequencies/degen (from CPFIT): 489 cm^{-1}/8.260, 1626 cm^{-1}/10.330, 3600 cm^{-1}/5.410
Lennard-Jones parameters: σ = 4.90 Å, ϵ/k = 415 K

k_1 from $CH_3 + CH_3O$, Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).
k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.
k_2 based on k_{-2} = 1.00E+14.
k_3 based on k_{-3} = 2.11E+13 from NIST fitting for $CH_3 + OH$.
k_4 based on k_{-4} = 2.00E+14.

C.10 $CH_3O + CH_2O \leftrightarrow [CH_3OCH_2O\cdot]^* \rightarrow$ Product



	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$CH_3O + CH_2O \rightarrow CH_3OCH_2O\cdot$	5.19E+11	6.6
-1	$CH_3OCH_2O\cdot \rightarrow CH_3O + CH_2O$	1.35E+24 $T^{-2.86}e^{-.00002T}$	15.7
2	$CH_3OCH_2O\cdot \rightarrow CH_3OC\cdot H_2 + O$	4.83E+15	96.0
3	$CH_3OCH_2O\cdot \rightarrow C\cdot H_2OCH_2OH$	2.08E+9 $T^{1.0}$	15.9
-3	$C\cdot H_2OCH_2OH \rightarrow CH_3OCH_2O\cdot$	2.74E+11 $T^{-0.14}e^{+.0008T}$	22.5
4	$C\cdot H_2OCH_2OH \rightarrow CH_2OH + CH_2O$	4.31E+12	19.8

frequencies/degen (from CPFIT): 558 cm^{-1}/7.863, 1731 cm^{-1}/9.557, 3599 cm^{-1}/3.580
Lennard-Jones parameters: σ = 4.61 Å, ϵ/k = 415 K

k_1 based on CH_3 + CO addition from Anastasi et al., *J. Chem. Soc. Faraday Trans. 1*, **878**, 2423 (1982).

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

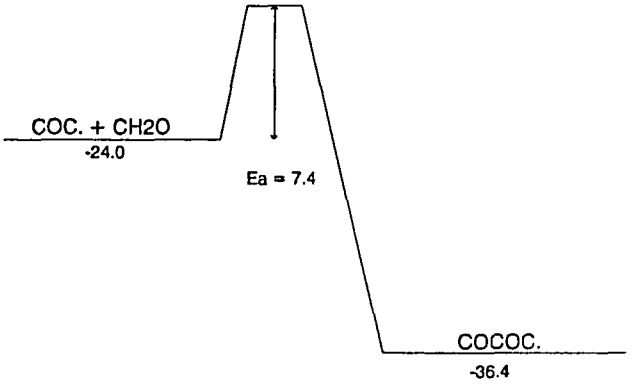
k_2 MR from k_{-2} = 7.00E+13 taken as CH_3 + O, Dean, 1987.

k_3 $A_3 = (ekT/h)[\exp(\triangle S^{\#}/R)](3) = (10^{10.71}T)(10^{\triangle S\#/4.6})(3)$, $\triangle S^{\#}$ = -8.6(loss of 2 rotors); Ea_3 = ring strain (5.9) + $E_{abstraction}$ (10).

k_{-3} MR with the temperature range of 298 K and 2000 K.

k_4 MR, A_{-4} = 1.06E+11 taken as one half of CH_3 addition to C_2H_4 and Ea_{-4} = 7.4 from Baulch et.al., 1992.

C.11 $COC\cdot + CH_2O \leftrightarrow [COCOC\cdot]^* \rightarrow$ Product



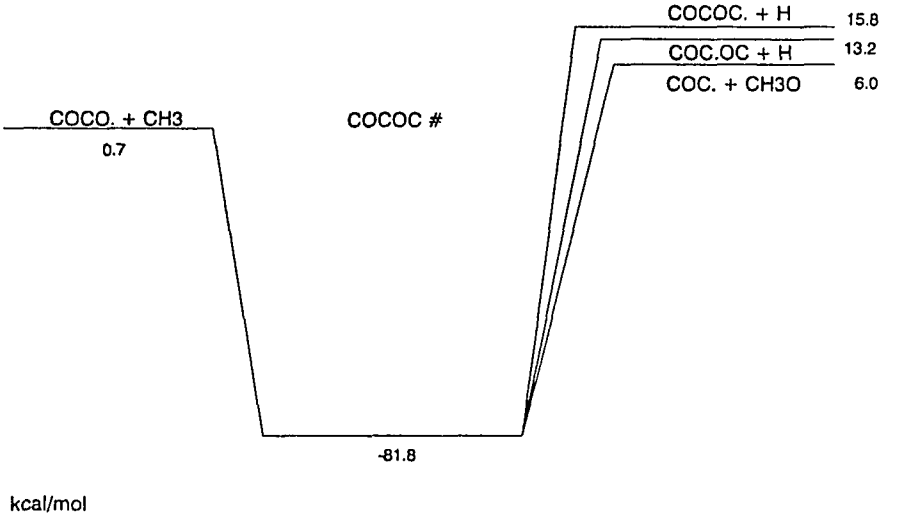
	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$COC\cdot + CH_2O \rightarrow COCOC\cdot$	1.06E+11	7.4
-1	$COCOC\cdot \rightarrow COC\cdot + CH_2O$	$1.91E+20\ T^{-2.78}e^{-0.00002T}$	19.1

frequencies/degen (from CPFIT): 631 cm^{-1}/7.878, 1899 cm^{-1}/7.530, 3593 cm^{-1}/2.592
Lennard-Jones parameters: $\sigma = 4.31$ Å, $\epsilon/k = 395$ K

k_1 A_1 taken as one half of CH_3 addition to C_2H_4 and $Ea_1 = 7.4$ from Baulch et.al., 1992.

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

C.12 COCO· + CH_3 → ↔ [COCOC]* → Product



	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	COCO· + CH_3 → COCOC	1.21E+13	0.0
-1	COCOC → COCO· + CH_3	5.81E+15	82.5
2	COCOC → COC· + CH_3O	5.65E+15	87.8
3	COCOC → COC·OC + H	3.06E+15	94.4
4	COCOC → COCOC· + H	2.00E+16	97.6

frequencies/degen (from CPFIT): 566 cm^{-1}/14.05, 1802 cm^{-1}/13.09, 3598 cm^{-1}/5.86
Lennard-Jones parameters: σ = 5.28 Å, ϵ/k = 379 K

k_1 Taken as for $CH_3O + CH_3$ → COC from Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).

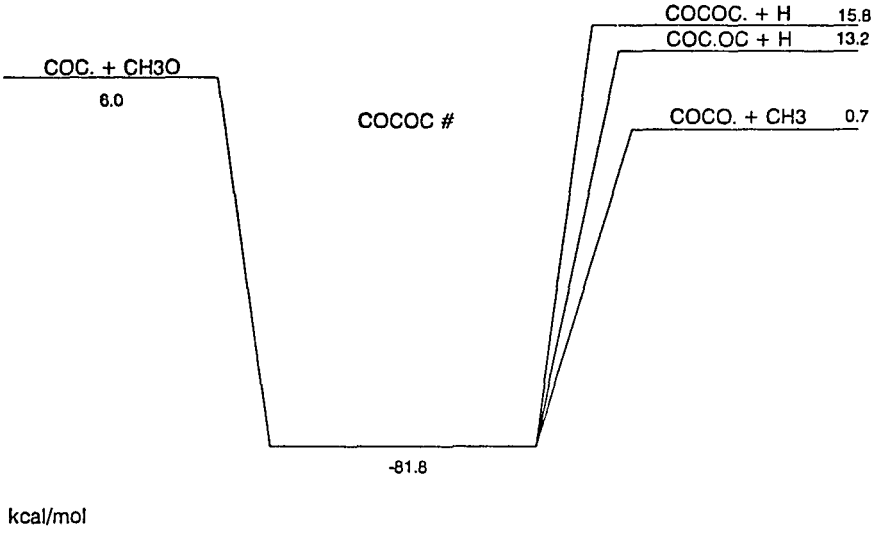
k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 MR with k_{-2} = 1.21E+13 as $CH_3O + CH_3$ → COC from Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).

k_3 based on k_{-3} = 1.00E+14.

k_4 based on k_{-4} = 6.40E+13.

C.13 $COC\cdot + CH_3O \rightarrow \leftrightarrow [COCOC]^* \rightarrow$ Product



	Reaction	A (s^{-1} or $cm^3/(mol*s)$)	Ea (kcal/mol)
1	$COC\cdot + CH_3O \rightarrow COCOC$	1.21E+13	0.0
-1	$COCOC \rightarrow COC\cdot + CH_3O$	5.65E+15	87.8
2	$COCOC \rightarrow COCO\cdot + CH_3$	5.81E+15	82.5
3	$COCOC \rightarrow COC\cdot OC + H$	3.06E+15	94.4
4	$COCOC \rightarrow COCOC\cdot + H$	2.00E+16	97.6

frequencies/degen (from CPFIT): 566 cm^{-1}/14.05, 1802 cm^{-1}/13.09, 3598 cm^{-1}/5.86
Lennard-Jones parameters: σ = 5.28 Å, ϵ/k = 379 K

k_1 Taken as $CH_3O + CH_3 \rightarrow COC$ from Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).

k_{-1} Microscopic Reversibility (MR) with the temperature range of 298 K and 2000 K.

k_2 MR with k_{-2} = 1.21E+13 as $CH_3O + CH_3 \rightarrow COC$ from Tsang et al., *J. Phys. Chem.*, **15**, 1087 (1986).

k_3 based on k_{-3} = 1.00E+14.

k_4 based on k_{-4} = 6.40E+13.

APPENDIX D

FORTRAN PROGRAMS

D.1 THERMCAL

```
C
      PROGRAM THERMCAL
C
C-----------------------------------------------------------------------C
c***************changed W.Ing 06-15-94, 08-01-95**********************
C
C            thermodynamic properties of species
      IMPLICIT NONE
C
      include 'thm_par.fi'
      include 'thm_cfg.fi'
C
      CHARACTER comment(Max_Group)*70,
     2          specy_name*80, specy_formula*40,
     3          specy_group*80,
     4          specy_element(Max_Element)*4,
     5          specy_comment*70, specy_phase, line*200, KEY(5)*3,
     6          SUB(20)*100, UPCASE*4, str*100, qty*100
C
      INTEGER  specy_group_qty, specy_symmetry,specy_len,
     1         specy_element_qty(Max_Element),Nloop,ILASCH,
     2         ILEN, IPPLEN, NSUB, NKEY, Isymmetry(5), NVAL, Ierr,
     3         I, Istr, Igroup, Ifirch, Islash, Iqty(5), Ifound, M,
     4         Irotor

      INTEGER*2 year,month,day
C
      REAL*8 specy_Hf, specy_S, specy_Cp300, specy_Cp400, specy_Cp500,
     1       specy_Cp600, specy_Cp800, specy_Cp1000, specy_Cp1500,
     2       specy_rotor, rotor(5)
C
      LOGICAL therm, nasa, init_err, io_err, ErrCp1500, elem_err
C
      DATA KEY/'REF','XXX','XXX','END','!'/
C
C     Initialize variables
C-----------------------------------------------------------------------C
C
C     Call Initialization subroutines, 02/95 ING
C
C      CALL TITLE
c          DO 10 M = 1, max_element
c              specy_element(M) = ' '
c              specy_element_qty(M) = 0
c  10     continue
      CALL THERM_IO(io_err)
      if (io_err) GO TO 6000
      CALL THERM_INIT(init_err)
      if (init_err) GO TO 6000
      CALL GETDAT(year,month,day)
      year = year - 1900
      write(LOG,10)month,day,year
   10 format(' Thermcal log file, date : ',I2,'/',I2,'/',I2)
```

```
C
      write(lst,*)'THERMO'
      write(lst,50)
   50 format(' SPECIES       Hf       S     Cp 300    400     500     60
     10     800    1000     1500     DATE          ELEMENTS')
C      CALL GET_GROUPS
C
      READ (INP,'(A)',END=5000) LINE
  100 CONTINUE
      LINE = ' '
      READ (INP,'(A)',END=5000) LINE
  105 CONTINUE
      ILEN = IPPLEN(LINE)
      IF (ILEN .EQ. 0) GO TO 100
C
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C
C        IS THERE A KEYWORD?
C
      CALL CKCOMP ( UPCASE(SUB(1), 1) , KEY, 5, NKEY)
      IF (NKEY.EQ.5) GO TO 100
      CALL CKCOMP ( UPCASE(SUB(1), 3) , KEY, 4, NKEY)
      IF (NKEY.EQ.4) GO TO 5000
C      WRITE(*,*)UPCASE(SUB(1),3),KEY(1),NKEY
C      IF (NKEY .GT. 0) ITASK = 0
C
      IF (NKEY.EQ.1) THEN
          READ (INP,'(A)',END=5000) LINE
          ILEN = IPPLEN(LINE)
          WRITE(LST,120)LINE(2:ILEN)
  120     FORMAT(1X,A)
C          WRITE(LST,*)LINE
C
      ELSE
C          write(*,*)' group (1-3) = ',group(1),' ',group(2),' ',group(3)
          specy_name = SUB(1)
          specy_formula = SUB(2)
C
          specy_len = IPPLEN(specy_formula)
          DO 150 M = 1, max_element
              specy_element(M) = ' '
              specy_element_qty(M) = 0
  150     continue
c          WRITE(*,*)specy_len,specy_formula(:specy_len)
          call find_elem(specy_formula,specy_len,specy_element,
     1                   specy_element_qty,element,element_qty,
     2                   elem_err,LOG)
          IF (elem_err) THEN
              Ilen = IPPLEN(specy_name)
              write(LOG,160)specy_name(:Ilen)
  160         format(1x,'  for specy  " ',A,' "')
              write(*,170)specy_name(:Ilen)
  170         format(1x,'  specy  "',A,'" ERROR !!!')
              READ (INP,'(A)',END=5000) LINE
              GO TO 100
          ENDIF
C
          call IPPARI (SUB(3), -1, 1, Isymmetry, nval, Ierr, LOG)
          IF (Ierr.NE. 0) THEN
              Ilen = IPPLEN(specy_name)
              WRITE(LOG,310)specy_name(:Ilen)
  310         FORMAT(' "',A,'" input symmetry  ERROR !!!')
              WRITE(*,320)specy_name(:Ilen)
  320         FORMAT(' "',A,'" input symmetry  ERROR !!!')
              READ (INP,'(A)',END=5000) LINE
              GO TO 6000
```

```
            ENDIF
            specy_symmetry = Isymmetry(1)
            call IPPARR (SUB(4), -1, 1, rotor, nval, Ierr, LOG)
            IF (Ierr.NE. 0) THEN
                Ilen = IPPLEN(specy_name)
                WRITE(LOG,330)specy_name(:Ilen)
  330           FORMAT(' "',A,'" input rotor  ERROR !!!')
                WRITE(*,340)specy_name(:Ilen)
  340           FORMAT(' "',A,'" input rotor  ERROR !!!')
                READ (INP,'(A)',END=5000) LINE
                GO TO 6000
            ENDIF
            specy_rotor = rotor(1)
  400       CONTINUE
            READ (INP,'(A)',END=5000) LINE
            ILEN = IPPLEN(LINE)
            IF (ILEN .EQ. 0) GO TO 400
            CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C
            specy_Hf     = 0.0
            specy_S      = 0.0
            specy_Cp300  = 0.0
            specy_Cp400  = 0.0
            specy_Cp500  = 0.0
            specy_Cp600  = 0.0
            specy_Cp800  = 0.0
            specy_Cp1000 = 0.0
            specy_Cp1500 = 0.0
C
            Nloop = INT(NSUB/2.0)
            IF (MOD(NSUB,2).NE. 0) THEN
                WRITE(*,410)specy_name
  410           FORMAT(' Unbalanced groups and quantities for specy : ',A)
                WRITE(LOG,420)specy_name
  420           FORMAT(' Unbalanced groups and quantities for specy : ',A)
                GO TO 100
            ENDIF
            ErrCp1500 = .FALSE.
            DO 500 I = 1, Nloop
C                str = SUB(I)
C                Istr = LEN(str)
C                Igroup = IFIRCH(STR)
C                Islash = INDEX(str,'*')
C                specy_group = str(Igroup:Islash-1)
C                qty = str(Islash+1:Istr)
                str = SUB(I*2-1)
                Istr = LEN(str)
                Igroup = IFIRCH(STR)
                Islash = INDEX(str,' ')
                if (Islash.eq.0) Islash = Istr
                specy_group = str(Igroup:Islash-1)
                qty = SUB(I*2)
                call IPPARI (qty, -1, 1, Iqty, nval, Ierr, LOG)
                specy_group_qty = Iqty(1)
                IF (Ierr .NE. 0) THEN
                    WRITE(LOG,421)specy_formula(:specy_len),
     1                               str(Igroup:Islash-1)
  421               FORMAT(' formula "',A,'" group "',
     1                 A,'" quantity  ERROR !!!')
                    WRITE(*,422)specy_formula(:specy_len),
     1                               str(Igroup:Islash-1)
  422               FORMAT(' formula "',A,'" group "',
     1                 A,'" quantity  ERROR !!!')
                    GO TO 100
                ENDIF
                CALL CKCOMP (specy_group, group, group_total, Ifound)
```

```
C              write(*,*)specy_group,group_total
C              write(*,*)group(1),group(150),group(350)
               IF (Ifound.EQ. 0) THEN
                   ILEN = ILASCH(specy_group)
                   WRITE(*,430)specy_group(:ILEN), specy_name
  430              FORMAT(' Group "',A,'" not found for specy "',A,'"')
                   WRITE(LOG,440)specy_group(:ILEN), specy_name
  440              FORMAT(' Group "',A,'" not found for specy "',A,'"')
                   GO TO 100
               ENDIF
C              write(*,*)' specy_group = ',specy_group
C              write(*,*)'group_total = ',group_total
C              write(*,*)'Ifound = ',Ifound
               specy_Hf     = specy_Hf     + group_Hf(Ifound)
     1                                      * specy_group_qty
               specy_S      = specy_S      + group_S(Ifound)
     1                                      * specy_group_qty
               specy_Cp300  = specy_Cp300  + group_Cp300(Ifound)
     1                                      * specy_group_qty
               specy_Cp400  = specy_Cp400  + group_Cp400(Ifound)
     1                                      * specy_group_qty
               specy_Cp500  = specy_Cp500  + group_Cp500(Ifound)
     1                                      * specy_group_qty
               specy_Cp600  = specy_Cp600  + group_Cp600(Ifound)
     1                                      * specy_group_qty
               specy_Cp800  = specy_Cp800  + group_Cp800(Ifound)
     1                                      * specy_group_qty
               specy_Cp1000 = specy_Cp1000 + group_Cp1000(Ifound)
     1                                      * specy_group_qty
               specy_Cp1500 = specy_Cp1500 + group_Cp1500(Ifound)
     1                                      * specy_group_qty
               IF (group_Cp1500(Ifound).EQ.0.0) ErrCp1500 = .true.
C              write(lst,*)specy_group,' ',group_Cp1500(Ifound),
C    1                                        ' ',specy_group_qty
  500          CONTINUE
               IF (ErrCp1500) specy_Cp1500 = 0.0
               specy_S = specy_S - 1.98717 * ALOG(specy_symmetry*1.0)
C
               Irotor = INT(specy_rotor)
               write(LST,600)specy_name,specy_Hf,specy_S,specy_Cp300,
     1                       specy_Cp400,specy_Cp500,specy_Cp600,
     2                       specy_Cp800,specy_Cp1000,specy_Cp1500,
     3                       month,day,year,
     4                       specy_element(1),specy_element_qty(1),
     5                       specy_element(2),specy_element_qty(2),
     6                       specy_element(3),specy_element_qty(3),
     7                       specy_element(4),specy_element_qty(4),
     8                       Irotor
  600          format(' ',A9,F8.2,' ',F8.2,' ',7f8.2,1x,I2,'/',I2,'/',I2,
     1                 2x,'ThmCal',2x,4(A2,I3,1x),'G',I2)
           ENDIF
C
           go to 100
 5000      write(*,*)'...End of calculations...'
c           WRITE(LST,5100)
c 5100    FORMAT('END')
           write(*,*)' '
           write(*,*)' List file created as : ',File_lst
           write(*,*)' '
           go to 7000
 6000      write(*,*)'...error in input file...',File_inp
 7000      continue
           close (log)
           close (lst)
           close (inp)
           end
```

```
C
C-----------------------------------------------------------------------C
C
      SUBROUTINE therm_init(init_err)
      implicit none
      include 'thm_par.fi'
      include 'thm_cfg.fi'
C     local declaration
      character line*100, sub(20)*100, key(4)*5, UPCASE*4, KEY_WD*4,
     1          group_n*20
      integer cfg, ilen, nsub, nkey, nfile, nelem, nhelp, nkeys,
     1        Ifile(5), Ielem(5), Ihelp(5), Ikeys(5), nval, Igroup(5),
     2        Istart, Istr, Ierr, group_idx, IPPLEN, I, J, ngroup,
     3        Ifirch, Icoma, Ierr2, Ierr3
      real*8 Igp(9)
      logical init_err, BD
      data cfg/35/, key/'FILE','ELEM','HELP','KEYS'/
C
      OPEN(cfg,FILE='therm.cfg',STATUS='OLD',ERR=30)
      GO TO 50
 30   CONTINUE
      WRITE(*,40)
 40   FORMAT(' Configuration file THERM.CFG is not in the working direct
     +ory !!!')
      init_err = .true.
      return
 50   CONTINUE
      init_err = .false.
  100 CONTINUE
      LINE = ' '
      READ (CFG,'(A)',END=5000) LINE
  105 CONTINUE
      ILEN = IPPLEN(LINE)
      IF (ILEN .EQ. 0) GO TO 100
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      KEY_WD = SUB(1)(2:5)
      CALL CKCOMP ( UPCASE(KEY_WD, 4) , KEY, 4, NKEY)
C      write(*,*)' key = ',key_wd,' nkey = ',nkey
C      IF (NKEY .GT. 0) ITASK = 0
C
      IF (NKEY.EQ.1) THEN
          call IPPARI (SUB(2), -1, 1, Ifile, nval, Ierr, LOG)
C          write(*,*)'..group files = ',Ifile(1), ' Ierr = ',Ierr
          IF (Ierr.NE. 0) GO TO 6000
          nfile = Ifile(1)
          group_total = 0
          do 300 I = 1, nfile
              READ (CFG,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              IF (ILEN .EQ. 0) GO TO 100
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              BD = .false.
C     ADDED FOR RADICAL CORRECTION
              IF (UPCASE(SUB(1),2).EQ.'BD') BD=.TRUE.
              OPEN(100+I,FILE=sub(1),STATUS='OLD',ERR=130)
C              READ (100+I,'(A)',END=5000) LINE
              Ierr2 = 0
              Ierr3 = 0
  110         continue
              READ (100+I,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              IF (ILEN .EQ. 0) GO TO 6000
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              Istr = IPPLEN(sub(1))
              Istart = IFIRCH(sub(1))
              Icoma = INDEX(sub(1),',')
```

```
                IF (Icoma.EQ.0) THEN
                    group_n = line(Istart:Istr)
                ELSE
                    group_n = line(Istart:Icoma-1)
                ENDIF
                call IPPARI (group_n, -1, 1, Igroup, nval, Ierr, LOG)
                IF (Ierr.NE.0.AND.IERR2.NE.0.AND.IERR3.NE.0) THEN
                    GO TO 6000
                ELSEIF (Ierr.NE.0.AND.IERR2.NE.0) THEN
                    Ierr3 = 1
                    GO TO 110
                ELSEIF (Ierr.NE.0) THEN
                    Ierr2 = 1
                    GO TO 110
                ENDIF
                ngroup = Igroup(1)
                group_idx = group_total + 1
C               write(*,*)' group(',group_total,') = ',group(group_total)
                group_total = ngroup + group_total
C               write(*,*)'----group_total = ',group_total
                DO 120 J = 1, ngroup
                    READ (100+I,'(A)',END=6000) LINE
                    ILEN = IPPLEN(LINE)
                    IF (ILEN .EQ. 0) GO TO 6000
                    CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                    Istr = IPPLEN(sub(1))
                    Istart = IFIRCH(sub(1))
                    Icoma = INDEX(sub(1),',')
C                    write(*,*)'group_idx=',group_idx,' Istart=',Istart
C     1                          ,' Icoma=',Icoma,' J=',J
                    IF (Icoma.EQ.0) THEN
                        group(group_idx) = line(Istart:Istr)
                    ELSE
                        group(group_idx) = line(Istart:Icoma-1)
                    ENDIF
c
                    call IPPARR (line(Istr+2:79), -1, 11, Igp, nval,
     1                                Ierr, LOG)
C     CORRECT THE RADICAL OF LOSING ONE H ATOM FOR BD GROUPS
                    IF (BD) THEN
                        group_Hf(group_idx) = Igp(1)-52.1
                    ELSE
                        group_Hf(group_idx) = Igp(1)
                    ENDIF
                    group_S(group_idx) = Igp(2)
                    group_Cp300(group_idx) = Igp(3)
                    group_Cp400(group_idx) = Igp(4)
                    group_Cp500(group_idx) = Igp(5)
                    group_Cp600(group_idx) = Igp(6)
                    group_Cp800(group_idx) = Igp(7)
                    group_Cp1000(group_idx) = Igp(8)
                    IF (nval.NE. 8) group_Cp1500(group_idx) = Igp(9)
C                    write(lst,*)group(group_idx),nval
                    group_idx = group_idx + 1
  120           continue
                GO TO 200
  130           CONTINUE
                WRITE(*,140)sub(1)
  140           FORMAT(' Can't find group file : ',A20)
                init_err = .true.
                return
  200           CLOSE(100+I)
  300       continue
            go to 100
        ELSEIF (NKEY.EQ.2) THEN
            call IPPARI (SUB(2), -1, 1, Ielem, nval, Ierr, LOG)
```

```
          IF (Ierr.NE. 0) GO TO 6000
          element_qty = Ielem(1)
c          write(*,*)' elem_qty = ',element_qty
c          group_total = 0
          do 500 I = 1, element_qty
  400         continue
              READ (CFG,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              IF (ILEN .EQ. 0) GO TO 100
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              ILEN = IPPLEN(SUB(1))
              element(I) = UPCASE(SUB(1),ILEN)
c              write(*,*)' I = ',I,' element = ',element(I)
  500     continue
          go to 100
      ELSEIF (NKEY.EQ.3) THEN
  600     continue
          go to 5000
      ELSEIF (NKEY.EQ.4) THEN
  700     continue
          go to 5000
      ELSE
          WRITE(*,*)'Errors in configuration file THERM.CFG !!!'
          init_err = .true.
      ENDIF
 5000 continue
      go to 7000
 6000 WRITE(*,*)'Errors in configuration file THERM.CFG !!!'
      init_err = .true.
 7000 continue
      close (cfg)
      return
      end

C
C-----------------------------------------------------------------------C
      SUBROUTINE find_elem(specy_formula,specy_len,specy_element,
     1                     specy_element_qty,element,element_qty,
     2                     elem_err,LOG)
      IMPLICIT NONE
C
      CHARACTER*(*) specy_formula, specy_element(*), element(*)
C
      CHARACTER numeric(10), char(26), elem_ch1, elem_ch2,qty_ch*3,
     1          paren(2), UPCASE*4
C
      INTEGER  specy_element_qty(*), LOG, I, J, K, Ifirch, Islash,
     1         Ilen, NKEY, specy_len, Index2, sym(40),
     2         qty(2), Ierr, element_qty, IPPLEN, specy_n(20),
     3         specy_n_temp(20), nval, M
C
      LOGICAL elem_err, paren_left, paren_right
C
      DATA    paren/'(',')'/,
     4         sym/40*0/
C
      elem_err = .false.
      Ilen = IPPLEN(specy_formula)
      IF (Ilen.EQ.0) THEN
          elem_err = .true.
          write(LOG,*)' ...ERROR...  specy formula did not specify.'
          return
      ENDIF
C
C
      DO 30 I = 1, 20
```

```
          specy_n(I) = 0
          specy_n_temp(I) = 0
   30 CONTINUE
      DO 100 I = 1, specy_len
          K = ICHAR(specy_formula(I:I))
          paren_left = .false.
          paren_right = .false.
          IF (K.GE.48.AND. K.LE.57) THEN
              sym(I) = 2
          ELSEIF (K.GE.65.AND. K.LE.90) THEN
              sym(I) = 1
          ELSEIF (K.GE.97.AND. K.LE.122) THEN
              sym(I) = 1
              specy_formula(I:I) = UPCASE(specy_formula(I:I),1)
          ELSEIF (K.EQ.40.) THEN
              sym(I) = 3
          ELSEIF (K.EQ.41) THEN
              sym(I) = 4
          ELSE
              elem_err = .true.
              write(LOG,50)specy_formula(:Ilen),specy_formula(I:I)
   50         format(1x,'  specy formula "',A,'" contains unknown',
     1                    ' character "',A,'"')
              return
          ENDIF
  100 CONTINUE
      Index2 = 1
c      write(*,*)' sym = ',sym
  200 CONTINUE
c      write(*,*)' Index2 = ',Index2,' sym = ',sym(Index2),' Ilen = ',
c     1           Ilen,' ',specy_len,' ',specy_formula
c      write(*,*)sym(I)
      IF (sym(Index2).EQ.1) THEN
          IF (paren_left) THEN
              CALL sub_elem(Index2,specy_formula,specy_n_temp,sym,
     1                      specy_len,element,element_qty,elem_err,LOG)
              GO TO 300
          ELSE
              CALL sub_elem(Index2,specy_formula,specy_n,sym,
     1                      specy_len,element,element_qty,elem_err,LOG)
              GO TO 300
          ENDIF
      ELSEIF (sym(Index2).EQ.3) THEN
          paren_left = .true.
          Index2 = Index2 + 1
          GO TO 300
      ELSEIF (sym(Index2).EQ.4) THEN
          IF (paren_left) THEN
              Index2 = Index2 + 1
              qty(1) = 1
              IF (sym(Index2).EQ.2) THEN
                  IF (sym(Index2+1).EQ.2) THEN
                      qty_ch = '  '
                      qty_ch = specy_formula(Index2:Index2+1)
                      CALL IPPARI(qty_ch, -1, 1, qty, nval, Ierr, LOG)
                      Index2 = Index2 + 2
                  ELSE
                      qty_ch = '  '
                      qty_ch = specy_formula(Index2:Index2)
                      CALL IPPARI(qty_ch, -1, 1, qty, nval, Ierr, LOG)
                      Index2 = Index2 + 1
                  ENDIF
              ENDIF
              DO 220 J = 1, element_qty
                  specy_n(J) = specy_n(J) + qty(1)*specy_n_temp(J)
  220         CONTINUE
```

```
          ELSE
              elem_err = .true.
              write(LOG,240)specy_formula(:Ilen)
  240         format(1x,'  specy formula "',A,'" ... error ...')
              return
          ENDIF
      ELSE
          elem_err = .true.
          write(LOG,260)specy_formula(:Ilen)
  260     format(1x,'  specy formula "',A,'" ... error ...')
          return
      ENDIF
  300 CONTINUE
c      IF (Index2.GT.Ilen) return
      IF (Index2.LE.Ilen) GO TO 200
      K = 0
      DO 400 J = 1, element_qty
          IF (specy_n(J).GT.0) K = K + 1
  400 CONTINUE
      IF (K.GT.4) THEN
          elem_err = .true.
          write(LOG,500)specy_formula(:Ilen)
  500     format(1x,'  specy formula "',A,'" more than 4 elements')
      ENDIF
      IF (specy_n(1).NE.0) THEN
          M = 1
          DO 600 J = 1, element_qty
              IF (specy_n(J).NE.0) THEN
                  specy_element(M) = element(J)
                  specy_element_qty(M) = specy_n(J)
                  M = M + 1
              ENDIF
  600     CONTINUE
      ELSEIF (specy_n(2).NE.0.AND.K.LT.3) THEN
          M = 3
          specy_element(1) = element(1)
          specy_element_qty(1) = 0
          specy_element(2) = element(2)
          specy_element_qty(2) = 0
          DO 700 J = 3, element_qty
              IF (specy_n(J).NE.0) THEN
                  specy_element(M) = element(J)
                  specy_element_qty(M) = specy_n(J)
                  M = M + 1
              ENDIF
  700     CONTINUE
      ELSEIF (specy_n(2).NE.0.AND.K.LT.4) THEN
          M = 2
          specy_element(1) = element(2)
          specy_element_qty(1) = 0
          DO 800 J = 2, element_qty
              IF (specy_n(J).NE.0) THEN
                  specy_element(M) = element(J)
                  specy_element_qty(M) = specy_n(J)
                  M = M + 1
              ENDIF
  800     CONTINUE
      ELSE
          M = 1
          DO 900 J = 1, element_qty
              IF (specy_n(J).NE.0) THEN
                  specy_element(M) = element(J)
                  specy_element_qty(M) = specy_n(J)
                  M = M + 1
              ENDIF
  900     CONTINUE
```

```
      ENDIF
C
      return
      end
C
C-----------------------------------------------------------------------C
      SUBROUTINE sub_elem(Index2,specy_formula,specy_n,sym,
     1                    specy_len,element,element_qty,elem_err,LOG)
      IMPLICIT NONE
C
      CHARACTER*(*) specy_formula, element(*)
C
      CHARACTER elem_ch, elem_ch2*2, qty_ch*3
C
      INTEGER  LOG, I, J, K, Ifirch, Islash,
     1         Ilen, NKEY, specy_len, Index2, sym(40),
     2         qty(2), Ierr, M, element_qty, specy_n(*), elem_found,
     3         nval
C
      LOGICAL elem_err, paren_left, paren_right
C
      elem_found = 0
      IF (sym(Index2).NE.1) THEN
          elem_err = .true.
          write(LOG,100)specy_formula(:Ilen)
  100     format(1x,'  specy formula "',A,'" ... error ...')
          return
      ENDIF
      elem_ch  = specy_formula(Index2:Index2)
      CALL CKCOMP ( elem_ch, element, element_qty, NKEY)
c      write(*,*)elem_ch,element_qty,NKEY
c      do 150 i = 1, element_qty
c      write(*,*)element(i)
c  150 continue
      IF (NKEY.EQ.0) THEN
          elem_err = .true.
          write(LOG,200)specy_formula(:Ilen)
  200     format(1x,'  specy formula "',A,'" ... error ...')
          return
      ENDIF
      elem_found = NKEY
      qty(1) = 1
      IF (Index2+1.GT.specy_len) THEN
          specy_n(elem_found) = specy_n(elem_found) + qty(1)
          Index2 = Index2 + 1
          return
      ENDIF
      IF (sym(Index2+1).EQ.3.OR.sym(Index2+1).EQ.4) THEN
          specy_n(elem_found) = specy_n(elem_found) + 1
          Index2 = Index2 + 1
          return
      ELSEIF (sym(Index2+1).EQ.2) THEN
          M = 1
  300     CONTINUE
          IF ((Index2+M+1).LE.specy_len.AND.sym(Index2+M+1).EQ.2) THEN
              M = M + 1
              GO TO 300
          ENDIF
          qty_ch = ' '
          qty_ch = specy_formula(Index2+1:Index2+M)
          CALL IPPARI(qty_ch, -1, 1, qty, nval, Ierr, LOG)
          specy_n(elem_found) = specy_n(elem_found) + qty(1)
          Index2 = Index2 + M + 1
c          write(*,*)' Index2 = ',Index2,' qty = ',qty(1)
          return
      ELSEIF (sym(Index2+1).EQ.1) THEN
```

```
            elem_ch2  = specy_formula(Index2:Index2+1)
            CALL CKCOMP (elem_ch2, element, element_qty, NKEY)
            IF (NKEY.EQ.0) THEN
                specy_n(elem_found) = specy_n(elem_found) + 1
                Index2 = Index2 + 1
                return
            ELSE
                elem_found = NKEY
                Index2 = Index2 + 2
            ENDIF
            M = 0
   400      CONTINUE
            IF ((Index2+M+1).LE.specy_len.AND.sym(Index2+M).EQ.2) THEN
                M = M + 1
                GO TO 400
            ENDIF
            IF (M.GT.0) THEN
                qty_ch = '  '
                qty_ch = specy_formula(Index2:Index2+M-1)
                CALL IPPARI(qty_ch, -1, 1, qty, nval, Ierr, LOG)
                specy_n(elem_found) = specy_n(elem_found) + qty(1)
                Index2 = Index2 + M
                return
            ELSE
                specy_n(elem_found) = specy_n(elem_found) + 1
                return
            ENDIF
        ELSE
            specy_n(elem_found) = specy_n(elem_found) + 1
            Index2 = Index2 + 1
        ENDIF
        return
        end
C
C-----------------------------------------------------------------------C
        SUBROUTINE CKISUB (LINE, SUB, NSUB)
C
C       Generates an array of CHAR*(*) substrings from a CHAR*(*) string,
C       using blanks or tabs as delimiters
C
C       Input:  LINE  - a CHAR*(*) line
C       Output: SUB   - a CHAR*(*) array of substrings
C               NSUB  - number of substrings found
C       A '!' will comment out a line, or remainder of the line.
C                                       F. Rupley, Div. 8245, 5/15/86
C-----------------------------------------------------------------------C
C*****precision > double
        IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C       IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
        CHARACTER*(*) SUB(*), LINE
        NSUB = 0
C
        DO 5 N = 1, LEN(LINE)
          IF (ICHAR(LINE(N:N)) .EQ. 9) LINE(N:N) = ' '
     5  CONTINUE
C
        IF (IPPLEN(LINE) .LE. 0) RETURN
C
        ILEN = ILASCH(LINE)
C
        NSTART = IFIRCH(LINE)
    10  CONTINUE
```

```
      ISTART = NSTART
      NSUB = NSUB + 1
      SUB(NSUB) = ' '
C
      DO 100 I = ISTART, ILEN
         ILAST = INDEX(LINE(ISTART:),' ') - 1
         IF (ILAST .GT. 0) THEN
            ILAST = ISTART + ILAST - 1
         ELSE
            ILAST = ILEN
         ENDIF
         SUB(NSUB) = LINE(ISTART:ILAST)
         IF (ILAST .EQ. ILEN) RETURN
C
         NSTART = ILAST + IFIRCH(LINE(ILAST+1:))
C
C        Does SUB have any slashes?
C
         I1 = INDEX(SUB(NSUB),'~')
         IF (I1 .LE. 0) THEN
            IF (LINE(NSTART:NSTART) .NE. '~') GO TO 10
            NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
            IND = INDEX(SUB(NSUB),' ')
            SUB(NSUB)(IND:) = LINE(NSTART:NEND)
            IF (NEND .EQ. ILEN) RETURN
            NSTART = NEND + IFIRCH(LINE(NEND+1:))
            GO TO 10
         ENDIF
C
C        Does SUB have 2 slashes?
C
         I2 = INDEX(SUB(NSUB)(I1+1:),'~')
         IF (I2 .GT. 0) GO TO 10
C
         NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
         IND = INDEX(SUB(NSUB),' ') + 1
         SUB(NSUB)(IND:) = LINE(NSTART:NEND)
         IF (NEND .EQ. ILEN) RETURN
         NSTART = NEND + IFIRCH(LINE(NEND+1:))
         GO TO 10
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPNPAR (LINE, NPAR, IPAR, ISTART)
C
C     Returns CHAR*(*) IPAR substring of CHAR*(*) string LINE which
C     contains NPAR real parameters
C
C     Input:     LINE - a CHAR*(*) line
C                NPAR - number of parameters expected
C     Output:    IPAR - the substring of parameters only
C                ISTART - the starting location of IPAR substring
C     A '!' will comment out a line, or remainder of the line.
C                                    F. Rupley, Div. 8245, 5/14/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) LINE,IPAR
C
C----------Find Comment String (! signifies comment)
```

```
C
      ILEN = IPPLEN(LINE)
      ISTART = 0
      N = 0
      IF (ILEN.GT.0) THEN
         DO 40 I = ILEN, 1, -1
            ISTART = I
            IPAR = ' '
            IPAR = LINE(ISTART:ILEN)
            IF (LINE(I:I).NE.' ') THEN
               IF (I .EQ. 1) RETURN
               IF (LINE(I-1:I-1) .EQ. ' ') THEN
                  N = N + 1
                  IF (N .EQ. NPAR) RETURN
               ENDIF
            ENDIF
   40    CONTINUE
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPPARI(STRING, ICARD, NEXPEC, IVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARI
C   REFER TO  IPGETI
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851725   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses integer variables from a character variable.  Called
C            by IPGETI, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARI may be used for parsing an input record that contains integer
C  values, but was read into a character variable instead of directly
C  into integer variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARI to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of IVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1,   2,,40000   , ,60'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set IVAL(1) = 1
C     (2) set IVAL(2) = 2
C     (3) leave IVAL(3) unchanged
C     (4) set IVAL(4) = 40000
C     (5) leave IVAL(5) unchanged
C     (6) set IVAL(6) = 60
C
C   IPPARI will print diagnostics on the default output device, if
C   desired.
C
C   IPPARI is part of IOPAK, and is written in ANSI FORTRAN 77
```

```
C
C   Examples:
C
C      Assume IVAL = (0, 0, 0) and NEXPEC = 3 on entry:
C
C   input string           IVAL on exit             IERR    NFOUND
C   ------------           ---------------------    ----    ------
C   '  2 ,   3 45 '        (2, 3, 45)                 0       3
C   '2.15,,3'              (2, 0, 3)                  1       0
C   '3X, 25, 2'            (0, 0, 0)                  1       0
C   '10000'                (10000, 0, 0)              2       1
C
C      Assume IVAL = (0, 0, 0, 0) and NEXPEC = -4 on entry:
C
C   input string           IVAL on exit             IERR    NFOUND
C   ------------           ---------------------    ----    ------
C   '1, 2'                 (1, 2)                     0       2
C   ',,37  400'            (0, 0, 37, 400)            0       4
C   ' 1,,-3,,5'            (1, 0, -3, 0)              3       4
C
C   arguments: (I=input,O=output)
C   ----------------------------
C   STRING (I) - the character string to be parsed.
C
C   ICARD  (I) - data statement number, and error processing flag
C          < 0 : no error messages printed
C          = 0 : print error messages, but not ICARD
C          > 0 : print error messages, and ICARD
C
C   NEXPEC (I) - number of real variables expected to be input.  If
C          < 0, the number is unknown, and any number of values
C          between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C   PROMPT (I) - prompting string, character type.  A question
C          mark will be added to form the prompt at the screen.
C
C   IVAL (I,O) - the integer value or values to be modified.  On entry,
C        the values are printed as defaults.  The formal parameter
C        corresponding to IVAL must be dimensioned at least NEXPEC
C        in the calling program if NEXPEC > 1.
C
C   NFOUND (O) - the number of real values represented in STRING,
C          only in the case that there were as many or less than
C          NEXPEC.
C
C   IERR (O) - error flag:
C        = 0 if no errors found
C        = 1 syntax errors or illegal values found
C        = 2 for too few values found (NFOUND < NEXPEC)
C        = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C    REFERENCES  (NONE)
C    ROUTINES CALLED  IFIRCH,ILASCH
C    END PROLOGUE  IPPARI
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION IVAL(*)
```

```
      CHARACTER *8 FMT(14)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARI
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set false when a space follows
C--- an integer value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR .OR. NC .EQ. IE) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into integer array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      IES = NC - 1
      NCH = IES - IBS + 1
      DATA FMT/' (I1)', ' (I2)', ' (I3)', ' (I4)', ' (I5)',
     1   ' (I6)', ' (I7)', ' (I8)', ' (I9)', '(I10)',
     2   '(I11)', '(I12)', '(I13)', '(I14)'/
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
```

```
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) IVAL(NFOUND)
      GO TO 450
400   CONTINUE
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0)RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1    WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      SUBROUTINE IPPARR(STRING, ICARD, NEXPEC, RVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARR
C   REFER TO  IPGETR
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851625   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses real variables from a character variable.  Called
C            by IPGETR, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARR may be used for parsing an input record that contains real
C  values, but was read into a character variable instead of directly
C  into real variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARR to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of RVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1.,   2,,4.e-5   , ,6.e-6'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set RVAL(1) = 1.0
C     (2) set RVAL(2) = 2.0
C     (3) leave RVAL(3) unchanged
C     (4) set RVAL(4) = 4.0E-05
C     (5) leave RVAL(5) unchanged
C     (6) set RVAL(6) = 6.0E-06
C
```

```
C   IPPARR will print diagnostics on the default output device, if
C   desired.
C
C   IPPARR is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume RVAL = (0., 0., 0.) and NEXPEC = 3 on entry:
C
C   input string           RVAL on exit             IERR    NFOUND
C   ------------           ---------------------    ----    ------
C  '  2.34e-3,  3 45.1'    (2.34E-03, 3.0, 45.1)      0       3
C  '2,,3.-5'              (2.0, 0.0, 3.0E-05)        0       3
C  ',1.4,0.028E4'         (0.0, 1.4, 280.0)          0       3
C  '1.0, 2.a4, 3.0'       (1.0, 0.0, 0.0)            1       1
C  '1.0'                  (1.0, 0.0, 0.0)            2       1
C
C      Assume RVAL = (0.,0.,0.,0.) and NEXPEC = -4 on entry:
C
C   input string           RVAL on exit             IERR    NFOUND
C   ------------           ---------------------    ----    ------
C  '1.,2.'                (1.0, 2.0)                 0       2
C  ',,3  4.0'             (0.0, 0.0, 3.0, 4.0)       0       4
C  '1,,3,,5.0'            (0.0, 0.0, 3.0, 0.0)       3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  RVAL (I,O) - the real value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to RVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARR
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
```

```
C*****END precision > single
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION RVAL(*)
      CHARACTER *8 FMT(22)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARR
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set negative when a space follows
C--- a real value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into real array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      DATA FMT/     ' (E1.0)', ' (E2.0)', ' (E3.0)', ' (E4.0)',
     1    ' (E5.0)', ' (E6.0)', ' (E7.0)', ' (E8.0)', ' (E9.0)',
     2    '(E10.0)', '(E11.0)', '(E12.0)', '(E13.0)', '(E14.0)',
```

```
     3   '(E15.0)', '(E16.0)', '(E17.0)', '(E18.0)', '(E19.0)',
     4   '(E20.0)', '(E21.0)', '(E22.0)'/
      IES = NC - 1
      NCH = IES - IBS + 1
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) RVAL(NFOUND)
      GO TO 450
400   CONTINUE
      WRITE (LOUT, 555) STRING(IBS:IES)
  555 FORMAT (A)
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0) RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1   WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
```

```
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END
      FUNCTION ILASCH(STRING)
C   BEGIN PROLOGUE  ILASCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines last significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the last non-blank character in a string of
C  arbitrary length.  If no characters are found, ILASCH is set = 0.
C  When used with the companion routine IFIRCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C  Note that the FORTRAN intrinsic function LEN returns the length
C  of a character string as declared, rather than as filled.  The
C  declared length includes leading and trailing blanks, and thus is
C  not useful in generating 'significant' substrings.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) STRING
C
C***FIRST EXECUTABLE STATEMENT ILASCH
      NLOOP = LEN(STRING)
      IF (NLOOP.EQ.0) THEN
         ILASCH = 0
         RETURN
      ENDIF
C
      DO 100 I = NLOOP, 1, -1
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
120   CONTINUE
      ILASCH = I
      END
```

```
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKCOMP (IST, IRAY, II, I)
C
C  START PROLOGUE
C
C  SUBROUTINE CKCOMP (IST, IRAY, II, I)*
C     Returns the index of an element of a reference character
C     string array which corresponds to a character string;
C     leading and trailing blanks are ignored.
C
C
C  INPUT
C     IST   - A character string.
C                   Data type - CHARACTER*(*)
C     IRAY  - An array of character strings;
C             dimension IRAY(*) at least II
C                   Data type - CHARACTER*(*)
C     II    - The length of IRAY.
C                   Data type - integer scalar.
C
C  OUTPUT
C     I     - The first integer location in IRAY in which IST
C             corresponds to IRAY(I); if IST is not also an
C             entry in IRAY, I=0.
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) IST, IRAY(*)
C
      I = 0
      DO 10 N = II, 1, -1
         IS1 = IFIRCH(IST)
         IS2 = ILASCH(IST)
         IR1 = IFIRCH(IRAY(N))
         IR2 = ILASCH(IRAY(N))
         IF ( IS2.GE.IS1 .AND. IS2.GT.0 .AND.
     1        IR2.GE.IR1 .AND. IR2.GT.0 .AND.
     2        IST(IS1:IS2).EQ.IRAY(N)(IR1:IR2) ) I=N
   10 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKUNIT (LINE, AUNITS, EUNITS, IUNITS)
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
      CHARACTER*(*) LINE, IUNITS, AUNITS, EUNITS
      CHARACTER*4 UPCASE
C
      AUNITS = ' '
      EUNITS = ' '
      IUNITS = ' '
      DO 85 N = 1, ILASCH(LINE)-3
```

```
            IND = ILASCH(IUNITS)
            IF (EUNITS .EQ. ' ') THEN
               IF (UPCASE(LINE(N:), 4)     .EQ. 'CAL/') THEN
                  EUNITS = 'CAL/'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'E units cal/mole'
                  ELSE
                     IUNITS(IND:) = ', E units cal/mole'
                  ENDIF
               ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KCAL') THEN
                  EUNITS = 'KCAL'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'E units Kcal/mole'
                  ELSE
                     IUNITS(IND:) = ', E units Kcal/mole'
                  ENDIF
               ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'JOUL') THEN
                  EUNITS = 'JOUL'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'E units Joules/mole'
                  ELSE
                     IUNITS(IND:) = ', E units Joules/mole'
                  ENDIF
               ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KJOU') THEN
                  EUNITS = 'KJOU'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'E units Kjoule/mole'
                  ELSE
                     IUNITS(IND:) = ', E units Kjoule/mole'
                  ENDIF
               ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KELV') THEN
                  EUNITS = 'KELV'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'E units Kelvins'
                  ELSE
                     IUNITS(IND:) = ', E units Kelvins'
                  ENDIF
               ENDIF
            ENDIF
            IF (AUNITS .EQ. ' ') THEN
               IF (UPCASE(LINE(N:), 4) .EQ. 'MOLE') THEN
                  IF (UPCASE(LINE(N+4:), 1) .EQ. 'S') THEN
                     AUNITS = 'MOLE'
                     IF (IUNITS .EQ. ' ') THEN
                        IUNITS = 'A units mole-cm-sec-K'
                     ELSE
                        IUNITS(IND:) = ', A units mole-cm-sec-K'
                     ENDIF
                  ELSEIF (UPCASE(LINE(N+4:), 4) .EQ. 'CULE') THEN
                     AUNITS = 'MOLC'
                     IF (IUNITS .EQ. ' ') THEN
                        IUNITS = 'A units molecules'
                     ELSE
                        IUNITS(IND:) = ', A units molecules'
                     ENDIF
                  ENDIF
               ENDIF
            ENDIF
      85 CONTINUE
C
         IF (AUNITS .EQ. ' ') THEN
            AUNITS = 'MOLE'
            IND = ILASCH(IUNITS) + 1
            IF (IND .GT. 1) THEN
               IUNITS(IND:) = ', A units mole-cm-sec-K'
            ELSE
```

```
            IUNITS(IND:) = ' A units mole-cm-sec-K'
         ENDIF
      ENDIF
C
      IF (EUNITS .EQ. ' ') THEN
         EUNITS = 'CAL/'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', E units cal/mole'
         ELSE
            IUNITS(IND:) = ' E units cal/mole'
         ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      INTEGER FUNCTION IPPLEN (LINE)
C
C  BEGIN PROLOGUE
C
C  FUNCTION IPPLEN (LINE)
C     Returns the effective length of a character string, i.e.,
C     the index of the last character before an exclamation mark (!)
C     indicating a comment.
C
C  INPUT
C     LINE  - A character string.
C
C  OUTPUT
C     IPPLEN - The effective length of the character string.
C
C  END PROLOGUE
C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER LINE*(*)
C
      IN = IFIRCH(LINE)
      IF (IN.EQ.0 .OR. LINE(IN:IN).EQ.'!') THEN
         IPPLEN = 0
      ELSE
         IN = INDEX(LINE,'!')
         IF (IN .EQ. 0) THEN
            IPPLEN = ILASCH(LINE)
         ELSE
            IPPLEN = ILASCH(LINE(:IN-1))
         ENDIF
      ENDIF
      RETURN
      END
C
      CHARACTER*(*) FUNCTION UPCASE(ISTR, ILEN)
      CHARACTER ISTR*(*), LCASE(26)*1, UCASE(26)*1
      DATA LCASE /'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1            'n','o','p','q','r','s','t','u','v','w','x','y','z'/,
     2     UCASE /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     3            'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
C
```

```
      UPCASE = ' '
      UPCASE = ISTR(:ILEN)
      JJ = MIN (LEN(UPCASE), LEN(ISTR), ILEN)
      DO 10 J = 1, JJ
         DO 10 N = 1,26
            IF (ISTR(J:J) .EQ. LCASE(N)) UPCASE(J:J) = UCASE(N)
   10 CONTINUE
      RETURN
      END



      SUBROUTINE THERM_IO(ERROR)
C
      IMPLICIT NONE
      INCLUDE 'THM_PAR.FI'
C
      CHARACTER FILE_NAME*20
      INTEGER IDOT, ISTART, IFIRCH
      LOGICAL ERROR
C
      ERROR = .FALSE.
    5 WRITE(*,10)
   10 FORMAT(' Please enter the input file name : ')
      READ(*,'(A20)')FILE_INP
      OPEN(INP,FILE=FILE_INP,STATUS='OLD',ERR=30)
      GO TO 50
   30 CONTINUE
      WRITE(*,40)FILE_INP
   40 FORMAT(' Can not find file : ',A20)
      GO TO 5
   50 CONTINUE
      ISTART = IFIRCH(FILE_INP)
      IDOT = INDEX(FILE_INP,'.')
      FILE_NAME = FILE_INP(ISTART:IDOT-1)
C      FILE_LOG = FILE_NAME + '.' + 'LOG'
      FILE_LOG = 'THERMCAL.LOG'
C      FILE_LOG(IDOT:IDOT) = '.'
C      FILE_LOG(IDOT+1:IDOT+3) = 'LOG'
C      FILE_LST = FILE_NAME + '.' + 'LST'
      FILE_LST = FILE_NAME
      FILE_LST(IDOT:IDOT) = '.'
      FILE_LST(IDOT+1:IDOT+3) = 'LST'
C      FILE_DAT = FILE_NAME + '.' + 'DAT'
C
      OPEN(LOG,FILE=FILE_LOG,STATUS='UNKNOWN',ERR=230)
      CLOSE(LOG,STATUS='DELETE')
      OPEN(LOG,FILE=FILE_LOG,STATUS='NEW',ERR=230)
      GO TO 250
  230 CONTINUE
      WRITE(*,240)FILE_LOG
  240 FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 350
  250 CONTINUE
C
      OPEN(LST,FILE=FILE_LST,STATUS='UNKNOWN',ERR=330)
      CLOSE(LST,STATUS='DELETE')
      OPEN(LST,FILE=FILE_LST,STATUS='NEW',ERR=330)
      GO TO 350
  330 CONTINUE
      WRITE(*,340)FILE_LST
  340 FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 350
```

```
 350  CONTINUE
C
      RETURN
      END
```

D.2 THERMSRT

```
C
      PROGRAM THERMCVT
C
C-----------------------------------------------------------------------C
c***************changed W.Ing 06-15-94, 08-01-95**********************
C
      IMPLICIT NONE
C
      include 'thm_par2.fi'
C      include 'thm_cfg.fi'
C
      CHARACTER specy_name*70, specy_formula*40,
     3          specy_group(Max_Specy_Group)*14,KEY2(3)*4,
     4          specy_element(Max_Element)*4, str_int*8,
     5          specy_comment*70, specy_phase, line*100, KEY(5)*5,
     6          SUB(20)*100, UPCASE*5, str*100, qty*100, NAME*200
C
      INTEGER  specy_group_qty(Max_Specy_Group), specy_symmetry,
     1         specy_element_qty(Max_Element), Ngroup,ILASCH,
     2         ILEN, IPPLEN, NSUB, NKEY, Isymmetry(5), NVAL, Ierr,
     3         I, Istr, Igroup(5), Ifirch, Islash, Iqty(5), Ifound,J,
     4         Index, Itotal, Iname, NKEY2, Nloop
C
      REAL*8 specy_Hf, specy_S, specy_Cp300, specy_Cp400, specy_Cp500,
     1       specy_Cp600, specy_Cp800, specy_Cp1000, specy_Cp1500,
     2       specy_rotor, rotor(5)
C
      LOGICAL therm, nasa, err, io_err
C
      DATA KEY/'UNITS','RADIC','XXX','END','!'/
      DATA KEY2/'SYMM','NROT','ENDS'/
C
C
C     Call Initialization subroutines, 02/95 ING
C
C      CALL TITLE
      CALL THERM_IO(io_err)
      if (io_err) GO TO 6000
C      CALL THERM_INIT(init_err)
C      if (init_err) GO TO 6000
C
      write(lst,*)'THERM BATCH INPUT FILE'
C      write(lst,50)
C   50 format(' SPECIES       Hf       S     Cp 300    400     500     60
C     10     800    1000     1500     DATE          ELEMENTS')
C
  100 CONTINUE
      specy_rotor = 0.0
      specy_symmetry = 1
      LINE = ' '
      READ (INP,'(A)',END=5000) LINE
  105 CONTINUE
      ILEN = IPPLEN(LINE)
      IF (ILEN .EQ. 0) GO TO 100
      READ (INP,'(A)',END=5000) LINE
      READ (INP,'(A)',END=5000) LINE
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      specy_name = SUB(1)
      specy_formula = SUB(2)
```

```
C
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C
C        IS THERE A KEYWORD?
C
      CALL CKCOMP ( UPCASE(SUB(1), 5) , KEY, 2, NKEY)
C      WRITE(*,*)UPCASE(SUB(1),3),KEY(1),NKEY
C      IF (NKEY .GT. 0) ITASK = 0
C
C      WRITE(*,*)UPCASE(SUB(1), 5)
C      WRITE(*,*)NKEY
      IF (NKEY.EQ.2) THEN
          READ (INP,'(A)',END=5000) LINE
          READ (INP,'(A)',END=5000) LINE
          READ (INP,'(A)',END=5000) LINE
      ELSEIF (NKEY.EQ.1) THEN
C     DO NOTHING FOR MOLECULES
      ELSE
C     ERROR FOR NEITHER MOLECULES NOR RADICALS
          GO TO 6000
      ENDIF
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      call IPPARI (SUB(2), -1, 1, Igroup, nval, Ierr, LOG)
      Ngroup = Igroup(1)
      IF (Ierr.NE.0 .OR. Ngroup.EQ.0) THEN
  160     CONTINUE
          WRITE(*,*)' specy "',specy_name,'" group number error !!!'
          READ (INP,'(A)',END=5000) LINE
          ILEN = IPPLEN(LINE)
          CALL CKISUB (LINE(:ILEN), SUB, NSUB)
          CALL CKCOMP ( UPCASE(SUB(1), 4) , KEY2, 3, NKEY2)
          IF (NKEY2.NE.3) GO TO 160
          GO TO 100
      ENDIF
C GO TO 6000
      READ (INP,'(A)',END=5000) LINE
      IF (Ngroup.LE.5) THEN
          DO 200 I = 1, Ngroup
              READ (INP,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              specy_group(I) = SUB(3)
              call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
              specy_group_qty(I) = Iqty(1)
  200     CONTINUE
      ELSE
          Nloop = INT(Ngroup/2.0)
C          write(*,*)'Ngroup = ',Ngroup,' Nloop = ',Nloop
          IF(MOD(Ngroup,2).EQ.0) THEN
              DO 210 I = 1, Nloop
                  READ (INP,'(A)',END=5000) LINE
                  ILEN = IPPLEN(LINE)
                  CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                  specy_group(I) = SUB(3)
                  call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(I) = Iqty(1)
                  specy_group(Nloop+I) = SUB(9)
                  call IPPARI (SUB(11), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(Nloop+I) = Iqty(1)
C         write(*,*)I,specy_group(I),specy_group_qty(I)
C         write(*,*)Nloop+I,specy_group(Nloop+I),specy_group_qty(Nloop+I)
  210         CONTINUE
```

```
            ELSE
                DO 220 I = 1, Nloop
                    READ (INP,'(A)',END=5000) LINE
                    ILEN = IPPLEN(LINE)
                    CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                    specy_group(I) = SUB(3)
                    call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
                    specy_group_qty(I) = Iqty(1)
                    specy_group(Nloop+I+1) = SUB(9)
                    call IPPARI (SUB(11), -1, 1, Iqty, nval, Ierr, LOG)
                    specy_group_qty(Nloop+I+1) = Iqty(1)
  220           CONTINUE
                READ (INP,'(A)',END=5000) LINE
                ILEN = IPPLEN(LINE)
                CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                specy_group(Nloop+1) = SUB(3)
                call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
                specy_group_qty(Nloop+1) = Iqty(1)
            ENDIF
        ENDIF
C           WRITE(LST,*)LINE
C       READ (INP,'(A)',END=5000) LINE
C       READ (INP,'(A)',END=5000) LINE
C       READ (INP,'(A)',END=5000) LINE
  250 CONTINUE
        READ (INP,'(A)',END=5000) LINE
        ILEN = IPPLEN(LINE)
        CALL CKISUB (LINE(:ILEN), SUB, NSUB)
        CALL CKCOMP ( UPCASE(SUB(1), 4) , KEY2, 3, NKEY2)
C       WRITE(*,*)UPCASE(SUB(1), 4)
        IF (NKEY2.EQ.0) GO TO 250
        IF (NKEY2.EQ.3) GO TO 280
        IF (NKEY2.EQ.2) THEN
            call IPPARR (SUB(2), -1, 1, rotor, nval, Ierr, LOG)
            IF (Ierr.NE. 0) GO TO 6000
            specy_rotor = rotor(1)
C           READ (INP,'(A)',END=5000) LINE
C           ILEN = IPPLEN(LINE)
C           CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C           call IPPARI (SUB(2), -1, 1, Isymmetry, nval, Ierr, LOG)
C           IF (Ierr.NE. 0) GO TO 6000
C           specy_symmetry = Isymmetry(1)
        ELSE
            call IPPARI (SUB(2), -1, 1, Isymmetry, nval, Ierr, LOG)
C           write(*,*)line
C           write(*,*)sub(2)
            IF (Ierr.NE. 0) GO TO 6000
            specy_symmetry = Isymmetry(1)
        ENDIF
        GO TO 250
C
  280 CONTINUE
      write(LST,300)specy_name,specy_formula,
     2              specy_symmetry,specy_rotor
  300 format(A12,' ',A12,' ',I4,' ',F4.1)
C       write(LST,300)specy_name(:ILASCH(specy_name)),
C     1               specy_formula(:ILASCH(specy_name)),
C     2               specy_symmetry,specy_rotor
C  300 format(A,' ',A,' ',I3,' ',F4.1)
c       Index = 1
        Itotal = 0
        DO 350 J = 1, Ngroup
            Itotal = Itotal + 1
            NAME(Itotal:Itotal) = ' '
            Index = Itotal + 1
            Iname = IPPLEN(specy_group(J))
```

```
          Itotal = Itotal + Iname
          NAME(Index:Itotal) = specy_group(J)
          NAME(Itotal+1:Itotal+1) = ' '
          CALL CKI2CH (specy_group_qty(J), str_int, Iname, err)
          NAME(Itotal+2:Itotal+Iname+1) = str_int
          Itotal = Itotal + 1 + Iname
  350 CONTINUE
      ILEN = IPPLEN(NAME)
c      write(*,*)'..Itotal = ',itotal
      write(lst,360)NAME(:Itotal)
  360 format(' ',A)
C      write(LST,310)(specy_group(J)(:ILASCH(specy_GROUP(J))),
C     1           specy_group_qty(J)(:ILASCH(specy_GROUP(J))),J=1,Ngroup)
C  310 format(' ',A,'*',I)
C
C     DUMP USELESS SPECY-END INFO FOR MOLECULES AND RADICALS
C
C      IF (NKEY.EQ.2) THEN
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C      ELSE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C      ENDIF
C
      go to 100
 5000 write(*,*)'...end of conversion...'
      WRITE(LST,5100)
 5100 FORMAT('END')
      write(*,*)' '
      write(*,*)' new ThermCal input file created as - ',File_lst
      write(*,*)' '
      go to 7000
 6000 write(*,*)'...error in input file...'
 7000 continue
      end
C-----------------------------------------------------------------------C
      SUBROUTINE CKISUB (LINE, SUB, NSUB)
C
C     Generates an array of CHAR*(*) substrings from a CHAR*(*) string,
C     using blanks or tabs as delimiters
C
C     Input:  LINE  - a CHAR*(*) line
C     Output: SUB   - a CHAR*(*) array of substrings
C             NSUB  - number of substrings found
C     A '!' will comment out a line, or remainder of the line.
C                                        F. Rupley, Div. 8245, 5/15/86
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) SUB(*), LINE
      NSUB = 0
C
      DO 5 N = 1, LEN(LINE)
        IF (ICHAR(LINE(N:N)) .EQ. 9) LINE(N:N) = ' '
    5 CONTINUE
C
      IF (IPPLEN(LINE) .LE. 0) RETURN
```

```
C
      ILEN = ILASCH(LINE)
C
      NSTART = IFIRCH(LINE)
   10 CONTINUE
      ISTART = NSTART
      NSUB = NSUB + 1
      SUB(NSUB) = ' '
C
      DO 100 I = ISTART, ILEN
         ILAST = INDEX(LINE(ISTART:),' ') - 1
         IF (ILAST .GT. 0) THEN
            ILAST = ISTART + ILAST - 1
         ELSE
            ILAST = ILEN
         ENDIF
         SUB(NSUB) = LINE(ISTART:ILAST)
         IF (ILAST .EQ. ILEN) RETURN
C
         NSTART = ILAST + IFIRCH(LINE(ILAST+1:))
C
C        Does SUB have any slashes?
C
         I1 = INDEX(SUB(NSUB),'~')
         IF (I1 .LE. 0) THEN
            IF (LINE(NSTART:NSTART) .NE. '~') GO TO 10
            NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
            IND = INDEX(SUB(NSUB),' ')
            SUB(NSUB)(IND:) = LINE(NSTART:NEND)
            IF (NEND .EQ. ILEN) RETURN
            NSTART = NEND + IFIRCH(LINE(NEND+1:))
            GO TO 10
         ENDIF
C
C        Does SUB have 2 slashes?
C
         I2 = INDEX(SUB(NSUB)(I1+1:),'~')
         IF (I2 .GT. 0) GO TO 10
C
         NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
         IND = INDEX(SUB(NSUB),' ') + 1
         SUB(NSUB)(IND:) = LINE(NSTART:NEND)
         IF (NEND .EQ. ILEN) RETURN
         NSTART = NEND + IFIRCH(LINE(NEND+1:))
         GO TO 10
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPNPAR (LINE, NPAR, IPAR, ISTART)
C
C     Returns CHAR*(*) IPAR substring of CHAR*(*) string LINE which
C     contains NPAR real parameters
C
C     Input:     LINE - a CHAR*(*) line
C                NPAR - number of parameters expected
C     Output:    IPAR - the substring of parameters only
C                ISTART - the starting location of IPAR substring
C     A '!' will comment out a line, or remainder of the line.
C                                        F. Rupley, Div. 8245, 5/14/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
```

```
C*****END precision > single
C
      CHARACTER*(*) LINE,IPAR
C
C----------Find Comment String (! signifies comment)
C
      ILEN = IPPLEN(LINE)
      ISTART = 0
      N = 0
      IF (ILEN.GT.0) THEN
         DO 40 I = ILEN, 1, -1
            ISTART = I
            IPAR = ' '
            IPAR = LINE(ISTART:ILEN)
            IF (LINE(I:I).NE.' ') THEN
               IF (I .EQ. 1) RETURN
               IF (LINE(I-1:I-1) .EQ. ' ') THEN
                  N = N + 1
                  IF (N .EQ. NPAR) RETURN
               ENDIF
            ENDIF
   40    CONTINUE
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPPARI(STRING, ICARD, NEXPEC, IVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARI
C   REFER TO  IPGETI
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851725   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses integer variables from a character variable.  Called
C            by IPGETI, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARI may be used for parsing an input record that contains integer
C  values, but was read into a character variable instead of directly
C  into integer variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARI to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of IVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1,   2,,40000   , ,60'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set IVAL(1) = 1
C     (2) set IVAL(2) = 2
C     (3) leave IVAL(3) unchanged
C     (4) set IVAL(4) = 40000
C     (5) leave IVAL(5) unchanged
C     (6) set IVAL(6) = 60
```

```
C
C   IPPARI will print diagnostics on the default output device, if
C   desired.
C
C   IPPARI is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume IVAL = (0, 0, 0) and NEXPEC = 3 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ---------------------   ----    ------
C  '  2 ,   3 45 '         (2, 3, 45)               0       3
C  '2.15,,3'              (2, 0, 3)                1       0
C  '3X, 25, 2'            (0, 0, 0)                1       0
C  '10000'                (10000, 0, 0)            2       1
C
C      Assume IVAL = (0, 0, 0, 0) and NEXPEC = -4 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ---------------------   ----    ------
C  '1, 2'                 (1, 2)                   0       2
C  ',,37  400'            (0, 0, 37, 400)          0       4
C  ' 1,,-3,,5'            (1, 0, -3, 0)            3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  IVAL (I,O) - the integer value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to IVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARI
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C     IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
```

```
C*****END precision > single
C
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION IVAL(*)
      CHARACTER *8 FMT(14)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARI
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set false when a space follows
C--- an integer value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR .OR. NC .EQ. IE) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into integer array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      IES = NC - 1
      NCH = IES - IBS + 1
```

```
      DATA FMT/' (I1)', ' (I2)', ' (I3)', ' (I4)', ' (I5)',
     1   ' (I6)', ' (I7)', ' (I8)', ' (I9)', '(I10)',
     2   '(I11)', '(I12)', '(I13)', '(I14)'/
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) IVAL(NFOUND)
      GO TO 450
400   CONTINUE
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0)RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1    WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      SUBROUTINE IPPARR(STRING, ICARD, NEXPEC, RVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARR
C   REFER TO  IPGETR
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851625   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses real variables from a character variable.  Called
C            by IPGETR, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARR may be used for parsing an input record that contains real
C  values, but was read into a character variable instead of directly
C  into real variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARR to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of RVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1.,   2,,4.e-5   , ,6.e-6'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set RVAL(1) = 1.0
C     (2) set RVAL(2) = 2.0
```

```
C      (3) leave RVAL(3) unchanged
C      (4) set RVAL(4) = 4.0E-05
C      (5) leave RVAL(5) unchanged
C      (6) set RVAL(6) = 6.0E-06
C
C   IPPARR will print diagnostics on the default output device, if
C   desired.
C
C   IPPARR is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume RVAL = (0., 0., 0.) and NEXPEC = 3 on entry:
C
C   input string           RVAL on exit              IERR    NFOUND
C   ------------           ----------------------    ----    ------
C '  2.34e-3,  3 45.1'     (2.34E-03, 3.0, 45.1)       0       3
C '2,,3.-5'                (2.0, 0.0, 3.0E-05)         0       3
C ',1.4,0.028E4'           (0.0, 1.4, 280.0)           0       3
C '1.0, 2.a4, 3.0'         (1.0, 0.0, 0.0)             1       1
C '1.0'                    (1.0, 0.0, 0.0)             2       1
C
C      Assume RVAL = (0.,0.,0.,0.) and NEXPEC = -4 on entry:
C
C   input string           RVAL on exit              IERR    NFOUND
C   ------------           ----------------------    ----    ------
C '1.,2.'                  (1.0, 2.0)                  0       2
C ',,3  4.0'               (0.0, 0.0, 3.0, 4.0)        0       4
C '1,,3,,5.0'              (0.0, 0.0, 3.0, 0.0)        3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  RVAL (I,O) - the real value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to RVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARR
C*****precision > double
```

```
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION RVAL(*)
      CHARACTER *8 FMT(22)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARR
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set negative when a space follows
C--- a real value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into real array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
```

```
      ENDIF
C
      DATA FMT/     ' (E1.0)', ' (E2.0)', ' (E3.0)', ' (E4.0)',
     1   ' (E5.0)', ' (E6.0)', ' (E7.0)', ' (E8.0)', ' (E9.0)',
     2   '(E10.0)', '(E11.0)', '(E12.0)', '(E13.0)', '(E14.0)',
     3   '(E15.0)', '(E16.0)', '(E17.0)', '(E18.0)', '(E19.0)',
     4   '(E20.0)', '(E21.0)', '(E22.0)'/
      IES = NC - 1
      NCH = IES - IBS + 1
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) RVAL(NFOUND)
      GO TO 450
400   CONTINUE
      WRITE (LOUT, 555) STRING(IBS:IES)
  555 FORMAT (A)
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0) RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1   WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
```

```
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END
      FUNCTION ILASCH(STRING)
C   BEGIN PROLOGUE  ILASCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines last significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the last non-blank character in a string of
C  arbitrary length.  If no characters are found, ILASCH is set = 0.
C  When used with the companion routine IFIRCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C  Note that the FORTRAN intrinsic function LEN returns the length
C  of a character string as declared, rather than as filled.  The
C  declared length includes leading and trailing blanks, and thus is
C  not useful in generating 'significant' substrings.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) STRING
C
C***FIRST EXECUTABLE STATEMENT ILASCH
      NLOOP = LEN(STRING)
      IF (NLOOP.EQ.0) THEN
         ILASCH = 0
         RETURN
      ENDIF
C
      DO 100 I = NLOOP, 1, -1
         IF (STRING(I:I) .NE. ' ') GO TO 120
```

```
100   CONTINUE
C
120   CONTINUE
      ILASCH = I
      END
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKCOMP (IST, IRAY, II, I)
C
C  START PROLOGUE
C
C  SUBROUTINE CKCOMP (IST, IRAY, II, I)*
C     Returns the index of an element of a reference character
C     string array which corresponds to a character string;
C     leading and trailing blanks are ignored.
C
C
C  INPUT
C     IST   - A character string.
C                   Data type - CHARACTER*(*)
C     IRAY  - An array of character strings;
C             dimension IRAY(*) at least II
C                   Data type - CHARACTER*(*)
C     II    - The length of IRAY.
C                   Data type - integer scalar.
C
C  OUTPUT
C     I     - The first integer location in IRAY in which IST
C             corresponds to IRAY(I); if IST is not also an
C             entry in IRAY, I=0.
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) IST, IRAY(*)
C
      I = 0
      DO 10 N = II, 1, -1
         IS1 = IFIRCH(IST)
         IS2 = ILASCH(IST)
         IR1 = IFIRCH(IRAY(N))
         IR2 = ILASCH(IRAY(N))
         IF ( IS2.GE.IS1 .AND. IS2.GT.0 .AND.
     1        IR2.GE.IR1 .AND. IR2.GT.0 .AND.
     2        IST(IS1:IS2).EQ.IRAY(N)(IR1:IR2) ) I=N
   10 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKUNIT (LINE, AUNITS, EUNITS, IUNITS)
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
      CHARACTER*(*) LINE, IUNITS, AUNITS, EUNITS
      CHARACTER*4 UPCASE
```

```
C
      AUNITS = ' '
      EUNITS = ' '
      IUNITS = ' '
      DO 85 N = 1, ILASCH(LINE)-3
         IND = ILASCH(IUNITS)
         IF (EUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4)      .EQ. 'CAL/') THEN
               EUNITS = 'CAL/'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units cal/mole'
               ELSE
                  IUNITS(IND:) = ', E units cal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KCAL') THEN
               EUNITS = 'KCAL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kcal/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kcal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'JOUL') THEN
               EUNITS = 'JOUL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Joules/mole'
               ELSE
                  IUNITS(IND:) = ', E units Joules/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KJOU') THEN
               EUNITS = 'KJOU'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kjoule/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kjoule/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KELV') THEN
               EUNITS = 'KELV'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kelvins'
               ELSE
                  IUNITS(IND:) = ', E units Kelvins'
               ENDIF
            ENDIF
         ENDIF
         IF (AUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4) .EQ. 'MOLE') THEN
               IF (UPCASE(LINE(N+4:), 1) .EQ. 'S') THEN
                  AUNITS = 'MOLE'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units mole-cm-sec-K'
                  ELSE
                     IUNITS(IND:) = ', A units mole-cm-sec-K'
                  ENDIF
               ELSEIF (UPCASE(LINE(N+4:), 4) .EQ. 'CULE') THEN
                  AUNITS = 'MOLC'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units molecules'
                  ELSE
                     IUNITS(IND:) = ', A units molecules'
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
   85 CONTINUE
C
      IF (AUNITS .EQ. ' ') THEN
```

```
         AUNITS = 'MOLE'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', A units mole-cm-sec-K'
         ELSE
            IUNITS(IND:) = ' A units mole-cm-sec-K'
         ENDIF
      ENDIF
C
      IF (EUNITS .EQ. ' ') THEN
         EUNITS = 'CAL/'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', E units cal/mole'
         ELSE
            IUNITS(IND:) = ' E units cal/mole'
         ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      INTEGER FUNCTION IPPLEN (LINE)
C
C  BEGIN PROLOGUE
C
C  FUNCTION IPPLEN (LINE)
C     Returns the effective length of a character string, i.e.,
C     the index of the last character before an exclamation mark (!)
C     indicating a comment.
C
C  INPUT
C     LINE  - A character string.
C
C  OUTPUT
C     IPPLEN - The effective length of the character string.
C
C  END PROLOGUE
C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER LINE*(*)
C
      IN = IFIRCH(LINE)
      IF (IN.EQ.0 .OR. LINE(IN:IN).EQ.'!') THEN
         IPPLEN = 0
      ELSE
         IN = INDEX(LINE,'!')
         IF (IN .EQ. 0) THEN
            IPPLEN = ILASCH(LINE)
         ELSE
            IPPLEN = ILASCH(LINE(:IN-1))
         ENDIF
      ENDIF
      RETURN
      END
C
      CHARACTER*(*) FUNCTION UPCASE(ISTR, ILEN)
      CHARACTER ISTR*(*), LCASE(26)*1, UCASE(26)*1
```

```
      DATA LCASE /'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1            'n','o','p','q','r','s','t','u','v','w','x','y','z'/,
     2     UCASE /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     3            'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
C
      UPCASE = ' '
      UPCASE = ISTR(:ILEN)
      JJ = MIN (LEN(UPCASE), LEN(ISTR), ILEN)
      DO 10 J = 1, JJ
         DO 10 N = 1,26
            IF (ISTR(J:J) .EQ. LCASE(N)) UPCASE(J:J) = UCASE(N)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CKI2CH (NUM, STR, I, KERR)
C
C  START PROLOGUE
C
C  SUBROUTINE CKI2CH (NUM, STR, I, KERR)
C     Returns a character string representation of an integer
C     and the effective length of the string.
C
C  INPUT
C     NUM    - A number to be converted to a character string;
C              the maximum magnitude of NUM is machine-dependent.
C                   Data type - integer scalar.
C
C  OUTPUT
C     STR    - A left-justified character string representing NUM
C                   Data type - CHARACTER*(*)
C     I      - The effective length of the character string
C                   Data type - integer scalar
C     KERR   - Error flag;  character length errors will result in
C              KERR=.TRUE.
C                    Data type - logical
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STR*(*), IST(10)*(1)
      LOGICAL KERR
      DATA IST/'0','1','2','3','4','5','6','7','8','9'/
      BIGI = 2147483647.
C
      I = 0
      STR = ' '
      ILEN = LEN(STR)
      KERR = .FALSE.
      IF (ILEN.LT.1 .OR. ABS(NUM).GT.BIGI) THEN
         KERR = .TRUE.
         RETURN
      ENDIF
C
      IF (NUM .EQ. 0) THEN
         STR = '0'
         I = 1
         RETURN
      ELSEIF (NUM .LT. 0) THEN
         STR(1:) = '-'
      ENDIF
```

```
C
      INUM = ABS(NUM)
      NCOL = NINT(LOG10(REAL(INUM))) + 1
C
      DO 10 J = NCOL, 1, -1
         IDIV = INUM / 10.0**(J-1)
         IF (J.EQ.NCOL .AND. IDIV.EQ.0) GO TO 10
         LT = ILASCH(STR)
         IF (LT .EQ. ILEN) THEN
            STR = ' '
            KERR = .TRUE.
            RETURN
         ENDIF
         STR(LT+1:) = IST(IDIV+1)
         INUM = INUM - IDIV*10.0**(J-1)
   10 CONTINUE
      I = ILASCH(STR)
C
      RETURN
      END
C
C-----------------------------------------------------------------------C

      SUBROUTINE THERM_IO(ERROR)
C
      IMPLICIT NONE
      INCLUDE 'THM_PAR3.FI'
C
C      CHARACTER FILE_NAME*20
C      INTEGER IDOT, ISTART, IFIRCH
      LOGICAL ERROR
C
      ERROR = .FALSE.
    5 WRITE(*,10)
   10 FORMAT(' Please enter the input file name : ')
      READ(*,'(A20)')FILE_INP
      OPEN(INP,FILE=FILE_INP,STATUS='OLD',ERR=30)
      GO TO 50
   30 CONTINUE
      WRITE(*,40)FILE_INP
   40 FORMAT(' Can not find file : ',A20)
      GO TO 5
   50 CONTINUE
C
      ERROR = .FALSE.
   55 WRITE(*,110)
  110 FORMAT(' Please enter the output sorted file name : ')
      READ(*,'(A20)')FILE_OUT
      OPEN(LOUT,FILE=FILE_OUT,STATUS='UNKNOWN',ERR=130)
      CLOSE(LOUT,status='delete')
      OPEN(LOUT,FILE=FILE_OUT,STATUS='NEW',ERR=130)
      GO TO 150
  130 CONTINUE
      WRITE(*,140)FILE_OUT
  140 FORMAT(' File I/O Error : ',A20)
      GO TO 350
  150 CONTINUE
C
      FILE_LOG = 'THERMSRT.LOG'
C
      OPEN(LOG,FILE=FILE_LOG,STATUS='UNKNOWN',ERR=230)
      CLOSE(LOG,status='delete')
      OPEN(LOG,FILE=FILE_LOG,STATUS='NEW',ERR=230)
      GO TO 450
  230 CONTINUE
      WRITE(*,240)FILE_LOG
```

```
 240  FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 450
 350  CONTINUE
      ERROR = .TRUE.
 450  CONTINUE
C
      RETURN
      END
```

D.3 THERMCVT

```
C
      PROGRAM THERMCVT
C
C-----------------------------------------------------------------------C
c***************changed W.Ing 06-15-94, 08-01-95**********************
C
      IMPLICIT NONE
C
      include 'thm_par2.fi'
C      include 'thm_cfg.fi'
C
      CHARACTER specy_name*70, specy_formula*40,
     3          specy_group(Max_Specy_Group)*14,KEY2(3)*4,
     4          specy_element(Max_Element)*4, str_int*8,
     5          specy_comment*70, specy_phase, line*100, KEY(5)*5,
     6          SUB(20)*100, UPCASE*5, str*100, qty*100, NAME*200
C
      INTEGER  specy_group_qty(Max_Specy_Group), specy_symmetry,
     1         specy_element_qty(Max_Element), Ngroup,ILASCH,
     2         ILEN, IPPLEN, NSUB, NKEY, Isymmetry(5), NVAL, Ierr,
     3         I, Istr, Igroup(5), Ifirch, Islash, Iqty(5), Ifound,J,
     4         Index, Itotal, Iname, NKEY2, Nloop
C
      REAL*8 specy_Hf, specy_S, specy_Cp300, specy_Cp400, specy_Cp500,
     1       specy_Cp600, specy_Cp800, specy_Cp1000, specy_Cp1500,
     2       specy_rotor, rotor(5)
C
      LOGICAL therm, nasa, err, io_err
C
      DATA KEY/'UNITS','RADIC','XXX','END','!'/
      DATA KEY2/'SYMM','NROT','ENDS'/
C
C
C     Call Initialization subroutines, 02/95 ING
C
C      CALL TITLE
      CALL THERM_IO(io_err)
      if (io_err) GO TO 6000
C      CALL THERM_INIT(init_err)
C      if (init_err) GO TO 6000
C
      write(lst,*)'THERM BATCH INPUT FILE'
C      write(lst,50)
C   50 format(' SPECIES       Hf       S     Cp 300    400     500     60
C     10     800    1000     1500     DATE          ELEMENTS')
C
  100 CONTINUE
      specy_rotor = 0.0
      specy_symmetry = 1
      LINE = ' '
      READ (INP,'(A)',END=5000) LINE
  105 CONTINUE
      ILEN = IPPLEN(LINE)
      IF (ILEN .EQ. 0) GO TO 100
      READ (INP,'(A)',END=5000) LINE
      READ (INP,'(A)',END=5000) LINE
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      specy_name = SUB(1)
```

```
      specy_formula = SUB(2)
C
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C
C        IS THERE A KEYWORD?
C
      CALL CKCOMP ( UPCASE(SUB(1), 5) , KEY, 2, NKEY)
C      WRITE(*,*)UPCASE(SUB(1),3),KEY(1),NKEY
C      IF (NKEY .GT. 0) ITASK = 0
C
C      WRITE(*,*)UPCASE(SUB(1), 5)
C      WRITE(*,*)NKEY
      IF (NKEY.EQ.2) THEN
          READ (INP,'(A)',END=5000) LINE
          READ (INP,'(A)',END=5000) LINE
          READ (INP,'(A)',END=5000) LINE
      ELSEIF (NKEY.EQ.1) THEN
C     DO NOTHING FOR MOLECULES
      ELSE
C     ERROR FOR NEITHER MOLECULES NOR RADICALS
          GO TO 6000
      ENDIF
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      call IPPARI (SUB(2), -1, 1, Igroup, nval, Ierr, LOG)
      Ngroup = Igroup(1)
      IF (Ierr.NE.0 .OR. Ngroup.EQ.0) THEN
  160     CONTINUE
          WRITE(*,*)' specy "',specy_name,'" group number error !!!'
          READ (INP,'(A)',END=5000) LINE
          ILEN = IPPLEN(LINE)
          CALL CKISUB (LINE(:ILEN), SUB, NSUB)
          CALL CKCOMP ( UPCASE(SUB(1), 4) , KEY2, 3, NKEY2)
          IF (NKEY2.NE.3) GO TO 160
          GO TO 100
      ENDIF
C GO TO 6000
      READ (INP,'(A)',END=5000) LINE
      IF (Ngroup.LE.5) THEN
          DO 200 I = 1, Ngroup
              READ (INP,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              specy_group(I) = SUB(3)
              call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
              specy_group_qty(I) = Iqty(1)
  200     CONTINUE
      ELSE
          Nloop = INT(Ngroup/2.0)
C          write(*,*)'Ngroup = ',Ngroup,' Nloop = ',Nloop
          IF(MOD(Ngroup,2).EQ.0) THEN
              DO 210 I = 1, Nloop
                  READ (INP,'(A)',END=5000) LINE
                  ILEN = IPPLEN(LINE)
                  CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                  specy_group(I) = SUB(3)
                  call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(I) = Iqty(1)
                  specy_group(Nloop+I) = SUB(9)
                  call IPPARI (SUB(11), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(Nloop+I) = Iqty(1)
C         write(*,*)I,specy_group(I),specy_group_qty(I)
C         write(*,*)Nloop+I,specy_group(Nloop+I),specy_group_qty(Nloop+I)
```

```
  210         CONTINUE
          ELSE
              DO 220 I = 1, Nloop
                  READ (INP,'(A)',END=5000) LINE
                  ILEN = IPPLEN(LINE)
                  CALL CKISUB (LINE(:ILEN), SUB, NSUB)
                  specy_group(I) = SUB(3)
                  call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(I) = Iqty(1)
                  specy_group(Nloop+I+1) = SUB(9)
                  call IPPARI (SUB(11), -1, 1, Iqty, nval, Ierr, LOG)
                  specy_group_qty(Nloop+I+1) = Iqty(1)
  220         CONTINUE
              READ (INP,'(A)',END=5000) LINE
              ILEN = IPPLEN(LINE)
              CALL CKISUB (LINE(:ILEN), SUB, NSUB)
              specy_group(Nloop+1) = SUB(3)
              call IPPARI (SUB(5), -1, 1, Iqty, nval, Ierr, LOG)
              specy_group_qty(Nloop+1) = Iqty(1)
          ENDIF
      ENDIF
C          WRITE(LST,*)LINE
C      READ (INP,'(A)',END=5000) LINE
C      READ (INP,'(A)',END=5000) LINE
C      READ (INP,'(A)',END=5000) LINE
  250 CONTINUE
      READ (INP,'(A)',END=5000) LINE
      ILEN = IPPLEN(LINE)
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
      CALL CKCOMP ( UPCASE(SUB(1), 4) , KEY2, 3, NKEY2)
C      WRITE(*,*)UPCASE(SUB(1), 4)
      IF (NKEY2.EQ.0) GO TO 250
      IF (NKEY2.EQ.3) GO TO 280
      IF (NKEY2.EQ.2) THEN
          call IPPARR (SUB(2), -1, 1, rotor, nval, Ierr, LOG)
          IF (Ierr.NE. 0) GO TO 6000
          specy_rotor = rotor(1)
C          READ (INP,'(A)',END=5000) LINE
C          ILEN = IPPLEN(LINE)
C          CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C          call IPPARI (SUB(2), -1, 1, Isymmetry, nval, Ierr, LOG)
C          IF (Ierr.NE. 0) GO TO 6000
C          specy_symmetry = Isymmetry(1)
      ELSE
          call IPPARI (SUB(2), -1, 1, Isymmetry, nval, Ierr, LOG)
C          write(*,*)line
C          write(*,*)sub(2)
          IF (Ierr.NE. 0) GO TO 6000
          specy_symmetry = Isymmetry(1)
      ENDIF
      GO TO 250
C
  280 CONTINUE
      write(LST,300)specy_name,specy_formula,
     2              specy_symmetry,specy_rotor
  300 format(A12,' ',A12,' ',I4,' ',F4.1)
C      write(LST,300)specy_name(:ILASCH(specy_name)),
C     1              specy_formula(:ILASCH(specy_name)),
C     2              specy_symmetry,specy_rotor
C  300 format(A,' ',A,' ',I3,' ',F4.1)
c      Index = 1
      Itotal = 0
      DO 350 J = 1, Ngroup
          Itotal = Itotal + 1
          NAME(Itotal:Itotal) = ' '
          Index = Itotal + 1
```

```
          Iname = IPPLEN(specy_group(J))
          Itotal = Itotal + Iname
          NAME(Index:Itotal) = specy_group(J)
          NAME(Itotal+1:Itotal+1) = ' '
          CALL CKI2CH (specy_group_qty(J), str_int, Iname, err)
          NAME(Itotal+2:Itotal+Iname+1) = str_int
          Itotal = Itotal + 1 + Iname
  350 CONTINUE
      ILEN = IPPLEN(NAME)
c      write(*,*)'..Itotal = ',itotal
      write(lst,360)NAME(:Itotal)
  360 format(' ',A)
C      write(LST,310)(specy_group(J)(:ILASCH(specy_GROUP(J))),
C     1           specy_group_qty(J)(:ILASCH(specy_GROUP(J))),J=1,Ngroup)
C  310 format(' ',A,'*',I)
C
C     DUMP USELESS SPECY-END INFO FOR MOLECULES AND RADICALS
C
C      IF (NKEY.EQ.2) THEN
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C      ELSE
C          READ (INP,'(A)',END=5000) LINE
C          READ (INP,'(A)',END=5000) LINE
C      ENDIF
C
      go to 100
 5000 write(*,*)'...end of conversion...'
      WRITE(LST,5100)
 5100 FORMAT('END')
      write(*,*)' '
      write(*,*)' new ThermCal input file created as - ',File_lst
      write(*,*)' '
      go to 7000
 6000 write(*,*)'...error in input file...'
 7000 continue
      end
C-----------------------------------------------------------------------C
      SUBROUTINE CKISUB (LINE, SUB, NSUB)
C
C     Generates an array of CHAR*(*) substrings from a CHAR*(*) string,
C     using blanks or tabs as delimiters
C
C     Input:  LINE  - a CHAR*(*) line
C     Output: SUB   - a CHAR*(*) array of substrings
C             NSUB  - number of substrings found
C     A '!' will comment out a line, or remainder of the line.
C                                    F. Rupley, Div. 8245, 5/15/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) SUB(*), LINE
      NSUB = 0
C
      DO 5 N = 1, LEN(LINE)
        IF (ICHAR(LINE(N:N)) .EQ. 9) LINE(N:N) = ' '
    5 CONTINUE
C
```

```
      IF (IPPLEN(LINE) .LE. 0) RETURN
C
      ILEN = ILASCH(LINE)
C
      NSTART = IFIRCH(LINE)
   10 CONTINUE
      ISTART = NSTART
      NSUB = NSUB + 1
      SUB(NSUB) = ' '
C
      DO 100 I = ISTART, ILEN
         ILAST = INDEX(LINE(ISTART:),' ') - 1
         IF (ILAST .GT. 0) THEN
            ILAST = ISTART + ILAST - 1
         ELSE
            ILAST = ILEN
         ENDIF
         SUB(NSUB) = LINE(ISTART:ILAST)
         IF (ILAST .EQ. ILEN) RETURN
C
         NSTART = ILAST + IFIRCH(LINE(ILAST+1:))
C
C        Does SUB have any slashes?
C
         I1 = INDEX(SUB(NSUB),'~')
         IF (I1 .LE. 0) THEN
            IF (LINE(NSTART:NSTART) .NE. '~') GO TO 10
            NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
            IND = INDEX(SUB(NSUB),' ')
            SUB(NSUB)(IND:) = LINE(NSTART:NEND)
            IF (NEND .EQ. ILEN) RETURN
            NSTART = NEND + IFIRCH(LINE(NEND+1:))
            GO TO 10
         ENDIF
C
C        Does SUB have 2 slashes?
C
         I2 = INDEX(SUB(NSUB)(I1+1:),'~')
         IF (I2 .GT. 0) GO TO 10
C
         NEND = NSTART + INDEX(LINE(NSTART+1:),'~')
         IND = INDEX(SUB(NSUB),' ') + 1
         SUB(NSUB)(IND:) = LINE(NSTART:NEND)
         IF (NEND .EQ. ILEN) RETURN
         NSTART = NEND + IFIRCH(LINE(NEND+1:))
         GO TO 10
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPNPAR (LINE, NPAR, IPAR, ISTART)
C
C     Returns CHAR*(*) IPAR substring of CHAR*(*) string LINE which
C     contains NPAR real parameters
C
C     Input:     LINE - a CHAR*(*) line
C                NPAR - number of parameters expected
C     Output:    IPAR - the substring of parameters only
C                ISTART - the starting location of IPAR substring
C     A '!' will comment out a line, or remainder of the line.
C                                    F. Rupley, Div. 8245, 5/14/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
```

```
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) LINE,IPAR
C
C----------Find Comment String (! signifies comment)
C
      ILEN = IPPLEN(LINE)
      ISTART = 0
      N = 0
      IF (ILEN.GT.0) THEN
         DO 40 I = ILEN, 1, -1
            ISTART = I
            IPAR = ' '
            IPAR = LINE(ISTART:ILEN)
            IF (LINE(I:I).NE.' ') THEN
               IF (I .EQ. 1) RETURN
               IF (LINE(I-1:I-1) .EQ. ' ') THEN
                  N = N + 1
                  IF (N .EQ. NPAR) RETURN
               ENDIF
            ENDIF
   40    CONTINUE
      ENDIF
      RETURN
      END
C-------------------------------------------------------------------------C
      SUBROUTINE IPPARI(STRING, ICARD, NEXPEC, IVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARI
C   REFER TO  IPGETI
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851725   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses integer variables from a character variable.  Called
C            by IPGETI, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARI may be used for parsing an input record that contains integer
C  values, but was read into a character variable instead of directly
C  into integer variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARI to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of IVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1,   2,,40000   , ,60'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set IVAL(1) = 1
C     (2) set IVAL(2) = 2
C     (3) leave IVAL(3) unchanged
C     (4) set IVAL(4) = 40000
C     (5) leave IVAL(5) unchanged
```

```
C     (6) set IVAL(6) = 60
C
C   IPPARI will print diagnostics on the default output device, if
C   desired.
C
C   IPPARI is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume IVAL = (0, 0, 0) and NEXPEC = 3 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ----------------------  ----    ------
C   '  2 ,   3 45 '        (2, 3, 45)                0       3
C   '2.15,,3'              (2, 0, 3)                 1       0
C   '3X, 25, 2'            (0, 0, 0)                 1       0
C   '10000'                (10000, 0, 0)             2       1
C
C      Assume IVAL = (0, 0, 0, 0) and NEXPEC = -4 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ----------------------  ----    ------
C   '1, 2'                 (1, 2)                    0       2
C   ',,37  400'            (0, 0, 37, 400)           0       4
C   ' 1,,-3,,5'            (1, 0, -3, 0)             3       4
C
C   arguments: (I=input,O=output)
C   -----------------------------
C   STRING (I) - the character string to be parsed.
C
C   ICARD  (I) - data statement number, and error processing flag
C          < 0 : no error messages printed
C          = 0 : print error messages, but not ICARD
C          > 0 : print error messages, and ICARD
C
C   NEXPEC (I) - number of real variables expected to be input.  If
C          < 0, the number is unknown, and any number of values
C          between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C   PROMPT (I) - prompting string, character type.  A question
C          mark will be added to form the prompt at the screen.
C
C   IVAL (I,O) - the integer value or values to be modified.  On entry,
C        the values are printed as defaults.  The formal parameter
C        corresponding to IVAL must be dimensioned at least NEXPEC
C        in the calling program if NEXPEC > 1.
C
C   NFOUND (O) - the number of real values represented in STRING,
C          only in the case that there were as many or less than
C          NEXPEC.
C
C   IERR (O) - error flag:
C        = 0 if no errors found
C        = 1 syntax errors or illegal values found
C        = 2 for too few values found (NFOUND < NEXPEC)
C        = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C    REFERENCES  (NONE)
C    ROUTINES CALLED  IFIRCH,ILASCH
C    END PROLOGUE  IPPARI
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
```

```
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION IVAL(*)
      CHARACTER *8 FMT(14)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARI
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set false when a space follows
C--- an integer value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR .OR. NC .EQ. IE) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into integer array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      IES = NC - 1
```

```
      NCH = IES - IBS + 1
      DATA FMT/' (I1)', ' (I2)', ' (I3)', ' (I4)', ' (I5)',
     1   ' (I6)', ' (I7)', ' (I8)', ' (I9)', '(I10)',
     2   '(I11)', '(I12)', '(I13)', '(I14)'/
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) IVAL(NFOUND)
      GO TO 450
400   CONTINUE
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0)RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1    WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      SUBROUTINE IPPARR(STRING, ICARD, NEXPEC, RVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARR
C   REFER TO  IPGETR
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851625   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses real variables from a character variable.  Called
C            by IPGETR, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARR may be used for parsing an input record that contains real
C  values, but was read into a character variable instead of directly
C  into real variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARR to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of RVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1.,   2,,4.e-5   , ,6.e-6'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set RVAL(1) = 1.0
```

```
C      (2) set RVAL(2) = 2.0
C      (3) leave RVAL(3) unchanged
C      (4) set RVAL(4) = 4.0E-05
C      (5) leave RVAL(5) unchanged
C      (6) set RVAL(6) = 6.0E-06
C
C   IPPARR will print diagnostics on the default output device, if
C   desired.
C
C   IPPARR is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume RVAL = (0., 0., 0.) and NEXPEC = 3 on entry:
C
C   input string           RVAL on exit              IERR    NFOUND
C   ------------           ----------------------    ----    ------
C  '  2.34e-3,  3 45.1'    (2.34E-03, 3.0, 45.1)      0       3
C  '2,,3.-5'               (2.0, 0.0, 3.0E-05)        0       3
C  ',1.4,0.028E4'          (0.0, 1.4, 280.0)          0       3
C  '1.0, 2.a4, 3.0'        (1.0, 0.0, 0.0)            1       1
C  '1.0'                   (1.0, 0.0, 0.0)            2       1
C
C      Assume RVAL = (0.,0.,0.,0.) and NEXPEC = -4 on entry:
C
C   input string           RVAL on exit              IERR    NFOUND
C   ------------           ----------------------    ----    ------
C  '1.,2.'                 (1.0, 2.0)                 0       2
C  ',,3  4.0'              (0.0, 0.0, 3.0, 4.0)       0       4
C  '1,,3,,5.0'             (0.0, 0.0, 3.0, 0.0)       3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  RVAL (I,O) - the real value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to RVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARR
```

```
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION RVAL(*)
      CHARACTER *8 FMT(22)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARR
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set negative when a space follows
C--- a real value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into real array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
```

```
         GO TO 500
       ENDIF
C
       DATA FMT/     ' (E1.0)', ' (E2.0)', ' (E3.0)', ' (E4.0)',
      1    ' (E5.0)', ' (E6.0)', ' (E7.0)', ' (E8.0)', ' (E9.0)',
      2    '(E10.0)', '(E11.0)', '(E12.0)', '(E13.0)', '(E14.0)',
      3    '(E15.0)', '(E16.0)', '(E17.0)', '(E18.0)', '(E19.0)',
      4    '(E20.0)', '(E21.0)', '(E22.0)'/
       IES = NC - 1
       NCH = IES - IBS + 1
       ITEMP = ' '
       ITEMP = STRING(IBS:IES)
       READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) RVAL(NFOUND)
       GO TO 450
400    CONTINUE
       WRITE (LOUT, 555) STRING(IBS:IES)
  555  FORMAT (A)
       IERR = 1
       GO TO 510
450    CONTINUE
       NC = NC + 1
       IF (NC .LE. IE) GO TO 100
C
500    CONTINUE
       IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510    CONTINUE
C
       IF (IERR .EQ. 0 .OR. ICARD .LT. 0) RETURN
       IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
      1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
       IF (IERR .EQ. 1)
      1   WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
       IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
      1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
      2   '  NUMBER EXPECTED = ', NEXPEC
       IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
      1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
       END
C
       FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
```

```
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END
      FUNCTION ILASCH(STRING)
C   BEGIN PROLOGUE  ILASCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines last significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C----------------------------------------------------------------------
C  IFIRCH locates the last non-blank character in a string of
C  arbitrary length.  If no characters are found, ILASCH is set = 0.
C  When used with the companion routine IFIRCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C  Note that the FORTRAN intrinsic function LEN returns the length
C  of a character string as declared, rather than as filled.  The
C  declared length includes leading and trailing blanks, and thus is
C  not useful in generating 'significant' substrings.
C----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) STRING
C
C***FIRST EXECUTABLE STATEMENT ILASCH
      NLOOP = LEN(STRING)
      IF (NLOOP.EQ.0) THEN
         ILASCH = 0
         RETURN
      ENDIF
C
      DO 100 I = NLOOP, 1, -1
```

```
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
120   CONTINUE
      ILASCH = I
      END
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKCOMP (IST, IRAY, II, I)
C
C  START PROLOGUE
C
C  SUBROUTINE CKCOMP (IST, IRAY, II, I)*
C     Returns the index of an element of a reference character
C     string array which corresponds to a character string;
C     leading and trailing blanks are ignored.
C
C
C  INPUT
C     IST   - A character string.
C                   Data type - CHARACTER*(*)
C     IRAY  - An array of character strings;
C             dimension IRAY(*) at least II
C                   Data type - CHARACTER*(*)
C     II    - The length of IRAY.
C                   Data type - integer scalar.
C
C  OUTPUT
C     I     - The first integer location in IRAY in which IST
C             corresponds to IRAY(I); if IST is not also an
C             entry in IRAY, I=0.
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) IST, IRAY(*)
C
      I = 0
      DO 10 N = II, 1, -1
         IS1 = IFIRCH(IST)
         IS2 = ILASCH(IST)
         IR1 = IFIRCH(IRAY(N))
         IR2 = ILASCH(IRAY(N))
         IF ( IS2.GE.IS1 .AND. IS2.GT.0 .AND.
     1        IR2.GE.IR1 .AND. IR2.GT.0 .AND.
     2        IST(IS1:IS2).EQ.IRAY(N)(IR1:IR2) ) I=N
   10 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKUNIT (LINE, AUNITS, EUNITS, IUNITS)
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
      CHARACTER*(*) LINE, IUNITS, AUNITS, EUNITS
```

```
      CHARACTER*4 UPCASE
C
      AUNITS = ' '
      EUNITS = ' '
      IUNITS = ' '
      DO 85 N = 1, ILASCH(LINE)-3
         IND = ILASCH(IUNITS)
         IF (EUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4)     .EQ. 'CAL/') THEN
               EUNITS = 'CAL/'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units cal/mole'
               ELSE
                  IUNITS(IND:) = ', E units cal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KCAL') THEN
               EUNITS = 'KCAL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kcal/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kcal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'JOUL') THEN
               EUNITS = 'JOUL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Joules/mole'
               ELSE
                  IUNITS(IND:) = ', E units Joules/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KJOU') THEN
               EUNITS = 'KJOU'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kjoule/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kjoule/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KELV') THEN
               EUNITS = 'KELV'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kelvins'
               ELSE
                  IUNITS(IND:) = ', E units Kelvins'
               ENDIF
            ENDIF
         ENDIF
         IF (AUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4) .EQ. 'MOLE') THEN
               IF (UPCASE(LINE(N+4:), 1) .EQ. 'S') THEN
                  AUNITS = 'MOLE'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units mole-cm-sec-K'
                  ELSE
                     IUNITS(IND:) = ', A units mole-cm-sec-K'
                  ENDIF
               ELSEIF (UPCASE(LINE(N+4:), 4) .EQ. 'CULE') THEN
                  AUNITS = 'MOLC'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units molecules'
                  ELSE
                     IUNITS(IND:) = ', A units molecules'
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
   85 CONTINUE
C
```

```
      IF (AUNITS .EQ. ' ') THEN
         AUNITS = 'MOLE'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', A units mole-cm-sec-K'
         ELSE
            IUNITS(IND:) = ' A units mole-cm-sec-K'
         ENDIF
      ENDIF
C
      IF (EUNITS .EQ. ' ') THEN
         EUNITS = 'CAL/'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', E units cal/mole'
         ELSE
            IUNITS(IND:) = ' E units cal/mole'
         ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      INTEGER FUNCTION IPPLEN (LINE)
C
C  BEGIN PROLOGUE
C
C  FUNCTION IPPLEN (LINE)
C     Returns the effective length of a character string, i.e.,
C     the index of the last character before an exclamation mark (!)
C     indicating a comment.
C
C  INPUT
C     LINE  - A character string.
C
C  OUTPUT
C     IPPLEN - The effective length of the character string.
C
C  END PROLOGUE
C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER LINE*(*)
C
      IN = IFIRCH(LINE)
      IF (IN.EQ.0 .OR. LINE(IN:IN).EQ.'!') THEN
         IPPLEN = 0
      ELSE
         IN = INDEX(LINE,'!')
         IF (IN .EQ. 0) THEN
            IPPLEN = ILASCH(LINE)
         ELSE
            IPPLEN = ILASCH(LINE(:IN-1))
         ENDIF
      ENDIF
      RETURN
      END
C
      CHARACTER*(*) FUNCTION UPCASE(ISTR, ILEN)
```

```
      CHARACTER ISTR*(*), LCASE(26)*1, UCASE(26)*1
      DATA LCASE /'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1           'n','o','p','q','r','s','t','u','v','w','x','y','z'/,
     2     UCASE /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     3           'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
C
      UPCASE = ' '
      UPCASE = ISTR(:ILEN)
      JJ = MIN (LEN(UPCASE), LEN(ISTR), ILEN)
      DO 10 J = 1, JJ
         DO 10 N = 1,26
            IF (ISTR(J:J) .EQ. LCASE(N)) UPCASE(J:J) = UCASE(N)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CKI2CH (NUM, STR, I, KERR)
C
C  START PROLOGUE
C
C  SUBROUTINE CKI2CH (NUM, STR, I, KERR)
C     Returns a character string representation of an integer
C     and the effective length of the string.
C
C  INPUT
C     NUM    - A number to be converted to a character string;
C              the maximum magnitude of NUM is machine-dependent.
C                   Data type - integer scalar.
C
C  OUTPUT
C     STR    - A left-justified character string representing NUM
C                   Data type - CHARACTER*(*)
C     I      - The effective length of the character string
C                   Data type - integer scalar
C     KERR   - Error flag;  character length errors will result in
C              KERR=.TRUE.
C                    Data type - logical
C
C  END PROLOGUE
C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STR*(*), IST(10)*(1)
      LOGICAL KERR
      DATA IST/'0','1','2','3','4','5','6','7','8','9'/
      BIGI = 2147483647.
C
      I = 0
      STR = ' '
      ILEN = LEN(STR)
      KERR = .FALSE.
      IF (ILEN.LT.1 .OR. ABS(NUM).GT.BIGI) THEN
         KERR = .TRUE.
         RETURN
      ENDIF
C
      IF (NUM .EQ. 0) THEN
         STR = '0'
         I = 1
         RETURN
      ELSEIF (NUM .LT. 0) THEN
         STR(1:) = '-'
```

```
      ENDIF
C
      INUM = ABS(NUM)
      NCOL = NINT(LOG10(REAL(INUM))) + 1
C
      DO 10 J = NCOL, 1, -1
         IDIV = INUM / 10.0**(J-1)
         IF (J.EQ.NCOL .AND. IDIV.EQ.0) GO TO 10
         LT = ILASCH(STR)
         IF (LT .EQ. ILEN) THEN
            STR = ' '
            KERR = .TRUE.
            RETURN
         ENDIF
         STR(LT+1:) = IST(IDIV+1)
         INUM = INUM - IDIV*10.0**(J-1)
   10 CONTINUE
      I = ILASCH(STR)
C
      RETURN
      END
C
C-------------------------------------------------------------------------C


      SUBROUTINE THERM_IO(ERROR)
C
      IMPLICIT NONE
      INCLUDE 'THM_PAR2.FI'
C
      CHARACTER FILE_NAME*20
      INTEGER IDOT, ISTART, IFIRCH
      LOGICAL ERROR
C
      ERROR = .FALSE.
    5 WRITE(*,10)
   10 FORMAT(' Enter the file name for conversion : ')
      READ(*,'(A20)')FILE_INP
      OPEN(INP,FILE=FILE_INP,STATUS='OLD',ERR=30)
      GO TO 50
   30 CONTINUE
      WRITE(*,40)FILE_INP
   40 FORMAT(' Can not find file : ',A20)
      GO TO 5
   50 CONTINUE
      ISTART = IFIRCH(FILE_INP)
      IDOT = INDEX(FILE_INP,'.')
      FILE_NAME = FILE_INP(ISTART:IDOT-1)
C      FILE_LOG = FILE_NAME + '.' + 'LOG'
      FILE_LOG = 'THERMCVT.LOG'
C      FILE_LOG(IDOT:IDOT) = '.'
C      FILE_LOG(IDOT+1:IDOT+3) = 'LOG'
C      FILE_LST = FILE_NAME + '.' + 'LST'
      FILE_LST = FILE_NAME
      FILE_LST(IDOT:IDOT) = '.'
      FILE_LST(IDOT+1:IDOT+3) = 'THM'
C      FILE_DAT = FILE_NAME + '.' + 'DAT'
C
      OPEN(LOG,FILE=FILE_LOG,STATUS='UNKNOWN',ERR=230)
      GO TO 250
  230 CONTINUE
      WRITE(*,240)FILE_LOG
  240 FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 350
```

```
 250  CONTINUE
C
      OPEN(LST,FILE=FILE_LST,STATUS='UNKNOWN',ERR=330)
      GO TO 350
 330  CONTINUE
      WRITE(*,340)FILE_LST
 340  FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 350
 350  CONTINUE
C
      RETURN
      END
```

D.4 INTP4

```
C
      PROGRAM CKINTP
C
C-----------------------------------------------------------------------C
C     VERSION 4.0                W.ING
C        CHEBYSHEV POLYNOMIALS
C     VERSION 3.0
C     CHANGES FROM VERSION 1.0
C     1.  Changed from REAL*8 to DOUBLE PRECISION
C     CHANGES FROM VERSION 1.1
C     1.  Changed CHARACTER*100 to CHARACTER*80
C     2.  Added THERMO "ALL" option
C     3.  Write LENICK, LENRCK, LENCCK to linking file
C     4.  Allow reaction species to end in '=' or '-'
C     5.  Allow real values of elemental composition in THERMO cards
C     6.  Allow upper/lower case input
C     CHANGES FROM VERSION 1.2
C     1.  Reaction delimiters are now "=" or "<=>" if reversible,
C                                        " =>" if irreversible.
C     2.  Fixed an error with IFIRCH(LINE) in IPPLEN
C     CHANGES FROM VERSION 1.3
C     1.  Add "unix" change blocks
C     CHANGES FROM VERSION 1.4
C     1.  Modify OPEN statements
C     CHANGES FROM VERSION 1.5
C     1.  Correct molecules to moles unit conversion
C     2.  Correct UPCASE to avoid dimensioning errors
C     CHANGES FROM VERSION 1.7
C     1.  Further correction of molecules conversion for fall-off
C         and third-body reactions
C     CHANGES FOR VERSION 1.8
C     1.  Change Subroutine CKUNIT to parse LINE instead of SUB(*)
C         in order to correct misinterpretation of unit strings
C         with slashes.
C     CHANGES FOR VERSION 1.9
C     1.  First record of linking file now consists of a character
C         string version, precision, and logical error flag
C     CHANGES FOR VERSION 2.0
C     1.  Error in UPCASE could cause interpreter to ignore some
C         keywords.
C     CHANGES FOR VERSION 2.1
C     1.  10/18/90 (F. Rupley):
C         Error in scaling the pre-exponential constants RPAR(3,*)
C         where REV is declared, and FPAL(3,*) for fall-off reactions,
C         as RPAR(3,II)*EFAC should read RPAR(3,NREV), and
C            FPAL(3,II)*EFAC should read FPAL(3,NFAL).
C         This error was introduced in CKINTERP.15 during refinement
C         Dof units conversion routines.
C     2.  Subroutine CKDUP modified to recognize that two reactions
C         may be duplicate except for a third-body species in a
C         fall-off reaction.
C     CHANGES FOR VERSION 2.2
C     1.  11/14/90 (F. Rupley per M. Coltrin):
C         Initialize variable NCHRG
C      CHANGES FOR VERSION 2.3
C     1.  In CKPREAC, error correction of 10/18/90 (above, V2.1).
C     CHANGES FOR VERSION 2.4
C     1.  Additional checking of TLO,TMID,THI for species -
C         a) set initial values at -1.
C         b) if user has not provided a TLO,TMID, or THI, use the
```

```
C           values provided by THERMO.DAT.
C        c) check that TLO < THI, TLO <= TMID <= THI
C     CHANGES FOR VERSION 2.5
C     1.  Need to get TLO,THI,TMID from database BEFORE reading
C         user's THERMO data (unless THERMO ALL option is used)
C     CHANGES FOR VERSION 2.6
C     1.  LENRCK lengthened by II+NREV to reflect additional
C         work space needed by CKRAT for a 4th parameter
C         (perturbation factor).
C     CHANGES FOR VERSION 2.7
C     1.  Two otherwise duplicate reactions are unique if one
C         is a third body reaction and the other not.
C     CHANGES FOR VERSION 2.8
C     1.  Change output format to print all 16 characters for
C         a species name.
C     CHANGES FOR VERSION 2.9 (2/24/92 F. Rupley)
C     1.  Check that reverse (REV) parameters were given when
C         RTL reverse Teller-Landauer parameters are given.
C     2.  Add 2*II to length of real work space
C     CHANGES FOR VERSION 3.0 (4/13/92 F. Rupley per M. Coltrin)
C     1.  Correct logic in CKDUP, add argument to call list.
C
C
C     CKINTP interprets a formatted ASCII representation of a
C     chemical reaction mechanism and creates the binary file LINK
C     required by CHEMKIN.  CKINTP is dimensioned as follows:
c
c****************changed W.Ing 06-15-94, 08-01-95***********************
C
C     MDIM = maximum number of elements in a problem;              (20)
C     KDIM = maximum number of species in a problem;              (500)
C     MAXTP= maximum number of temperatures used to fit             (3)
C            thermodynamic properties of species
C     NPC  = number of polynomial coefficients to fits              (5)
C     NPCP2= number of fit coefficients for a temperature range     (7)
C     IDIM = maximum number of reactions in a mechanism;         (1500)
C     NPAR = number of Arrhenius parameters in a reaction;          (3)
C     NLAR = number of Landau-Teller parameters in a reaction;      (2)
C     NFAR = number of fall-off parameters in a reaction;          (70)
C     MAXSP= maximum number of species in a reaction                (6)
C     MAXTB= maximum number of third bodies for a reaction         (10)
C     LSYM = character string length of element and species names (16)
C
C     User input is read from LIN (Unit15), a thermodynamic database
C     is read from LTHRM (Unit17), printed output is assigned to LOUT
C     (Unit16), and binary linking data is written to LINC (Unit25).
C
C     REQUIRED ELEMENT INPUT: (Subroutine CKCHAR)           (DIMENSION)
C
C        The word 'ELEMENTS' followed by a list of element
C        names, terminated by the word 'END';
C
C        The resulting element data stored in LINK is:
C        MM       - integer number of elements found
C        ENAME(*) - CHARACTER*(*) array of element names         (MDIM)
C        AWT(*)   - real array of atomic weights;                (MDIM)
C                   default atomic weights are those on
C                   atomic weight charts; if an element
C                   is not on the periodic chart, or if
C                   it is desirable to alter its atomic
C                   weight, this value must be included
C                   after the element name, enclosed by
C                   slashed, i.e., D/2.014/
C
C     REQUIRED SPECIES INPUT: (Subroutine CKCHAR)
C
```

```
C        The word 'SPECIES' followed by a list of species
C        names, terminated by the word 'END';
C
C        The resulting species data stored in LINK is:
C        KK        - integer number of species found
C        KNAME(*) - CHARACTER*(*) array of species names          (KDIM)
C
C     OPTIONAL THERMODYNAMIC DATA: (Subroutine CKTHRM)
C     (If this feature is not used, thermodynamic properties are
C     obtained from a CHEMKIN database.)  The format for this option
C     is the word 'THERMO' followed by any number of 4-line data sets:
C
C     Line 1: species name, optional comments, elemental composition,
C             phase, T(low), T(high), T(mid), additional elemental
C             composition, card number (col. 80);
C             format(A10,A14,4(A2,I3),A1,E10.0,E10.0,E8.0,(A2,I3),I1)
C     Line 2: coefficients a(1--5) for upper temperature range,
C             card number (col. 80);
C             format(5(e15.0),I1)
C     Line 3: coefficients a(6--7) for upper temperature range,
C             coefficients a(1--3) for lower temperature range,
C             card number (col. 80);
C             format(5(e15.0),I1)
C     Line 4: coefficients a(4--7) for lower temperature range,
C             card number (col. 80);
C             format(4(e15.0),I1)
C
C     End of THERMO data is indicated by 'END' line or new keyword.
C
C        The resulting thermodynamic data stored in LINK are:
C        WTM(*)    - real array of molecular weights              (KDIM)
C        KNCF(*,*)- integer composition of species          (MDIM,KDIM)
C        KPHSE(*) - integer phase of a species;                 (KDIM)
C                   -1(solid), 0(gas), +1(liquid).
C        KCHRG(*) - ionic charge of a species;                  (KDIM)
C                   = 0 except in presence/absence of electrons
C                   = +n in absence of n electrons
C                   = -n in presence of n electons
C        NCHRG     - integer number of species with KCHRG<>0
C        NT(*)     - array of number of temperatures used       (KDIM)
C                   in fits
C        T(*,*)    - array of temperatures used in fits     (MAXTP,KDIM)
C        A(N,L,K) - Thermodynamic properties for       (NPC+2,NTR,KDIM)
C                   species K consists of polynomial
C                   coefficients for fits to
C                   CP/R = SUM (A(N,L,K)*Temperature**(N-1), N=1,NPC+2)
C                          where  T(L,K) <= Temperature < T(L+1,K),
C                   and,
C                   N=NPC+1 is formation enthalpy HO/R = A(NPC+1,L,K),
C                   N=NPC+2 is formation entropy  SO/R = A(NPC+2,L,K)
C
C     OPTIONAL REACTION INPUT:
C     Reaction data is input after all ELEMENT, SPECIES and THERMO
C     data in the following format:
C
C     1) (Subroutine CKREAC)
C        The first line contains the keyword 'REACTIONS' and an
C        optional description of units:
C
C           'MOLES' - (default), pre-exponential units are moles-sec-K;
C           'MOLECULES' - pre-exponential units are molecules and
C                         will be converted to moles.
C           'KELVINS' - activation energies are Kelvins, else the
C                       activation energies are converted to Kelvins;
C           'CAL/MOLE' - (default), activation energies are cal/mole;
C           'KCAL/MOLE' - activation energies are Kcal/mole;
```

```
C           'JOULES/MOLE' - activation energies are joules/mole;
C           'KJOULES/MOLE' - activation energies are Kjoules/mole.
C
C        A description of each reaction is expected to follow.
C        Required format for a reaction is a list of '+'-delimited
C        reactants, followed by a list of '+'-delimited reactants,
C        each preceded by its stoichiometric coefficient if greater
C        than 1;  separating the reactants from the products is a '='
C        if reversible reaction, else a '=>'.  Following the reaction
C        string on the same line are the space-delimited Arrhenius
C        coefficients.
C
C        If the reaction contains a third body, this is indicated by
C        by the presence of an 'M' as a reactant or product or both,
C        and enhancement factors for third-bodies may be defined on
C        additional lines as described in (2).
C
C        If the reaction contains a radiation wavelength, this is
C        indicated by the presence of an 'HV' either as a reactant
C        or as a product.  Unless otherwise defined on additional
C        lines as described in (2), the value of the wavelength is
C        -1.0 if a reactant or +1.0 if a product.
C
C        If the reaction is a fall-off reaction, this is indicated
C        either by a '(+M)' or a '(+KNAME(K))', and there must be
C        additional lines as described in (2) to define fall-off
C        parameters.
C
C     2)  (Subroutine CKAUXL)
C        Additional information for a reaction is given on lines
C        immediately following the reaction description; this data
C        will consist of a 'keyword' to denote the type of data,
C        followed by a '/', then the required parameters for the
C        keyword, followed by another '/'.  There may be more than
C        one keyword per line, and there may be any number of lines.
C        The keywords and required parameters are as follows:
C
C        KNAME(K)/efficiency value/ - species (K) is an enhanced
C                third body in the reaction
C        HV/wavelength/ - radiation wavelength parameter
C        LT/val1 val2/ - Landau-Teller coefficients
C        LOW/val1 val2 val3/ - low fall-off parameters
C        TROE/val1 val2 val3 val4/ - Troe fall-off parameters;
C                                    if val4 is omitted, a default
C                                    parameter will be used
C
C        ADDED W.ING, 06/15/95**********************************
C
C        EXTROE/val1 val2 val3 val4 val5/ -
C                                   Extended Troe fall-off parameters;
C
C                                   val1-val4 = Troe parameters
C                                   val5 = well # (for cal. ext reduced
C                                                  pressure)
C
C        ***********************************************************
C
C        SRI/val1 val2 val3 val4/ - SRI fall-off parameters;
C                                   if val4 is omitted, a default
C                                   parameter will be used
C           (it is an error to have both LT and Fall-off defined)
C        REV/par1 par2 par3/ - reverse parameters given
C        RLT/val1 val2/ - Landau-Teller coefficients for reverse
C           (it is an error if REV given and not RLT)
C
C     The end of all reaction data is indicated by an 'END' card or
```

```
C     <eof>.
C
C     Resulting reaction data stored in LINC are:
C       II         - integer number of reactions found
C       PAR(*,*)   - array of real Arrhenius coefficients   (NPAR,IDIM)
C       NSPEC(*)   - total number of species in a reaction       (IDIM)
C                    if NSPEC < 0, reaction is irreversible
C       NREAC(*)   - number of reactants only                    (IDIM)
C       NUNK(*,*)  - array of species numbers for reaction (MAXSP,IDIM)
C       NU(*,*)    - array of stoichiometric coefficients  (MAXSP,IDIM)
C                    of species in a reaction, negative=reactant,
C                    positive=product
C
C       NWL        - number of reactions with radiation wavelength
C       IWL(*)     - integer reaction numbers                    (IDIM)
C       WL(*)      - real radiation wavelengths                  (IDIM)
C
C       NTHB       - number of reactions with third bodies
C       ITHB       - integer reaction numbers                    (IDIM)
C       NTBS(*)    - total number of enhanced species for NTHB   (IDIM)
C       NKTB(*,*)  - species numbers of enhanced species   (MAXTB,IDIM)
C       AIK(*,*)   - enhancement factors                   (MAXTB,IDIM)
C
C       NFAL       - number of fall-off reactions
C       IFAL(*)    - integer reaction numbers                    (IDIM)
C       KFAL(*)    - integer species number for which
C                    concentrations are a factor in fall-off
C                    calculation
C       IFOP(*)    - integer fall-off type number                (IDIM)
C                    = 0 if fall-off reaction is found
C                    = 1 for Lindemann form
C                    = 2 for 6-parameter Troe form
C                    = 3 for 7-parameter Troe form
C                    = 4 for SRI form
C                    = 5 for 8-parameter Extended Troe form
C       PFAL(*,*)  - fall-off parameters                   (NFAR,IDIM)
C
C       NLAN       - number of reactions with Landau-Teller
C       ILAN       - integer reaction numbers                    (IDIM)
C       PLAN       - Landau-Teller parameters                 (NLAR,IDIM)
C
C       NREV       - number of reactions with reverse parameters
C       IREV(*)    - integer reaction numbers                    (IDIM)
C       RPAR(*,*)  - parameters                               (NPAR,IDIM)
C
C       NRLT       - number of reactions with reverse parameters
C                    and Landau-Teller parameters
C       IRLT(*)    - integer reaction numbers                    (IDIM)
C       RLAN(*,*)  - reverse Teller-Laudauer parameters       (NLAR,IDIM)
C
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      PARAMETER (MDIM=20, KDIM=500, MKDIM=MDIM*KDIM, IDIM=1500, LSYM=16,
     1           NPAR=3, NPIDIM=IDIM*NPAR, NPC=5, NPCP2=NPC+2, MAXTP=3,
     2           NTR=MAXTP-1, NKTDIM=NTR*NPCP2*KDIM, MAXSP=6, MAXTB=10,
     3           NLAR=2, NSIDIM=MAXSP*IDIM, NTIDIM=MAXTB*IDIM,
     4           NLIDIM=NLAR*IDIM, NFAR=70, NFIDIM=NFAR*IDIM,
     5           NTDIM=KDIM*MAXTP, NIDIM=9*IDIM, LIN=5, LOUT=6,
     6           LTHRM=17, LINC=25)
C
```

```
      CHARACTER KNAME(KDIM)*(LSYM), ENAME(MDIM)*(LSYM), SUB(80)*80,
     1          KEY(5)*4, LINE*80, IUNITS*80, AUNITS*4, EUNITS*4,
     2          UPCASE*4, VERS*(LSYM), PREC*(LSYM)
C
      DIMENSION AWT(MDIM), KNCF(MDIM,KDIM), WTM(KDIM), KPHSE(KDIM),
     1          KCHRG(KDIM), A(NPCP2,NTR,KDIM), T(MAXTP,KDIM), NT(KDIM),
     2          NSPEC(IDIM), NREAC(IDIM), NU(MAXSP,IDIM),
     3          NUNK(MAXSP,IDIM), PAR(NPAR,IDIM), IDUP(IDIM),IREV(IDIM),
     4          RPAR(NPAR,IDIM), ILAN(IDIM), PLAN(NLAR,IDIM),
     5          IRLT(IDIM), RLAN(NLAR,IDIM), IWL(IDIM),  WL(IDIM),
     6          IFAL(IDIM), IFOP(IDIM), KFAL(IDIM), PFAL(NFAR,IDIM),
     7          ITHB(IDIM),NTBS(IDIM),AIK(MAXTB,IDIM),NKTB(MAXTB,IDIM)
      DIMENSION VALUE(5)
C
      LOGICAL KERR, THERMO, ITHRM(KDIM)
C
C     Initialize variables
C
      DATA KEY/'ELEM','SPEC','THER','REAC','END'/, KERR/.FALSE./,
     1     ITASK,NCHRG,MM,KK,II,NLAN,NFAL,NTHB,NREV,NRLT,NWL/11*0/,
     2     ENAME,AWT/MDIM*' ',MDIM*0.0/, THERMO/.TRUE./,
     3     T/NTDIM*-1.0/, KNAME,WTM,NT,KPHSE,KCHRG,ITHRM
     4     /KDIM*' ', KDIM*0.0, KDIM*3, KDIM*0, KDIM*0, KDIM*.FALSE./,
     5     WL,IFOP,NTBS,IDUP /IDIM*0.0, IDIM*-1, IDIM*0, IDIM*0/,
     6     NSPEC,NREAC,IREV,ILAN,IRLT,IWL,IFAL,KFAL,ITHB/NIDIM*0/
C
      DATA NUNK,NU/NSIDIM*0, NSIDIM*0/, NKTB,AIK/NTIDIM*0,NTIDIM*-1.0/
      DATA PAR,RPAR/NPIDIM*0.0, NPIDIM*0.0/
      DATA PLAN,RLAN/NLIDIM*0.0, NLIDIM*0.0/
      DATA PFAL/NFIDIM*0.0/, KNCF/MKDIM*0.0/, A/NKTDIM*0.0/
C-----------------------------------------------------------------------C
C

C
C     ADD FILEIO SUBROUTINE 02/94 ING
C
      CALL FILEIO_INT
C
C
C


      VERS = '3.0'
      WRITE  (LOUT, 15) VERS(:3)
   15 FORMAT (/
     1' CHEMKIN INTERPRETER OUTPUT: CHEMKIN-II Version ',A,' Jun. 1991'
C*****precision > double
     2/'                                 DOUBLE PRECISION'/)
      PREC = 'DOUBLE'
C*****END precision > double
C*****precision > single
C     2/'                                  SINGLE PRECISION'/)
C      PREC = 'SINGLE'
C*****END precision > single
C
C        START OF MECHANISM INTERPRETATION
C
  100 CONTINUE
      LINE = ' '
      READ (LIN,'(A)',END=5000) LINE
  105 CONTINUE
      ILEN = IPPLEN(LINE)
      IF (ILEN .EQ. 0) GO TO 100
C
      CALL CKISUB (LINE(:ILEN), SUB, NSUB)
C
C        IS THERE A KEYWORD?
```

```
C
      CALL CKCOMP ( UPCASE(SUB(1), 4) , KEY, 5, NKEY)
      IF (NKEY .GT. 0) ITASK = 0
C
      IF (NKEY.EQ.1 .OR. NKEY.EQ.2) THEN
C
C        ELEMENT OR SPECIES DATA
C
         ITASK = NKEY
         IF (NSUB .EQ. 1) GO TO 100
C
         DO 25 N = 2, NSUB
            SUB(N-1) = ' '
            SUB(N-1) = SUB(N)
   25    CONTINUE
         NSUB = NSUB-1
C
      ELSEIF (NKEY .EQ. 3) THEN
C
C        THERMODYNAMIC DATA
C
         IF (NSUB .GT. 1) THEN
            IF ( UPCASE(SUB(2), 3) .EQ. 'ALL') THEN
               THERMO = .FALSE.
               READ (LIN,'(A)') LINE
               CALL IPPARR (LINE, -1, 3, VALUE, NVAL, IER, LOUT)
               IF (NVAL .NE. 3 .OR. IER.NE.0) THEN
                  KERR = .TRUE.
                  WRITE (LOUT, 333)
               ELSE
                  TLO = VALUE(1)
                  TMID = VALUE(2)
                  THI = VALUE(3)
               ENDIF
            ENDIF
         ELSE
C
C           USE THERMODYNAMIC DATABASE FOR DEFAULT TLO,TMID,THI
C*****OPEN statement > vms
C            OPEN (LTHRM, STATUS='OLD', FORM='FORMATTED', READONLY,
C     1            SHARED)
C*****END OPEN statement > vms
C*****OPEN statement > unix
C            OPEN (LTHRM, FORM='FORMATTED', FILE='thermdat')
C*****END OPEN statement > unix
C
            READ (LTHRM,'(A)') LINE
            READ (LTHRM,'(A)') LINE
            CALL IPPARR (LINE, -1, 3, VALUE, NVAL, IER, LOUT)
            IF (NVAL .NE. 3 .OR. IER.NE.0) THEN
               KERR = .TRUE.
               WRITE (LOUT, 333)
            ELSE
               TLO = VALUE(1)
               TMID = VALUE(2)
               THI = VALUE(3)
            ENDIF
            REWIND (LTHRM)
            CLOSE (LTHRM)
         ENDIF
C
         CALL CKTHRM (LIN, MDIM, ENAME, MM, AWT, KNAME, KK, KNCF,
     1                KPHSE, KCHRG, WTM, MAXTP, NT, NTR, TLO, TMID,
     2                THI, T, NPCP2, A, ITHRM, KERR, LOUT, LINE)
C
         IF (.NOT. THERMO)
```

```
     1      CALL CKPRNT (MDIM, MAXTP, MM, ENAME, KK, KNAME, WTM, KPHSE,
     2                   KCHRG, NT, T, TLO, TMID, THI, KNCF, ITHRM,
     3                   LOUT, KERR)
         I1 = IFIRCH(LINE)
         IF (UPCASE(LINE(I1:), 4) .EQ. 'REAC') GO TO 105
C
       ELSEIF (NKEY .EQ. 4) THEN
C
         ITASK = 4
C        START OF REACTIONS; ARE UNITS SPECIFIED?
         CALL CKUNIT (LINE(:ILEN), AUNITS, EUNITS, IUNITS)
C
         IF (THERMO) THEN
C
C           THERMODYNAMIC DATA
C*****OPEN statement > vms
C            OPEN (LTHRM, STATUS='OLD', FORM='FORMATTED', READONLY,
C     1            SHARED)
C*****END OPEN statement > vms
C*****OPEN statement > unix
C            OPEN (LTHRM, FORM='FORMATTED', FILE='thermdat')
C*****END OPEN statement > unix
C
            READ (LTHRM,'(A)') LINE
            READ (LTHRM,'(A)') LINE
            CALL IPPARR (LINE, -1, 3, VALUE, NVAL, IER, LOUT)
            IF (NVAL .NE. 3 .OR. IER.NE.0) THEN
               KERR = .TRUE.
               WRITE (LOUT, 333)
            ELSE
               TLO = VALUE(1)
               TMID = VALUE(2)
               THI = VALUE(3)
            ENDIF
            CALL CKTHRM (LTHRM, MDIM, ENAME, MM, AWT, KNAME, KK, KNCF,
     1                   KPHSE, KCHRG, WTM, MAXTP, NT, NTR, TLO, TMID,
     2                   THI, T, NPCP2, A, ITHRM, KERR, LOUT, LINE)
            CALL CKPRNT (MDIM, MAXTP, MM, ENAME, KK, KNAME, WTM, KPHSE,
     1                   KCHRG, NT, T, TLO, TMID, THI, KNCF, ITHRM,
     2                   LOUT, KERR)
            THERMO = .FALSE.
         ENDIF
C
         WRITE (LOUT, 1800)
         GO TO 100
       ENDIF
C
       IF (ITASK .EQ. 1) THEN
C
C        ELEMENT DATA
C
         IF (MM .EQ. 0) THEN
            WRITE (LOUT, 200)
            WRITE (LOUT, 300)
            WRITE (LOUT, 200)
         ENDIF
C
         IF (NSUB .GT. 0) THEN
            M1 = MM +1
            CALL CKCHAR (SUB, NSUB, MDIM, ENAME, AWT, MM, KERR, LOUT)
            DO 110 M = M1, MM
               IF (AWT(M) .LE. 0) CALL CKAWTM (ENAME(M), AWT(M))
               WRITE (LOUT, 400) M,ENAME(M)(:4),AWT(M)
               IF (AWT(M) .LE. 0) THEN
                  KERR = .TRUE.
                  WRITE (LOUT, 1000) ENAME(M)
```

```
               ENDIF
  110        CONTINUE
           ENDIF
C
        ELSEIF (ITASK .EQ. 2) THEN
C
C          PROCESS SPECIES DATA
C
           IF (KK .EQ. 0) WRITE (LOUT, 200)
           IF (NSUB .GT. 0)
     1     CALL CKCHAR (SUB, NSUB, KDIM, KNAME, WTM, KK, KERR, LOUT)
C
        ELSEIF (ITASK .EQ. 4) THEN
C
C          PROCESS REACTION DATA
C
           IND = 0
           DO 120 N = 1, NSUB
              IND = MAX(IND, INDEX(SUB(N),'/'))
              IF (UPCASE(SUB(N), 3) .EQ. 'DUP') IND = MAX(IND,1)
  120      CONTINUE
           IF (IND .GT. 0) THEN
C
C             AUXILIARY REACTION DATA
C
              CALL CKAUXL (SUB, NSUB, II, KK, KNAME, LOUT, MAXSP, NPAR,
     1                     NSPEC, NTHB, ITHB, NTBS, MAXTB, NKTB, AIK,
     2                     NFAL, IFAL, IDUP, NFAR, PFAL, IFOP, KFAL, NLAN,
     3                     ILAN, NLAR, PLAN, NREV, IREV, RPAR,
     4                     NRLT, IRLT, RLAN, NWL, IWL, WL, KERR)
C
           ELSE
C
C             THIS IS A REACTION STRING
C
              IF (II .LT. IDIM) THEN
C
                 IF (II .GT. 0)
C
C                CHECK PREVIOUS REACTION FOR COMPLETENESS
C
     1           CALL CPREAC (II, MAXSP, NSPEC, NPAR, PAR, RPAR,
     2                        AUNITS, EUNITS, NREAC, NUNK, NU, KCHRG,
     3                        MDIM, MM, KNCF, IDUP, NFAL, IFAL, KFAL,
     4                        NFAR, PFAL, IFOP, NREV, IREV, NTHB, ITHB,
     5                        NLAN, ILAN, NRLT, IRLT, KERR, LOUT)
C
C                NEW REACTION
C
                 II = II+1
                 CALL CKREAC (LINE(:ILEN), II, KK, KNAME, LOUT, MAXSP,
     1                        NSPEC, NREAC, NUNK, NU, NPAR, PAR,
     2                        NTHB, ITHB, NFAL, IFAL, KFAL, NWL,
     3                        IWL, WL, KERR)
C
              ELSE
                 WRITE (LOUT, 1070)
                 KERR = .TRUE.
              ENDIF
C
           ENDIF
        ENDIF
        GO TO 100
C
 5000 CONTINUE
C
```

```
C     END OF INPUT
C
      IF (II .GT. 0) THEN
C
C               CHECK FINAL REACTION FOR COMPLETENESS
C
           CALL CPREAC (II, MAXSP, NSPEC, NPAR, PAR, RPAR, AUNITS,
     1                  EUNITS, NREAC, NUNK, NU, KCHRG, MDIM, MM,
     2                  KNCF, IDUP, NFAL, IFAL, KFAL, NFAR, PFAL, IFOP,
     3                  NREV, IREV, NTHB, ITHB, NLAN, ILAN, NRLT,
     4                  IRLT, KERR, LOUT)
C
C               CHECK REACTIONS DECLARED AS DUPLICATES
C
          DO 500 I = 1, II
             IF (IDUP(I) .LT. 0) THEN
                KERR = .TRUE.
                WRITE (LOUT, 1095) I
             ENDIF
  500     CONTINUE
C
          WRITE (LOUT, '(/1X,A)') ' NOTE: '//IUNITS(:ILASCH(IUNITS))
C
      ELSEIF (THERMO) THEN
C
C        THERE WAS NO REACTION DATA, MAKE SURE SPECIES DATA IS COMPLETE
C
C*****OPEN statement > vms
C             OPEN (LTHRM, STATUS='OLD', FORM='FORMATTED', READONLY,
C     1             SHARED)
C*****END OPEN statement > vms
C*****OPEN statement > unix
C             OPEN (LTHRM, FORM='FORMATTED', FILE='thermdat')
C*****END OPEN statement > unix
C
          READ (LTHRM,'(A)') LINE
          READ (LTHRM,'(A)') LINE
          CALL IPPARR (LINE, -1, 3, VALUE, NVAL, IER, LOUT)
          IF (NVAL .NE. 3 .OR. IER.NE.0) THEN
             KERR = .TRUE.
             WRITE (LOUT, 333)
          ELSE
             TLO = VALUE(1)
             TMID = VALUE(2)
             THI = VALUE(3)
          ENDIF
          CALL CKTHRM (LTHRM, MDIM, ENAME, MM, AWT, KNAME, KK, KNCF,
     1                 KPHSE, KCHRG, WTM, MAXTP, NT, NTR, TLO, TMID,
     2                 THI, T, NPCP2, A, ITHRM, KERR, LOUT, LINE)
          CALL CKPRNT (MDIM, MAXTP, MM, ENAME, KK, KNAME, WTM, KPHSE,
     1                 KCHRG, NT, T, TLO, TMID, THI, KNCF, ITHRM,
     2                 LOUT, KERR)
       ENDIF
C
       DO 1150 K = 1, KK
          IF (KCHRG(K) .NE. 0) NCHRG = NCHRG+1
 1150  CONTINUE
C
       LENICK = 1 + (3 + MM)*KK + (2 + 2*MAXSP)*II + NLAN + NRLT
     1            + 3*NFAL + (2 + MAXTB)*NTHB + NREV + NWL
       LENCCK = MM + KK
       LENRCK = 3 + MM + KK*(5 + MAXTP + NTR*NPCP2) + II*7 + NREV
     1            + NPAR*(II + NREV) + NLAR*(NLAN + NRLT)
     2            + NFAR*NFAL + MAXTB*NTHB + NWL
C
C     OPEN LINKING FILE
```

```
C
C*****OPEN statement > vms
C      OPEN (LINC, STATUS='NEW', FORM='UNFORMATTED')
C*****END OPEN statement > vms
C*****OPEN statement > unix
C      OPEN (LINC, FORM='UNFORMATTED', FILE='cklink')
C*****END OPEN statement > unix
C
      WRITE (LINC) VERS, PREC, KERR
      WRITE (LINC) LENICK, LENRCK, LENCCK, MM, KK, II, MAXSP,
     1             MAXTB, MAXTP, NPC, NPAR, NLAR, NFAR, NREV, NFAL,
     2             NTHB, NLAN, NRLT, NWL, NCHRG
      WRITE (LINC) (ENAME(M), AWT(M), M = 1, MM)
      WRITE (LINC) (KNAME(K), (KNCF(M,K),M=1,MM), KPHSE(K),
     1              KCHRG(K), WTM(K), NT(K), (T(L,K),L=1,MAXTP),
     2              ((A(M,L,K), M=1,NPCP2), L=1,NTR), K = 1, KK)
C
      IF (II .GT. 0) THEN
C
         WRITE (LINC) (NSPEC(I), NREAC(I), (PAR(N,I), N = 1, NPAR),
     1          (NU(M,I), NUNK(M,I), M = 1, MAXSP), I = 1, II)
C
         IF (NREV .GT. 0) WRITE (LINC)
     1      (IREV(N),(RPAR(L,N),L=1,NPAR),N=1,NREV)
C
         IF (NFAL .GT. 0) WRITE (LINC)
     1      (IFAL(N),IFOP(N),KFAL(N),(PFAL(L,N),L=1,NFAR), N = 1, NFAL)
C
         IF (NTHB .GT. 0) WRITE (LINC)
     1      (ITHB(N),NTBS(N),(NKTB(M,N),AIK(M,N),M=1,MAXTB),N=1,NTHB)
C
         IF (NLAN .GT. 0) WRITE (LINC)
     1      (ILAN(N), (PLAN(L,N), L = 1, NLAR), N = 1, NLAN)
C
         IF (NRLT .GT. 0) WRITE (LINC)
     1      (IRLT(N), (RLAN(L,N), L = 1, NLAR), N=1,NRLT)
C
         IF (NWL .GT. 0) WRITE (LINC) (IWL(N), WL(N), N = 1, NWL)
      ENDIF
C
      IF (KERR) THEN
C
         WRITE (LOUT, '(//A)')
     1   ' WARNING...THERE IS AN ERROR IN THE LINKING FILE'
C
      ELSE
C
         IF (II .LE. 0) WRITE (LOUT, '(/A)')
     1      ' WARNING...NO REACTION INPUT FOUND; ',
     2      ' LINKING FILE HAS NO REACTION INFORMATION ON IT.'
C
         WRITE (LOUT, '(///A)')
     1   ' NO ERRORS FOUND ON INPUT...CHEMKIN LINKING FILE WRITTEN.'
C
         WRITE (LOUT, '(/A,3(/A,I6))')
     1      ' WORKING SPACE REQUIREMENTS ARE',
     2      '    INTEGER:    ',LENICK,
     3      '    REAL:       ',LENRCK,
     4      '    CHARACTER:  ',LENCCK
      ENDIF
C-----------------------------------------------------------------------C
C
C     FORMATS
C
  200 FORMAT (26X,20('-'))
  300 FORMAT (26X,'ELEMENTS',5X,'ATOMIC',/26X,'CONSIDERED',3X,'WEIGHT')
```

```
  333 FORMAT (/6X,'Error...no TLO,TMID,THI given for THERMO ALL...'/)
  400 FORMAT (25X,I3,'. ',A4,G15.6)
C
 1000 FORMAT (6X,'Error...no atomic weight for element ',A)
 1070 FORMAT (6X,'Error...more than IDIM reactions...')
 1095 FORMAT (6X,'Error...no duplicate declared for reaction no.',I3)
 1800 FORMAT (///54X, '(k = A T**b exp(-E/RT))',/,
     1        6X,'REACTIONS CONSIDERED',30X,'A',8X,'b',8X,'E',/)
C
C
C
      CLOSE(LOUT)
      CLOSE(LINC)
C
C
C
      STOP
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKCHAR (SUB, NSUB, NDIM, STRAY, RAY, NN, KERR, LOUT)
C
C     Extracts names and real values from an array of CHAR*(*)
C     substrings; stores names in STRAY array, real values in RAY;
C     i.e. can be used to store element and atomic weight data,
C     species names, etc.
C
C     Input:   SUB(N),N=1,NSUB  - array of CHAR*(*) substrings
C              NSUB             - number of substrings
C              NDIM             - size of STRAY,RAY arrays
C              NN               - actual number of STRAY found
C              STRAY(N),N=1,NN  - CHAR*(*) array
C              RAY(N),N=1,NN    - Real array
C              LOUT             - output unit for error messages
C     Output:  NN               - incremented if more STRAY found
C              STRAY(N),N=1,NN  - incremented array of STRAY
C              RAY(N),N=1,NN    - incremented array of reals
C              KERR             - logical, .TRUE. = error in data
C
C                                      F. Rupley, Div. 8245, 2/5/88
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION RAY(*), PAR(1)
      CHARACTER SUB(*)*(*), STRAY(*)*(*), ISTR*80, UPCASE*4
      LOGICAL KERR
C
      ILEN = LEN(STRAY(1))
C
      DO 200 N = 1, NSUB
         IF ( UPCASE(SUB(N), 3) .EQ. 'END') RETURN
         ISTR = ' '
         I1 = INDEX(SUB(N),'/')
         IF (I1 .EQ .1) THEN
            KERR = .TRUE.
            WRITE (LOUT, 130) SUB(N)(:ILASCH(SUB(N)))
         ELSE
            IF (I1 .LE. 0) THEN
               ISTR = SUB(N)
            ELSE
               ISTR = SUB(N)(:I1-1)
            ENDIF
```

```
            CALL CKCOMP (ISTR, STRAY, NN, INUM)
C
            IF (INUM .GT. 0) THEN
               WRITE (LOUT, 100) SUB(N)(:ILASCH(SUB(N)))
            ELSE
               IF (NN .LT. NDIM) THEN
                  IF (ISTR(ILEN+1:) .NE. ' ') THEN
                     WRITE (LOUT, 120) SUB(N)(:ILASCH(SUB(N)))
                     KERR = .TRUE.
                  ELSE
                     NN = NN + 1
                     STRAY(NN) = ' '
                     STRAY(NN) = ISTR(:ILEN)
                     IF (I1 .GT. 0) THEN
                        I2 = I1 + INDEX(SUB(N)(I1+1:),'/')
                        ISTR = ' '
                        ISTR = SUB(N)(I1+1:I2-1)
                        CALL IPPARR (ISTR, 1, 1, PAR, NVAL, IER, LOUT)
                        IF (IER .EQ. 0) THEN
                           RAY(NN) = PAR(1)
                        ELSE
                           KERR = .TRUE.
                        ENDIF
                     ENDIF
                  ENDIF
               ELSE
                  WRITE (LOUT, 110) SUB(N)(:ILASCH(SUB(N)))
                  KERR = .TRUE.
               ENDIF
            ENDIF
         ENDIF
  200 CONTINUE
C
  100 FORMAT (6X,'Warning...duplicate array element ignored...',A)
  110 FORMAT (6X,'Error...character array size too small for  ...',A)
  120 FORMAT (6X,'Error...character array element name too long...',A)
  130 FORMAT (6X,'Error...misplaced value...',A)
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKAWTM (ENAME, AWT)
C
C     Returns atomic weight of element ENAME.
C     Input:   ENAME - CHAR*(*) element name
C     Output:  AWT   - real atomic weight
C
C                                        F. Rupley, Div. 8245, 11/11/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      PARAMETER (NATOM = 102)
      DIMENSION ATOM(NATOM)
      CHARACTER ENAME*(*), IATOM(NATOM)*2, UPCASE*2
C
      DATA (IATOM(I),ATOM(I),I=1,40) /
     *'H ',  1.00797, 'HE',  4.00260, 'LI',  6.93900, 'BE',  9.01220,
     *'B ', 10.81100, 'C ', 12.01115, 'N ', 14.00670, 'O ', 15.99940,
     *'F ', 18.99840, 'NE', 20.18300, 'NA', 22.98980, 'MG', 24.31200,
     *'AL', 26.98150, 'SI', 28.08600, 'P ', 30.97380, 'S ', 32.06400,
     *'CL', 35.45300, 'AR', 39.94800, 'K ', 39.10200, 'CA', 40.08000,
     *'SC', 44.95600, 'TI', 47.90000, 'V ', 50.94200, 'CR', 51.99600,
     *'MN', 54.93800, 'FE', 55.84700, 'CO', 58.93320, 'NI', 58.71000,
```

```
     *'CU', 63.54000, 'ZN', 65.37000, 'GA', 69.72000, 'GE', 72.59000,
     *'AS', 74.92160, 'SE', 78.96000, 'BR', 79.90090, 'KR', 83.80000,
     *'RB', 85.47000, 'SR', 87.62000, 'Y ', 88.90500, 'ZR', 91.22000/
C
      DATA (IATOM(I),ATOM(I),I=41,80) /
     *'NB', 92.90600, 'MO', 95.94000, 'TC', 99.00000, 'RU',101.07000,
     *'RH',102.90500, 'PD',106.40000, 'AG',107.87000, 'CD',112.40000,
     *'IN',114.82000, 'SN',118.69000, 'SB',121.75000, 'TE',127.60000,
     *'I ',126.90440, 'XE',131.30000, 'CS',132.90500, 'BA',137.34000,
     *'LA',138.91000, 'CE',140.12000, 'PR',140.90700, 'ND',144.24000,
     *'PM',145.00000, 'SM',150.35000, 'EU',151.96000, 'GD',157.25000,
     *'TB',158.92400, 'DY',162.50000, 'HO',164.93000, 'ER',167.26000,
     *'TM',168.93400, 'YB',173.04000, 'LU',174.99700, 'HF',178.49000,
     *'TA',180.94800, 'W ',183.85000, 'RE',186.20000, 'OS',190.20000,
     *'IR',192.20000, 'PT',195.09000, 'AU',196.96700, 'HG',200.59000/
C
      DATA (IATOM(I),ATOM(I),I=81,NATOM) /
     *'TL',204.37000, 'PB',207.19000, 'BI',208.98000, 'PO',210.00000,
     *'AT',210.00000, 'RN',222.00000, 'FR',223.00000, 'RA',226.00000,
     *'AC',227.00000, 'TH',232.03800, 'PA',231.00000, 'U ',238.03000,
     *'NP',237.00000, 'PU',242.00000, 'AM',243.00000, 'CM',247.00000,
     *'BK',249.00000, 'CF',251.00000, 'ES',254.00000, 'FM',253.00000,
     *'D ',002.01410, 'E',5.45E-4/
C
      CALL CKCOMP ( UPCASE(ENAME, 2), IATOM, NATOM, L)
      IF (L .GT. 0) AWT = ATOM(L)
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKTHRM (LUNIT, MDIM, ENAME, MM, AWT, KNAME, KK, KNCF,
     1                   KPHSE, KCHRG, WTM, MAXTP, NT, NTR, TLO, TMID,
     2                   THI, T, NPCP2, A, ITHRM, KERR, LOUT, ISTR)
C
C     Finds thermodynamic data and elemental composition for species
C     Input:  LUNIT  - unit number for input of thermo properties
C             MDIM   - maximum number of elements allowed
C             ENAME(M),M=1,MM  - array of CHAR*(*) element names
C             MM     - total number of elements declared
C             AWT(M),M=1,MM    - array of atomic weights for elements
C             KNAME(K),K=1,KK  - array of CHAR*(*) species names
C             KK     - total number of species declared
C             LOUT   - output unit for messages
C             NT(K),K=1,KK - number of temperature values
C             NTR - number of temperature ranges
C     Output: KNCF(M,K) - elemental composition of species
C             KPHSE(K),K=1,KK - integer array, species phase
C             KCHRG(K),K=1,KK - integer array of species charge
C                      =0, if no electrons,
C                      =(-1)*number of electrons present
C             WTM(K),K=1,KK - array of molecular weights of species
C             A(M,L,K)- array of thermodynamic coefficients
C             T(N),N=1,NT - array of temperatures
C             KERR   - logical error flag
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION WTM(*), NT(*), T(MAXTP,*), KPHSE(*), KNCF(MDIM,*),
     1          KCHRG(*), A(NPCP2,NTR,*), AWT(*), VALUE(5)
      CHARACTER ENAME(*)*(*), KNAME(*)*(*), LINE(4)*80, ELEM*16
      CHARACTER UPCASE*4, ISTR*80, SUB(80)*80
      LOGICAL KERR, ITHRM(*)
```

```
C
      IF (MM.LE.0 .OR. KK.LE.0) WRITE (LOUT, 80)
C
      GO TO 20
   10 CONTINUE
      ISTR = ' '
      READ (LUNIT,'(A)',END=40) ISTR
   20 CONTINUE
      ILEN = IPPLEN(ISTR)
      IF (ILEN .LE. 0) GO TO 10
C
      CALL CKISUB (ISTR(:ILEN), SUB, NSUB)
      CALL CKCOMP (SUB(1), KNAME, KK, K)
      IF (K .EQ. 0) THEN
         IF (UPCASE(SUB(1), 3) .EQ. 'END' .OR.
     1       UPCASE(SUB(1), 4) .EQ. 'REAC') RETURN
         GO TO 10
      ENDIF
C
      IF (ITHRM(K)) GO TO 10
      ITHRM(K) = .TRUE.
      LINE(1) = ' '
      LINE(1) = ISTR
      DO 25 L = 2, 4
         LINE(L) = ' '
         READ (LUNIT,'(A)',END=40) LINE(L)
   25 CONTINUE
C
      ICOL = 20
      DO 60 I = 1, 5
         ICOL = ICOL + 5
         IF (I .EQ. 5) ICOL = 74
         ELEM  = LINE(1)(ICOL:ICOL+1)
         IELEM = 0
C
         IF (LINE(1)(ICOL+2:ICOL+4) .NE. ' ') THEN
            CALL IPPARR
     1      (LINE(1)(ICOL+2:ICOL+4), 0, 1, VALUE, NVAL, IER, LOUT)
            IELEM = VALUE(1)
         ENDIF
C
         IF (ELEM.NE.' ' .AND. IELEM.NE.0) THEN
            IF (UPCASE(ELEM, 1) .EQ. 'E')
     1           KCHRG(K)=KCHRG(K)+IELEM*(-1)
            CALL CKCOMP (ELEM, ENAME, MM, M)
            IF (M .GT. 0) THEN
               KNCF(M,K) = IELEM
               WTM(K) = WTM(K) + AWT(M)*FLOAT(IELEM)
            ELSE
               WRITE (LOUT, 100) ELEM,KNAME(K)(:10)
               KERR = .TRUE.
            ENDIF
         ENDIF
   60 CONTINUE
C
      IF (UPCASE(LINE(1)(45:),1) .EQ. 'L') KPHSE(K)=1
      IF (UPCASE(LINE(1)(45:),1) .EQ. 'S') KPHSE(K)=-1
C
C-----Currently allows for three temperatures, two ranges;
C     in future, NT(K) may vary, NTR = NT(K)-1
C
      T(1,K) = TLO
      IF (LINE(1)(46:55) .NE. ' ') CALL IPPARR
     1   (LINE(1)(46:55), 0, 1, T(1,K), NVAL, IER, LOUT)
C
      T(2,K) = TMID
```

```
      IF (LINE(1)(66:73) .NE. ' ') CALL IPPARR
     1   (LINE(1)(66:73), 0, 1, T(2,K), NVAL, IER, LOUT)
C
      T(NT(K),K) = THI
      IF (LINE(1)(56:65) .NE. ' ') CALL IPPARR
     1   (LINE(1)(56:65), 0, 1, T(NT(K),K), NVAL, IER, LOUT)
C
      READ (LINE(2)(:75),'(5E15.8)') (A(I,NTR,K),I=1,5)
      READ (LINE(3)(:75),'(5E15.8)')
     1            (A(I,NTR,K),I=6,7),(A(I,1,K),I=1,3)
      READ (LINE(4)(:60),'(4E15.8)') (A(I,1,K),I=4,7)
      GO TO 10
C
   40 RETURN
   80 FORMAT (6X,'Warning...THERMO cards misplaced will be ignored...')
  100 FORMAT (6X,'Error...element...',A,'not declared for...',A)
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKREAC (LINE, II, KK, KNAME, LOUT, MAXSP, NSPEC, NREAC,
     1                   NUNK, NU, NPAR, PAR, NTHB, ITHB,
     2                   NFAL, IFAL, KFAL, NWL, IWL, WL, KERR)
C
C     CKREAC parses the main CHAR*(*) line representing a gas-phase
C     reaction; first, the real Arrhenius parameters are located and
C     stored in PAR(N,I),N=1,NPAR, where I is the reaction number;
C     then a search is made over the reaction string:
C
C     '=','<=>': reaction I is reversible;
C     '=>'     : reaction I is irreversible;
C
C     '(+[n]KNAME(K))': reaction I is a fall-off reaction;
C                       NFAL is incremented, the total number of
C                       fall-off reactions;
C                       IFAL(NFAL)=I, KFAL(NFAL)=K;
C                       this species is eliminated from consideration
C                       as a reactant or product in this reaction.
C
C     '(+M)'   : reaction I is a fall-off reaction;
C                NFAL is incremented, IFAL(NFAL)=I, KFAL(NFAL)=0;
C
C     '+[n]KNAME(K)': NSPEC(I) is incremented, the total number of
C                     species for this reaction;
C                     n is an optional stoichiometric coefficient
C                     of KNAME(K), if omitted, n=1;
C                     if this string occurs before the =/-,
C                     NREAC(I) is incremented, the total number of
C                     reactants for this reaction, NUNK(N,I)=K, and
C                     NU(N,I) = -n, where N=1-3 is reserved for
C                     reactants;
C                     if this string occurs after the =/-,
C                     NUNK(N,I) = K, and NU(N,I) = n, where N=4-6
C                     is reserved for products;
C
C     '+M' : I is a third-body reaction; NTHB is incremented, the
C            total number of third-body reactions, and ITHB(NTHB)=I.
C
C     Input:  LINE  - a CHAR*(*) line (from data file)
C             II    - the index of this reaction, and the total number
C                     of reactions found so far.
C             KK    - actual integer number of species
C             KNAME(K),K=1,KK - array of CHAR*(*) species names
C             LOUT  - output unit for error messages
C             MAXSP - maximum number of species allowed in reaction
C             NPAR  - number of parameters expected
C     A '!' will comment out a line, or remainder of the line.
C
```

```
C     Output: NSPEC - total number of reactants+products in reaction
C             NREAC - number of reactants
C             NUNK  - species numbers for the NSPEC species
C             NU    - stoichiometric coefficients for the NSPEC spec.
C             NFAL  - total number of fall-off reactions
C             IFAL  - reaction numbers for the NFAL reactions
C             KFAL  - 3rd body species numbers for the NFAL reactions
C             NTHB  - total number of 3rd-body reactions
C             ITHB  - reaction numbers for the NTHB reactions
C             NWL   - number of radiation-enhanced reactions
C             IWL   - reaction numbers for the NWL reactions
C             WL    - radiation wavelengths for the NWL reactions
C             KERR  - logical, .TRUE. = error in data file
C
C                                        F. Rupley, Div. 8245, 5/13/86
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), NREAC(*), NUNK(MAXSP,*), NU(MAXSP,*),
     1          PAR(NPAR,*), IFAL(*), KFAL(*), ITHB(*), IWL(*), WL(*),
     2          IPLUS(20)
      CHARACTER KNAME(*)*(*), LINE*(*), INUM(10)*1, UPCASE*4
      CHARACTER*80 ISTR, IREAC, IPROD, ISPEC, INAME, ITEMP
      LOGICAL KERR, LTHB, LWL
      DATA INUM/'0','1','2','3','4','5','6','7','8','9'/
C
      LTHB = .FALSE.
      LWL = .FALSE.
      NSPEC(II) = 0
      NREAC(II) = 0
C
C----------Find NPAR real parameters-----------------------
C
      CALL IPNPAR (LINE, NPAR, ISTR, ISTART)
      CALL IPPARR (ISTR, 1, NPAR, PAR(1,II), NVAL, IER, LOUT)
      IF (IER .NE. 0) KERR = .TRUE.
C
C-----Remove blanks from reaction string
C
      INAME = ' '
      ILEN = 0
      DO 10 I = 1, ISTART-1
         IF (LINE(I:I) .NE. ' ') THEN
            ILEN = ILEN+1
            INAME(ILEN:ILEN) = LINE(I:I)
         ENDIF
   10 CONTINUE
C
C-----Find reaction string, product string
C
      I1 = 0
      I2 = 0
      DO 25 I = 1, ILEN
         IF (I1 .LE. 0) THEN
            IF (INAME(I:I+2) .EQ. '<=>') THEN
               I1 = I
               I2 = I+2
               IR = 1
            ELSEIF (INAME(I:I+1) .EQ. '=>') THEN
               I1 = I
               I2 = I+1
```

```
                IR = -1
              ELSEIF (I.GT.1 .AND. INAME(I:I).EQ.'='
     1                     .AND. INAME(I-1:I-1).NE.'=') THEN
                I1 = I
                I2 = I
                IR = 1
              ENDIF
            ENDIF
   25 CONTINUE
C
      IF (ILASCH(INAME).GE.45 .AND. I1.GT.0) THEN
         WRITE (LOUT, 1900) II,INAME(:I1-1),(PAR(N,II),N=1,NPAR)
         WRITE (LOUT, 1920) INAME(I1:)
      ELSE
          WRITE (LOUT, 1900) II,INAME(:45),(PAR(N,II),N=1,NPAR)
      ENDIF
C
      IREAC = ' '
      IPROD = ' '
      IF (I1 .GT. 0) THEN
         IREAC = INAME(:I1-1)
         IPROD = INAME(I2+1:)
      ELSE
C
C-----did not find delimiter
C
         WRITE (LOUT, 660)
         KERR = .TRUE.
         RETURN
      ENDIF
C
      IF (INDEX(IREAC,'=>').GT.0 .OR. INDEX(IPROD,'=>').GT.0) THEN
C
C-----more than one '=>'
C
         WRITE (LOUT, 800)
         KERR = .TRUE.
         RETURN
      ENDIF
C
C-----Is this a fall-off reaction?
C
      IF (INDEX(IREAC,'(+').GT.0 .OR. INDEX(IPROD,'(+').GT.0) THEN
         KRTB = 0
         KPTB = 0
         DO 300 J = 1, 2
            ISTR = ' '
            KTB  = 0
            IF (J .EQ. 1) THEN
               ISTR = IREAC
            ELSE
               ISTR = IPROD
            ENDIF
C
            DO 35 N = 1, ILASCH(ISTR)-1
               IF (ISTR(N:N+1) .EQ. '(+') THEN
                  I1 = N+2
                  I2 = I1 + INDEX(ISTR(I1:),')')-1
                  IF (I2 .GT. I1) THEN
                     IF (ISTR(I1:I2-1).EQ.'M' .OR.
     1                   ISTR(I1:I2-1).EQ.'m') THEN
                        IF (KTB .NE. 0) THEN
                           WRITE (LOUT, 630)
                           KERR = .TRUE.
                           RETURN
                        ELSE
```

```
                              KTB = -1
                           ENDIF
                        ELSE
                           CALL CKCOMP (ISTR(I1:I2-1), KNAME, KK, KNUM)
                           IF (KNUM .GT. 0) THEN
                              IF (KTB .NE. 0) THEN
                                 WRITE (LOUT, 630)
                                 KERR = .TRUE.
                                 RETURN
                              ELSE
                                 KTB = KNUM
                              ENDIF
                           ENDIF
                        ENDIF
                        IF (KTB .NE. 0) THEN
                           ITEMP = ' '
                           IF (I1 .EQ. 1) THEN
                              ITEMP = ISTR(I2+1:)
                           ELSE
                              ITEMP = ISTR(:I1-3)//ISTR(I2+1:)
                           ENDIF
                           IF (J .EQ. 1) THEN
                              IREAC = ' '
                              IREAC = ITEMP
                              KRTB = KTB
                           ELSE
                              IPROD = ' '
                              IPROD = ITEMP
                              KPTB = KTB
                           ENDIF
                        ENDIF
                     ENDIF
                  ENDIF
   35          CONTINUE
  300       CONTINUE
C
          IF (KRTB.NE.0 .OR. KPTB.NE.0) THEN
C
C            does product third-body match reactant third-body
C
             IF (KRTB.LE.0 .AND. KPTB.LE.0) THEN
C
                NFAL = NFAL + 1
                IFAL(NFAL) = II
                KFAL(NFAL) = 0
C
                LTHB = .TRUE.
                NTHB = NTHB + 1
                ITHB(NTHB) = II
C
             ELSEIF (KRTB .EQ. KPTB) THEN
                NFAL = NFAL + 1
                IFAL(NFAL) = II
                KFAL(NFAL) = KRTB
C
             ELSE
C
                WRITE (LOUT, 640)
                KERR = .TRUE.
                RETURN
             ENDIF
          ENDIF
       ENDIF
C
C----------Find reactants, products-------------------------
C
```

```
      DO 600 J = 1, 2
         ISTR = ' '
         LTHB = .FALSE.
         IF (J .EQ. 1) THEN
            ISTR = IREAC
            NS = 0
         ELSE
            ISTR = IPROD
            NS = 3
         ENDIF
C
C-----------store pointers to '+'-signs
C
         NPLUS = 1
         IPLUS(NPLUS) = 0
         DO 500 L = 2, ILASCH(ISTR)-1
            IF (ISTR(L:L).EQ.'+') THEN
               NPLUS = NPLUS + 1
               IPLUS(NPLUS) = L
            ENDIF
  500    CONTINUE
         NPLUS = NPLUS + 1
         IPLUS(NPLUS) = ILASCH(ISTR)+1
C
         NSTART = 1
  505    CONTINUE
         N1 = NSTART
         DO 510 N = NPLUS, N1, -1
            ISPEC = ' '
            ISPEC = ISTR(IPLUS(N1)+1 : IPLUS(N)-1)
C
            IF (UPCASE(ISPEC, 1) .EQ. 'M') THEN
               IF (LTHB) THEN
                  WRITE (LOUT, 900)
                  KERR = .TRUE.
                  RETURN
               ELSEIF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) THEN
                  WRITE (LOUT, 640)
                  KERR = .TRUE.
                  RETURN
               ELSE
                  LTHB = .TRUE.
                  IF (NTHB.EQ.0 .OR.
     1               (NTHB.GT.0.AND.ITHB(NTHB).NE.II)) THEN
                     NTHB = NTHB + 1
                     ITHB(NTHB) = II
                  ENDIF
                  IF (N .EQ. NPLUS) GO TO 600
                  NSTART = N
                  GO TO 505
               ENDIF
C
            ELSEIF (UPCASE(ISPEC, 2) .EQ. 'HV') THEN
               IF (LWL) THEN
                  WRITE (LOUT, 670)
                  KERR = .TRUE.
                  RETURN
               ELSE
                  LWL = .TRUE.
                  NWL = NWL + 1
                  IWL(NWL) = II
                  WL(NWL) = 1.0
                  IF (J .EQ. 1) WL(NWL) = -1.0
                  IF (N .EQ. NPLUS) GO TO 600
                  NSTART = N
                  GO TO 505
```

```
               ENDIF
            ENDIF
C
C-----------does this string start with a one- or two-digit number?
C
            IVAL = 0
            CALL CKCOMP (ISPEC(1:1), INUM, 10, I1)
            CALL CKCOMP (ISPEC(2:2), INUM, 10, I2)
            IF (I1 .GT. 0) THEN
               ITEMP = ' '
               IF (I2 .GT. 0) THEN
                  IVAL = 10*(I1-1) + (I2-1)
                  ITEMP = ISPEC(3:)
               ELSE
                  IVAL = I1-1
                  ITEMP = ISPEC(2:)
               ENDIF
               ISPEC = ' '
               ISPEC = ITEMP
            ENDIF
C
            CALL CKCOMP (ISPEC, KNAME, KK, KNUM)
            IF (KNUM .EQ. 0) THEN
               IF ((N-N1) .GT. 1) GO TO 510
               WRITE (LOUT, 680) ISPEC(:ILASCH(ISPEC))
               KERR = .TRUE.
            ELSE
C
C--------------a species has been found
C
               IVAL = MAX(IVAL,1)
               IF (J .EQ. 1) IVAL = -IVAL
               NNUM = 0
               DO 111 K = 1, NS
                  IF (KNUM.EQ.NUNK(K,II) .AND.
     1                NU(K,II)/IVAL.GT.0) THEN
C
C-----------------increment species coefficient count
C
                     NNUM=K
                     NU(NNUM,II) = NU(NNUM,II) + IVAL
                  ENDIF
  111          CONTINUE
               IF (NNUM .LE. 0) THEN
C
C------------------are there too many species?
C
                  IF (J.EQ.1 .AND. NS.EQ.3) THEN
                     WRITE (LOUT, 690)
                     KERR = .TRUE.
                     RETURN
                  ELSEIF (J.EQ.2 .AND. NS.EQ.MAXSP) THEN
                     WRITE (LOUT, 700)
                     KERR = .TRUE.
                     RETURN
                  ELSE
C
C-------------------increment species count
C
                     NS = NS + 1
                     NSPEC(II) = NSPEC(II)+1
                     IF (J .EQ. 1) NREAC(II) = NS
                     NUNK(NS,II) = KNUM
                     NU(NS,II)   = IVAL
                  ENDIF
               ENDIF
```

```
            ENDIF
            IF (N .EQ. NPLUS) GO TO 600
            NSTART = N
            GO TO 505
C
  510    CONTINUE
  600 CONTINUE
C
      NSPEC(II) = IR*NSPEC(II)
C
  630 FORMAT (6X,'Error...more than one fall-off declaration...')
  640 FORMAT (6X,'Error in fall-off declaration...')
  650 FORMAT (6X,'Error...reaction string not found...')
  660 FORMAT (6X,'Error in reaction...')
  670 FORMAT (6X,'Error in HV declaration...')
  680 FORMAT (6X,'Error...undeclared species...',A)
  690 FORMAT (6X,'Error...more than 3 reactants...')
  700 FORMAT (6X,'Error...more than 3 products...')
  800 FORMAT (6X,'Error in reaction delimiter...')
  900 FORMAT (6X,'Error in third-body declaration...')
C 1900 FORMAT (I4,'. ',A,T51,E10.3,F7.3,F11.3)
 1900 FORMAT (I4,'. ', A, T53, 1PE8.2, 2X, 0PF5.1, 2X, F9.1)
 1920 FORMAT (6X,A)
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKAUXL (SUB, NSUB, II, KK, KNAME, LOUT, MAXSP, NPAR,
     1                   NSPEC, NTHB, ITHB, NTBS, MAXTB, NKTB, AIK,
     2                   NFAL, IFAL, IDUP, NFAR, PFAL, IFOP, KFAL, NLAN,
     3                   ILAN, NLAR, PLAN, NREV, IREV, RPAR,
     4                   NRLT, IRLT, RLAN, NWL, IWL, WL, KERR)
C
C     CKAUXL parses the auxiliary CHAR*(*) lines representing
C     additional options for a gas-phase reaction; data is stored
C     based on finding a 'keyword' followed by its required
C     parameters:
C
C     KNAME(K)/val1/: this is an enhanced third-body;
C
C        if ITHB(NTHB) <> I, this is an error, reaction I is not a
C                            third-body reaction;
C        else NTBS(NTHB) is incremented,
C             AIK(NTBS(NTHB),NTHB) = K,
C             NKTB(NTBS(NTHB)),NTHB) = val1;
C
C     (LOW,TROE, and SRI define fall-off data):
C
C     LOW/val1 val2 val3/: PFAL(N,NFAL) = val(N),N=1,3;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
C                         parameters;
C        if IFOP(NFAL)>0, this is an error, LOW already declared;
C        else
C           IFOP(NFAL) = ABS(IFOP(NFAL))
C
C     TROE/val1 val2 val3 [val4]/: PFAL(N,NFAL) = val(N),N=4,7;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
```

```
C                          parameters;
C        if ABS(IFOP(NFAL)).GT.1, this is an error,
C        else
C        if 3 TROE values, IFOP(NFAL) = 3*IFOP(NFAL);
C        if 4 TROE values, IFOP(NFAL) = 4*IFOP(NFAL);
C
C     SRI/val1 val2 val3/: PFAL(N,NFAL) = val(N),N=4,6;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
C                         parameters;
C        if ABS(IFOP(NFAL))>1, this is an error;
C        else
C        if IFOP(NFAL)= 2*IFOP(NFAL);
C
C     LT/val1 val2/:
C        if IFAL(NFAL)=I, this is an error, cannot have fall-off and
C                         T-L numbers;
C        else increment NLAN, the number of T-L reactions,
C             ILAN(NLAN)=I, PLAN(N,NLAN)=val(N),N=1,2
C        if IREV(NREV)=I, need IRLT(NRLT)=I.
C
C     REV[ERSE]/val1 val2 val3/ :
C        if IFAL(NFAL)=I, this is an error;
C        if IREV(NREV)=I, this is an error, REV already declared;
C        if NSPEC(I)<0, this an error, as I is irreversible;
C        else increment NREV, the number of reactions with reverse
C             parameters given,
C             IREV(NREV)=I, RPAR(N,NREV)=val(N),N=1,3;
C             if ILAN(NLAN)=I, need IRLT(NRLT)=I;
C             if IRLT(NRLT)=I, need ILAN(NRLT)=I.
C
C     RLT/val1 val2/:
C       if IFAL(NFAL)=I, this is an error, cannot have fall-off and
C                        T-L numbers;
C       if IRLT(NRLT)=I, this is an error, RLT already declared;
C       else increment NRLT, the number of reactions with BOTH
C                      reverse parameters given, and T-L numbers;
C            IRLT(NRLT)=I, RLAN(N,NRLT)=val(N),N=1,2;
C            if IREV(NREV)<>I, need IREV(NREV)=I;
C            if ILAN(NREV)<>I, need ILAN(NLAN)=I;
C
C    DUP[LICATE]:
C       This reaction is allowed to be duplicated.
C
C     Input:  LINE - CHAR*(*) auxiliary information string
C             KK   - total number of species declared
C             KNAME- CHAR*(*) species names
C             LOUT - output unit for error messages
C             MAXSP- maximum third bodies allowed in a reaction
C     Output: NTHB - total number of reactions with third bodies
C             ITHB - array of third-body reaction numbers
C             AIK  - non-zero third body enhancement factors
C             NKTB - array of species numbers for the third body
C                          enchancement factors
C             NFAL - total number of fall-off reactions
C             IFAL - array of fall-off reaction numbers
C             IFOP - array of fall-off type
C             PFAL - fall-off parameters
C             NLAN - total number of Landau-Teller reactions
C             ILAN - array of T-L reaction numbers
C             NLAR - number of Landau-Teller numbers allowed
C             PLAN - array of Landau-Teller numbers
```

```
C             NRLT - total number of 'reverse' T-L reactions
C             IRLT - array of 'reverse' T-L reaction numbers
C             RLAN - array of 'reverse' Landau-Teller numbers
C             NWL  - total number of radiation-enhanced reactions
C             IWL  - array of radiation-enhanced reaction numbers
C             WL   - array of wavelengths
C             KERR - logical, = .TRUE. if error found
C                                    F. Rupley, Div. 8245, 5/27/87
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), ITHB(*), NTBS(*), NKTB(MAXTB,*), IDUP(*),
     1          AIK(MAXTB,*), IFAL(*), KFAL(*), IFOP(*), PFAL(NFAR,*),
     2          ILAN(*), PLAN(NLAR,*), IREV(*), RPAR(NPAR,*), IRLT(*),
     3          RLAN(NLAR,*), IWL(*), WL(*), VAL(1)
      CHARACTER SUB(*)*(*), KNAME(*)*(*), KEY*80, RSTR*80, UPCASE*4
      CHARACTER SUB_CH (80)*80,RSTR_CH*80,LINE_CH*80
      LOGICAL KERR, LLAN, LRLT, LTHB, LFAL, LTRO, LSRI, LWL, LREV
C
      LTHB = (NTHB.GT.0 .AND. ITHB(NTHB).EQ.II)
      LFAL = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II)
      LWL  = (NWL .GT.0 .AND. IWL(NWL)  .EQ.II)
      LREV = (NREV.GT.0 .AND. IREV(NREV).EQ.II)
      LLAN = (NLAN.GT.0 .AND. ILAN(NLAN).EQ.II)
      LRLT = (NRLT.GT.0 .AND. IRLT(NRLT).EQ.II)
      LTRO = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).GT.2)
      LSRI = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).EQ.2)
C
      DO 500 N = 1, NSUB
         ILEN = ILASCH(SUB(N))
         KEY = ' '
         IF ( UPCASE(SUB(N), 3) .EQ. 'DUP') THEN
            IDUP(II) = -1
            WRITE (LOUT, 4000)
            GO TO 500
         ELSE
            I1 = INDEX(SUB(N),'/')
            I2 = INDEX(SUB(N)(I1+1:),'/')
            IF (I1.LE.0 .OR. I2.LE.0) THEN
               KERR = .TRUE.
               WRITE (LOUT, 2090) SUB(N)(:ILEN)
               GO TO 500
            ENDIF
            KEY = SUB(N)(:I1-1)
            RSTR = ' '
            RSTR = SUB(N)(I1+1:I1+I2-1)
         ENDIF
C   **************W.ING 6/16/95**********************
         IF (UPCASE(KEY, 3).EQ.'LOW' .OR.
     1       UPCASE(KEY, 4).EQ.'TROE'.OR.
     2       UPCASE(KEY, 4).EQ.'EXTR'.OR.
     3       UPCASE(KEY, 4).EQ.'CHEB'.OR.
     4       UPCASE(KEY, 3).EQ.'SRI') THEN
C
C        FALL-OFF DATA
C
            IF ((.NOT.LFAL) .OR. LLAN .OR. LRLT .OR. LREV) THEN
               KERR = .TRUE.
               IF (.NOT. LFAL) WRITE (LOUT, 1050) SUB(N)(:ILEN)
               IF (LLAN)       WRITE (LOUT, 1060) SUB(N)(:ILEN)
               IF (LRLT)       WRITE (LOUT, 1070) SUB(N)(:ILEN)
```

```
            IF (LREV)       WRITE (LOUT, 1090) SUB(N)(:ILEN)
          ELSE
C
            IF (UPCASE(KEY, 3) .EQ. 'LOW') THEN
               IF (IFOP(NFAL) .GT. 0) THEN
                  WRITE (LOUT, 2000) SUB(N)(:ILEN)
                  KERR = .TRUE.
               ELSE
                  IFOP(NFAL) = ABS(IFOP(NFAL))
                  CALL IPPARR (RSTR,1,3,PFAL(1,NFAL),NVAL,IER,LOUT)
                  IF (IER .NE. 0) KERR = .TRUE.
                  WRITE (LOUT, 3050) (PFAL(L,NFAL),L=1,3)
               ENDIF
C
            ELSEIF (UPCASE(KEY, 4) .EQ. 'TROE') THEN
               IF (LTRO .OR. LSRI) THEN
                  KERR = .TRUE.
                  IF (LTRO) WRITE (LOUT, 2010) SUB(N)(:ILEN)
                  IF (LSRI) WRITE (LOUT, 2030) SUB(N)(:ILEN)
               ELSE
                  LTRO = .TRUE.
                  CALL IPPARR (RSTR,1,-4,PFAL(4,NFAL),NVAL,IER,LOUT)
                  IF (NVAL .EQ. 3) THEN
                     IFOP(NFAL) = 3*IFOP(NFAL)
                     WRITE (LOUT, 3080) (PFAL(L,NFAL),L=4,6)
                  ELSEIF (NVAL .EQ. 4) THEN
                     IFOP(NFAL) = 4*IFOP(NFAL)
                     WRITE (LOUT, 3090) (PFAL(L,NFAL),L=4,7)
                  ELSE
                     WRITE (LOUT, 2020) SUB(N)(:ILEN)
                     KERR = .TRUE.
                  ENDIF
               ENDIF
C
C   **************W.ING 8/10/95************************
            ELSEIF (UPCASE(KEY, 4) .EQ. 'CHEB') THEN
C                LTRO = .TRUE.
                  CALL IPPARR (RSTR,1,-8,PFAL(4,NFAL),NVAL,IER,LOUT)
                  N_CHEB = PFAL(4,NFAL)
                  M_CHEB = PFAL(5,NFAL)
                  NPARA_CHEB = N_CHEB*M_CHEB + 2
                  INDEX_CHEB = 0
                  IFOP(NFAL) = 6*IFOP(NFAL)
c                  write(*,*)'..NFAL..IFOP...',NFAL,IFOP(NFAL)
C                  WRITE(*,' ..NPARA_CHEX..')
C                  CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                  INDEX_CHEB = INDEX_CHEB + NVAL
C                  WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
c                  WRITE(*,*)' ..NPARA..INDEX..NVAL.',NPARA_CHEB,NVAL
c                  WRITE(*,*)'...N...M...',N_CHEB,M_CHEB
                  WRITE (LOUT, 3098) (PFAL(L,NFAL),L=4,NVAL+3)
101               CONTINUE
                  LINE_CH = ' '
                  READ (5,'(A)') LINE_CH
c                  WRITE(*,*)' ..LINE_CH..'
c                  WRITE(*,*)LINE_CH
C
                  ILEN = IPPLEN(LINE_CH)
                  CALL CKISUB (LINE_CH(:ILEN), SUB_CH, NSUB_CH)
C                      KEY = ' '
                  DO 151 N_CH = 1, NSUB_CH
                     IILEN = ILASCH(SUB_CH(NSUB_CH))
c                     WRITE(*,*)'   NSUB_CH = ',NSUB_CH
c                     WRITE(*,*)SUB_CH(NSUB_CH)
                     II1 = INDEX(SUB_CH(NSUB_CH),'/')
                     II2 = INDEX(SUB_CH(NSUB_CH)(II1+1:),'/')
```

```
c                        WRITE(*,*)'....II1..II2...',II1,II2
                        IF (II1.LE.0 .OR. II2.LE.0) THEN
                          KERR = .TRUE.
                          WRITE (LOUT, 2090) SUB_CH(N_CH)(:IILEN)
C                          GO TO 199
                        ENDIF
C                        KEY = SUB(N)(:I1-1)
                        RSTR_CH = ' '
                        RSTR_CH = SUB_CH(NSUB_CH)(II1+1:II1+II2-1)
c                        WRITE(*,*)RSTR_CH
                      CALL IPPARR (RSTR_CH,1,-8,PFAL(INDEX_CHEB+4,NFAL)
     2                                ,NVAL,IER,LOUT)
c                        WRITE(*,*)'...INDEX..NVAL..',INDEX_CHEB,NVAL
c                        WRITE(*,*)'...IER..= ',IER
C                        INDEX_CHEB = INDEX_CHEB + NVAL
c                        WRITE (*, 3098) (PFAL(L,NFAL),L=INDEX_CHEB+4
c     2                                  ,NVAL+INDEX_CHEB+3)
                        WRITE (LOUT, 3098) (PFAL(L,NFAL),L=INDEX_CHEB+4
     2                                  ,NVAL+INDEX_CHEB+3)
                        INDEX_CHEB = INDEX_CHEB + NVAL
c                       WRITE(*,*)'..INDEX...',INDEX_CHEB
151                   CONTINUE
                      IF (INDEX_CHEB.LT.NPARA_CHEB) GOTO 101
199                   CONTINUE
                      IF (INDEX_CHEB .EQ. NPARA_CHEB) THEN
C
                      ELSE
                         WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
                         KERR = .TRUE.
                      ENDIF
C   ************************************************
C   **************W.ING 6/16/95***********************
                ELSEIF (UPCASE(KEY, 4) .EQ. 'EXTR') THEN
                   IF (LTRO .OR. LSRI) THEN
                      KERR = .TRUE.
                      IF (LTRO) WRITE (LOUT, 2015) SUB(N)(:ILEN)
                      IF (LSRI) WRITE (LOUT, 2035) SUB(N)(:ILEN)
                   ELSE
                      LTRO = .TRUE.
                      CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                      IF (NVAL .EQ. 5) THEN
                         IFOP(NFAL) = 5*IFOP(NFAL)
C                         WRITE(*,*)IFOP(NFAL)
                         WRITE (LOUT, 3095) (PFAL(L,NFAL),L=4,8)
                      ELSE
                         WRITE (LOUT, 2025) SUB(N)(:ILEN)
                         KERR = .TRUE.
                      ENDIF
                   ENDIF
C   ************************************************
                ELSEIF (UPCASE(KEY, 3) .EQ. 'SRI') THEN
                   IF (LTRO .OR. LSRI) THEN
                      KERR = .TRUE.
                      IF (LTRO) WRITE (LOUT, 2030) SUB(N)(:ILEN)
                      IF (LSRI) WRITE (LOUT, 2040) SUB(N)(:ILEN)
                   ELSE
                      LSRI = .TRUE.
                      IFOP(NFAL) = 2*IFOP(NFAL)
                      CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                      IF (NVAL .EQ. 3) THEN
                         PFAL(7,NFAL) = 1.0
                         PFAL(8,NFAL) = 0.0
                         WRITE (LOUT, 3060) (PFAL(L,NFAL),L=4,6)
                      ELSEIF (NVAL .EQ. 5) THEN
                         WRITE (LOUT, 3070) (PFAL(L,NFAL),L=4,8)
                      ELSE
```

```
                  WRITE (LOUT, 2020) SUB(N)(:ILEN)
                  KERR = .TRUE.
               ENDIF
            ENDIF
         ENDIF
      ENDIF
C
      ELSEIF (UPCASE(KEY, 3) .EQ. 'REV') THEN
C
C     REVERSE ARRHENIUS PARAMETERS
C
         IF (LFAL .OR. LREV .OR. NSPEC(II).LT.0) THEN
            KERR = .TRUE.
            IF (LFAL) WRITE (LOUT, 1090) SUB(N)(:ILEN)
            IF (LREV) WRITE (LOUT, 2050) SUB(N)(:ILEN)
            IF (NSPEC(II) .LT. 0) WRITE (LOUT, 2060) SUB(N)(:ILEN)
         ELSE
            LREV = .TRUE.
            NREV = NREV+1
            IREV(NREV) = II
            CALL IPPARR (RSTR,1,NPAR,RPAR(1,NREV),NVAL,IER,LOUT)
            IF (IER .NE. 0) KERR = .TRUE.
            WRITE (LOUT, 1900) '   Reverse Arrhenius coefficients:',
     1                         (RPAR(L,NREV),L=1,3)
         ENDIF
C
      ELSEIF (UPCASE(KEY, 3) .EQ. 'RLT') THEN
C
C     REVERSE LANDAU-TELLER PARAMETERS
C
         IF (LFAL .OR. LRLT .OR. NSPEC(II).LT.0) THEN
            KERR = .TRUE.
            IF (LFAL) WRITE (LOUT, 1070) SUB(N)(:ILEN)
            IF (LRLT) WRITE (LOUT, 2080) SUB(N)(:ILEN)
            IF (NSPEC(II) .LT. 0) WRITE (LOUT, 1080) SUB(N)(:ILEN)
         ELSE
            LRLT = .TRUE.
            NRLT = NRLT + 1
            IRLT(NRLT) = II
            CALL IPPARR (RSTR,1,NLAR,RLAN(1,NRLT),NVAL,IER,LOUT)
            IF (IER .NE. 0) KERR = .TRUE.
            WRITE (LOUT, 3040) (RLAN(L,NRLT),L=1,2)
         ENDIF
C
      ELSEIF (UPCASE(KEY, 2) .EQ. 'HV') THEN
C
C     RADIATION WAVELENGTH ENHANCEMENT FACTOR
C
         IF (.NOT.LWL) THEN
            WRITE (LOUT, 1000) SUB(N)(:ILEN)
            KERR = .TRUE.
         ELSE
            CALL IPPARR (RSTR,1,1,VAL,NVAL,IER,LOUT)
            IF (IER .EQ. 0) THEN
               WL(NWL) = WL(NWL)*VAL(1)
               WRITE (LOUT, 3020) ABS(WL(NWL))
            ELSE
               WRITE (LOUT, 1000) SUB(N)(:ILEN)
               KERR = .TRUE.
            ENDIF
         ENDIF
C
      ELSEIF (UPCASE(KEY, 2) .EQ. 'LT') THEN
C
C     LANDAU-TELLER PARAMETERS
C
```

```
            IF (LFAL .OR. LLAN) THEN
               KERR = .TRUE.
               IF (LFAL) WRITE (LOUT, 1060) SUB(N)(:ILEN)
               IF (LLAN) WRITE (LOUT, 2070) SUB(N)(:ILEN)
            ELSE
               LLAN = .TRUE.
               NLAN = NLAN + 1
               ILAN(NLAN) = II
               CALL IPPARR (RSTR,1,NLAR,PLAN(1,NLAN),NVAL,IER,LOUT)
               IF (IER .NE. 0) THEN
                  WRITE (LOUT, 1010) SUB(N)(:ILEN)
                  KERR = .TRUE.
               ENDIF
               WRITE (LOUT, 3000) (PLAN(L,NLAN),L=1,2)
            ENDIF
C
         ELSE
C
C        ENHANCED THIRD BODIES
C
            CALL CKCOMP (KEY, KNAME, KK, K)
            IF (K .EQ. 0) THEN
               WRITE (LOUT, 1040) KEY(:ILASCH(KEY))
               KERR = .TRUE.
            ELSE
               IF (.NOT.LTHB) THEN
                  KERR = .TRUE.
                  WRITE (LOUT, 1020) SUB(N)(:ILEN)
               ELSE
                  IF (NTBS(NTHB) .EQ. MAXTB) THEN
                     KERR = .TRUE.
                     WRITE (LOUT, 1030) SUB(N)(:ILEN)
                  ELSE
                     CALL IPPARR (RSTR, 1, 1, VAL, NVAL, IER, LOUT)
                     IF (IER .EQ. 0) THEN
                        WRITE (LOUT, 3010) KNAME(K),VAL(1)
                        NTBS(NTHB) = NTBS(NTHB) + 1
                        NKTB(NTBS(NTHB),NTHB) = K
                        AIK(NTBS(NTHB),NTHB) = VAL(1)
                     ELSE
                        WRITE (LOUT, 1020) SUB(N)(:ILEN)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
  500 CONTINUE
C
C     FORMATS
C
 1000 FORMAT (6X,'Error in HV declaration...',A)
 1010 FORMAT (6X,'Error in LT declaration..',A)
 1020 FORMAT (6X,'Error in third body declaration...',A)
 1030 FORMAT (6X,'Error...more than MAXTB third bodies...',A)
 1040 FORMAT (6X,'Error...undeclared species...',A)
 1050 FORMAT (6X,'Error...this is not a fall-off reaction...',A)
 1060 FORMAT (6X,'Error...LT declared in fall-off reaction...',A)
 1070 FORMAT (6X,'Error...RLT declared in fall-off reaction...',A)
 1080 FORMAT (6X,'Error...RLT declared in irreversible reaction...',A)
 1090 FORMAT (6X,'Error...REV declared in fall-off reaction...',A)
 2000 FORMAT (6X,'Error...LOW declared more than once...',A)
 2010 FORMAT (6X,'Error...TROE declared more than once...',A)
 2015 FORMAT (6X,'Error...EXTROE declared more than once...',A)
 2020 FORMAT (6X,'Error in fall-off parameters...',A)
 2025 FORMAT (6X,'Error in EXTROE fall-off parameters...',A)
```

```
 2028 FORMAT (6X,'Error in CHEB fall-off parameters...',A)
 2030 FORMAT (6X,'Error...cannot use both TROE and SRI...',A)
 2035 FORMAT (6X,'Error...cannot use TROE, EXTROE or SRI...',A)
 2040 FORMAT (6X,'Error...SRI declared more than once...',A)
 2050 FORMAT (6X,'Error...REV declared more than once...',A)
 2060 FORMAT (6X,'Error...REV declared for irreversible reaction...',A)
 2070 FORMAT (6X,'Error...LT declared more than once...',A)
 2080 FORMAT (6X,'Error...RLT declared more than once...',A)
 2090 FORMAT (6X,'Error in auxiliary data...',A)
 3000 FORMAT (9X,'Landau-Teller parameters: B=',E12.5,', C=',E12.5)
 3010 FORMAT (9X,A16,' Enhanced by ',1PE12.3)
 3020 FORMAT (9X,'Radiation wavelength (A): ',F10.2)
C 1900 FORMAT (6X,A,T51,E10.3,F7.3,F11.3)
 1900 FORMAT (6X, A, T53, 1PE8.2, 2X, 0PF5.1, 2X, F9.1)
 3040 FORMAT (9X,'Reverse Landau-Teller parameters: B=',E12.5,
     1           ', C=',E12.5)
 3050 FORMAT (6X,'Low pressure limit:',3E13.5)
 3060 FORMAT (6X,'SRI centering:     ',3E13.5)
 3070 FORMAT (6X,'SRI centering:     ',5E13.5)
 3080 FORMAT (6X,'TROE centering:    ',3E13.5)
 3090 FORMAT (6X,'TROE centering:    ',4E13.5)
 3095 FORMAT (6X,'EXTROE centering:  ',5E13.5)
 3098 FORMAT (6X,'CHEB Polynomials:  ',8E13.5)
 4000 FORMAT (6X,'Declared duplicate reaction...')
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKPRNT (MDIM, MAXTP, MM, ENAME, KK, KNAME, WTM,
     1                   KPHSE, KCHRG, NT, T, TLO, TMID, THI, KNCF,
     2                   ITHRM, LOUT, KERR)
C
C     Prints species interpreter output and checks for completeness.
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION WTM(*), KPHSE(*), KCHRG(*), T(MAXTP,*),
     1          NT(*), KNCF(MDIM,*), IPLUS(10)
      LOGICAL KERR, ITHRM(*)
      CHARACTER ENAME(*)*(*), KNAME(*)*(*), IPHSE(3)*1, INUM(10)*1
      DATA IPHSE/'S','G','L'/
      DATA INUM/'0','1','2','3','4','5','6','7','8','9'/
C
      WRITE (LOUT, 400) (ENAME(M), M = 1, MM)
      WRITE (LOUT, 300)
C
      DO 100 K = 1, KK
C
         IF (T(1,K) .LT. 0.0) T(1,K) = TLO
         IF (T(2,K) .LT. 0.0) T(2,K) = TMID
         IF (T(3,K) .LT. 0.0) T(NT(K),K) = THI
         WRITE (LOUT, 500) K, KNAME(K), IPHSE(KPHSE(K)+2), KCHRG(K),
     1                     WTM(K), T(1,K), T(NT(K),K), (KNCF(M,K),M=1,MM)
         IF (T(1,K) .GE. T(NT(K),K)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 240)
         ENDIF
         IF (T(1,K) .GT. T(2,K)) THEN
            WRITE (LOUT, 250)
            KERR = .TRUE.
         ENDIF
         IF (T(NT(K),K) .LT. T(2,K)) THEN
            WRITE (LOUT, 260)
```

```
            KERR = .TRUE.
         ENDIF
C
C        each species must have thermodynamic data
C
         IF (.NOT. ITHRM(K)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 200)
         ENDIF
C
C        a species cannot start with a number
C
         CALL CKCOMP (KNAME(K)(:1), INUM, 10, I)
         IF (I .GT. 0) THEN
            KERR = .TRUE.
            WRITE (LOUT, 210)
         ENDIF
C
C        if '+' sign is used in a species name,
C           examples of legal species symbols with + are:
C           OH(+)2, OH(+2), OH+, OH++, OH+++, OH(+), OH(++),
C           OH[+OH], OH2+, OH+2
C
C           examples of illegal species symbols with + are:
C           +OH         (symbol starts with a +, this will cause
C                        confusion in a reaction)
C           OH(+OH)     (symbol in parentheses is another species-
C                        this arrangement is reserved for a fall-off
C                        reaction)
C           OH+OH       (plus delimits other species names, this
C                        will cause confusion in a reaction)
C
         NPLUS = 0
         DO 50 N = 1, ILASCH(KNAME(K))
            IF (KNAME(K)(N:N) .EQ. '+') THEN
               NPLUS = NPLUS + 1
               IPLUS(NPLUS) = N
            ENDIF
   50    CONTINUE
         DO 60 N = 1, NPLUS
            I1 = IPLUS(N)
            IF (I1 .EQ. 1) THEN
               WRITE (LOUT, 220)
               KERR = .TRUE.
            ELSE
C
C              is there another species name in parentheses
C
               IF (KNAME(K)(I1-1:I1-1) .EQ. '(') THEN
                  I1 = I1 + 1
                  I2 = I1 + INDEX(KNAME(K)(I1:),')')-1
                  IF (I2 .GT. I1) THEN
                     CALL CKCOMP (KNAME(K)(I1:I2-1), KNAME, KK, KNUM)
                     IF (KNUM .GT. 0) THEN
                        WRITE (LOUT, 230)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
               ENDIF
C
C              is there another species name after a +
C
               I1 = I1 + 1
               IF (N .LT. NPLUS) THEN
                  DO 55 L = N+1, NPLUS
                     I2 = IPLUS(L)
```

```
                  IF (I2 .GT. I1) THEN
                     CALL CKCOMP (KNAME(K)(I1:I2-1),KNAME,KK,KNUM)
                     IF (KNUM .GT. 0) THEN
                        WRITE (LOUT, 230)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
   55          CONTINUE
            ENDIF
C
            I2 = ILASCH(KNAME(K))
            IF (I2 .GE. I1) THEN
               CALL CKCOMP (KNAME(K)(I1:I2), KNAME, KK, KNUM)
               IF (KNUM .GT. 0) THEN
                  WRITE (LOUT, 230)
                  KERR = .TRUE.
               ENDIF
            ENDIF
         ENDIF
   60    CONTINUE
C
  100 CONTINUE
      WRITE (LOUT, 300)
      RETURN
C
  200 FORMAT (6X,'Error...no thermodynamic properties for species')
  210 FORMAT (6X,'Error...species starts with a number')
  220 FORMAT (6X,'Error...species starts with a plus')
  230 FORMAT (6X,'Error...illegal + in species name')
  240 FORMAT (6X,'Error...High temperature must be < Low temperature')
  250 FORMAT (6X,'Error...Low temperature must be <= Mid temperature')
  260 FORMAT (6X,'Error...High temperature must be => Mid temperature')
  300 FORMAT (1X,79('-'))
C  400 FORMAT (1X,79('-'),/21X,'C',/18X,'P',2X,'H',/18X,'H',2X,'A',
C     1        /18X,'A',2X,'R',/1X,'SPECIES',10X,'S',2X,'G',2X,
C     2        'MOLECULAR',3X,'TEMPERATURE',4X,'ELEMENT COUNT',/1X,
C     3        'CONSIDERED',7X,'E',2X,'E',2X,'WEIGHT',6X,'LOW',5X,
C     4        'HIGH',3X,15(A3),/1X,79('-'))
C  500 FORMAT (I4,'. ',A10,2X,A1,I3,F11.5,2(F8.1),15(I3))
C
  400 FORMAT (1X,79('-'),/T26,'C',/T24,'P H',/T24,'H A',/T24,'A R',
     1       /1X,'SPECIES',T24,'S G',T28,'MOLECULAR',T38,'TEMPERATURE',
     2       T52,'ELEMENT COUNT',
     3       /1X,'CONSIDERED',T24,'E E',T28,'WEIGHT',T38,'LOW',
     4       T45,'HIGH',T52,15(A3))
  500 FORMAT (1X,I3,'. ',A16,T24,A1,T26,I1,T28,F9.5,T38,F6.1,T45,F6.1,
     1       T51,15(I3))
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CPREAC (II, MAXSP, NSPEC, NPAR, PAR, RPAR, AUNITS,
     1                   EUNITS, NREAC, NUNK, NU, KCHRG, MDIM, MM, KNCF,
     2                   IDUP, NFAL, IFAL, KFAL, NFAR, PFAL, IFOP, NREV,
     3                   IREV, NTHB, ITHB, NLAN, ILAN, NRLT, IRLT, KERR,
     4                   LOUT)
C
C     Prints reaction interpreter output and checks for reaction
C     balance, duplication, and missing data in 'REV' reactions;
C     correct units of Arrhenius parameters
C
C     Input: II     - the index number of the reaction
C            MAXSP  - maximum number of species allowed in a reaction
C            NSPEC  - array of the number of species in the reactions
C            NPAR   - the number of Arrhenius parameters required
C            PAR    - matrix of Arrhenius parameters for the reactions
C            RPAR   - matrix of reverse Arrhenius parameters for the
C                     reactions which declared them
```

```
C            AUNITS - character string which describes the input units
C                     of A, the pre-exponential factor PAR(1,I)
C            EUNITS - character string which describes the input units
C                     of E, the activation energy PAR(3,I)
C            NREAC  - array of the number of reactants in the reactions
C            NUNK   - matrix of the species numbers of the reactants
C                     and products in the reactions
C            NU     - matrix of the stoichiometric coefficients of the
C                     reactants and products in the reactions
C            KCHRG  - array of the electronic charges of the species
C            MDIM   - the maximum number of elements allowed
C            MM     - the actual number of elements declared
C            KNCF   - matrix of elemental composition of the species
C            IDUP   - array of integer flags to indicate duplicate
C                     reactions
C            NFAL   - total number of reactions with fall-off
C            IFAL   - array of the NFAL reaction numbers
C            NFAR   - maximum number of fall-off parameters allowed
C            PFAL   - matrix of fall-off parameters for the NFAL
C                     reactions
C            IFOP   - array of integer fall-off types for the NFAL
C                     reactions
C            NREV   - total number of reactions with reverse parameters
C            IREV   - array of the NREV reaction numbers
C            NTHB   - total number of reactions with third-bodies
C            ITHB   - array of the NTHB reaction numbers
C            NLAN   - total number of reactions with Landauer-Teller
C                     parameters
C            ILAN   - array of the NLAN reaction numbers
C            NRLT   - total number of reactions with reverse
C                     Landauer-Teller parameters
C            IRLT   - array of the NRLT reaction numbers
C            KERR   - logical error flag
C            LOUT   - unit number for output messages
C
C-----------------------------------------------------------------------C
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), PAR(NPAR,*), RPAR(NPAR,*), NREAC(*),
     1          NUNK(MAXSP,*), NU(MAXSP,*), KCHRG(*), KNCF(MDIM,*),
     2          IDUP(*), IFAL(*), KFAL(*), PFAL(NFAR,*), IFOP(*),
     3          IREV(*), ITHB(*), ILAN(*), IRLT(*)
      CHARACTER*(*) AUNITS, EUNITS
      LOGICAL IERR,KERR,LREV,LLAN,LRLT
C
      CALL CKBAL (MAXSP, NUNK(1,II), NU(1,II), MDIM, MM, KCHRG, KNCF,
     1            IERR)
C
      IF (IERR) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1060)
      ENDIF
C
      CALL CKDUP (II, MAXSP, NSPEC, NREAC, NU, NUNK, NFAL, IFAL, KFAL,
     1            ISAME)
C
      IF (ISAME .GT. 0) THEN
         IF (IDUP(ISAME).NE.0 .AND. IDUP(II).NE.0) THEN
            IDUP(ISAME) = ABS(IDUP(ISAME))
            IDUP(II)    = ABS(IDUP(II))
```

```
            ELSE
               N1 = 0
               N2 = 0
               IF (NTHB .GT. 1) THEN
                  DO 150 N = 1, NTHB
                     IF (ITHB(N) .EQ. ISAME) N1 = 1
                     IF (ITHB(N) .EQ. II)    N2 = 1
  150             CONTINUE
               ENDIF
               IF (N1 .EQ. N2) THEN
                  KERR = .TRUE.
                  WRITE (LOUT, 1050) ISAME
               ENDIF
            ENDIF
         ENDIF
C
         IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).LT.0) THEN
            KERR = .TRUE.
            WRITE (LOUT, 1020)
         ENDIF
C
         LREV = (NREV.GT.0 .AND. IREV(NREV).EQ.II)
         LLAN = (NLAN.GT.0 .AND. ILAN(NLAN).EQ.II)
         LRLT = (NRLT.GT.0 .AND. IRLT(NRLT).EQ.II)
         IF (LREV .AND. LLAN .AND. (.NOT.LRLT)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 1030)
         ENDIF
         IF (LRLT .AND. (.NOT.LLAN)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 1040)
         ENDIF
         IF (LRLT .AND. (.NOT.LREV)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 1045)
         ENDIF
C
         IF (EUNITS .EQ. 'KELV') THEN
            EFAC = 1.0
         ELSEIF (EUNITS .EQ. 'CAL/') THEN
C           convert E from cal/mole to Kelvin
            EFAC = 1.0 / 1.987
         ELSEIF (EUNITS .EQ. 'KCAL') THEN
C           convert E from kcal/mole to Kelvin
            EFAC = 1000.0 / 1.987
         ELSEIF (EUNITS .EQ. 'JOUL') THEN
C           convert E from Joules/mole to Kelvin
            EFAC = 1.0 / 8.314
         ELSEIF (EUNITS .EQ. 'KJOU') THEN
C           convert E from Kjoules/mole to Kelvin
            EFAC = 1000.0 / 8.314
         ENDIF
         PAR(3,II) = PAR(3,II) * EFAC
C
C         IF (NREV.GT.0 .AND. IREV(NREV).EQ.II) RPAR(3,II)=RPAR(3,II)*EFAC
C         IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) PFAL(3,II)=PFAL(3,II)*EFAC
C
         IF (NREV.GT.0 .AND. IREV(NREV).EQ.II)
        1    RPAR(3,NREV) = RPAR(3,NREV) * EFAC
         IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II)
        1    PFAL(3,NFAL) = PFAL(3,NFAL) * EFAC
C
         IF (AUNITS .EQ. 'MOLC') THEN
            NSTOR = 0
            NSTOP = 0
            DO 50 N = 1, MAXSP
```

```
            IF (NU(N,II) .LT. 0) THEN
C              sum of stoichiometric coefficients of reactants
               NSTOR = NSTOR + ABS(NU(N,II))
            ELSEIF (NU(N,II) .GT. 0) THEN
C              sum of stoichiometric coefficients of products
               NSTOP = NSTOP + NU(N,II)
            ENDIF
   50     CONTINUE
C
          AVAG = 6.023E23
C
          IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) THEN
C
C            fall-off reaction, "(+M)" or "(+species name)" does not
C            count except in "LOW" A-factor;
C            reverse-rate declarations are not allowed
C
             IF (NSTOR.GT.0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
             NSTOR = NSTOR + 1
             IF (NSTOR.GT.0) PFAL(1,NFAL) = PFAL(1,NFAL)*AVAG**(NSTOR-1)
C
          ELSEIF (NTHB.GT.0 .AND. ITHB(NTHB).EQ.II) THEN
C
C            third body reaction, "+M" counts as species in
C            forward and reverse A-factor conversion
C
             NSTOR = NSTOR + 1
             NSTOP = NSTOP + 1
             IF (NSTOR.GT.0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
             IF (NREV.GT.0 .AND. IREV(NREV).EQ.II .AND. NSTOP.GT.0)
     1           RPAR(1,NREV) = RPAR(1,NREV) * AVAG**(NSTOP-1)
C
          ELSE
C
C            not third-body or fall-off reaction, but may have
C            reverse rates.
C
             IF (NSTOR .GT. 0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
             IF (NREV.GT.0 .AND. IREV(NREV).EQ.II .AND. NSTOP.GT.0)
     1           RPAR(1,NREV) = RPAR(1,NREV) * AVAG**(NSTOP-1)
          ENDIF
       ENDIF
C
 1020 FORMAT (6X,'Error...no LOW parameters given for fall-off...')
 1030 FORMAT (6X,'Error...reverse T-L required...')
 1040 FORMAT (6X,'Error...forward T-L required...')
 1045 FORMAT (6X,'Error...REV parameters must be given with RTL...')
 1050 FORMAT (6X,'Error...undeclared duplicate to reaction number ',I3)
 1060 FORMAT (6X,'Error...reaction does not balance...')
       RETURN
       END
C-----------------------------------------------------------------------C
       SUBROUTINE CKBAL (NSPEC, KSPEC, KCOEF, MDIM, MM, KCHRG, KNCF,
      1                  IERR)
C
C      Checks elemental balance of reactants vs. products.
C      Checks charge balance of reaction.
C
C      Input:  NSPEC - total number of species in this reaction
C              KSPEC(N),N=1,NSPEC- array of species numbers in reaction
C              KCOEF(N) - stoichiometric coefficients of the species
C              MDIM  - maximum number of elements allowed
C              MM    - actual integer number of elements
C              KCHRG(K) - ionic charge Kth species
C              KNCF(M,K)- integer elemental composition of Kth species
C      Output: KERR  - logical, =.TRUE. if reaction does not balance
```

```
C                                        F. Rupley, Div. 8245, 5/13/86
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION KSPEC(*), KCOEF(*), KNCF(MDIM,*), KCHRG(*)
      LOGICAL IERR
C
      IERR = .FALSE.
C
C     charge balance
C
      KBAL = 0
      DO 50 N = 1, ABS(NSPEC)
         KBAL = KBAL + KCOEF(N)*KCHRG(KSPEC(N))
   50 CONTINUE
      IF (KBAL .NE. 0) IERR = .TRUE.
C
C     element balance
C
      DO 100 M = 1, MM
         MBAL = 0
         DO 80 N = 1, ABS(NSPEC)
            MBAL = MBAL + KCOEF(N)*KNCF(M,KSPEC(N))
   80    CONTINUE
         IF (MBAL .NE. 0) IERR = .TRUE.
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKDUP (I, MAXSP, NS, NR, NU, NUNK, NFAL, IFAL, KFAL,
     1                  ISAME)
C
C     Checks reaction I against the (I-1) reactions for duplication
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NS(*), NR(*), NU(MAXSP,*), NUNK(MAXSP,*), IFAL(*),
     1          KFAL(*)
C
      ISAME = 0
      NRI = NR(I)
      NPI = ABS(NS(I)) - NR(I)
C
      DO 500 J = 1, I-1
C
         NRJ = NR(J)
         NPJ = ABS(NS(J)) - NR(J)
C
         IF (NRJ.EQ.NRI .AND. NPJ.EQ.NPI) THEN
C
            NSAME = 0
            DO 20 N = 1, MAXSP
               KI = NUNK(N,I)
               NI = NU(N,I)
C
               DO 15 L = 1, MAXSP
```

```
                  KJ = NUNK(L,J)
                  NJ = NU(L,J)
                  IF (NJ.NE.0 .AND. KJ.EQ.KI .AND. NJ.EQ.NI)
     1            NSAME = NSAME + 1
    15         CONTINUE
    20       CONTINUE
C
           IF (NSAME .EQ. ABS(NS(J))) THEN
C
C          same products, reactants, coefficients, check fall-off
C          third body
C
              IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.I) THEN
                 DO 22 N = 1, NFAL-1
                    IF (J.EQ.IFAL(N) .AND. KFAL(N).EQ.KFAL(NFAL)) THEN
                       ISAME = J
                       RETURN
                    ENDIF
    22             CONTINUE
                   RETURN
                ENDIF
C
                ISAME = J
                RETURN
             ENDIF
          ENDIF
C
          IF (NPI.EQ.NRJ .AND. NPJ.EQ.NRI) THEN
C
             NSAME = 0
             DO 30 N = 1, MAXSP
                KI = NUNK(N,I)
                NI = NU(N,I)
C
                DO 25 L = 1, MAXSP
                   KJ = NUNK(L,J)
                   NJ = NU(L,J)
                   IF (NJ.NE.0 .AND. KJ.EQ.KI .AND. -NJ.EQ.NI)
     1             NSAME = NSAME + 1
    25          CONTINUE
    30       CONTINUE
C
             IF (NSAME.EQ.ABS(NS(J)) .AND.
     1           (NS(J).GT.0 .OR. NS(I).GT.0)) THEN
C
C            same products as J reactants, and vice-versa
C
                IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.I) THEN
                   DO 32 N = 1, NFAL-1
                      IF (J.EQ.IFAL(N) .AND. KFAL(N).EQ.KFAL(NFAL)) THEN
                         ISAME = J
                         RETURN
                      ENDIF
    32             CONTINUE
                   RETURN
                ENDIF
C
                ISAME = J
                RETURN
             ENDIF
          ENDIF
C
  500 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
```

```
      SUBROUTINE CKISUB (LINE, SUB, NSUB)
C
C     Generates an array of CHAR*(*) substrings from a CHAR*(*) string,
C     using blanks or tabs as delimiters
C
C     Input:  LINE  - a CHAR*(*) line
C     Output: SUB   - a CHAR*(*) array of substrings
C             NSUB  - number of substrings found
C     A '!' will comment out a line, or remainder of the line.
C                                        F. Rupley, Div. 8245, 5/15/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) SUB(*), LINE
      NSUB = 0
C
      DO 5 N = 1, LEN(LINE)
        IF (ICHAR(LINE(N:N)) .EQ. 9) LINE(N:N) = ' '
    5 CONTINUE
C
      IF (IPPLEN(LINE) .LE. 0) RETURN
C
      ILEN = ILASCH(LINE)
C
      NSTART = IFIRCH(LINE)
   10 CONTINUE
      ISTART = NSTART
      NSUB = NSUB + 1
      SUB(NSUB) = ' '
C
      DO 100 I = ISTART, ILEN
         ILAST = INDEX(LINE(ISTART:),' ') - 1
         IF (ILAST .GT. 0) THEN
            ILAST = ISTART + ILAST - 1
         ELSE
            ILAST = ILEN
         ENDIF
         SUB(NSUB) = LINE(ISTART:ILAST)
         IF (ILAST .EQ. ILEN) RETURN
C
         NSTART = ILAST + IFIRCH(LINE(ILAST+1:))
C
C        Does SUB have any slashes?
C
         I1 = INDEX(SUB(NSUB),'/')
         IF (I1 .LE. 0) THEN
            IF (LINE(NSTART:NSTART) .NE. '/') GO TO 10
            NEND = NSTART + INDEX(LINE(NSTART+1:),'/')
            IND = INDEX(SUB(NSUB),' ')
            SUB(NSUB)(IND:) = LINE(NSTART:NEND)
            IF (NEND .EQ. ILEN) RETURN
            NSTART = NEND + IFIRCH(LINE(NEND+1:))
            GO TO 10
         ENDIF
C
C        Does SUB have 2 slashes?
C
         I2 = INDEX(SUB(NSUB)(I1+1:),'/')
         IF (I2 .GT. 0) GO TO 10
C
         NEND = NSTART + INDEX(LINE(NSTART+1:),'/')
```

```
         IND = INDEX(SUB(NSUB),' ') + 1
         SUB(NSUB)(IND:) = LINE(NSTART:NEND)
         IF (NEND .EQ. ILEN) RETURN
         NSTART = NEND + IFIRCH(LINE(NEND+1:))
         GO TO 10
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPNPAR (LINE, NPAR, IPAR, ISTART)
C
C     Returns CHAR*(*) IPAR substring of CHAR*(*) string LINE which
C     contains NPAR real parameters
C
C     Input:     LINE - a CHAR*(*) line
C                NPAR - number of parameters expected
C     Output:    IPAR - the substring of parameters only
C                ISTART - the starting location of IPAR substring
C     A '!' will comment out a line, or remainder of the line.
C                                    F. Rupley, Div. 8245, 5/14/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) LINE,IPAR
C
C----------Find Comment String (! signifies comment)
C
      ILEN = IPPLEN(LINE)
      ISTART = 0
      N = 0
      IF (ILEN.GT.0) THEN
         DO 40 I = ILEN, 1, -1
            ISTART = I
            IPAR = ' '
            IPAR = LINE(ISTART:ILEN)
            IF (LINE(I:I).NE.' ') THEN
               IF (I .EQ. 1) RETURN
               IF (LINE(I-1:I-1) .EQ. ' ') THEN
                  N = N + 1
                  IF (N .EQ. NPAR) RETURN
               ENDIF
            ENDIF
   40    CONTINUE
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPPARI(STRING, ICARD, NEXPEC, IVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARI
C   REFER TO  IPGETI
C   DATE WRITTEN  850625    (YYMMDD)
C   REVISION DATE 851725    (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses integer variables from a character variable.  Called
C            by IPGETI, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARI may be used for parsing an input record that contains integer
```

```
C  values, but was read into a character variable instead of directly
C  into integer variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARI to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of IVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1,   2,,40000   , ,60'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set IVAL(1) = 1
C     (2) set IVAL(2) = 2
C     (3) leave IVAL(3) unchanged
C     (4) set IVAL(4) = 40000
C     (5) leave IVAL(5) unchanged
C     (6) set IVAL(6) = 60
C
C   IPPARI will print diagnostics on the default output device, if
C   desired.
C
C   IPPARI is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume IVAL = (0, 0, 0) and NEXPEC = 3 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ----------------------  ----    ------
C  '  2 ,   3 45 '         (2, 3, 45)                0       3
C  '2.15,,3'               (2, 0, 3)                 1       0
C  '3X, 25, 2'             (0, 0, 0)                 1       0
C  '10000'                 (10000, 0, 0)             2       1
C
C      Assume IVAL = (0, 0, 0, 0) and NEXPEC = -4 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ----------------------  ----    ------
C  '1, 2'                  (1, 2)                    0       2
C  ',,37  400'             (0, 0, 37, 400)           0       4
C  ' 1,,-3,,5'             (1, 0, -3, 0)             3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
```

```
C
C  IVAL (I,O) - the integer value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to IVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARI
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION IVAL(*)
      CHARACTER *8 FMT(14)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARI
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set false when a space follows
C--- an integer value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR .OR. NC .EQ. IE) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
```

```
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into integer array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      IES = NC - 1
      NCH = IES - IBS + 1
      DATA FMT/' (I1)', ' (I2)', ' (I3)', ' (I4)', ' (I5)',
     1    ' (I6)', ' (I7)', ' (I8)', ' (I9)', '(I10)',
     2    '(I11)', '(I12)', '(I13)', '(I14)'/
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) IVAL(NFOUND)
      GO TO 450
400   CONTINUE
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0)RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1    WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      SUBROUTINE IPPARR(STRING, ICARD, NEXPEC, RVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARR
C   REFER TO  IPGETR
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851625   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses real variables from a character variable.  Called
C            by IPGETR, the IOPAK routine used for interactive input.
```

```
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARR may be used for parsing an input record that contains real
C  values, but was read into a character variable instead of directly
C  into real variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARR to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of RVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1.,   2,,4.e-5   , ,6.e-6'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set RVAL(1) = 1.0
C     (2) set RVAL(2) = 2.0
C     (3) leave RVAL(3) unchanged
C     (4) set RVAL(4) = 4.0E-05
C     (5) leave RVAL(5) unchanged
C     (6) set RVAL(6) = 6.0E-06
C
C   IPPARR will print diagnostics on the default output device, if
C   desired.
C
C   IPPARR is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume RVAL = (0., 0., 0.) and NEXPEC = 3 on entry:
C
C   input string           RVAL on exit             IERR    NFOUND
C   ------------           ----------------------   ----    ------
C  '  2.34e-3,  3 45.1'    (2.34E-03, 3.0, 45.1)     0       3
C  '2,,3.-5'               (2.0, 0.0, 3.0E-05)       0       3
C  ',1.4,0.028E4'          (0.0, 1.4, 280.0)         0       3
C  '1.0, 2.a4, 3.0'        (1.0, 0.0, 0.0)           1       1
C  '1.0'                   (1.0, 0.0, 0.0)           2       1
C
C      Assume RVAL = (0.,0.,0.,0.) and NEXPEC = -4 on entry:
C
C   input string           RVAL on exit             IERR    NFOUND
C   ------------           ----------------------   ----    ------
C  '1.,2.'                 (1.0, 2.0)                0       2
C  ',,3  4.0'              (0.0, 0.0, 3.0, 4.0)      0       4
C  '1,,3,,5.0'             (0.0, 0.0, 3.0, 0.0)      3       4
C
C  arguments: (I=input,O=output)
C  ---------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
```

```
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  RVAL (I,O) - the real value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to RVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARR
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION RVAL(*)
      CHARACTER *8 FMT(22)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARR
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set negative when a space follows
C--- a real value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
```

```
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into real array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      DATA FMT/     ' (E1.0)', ' (E2.0)', ' (E3.0)', ' (E4.0)',
     1    ' (E5.0)', ' (E6.0)', ' (E7.0)', ' (E8.0)', ' (E9.0)',
     2    '(E10.0)', '(E11.0)', '(E12.0)', '(E13.0)', '(E14.0)',
     3    '(E15.0)', '(E16.0)', '(E17.0)', '(E18.0)', '(E19.0)',
     4    '(E20.0)', '(E21.0)', '(E22.0)'/
      IES = NC - 1
      NCH = IES - IBS + 1
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) RVAL(NFOUND)
      GO TO 450
400   CONTINUE
      WRITE (LOUT, 555) STRING(IBS:IES)
  555 FORMAT (A)
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0) RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1    '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1   WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1    ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2    '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1    ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
```

```
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END
      FUNCTION ILASCH(STRING)
C   BEGIN PROLOGUE  ILASCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines last significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the last non-blank character in a string of
C  arbitrary length.  If no characters are found, ILASCH is set = 0.
C  When used with the companion routine IFIRCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C  Note that the FORTRAN intrinsic function LEN returns the length
C  of a character string as declared, rather than as filled.  The
```

```
C  declared length includes leading and trailing blanks, and thus is
C  not useful in generating 'significant' substrings.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) STRING
C
C***FIRST EXECUTABLE STATEMENT ILASCH
      NLOOP = LEN(STRING)
      IF (NLOOP.EQ.0) THEN
         ILASCH = 0
         RETURN
      ENDIF
C
      DO 100 I = NLOOP, 1, -1
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
120   CONTINUE
      ILASCH = I
      END
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKCOMP (IST, IRAY, II, I)
C
C  START PROLOGUE
C
C  SUBROUTINE CKCOMP (IST, IRAY, II, I)*
C     Returns the index of an element of a reference character
C     string array which corresponds to a character string;
C     leading and trailing blanks are ignored.
C
C
C  INPUT
C     IST   - A character string.
C                   Data type - CHARACTER*(*)
C     IRAY  - An array of character strings;
C             dimension IRAY(*) at least II
C                   Data type - CHARACTER*(*)
C     II    - The length of IRAY.
C                   Data type - integer scalar.
C
C  OUTPUT
C     I     - The first integer location in IRAY in which IST
C             corresponds to IRAY(I); if IST is not also an
C             entry in IRAY, I=0.
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
```

```
      CHARACTER*(*) IST, IRAY(*)
C
      I = 0
      DO 10 N = II, 1, -1
         IS1 = IFIRCH(IST)
         IS2 = ILASCH(IST)
         IR1 = IFIRCH(IRAY(N))
         IR2 = ILASCH(IRAY(N))
         IF ( IS2.GE.IS1 .AND. IS2.GT.0 .AND.
     1        IR2.GE.IR1 .AND. IR2.GT.0 .AND.
     2        IST(IS1:IS2).EQ.IRAY(N)(IR1:IR2) ) I=N
   10 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKUNIT (LINE, AUNITS, EUNITS, IUNITS)
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
      CHARACTER*(*) LINE, IUNITS, AUNITS, EUNITS
      CHARACTER*4 UPCASE
C
      AUNITS = ' '
      EUNITS = ' '
      IUNITS = ' '
      DO 85 N = 1, ILASCH(LINE)-3
         IND = ILASCH(IUNITS)
         IF (EUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4)     .EQ. 'CAL/') THEN
               EUNITS = 'CAL/'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units cal/mole'
               ELSE
                  IUNITS(IND:) = ', E units cal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KCAL') THEN
               EUNITS = 'KCAL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kcal/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kcal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'JOUL') THEN
               EUNITS = 'JOUL'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Joules/mole'
               ELSE
                  IUNITS(IND:) = ', E units Joules/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KJOU') THEN
               EUNITS = 'KJOU'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kjoule/mole'
               ELSE
                  IUNITS(IND:) = ', E units Kjoule/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KELV') THEN
               EUNITS = 'KELV'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units Kelvins'
               ELSE
```

```
                  IUNITS(IND:) = ', E units Kelvins'
               ENDIF
            ENDIF
         ENDIF
         IF (AUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4) .EQ. 'MOLE') THEN
               IF (UPCASE(LINE(N+4:), 1) .EQ. 'S') THEN
                  AUNITS = 'MOLE'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units mole-cm-sec-K'
                  ELSE
                     IUNITS(IND:) = ', A units mole-cm-sec-K'
                  ENDIF
               ELSEIF (UPCASE(LINE(N+4:), 4) .EQ. 'CULE') THEN
                  AUNITS = 'MOLC'
                  IF (IUNITS .EQ. ' ') THEN
                     IUNITS = 'A units molecules'
                  ELSE
                     IUNITS(IND:) = ', A units molecules'
                  ENDIF
               ENDIF
            ENDIF
   85 CONTINUE
C
      IF (AUNITS .EQ. ' ') THEN
         AUNITS = 'MOLE'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', A units mole-cm-sec-K'
         ELSE
            IUNITS(IND:) = ' A units mole-cm-sec-K'
         ENDIF
      ENDIF
C
      IF (EUNITS .EQ. ' ') THEN
         EUNITS = 'CAL/'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', E units cal/mole'
         ELSE
            IUNITS(IND:) = ' E units cal/mole'
         ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      INTEGER FUNCTION IPPLEN (LINE)
C
C  BEGIN PROLOGUE
C
C  FUNCTION IPPLEN (LINE)
C     Returns the effective length of a character string, i.e.,
C     the index of the last character before an exclamation mark (!)
C     indicating a comment.
C
C  INPUT
C     LINE  - A character string.
C
C  OUTPUT
C     IPPLEN - The effective length of the character string.
C
C  END PROLOGUE
```

```
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C     IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER LINE*(*)
C
      IN = IFIRCH(LINE)
      IF (IN.EQ.0 .OR. LINE(IN:IN).EQ.'!') THEN
         IPPLEN = 0
      ELSE
         IN = INDEX(LINE,'!')
         IF (IN .EQ. 0) THEN
            IPPLEN = ILASCH(LINE)
         ELSE
            IPPLEN = ILASCH(LINE(:IN-1))
         ENDIF
      ENDIF
      RETURN
      END
C
      CHARACTER*(*) FUNCTION UPCASE(ISTR, ILEN)
      CHARACTER ISTR*(*), LCASE(26)*1, UCASE(26)*1
      DATA LCASE /'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1            'n','o','p','q','r','s','t','u','v','w','x','y','z'/,
     2     UCASE /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     3            'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
C
      UPCASE = ' '
      UPCASE = ISTR(:ILEN)
      JJ = MIN (LEN(UPCASE), LEN(ISTR), ILEN)
      DO 10 J = 1, JJ
         DO 10 N = 1,26
            IF (ISTR(J:J) .EQ. LCASE(N)) UPCASE(J:J) = UCASE(N)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE FILEIO_INT

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (LIN=5, LOUT=6, LTHRM=17, LINC=25)
      CHARACTER FILE_MECH*20,FILE_THM*20,FILE_INP*20,
     1          FILE_MOUT*20,FILE_OUT*20,FILE_LINK*20
      COMMON/FILE_IO/FILE_MECH,FILE_THM,FILE_INP,
     1               FILE_MOUT,FILE_OUT,FILE_LINK



  5   WRITE(*,10)
 10   FORMAT(' Please enter the input mechanism file name : ')
      READ(*,'(A20)')FILE_MECH
      OPEN(UNIT=5,FILE=FILE_MECH,STATUS='OLD',ERR=30)

      GO TO 50
 30   CONTINUE
      WRITE(*,40)FILE_MECH
 40   FORMAT(' Can not find file : ',A20)
      GO TO 5
 50   CONTINUE

 105  WRITE(*,110)
```

```
110   FORMAT(' Please enter the NASA thermo file name : ')
      READ(*,'(A20)')FILE_THM
      OPEN(UNIT=17,FILE=FILE_THM,STATUS='OLD',ERR=130)

      GO TO 150
130   CONTINUE
      WRITE(*,140)FILE_THM
140   FORMAT(' Can not find file : ',A20)
      GO TO 105
150   CONTINUE

      OPEN(UNIT=6,FILE='CKINTERP.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=25,FILE='CHEMKIN.LNK',STATUS='UNKNOWN',
     1                 FORM='BINARY')


      RETURN
      END
```

D.5 CKLIB4

```
C
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKRAT  (RCKWRK, ICKWRK, II, KK, MAXSP, MAXTB, RU, PATM,
     1                   T, C, NSPEC, NU, NUNK, NPAR, PAR, NREV, IREV,
     2                   RPAR, NFAL, IFAL, IFOP, KFAL, NFAR, FPAR, NLAN,
     3                   NLAR, ILAN, PLT, NRLT, IRLT, RPLT, NTHB, ITHB,
     4                   NTBS, AIK, NKTB, SMH, RKF, RKR, EQK, CTB)
C
C  START PROLOGUE
C
C  SUBROUTINE CKRAT  (RCKWRK, ICKWRK, II, KK, MAXSP, MAXTB, RU, PATM,
C 1                   T, C, NSPEC, NU, NUNK, NPAR, PAR, NREV, IREV,
C 2                   RPAR, NFAL, IFAL, IFOP, KFAL, NFAR, FPAR, NLAN,
C 3                   NLAR, ILAN, PLT, NRLT, IRLT, RPLT, NTHB, ITHB,
C 4                   NTBS, AIK, NKTB, SMH, RKF, RKR, EQK, CTB)
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H, O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H, O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION RCKWRK(*), ICKWRK(*), C(*), NSPEC(*), NU(MAXSP,*),
     1          NUNK(MAXSP,*), PAR(NPAR,*), IREV(*), RPAR(NPAR,*),
     2          ILAN(*), IRLT(*), PLT(NLAR,*), RPLT(NLAR,*),
     3          IFAL(*), IFOP(*), KFAL(*), FPAR(NFAR,*), ITHB(*),
     4          NTBS(*), AIK(MAXTB,*), NKTB(MAXTB,*), SMH(*),
     5          RKF(*), RKR(*), EQK(*), CTB(*)
C
      COMMON /MACH/ SMALL,BIG,EXPARG
C
c      WRITE(*,*)'...CALL...CKRAT....'
      ALOGT = LOG(T)
C
      DO 20 I = 1, II
         CTB(I) = 1.0
         RKF(I) = PAR(1,I) * EXP(PAR(2,I)*ALOGT - PAR(3,I)/T)
   20 CONTINUE
C
C        Landau-Teller reactions
C
      DO 25 N = 1, NLAN
         I = ILAN(N)
         TFAC = PLT(1,N)/T**(1.0/3.0) + PLT(2,N)/T**(2.0/3.0)
         RKF(I) = RKF(I) * EXP(TFAC)
   25 CONTINUE
C
      CALL CKSMH (T, ICKWRK, RCKWRK, SMH)
      DO 50 I = 1, II
          SUMSMH = 0.0
          DO 40 N = 1, MAXSP
             IF (NUNK(N,I).NE.0) SUMSMH=SUMSMH+NU(N,I)*SMH(NUNK(N,I))
   40     CONTINUE
C
          EQK(I) = EXP(MIN(SUMSMH,EXPARG))
   50 CONTINUE
```

```
C
      PFAC = PATM / (RU*T)
      DO 60 I = 1, II
         NUSUMK = NU(1,I)+NU(2,I)+NU(3,I)+NU(4,I)+NU(5,I)+NU(6,I)
         EQK(I) = EQK(I) * PFAC**NUSUMK
C
C     RKR=0.0 for irreversible reactions, else RKR=RKF/MAX(EQK,SMALL)
C
         RKR(I) = 0.0
         IF (NSPEC(I).GT.0) RKR(I) = RKF(I) / MAX(EQK(I),SMALL)
   60 CONTINUE
C
C     if reverse parameters have been given:
C
      DO 70 N = 1, NREV
         I = IREV(N)
         RKR(I) = RPAR(1,N) * EXP(RPAR(2,N)*ALOGT - RPAR(3,N)/T)
         EQK(I) = RKF(I)/RKR(I)
   70 CONTINUE
C
C     if reverse Landau-Teller parameters have been given:
C
      DO 75 N = 1, NRLT
         I = IRLT(N)
         TFAC = RPLT(1,N)/T**(1.0/3.0) + RPLT(2,N)/T**(2.0/3.0)
         RKR(I) = RKR(I) * EXP(TFAC)
         EQK(I) = RKF(I)/RKR(I)
   75 CONTINUE
C
C     third-body reactions
C
      CTOT = 0.0
      DO 10 K = 1, KK
         CTOT = CTOT + C(K)
   10 CONTINUE
C
      DO 80 N = 1, NTHB
         CTB(ITHB(N)) = CTOT
         DO 80 L = 1, NTBS(N)
            CTB(ITHB(N)) = CTB(ITHB(N)) + (AIK(L,N)-1.0)*C(NKTB(L,N))
   80 CONTINUE
C
C     If fall-off (pressure dependence):
C
c      WRITE(*,*)'....NFAL = ',NFAL
      DO 90 N = 1, NFAL
c        WRITE(*,*)'....    IFOP(',N,') = ',IFOP(N)
C
C        CONCENTRATION OF THIRD BODY
C
         IF (KFAL(N) .EQ. 0) THEN
            CTHB = CTB(IFAL(N))
            CTB(IFAL(N)) = 1.0
         ELSE
            CTHB = C(KFAL(N))
         ENDIF
C
         RKLOW = FPAR(1,N) * EXP(FPAR(2,N)*ALOGT - FPAR(3,N)/T)
         PR = RKLOW*CTHB / RKF(IFAL(N))
         PRLOG = LOG10(MAX(PR,SMALL))
C
         IF (IFOP(N) .EQ. 1) THEN
C
C           LINDEMANN FORM
C
            FC = 1.0
```

```
C
      ELSE
C
         IF (IFOP(N) .EQ. 2) THEN
C
C           SRI FORM
C
            XP = 1.0/(1.0 + PRLOG**2)
            FC = ((FPAR(4,N)*EXP(-FPAR(5,N)/T) + EXP(-T/FPAR(6,N)))
     1           **XP) * FPAR(7,N) * T**FPAR(8,N)
C
         ELSE
C
C           6-PARAMETER TROE FORM
C
            FCENT = (1.0-FPAR(4,N)) * EXP(-T/FPAR(5,N))
     1            + FPAR(4,N) * EXP(-T/FPAR(6,N))
C
C           7-PARAMETER TROE FORM
C
            IF (IFOP(N) .EQ. 4) FCENT = FCENT + EXP(-FPAR(7,N)/T)
C
C
C           8-PARAMETER EXT-TROE FORM
C
            IF (IFOP(N) .EQ. 5) FCENT = FCENT + EXP(-FPAR(7,N)/T)
            IF (IFOP(N) .EQ. 5) PR = PR * (CTHB)**(FPAR(8,N)-1.0)
C
            IF (IFOP(N) .EQ. 6) THEN
C
C     W.Ing 08/01/95
C     Now we defined     Tmin = 300.   (K)
C                        Tmax = 2500.  (K)
C                        Pmin = 1.0E-03   (ATM)
C                        Pmax = 1.0E+02   (ATM)
C   ***************************************************************
C   *  which can be redefined but make sure Cheby's polynomials   *
C   *  refitting with corresponding T & P range.  IMPORTANT !!!   *
C   ***************************************************************
C
               Tmin = 300.
               Tmax = 2500.
               Pmin = 1.0E-03
               Pmax = 1.0E+02
               T_cheb = (2.*(1./T)-(1/Tmin)-(1/Tmax)) /
     2                  ((1/Tmax)-(1/Tmin))
               P_cheb = (2.*dlog10(PATM)-dlog10(Pmin)-
     2                   dlog10(Pmax)) / (dlog10(Pmax)-dlog10(Pmin))
               RK_tempLog = 0.0
               DO 82 I = 1, FPAR(4,N)
                  DO 81 J = 1, FPAR(5,N)
c           write(*,*)RK_tempLog,((I-1)*M_cheb+J),X((I-1)*M_cheb+J),
c     2                 PhiCh(I,T_cheb),PhiCh(J,P_cheb)
                    RK_tempLog = RK_tempLog + FPAR((I-1)*
     2                           FPAR(5,N)+J+5,N)*PhiCh(I,T_cheb)*
     3                           PhiCh(J,P_cheb)
81                CONTINUE
82             CONTINUE
               FC = 10.0**RK_tempLog
            ELSE
               FCLOG = LOG10(MAX(FCENT,SMALL))
               XN    = 0.75 - 1.27*FCLOG
               CPRLOG= PRLOG - (0.4 + 0.67*FCLOG)
               FLOG = FCLOG/(1.0 + (CPRLOG/(XN-0.14*CPRLOG))**2)
               FC = 10.0**FLOG
            ENDIF
```

```
            ENDIF
         ENDIF
         IF (IFOP(N) .EQ. 6) THEN
            RKF(IFAL(N)) = FC
            RKR(IFAL(N)) = RKF(IFAL(N)) * FC
         ELSE
            PCOR = FC * PR/(1.0+PR)
            RKF(IFAL(N)) = RKF(IFAL(N)) * PCOR
            RKR(IFAL(N)) = RKR(IFAL(N)) * PCOR
         ENDIF
c         WRITE(*,*)'..IFOP = ',IFOP(N),'....RKF(',N,') = ',RKF(N)
c         WRITE(*,*)'..IFOP = ',IFOP(N),'....RKR(',N,') = ',RKR(N)
   90 CONTINUE
C
C     Multiply by the product of reactants and product of products
C     PAR(4,I) is a perturbation factor
C
      DO 150 I = 1, II
         RKF(I) = RKF(I)*CTB(I)*C(NUNK(1,I))**IABS(NU(1,I))*PAR(4,I)
         RKR(I) = RKR(I)*CTB(I)*C(NUNK(4,I))**NU(4,I)       *PAR(4,I)
         IF (NUNK(2,I) .NE. 0) THEN
            RKF(I)= RKF(I) * C(NUNK(2,I))**IABS(NU(2,I))
            IF (NUNK(3,I) .NE. 0)
     1         RKF(I) = RKF(I) * C(NUNK(3,I))**IABS(NU(3,I))
         ENDIF
         IF (NUNK(5,I) .NE. 0) THEN
            RKR(I) = RKR(I) * C(NUNK(5,I))**NU(5,I)
            IF (NUNK(6,I) .NE. 0) RKR(I) = RKR(I)*C(NUNK(6,I))**NU(6,I)
         ENDIF
  150 CONTINUE
C
      RETURN
      END
C


C
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKRATT (RCKWRK, ICKWRK, II, MAXSP, RU, PATM, T, NSPEC,
     1                   NU, NUNK, NPAR, PAR, NREV, IREV, RPAR, NLAN,
     2                   NLAR, ILAN, PLT, NRLT, IRLT, RPLT, SMH, RKFT,
     3                   RKRT, EQK)
C
C  START PROLOGUE
C
C  SUBROUTINE CKRATT (RCKWRK, ICKWRK, II, MAXSP, RU, PATM, T, NSPEC,
C 1                   NU, NUNK, NPAR, PAR, NREV, IREV, RPAR, NLAN,
C 2                   NLAR, ILAN, PLT, NRLT, IRLT, RPLT, SMH, RKFT,
C 3                   RKRT, EQK)
C
C  END PROLOGUE
C
C*****precision > double
        IMPLICIT DOUBLE PRECISION (A-H, O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C        IMPLICIT REAL (A-H, O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION RCKWRK(*), ICKWRK(*), NSPEC(*), NU(MAXSP,*),
     1          NUNK(MAXSP,*), PAR(NPAR,*), IREV(*), RPAR(NPAR,*),
     2          ILAN(*), IRLT(*), PLT(NLAR,*), RPLT(NLAR,*), SMH(*),
     3          RKFT(*), RKRT(*), EQK(*)
C
      COMMON /MACH/ SMALL,BIG,EXPARG
```

```
C
      ALOGT = LOG(T)
C
      DO 20 I = 1, II
         RKFT(I) = PAR(1,I) * EXP(PAR(2,I)*ALOGT - PAR(3,I)/T)
   20 CONTINUE
C
C        Landau-Teller reactions
C
      DO 25 N = 1, NLAN
         I = ILAN(N)
         TFAC = PLT(1,N)/T**(1.0/3.0) + PLT(2,N)/T**(2.0/3.0)
         RKFT(I) = RKFT(I) * EXP(TFAC)
   25 CONTINUE
C
       CALL CKSMH (T, ICKWRK, RCKWRK, SMH)
       DO 50 I = 1, II
          SUMSMH = 0.0
          DO 40 N = 1, MAXSP
             IF (NUNK(N,I).NE.0) SUMSMH=SUMSMH+NU(N,I)*SMH(NUNK(N,I))
   40     CONTINUE
          EQK(I) = EXP(MIN(SUMSMH,EXPARG))
   50 CONTINUE
C
      PFAC = PATM / (RU*T)
      DO 60 I = 1, II
         NUSUMK = NU(1,I)+NU(2,I)+NU(3,I)+NU(4,I)+NU(5,I)+NU(6,I)
         EQK(I) = EQK(I) * PFAC**NUSUMK
C
C     RKRT=0.0 for irreversible reactions, else RKRT=RKFT/MAX(EQK,SMALL)
C
         RKRT(I) = 0.0
         IF (NSPEC(I).GT.0) RKRT(I) = RKFT(I) / MAX(EQK(I),SMALL)
   60 CONTINUE
C
C     if reverse parameters have been given:
C
      DO 70 N = 1, NREV
         I = IREV(N)
         RKRT(I) = RPAR(1,N) * EXP(RPAR(2,N)*ALOGT - RPAR(3,N)/T)
         EQK(I)  = RKFT(I)/RKRT(I)
   70 CONTINUE
C
C     if reverse Landau-Teller parameters have been given:
C
      DO 75 N = 1, NRLT
         I = IRLT(N)
         TFAC = RPLT(1,N)/T**(1.0/3.0) + RPLT(2,N)/T**(2.0/3.0)
         RKRT(I) = RKRT(I) * EXP(TFAC)
         EQK(I) = RKFT(I)/RKRT(I)
   75 CONTINUE
C
      DO 80 I = 1, II
         RKFT(I) = RKFT(I) * PAR(4,I)
         RKRT(I) = RKRT(I) * PAR(4,I)
   80 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKRATX (II, KK, MAXSP, MAXTB, T, C, NSPEC, NU, NUNK,
     1                   NFAL, IFAL, IFOP, KFAL, NFAR, FPAR, NTHB, ITHB,
     2                   NTBS, AIK, NKTB, RKFT, RKRT, RKF, RKR, CTB,
     3                   PATM)
```

```
C
C  START PROLOGUE
C
C  SUBROUTINE CKRATX (II, KK, MAXSP, MAXTB, T, C, NSPEC, NU, NUNK,
C 1                   NFAL, IFAL, IFOP, KFAL, NFAR, FPAR, NTHB, ITHB,
C 2                   NTBS, AIK, NKTB, RKFT, RKRT, RKF, RKR, CTB)
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H, O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H, O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION C(*), NSPEC(*), NU(MAXSP,*), NUNK(MAXSP,*),
     1          IFAL(*), IFOP(*), KFAL(*), FPAR(NFAR,*), ITHB(*),
     2          NTBS(*), AIK(MAXTB,*), NKTB(MAXTB,*), RKFT(*),
     3          RKRT(*), RKF(*), RKR(*), CTB(*)
C
      COMMON /MACH/ SMALL,BIG,EXPARG
C
      DO 20 I = 1, II
         CTB(I) = 1.0
         RKF(I) = RKFT(I)
         RKR(I) = RKRT(I)
   20 CONTINUE
C
C     third-body reactions
C
      CTOT = 0.0
      DO 10 K = 1, KK
         CTOT = CTOT + C(K)
   10 CONTINUE
C
      DO 80 N = 1, NTHB
         CTB(ITHB(N)) = CTOT
         DO 80 L = 1, NTBS(N)
            CTB(ITHB(N)) = CTB(ITHB(N)) + (AIK(L,N)-1.0)*C(NKTB(L,N))
   80 CONTINUE
C
C     If fall-off (pressure correction):
C
      ALOGT = LOG(T)
C
      DO 90 N = 1, NFAL
C
C          WRITE(*,*)' IFOP(',N,') = ',IFOP(N)
         RKLOW = FPAR(1,N) * EXP(FPAR(2,N)*ALOGT - FPAR(3,N)/T)
C          WRITE(*,*)' RKLOW = ',RKLOW
C
C        CONCENTRATION OF THIRD BODY
C
         IF (KFAL(N) .EQ. 0) THEN
            PR = RKLOW * CTB(IFAL(N)) / RKF(IFAL(N))
            CTB(IFAL(N)) = 1.0
         ELSE
            PR = RKLOW * C(KFAL(N)) / RKF(IFAL(N))
         ENDIF
C
         PCOR = PR / (1.0 + PR)
C
         IF (IFOP(N) .GT. 1) THEN
            PRLOG = LOG10(MAX(PR,SMALL))
C
```

```
            IF (IFOP(N) .EQ. 2) THEN
C
C              8-PARAMETER SRI FORM
C
               XP = 1.0/(1.0 + PRLOG**2)
               FC = ((FPAR(4,N)*EXP(-FPAR(5,N)/T) + EXP(-T/FPAR(6,N)))
     1              **XP) * FPAR(7,N) * T**FPAR(8,N)
C
            ELSE
C
C              6-PARAMETER TROE FORM
C
C               WRITE(*,*)'  BEFORE EXP'
               IF (IFOP(N).NE.6) FCENT = (1.0-FPAR(4,N)) *
     1            EXP(-T/FPAR(5,N)) + FPAR(4,N) * EXP(-T/FPAR(6,N))
C               WRITE(*,*)' FCENT = ',FCENT
C
C              7-PARAMETER TROE FORM
C
               IF (IFOP(N) .EQ. 4) FCENT = FCENT + EXP(-FPAR(7,N)/T)
C
C     W.ING 07/11/95
C
C              8-PARAMETER EXT-TROE FORM
C
               IF (IFOP(N) .EQ. 5) FCENT = FCENT + EXP(-FPAR(7,N)/T)
               IF (IFOP(N) .EQ. 5) PR = PR * (CTHB)**(FPAR(8,N)-1.0)
C
               IF (IFOP(N) .EQ. 6) THEN
C
C     W.Ing 08/01/95
C     Now we defined      Tmin = 300.  (K)
C                         Tmax = 2500. (K)
C                         Pmin = 1.0E-03  (ATM)
C                         Pmax = 1.0E+02  (ATM)
C     ***************************************************************
C     *  which can be redefined but make sure Cheby's polynomials   *
C     *  refitting with corresponding T & P range.  IMPORTANT !!!   *
C     ***************************************************************
C
                  Tmin = 300.
                  Tmax = 2500.
                  Pmin = 1.0E-03
                  Pmax = 1.0E+02
                  P = CTOT * 8.314E+07 * T / 1.01325E6
                  T_cheb = (2.*(1./T)-(1/Tmin)-(1/Tmax)) /
     2                     ((1/Tmax)-(1/Tmin))
                  P_cheb = (2.*dlog10(P)-dlog10(Pmin)-
     2                      dlog10(Pmax)) / (dlog10(Pmax)-dlog10(Pmin))
C                  WRITE(*,*)' T = ',T,' T_cheb = ',T_cheb
C                  WRITE(*,*)' P = ',P,' P_cheb = ',P_cheb
                  RK_tempLog = 0.0
                  DO 82 I = 1, FPAR(4,N)
                     DO 81 J = 1, FPAR(5,N)
C            write(*,*)RK_tempLog,((I-1)*M_cheb+J),X((I-1)*M_cheb+J),
C     2                 PhiCh(I,T_cheb),PhiCh(J,P_cheb)
                     RK_tempLog = RK_tempLog + FPAR((I-1)*
     2                            FPAR(5,N)+J+5,N)*PhiCh(I,T_cheb)*
     3                            PhiCh(J,P_cheb)
81                   CONTINUE
82                CONTINUE
C                  WRITE(*,*)'...RK_TEMPLOG = ',RK_tempLog
                  FC = 10.0**RK_tempLog
C                  WRITE(*,*)'...FC = ',FC
               ELSE
C                  FCLOG = LOG10(MAX(FCENT,SMALL))
```

```
                  XN    = 0.75 - 1.27*FCLOG
                  CPRLOG= PRLOG - (0.4 + 0.67*FCLOG)
                  FLOG = FCLOG/(1.0 + (CPRLOG/(XN-0.14*CPRLOG))**2)
                  FC = 10.0**FLOG
               ENDIF
            ENDIF
         ENDIF
         IF (IFOP(N) .EQ. 6) THEN
            RKF(IFAL(N)) = FC
C            WRITE(*,*)' RKR = ',RKR(IFAL(N)),' RKFT = ',RKFT(IFAL(N))
            RKR(IFAL(N)) = RKR(IFAL(N)) * FC / RKFT(IFAL(N))
         ELSE
            IF (IFOP(N) .GT. 1) PCOR = FC * PCOR
            RKF(IFAL(N)) = RKF(IFAL(N)) * PCOR
            RKR(IFAL(N)) = RKR(IFAL(N)) * PCOR
         ENDIF
c         WRITE(*,*)'..IFOP = ',IFOP(N),'....RKF(',IFAL(N),') = ',
c     2                          RKF(IFAL(N))
c         WRITE(*,*)'..IFOP = ',IFOP(N),'....RKR(',IFAL(N),') = ',
c     2                          RKR(IFAL(N))
   90 CONTINUE
C
CC               FCLOG = LOG10(MAX(FCENT,SMALL))
C               XN    = 0.75 - 1.27*FCLOG
C               CPRLOG= PRLOG - (0.4 + 0.67*FCLOG)
C               FLOG = FCLOG/(1.0 + (CPRLOG/(XN-0.14*CPRLOG))**2)
C               FC = 10.0**FLOG
C            ENDIF
C            PCOR = FC * PCOR
C         ENDIF
C
C         RKF(IFAL(N)) = RKF(IFAL(N)) * PCOR
C         RKR(IFAL(N)) = RKR(IFAL(N)) * PCOR
C   90 CONTINUE
C
C     Multiply by the product of reactants and product of products
C
      DO 150 I = 1, II
         RKF(I) = RKF(I)*CTB(I)*C(NUNK(1,I))**IABS(NU(1,I))
         RKR(I) = RKR(I)*CTB(I)*C(NUNK(4,I))**NU(4,I)
         IF (NUNK(2,I) .NE. 0) THEN
            RKF(I) = RKF(I) * C(NUNK(2,I))**IABS(NU(2,I))
            IF (NUNK(3,I) .NE. 0)
     1         RKF(I) = RKF(I) * C(NUNK(3,I))**IABS(NU(3,I))
         ENDIF
         IF (NUNK(5,I) .NE. 0) THEN
            RKR(I) = RKR(I) * C(NUNK(5,I))**NU(5,I)
            IF (NUNK(6,I) .NE. 0)
     1         RKR(I) = RKR(I) * C(NUNK(6,I))**NU(6,I)
         ENDIF
  150 CONTINUE
C
      RETURN
      END
```

D.6 MECHCVT

```
PROGRAM MECHCVT
C
C-----------------------------------------------------------------------C
c***************changed W.Ing 06-15-94, 08-01-95***********************
C
C     MDIM = maximum number of elements in a problem;                (20)
C     KDIM = maximum number of species in a problem;                (500)
C     MAXTP= maximum number of temperatures used to fit               (3)
C            thermodynamic properties of species
C     NPC  = number of polynomial coefficients to fits               (5)
C     NPCP2= number of fit coefficients for a temperature range      (7)
C     IDIM = maximum number of reactions in a mechanism;            (1500)
C     NPAR = number of Arrhenius parameters in a reaction;           (3)
C     NLAR = number of Landau-Teller parameters in a reaction;       (2)
C     NFAR = number of fall-off parameters in a reaction;           (66)
C     MAXSP= maximum number of species in a reaction                 (6)
C     MAXTB= maximum number of third bodies for a reaction          (10)
C     LSYM = character string length of element and species names   (16)
C
C     User input is read from LIN (Unit15), a thermodynamic database
C     is read from LTHRM (Unit17), printed output is assigned to LOUT
C     (Unit16), and binary linking data is written to LINC (Unit25).
C
C     REQUIRED ELEMENT INPUT: (Subroutine CKCHAR)           (DIMENSION)
C
C        The word 'ELEMENTS' followed by a list of element
C        names, terminated by the word 'END';
C
C        The resulting element data stored in LINK is:
C        MM       - integer number of elements found
C        ENAME(*) - CHARACTER*(*) array of element names         (MDIM)
C        AWT(*)   - real array of atomic weights;                (MDIM)
C                   default atomic weights are those on
C                   atomic weight charts; if an element
C                   is not on the periodic chart, or if
C                   it is desirable to alter its atomic
C                   weight, this value must be included
C                   after the element name, enclosed by
C                   slashed, i.e., D/2.014/
C
C     REQUIRED SPECIES INPUT: (Subroutine CKCHAR)
C
C        The word 'SPECIES' followed by a list of species
C        names, terminated by the word 'END';
C
C        The resulting species data stored in LINK is:
C        KK       - integer number of species found
C        KNAME(*) - CHARACTER*(*) array of species names         (KDIM)
C
C     OPTIONAL THERMODYNAMIC DATA: (Subroutine CKTHRM)
C     (If this feature is not used, thermodynamic properties are
C     obtained from a CHEMKIN database.)  The format for this option
C     is the word 'THERMO' followed by any number of 4-line data sets:
C
C     Line 1: species name, optional comments, elemental composition,
C             phase, T(low), T(high), T(mid), additional elemental
C             composition, card number (col. 80);
C             format(A10,A14,4(A2,I3),A1,E10.0,E10.0,E8.0,(A2,I3),I1)
C     Line 2: coefficients a(1--5) for upper temperature range,
C             card number (col. 80);
```

```
C             format(5(e15.0),I1)
C     Line 3: coefficients a(6--7) for upper temperature range,
C             coefficients a(1--3) for lower temperature range,
C             card number (col. 80);
C             format(5(e15.0),I1)
C     Line 4: coefficients a(4--7) for lower temperature range,
C             card number (col. 80);
C             format(4(e15.0),I1)
C
C     End of THERMO data is indicated by 'END' line or new keyword.
C
C        The resulting thermodynamic data stored in LINK are:
C        WTM(*)    - real array of molecular weights              (KDIM)
C        KNCF(*,*)- integer composition of species          (MDIM,KDIM)
C        KPHSE(*) - integer phase of a species;                 (KDIM)
C                   -1(solid), 0(gas), +1(liquid).
C        KCHRG(*) - ionic charge of a species;                  (KDIM)
C                   = 0 except in presence/absence of electrons
C                   = +n in absence of n electrons
C                   = -n in presence of n electons
C        NCHRG     - integer number of species with KCHRG<>0
C        NT(*)     - array of number of temperatures used       (KDIM)
C                   in fits
C        T(*,*)    - array of temperatures used in fits     (MAXTP,KDIM)
C        A(N,L,K) - Thermodynamic properties for       (NPC+2,NTR,KDIM)
C                   species K consists of polynomial
C                   coefficients for fits to
C                   CP/R = SUM (A(N,L,K)*Temperature**(N-1), N=1,NPC+2)
C                        where  T(L,K) <= Temperature < T(L+1,K),
C                   and,
C                   N=NPC+1 is formation enthalpy HO/R = A(NPC+1,L,K),
C                   N=NPC+2 is formation entropy  SO/R = A(NPC+2,L,K)
C
C     OPTIONAL REACTION INPUT:
C     Reaction data is input after all ELEMENT, SPECIES and THERMO
C     data in the following format:
C
C     1) (Subroutine CKREAC)
C        The first line contains the keyword 'REACTIONS' and an
C        optional description of units:
C
C           'MOLES' - (default), pre-exponential units are moles-sec-K;
C           'MOLECULES' - pre-exponential units are molecules and
C                         will be converted to moles.
C           'KELVINS' - activation energies are Kelvins, else the
C                       activation energies are converted to Kelvins;
C           'CAL/MOLE' - (default), activation energies are cal/mole;
C           'KCAL/MOLE' - activation energies are Kcal/mole;
C           'JOULES/MOLE' - activation energies are joules/mole;
C           'KJOULES/MOLE' - activation energies are Kjoules/mole.
C
C        A description of each reaction is expected to follow.
C        Required format for a reaction is a list of '+'-delimited
C        reactants, followed by a list of '+'-delimited reactants,
C        each preceded by its stoichiometric coefficient if greater
C        than 1;  separating the reactants from the products is a '='
C        if reversible reaction, else a '=>'.  Following the reaction
C        string on the same line are the space-delimited Arrhenius
C        coefficients.
C
C        If the reaction contains a third body, this is indicated by
C        by the presence of an 'M' as a reactant or product or both,
C        and enhancement factors for third-bodies may be defined on
C        additional lines as described in (2).
C
C        If the reaction contains a radiation wavelength, this is
```

```
C       indicated by the presence of an 'HV' either as a reactant
C       or as a product.  Unless otherwise defined on additional
C       lines as described in (2), the value of the wavelength is
C       -1.0 if a reactant or +1.0 if a product.
C
C       If the reaction is a fall-off reaction, this is indicated
C       either by a '(+M)' or a '(+KNAME(K))', and there must be
C       additional lines as described in (2) to define fall-off
C       parameters.
C
C    2) (Subroutine CKAUXL)
C       Additional information for a reaction is given on lines
C       immediately following the reaction description; this data
C       will consist of a 'keyword' to denote the type of data,
C       followed by a '/', then the required parameters for the
C       keyword, followed by another '/'.  There may be more than
C       one keyword per line, and there may be any number of lines.
C       The keywords and required parameters are as follows:
C
C       KNAME(K)/efficiency value/ - species (K) is an enhanced
C              third body in the reaction
C       HV/wavelength/ - radiation wavelength parameter
C       LT/val1 val2/ - Landau-Teller coefficients
C       LOW/val1 val2 val3/ - low fall-off parameters
C       TROE/val1 val2 val3 val4/ - Troe fall-off parameters;
C                                   if val4 is omitted, a default
C                                   parameter will be used
C
C       ADDED W.ING, 06/15/95**********************************
C
C       EXTROE/val1 val2 val3 val4 val5/ -
C                                  Extended Troe fall-off parameters;
C
C                                  val1-val4 = Troe parameters
C                                  val5 = well # (for cal. ext reduced
C                                                 pressure)
C
C       ******************************************************
C
C       SRI/val1 val2 val3 val4/ - SRI fall-off parameters;
C                                  if val4 is omitted, a default
C                                  parameter will be used
C          (it is an error to have both LT and Fall-off defined)
C       REV/par1 par2 par3/ - reverse parameters given
C       RLT/val1 val2/ - Landau-Teller coefficients for reverse
C          (it is an error if REV given and not RLT)
C
C    The end of all reaction data is indicated by an 'END' card or
C    <eof>.
C
C    Resulting reaction data stored in LINC are:
C      II        - integer number of reactions found
C      PAR(*,*)  - array of real Arrhenius coefficients   (NPAR,IDIM)
C      NSPEC(*)  - total number of species in a reaction       (IDIM)
C                  if NSPEC < 0, reaction is irreversible
C      NREAC(*)  - number of reactants only                    (IDIM)
C      NUNK(*,*) - array of species numbers for reaction (MAXSP,IDIM)
C      NU(*,*)   - array of stoichiometric coefficients  (MAXSP,IDIM)
C                  of species in a reaction, negative=reactant,
C                  positive=product
C
C      NWL       - number of reactions with radiation wavelength
C      IWL(*)    - integer reaction numbers                    (IDIM)
C      WL(*)     - real radiation wavelengths                  (IDIM)
C
C      NTHB      - number of reactions with third bodies
```

```
C       ITHB       - integer reaction numbers                        (IDIM)
C       NTBS(*)    - total number of enhanced species for NTHB       (IDIM)
C       NKTB(*,*)  - species numbers of enhanced species     (MAXTB,IDIM)
C       AIK(*,*)   - enhancement factors                     (MAXTB,IDIM)
C
C       NFAL       - number of fall-off reactions
C       IFAL(*)    - integer reaction numbers                        (IDIM)
C       KFAL(*)    - integer species number for which
C                    concentrations are a factor in fall-off
C                    calculation
C       IFOP(*)    - integer fall-off type number                    (IDIM)
C                    = 0 if fall-off reaction is found
C                    = 1 for Lindemann form
C                    = 2 for 6-parameter Troe form
C                    = 3 for 7-parameter Troe form
C                    = 4 for SRI form
C                    = 5 for 8-parameter Extended Troe form
C       PFAL(*,*)  - fall-off parameters                     (NFAR,IDIM)
C
C       NLAN       - number of reactions with Landau-Teller
C       ILAN       - integer reaction numbers                        (IDIM)
C       PLAN       - Landau-Teller parameters                (NLAR,IDIM)
C
C       NREV       - number of reactions with reverse parameters
C       IREV(*)    - integer reaction numbers                        (IDIM)
C       RPAR(*,*)  - parameters                              (NPAR,IDIM)
C
C       NRLT       - number of reactions with reverse parameters
C                    and Landau-Teller parameters
C       IRLT(*)    - integer reaction numbers                        (IDIM)
C       RLAN(*,*)  - reverse Teller-Laudauer parameters      (NLAR,IDIM)
C
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
C      PARAMETER (MDIM=20, KDIM=500, MKDIM=MDIM*KDIM, IDIM=1500, LSYM=16,
C     1           NPAR=3, NPIDIM=IDIM*NPAR, NPC=5, NPCP2=NPC+2, MAXTP=3,
C     2           NTR=MAXTP-1, NKTDIM=NTR*NPCP2*KDIM, MAXSP=6, MAXTB=10,
C     3           NLAR=2, NSIDIM=MAXSP*IDIM, NTIDIM=MAXTB*IDIM,
C     4           NLIDIM=NLAR*IDIM, NFAR=70, NFIDIM=NFAR*IDIM,
C     5           NTDIM=KDIM*MAXTP, NIDIM=9*IDIM)
      PARAMETER  (LIN=5, LOUT=6,
     6           LOG=15, LINC=25)
C
      CHARACTER SUB(120)*120,SUB_CH(120)*120,RSTR*120,RSTR_CH*120,
     1          KEY(5)*4, LINE*100,
     2          UPCASE*4, FILE_LOG*20,
     3          LINE1*120,LINE2*120,LINE3*120,LINE_CH*120
C
C      DIMENSION AWT(MDIM), KNCF(MDIM,KDIM), WTM(KDIM), KPHSE(KDIM),
C     1          KCHRG(KDIM), A(NPCP2,NTR,KDIM), T(MAXTP,KDIM), NT(KDIM),
C     2          NSPEC(IDIM), NREAC(IDIM), NU(MAXSP,IDIM),
C     3          NUNK(MAXSP,IDIM), PAR(NPAR,IDIM), IDUP(IDIM),IREV(IDIM),
C     4          RPAR(NPAR,IDIM), ILAN(IDIM), PLAN(NLAR,IDIM),
C     5          IRLT(IDIM), RLAN(NLAR,IDIM), IWL(IDIM),  WL(IDIM),
C     6          IFAL(IDIM), IFOP(IDIM), KFAL(IDIM), PFAL(NFAR,IDIM),
C     7          ITHB(IDIM),NTBS(IDIM),AIK(MAXTB,IDIM),NKTB(MAXTB,IDIM)
C      DIMENSION VALUE(5)
C
      REAL*8 T_LOW,T_HIGH,PHICH,T,P,K_fit,lnK,T_cheb, P_cheb
      CHARACTER*100 ISTR, IREAC, IPROD, ISPEC, INAME, ITEMP
```

```
      DIMENSION CHEB(70),T(51),RK(51),lnK(51),PAR(3)
C      CHARACTER*80 ISTR, IREAC, IPROD, ISPEC, INAME, ITEMP
      LOGICAL KERR,M_PPP,COMM
      CHARACTER FILE_IN*20,FILE_OUT*20
      COMMON/FILE_IO/FILE_IN,FILE_OUT
C      , THERMO, ITHRM(KDIM)
C
C     Initialize variables
C
C      DATA KEY/'ELEM','SPEC','THER','REAC','END'/
C      , KERR/.FALSE./
C     1     ITASK,NCHRG,MM,KK,II,NLAN,NFAL,NTHB,NREV,NRLT,NWL/11*0/,
C     2     ENAME,AWT/MDIM*' ',MDIM*0.0/, THERMO/.TRUE./,
C     3     T/NTDIM*-1.0/, KNAME,WTM,NT,KPHSE,KCHRG,ITHRM
C     4     /KDIM*' ', KDIM*0.0, KDIM*3, KDIM*0, KDIM*0, KDIM*.FALSE./,
C     5     WL,IFOP,NTBS,IDUP /IDIM*0.0, IDIM*-1, IDIM*0, IDIM*0/,
C     6     NSPEC,NREAC,IREV,ILAN,IRLT,IWL,IFAL,KFAL,ITHB/NIDIM*0/
C
C      DATA NUNK,NU/NSIDIM*0, NSIDIM*0/, NKTB,AIK/NTIDIM*0,NTIDIM*-1.0/
C      DATA PAR,RPAR/NPIDIM*0.0, NPIDIM*0.0/
C      DATA PLAN,RLAN/NLIDIM*0.0, NLIDIM*0.0/
C      DATA PFAL/NFIDIM*0.0/, KNCF/MKDIM*0.0/, A/NKTDIM*0.0/
C-----------------------------------------------------------------------C
C
C
C     ADD FILEIO SUBROUTINE 02/94 ING
C
      CALL FILEIO_CVT
      FILE_LOG = 'MECHCVT.LOG'
      OPEN(UNIT=15,FILE=FILE_LOG,STATUS='UNKNOWN',FORM='FORMATTED')
    5 WRITE(*,6)
    6 FORMAT(' Enter PRESSURE (atm) for output mechanism : ')
      READ(*,*)P
      IF (P.LE.1E2 .OR. P.GE.1E-3) GO TO 8
      IF (P.GT.1E2 .OR. P.LT.1E-3) THEN
          WRITE(*,7)P
    7     FORMAT(' Pressure " ',1pe8.2,' atm " is out of 1.0e-3 and 1.0e
     2+2 atm range' )
          write(*,*)'  please re-enter a valid pressure '
          write(*,*)' '
      ENDIF
      GO TO 5
    8 CONTINUE
      WRITE(*,*)' Enter Temperature range (K) for output mechanism '
   51 WRITE(*,*)'   Temperature (low) = '
      READ(*,*)T_LOW
      IF (T_LOW.LE.2500. .OR. T_LOW.GE.300.) GO TO 53
      IF (P.GT.1E2 .OR. P.LT.1E-3) THEN
          WRITE(*,52)T_LOW
   52     FORMAT(' Temperature " ',f7.1,' K " is out of 300 and 2500 K r
     2ange' )
          write(*,*)'  please re-enter a valid temperature '
          write(*,*)' '
      ENDIF
      GO TO 51
   53 CONTINUE
   54 WRITE(*,*)'   Temperature (high) = '
      READ(*,*)T_HIGH
      IF (T_LOW.LE.2500 .OR. T_LOW.GE.300) GO TO 56
      IF (T_HIGH.GT.2500 .OR. T_HIGH.LT.300) THEN
          WRITE(*,55)T_HIGH
   55     FORMAT(' Temperature " ',f7.1,' K " is out of 300 and 2500 K r
     2ange' )
          write(*,*)'  please re-enter a valid temperature '
          write(*,*)' '
      ENDIF
```

```
      GO TO 54
   56 CONTINUE
      IF (T_LOW .GT.T_HIGH) THEN
          T_TEMP = T_LOW
          T_LOW = T_HIGH
          T_HIGH = T_TEMP
      ENDIF
      IF (T_LOW.EQ.T_HIGH) THEN
          T_HIGH = T_HIGH + 10.
          T_LOW = T_LOW - 10.
      ENDIF
C
C
C        START OF MECHANISM CONVERSION
C
      R=1.987
      M = 0
      LINE3 = ' '
      LINE2 = ' '
      LINE1 = ' '
      ILEN3 = 0
      ILEN2 = 0
      ILEN1 = 0
      COMM = .FALSE.
  100 CONTINUE
      IF (.NOT.COMM) THEN
          M = M + 1
          LINE3 = LINE2
          LINE2 = LINE1
          LINE1 = ' '
          ILEN3 = ILEN2
          ILEN2 = ILEN1
          ILEN1 = 0
      ENDIF
      COMM = .FALSE.
      DO 102 I = 1, 120
          SUB(I) = ' '
  102 CONTINUE
      READ (LIN,'(A)',END=5000) LINE1
C      write(*,*)line1
  105 CONTINUE
      ILEN1 = IPPLEN(LINE1)
      IF (LINE1(1:1) .EQ. '!') THEN
          M = M -1
          COMM = .TRUE.
          GO TO 100
      ENDIF
C      IF (ILEN1 .EQ. 0) GO TO 100
C
      CALL CKISUB (LINE1(:ILEN1), SUB, NSUB)
C
C
C
      N = 0
      IF (UPCASE(SUB(1),4) .EQ. 'CHEB') THEN

          LINE = LINE3

C
C
C
C
C
          I1 = INDEX(SUB(NSUB),'/')
          I2 = INDEX(SUB(NSUB)(I1+1:),'/')
c          WRITE(*,*)' I1 = ',I1,' I2 = ',I2
```

```
            IF (I1.LE.0 .OR. I2.LE.0) THEN
               KERR = .TRUE.
               WRITE (LOUT, 2090) SUB(NSUB)(:ILEN)
C               GO TO 500
            ENDIF
C            KEY = SUB(N)(:I1-1)
            RSTR = ' '
            RSTR = SUB(NSUB)(I1+1:I1+I2-1)
C            WRITE(*,*)RSTR
            CALL IPPARR (RSTR,1,-8,CHEB(1),NVAL,IER,LOUT)
            N_CHEB = CHEB(1)
            M_CHEB = CHEB(2)
            NPARA_CHEB = N_CHEB*M_CHEB + 2
            INDEX_CHEB = 0
C            write(*,*)'..NPARA....',NPARA_CHEB,N_CHEB,M_CHEB
C                      CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
            INDEX_CHEB = INDEX_CHEB + NVAL
C                      WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
C                      WRITE(*,*)' ..NPARA..INDEX..NVAL.',NPARA_CHEB,NVAL
C                      WRITE(*,*)'...N...M...',N_CHEB,M_CHEB
C                      WRITE (LOUT, 3098) (PFAL(L,NFAL),L=4,NVAL+3)
101         CONTINUE
            LINE_CH = ' '
            READ (LIN,'(A)') LINE_CH
C            WRITE(*,*)' ..LINE_CH..'
C            WRITE(*,*)LINE_CH
C
            ILEN = IPPLEN(LINE_CH)
C            WRITE(*,*)ILEN
            CALL CKISUB (LINE_CH(:ILEN), SUB_CH, NSUB_CH)
C                        KEY = ' '
            DO 151 N_CH = 1, NSUB_CH
                IILEN = ILASCH(SUB_CH(NSUB_CH))
C                WRITE(*,*)'   NSUB_CH = ',NSUB_CH
C                WRITE(*,*)SUB_CH(NSUB_CH)
                II1 = INDEX(SUB_CH(NSUB_CH),'/')
                II2 = INDEX(SUB_CH(NSUB_CH)(II1+1:),'/')
C                WRITE(*,*)'....II1..II2...',II1,II2
                IF (II1.LE.0 .OR. II2.LE.0) THEN
                    KERR = .TRUE.
                    WRITE (LOUT, 2090) SUB_CH(N_CH)(:IILEN)
C                    GO TO 199
                ENDIF
C                        KEY = SUB(N)(:I1-1)
                RSTR_CH = ' '
                RSTR_CH = SUB_CH(NSUB_CH)(II1+1:II1+II2-1)
C                WRITE(*,*)RSTR_CH
                CALL IPPARR (RSTR_CH,1,-8,CHEB(INDEX_CHEB+1)
     2                                    ,NVAL,IER,LOUT)
C                WRITE(*,*)'...INDEX..NVAL..',INDEX_CHEB,NVAL
C                WRITE(*,*)'...IER..= ',IER
C                        INDEX_CHEB = INDEX_CHEB + NVAL
C                WRITE (*, 3098) (CHEB(L),L=INDEX_CHEB+1
C     2                                    ,NVAL+INDEX_CHEB)
C                WRITE (LOUT, 3098) (PFAL(L,NFAL),L=INDEX_CHEB+4
C     2                                    ,NVAL+INDEX_CHEB+3)
                INDEX_CHEB = INDEX_CHEB + NVAL
C                WRITE(*,*)'..INDEX...',INDEX_CHEB
151         CONTINUE
            IF (INDEX_CHEB.LT.NPARA_CHEB) GOTO 101
199         CONTINUE
            IF (INDEX_CHEB .EQ. NPARA_CHEB) THEN
C
            ELSE
                WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
                KERR = .TRUE.
```

```
          ENDIF
C    ***************************************************
C    ***************W.ING 6/16/95************************


C201        N = N + 1
C           WRITE(LOUT,*)'.....CALL SUB CHEB....',N




C           WRITE(LOUT,810)LINE3
C           LINE3 = LINE2
C           LINE2 = LINE1
C           ILEN3 = ILEN2
C           ILEN2 = ILEN1
C           READ (LIN,'(A)',END=5000) LINE1
C           ILEN1 = IPPLEN(LINE1)
C           CALL CKISUB (LINE1(:ILEN1), SUB, NSUB)
C           IF(UPCASE(SUB(1),4) .EQ. 'CHEB') GOTO 101
C           WRITE(LOUT,810)LINE3
           LINE3 = ' '
           LINE2 = ' '
           LINE1 = ' '
           ILEN3 = 0
           ILEN2 = 0
           ILEN1 = 0
c           DO 299 N = 1, NPARA_CHEB
c               WRITE(LOG,*)' CHEB(',N,') = ',CHEB(N)
c299        CONTINUE
           Tmin = 300.
           Tmax = 2500.
           Pmin = 1.0E-03
           Pmax = 1.0E+02
C
C----------Find NPAR real parameters-----------------------
C
           NPAR = 3
           CALL IPNPAR (LINE, NPAR, ISTR, ISTART)
           CALL IPPARR (ISTR, 1, NPAR, PAR(1), NVAL, IER, LOUT)
           IF (IER .NE. 0) KERR = .TRUE.
C
C-----Remove blanks from reaction string
C
c           INAME = ' '
c           ILEN = 0
c           DO 10 I = 1, ISTART-1
c               IF (LINE(I:I) .NE. ' ') THEN
c                   ILEN = ILEN+1
c                   INAME(ILEN:ILEN) = LINE(I:I)
c               ENDIF
c   10       CONTINUE
           INAME = ' '
           ILEN = 0
           M_PPP = .TRUE.
           DO 10 I = 1, ISTART-1
               IF (LINE(I:I) .NE. '(' .AND. M_PPP ) THEN
                   ILEN = ILEN+1
                   INAME(ILEN:ILEN) = LINE(I:I)
               ELSEIF (LINE(I:I).EQ.'('.OR..NOT.M_PPP) THEN
                   ILEN = ILEN+1
                   INAME(ILEN:ILEN) = ' '
               ENDIF
               IF (LINE(I:I) .EQ. '(') M_PPP = .FALSE.
```

```
            IF (LINE(I:I) .EQ. ')') M_PPP = .TRUE.
   10     CONTINUE
C
C-----Find reaction string, product string
C
          I1 = 0
          I2 = 0
          DO 25 I = 1, ILEN
              IF (I1 .LE. 0) THEN
                  IF (INAME(I:I+2) .EQ. '<=>') THEN
                      I1 = I
                      I2 = I+2
                      IR = 1
                  ELSEIF (INAME(I:I+1) .EQ. '=>') THEN
                      I1 = I
                      I2 = I+1
                      IR = -1
                  ELSEIF (I.GT.1 .AND. INAME(I:I).EQ.'='
     1                .AND. INAME(I-1:I-1).NE.'=') THEN
                      I1 = I
                      I2 = I
                      IR = 1
                  ENDIF
              ENDIF
   25     CONTINUE
C
          IF (ILASCH(INAME).GE.45 .AND. I1.GT.0) THEN
              WRITE (LOUT, 1900) INAME(:I1-1),Afit,rnfit,Eafit*1000.
              WRITE (LOUT, 1920) INAME(I1:)
          ELSE
              WRITE (LOUT, 1900) INAME(:45),Afit,rnfit,Eafit*1000.,
     1                           FILE_IN,AVGERR
          ENDIF
C          IF (T_HIGH .EQ. T_LOW) THEN
C              N_TFIT = 1
C              nK = 1
C          ELSE
              N_TFIT = 21
C              nK = 1
C          ENDIF
          DO 301 NFIT = 1, N_TFIT
              T(NFIT) = T_LOW + (T_HIGH-T_LOW)/(N_TFIT-1)*(NFIT-1)
c              WRITE(*,*)' T(NFIT) = ',T(NFIT)
              T_cheb = (2.*(1./T(NFIT))-(1/Tmin)-(1/Tmax)) /
     2                       ((1/Tmax)-(1/Tmin))
c              WRITE(*,*)' T_cheb = ',t_cheb
              P_cheb = (2.*dlog10(P)-dlog10(Pmin)-
     2                        dlog10(Pmax)) / (dlog10(Pmax)-dlog10(Pmin))
c              WRITE(*,*)' P_cheb = ',P_cheb
              RK_tempLog = 0.0
c to avoid ms fortran compiler significant digits error
              IF (t_cheb .ge. 0.9999999999999) t_cheb = 1.0
              DO 82 I = 1, CHEB(1)
                 DO 81 J = 1, CHEB(2)
c            write(*,*)RK_tempLog,((I-1)*M_cheb+J),X((I-1)*M_cheb+J),
c     2                  PhiCh(I,T_cheb),PhiCh(J,P_cheb)
C                   WRITE(*,*)T_LOW,T_HIGH,NFIT,T(NFIT)
c                   WRITE(*,77)I,J,NFIT,T(NFIT),T_CHEB,P_CHEB
c77                 FORMAT('I=',I2,'J=',I2,'T(',I2,')=',F7.2,' ',2F9.2)
c                     TT = PHICH(J,P_CHEB)
c                     WRITE(*,*)'P_CHEB = ',TT
c                     TT = PHICH(I,T_CHEB)
c                     WRITE(*,*)'T_CHEB = ',TT
                    RK_tempLog = RK_tempLog + CHEB((I-1)*
     2                     CHEB(2)+J+2)*PhiCh(I,T_cheb)*
     3                     PhiCh(J,P_cheb)
```

```
81                CONTINUE
82              CONTINUE
                RK(NFIT) = 10.0**RK_tempLog
301         CONTINUE
            nK = 21
            do 45 I=1,nK
                lnK(I) = dlog(RK(I))
45          continue
            nlow = 1
            nhigh = nK
            WRITE(LOG,*)' '
            WRITE(LOG,*)' '
            IF (ILASCH(INAME).GE.45 .AND. I1.GT.0) THEN
                WRITE (LOG, 1800) INAME(:I1-1)
                WRITE (LOG, 1820) INAME(I1:)
            ELSE
                WRITE (LOG, 1800) INAME(:45)
            ENDIF
            WRITE(LOG,*)' '
            call arrnlin(LOG,0.0d0,nK,3,T,lnK,nlow,nhigh,
     2                  Afit,rnfit,Eafit,errpar)
            avgerr=0.0
            do 300 I = 1,nK
                K_fit = Afit*T(I)**rnfit*dexp(-Eafit*1000./R/T(I))
                error = 100.d0* (rK(I) - K_fit) /
     2            dmin1(rK(I), K_fit)
                avgerr = avgerr + dabs(error)
                WRITE(LOG,277)T(I),rK(I),K_fit,error
277             format(10X,'T = ',f6.1,' K = ',1pe9.2,' K_fit = ',e9.2,
     1                  ' error =',0pf9.2)
300         continue
            avgerr = avgerr/nK
            write(LOG,*)'  '
            write(LOG,320) Afit,rnfit,Eafit,avgerr
320         format(8X,'Fitted  A =',1pe10.3,' n =',0pf6.2,' Ea =',
     2      1pe10.3,' avg error ',0pf7.2,' %')
C
C
C
C
C
C
C
C
C
          ELSE
              IF (LINE3.NE.' ' .AND. M.GT.2) WRITE(LOUT,810)LINE3
          ENDIF
          GOTO 100

5000      IF (LINE3 .NE. ' ') WRITE(LOUT,810) LINE3
          IF (LINE2 .NE. ' ') WRITE(LOUT,810) LINE2
C          WRITE(*,*)'...END'
810       FORMAT(A80)
1800      FORMAT ( A)
1820      FORMAT (6X,A)
1900      FORMAT ( A, T47, 1PE8.2, 2X, 0PF7.3, 1X, F8.0,' ! ',A12,
         1         ' avg err ',f6.2,'%')
1920      FORMAT (6X,A)
2028      FORMAT (6X,'Error in CHEB fall-off parameters...',A)
2090      FORMAT (6X,'Error in auxiliary data...',A)
3098      FORMAT (6X,'CHEB Polynomials:  ',8E13.5)
          CLOSE (LIN)
          CLOSE (LOG)
          CLOSE (LOUT)
          STOP
          END
```

```
C
C
C***PHICH*************************************
C
      FUNCTION PhiCh(i,x)
      implicit none
c     local variables
      integer i
      real*8 x,PhiCh
c
      PhiCh = DCOS((i-1)*DACOS(x))
      return
      end
C
c*ARRNLIN*********************************
      subroutine arrnlin(lerr,dummy1,nData,npara,xin,yin,nlow,nhigh,
     +         Afit,rnfit,Eafit,chiSq)
      implicit none
c     local variables
      integer mxWells, mxProds, npara
      parameter (mxWells = 12, mxProds = 10)
      integer mxTPts, mxPPts
      parameter (mxTPts = 20, mxPPts = 61)
      integer mxColls
      parameter (mxColls = 10)
      integer mxFreqs
      parameter (mxFreqs = 6)
      integer mxFitP
      parameter (mxFitP = 20)
      integer mxKfit
      parameter (mxKFit = 6)
      integer i
      integer lerr,nData,nlow,nhigh
      real*8 xin(mxTpts),yin(mxTpts),Afit,rnfit,Eafit,dummy1,chiSq
      integer listA(3)
      real*8 sig(mxTpts),A(3),covar(3,3)
      external fArrn,fArrn3
c     end declarations
c     nlow and nhigh are part of old autoranging stuff - here, we fix and
c     leave alone
      nlow = 1
      nhigh = Ndata
c     perform some initializing
      do 20 i = 1,nData
         sig(i) = 1.d0
20    continue
c     listA is a vector telling what coefficients to adjust;
c     we want all of them
      do 50 i = 1,npara
         listA(i) = i
50    continue
c     note we added add'l argument lerr to this sub
      if (npara.eq.2) then
      call lfit (xin,yin,sig,nData,A,npara,listA,npara,covar,
     +   npara,chiSq,fArrn,lerr)
      endif
      if (npara.eq.3) then
        call lfit (xin,yin,sig,nData,A,npara,listA,npara,covar,
     +   npara,chiSq,fArrn3,lerr)
      endif
c     set parameters for output
      if (npara.eq.2) then
          Afit = dexp(A(1))
          Eafit = -A(2)*1.987/1000.
      endif
      if (npara.eq.3) then
```

```
            Afit = dexp(A(1))
            rnfit = A(2)
            Eafit = -A(3)*1.987/1000.
         endif
         return
         end
c***FARRN**************************
c     extern basis functions for lsq fit of arrhenius coefficients
         subroutine fArrn(x,VecCoefs,nCoefs)
         implicit none
         integer nCoefs
         real*8 x,VecCoefs(nCoefs)
         VecCoefs(1) = 1.d0
         VecCoefs(2) = 1/x
         return
         end

c***FARRN3*************************
c     extern basis functions for lsq fit of arrhenius coefficients
         subroutine fArrn3(x,VecCoefs,nCoefs)
         implicit none
         integer nCoefs
         real*8 x,VecCoefs(nCoefs)
         VecCoefs(1) = 1.d0
         VecCoefs(2) = dlog(x)
         VecCoefs(3) = 1/x
         return
         end
c this is all matrix solving stuff from press et al.,numerical recipiesc
*****ludcmp*************************************************************
         subroutine ludcmp(A,n,np,indx,d,vv,ierr,lout)
c     this is adaped from numerical recipes by press et al.  <ayc 3/93>
c     this is an lu decomposer and the output is passed to susequent
routines
c     Note: change of inputs - esp array vv so it can be fixed in size
c     from above
c     A is input array of n equations and is physically dimensioned np x np
c     note input matrix A is changed by this routine to give LU decomp'ed A
c     indx is output 1-D array of pivot info - this along with variable D is
c     passed to companion subroutines
c     note: code assumes no-trip do's (fortran 77 standard)
         implicit none
         real*8 tiny
c         parameter(tiny = 1.d-175)
         integer n,np,indx(np),imax,ierr,lout
         integer i,j,k
         real*8 A(np,np),vv(np),d,aamax,sum,dum
         ierr = 0
         tiny=1.d-175
         d = 1.d0
         do 120 i = 1,n
            aamax = 1.d0
               do 110 j = 1,n
               if (dabs(A(i,j)).gt.aamax) aamax = dabs(A(i,j))
110         continue
            if (aamax.le.tiny) then
               write(lout,*) 'ERROR: singular matrix.'
               ierr = 1
               return
            endif
               vv(i) = 1.d0/aamax
120      continue
         do 190 j = 1,n
            do 140 i = 1,j-1
               sum = A(i,j)
                  do 130 k= 1,i-1
```

```
               sum = sum - A(i,k)*A(k,j)
130         continue
            A(i,j) = sum
140      continue
         aamax = 0.d0
         do 160 i = j,n
            sum = A(i,j)
               do 150 k = 1,j-1
               sum = sum - A(i,k)*A(k,j)
150         continue
            A(i,j) = sum
            dum = vv(i)*dabs(sum)
               if (dum.ge.aamax) then
               imax = i
               aamax = dum
            endif
160      continue
         if (j.ne.imax) then
            do 170 k = 1,n
               dum = A(imax,k)
               A(imax,k) = A(j,k)
               A(j,k) = dum
170         continue
            d = -d
            vv(imax) = vv(j)
         endif
                     indx(j) = imax
         if (dabs(A(j,j)).le.tiny) A(j,j) = dsign(tiny,A(j,j))
            if (j.ne.n) then
            dum = 1.d0/A(j,j)
               do 180 i = j+1,n
               A(i,j) = A(i,j)*dum
180         continue
         endif
190   continue
      return
      end
c*****lubksb**********************************************************************
      subroutine lubksb(A,n,np,indx,B)
c     this is taken from numerical recipes by press et al.   <ayc 3/93>
c     A is LU decomp'ed array of n equations, physically dimensioned np x np
c     and was obtained by ludcmp
c     indx is 1-D array of pivot info from lubksb
c     B is input column vector (A X = B) but is changed on output to give X
      implicit none
      integer n,np,indx(np),ii,ll
      integer i,j
      real*8 A(np,np),B(np),sum
      ii = 0
      do 120 i = 1,n
         ll = indx(i)
         sum = B(ll)
         B(ll) = B(i)
            if (ii.ne.0) then
            do 110 j = ii,i-1
                sum = sum - A(i,j)*B(j)
110         continue
         elseif (sum.ne.0.d0) then
            ii = i
         endif
            B(i) = sum
120   continue
      do 140 i = n,1,-1
         sum = B(i)
         do 130 j = i+1,n
            sum = sum - A(i,j)*B(j)
```

```
130       continue
          B(i) = sum/A(i,i)
140    continue
       return
       end
c*****lfit*************************************************************
       subroutine lfit(x,y,sig,ndata,a,ma,lista,mfit,covar,ncvm,
      2    chisq,funcs,lerr)
c     this is taken from numerical recipes by press et al.   <ayc 4/93>
c     its pretty complicated so you'd better see description in book
c     comments in code are mine <ayc>
c     this program allows you to only adjust mfit of ma coefficients,
c     while keeping the rest fixed - lista is an input vector specifying
c     what parameters you want adjusted
       implicit none
c     mmax is max # of coefficients
       integer mmax
       parameter (mmax = 10)
       integer ndata,ma,lista(ma),mfit,ncvm
       integer kk,ihit,i,j,k
       real*8 x(ndata),y(ndata),sig(ndata),a(ma),covar(ncvm,ncvm),
      2    beta(mmax),afunc(mmax),chisq
       real*8 ym,sig2i,wt,sum
       external funcs
c     the following are for ldu routines which we now use
c     instead of gaussj
       integer indx(mmax),ierr,lerr
       real*8 d,vv(mmax)
c     done declarations
       kk = mfit + 1
       do 12 j = 1,ma
       ihit = 0
       do 11 k = 1,mfit
          if (lista(k).eq.j) ihit = ihit + 1
11        continue
          if (ihit.eq.0) then
             lista(kk) = j
             kk = kk + 1
          else if (ihit.gt.1) then
             stop 'improper set in lista'
          endif
12     continue
       if (kk.ne.(ma+1)) stop 'improper set in lista'
          do 14 j = 1,mfit
          do 13 k = 1,mfit
             covar(j,k) = 0.d0
13        continue
          beta(j) = 0.d0
14     continue
       do 18 i = 1,ndata
          call funcs(x(i),afunc,ma)
          ym = y(i)
             if (mfit.lt.ma) then
             do 15 j = mfit+1,ma
                ym = ym - a(lista(j))*afunc(lista(j))
15           continue
          endif
                   sig2i = 1.d0/sig(i)**2
             do 17 j = 1,mfit
             wt = afunc(lista(j))*sig2i
             do 16 k = 1,j
                covar(j,k) = covar(j,k) + wt*afunc(lista(k))
16           continue
             beta(j) = beta(j) + ym*wt
17        continue
18     continue
```

```
      if (mfit.gt.1) then
         do 21 j = 2,mfit
            do 19 k = 1,j-1
               covar(k,j) = covar(j,k)
19          continue
21       continue
      endif
c     we have replaced this with call to ldu solver
c      call gaussj(covar,mfit,ncvm,beta,1,1)
c     we use lud routines instead of gaussj
c     we aren't going to do anything with inverse
c     so we just solve to get solution vector
c     NOTE: we have changed some of the inputs of these
c     routines as compared to book!!
      call ludcmp(covar,mfit,ncvm,indx,d,vv,ierr,lerr)
      call lubksb(covar,mfit,ncvm,indx,beta)
c     this has totally messed up covar but we don't
c     need it anymore - see book to get the 3 lines
c     of code to compute inverse of covar from here
      do 22 j = 1,mfit
         a(lista(j)) = beta(j)
22    continue
      chisq = 0.d0
      do 24 i = 1,ndata
         call funcs(x(i),afunc,ma)
                sum = 0.d0
              do 23 j = 1,ma
            sum = sum + a(j)*afunc(j)
23       continue
         chisq = chisq + ((y(i)-sum)/sig(i))**2
24    continue
c     call covsrt(covar,ncvm,ma,lista,mfit)
c     this sorts covariant matrix to original ordering
c     of fitting coefficients; here we neither changed
c     ordering nor do we care about this matrix so
c     we don't worry about it - note with gaussj,
c     covar would have been the inverse of the original;
c     here its become befuddled - but if we cared we
c     could fix it
      return
      end
C
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKCHAR (SUB, NSUB, NDIM, STRAY, RAY, NN, KERR, LOUT)
C
C     Extracts names and real values from an array of CHAR*(*)
C     substrings; stores names in STRAY array, real values in RAY;
C     i.e. can be used to store element and atomic weight data,
C     species names, etc.
C
C     Input:   SUB(N),N=1,NSUB  - array of CHAR*(*) substrings
C              NSUB             - number of substrings
C              NDIM             - size of STRAY,RAY arrays
C              NN               - actual number of STRAY found
C              STRAY(N),N=1,NN  - CHAR*(*) array
C              RAY(N),N=1,NN    - Real array
C              LOUT             - output unit for error messages
C     Output:  NN               - incremented if more STRAY found
C              STRAY(N),N=1,NN  - incremented array of STRAY
C              RAY(N),N=1,NN    - incremented array of reals
C              KERR             - logical, .TRUE. = error in data
C
C                                      F. Rupley, Div. 8245, 2/5/88
C-----------------------------------------------------------------------C
C*****precision > double
```

```
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION RAY(*), PAR(1)
      CHARACTER SUB(*)*(*), STRAY(*)*(*), ISTR*80, UPCASE*4
      LOGICAL KERR
C
      ILEN = LEN(STRAY(1))
C
      DO 200 N = 1, NSUB
         IF ( UPCASE(SUB(N), 3) .EQ. 'END') RETURN
         ISTR = ' '
         I1 = INDEX(SUB(N),'/')
         IF (I1 .EQ .1) THEN
            KERR = .TRUE.
            WRITE (LOUT, 130) SUB(N)(:ILASCH(SUB(N)))
         ELSE
            IF (I1 .LE. 0) THEN
               ISTR = SUB(N)
            ELSE
               ISTR = SUB(N)(:I1-1)
            ENDIF
            CALL CKCOMP (ISTR, STRAY, NN, INUM)
C
            IF (INUM .GT. 0) THEN
               WRITE (LOUT, 100) SUB(N)(:ILASCH(SUB(N)))
            ELSE
               IF (NN .LT. NDIM) THEN
                  IF (ISTR(ILEN+1:) .NE. ' ') THEN
                     WRITE (LOUT, 120) SUB(N)(:ILASCH(SUB(N)))
                     KERR = .TRUE.
                  ELSE
                     NN = NN + 1
                     STRAY(NN) = ' '
                     STRAY(NN) = ISTR(:ILEN)
                     IF (I1 .GT. 0) THEN
                        I2 = I1 + INDEX(SUB(N)(I1+1:),'/')
                        ISTR = ' '
                        ISTR = SUB(N)(I1+1:I2-1)
                        CALL IPPARR (ISTR, 1, 1, PAR, NVAL, IER, LOUT)
                        IF (IER .EQ. 0) THEN
                           RAY(NN) = PAR(1)
                        ELSE
                           KERR = .TRUE.
                        ENDIF
                     ENDIF
                  ENDIF
               ELSE
                  WRITE (LOUT, 110) SUB(N)(:ILASCH(SUB(N)))
                  KERR = .TRUE.
               ENDIF
            ENDIF
         ENDIF
  200 CONTINUE
C
  100 FORMAT (6X,'Warning...duplicate array element ignored...',A)
  110 FORMAT (6X,'Error...character array size too small for  ...',A)
  120 FORMAT (6X,'Error...character array element name too long...',A)
  130 FORMAT (6X,'Error...misplaced value...',A)
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKAWTM (ENAME, AWT)
C
```

```
C     Returns atomic weight of element ENAME.
C     Input:   ENAME - CHAR*(*) element name
C     Output:  AWT   - real atomic weight
C
C                                        F. Rupley, Div. 8245, 11/11/86
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C     IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      PARAMETER (NATOM = 102)
      DIMENSION ATOM(NATOM)
      CHARACTER ENAME*(*), IATOM(NATOM)*2, UPCASE*2
C
      DATA (IATOM(I),ATOM(I),I=1,40) /
     *'H ',  1.00797, 'HE',  4.00260, 'LI',  6.93900, 'BE',  9.01220,
     *'B ', 10.81100, 'C ', 12.01115, 'N ', 14.00670, 'O ', 15.99940,
     *'F ', 18.99840, 'NE', 20.18300, 'NA', 22.98980, 'MG', 24.31200,
     *'AL', 26.98150, 'SI', 28.08600, 'P ', 30.97380, 'S ', 32.06400,
     *'CL', 35.45300, 'AR', 39.94800, 'K ', 39.10200, 'CA', 40.08000,
     *'SC', 44.95600, 'TI', 47.90000, 'V ', 50.94200, 'CR', 51.99600,
     *'MN', 54.93800, 'FE', 55.84700, 'CO', 58.93320, 'NI', 58.71000,
     *'CU', 63.54000, 'ZN', 65.37000, 'GA', 69.72000, 'GE', 72.59000,
     *'AS', 74.92160, 'SE', 78.96000, 'BR', 79.90090, 'KR', 83.80000,
     *'RB', 85.47000, 'SR', 87.62000, 'Y ', 88.90500, 'ZR', 91.22000/
C
      DATA (IATOM(I),ATOM(I),I=41,80) /
     *'NB', 92.90600, 'MO', 95.94000, 'TC', 99.00000, 'RU',101.07000,
     *'RH',102.90500, 'PD',106.40000, 'AG',107.87000, 'CD',112.40000,
     *'IN',114.82000, 'SN',118.69000, 'SB',121.75000, 'TE',127.60000,
     *'I ',126.90440, 'XE',131.30000, 'CS',132.90500, 'BA',137.34000,
     *'LA',138.91000, 'CE',140.12000, 'PR',140.90700, 'ND',144.24000,
     *'PM',145.00000, 'SM',150.35000, 'EU',151.96000, 'GD',157.25000,
     *'TB',158.92400, 'DY',162.50000, 'HO',164.93000, 'ER',167.26000,
     *'TM',168.93400, 'YB',173.04000, 'LU',174.99700, 'HF',178.49000,
     *'TA',180.94800, 'W ',183.85000, 'RE',186.20000, 'OS',190.20000,
     *'IR',192.20000, 'PT',195.09000, 'AU',196.96700, 'HG',200.59000/
C
      DATA (IATOM(I),ATOM(I),I=81,NATOM) /
     *'TL',204.37000, 'PB',207.19000, 'BI',208.98000, 'PO',210.00000,
     *'AT',210.00000, 'RN',222.00000, 'FR',223.00000, 'RA',226.00000,
     *'AC',227.00000, 'TH',232.03800, 'PA',231.00000, 'U ',238.03000,
     *'NP',237.00000, 'PU',242.00000, 'AM',243.00000, 'CM',247.00000,
     *'BK',249.00000, 'CF',251.00000, 'ES',254.00000, 'FM',253.00000,
     *'D ',002.01410, 'E',5.45E-4/
C
      CALL CKCOMP ( UPCASE(ENAME, 2), IATOM, NATOM, L)
      IF (L .GT. 0) AWT = ATOM(L)
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKTHRM (LUNIT, MDIM, ENAME, MM, AWT, KNAME, KK, KNCF,
     1                   KPHSE, KCHRG, WTM, MAXTP, NT, NTR, TLO, TMID,
     2                   THI, T, NPCP2, A, ITHRM, KERR, LOUT, ISTR)
C
C     Finds thermodynamic data and elemental composition for species
C     Input:  LUNIT  - unit number for input of thermo properties
C             MDIM   - maximum number of elements allowed
C             ENAME(M),M=1,MM  - array of CHAR*(*) element names
C             MM     - total number of elements declared
C             AWT(M),M=1,MM    - array of atomic weights for elements
C             KNAME(K),K=1,KK  - array of CHAR*(*) species names
C             KK     - total number of species declared
```

```
C             LOUT   - output unit for messages
C             NT(K),K=1,KK - number of temperature values
C             NTR - number of temperature ranges
C     Output: KNCF(M,K) - elemental composition of species
C             KPHSE(K),K=1,KK - integer array, species phase
C             KCHRG(K),K=1,KK - integer array of species charge
C                      =0, if no electrons,
C                      =(-1)*number of electrons present
C             WTM(K),K=1,KK - array of molecular weights of species
C             A(M,L,K)- array of thermodynamic coefficients
C             T(N),N=1,NT - array of temperatures
C             KERR   - logical error flag
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION WTM(*), NT(*), T(MAXTP,*), KPHSE(*), KNCF(MDIM,*),
     1          KCHRG(*), A(NPCP2,NTR,*), AWT(*), VALUE(5)
      CHARACTER ENAME(*)*(*), KNAME(*)*(*), LINE(4)*80, ELEM*16
      CHARACTER UPCASE*4, ISTR*80, SUB(80)*80
      LOGICAL KERR, ITHRM(*)
C
      IF (MM.LE.0 .OR. KK.LE.0) WRITE (LOUT, 80)
C
      GO TO 20
   10 CONTINUE
      ISTR = ' '
      READ (LUNIT,'(A)',END=40) ISTR
   20 CONTINUE
      ILEN = IPPLEN(ISTR)
      IF (ILEN .LE. 0) GO TO 10
C
      CALL CKISUB (ISTR(:ILEN), SUB, NSUB)
      CALL CKCOMP (SUB(1), KNAME, KK, K)
      IF (K .EQ. 0) THEN
         IF (UPCASE(SUB(1), 3) .EQ. 'END' .OR.
     1       UPCASE(SUB(1), 4) .EQ. 'REAC') RETURN
         GO TO 10
      ENDIF
C
      IF (ITHRM(K)) GO TO 10
      ITHRM(K) = .TRUE.
      LINE(1) = ' '
      LINE(1) = ISTR
      DO 25 L = 2, 4
         LINE(L) = ' '
         READ (LUNIT,'(A)',END=40) LINE(L)
   25 CONTINUE
C
      ICOL = 20
      DO 60 I = 1, 5
         ICOL = ICOL + 5
         IF (I .EQ. 5) ICOL = 74
         ELEM  = LINE(1)(ICOL:ICOL+1)
         IELEM = 0
C
         IF (LINE(1)(ICOL+2:ICOL+4) .NE. ' ') THEN
            CALL IPPARR
     1      (LINE(1)(ICOL+2:ICOL+4), 0, 1, VALUE, NVAL, IER, LOUT)
            IELEM = VALUE(1)
         ENDIF
C
```

```
            IF (ELEM.NE.' ' .AND. IELEM.NE.0) THEN
               IF (UPCASE(ELEM, 1) .EQ. 'E')
     1            KCHRG(K)=KCHRG(K)+IELEM*(-1)
               CALL CKCOMP (ELEM, ENAME, MM, M)
               IF (M .GT. 0) THEN
                  KNCF(M,K) = IELEM
                  WTM(K) = WTM(K) + AWT(M)*FLOAT(IELEM)
               ELSE
                  WRITE (LOUT, 100) ELEM,KNAME(K)(:10)
                  KERR = .TRUE.
               ENDIF
            ENDIF
   60 CONTINUE
C
      IF (UPCASE(LINE(1)(45:),1) .EQ. 'L') KPHSE(K)=1
      IF (UPCASE(LINE(1)(45:),1) .EQ. 'S') KPHSE(K)=-1
C
C-----Currently allows for three temperatures, two ranges;
C     in future, NT(K) may vary, NTR = NT(K)-1
C
      T(1,K) = TLO
      IF (LINE(1)(46:55) .NE. ' ') CALL IPPARR
     1   (LINE(1)(46:55), 0, 1, T(1,K), NVAL, IER, LOUT)
C
      T(2,K) = TMID
      IF (LINE(1)(66:73) .NE. ' ') CALL IPPARR
     1   (LINE(1)(66:73), 0, 1, T(2,K), NVAL, IER, LOUT)
C
      T(NT(K),K) = THI
      IF (LINE(1)(56:65) .NE. ' ') CALL IPPARR
     1   (LINE(1)(56:65), 0, 1, T(NT(K),K), NVAL, IER, LOUT)
C
      READ (LINE(2)(:75),'(5E15.8)') (A(I,NTR,K),I=1,5)
      READ (LINE(3)(:75),'(5E15.8)')
     1            (A(I,NTR,K),I=6,7),(A(I,1,K),I=1,3)
      READ (LINE(4)(:60),'(4E15.8)') (A(I,1,K),I=4,7)
      GO TO 10
C
   40 RETURN
   80 FORMAT (6X,'Warning...THERMO cards misplaced will be ignored...')
  100 FORMAT (6X,'Error...element...',A,'not declared for...',A)
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKREAC (LINE, II, KK, KNAME, LOUT, MAXSP, NSPEC, NREAC,
     1                   NUNK, NU, NPAR, PAR, NTHB, ITHB,
     2                   NFAL, IFAL, KFAL, NWL, IWL, WL, KERR)
C
C     CKREAC parses the main CHAR*(*) line representing a gas-phase
C     reaction; first, the real Arrhenius parameters are located and
C     stored in PAR(N,I),N=1,NPAR, where I is the reaction number;
C     then a search is made over the reaction string:
C
C     '=','<=>': reaction I is reversible;
C     '=>'     : reaction I is irreversible;
C
C     '(+[n]KNAME(K))': reaction I is a fall-off reaction;
C                       NFAL is incremented, the total number of
C                       fall-off reactions;
C                       IFAL(NFAL)=I, KFAL(NFAL)=K;
C                       this species is eliminated from consideration
C                       as a reactant or product in this reaction.
C
C     '(+M)'   : reaction I is a fall-off reaction;
C                NFAL is incremented, IFAL(NFAL)=I, KFAL(NFAL)=0;
C
C     '+[n]KNAME(K)': NSPEC(I) is incremented, the total number of
```

```
C                    species for this reaction;
C                    n is an optional stoichiometric coefficient
C                    of KNAME(K), if omitted, n=1;
C                    if this string occurs before the =/-,
C                    NREAC(I) is incremented, the total number of
C                    reactants for this reaction, NUNK(N,I)=K, and
C                    NU(N,I) = -n, where N=1-3 is reserved for
C                    reactants;
C                    if this string occurs after the =/-,
C                    NUNK(N,I) = K, and NU(N,I) = n, where N=4-6
C                    is reserved for products;
C
C     '+M' : I is a third-body reaction; NTHB is incremented, the
C            total number of third-body reactions, and ITHB(NTHB)=I.
C
C     Input:  LINE  - a CHAR*(*) line (from data file)
C             II    - the index of this reaction, and the total number
C                     of reactions found so far.
C             KK    - actual integer number of species
C             KNAME(K),K=1,KK - array of CHAR*(*) species names
C             LOUT  - output unit for error messages
C             MAXSP - maximum number of species allowed in reaction
C             NPAR  - number of parameters expected
C     A '!' will comment out a line, or remainder of the line.
C
C     Output: NSPEC - total number of reactants+products in reaction
C             NREAC - number of reactants
C             NUNK  - species numbers for the NSPEC species
C             NU    - stoichiometric coefficients for the NSPEC spec.
C             NFAL  - total number of fall-off reactions
C             IFAL  - reaction numbers for the NFAL reactions
C             KFAL  - 3rd body species numbers for the NFAL reactions
C             NTHB  - total number of 3rd-body reactions
C             ITHB  - reaction numbers for the NTHB reactions
C             NWL   - number of radiation-enhanced reactions
C             IWL   - reaction numbers for the NWL reactions
C             WL    - radiation wavelengths for the NWL reactions
C             KERR  - logical, .TRUE. = error in data file
C
C                                        F. Rupley, Div. 8245, 5/13/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), NREAC(*), NUNK(MAXSP,*), NU(MAXSP,*),
     1          PAR(NPAR,*), IFAL(*), KFAL(*), ITHB(*), IWL(*), WL(*),
     2          IPLUS(20)
      CHARACTER KNAME(*)*(*), LINE*(*), INUM(10)*1, UPCASE*4
      CHARACTER*80 ISTR, IREAC, IPROD, ISPEC, INAME, ITEMP
      LOGICAL KERR, LTHB, LWL
      DATA INUM/'0','1','2','3','4','5','6','7','8','9'/
C
      LTHB = .FALSE.
      LWL = .FALSE.
      NSPEC(II) = 0
      NREAC(II) = 0
C
C----------Find NPAR real parameters-----------------------
C
      CALL IPNPAR (LINE, NPAR, ISTR, ISTART)
      CALL IPPARR (ISTR, 1, NPAR, PAR(1,II), NVAL, IER, LOUT)
      IF (IER .NE. 0) KERR = .TRUE.
```

```
C
C-----Remove blanks from reaction string
C
      INAME = ' '
      ILEN = 0
      DO 10 I = 1, ISTART-1
         IF (LINE(I:I) .NE. ' ') THEN
            ILEN = ILEN+1
            INAME(ILEN:ILEN) = LINE(I:I)
         ENDIF
   10 CONTINUE
C
C-----Find reaction string, product string
C
      I1 = 0
      I2 = 0
      DO 25 I = 1, ILEN
         IF (I1 .LE. 0) THEN
            IF (INAME(I:I+2) .EQ. '<=>') THEN
               I1 = I
               I2 = I+2
               IR = 1
            ELSEIF (INAME(I:I+1) .EQ. '=>') THEN
               I1 = I
               I2 = I+1
               IR = -1
            ELSEIF (I.GT.1 .AND. INAME(I:I).EQ.'='
     1                     .AND. INAME(I-1:I-1).NE.'=') THEN
               I1 = I
               I2 = I
               IR = 1
            ENDIF
         ENDIF
   25 CONTINUE
C
      IF (ILASCH(INAME).GE.45 .AND. I1.GT.0) THEN
         WRITE (LOUT, 1900) II,INAME(:I1-1),(PAR(N,II),N=1,NPAR)
         WRITE (LOUT, 1920) INAME(I1:)
      ELSE
          WRITE (LOUT, 1900) II,INAME(:45),(PAR(N,II),N=1,NPAR)
      ENDIF
C
      IREAC = ' '
      IPROD = ' '
      IF (I1 .GT. 0) THEN
         IREAC = INAME(:I1-1)
         IPROD = INAME(I2+1:)
      ELSE
C
C-----did not find delimiter
C
         WRITE (LOUT, 660)
         KERR = .TRUE.
         RETURN
      ENDIF
C
      IF (INDEX(IREAC,'=>').GT.0 .OR. INDEX(IPROD,'=>').GT.0) THEN
C
C-----more than one '=>'
C
         WRITE (LOUT, 800)
         KERR = .TRUE.
         RETURN
      ENDIF
C
C-----Is this a fall-off reaction?
```

```
C
      IF (INDEX(IREAC,'(+').GT.0 .OR. INDEX(IPROD,'(+').GT.0) THEN
         KRTB = 0
         KPTB = 0
         DO 300 J = 1, 2
            ISTR = ' '
            KTB  = 0
            IF (J .EQ. 1) THEN
               ISTR = IREAC
            ELSE
               ISTR = IPROD
            ENDIF
C
            DO 35 N = 1, ILASCH(ISTR)-1
               IF (ISTR(N:N+1) .EQ. '(+') THEN
                  I1 = N+2
                  I2 = I1 + INDEX(ISTR(I1:),')')-1
                  IF (I2 .GT. I1) THEN
                     IF (ISTR(I1:I2-1).EQ.'M' .OR.
     1                   ISTR(I1:I2-1).EQ.'m') THEN
                        IF (KTB .NE. 0) THEN
                           WRITE (LOUT, 630)
                           KERR = .TRUE.
                           RETURN
                        ELSE
                           KTB = -1
                        ENDIF
                     ELSE
                        CALL CKCOMP (ISTR(I1:I2-1), KNAME, KK, KNUM)
                        IF (KNUM .GT. 0) THEN
                           IF (KTB .NE. 0) THEN
                              WRITE (LOUT, 630)
                              KERR = .TRUE.
                              RETURN
                           ELSE
                              KTB = KNUM
                           ENDIF
                        ENDIF
                     ENDIF
                     IF (KTB .NE. 0) THEN
                        ITEMP = ' '
                        IF (I1 .EQ. 1) THEN
                           ITEMP = ISTR(I2+1:)
                        ELSE
                           ITEMP = ISTR(:I1-3)//ISTR(I2+1:)
                        ENDIF
                        IF (J .EQ. 1) THEN
                           IREAC = ' '
                           IREAC = ITEMP
                           KRTB = KTB
                        ELSE
                           IPROD = ' '
                           IPROD = ITEMP
                           KPTB = KTB
                        ENDIF
                     ENDIF
                  ENDIF
               ENDIF
   35       CONTINUE
  300    CONTINUE
C
         IF (KRTB.NE.0 .OR. KPTB.NE.0) THEN
C
C           does product third-body match reactant third-body
C
            IF (KRTB.LE.0 .AND. KPTB.LE.0) THEN
```

```
C
                NFAL = NFAL + 1
                IFAL(NFAL) = II
                KFAL(NFAL) = 0
C
                LTHB = .TRUE.
                NTHB = NTHB + 1
                ITHB(NTHB) = II
C
             ELSEIF (KRTB .EQ. KPTB) THEN
                NFAL = NFAL + 1
                IFAL(NFAL) = II
                KFAL(NFAL) = KRTB
C
             ELSE
C
                WRITE (LOUT, 640)
                KERR = .TRUE.
                RETURN
             ENDIF
          ENDIF
       ENDIF
C
C----------Find reactants, products-------------------------
C
       DO 600 J = 1, 2
          ISTR = ' '
          LTHB = .FALSE.
          IF (J .EQ. 1) THEN
             ISTR = IREAC
             NS = 0
          ELSE
             ISTR = IPROD
             NS = 3
          ENDIF
C
C-----------store pointers to '+'-signs
C
          NPLUS = 1
          IPLUS(NPLUS) = 0
          DO 500 L = 2, ILASCH(ISTR)-1
             IF (ISTR(L:L).EQ.'+') THEN
                NPLUS = NPLUS + 1
                IPLUS(NPLUS) = L
             ENDIF
  500     CONTINUE
          NPLUS = NPLUS + 1
          IPLUS(NPLUS) = ILASCH(ISTR)+1
C
          NSTART = 1
  505     CONTINUE
          N1 = NSTART
          DO 510 N = NPLUS, N1, -1
             ISPEC = ' '
             ISPEC = ISTR(IPLUS(N1)+1 : IPLUS(N)-1)
C
             IF (UPCASE(ISPEC, 1) .EQ. 'M') THEN
                IF (LTHB) THEN
                   WRITE (LOUT, 900)
                   KERR = .TRUE.
                   RETURN
                ELSEIF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) THEN
                   WRITE (LOUT, 640)
                   KERR = .TRUE.
                   RETURN
                ELSE
```

```
                  LTHB = .TRUE.
                  IF (NTHB.EQ.0 .OR.
     1               (NTHB.GT.0.AND.ITHB(NTHB).NE.II)) THEN
                     NTHB = NTHB + 1
                     ITHB(NTHB) = II
                  ENDIF
                  IF (N .EQ. NPLUS) GO TO 600
                  NSTART = N
                  GO TO 505
               ENDIF
C
            ELSEIF (UPCASE(ISPEC, 2) .EQ. 'HV') THEN
               IF (LWL) THEN
                  WRITE (LOUT, 670)
                  KERR = .TRUE.
                  RETURN
               ELSE
                  LWL = .TRUE.
                  NWL = NWL + 1
                  IWL(NWL) = II
                  WL(NWL) = 1.0
                  IF (J .EQ. 1) WL(NWL) = -1.0
                  IF (N .EQ. NPLUS) GO TO 600
                  NSTART = N
                  GO TO 505
               ENDIF
            ENDIF
C
C-----------does this string start with a one- or two-digit number?
C
            IVAL = 0
            CALL CKCOMP (ISPEC(1:1), INUM, 10, I1)
            CALL CKCOMP (ISPEC(2:2), INUM, 10, I2)
            IF (I1 .GT. 0) THEN
               ITEMP = ' '
               IF (I2 .GT. 0) THEN
                  IVAL = 10*(I1-1) + (I2-1)
                  ITEMP = ISPEC(3:)
               ELSE
                  IVAL = I1-1
                  ITEMP = ISPEC(2:)
               ENDIF
               ISPEC = ' '
               ISPEC = ITEMP
            ENDIF
C
            CALL CKCOMP (ISPEC, KNAME, KK, KNUM)
            IF (KNUM .EQ. 0) THEN
               IF ((N-N1) .GT. 1) GO TO 510
               WRITE (LOUT, 680) ISPEC(:ILASCH(ISPEC))
               KERR = .TRUE.
            ELSE
C
C--------------a species has been found
C
               IVAL = MAX(IVAL,1)
               IF (J .EQ. 1) IVAL = -IVAL
               NNUM = 0
               DO 111 K = 1, NS
                  IF (KNUM.EQ.NUNK(K,II) .AND.
     1                NU(K,II)/IVAL.GT.0) THEN
C
C-----------------increment species coefficient count
C
                     NNUM=K
                     NU(NNUM,II) = NU(NNUM,II) + IVAL
```

```
                ENDIF
  111         CONTINUE
              IF (NNUM .LE. 0) THEN
C
C-----------------are there too many species?
C
                 IF (J.EQ.1 .AND. NS.EQ.3) THEN
                    WRITE (LOUT, 690)
                    KERR = .TRUE.
                    RETURN
                 ELSEIF (J.EQ.2 .AND. NS.EQ.MAXSP) THEN
                    WRITE (LOUT, 700)
                    KERR = .TRUE.
                    RETURN
                 ELSE
C
C--------------------increment species count
C
                    NS = NS + 1
                    NSPEC(II) = NSPEC(II)+1
                    IF (J .EQ. 1) NREAC(II) = NS
                    NUNK(NS,II) = KNUM
                    NU(NS,II)   = IVAL
                 ENDIF
              ENDIF
           ENDIF
           IF (N .EQ. NPLUS) GO TO 600
           NSTART = N
           GO TO 505
C
  510    CONTINUE
  600 CONTINUE
C
      NSPEC(II) = IR*NSPEC(II)
C
  630 FORMAT (6X,'Error...more than one fall-off declaration...')
  640 FORMAT (6X,'Error in fall-off declaration...')
  650 FORMAT (6X,'Error...reaction string not found...')
  660 FORMAT (6X,'Error in reaction...')
  670 FORMAT (6X,'Error in HV declaration...')
  680 FORMAT (6X,'Error...undeclared species...',A)
  690 FORMAT (6X,'Error...more than 3 reactants...')
  700 FORMAT (6X,'Error...more than 3 products...')
  800 FORMAT (6X,'Error in reaction delimiter...')
  900 FORMAT (6X,'Error in third-body declaration...')
C 1900 FORMAT (I4,'. ',A,T51,E10.3,F7.3,F11.3)
 1900 FORMAT (I4,'. ', A, T53, 1PE8.2, 2X, 0PF5.1, 2X, F9.1)
 1920 FORMAT (6X,A)
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKAUXL (SUB, NSUB, II, KK, KNAME, LOUT, MAXSP, NPAR,
     1                   NSPEC, NTHB, ITHB, NTBS, MAXTB, NKTB, AIK,
     2                   NFAL, IFAL, IDUP, NFAR, PFAL, IFOP, KFAL, NLAN,
     3                   ILAN, NLAR, PLAN, NREV, IREV, RPAR,
     4                   NRLT, IRLT, RLAN, NWL, IWL, WL, KERR)
C
C     CKAUXL parses the auxiliary CHAR*(*) lines representing
C     additional options for a gas-phase reaction; data is stored
C     based on finding a 'keyword' followed by its required
C     parameters:
C
C     KNAME(K)/val1/: this is an enhanced third-body;
C
C        if ITHB(NTHB) <> I, this is an error, reaction I is not a
C                            third-body reaction;
```

```
C        else NTBS(NTHB) is incremented,
C             AIK(NTBS(NTHB),NTHB) = K,
C             NKTB(NTBS(NTHB)),NTHB) = val1;
C
C     (LOW,TROE, and SRI define fall-off data):
C
C     LOW/val1 val2 val3/: PFAL(N,NFAL) = val(N),N=1,3;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
C                         parameters;
C        if IFOP(NFAL)>0, this is an error, LOW already declared;
C        else
C           IFOP(NFAL) = ABS(IFOP(NFAL))
C
C     TROE/val1 val2 val3 [val4]/: PFAL(N,NFAL) = val(N),N=4,7;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
C                         parameters;
C        if ABS(IFOP(NFAL)).GT.1, this is an error,
C        else
C        if 3 TROE values, IFOP(NFAL) = 3*IFOP(NFAL);
C        if 4 TROE values, IFOP(NFAL) = 4*IFOP(NFAL);
C
C     SRI/val1 val2 val3/: PFAL(N,NFAL) = val(N),N=4,6;
C
C        if IFAL(NFAL)<>I, this is an error, reaction I is not a
C                          fall-off reaction;
C        if ILAN(NLAN)=I, this is an error, cannot have T-L numbers.
C        if IRLT(NRLT)=I, this is an error,          "
C        if IREV(NREV)=I, this is an error, cannot declare reverse
C                         parameters;
C        if ABS(IFOP(NFAL))>1, this is an error;
C        else
C        if IFOP(NFAL)= 2*IFOP(NFAL);
C
C     LT/val1 val2/:
C        if IFAL(NFAL)=I, this is an error, cannot have fall-off and
C                         T-L numbers;
C        else increment NLAN, the number of T-L reactions,
C             ILAN(NLAN)=I, PLAN(N,NLAN)=val(N),N=1,2
C        if IREV(NREV)=I, need IRLT(NRLT)=I.
C
C     REV[ERSE]/val1 val2 val3/ :
C        if IFAL(NFAL)=I, this is an error;
C        if IREV(NREV)=I, this is an error, REV already declared;
C        if NSPEC(I)<0, this an error, as I is irreversible;
C        else increment NREV, the number of reactions with reverse
C             parameters given,
C             IREV(NREV)=I, RPAR(N,NREV)=val(N),N=1,3;
C             if ILAN(NLAN)=I, need IRLT(NRLT)=I;
C             if IRLT(NRLT)=I, need ILAN(NRLT)=I.
C
C     RLT/val1 val2/:
C       if IFAL(NFAL)=I, this is an error, cannot have fall-off and
C                        T-L numbers;
C       if IRLT(NRLT)=I, this is an error, RLT already declared;
C       else increment NRLT, the number of reactions with BOTH
C                      reverse parameters given, and T-L numbers;
```

```
C            IRLT(NRLT)=I, RLAN(N,NRLT)=val(N),N=1,2;
C            if IREV(NREV)<>I, need IREV(NREV)=I;
C            if ILAN(NREV)<>I, need ILAN(NLAN)=I;
C
C     DUP[LICATE]:
C        This reaction is allowed to be duplicated.
C
C      Input:  LINE - CHAR*(*) auxiliary information string
C              KK   - total number of species declared
C              KNAME- CHAR*(*) species names
C              LOUT - output unit for error messages
C              MAXSP- maximum third bodies allowed in a reaction
C      Output: NTHB - total number of reactions with third bodies
C              ITHB - array of third-body reaction numbers
C              AIK  - non-zero third body enhancement factors
C              NKTB - array of species numbers for the third body
C                          enchancement factors
C              NFAL - total number of fall-off reactions
C              IFAL - array of fall-off reaction numbers
C              IFOP - array of fall-off type
C              PFAL - fall-off parameters
C              NLAN - total number of Landau-Teller reactions
C              ILAN - array of T-L reaction numbers
C              NLAR - number of Landau-Teller numbers allowed
C              PLAN - array of Landau-Teller numbers
C              NRLT - total number of 'reverse' T-L reactions
C              IRLT - array of 'reverse' T-L reaction numbers
C              RLAN - array of 'reverse' Landau-Teller numbers
C              NWL  - total number of radiation-enhanced reactions
C              IWL  - array of radiation-enhanced reaction numbers
C              WL   - array of wavelengths
C              KERR - logical, = .TRUE. if error found
C                                          F. Rupley, Div. 8245, 5/27/87
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), ITHB(*), NTBS(*), NKTB(MAXTB,*), IDUP(*),
     1          AIK(MAXTB,*), IFAL(*), KFAL(*), IFOP(*), PFAL(NFAR,*),
     2          ILAN(*), PLAN(NLAR,*), IREV(*), RPAR(NPAR,*), IRLT(*),
     3          RLAN(NLAR,*), IWL(*), WL(*), VAL(1)
      CHARACTER SUB(*)*(*), KNAME(*)*(*), KEY*80, RSTR*80, UPCASE*4
      CHARACTER SUB_CH (80)*80,RSTR_CH*80,LINE_CH*80
      LOGICAL KERR, LLAN, LRLT, LTHB, LFAL, LTRO, LSRI, LWL, LREV
C
      LTHB = (NTHB.GT.0 .AND. ITHB(NTHB).EQ.II)
      LFAL = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II)
      LWL  = (NWL .GT.0 .AND. IWL(NWL)  .EQ.II)
      LREV = (NREV.GT.0 .AND. IREV(NREV).EQ.II)
      LLAN = (NLAN.GT.0 .AND. ILAN(NLAN).EQ.II)
      LRLT = (NRLT.GT.0 .AND. IRLT(NRLT).EQ.II)
      LTRO = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).GT.2)
      LSRI = (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).EQ.2)
C
      DO 500 N = 1, NSUB
         ILEN = ILASCH(SUB(N))
         KEY = ' '
         IF ( UPCASE(SUB(N), 3) .EQ. 'DUP') THEN
            IDUP(II) = -1
            WRITE (LOUT, 4000)
            GO TO 500
         ELSE
```

```
            I1 = INDEX(SUB(N),'/')
            I2 = INDEX(SUB(N)(I1+1:),'/')
            IF (I1.LE.0 .OR. I2.LE.0) THEN
               KERR = .TRUE.
               WRITE (LOUT, 2090) SUB(N)(:ILEN)
               GO TO 500
            ENDIF
            KEY = SUB(N)(:I1-1)
            RSTR = ' '
            RSTR = SUB(N)(I1+1:I1+I2-1)
         ENDIF
C    **************W.ING 6/16/95************************
         IF (UPCASE(KEY, 3).EQ.'LOW' .OR.
     1       UPCASE(KEY, 4).EQ.'TROE'.OR.
     2       UPCASE(KEY, 4).EQ.'EXTR'.OR.
     3       UPCASE(KEY, 4).EQ.'CHEB'.OR.
     4       UPCASE(KEY, 3).EQ.'SRI') THEN
C
C        FALL-OFF DATA
C
            IF ((.NOT.LFAL) .OR. LLAN .OR. LRLT .OR. LREV) THEN
               KERR = .TRUE.
               IF (.NOT. LFAL) WRITE (LOUT, 1050) SUB(N)(:ILEN)
               IF (LLAN)       WRITE (LOUT, 1060) SUB(N)(:ILEN)
               IF (LRLT)       WRITE (LOUT, 1070) SUB(N)(:ILEN)
               IF (LREV)       WRITE (LOUT, 1090) SUB(N)(:ILEN)
            ELSE
C
               IF (UPCASE(KEY, 3) .EQ. 'LOW') THEN
                  IF (IFOP(NFAL) .GT. 0) THEN
                     WRITE (LOUT, 2000) SUB(N)(:ILEN)
                     KERR = .TRUE.
                  ELSE
                     IFOP(NFAL) = ABS(IFOP(NFAL))
                     CALL IPPARR (RSTR,1,3,PFAL(1,NFAL),NVAL,IER,LOUT)
                     IF (IER .NE. 0) KERR = .TRUE.
                     WRITE (LOUT, 3050) (PFAL(L,NFAL),L=1,3)
                  ENDIF
C
               ELSEIF (UPCASE(KEY, 4) .EQ. 'TROE') THEN
                  IF (LTRO .OR. LSRI) THEN
                     KERR = .TRUE.
                     IF (LTRO) WRITE (LOUT, 2010) SUB(N)(:ILEN)
                     IF (LSRI) WRITE (LOUT, 2030) SUB(N)(:ILEN)
                  ELSE
                     LTRO = .TRUE.
                     CALL IPPARR (RSTR,1,-4,PFAL(4,NFAL),NVAL,IER,LOUT)
                     IF (NVAL .EQ. 3) THEN
                        IFOP(NFAL) = 3*IFOP(NFAL)
                        WRITE (LOUT, 3080) (PFAL(L,NFAL),L=4,6)
                     ELSEIF (NVAL .EQ. 4) THEN
                        IFOP(NFAL) = 4*IFOP(NFAL)
                        WRITE (LOUT, 3090) (PFAL(L,NFAL),L=4,7)
                     ELSE
                        WRITE (LOUT, 2020) SUB(N)(:ILEN)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
C
C    **************W.ING 8/10/95************************
               ELSEIF (UPCASE(KEY, 4) .EQ. 'CHEB') THEN
C                    LTRO = .TRUE.
                     CALL IPPARR (RSTR,1,-8,PFAL(4,NFAL),NVAL,IER,LOUT)
                     N_CHEB = PFAL(4,NFAL)
                     M_CHEB = PFAL(5,NFAL)
                     NPARA_CHEB = N_CHEB*M_CHEB + 2
```

```
                    INDEX_CHEB = 0
                    IFOP(NFAL) = 6*IFOP(NFAL)
c                    write(*,*)'..NFAL..IFOP...',NFAL,IFOP(NFAL)
C                    WRITE(*,' ..NPARA_CHEX..')
C                    CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                    INDEX_CHEB = INDEX_CHEB + NVAL
C                    WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
c                    WRITE(*,*)' ..NPARA..INDEX..NVAL.',NPARA_CHEB,NVAL
c                    WRITE(*,*)'...N...M...',N_CHEB,M_CHEB
                    WRITE (LOUT, 3098) (PFAL(L,NFAL),L=4,NVAL+3)
101                 CONTINUE
                    LINE_CH = ' '
                    READ (5,'(A)') LINE_CH
c                    WRITE(*,*)' ..LINE_CH..'
c                    WRITE(*,*)LINE_CH
C
                    ILEN = IPPLEN(LINE_CH)
                    CALL CKISUB (LINE_CH(:ILEN), SUB_CH, NSUB_CH)
C                        KEY = ' '
                    DO 151 N_CH = 1, NSUB_CH
                       IILEN = ILASCH(SUB_CH(NSUB_CH))
c                       WRITE(*,*)'   NSUB_CH = ',NSUB_CH
c                       WRITE(*,*)SUB_CH(NSUB_CH)
                       II1 = INDEX(SUB_CH(NSUB_CH),'/')
                       II2 = INDEX(SUB_CH(NSUB_CH)(II1+1:),'/')
c                       WRITE(*,*)'....II1..II2...',II1,II2
                       IF (II1.LE.0 .OR. II2.LE.0) THEN
                         KERR = .TRUE.
                         WRITE (LOUT, 2090) SUB_CH(N_CH)(:IILEN)
C                         GO TO 199
                       ENDIF
C                       KEY = SUB(N)(:I1-1)
                       RSTR_CH = ' '
                       RSTR_CH = SUB_CH(NSUB_CH)(II1+1:II1+II2-1)
c                       WRITE(*,*)RSTR_CH
                    CALL IPPARR (RSTR_CH,1,-8,PFAL(INDEX_CHEB+4,NFAL)
     2                                ,NVAL,IER,LOUT)
c                       WRITE(*,*)'...INDEX..NVAL..',INDEX_CHEB,NVAL
c                       WRITE(*,*)'...IER..= ',IER
C                       INDEX_CHEB = INDEX_CHEB + NVAL
                       WRITE (*, 3098) (PFAL(L,NFAL),L=INDEX_CHEB+4
     2                                ,NVAL+INDEX_CHEB+3)
                       WRITE (LOUT, 3098) (PFAL(L,NFAL),L=INDEX_CHEB+4
     2                                ,NVAL+INDEX_CHEB+3)
                       INDEX_CHEB = INDEX_CHEB + NVAL
c                     WRITE(*,*)'..INDEX...',INDEX_CHEB
151                 CONTINUE
                    IF (INDEX_CHEB.LT.NPARA_CHEB) GOTO 101
199                 CONTINUE
                    IF (INDEX_CHEB .EQ. NPARA_CHEB) THEN
C
                    ELSE
                       WRITE (LOUT, 2028) SUB_CH(N)(:ILEN)
                       KERR = .TRUE.
                    ENDIF
C   ************************************************
C   **************W.ING 6/16/95************************
                ELSEIF (UPCASE(KEY, 4) .EQ. 'EXTR') THEN
                   IF (LTRO .OR. LSRI) THEN
                      KERR = .TRUE.
                      IF (LTRO) WRITE (LOUT, 2015) SUB(N)(:ILEN)
                      IF (LSRI) WRITE (LOUT, 2035) SUB(N)(:ILEN)
                   ELSE
                      LTRO = .TRUE.
                      CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                      IF (NVAL .EQ. 5) THEN
```

```
                        IFOP(NFAL) = 5*IFOP(NFAL)
C                        WRITE(*,*)IFOP(NFAL)
                        WRITE (LOUT, 3095) (PFAL(L,NFAL),L=4,8)
                     ELSE
                        WRITE (LOUT, 2025) SUB(N)(:ILEN)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
C    ************************************************
               ELSEIF (UPCASE(KEY, 3) .EQ. 'SRI') THEN
                  IF (LTRO .OR. LSRI) THEN
                     KERR = .TRUE.
                     IF (LTRO) WRITE (LOUT, 2030) SUB(N)(:ILEN)
                     IF (LSRI) WRITE (LOUT, 2040) SUB(N)(:ILEN)
                  ELSE
                     LSRI = .TRUE.
                     IFOP(NFAL) = 2*IFOP(NFAL)
                     CALL IPPARR (RSTR,1,-5,PFAL(4,NFAL),NVAL,IER,LOUT)
                     IF (NVAL .EQ. 3) THEN
                        PFAL(7,NFAL) = 1.0
                        PFAL(8,NFAL) = 0.0
                        WRITE (LOUT, 3060) (PFAL(L,NFAL),L=4,6)
                     ELSEIF (NVAL .EQ. 5) THEN
                        WRITE (LOUT, 3070) (PFAL(L,NFAL),L=4,8)
                     ELSE
                        WRITE (LOUT, 2020) SUB(N)(:ILEN)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
               ENDIF
C
         ELSEIF (UPCASE(KEY, 3) .EQ. 'REV') THEN
C
C        REVERSE ARRHENIUS PARAMETERS
C
            IF (LFAL .OR. LREV .OR. NSPEC(II).LT.0) THEN
               KERR = .TRUE.
               IF (LFAL) WRITE (LOUT, 1090) SUB(N)(:ILEN)
               IF (LREV) WRITE (LOUT, 2050) SUB(N)(:ILEN)
               IF (NSPEC(II) .LT. 0) WRITE (LOUT, 2060) SUB(N)(:ILEN)
            ELSE
               LREV = .TRUE.
               NREV = NREV+1
               IREV(NREV) = II
               CALL IPPARR (RSTR,1,NPAR,RPAR(1,NREV),NVAL,IER,LOUT)
               IF (IER .NE. 0) KERR = .TRUE.
               WRITE (LOUT, 1900) '   Reverse Arrhenius coefficients:',
     1                            (RPAR(L,NREV),L=1,3)
            ENDIF
C
         ELSEIF (UPCASE(KEY, 3) .EQ. 'RLT') THEN
C
C        REVERSE LANDAU-TELLER PARAMETERS
C
            IF (LFAL .OR. LRLT .OR. NSPEC(II).LT.0) THEN
               KERR = .TRUE.
               IF (LFAL) WRITE (LOUT, 1070) SUB(N)(:ILEN)
               IF (LRLT) WRITE (LOUT, 2080) SUB(N)(:ILEN)
               IF (NSPEC(II) .LT. 0) WRITE (LOUT, 1080) SUB(N)(:ILEN)
            ELSE
               LRLT = .TRUE.
               NRLT = NRLT + 1
               IRLT(NRLT) = II
               CALL IPPARR (RSTR,1,NLAR,RLAN(1,NRLT),NVAL,IER,LOUT)
               IF (IER .NE. 0) KERR = .TRUE.
```

```
            WRITE (LOUT, 3040) (RLAN(L,NRLT),L=1,2)
         ENDIF
C
      ELSEIF (UPCASE(KEY, 2) .EQ. 'HV') THEN
C
C     RADIATION WAVELENGTH ENHANCEMENT FACTOR
C
         IF (.NOT.LWL) THEN
            WRITE (LOUT, 1000) SUB(N)(:ILEN)
            KERR = .TRUE.
         ELSE
            CALL IPPARR (RSTR,1,1,VAL,NVAL,IER,LOUT)
            IF (IER .EQ. 0) THEN
               WL(NWL) = WL(NWL)*VAL(1)
               WRITE (LOUT, 3020) ABS(WL(NWL))
            ELSE
               WRITE (LOUT, 1000) SUB(N)(:ILEN)
               KERR = .TRUE.
            ENDIF
         ENDIF
C
      ELSEIF (UPCASE(KEY, 2) .EQ. 'LT') THEN
C
C     LANDAU-TELLER PARAMETERS
C
         IF (LFAL .OR. LLAN) THEN
            KERR = .TRUE.
            IF (LFAL) WRITE (LOUT, 1060) SUB(N)(:ILEN)
            IF (LLAN) WRITE (LOUT, 2070) SUB(N)(:ILEN)
         ELSE
            LLAN = .TRUE.
            NLAN = NLAN + 1
            ILAN(NLAN) = II
            CALL IPPARR (RSTR,1,NLAR,PLAN(1,NLAN),NVAL,IER,LOUT)
            IF (IER .NE. 0) THEN
               WRITE (LOUT, 1010) SUB(N)(:ILEN)
               KERR = .TRUE.
            ENDIF
            WRITE (LOUT, 3000) (PLAN(L,NLAN),L=1,2)
         ENDIF
C
      ELSE
C
C     ENHANCED THIRD BODIES
C
         CALL CKCOMP (KEY, KNAME, KK, K)
         IF (K .EQ. 0) THEN
            WRITE (LOUT, 1040) KEY(:ILASCH(KEY))
            KERR = .TRUE.
         ELSE
            IF (.NOT.LTHB) THEN
               KERR = .TRUE.
               WRITE (LOUT, 1020) SUB(N)(:ILEN)
            ELSE
               IF (NTBS(NTHB) .EQ. MAXTB) THEN
                  KERR = .TRUE.
                  WRITE (LOUT, 1030) SUB(N)(:ILEN)
               ELSE
                  CALL IPPARR (RSTR, 1, 1, VAL, NVAL, IER, LOUT)
                  IF (IER .EQ. 0) THEN
                     WRITE (LOUT, 3010) KNAME(K),VAL(1)
                     NTBS(NTHB) = NTBS(NTHB) + 1
                     NKTB(NTBS(NTHB),NTHB) = K
                     AIK(NTBS(NTHB),NTHB) = VAL(1)
                  ELSE
                     WRITE (LOUT, 1020) SUB(N)(:ILEN)
```

```
                        KERR = .TRUE.
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
  500 CONTINUE
C
C     FORMATS
C
 1000 FORMAT (6X,'Error in HV declaration...',A)
 1010 FORMAT (6X,'Error in LT declaration..',A)
 1020 FORMAT (6X,'Error in third body declaration...',A)
 1030 FORMAT (6X,'Error...more than MAXTB third bodies...',A)
 1040 FORMAT (6X,'Error...undeclared species...',A)
 1050 FORMAT (6X,'Error...this is not a fall-off reaction...',A)
 1060 FORMAT (6X,'Error...LT declared in fall-off reaction...',A)
 1070 FORMAT (6X,'Error...RLT declared in fall-off reaction...',A)
 1080 FORMAT (6X,'Error...RLT declared in irreversible reaction...',A)
 1090 FORMAT (6X,'Error...REV declared in fall-off reaction...',A)
 2000 FORMAT (6X,'Error...LOW declared more than once...',A)
 2010 FORMAT (6X,'Error...TROE declared more than once...',A)
 2015 FORMAT (6X,'Error...EXTROE declared more than once...',A)
 2020 FORMAT (6X,'Error in fall-off parameters...',A)
 2025 FORMAT (6X,'Error in EXTROE fall-off parameters...',A)
 2028 FORMAT (6X,'Error in CHEB fall-off parameters...',A)
 2030 FORMAT (6X,'Error...cannot use both TROE and SRI...',A)
 2035 FORMAT (6X,'Error...cannot use TROE, EXTROE or SRI...',A)
 2040 FORMAT (6X,'Error...SRI declared more than once...',A)
 2050 FORMAT (6X,'Error...REV declared more than once...',A)
 2060 FORMAT (6X,'Error...REV declared for irreversible reaction...',A)
 2070 FORMAT (6X,'Error...LT declared more than once...',A)
 2080 FORMAT (6X,'Error...RLT declared more than once...',A)
 2090 FORMAT (6X,'Error in auxiliary data...',A)
 3000 FORMAT (9X,'Landau-Teller parameters: B=',E12.5,', C=',E12.5)
 3010 FORMAT (9X,A16,' Enhanced by ',1PE12.3)
 3020 FORMAT (9X,'Radiation wavelength (A): ',F10.2)
C 1900 FORMAT (6X,A,T51,E10.3,F7.3,F11.3)
 1900 FORMAT (6X, A, T53, 1PE8.2, 2X, 0PF5.1, 2X, F9.1)
 3040 FORMAT (9X,'Reverse Landau-Teller parameters: B=',E12.5,
     1           ', C=',E12.5)
 3050 FORMAT (6X,'Low pressure limit:',3E13.5)
 3060 FORMAT (6X,'SRI centering:     ',3E13.5)
 3070 FORMAT (6X,'SRI centering:     ',5E13.5)
 3080 FORMAT (6X,'TROE centering:    ',3E13.5)
 3090 FORMAT (6X,'TROE centering:    ',4E13.5)
 3095 FORMAT (6X,'EXTROE centering:  ',5E13.5)
 3098 FORMAT (6X,'CHEB Polynomials:  ',8E13.5)
 4000 FORMAT (6X,'Declared duplicate reaction...')
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKPRNT (MDIM, MAXTP, MM, ENAME, KK, KNAME, WTM,
     1                   KPHSE, KCHRG, NT, T, TLO, TMID, THI, KNCF,
     2                   ITHRM, LOUT, KERR)
C
C     Prints species interpreter output and checks for completeness.
C-----------------------------------------------------------------------C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION WTM(*), KPHSE(*), KCHRG(*), T(MAXTP,*),
     1          NT(*), KNCF(MDIM,*), IPLUS(10)
```

```
      LOGICAL KERR, ITHRM(*)
      CHARACTER ENAME(*)*(*), KNAME(*)*(*), IPHSE(3)*1, INUM(10)*1
      DATA IPHSE/'S','G','L'/
      DATA INUM/'0','1','2','3','4','5','6','7','8','9'/
C
      WRITE (LOUT, 400) (ENAME(M), M = 1, MM)
      WRITE (LOUT, 300)
C
      DO 100 K = 1, KK
C
         IF (T(1,K) .LT. 0.0) T(1,K) = TLO
         IF (T(2,K) .LT. 0.0) T(2,K) = TMID
         IF (T(3,K) .LT. 0.0) T(NT(K),K) = THI
         WRITE (LOUT, 500) K, KNAME(K), IPHSE(KPHSE(K)+2), KCHRG(K),
     1                     WTM(K), T(1,K), T(NT(K),K), (KNCF(M,K),M=1,MM)
         IF (T(1,K) .GE. T(NT(K),K)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 240)
         ENDIF
         IF (T(1,K) .GT. T(2,K)) THEN
            WRITE (LOUT, 250)
            KERR = .TRUE.
         ENDIF
         IF (T(NT(K),K) .LT. T(2,K)) THEN
            WRITE (LOUT, 260)
            KERR = .TRUE.
         ENDIF
C
C        each species must have thermodynamic data
C
         IF (.NOT. ITHRM(K)) THEN
            KERR = .TRUE.
            WRITE (LOUT, 200)
         ENDIF
C
C        a species cannot start with a number
C
         CALL CKCOMP (KNAME(K)(:1), INUM, 10, I)
         IF (I .GT. 0) THEN
            KERR = .TRUE.
            WRITE (LOUT, 210)
         ENDIF
C
C        if '+' sign is used in a species name,
C           examples of legal species symbols with + are:
C           OH(+)2, OH(+2), OH+, OH++, OH+++, OH(+), OH(++),
C           OH[+OH], OH2+, OH+2
C
C           examples of illegal species symbols with + are:
C           +OH         (symbol starts with a +, this will cause
C                        confusion in a reaction)
C           OH(+OH)     (symbol in parentheses is another species-
C                        this arrangement is reserved for a fall-off
C                        reaction)
C           OH+OH       (plus delimits other species names, this
C                        will cause confusion in a reaction)
C
         NPLUS = 0
         DO 50 N = 1, ILASCH(KNAME(K))
            IF (KNAME(K)(N:N) .EQ. '+') THEN
               NPLUS = NPLUS + 1
               IPLUS(NPLUS) = N
            ENDIF
   50    CONTINUE
         DO 60 N = 1, NPLUS
            I1 = IPLUS(N)
```

```
            IF (I1 .EQ. 1) THEN
               WRITE (LOUT, 220)
               KERR = .TRUE.
            ELSE
C
C              is there another species name in parentheses
C
               IF (KNAME(K)(I1-1:I1-1) .EQ. '(') THEN
                  I1 = I1 + 1
                  I2 = I1 + INDEX(KNAME(K)(I1:),')')-1
                  IF (I2 .GT. I1) THEN
                     CALL CKCOMP (KNAME(K)(I1:I2-1), KNAME, KK, KNUM)
                     IF (KNUM .GT. 0) THEN
                        WRITE (LOUT, 230)
                        KERR = .TRUE.
                     ENDIF
                  ENDIF
               ENDIF
C
C              is there another species name after a +
C
               I1 = I1 + 1
               IF (N .LT. NPLUS) THEN
                  DO 55 L = N+1, NPLUS
                     I2 = IPLUS(L)
                     IF (I2 .GT. I1) THEN
                        CALL CKCOMP (KNAME(K)(I1:I2-1),KNAME,KK,KNUM)
                        IF (KNUM .GT. 0) THEN
                           WRITE (LOUT, 230)
                           KERR = .TRUE.
                        ENDIF
                     ENDIF
   55             CONTINUE
               ENDIF
C
               I2 = ILASCH(KNAME(K))
               IF (I2 .GE. I1) THEN
                  CALL CKCOMP (KNAME(K)(I1:I2), KNAME, KK, KNUM)
                  IF (KNUM .GT. 0) THEN
                     WRITE (LOUT, 230)
                     KERR = .TRUE.
                  ENDIF
               ENDIF
            ENDIF
   60    CONTINUE
C
  100 CONTINUE
      WRITE (LOUT, 300)
      RETURN
C
  200 FORMAT (6X,'Error...no thermodynamic properties for species')
  210 FORMAT (6X,'Error...species starts with a number')
  220 FORMAT (6X,'Error...species starts with a plus')
  230 FORMAT (6X,'Error...illegal + in species name')
  240 FORMAT (6X,'Error...High temperature must be < Low temperature')
  250 FORMAT (6X,'Error...Low temperature must be <= Mid temperature')
  260 FORMAT (6X,'Error...High temperature must be => Mid temperature')
  300 FORMAT (1X,79('-'))
C  400 FORMAT (1X,79('-'),/21X,'C',/18X,'P',2X,'H',/18X,'H',2X,'A',
C     1        /18X,'A',2X,'R',/1X,'SPECIES',10X,'S',2X,'G',2X,
C     2        'MOLECULAR',3X,'TEMPERATURE',4X,'ELEMENT COUNT',/1X,
C     3        'CONSIDERED',7X,'E',2X,'E',2X,'WEIGHT',6X,'LOW',5X,
C     4        'HIGH',3X,15(A3),/1X,79('-'))
C  500 FORMAT (I4,'. ',A10,2X,A1,I3,F11.5,2(F8.1),15(I3))
C
  400 FORMAT (1X,79('-'),/T26,'C',/T24,'P H',/T24,'H A',/T24,'A R',
```

```
     1       /1X,'SPECIES',T24,'S G',T28,'MOLECULAR',T38,'TEMPERATURE',
     2       T52,'ELEMENT COUNT',
     3       /1X,'CONSIDERED',T24,'E E',T28,'WEIGHT',T38,'LOW',
     4       T45,'HIGH',T52,15(A3))
  500 FORMAT (1X,I3,'. ',A16,T24,A1,T26,I1,T28,F9.5,T38,F6.1,T45,F6.1,
     1       T51,15(I3))
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CPREAC (II, MAXSP, NSPEC, NPAR, PAR, RPAR, AUNITS,
     1                   EUNITS, NREAC, NUNK, NU, KCHRG, MDIM, MM, KNCF,
     2                   IDUP, NFAL, IFAL, KFAL, NFAR, PFAL, IFOP, NREV,
     3                   IREV, NTHB, ITHB, NLAN, ILAN, NRLT, IRLT, KERR,
     4                   LOUT)
C
C     Prints reaction interpreter output and checks for reaction
C     balance, duplication, and missing data in 'REV' reactions;
C     correct units of Arrhenius parameters
C
C     Input: II     - the index number of the reaction
C            MAXSP  - maximum number of species allowed in a reaction
C            NSPEC  - array of the number of species in the reactions
C            NPAR   - the number of Arrhenius parameters required
C            PAR    - matrix of Arrhenius parameters for the reactions
C            RPAR   - matrix of reverse Arrhenius parameters for the
C                     reactions which declared them
C            AUNITS - character string which describes the input units
C                     of A, the pre-exponential factor PAR(1,I)
C            EUNITS - character string which describes the input units
C                     of E, the activation energy PAR(3,I)
C            NREAC  - array of the number of reactants in the reactions
C            NUNK   - matrix of the species numbers of the reactants
C                     and products in the reactions
C            NU     - matrix of the stoichiometric coefficients of the
C                     reactants and products in the reactions
C            KCHRG  - array of the electronic charges of the species
C            MDIM   - the maximum number of elements allowed
C            MM     - the actual number of elements declared
C            KNCF   - matrix of elemental composition of the species
C            IDUP   - array of integer flags to indicate duplicate
C                     reactions
C            NFAL   - total number of reactions with fall-off
C            IFAL   - array of the NFAL reaction numbers
C            NFAR   - maximum number of fall-off parameters allowed
C            PFAL   - matrix of fall-off parameters for the NFAL
C                     reactions
C            IFOP   - array of integer fall-off types for the NFAL
C                     reactions
C            NREV   - total number of reactions with reverse parameters
C            IREV   - array of the NREV reaction numbers
C            NTHB   - total number of reactions with third-bodies
C            ITHB   - array of the NTHB reaction numbers
C            NLAN   - total number of reactions with Landauer-Teller
C                     parameters
C            ILAN   - array of the NLAN reaction numbers
C            NRLT   - total number of reactions with reverse
C                     Landauer-Teller parameters
C            IRLT   - array of the NRLT reaction numbers
C            KERR   - logical error flag
C            LOUT   - unit number for output messages
C
C-----------------------------------------------------------------------C
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
```

```
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NSPEC(*), PAR(NPAR,*), RPAR(NPAR,*), NREAC(*),
     1          NUNK(MAXSP,*), NU(MAXSP,*), KCHRG(*), KNCF(MDIM,*),
     2          IDUP(*), IFAL(*), KFAL(*), PFAL(NFAR,*), IFOP(*),
     3          IREV(*), ITHB(*), ILAN(*), IRLT(*)
      CHARACTER*(*) AUNITS, EUNITS
      LOGICAL IERR,KERR,LREV,LLAN,LRLT
C
      CALL CKBAL (MAXSP, NUNK(1,II), NU(1,II), MDIM, MM, KCHRG, KNCF,
     1            IERR)
C
      IF (IERR) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1060)
      ENDIF
C
      CALL CKDUP (II, MAXSP, NSPEC, NREAC, NU, NUNK, NFAL, IFAL, KFAL,
     1            ISAME)
C
      IF (ISAME .GT. 0) THEN
         IF (IDUP(ISAME).NE.0 .AND. IDUP(II).NE.0) THEN
            IDUP(ISAME) = ABS(IDUP(ISAME))
            IDUP(II)    = ABS(IDUP(II))
         ELSE
            N1 = 0
            N2 = 0
            IF (NTHB .GT. 1) THEN
               DO 150 N = 1, NTHB
                  IF (ITHB(N) .EQ. ISAME) N1 = 1
                  IF (ITHB(N) .EQ. II)    N2 = 1
  150          CONTINUE
            ENDIF
            IF (N1 .EQ. N2) THEN
               KERR = .TRUE.
               WRITE (LOUT, 1050) ISAME
            ENDIF
         ENDIF
      ENDIF
C
      IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II .AND. IFOP(NFAL).LT.0) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1020)
      ENDIF
C
      LREV = (NREV.GT.0 .AND. IREV(NREV).EQ.II)
      LLAN = (NLAN.GT.0 .AND. ILAN(NLAN).EQ.II)
      LRLT = (NRLT.GT.0 .AND. IRLT(NRLT).EQ.II)
      IF (LREV .AND. LLAN .AND. (.NOT.LRLT)) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1030)
      ENDIF
      IF (LRLT .AND. (.NOT.LLAN)) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1040)
      ENDIF
      IF (LRLT .AND. (.NOT.LREV)) THEN
         KERR = .TRUE.
         WRITE (LOUT, 1045)
      ENDIF
C
      IF (EUNITS .EQ. 'KELV') THEN
         EFAC = 1.0
      ELSEIF (EUNITS .EQ. 'CAL/') THEN
C        convert E from cal/mole to Kelvin
```

```
            EFAC = 1.0 / 1.987
         ELSEIF (EUNITS .EQ. 'KCAL') THEN
C           convert E from kcal/mole to Kelvin
            EFAC = 1000.0 / 1.987
         ELSEIF (EUNITS .EQ. 'JOUL') THEN
C           convert E from Joules/mole to Kelvin
            EFAC = 1.0 / 8.314
         ELSEIF (EUNITS .EQ. 'KJOU') THEN
C           convert E from Kjoules/mole to Kelvin
            EFAC = 1000.0 / 8.314
         ENDIF
         PAR(3,II) = PAR(3,II) * EFAC
C
C         IF (NREV.GT.0 .AND. IREV(NREV).EQ.II) RPAR(3,II)=RPAR(3,II)*EFAC
C         IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) PFAL(3,II)=PFAL(3,II)*EFAC
C
         IF (NREV.GT.0 .AND. IREV(NREV).EQ.II)
        1    RPAR(3,NREV) = RPAR(3,NREV) * EFAC
         IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II)
        1    PFAL(3,NFAL) = PFAL(3,NFAL) * EFAC
C
         IF (AUNITS .EQ. 'MOLC') THEN
            NSTOR = 0
            NSTOP = 0
            DO 50 N = 1, MAXSP
               IF (NU(N,II) .LT. 0) THEN
C                 sum of stoichiometric coefficients of reactants
                  NSTOR = NSTOR + ABS(NU(N,II))
               ELSEIF (NU(N,II) .GT. 0) THEN
C                 sum of stoichiometric coefficients of products
                  NSTOP = NSTOP + NU(N,II)
               ENDIF
   50       CONTINUE
C
            AVAG = 6.023E23
C
            IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.II) THEN
C
C              fall-off reaction, "(+M)" or "(+species name)" does not
C              count except in "LOW" A-factor;
C              reverse-rate declarations are not allowed
C
               IF (NSTOR.GT.0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
               NSTOR = NSTOR + 1
               IF (NSTOR.GT.0) PFAL(1,NFAL) = PFAL(1,NFAL)*AVAG**(NSTOR-1)
C
            ELSEIF (NTHB.GT.0 .AND. ITHB(NTHB).EQ.II) THEN
C
C              third body reaction, "+M" counts as species in
C              forward and reverse A-factor conversion
C
               NSTOR = NSTOR + 1
               NSTOP = NSTOP + 1
               IF (NSTOR.GT.0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
               IF (NREV.GT.0 .AND. IREV(NREV).EQ.II .AND. NSTOP.GT.0)
        1          RPAR(1,NREV) = RPAR(1,NREV) * AVAG**(NSTOP-1)
C
            ELSE
C
C              not third-body or fall-off reaction, but may have
C              reverse rates.
C
               IF (NSTOR .GT. 0) PAR(1,II) = PAR(1,II) * AVAG**(NSTOR-1)
               IF (NREV.GT.0 .AND. IREV(NREV).EQ.II .AND. NSTOP.GT.0)
        1          RPAR(1,NREV) = RPAR(1,NREV) * AVAG**(NSTOP-1)
            ENDIF
```

```
      ENDIF
C
 1020 FORMAT (6X,'Error...no LOW parameters given for fall-off...')
 1030 FORMAT (6X,'Error...reverse T-L required...')
 1040 FORMAT (6X,'Error...forward T-L required...')
 1045 FORMAT (6X,'Error...REV parameters must be given with RTL...')
 1050 FORMAT (6X,'Error...undeclared duplicate to reaction number ',I3)
 1060 FORMAT (6X,'Error...reaction does not balance...')
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKBAL (NSPEC, KSPEC, KCOEF, MDIM, MM, KCHRG, KNCF,
     1                  IERR)
C
C     Checks elemental balance of reactants vs. products.
C     Checks charge balance of reaction.
C
C     Input:  NSPEC - total number of species in this reaction
C             KSPEC(N),N=1,NSPEC- array of species numbers in reaction
C             KCOEF(N) - stoichiometric coefficients of the species
C             MDIM  - maximum number of elements allowed
C             MM    - actual integer number of elements
C             KCHRG(K) - ionic charge Kth species
C             KNCF(M,K)- integer elemental composition of Kth species
C     Output: KERR  - logical, =.TRUE. if reaction does not balance
C                                     F. Rupley, Div. 8245, 5/13/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION KSPEC(*), KCOEF(*), KNCF(MDIM,*), KCHRG(*)
      LOGICAL IERR
C
      IERR = .FALSE.
C
C     charge balance
C
      KBAL = 0
      DO 50 N = 1, ABS(NSPEC)
         KBAL = KBAL + KCOEF(N)*KCHRG(KSPEC(N))
   50 CONTINUE
      IF (KBAL .NE. 0) IERR = .TRUE.
C
C     element balance
C
      DO 100 M = 1, MM
         MBAL = 0
         DO 80 N = 1, ABS(NSPEC)
            MBAL = MBAL + KCOEF(N)*KNCF(M,KSPEC(N))
   80    CONTINUE
         IF (MBAL .NE. 0) IERR = .TRUE.
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKDUP (I, MAXSP, NS, NR, NU, NUNK, NFAL, IFAL, KFAL,
     1                  ISAME)
C
C     Checks reaction I against the (I-1) reactions for duplication
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
```

```
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      DIMENSION NS(*), NR(*), NU(MAXSP,*), NUNK(MAXSP,*), IFAL(*),
     1          KFAL(*)
C
      ISAME = 0
      NRI = NR(I)
      NPI = ABS(NS(I)) - NR(I)
C
      DO 500 J = 1, I-1
C
         NRJ = NR(J)
         NPJ = ABS(NS(J)) - NR(J)
C
         IF (NRJ.EQ.NRI .AND. NPJ.EQ.NPI) THEN
C
            NSAME = 0
            DO 20 N = 1, MAXSP
               KI = NUNK(N,I)
               NI = NU(N,I)
C
               DO 15 L = 1, MAXSP
                  KJ = NUNK(L,J)
                  NJ = NU(L,J)
                  IF (NJ.NE.0 .AND. KJ.EQ.KI .AND. NJ.EQ.NI)
     1            NSAME = NSAME + 1
   15          CONTINUE
   20       CONTINUE
C
            IF (NSAME .EQ. ABS(NS(J))) THEN
C
C           same products, reactants, coefficients, check fall-off
C           third body
C
               IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.I) THEN
                  DO 22 N = 1, NFAL-1
                     IF (J.EQ.IFAL(N) .AND. KFAL(N).EQ.KFAL(NFAL)) THEN
                        ISAME = J
                        RETURN
                     ENDIF
   22             CONTINUE
                  RETURN
               ENDIF
C
               ISAME = J
               RETURN
            ENDIF
         ENDIF
C
         IF (NPI.EQ.NRJ .AND. NPJ.EQ.NRI) THEN
C
            NSAME = 0
            DO 30 N = 1, MAXSP
               KI = NUNK(N,I)
               NI = NU(N,I)
C
               DO 25 L = 1, MAXSP
                  KJ = NUNK(L,J)
                  NJ = NU(L,J)
                  IF (NJ.NE.0 .AND. KJ.EQ.KI .AND. -NJ.EQ.NI)
     1            NSAME = NSAME + 1
   25          CONTINUE
   30       CONTINUE
```

```
C
            IF (NSAME.EQ.ABS(NS(J)) .AND.
     1          (NS(J).GT.0 .OR. NS(I).GT.0)) THEN
C
C           same products as J reactants, and vice-versa
C
               IF (NFAL.GT.0 .AND. IFAL(NFAL).EQ.I) THEN
                  DO 32 N = 1, NFAL-1
                     IF (J.EQ.IFAL(N) .AND. KFAL(N).EQ.KFAL(NFAL)) THEN
                        ISAME = J
                        RETURN
                     ENDIF
   32                CONTINUE
                     RETURN
                  ENDIF
C
                  ISAME = J
                  RETURN
               ENDIF
            ENDIF
C
  500 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE CKISUB (LINE, SUB, NSUB)
C
C     Generates an array of CHAR*(*) substrings from a CHAR*(*) string,
C     using blanks or tabs as delimiters
C
C     Input:  LINE  - a CHAR*(*) line
C     Output: SUB   - a CHAR*(*) array of substrings
C             NSUB  - number of substrings found
C     A '!' will comment out a line, or remainder of the line.
C                                        F. Rupley, Div. 8245, 5/15/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) SUB(*), LINE
      NSUB = 0
C
      DO 5 N = 1, LEN(LINE)
        IF (ICHAR(LINE(N:N)) .EQ. 9) LINE(N:N) = ' '
    5 CONTINUE
C
      IF (IPPLEN(LINE) .LE. 0) RETURN
C
      ILEN = ILASCH(LINE)
C
      NSTART = IFIRCH(LINE)
   10 CONTINUE
      ISTART = NSTART
      NSUB = NSUB + 1
      SUB(NSUB) = ' '
C
      DO 100 I = ISTART, ILEN
         ILAST = INDEX(LINE(ISTART:),' ') - 1
         IF (ILAST .GT. 0) THEN
            ILAST = ISTART + ILAST - 1
         ELSE
            ILAST = ILEN
```

```
         ENDIF
         SUB(NSUB) = LINE(ISTART:ILAST)
         IF (ILAST .EQ. ILEN) RETURN
C
         NSTART = ILAST + IFIRCH(LINE(ILAST+1:))
C
C        Does SUB have any slashes?
C
         I1 = INDEX(SUB(NSUB),'/')
         IF (I1 .LE. 0) THEN
            IF (LINE(NSTART:NSTART) .NE. '/') GO TO 10
            NEND = NSTART + INDEX(LINE(NSTART+1:),'/')
            IND = INDEX(SUB(NSUB),' ')
            SUB(NSUB)(IND:) = LINE(NSTART:NEND)
            IF (NEND .EQ. ILEN) RETURN
            NSTART = NEND + IFIRCH(LINE(NEND+1:))
            GO TO 10
         ENDIF
C
C        Does SUB have 2 slashes?
C
         I2 = INDEX(SUB(NSUB)(I1+1:),'/')
         IF (I2 .GT. 0) GO TO 10
C
         NEND = NSTART + INDEX(LINE(NSTART+1:),'/')
         IND = INDEX(SUB(NSUB),' ') + 1
         SUB(NSUB)(IND:) = LINE(NSTART:NEND)
         IF (NEND .EQ. ILEN) RETURN
         NSTART = NEND + IFIRCH(LINE(NEND+1:))
         GO TO 10
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPNPAR (LINE, NPAR, IPAR, ISTART)
C
C     Returns CHAR*(*) IPAR substring of CHAR*(*) string LINE which
C     contains NPAR real parameters
C
C     Input:     LINE - a CHAR*(*) line
C                NPAR - number of parameters expected
C     Output:    IPAR - the substring of parameters only
C                ISTART - the starting location of IPAR substring
C     A '!' will comment out a line, or remainder of the line.
C                                        F. Rupley, Div. 8245, 5/14/86
C-----------------------------------------------------------------------C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) LINE,IPAR
C
C----------Find Comment String (! signifies comment)
C
      ILEN = IPPLEN(LINE)
      ISTART = 0
      N = 0
      IF (ILEN.GT.0) THEN
         DO 40 I = ILEN, 1, -1
            ISTART = I
            IPAR = ' '
            IPAR = LINE(ISTART:ILEN)
            IF (LINE(I:I).NE.' ') THEN
```

```
               IF (I .EQ. 1) RETURN
               IF (LINE(I-1:I-1) .EQ. ' ') THEN
                  N = N + 1
                  IF (N .EQ. NPAR) RETURN
               ENDIF
            ENDIF
   40    CONTINUE
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------------C
      SUBROUTINE IPPARI(STRING, ICARD, NEXPEC, IVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARI
C   REFER TO  IPGETI
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851725   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses integer variables from a character variable.  Called
C            by IPGETI, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARI may be used for parsing an input record that contains integer
C  values, but was read into a character variable instead of directly
C  into integer variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARI to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of IVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1,   2,,40000   , ,60'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set IVAL(1) = 1
C     (2) set IVAL(2) = 2
C     (3) leave IVAL(3) unchanged
C     (4) set IVAL(4) = 40000
C     (5) leave IVAL(5) unchanged
C     (6) set IVAL(6) = 60
C
C   IPPARI will print diagnostics on the default output device, if
C   desired.
C
C   IPPARI is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume IVAL = (0, 0, 0) and NEXPEC = 3 on entry:
C
C   input string           IVAL on exit            IERR    NFOUND
C   ------------           ---------------------   ----    ------
C  '  2 ,   3 45 '         (2, 3, 45)                0       3
C  '2.15,,3'              (2, 0, 3)                 1       0
C  '3X, 25, 2'            (0, 0, 0)                 1       0
```

```
C  '10000'                  (10000, 0, 0)             2       1
C
C      Assume IVAL = (0, 0, 0, 0) and NEXPEC = -4 on entry:
C
C   input string            IVAL on exit              IERR    NFOUND
C   ------------            ----------------------    ----    ------
C  '1, 2'                   (1, 2)                      0       2
C  ',,37  400'              (0, 0, 37, 400)             0       4
C  ' 1,,-3,,5'              (1, 0, -3, 0)               3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  IVAL (I,O) - the integer value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to IVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARI
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION IVAL(*)
      CHARACTER *8 FMT(14)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARI
      IERR   = 0
      NFOUND = 0
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
```

```
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set false when a space follows
C--- an integer value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR .OR. NC .EQ. IE) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into integer array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      IES = NC - 1
      NCH = IES - IBS + 1
      DATA FMT/' (I1)', ' (I2)', ' (I3)', ' (I4)', ' (I5)',
     1   ' (I6)', ' (I7)', ' (I8)', ' (I9)', '(I10)',
     2   '(I11)', '(I12)', '(I13)', '(I14)'/
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) IVAL(NFOUND)
      GO TO 450
400   CONTINUE
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
```

```
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0)RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1    WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      SUBROUTINE IPPARR(STRING, ICARD, NEXPEC, RVAL, NFOUND, IERR, LOUT)
C   BEGIN PROLOGUE  IPPARR
C   REFER TO  IPGETR
C   DATE WRITTEN  850625   (YYMMDD)
C   REVISION DATE 851625   (YYMMDD)
C   CATEGORY NO.  J3.,J4.,M2.
C   KEYWORDS  PARSE
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Parses real variables from a character variable.  Called
C            by IPGETR, the IOPAK routine used for interactive input.
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IPPARR may be used for parsing an input record that contains real
C  values, but was read into a character variable instead of directly
C  into real variables.
C  The following benefits are gained by this approach:
C    - specification of only certain elements of the array is allowed,
C      thus letting the others retain default values
C    - variable numbers of values may be input in a record, up to a
C      specified maximum
C    - control remains with the calling program in case of an input
C      error
C    - diagnostics may be printed by IPPARR to indicate the nature
C      of input errors
C
C   The contents of STRING on input indicate which elements of RVAL
C   are to be changed from their entry values, and values to which
C   they should be changed on exit.  Commas and blanks serve as
C   delimiters, but multiple blanks are treated as a single delimeter.
C   Thus, an input record such as:
C     '   1.,   2,,4.e-5   , ,6.e-6'
C   is interpreted as the following set of instructions by IPGETR:
C
C     (1) set RVAL(1) = 1.0
C     (2) set RVAL(2) = 2.0
C     (3) leave RVAL(3) unchanged
C     (4) set RVAL(4) = 4.0E-05
C     (5) leave RVAL(5) unchanged
C     (6) set RVAL(6) = 6.0E-06
C
C   IPPARR will print diagnostics on the default output device, if
C   desired.
C
C   IPPARR is part of IOPAK, and is written in ANSI FORTRAN 77
C
C   Examples:
C
C      Assume RVAL = (0., 0., 0.) and NEXPEC = 3 on entry:
C
C   input string           RVAL on exit            IERR    NFOUND
```

```
C   -------------          ----------------------  ----    ------
C  '  2.34e-3,  3 45.1'    (2.34E-03, 3.0, 45.1)     0       3
C  '2,,3.-5'               (2.0, 0.0, 3.0E-05)       0       3
C  ',1.4,0.028E4'          (0.0, 1.4, 280.0)         0       3
C  '1.0, 2.a4, 3.0'        (1.0, 0.0, 0.0)           1       1
C  '1.0'                   (1.0, 0.0, 0.0)           2       1
C
C      Assume RVAL = (0.,0.,0.,0.) and NEXPEC = -4 on entry:
C
C   input string           RVAL on exit            IERR    NFOUND
C   ------------           ----------------------  ----    ------
C  '1.,2.'                 (1.0, 2.0)                0       2
C  ',,3  4.0'              (0.0, 0.0, 3.0, 4.0)      0       4
C  '1,,3,,5.0'             (0.0, 0.0, 3.0, 0.0)      3       4
C
C  arguments: (I=input,O=output)
C  ----------------------------
C  STRING (I) - the character string to be parsed.
C
C  ICARD  (I) - data statement number, and error processing flag
C         < 0 : no error messages printed
C         = 0 : print error messages, but not ICARD
C         > 0 : print error messages, and ICARD
C
C  NEXPEC (I) - number of real variables expected to be input.  If
C         < 0, the number is unknown, and any number of values
C         between 0 and abs(nexpec) may be input.  (see NFOUND)
C
C  PROMPT (I) - prompting string, character type.  A question
C         mark will be added to form the prompt at the screen.
C
C  RVAL (I,O) - the real value or values to be modified.  On entry,
C       the values are printed as defaults.  The formal parameter
C       corresponding to RVAL must be dimensioned at least NEXPEC
C       in the calling program if NEXPEC > 1.
C
C  NFOUND (O) - the number of real values represented in STRING,
C         only in the case that there were as many or less than
C         NEXPEC.
C
C  IERR (O) - error flag:
C       = 0 if no errors found
C       = 1 syntax errors or illegal values found
C       = 2 for too few values found (NFOUND < NEXPEC)
C       = 3 for too many values found (NFOUND > NEXPEC)
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  IFIRCH,ILASCH
C   END PROLOGUE  IPPARR
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER STRING*(*), ITEMP*80
      DIMENSION RVAL(*)
      CHARACTER *8 FMT(22)
      LOGICAL OKINCR
C
C   FIRST EXECUTABLE STATEMENT  IPPARR
      IERR   = 0
      NFOUND = 0
```

```
      NEXP = IABS(NEXPEC)
      IE = ILASCH(STRING)
      IF (IE .EQ. 0) GO TO 500
      NC = 1
C
C--- OKINCR is a flag that indicates it's OK to increment
C--- NFOUND, the index of the array into which the value
C--- should be read.  It is set negative when a space follows
C--- a real value substring, to keep incrementing from
C--- occurring if a comma should be encountered before the
C--- next value.
C
      OKINCR = .TRUE.
C
C--- begin overall loop on characters in string
C
100   CONTINUE
C
      IF (STRING(NC:NC) .EQ. ',') THEN
         IF (OKINCR) THEN
            NFOUND = NFOUND + 1
         ELSE
            OKINCR = .TRUE.
         ENDIF
C
         GO TO 450
      ENDIF
      IF (STRING(NC:NC) .EQ. ' ') GO TO 450
C
C--- first good character (non-delimeter) found - now find
C--- last good character
C
      IBS = NC
160   CONTINUE
      NC = NC + 1
      IF (NC .GT. IE) GO TO 180
      IF (STRING(NC:NC) .EQ. ' ')THEN
         OKINCR = .FALSE.
      ELSEIF (STRING(NC:NC) .EQ. ',')THEN
         OKINCR = .TRUE.
      ELSE
         GO TO 160
      ENDIF
C
C--- end of substring found - read value into real array
C
180   CONTINUE
      NFOUND = NFOUND + 1
      IF (NFOUND .GT. NEXP) THEN
         IERR = 3
         GO TO 500
      ENDIF
C
      DATA FMT/     ' (E1.0)', ' (E2.0)', ' (E3.0)', ' (E4.0)',
     1    ' (E5.0)', ' (E6.0)', ' (E7.0)', ' (E8.0)', ' (E9.0)',
     2   '(E10.0)', '(E11.0)', '(E12.0)', '(E13.0)', '(E14.0)',
     3   '(E15.0)', '(E16.0)', '(E17.0)', '(E18.0)', '(E19.0)',
     4   '(E20.0)', '(E21.0)', '(E22.0)'/
      IES = NC - 1
      NCH = IES - IBS + 1
      ITEMP = ' '
      ITEMP = STRING(IBS:IES)
      READ (ITEMP(1:NCH), FMT(NCH), ERR = 400) RVAL(NFOUND)
      GO TO 450
400   CONTINUE
      WRITE (LOUT, 555) STRING(IBS:IES)
```

```
  555 FORMAT (A)
      IERR = 1
      GO TO 510
450   CONTINUE
      NC = NC + 1
      IF (NC .LE. IE) GO TO 100
C
500   CONTINUE
      IF (NEXPEC .GT. 0 .AND. NFOUND .LT. NEXP) IERR = 2
510   CONTINUE
C
      IF (IERR .EQ. 0 .OR. ICARD .LT. 0) RETURN
      IF (ICARD .NE. 0) WRITE (LOUT, '(A,I3)')
     1   '!! ERROR IN DATA STATEMENT NUMBER', ICARD
      IF (IERR .EQ. 1)
     1   WRITE (LOUT, '(A)')'SYNTAX ERROR, OR ILLEGAL VALUE'
      IF (IERR .EQ. 2) WRITE (LOUT, '(A,I2, A, I2)')
     1   ' TOO FEW DATA ITEMS.  NUMBER FOUND = ' , NFOUND,
     2   '  NUMBER EXPECTED = ', NEXPEC
      IF (IERR .EQ. 3) WRITE (LOUT, '(A,I2)')
     1   ' TOO MANY DATA ITEMS.  NUMBER EXPECTED = ', NEXPEC
      END
C
      FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
```

```
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END
      FUNCTION ILASCH(STRING)
C   BEGIN PROLOGUE  ILASCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines last significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the last non-blank character in a string of
C  arbitrary length.  If no characters are found, ILASCH is set = 0.
C  When used with the companion routine IFIRCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C  Note that the FORTRAN intrinsic function LEN returns the length
C  of a character string as declared, rather than as filled.  The
C  declared length includes leading and trailing blanks, and thus is
C  not useful in generating 'significant' substrings.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) STRING
C
C***FIRST EXECUTABLE STATEMENT ILASCH
      NLOOP = LEN(STRING)
      IF (NLOOP.EQ.0) THEN
         ILASCH = 0
         RETURN
      ENDIF
C
      DO 100 I = NLOOP, 1, -1
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
120   CONTINUE
      ILASCH = I
      END
C-----------------------------------------------------------------------C
C
      SUBROUTINE CKCOMP (IST, IRAY, II, I)
C
C  START PROLOGUE
C
C  SUBROUTINE CKCOMP (IST, IRAY, II, I)*
C     Returns the index of an element of a reference character
C     string array which corresponds to a character string;
C     leading and trailing blanks are ignored.
```

```
C
C
C  INPUT
C     IST   - A character string.
C                   Data type - CHARACTER*(*)
C     IRAY  - An array of character strings;
C             dimension IRAY(*) at least II
C                   Data type - CHARACTER*(*)
C     II    - The length of IRAY.
C                   Data type - integer scalar.
C
C  OUTPUT
C     I     - The first integer location in IRAY in which IST
C             corresponds to IRAY(I); if IST is not also an
C             entry in IRAY, I=0.
C
C  END PROLOGUE
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER*(*) IST, IRAY(*)
C
      I = 0
      DO 10 N = II, 1, -1
         IS1 = IFIRCH(IST)
         IS2 = ILASCH(IST)
         IR1 = IFIRCH(IRAY(N))
         IR2 = ILASCH(IRAY(N))
         IF ( IS2.GE.IS1 .AND. IS2.GT.0 .AND.
     1        IR2.GE.IR1 .AND. IR2.GT.0 .AND.
     2        IST(IS1:IS2).EQ.IRAY(N)(IR1:IR2) ) I=N
   10 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------C
      SUBROUTINE CKUNIT (LINE, AUNITS, EUNITS, IUNITS)
C
C*****precision > double
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
      CHARACTER*(*) LINE, IUNITS, AUNITS, EUNITS
      CHARACTER*4 UPCASE
C
      AUNITS = ' '
      EUNITS = ' '
      IUNITS = ' '
      DO 85 N = 1, ILASCH(LINE)-3
         IND = ILASCH(IUNITS)
         IF (EUNITS .EQ. ' ') THEN
            IF (UPCASE(LINE(N:), 4)     .EQ. 'CAL/') THEN
               EUNITS = 'CAL/'
               IF (IUNITS .EQ. ' ') THEN
                  IUNITS = 'E units cal/mole'
               ELSE
                  IUNITS(IND:) = ', E units cal/mole'
               ENDIF
            ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KCAL') THEN
```

```
                EUNITS = 'KCAL'
                IF (IUNITS .EQ. ' ') THEN
                   IUNITS = 'E units Kcal/mole'
                ELSE
                   IUNITS(IND:) = ', E units Kcal/mole'
                ENDIF
             ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'JOUL') THEN
                EUNITS = 'JOUL'
                IF (IUNITS .EQ. ' ') THEN
                   IUNITS = 'E units Joules/mole'
                ELSE
                   IUNITS(IND:) = ', E units Joules/mole'
                ENDIF
             ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KJOU') THEN
                EUNITS = 'KJOU'
                IF (IUNITS .EQ. ' ') THEN
                   IUNITS = 'E units Kjoule/mole'
                ELSE
                   IUNITS(IND:) = ', E units Kjoule/mole'
                ENDIF
             ELSEIF (UPCASE(LINE(N:), 4) .EQ. 'KELV') THEN
                EUNITS = 'KELV'
                IF (IUNITS .EQ. ' ') THEN
                   IUNITS = 'E units Kelvins'
                ELSE
                   IUNITS(IND:) = ', E units Kelvins'
                ENDIF
             ENDIF
          ENDIF
          IF (AUNITS .EQ. ' ') THEN
             IF (UPCASE(LINE(N:), 4) .EQ. 'MOLE') THEN
                IF (UPCASE(LINE(N+4:), 1) .EQ. 'S') THEN
                   AUNITS = 'MOLE'
                   IF (IUNITS .EQ. ' ') THEN
                      IUNITS = 'A units mole-cm-sec-K'
                   ELSE
                      IUNITS(IND:) = ', A units mole-cm-sec-K'
                   ENDIF
                ELSEIF (UPCASE(LINE(N+4:), 4) .EQ. 'CULE') THEN
                   AUNITS = 'MOLC'
                   IF (IUNITS .EQ. ' ') THEN
                      IUNITS = 'A units molecules'
                   ELSE
                      IUNITS(IND:) = ', A units molecules'
                   ENDIF
                ENDIF
             ENDIF
          ENDIF
   85 CONTINUE
C
      IF (AUNITS .EQ. ' ') THEN
         AUNITS = 'MOLE'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', A units mole-cm-sec-K'
         ELSE
            IUNITS(IND:) = ' A units mole-cm-sec-K'
         ENDIF
      ENDIF
C
      IF (EUNITS .EQ. ' ') THEN
         EUNITS = 'CAL/'
         IND = ILASCH(IUNITS) + 1
         IF (IND .GT. 1) THEN
            IUNITS(IND:) = ', E units cal/mole'
         ELSE
```

```
            IUNITS(IND:) = ' E units cal/mole'
         ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------C
C
      INTEGER FUNCTION IPPLEN (LINE)
C
C  BEGIN PROLOGUE
C
C  FUNCTION IPPLEN (LINE)
C     Returns the effective length of a character string, i.e.,
C     the index of the last character before an exclamation mark (!)
C     indicating a comment.
C
C  INPUT
C     LINE  - A character string.
C
C  OUTPUT
C     IPPLEN - The effective length of the character string.
C
C  END PROLOGUE
C
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER LINE*(*)
C
      IN = IFIRCH(LINE)
      IF (IN.EQ.0 .OR. LINE(IN:IN).EQ.'!') THEN
         IPPLEN = 0
      ELSE
         IN = INDEX(LINE,'!')
         IF (IN .EQ. 0) THEN
            IPPLEN = ILASCH(LINE)
         ELSE
            IPPLEN = ILASCH(LINE(:IN-1))
         ENDIF
      ENDIF
      RETURN
      END
C
      CHARACTER*(*) FUNCTION UPCASE(ISTR, ILEN)
      CHARACTER ISTR*(*), LCASE(26)*1, UCASE(26)*1
      DATA LCASE /'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1           'n','o','p','q','r','s','t','u','v','w','x','y','z'/,
     2     UCASE /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     3           'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
C
      UPCASE = ' '
      UPCASE = ISTR(:ILEN)
      JJ = MIN (LEN(UPCASE), LEN(ISTR), ILEN)
      DO 10 J = 1, JJ
         DO 10 N = 1,26
            IF (ISTR(J:J) .EQ. LCASE(N)) UPCASE(J:J) = UCASE(N)
   10 CONTINUE
      RETURN
      END
```

```
SUBROUTINE FILEIO_CVT

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (LIN=5, LOUT=6, LOG=15, LINC=25)
      CHARACTER FILE_IN*20,FILE_OUT*20
      COMMON/FILE_IO/FILE_IN,FILE_OUT


   5  WRITE(*,10)
  10  FORMAT(' Please enter the MASTER mechanism file name : ')
      READ(*,'(A20)')FILE_IN
      OPEN(UNIT=5,FILE=FILE_IN,STATUS='OLD',ERR=30)

      GO TO 50
  30  CONTINUE
      WRITE(*,40)FILE_IN
  40  FORMAT(' Can not find file : ',A20)
      GO TO 5
  50  CONTINUE

 105  WRITE(*,110)
 110  FORMAT(' Please enter the output mechanism file name : ')
      READ(*,'(A20)')FILE_out
      OPEN(UNIT=6,FILE=FILE_OUT,STATUS='UNKNOWN',FORM='FORMATTED',
     2             ERR=130)

      GO TO 150
 130  CONTINUE
      WRITE(*,140)FILE_OUT
 140  FORMAT(' Can not find file : ',A20)
      GO TO 105
 150  CONTINUE

C      OPEN(UNIT=6,FILE='CKINTERP.OUT',STATUS='UNKNOWN')
C      OPEN(UNIT=25,FILE='CHEMKIN.LNK',STATUS='UNKNOWN',
C     1                  FORM='BINARY')


      RETURN
      END
```

D.7 CHEBYSHEV POLY FITTINGS

```
SUBROUTINE file_IO(Error)
C
      IMPLICIT NONE
      CHARACTER FILE_INP*20, FILE_LOG*20, FILE_COL*20,
     1          FILE_KIN*20, FILE_OUT*20, FILE_NAME*20, FILE_FIT*20
      INTEGER INP, LOG, COL, KIN, OUT, Istart,Idot,Ifirch, FIT
      logical Error
C      PARAMETER (MG_file=20, Max_Specy_Group=50,
C     1           Max_Element=4 )
      PARAMETER (INP=10, LOG=20, COL=30, KIN=55, FIT=95, OUT=99)
      COMMON/FILE/FILE_INP,FILE_LOG,FILE_COL,FILE_KIN,FILE_OUT,FILE_NAME
     1           ,FILE_FIT
C      INCLUDE 'THM_PAR.FI'
C
C      CHARACTER FILE_NAME*20
C      INTEGER IDOT, ISTART, IFIRCH
C      LOGICAL ERROR
C
      ERROR = .FALSE.
   5  WRITE(*,10)
  10  FORMAT(' enter the input file name : ')
      READ(*,'(A20)')FILE_INP
      OPEN(INP,FILE=FILE_INP,STATUS='OLD',ERR=30)
      GO TO 50
  30  CONTINUE
      WRITE(*,40)FILE_INP
  40  FORMAT(' Can not find file : ',A20)
      GO TO 5
  50  CONTINUE
      ISTART = IFIRCH(FILE_INP)
      IDOT = INDEX(FILE_INP,'.')
      FILE_NAME = FILE_INP(ISTART:IDOT-1)
C
      FILE_LOG = FILE_NAME
      FILE_LOG(IDOT:IDOT) = '.'
      FILE_LOG(IDOT+1:IDOT+3) = 'LOG'
C
      FILE_COL = FILE_NAME
      FILE_COL(IDOT:IDOT) = '.'
      FILE_COL(IDOT+1:IDOT+3) = 'COL'
C
      FILE_KIN = FILE_NAME
      FILE_KIN(IDOT:IDOT) = '.'
      FILE_KIN(IDOT+1:IDOT+3) = 'KIN'
C
      FILE_OUT = FILE_NAME
      FILE_OUT(IDOT:IDOT) = '.'
      FILE_OUT(IDOT+1:IDOT+3) = 'OUT'
C
      FILE_FIT = FILE_NAME
      FILE_FIT(IDOT:IDOT) = '.'
      FILE_FIT(IDOT+1:IDOT+3) = 'FIT'
C
      OPEN(LOG,FILE=FILE_LOG,STATUS='UNKNOWN',ERR=230)
      CLOSE(LOG,STATUS='DELETE')
      OPEN(LOG,FILE=FILE_LOG,STATUS='NEW')
      GO TO 250
 230  CONTINUE
      WRITE(*,240)FILE_LOG
 240  FORMAT(' File I/O Error : ',A20)
```

```
      ERROR = .TRUE.
      GO TO 550
 250  CONTINUE
C
      OPEN(COL,FILE=FILE_COL,STATUS='UNKNOWN',ERR=330)
      CLOSE(COL,STATUS='DELETE')
      OPEN(COL,FILE=FILE_COL,STATUS='NEW')
      GO TO 350
 330  CONTINUE
      WRITE(*,340)FILE_COL
 340  FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 550
 350  CONTINUE
C
      OPEN(KIN,FILE=FILE_KIN,STATUS='UNKNOWN',ERR=430)
      CLOSE(KIN,STATUS='DELETE')
      OPEN(KIN,FILE=FILE_KIN,STATUS='NEW')
      GO TO 450
 430  CONTINUE
      WRITE(*,440)FILE_KIN
 440  FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 550
 450  CONTINUE
C
      OPEN(FIT,FILE=FILE_FIT,STATUS='UNKNOWN',ERR=530)
      CLOSE(FIT,STATUS='DELETE')
      OPEN(FIT,FILE=FILE_FIT,STATUS='NEW')
C
      OPEN(OUT,FILE=FILE_OUT,STATUS='UNKNOWN',ERR=530)
      CLOSE(OUT,STATUS='DELETE')
      OPEN(OUT,FILE=FILE_OUT,STATUS='NEW')
      GO TO 550
 530  CONTINUE
      WRITE(*,540)FILE_OUT
 540  FORMAT(' File I/O Error : ',A20)
      ERROR = .TRUE.
      GO TO 550
 550  CONTINUE
C
      RETURN
      END
C
C
      FUNCTION IFIRCH(STRING)
C   BEGIN PROLOGUE  IFIRCH
C   DATE WRITTEN   850626
C   REVISION DATE  850626
C   CATEGORY NO.  M4.
C   KEYWORDS  CHARACTER STRINGS,SIGNIFICANT CHARACTERS
C   AUTHOR  CLARK,G.L.,GROUP C-3 LOS ALAMOS NAT'L LAB
C   PURPOSE  Determines first significant (non-blank) character
C            in character variable
C   DESCRIPTION
C
C-----------------------------------------------------------------------
C  IFIRCH locates the first non-blank character in a string of
C  arbitrary length.  If no characters are found, IFIRCH is set = 0.
C  When used with the companion routine ILASCH, the length of a string
C  can be determined, and/or a concatenated substring containing the
C  significant characters produced.
C-----------------------------------------------------------------------
C
C   REFERENCES  (NONE)
C   ROUTINES CALLED  (NONE)
```

```
C   END PROLOGUE IFIRCH
C*****precision > double
       IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
C*****END precision > double
C
C*****precision > single
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END precision > single
C
      CHARACTER* (*)STRING
C
C   FIRST EXECUTABLE STATEMENT IFIRCH
      NLOOP = LEN(STRING)
C
      IF (NLOOP .EQ. 0) THEN
         IFIRCH = 0
         RETURN
      ENDIF
C
      DO 100 I = 1, NLOOP
         IF (STRING(I:I) .NE. ' ') GO TO 120
100   CONTINUE
C
      IFIRCH = 0
      RETURN
120   CONTINUE
      IFIRCH = I
      END


c***FITF*************************************
      subroutine FitF(lecho)
      implicit none
      include 'cdparams.fh'
      include 'cdrange.fh'
      include 'cdffunc.fh'
      include 'cdffit.fh'
      include 'cdtemp.fh'
c    local variables
      integer iter,lWK,INFO,iWK(mxFitP),M,J,it,ip,I
      integer Iparam(7),lecho,mxIter,mxFEval,mxGEval
c      real*8 Rparam(7),Xguess(mxFitP),Xscale(mxFitP),Fvalue,
c     2   X(mxFitP),WK(mxFitP*(mxFitP+8)),Fscale
c
c     This block has been changed to incorporated with HYBRD1
c
      real*8 Rparam(7),Xguess(mxFitP),Xscale(mxFitP),
     2   Fvalue(mxTPts * mxPPts),
     3   X(mxFitP),Fscale,TOL,FNORM,ENORM,
     4   WK(mxFitP*mxTPts * mxPPts + 5*mxFitP + mxTPts * mxPPts + 20)
      real*8 T_cheb,P_cheb,RK_tempLog,PhiCh
      external FCN
c    end declarations
c    get initial guess for solution; scaling info
C
C       call initX(Nfit,Xguess,Xscale,Fscale,funstr,Fopt)
C
C     CHANGED 02/95
      iTempWell = iWell
      iTempProd = iProd
      if (Fopt.eq.'CHEB') Nfit = N_cheb*M_cheb
C     write(*,*)Nfit,N_cheb,M_cheb
      call initX(Nfit,X,Xscale,Fscale,funstr,Fopt)
C
C
```

```
c     intialize minimizer
c     flag to reset parameters
       Iparam(1) = 1
c        call DU4INF(Iparam,Rparam)
c     change some parameters for minimizer
        call resetP(Iparam,Rparam,mxIter,mxFEval,mxGEval)
c     U2INF is the IMSL minimization fitting routine
c     NOTE U2INF is really for single precision but we
c     always compile with opt CRAY real*8 ->> single precision
c     (changed to du2inf for SGI  <ayc 3/94>)

c
c      This block has been changed by W. Ing
c      sub hybrid1 from MINPACK was incorporated to replace
c      IMSL minimization routine U2INF.
c
c       call DU2INF(FCN,Nfit,Xguess,Xscale,Fscale,Iparam,Rparam,
c      2   X,Fvalue,WK)
c
c       lWK = mxFitP*(3*mxFitP+13)
       TOL = 1.0D-10
       M = ntemps*npres
       lwk = mxFitP*M + 5*mxFitP + M + 20
c       Fvalue(1) = 1.d80
c       call HYBRD1(FCN,Nfit,X,Fvalue,TOL,INFO,WK,lWK)
       call LMDIF1(FCN,M,Nfit,X,Fvalue,TOL,INFO,iWK,WK,lWK)
       FNORM=ENORM(M,Fvalue)
c       WRITE(*,1000)FNORM,INFO,(X(J),J=1,Nfit)
c1000   FORMAT(5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7//
c      2       5X,15H EXIT PARAMETER,16X,I10//
c      3       5X,27H FINAL APPROXIMATE SOLUTION,//5X,4D15.7)

C       write(*,*)'...after minizator...'

C
C
C
c     test for convergence
C       if (Iparam(3).ge.mxIter) write(lecho,100) mxIter
C100    format(/,' ERROR: No convergence in ',i5,' iterations.')
C       if (Iparam(4).ge.mxFEval) write(lecho,110) mxFEval
C110    format(/,' ERROR: No convergence in ',i5,
C      2    ' function evaluations.')
C            if (Iparam(5).ge.mxGEval) write(lecho,120) mxGEval
C120    format(/,' ERROR: No convergence in ',i5,
C      2    ' gradient evaluations.')
C
       if (INFO.eq.0) write(lecho,100)
100    format(/,' ERROR: Improper input parameters ')
       if (INFO.eq.4) write(lecho,101)
101    format(/,' WARNING: Fvalue is orthogonal to the columns of the Jac
      2obian to machine precision.')
       if (INFO.eq.5) write(lecho,102)
102    format(/,' WARNING: Number of calls to FCN has reached or exceeded
      2 200*Nfit')
       if (INFO.eq.6) write(lecho,103)
103      format(/,' WARNING: TOL is too small.  No further reduction in t
      2he sum of squares in possible')
       if (INFO.eq.7) write(lecho,104)
104      format(/,' WARNING: TOL is too small.  No further improvment in
      2the approximate solution X is possible')
       if (INFO.eq.1) write(lecho,105)
105    format(/,' F Fitting : ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
      2 IN THE SUM OF SQUARES IS AT MOST TOL.')
       if (INFO.eq.2) write(lecho,106)
106    format(/,' F Fitting : ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
```

```
     2 BETWEEN X AND THE SOLUTION IS AT MOST TOL.')
      if (INFO.eq.3) write(lecho,107)
107   format(/,' F Fitting : ALGORITHM ESTIMATES THAT THE RELATIVE ERROR
     2 both IN THE SUM OF SQUARES IS AT MOST TOL AND BETWEEN X AND THE
     3 SOLUTION IS AT MOST TOL.')
C
C      write(*,*)'...before out...'
c      write(*,*)(X(J),J=1,Nfit)
c     fill out fit parameters into common variables
C      write(*,*)'....before FitVec filled out...'
C      write(*,*)'....Nfit = '
C      write(*,*)Nfit
C      write(*,*)'....INFO = '
C      write(*,*)INFO
      do 150 iter = 1,Nfit
         FitVec(iter) = X(iter)
150    continue
      if (Fopt.eq.'CHEB') then
         do 153 it = 1, ntemps
            do 153 ip = 1, npres
      T_cheb = (2.*(1./temp(it))-(1/Tmin)-(1/Tmax)) /
     2         ((1/Tmax)-(1/Tmin))
      P_cheb = (2.*dlog10(pres(1,ip))-dlog10(Pmin)-dlog10(Pmax))/
     2         (dlog10(Pmax)-dlog10(Pmin))
               RK_tempLog = 0.0
               DO 152 I = 1, N_Cheb
                  DO 151 J = 1, M_Cheb
                     RK_tempLog = RK_tempLog + X((I-1)*M_cheb+J) *
     2                           PhiCh(I,T_cheb)*PhiCh(J,P_cheb)
151                CONTINUE
152             CONTINUE
c*************************write k & fit cheb K compare output
153      continue
      endif
C      write(*,*)'....after FitVec filled out...'
C      write(*,*)'....X1-3 = '
C      write(*,*)FitVec(1),FitVec(2),FitVec(3)
C      write(*,*)'....lecho, iWell, iProd = '
C      write(*,*)lecho,iWell,iProd
      return
      end
c***RESETP************************************
      subroutine resetP(iparam,Rparam,mxI,mxF,mxG)
c     sets up parameters for minimizer - see IMSL doc.
      parameter (mxIter = 1000, mxFEval = 4000, mxGEval = 4000)
      integer Iparam(7),mxI,mxF,mxG
      real*8 Rparam(7)
c     all these things have defaults, but some
c     we reset
c     iparam(1) was control flag which we already invoked
c     iparam(2) is # of machine-dependent good digits
c     #3 is max # of iterations (def: 100)
      Iparam(3) = mxIter
      mxI = mxIter
c     #4 is max # of function evaluations (def: 400)
      Iparam(4) = mxFEval
        mxF = mxFEval
c     #5 is max # of gradient evaluations (def: 400)
      Iparam(5) = mxGEval
        mxG = mxGEval
c     iparam(6) def = 0 - init Hessian to Identity
c     iparam(7) is not used
c     Rparam defaults look ok - based on machine
c     dependent properties - i don't think we can
c     do better - so we leave alone this time
      return
```

```
      end
c***FCN*************************************
c     subroutine to define objective function
c     see IMSL documentation
c
c     02/95 changed to use HYBRD1
c
c      subroutine FCN(N, X, F)
c
      subroutine FCN(M, N, X, F, IFLAG)
      implicit none
      include 'cdparams.fh'
      include 'cdrange.fh'
      include 'cdffunc.fh'
      include 'cdffit.fh'
      include 'cdtemp.fh'
      include 'cdrates.fh'
c     local variables
      integer N,IFLAG,M
      integer it, ip, ii, ntt
      real*8 X(N),error,cumerr,Ffit,F(N)
      real*8 getFfit
c     end declarations
c      cumerr = 0.d0
c         write(*,*)'....x1..x2..x3..'
c         write(*,*)x(1),x(2),x(3)
      do 200 it = 1,ntemps
        cumerr = 0.d0
        do 100 ip = 1,npres
           Ffit = getFfit(N,X,temp(it),Pr(it,ip),Fopt,iTempWell,
     2                  iTempProd,it,ip)
           if (Fopt.eq."CHEB") then
              error = DABS((Ffit-DLOG10(RK(iTempwell,iTempprod,it,ip))))
c              /
c     2                          RK(iTempwell,iTempprod,it,ip) )
c            error = (dlog10(Fcalc(it,ip)) - dlog10(Ffit))**2
           else
c
c       ADDED FOR DEBUG W.ING
c
c           open(unit=15,status='unknown',file='fort.15')
           if (Ffit.eq.0) Ffit=1.0d-10
           if (Fcalc(it,ip).eq.0) Fcalc(it,ip)=1.0d-10
           error = DABS((dlog10(Fcalc(it,ip)) - dlog10(Ffit)))
           endif
           ntt = (it-1)*npres + ip
c          write(15,*)'......it...ip....erroe...logRK........logRK_cal'
c          write(15,92)it,ip,error,DLOG(RK(iTempwell,iTempprod,it,ip))
c     2                          ,Ffit
c92        format('  ',I2,'  ',I2,'  ',F8.4,'  ',F8.4,'  ',F8.4)
           F(ntt) = error
c            cumerr = cumerr + error
c            WRITE(*,*)temp(it),pr(it,ip),error,cumerr
100      continue
c         F(it) = cumerr
c         WRITE(*,*)F(it)
200   continue
c
c     test
c
C      do 200 ii = 1, N
C         F(ii) = 1.0d-6
C200   continue
c      write(*,*)(X(it),it=1,N)
c      F(1) = cumerr
      return
```

```
      end
c***FITFOUT**************************************
      subroutine FitFout(option,lfout1,lfout2,iwell,iprod)
c     fill out output files with Fcalc vs T, P
      implicit none
      include 'cdparams.fh'
      include 'cdwell0.fh'
      include 'cdlabels.fh'
      include 'cdrange.fh'
      include 'cdffunc.fh'
      include 'cdffit.fh'
c     local variables
      character option*8,RCname*20
      integer it,ip,iwell,iprod,iter
      integer lfout1,lfout2,itEmax,ipEmax,nFlt90,nEgt10
      real*8 Ffit,error,errormx,toterr
      real*8 getFfit
      character tab
c     end declarations
      tab = char(9)
      RCname = PDname(inpwell,inpchan)
      write(lfout1,50) option,inpwell,iwell,iprod,
     2   RCname,PDname(iwell,iprod)
50    format(//,1x,a,'(',i2,') -> ',i2,':',i2,'  ',a,' = ',a)
      write(lfout1,150) tab,tab,tab,tab,tab,tab,tab,tab
150   format(/,1x,' T (K) ',a,' 1000/T',a,' log P ',a,' log Pr',a,
     2   ' Fcalc ',a,'-lnFcal',a,' F fit ',a,'-lnFfit',a,' %error')
      nFlt90 = 0
      nEgt10 = 0
      errormx = 0.d0
      toterr = 0.d0
      do 350 it = 1,ntemps
        do 300 ip = 1,npres
          Ffit = getFfit(Nfit,FitVec,temp(it),Pr(it,ip),Fopt,
     2                   iWell,iProd,it,ip)
          if (Ffit.lt.9.d-1) nFlt90 = nFlt90 + 1
             error = (Fcalc(it,ip)-Ffit) / dmin1(Fcalc(it,ip),Ffit)
          if (dabs(error).gt.1.d-1) nEgt10 = nEgt10 + 1
                toterr = toterr + dabs(error)
                if (dabs(error).gt.errormx) then
             errormx = dabs(error)
             itEmax = it
             ipEmax = ip
          endif
c
          if (Ffit.eq.0) Ffit=1.0d-1
          if (Pr(it,ip).eq.0) Pr(it,ip)=1.0d-1
          if (pres(it,ip).eq.0) pres(it,ip)=1.0d-1
          if (Fcalc(it,ip).eq.0) Fcalc(it,ip)=1.0d-1
c
c
c
c
c
c           write(lfout1,280) temp(it),tab,1000.d0/temp(it),
c     2          tab,dlog10(pres(it,ip)),tab,dlog10(Pr(it,ip)),
c     3          tab,Fcalc(it,ip),tab,-dlog(Fcalc(it,ip)),
c     4          tab,Ffit,tab,-dlog(Ffit),tab,error*100.d0
c280        format(' ',f7.2,a,f7.4,a,f7.3,a,f7.3,a,f7.4,a,f7.3,
c     2          a,f7.4,a,f7.3,a,f7.2)
          write(lfout1,280) temp(it),tab,1000.d0/temp(it),
     2          tab,dlog10(pres(it,ip)),tab,dlog10(Pr(it,ip)),
     3          tab,Fcalc(it,ip),tab,(Fcalc(it,ip)),
     4          tab,Ffit,tab,(Ffit),tab,error*100.d0
280       format(' ',f7.2,a,f7.4,a,f7.3,a,f7.3,a,f7.4,a,f7.3,
     2          a,f7.4,a,f7.3,a,f10.2)
```

```
300       continue
350    continue
c     now write summary output to lfout2
       write(lfout2,50) option,inpwell,iwell,iprod,
      2    RCname,PDname(iwell,iprod)
       write(lfout2,400) funstr
400    format(/,' Fit: ',a,/)
       do 430 iter = 1,Nfit
         write(lfout2,410) iter,FitVec(iter)
410        format(' Param(',i2,'): ',1pe14.7)
430    continue
       write(lfout2,450) npres*ntemps,nFlt90,nEgt10,
      2    100.d0*toterr/(npres*ntemps)
450    format(/,' Total points: ',i3,'  # F''s < 0.9: ',i3,
      2    '  # err > 10%: ',i3,'  avg err%: ',f5.2)
       Ffit = getFfit(Nfit,FitVec,temp(itEmax),
      2    Pr(itEmax,ipEmax),Fopt,iWell,iProd,itEmax,ipEmax)
C
C
           if (pres(itEmax,ipEmax).eq.0) pres(itEmax,ipEmax)=1.0d-1
C
C
c       write(lfout2,460) errormx*100.d0,temp(itEmax),
c      2    dlog10(pres(itEmax,ipEmax)),Fcalc(itEmax,ipEmax),Ffit
c460    format(/,' Worst error%: ',f5.1'  T: ',f6.1,'  logP: ',f5.2,
c      2    '  Fcalc: ',f7.4,'  Ffit: ',f7.4)
       write(lfout2,460) errormx*100.d0,temp(itEmax),
      2    (pres(itEmax,ipEmax)),Fcalc(itEmax,ipEmax),Ffit
460    format(/,' Worst error%: ',f8.1'  T: ',f6.1,'  logP: ',f5.2,
      2    '  Fcalc: ',f10.4,'  Ffit: ',f10.4)
          return
       end
c
c*************************************************************************
c***FCHEBOUT*************************************
       subroutine FChebOut(option,lfout1,lfout2,lfout3,iwell,iprod)
c     fill out output files with Fcalc vs T, P
       implicit none
       include 'cdparams.fh'
       include 'cdwell0.fh'
       include 'cdlabels.fh'
       include 'cdrange.fh'
       include 'cdffunc.fh'
       include 'cdffit.fh'
       include 'cdrates.fh'
       include 'cdkfit.fh'
       include 'cdcolls.fh'
c     local variables
       character option*8,RCname*20,Pname*20
       integer it,ip,iwell,iprod,iter,I,J,irest
       integer lfout1,lfout2,itEmax,ipEmax,nFlt90,nEgt10,lfout3
       real*8 Ffit,error,errormx,toterr
       real*8 getFfit,RK_tempLog,RK_temp,PhiCh,a,rn,ea
       character tab
       integer*2 year,month,day
       CHARACTER FILE_INP*20, FILE_LOG*20, FILE_COL*20,
      1          FILE_KIN*20, FILE_OUT*20, FILE_NAME*20
       COMMON/FILE/FILE_INP,FILE_LOG,FILE_COL,FILE_KIN,FILE_OUT,FILE_NAME
c     end declarations
       tab = char(9)
       RCname = PDname(inpwell,inpchan)
       write(lfout1,50) option,inpwell,iwell,iprod,
      2    RCname,PDname(iwell,iprod)
50     format(//,1x,a,'(',i2,') -> ',i2,':',i2,'  ',a,' = ',a)
       write(lfout1,150) tab,tab,tab,tab,tab,tab,tab,tab
```

```
150   format(/,1x,' T (K) ',a,' 1000/T',a,' log P ',a,' log Pr',a,
     2    ' Kcalc ',a,'logKcal',a,' K fit ',a,'logKfit',a,' %error')
      nFlt90 = 0
      nEgt10 = 0
      errormx = 0.d0
      toterr = 0.d0
      do 350 it = 1,ntemps
        do 300 ip = 1,npres
          RK_tempLog = getFfit(Nfit,FitVec,temp(it),Pr(it,ip),Fopt,
     2                 iWell,iProd,it,ip)
          RK_temp = 10**(RK_tempLog)
          if (Ffit.lt.9.d-1) nFlt90 = nFlt90 + 1
             error = (RK_temp-RK(iwell,iprod,it,ip)) /
     2              RK(iwell,iprod,it,ip)
          if (dabs(error).gt.1.d-1) nEgt10 = nEgt10 + 1
          toterr = toterr + dabs(error)
          if (dabs(error).gt.errormx) then
             errormx = dabs(error)
             itEmax = it
             ipEmax = ip
          endif
c
          if (Ffit.eq.0) Ffit=1.0d-1
          if (Pr(it,ip).eq.0) Pr(it,ip)=1.0d-1
          if (pres(it,ip).eq.0) pres(it,ip)=1.0d-1
          if (Fcalc(it,ip).eq.0) Fcalc(it,ip)=1.0d-1
c
c           write(1fout1,280) temp(it),tab,1000.d0/temp(it),
c     2        tab,dlog10(pres(it,ip)),tab,dlog10(Pr(it,ip)),
c     3        tab,Fcalc(it,ip),tab,-dlog(Fcalc(it,ip)),
c     4        tab,Ffit,tab,-dlog(Ffit),tab,error*100.d0
c280        format(' ',f7.2,a,f7.4,a,f7.3,a,f7.3,a,f7.4,a,f7.3,
c     2        a,f7.4,a,f7.3,a,f7.2)
c
          write(1fout1,280) temp(it),tab,1000.d0/temp(it),
     2        tab,dlog10(pres(it,ip)),tab,dlog10(Pr(it,ip)),
     3        tab,RK(iwell,iprod,it,ip),tab,
     4        DLOG10(RK(iwell,iprod,it,ip)),
     4        tab,RK_temp,tab,DLOG10(RK_temp),tab,error*100.d0
280       format(' ',f7.2,a,f7.4,a,f7.3,a,f7.3,a,E12.4,a,f7.3,
     2        a,E12.4,a,f7.3,a,F8.3)

300     continue
350   continue
c     now write summary output to 1fout2
      write(1fout2,50) option,inpwell,iwell,iprod,
     2   RCname,PDname(iwell,iprod)
      write(1fout2,400) funstr
400   format(/,' Fit: ',a,/)
      do 430 iter = 1,Nfit
        write(1fout2,410) iter,FitVec(iter)
410     format(' Param(',i2,'): ',1pe14.7)
430   continue
      write(1fout2,450) npres*ntemps,nFlt90,nEgt10,
     2   100.d0*toterr/(npres*ntemps)
450   format(/,' Total points: ',i3,'  # F''s < 0.9: ',i3,
     2    '  # err > 10%: ',i3,'  avg err%: ',f5.2)
      Ffit = getFfit(Nfit,FitVec,temp(itEmax),
     2   Pr(itEmax,ipEmax),Fopt,iWell,iProd,itEmax,ipEmax)
C
C
        if (pres(itEmax,ipEmax).eq.0) pres(itEmax,ipEmax)=1.0d-1
C
C
c      write(1fout2,460) errormx*100.d0,temp(itEmax),
c     2   dlog10(pres(itEmax,ipEmax)),Fcalc(itEmax,ipEmax),Ffit
```

```
c460   format(/,' Worst error%: ',f5.1'  T: ',f6.1,'  logP: ',f5.2,
c     2    '  Fcalc: ',f7.4,'  Ffit: ',f7.4)
      write(lfout2,460) errormx*100.d0,temp(itEmax),
     2   (pres(itEmax,ipEmax)),RK(iwell,iprod,itEmax,ipEmax),10**fFIT
460   format(/,' Worst error%: ',f8.1'  T: ',f6.1,'     P: ',f5.2,
     2    '  Kcalc: ',e11.4,'  Kfit: ',e11.4)
C
c     now write summary output to lfout3
C
      call concat(PDname(inpwell,inpchan),' (+M)',RCname)
      call concat(PDname(iwell,iprod),' (+M)',Pname)
c      RCname = PDname(inpwell,inpchan)
      a = 1.0
      rn = 0.0
      ea = 0.0
      call getdat(year,month,day)
      year = year - 1900
      irest = MOD(Nfit-4,5)
      write(lfout3,500) RCname,Pname,a,rn,ea,FILE_NAME,month,year
  500 format(a,' <=> ',a,1pe9.2,'  ',0pf7.3,'  ',f7.0,1x,
     2   '! ',A8,1x,I2,'/',I2)
      write(lfout3,510)
  510 format('    LOW / 1.0 0.0 0.0 /')
      write(lfout3,520)n_cheb,m_cheb,(FitVec(iter),iter=1,4)
  520 format('    CHEB/ ',I1,3x,I1,5x,1p4(1x,e11.4),'/')
      write(lfout3,530)(FitVec(iter),iter=5,Nfit-irest)
  530 format(1('    CHEB/',1p5(1x,e11.4),'/'))
      if (irest.eq.1) then
          write(lfout3,540)(FitVec(iter),iter=Nfit-irest+1,Nfit)
  540     format('    CHEB/',1p1x,e11.4,'/')
      elseif (irest.eq.2) then
          write(lfout3,550)(FitVec(iter),iter=Nfit-irest+1,Nfit)
  550     format('    CHEB/',1p2(1x,e11.4),'/')
      elseif (irest.eq.3) then
          write(lfout3,560)(FitVec(iter),iter=Nfit-irest+1,Nfit)
  560     format('    CHEB/',1p3(1x,e11.4),'/')
      elseif (irest.eq.4) then
          write(lfout3,570)(FitVec(iter),iter=Nfit-irest+1,Nfit)
  570     format('    CHEB/',1p4(1x,e11.4),'/')
      endif
c      do 430 iter = 1,Nfit
c          write(lfout3,600) iter,FitVec(iter)
c  600     format(' Param(',i2,'): ',1pe14.7)
c430   continue
cCH3 + OH (+M) = CH2O + H2 (+M)      1.00E+00    0.00         0. !4.04E+12
-0.51     2839. !CHEM241
c   LOW / 1.0 0.0 0.0 /
c CHEB/7 3 1.0044196E+01 -7.3333131E-01 -2.9292355E-01/
c CHEB/ 5.4861E-01   7.0805E-01   2.3241E-01   5.8694E-02   9.3565E-02/
c CHEB/8.3981941E-02 -2.2423983E-02 -3.0251313E-02 4.4687742E-04 -
1.7928843E-02/
c CHEB/-2.8764778E-02 -1.5347567E-02 -7.4310585E-03 -1.3008562E-02 -
9.7005805E-03/
c CHEB/-1.8067830E-03 -3.2013939E-03 -3.1585273E-03/
c
      return
      end


c***INITX**********************************
c     sub to return initial-guess solution and scaling parameters
      subroutine initX(Npar,X0,Xscale,Fscale,IDstr,option)
      implicit none
c     local variables
      integer Npar
      real*8 X0(*),Xscale(*),Fscale
```

```
      character IDstr*(*),option*8
c     end declarations
c     IMPORTANT: these options are set in getparams
      if (option.eq.'Troe') then
        call initTroe(Npar,X0,Xscale,Fscale,IDstr)
           else if (option.eq.'CHEB') then
        call initCheb(Npar,X0,Xscale,Fscale,IDstr)
           else if (option.eq.'SRI') then
        call initSRI(Npar,X0,Xscale,Fscale,IDstr)
           else if (option.eq.'FRENK') then
        call initFreK(Npar,X0,Xscale,Fscale,IDstr)
           else
        call initDEF(Npar,X0,Xscale,Fscale,IDstr)
                 endif
           return
      end
c***GETFFIT*************************************
c     routine to return fitted F
      function getFfit(Npar,X,T,P,option,iWell,iProd,it,ip)
      implicit none
c     local variables
      integer Npar
      real*8 X(*),T,P,it,ip,iWell,iProd
      real*8 getFfit,getTroe,getSRI,getDef,getFreK,getCheb
      character option*8
c     end declarations
      if (option.eq.'Troe') then
        getFfit = getTroe(Npar,X,T,P)
           else if (option.eq.'CHEB') then
        getFfit = getCheb(Npar,X,T,P,iWell,iProd,it,ip)
           else if (option.eq.'SRI') then
        getFfit = getSRI(Npar,X,T,P)
           else if (option.eq.'FRENK') then
        getFfit = getFreK(Npar,X,T,P,iWell,iProd,it,ip)
           else
        getFfit = getDef(Npar,X,T,P)
      endif
      return
      end
c***INITCHEB************************************
c     companion routine to getFreK for initializing variables
      subroutine initCheb(Npar,X0,Xscale,Fscale,IDstr)
      implicit none
c     local variables
      integer iter
      integer Npar,Mpar
      real*8 X0(*),Xscale(*),Fscale
      character IDstr*(*)
c     end declarations
c       Npar = 3
c     no better idea so set Xscale to 1 - see IMSL description;
c     set all x's to order 1 (we define T's as x's * 1000)
c     enter FIT function ID string here
      IDstr = 'M,N-Parameter Chebyshev Fit in T/1000 (w/atan)'
      do 10 iter = 1,Npar
        Xscale(iter) = 1.d0
        X0(iter) = 5.d0
10    continue
      Fscale = 1.d0
c     Now we reset the a factor
c       X0(3) = 2.0
      return
      end
c***GETCHEB*************************************
c     function to return defaulf "a" fitting function
      function getCheb(Npar,X,T,P,iWell,iProd,it,ip)
```

```
      implicit none
      include 'cdparams.fh'
      include 'cdrange.fh'
      include 'cdrates.fh'
      include 'cdlimfit.fh'
      include 'cdffunc.fh'
c     local variables
      integer Npar,it,ip,iWell,iProd,I,J
      real*8 X(*),T,P,a
      real*8 getCheb,kFit,PhiCh,T_cheb,P_cheb,RK_tempLog
c     end declarations
C       a = X(1)*dexp(-((dlog10(P)-X(2))/X(3))**2)-(1/nPr(iwell,iprod))
C       kFit = (RK(iwell,iprod,it,0)**a + RK(iwell,iprod,it,npres+1)**a)
C      2       **(1/a)
       T_cheb = (2.*(1./temp(it))-(1/Tmin)-(1/Tmax)) /
      2         ((1/Tmax)-(1/Tmin))
       P_cheb = (2.*dlog10(pres(1,ip))-dlog10(Pmin)-dlog10(Pmax))/
      2         (dlog10(Pmax)-dlog10(Pmin))
       RK_tempLog = 0.0
       DO 20 I = 1, N_Cheb
          DO 10 J = 1, M_Cheb
c            write(*,*)RK_tempLog,((I-1)*M_cheb+J),X((I-1)*M_cheb+J),
c     2                  PhiCh(I,T_cheb),PhiCh(J,P_cheb)
            RK_tempLog = RK_tempLog
     2             + X((I-1)*M_cheb+J)*PhiCh(I,T_cheb)*PhiCh(J,P_cheb)
10        CONTINUE
20     CONTINUE
c       RK_Cheb(iwell,iprod,it,ip)=10**RK_tempLog

c       write(*,*)'...it....ip....a...k...kfit...'
c       write(*,*)it,ip,a,RK(iwell,iprod,it,ip),kFit
c       write(*,*)'...it...ip...'
c       write(*,*)it,ip
c       getCheb = RK_Cheb(iwell,iprod,it,ip)
       getCheb = RK_tempLog
c       write(*,*)'...logk0**a+kinf**a-logk**a........getCheb = ...'
c       write(*,*)getCheb
       return
       end
c***PHICH*************************************
C
       FUNCTION PhiCh(i,x)
       implicit none
c      local variables
       integer i
       real*8 x,PhiCh
c
       PhiCh = DCOS((i-1)*DACOS(x))
       return
       end
c***INITFREK**********************************
c     companion routine to getFreK for initializing variables
       subroutine initFreK(Npar,X0,Xscale,Fscale,IDstr)
       implicit none
c     local variables
       integer iter
       integer Npar
       real*8 X0(*),Xscale(*),Fscale
       character IDstr*(*)
c     end declarations
       Npar = 3
c     no better idea so set Xscale to 1 - see IMSL description;
c     set all x's to order 1 (we define T's as x's * 1000)
c     enter FIT function ID string here
       IDstr = '3-Parameter Frenklach Fit in T/1000 (w/atan)'
       do 10 iter = 1,Npar
```

```
        Xscale(iter) = 1.d1
        X0(iter) = 5.d-1
10     continue
       Fscale = 1.d0
c     Now we reset the a factor
       X0(3) = 2.0
       return
       end
c***GETFREK*************************************
c     function to return defaulf "a" fitting function
       function getFreK(Npar,X,T,P,iWell,iProd,it,ip)
       implicit none
       include 'cdparams.fh'
       include 'cdrange.fh'
       include 'cdrates.fh'
       include 'cdlimfit.fh'
       include 'cdffunc.fh'
c     local variables
       integer Npar,it,ip,iWell,iProd
       real*8 X(*),T,P,a
       real*8 getFreK,kFit
c     end declarations
       a = X(1)*dexp(-((dlog10(P)-X(2))/X(3))**2)-(1/nPr(iwell,iprod))
       kFit = (RK(iwell,iprod,it,0)**a + RK(iwell,iprod,it,npres+1)**a)
      2       **(1/a)
c       write(*,*)'...it....ip....a...k...kfit...'
c       write(*,*)it,ip,a,RK(iwell,iprod,it,ip),kFit
c       write(*,*)'...it...ip...'
c       write(*,*)it,ip
       getFreK = dlog10(RK(iwell,iprod,it,0)**a
      2               + RK(iwell,iprod,it,npres+1)**a)
      3        - dlog10(RK(iwell,iprod,it,ip)**a)
c       write(*,*)'...logk0**a+kinf**a-logk**a........getFreK = ...'
c       write(*,*)getFreK
       return
       end
c***INITDEF*************************************
c     companion routine to getDef for initializing variables
       subroutine initDef(Npar,X0,Xscale,Fscale,IDstr)
       implicit none
c     local variables
       integer iter
       integer Npar,iamp,ioff,iwidth,iasym
       real*8 X0(*),Xscale(*),Fscale
       character IDstr*(*)
c     end declarationsc    make consistent with getDef
        data iamp,iwidth,iasym,ioff/1,6,9,12/
       Npar = 12
c     no better idea so set Xscale to 1 - see IMSL description;
c     also zero out all X0's before setting non-zero values
c     ok - let's try not quite zeroing out (we want init vector
c     all the same order of magnitude)
c     enter FIT function ID string here
       IDstr = 'G (12): 5 amp, 3 width, 3 asym, 1 shift, exp = 1/2'
        do 10 iter = 1,Npar
        Xscale(iter) = 1.d0
        X0(iter) = 1.d-1
10     continue
       Fscale = 1.d0
c     Now we reset the amplitude and width
       X0(iamp) = 1.d0
       X0(iwidth) = 2.d0
       return
       end
c***GETDEF*************************************
c     function to return defaulf F fitting function
```

```
      function getDef(Npar,X,T,P)
      implicit none
c     local variables
      integer Npar,iamp,ioff,iwidth,iasym
      real*8 X(*),T,P,FcentLn,FLShape,Plgoff,Width,asym,Plg,
     2   arg,Tnm
      real*8 getDef
c     end declarations
          data iamp,iwidth,iasym,ioff/1,6,9,12/
c     IMPORTANT: data statement must be consistent with initDef
c               IDstr set in initDef
c     we invent new scaled T
      Tnm = (1.d-3*T)**0.5
c     use base 10 log for P
      Plg = dlog10(P)
c     a: we use first 5 parameters for amplitude function
c     and we try fitting to cubic in 1000./T
          FcentLn = X(iamp+ ) X(iamp+1)/(Tnm) + X(iamp+2)/(Tnm*Tnm)
     2   + X(iamp+3)/(Tnm*Tnm*Tnm) + X(iamp+4)/(Tnm*Tnm*Tnm*Tnm)
c     the rest are lineshape parameters:
c     b: we use 3 parameters for Half-width
      Width = X(iwidth) + X(iwidth+1)/(Tnm) + X(iwidth+2)/(Tnm*Tnm)
c     c: we use 3 parameters for asymmetry parameter
          asym = X(iasym) + X(iasym+1)/(Tnm) + X(iasym+2)/(Tnm*Tnm)
c     d: we use 1 parameter for offset from 0
      Plgoff =  X(ioff)
c     now the lineshape function
c     Lorentzian: we don't use it but we keep it just in case
c      FLShape = 1.d0 /
c     2    ( 1.d0 + ((Plg + Plgoff)/(Width + asym*Plg))**2 )
c     Gaussian:
      FLShape = dexp( -((Plg + Plgoff)/(Width + asym*Plg))**2 )
c     finally return the solution; but constrain argument
c     to exponential to avoid overflows
            arg = -FLShape*FcentLn
      if (arg.gt.100.d0) arg = 100.d0
      if (arg.lt.-100.d0) arg = -100.d0
      getDef = dexp(arg)
      return
      end
c***INITTROE*************************************
c     companion routine to getTroe for initializing variables
      subroutine initTroe(Npar,X0,Xscale,Fscale,IDstr)
      implicit none
c     local variables
      integer iter
      integer Npar
      real*8 X0(*),Xscale(*),Fscale
      character IDstr*(*)
c     end declarations
      Npar = 4
c     no better idea so set Xscale to 1 - see IMSL description;
c     set all x's to order 1 (we define T's as x's * 1000)
c     enter FIT function ID string here
      IDstr = '4-Parameter Troe Fit in T/1000 (w/atan)'
              do 10 iter = 1,Npar
        Xscale(iter) = 1.d0
        X0(iter) = 5.d-1
10    continue
      Fscale = 1.d0
c     Now we reset the a factor
      X0(1) = 4.d-1
      return
      end
c***GETTROE**************************************
c     function to return defaulf F fitting function
```

```
      function getTroe(Npar,X,T,P)
      implicit none
c     local variables
      integer Npar,icoef
      real*8 X(*),T,P,TStar(3),a,Fcent,
     2   FLShape,Rn,Rc,Rd,Plg
      real*8 getTroe
c     end declarations
         data Rd /0.14/
c     we try atan function to constrain a between 0 and 1
      a = datan(X(1))/3.14159265359d0 + .5d0
c     we wish to constrain T's to be positive
      do 10 icoef = 1,3
        Tstar(icoef) = 1.d3*dabs(X(icoef+1))
10    continue
c     use base 10 log for P
      Plg = dlog10(P)
c     this is right out of the Chemkin II manual, pg 23
          Fcent = (1.d0 - a)*dexp(-T/TStar(3)) + a*dexp(-T/TStar(1))
     2   + dexp(-TStar(2)/T)
          Rc = -0.4d0 - 0.67d0*dlog10(Fcent)
      Rn = 0.75 - 1.27*dlog10(Fcent)
          FLShape = 1.d0 /
     2   ( 1.d0 + ( (Plg + Rc)/ (Rn - Rd*(Plg + Rc)) )**2 )
      getTroe = Fcent**(FLShape)
          return
      end
c***INITSRI*************************************
c     companion routine to getSRI for initializing variables
      subroutine initSRI(Npar,X0,Xscale,Fscale,IDstr)
      implicit none
c     local variables
      integer iter
      integer Npar
      real*8 X0(*),Xscale(*),Fscale
      character IDstr*(*)
c     end declarations
      Npar = 3
c     no better idea so set Xscale to 1 - see IMSL description;
c     enter FIT function ID string here
      IDstr = '3-Parameter SRI Fit in T/1000'
               do 10 iter = 1,Npar
        Xscale(iter) = 1.d0
        X0(iter) = 1.d0
10    continue
      Fscale = 1.d0
c     Now we reset the a factor
      X0(1) = 4.d-1
      return
      end
c***GETSRI**************************************
c     function to return defaulf F fitting function
      function getSRI(Npar,X,T,P)
      implicit none
c     local variables
      integer Npar
      real*8 X(Npar),T,P,a,b,c,Fcent,FLShape,Plg
      real*8 getSRI
c     end declarations
c     (tried an equiv statement here, but compiler didn't
c     like it - probably because of passed variable)
      a = dabs(X(1))
      b = 1.d3*dabs(X(2))
      c = 1.d3*dabs(X(3))
c     use base 10 log for P
      Plg = dlog10(P)
```

```
c     this is right out of the Chemkin II manual, pg 24 -
c     but we have not included the optional d and e factors,
c     because we don't see that kind of limiting behavior
          Fcent = a*dexp(-b/T) + dexp(-T/c)
          FLShape = 1.d0 / ( 1.d0 + Plg**2 )
       getSRI = Fcent**(FLShape)
         return
       end
c  we tried these subs but we don't use them anymore
c**BOUNCR**************************************
c     func to reflect fitting parameter to right of of boundary
       function bouncR(xval,xlim,xzone)
       implicit none
       real*8 xval,xlim,xzone
       real*8 bouncR
c     bouncR is the reflection of xval to the other side
c     of xlim but within the region defined by xzone
c     xzone should be small compared to expected magnitude of
c     xval (because legitimate xval's could be stuck in the
c     zone); still xzone should be large enough that things
c     put there don't simple vanish into the noise
c     the test condition (xval < xlim) is assumed
       bouncR = xlim + xzone/(1.d0 + (xlim - xval))
            return
       end
c**BOUNCL**************************************
c     func to reflect fitting parameter to left of of boundary
       function bouncL(xval,xlim,xzone)
       implicit none
           real*8 xval,xlim,xzone
       real*8 bouncL
c     bouncL is the reflection of xval to the other side
c     of xlim but within the region defined by xzone
c     xzone should be small compared to expected magnitude of
c     xval (because legitimate xval's could be stuck in the
c     zone); still xzone should be large enough that things
c     put there don't simple vanish into the noise
c     the test condition (xval > xlim) is assumed
       bouncL = xlim - xzone/(1.d0 + (xval - xlim))
            return
       end


program chemAct
c     version 5.3  <ing 5/95>
c <v5.2>  changed file input style to prompting users keyin input file
name,
c          output to fort.20, fort.30.... no longer used, changed to
c                  <input_put_file_name>.log,
c                  <input_put_file_name>.kin,
c                  <input_put_file_name>.col,
c                  <input_put_file_name>.out                          <ing
10/94>
c <v5.3>  added CHEB keyboard for chebyshov polynomials,
c          implimented subroutines for SGI version only to pc, so now
c          fitglobal, troe...keyword can be used, ....hooray          <ing
5/95>
c*****start here
c     non-standard implcit, but almost universal;
c     if problems switch to character or logical
       implicit none
c     grab parameter statments for commons
       include 'cdparams.fh'
c      grab only those commons needed for main
        include 'cdwell0.fh'
        include 'cdrange.fh'
```

```
        include 'cdrates.fh'
        include 'cdlimfit.fh'
        include 'cdcontrl.fh'
        include 'cdTemp.fh'
        include 'cdFfit.fh'
c     delcare local variables
      character option*8,fstamp*120
c      integer iWell,iProd,ip,n
      integer*2 hr_i,min_i,sec_i,hun_i,hr,min,sec,hun
      integer ip,n
      integer Nluout
      real*8 dE
      logical proceed,error
c     assign lu's; Nluout is in data statement
      integer linput,lecho,lkout,ldiag,lfout,ltout,lckout,ltabout,
     2   lffit,lfsum,lrecon,luarr(11),lusave(11),lplot
      equivalence (luarr(1),lecho),(luarr(2),lkout),(luarr(3),ldiag),
     2   (luarr(4),lfout),(luarr(5),ltout),(luarr(6),lckout),
     3   (luarr(7),ltabout),(luarr(8),lffit),(luarr(9),lfsum),
     4   (luarr(10),lrecon),(luarr(11),lplot)
      data linput,lecho,lkout,ldiag,lfout,ltout,lckout,ltabout,
     2   lffit,lfsum,lrecon,lplot,Nluout
     3        /10,20,25,30,35,45,55,65,75,85,95,99,11/
      data lusave/ 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1/
C      common/io/linput,lecho,lkout,ldiag,lfout,ltout,lckout,ltabout,
C     2   lffit,lfsum,lrecon, Nluout
c******end declarations ***********
c     default T and P range assigned through block data getrange
c     these defaults can be overwritten if option specified
c     open and read file header from input file
c       open(linput, file = 'fort.10', status = 'old')
      call file_io(error)
      call gettim(hr_i,min_i,sec_i,hun_i)
      read(linput,'(a)') fstamp
c     open all output files and stamp header
      call initfs(Nluout,luarr,lusave,fstamp)
c     read all other input parameters from input file and close
      call getparams(linput,lecho,dE)
      close(linput, status = 'keep')
c     echo generic parameters to file
       call echoparams(lecho)
c     loop through remainder of program doing chemact calculation first
c     then do dissoc calculations if desired
      write(*,*)' '
      write(*,*) 'program running...'
      do 200 n = 0,nWells
c       set switches
          if ((n.eq.0).and.(chemact)) then
                option = 'Chemact'
            proceed = .true.
          else if ((n.ge.1).and.(dissoc)) then
            option = 'Dissoc'
            inpWell = n
            inpchan = 0
            proceed = .true.
          else
            proceed = .false.
          endif
          if (proceed) then
c           get array idpWell giving depth of complex on isomer chart;
c           also compute Well depths in kcal; and Emax
C            write(*,*)' '
C            write(*,*) 'program running...'
c            write(*,*)'setcalc'
             call setCalc(lecho,isomers,option)
c               write(*,*) 'echo2'
```

```
            call echo2(lecho,option)
c           now do actual calculation
            if (option.eq.'Chemact') then
                write(*,*)' '
                write(*,*)'chemical activation...'
c            write(*,*) 'compChem'
               call compCHEM(lecho,ldiag,rkold,dE)
                write(*,*)'                          done'
            else
                write(*,*)' '
                write(*,110)n
  110           format(' unimolecular dissociation : well # ',i2)
c            write(*,*) 'compDiss'
               call compDISS(lecho,ldiag,rkold,dE)
                write(*,*)'                                    ...done'
            endif
c           finish calculation
c            write(*,*) 'cleanup'
            call cleanup(option)
            write(*,*)'   fitting and writting output...'
c            write(*,*) 'output'
c           now we output results; first we print table
            if (table) call tabout(option,ltabout)
c           now we loop over all reactions and output and or fit
c           remember iProd = 0 is stabilization channel for chemact
c           note common blocks kfit and Ffit only store one reaction's
c           worth of parameters at a time
            do 100 iWell = 1,nWells
               do 80 iProd = 0,NProds(iWell)
c                 the boolean addTerm indicates whether we must include
c                 secondary low p limit term
                  if ((option.eq.'Dissoc').and.(noDExit(iWell))
     2                .and.(iProd.eq.0)) then
                     addTerm = .true.
                  else
                     addTerm = .false.
                  endif
c              don't output dissoc channels where dissociating isomer
c              is Product - this is a usless rate and we integrated for
c              an E > 0 so we didn't calculate this correctly anyway
               if (.not.((option.eq.'Dissoc').and.(iWell.eq.inpWell)
     2             .and.(iProd.eq.0))) then
c                 fit limiting arrhenius factors
                  if ((fitglobal.or.fitrange).and.(nTemps.gt.2)) then
                     if (addTerm)
     2                  call Kfitout(option,lecho,lkout,iWell,
     3                  iProd,-1)
                     call Kfitout(option,lecho,lkout,iWell,iProd,0)
                     call Kfitout(option,lecho,lkout,iWell,iProd,
     2                  npres+1)
                  endif
c                 fit arrhenius factors over p range
                  if (fitrange.and.(nTemps.gt.2)) then
                     do 70 ip = 1,npres
c ** w.ing, skip normal 3-parameters arrn form output, if cheb selected
                        if (Fopt.ne.'CHEB') then
c                        write(*,*)'   fitting arrhenius parameters..'
                         call Kfitout(option,lecho,lkout,iWell,
     2                                    iProd,ip)
                         call CKoutput(option,lckout,iWell,iProd,ip)
                        endif
c ****************************************
70                   continue
                  endif
c                 fit lineshape factors; make sure enough parameters
                  if ((fitglobal).and.(nTemps.gt.3).and.
```

```
     2                  (nPres.gt.5)) then
c                        write(*,*)'   fitting f functions..'
                        if (Fopt.eq.'CHEB') then
                          write(*,120)iwell,iprod
  120                     format('    fitting chebyshev polynomials...',
     1                           'well #',i2,' channel # ',i2)
                          call kLimFit(option,lecho,lfsum,iWell,iProd)
                          call calcF(option,iWell,iProd)
                          call FitF(lecho)
c                          call FChebOut(option,lffit,lfsum,lckout,
                          call FChebOut(option,lffit,lecho,lckout,
     1                                  iWell,iProd)
                          call reConst(option,lrecon,iWell,iProd)
                        else
                          write(*,130)iwell,iprod
  130                     format('    fitting f functions....well #',
     1                           i2,' channel # ',i2)
                          call kLimFit(option,lecho,lfsum,iWell,iProd)
                          call calcF(option,iWell,iProd)
                          if (showline) then
                              call Foutput(option,lfout,iWell,iProd)
                              call Toutput(option,ltout,iWell,iProd)
                          endif
c      w.ing 02/95 add parameters (iWell and iProd) for calling FitF sub
                          call FitF(lecho)
                          call FitFout(option,lffit,lfsum,iWell,iProd)
                          call reConst(option,lrecon,iWell,iProd)
                        endif
                      endif
                    endif
80                continue
100             continue
              endif
200   continue
c     close all output files
999   continue
c      call closfs(Nluout,luarr,lusave)
      call gettim(hr,min,sec,hun)
      if (sec.lt.sec_i) then
          sec = sec + 60 - sec_i
          min = min - 1
      else
          sec = sec - sec_i
      endif
      if (min.lt.min_i) then
          min = min + 60 - min_i
          hr = hr - 1
      else
          min = min - min_i
      endif
      hr = hr - hr_i
      write(lecho,300)hr,min,sec
  300 format(1x,'execution time : ',i2,' hours ',i2,' minutes ',i2,
     1           ' seconds')
      write(*,310)hr,min,sec
  310 format(1x,'execution time : ',i2,' hours ',i2,' minutes ',i2,
     1           ' seconds')
      call closfs(Nluout,luarr,lusave)
      stop
      end
c***INITFS****************************
      subroutine initfs(Nlu,lu,lusave,string)
c     opens and clears out all output files and stamps header
      implicit none
c     local variables
      integer Nlu,lu(Nlu),lusave(Nlu)
```

```
      integer i
      character string*(*),num*2,fname*8
c     end declarations
      do 20 i = 1,Nlu
          if (lusave(i).eq.0) then
              write(num,'(i2)') lu(i)
              call concat('fort.',num,fname)
              open(lu(i), file = fname, status = 'unknown')
              close(lu(i), status = 'delete')
              open(lu(i), file = fname, status = 'new')
              write(lu(i),'(1x,a)') string
          endif
20    continue
      return
      end
c***CLOSFS*****************************
      subroutine closfs(Nlu,lu,lusave)
c     closes output files
      implicit none
c     local variables
      integer Nlu,lu(Nlu),lusave(Nlu)
      integer i
c     end declarations
      do 20 i = 1,Nlu
          if (lusave(i).eq.1) then
              close(lu(i), status = 'keep')
          else
              close(lu(i), status = 'delete')
          endif
20    continue
      return
      end
c GETRANGE******************
      block data getrange
        implicit none
      include 'cdparams.fh'
      include 'cdrange.fh'
c     local variables
      integer i,j
c     end declarations
      data ntemps,npres/13,61/
      data (temp(i),i=1,13) /300.,400.,500.,600.,800.,1000.,1200.,1500.,
     2   2000.,2500.,3200.,4000.,4800./
      data ((pres(i,j),i=1,13),j=1,21) /13*1.d-10,13*2.d-10,13*5.d-10,
     2   13*1.d-9,13*2.d-9,13*5.d-9,13*1.d-8,13*2.d-8,13*5.d-8,13*1.d-7,
     3   13*2.d-7,13*5.d-7,13*1.d-6,13*2.d-6,13*5.d-6,13*1.d-5,13*2.d-5,
     4   13*5.d-5,13*1.d-4,13*2.d-4,13*5.d-4/
      data ((pres(i,j),i=1,13),j=22,42) /13*1.d-3,13*2.d-3,13*5.d-3,
     2   13*1.d-2,13*2.d-2,13*5.d-2,13*1.d-1,13*2.d-1,13*5.d-1,13*1.d0,
     3   13*2.d0,13*5.d0,13*1.d1,13*2.d1,13*5.d1,13*1.d2,13*2.d2,
     4   13*5.d2,13*1.d3,13*2.d3,13*5.d3/
      data ((pres(i,j),i=1,13),j=43,61) /13*1.d4,13*2.d4,13*5.d4,
     2   13*1.d5,13*2.d5,13*5.d5,13*1.d6,13*2.d6,13*5.d6,13*1.d7,
     3   13*2.d7,13*5.d7,13*1.d8,13*2.d8,13*5.d8,13*1.d9,13*2.d9,
     4   13*5.d9,13*1.d10/
        end
c***LOOKUP*****************************
      subroutine lookup(sname,rmass,sig,ek,deltaE,noinfo)
      implicit none
        integer mxdata
      parameter (mxdata = 10)
c     local variables
      logical noinfo
      character sname*10
      integer n
      integer ndata
```

```
      real*8 rmass,sig,ek,deltaE
      character species(mxdata)*10
      real*8 smass(mxdata),ssig(mxdata),sek(mxdata),sdeltaE(mxdata)
c     do assignments here
      data ndata/6/
      data species(1) /'N2'/
      data smass(1),ssig(1),sek(1),sdeltaE(1)/28.0, 3.621, 97.5, 830./
           data species(2) /'AR'/
      data smass(2),ssig(2),sek(2),sdeltaE(2)/40.0, 3.33, 136.5, 630./
      data species(3) /'HE'/
      data smass(3),ssig(3),sek(3),sdeltaE(3)/4.0, 2.6, 10.2, 431./
      data species(4) /'CH4'/
      data smass(4),ssig(4),sek(4),sdeltaE(4)/16.0, 3.746, 141.4, 2100./
      data species(5) /'C3H8'/
      data smass(5),ssig(5),sek(5),sdeltaE(5)/44.0, 4.98, 266.8, 4200./
      data species(6) /'SF6'/
      data smass(6),ssig(6),sek(6),sdeltaE(6)/146.0, 5.13, 222.1, 3400./
            noinfo = .true.
      do 10 n = 1,ndata
         if (sname.eq.species(n)) then
            rmass = smass(n)
            sig = ssig(n)
            ek = sek(n)
            deltaE = sdeltaE(n)
            noinfo = .false.
         endif
10    continue
      return
      end
```

REFERENCES

1 *Chemical Engineering Progress* **June 1989**.

2 Norton, T.; Dryer, F.; Yetter, R. *Twenty-forth Symposium (International) on Combustion*, The Combustion Institute: Pittsburgh **1990**.

3 *Chemical Engineering Progress* **Nov. 1989**, p13.

4 Gulder, O.L. *Combustion Science and Technology* **1983**, 33, 179.

5 Norton, T. S.; Dryer, F. L. *Combust, Sci. and Tech.* **1989** 63, 107.

6 *Chem. & Eng. News* **1985**, 63, 12.

7 Anderson, E. *Chem. & Eng. News* **1986**, 64, 8.

8 *Oil & Gas J.* **1988**, 86, 28.

9 *Chem. & Eng. News* **1990**, 68, 9.

10 *Oil & Gas J.* **1987**, 85, 24.

11 *Remirez, R. Chem. Eng.* **1987**, 94, 19.

12 *Chem. & Eng. News* **1990**, 68, 17.

13 *Chem. Eng.* **1986**, 93, 22.

14 Al-Jarallah, A. M.; Lee, A. K. K. *Hydrocarbon Processing* **1988**, 67, 51.

15 *Process Eng.* **1988**, 69, 19.

16 Masters, K. R.; Prohaska, E. A. *Hydrocarbon Processing* **1988**, 67, 48.

17 Lange, P. M.; Martinola, F.; Oecki, S. *Hydrocarbon Processing* **1985**, 64, 51.

18 Lange, P. M.; Martinola, F.; Oecki, S. *Hydrocarbon Processing* **1985**, 64, 51.

19 Herwig, J.; Schleppinghoff, B.; Schulwitz, S. *Hydrocarbon Processing* **1984**, 63, 86.

20 Gaffney, J. *Combustion Science and Technology* **1980**, 24, 89.

21 Sorek, H. *Combustion Science and Technology* **1984**, 41, 203.

22 Smith, S.R.; Gordon, A.S. *J. Phys. Chem.* **1956**, 60.

23 Fletcher, J.C.N. *Proc. Roy. Soc.*(London) **1934**, A147, 119.

24 Someno, F. *Bull. Ins. Phys. Chem. Research,* Tokyo **1942**, 21, 277.

25 Cooke, D.F.; Dodson, M.G.; Williams, A.*Combustion and Flame* **1971**, 16, 233.

26 Bowman, C.T. *Combustion and Flame* **1975**, 25, 343.

27 Aronowitz, D.; Santoro, R.J.; Dryer, F.L.; Glassman, I *Seventeenth Symposium (International) on Combustion* **1978**, 633.

28 Westbrook, C.K.; Creighton, J.; Lund, C.; Dryer, F.L. *J. Phys. Chem.* **1977**, 81, 2542.

29 Vandooren J.; Van Tiggelen, P.J I *Eighteenth Symposium (International) on Combustion* **1981**, 473.

30 Grotheer, H.; Riekert, G.; Walter, D; Just, T. *Twenty-second Symposium (International) on Combustion* **1988**, 963.

31 Norton, T. S.; Dryer, F. L. *Int. J. Chem. Kinet.* **1990**, 22, 219.

32 Norton, T. S.; Dryer, F. L. *Twenty-third Symposium (International) on Combustion* **1990**, 179.

33 Westbrook, C. K.; Dryer, F. L. *Combust. Sci. and Tech.* **1979**, 20, 125.

34 Wallingtom, T. J.; Dagant, P.; Liu, R.; Kurylo, M. *J. Env. Sci. Tech.* **1988**, 22, 842.

35 Cox, R. A.; Goldstone, A. *Proceedings of the 2nd European Symposium on the Physico Chemical Behavior of Atmospheric Pollutants*; Riedel: Dorecht, Holland, **1988**, 112.

36 Daly, N. J.; Wentrup, C. *Aust. J. Chem.* **1968**, 21, 2711.

37 Choo, K. Y.; Golden, D. M.; Benson, S. W. *Int. J. Chem. Kin.* **1974**, 6, 631.

38 Brocard, J. C.; Baronnet, F. *Combust. & Flame* **1983**, 52, 25.

39 Dunphy, M.; Simmie, J. M. *Combust. Sci. Tech.* **1989**, 66, 157.

40 Norton, T. S.; Dryer, F. L. *Twenty-forth Symposium (International) on Combustion*, The Combustion Institute: Pittsburgh **1990**.

41 Lay, T. H.; Bozzelli, J. W. *The Eastern States Section of The Combustion Institute*: Princeton Univ., New Jersey **1993**, 449.

42 Lay, T. H. *Ph.D. Dissertation*, New Jersey Institute of Technology, Newark **1995**.

43 Lay, T. H.; Ritter, E. R.; Dean, A. M.; Bozzelli, J. W. submitted to *J. Phys. Chem.* **1995**.

44 Dean, A. M. *J. Phys. Chem.***1985**, 89, 4600.

45 Dean, A. M.; Bozzelli J. W.; Ritter, E. R. *Combust. Sci. and Tech.* **1991**, 80, 63.

46 Chang, A.Y., Bozzelli, J.W., and Dean, A.M., (To be Published). Personnel communication, Chang, A.Y., and Dean, A.M., Exxon Research and Engineering, Annandale, NJ.

47 Gilbert, R. G.; Luther, K.; Troe, J. *Ber. Bunsenges. Phys. Chem.* **1983**, 87, 169.

48 Hirschfelder, J. O.; Curtiss, C. F.; Bird, R. B. *Molecular Theory of Gases and Liquids*, 2nd ed., Wiley, London **1963**.

49 Harrington, R. E.; Rabinovitch, B. S.; Hoare, M. R. *J. Chem. Phys.* **1960**, 33, 744. Kohlmaier, G. H.; Rabinovitch, B. S. *ibid.* **1963**, 38. Ireton, R. C.; Rabinovitch, B. S. *J. Phys. Chem.* **1974**, 78.

50 Reid, R. C.; Prusnitz, J. M.; Poling, B. E. *The properties of Gases and Liquids* 3nd ed., McGraw-Hill, Singapore **1988**.

51 Westmoreland, P. R.; Howard, J. B.; Longwell, J. P.; Dean, A. M. *AIChE Journal* **1986**, 32, 1971.

52 Dean, A. M.; Westmoreland, P. R. *Int. J. Chem. Kinet.* **1987**, 19, 207.

53 Westmoreland, P. R. *Combust. Sci. and Tech.* **1992**, 82, 151.

54 Ritter, E. R.; Bozzelli, J. W.; Dean, A. M. *J. Phys. Chem.* **1990**, 94, 2493.

55 Bozzelli, J. W.; Dean, A. M. *J. Phys. Chem.* **1990**, 94, 3313.

56 Bozzelli, J. W.; Dean, A. M. *J. Phys. Chem.* **1993**, 97, 4427.

57 Bozzelli, J. W.; Chang, A. Y.; Dean, A. M. *Twenty-fifth Symposium (International) on Combustion*, The Combustion Institute. **1994**.

58 Ho, W. P.; Barat, R. B.; Bozzelli, J. W. *Combust. and Flame* **1992**, 88, 265.

59 Tavakoli, J.; Bozzelli, J. W. Collected Papers in Heat Transfer HTD Vol.-104, K.T. Yang, ed., ASME, **1988**, 2, 147.

60 Won, Y. S.; Bozzelli, J. W. *Combust. Sci. and Tech.* **1992**, 85, 345.

61 Benson, S.W. *Thermochemical Kinetics*, 2nd ed., Wiley, New York, **1976**.

62 Ritter, E. R. CPFIT Computer Code, *J. Chem. Inf. Computat. Sci.* **1991**, 31, 400.

63 Bozzelli, J. W.; Chang, A. Y.; Dean, A. M. submitted to *J. Phys. Chem.* **1995**.

64 Ben-Amotz, D.; Herschbach, D. *J. Phys. Chem.* **1990**, 94, 3393.

65 Robinson, P. J.; Holbrook, K. A. *Unimolecular Reactions*; Wiley-Interscience; New York, **1972**.

66 Forst, W. *Theory of Unimolecular Reactions*; Academic Press; New York, **1973**.

67 Gilbert, R. G.; Smith, S. C. *Theory of Unimolecular and Recombination Reactions*; Blackwell Scientific; Oxford and Cambridge, Mass., **1990**.

68 Quack, M.; Troe, J. *Ber. Bunsenges. Phys. Chem.* **1974**, 78, 240.

69 Quack, M.; Troe, J. *Ber. Bunsenges. Phys. Chem.* **1975**, 79, 140.

70 Quack, M.; Troe, J. *Ber. Bunsenges. Phys. Chem.* **1975**, 79, 469.

71 Quack, M.; Troe, J. *Ber. Bunsenges. Phys. Chem.* **1977**, 81, 329.

72 Stewart, P. H.; Larson, C. W.; Golden, D. *Combus. and Flame*, **1989**, 75, 25.

73 Kazakov, A.; Wang, H.; Frenklack, M. *American Chem. Soc.* **1994**, 98, 10598.

74 Gardiner, W. C. *The 12th IMACS World Congress on Scientific Computations*, Paris, **1988**, 58.

75 Venkatesh, P.; Chang, A. Y.; Dean, A. M.; Cohen, M. H.; Carr, R. W. (To be Published), Personnel communication.

76 Fox, L.; Parker, I. B. *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, London, **1968**.

77 Rivlin, T. J. *Chebyshev Polynimials From Approximation Theory to Algebra and Number Theory*, Second Edition, Wiley-Interscience, New York, **1990**.

78 Ritter, E.R.; Bozzelli, J.W. *Int. J. Chem. Kinet.* **1991**, 23, 767.

79 Benson, S.W. *Thermochemical Kinetics*, 2nd ed., Wiley, New York, **1976**.

80 Kee, R. J.; Rupley, F. M.; Miller, J. A. *CHEMKIN-II, A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics*, Sandia National Laboratories, Livermore, CA, **1990**.

81 Ritter, E. R.; Bozzelli J. W. The Eastern States Section of The Combuion Institute, Princeton University, New Jersey, p.459, **1993**.

82 Ritter, E. R. Ph.D. Dissertation, New Jersey Institute of Technology, **1989**.

83 Ritter, E. R. *J. Chem. Inf. Computat. Sci.* **1991**, 31, 400.

84 Ritter, E. R.; Bozzelli, J. W.; Dean A.M. *J. Phys. Chem.* **1990**, 94, 2493.

85 Radford, H.E. *Chem. Phys. Lett.* **1980**, 71, 195.

86 Wang, W. C.; Suto, M.; Lee, L. C. *J. Chem. Phys.* **1984**, 81, 3122.

87 Grotheer, H. H.; Riekert, G.; Meier, U.; Just, Th. *Ber. Bunsen-Ges. Phys. Chem.* **1985**, 89, 187.

88 Payne, W. A.; Bruuing, J.; Mitchell, M. B.; Stief, L. J. *Int. J. Chem. Kinet.* **1988**, 20, 63.

89 Dobe, J. E.; Warnatz, J. *Ber. Bunsen-Ges Phys. Chem.* **1983**, 87, 1040.

90 Pagsberg, P.; Munk, J. *J. Phys. Chem.* **1989**, 93, 5162.

91 Nesbitt, F. L.; Payne, W. A.; Stief, L. J. *J. Phys. Chem.* **1988**, 92, 4030.

92 Miyoshi, A.; Matsui, H.; Washida, N. *J. Phys. Chem.* **1990**, 94, 3016.

93 Veyret, B.; Lesclaux, R.; Rayez, M.; Rayez, J. *J. Phys. Chem.* **1989**, 93, 2368.

94 Veyret, B.; Rayez, J.; Lesclaux, R. *J. Phys. Chem.* **1982**, 86, 3424.

95 Evleth, E. M.; Melius, C. F.; Rayez, J. C.; Rayez, W.; Forst, W. *J. Phys. Chem.* **1993**, 97, 5040.

96 Walch, S. P. *American Inst. Phys.* **1993**, 98, 3163.

97 Cobos, C. J.; Hippler, H.; Luther, K.; Ravishankara, A. R.; Troe, J *J. Phys. Chem.* **1985**, 89, 4332.

98 Semmes, D. H.; Ravishankara, A. R.; Gump-Perkins, C. A.; Wine, P. H. *Int. J. Chem. Kinet.* **1985**, 17, 303.

99 Daly, N. J.; Wentrup, C. *Aust. J. Chem.* **1968**, 21, 2711.

100 Choo, K. Y.; Golden, D. M.; Benson, S. W. *Int. J. Chem. Kin.* **1974**, 6, 631.