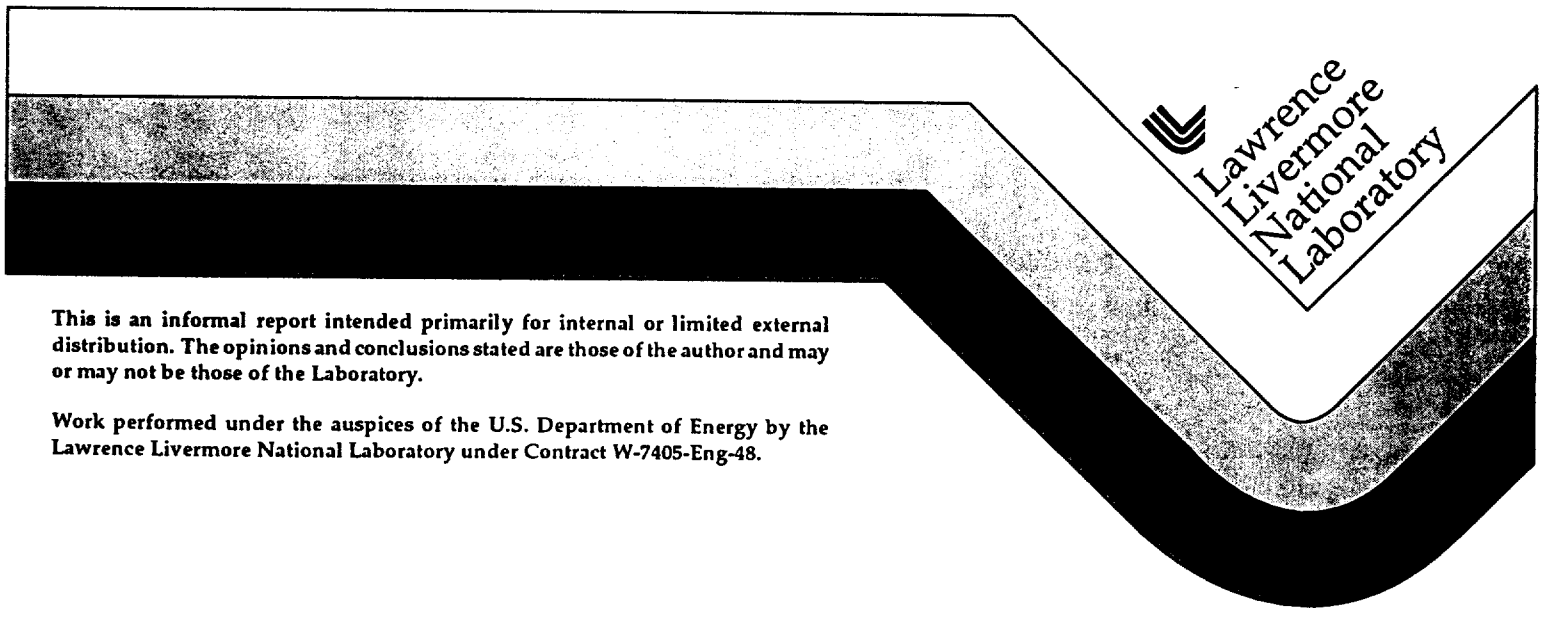
231033

UCRL-ID-126220

A Process and Economic Model of In-Field Heavy Oil Upgrading Using Aqueous Pyrolysis

C.B. Thorsness
W.C. Miller

January 21, 1997



Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

A Process and Economic Model of In-Field Heavy Oil Upgrading Using Aqueous Pyrolysis

C. B. Thorsness
W. C. Miller

Abstract

A process and economic model for aqueous pyrolysis in-field upgrading of heavy oil has been developed. The model has been constructed using the ASPEN PLUS chemical process simulator. The process features cracking of heavy oil at moderate temperatures in the presence of water to increase oil quality and thus the value of the oil. Calculations with the model indicate that for a 464 Mg/day (3,000 bbl/day) process, which increases the oil API gravity of the processed oil from 13.5° to 22.4°, the required value increase of the oil would need to be at least $2.80/Mg·°API ($0.40/bbl ·°API) to make the process economically attractive. This level of upgrading has been demonstrated in preliminary experiments with candidate catalysts. For improved catalysts capable of halving the coke make and increasing the pyrolysis rate, a required price increase for the oil as low as $1.34/Mg·°API ($0.21/bbl ·°API) has been calculated.

INTRODUCTION

Heavy crude oil with an API gravity below 20° generally has a considerably lower market value then benchmark crudes because of undesirable physical and chemical properties (e.g., high viscosity and high heteroatom content). One strategy to increase the value of these crude oils and to allow easier transport by existing pipelines is in-field upgrading. Aqueous pyrolysis is one candidate for in-field upgrading[1] of heavy oils. It involves heating an oil-water mixture under pressure, in the presence of a suitable catalyst, to crack and coke the heavy constituents and thus improve the oil quality. This process is attractive since the heavy oils are often associated with some thermally assisted recovery processes, such as steam flooding. As a result the oils are often produced at somewhat elevated temperatures and with considerable water. The produced fluids, therefore, often already require some form of a dewatering step to allow them to be marketed, and the Aqueous Pyrolysis process can be viewed as an extension of the dewatering process.

The Aqueous Pyrolysis process is related to standard vis-breaking and coking operations. However, it uses water and other additives (catalysts) to allow operation at modest temperatures with a minimum of coke make. The processing is intended

to be closely coupled to a field site and is aimed at more modest changes in oil properties than standard refinery based operations.

To evaluate the potential of the process and to allow performance goals for a catalyst system to be developed, it is useful to have a means of estimating the cost of a proposed process using a given crude oil feed. The purpose of this report is to describe such a model and report preliminary cost estimates. The model was developed using a commercially available process simulator, ASPEN PLUS.

PROCESS DESCRIPTION

A flowsheet of the proposed process is shown in Fig. 1. The primary process units include a high pressure pump, high pressure heat exchangers, a high pressure heater and a high pressure reactor vessel. In the following description, typical temperatures, compositions, and pressures are given to help clarify the discussion. These values are for results for a defined base case described later; however, the specific process to be modeled need not necessarily conform to these values.

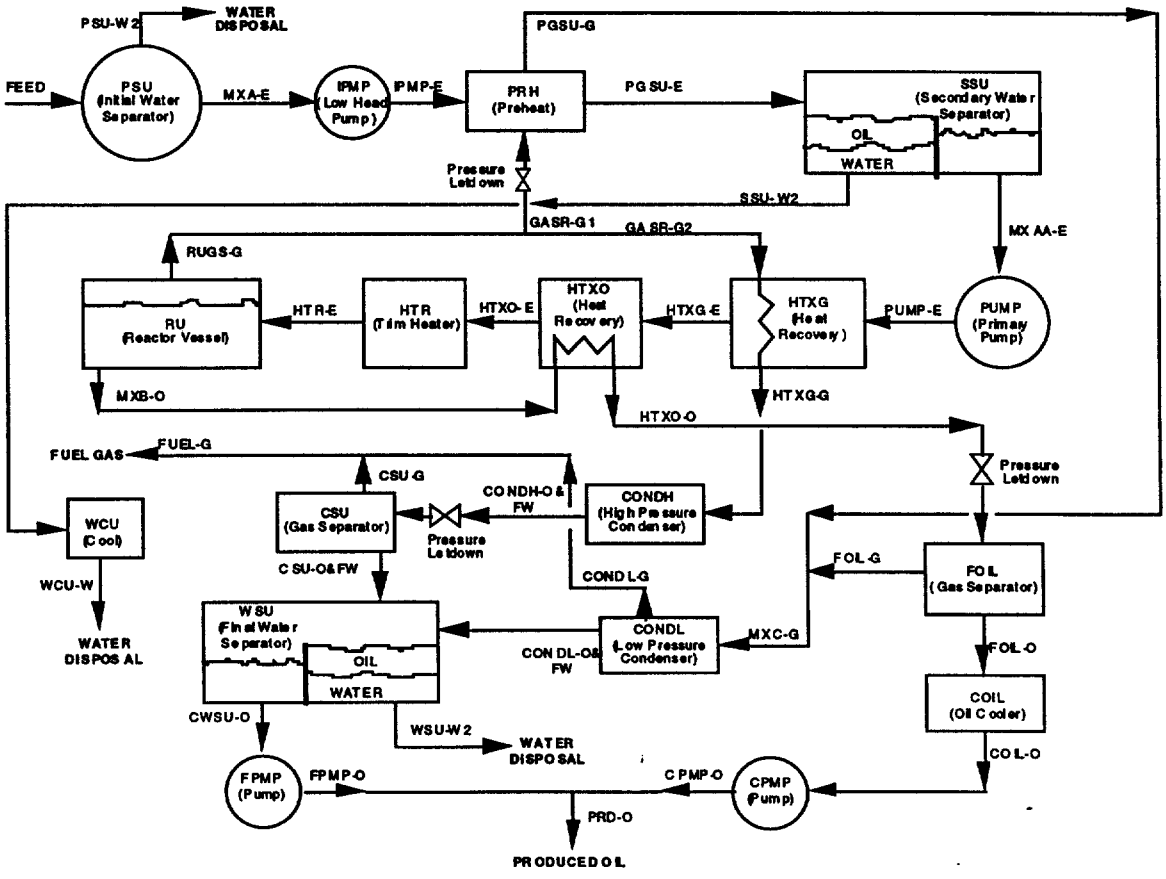


Figure 1. Aqueous pyrolysis process flowsheet.

It is assumed that produced fluids from a thermal recovery operation (labeled FEED in flowsheet) enter the process at a modest temperature, 38°C, and a water/oil ratio of 10:1. The first unit, labeled PSU, is a simple settling tank used to separate the bulk of the water, stream PSU-W2, from the oil. The crude oil and remaining water, 30% by weight, are fed to a pump which raises their pressure slightly (0.7 MPa) to accommodate further dewatering.

The fluid then moves to vessel PRH where it is heated by combining it with hot vapor from downstream processing. This facilitates further dewatering of the oil. The liquid outlet stream, PGSU-E, is heated to 78°C in this operation. This heated fluid then enters a water separation vessel, SSU, where the water content in the primary oil stream, MXAA-E, is reduced to 10%. The water exiting the water separator goes to a small heat exchanger, WCU, where it is cooled to 25°C before it leaves the battery limits as waste water.

The primary process stream, MXAA-E, is fed to the main high pressure pump, PUMP, where the pressure is increased to 13.8 MPa. The stream passes to two heat-recovery heat exchangers and a final trim heater. The first heat exchanger, HTXG, uses hot vapors from the reactor vessel to heat the process stream to 270°C. The second heat exchanger, HTXO, uses the hot liquid exiting the reactor to further heat the stream. The stream exits HTXO at 317°C then enters a fired heater, HTR, which heats the stream to the desired final reaction temperature (438°C).

The stream then enters the reactor vessel, RU. The reactor vessel allows sufficient residence time (two hours) at elevated temperature for the desired upgrading to occur. The reactor vessel separates the product stream into liquid, MXB-O, and vapor, RUGS-G, product streams.

The liquid product stream exiting the reactor, MXB-O, passes through the heat recovery heat exchanger and then goes to the pressure letdown station. Downstream of the pressure letdown is a separator vessel, FOIL, which separates liquid and vapor phases. The vapor, FOIL-G, and liquid, FOIL-O, exit the separator vessel at 284°C and 0.2 MPa. The liquid stream, which is essentially water free, passes through an air cooler and emerges at 93°C. The exiting vapor stream, FOIL-G, is then combined with the vapor stream from the PRH unit and enters the low pressure condenser unit, CONDL. CONDL uses chilled water to drop the temperature of the exiting streams to 24°C.

The part of the vapor product stream exiting the reactor, RUGS-G, which is not used for preheating passes through the heat recovery heat exchanger and then to a high pressure chilled water heat exchanger, CONDH. In the flowsheet, this unit is labeled as a condenser although in reality the stream entering it is essentially all liquid at a temperature of 117°C. The CONDH heat exchanger cools the stream to 24°C. The stream is then letdown in pressure to 0.2 MPa and vapor and liquid streams are separated in vessel CSU.

The vapor exists the CSU unit at 17°C and is combined with the vapor CONDL-G from the low pressure condenser. The combined stream exists the battery as a fuel gas containing low molecular weight hydrocarbons and carbon dioxide.

The liquid streams leaving CONDH and CONDL contain free water. This water is removed in a final water separation vessel, WSU. (Water/oil ratio entering the unit is 0.1.)

The two product oil streams are pumped to pressure (1.03 MPa, 150 psi) and combined before exiting the battery limits.

Several operations have been omitted from the flowsheet. One is the introduction and possible recovery of a catalyst. Introduction of a catalyst would be a straight forward operation and have little influence on the basic energy and material flows. Any recovery would depend on the cost of the catalyst and details of its behavior in the various fluid streams. These operations could be added to the model in the future if a specific catalyst recovery operation were to be defined.
Another operation that has been omitted is the removal of coke from the system. The tacit assumption made by the continuous nature of the process is that any coke produced is carried out of the processing units by the flowing streams. An ideal catalyst system would yield very low coke formation rates. However, if the coke levels were unacceptably high in the product oil stream then some operation would have to be added to reduce them.

THE ASPEN PLUS PROCESS MODEL

The flowsheet described above has been modeled using the ASPEN PLUS process simulator. The ASPEN PLUS program is a steady-state modular flowsheet simulator in which process models are constructed by linking together basic modules. ASPEN PLUS refers to these modules as unit-operation blocks. These blocks are linked by material and energy streams. Chemical species in the material streams have fundamental properties defined by selecting various thermodynamic models. ASPEN PLUS has a large data base of conventional chemical species and in addition allows nonconventional species to be defined. ASPEN PLUS also allows user supplied FORTRAN coding to be integrated into the simulation. A feature which has been heavily utilized in developing the Aqueous Pyrolysis flowsheet model.

In this section, the species and reactions considered by the model are outlined. This is followed by a description of the actual ASPEN PLUS computational modules used to simulate the flowsheet. The next section describes the ASPEN PLUS economic model which obtains its primary inputs from the

flowsheet model. The flowsheet model is constructed to allow different feed compositions; however, to help clarify the description, reference to a specific feed is made. A complete listing of the ASPEN PLUS input file is provided in Appendix I. The following model description does not try to repeat all the information available in the ASPEN PLUS documentation. The reader is referred to that documentation for a more detailed description of model elements.

Components

Three basic types of chemical components or species are used in the model: conventional components, pseudocomponents, and solids. The complicated crude oil composition is simulated using a set of pseudocomponents. These pseudocomponents and their properties were obtained by using facilities available in ASPEN PLUS[2]. A set of six pseudocomponents representing the crude oil are defined in the model representing different boiling point fractions. Their properties are set by the ASPEN PLUS model based on measured boiling point curves. Table 1 lists the components along with their important characterizing properties for the base case. For other crude oils the properties can vary, but the pseudocomponent names would be preserved.

Table 1. Pseudocomponents used to simulate crude oil.

Name	MW gm/mole	wt %	Specific gravity	Boiling Point (°C)	Critical Temp (°C)	Critical Pres (MPa)	Acentric Factor	API Gravity (°)
HNAPH	142	5.00	0.844	186	387	2.61	0.37	36.1
KERO	178	6.00	0.877	241	443	2.19	0.46	29.9
AGO	228	12.00	0.911	304	504	1.83	0.57	23.7
LVGO	307	16.10	0.954	388	583	1.48	0.73	16.9
HVGO	411	26.80	1	483	668	1.21	1	10.4
VR	540	34.10	1.04	588	759	0.99	1.29	4.4

The ASPEN PLUS PROPERTIES option computes water solubility in the oil phase. Coefficients of the solubility parameters for the oil pseudocomponents have been altered to extend their use to higher temperatures. This is a crude treatment of the solubility at elevated temperatures based on very limited information. This extension is discussed in another report[3].

In addition to the pseudocomponents used to define the feed oil stream, a number of other components are defined in the model. A seventh pseudocomponent, P-OIL, used in defining the overall aqueous pyrolysis reaction is based on an assumed boiling point and API gravity. In addition, solid carbon is used to represent coke (COKE is the component name in the model). Finally a group of conventional components are included: H2O, CH4, C2H6, C3H8, HHC, H2S, CO2, and H2. These names identify the actual

species within the model with the exception of HHC. This is meant to be a hydrocarbon species needed to complete the definition of the pyrolysis reaction. In the base case this component is normal butane.

Aqueous Pyrolysis Reaction

The pyrolysis reaction considered in the model is a simplified representation of what occurs. Its structure is based on limited information on potential systems of interest. Reference 3 discusses the development of the simple model. It is assumed that the crude oil reacts at some rate according to a simple first order decomposition reaction where the rate constant is given by an Arrhenius' form. The model assumes that all crude components react at a similar rate and produce coke, light hydrocarbons, hydrogen sulfide, carbon dioxide, and a light oil fraction. The reaction stoichiometry used in the base case is listed in Table 2.

Table 2. Base case reaction stoichiometry.

	Weight Fraction
Coke	0.2
P-OIL	0.6
CO_2	0.028
H_2S	0.014
H_2	0.0006
CH_4	0.053
C_2's	0.034
C_3's	0.038
C_4's	0.033

In the model, care is taken to preserve mass during reaction, but atomic balances are not strictly enforced since atomic compositions of oil and coke components are not explicitly defined.

Unit-Operation Blocks

The basic process flowsheet unit operations, shown in Fig. 1 do not map one-to-one into ASPEN PLUS unit-operation blocks. The ASPEN PLUS unit-operation blocks used to model the process flowsheet have been organized into five areas. These areas represents different aspects of the process and are shown in Fig. 2. In this figure material streams are shown with solid lines and energy streams with broken lines. The stream names used here, as in Fig. 1, are those names actually defined and used in the ASPEN PLUS model.

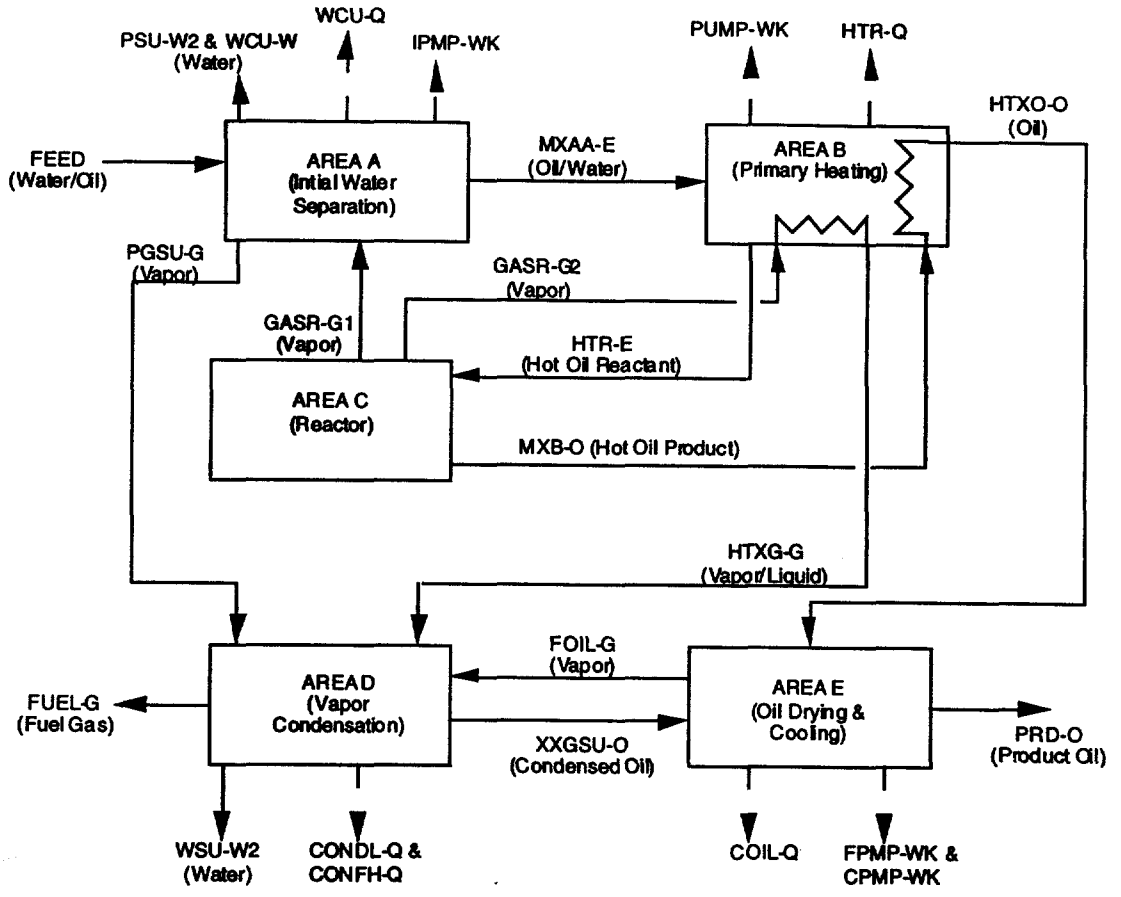


Figure 2. Process areas used in organizing the ASPEN PLUS unit-operation blocks.

In Fig. 3, Area A ASPEN PLUS blocks and streams are shown. This area models units PSU, IPMP, PRH, SSU, and WCU in the overall process flowsheet. The names shown are those of the actual ASPEN PLUS blocks. The stream names which appear on this area and the other area diagrams have been used to name streams on the overall process flowsheet, Fig. 1. The material stream names follow the convention of being composed of the unit name from which they originate, and a suffix which indicates the stream is primarily an oil/water stream (-E), a water stream (-W), a free water stream (-FW), an oil stream (-O), or a vapor stream (-G). These conventions hold fairly well but in certain cases do not necessarily reflect the dominate phase. Energy flows are also shown (broken lines) and have a suffix indicating if it is a thermal energy stream (-Q) or a mechanical energy stream (-WK). By ASPEN PLUS convention, the direction of the energy streams is always out of a block; thus, energy input streams always have a negative sign. Not included in this summary figure are streams which exist in the model, but which are generally zero.

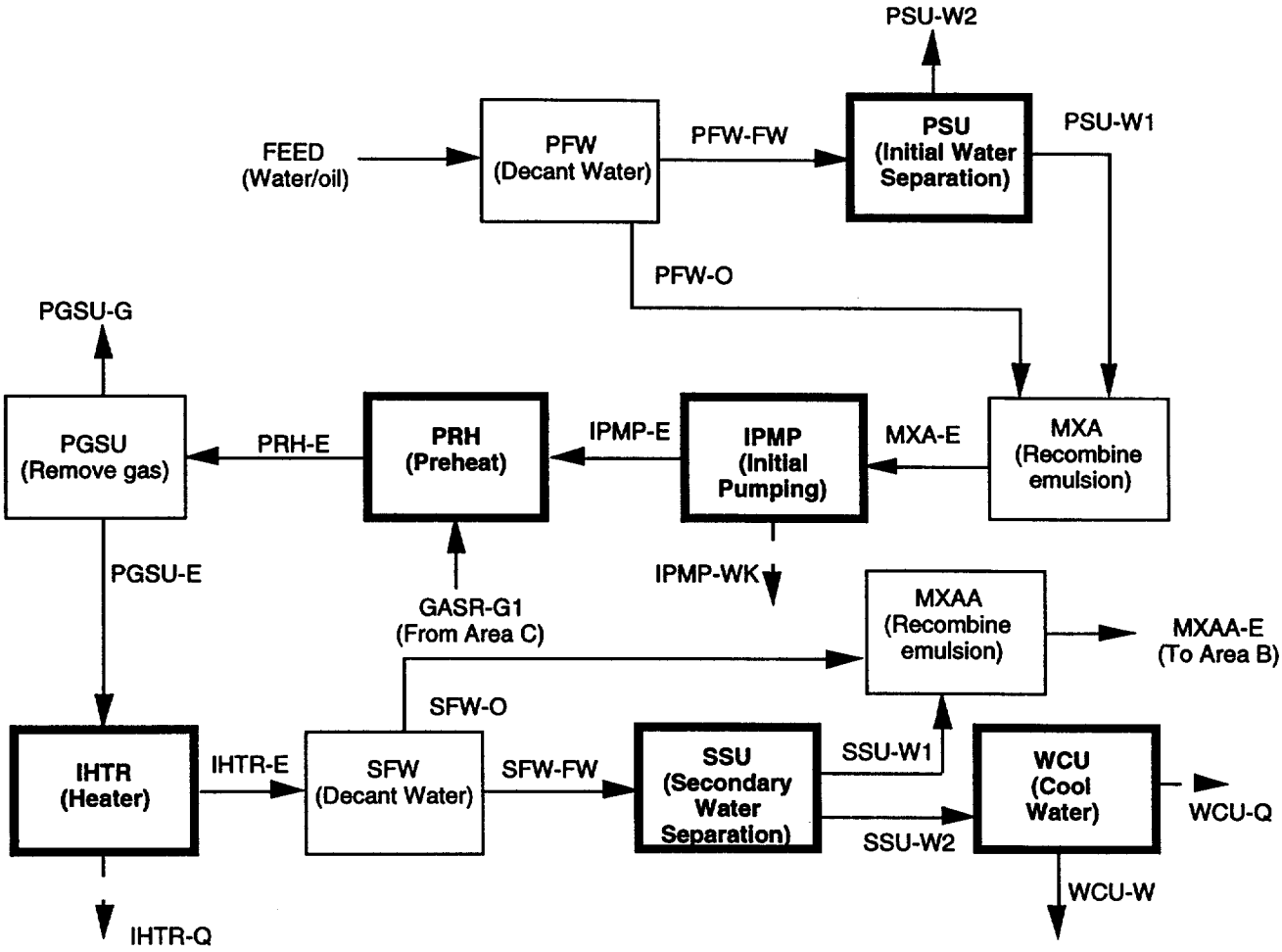


Figure 3. Model Area A, Initial Water Separation

The streams shown in the ASPEN PLUS diagrams do not necessarily represent individual streams in the actual process. For example, the free water streams represent a phase which coexists with a separate oil rich phase. Such artificial separation of streams are repeatedly used in the model to allow control over the actual amount of water remaining in a given stream. The amount of water present in the liquid phase at any point in the system in the true process is the sum of the amount dissolved in the oil phase and the amount of free water. Since the amount of free water is a function of the efficiency of separation that has occurred, plus the presence or absence of an emulsion phase, the model allows the water concentration to be set by the user at important points in the process sequence.

The function of the ASPEN PLUS unit-operation blocks for Area A are listed below. Also listed are the FORTRAN blocks associated with the unit-operation blocks. Several of the ASPEN PLUS unit-operation blocks are

given the same names as those used for actual unit operations shown on the overall process flowsheet (Fig. 1). These blocks are closely associated with the flowsheet unit operation, but often they are only a part of the ASPEN PLUS model for that operation. The name in {} is the functional name of the ASPEN PLUS module (or designated as a FORTRAN code block):

- PFW {MIXER} — Separates free water from oil.
- PSU {FSPLIT} — In conjunction with the FORTRAN block FPSU sets the amount of water to be removed from the system by the first separation.
- FPSU {FORTRAN} — Sets split parameters for PSU based on stream compositions and user defined separation criteria.
- MXA {MIXER} — Recombines streams to represent true process flowsheet stream.
- IPMP {PUMP} — Models the pumping of process stream to intermediate pressure.
- PRH {MIXER} — Combines vapor and process stream. Along with PGSU block it models the PRH preheat vessel.
- PGSU {FLASH2} — Models separation of gas and liquid phases in the PRH vessel.
- IHTR {HEATER} — A unit not shown in the process flowsheet because it is generally not needed. Based on user input it can be used to model the further heating of the process stream prior to entering the SSU unit.
- FIHTR {FORTRAN} — Used to set exit temperature from IHTR based on user input. Handles the case when IHTR not needed.
- SFW {MIXER} — Separates free water from oil.
- SSU {FSPLIT} — In conjunction with the FORTRAN block FSSU sets the amount of water to be removed from the system by the second separation.
- FSSU {FORTRAN} — Sets split parameters for SSU based on stream compositions and user defined separation criteria.
- MXAA {MIXER} — Recombines streams to represent true flowsheet stream.
- WCU {HEATER} — Sets the desired outlet temperature for waste water stream WCU-W and computes the required amount of cooling.

Figure 4 contains the ASPEN PLUS unit-operation blocks for Area B which model the high pressure pumping and stream heating operations:

- XGSU {FASH2} — Insures that no vapor phase is present in PUMP feed. The vapor stream exiting this module is always essentially zero.
- PUMP {PUMP} — The primary pump in the system which raises input stream to processing pressure.
- HTXG {HEATX} — Heat exchange between hot vapor and incoming feed.
- HTXO {HEATX} — Heat exchange between hot liquid and incoming feed.
- HTR {HEATER} — Trim temperature to desired reaction temperature.

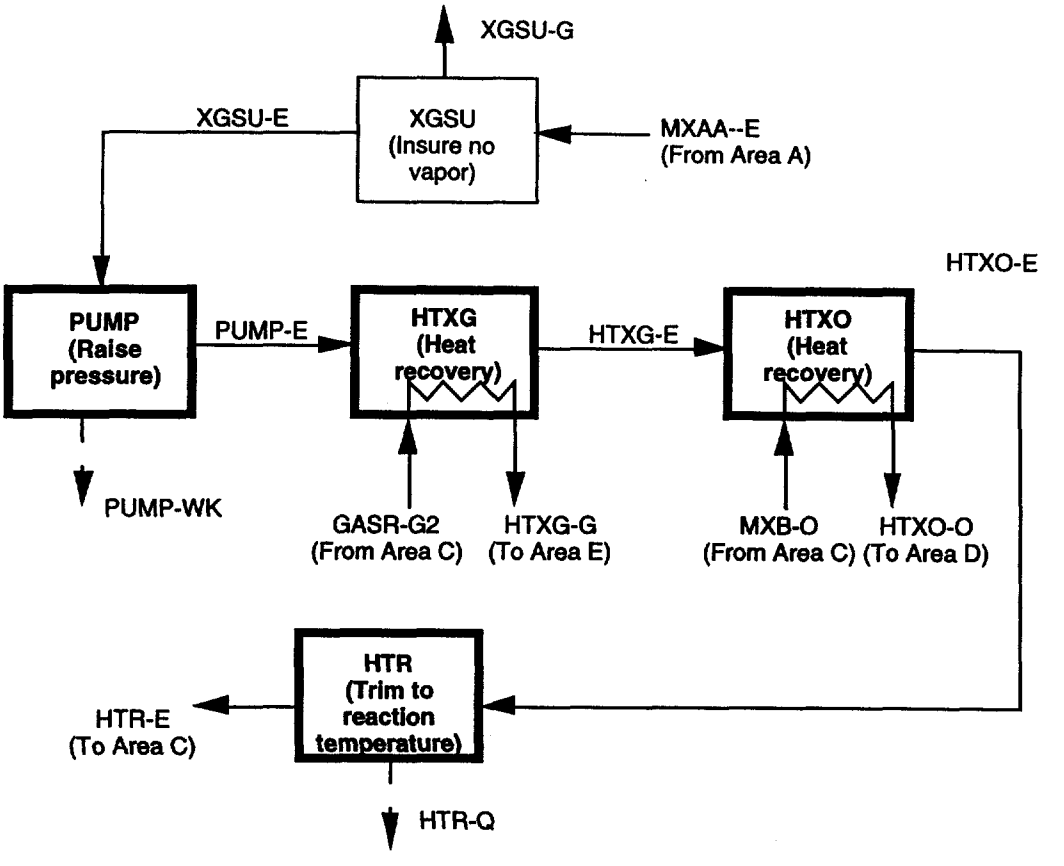


Figure 4. Model Area B, Primary Heating.

Figure 5 contains the ASPEN PLUS unit-operation blocks for Area C which model the reactor vessel. The model assumes that reactions within the reactor vessel can be approximated using three CSTRs in series. The pyrolysis reactions are only allowed to occur in the liquid phase.

- RU1, RU2, RU3 {RCSTR} — Three RCSTR unit-operation blocks which model the pyrolysis.
- FRU {FORTRAN} — Sets each RCSTR block volume to one third the total reactor volume. The total reactor volume is based on user requested residence time and the average of the liquid-phase volumetric flow into block RU1 (stream HTR-E) and out of block RU3 (stream RU3-E).
- RUGS {FLASH2} — Separate out the gas phase.
- GASR {FSPLIT} — Split the required amount to be recycled for heating the incoming feed in the PRH vessel. The split fraction is set using an ASPEN PLUS DESIGN-SPEC block, SPEC1. This design specification adjusts the split fraction until the desired temperature leaving the PRH vessel (stream PGSU-E) is reached.
- RFW {MIXER} — Separates free water from oil.

- RSU {FSPLIT} — In conjunction with the FORTRAN block FRSU sets the amount of water to be removed from the system by the second separation. In practice the model has only been used to look at cases where no free water exists, so the RSU-W2 stream flow rate is always zero.
- FRSU {FORTRAN} — Sets split parameters for SSU based on stream compositions and user defined separation criteria.
- MXB {MIXER} — Recombines streams to represent true flowsheet stream.

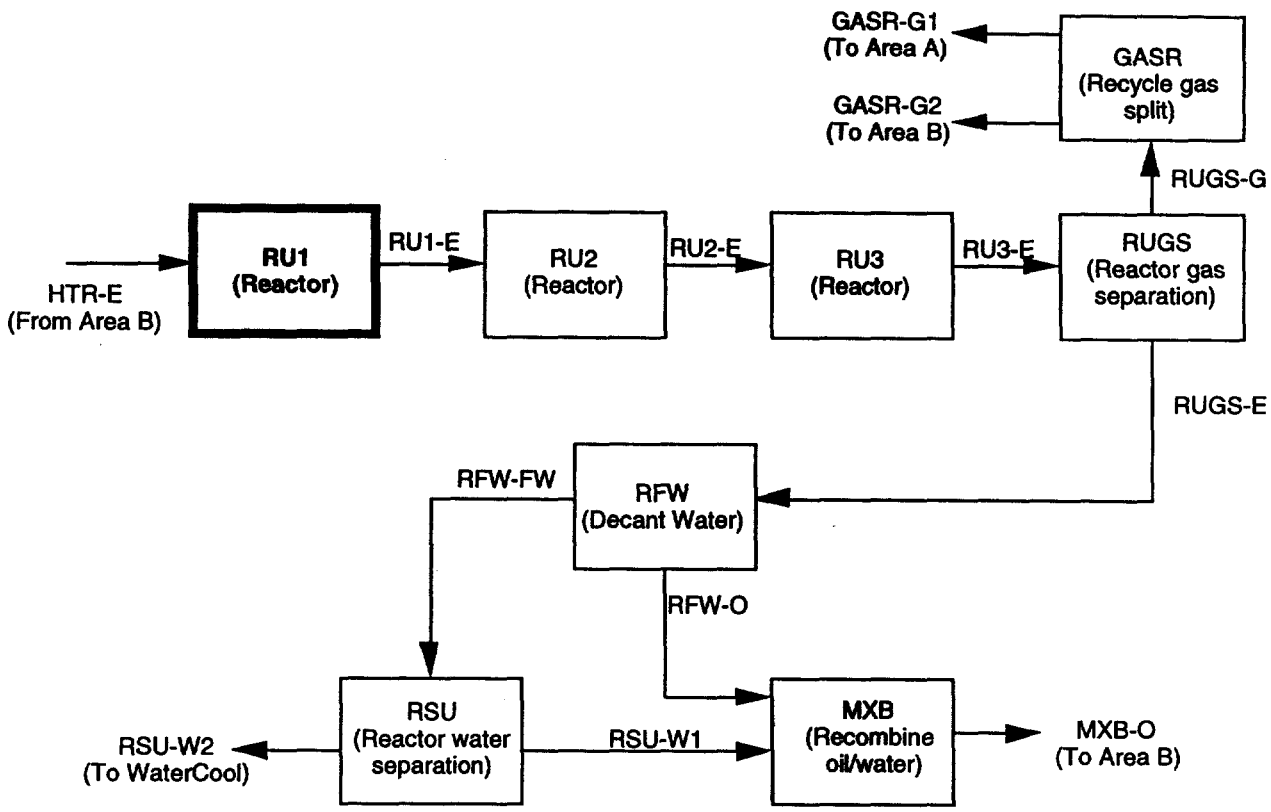


Figure 5. Model Area C, Reactor.

Figure 6 contains the ASPEN PLUS unit-operation blocks for Area E which model vapor condensation:

- MXC {MIXER} — Mix streams before entering low pressure condenser.
- CONDL {FLASH2} — Low pressure condenser, outlet temperature set.
- CONDH {FLASH2} — High pressure condenser/cooler, outlet temperature set.
- CSU {FLASH2} — Flash stream to low pressure and separate liquid and vapor streams.
- MXG {MIXER} — Combine fuel gas streams.

- MXWSU {MIXER} — Combine liquid streams from condensers and separate free water from oil.
- WSU {FSPLIT} — In conjunction with the FORTRAN block FWSU sets the amount of water to be removed from the system by the final, WSU, separation.
- FWSU {FORTRAN} — Sets split parameters for WSU based on stream compositions and user defined separation criteria.
- CWSU {MIXER} — Recombines streams to represent true flowsheet stream.
- XXGSU {FLASH2} — Separates trace vapor phase from liquid phase to insure pump block will operate properly.

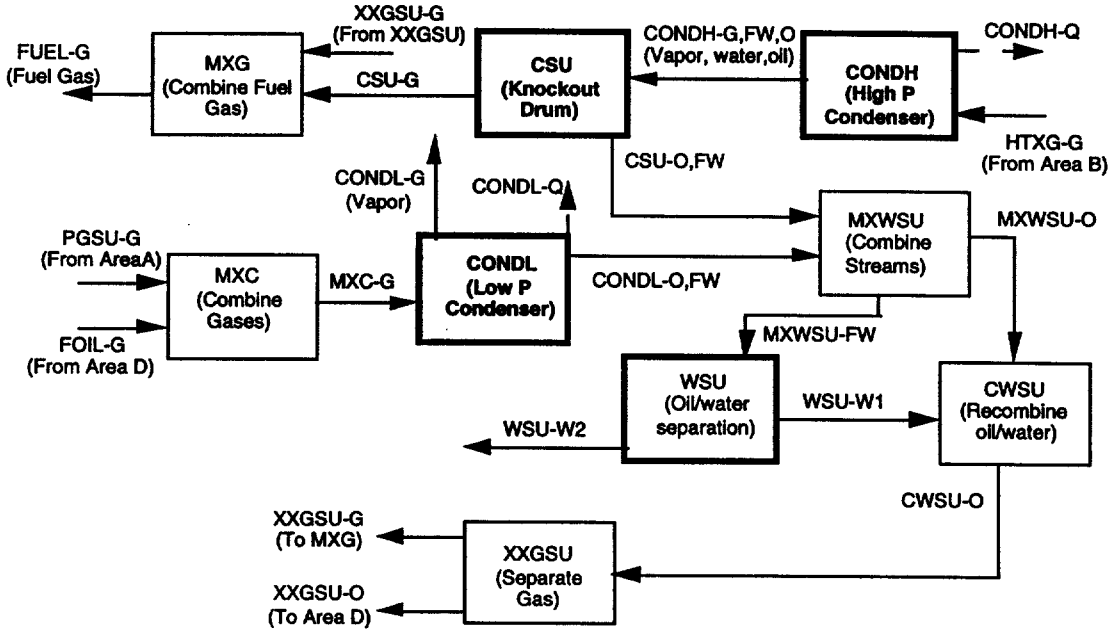


Figure 6. Model Area D, Oil Drying and Cooling

Figure 7 contains the ASPEN PLUS unit-operation blocks for Area E, which model oil drying and cooling operations, as well as low pressure pumping to reach desired delivery pressure for the product oil:

- FOIL {FLASH2} — Flash hot oil/water to cool and remove water.
- COIL {HEATER} — Cool oil.
- CPMP & FPMP {PUMP} — Low pressure oil pumps to set desired delivery pressure.
- PRD {MIXER} — Combine oil product streams.

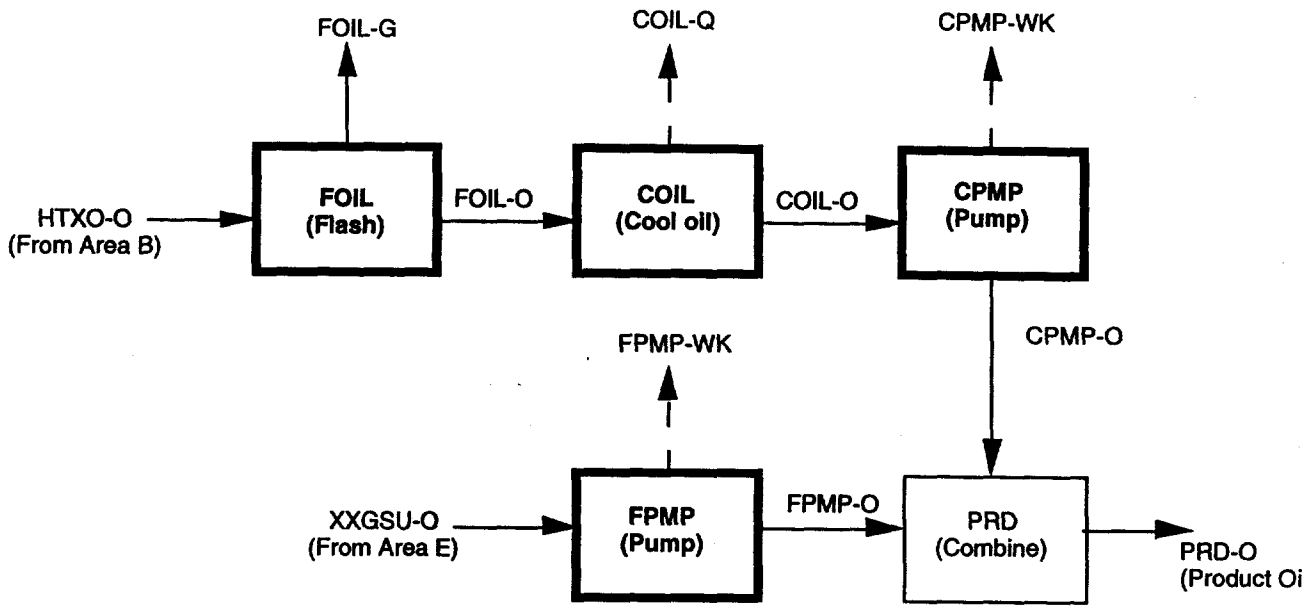


Figure 7. Model Area E, Gas Colling.

Input for Unit-Operation Blocks

Input parameters to the model are defined in the ASPEN PLUS input language file. This file not only allows specification of input parameters but also serves to define the entire flowsheet (and economic) model for the ASPEN PLUS simulation program. A listing of this file, with base case parameters, is provided in Appendix I.

The ASPEN PLUS input language file provides a great deal of flexibility in changing model parameters. Literally, thousands of model parameters can be altered by changing entries in the file. This great flexibility can lead to potential problems. The most troublesome is to distinguish the parameters which were not intended to be altered by changing the file entries from those parameters which were intended to be variable inputs to the model.

The model was constructed with a certain set parameters in mind as primary model variables. These parameters have been highlighted in the input language file in several ways. One method employed was to construct a FORTRAN code block in which FORTRAN variables are set to important process parameters such as reactor temperature, pressure, and residence time.

The following is a complete list of these parameters:

- **pr** — The desired pressure of the reactor vessel in pascals (Pa). This pressure also sets the pressure of the HTR heater and the heat recovery heat exchangers.
- **tr** — The desired temperature at the outlet of the HTR heater and the inlet of the reactor in degrees celcius (°C).
- **pri** — The pressure of operation for the PRH vapor driven preheat vessel in pascals (Pa). This pressure will also set the pressure of the secondary water separation vessel, SSU.
- **tihtr** — The desired temperature for the stream entering the secondary water separation unit, SSU. Heating is provided by hot vapor recycle. The amount of recycle is set by the DESIGN-SPEC SPEC1. If their is insufficient energy in the vapor recycle, or if vapor recycle is undesirable, SPEC1 can be deactivated and the heating will be provided by the fired heater IHTR.
- **rtres** — Desired residence time in the high pressure reactor vessel in minutes.
- **wor1** — Desired water-to-oil ratio on a weight basis exiting the first separator, PSU. The water mass used in this ratio includes both free water and water dissolved in the oil phase; however, the water-to-oil ratio will not be lowered below the point where the free water phase disappears, regardless of the input value. Water in the vapor phase is not considered in computing the ratios.
- **wor2** — Desired water-to-oil ratio on a weight basis exiting the second separator, SSU. See comments above in wor1.
- **wor3** — Desired water-to-oil ratio on a weight basis exiting the reactor vessel, RU. See comments above in wor1.
- **wor4** — Desired water-to-oil ratio on a weight basis exiting the final separator, WSU. See comments above in wor1.
- **worf** — Desired water-to-oil ratio on a weight basis in the final PRD-O. This feature is not active in the base case. It can be activated by uncommenting the DESIGN-SPEC SPEC2. SPEC2 attempts to match the desired final water content by varying the temperature difference between streams HTXG-E and HTXO-O, entering and leaving the heat exchanger HTXO. If this feature is activated the user should clearly understand the interaction between the worf and the wor4 parameters.
- **prc** — Delivery pressure for final oil. Used to set outlet pressure of CPMP pump.
- **fgr** — Sets initial guess for fraction of reactor gas recycled to the PRH unit. If the DESIGN-SPEC SPEC1 is commented out this will be the recycle ratio used.
- **tao** — Set the approach temperature (°C) for the liquid stream heat recovery heat exchanger, HTXO. The approach temperature of exiting stream HTXO-O and entering stream HTXG-E is set by this parameter.

- **tag** — Set the approach temperature (°C) for the vapor stream heat recovery heat exchanger, HTXG. The approach temperature of exiting stream HTXG-G and entering stream PUMP-E is set by this parameter.
- **wtfc & wtfo** — Sets the stoichiometry for the coke and product oil, respectively, in the pyrolysis reactions. These are ratio of weights of product to weights of reacted oil. The remaining product is assumed to be composed of gas/vapor components.
- **xch4, xc2h6, xc3h8, xco2, xh2, xhhc, & xh2s** — Relative moles of gas/vapor produced by the pyrolysis reaction. These constituents represents all products of the pyrolysis except for coke and product oil. Each parameter defines the moles of component. The name of the parameter is constructed from the name of the individual component by adding the prefix 'x' (i.e. xch4 is the parameter for component CH4). The values input are normalized to give the composition of the products, exclusive of the coke and product oil.
- **akin** — Pre-exponential factor in the Arrhenius' rate expression for the pyrolysis reactions. The units are reciprocal seconds (s^{-1}). The activation energy is set in the RU1, RU2 and RU3 RCSTR blocks.

The second type of model variables are parameters set using the ASPEN PLUS input language directly, and as a result cannot be centralized in one section of the input file. These inputs have been divided into several categories. An index of the inputs is supplied in comments in the input file, while the actual lines on which parameters are to be entered are designated with a comment of the form 'Data #n' where 'n' is a number. These locations are easily found using an editor. Those parameters designated with numbers 0-2 need to be changed if the crude oil composition or feed flow rate changes, while those with numbers between 21-28 are less likely to need modification. Note the parameters with numbers greater than 100 are cost parameters only and are described later.

Primary process model input parameters found on or below 'Data #n' lines:

- **0** — The temperature, pressure and flow rates of feed water and CRUDE. CRUDE is the name given to the input oil defined by pseudocomponents (see below). H2O is the name used for water. The default units are celcius, pascals, and kilograms per second. However any units known to ASPEN PLUS[4] may be used if properly designated.
- **1** — This section is used to define the properties of the crude oil components. These are set using the ASPEN PLUS Assay Data Analysis and Pseudocomponent Correlation System. In the base case example in Appendix I the crude oil is characterized using API gravity and D86 or vacuum distillation curves for six cuts. The BLEND CRUDE statement is used to define amounts of these cuts in the composite CRUDE used in the model.

- 2 — This section defines water solubility parameters for the crude oil components. These values override the values assigned automatically by ASPEN PLUS.

Secondary process model input parameters found on or below 'Data #n' lines:

- 21 — Redefine activation energies or power law exponent for each reaction. Note that the pre-exponential term can be redefined only if appropriate coding is removed from the FINPUT FORTRAN block.
- 22 — Heat transfer coefficients for heat recovery heat exchangers. These parameters only effect the computed heat exchanger surface. They do not directly influence the material and energy balance calculations since the approach temperatures are used to determine operation of the heat exchangers. However, these parameters will influence the cost calculations.
- 23 — Parameters which can influence the exact size of the reactor vessel. Changes can have a small influence on energy and material balances by slightly modifying the actual residence time in the reactor. These parameters have more influence on cost calculations.
- 24 — Outlet temperature of process stream from CONDL unit.
- 25 — Outlet temperature of process stream from CONDH unit.
- 26 — Flash pressure at the CSU unit.
- 27 — Set the API gravity and normal boiling point (°C) of the pyrolysis product oil P-OIL.
- 28 — Define the two pyrolysis reaction products COKE and HHC. The FORTRAN block SETR will compute the proper stoichiometric coefficients for the products chosen. The components are changed by changing the middle name. This name must be in the ASPEN PLUS data base or defined elsewhere in the input. (e.g., to change the HHC component from normal butane to normal pentane change the entry from 'HHC C4H10-1 HHC' to 'HHC C5H12-1 HHC').

ECONOMIC MODEL

An economic evaluation of the process has been incorporated into the model using procedures available in the ASPEN PLUS system. The economic calculations draw directly on results from the process simulation to compute equipment size/cost and operating costs. Cash flow analysis and profitability analysis are carried out using procedures outlined in the ASPEN PLUS reference manual[5]. A start date for the project of June 1994 was used.

In general, default values available from ASPEN PLUS are used for various cost factors and economic parameters related to the estimation of total investment and profitability. However, some default values are overridden:

- Economic life is set to 20 years.
- Costs associated with land acquisition and development are set to zero. It is assumed the process will be installed at an existing oil collection/processing facility.
- The cost for service buildings is set to zero. It is assumed general service buildings are already available.
- The total capital contingency factor is set to 15% of the Total Direct and Indirect Cost.
- Maintenance costs, including supplies and labor, are set at 4% of the Total Depreciable Costs per year.
- A single factor of 4% of the Total Depreciable Costs per year is used for property taxes, insurance, and general overhead.
- The interest rate on equity is set to 10%.

Cost tables available within ASPEN PLUS are used for some equipment while other estimates are obtained from correlations taken from Walas[6]. The Walas correlations were used when high pressures were involved. Many equipment cost estimates in ASPEN PLUS do not adequately cover the high pressure range, and it can result in considerable cost under estimation. FORTRAN routines which are based on the Walas' correlations have been developed. Listings of these routines are presented in Appendix II. These routines are called during the ASPEN PLUS execution to establish equipment costs as needed. In the following discussion these routines are referred to as Walas Cost Routines.

Even though considerable detail on the cost analysis is reported by ASPEN PLUS, it should be remembered that the estimates are only somewhere between an "order of magnitude estimate" and a "study estimate." The basic approach used is based on estimating purchase costs of major pieces of equipment and using standard factors for installation, instrumentation, etc.

Sizing of all items include a peak allowance capacity factor. This factor defines the ratio between nominal design flows and volumes, and the flows and volumes used to size equipment. For those units costed using ASPEN PLUS algorithms, this capacity factor is set by the "PEAK" parameter; for others, this is set by the variable "capfac" in the FORTRAN INPUT block. The ASPEN PLUS default value for "PEAK" is 1.06. This value is used in the base case.

Several of the vessels are sized based on entrainment limits, as well as residence time considerations. A series of FORTRAN routines were coded to allow the entrainment limits to be incorporated with residence time specifications in determining vessel sizes. These routines are based on an

article by R. N. Watkins[7] and consider both horizontal or vertical vessels arrangements. A listing of the routines is provided in Appendix III.

Each unit shown in the process flowsheet, Fig. 1, is costed. The general methods used for each unit are outlined below. In a number of cases the equipment sizing and costing do not use standard ASPEN PLUS algorithms, and as a result certain information in the ASPEN PLUS generated output file are not valid for these units. These items are noted for each unit.

PSU Unit (Cost Block C-PSU)

This initial separation vessel is costed as a tank with enough residence time to hold 60 minutes of incoming flow (stream FEED). The ASPEN PLUS cost block TANK is used to obtain the unit cost. All ASPEN PLUS output for this unit is valid.

PRH Unit (Cost Block C-PRH)

The size of this preheat vessel is determined by using the Watkins routines and assuming a 5 minute liquid residence time. Vapor and liquid flow rates and properties are set to those in the stream exiting the unit (stream PRH-E). The sizing routine is call by FORTRAN block FCPRH. The ASPEN PLUS cost block V-VESSEL is used to cost the unit. The vessel is assumed to be carbon steel with a corrosion allowance, 0.32 cm (0.125 in.) for the base case. ASPEN PLUS cost output related to this unit is valid except for "PEAK CAPACITY" ALLOWANCE FACTOR" and "VELOCITY RATIO."

SSU Unit (Cost Block C-SSU)

This separator vessel is sized using a 5 minute liquid residence time and assuming the vessel is 60% full. The liquid flow rate is obtained from stream PGSU-E. The ASPEN PLUS cost block H-VESSEL is used to cost the unit. A charge of 10% of the purchase cost has been added for vessel internals. All ASPEN PLUS cost output related to this unit is valid.

IPMP Unit (Cost Block C-IPMP)

This low pressure pump size is based on a the IPMP-E stream flow and the pressure difference between the PRH and PSU units. The ASPEN PLUS cost block PUMP, assuming a stainless steel SS-ANSI pump type, is used to cost the unit. The pump is assumed to be driven by an electric motor. All ASPEN PLUS cost output related to this unit is valid.

CPMP Unit (Cost Block C-CPMP)

This low pressure pump size is based on a the COIL-O stream flow and the pressure difference between the COIL unit and the desired product pressure. The ASPEN PLUS cost block PUMP, assuming a carbon steel SS-ANSI pump type, is used to cost the unit. The pump is assumed to be driven by an electric motor. All ASPEN PLUS cost output related to this unit is valid.

FPMP Unit (Cost Block C-FPMP)

This low pressure pump size is based on a the CWSU-O stream flow and the pressure difference between the WSU unit and the desired product pressure. The ASPEN PLUS cost block PUMP, assuming a carbon steel SS-ANSI pump type, is used to cost the unit. The pump is assumed to be driven by an electric motor. All ASPEN PLUS cost output related to this unit is valid.

PUMP Unit (Cost Block C-PUMP)

The high pressure pump cost was based on the Walas Cost routine "usrpmp." This routine is call by FORTRAN block FC-PUMP. The MXAA-E stream flow rate and the pressure difference between this stream and stream PUMP-E is used to size the pump. The pump is assumed to be stainless steel construction and driven by an electric motor. Because of the use of the Walas Cost Routine, the following ASPEN PLUS cost output is not valid for this unit "PUMP-TYPE," "MATERIAL OF CONSTRUCTION," "PEAK CAPACITY ALLOWANCE FACTOR," "MATERIAL OF CONSTRUCTION FACTOR," "PUMP EFFICIENCY," "MOTOR EFFICIENCY," "POWER REQUIRED PER PUMP,"and "CARBON STEEL COST."

HTXG Unit (Cost Block C-HTXG)

This heat recovery heat exchanger is sized (the required heat transfer area) by the ASPEN PLUS unit-operation block HEATX (user block HTXG). The cost is obtained from the Walas Cost routine "usrht1," called by FORTRAN block FCHTXG. The materials of construction are assumed to be stainless steel. Because of the use of the Walas Cost routine, the following ASPEN PLUS cost output is not valid for this unit "HEAT EXCHANGER TYPE," "SHELL MATERIAL," "TUBE MATERIAL," PEAK CAPACITY FACTOR," "NUMBER OF SHELL PASSES," NUMBER OF TUBE PASSES," "HEAT TRANSFER COEFFICIENT," "MATERIAL OF CONSTRUCTION FACTOR," and "CARBON STEEL COST."

HTXO Unit (Cost Block C-HTXO)

This heat recovery heat exchanger is sized by the ASPEN PLUS unit-operation block HEATX (user block HTXO). The cost is obtained from the Walas Cost routine "usrht1," called by FORTRAN block FCHTXO. The materials of construction are assumed to be stainless steel. The list of ASPEN PLUS cost output which is not valid is the same as that for the HTXG unit above.

HTR Unit (Cost Block C-HTR)

This fired heater is sized using the computed heat stream HTR-Q and an assumed efficiency. The base case efficiency factor is 0.75. The cost is obtained from the Walas Cost routine "usrhtr," called by FORTRAN block FCHTR. This routine is used because the ASPEN PLUS cost correlations do not go to a high enough pressure. The materials of construction are assumed to be carbon steel since no free water is present at this stage of the process. Because of the use of the Walas Cost routine, the following ASPEN PLUS cost output is not valid for this unit "FIRED HEATER TYPE," PEAK CAPACITY FACTOR," "MATERIAL OF CONSTRUCTION FACTOR," "THERMAL EFFICIENCY," and "CARBON STEEL COST."

RU Unit (Cost Block C-RU)

The size of the reactor vessel is determined by using the Watkins routines and assuming a specified liquid residence time. Vapor and liquid flow rates and properties are set to the average of those in streams entering and exiting the reactor, streams HTR-E and RU-E respectively (note RU-E is equal to the sum of streams RUGS-G and MXB-O). The sizing routine is call by FORTRAN block FRU. The ASPEN PLUS cost block H-VESSEL is used to cost the unit. The vessel is assumed to be carbon steel with a corrosion allowance, 0.32 cm (0.125 in.) for the base case. ASPEN PLUS cost output related to this unit is valid except for "PEAK CAPACITY ALLOWANCE FACTOR."

FOIL Unit (Cost Block C-FOIL)

The size of this flash vessel is determined by using the Watkins routines and assuming a 5 minute liquid residence time. Vapor and liquid flow rates and properties are set to those in the stream exiting the unit, streams FOIL-O and -G. The sizing routine is called by FORTRAN block FCFOIL. The ASPEN PLUS cost block H-VESSEL is used to cost the unit. The vessel is assumed to be carbon steel with a corrosion allowance, 0.32 cm (0.125 in.) for the base case. ASPEN PLUS cost output related to this unit is valid except for "PEAK CAPACITY ALLOWANCE FACTOR."

COIL Unit (Cost Block C-COIL)

This air cooler used to cool the oil stream is sized and costed using the aspen cost block AIRCOOL. Carbon steel construction is assumed. All ASPEN PLUS output for this unit is valid.

CONDH Unit (Cost Block C-CONDH)

This heat exchanger used to cool the mostly condensed vapor stream HTXG-G is sized by the ASPEN PLUS unit-operation block HEATX (user block CONDH). The cost is obtained from the Walas Cost routine "usrht1," called by FORTRAN block FCCONDH. The materials of construction are assumed to be stainless steel. The list of ASPEN PLUS cost output which is not valid is the same as that for the HTXG unit above.

CSU Unit (Cost Block C-CSU)

The size of this flash vessel is determined by using the Watkins routines and assuming a 5 minute liquid residence time. Vapor and liquid flow rates and properties are set to those in the stream exiting the unit, streams CSU-O -FW and -G. The sizing routine is call by FORTRAN block FCCSU. The ASPEN PLUS cost block H-VESSEL is used to cost the unit. The vessel is assumed to be carbon steel with a corrosion allowance, 0.32 cm (0.125 in.) for the base case. ASPEN PLUS cost output related to this unit is valid except for "PEAK CAPACITY ALLOWANCE FACTOR" and "LIQUID STREAM FLOW."

CONDL Unit (Cost Block C-CONDL)

This unit condenses vapor from the FOIL flash. The unit is sized and costed by ASPEN PLUS unit-operation and cost blocks HEATX. The materials of construction are assumed to be stainless steel. All ASPEN PLUS output for this unit is valid.

WCU Unit (Cost Block C-WCU)

This unit cools the water stream, SSU-W2, exiting the oil-water spearator SSU. The unit is sized and costed by ASPEN PLUS unit-operation and cost blocks HEATX. The materials of construction of the shell and tube heat exchanger are assumed to be stainless steel and carbon steel respectively. All ASPEN PLUS output for this unit is valid.

WSU Unit (Cost Block C-WSU)

This separator vessel is sized using a 5 minute liquid residence time and assuming the vessel is 60% full. The liquid flow rate is obtained from streams CONDL and CSU -O and -FW. The ASPEN PLUS cost block H-VESSEL is used to cost the unit. A charge of 10% of the purchase cost has been added for vessel internals. All ASPEN PLUS cost output related to this unit is valid.

Other Cost Parameters

As with the process parameters cost related parameters are input to the model through the FORTRAN INPUT block and through labeled lines. Only one purely cost related parameter is set by the a FORTRAN INPUT variable. This is capfac which is the capacity factor used by all non ASPEN PLUS sizing/costing routines. It is equivalent to the PEAK parameter used in ASPEN PLUS blocks. In the base case it is set to equal the ASPEN PLUS default value of 1.06.

A number of cost parameters which are input using the ASPEN PLUS input language are noted in the input file using the 'Data #1nm' tag. The 'nm' is a particular number. These lines include the following:

- **101** — Residence time of the PSU, PRH, SSU, and WSU units. Base case values are 5 minutes for the PRH, SSU and WSU units and 60 minutes for the PSU unit.
- **102** — Utility costs. Base case values for January 1994: electricity $0.05/kW-hr, natural gas $2.1/GJ, and cooling water $0.088/Mg ($0.04/thousand-lbm).
- **103** — Labor costs and number of operators. Base case values for June 1991 is $16/hr for operators and $20/hr for June 1995 construction labor. The number of operators is set at one.
- **104** — Corrosion allowances. Base case 0.32 cm (0.125 in.)
- **105** — Parameters used by the Watkins' routines in sizing the FOIL flash vessel.
- **106** — Parameters used by the Watkins' routines in sizing the CSU flash vessel.
- **107** — Cooling water temperatures. The base case uses a supply temperature of 16°C (60°F) and a return temperature of 60°C (140°F).
- **108** — Fired heater HTR efficiency. The efficiency is defined as the heat transferred to the process fluid divided by the heat of combustion of the consumed natural gas. The base case uses 0.75.
- **109** — Parameters used by the Watkins' routines in sizing the PSU preheat vessel.
- **110** — Pump type for the ASPEN PLUS sized units IPMP, FPMP and CPMP.

- **111** — Cost factor, applied to purchase price, for internals for separation units SSU and WSU.

For the most part ASPEN PLUS default values were used for other cost parameters. However, several default values were changed:

- **COST-INDEX** — ASPEN PLUS version 9.2-1 contains cost index information up to March 1995. For dates beyond that a 0.03 escalation factor was used for all indexes except the "FUEL" index which was set at 0.05.
- **CONTINGENCY** — As defined by ASPEN PLUS the "PROJECT-BASIS" contingency was set at 0.05 and the "PROJECT-DEFINITION" contingency at 0.10.
- **OPERATING-COST** — "OTHER-LABOR MAINTENANCE" is set to zero since maintenance costs are included in "SUPPLIES MAINTENANCE FACTOR" which is set to 0.04. The "GENERAL-WORKS" parameters "GEN-ADMIN" and "TAX" are set to zero and these costs are lumped together in the "ADDITIONAL FACTOR" which is set to 0.04.

ASPEN PLUS OUTPUT FILES

ASPEN PLUS produces a number of output files upon successful execution. The complete list is given in the ASPEN PLUS documentation[8]. The results of the simulation are present in the file "name.rep," where name is the user specified name. This file contains results originating from the ASPEN PLUS as well as results from user subroutines. User subroutines usrht1, usrpmp, usrhtr, sepv, and seph also write information to this file. These results are located near the beginning of the file just after the table of contents. The user routine output gives information about calculations performed by the user routines. These results are labeled according to the process unit with which they are associated.

Appearing just after the user results from routines associated with process units is a set of results computed by the FORTRAN block OUT. These results summarize selected performance information. At the top is a short section which summarizes the performance of HTXG and HTXO heat exchangers. This is followed by the fraction of reactor vapor flow used in the PRH vessel. Next the amount of water in the liquid oil phase at the reactor exit and in the product stream PRD-O are reported. Next is information on free water flow, vapor flow exiting the reactor, and summary injection, and production oil flows and gravity. A section on gas composition of the FUEL-G stream follows along with the energy usage of the HTR fired heater and the fraction of reaction product P-OIL which exits in the FUEL-G stream. The last section reports some economic numbers which give the required increase in oil value that leads to a profitable operation (as defined by the cash flow analysis).

Two numbers are given: one assuming the oil has no initial value, and one assuming that the input oil is worth $64.6/Mg ($10/bbl). Since some oil is destroyed in the process, the required profit is a function of the feed oil value.

BASE CASE RESULTS

The model input language file given in Appendix I was run with ASPEN PLUS version 9.2-1 software. ASPEN PLUS produces an output file (name.rep) of more than one hundred pages. It reports results for all process streams, unit-operation blocks, cost blocks, and cash flow analysis. In the following two subsections, selected results from this output file for the base case conditions are presented.

Flowsheet Results

Figure 8 lists computed results for all major process flow streams, including the energy streams. The location of these streams can be found by referring to the overall process flowsheets representations given in Figs. 1 and 2. Specific information on detailed streams within the model system can be found by referring to the ASPEN PLUS module layouts given in Figs. 3-7.

STREAM ID	COIL-O	CONDH-G	CONDH-FW	CONDH-O	CONDL-FW
TOTAL STREAM:					
KG/SEC	2.9457	0.0	0.4292	2.0117	5.6752-02
PHASE:	LIQUID	MISSING	LIQUID	LIQUID	LIQUID
COMPONENTS: KG/SEC					
H2O	1.1362-03	0.0	0.4292	1.2256-04	5.6752-02
CH4	3.9074-05	0.0	0.0	6.2683-02	0.0
C2H6	1.0152-04	0.0	0.0	3.9435-02	0.0
C3H8	1.3433-04	0.0	0.0	4.6943-02	0.0
HHC	2.3027-04	0.0	0.0	4.0464-02	0.0
H2S	3.0895-05	0.0	0.0	1.6994-02	0.0
CO2	2.5334-05	0.0	0.0	3.2109-02	0.0
H2	3.4292-07	0.0	0.0	6.7517-04	0.0
P-OIL	2.4868-02	0.0	0.0	0.6814	0.0
HNAPH	1.6349-02	0.0	0.0	0.1646	0.0
KERO	4.0230-02	0.0	0.0	0.1892	0.0
AGO	0.1568	0.0	0.0	0.3186	0.0
LVGO	0.3723	0.0	0.0	0.2682	0.0
HVGO	0.8517	0.0	0.0	0.1326	0.0
VR	1.1827	0.0	0.0	1.7608-02	0.0
TOTAL FLOW:					
KMOL/SEC	6.8163-03	0.0	2.3828-02	1.9667-02	3.1502-03
KG/SEC	2.6469	0.0	0.4292	2.0117	5.6752-02
CUM/SEC	2.7878-03	0.0	4.3160-04	3.1677-03	5.7061-05
STATE VARIABLES:					
TEMP C	93.3333	MISSING	23.8888	23.8888	23.8888
PRES N/SQM	2.0684+05	1.3780+07	1.3780+07	1.3780+07	2.0684+05
VFRAC	0.0	MISSING	0.0	0.0	0.0
LFRAC	1.0000	MISSING	1.0000	1.0000	1.0000
SFRAC	0.0	MISSING	0.0	0.0	0.0
MASSVFRA	0.0	MISSING	0.0	0.0	0.0
DENSITY:					
KMOL/CUM	2.4450	MISSING	55.2078	6.2087	55.2078
KG/CUM	949.4536	MISSING	994.5850	635.0885	994.5850
AVG MW	388.3201	MISSING	18.0152	102.2900	18.0152
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.2988	0.0	0.0	0.0	0.0

Figure 8. Base case material and energy streams.

STREAM ID	CONDL-G	CONDL-O	CPMP-O	CSU-FW	CSU-G
TOTAL STREAM:					
KG/SEC	5.4992-02	0.3640	2.9457	0.4279	0.2067
PHASE:	VAPOR	LIQUID	LIQUID	LIQUID	VAPOR
COMPONENTS: KG/SEC					
H2O	5.5571-04	6.4248-06	1.1362-03	0.4279	1.4303-03
CH4	1.6644-02	2.2042-04	3.9074-05	0.0	6.1704-02
C2H6	1.0524-02	6.5765-04	1.0152-04	0.0	3.6569-02
C3H8	8.1526-03	1.6470-03	1.3433-04	0.0	3.7107-02
HHC	4.2730-03	4.0504-03	2.3027-04	0.0	1.7609-02
H2S	2.8270-03	4.3717-04	3.0895-05	0.0	1.4118-02
CO2	9.1231-03	2.2421-04	2.5334-05	0.0	3.1123-02
H2	2.2420-04	3.3380-07	3.4292-07	0.0	6.7406-04
P-OIL	2.5980-03	0.1877	2.4868-02	0.0	6.2205-03
HNAPH	6.2446-05	4.8467-02	1.6349-02	0.0	1.3456-04
KERO	7.1810-06	4.7524-02	4.0230-02	0.0	1.7399-05
AGO	5.1397-07	5.3167-02	0.1568	0.0	1.7785-06
LVGO	2.7063-09	1.7498-02	0.3723	0.0	2.2180-08
HVGO	1.0028-12	2.3668-03	0.8517	0.0	2.6888-11
VR	0.0	8.8961-05	1.1827	0.0	0.0
TOTAL FLOW:					
KMOL/SEC	2.1025-03	2.7911-03	6.8163-03	2.3754-02	7.8002-03
KG/SEC	5.4992-02	0.3640	2.6469	0.4279	0.2067
CUM/SEC	2.4867-02	4.4915-04	2.7875-03	4.2779-04	9.0317-02
STATE VARIABLES:					
TEMP C	23.8888	23.8888	93.9652	17.9158	17.9158
PRES N/SQM	2.0684+05	2.0684+05	1.0340+06	2.0684+05	2.0684+05
VFRAC	1.0000	0.0	0.0	0.0	1.0000
LFRAC	0.0	1.0000	1.0000	1.0000	0.0
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	1.0000	0.0	0.0	0.0	1.0000
DENSITY:					
KMOL/CUM	8.4548-02	6.2143	2.4452	55.5261	8.6364-02
KG/CUM	2.2114	810.6073	949.5552	1000.3185	2.2887
AVG MW	26.1557	130.4410	388.3201	18.0152	26.5007
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.0	0.0	0.2988	0.0	0.0

Figure 8. (continued). Base case material and energy streams.

STREAM ID	CSU-O	CWSU-O	FEED	FOIL-G	FOIL-O
TOTAL STREAM:					
KG/SEC	1.8064	2.1921	60.4725	0.4548	2.9457
PHASE:	LIQUID	LIQUID	LIQUID	VAPOR	LIQUID
COMPONENTS: KG/SEC					
H2O	2.3491-05	2.1704-02	55.0977	5.6235-02	1.1362-03
CH4	9.7843-04	1.1988-03	0.0	8.0163-03	3.9074-05
C2H6	2.8663-03	3.5240-03	0.0	8.1644-03	1.0152-04
C3H8	9.8354-03	1.1482-02	0.0	7.8249-03	1.3433-04
HHC	2.2855-02	2.6906-02	0.0	7.7205-03	2.3027-04
H2S	2.8760-03	3.3131-03	0.0	2.4160-03	3.0895-05
CO2	9.8610-04	1.2103-03	0.0	5.2482-03	2.5334-05
H2	1.1072-06	1.4410-06	0.0	9.4527-05	3.4292-07
P-OIL	0.6751	0.8629	0.0	0.1900	2.4868-02
HNAPH	0.1644	0.2129	0.2659	4.8439-02	1.6349-02
KERO	0.1891	0.2367	0.3257	4.7513-02	4.0230-02
AGO	0.3186	0.3717	0.6434	5.3164-02	0.1568
LVGO	0.2682	0.2857	0.8671	1.7498-02	0.3723
HVGO	0.1326	0.1350	1.4412	2.3668-03	0.8517
VR	1.7608-02	1.7697-02	1.8312	8.8961-05	1.1827
TOTAL FLOW:					
KMOL/SEC	1.1941-02	1.5935-02	3.0746	7.0908-03	6.8163-03
KG/SEC	1.8064	2.1921	60.4725	0.4548	2.6469
CUM/SEC	2.1376-03	2.6079-03	6.1748-02	0.1569	3.2445-03
STATE VARIABLES:					
TEMP C	17.9158	18.8297	37.7777	284.0831	284.0831
PRES N/SQM	2.0684+05	2.0684+05	6.8948+04	2.0684+05	2.0684+05
VFRAC	0.0	0.0	0.0	1.0000	0.0
LFRAC	1.0000	1.0000	1.0000	0.0	1.0000
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	0.0	0.0	0.0	1.0000	0.0
ENTHALPY:					
J/KMOL	-2.5481+08	-2.5066+08	-2.8618+08	-1.5961+08	-4.3459+08
J/KG	-1.6844+06	-1.8222+06	-1.4550+07	-2.4885+06	-1.1191+06
WATT	-3.0427+06	-3.9944+06	-8.7990+08	-1.1318+06	-2.9623+06
ENTROPY:					
J/KMOL-K	-9.3766+05	-8.5657+05	-1.6949+05	-2.4426+05	-1.9141+06
J/KG-K	-6198.3405	-6226.6539	-8617.6019	-3808.2895	-4929.1000
DENSITY:					
KMOL/CUM	5.5863	6.1104	49.7936	4.5186-02	2.1008
KG/CUM	845.0744	840.5804	979.3403	2.8982	815.8198
AVG MW	151.2754	137.5650	19.6679	64.1396	388.3201
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.0	0.0	0.0	0.0	0.2988

STREAM ID	FPMP-O	FUEL-G	GASR-G1	GASR-G2	HTR-E
TOTAL STREAM:					
KG/SEC	2.1921	0.2617	0.5212	2.4410	6.3628
PHASE:	LIQUID	MIXED	VAPOR	MIXED	MIXED
COMPONENTS: KG/SEC					
H2O	2.1704-02	1.9860-03	9.1685-02	0.4293	0.5784
CH4	1.1988-03	7.8348-02	1.3384-02	6.2683-02	4.5366-03
C2H6	3.5240-03	4.7093-02	8.4205-03	3.9435-02	5.4032-03
C3H8	1.1482-02	4.5260-02	1.0024-02	4.6943-02	8.0488-03
HHC	2.6906-02	2.1882-02	8.6402-03	4.0464-02	8.0374-03
H2S	3.3131-03	1.6945-02	3.6287-03	1.6994-02	2.7805-03
CO2	1.2103-03	4.0246-02	6.8562-03	3.2109-02	2.7571-03
H2	1.4410-06	8.9826-04	1.4417-04	6.7517-04	1.4157-05
P-OIL	0.8629	8.8185-03	0.1455	0.6814	0.1451
HNAPH	0.2129	1.9701-04	3.5151-02	0.1646	0.3009
KERO	0.2367	2.4580-05	4.0400-02	0.1892	0.3661
AGO	0.3717	2.2924-06	6.8031-02	0.3186	0.7115
LVGO	0.2857	2.4886-08	5.7280-02	0.2682	0.9244
HVGO	0.1350	2.7890-11	2.8323-02	0.1326	1.4696
VR	1.7697-02	0.0	3.7599-03	1.7608-02	1.8350
TOTAL FLOW:					
KMOL/SEC	1.5935-02	9.9026-03	9.2874-03	4.3495-02	5.1696-02
KG/SEC	2.1921	0.2617	0.5212	2.4410	6.3628
CUM/SEC	2.6090-03	0.1152	2.9435-03	1.3785-02	1.5771-02
STATE VARIABLES:					
TEMP C	19.7494	19.2504	422.5409	422.5409	437.8000
PRES N/SQM	1.0340+06	2.0684+05	1.3780+07	1.3780+07	1.3780+07
VFRAC	0.0	0.9999	1.0000	1.0000	0.6587
LFRAC	1.0000	7.2014-05	0.0	1.0003-07	0.3412
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	0.0	0.9999	1.0000	1.0000	0.3184
DENSITY:					
KMOL/CUM	6.1078	8.5957-02	3.1551	3.1551	3.2779
KG/CUM	840.2285	2.2716	177.0756	177.0756	403.4613
AVG MW	137.5650	26.4274	56.1222	56.1222	123.0828

Figure 8. (continued). Base case material and energy streams.

STREAM ID	HTXG-E	HTXG-G	HTXO-E	HTXO-O	IPMP-E
TOTAL STREAM:					
KG/SEC	6.3628	2.4410	6.3628	3.4005	6.9872
PHASE:	LIQUID	MIXED	MIXED	LIQUID	LIQUID
COMPONENTS: KG/SEC					
H2O	0.5784	0.4293	0.5784	5.7371-02	1.6124
CH4	4.5366-03	6.2683-02	4.5366-03	8.0553-03	0.0
C2H6	5.4032-03	3.9435-02	5.4032-03	8.2660-03	0.0
C3H8	8.0488-03	4.6943-02	8.0488-03	7.9592-03	0.0
HHC	8.0374-03	4.0464-02	8.0374-03	7.9507-03	0.0
H2S	2.7805-03	1.6994-02	2.7805-03	2.4469-03	0.0
CO2	2.7571-03	3.2109-02	2.7571-03	5.2735-03	0.0
H2	1.4157-05	6.7517-04	1.4157-05	9.4870-05	0.0
P-OIL	0.1451	0.6814	0.1451	0.2148	0.0
HNAPH	0.3009	0.1646	0.3009	6.4789-02	0.2659
KERO	0.3661	0.1892	0.3661	8.7743-02	0.3257
AGO	0.7115	0.3186	0.7115	0.2100	0.6434
LVGO	0.9244	0.2682	0.9244	0.3898	0.8671
HVGO	1.4696	0.1326	1.4696	0.8541	1.4412
VR	1.8350	1.7608-02	1.8350	1.1828	1.8312
TOTAL FLOW:					
KMOL/SEC	5.1696-02	4.3495-02	5.1696-02	1.3907-02	0.1057
KG/SEC	6.3628	2.4410	6.3628	3.1017	6.9872
CUM/SEC	8.1749-03	3.8768-03	9.8356-03	4.2424-03	7.2326-03
STATE VARIABLES:					
TEMP C	269.8346	117.3484	317.0457	303.1346	38.0129
PRES N/SQM	1.3780+07	1.3780+07	1.3780+07	1.3780+07	6.9000+05
VFRAC	0.0	3.3614-03	0.2932	0.0	0.0
LFRAC	1.0000	0.9966	0.7067	1.0000	1.0000
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	0.0	1.3656-03	0.1425	0.0	0.0
DENSITY:					
KMOL/CUM	6.3237	11.2192	5.2560	3.2781	14.6264
KG/CUM	778.3435	629.6516	646.9247	731.1273	966.0857
AVG MW	123.0828	56.1222	123.0828	223.0313	66.0504
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.0	0.0	0.0	0.2988	0.0

STREAM ID	MXA-E	MXAA-E	MXB-O	MXC-G	PGSU-E
TOTAL STREAM:					
KG/SEC	6.9872	6.3628	3.4005	0.4758	7.4874
PHASE:	LIQUID	MIXED	MIXED	MIXED	LIQUID
COMPONENTS: KG/SEC					
H2O	1.6124	0.5784	5.7371-02	5.7314-02	1.7030
CH4	0.0	4.5366-03	8.0553-03	1.6864-02	4.5366-03
C2H6	0.0	5.4032-03	8.2660-03	1.1182-02	5.4032-03
C3H8	0.0	8.0488-03	7.9592-03	9.7996-03	8.0488-03

HHC	0.0	8.0374-03	7.9507-03	8.3233-03	8.0374-03
H2S	0.0	2.7805-03	2.4469-03	3.2642-03	2.7805-03
CO2	0.0	2.7571-03	5.2735-03	9.3473-03	2.7571-03
H2	0.0	1.4157-05	9.4870-05	2.2454-04	1.4157-05
P-OIL	0.0	0.1451	0.2148	0.1903	0.1451
HNAPH	0.2659	0.3009	6.4789-02	4.8529-02	0.3009
KERO	0.3257	0.3661	8.7743-02	4.7531-02	0.3661
AGO	0.6434	0.7115	0.2100	5.3167-02	0.7115
LVGO	0.8671	0.9244	0.3898	1.7498-02	0.9244
HVGO	1.4412	1.4696	0.8541	2.3668-03	1.4696
VR	1.8312	1.8350	1.1828	8.8961-05	1.8350
TOTAL FLOW:					
KMOL/SEC	0.1057	5.1696-02	1.3907-02	8.0438-03	0.1141
KG/SEC	6.9872	6.3628	3.1017	0.4758	7.4874
CUM/SEC	7.2331-03	6.9633-03	4.2038-03	0.1754	8.1570-03
STATE VARIABLES:					
TEMP C	37.7777	76.7789	422.5409	275.3309	76.7766
PRES N/SQM	6.8948+04	6.9000+05	1.3780+07	2.0684+05	6.9000+05
VFRAC	0.0	8.1933-08	7.3343-07	0.9999	0.0
LFRAC	1.0000	1.0000	1.0000	3.9598-05	1.0000
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	0.0	1.4685-08	1.4473-07	0.9997	0.0
DENSITY:					
KMOL/CUM	14.6253	7.4240	3.3082	4.5838-02	13.9905
KG/CUM	966.0111	913.7691	737.8383	2.7115	917.9158
AVG MW	66.0504	123.0828	223.0313	59.1539	65.6098
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.0	0.0	0.2988	0.0	0.0

Figure 8. (continued). Base case material and energy streams.

STREAM ID	PGSU-G	PSU-W2	PRD-O	PUMP-E	RUGS-G
TOTAL STREAM:					
KG/SEC	2.1025-02	53.4852	5.1379	6.3628	2.9622
PHASE:	VAPOR	LIQUID	LIQUID	LIQUID	VAPOR
COMPONENTS: KG/SEC					
H2O	1.0791-03	53.4852	2.2841-02	0.5784	0.5210
CH4	8.8478-03	0.0	1.2379-03	4.5366-03	7.6067-02
C2H6	3.0173-03	0.0	3.6255-03	5.4032-03	4.7856-02
C3H8	1.9747-03	0.0	1.1617-02	8.0488-03	5.6966-02
HHC	6.0287-04	0.0	2.7136-02	8.0374-03	4.9105-02
H2S	8.4821-04	0.0	3.3440-03	2.7805-03	2.0623-02
CO2	4.0991-03	0.0	1.2356-03	2.7571-03	3.8966-02
H2	1.3001-04	0.0	1.7839-06	1.4157-05	8.1933-04
P-OIL	3.1342-04	0.0	0.8877	0.1451	0.8269
HNAPH	9.0064-05	0.0	0.2293	0.3009	0.1997
KERO	1.8361-05	0.0	0.2769	0.3661	0.2296
AGO	3.5552-06	0.0	0.5286	0.7115	0.3866
LVGO	1.4076-07	0.0	0.6580	0.9244	0.3255
HVGO	1.5237-09	0.0	0.9867	1.4696	0.1609
VR	4.8023-12	0.0	1.2004	1.8350	2.1368-02
TOTAL FLOW:					
KMOL/SEC	9.5306-04	2.9688	2.2752-02	5.1696-02	5.2782-02
KG/SEC	2.1025-02	53.4852	4.8390	6.3628	2.9622
CUM/SEC	3.9608-03	5.4515-02	5.4351-03	6.9524-03	1.6729-02
STATE VARIABLES:					
TEMP C	76.7766	37.7777	57.3291	84.0484	422.5409
PRES N/SQM	6.9000+05	6.8948+04	1.0340+06	1.3780+07	1.3780+07
VFRAC	1.0000	0.0	0.0	0.0	1.0000
LFRAC	0.0	1.0000	1.0000	1.0000	0.0
SFRAC	0.0	0.0	0.0	0.0	0.0
MASSVFRA	1.0000	0.0	0.0	0.0	1.0000
DENSITY:					
KMOL/CUM	0.2406	54.4598	4.1860	7.4357	3.1551
KG/CUM	5.3081	981.1088	890.3333	915.2105	177.0756
AVG MW	22.0601	18.0152	212.6903	123.0828	56.1222
SUBSTREAM: CISOLID	STRUCTURE: CONVENTIONAL				
COMPONENTS: KG/SEC					
COKE	0.0	0.0	0.2988	0.0	0.0

STREAM ID	SSU-W2	WCU-W	WSU-W2
TOTAL STREAM:			
KG/SEC	1.1246	1.1246	0.4630
PHASE:	LIQUID	LIQUID	LIQUID
COMPONENTS: KG/SEC			
H2O	1.1246	1.1246	0.4630
TOTAL FLOW:			
KMOL/SEC	6.2425-02	6.2425-02	2.5701-02
KG/SEC	1.1246	1.1246	0.4630
CUM/SEC	1.1938-03	1.1307-03	4.6327-04
STATE VARIABLES:			
TEMP C	76.7772	23.8888	18.8297
PRES N/SQM	6.9000+05	6.9000+05	2.0684+05
VFRAC	0.0	0.0	0.0
LFRAC	1.0000	1.0000	1.0000

SFRAC		0.0	0.0	0.0		
MASSVFRA		0.0	0.0	0.0		
DENSITY:						
KMOL/CUM		52.2933	55.2078	55.4775		
KG/CUM		942.0800	994.5850	999.4437		
AVG MW		18.0152	18.0152	18.0152		
STREAM ID		WCU-Q	HTR-Q	CONDH-Q	CONDL-Q	COIL-Q
FROM :		WCU	HTR	CONDH	CONDL	COIL
CLASS:		HEAT	HEAT	HEAT	HEAT	HEAT
Q	WATT	2.4832+05	-2.1709+06	5.6454+05	4.9995+05	1.2176+06
STREAM ID		IPMP-WK	PUMP-WK	CPMP-WK	FPMP-WK	
FROM :		IPMP	PUMP	CPMP	FPMP	
CLASS:		WORK	WORK	WORK	WORK	
P	WATT	8123.7674	1.6629+05	5470.4567	5241.9090	

Figure 8. (continued). Base case material and energy streams.

The base case describes a process treating 464 Mg/day (3,000 bbl/day) crude oil. The reactor pressure and temperature are 13.8 Mpa and 438°C and the liquid residence time two hours. These conditions lead to a computed increase of oil quality as measured by API gravity of 8.9° (13.5° for the feed and 22.4° for the product oil) and a loss of 16% to gas and coke.

Most of the energy need to heat the process streams is supplied by heat recovery operations. Examination of the results show that 2.17 MW of the energy required to heat the incoming stream is supplied by the fired heater, HTR. This represents 32% of the total needed. The rest of the energy is obtained through heat recovery operations, 41% is supplied by the HTXG heat exchanger, 16% by the HTXO heat exchanger and 11% in the preheat operation (PRH unit). Approximately 18% of the vapor stream from the reactor is diverted to the PRH unit. This represents 8% of the total mass entering the reactor unit.

The produced fuel gas flow, FUEL-G, is 0.26 kg/s (9.9 mol/s). The estimated heat of combustion of this gas is 296 kJ/mol. This stream could theoretically supply 2.93 MW of heat energy. If this stream were used to supply gas for a fired heater operating at a 75% efficiency, 2.2 MW of process heat would be available. As it turns out this is almost exactly the energy obtained from the fired heater, HTR.

Only a small portion of the produced oil product, P-OIL, exits in the fuel gas stream. Of the total 1.05 kg/s of P-OIL produced about 0.01 kg/s exits in the fuel gas.

A substantial fraction of the flow exists in the reactor in the vapor phase. Of the total 6.36 kg/s entering the reactor 47%, 2.96 kg/s, exits as vapor. The vapor phase is not all generated in the reactor. Vapor, on a weight basis, is produced in nearly equal amounts in heat recovery heat exchanger HTXO, the fired heater HTR, and the reactor vessel.

The stream enters the reactor at 438°C and exists at 423°C. The small drop is a result of the net endothermic nature of the assumed pyrolysis reaction. This is in part a result of the fact that the products tend to be vapor at the conditions of the reactor.

The weight fraction of water remaining in the produced oil stream is only 0.4%. Essentially all this water comes from the FPMP-O stream. The flashing of the hot product stream on the CPMP-O side reduces the water content in the CPMP-O stream to 0.04 wt%. A little over one-half the oil comes from the CPMP-O stream.

The basic premise of the process is that it is based on aqueous pyrolysis of the oil. This implies that water is present in the liquid phase during pyrolysis. The free water in the system goes away in the HTXO heat exchanger. However, some amount of water remains dissolved in the liquid phase. The amount estimated by the model is based on assumptions about the activity of water in the oil phase. The amount of water in the liquid phase entering the reactor is approximately 4 wt% and exiting in the liquid phase about 1.9 wt%. The reactor pressure 13.8 MPa (2000 psia) was chosen to maintain a computed water content of approximately 2 wt% in the reactor effluent.

Equipment Sizing and Economic Results

The basis of the process cost estimate revolves around the equipment costs. The list of equipment costed and selected sizing information are given in Table 3. Included in the table is a net cost factor which is used to compute the installed cost of the equipment from its purchase cost. ASPEN PLUS contains a set of default parameters for material and labor involved in piping, concrete, steel, electrical, instrumentation, insulation, paint, and setting work for each of its cost BLOCK models. These default values were used in all cases and result in the net cost factors shown.

The most expensive piece of equipment in the process is the reactor vessel. It represents over 40% of the total cost of equipment. The next most expensive item is the fired heater which accounts for approximately 19% of the total. Vendor "budget quotes" for these two items as well as the high pressure pump, were obtained as a check of the estimates used. Sizes vary somewhat from those of this base case. The quotes are for the following items:

- Reactor Vessel (RU) — Cost estimated by model correlations is \$537k for 47 m^3, 13.8 MPa (2000 psi) vessel; vendor budget estimate is \$700k for a 43 m^3 vessel, 13.8 MPa (vessel weight of 1.23×10^5 kg). The cost of a 43 m^3 vessel based on model correlations is \$460k (9/96 purchase), indicating the model may be underestimating the reactor vessel costs.

- Fired Heater (HT.) — Cost estimated by model correlations is $296k for a 2.9 MW gas-fired unit operating at 13.8 MPa (2000 psi); vendor budget estimate is $265k for a 5.3 MW unit. The cost of a 5.3 MW based on model correlations is $496k, indicating the model may be over estimating the fired heater costs.
- High Pressure Pump (PUMP) — Cost estimated by model correlations is $95k for a 0.44 m^3/min (200 gpm) unit; vendor budget estimate is $75k for a 0.5 m^3/min (133 gpm) unit. The cost of a 0.5 m^3/min pump based on model correlations is $99k, indicting the model may be over estimating

Table 3. Equipment costs for base case.

Unit	Type	Size	Purchase Cost (k$)	Composite Cost Factor	Total Cost (k$)
COIL	Air Cooler	34 m^2	10.8	1.52	16.4
CONDH	Heat Exchanger	28 m^2	33.6	2.24	75.2
CONDL	Heat Exchanger	10 m^2	6.9	2.12	14.6
CPMP	Pump	2.6 kg/s, 13 kW	3.1	1.94	6.0
CSU	Flash Vessel	1.1 m^3	3.6	2.22	8.0
FOIL	Flash Vessel	1.3 m^3	5.2	2.21	11.5
FPMP	Pump	2.2 kg/s, 13 kW	3.1	1.94	6.0
HTXG	Heat Exchanger	124 m^2	123.0	2.24	275.1
HTXO	Heat Exchanger	31 m^2	38.4	2.24	85.9
HTR	Fired Heater	2.9 MW	295.7	1.72	507.5
IPMP	Pump	7 kg/s, 17 kW	4.9	1.73	8.5
PRH	Vessel	4.6 m^3	10.6	2.98	31.6
PSU	Separator Tank	283 m^3	58.1	4.19	243.7
PUMP	Pump	6.4 kg/s, 166 kW	94.7	1.98	187.7
RU	Reactor Vessel	47 m^3	537.2	2.21	1189.2
SSU	Separator Vessel	4.3 m^3	9.3	2.22	20.6
WCU	Heat Exchanger	30 m^2	11.8	2.08	24.6
WSU	Separator Vessel	17 m^3	4.5	2.22	10.0
Total			1,255		2,722

In one case, the fired heater, the model estimates are substantially higher than the vendor quote and in another, the vessel, they are substantially lower. In the third case, the pump, the model estimates are slightly higher. The net difference between the model base case and the vendor quotes for these three units is approximately $138k (i.e., the model estimates are $138k lower than the vendor quotes). This is about 11% of the total equipment purchase costs. The bulk of the difference is in the cost of the pressure vessel. Further analysis would need to be done to determine if this difference holds up when other vendors and/or ancillary costs are considered. The model bases the ancillary costs on the purchase cost. For the case of the pressure vessel, it

probably could be argued that the cost factors should be lower since similar ancillary equipment would be needed for a lower pressure, hence, a less expensive vessel. The model estimates an installed cost for the vessel of $1190k. Using the vendor estimate of $700k for this vessel, the difference would leave over $400k for installation and ancillary equipment.

ASPEN PLUS uses the total equipment costs and a series of factors to compute the total plant capital cost. The resulting costs are summarized for the base case in Table 4. The cost of process units, $3,655k, shown in this Table is larger than the $2,722k shown in Table 3. This is a result of adding costs for the following: unlisted equipment, building materials and spares. The total direct and indirect capital cost (physical plant cost) comes to $6,221k.

Table 4. Total capital and investment costs for the base case in thousands of dollars.

Process Units	3,655
Contractor	1,537
Other	1029
Total Direct & Indirect	**6,221**
Contingency	933
Total Depreciable Capital	**7,154**
Working Capital	427
Startup Cost	531
Total Investment	**8,112**

Operating cost estimates are based on utility consumption computed in the process flow sheet model, the assumed number of operators per shift, and factors which estimate a variety of costs relative to original capital investment. These later items include maintenance, taxes, insurance, and general works charges. Operating costs are summarized in Table 5. The total operating cost for the base case is approximately $1.1M/yr., of which 26% is utility costs.

In addition to capital and operating costs, the model performs a profitability analysis which computes the required selling price of the product oil (stream PRD-O). For the base case the reported value is $17.81/Mg ($2.52/bbl). This result is for the case in which the incoming oil is assumed to have no value

Table 5. Operating costs for the base case in thousands of dollars per year (1st year dollars).

Electricity	88
Cooling Water	20
Natural Gas	189
Subtotal	**297**
Labor	241
Supplies	300
Other	286
Total	**1,124**

and, thus, the selling price is really a measure of the increase in selling price of the oil provided by the aqueous pyrolysis processing. Actually, the increase in price must be more than this increment because some oil is lost to coke and gas formation. This additional increment is dependent on the actual value of the feed oil. If the incoming oil has a value of $64.6/Mg ($10/bbl), then the product oil must have an increased value of $25.05/Mg ($3.55/bbl). This would represent a $2.81/Mg ($0.40/bbl) increment per unit increase in API gravity. It should be noted that this result excludes any costs for catalyst, coke removal, and waste water treatment.

SENSITIVITY

A series of runs have been made with the model to determine the sensitivity of the economics to various model parameters/assumptions. The results of these calculations are described below. All parameters are those of the base case except as noted. Of primary interest is the required increment in selling price of the product oil. In all cases described below it is assumed that the incoming crude oil has a value of $64.6/Mg ($10/bbl).

Flow Rate

Not surprisingly, the overall cost of processing declines with increased flow rates. Calculations for crude oil flow rates from 155-3,096 Mg/day (1,000-20,000 bbl/day) show a drop in required incremental oil selling price from $35.29/Mg to $20.36/Mg. For these cases only the flow rates change, the fraction of oil reacted and thus the increase in API gravity remains constant. The API gravity increases from 13.5° to 22.4° for all cases.

The change in required incremental selling price is shown in Fig. 9. The change in price is steepest at low flows and begins to level out at flows higher than the base case flow of 464 Mg/day (3,000 bbl/day). The figure also shows the required plant cost per unit oil processed and the gross operating costs.

The small increase in price and operating costs at intermediate flows was a result of the changes in the number of operators. One operator was assumed to be required for flows at and below 1,161 Mg/day (7,500 bbl/day). This was increased to two operators at flows up to and including 2,322 Mg/day (15,000 bbl/day) and three operators at 3,096 Mg/day (20,000 bbl/day).

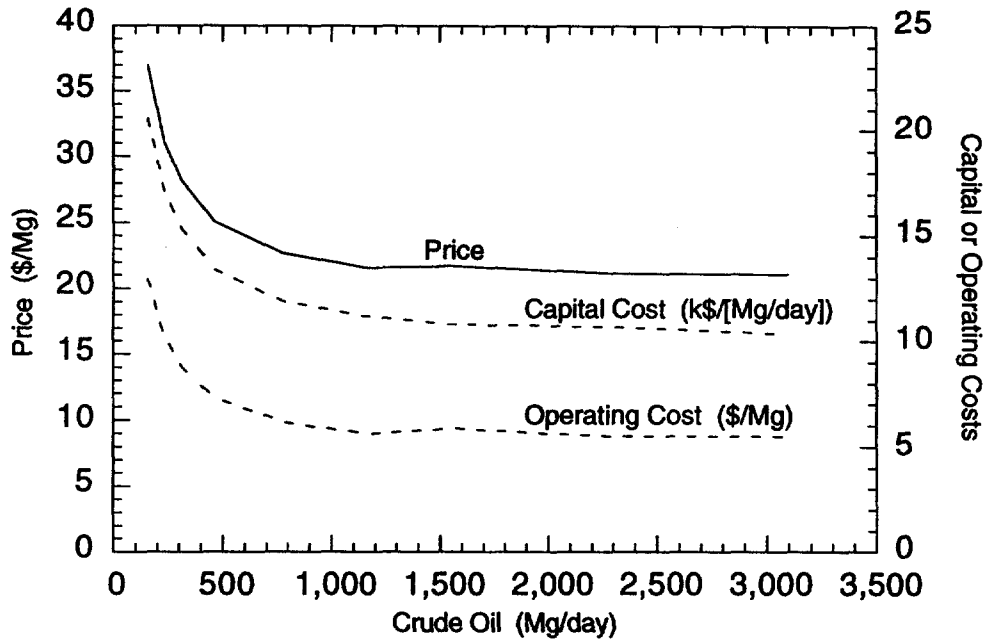


Figure 9. Required incremental selling price as a function of size of operation (Capital Costs shown are physical plant cost).

Residence Time

A series of runs were made in which residence time in the reactor was changed. Selected results are listed in Table 6. Reducing residence time directly reduces the size of the reactor. Since the reactor is the most costly piece of equipment, the cost of processing a unit of crude oil decreases with decreasing residence time. However, the reduced residence time leads to a reduced extent of reaction. Thus, the amount of upgrading, as measured by change in API gravity, goes down with decreasing residence time. The net effect is that the cost per unit oil per unit increase in API gravity goes down as residence time goes up (see Fig. 10).

Figure 10 indicates that a minimum in cost is not reached, even with 4 hours of residence time. However, at 4 hours residence time over 30% of the

incoming heavy oil has been converted to the product P-OIL and the change in price with increased residence time is small.

Residence Time (hrs)	Reactor Volume (m^3)	Product Oil Gravity (°API)	Crude to P-OIL (wt. %)	Product to Feed Ratio	Incremental Price [$64.6/Mg Feed] ($/Mg)
4	79.0	29.2	30.9	0.739	34.48
3	64.5	25.8	25.5	0.787	30.14
2	46.7	22.4	19.4	0.840	25.05
1	25.3	18.7	12.0	0.903	19.33
0.5	13.1	16.7	7.4	0.942	16.03
0.25	8.4	15.5	4.5	0.966	14.32

Table 6. Selected model results as a function of assumed reactor residence time.

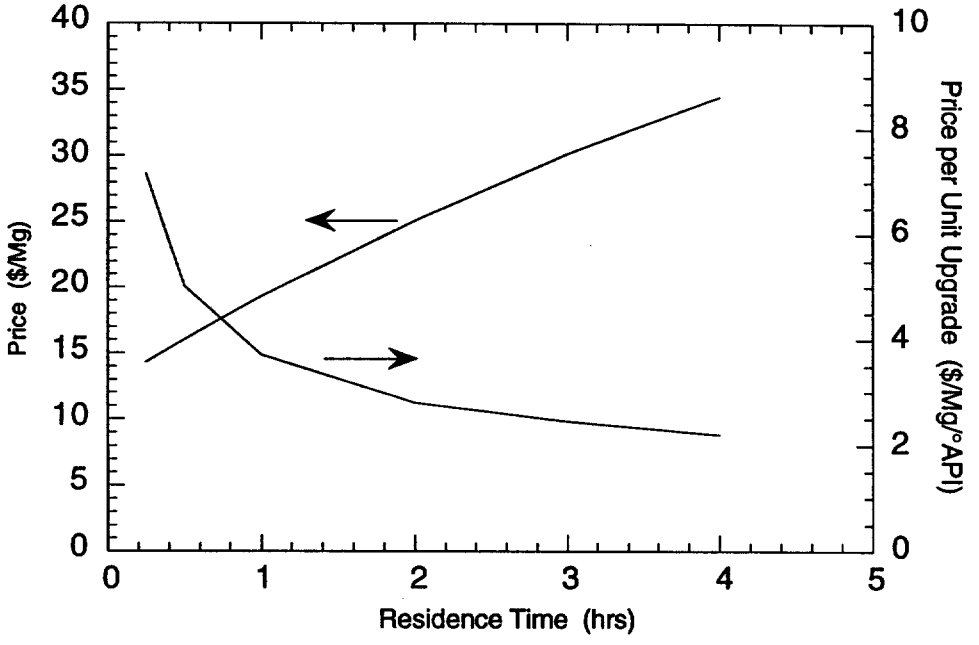


Figure 10. Required incremental selling price and incremental selling price per unit upgrade in API gravity as a function of reactor residence time (assumed input crude oil price is $64.6/Mg).

Pressure

Operating pressure has a fairly strong influence on process economics since the highest cost items, particularly the reactor vessel, are directly related to operating pressure. Also, pressure influences the operation and, therefore,

the process economics in a more complex way than does a change in residence time.

Some important changes in process performance resulting from alterations in operating pressure are shown in Table 7. With a halving of operating pressure, the reactor vessel cost drops by more than a factor of two. The drop is more than directly proportional to pressure because the required vessel size to maintain a liquid residence time of two hours also drops somewhat with pressure. The reason for this is the following. Even at the lower pressure the gas velocities are relatively low within the vessel, therefore, the entrainment does not play a large role in determining vessel volume. The volume is directly related to average liquid flow. For the lower pressures, more of the product oil enters the vapor phase which leads to a higher effective residence time for the heavier components. As a result the net pyrolysis increases somewhat with pressure as seen by the increase in API gravity and reduced product to feed ratio. (The product oil amount goes down with increased reaction because of the production of coke and gas.)

Table 7. Selected model results as a function of assumed reactor pressure.

Reactor Pressure (MPa)	Reactor Cost (k$)	Plant Cost (M$)	Product Oil Gravity (°API)	Product to Feed Ratio	Water in Liquid Phase Exiting Reactor (wt.%)	Incremental Price [$64.6/Mg Feed] ($/Mg)
13.78	537	6.22	22.4	0.84	1.9	25.05
12	506	6.11	22.5	0.84	1.6	24.87
11	434	5.83	25.2	0.80	1.2	26.57
10	351	5.17	27.4	0.77	1.0	27.00
8	274	4.84	29.0	0.75	0.7	27.44
6.89	211	4.48	29.4	0.74	0.6	26.83

The incremental price increase required, the last column in Table 7, is a complicated function of pressure because of competing effects. Lower pressure reduces capital and operating costs, but increased extent of reaction reduces the amount of product oil. The net effect is a relatively flat incremental price as a function of pressure.

However, as shown in Fig. 11, the price per unit increase in API gravity goes down with pressure. At the lower pressure 6.89 MPa (1000 psi) it is only 60% of the price at 13.78 MPa (2000 psi). The steep rise between 10 and 12 MPa is due to the discretized nature of the pricing used for certain items as a function of pressure, primarily the heat exchange units.

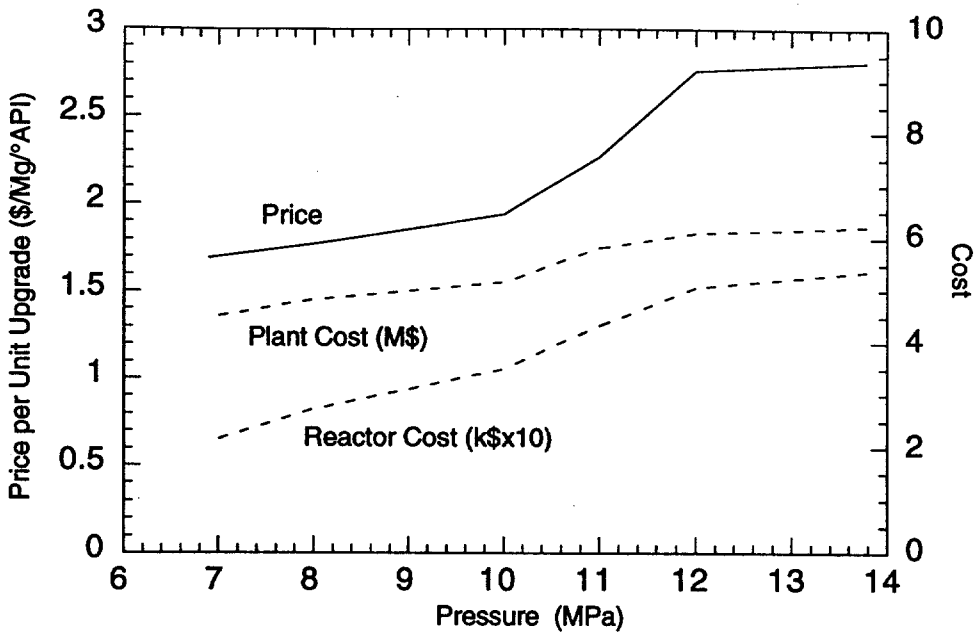


Figure 11. Required incremental selling price per unit upgrade in API gravity, reactor cost and total plant cost as a function of reactor pressure (assumed input crude oil price is $64.6/Mg).

As the pressure is lowered, the amount of water in the oil phase in the reactor decreases, see Table 7. Thus, if the water content of the oil phase in the reactor is an important parameter in the performance of the pyrolysis reactions, lower pressure operation may have an adverse effect not captured by the model.

Temperature

Like pressure, temperature has a relatively complicated influence on computed results. Computed results for four inlet temperatures are shown in Table 8. The temperature strongly affects the extent of reaction. This in turn influences the product oil gravity, the product-to-feed ratio, reactor volume, and the water content in liquid phase in the reactor.

Notice that the reactor volume steadily decreases with increasing temperature. This is a result of the increased production of P-OIL and a reduced liquid volume. However, the reactor cost does not monotonically decrease. This is a result of a correction made by ASPEN PLUS in computing vessel costs for changes in material strength. Algorithms in ASPEN PLUS begin to degrade the strength of carbon steel at temperatures above 350°C, and therefore increases required wall thickness, and thus vessel costs, for a given pressure of operation.

Table 8. Selected model results as a function of assumed reactor inlet temperature.

Temperature (°C)	Reactor Volume (m^3)	Reactor Cost (k$)	Product Oil Gravity (°API)	Product to Feed Ratio	Water in Liquid Phase Exiting Reactor (wt.%)	Incremental Price [$64.6/Mg Feed] ($/Mg)
425	53.6	520	19.5	0.89	2.4	21.80
437.8	46.7	537	22.4	0.84	1.9	25.05
445	41.5	490	24.7	0.81	1.6	26.78
455	32.2	574	28.6	0.75	1.2	31.34

The model indicates that the price per unit increase in API gravity decreases as temperature increases, see Fig. 12. This is a result of the increased extent of reaction, which increases the product oil gravity. As with pressure, operation at the lower cost, in this case higher temperature, results in lower water content in the liquid phase in the reactor which may result in adverse effects not captured by the model.

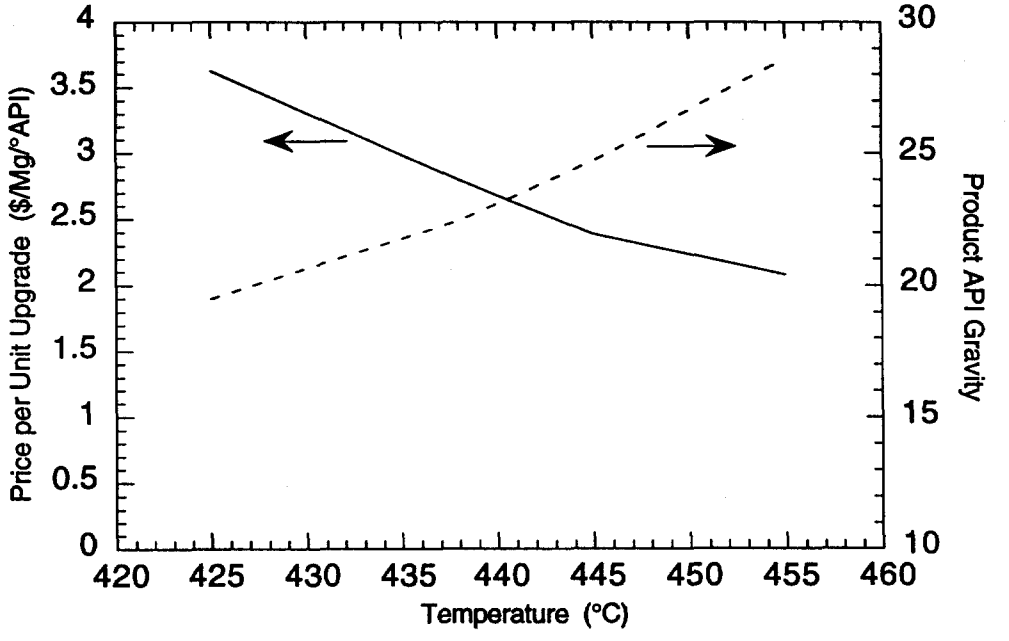


Figure 12. Required incremental selling price per unit upgrade in API gravity and product API gravity cost as a function of reactor inlet temperature (assumed input crude oil price is $64.6/Mg).

Constant Gravity

The primary controllable parameters in the process are reactor pressure, temperature, and residence time. These parameters can be manipulated so

that the same net product oil is produced using different combinations of parameters. A series of runs needed to be done to determine the relative economics of producing an oil with an API gravity equal to the base case, approximately 22.4. An additional constraint was placed on the system which required the water content of the liquid exiting the reactor to be equal to the base case, approximately 1.85 wt%.

Results of the calculations are shown in Table 9. These results cover a temperature range from 400-450°C which resulted in required residence times from 1-10 hours. At 425°C, and above, the economic performance is essentially unchanged. At the two lower temperatures the costs are somewhat higher. It appears that for a given reaction extent and specified water content the economics are relatively insensitive to the choice of pressure/temperature operating point.

Table 9. Selected model results for cases in which product oil API gravity and water content in the exiting reactor liquid were maintained approximately equal to the base case, 22.4 and 1.85 wt% respectively.

Temperature (°C)	Pressure (MPa)	Residence Time (hrs)	Reactor Volume (m^3)	Incremental Price [$64.6/Mg Feed] ($/Mg)	Incremental Price per Unit API [$64.6/Mg Feed] ($/Mg/°API)
400	10.0	10	170	30.31	3.52
415	11.1	5	135	28.44	3.31
425	12.0	3.5	90	26.08	3.03
437.8	13.78	2	47	25.05	2.81
450	17.0	1	21	25.50	2.97

Aqueous Pyrolysis Reaction

A series of runs were done to look at the influence of changing the kinetic and stoichiometric parameters of the assumed pyrolysis reaction. In one set of runs it is assumed that the reaction rate is 10 times faster than the base case. This was done by increasing the pre-exponential factor in the kinetic rate constant expression by a factor of 10. In the second set of runs it is assumed that no coke is produced during the reaction and the amounts of product oil increases to compensate. For all runs the product oil gravity and amount of water in the liquid exiting the reactor was held constant at values equal to the base case results.

For each assumption two sets of computed results are shown in Table 10, one in which the residence time was held equal to the base case, two hours, and a second in which the temperature was held equal to the base case. For the

constant residence time runs, the system pressure and temperature were adjusted to maintain the desired operation; and for the base case temperature runs, the residence time was adjusted for the faster kinetics case and the residence time and pressure were adjusted for the no coke case to give the required process performance.

Table 10. Selected model results for runs in which product oil API gravity and water content in the exiting reactor liquid were maintained approximately equal to the base case, 22.4 and 1.85 wt% respectively.

Case	Temperature (°C)	Pressure (MPa)	Residence Time (hrs)	Reactor Volume (m^3)	Reactor Cost (k$)	Operating Cost (M$/yr)	Incremental Price [$64.6/Mg Feed] ($/Mg)	Incremental Price per Unit API [$64.6/Mg Feed] ($/Mg/°API)
Base	437.8	13.8	2.00	47	537	1.12	25.05	2.81
10x rate	372.0	8.9	2.00	61	289	0.92	21.54	2.45
10x rate	437.8	13.8	0.20	8	190	0.96	21.50	2.42
No coke	437.8	18.0	0.85	20	418	1.16	20.69	2.32
No coke	423.0	13.0	2.00	51	470	1.05	18.77	2.11

The results in Table 10 indicate modest improvement in economics over the base for these more optimistic assumptions about the pyrolysis reaction. For the cases using the faster kinetics about 15% lowering in price is computed. This was true whether the faster rate was compensated for by lowering the temperature and pressure or by reducing the residence time in the reactor. For the cases in which zero coke production was assumed, it was more advantages to reduce the temperature to compensate than to reduce the residence time. The best of the no coke runs indicates a 25% improvement in process economics over the base case.

Burning Fuel Gas

As mentioned previously, the amount of combustion energy present in the produced fuel gas for the base case is essentially equal to the amount of energy needed to fuel the fired heater. In the base case the fired heater fuel is assumed to be natural gas. The fired heater is computed to require 2.89 MW of combustible gas energy. At an indexed price of $2.29/GJ (2nd quarter 1995) the yearly cost for natural gas is $189k. This is 17% of the annual operating cost. If it is assumed that the produced fuel gas can be used instead of natural gas to fire the heater, then the total operating cost drops and, consequently, the required incremental price of the product oil would be reduced.

A case was run in which no charge was taken for fueling the fired heater, HTR. This resulted in a drop of required incremental oil price from

$25.05/Mg for the base case to $23.25/Mg, just under a 10% drop. These costs are for feed oil assumed to be worth $64.6/Mg ($10/bbl).

Stainless Steel Reactor

The base case assumes that the reactor vessel can be made from carbon steel with a corrosion allowance of 0.32 cm (0.125 in.). If for a given system the corrosion of the carbon steel is unacceptable, then an alloy vessel would be required. Other key elements of the system were already assumed to be stainless steel in the base case (e.g., heat exchanger tubes and high pressure pump).

The switch to stainless steel for the reactor vessel increases the estimated vessel purchase cost from $537k to $983k. This results in an increase in the plant capital cost of about 20% and an increase in required selling price from the base case value of $25.05/Mg to a value of $27.93/Mg (slightly more than a 10% increase).

Minimum Cost

The base case assumptions leads to a required oil price increase of $2.81/Mg·°API. This is higher than what has been estimated as a reasonable expectation for price increase of approximately $1.60/Mg·°API ($0.25/bbl·°API)[9]. This level was for the most part not reached by the cases considered in the sensitivity analysis describe above.

It is of interest to determine whether such price levels could be reached for a process in which several factors are shifted to more optimistic values. A model run was done in which the base case parameters were maintained except for the following:

- Faster reaction kinetics were assumed; the pre-exponential kinetic factor was increased ten fold.
- The coke make was set to zero and the product oil yield increased to compensate.
- The fired heater was assumed to be fed with produced fuel gas.
- Reactor residence times were chosen so that an increase in API gravity to about 25° was achieved.

Selected results from the calculations are listed in Table 11. Two cases were run, each using a different assumed temperature at the reactor inlet. The pressures were adjusted in each case to maintain a water content in the exit reactor liquid of approximately 2 wt%. Differences in results for the two cases were minor. Both show a required price increase below the $1.60/Mg·°API level. In fact the $1.60/Mg·°API can be achieved even assuming that that

instead of no coke production, coke production is cut in half from the levels of the base case.

Table 11. Selected model results for the base case and cases in which more optimistic assumptions are made.

	Temperature (C)	Pressure (MPa)	Residence Time (hrs)	Reactor Volume (m^3)	Product Oil Gravity (°API)	Reactor Cost (% of Listed Equipment)	Capital Costs (M$)	Operating Cost (M$/yr)	Incremental Price per Unit API [$64.6/Mg Feed] ($/Mg/°API)
Base	437.8	13.78	2	46.7	22.4	44	6.22	1.12	2.81
	360	10	2.4	72.1	25.2	38	4.68	0.78	1.39
	375	11	1.1	31.7	25.0	25	4.31	0.75	1.34

CONCLUSIONS

An ASPEN PLUS model of in-field aqueous pyrolysis upgrading of heavy oils has been developed. The model indicates that for a 464 Mg/day (3,00 bbl/day) process, which increases the oil API gravity of the processed oil from 13.5° to 22.4°, the required value increase of the oil must be about $2.80/Mg·°API (for a crude oil initially worth $64.6/Mg). This level of upgrading has been demonstrated in preliminary experiments with candidate catalysts[3].

As expected, the reactor vessel is the most costly piece of equipment in the process, accounting for approximately 40% of the equipment costs for the base case. If a stainless steel reactor vessel is required, this percentage increases to about 55%. This increased cost raises the required selling price of the oil about 10%.

A vendor "budget" quote for the reactor vessel was obtained which was substantially higher than the estimate used in the model. However, the manner in which installed costs are estimated in the model probably overestimates ancillary costs for the vessel. Further work on the installed reactor vessel costs is required to resolve this issue.

The economic performance of the process is a function of a number of factors, including operating pressure, temperature, reaction kinetics, reaction stoichiometry and total flow rate. Substantial changes in reaction stoichiometry and kinetics are required to improve the economic performance.

It is estimated that a required selling price increase for the oil of $1.34/Mg·°API ($0.21/bbl ·°API) can be reached if a catalyst can be found which both yields one-half the coke make and an order of magnitude more rapid kinetics than that already demonstrated.

REFERENCES

1. J. G. Reynolds, A. M. Murray, E. V. Nuxoll, and G. A. Fox, "Upgrading of Heavy Oil from the San Joaquin Valley of California by Aqueous Pyrolysis," Lawrence Livermore National Laboratory, Livermore, California, UCRL-122472 (1995).

2. *ASPEN PLUS Reference Manual — User Guide Volume 1,* (Aspen Technology, Inc., Cambridge, Massachusetts, 1994) Chapter 9.

3. C. B. Thorsness and J. G. Reynolds "Modeling a Set of Heavy Oil Aqueous Pyrolysis Experiments," Lawrence Livermore National Laboratory, Livermore, California, UCRL-ID-125673 (1996).

4. *ASPEN PLUS Reference Manual — Input Language Guide - Volume 1,* (Aspen Technology, Inc., Cambridge, Massachusetts, 1996) Chapter 3.

5. *ASPEN PLUS Reference Manual — Volume 4 — Costing and Pressure Relief,* (Aspen Technology, Inc., Cambridge, Massachusetts, 1994) Chapters 1-7, 9, 10, 12-15.

6. Stanley M. Walas, *Chemical Process Equipment — Selection and Design,* (Butterworth-Heinemann, Boston 1990), pp. 663-669.

7. R. N. Watkins, "Sizing Separators and Accumulators," *Hydrocarbon Processing, Vol. 46,* pp. 253-256 (1967).

8. *ASPEN PLUS Reference Manual — Volume 5 — System Management* (Aspen Technology, Inc., Cambridge, Massachusetts, 1994) Chapter 1 pp. 3-5.

9. J. G. Reynolds, Personal Communications.

APPENDIX I

ASPEN PLUS INPUT FILE

Below is a listing of the ASPEN PLUS input language file for the base case. This input files has run successfully with ASPEN PLUS Release 9.2-1 on a HP-9000/730 computer workstation running HPUX 10.10. Just under three minutes are required for execution on a dedicated machine. In order to execute successfully, the user FORTRAN routines called by the model must be available as object files at the time of execution. These files can be generated from the source listings given in Appendices II and III using the "aspcomp" command provided as part of the ASPEN PLUS software.

This base case completes execution with two warnings. These are listed at the end of the output file. The first warning concerns the unit operation block RSU. The RSU routine operates on the free-water exiting the reactor to establish a user requested split of free-water. Since no free-water is present in this case, and for nearly all cases, the ASPEN PLUS module, FSPLIT, which implements the RSU unit reports a zero feed stream warning.

The other warning concerns costing of the two pumps, FPMP and CPMP, which pump the product oil to the requested outlet pressure. For the base case, the oil flows are below the lower cutoff limit of the ASPEN PLUS cost correlations for the SS-ANSI pump type. As a result, ASPEN PLUS uses the cost of the smallest pump in its database. For the base case the volumetric flow through these pumps, about 0.003 m^3/s, is just below the low flow cutoff, 0.00316 m^3/s (50 gpm). Consequently the cost estimates are satisfactory. In addition, the cost of these pumps contribute less than 1% of cost of the process for the base case.

Input file:

```
TITLE 'Hydrous Pyrolysis Model (HPM) [Rev 5.7]'

;=============================================================================
; Normal runs complete with errors & warning errors.  However, they
; should be checked for new situations.
;=============================================================================

;=============================================================================
;  Revisions
;=============================================================================
;  Rev    When      What
;  5.7  17-oct-96 Documented version

;=============================================================================
;  Overall stream description
;=============================================================================
;
```

```
;     Input Streams:
;       FEED - Oil & water input

;     Output Streams:
;       Material
;       PSU-W2   - Primary water decant stream from 1st sep.
;       WCU-W    - Water decant stream from second sep.
;       WSU-W2   - Final  water decant stream
;       PRD-O    - Product oil stream
;       FUEL-G   - Fuel gas

;       Thermal energy
;       HTR-Q    - Heat added by trim heater
;       COIL-Q   - Active cooling of product oil stream
;       IHTR-Q   - Heat delivered to heater upstream of secondary water
;                  spearation (usually zero)
;       WCU-Q    - Cooling of water from the secondary separator
;       CONDL-Q  - Heat removed in low pressure condeneser
;       CONDH-Q  - Heat removed in high pressure condeneser
;
;       Mechanical energy
;       PUMP-WK  - Primary pump
;       FPMP-WK  - Product oil pump
;       CPMP-WK  - Product oil pump


;=============================================================================
;  Other system parameters not include in input section
;=============================================================================

;     Data (found using 'Data #n' search):
;       n   Description ..............................
;   Priamry process parameters
;       0 - Feed temperature, pressure & flow rates
;       1 - Oil properties
;       2 - Water solubility parameters
;
;   Secondary process  parameters
;      21 - Additional kinetic parameters
;      22 - Heat recovery overall heat trans. coefs
;      23 - Parameters associated with reactor gas/liq separation
;      24 - Outlet temperature of CONDL
;      25 - Outlet temperature of CONDH
;      26 - Pressure of CSU unit
;      27 - P-OIL properties
;      28 - Define HHC & COKE components
;
;   Economic parameters
;      101 - Residence time in vessels
;      102 - Cost of electricity, natural gas, & cooling water
;      103 - Labor rates
;      104 - Corrosion
;      105 - Parameters associated with FOIL gas/liq separation
;      106 - Parameters associated with CSU  gas/liq separation
;      107 - Cooling water temperatures
;      108 - Heater efficiency
;      109 - Parameters associated with PSU preheat vessel
;      110 - Pump types
;      111 - Vessel internals
;


;SIMULATE STOP=PRD
```

```
;=====================================================================
;  Input Model Parameters
;=====================================================================

FORTRAN FINPUT
F      common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
F      common /usr2/ akin, xch4,xc2h6,xc3h8,xhhc,xco2,xh2,xh2s,wtfc,wtfo
F      common /usr3/ capfac
     DEFINE tr    BLOCK-VAR  BLOCK=HTR  SENTENCE=PARAM  VARIABLE=TEMP
     DEFINE pr    BLOCK-VAR  BLOCK=PUMP SENTENCE=PARAM  VARIABLE=PRES
     DEFINE pri   BLOCK-VAR  BLOCK=IPMP SENTENCE=PARAM  VARIABLE=PRES
     DEFINE prc   BLOCK-VAR  BLOCK=CPMP SENTENCE=PARAM  VARIABLE=PRES
     DEFINE prf   BLOCK-VAR  BLOCK=FPMP SENTENCE=PARAM  VARIABLE=PRES
     DEFINE tao   BLOCK-VAR  BLOCK=HTXO SENTENCE=PARAM  VARIABLE=DELT-HOT
     DEFINE tag   BLOCK-VAR  BLOCK=HTXG SENTENCE=PARAM  VARIABLE=DELT-HOT
     DEFINE fgr   BLOCK-VAR  BLOCK=GASR SENTENCE=FRAC   VARIABLE=FRAC
;
;  Reactor pressure (Pa) & inlet temperature (C)
F      pr=13.78e6
F      tr=437.8

;  Initial pressure (Pa) for vapor heat recovery
F      pri=0.69e6

;  Set temperature (C) for feed stream to secondary water speratore,SSU.
;     Either uses hot vapor to heat or IHTR heater depending on coding.
;     (negative for no heat)
F      tihtr=76.7

;  Reactor residence time in minutes
F      rtres=2*60
;
;  Water to oil ratio at separators ; wor1 - at PSU.
;                                     wor2 - at SSU
;                                     wor3 - at RSU
;                                     wor4 - at WSU
;  Total mass basis, includes dissolved water
F      wor1=0.30
F      wor2=0.10
F      wor3=0.01
F      wor4=0.01
;
;  Desired final water oil ratio, only active in DESIGN SPEC SPEC2 is active
F      worf=0.009

;  Desired final oil delivery pressure (Pa)
F      prc=1.034e6
;  The next statement is not input and should not be altered.  It sets
;     the ouput of the two final pumps to the same pressure.
F      prf=prc

;  Split of gas from reactor, fraction recycled to PRH.  Initial guess only.
;     Sets fraction absolutely only if DESIGN SPEC1 is removed.
F      fgr=0.0

;  Approach temperatures (C) of oil & vapor heat recovery htx's
;  (39 is about optimum for heat recovery, however need hot enough
;   to establish final water/oil ratio in final flash, if significant
;   gas heating available then optimize heat recovery)
;  tao - oil heat recovery;  tag - gas heat recovery
F      tao=33.3
F      tag=33.3
;
; Basic reaction stoich.
```

```
;    wtfc-weight fraction coke
;    wtfo-weight fraction oil component
;    (weight fraction vapor/gas by difference)
;    xch4, xc2h6, ... relative moles of vapor/gas components.  The amounts
;          are relative to toal vapor/gas product, not reacted oil.
F     wtfc=0.2
F     wtfo=0.6
;
F     xch4=1.0
F     xc2h6=0.34
F     xc3h8=0.26
F     xco2=0.19
F     xh2=0.09
F     xhhc=0.17
F     xh2s=0.12

;  Input pre-exponential factor for kinetics (1/s)
F     akin=1.7e8

;  capacity factor for all non-ASPEN cost modules (ASPEN modules default
;       to 1.06)
F     capfac=1.06

    EXECUTE FIRST


FORTRAN SETR   ; set kinetic and stoichiometric constants
F     common /plex/plx(1)
F     common /usr2/ akin,xch4,xc2h6,xc3h8,xhhc,xco2,xh2,xh2s,wtfc,wtfo
F     dimension iplx(1)
F     equivalence (iplx(1),plx(1))
;
   DEFINE ac1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=CISOLID ID3=COKE
   DEFINE ach41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=CH4
   DEFINE ac2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=C2H6
   DEFINE ac3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=C3H8
   DEFINE aco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=CO2
   DEFINE ah21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=H2
   DEFINE ahhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=hhc
   DEFINE ah2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=H2S
   DEFINE ao1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
        ID1=1  ID2=MIXED  ID3=P-OIL
;
   DEFINE bc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=CISOLID ID3=COKE
   DEFINE bch41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=MIXED  ID3=CH4
   DEFINE bc2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=MIXED  ID3=C2H6
   DEFINE bc3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=MIXED  ID3=C3H8
   DEFINE bco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=MIXED  ID3=CO2
   DEFINE bh21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
        ID1=2  ID2=MIXED  ID3=H2
   DEFINE bhhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
```

```
    ID1=2  ID2=MIXED  ID3=hhc
DEFINE bh2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=H2S
DEFINE bo1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=P-OIL
;
DEFINE cc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=CISOLID ID3=COKE
DEFINE cch41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=CH4
DEFINE cc2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=C2H6
DEFINE cc3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=C3H8
DEFINE cco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=CO2
DEFINE ch21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=H2
DEFINE chhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=hhc
DEFINE ch2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=H2S
DEFINE co1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=P-OIL
;
DEFINE dc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=CISOLID ID3=COKE
DEFINE dch41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=CH4
DEFINE dc2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C2H6
DEFINE dc3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C3H8
DEFINE dco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=CO2
DEFINE dh21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2
DEFINE dhhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=hhc
DEFINE dh2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2S
DEFINE do1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=P-OIL
;
DEFINE ec1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=CISOLID ID3=COKE
DEFINE ech41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CH4
DEFINE ec2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C2H6
DEFINE ec3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C3H8
DEFINE eco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CO2
DEFINE eh21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2
DEFINE ehhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=hhc
DEFINE eh2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2S
DEFINE eo1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=P-OIL
;
DEFINE fc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
```

```
    ID1=6  ID2=CISOLID ID3=COKE
DEFINE fch41 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CH4
DEFINE fc2h61 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C2H6
DEFINE fc3h81 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C3H8
DEFINE fco21 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CO2
DEFINE fh21  BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=H2
DEFINE fhhc1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=hhc
DEFINE fh2s1 BLOCK-VAR BLOCK=RU1 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=H2S
DEFINE fo1  BLOCK-VAR BLOCK=RU1  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=P-OIL
;
DEFINE ac2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=CISOLID ID3=COKE
DEFINE ach42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=CH4
DEFINE ac2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=C2H6
DEFINE ac3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=C3H8
DEFINE aco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=CO2
DEFINE ah22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=H2
DEFINE ahhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=hhc
DEFINE ah2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=H2S
DEFINE ao2  BLOCK-VAR BLOCK=RU2  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=1  ID2=MIXED  ID3=P-OIL
;
DEFINE bc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=CISOLID ID3=COKE
DEFINE bch42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=CH4
DEFINE bc2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=C2H6
DEFINE bc3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=C3H8
DEFINE bco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=CO2
DEFINE bh22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=H2
DEFINE bhhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=hhc
DEFINE bh2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=H2S
DEFINE bo2  BLOCK-VAR BLOCK=RU2  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=2  ID2=MIXED  ID3=P-OIL
;
DEFINE cc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=CISOLID ID3=COKE
DEFINE cch42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=CH4
DEFINE cc2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=C2H6
DEFINE cc3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=C3H8
```

```
DEFINE cco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=CO2
DEFINE ch22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=H2
DEFINE chhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=hhc
DEFINE ch2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=3  ID2=MIXED  ID3=H2S
DEFINE co2   BLOCK-VAR BLOCK=RU2  SENTENCE=STOIC VARIABLE=COEF  &
    ID1=3  ID2=MIXED  ID3=P-OIL
;
DEFINE dc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=CISOLID ID3=COKE
DEFINE dch42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=CH4
DEFINE dc2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C2H6
DEFINE dc3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C3H8
DEFINE dco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=CO2
DEFINE dh22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2
DEFINE dhhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=hhc
DEFINE dh2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2S
DEFINE do2   BLOCK-VAR BLOCK=RU2  SENTENCE=STOIC VARIABLE=COEF  &
    ID1=4  ID2=MIXED  ID3=P-OIL
;
DEFINE ec2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=CISOLID ID3=COKE
DEFINE ech42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CH4
DEFINE ec2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C2H6
DEFINE ec3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C3H8
DEFINE eco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CO2
DEFINE eh22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2
DEFINE ehhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=hhc
DEFINE eh2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2S
DEFINE eo2   BLOCK-VAR BLOCK=RU2  SENTENCE=STOIC VARIABLE=COEF  &
    ID1=5  ID2=MIXED  ID3=P-OIL
;
DEFINE fc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=CISOLID ID3=COKE
DEFINE fch42 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CH4
DEFINE fc2h62 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C2H6
DEFINE fc3h82 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C3H8
DEFINE fco22 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CO2
DEFINE fh22  BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=H2
DEFINE fhhc2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=hhc
DEFINE fh2s2 BLOCK-VAR BLOCK=RU2 SENTENCE=STOIC VARIABLE=COEF   &
```

```
      ID1=6   ID2=MIXED   ID3=H2S
   DEFINE fo2   BLOCK-VAR BLOCK=RU2   SENTENCE=STOIC VARIABLE=COEF    &
      ID1=6   ID2=MIXED   ID3=P-OIL
;
   DEFINE ac3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=CISOLID ID3=COKE
   DEFINE ach43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=CH4
   DEFINE ac2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=C2H6
   DEFINE ac3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=C3H8
   DEFINE aco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=CO2
   DEFINE ah23   BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=H2
   DEFINE ahhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=hhc
   DEFINE ah2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=H2S
   DEFINE ao3   BLOCK-VAR BLOCK=RU3   SENTENCE=STOIC VARIABLE=COEF    &
      ID1=1   ID2=MIXED   ID3=P-OIL
;
   DEFINE bc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=CISOLID ID3=COKE
   DEFINE bch43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=CH4
   DEFINE bc2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=C2H6
   DEFINE bc3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=C3H8
   DEFINE bco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=CO2
   DEFINE bh23   BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=H2
   DEFINE bhhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=hhc
   DEFINE bh2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=H2S
   DEFINE bo3   BLOCK-VAR BLOCK=RU3   SENTENCE=STOIC VARIABLE=COEF    &
      ID1=2   ID2=MIXED   ID3=P-OIL
;
   DEFINE cc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=CISOLID ID3=COKE
   DEFINE cch43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=CH4
   DEFINE cc2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=C2H6
   DEFINE cc3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=C3H8
   DEFINE cco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=CO2
   DEFINE ch23   BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=H2
   DEFINE chhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=hhc
   DEFINE ch2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=H2S
   DEFINE co3   BLOCK-VAR BLOCK=RU3   SENTENCE=STOIC VARIABLE=COEF    &
      ID1=3   ID2=MIXED   ID3=P-OIL
;
   DEFINE dc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
      ID1=4   ID2=CISOLID ID3=COKE
   DEFINE dch43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF    &
```

```
    ID1=4  ID2=MIXED  ID3=CH4
DEFINE dc2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C2H6
DEFINE dc3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=C3H8
DEFINE dco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=CO2
DEFINE dh23  BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2
DEFINE dhhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=hhc
DEFINE dh2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=H2S
DEFINE do3  BLOCK-VAR BLOCK=RU3  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=4  ID2=MIXED  ID3=P-OIL
;
DEFINE ec3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=CISOLID ID3=COKE
DEFINE ech43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CH4
DEFINE ec2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C2H6
DEFINE ec3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=C3H8
DEFINE eco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=CO2
DEFINE eh23  BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2
DEFINE ehhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=hhc
DEFINE eh2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=H2S
DEFINE eo3  BLOCK-VAR BLOCK=RU3  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=5  ID2=MIXED  ID3=P-OIL
;
DEFINE fc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=CISOLID ID3=COKE
DEFINE fch43 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CH4
DEFINE fc2h63 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C2H6
DEFINE fc3h83 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=C3H8
DEFINE fco23 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=CO2
DEFINE fh23  BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=H2
DEFINE fhhc3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=hhc
DEFINE fh2s3 BLOCK-VAR BLOCK=RU3 SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=H2S
DEFINE fo3  BLOCK-VAR BLOCK=RU3  SENTENCE=STOIC VARIABLE=COEF   &
    ID1=6  ID2=MIXED  ID3=P-OIL

DEFINE aea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
    ID1=1
DEFINE bea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
    ID1=2
DEFINE cea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
    ID1=3
DEFINE dea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
    ID1=4
DEFINE eea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
    ID1=5
```

```
    DEFINE fea1 BLOCK-VAR BLOCK=RU1  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=6
    DEFINE aea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=1
    DEFINE bea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=2
    DEFINE cea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=3
    DEFINE dea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=4
    DEFINE eea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=5
    DEFINE fea2 BLOCK-VAR BLOCK=RU2  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=6
    DEFINE aea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=1
    DEFINE bea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=2
    DEFINE cea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=3
    DEFINE dea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=4
    DEFINE eea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=5
    DEFINE fea3 BLOCK-VAR BLOCK=RU3  SENTENCE=RATE-CON VARIABLE=PRE-EXP &
        ID1=6
;
;  define gas mw's for later use
F     ioff=ifcmnc('MW')
F     n=kccidc('CH4')
F     wch4=plx(ioff+n)
F     n=kccidc('C2H6')
F     wc2h6=plx(ioff+n)
F     n=kccidc('C3H8')
F     wc3h8=plx(ioff+n)
F     n=kccidc('CO2')
F     wco2 =plx(ioff+n)
F     n=kccidc('H2')
F     wh2  =plx(ioff+n)
F     n=kccidc('H2S')
F     wh2s =plx(ioff+n)
;  COKE
F     n=kccidc('COKE')
F     wc   =plx(ioff+n)
;  P-OIL
F     n=kccidc('P-OIL')
F     wo   =plx(ioff+n)
;  HHC
F     n=kccidc('HHC')
F     whhc =plx(ioff+n)
;
;  normalize gas
F     sum=xch4+xc2h6+xc3h8+xco2+xh2+xhhc+xh2s
F     xch4=xch4/sum
F     xc2h6=xc2h6/sum
F     xc3h8=xc3h8/sum
F     xco2=xco2/sum
F     xh2=xh2/sum
F     xhhc=xhhc/sum
F     xh2s=xh2s/sum
;
;  compute gas average mw
F     wmix=xch4*wch4+xc2h6*wc2h6+xc3h8*wc3h8+xco2*wco2+xh2*wh2
F    &      +xhhc*whhc+xh2s*wh2s
```

```
;
;  For VR reaction
F     n=kccidc('VR')
F     wm    =plx(ioff+n)
F     wtc=wm*wtfc
F     wto=wm*wtfo
F     wtg=wm-wtc-wto
F     xmolg=wtg/wmix
F     ac1=wtc/wc
F     ao1=wto/wo
F     ach41 =xmolg*xch4
F     ac2h61=xmolg*xc2h6
F     ac3h81=xmolg*xc3h8
F     aco21 =xmolg*xco2
F     ah21  =xmolg*xh2
F     ahhc1 =xmolg*xhhc
F     ah2s1 =xmolg*xh2s

F     aea1=akin
F     aea2=akin
F     aea3=akin

;  For HVGO reaction
F     n=kccidc('HVGO')
F     wm    =plx(ioff+n)
F     wtc=wm*wtfc
F     wto=wm*wtfo
F     wtg=wm-wtc-wto
F     xmolg=wtg/wmix
F     bc1=wtc/wc
F     bo1=wto/wo
F     bch41 =xmolg*xch4
F     bc2h61=xmolg*xc2h6
F     bc3h81=xmolg*xc3h8
F     bco21 =xmolg*xco2
F     bh21  =xmolg*xh2
F     bhhc1 =xmolg*xhhc
F     bh2s1 =xmolg*xh2s

F     bea1=akin
F     bea2=akin
F     bea3=akin
;
;  For LVGO reaction
F     n=kccidc('LVGO')
F     wm    =plx(ioff+n)
F     wtc=wm*wtfc
F     wto=wm*wtfo
F     wtg=wm-wtc-wto
F     xmolg=wtg/wmix
F     cc1=wtc/wc
F     co1=wto/wo
F     cch41 =xmolg*xch4
F     cc2h61=xmolg*xc2h6
F     cc3h81=xmolg*xc3h8
F     cco21 =xmolg*xco2
F     ch21  =xmolg*xh2
F     chhc1 =xmolg*xhhc
F     ch2s1 =xmolg*xh2s

F     cea1=akin
F     cea2=akin
F     cea3=akin
;
```

```
;  For AGO reaction
F      n=kccidc('AGO')
F      wm    =plx(ioff+n)
F      wtc=wm*wtfc
F      wto=wm*wtfo
F      wtg=wm-wtc-wto
F      xmolg=wtg/wmix
F      dc1=wtc/wc
F      do1=wto/wo
F      dch41 =xmolg*xch4
F      dc2h61=xmolg*xc2h6
F      dc3h81=xmolg*xc3h8
F      dco21 =xmolg*xco2
F      dh21  =xmolg*xh2
F      dhhc1 =xmolg*xhhc
F      dh2s1 =xmolg*xh2s

F      dea1=akin
F      dea2=akin
F      dea3=akin
;
;  For KERO reaction
F      n=kccidc('KERO')
F      wm    =plx(ioff+n)
F      wtc=wm*wtfc
F      wto=wm*wtfo
F      wtg=wm-wtc-wto
F      xmolg=wtg/wmix
F      ec1=wtc/wc
F      eo1=wto/wo
F      ech41 =xmolg*xch4
F      ec2h61=xmolg*xc2h6
F      ec3h81=xmolg*xc3h8
F      eco21 =xmolg*xco2
F      eh21  =xmolg*xh2
F      ehhc1 =xmolg*xhhc
F      eh2s1 =xmolg*xh2s

F      eea1=akin
F      eea2=akin
F      eea3=akin
;
;  For HNAPH reaction
F      n=kccidc('HNAPH')
F      wm    =plx(ioff+n)
F      wtc=wm*wtfc
F      wto=wm*wtfo
F      wtg=wm-wtc-wto
F      xmolg=wtg/wmix
F      fc1=wtc/wc
F      fo1=wto/wo
F      fch41 =xmolg*xch4
F      fc2h61=xmolg*xc2h6
F      fc3h81=xmolg*xc3h8
F      fco21 =xmolg*xco2
F      fh21  =xmolg*xh2
F      fhhc1 =xmolg*xhhc
F      fh2s1 =xmolg*xh2s

F      fea1=akin
F      fea2=akin
F      fea3=akin
;
; set reactors 2 & 3 to reactor 1 stoic
```

```
F     ac2=ac1
F     ao2=ao1
F     ach42 =ach41
F     ac2h62=ac2h61
F     ac3h82=ac3h81
F     aco22 =aco21
F     ah22  =ah21
F     ahhc2=ahhc1
F     ah2s2 =ah2s1
F     ac3=ac1
F     ao3=ao1
F     ach43 =ach41
F     ac2h63=ac2h61
F     ac3h83=ac3h81
F     aco23 =aco21
F     ah23  =ah21
F     ahhc3=ahhc1
F     ah2s3 =ah2s1
;
F     bc2=bc1
F     bo2=bo1
F     bch42 =bch41
F     bc2h62=bc2h61
F     bc3h82=bc3h81
F     bco22 =bco21
F     bh22  =bh21
F     bhhc2=bhhc1
F     bh2s2 =bh2s1
F     bc3=bc1
F     bo3=bo1
F     bch43 =bch41
F     bc2h63=bc2h61
F     bc3h83=bc3h81
F     bco23 =bco21
F     bh23  =bh21
F     bhhc3=bhhc1
F     bh2s3 =bh2s1
;
F     cc2=cc1
F     co2=co1
F     cch42 =cch41
F     cc2h62=cc2h61
F     cc3h82=cc3h81
F     cco22 =cco21
F     ch22  =ch21
F     chhc2=chhc1
F     ch2s2 =ch2s1
F     cc3=cc1
F     co3=co1
F     cch43 =cch41
F     cc2h63=cc2h61
F     cc3h83=cc3h81
F     cco23 =cco21
F     ch23  =ch21
F     chhc3=chhc1
F     ch2s3 =ch2s1
;
F     dc2=dc1
F     do2=do1
F     dch42 =dch41
F     dc2h62=dc2h61
F     dc3h82=dc3h81
F     dco22 =dco21
F     dh22  =dh21
```

```
F      dhhc2=dhhc1
F      dh2s2 =dh2s1
F      dc3=dc1
F      do3=do1
F      dch43 =dch41
F      dc2h63=dc2h61
F      dc3h83=dc3h81
F      dco23 =dco21
F      dh23  =dh21
F      dhhc3=dhhc1
F      dh2s3 =dh2s1
;
F      ec2=ec1
F      eo2=eo1
F      ech42 =ech41
F      ec2h62=ec2h61
F      ec3h82=ec3h81
F      eco22 =eco21
F      eh22  =eh21
F      ehhc2=ehhc1
F      eh2s2 =eh2s1
F      ec3=ec1
F      eo3=eo1
F      ech43 =ech41
F      ec2h63=ec2h61
F      ec3h83=ec3h81
F      eco23 =eco21
F      eh23  =eh21
F      ehhc3=ehhc1
F      eh2s3 =eh2s1
;
F      fc2=fc1
F      fo2=fo1
F      fch42 =fch41
F      fc2h62=fc2h61
F      fc3h82=fc3h81
F      fco22 =fco21
F      fh22  =fh21
F      fhhc2=fhhc1
F      fh2s2 =fh2s1
F      fc3=fc1
F      fo3=fo1
F      fch43 =fch41
F      fc2h63=fc2h61
F      fc3h83=fc3h81
F      fco23 =fco21
F      fh23  =fh21
F      fhhc3=fhhc1
F      fh2s3 =fh2s1
;
   EXECUTE BEFORE RU1


;=============================================================================
;  Control
;=============================================================================

ACCOUNT-INFO ACCOUNT=HPASPEN PROJECT-ID=P  &
        PROJECT-NAME="Oil Upgrading" USER-NAME="CHE"

IN-UNITS SI TEMPERATURE=C

;OUT-UNITS ENG VOLUME-FLOW='BBL/DAY' ENTHALPY-FLO='MMBTU/HR'  &
;         VOLUME=BBL HEAD=FT HEAT=MMBTU
```

```
REPORT  LINES=78

RUN-CONTROL  MAX-TIME=180

CONV-OPTIONS
   PARAM TEAR-METHOD=BROYDEN
   WEGSTEIN MAXIT=60

;  allows property parameters to be examined
PROPERTY-REPORT PROJECT

;----- Report Style 1 -----------------------------------------------
;STREAM-REPORT NOMOLEFLOW STDVOLFLOW PROPERTIES=PETRO TOTAL

PROP-SET PETRO CPMX VLSTDMX APISTD SGSTD WAT TBPCRV D86CRV  &
        D1160CRV UNITS='BBL/DAY' 'BBL/HR' SUBSTREAM=MIXED   &
        BASIS=DRY

PROP-SET PETRO2 VLSTDMX APISTD SGSTD WAT  &
        UNITS='BBL/DAY' 'BBL/HR' SUBSTREAM=MIXED   &
        BASIS=DRY

PROP-SET VISC MUMX KMX  PHASE=L1

PROP-SET VISCG MUMX PHASE=V

;
PROP-SET TOTAL CPMX TBUB PBUB MASSVFRAC

;----- Report Style 2 -----------------------------------------------
STREAM-REPORT MASSFLOW PROPERTIES=PETRO2 VISC VISCG O1 O2 O5 ; O1 & O2 see OUT section


;----- Report Style 3 -----------------------------------------------
;  Primary flowsheet streams
;STREAM-REPORT WIDE MASSFLOW NOZEROFLOW PROPERTIES=O5 &
;  INCL-STREAMS= &
;      COIL-O CONDH-G CONDH-FW CONDH-O &
;      CONDL-FW CONDL-G CONDL-O CPMP-O CSU-FW CSU-G CSU-O CWSU-O &
;      FEED FOIL-G FOIL-O FPMP-O FUEL-G &
;      GASR-G1 GASR-G2 &
;      HTR-E HTXG-E HTXG-G HTXO-E HTXO-O &
;      IPMP-E &
;      MXA-E MXAA-E MXB-O MXC-G &
;      PGSU-E PGSU-G PSU-W2 PRD-O PUMP-E &
;      RUGS-G &
;      SSU-W2 &
;      WCU-W WSU-W2 &
;      WCU-Q HTR-Q CONDH-Q CONDH-Q COIL-Q  &
;      IPMP-WK PUMP-WK CPMP-WK FPMP-WK

;  Property sets used in computations & printout

PROP-SET PRHOL RHOMX PHASE=L  UNITS='KG/CUM'
PROP-SET PRHOV RHOMX PHASE=V  UNITS='KG/CUM'
PROP-SET PFLWL MASSFLMX PHASE=L

PROP-SET O1 APISTD
PROP-SET O2 MASSFLMX PHASE=L2
PROP-SET O3 MASSFLMX PHASE=L1
PROP-SET O4 MUMX     PHASE=L1
PROP-SET O5 MASSVFRAC
```

```
;=============================================================================
; Component & Property Setup
;=============================================================================

SIM-OPTIONS FREE-WATER=YES

INSERT * API  ; use API for liquid volumes in all option sets
;PROPERTIES   PRMHV2  SOLU-WATER=0
PROPERTIES   GRAYSON  SOLU-WATER=1
;PROPERTIES   PSRK  SOLU-WATER=0

DATABANKS PURECOMP  / AQUEOUS  / SOLIDS  / INORGANIC  /  NOASPENPCD

PROP-SOURCES PURECOMP  / AQUEOUS  / SOLIDS  / INORGANIC

;  Section to change solubility computation.                    Data #2
PROP-DATA
     PROP-LIST          WATSOL      ; exp(c1+c2/T+c3*T)
;                         Ci1          Ci2            Ci3
;  set low MW to zero
       PVAL  C2H6         -10          0.0            0
       PVAL  C3H8         -10          0.0            0
       PVAL  HHC          -10          0.0            0
       PVAL  P-OIL        -10          0.0            0
;  Default values
;      PVAL  HNAPH      7.35939      -4352.68         0
;      PVAL  KERO       7.24358      -4328.52         0
;      PVAL  AGO        7.12479      -4303.77         0
;      PVAL  LVGO       6.98712      -4275.15         0
;      PVAL  HVGO       6.85205      -4247.13         0
;      PVAL  VR         6.72077      -4219.96         0

;  Active values  (parameters with maxima in solubility curve)
       PVAL  HNAPH        24.35       -8000        -0.019
       PVAL  KERO         24.35       -8000        -0.019
       PVAL  AGO          24.35       -8000        -0.019
       PVAL  LVGO         24.35       -8000        -0.019
       PVAL  HVGO         24.35       -8000        -0.019
       PVAL  VR           24.35       -8000        -0.019


COMPONENTS                                                      ; Data #28
    H2O H2O H2O  / CH4 CH4 CH4 / C2H6 C2H6 C2H6 / C3H8 C3H8 C3H8 &
  / HHC C4H10-1 HHC  / H2S H2S H2S / CO2 CO2 CO2 / H2 H2 H2 &
  / P-OIL  / COKE  C  COKE

PC-USER
;    PC-DEF ASPEN  P-LE    NBP=   0   API=60
;    PC-DEF ASPEN  P-LGASO NBP= 51.7  API=55
;    PC-DEF ASPEN  P-LNAPH NBP= 121.1 API=50
;    PC-DEF ASPEN  P-HNAPH NBP= 176.7 API=45
;    PC-DEF ASPEN  P-KERO  NBP= 232.2 API=35
;    PC-DEF ASPEN  P-AGO   NBP= 301.7 API=30
;    PC-DEF ASPEN  P-LVGO  NBP= 385   API=25
;    PC-DEF ASPEN  P-HVGO  NBP= 482.2 API=20
;    PC-DEF ASPEN  P-VR    NBP= 648.9 API=10
     PC-DEF ASPEN  P-OIL   NBP= 121.1 API=50     ; NBP in C    ; Data #27

;  Oil composition section                                      ; Data #1
PC-CALC
     PC-SET CRUDE
     PC-IDS OPTION=LIST &
           LIST=LE LGASO LNAPH HNAPH KERO AGO LVGO HVGO    VR
;    CUTS LIST= 0  60  175   300   400  500  650  800  1000    1600
```

```
    CUTS LIST= 0 15.6 79.4 148.9 204.4 260 343.3 426.7 537.8 871.1

ADA-SETUP
    ADA-SETUP PROCEDURE=REL9

ASSAY CUT1      ; made up
    ASSAY-DATA API=37.1
;<F>DIST-CURVE D86   0  360 / 20  365 / 80  370 / 100 375
    DIST-CURVE D86   0  182.2 / 20  185.0 / 80  187.8 / 100 190.6

ASSAY CUT2
    ASSAY-DATA API=29.9
;<F>DIST-CURVE D86   0  430                                  &
;                  /  5  446 / 10  450 / 20  456 / 30  462 &
;                  / 40  466 / 50  470 / 60  476 / 70  482 &
;                  / 80  490 / 90  504 / 95  508 / 99  514
    DIST-CURVE D86   0  221.1                                &
                   /  5  230.0 / 10  232.2 / 20  235.6 / 30  238.9 &
                   / 40  241.1 / 50  243.3 / 60  246.7 / 70  250.0 &
                   / 80  254.4 / 90  262.2 / 95  264.4 / 99  267.8

ASSAY CUT3
    ASSAY-DATA API=23.7
;<F>DIST-CURVE D86   0  520                                  &
;                  /  5  544 / 10  558 / 20  562 / 30  568 &
;                  / 40  570 / 50  574 / 60  580 / 70  584 &
;                  / 80  590 / 90  600 / 95  610 / 99  618
    DIST-CURVE D86   0  271.1                                  &
                   /  5  284.4 / 10  292.2 / 20  294.4 / 30  297.8 &
                   / 40  298.9 / 50  301.1 / 60  304.4 / 70  306.7 &
                   / 80  310.0 / 90  315.6 / 95  321.1 / 99  325.6

ASSAY CUT4      ; Vac
    ASSAY-DATA API=16.0
;<F>DIST-CURVE TBPLV 0  637                                  &
;                  /  5  664 / 10  683 / 20  688 / 30  698 &
;                  / 40  708 / 50  718 / 60  737 / 70  755 &
;                  / 80  778 / 90  806 / 95  827 / 99  844
    DIST-CURVE TBPLV 0  336.1                                  &
                   /  5  351.1 / 10  361.7 / 20  364.4 / 30  370.0 &
                   / 40  375.6 / 50  381.1 / 60  391.7 / 70  401.7 &
                   / 80  414.4 / 90  430.0 / 95  441.7 / 99  451.1

ASSAY CUT5     ; Vac
    ASSAY-DATA API=13.8
;<F>DIST-CURVE TBPLV 0  686                                  &
;                  /  5  734 / 10  760 / 20  797 / 30  816 &
;                  / 40  834 / 50  851 / 60  868 / 70  887 &
;                  / 80  908 / 90  938 / 95  952 / 99  973
    DIST-CURVE TBPLV 0  363.3                                  &
                   /  5  390.0 / 10  404.4 / 20  425.0 / 30  435.6 &
                   / 40  445.6 / 50  455.0 / 60  464.4 / 70  475.0 &
                   / 80  486.7 / 90  503.3 / 95  511.1 / 99  522.8

ASSAY CUT6     ; Vac
    ASSAY-DATA API=4.6
;<F>DIST-CURVE TBPLV 0  917                                    &
;                  /  5  938 / 10  979 / 20  998              &
;                 / 40 1045 / 60 1085 / 80 1130 / 90 1165
    DIST-CURVE TBPLV 0  491.7                                    &
                   /  5  503.3 / 10 526.1 / 20 536.7           &
                  / 40  562.8 / 60 585.0 / 80 610.0 / 90 629.4
                 ;  Above 20% from log probability curve
```

```
BLEND CRUDE
;  core labs crude API 12
;   MASS-FRAC CUT1 0.0087 /  CUT2 0.0536 /  CUT3 0.108 / &
;             CUT4 0.1479 /  CUT5 0.2194 /  CUT6 0.4608
;  add lights to get API of 13.5
    MASS-FRAC CUT1 0.0496 /  CUT2 0.0764 /  CUT3 0.1007 / &
              CUT4 0.1378 /  CUT5 0.2045 /  CUT6 0.4295


;=====================================================================
; Flowsheet
;=====================================================================

FLOWSHEET

; Area A -------------------------------------------------------
    BLOCK PFW    IN=FEED              OUT=PFW-O PFW-FW
    BLOCK PSU    IN=PFW-FW            OUT=PSU-W1 PSU-W2
    BLOCK MXA    IN=PSU-W1 PFW-O      OUT=MXA-E
    BLOCK IPMP   IN=MXA-E             OUT=IPMP-E IPMP-WK
    BLOCK PRH    IN=IPMP-E GASR-G1    OUT=PRH-E
    BLOCK PGSU   IN=PRH-E             OUT=PGSU-G PGSU-E
    BLOCK IHTR   IN=PGSU-E            OUT=IHTR-E IHTR-Q
    BLOCK SFW    IN=IHTR-E            OUT=SFW-O SFW-FW
    BLOCK SSU    IN=SFW-FW            OUT=SSU-W1 SSU-W2
    BLOCK WCU    IN=SSU-W2            OUT=WCU-W WCU-Q
    BLOCK MXAA   IN=SSU-W1 SFW-O      OUT=MXAA-E
; Area B -------------------------------------------------------
    BLOCK XGSU   IN=MXAA-E            OUT=XGSU-G XGSU-E
    BLOCK PUMP   IN=XGSU-E            OUT=PUMP-E  PUMP-WK
    BLOCK HTXG   IN=GASR-G2 PUMP-E    OUT=HTXG-G  HTXG-E
    BLOCK HTXO   IN=HTXG-E MXB-O      OUT=HTXO-E  HTXO-O
    BLOCK HTR    IN=HTXO-E            OUT=HTR-E  HTR-Q
; Area C -------------------------------------------------------
    BLOCK RU1    IN=HTR-E             OUT=RU1-E
    BLOCK RU2    IN=RU1-E             OUT=RU2-E
    BLOCK RU3    IN=RU2-E             OUT=RU3-E
    BLOCK RUGS   IN=RU3-E             OUT=RUGS-G RUGS-E
    BLOCK GASR   IN=RUGS-G            OUT=GASR-G1 GASR-G2
    BLOCK RFW    IN=RUGS-E            OUT=RFW-O  RFW-FW
    BLOCK RSU    IN=RFW-FW            OUT=RSU-W1 RSU-W2
    BLOCK MXB    IN=RSU-W1 RFW-O      OUT=MXB-O
; Area D -------------------------------------------------------
    BLOCK MXC    IN=PGSU-G  FOIL-G    OUT=MXC-G
    BLOCK CONDL  IN=MXC-G             OUT=CONDL-G CONDL-O CONDL-FW CONDL-Q
    BLOCK CONDH  IN=HTXG-G            OUT=CONDH-G CONDH-O CONDH-FW CONDH-Q
    BLOCK CSU    IN=CONDH-G CONDH-O CONDH-FW &
                                  OUT=CSU-G CSU-O CSU-FW
    BLOCK MXG    IN=CONDL-G CSU-G XXGSU-G &
                                  OUT=FUEL-G
    BLOCK MXWSU  IN=CONDL-FW CONDL-O CSU-FW CSU-O &
                                  OUT=MXWSU-O MXWSU-FW
    BLOCK WSU    IN=MXWSU-FW          OUT=WSU-W1 WSU-W2
    BLOCK CWSU   IN=MXWSU-O  WSU-W1   OUT=CWSU-O
    BLOCK XXGSU  IN=CWSU-O            OUT=XXGSU-G XXGSU-O
; Area E -------------------------------------------------------
    BLOCK FOIL   IN=HTXO-O            OUT=FOIL-G FOIL-O FOIL-Q
    BLOCK COIL   IN=FOIL-O            OUT=COIL-O COIL-Q
    BLOCK CPMP   IN=COIL-O            OUT=CPMP-O CPMP-WK
    BLOCK PRD    IN=CPMP-O FPMP-O     OUT=PRD-O
    BLOCK FPMP   IN=XXGSU-O           OUT=FPMP-O FPMP-WK

;=====================================================================
; Streams
```

```
;=============================================================================

DEF-STREAMS MIXCISLD ALL
DEF-STREAMS HEAT HTR-Q COIL-Q IHTR-Q CONDL-Q CONDH-Q FOIL-Q WCU-Q
DEF-STREAMS WORK PUMP-WK IPMP-WK CPMP-WK FPMP-WK

STREAM FEED
     SUBSTREAM MIXED TEMP=100 <F> PRES=10 <psi>     ;            Data #0
     STDVOL-FLOW CRUDE  3000  <BBL/DAY>
     STDVOL-FLOW H2O    30000  <BBL/DAY>


;=============================================================================
; Block specifications
;=============================================================================

; Area A -------------------------------------------------------

BLOCK PFW  MIXER
     DESCRIPTION 'Artifical separation of free water'

BLOCK PSU     FSPLIT
     DESCRIPTION 'Initial water separation'
     PARAM NPHASE=1  PHASE=L
     FRAC  PSU-W1  0.5  ; split set in FPSU
FORTRAN FPSU
F      common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
     DEFINE fwato MASS-FLOW  STREAM=PFW-O  COMPONENT=H2O
     DEFINE foilt STREAM-VAR STREAM=PFW-O  VARIABLE=MASS-FLOW
     DEFINE fwat  STREAM-VAR STREAM=PFW-FW VARIABLE=MASS-FLOW
     DEFINE wsplt BLOCK-VAR  BLOCK=PSU  SENTENCE=FRAC &
                 VARIABLE=FRAC  ID1=PSU-W1
F      wtot=fwato+fwat
F      foil=foilt-fwato
F      foil=max(foil,0.0001)
F      wor=wtot/foil
F      if (wor .gt. wor1) then
F         w=foil*wor1
F         ws=w-fwato
F         if (ws .gt. 0.0 .and. fwat .gt. 0.0) then
F            wsplt=ws/fwat
F         else
F            wsplt=0.0
F         endif
F      else
F         wsplt=1.0
F      endif
     READ-VARS  fwato foilt fwat
     WRITE-VARS wsplt

BLOCK MXA  MIXER
     DESCRIPTION 'Recombine oil/water '

BLOCK IPMP  PUMP
     DESCRIPTION 'Initial pumping to raise pressure for heat reovery'
     PARAM  PRES=100 <psi>     ; Pressure set in FINPUT

BLOCK PRH  MIXER
     DESCRIPTION 'Combine hot reactor vapors'

BLOCK  PGSU     FLASH2
     DESCRIPTION 'Separate out gas phase'
     PARAM  DUTY=0
```

```
BLOCK IHTR  HEATER
    DESCRIPTION 'Trim temperature prior to setting reator water level'
    PARAM TEMP=300 <F> PRES=0 <psi>    ; Temperature set in FIHTR
FORTRAN FIHTR
F      common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
    DEFINE tin   STREAM-VAR STREAM=PGSU-E  VARIABLE=TEMP
    DEFINE tset  BLOCK-VAR  BLOCK=IHTR  SENTENCE=PARAM  VARIABLE=TEMP
F      if (tihtr .le. tin) then
F         tset=tin
F      else
F         tset=tihtr
F      endif
    READ-VARS  tin
    WRITE-VARS tset

BLOCK SFW  MIXER
    DESCRIPTION 'Artifical separation of free water'

BLOCK SSU    FSPLIT
    DESCRIPTION 'Secondary water separation'
    PARAM NPHASE=1  PHASE=L
    FRAC  SSU-W1  0.5  ; split set in FSSU
FORTRAN FSSU
F      common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
    DEFINE fwato MASS-FLOW  STREAM=SFW-O  COMPONENT=H2O
    DEFINE foilt STREAM-VAR STREAM=SFW-O  VARIABLE=MASS-FLOW
    DEFINE fwat  STREAM-VAR STREAM=SFW-FW VARIABLE=MASS-FLOW
    DEFINE wsplt BLOCK-VAR  BLOCK=SSU  SENTENCE=FRAC &
                VARIABLE=FRAC  ID1=SSU-W1
F      wtot=fwato+fwat
F      foil=foilt-fwato
F      foil=max(foil,0.0001)
F      wor=wtot/foil
F      if (wor .gt. wor2) then
F         w=foil*wor2
F         ws=w-fwato
F         if (ws .gt. 0.0 .and. fwat .gt. 0.0) then
F            wsplt=ws/fwat
F         else
F            wsplt=0.0
F         endif
F      else
F         wsplt=1.0
F      endif
    READ-VARS  fwato foilt fwat
    WRITE-VARS wsplt

BLOCK WCU  HEATER
    DESCRIPTION 'Cool water from SSU'
    PARAM TEMP=75 <F>  PRES=0

BLOCK MXAA MIXER
    DESCRIPTION 'Recombine oil/water '

; Area B ------------------------------------------------------

BLOCK  XGSU    FLASH2
    DESCRIPTION 'Separate out gas phase to insure PUMP calculation ok'
    PARAM  DUTY=0

BLOCK PUMP  PUMP
    DESCRIPTION 'Pump primary emulsion to desired pressure'
    PARAM  PRES=2000 <psi>    ; Reactor pressure set in FINPUT
```

```
BLOCK HTXG  HEATX
    DESCRIPTION 'Heat recovery heat exchanger (gas/oil)'
    PARAM  DELT-HOT=70 <F>  ; Approach temp set in FINPUT
    HEAT-TR-COEF  U=50 <BTU/HR-SQFT-R>   ; Overall heat tx coef  Data #22
    FEEDS      HOT=GASR-G2 COLD=PUMP-E
    PRODUCTS  HOT=HTXG-G  COLD=HTXG-E

BLOCK HTXO  HEATX
    DESCRIPTION 'Heat recovery heat exchanger  (oil/oil)'
    PARAM  DELT-HOT=70  ; Approach temp (F) set in FINPUT
;   HEAT-TR-COEF  U=60  <BTU/HR-SQFT-R>  ; Overall heat tx coef  Data #22
    HEAT-TR-COEF  L-L=30  <BTU/HR-SQFT-R> &
                L-B=100 <BTU/HR-SQFT-R> &
                L-V=10  <BTU/HR-SQFT-R>  ;                    Data #22
    FLASH-SPECS HTXO-O NPHASE=1 PHASE=L
    FEEDS      HOT=MXB-O   COLD=HTXG-E
    PRODUCTS  HOT=HTXO-O  COLD=HTXO-E

BLOCK HTR  HEATER
    DESCRIPTION 'Trim temperature for reactor/separator'
    PARAM TEMP=600 <F> PRES=0   ; Reactor temperature set in FINPUT

; Area C ------------------------------------------------------

BLOCK RU1  RCSTR
    DESCRIPTION 'Reactor unit 1'
    PARAM DUTY=0 NPHASE=2  PHASE=L VOL=10 <BBL> REACT-VOL=10 <BBL> PRES=0
              ; reaction controlled by react-vol set in FRU

    STOIC 1 MIXED VR    -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 1 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 1 VR    1   ;                                    Data #21

    STOIC 2 MIXED HVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 2 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 2 HVGO  1   ;                                    Data #21

    STOIC 3 MIXED LVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 3 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 3 LVGO  1   ;                                    Data #21

    STOIC 4 MIXED AGO   -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 4 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 4 AGO   1   ;                                    Data #21

    STOIC 5 MIXED KERO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 5 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 5 KERO  1   ;                                    Data #21

    STOIC 6 MIXED HNAPH -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 6 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 6 HNAPH 1   ;                                    Data #21

BLOCK RU2  RCSTR
    DESCRIPTION 'Reactor unit 2'
    PARAM DUTY=0 NPHASE=2  PHASE=L VOL=10 <BBL> REACT-VOL=10 <BBL> PRES=0
              ; reaction controlled by react-vol set in FRU

    STOIC 1 MIXED VR    -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 1 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
    POWLAW-EXP 1 VR    1   ;                                    Data #21

    STOIC 2 MIXED HVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 2 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)   Data #21
```

```
    POWLAW-EXP 2 HVGO  1   ;                                        Data #21

    STOIC 3 MIXED LVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 3 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 3 LVGO  1   ;                                        Data #21

    STOIC 4 MIXED AGO   -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 4 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 4 AGO   1   ;                                        Data #21

    STOIC 5 MIXED KERO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 5 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 5 KERO  1   ;                                        Data #21

    STOIC 6 MIXED HNAPH -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 6 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 6 HNAPH 1   ;                                        Data #21

BLOCK RU3  RCSTR
    DESCRIPTION 'Reactor unit 3'
    PARAM DUTY=0 NPHASE=2  PHASE=L VOL=10 <BBL> REACT-VOL=10 <BBL> PRES=0
          ; reaction controlled by react-vol set in FRU

    STOIC 1 MIXED VR    -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 1 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 1 VR    1   ;                                        Data #21

    STOIC 2 MIXED HVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 2 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 2 HVGO  1   ;                                        Data #21

    STOIC 3 MIXED LVGO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 3 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 3 LVGO  1   ;                                        Data #21

    STOIC 4 MIXED AGO   -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 4 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 4 AGO   1   ;                                        Data #21

    STOIC 5 MIXED KERO  -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 5 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 5 KERO  1   ;                                        Data #21

    STOIC 6 MIXED HNAPH -1 / MIXED P-OIL * ; components & ceoffs set in SETR
    RATE-CON 6 14.2e7  39.74[kcal/mol] ; Eact & pre-exp (1/s)    Data #21
    POWLAW-EXP 6 HNAPH 1   ;                                        Data #21

FORTRAN FRU
F     common /usr3/ capfac
F     common /usr4/ isig,old(10),iold
; Compute number of vessels and volumes for total reactor system based
; on sep correlations.  Define appropiate parameters for reactor routines
; and costing module.
F     common /usr1/ rtres, tihtr, wor1, wor2, wor3, wor4, worf
F     common /plex/ b(1)
F     character*1 tdum
    DEFINE flwi  STREAM-VAR STREAM=HTR-E  VARIABLE=MASS-FLOW
    DEFINE flowli STREAM-PROP STREAM=HTR-E  PROPERTY=PFLWL
    DEFINE denli STREAM-PROP STREAM=HTR-E  PROPERTY=PRHOL
    DEFINE dengi STREAM-PROP STREAM=HTR-E  PROPERTY=PRHOV
    DEFINE flwo  STREAM-VAR STREAM=RU3-E  VARIABLE=MASS-FLOW
    DEFINE flowlo STREAM-PROP STREAM=RU3-E  PROPERTY=PFLWL
    DEFINE denlo STREAM-PROP STREAM=RU3-E  PROPERTY=PRHOL
    DEFINE dengo STREAM-PROP STREAM=RU3-E  PROPERTY=PRHOV
```

```
;
    DEFINE pres  BLOCK-VAR BLOCK=PUMP SENTENCE=PARAM  VARIABLE=PRES
    DEFINE vol1  BLOCK-VAR BLOCK=RU1 SENTENCE=PARAM  VARIABLE=VOL
    DEFINE volr1 BLOCK-VAR BLOCK=RU1 SENTENCE=PARAM  VARIABLE=REACT-VOL
    DEFINE vol2  BLOCK-VAR BLOCK=RU2 SENTENCE=PARAM  VARIABLE=VOL
    DEFINE volr2 BLOCK-VAR BLOCK=RU2 SENTENCE=PARAM  VARIABLE=REACT-VOL
    DEFINE vol3  BLOCK-VAR BLOCK=RU3 SENTENCE=PARAM  VARIABLE=VOL
    DEFINE volr3 BLOCK-VAR BLOCK=RU3 SENTENCE=PARAM  VARIABLE=REACT-VOL
    DEFINE volc  CBLOCK-VAR CBLOCK=C-RU SENTENCE=SIZING-DATA &
       VARIABLE=VOL
    DEFINE temp  BLOCK-VAR  BLOCK=HTR  SENTENCE=PARAM  VARIABLE=TEMP
    DEFINE diamc CBLOCK-VAR CBLOCK=C-RU SENTENCE=SIZING-DATA &
       VARIABLE=DIAM
    DEFINE presc CBLOCK-VAR CBLOCK=C-RU SENTENCE=SIZING-DATA &
       VARIABLE=PRES
    DEFINE tempc CBLOCK-VAR CBLOCK=C-RU SENTENCE=SIZING-DATA &
       VARIABLE=TEMP
    DEFINE nequip CBLOCK-VAR  CBLOCK=C-RU SENTENCE=COSTING-DATA &
       VARIABLE=NEQUIP
;  used average properties in/out of reactor (except on 1st pass)
;  (wti weighting factor for in flow, [1-wti] used for outflow)  Data #23
F      if (flwo.gt.0.0 .and. flwo.lt. 1e10) then
F         wti=0.50
F         wto=1.0-wti
F         flw=wto*flwo+wti*flwi
F         flowl=wto*flowlo+wti*flowli
F         deng=wto*dengo+wti*dengi
F         denl=wto*denlo+wti*denli
F      else
F         flw=flwi
F         flowl=flowli
F         deng=dengi
F         denl=denli
F      endif
F      flw=capfac*flw
F      flowl=capfac*flowl
;
;  set liquid volume in m**3 based on residence time and average liq flow
F     vliq=rtres*60*flowl/denl
F     flowg=flw-flowl
;  maximum vessel volume                                          Data #23
F     vmax=2521*exp(-pres/4.723e6)
;  minimum L/D ratio                                              Data #23
F     xldr=3.0
;  minimum diameter (m)                                           Data #23
F     dmin=5.0*0.3048
;  minimum void height if horizontal (m)                          Data #23
F     hmin=1.0*0.3048
F     write(nrpt,'(''C-RU'')')
;  protect against bad flows, use old values if set,
;     otherwise use default values
F     if (flowl.le.0.0.or.flowl.ge.1.0e10) then
F        if (isig.eq.1234) then
F           vol=old(1)
F           number=iold
F           vliq=old(2)
F           diam=old(3)
F        else
F           vol=10.0
F           number=1
F           vliq=8
F        endif
F     else
F        call sep(nrpt,'H','S','Y','N',flowg,flowl,vliq,vmax,denl,deng,
```

```
F     &          dum,dum,xldr,hmin,dmin,number,vol,xlen,diam,tdum)
F      endif
F      write(nrpt,'(''Debug vol,vliq '',2f10.3)') vol,vliq
F      volc=vol
F      vol1=number*volc/3.0
F      vol2=vol1
F      vol3=vol1
F      volr1=vliq/3.0
F      volr2=volr1
F      volr3=volr1
F      write(nrpt,'(''vol & volr 1-3'',6f8.0)')
F     &        vol1,volr1,vol2,volr2,vol3,volr3
F      nequip=number
F      diamc=diam
F      tempc=temp
F      presc=pres
F      isig=1234
F      iold=number
F      old(1)=vol
F      old(2)=vliq
F      old(3)=diam
     READ-VARS flowli denli dengi flowlo denlo dengo pres
     WRITE-VARS vol1 vol2 vol3 volr1 volr2 volr3 volc diamc nequip

BLOCK  RUGS     FLASH2
     DESCRIPTION 'Separate out gas phase '
     PARAM  DUTY=0

BLOCK  GASR  FSPLIT
     DESCRIPTION 'Split gas for recycle'
     FRAC GASR-G1 0.5   ; set in FINPUT

BLOCK RFW  MIXER
     DESCRIPTION 'Artifical separation of free water'

BLOCK RSU     FSPLIT
     DESCRIPTION 'Reactor water separation'
     PARAM  NPHASE=1  PHASE=L
     FRAC  RSU-W1 0.01
FORTRAN FRSU
F      common /usr1/ rtres,  tihtr, wor1, wor2, wor3, wor4, worf
     DEFINE fwato MASS-FLOW  STREAM=RFW-O  COMPONENT=H2O
     DEFINE foilt STREAM-VAR STREAM=RFW-O  VARIABLE=MASS-FLOW
     DEFINE fwat  STREAM-VAR STREAM=RFW-FW VARIABLE=MASS-FLOW
     DEFINE wsplt BLOCK-VAR  BLOCK=RSU  SENTENCE=FRAC &
                 VARIABLE=FRAC  ID1=RSU-W1
F      wtot=fwato+fwat
F      foil=foilt-fwato
F      foil=max(foil,0.0001)
F      wor=wtot/foil
F      if (wor .gt. wor3) then
F         w=foil*wor3
F         ws=w-fwato
F         if (ws .gt. 0.0 .and. fwat .gt. 0.0) then
F            wsplt=ws/fwat
F         else
F            wsplt=0.0
F         endif
F      else
F         wsplt=1.0
F      endif
     READ-VARS  fwato foilt fwat
     WRITE-VARS wsplt
```

```
BLOCK MXB  MIXER
    DESCRIPTION 'Recombine oil/water '

; Area D ------------------------------------------------------

BLOCK MXC  MIXER
    DESCRIPTION 'Combine gases prior to  low P condenser for convenience'

BLOCK CONDL FLASH2
    DESCRIPTION 'Gas condenser - low P'
    PARAM  TEMP=75 <F>  PRES=0  ; no pressure drop                Data #24

BLOCK CONDH FLASH2
    DESCRIPTION 'Gas condenser - high P'
    PARAM  TEMP=75 <F>  PRES=0  ; no pressure drop                Data #25

BLOCK CSU   FLASH2
    DESCRIPTION 'Liquid knock out for high pressure condenser'
    PARAM  PRES=30 <psi>    ;                                    Data #26

BLOCK MXG  MIXER
    DESCRIPTION 'Combine condenser gases'

BLOCK MXWSU  MIXER
    DESCRIPTION 'Combine streams from low P for separation tank'

BLOCK WSU    FSPLIT
    DESCRIPTION 'Low P oil water separation'
    PARAM  NPHASE=1  PHASE=L
    FRAC  WSU-W1 0.01
FORTRAN FWSU
F     common /usr1/ rtres,  tihtr, wor1, wor2, wor3, wor4, worf
    DEFINE fwato MASS-FLOW  STREAM=MXWSU-O   COMPONENT=H2O
    DEFINE foilt STREAM-VAR STREAM=MXWSU-O   VARIABLE=MASS-FLOW
    DEFINE fwat  STREAM-VAR STREAM=MXWSU-FW VARIABLE=MASS-FLOW
    DEFINE wsplt BLOCK-VAR  BLOCK=WSU  SENTENCE=FRAC &
                VARIABLE=FRAC  ID1=WSU-W1
F     wtot=fwato+fwat
F     foil=foilt-fwato
F     foil=max(foil,0.0001)
F     wor=wtot/foil
F     if (wor .gt. wor4) then
F        w=foil*wor4
F        ws=w-fwato
F        if (ws .gt. 0.0 .and. fwat .gt. 0.0) then
F           wsplt=ws/fwat
F        else
F           wsplt=0.0
F        endif
F     else
F        wsplt=1.0
F     endif
    READ-VARS  fwato foilt fwat
    WRITE-VARS wsplt

BLOCK CWSU  MIXER
    DESCRIPTION 'Recombine oil/water '

BLOCK  XXGSU    FLASH2
    DESCRIPTION 'Separate out gas phase to insure PUMP calculation ok'

; Area E ------------------------------------------------------

BLOCK FOIL FLASH2
```

```
    DESCRIPTION 'Flash oil product to remove water'
    PARAM  DUTY=0    PRES=30 <psi>

BLOCK COIL HEATER
    DESCRIPTION 'Cool final oil product'
    PARAM  TEMP=200 <F>

BLOCK CPMP  PUMP
    DESCRIPTION 'Raise oil pressure to desired delivery pressure'
    PARAM  PRES=200 <psi>  ; Pressure set in FINPUT

BLOCK FPMP  PUMP
    DESCRIPTION 'Raise oil pressure to desired delivery pressure'
    PARAM  PRES=200 <psi>  ; Pressure set in FINPUT

BLOCK PRD MIXER

;--------------------------------


;=============================================================================
;  Design Specifications
;=============================================================================

DESIGN-SPEC SPEC1  ; set gas recycle to achieve desired preheat temperature
F     common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
   DEFINE tmpo STREAM-VAR STREAM=PRH-E  VARIABLE=TEMP
   DEFINE tmpi STREAM-VAR STREAM=IPMP-E VARIABLE=TEMP
F     if (tihtr .lt. 0.0) then
F        err=tmpo-tmpi
F     else
F        err=tmpo-tihtr
F     endif
   SPEC err TO 0
   TOL-SPEC 0.1
   VARY BLOCK-VAR BLOCK=GASR SENTENCE=FRAC VARIABLE=FRAC ID1=GASR-G1
   LIMITS 0 1.0


;DESIGN-SPEC SPEC2  ; set final water content in oil at PRD-O by
;                   ; adjusting oil HTX to deliver proper temp
;F     common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
;   DEFINE tao  BLOCK-VAR  BLOCK=HTXO SENTENCE=PARAM  VARIABLE=DELT-HOT
;   DEFINE water MASS-FLOW  STREAM=PRD-O  COMPONENT=H2O
;   DEFINE flow  STREAM-VAR STREAM=PRD-O  VARIABLE=MASS-FLOW
;F     frac=water/flow
;   SPEC frac TO 'worf'
;   TOL-SPEC 0.0001
;   VARY BLOCK-VAR BLOCK=HTXO SENTENCE=PARAM VARIABLE=DELT-HOT
;   LIMITS 340 1200

;=============================================================================
;  Summary output
;=============================================================================


FORTRAN OUT
F     common /usr1/ rtres, tihtr,  wor1, wor2, wor3, wor4, worf
F     common /usr2/ akin,xch4,xc2h6,xc3h8,xhhc,xco2,xh2,xh2s,wtfc,wtfo
;  DEFINE deng3 STREAM-PROP STREAM=3GAS  PROPERTY=DENM

   DEFINE tohi  STREAM-VAR STREAM=MXB-O VARIABLE=TEMP
   DEFINE toho  STREAM-VAR STREAM=HTXO-O VARIABLE=TEMP
   DEFINE toci  STREAM-VAR STREAM=HTXG-E VARIABLE=TEMP
```

```
DEFINE toco  STREAM-VAR STREAM=HTXO-E VARIABLE=TEMP
DEFINE fo    STREAM-VAR STREAM=PUMP-E VARIABLE=MASS-FLOW
DEFINE ao    BLOCK-VAR  BLOCK=HTXO SENTENCE=RESULTS VARIABLE=AREA-CALC
DEFINE dutyo BLOCK-VAR  BLOCK=HTXO SENTENCE=RESULTS VARIABLE=DUTY

DEFINE tghi  STREAM-VAR STREAM=GASR-G2 VARIABLE=TEMP
DEFINE tgho  STREAM-VAR STREAM=HTXG-G VARIABLE=TEMP
DEFINE tgci  STREAM-VAR STREAM=PUMP-E VARIABLE=TEMP
DEFINE tgco  STREAM-VAR STREAM=HTXG-E VARIABLE=TEMP
DEFINE fg    STREAM-VAR STREAM=PUMP-E VARIABLE=MASS-FLOW
DEFINE ag    BLOCK-VAR  BLOCK=HTXG SENTENCE=RESULTS VARIABLE=AREA-CALC
DEFINE dutyg BLOCK-VAR  BLOCK=HTXG SENTENCE=RESULTS VARIABLE=DUTY

DEFINE grx  BLOCK-VAR  BLOCK=GASR SENTENCE=FRAC   VARIABLE=FRAC

DEFINE wtrx  MASS-FLOW  STREAM=RUGS-E  COMPONENT=H2O
DEFINE flrx   STREAM-VAR STREAM=RUGS-E VARIABLE=MASS-FLOW
DEFINE flrg   STREAM-VAR STREAM=RUGS-G VARIABLE=MASS-FLOW
DEFINE wtpx  MASS-FLOW  STREAM=PRD-O  COMPONENT=H2O
DEFINE flpx   STREAM-VAR STREAM=PRD-O  VARIABLE=MASS-FLOW

DEFINE apixi STREAM-PROP STREAM=PFW-O  PROPERTY=O1
DEFINE apixo STREAM-PROP STREAM=PRD-O  PROPERTY=O1
DEFINE foxi   STREAM-VAR STREAM=PFW-O  VARIABLE=MASS-FLOW
DEFINE wtix   MASS-FLOW  STREAM=PFW-O  COMPONENT=H2O
DEFINE foxo   STREAM-VAR STREAM=PRD-O  VARIABLE=MASS-FLOW

DEFINE chrx  MASS-FLOW STREAM=RU3-E SUBSTREAM=CISOLID COMPONENT=COKE
DEFINE poilx MASS-FLOW STREAM=RU3-E SUBSTREAM=MIXED   COMPONENT=P-OIL

DEFINE gch4  MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=CH4
DEFINE gc2h6 MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=C2H6
DEFINE gc3h8 MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=C3H8
DEFINE ghhc  MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=HHC
DEFINE gh2   MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=H2
DEFINE gpoil MOLE-FLOW STREAM=FUEL-G SUBSTREAM=MIXED   COMPONENT=P-OIL
DEFINE htrqx INFO-VAR  STREAM=HTR-Q  INFO=HEAT  VARIABLE=DUTY

DEFINE cost PROFIT-VAR SENTENCE=RESULTS VARIABLE=PROD-PRICE
DEFINE rhoo STREAM-PROP STREAM=PRD-O PROPERTY=PRHOL
DEFINE rhoi STREAM-PROP STREAM=PFW-O PROPERTY=PRHOL

DEFINE opoil MOLE-FLOW  STREAM=PRD-O   COMPONENT=P-OIL

DEFINE fwf1 STREAM-PROP STREAM=HTXG-E  PROPERTY=O2
DEFINE fwf2 STREAM-PROP STREAM=HTXO-E  PROPERTY=O2
DEFINE fwf3 STREAM-PROP STREAM=HTXO-O  PROPERTY=O2
DEFINE fof1 STREAM-PROP STREAM=HTXG-E  PROPERTY=O3
DEFINE fof2 STREAM-PROP STREAM=HTXO-E  PROPERTY=O3
DEFINE fof3 STREAM-PROP STREAM=HTXO-O  PROPERTY=O3

DEFINE v1   STREAM-PROP STREAM=HTXO-O  PROPERTY=O4

F     write(nrpt,'
F    &  (''====================================='',
F    &   ''====================================''/)')

F     write(nrpt,'(''Heat Exchanger Performance''
F    &   /3x,''HTXG'',
F    &   /3x,''Area (sqm):'',f8.0,''   sqm/kg/hr :'',f6.3,
F    &    3x,''Duty (MW):'',f6.1,
F    &   /9x,''  IN (C)   OUT (C) ''
F    &   /3x,'' Hot  '',2f8.0/3x,'' Cold '',2f8.0/3x,'' Delta'',2f8.0)')
F    &   ag,ag/fg,dutyg/1.0e6,tghi,tgho,tgci,tgco,tghi-tgco,tgho-tgci
```

```
F     write(nrpt,'(
F    &   /3x,''HTXO'',
F    &   /3x,''Area (sqm):'',f8.0,''   sqm/kg/hr :'',f6.3,
F    &    3x,''Duty (MW):'',f6.1,
F    &   /9x,''  IN (C)   OUT (C) ''
F    &   /3x,'' Hot  '',2f8.0/3x,'' Cold '',2f8.0/3x,'' Delta'',2f8.0)')
F    &   ao,ao/fo,dutyo/1.0e6,tohi,toho,toci,toco,tohi-toco,toho-toci

F     write(nrpt,'(/''Gas fraction for preheat: '',f6.3)') grx

F     write(nrpt,'(/''Water wt% at reactor & product: '',2f6.2)')
F    &    wtrx/flrx*100,wtpx/flpx*100

F     wtp1=0.0
F     wtp2=0.0
F     wtp3=0.0
F     if (fwf1.lt.1e30) wtp1=100*fwf1/(fwf1+fof1)
F     if (fwf2.lt.1e30) wtp1=100*fwf2/(fwf2+fof2)
F     if (fwf3.lt.1e30) wtp3=100*fwf3/(fwf3+fof3)
F     write(nrpt,'(/''Free-water wt% of liquid after: ''
F    &             /''   HTXG: '',f6.2,
F    &             /''   HTXO: '',f6.2,
F    &             /''   RUGS: '',f6.2)')
F    &      wtp1,wtp2,wtp3

F     write(nrpt,'(/''% flow exiting reactor as vapor:'',f5.1)')
F    &    100.0*flrg/(flrx+flrg)

F     write(nrpt,'(/''Injection & Production''
F    &   /3x,''API gravity: '',2f6.1
F    &   /3x,''Ratio oil flow by weight: '',f6.3)')
F    &    apixi,apixo,(foxo-chrx-wtpx)/(foxi-wtix)

F     write(nrpt,'(/''Reaction products''
F    &   /3x,''Weight fraction to coke: '',f6.3
F    &   /3x,''Weight fraction to P-OIL:'',f6.3)')
F    &    chrx/foxi,poilx/foxi

;  convert from kgmol/s to MW
F     hch4 =gch4 *118215*2324/1.0e6
F     hc2h6=gc2h6*207015*2324/1.0e6
F     hc3h8=gc3h8*294705*2324/1.0e6
F     hhhc =ghhc *381215*2324/1.0e6
F     hpoil=gpoil*468975*2324/1.0e6
F     hh2  =gh2  * 37906*2324/1.0e6
F     write(nrpt,'(/''Combustion energy in FUEL-G (HHV)''
F    &             /3x,''          mol/s    MW''
F    &             /3x,''CH4   '',f6.2,4x,f6.2
F    &             /3x,''C2H6  '',f6.2,4x,f6.2
F    &             /3x,''C3H8  '',f6.2,4x,f6.2
F    &             /3x,''HHC   '',f6.2,4x,f6.2
F    &             /3x,''P-OIL '',f6.2,4x,f6.2
F    &             /3x,''H2    '',f6.2,4x,f6.2
F    &             /3x,''----- '','' -----'',4x,'' -----''
F    &             /3x,''SUM   '',f6.2,4x,f6.2)')
F    &  gch4*1000,hch4,gc2h6*1000,hc2h6,gc3h8*1000,hc3h8,ghhc*1000,
F    &  hhhc,gpoil*1000,hpoil,gh2*1000,hh2,
F    &  (gch4+gc2h6+gc3h8+gc4h10+gpoil+gh2)*1000,
F    &  hch4+hc2h6+hc3h8+hc4h10+hpoil+hh2

F     write(nrpt,'(/''Primary heater (MW):'',f6.2)') -htrqx/1.0e6

F     write(nrpt,'(/''% P-OIL in FUEL-G:'',f5.1)')
F    &         100*gpoil/(opoil+gpoil)
```

```
F     write(nrpt,'(/''Viscosity estimates:''
F    &            /''  Stream   Viscosity (1.0e-3 Pa-s)'')')
F     write(nrpt,'( ''  HTX-O  '',f8.1)') v1*1000

F     write(nrpt,'(/
F    &   ''Required increase in oil value per unit of product'')')
F     write(nrpt,'(3x,''Basis, zero  cost oil in: '',
F    &   f5.2,'' $/Mg'',3x,''('',f5.2,'' $/BBL)'')')
F    &   cost*1000,cost*rhoo*0.159
;  assumed raw oil value, water free basis,  <val> ($/bbl)
;  & compute price increase per unit output (added cost to make up for
;  oil loss.
F     val=10
;  value on per kg basis
F     valw=val/(0.159*rhoi)
F     fprof=foxo*cost+(foxi-foxo)*valw
F     fprof=fprof/foxo
F     write(nprt,'(''foxi,foxo,valw,rhoo'',4f10.2)')foxi,foxo,valw,rhoo
F     write(nrpt,'(3x,''Basis, '',f4.1,'' $/BBL oil in: '',
F    &   f5.2,'' $/Mg'',3x,''('',f5.2,'' $/BBL)'')')
F    &   val,fprof*1000,fprof*rhoo*0.159
F

F     write(nrpt,'
F    &  (/''====================================='',
F    &    ''=====================================''/)')
   EXECUTE LAST


;##############################################################################
;
;  COSTING INFORMATION
;
;##############################################################################


;==============================================================================
;  Report & Control Options
;==============================================================================

CBLOCK-REPORT  NOSORT

ECONOMIC-REPORT NOCASHFLOW   ; supress cashflow & retrun table

;COSTING-OPTION  OPER-COST
COSTING-OPTION  PROFIT

PROFITABILITY
    ANALYSIS MODE=PROD-PRICE  IRR=0.10
;   ANALYSIS MODE=IRR
    ECONOMIC-LIFE  YEARS=20

;==============================================================================
;  Cost Information
;==============================================================================

PROJECT-DATES
   START  JUNE  1994

; set escalation for all indexes (beyond March 95?)
COST-INDEX
   EQUIPMENT  ESCALATION=0.03
```

```
   FABRICATED ESCALATION=0.03
   PUMP-COMPR ESCALATION=0.03
   LABOR      ESCALATION=0.03
   COMMODITY  ESCALATION=0.03
   BUILDING   ESCALATION=0.03
   CHEMICAL   ESCALATION=0.03
   FUEL       ESCALATION=0.05  ; based on ENERGETICS
   OPER-MAT   ESCALATION=0.03
   OPER-LAB   ESCALATION=0.03
   PLANT      ESCALATION=0.03

;  Remove cost associated with new site
SITE-COSTS
   SITE-DEVELOPMENT MAT-FAC=0.0 LAB-FAC=0.0
SERVICE-ITEMS
   SERVICE-BUILDINGS MAT-FAC=0.0 LAB-FAC=0.0
PLANT
   LAND FACTOR=0.0

LABOR-COSTS
   WAGES  RATE=20 MONTH=JUNE YEAR=1995  ; Unload Const. Labor $/hr   Data #103

CONTINGENCY
   PROCESS-BASIS      FACTOR=0.05
   PROJECT-DEFINITION FACTOR=0.1

OPERATING-COST
   OPERATING-LA NOPER=1  RATE=16 MONTH=JUNE YEAR=1991 ; unloaded cost $/hr   Data #103
   OTHER-LABOR MAINTENANCE=0.0 ; maintenance in supplies
   SUPPLIES MAINTENANCE=0.04 ; all maint including labor
   GENERAL-WORKS  GEN-ADMIN=0.0  TAX=0.0 INSURANCE=0.0 ; set to zero
   ADDITIONAL FACTOR=0.04  ; used to incorporate prop tax, ins & general overhead

UTILITY POWER ELECTRICITY
    DESCRIPTION  'Electricity'
    SOURCE       PURCHASED
    COST         PRICE=0.05 [$/kwhr] MONTH=JANUARY YEAR=1994  ;  Data #102

UTILITY NGAS  GAS
    DESCRIPTION  'Natural Gas'
    SOURCE       PURCHASED
    COST         PRICE=2.15e-9 [$/J]  MONTH=JANUARY YEAR=1994 ;  Data #102

UTILITY COOLW WATER
    DESCRIPTION  'Chilled water'
    PARAM TIN=60[F] TOUT=140[F] COMPONENT=H2O
    SOURCE       PURCHASED
    PROPERTIES   SYSOP12
    COST         PRICE=0.04e-3 [$/lb] MONTH=JUNE YEAR=1991 ;     Data #102

PRODUCT OIL
    REFERENCE  STREAM=PRD-O


UNIT CB1  ; For now lump everything together
   CBLOCKS C-PSU C-PRH C-IPMP C-CPMP C-FPMP C-IHTR C-SSU C-PUMP &
           C-HTXG C-HTXO C-HTR C-RU C-FOIL C-COIL C-CONDH C-CSU &
           C-CONDL C-WCU C-WSU


;=============================================================================
;  Cost Blocks
;=============================================================================
```

```
CBLOCK C-PSU TANK
    DESCRIPTION 'Initial water separation'
    REFERENCE   INLET STREAM=FEED
    SIZING-DATA RETEN-TIME=60 [min]   ;                          Data #101

CBLOCK C-PRH V-VESSEL
    DESCRIPTION 'Capture reactor vapors'
    REFERENCE   BLOCK=PRH
    SIZING-DATA RETEN-TIME=5 [min] & ;   only place holder true retention
                                     ;   set in FCPRH
                CORROSION=0.125 <in> ;                           Data #104

FORTRAN FCPRH
F     common /usr3/ capfac
F     character*1 tdum
; Compute size of preheat vessel
    DEFINE flowgx  STREAM-VAR STREAM=PGSU-G VARIABLE=MASS-FLOW
    DEFINE flwlx1  STREAM-PROP STREAM=PRH-E  PROPERTY=O2
    DEFINE flwlx2  STREAM-PROP STREAM=PRH-E  PROPERTY=O3
    DEFINE denlx   STREAM-PROP STREAM=PRH-E  PROPERTY=PRHOL
    DEFINE dengx   STREAM-PROP STREAM=PRH-E  PROPERTY=PRHOV
;
    DEFINE volc CBLOCK-VAR CBLOCK=C-PRH  SENTENCE=SIZING-DATA &
       VARIABLE=VOL
    DEFINE diamc CBLOCK-VAR CBLOCK=C-PRH  SENTENCE=SIZING-DATA &
       VARIABLE=DIAM
    DEFINE nequip CBLOCK-VAR  CBLOCK=C-PRH  SENTENCE=COSTING-DATA &
       VARIABLE=NEQUIP
;  total liquid flow
F     flowlx=capfac*(flwlx1+flwlx2)
;  set liquid volume in ft**3 based on rest minute residence time  Data #101
F     rest=5.0
F     vliq=rest*60.0*flowlx/denlx
;  maximum vessel volume                                          Data #109
F     vmax=16.0
;  minimum diameter (m)                                           Data #109
F     dmin=4.0*0.3048
F     write(nrpt,'(''C-PRH'')')
;  use vertical only for good contact
F     call sep(nrpt,'V','S','Y','N',flowgx,flowlx,vliq,vmax,denlx,
F    &   dengx,dum,dum,dum,dum,dmin,number,vol,xlen,diam,tdum)
F     volc=vol
F     nequip=number
F     diamc=diam
    READ-VARS flowgx denlx dengx


CBLOCK C-IPMP PUMP
    DESCRIPTION 'Pump initial emulsion to desired pressure'
    REFERENCE   BLOCK=IPMP
    COSTING-DATA TYPE=SS-ANSI NSTANDBY=0 MATERIAL=SS316    ; Data #110
    UTILITY       ELEC=POWER

CBLOCK C-CPMP PUMP
    DESCRIPTION 'Pump main product stream to desired pressure'
    REFERENCE   BLOCK=CPMP
    COSTING-DATA TYPE=SS-ANSI NSTANDBY=0 MATERIAL=CAST-ST    ; Data #110
    UTILITY       ELEC=POWER

CBLOCK C-FPMP PUMP
    DESCRIPTION 'Pump condenser oil to desired pressure'
    REFERENCE   BLOCK=FPMP
    COSTING-DATA TYPE=SS-ANSI NSTANDBY=0 MATERIAL=CAST-ST  ; Data #110
    UTILITY       ELEC=POWER
```

```
CBLOCK C-IHTR FIRED-HEATER
    DESCRIPTION 'Set temperature for water split prior to reaction'
    REFERENCE  BLOCK=IHTR
    COSTING-DATA MATERIAL=SS316
    UTILITY     GAS =NGAS
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
FORTRAN FCIHTR ;  sets fired heater costs
F     common /usr3/ capfac
    DEFINE pres  STREAM-VAR STREAM=PGSU-E VARIABLE=PRES
    DEFINE qx    INFO-VAR   STREAM=IHTR-Q    INFO=HEAT VARIABLE=DUTY
    DEFINE costx CBLOCK-VAR CBLOCK=C-IHTR  SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
;  convert to psi & MW
F     prs=pres/1000.0/6.695
F     qmw=capfac*(-qx/1.0e6)
F     write(nrpt,'(''C-IHTR'')')
F     if (qmw .gt. 0.01) then
F        call usrhtr(nrpt,'SS','BM',qmw,prs,costx)
F     else
F        costx=0.0
F     endif
    EXECUTE BEFORE C-IHTR

CBLOCK C-SSU H-VESSEL
    DESCRIPTION 'Secondary water separation'
    REFERENCE   BLOCK=SFW
    SIZING-DATA RETEN-TIME=5 [min] & ;(assume 60% liquid fill)  Data #101
              CORROSION=0.125 <in>                          ;  Data #104
    COSTING-DATA ADJUST=1.1 ; 10% for internals                 Data #111

CBLOCK C-PUMP PUMP  ; include electric motor
    DESCRIPTION 'Pump primary emulsion to desired pressure'
    REFERENCE   BLOCK=PUMP
    COSTING-DATA TYPE=HS-MS NSTANDBY=0 MATERIAL=SS316
    UTILITY    ELEC=POWER
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
FORTRAN FC-PUMP ;  sets pump+motor cost & power use
F     common /usr3/ capfac
F     character type*20
    DEFINE flw   STREAM-VAR STREAM=PUMP-E VARIABLE=MASS-FLOW
    DEFINE pin   STREAM-VAR STREAM=MXAA-E VARIABLE=PRES
    DEFINE pout  STREAM-VAR STREAM=PUMP-E VARIABLE=PRES
    DEFINE wrk   INFO-VAR   STREAM=PUMP-WK  INFO=WORK VARIABLE=POWER
    DEFINE fpow  BLOCK-VAR  BLOCK=PUMP SENTENCE=RESULTS VARIABLE=FLUID-POWER
    DEFINE epow  BLOCK-VAR  BLOCK=PUMP SENTENCE=RESULTS VARIABLE=ELEC-POWER
    DEFINE cost CBLOCK-VAR CBLOCK=C-PUMP SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
    DEFINE pwr  CBLOCK-VAR CBLOCK=C-PUMP SENTENCE=UTILITY   &
      VARIABLE=ELEC-RATE  ; used to set power in kW
F     flow=capfac*flw
;  convert to MPa
F     head=(pout-pin)/1.0e6
F     write(nrpt,'(''C-PUMP'')')
;  overall efficiency for sizing electric motor
F     eff=fpow/epow
F     call usrpml(nrpt,'SS','E',flow,head,eff,cost,dum,power,type)
;  uses work to set power for pump (this incudes efficiency computed by ASPEN)
F     pwr=wrk
    EXECUTE BEFORE C-PUMP


CBLOCK C-HTXG HEATX
    DESCRIPTION 'Heat recovery heat exchanger (gas/oil)'
```

```
    REFERENCE  SHELL  BLOCK=HTXG  STREAM=GASR-G2
    REFERENCE  TUBE   BLOCK=HTXG  STREAM=PUMP-E
    SIZING-DATA  NPASS-TUBE=1  ; avoids warning
    COSTING-DATA MAT-SHELL=SS316  MAT-TUBE=SS316
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
FORTRAN FCHTXG ; sets htx cost
F     common /usr3/ capfac
    DEFINE pres  STREAM-VAR STREAM=PUMP-E VARIABLE=PRES
    DEFINE area  BLOCK-VAR  BLOCK=HTXG  SENTENCE=RESULTS  &
              VARIABLE=AREA-CALC
    DEFINE cost CBLOCK-VAR CBLOCK=C-HTXG SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
F     areax=capfac*area
;  convert to MPa
F     prs=pres/1.0e6
F     write(nrpt,'(''C-HTXG'')')
F     call usrht1(nrpt,'SS316','UT',areax,prs,cost)
    EXECUTE BEFORE C-HTXG

CBLOCK C-HTXO HEATX
    DESCRIPTION 'Heat recovery heat exchanger (oil)'
    REFERENCE  SHELL  BLOCK=HTXO  STREAM=HTXG-E
    REFERENCE  TUBE   BLOCK=HTXO  STREAM=MXB-O
    SIZING-DATA  NPASS-TUBE=1  ; avoids warning
    COSTING-DATA MAT-SHELL=SS316  MAT-TUBE=SS316
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
FORTRAN FCHTXO ; sets htx cost
F     common /usr3/ capfac
    DEFINE pres  STREAM-VAR STREAM=HTXG-E VARIABLE=PRES
    DEFINE area  BLOCK-VAR  BLOCK=HTXO  SENTENCE=RESULTS  &
              VARIABLE=AREA-CALC
    DEFINE cost CBLOCK-VAR CBLOCK=C-HTXO SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
F     areax=capfac*area
;  pressure to MPa
F     prs=pres/1.0e6
F     write(nrpt,'(''C-HTXO'')')
F     call usrht1(nrpt,'SS316','UT',areax,prs,cost)
    EXECUTE BEFORE C-HTXO


CBLOCK C-HTR FIRED-HEATER
    DESCRIPTION 'Trim temperature for reactor/separator'
    REFERENCE  BLOCK=HTR
    COSTING-DATA MATERIAL=CS
    UTILITY     GAS =NGAS
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
FORTRAN FCHTR ;  sets fired heater costs
F     common /usr3/ capfac
    DEFINE pres  STREAM-VAR STREAM=PUMP-E VARIABLE=PRES
    DEFINE q     INFO-VAR   STREAM=HTR-Q    INFO=HEAT VARIABLE=DUTY
    DEFINE cost CBLOCK-VAR CBLOCK=C-HTR  SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
;  convert to psi & MW
F     prs=pres/1000.0/6.895
F     qmw=capfac*(-q/1.0e6)
;  efficiency                                                  Data #108
F     eff=0.75
F     qmw=qmw/eff
F     write(nrpt,'(''C-HTR'')')
F     call usrhtr(nrpt,'CS','BM',qmw,prs,cost)
    EXECUTE BEFORE C-HTR

CBLOCK C-RU H-VESSEL
```

```
    DESCRIPTION 'Reactor and high temperature separation'
    SIZING-DATA VOL=100 <bbl>  DIAM=5 <ft> & ; size, P & T set in FRU
              PRES=2000 <psi> TEMP=400 &
              CORROSION=0.125 <in>                              ;  Data #104
    COSTING-DATA NEQUIP=1    ; number of vessels set in FRU

CBLOCK C-FOIL H-VESSEL
    DESCRIPTION 'Flash to set final water'
    REFERENCE  LIQUID STREAM=FOIL-O
    ; require volume set in FCFOIL
    SIZING-DATA VOL=100 [bbl]      &
              CORROSION=0.125 <in>                              ;  Data #104

FORTRAN FCFOIL
F     common /usr3/ capfac
F     character*1 tdum
; Compute size of knowkout vessel for FOIL
    DEFINE flowgx  STREAM-VAR STREAM=FOIL-G VARIABLE=MASS-FLOW
    DEFINE flowlx  STREAM-VAR STREAM=FOIL-O VARIABLE=MASS-FLOW
    DEFINE denlx   STREAM-PROP STREAM=FOIL-O PROPERTY=PRHOL;
    DEFINE dengx   STREAM-PROP STREAM=FOIL-G PROPERTY=PRHOV;
;
    DEFINE volc CBLOCK-VAR CBLOCK=C-FOIL SENTENCE=SIZING-DATA &
       VARIABLE=VOL
    DEFINE diamc CBLOCK-VAR CBLOCK=C-FOIL SENTENCE=SIZING-DATA &
       VARIABLE=DIAM
    DEFINE nequip CBLOCK-VAR  CBLOCK=C-FOIL SENTENCE=COSTING-DATA &
       VARIABLE=NEQUIP
F     flwg=capfac*flowgx
F     flwl=capfac*flowlx
;  set liquid volume in m**3 based on 5 min residence time
F     vliq=5.0*60.0*flwl/denlx
;  maximum vessel volume                                          Data #105
F     vmax=16.0
;  minimum L/D ratio                                              Data #105
F     xldr=3.0
;  minimum diameter (m)                                           Data #105
F     dmin=2.0*0.3048
;  minimum void height if horizontal (m)                          Data #105
F     hmin=1.0*0.3048
F     write(nrpt,'(''C-FOIL'')')
F     call sep(nrpt,' ','S','Y','N',flwg,flwl,vliq,vmax,denlx,
F    &   dengx,dum,dum,xldr,hmin,dmin,number,vol,xlen,diam,tdum)
F     volc=vol
F     nequip=number
F     diamc=diam
    READ-VARS flowlx denlx dengx
    WRITE-VARS volc diamc nequip

CBLOCK C-COIL AIRCOOL
    DESCRIPTION 'Cool final oil product'
    REFERENCE  BLOCK=COIL
    SIZING-DATA  U=50 <BTU/HR-SQFT-R>  ; Bare surface (surfaces are finned)
                                       ; Based on oil/air system
    COSTING-DATA  CLASS=ORGANIC & ; ORGANIC doen't seem to matter
                MATERIAL=CS

CBLOCK C-CONDH HEATX
    DESCRIPTION 'Gas condenser - high pressure'
;  Note sizing data is overridden by user cost, only dummies
    REFERENCE SHELL BLOCK=CONDH
    REFERENCE TUBE UTILITY=COOLW
    SIZING-DATA  NPASS-TUBE=1 U=150 <BTU/HR-SQFT-R>
    COSTING-DATA MAT-SHELL=CS     MAT-TUBE=SS316
```

```
    USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT

FORTRAN FCCONDH ; sets htx cost
F     common /usr3/ capfac
    DEFINE t1     STREAM-VAR STREAM=HTXG-G  VARIABLE=TEMP
    DEFINE t2     STREAM-VAR STREAM=CONDH-O VARIABLE=TEMP
    DEFINE pres   STREAM-VAR STREAM=CONDH-O VARIABLE=PRES
    DEFINE qcool  INFO-VAR  STREAM=CONDH-Q  INFO=HEAT  VARIABLE=DUTY
    DEFINE cost CBLOCK-VAR CBLOCK=C-CONDH SENTENCE=USER-COST  &
      VARIABLE=PURCH-COST  ; used to set cost
;  ucool, based on computed film coefs liq/liq (W/sqm-C)
F     ucool=150*5.674
;  cool side set by assumed cooling water temps (C)          Data #107
F     dta=t1-60
F     dtb=t2-15.6
F     dtlm=(dta-dtb)/log(dta/dtb)
F     area=capfac*qcool/ucool/dtlm
F     write(nrpt,'(''C-CONDH'')')
F     write(nrpt,'(''dta,dtb,H-dtlm,qcool,area='',2f6.0,3(1pe12.3))')
F    &            dta,dtb,dtlm,qcool,area
;  convert pres to MPa
F     prs=pres/1.0e6
F     call usrht1(nrpt,'SS316','UT',area,prs,cost)
    EXECUTE BEFORE C-CONDH

CBLOCK C-CSU H-VESSEL
    DESCRIPTION 'Liquid knock out for high pressure condenser'
    REFERENCE LIQUID STREAM=CSU-O
    SIZING-DATA VOL=100 <bbl>  DIAM=5 <ft> & ; size set in FCSU
              CORROSION=0.125 <in>                              ;  DATA #104

FORTRAN FCSU
F     common /usr3/ capfac
F     character*1 tdum
; Compute size of knowkout vessel for CSU
    DEFINE flowgx  STREAM-VAR STREAM=CSU-G  VARIABLE=MASS-FLOW
    DEFINE flwlx1  STREAM-VAR STREAM=CSU-O  VARIABLE=MASS-FLOW
    DEFINE flwlx2  STREAM-VAR STREAM=CSU-FW VARIABLE=MASS-FLOW
    DEFINE denlx   STREAM-PROP STREAM=CSU-O  PROPERTY=PRHOL;
    DEFINE dengx   STREAM-PROP STREAM=CSU-G  PROPERTY=PRHOV;
;
    DEFINE volc CBLOCK-VAR CBLOCK=C-CSU SENTENCE=SIZING-DATA &
       VARIABLE=VOL
    DEFINE diamc CBLOCK-VAR CBLOCK=C-CSU SENTENCE=SIZING-DATA &
       VARIABLE=DIAM
    DEFINE nequip CBLOCK-VAR  CBLOCK=C-CSU SENTENCE=COSTING-DATA &
       VARIABLE=NEQUIP
;  set liquid volume in ft**3 based on 5 min residence time
F     flwl=capfac*(flwlx1+flwlx2)
F     flwg=capfac*flowgx
F     vliq=5.0*60.0*flwl/denlx
;  maximum vessel volume                                      Data #106
F     vmax=16.0
;  minimum L/D ratio                                          Data #106
F     xldr=3.0
;  minimum diameter (m)                                       Data #106
F     dmin=2.0*0.3048
;  minimum void height if horizontal (m)                      Data #106
F     hmin=1.0*0.3048
F     write(nrpt,'(''C-CSU'')')
F     call sep(nrpt,' ','S','Y','N',flwg,flwl,vliq,vmax,denlx,dengx,
F    &   dum,dum,xldr,hmin,dmin,number,vol,xlen,diam,tdum)
F     volc=vol
F     nequip=number
```

```
F     diamc=diam
    READ-VARS flwlx1 flwlx2 denlx dengx
    WRITE-VARS volc diamc nequip

CBLOCK C-CONDL HEATX
    DESCRIPTION 'Gas condenser - low pressure'
    REFERENCE SHELL BLOCK=CONDL
    REFERENCE TUBE UTILITY=COOLW
    SIZING-DATA  NPASS-TUBE=1 U=150 <BTU/HR-SQFT-R>
    COSTING-DATA MAT-SHELL=CS     MAT-TUBE=SS316
;  Commented out alternate htx costing
;   USER-COST  PURCH-COST=* INDEX=325 INDEX-TYPE=PLANT
;FORTRAN FCONDL ; sets htx cost
;    DEFINE t1    STREAM-VAR STREAM=MXC-G   VARIABLE=TEMP
;    DEFINE t2    STREAM-VAR STREAM=CONDL-G VARIABLE=TEMP
;    DEFINE pres  STREAM-VAR STREAM=CONDL-G VARIABLE=PRES
;;   DEFINE qcool  INFO-VAR  STREAM=CONDL-Q  INFO=HEAT  VARIABLE=DUTY
;    DEFINE cost CBLOCK-VAR CBLOCK=C-CONDL SENTENCE=USER-COST  &
;      VARIABLE=PURCH-COST  ; used to set cost
;;  pressure to MPa
;F     prs=pres/1.0e6
;;  ucool Walas (W/sqm-C)
;F     ucool=150*5.674
;;  cool side set by assumed cooling water temps              Data #107
;F     dta=t1-60
;F     dtb=t2-15.6
;F     dtlm=(dta-dtb)/log(dta/dtb)
;F     area=qcool/ucool/dtlm
;F     write(nrpt,'(''C-CONDL'')')
;F     write(nrpt,'(''L-dtlm,qcool,area='',3(1pe12.3))')
;F    &           dtlm,qcool,area
;F     call usrht1(nrpt,'SS316','UT',area,prs,cost)
;    EXECUTE BEFORE C-CONDL

CBLOCK C-WCU HEATX
    DESCRIPTION 'Cool water from separator'
    REFERENCE SHELL BLOCK=WCU
    REFERENCE TUBE UTILITY=COOLW
    SIZING-DATA  NPASS-TUBE=1 U=130 <BTU/HR-SQFT-R>  ; Based on film coefs
    COSTING-DATA MAT-SHELL=CS     MAT-TUBE=SS316

CBLOCK C-WSU H-VESSEL
    DESCRIPTION 'Condeser  water/oil separation'
    REFERENCE   BLOCK=MXWSU
    SIZING-DATA RETEN-TIME=5 [min] & ;(assume 60% liquid fill)  Data #101
              CORROSION=0.125 <in>                          ;  Data #104
    COSTING-DATA ADJUST=1.1 ; 10% for internals                 Data #111
```

APPENDIX II

WALAS BASED COST ROUTINES

Below is a listing of the FORTRAN subroutines which are used to obtain cost information for certain process units. These cost estimates are based on information in Walas' book, *Chemical Process Equipment - Selection and Design* (reference 6 in main body). In each case, the output costs are in 1985 dollars. In the "usrht1" routine for costing heat exchangers, the cost correlations have been extended beyond those available in Walas. This extension was done by using selected information from Purohit[II-1].

```
subroutine usrht1(luin,mat,type,areai,presi,cost)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Heat exchanger costs (1985) from Walas.  Shell & tube.
c
c  Limits for: shell & tube: 100-900 psig, 50-12000 sqft.
c                      (cbt extension to 2500 psi)
c              aircoolers 0-7 atm 50-200,000 sqft
c
c     In: luin - (0 no write, >0 wite input & results, <0 write input
c                 & do no calcs)
c         mat  - SS316, SS304 (more available)
c         type - Shell &  tube
c                    FH- fixed head
c                    UT - Utube
c                    KE - Kettle
c                  AC - air cooler
c         area - m**2
c         pres - MPa
c
c  Output: cost - $
c
c-------------------------------------------------------------
c
c  Rev. 1.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      implicit none
c  formal params
      character*(*) mat,type
      integer luin
      real*8 areai, presi, cost
c  local params
      integer lu
      real*8 area, pres, log_area,c
      real*8 fd, fm, fp, cb

c convert input to ft**2 & psi
      area=areai/0.0929
      pres=presi*1.0e3/6.895

      cost=-1
      lu=abs(luin)
      if (lu.ne.0) then
        write(lu,'(''+=+= usrhtx +=+=='')')
```

```
        write(lu,'(''Material '',a)') mat
        write(lu,'(''Type     '',a)') type
        write(lu,'(''area (sqm) '',f10.4)') areai
        write(lu,'(''pressure (MPa) '',f10.4)') presi
        if (luin.lt.0) return
      endif

c  Air cooler section
      if (type(1:2) .eq. 'AC' .or. type(1:2) .eq. 'ac') then
        c=24.6*(area/1000)**0.4
        cost=c*1.0e3
        return
      endif


c  Shell & tupe section

      if (area .le. 0.0) return
      log_area=log(area)

c  cb ------------------------------------------------------------
      cb=exp(8.821-0.30863*log_area+0.0681*(log_area)**2)

c  fd ------------------------------------------------------------
      if (type(1:2).eq.'FH' .or. type(1:2).eq.'fh') then
c  fixed head
        fd=exp(-1.1156+0.0906*log_area)
      elseif (type(1:2).eq.'UT' .or. type(1:2).eq.'ut') then
c  U-tube
        fd=exp(-0.9816+0.083*log_area)
      else
c  U-tube
        fd=exp(-0.9816+0.0830*log_area)
      endif

c  fp ------------------------------------------------------------
c  Pressure 100-300 psig
      if (pres .le.300) then
        fp=0.7771+0.04981*log_area
c  Pressure 300-600 psig
      elseif (pres .le. 600) then
         fp=1.0305+0.07140*log_area
c  Pressure 600-900 psig
      elseif (pres .le. 900) then
        fp=1.1400+0.12088*log_area
c  Extension by cbt using rough data from Che Eng '83
c  Pressure 900-1500 psig
      elseif (pres .le.1500) then
        fp=1.2+0.120*log_area
c  Pressure 1500-2000 psig
      elseif (pres .le.2000) then
        fp=1.2+0.19*log_area
c  Pressure 2000-2500 psig
      elseif (pres .le.2500) then
        fp=1.2+0.47*log_area
      else
        fp=1.2+0.47*log_area
      endif

c  fm ------------------------------------------------------------
      if (mat(1:5).eq.'SS316' .or. type(1:2).eq.'ss316') then
c  Stainless 316
        fm=0.8603+.23296*log_area
      elseif (mat(1:5).eq.'SS304' .or. type(1:2).eq.'ss304') then
```

```
c  Stainless 304
         fm=0.8193+.15984*log_area
       else
c  Stainless 304
         fm=0.8193+.15984*log_area
       endif

       cost=fd*fm*fp*cb

       if (lu.ne.0) then
         write(lu,'(''Cost (1985 $) '',f10.0)') cost
         if (area.gt.0.0)
      &      write(lu,'(''Cost ($/ft) '',f10.4)') cost/area
       endif

       return
       end


       subroutine usrmtr(luin,type,power, cost)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Cost (1985) of electric motors from Walas.
c
c  In:
c      luin - (0 no write, >0 wite input & results, <0 write input
c               & do no calcs)
c   type - type  EN3600 enclosed 3600 rpm
c   power - MW
c
c  Out:
c   cost - capital cost $
c
c-----------------------------------------------------------------------
c
c  Rev. 1.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       implicit none
c  formal params
       character type*(*)
       integer luin
       real*8 power, cost
c  local params
       integer n,lu
       real*8 hp, a1,a2,a3  ,c, fac
       real*8 hpmax, hp_unit


       cost=-1
       hp=power*1.0e6/746  ! horsepower

       lu=abs(luin)
       if (lu.ne.0) then
         write(lu,'(''+=+= usrmtr +=+='')')
         write(lu,'(''Type      '',a)') type
         write(lu,'(''Power (MW) '',f10.4)') power
         write(lu,'(''Power (hp) '',f10.4)') hp
         if (luin.lt.0) return
       endif

       if (type(1:6) .eq. 'EN3600') then
         hpmax=400.0
         n=hp/hpmax
```

```
        if (mod(hp,hpmax).gt.0.01*hpmax) n=n+1
        n=max(1,n)
        hp_unit=hp/n
        fac=log(hp_unit)

        if (hp_unit.le.7.5) then
           a1=5.1058
           a2=0.03316
           a3=0.15374
        elseif (hp_unit.le.250) then
           a1=3.8544
           a2=0.83311
           a3=0.02399
        elseif (hp_unit.le.400) then
           a1=5.3182
           a2=1.0847
           a3=-0.05695
         else
           return
         endif

        c=1.2*exp(a1+a2*fac+a3*fac**2)
        cost=n*c

      endif

      if (lu.ne.0) then
        write(lu,'(''Cost ($) '',f10.0)') cost
        write(lu,'(''Number of motors '',f10.0)') n
      endif

      return
      end


      subroutine usrpm1(luin,mat,ptype,flow,head,eff,cost,
     &                  motorcost,power,type_used)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Liquid pump capital costs (1985) from Walas.
c  Limits 50-5000 gpm & 50-3200 ft head.
c
c     In: luin - (0 no write, >0 write input & results, <0 write input
c                & do no calcs)
c        mat  - Material  SS-stainless  CS-carbon steel
c        type - If ' ' then auto selects pump. Otherwise
c               A or a -   One-stage, 1750 rpm, VSC  50-3500  50-200''
c               B or b -   One-stage, 1750 rpm, HSC 250-5000  50-500''
c               C or c -   One-stage, 3550 rpm, HSC 100-1500 100-450''
c               D or d -   Two-stage, 3550 rpm, HSC 50-1100  300-1100''
c               E or e - Multi-stage, 3550 rpm, HSC 100-1500 650-3200''
c
c        flow - kg/s  (density assumed to be 1000 kg/m**3)
c        head - MPa
c        eff - overall efficiency fraction (pump*electric)
c
c    Out: cost - total cost pump + motor ($)
c         motorcost - cost of motor ($)
c         power - power used in sizing motor (Mw)
c         type_used - type pump used
c
c-----------------------------------------------------------------
c
c  Rev. 1.0
```

```
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      implicit none
c  formal params
      character type_used*(*),mat*(*),ptype*(*)
      integer luin
      real*8 flow,head,eff,cost,power
c  local params
      integer lu
      real*8 q, hd,  motorcost,hp,s
      real*8 ft, fm, cb, b1, b2, b3, fac, quse

      lu=abs(luin)
      if (lu.ne.0) then
        write(lu,'(''+=+= usrpmp +=+='')')
        write(lu,'(''Material '',a)') mat
        write(lu,'(''flow (kg/s) '',f10.4)') flow
        write(lu,'(''head (MPa) '',f10.4)') head
        write(lu,'(''Overall efficiency '',f5.2)') eff
        if (luin .lt. 0) return
      endif

      cost=-1

      q=flow*60.0/3.785 ! gpm
      hd=head*1.0e6/2989.0  ! ft water

      if     (mat(1:2) .eq. 'SS' .or. mat(1:2) .eq. 'ss') then
         fm=2.0
      elseif (mat(1:2) .eq. 'CS' .or. mat(1:2) .eq. 'cs') then
         fm=1.35
      else
        fm=1.35
      endif

c  ft ----------------------------------------------------------
      if (ptype(1:1) .eq. ' ') then
        if(q.ge.50 .and. q.le.3500.and. hd.ge.50 .and. hd.le.200)then
           ptype='A'
        elseif(q.ge.250.and. q.le.5000.and. hd.ge.50.and.hd.le.500)then
           ptype='B'
        elseif(q.ge.100.and. q.le.1500.and.hd.ge.100.and.hd.le.450)then
           ptype='C'
         elseif(q.ge.50.and.q.le.1100.and. hd.ge.300.and.hd.le.1100)then
           ptype='D'
         elseif(q.ge.100.and.q.le.1500.and.hd.ge.650.and.hd.le.3200)then
           ptype='E'
        else
           ptype='E'
        endif
      endif

c  don't allow too low a flow to be used
      if   (ptype .eq. 'A' .or. ptype .eq. 'a')then
        quse=max(q,50)
        b1=5.1029
        b2=-1.2217
        b3=0.0771
        type_used='one-stage 1750 rpm, VSC'
      elseif(ptype .eq. 'B' .or. ptype .eq. 'b')then
        quse=max(q,250)
        b1=2.029
        b2=-0.2371
        b3=0.0102
```

```
        type_used='one-stage 1750 rpm, HSC'
      elseif(ptype .eq. 'C' .or. ptype .eq. 'd')then
        quse=max(q,100)
        b1=0.0632
        b2=0.2744
        b3=-0.0253
        type_used='one-stage 3500 rpm, HSC'
      elseif(ptype .eq. 'D' .or. ptype .eq. 'd')then
        quse=max(q,50)
        b1=13.7321
        b2=-2.8304
        b3=0.1542
        type_used='two-stage 3500 rpm, HSC'
      else   ! ltype='E'
        quse=max(q,100)
         b1=9.8849
        b2=-1.6164
        b3=0.0834
        type_used='multi-stage 3500 rpm, HSC'
      endif

      fac=log(quse*sqrt(hd))

c  cb ------------------------------------------------------------
      cb=1.55*exp(8.833-0.6019*fac+0.0519*fac**2)

      ft=exp(b1+b2*fac+b3*fac**2)

      cost=fm*ft*cb

      s=1.0  ! specific gravity
      hp=8.33*hd*s/33000 *q/eff  !Horsepower
      power=hp*746/1.0e6    ! Mw
      call usrmtr (luin,'EN3600',power, motorcost)

      if (lu.ne.0) then
        write(lu,'(''Type      '',a)') type_used
        write(lu,'(''Pump cost  ($)  '',f10.4)') cost
        write(lu,'(''Motor cost ($)  '',f10.4)') motorcost
        write(lu,'(''Power (hp) '',f10.4)') hp
        write(lu,'(''Power (Mw) '',f10.4)') power
      endif

      cost=cost+motorcost

      return
      end


      subroutine usrhtr (luin,mat,type,q,pres,cost)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Fired heater costs (1985) from Walas.
c
c  Limits Box 20-200 Mbtu/hr  (million Btu/hr), 0-3000 psi
c         Cylinder 2-30 MBtu/hr  0-1500
c
c     In: luin - (0 no write, >0 wite input & results, <0 write input
c                & do no calcs)
c     In: mat- CS-acrbon steel, SS-stianless
c        type- BL - Box low temp (<300C) ,
c              BM - Box med temp (300-500C),
c              BH - Box high temp (>500C)
c              Note: temps are approx.
```

```
c              CL - Cylindrical type
c           q - heat requirement  MW
c         pres - pressure (psi)
c
c     Out:
c       cost - negative is error, positive cost in $
c
c-----------------------------------------------------------------
c
c  Rev. 1.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      implicit none
c  formal params
      character*(*) mat,type
      integer luin
      real*8  q,pres,cost
c  local params
      integer lu
      real*8 c, millbtu_hr,k,fd,fp

      millbtu_hr=q*1.0e6/0.2929/1.0e6
      lu=abs(luin)
      if (lu.ne.0) then
        write(lu,'(''+=+= usrhtr +=+='')')
        write(lu,'(''Material '',a)') mat
        write(lu,'(''Type     '',a)') type
        write(lu,'(''q (MW) '',f10.4)') q
        write(lu,'(''q (million BTU/hr) '',f10.4)') millbtu_hr
        write(lu,'(''pressure (psig) '',f10.4)') pres
        if (luin.lt.0) return
      endif

c  Air cooler section

      if (type(1:1) .eq. 'B' .or. type(1:1) .eq. 'b') then
        if (mat(1:2).eq.'CS' .or. mat(1:2).eq.'cs') then
           k=25.5
        elseif (mat(1:2).eq.'SS' .or. mat(1:2).eq.'ss') then
           k=45.0
         else
           k=25.5
         endif

        if (type(2:2).eq. 'L' .or. type(2:2).eq.'l') then
           fd=0.0
        elseif (type(2:2).eq. 'M' .or. type(2:2).eq.'m') then
           fd=0.10
        elseif (type(2:2).eq. 'H' .or. type(2:2).eq.'h') then
           fd=0350
         else
           fd=0.0
         endif

        if (pres .le. 500.0) then
           fp=0.0
        elseif (pres .le. 1000.0) then
           fp=0.1
        elseif (pres .le. 1500.0) then
           fp=0.15
        elseif (pres .le. 2000.0) then
           fp=0.25
        elseif (pres .le. 2500.0) then
           fp=0.40
```

```
    elseif (pres .le. 3000.0) then
       fp=0.60
     else
       fp=0.6
     endif

    c=k*(1.0+fd+fp)*millbtu_hr**0.86
    cost=c*1.0e3

  else  ! this is Cylindrical
    if (mat(1:2).eq.'CS' .or. mat(1:2).eq.'cs') then
       k=27.3
    elseif (mat(1:2).eq.'SS' .or. mat(1:2).eq.'ss') then
       k=42.0
     else
       k=27.3
     endif

    fd=0  ! cylindrical

    if (pres .le. 500.0) then
       fp=0.0
    elseif (pres .le. 1000.0) then
       fp=0.15
    elseif (pres .le. 1500.0) then
       fp=0.20
    else
       fp=0.20
     endif

    c=k*(1.0+fd+fp)*millbtu_hr**0.86
    cost=c*1.0e3
  endif

  if (lu.ne.0) then
    write(lu,'(''Cost ($) '',f10.0)') cost
  endif

  return
  end
```

REFERENCES

II-1. G. P. Purohit, "Estimating costs of shell-and-tube heat exchangers," *Chemical Engineering, Vol. 90, No. 17,* pp, 54-67 (August 1983).

APPENDIX III

VESSEL SIZING ROUTINES, THE WATKINS' ROUTINES

Below is a listing of the FORTRAN subroutines which are used to obtain sizes for flash vessels. The routines are based on information from Watkins, reference 7. Calls to this set of routines are intended to be made through the "sep" routine, although direct calls to other subroutines can be made if needed.

```
subroutine sep(luin,config,units,mult,vtype,gas,liq,vliq,vmax,
     &    denl,deng,dp,visc,ldr,hmin,dmin,number,volume,len,diam,type)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Compute the size and number of vessels need in a flash drum
c  operation based on simple correlations in Walas for horizontal
c  and vertical drums.  Correlation for max velocity can be from
c  simple stokes flow 'S' or Newton based correlations from Watson
c  "Hydrocarbon Processing", 1967, 'N'.
c
c  Input: luin - write computational info to report file if nrpt
c                is ASPEN's lu lu.  Set to zero for no output.
c         config - Configuration. 'V' vertical only, 'H' horizontal only
c                  ' ' consider both
c         units - 'S' SI and 'E' English
c         mult - allow multiple vessels 'Y' or 'N'
c         vtype - type of velocity correlation to use stokes 'S' or
c                 'N' Newtonian
c         gas  - mass flow (kg/s) (lbs/hr)
c         liq - mass flow, only needed for 'N' vtype (kg/s) (lbs/hr)
c         vliq - Required liquid phase volume (m**3) (ft**3)
c         vmax - maximum vessel size (m**3) (ft**3)
c         denl - liquid density (kg/m**3) (lb/ft**3)
c         deng - gas density (kg/m**3) (lb/ft**3)
c         dp   - Maximum size of entrained (m) (micron)
c         visc - gas viscosity, only neede for vtype 'S' (Pa-s) (cp)
c         ldr  - Minimum desire L/D ratio
c         hmin - Minimum height of void space (m) (ft)
c                (1 ft Walas)
c         dmin - Minimum diameter, only for vtype='S' (m) (ft)
c
c  Output: number - number of vessels
c          volume - volume of each vessel (m**3) (ft**3)
c          len - drum length (m) (ft)
c          diam - drum diam (m) (ft)
c          type - type of drum 'H' horizontal, 'V' vertical
c
c-----------------------------------------------------------------------
c
c  Rev  1.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      character*1 config,mult,vtype,units,type
      integer luin,number
```

```
      real*8 gas,liq,vliq,vmax,denl,deng,dp,visc,volume,ldr,hmin
      real*8 len,diam,dmin
c  local params
      real*8 xgas,xliq,xvliq,xvmax,xdenl,xdeng,xdp,xvisc,xhmin
      real*8 xvolume,xlen,xdiam,xdmin

      if (units.eq.'E') then
c  convert to SI units
        xgas=gas*0.454/3600
        xliq=liq*0.454/3600
        xvliq=vliq*0.02832
        xvmax=vmax*0.02832
        xdenl=denl*16.02
        xdeng=deng*16.02
        xdp=dp*1.0d-6
        xvisc=visc*1.0e-3
        xhmin=hmin*0.3048
        xdmin=dmin*0.3048
      else
        xgas=gas
        xliq=liq
        xvliq=vliq
        xvmax=vmax
        xdenl=denl
        xdeng=deng
        xdp=dp
        xvisc=visc
        xhmin=hmin
        xdmin=dmin
      endif

c  call SI routine
      call sep1(luin,config,mult,vtype,xgas,xliq,xvliq,xvmax,xdenl,
     &    xdeng,xdp,xvisc,ldr,xhmin,xdmin,number,xvolume,xlen,xdiam,
     &    type)

      if (units.eq.'E') then
c  convert results to English units
        volume=xvolume/0.02832
        len=xlen/0.3048
        diam=xdiam/0.3048
      else
        volume=xvolume
        len=xlen
        diam=xdiam
      endif

      return
      end


      subroutine sep1(luin,config,mult,vtype,gas,liq,vliq,vmax,denl,
     &       deng,dp,visc,ldr,hmin,dmin,number,volume,len,diam,type)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Computation control routine.  See sep for description of computations
c  and input/output.
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      character*1 config,mult,vtype,type
      integer luin,number
      real*8 gas,vliq,vmax,denl,deng,dp,visc,volume,ldr,hmin,len,diam
```

```
      real*8 liq,dmin
c  local params
      integer lu,nmax
      real*8 vliqx,gasx,ratio,liqx
      parameter (nmax=20)

      lu=abs(luin)
      write(lu,1000)
 1000 format ("%%%%%%%%%%%%%%%%%%%%%%%% SEP %%%%%%%%%%%%%%%%%%%%%%%%")
      write (lu,'(   ''Parameters:''
     &             /3x,''Allow multiple vessels:'',a
     &             /3x,''Configuration:'',a)') mult,config

      number=0

      if (config .ne. 'V' .and. config .ne. 'H') then
c -------- Both configuration considered --------------------------
  100 number=number+1
         vliqx=vliq/number
        gasx=gas/number
        liqx=liq/number
         call sepv(luin,vtype,gasx,liqx,vliqx,vmax,denl,deng,dp,
     &                visc,dmin,volume,len,diam,ratio)
        type='V'

        if ( len/diam .gt. 5.0 .or. luin.lt.0) then

             call seph(luin,vtype,gasx,liqx,vliqx,vmax,denl,deng,dp,
     &            visc,ldr,hmin,dmin,volume,len,diam)
          type='H'
        endif
        if (luin.lt.0) then
           number=1
           volume=1
            goto 800
        endif
        if (mult.eq.'Y') then
           if (number.gt.nmax) then
              write(lu,'(''Max number iteration exceeded'')')
              goto 800
            endif
           if (volume.gt.vmax .and. diam.gt.1.1*dmin) then
              if (type.eq.'V' .and. ratio .lt. 3.0) goto 100
              if (type.eq.'H') goto 100
           endif
         endif

      elseif (config .eq. 'V' ) then
c -------- Only verical configuration considered --------------------------
  200 number=number+1
         vliqx=vliq/number
        gasx=gas/number
        liqx=liq/number
         call sepv(luin,vtype,gasx,liqx,vliqx,vmax,denl,deng,dp,
     &                visc,dmin,volume,len,diam,ratio)
        type='V'

        if (luin.lt.0) then
           number=1
           volume=1
            goto 800
        endif
        if (mult.eq.'Y') then
           if (number.gt.nmax) then
```

```
            write(lu,'(''Max number iteration exceeded'')')
            goto 800
          endif
          if (volume.gt.vmax .and. diam.gt.1.1*dmin) goto 200
        endif

      else
c -------- Only horizontal configuration considered ----------------------
  300 number=number+1
        vliqx=vliq/number
        gasx=gas/number
        liqx=liq/number
        call seph(luin,vtype,gasx,liqx,vliqx,vmax,denl,deng,dp,
     &            visc,ldr,hmin,dmin,volume,len,diam)
        type='H'
        if (luin.lt.0) then
          number=1
          volume=1
          goto 800
        endif
        if (mult.eq.'Y') then
          if (number.gt.nmax) then
            write(lu,'(''Max number iteration exceeded'')')
            goto 800
          endif
          if (volume.gt.vmax .and. diam.gt.1.1*dmin) goto 300
        endif
      endif

  800 write(lu,'(''Number of vessels : '',i3
     &          /''Volume per vessel, m**3 : '',f8.2
     &          /''Vessel type : '',a)')
     &    number,volume,type
      write(lu,1000)
      return
      end


      subroutine seph(luin,vtype,gas,liq,vliq,vmax,denl,deng,dp,visc,
     &                ldr,hmin,dmin,vol,len,diam)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Compute the size and number of vessels need in a flash drum
c  operation based on simple correlations in Walas for horizontal
c  drums.
c
c  Input: luin - write computational info to report file if nrpt
c                is ASPEN's lu lu.  Set to zero for no output.
c         vtype - type of velocity correlation to use stokes 'S' or
c                 'N' Newtonian
c         gas  - mass flow (kg/s)
c         liq - mass flow, only needed for 'N' vtype (kg/s) (lbs/hr)
c         vliq - Required liquid phase volume (m**3)
c         vmax - maximum vessel size (m**3)
c         denl - liquid density (kg/m**3)
c         deng - gas density (kg/m**3)
c         dp   - Maximum size of entrained  (m)
c         visc - gas viscosity (Pa-s)
c         dp   - Maximum size of entrained
c         ldr  - Minimum desire L/D ratio
c         hmin - Minimum height of void space (m)
c         dmin - Minimum diameter (m)
c
c  Output:  vol - volume of vessel (m**3)
```

```
c           len - length of vessel (m)
c           diam - diameter of vessel (m)
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      character*1 vtype
      integer luin
      real*8 gas,vliq,vmax,denl,deng,dp,visc,vol,ldr,hmin
      real*8 len,diam,dmin,liq
c  local params
      integer lu
      real*8 phi,h,pi,areamin,velmax,surfa
      data pi/3.1416/

      lu=abs(luin)
      write(lu,1000)
 1000 format ("%%%%%%%%%%%%%%%%%%%%%%%%%% SEPH %%%%%%%%%%%%%%%%%%%%%%%%")

      write (lu,'(   ''Parameters:''
     &             /3x,''Terminal velocity type : '',a
     &             /3x,''Gas flow, kg/s :'',1pe12.3
     &             /3x,''Liq flow, kg/s :'',1pe12.3
     &             /3x,''Liquid volume required, m**3 :'',1pe12.3
     &             /3x,''Maximum vessel volume, m**3 :'',1pe12.3
     &             /3x,''Liquid density, kg/m**3 :'',1pe12.3
     &             /3x,''Gas density, kg/m**3 :'',1pe12.3
     &             /3x,''Gas viscosity, Pa-s :'',1pe12.3
     &             /3x,''Min L/D :'',1pe12.3
     &             /3x,''Min void height, m :'',1pe12.3
     &             /3x,''Min diameter, m :'',1pe12.3
     &             /3x,''Maximum particle size, m :'',1pe12.3)')
     &   vtype,gas,liq,vliq,vmax,denl,deng,visc,ldr,hmin,dmin,dp
      if (luin.lt.0) return

c  compute max velocity
      if (vtype.eq.'S') then
        call sep2(deng,denl,visc,dp,velmax)
      else
        call sep3(deng,denl,gas,liq,velmax)
      endif
      velmax=velmax*1.25  ! horizontal drum

c  compute minimum required area
      areamin=gas/deng/velmax
      dphi=0.3/20

c  be sure l/d .ge. ldr
      do i=1,20
c     compute area based on void fraction
        phi=0.2+(i-1)*dphi
        area=areamin/phi
c  do not consider any area less than that for dmin
         area=max(area,(pi*dmin*dmin/4.0))

c     compute length based on vliq & area
        vol=vliq/(1.0-phi)
        len=vol/area
        call seph1(phi,vliq,len,h,diam)
        surfa=pi*diam*len+pi/2.0*diam**2
        write(lu,'(''phi,diam,len,diam-h,surfa,len/diam'',8f8.2)')
     &         phi,diam,len,(diam-h),surfa,len/diam
        if (len/diam.ge.ldr .and. h.ge.hmin) goto 200
      enddo
```

```
c  need longer drum to meet ldr minimum, compute len
      len=diam*ldr

 200  vol=len*pi*diam**2/4.0
      write (lu,'(''Computed results '',
     &          /3x,''velmax :'',1pe13.3
     &          /3x,''phi :'',1pe13.3
     &          /3x,''vol :'',1pe13.3
     &          /3x,''len :'',1pe13.3
     &          /3x,''diam :'',1pe13.3
     &          /3x,''len/diam:'',1pe13.3
     &          /3x,''h    :'',1pe13.3
     &            )'')
     &    velmax,phi,vol,len,diam,len/diam,h

      write (lu,1000)

      return
      end


      subroutine sephl(phi,vol,len,h,diam)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Given a liquid volume, fraction area fill nd drum length compute
c  liquid fill height, h, and drum diameter, diam.
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      real*8 phi,vol,len,h,diam
c  local params
      real*8 y ! distance from top to interface / radius
      real*8 dy,r,phix,pi
      data pi/3.1416/

      diam=sqrt(4.0*vol/(1.0-phi)/(len*pi))
      r=diam/2.0
      dy=0.01
      do i=1,100
        y=i*dy
        phix=(1/pi)*(acos(1-y)-(1-y)*sqrt(2.0*y-y**2))
cccc    write (6,'(''y,phi '',2f10.3)') y,phix
        if (phix.ge.phi) then
           h=(2-y)*r
           return
        endif
      enddo
      h=-99
      return
      end


      subroutine sep2(deng,denl,visc,dp,vel)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Stokes law.  Given density differences, paritcle diameter and
c  fluid viscosity compute minimum velocity to, entrain particle.
c  Assumes laminar, but used in estimates after Walas pg 614.
c
c  Input: deng - gas density (kg/m**3)
c         denl - liquid density (kg/m**3)
c         visc - gas viscosity (Pa-s)
```

```
c         dp - particle size (m)
c
c  Output: vel - terminal velocity (m/s)
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      real*8 deng,denl,visc,dp,vel
c  local params
      real*8 g
      data g/9.8/

      vel=g*(denl-deng)*dp**2/(18.0*visc)
      return
      end


      subroutine sep3(deng,denl,gas,liq,vel)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Newton's law (Watson).  Given densities and flows of gas & liq
c  computes maximum gas velocity to allow for 5% liq entrainment.
c
c  Input: deng - gas density (kg/m**3)
c         denl - liquid density (kg/m**3)
c         gas - gas flow rate (kg/s)
c         liq - liq flow rate (kg/s)
c
c  Output: vel - terminal velocity (m/s)
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      real*8 deng,denl,gas,liq,vel
c  local params
      real*8 x,kv
      integer nt,i
      parameter (nt=12)
      real*8 xt(nt),yt(nt)
      data xt/0.006,0.01,0.02,0.04,0.06,0.10,
     &       0.20,  0.40,0.60,1.00,2.00,4.0/
      data yt/0.23, 0.32,0.40,0.43,0.43,0.42,
     &        0.36, 0.27,0.21,0.14,0.075,0.03/

      x=liq/gas*sqrt(deng/denl)
      do i=1,nt
        if (x.le.xt(i)) then
           if (i.eq.1) then
              kv=yt(1)
           else
              kv=yt(i-1)+(x-xt(i-1))*(yt(i)-yt(i-1))/(xt(i)-xt(i-1))
           endif
           goto 100
        endif
      enddo
      kv=yt(nt)
  100 vel=kv*sqrt((denl-deng)/deng)  ! ft/s
      vel=vel/3.28 ! m/s
      return
      end


      subroutine sepv(luin,vtype,gas,liq,vliq,vmax,denl,deng,dp,visc,
```

```
     &                  dmin,vol,len,diam,ratio)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c
c  Compute the size and number of vessels need in a flash drum
c  operation based on simple correlations in Walas for vertical
c  drums.
c
c  Input: luin - write computational info to report file if nrpt
c                is ASPEN's lu lu.  Set to zero for no output.
c         vtype - type of velocity correlation to use stokes 'S' or
c                 'N' Newtonian
c         gas  - mass flow (kg/s)
c         liq - mass flow, only needed for 'N' vtype (kg/s) (lbs/hr)
c         vliq - Required liquid phase volume (m**3)
c         vmax - maximum vessel size (m**3)
c         denl - liquid density (kg/m**3)
c         deng - gas density (kg/m**3)
c         dp   - Maximum size of entrained (m)
c         visc - gas viscosity (Pa-s)
c         dmin - Minimum diameter (m)
c
c  Output:  volume - volume vessel (m**3)
c           len - length of vessel (m)
c           diam - diameter of vessel (m)
c           ratio - initially computed L/D ratio.  The routine
c                   expands to a minimum of L/D.
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c     implicit none
c  formal params
      character vtype*1
      integer luin
      real*8 gas,vliq,vmax,denl,deng,dp,visc,vol,len,diam,ratio,dmin
      real*8 liq
c  local params
      integer lu
      real*8 pi,velmax, lenvapor, lenliq
      data pi/3.1416/

      lu=abs(luin)
      write(lu,1000)
 1000 format ("%%%%%%%%%%%%%%%%%%%%%%%%% SEPV %%%%%%%%%%%%%%%%%%%%%%%%")

      write (lu,'(   ''Parameters:''
     &             /3x,''Terminal velocity type : '',a
     &             /3x,''Gas flow, kg/s :'',1pe12.3
     &             /3x,''Liq flow, kg/s :'',1pe12.3
     &             /3x,''Liquid volume required, m**3 :'',1pe12.3
     &             /3x,''Maximum vessel volume, m**3 :'',1pe12.3
     &             /3x,''Liquid density, kg/m**3 :'',1pe12.3
     &             /3x,''Gas density, kg/m**3 :'',1pe12.3
     &             /3x,''Gas viscosity, Pa-s :'',1pe12.3
     &             /3x,''Min diameter, m :'',1pe12.3
     &             /3x,''Maximum particle size, m :'',1pe12.3)')
     &   vtype,gas,liq,vliq,vmax,denl,deng,visc,dmin,dp
      if (luin.lt.0) return


c  compute max velocity
      if (vtype.eq.'S') then
        call sep2(deng,denl,visc,dp,velmax)
      else
        call sep3(deng,denl,gas,liq,velmax)
      endif
```

```
c  compute equired area
      area=gas/deng/velmax
c  do not consider any diameter less than dmin
      area=max(area,(pi*dmin*dmin/4.0))
      diam=sqrt(4.0*area/pi)
      lenvapor=1.68  ! based on Walas
      lenliq=vliq/area
      len=lenvapor+lenliq
      ratio=len/diam
      if (len/diam.lt.3.0) then
        write(lu,'(''Computed ratio L/D='',f8.2)') ratio
        len=3.0*diam ! L/D at least 3
      endif
      vol=len*pi*diam**2/4.0

      write (lu,'(''Computed results '',
     &          /3x,''velmax :'',1pe13.3
     &          /3x,''vol :'',1pe13.3
     &          /3x,''len :'',1pe13.3
     &          /3x,''diam :'',1pe13.3
     &          /3x,''len/diam:'',1pe13.3
     &            )')
     &    velmax,vol,len,diam,len/diam

      write (lu,1000)

      return
      end
```

Technical Information Department • **Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551