September 1995

MDC 95W5104

Molecular Sieve Bench Testing and Computer Modeling

Contract NAS8-38250-17

Final Report

McDonnell Douglas Aerospace - Huntsville

MCDONNELL DOUGLAS

Molecular Sieve Bench Testing and Computer Modeling

September 1995

MDC 95W5104

Prepared By:
Habib Mohamadinejad
Principal Engineer
Product Development Department

Approved By: Robert C DaLee
Robert C. DaLee
Senior Engineer
Product Development Department

Approved By: James B. Blackmon
Dr. James B. Blackmon
Director
Product Development Department

Prepared for the National Aeronautics and Space Administration
under NASA Contract NAS8-38250-17

McDonnell Douglas Aerospace - Huntsville
689 Discovery Drive, Huntsville, AL 35806 (205) 922-6600

PREFACE

This document has been prepared by the Product Development Group of McDonnell Douglas Aerospace - Huntsville (MDA-HSV) to document the mathematical model and numerical simulation of adsorption of H_2O and CO_2 on 5A zeolite. The work was performed under the Molecular Sieve Bench Testing and Computer Modeling contract (NAS8-38250-17) to ION Electronics.

THIS PAGE INTENTIONALLY LEFT BLANK

CONTENTS

Section **Page**

CONTENTS (continued)

Appendcies

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

NOMENCLATURE

a	Surface area of pellets per unit volume of pellet ft^2/ft^3
A	Surface area ft^2
B	Langmuir constant
C	Constant in Darcy equation
C	Gas stream concentration, lb mole/ft^3
C_{ip}	Gas phase concentration of ith component in the pores of particles, lb mole/ft^3
$\bar{C}_{ip}$	Volume average pore Gas phase concentration of ith component i, lb mole/ft^3
C_i^s	Gas phase concentration of ith component at the surface of the solid phase lb mole/ft^3
$C_{i,0}$	Gas phase concentration of ith component at boundary or initial lb mole/ft^3
c_{pg}	Heat capacity of gas phase, Btu/lbm-R
C_{ps}	Heat capacity of solid particle, Btu/lbm-R
C_{pw}	Heat capacity of column wall , Btu/lbm-R
d, d_t	Column diameter, ft
D	diffusivity, ft^2/hr
D_l	Axial diffusion, ft^2/hr
D_{ki}	Knudsen diffusion, ft^2/hr
D_{si}	Surface diffusion,ft^2/hr
D_e	Effective diffusion, ft^2/hr
D_{mi}	Molecular diffusion in mixture, ft^2/hr
D_o	Stagnant diffusivity, ft^2/sec
D_f	Fluid flow diffusivity, ft^2/sec
$D_{eff,x}$	Effective axial diffusivity , ft^2/sec
$D_{eff,r}$	Effective radial diffusivity, ft^2/sec
$D_{f,x}$	Axial diffusivity of fluid flow, ft^2/sec
$D_{f,r}$	Radial diffusivity of fluid flow, ft^2/sec
d_w	Bed diameter, ft
E_{si}	Surface diffusion energy of activation of species i
G	Superficial mass velocity, lb$_m$/ft^2 hr
h_c	Effective Heat transfer coefficient for column insulation, Btu/ft^2-hr
h_w	Heat transfer coefficient between the gas stream and the column wall, Btu/ft^2-hr
h_s	Heat transfer coefficient between the gas stream and the sorbent , Btu/ft^2-hr
ΔH	Heat of Adsorption , Btu/lb mole
K	Constant in Darcy equation
k	Fluid flow axial conductivity, dispersion, Btu/ft-hr-R
k_o	Stagnant thermal conductivity, Btu/ft-hr-R
$k_{eff,x}$	Effective axial conductivity, Btu/ft-hr-R
$k_{eff,r}$	Effective radial conductivity, Btu/ft-hr-R
$k_{f,x}$	axial conductivity of fluid flow, Btu/ft-hr-R
$k_{f,r}$	Radial conductivity of fluid flow, Btu/ft-hr-R
k_s	Solid thermal conductivity, Btu/ft-hr-R
k_{fi}	Fluid film mass transfer coefficient of ith component, ft/hr
$k_{eff,i}$	Effective mass transfer coefficient of ith component, ft/hr
M_i	Molecular weight of adsorbate i, lb/lb mole
n	Number of component
N_{Pe}	Peclet number, $c_{pg}\rho_g u/k$

NOMENCLATURE

N_{Re}	Reynolds number, $d_p \rho_g u/\mu$
N_{sc}	Schmidt number, $\mu/D\rho_g$
N_{ir}	Molar flux of component i, lb mole/ft^2-hr
P	Total pressure, mmHg or lbf/ft^2
P_i	Partial pressure of component i, mmHg or lbf/ft^2
P_{CA}	critical pressure of component A, Atm
P_{CB}	critical pressure of component B, Atm
q	Amount adsorbed in the solid, lb moles/ft^3 of solid
$\overset{*}{q}_i$	Solid phase concentration of i^{th} component in equilibrium with gas phase, lbs moles/ft^3 of solid
$\bar{q}_i$	Volume average solid phase concentration of component i, lb moles/ft^3 of solid
q_{mi}	Langmuir constant
r	Radial position, ft
r_e	Mean pore radius, A°
r_p	Radial spatial particle dimension, ft
R	Ideal gas constant 555 mm Hg ft^3/lb mol R
R_i	Inside wall diameter of column, ft
R_o	Outside wall diameter of column, ft
R_p	Particle radius, ft
t	Time , hr
T	Temperature R
T_o	Ambient temperature, R
T_g	gas temperature, R
T_w	Wall temperature, R
T_s	Solid temperature, R
T_{CA}	Critical temperature of component A, K
T_{CB}	Critical temperature of component B, K
u	Interstitial velocity, ft/hr
$V_\circ$	Velocity at the center of packed column, ft/hr
x	Axial position
x_i	Mole fraction of i^{th} component in the solid phase
y_i	Mole fraction of i^{th} component in the gas phase

GREEK

α	Intraparticle void fraction
$\alpha\beta$	Constant in effective conductivity equation
ε	External bed void volume
λ	Constant in effective conductivity equation
ϕ	Constant in stagnant conductivity equation
ρ_{pg}	Density of gas phase, lb mole/ft^3
ρ_s	Density of solid phase, lbs/ft^3
ρ_w	Density of column wall, lb mole/ft^3
τ_p	Pore tortuosity factor
τ_s	Surface tortuosity factor
π	Constant or spreading pressure
σ	Emissivity
ΔH	Heat of adsorption, BTU/lb of solid
ω	Over-Relaxation coefficient

SUBSCRIPTS

i	ith component
e	Effective
eff	Effective
ip	ith component in the pores
o	Outside, initial
p	Pore
pg	Gas phase
ps	Golid phase
s	Surface
t	total
w	Wall

SUPERSCRIPTS

-	Average value
*	Equilibrium value

SECTION 1

INTRODUCTION

Adsorption is a process involving the selective transfer of one or more solutes from fluid solutions onto and into solids. Physical adsorption is a surface phenomenon. When fluid is in contact with a solid surface, molecules in the fluid phase spontaneously concentrate on the surface without any chemical change. At low temperature, adsorption is usually caused by intermolecular forces, such as Van der waal forces (Ruthven, 1984). Adsorption phenomenon, like any other thermodynamic system which involves the interaction of two phases, is driven by chemical potential differences between the phases. This chemical potential determines the strength with which any given molecule is adsorbed. The differences in the surface energy of adsorbent and the properties of each kind of molecule cause differences in the amount adsorbed in certain kinds of surface, i.e., the selectivity. Separation can be accomplished through selective collection and concentration of fluid molecules onto a solid surface.

Separation of gas mixtures by adsorption can be obtained by using either single column or multi-column configuration. However, for both processes the adsorbent column is alternately saturated and regenerated in a cyclic manner.

One example of a cyclic selective process is the Four-Bed Molecule Sieve (4BMS) used in International Space Station Alpha for the removal of CO_2 as shown in Figure 1-1.

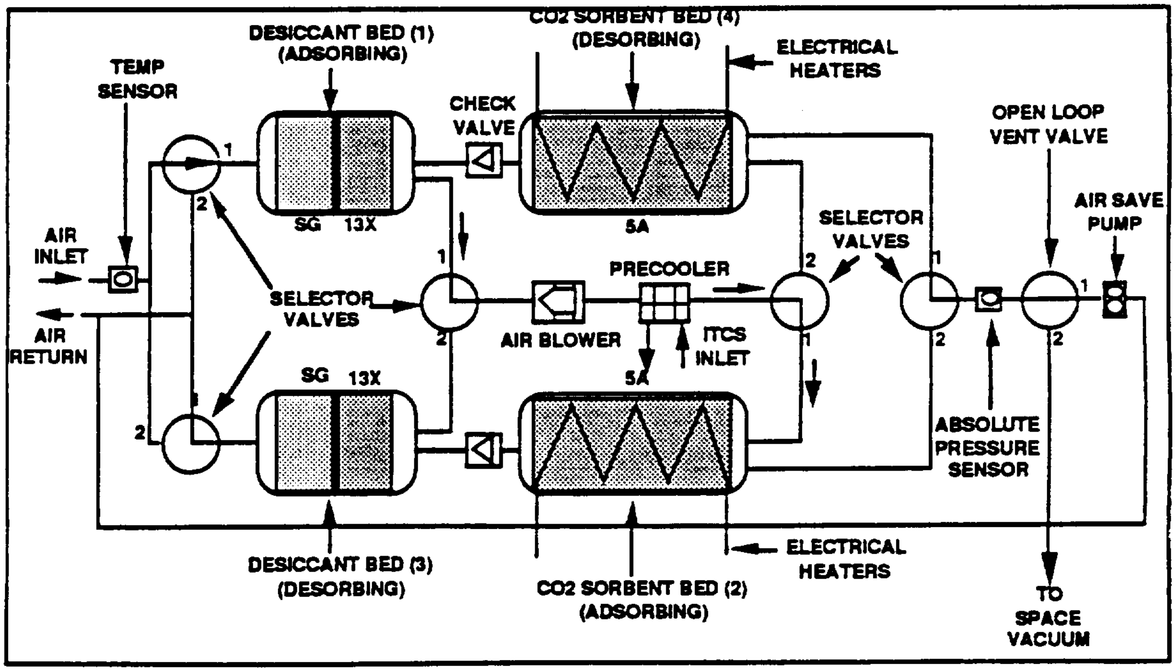


FIGURE 1-1 SPACE STATION 4-BED MOLESIEVE

As shown in Figure 1, the actual CO_2 removal system consists of four beds. Two beds operate in the adsorption mode (a desiccant and CO_2 sorbent bed) while the other set of identical beds desorb. The desiccant beds desorb through gas stripping while the sorbent beds are heated and subjected to a vacuum (e.g., the pressure and thermal swing process). Following the air flow path in Figure 1-1, cabin air laden with CO_2 and water enters at the system inlet, then enters desiccant bed (1), which consists of Silica Gel and 13X sorbent materials. The desiccant bed is required to remove virtually all water from the air stream. The dry air is next pulled through the air blower, which overcomes the system pressure drop. A heat exchanger, the precooler, removes the heat of compression and heat of adsorption from the desiccant bed before the air enters the 5A sorbent which is responsible for removing the CO_2 from air. Following the precooler, the 5A bed (2), CO_2 removal bed, performs the primary system function of removing CO_2. The other 5A bed (4) is concurrently desorbed with heat from embedded electrical heaters and allowed to vent to space vacuum . At the beginning of a new half cycle, the selector valves cycle to alternate the desorbing and adsorbing beds. Heated air from the hot, previously desorbing CO_2 sorbent bed, trips the downstream desiccant bed (3) of water, which is returned to the cabin in the return air stream.

Because of the complexity of the process, the design of an efficient 4BMS CO_2 removal system mission depends on many mission parameters, such as, duration, crew size, cost of power, volume, fluid interface properties, etc. A need for space vehicle CO_2 removal system models capable of accurately performing extrapolated hardware predictions is inevitable due to the change of the parameters which influences the CO_2 removal system capacity.

The purpose of this study is to (a) investigate the mathematical techniques required for a model capable of accurate extrapolated performance predictions and (b) obtain test data required to estimate mass transfer coefficients and verify the computer model. Models have been developed to demonstrate that the finite difference technique can be successfully applied to sorbents and conditions used in spacecraft CO_2 removal systems. The non isothermal, axially dispersed, plug flow model with linear driving force for 5X and pore diffusion for Silica gel are then applied to test data. A more complex model, a non-darcian model (2-dimensional), has also been developed for simulation of the test data. This model takes into account the channeling effect on column breakthrough.

SECTION 2

DEVELOPMENT AND ANALYSIS OF A MATHEMATICAL MODEL

2.1 Gas/Solid Equilibrium

Gas/solid equilibria is the thermodynamic limiting capacity for mass transfer in gas adsorption. It is one of the most important parameters for predicting solute separation from the bulk gas phase. Since the composition, temperature, and pressure change in a wide range with time and location within the adsorption column, a comprehensive gas/solid equilibrium model is needed.

2.1.1 Single Component Isotherm

The most popular single solute isotherms were correlated by Langmuir in 1918. The Langmuir approach assumed the rate of adsorption is equal to that of condensation. The final form of this equation is:

$$q_i^* = \frac{q_{mi} B_i P_i}{1 + B_i P_i} \qquad \text{eq. 2.1}$$

where B_i, and q_{mi} are functions of temperature and heat of adsorption. Some of the single solute isotherms were correlated by the Langmuir-Freundlich equation (Sips, 1948):

$$q_i^* = \frac{q_{mi} B_i P_i^{n_i}}{1 + B_i P_i^{n_i}}, \qquad \text{eq. 2.2}$$

where n_i is a constant close to one.

The pure isotherm constants of CO_2, H_2O, and N_2 on 13X zeolite are as follows:

for CO_2,

for T less or equal than 627.0 R

$$q_{m,CO_2} = (399.3942 - 1.93842889T + 3.25405x10^{-3}T^2 - 1.82262x10^{-6}T^3) / 44 / 100$$

$$B_{CO_2} = 35.35.358072159 - .17016733T + 2.74587x10^{-4}T^2 - 1.48412x10^{-7}T^3$$

$$n = -29.3861079 + .153889849T - 2.61225877x10^{-4}T^2 + 1.478185x10^{-7}T^3$$

for H_2O,

if partial pressure of H_2O less or equal than .4 mm Hg

for $T \leq$ 564.0 R

$$q_{m,H_2O} = (39.914452 - 8.87103x10^{-2}T + 6.839503x10^{-5}T^2) / 100 / 18.0$$

for T > 564.0 R

$$q_{m,H_2O} = (873.44464 - 3.867835T + 5.80375x10^{-3}T^2 - 2.9346685x10^{-6}T^3) / 100 / 18$$

$$n = 1.0$$

for partial presuure of greater than .4 mm Hg

$$q_{m,H_2O} = (95.0938247 - .217257754T + 1.4628603x10^{-4}T^2)/100/18$$

$$B_{H_2O} = 48.56664 - 2.262024x10^{-1}T + 3.91017x10^{-4}T^2 - 2.3790985x10^{-7}T^3$$

$$n = 1.0$$

and for N_2

$$q_{m,N_2} = .00961197026 - 1.637879912x10^{-5}T \qquad \begin{array}{l} \textit{for T less than 610 °R} \\ B_{N_2} = 4.5597278759x10^{-7}\exp(9628.9655743/T) \\ \textit{else} \\ B_{N_2} = 5.808906668x10^{-7}\exp(9911.5734593/T) \end{array}$$

where q is in lb mole of solute per lb of solid, T in Rankin, and P in mm Hg.

2.1.2 Multicomponent Isotherm

Two types of multicomponent correlation were used in this study. One is based on the Langmuir-Freundlich equation for multicomponent system

$$q_i^* = \frac{q_{mi}B_iP_i^{n_i}}{1+\sum_j B_jP_j^{n_j}}, \qquad \text{eq 2.3}$$

and the more reliable method is based on Ideal Adsorption Solution Theory (IAST) of Myers and Prausnitz (1965). For pure-gas adsorption and assuming ideal gas behavior:

$$\frac{\pi(p_i^*)A}{RT} = \int_0^p \frac{q_i^*}{p_i^*}dP_i^*, \qquad \text{eq 2.4}$$

where A is the surface area, π is the spreading pressure. The ideal solution assumption is applied to the adsorbed phase and the total amount adsorbed is then related to the mole fraction of each component:

$$1/q_t = \sum_i x_i/q_i^*. \qquad \text{eq 2.5}$$

The Raoult law is also applied to the relation between mole fraction in the gas phase and adsorbed phase:

$$P_i = Py_i = p_i^*(\pi)x_i, \qquad \text{eq 2.6}$$

where y_i and x_i are mole fraction of the i-th component in the gas phase and in adsorbed phase. respectively, and

$$\sum x_i = \sum y_i = 1 \qquad \text{eq 2.7}$$

The IST theory assumes that the spreading pressure is the same for all the components in the mixture.

The solution for the amount adsorbed in the solid phase in equilibrium with the gas phase concentration is obtained by an iterative procedure with the IST theory. For the case of the Langmuir-Freundlich single isotherm, from equations 2.2,2.4,2.6, and 2.7 one gets:

$$\sum_{i=1}^{i=n} \frac{P y_i}{\left(\dfrac{\exp\left(\dfrac{\pi A n_i}{R T q_{mi}} \right) - 1}{B_i} \right)^{\frac{1}{n_i}}} - 1 = 0 \qquad \text{eq 2.8}$$

The derivation is shown in Appendix A. Knowing the total pressure and the solid temperature, a value of π is guessed and solution will be obtained by iteration until equation 2.8 is satisfied.

2.2 Mathematical Model For Nonisothermal Multicomponent Adsorption In a Packed Bed

Separation of solutes from the bulk gas can be accomplished in a packed column which is filled with a specific adsorbent that preferentially adsorbs certain constituents from the bulk gas. In the simplest case of packed column adsorbent particles, the following mass transport processes are considered: axial dispersion in the interparticle fluid phase, fluid to particle mass transfer, and a reversible adsorption in the interior of the particle. Also, adsorption is accompanied by the evolution of heat, and temperature changes affect the adsorption equilibrium relation and, in some cases, adsorption rate. Thus, especially in the gas phase adsorption, not only the effects of mass transfer on adsorption rate but also effects of heat generation and heat transfer in the adsorbent bed must be taken into account.

The dynamic bed behavior can be modeled by heat and mass balance equations. The mathematical model will be used to estimate the breakthrough curve for a certain constituent in the bulk gas. In return, this enables one to obtain the necessary parameters for predicting the transient behavior of the temperature profile and concentration of the gas for different initial parameters such as inlet concentration, temperature, and the fluid velocity.

The heat and mass balances could not be solved analytically; therefore, these equations were solved numerically by finite differences methods, namely the Under-Relaxation and Newman methods. A FORTRAN code was written to find the numerical solutions to the transient equations.

In this study two types of mathematical models were used to predict the dynamic behavior of the bed. A one dimensional model was developed which takes into account only the mass and heat transfer rate of change in the axial direction of the bed. On the other hand, a two dimensional model would also estimate these rates in the radial direction of the bed, and therefore this model was also developed. The mathematical derivation of a two dimensional model is shown in Appendix B.

2.2.1 One Dimensional Adsorption Mathematical Model

In a one dimensional adsorption model in a packed column any rates of change of variables in the radial direction are assumed to be negligible compared to the axial direction. Also, the porosity of the bed is considered to be constant from the center to the column wall .

2.2.1.1 Mass Balance Equation

In the bulk stream of the gas within the bed, the material balance for the adsorbate concentration is

$$\frac{\partial C_i}{\partial t} = D_l \frac{\partial^2 C_i}{\partial x^2} - u\frac{\partial C_i}{\partial x} - \varepsilon \frac{\partial \bar{q}_i}{\partial t}$$

Boundary / Initial Condition

$$\text{at } t < 0,\ C_i = C_{i,o} \text{ for } 0 \le x \le L$$

$$\text{at } t < 0,\ \bar{q}_i = \bar{q}_{i,o} \text{ for } 0 \le x \le L$$

$$\text{at } t \ge 0,\ C_i = C_{i,o} \text{ for } x = 0 \qquad \text{eq 2.9}$$

$$\text{at } t \ge 0,\ \frac{\partial C_i}{\partial x} = 0 \text{ for } x = L$$

where C_i is the concentration of each component in the gas bulk, D_l is the axial dispersion, u is the interstitial velocity, ε is the void fraction, and $\bar{q}_i$ is the adsorbate concentration in the solid phase. The last boundary condition indicates a free stream boundary type. This boundary condition is prescribed for situation where the normal derivative of flux, the gradient of concentration or temperature, at the end of the packed column vanishes. In a numerical sense, the value of the last grid outside the packed bed is set equal to that of last grid of the packed bed.

2.2.1.2 Total Material Balance

Assuming the ideal gas law $C_i = {}^{Py_i}\!/_{RT}$ and knowing $\sum y_i = 1$, the above equation can be recast into an overall mass balance equation,

$$\frac{\partial P}{\partial t} = D_l \frac{\partial^2 P}{\partial x^2} - u\frac{\partial P}{\partial x} - P\frac{\partial u}{\partial x} + \frac{P}{T_g}\left(\frac{\partial T}{\partial t} - D_l \frac{\partial^2 T}{\partial x^2} + u\frac{\partial T}{\partial x}\right) - RT\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n}\frac{\partial \bar{q}_i}{\partial t} \qquad \text{eq 2.10}$$

This equation was used to compute the pressure drop along the axial direction in the bed.

2.2.1.3 Gas Phase Energy Equation

The change of gas temperature with respect to time is the result of heat flux from the solid to the gas plus convection of heat due to the fluid flow, as shown by the following equation:

$$\rho_g c_{pg}\frac{\partial T_g}{\partial t} = k_f\frac{\partial^2 T_g}{\partial x^2} - u\rho_g c_{pg}\frac{\partial T_g}{\partial x} + \frac{1-\varepsilon}{\varepsilon}h_s a_s(T_s - T_g) - \frac{4h_w}{\varepsilon d_w}(T_g - T_w)$$

Boundary / Initial Conditions

$$\text{at } t < 0,\ T_g = T_{g,0} \text{ for } 0 \le x \le L$$

$$\text{at } t \ge 0,\ T_g = T_i \text{ for } x = 0 \qquad \text{eq 2.11}$$

$$\text{at } t \ge 0,\ \frac{\partial T_g}{\partial x} = 0 \text{for } x = L$$

where T_g is bulk gas temperature, T_w is the wall temperature, T_s is the solid temperature, ρ_g is bulk density, c_{pg} is the specific heat of the bulk gas , k_f is the effective conductivity of the bulk gas, h_s is the heat transfer coefficient between the solid and the gas phase (Petrovic and Thodos, 1968), and h_w is the heat transfer coefficient between the wall and the gas phase.

2.2.1.4 Solid Phase Energy Equation

The following energy equation for the solid phase includes the term for heat flux from the solid phase to the gas phase plus heat generation due to adsorption.

$$\rho_s c_{ps}\frac{\partial T_s}{\partial t} = k_s\frac{\partial^2 T_s}{\partial x^2} + h_s a_s(T_g - T_s) - \sum_{i=1}^{n}\Delta H_i\frac{\partial \bar{q}_i}{\partial t}$$

Boundary / Initial Conditions

$$\text{at } t < 0,\ T_s = T_{s,0} \text{ for } 0 \le x \le L$$

$$\text{at } t \ge 0,\ T_s = T_i \text{ for } x = 0 \qquad \text{eq 2.12}$$

$$\text{at } t \ge 0,\ \frac{\partial T_s}{\partial x} = 0 \text{ for } x = L$$

2.2.1.5 Column Wall Energy Equation

The wall temperature T_w is given by

$$\rho_w c_{pw}\frac{\partial T_w}{\partial t} = 2\pi R_i h_w(T_g - T_w) - 2\pi R_o h_o(T_w - T_o)$$

Initial Condition

$$\text{at } t < 0,\ T_w = T_{w,0} \qquad \text{eq 2.13}$$

where T_o is the surrounding temperature, h_w and h_o are the heat transfer coefficients between the wall and fluid and between the wall and the surrounding, respectively. Axial conductivity in the canister wall is neglected since the area of heat transfer from the fluid to wall and from the wall to fluid is an order of magnitude larger than the area in the axial direction. This is analogous to heat conduction in a slab.

2.2.1.6 Momentum Equation

Bulk resistance to the fluid flow through a solid matrix, as compared to the resistance at and near the surfaces confining the solid matrix, was measured by Darcy (1956). The following equation characterizes the pressure drop in a packed column:

$$-\frac{dP}{dx}=\frac{\mu}{K}u \quad , \qquad \text{eq 2.14}$$

where K is the permeability of the solid and μ is the fluid viscosity. This empirical slug-flow equation, which is suitable for slip flow and creeping flow, is valid when the Reynolds number, based on particle diameter, is close to one.

For higher Reynolds number, cross flow within the porous media creates additional pressure drop. For this regime the Ergun equation could be used to estimate the pressure drop, see Ergun (1952) and Beavers (1969):

$$\frac{dP}{dx}=-\frac{\mu}{K}u-\rho C u^2 \quad , \qquad \text{eq 2.15}$$

where C is the inertial coefficient.

The Darcian terms contain the empirical coefficients K and C, which are given by relations developed by Ergun (1952) for flow in a packed bed:

$$K=\frac{d_p{}^2\varepsilon^3}{150(1-\varepsilon)^2} \quad , \qquad \text{eq 2.16}$$

and

$$C=\frac{1.75(1-\varepsilon)}{d_p\varepsilon^3} \quad , \qquad \text{eq 2.17}$$

where d_p is particle diameter.

Equations 2.10 and 2.15 are used to compute the total pressure and the velocity respectively in the axial direction. This was found to give better convergence than using them in the reverse order.

2.2.2 Solid Phase Transport Equation

Mass transfer of solute from bulk gas to sorbed state is driven by equilibrium isotherms. The mass balance equation inside the sorbent material depends on the structure of the pellet. Monodisperse pore-diffusion and bidisperse pore-diffusion are the two models used to predict the mass transport within the pellet. The schematics for the two types of sorbents are illustrated in Figures 2-1 and 2-2. In general, the mass transfer mechanism of an adsorption process includes four steps: fluid-film transfer, pore diffusion, surface adhesion, and surface diffusion. Since the surface adhesion rate approximates the order of the collision frequency of the gas on the solid surface, which is much greater than for the transport processes, equilibrium is assumed to be reached instantaneously at the interfaces (Yang, 1987). Adsorbates initially transfer from the bulk gas stream through an external film around the particles and reach the external surface of the particles. Molecules of adsorbate then diffuse into the pores of the particles, are adsorbed on the active sites, and then diffuse along the surface. The film transfer and pore diffusion are treated as sequential steps, since no accumulation of adsorbates are allowed; in other words, conservation of mass applies. However, pore and surface diffusion generally take place in parallel. Any combination of the resulting three steps can constitute the rate-controlling mechanism. The external mass transfer rate is defined in terms of an effective transfer coefficient (k_{fi}) as a linear driving force equation.

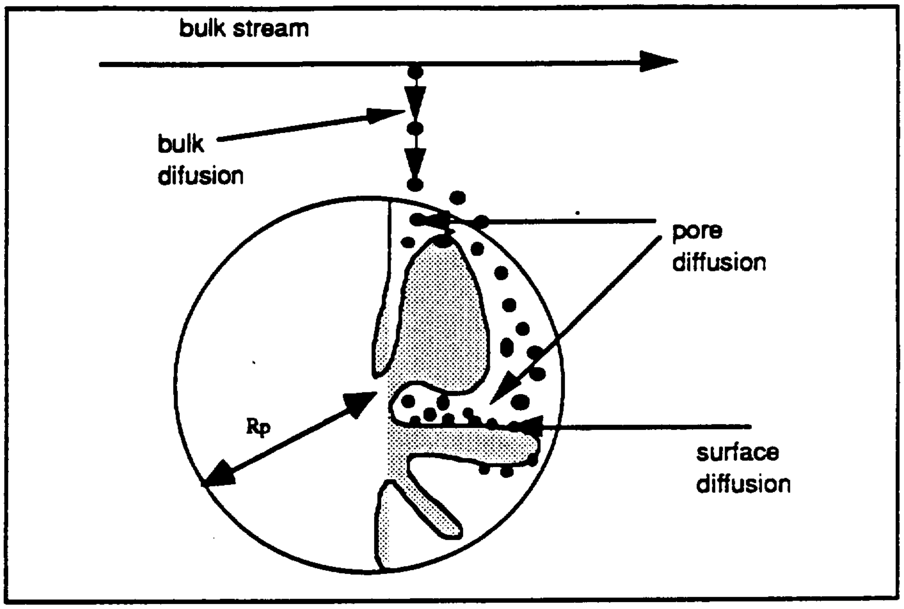


FIGURE 2-1 INTRAPARTICLE MECHANISM OF DIFFUSION FOR MONODISPERSE SORBENT

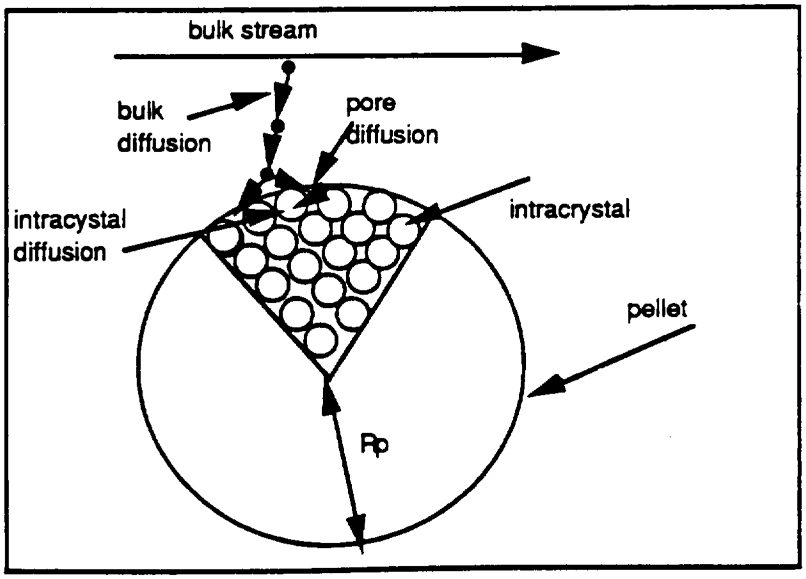


FIGURE 2-2 INTRAPARTICLE MECHANISMS OF BIDISPERSE-PORE SORBENT

$$N_{fi} = -k_{fi}(C_{bi} - C_{surf,i}) \qquad \text{eq 2.18}$$

For the low Reynolds number region covered in the present study, the correlation of Petrovic and Thodos (1968) is used:

$$k_{fi} = \frac{0.357}{\varepsilon}(N_{Re})^{0.64}(N_{sc,i})^{0.33}\frac{D_{mi}}{2R_p}, \qquad \text{eq 2.19}$$

for $3 < N_{Re} < 2000$. N_{Re} is the Reynolds number and N_{Sc} is Schmidt number.

Molecular diffusion of a component in a mixture is described by the Stefan-Maxwell equation. For single component diffusion in a mixture, however, the diffusion coefficient D_{mi} is approximately related to the binary coefficients by the following relationship (Bird et al., 1960)

$$D_{mi} = \frac{1 - y_i}{\sum_{j=1}^{n} \frac{y_j}{D_{i,j}}} \qquad \text{eq 2.20}$$

For binary mixtures at low pressure, $D_{i,j}$ can be estimated by the following equation, (Slattery and Bird, 1958):

$$D_{AB} = \frac{1}{p}(P_{cA}P_{cB})^{1/3}(T_{cA}T_{cB})^{5/12}\left(1/M_A + 1/M_B\right)^{1/2} a\left(T/\sqrt{T_{cA}T_{cB}}\right)^{b}$$

For nonpolar gaspairs:

$a = 2.745x10^{-4}$

$b = 1.823$

For H_2O with a nonpolar gas: eq 2.21

$a = 3.640x10^{-4}$

$b = 2.334$

The Knudsen diffusion coefficient can be estimated by the following equation (Satterfield, 1980):

$$D_{ki} = 4.67x10^{-4}\frac{r_e}{\tau_p}\left(\frac{T}{M_i}\right)^{0.5}, \qquad \text{eq 2.22}$$

where r_e is the mean diameter of the pore size and τ_p the pore tortuosity factor and is approximately between 1-4 (Satterfield, 1980).

The surface diffusion is an activated process which can be expressed by an Arrhenius-type expression (Smith, 1981):

$$D_{si} = D_{so,i}\exp\left(\frac{-E_{si}}{RT}\right) ,$$

eq 2.23

where E_{si} is the activation energy for the surface diffusion and $D_{so,i}$ is pre-exponential factor.

2.2.3 Monodisperse Pore Diffusion Model

The pore structure of pellets are simulated by this model are monodisperse. Sorbents in this category include activated carbon, alumina, and silica gel. The mass balance for component i inside the pores of a spherical particle at the axial location x in the bed yield:

$$\alpha\frac{\partial C_{ip}}{\partial t} + \frac{1}{r_p^2}\frac{\partial(r_p^2 N_{ir})}{\partial r_p} + \frac{\partial q_i}{\partial t} = 0 ,$$

eq 2.24

where α is the intraparticle void fraction, N_{ir} is the flux of component i, r_p is particle radius, and q_i is the amount adsorbed on the solid phase.

Equation 2.24 can be simplified by employing the following particle volume-averaged quantities:

$$\bar{C}_{ip} = \frac{3}{R_p^3}\int C_{ip} r_p^2 dr ,$$

eq 2.25

and

$$\bar{q}_{ip} = \frac{3}{R_p^3}\int q_{ip} r_p^2 dr .$$

eq 2.26

By integrating equation 2.24 with respect to r_p, using the volume averaged quantities and noting that $N_{ir}=0$ at $r=0$, the result is

$$\alpha\frac{\partial \bar{C}_{ip}}{\partial t} + \frac{3}{R_p}N_i^s + \frac{\partial \bar{q}_i}{\partial t} = 0 .$$

eq 2.27

To further simplify the model, parabolic concentration profile within the particle is assumed (Tsai, 1983), thus

$$C_{ip} = E_i + F_i r^2 ,$$

eq 2.28

where the constant E and F can be evaluated by using the surface conditions:

$$C_{ip} = C_{ip}^s \text{ at } r = R_p ,$$

eq 2.29

and

$$E_i = C_{ip}^s - F_i R_p^2 ,$$

eq 2.30

where C_{ip}^s is the concentration of i at the surface. By integrating equation 2.28 and using volume averaged quantity:

$$F_i = (C^s_{ip} - \bar{C}_{ip})\frac{5}{2R_p^2}. \qquad \text{eq 2.31}$$

Therefore, by inserting eq 2.31 and 2.30 into eq 2.28 and taking the derivative with respect to r at the surface of particle:

$$(\frac{\partial C_{ip}}{\partial r})_{r=R_p} = (C^s_{ip} - \bar{C}_{ip})5/R_p \qquad \text{eq 2.32}$$

The flux N_{ir} is the combined pore and solid diffusions. Pore diffusion is dominated by Knudsen diffusion with a constant diffusivity, D_{ik}. Surface diffusion D_s obeys Fick's law. Thus,

$$N_{ir} = -D_{ki}\frac{\partial C_{ip}}{\partial r} - D_{si}\frac{\partial q_i}{\partial r} \qquad \text{eq 2.33}$$

A local equilibrium can be assumed between the gas phase in the pore and the solid phase. Since the adsorbate concentrations are dilute, the solid phase concentration of each adsorbate may be assumed to be independent of the gas phase concentration of the other adsorbate. Therefore, the total flux of the i^{th} component at the pore mouth of the particle is

$$N_{ir} = -D_{ki}\frac{\partial C_{pi}}{\partial r}|_{r=R_p} - D_{si}\sum_j^n \frac{\bar{q}_i}{\partial C_{pj}}\frac{C_{pj}}{\partial r}|_{r=R_p} \quad i = 1,2,3\ldots n \qquad \text{eq 2.34}$$

Assuming gas-solid equilibrium within the pellet ,

$$\frac{\partial \bar{q}_i}{\partial C_p} = \frac{\partial \dot{q}_i}{\partial C_p}. \qquad \text{eq 2.35}$$

The time derivative of $\bar{q}_i$, is a function of T, and gas concentration in pore side of the pellet, C_{pj},

$$\frac{\partial \bar{q}_i}{\partial t} = \frac{\partial \bar{q}_i}{\partial T}\frac{\partial T}{\partial t} + \sum_{j=1}^{n}\frac{\partial \bar{q}_i}{\partial C_i}\frac{\partial C_i}{\partial t} \quad j = 1,2\ldots n \qquad \text{eq 2.36}$$

Equations 2.32-2.36 are substituted into equation 2.27, to obtain the final rate equation for monodisperse pellet is

$$(\varepsilon + \frac{\partial \bar{q}_i}{\partial \bar{C}_{ip}})\frac{\partial \bar{C}_{pi}}{\partial t} = \frac{15D_{si}}{R_p^2}\sum_j^n \frac{\partial \dot{q}_i}{\partial C_{pj}}(C^s_{pj} - \bar{C}_{pj}) + \frac{15D_{ki}}{R_p^2}(C^s_{pi} - \bar{C}_{pi})$$

$$- \frac{\partial \bar{q}_i}{\partial T_s} - \sum_{j=i}^{n}\frac{\partial \bar{q}_i}{\partial C_{pj}}\frac{\partial \bar{C}_{pj}}{\partial t} \qquad \text{eq 2.37}$$

2.2.4 Bidisperse Pore Diffusion Model for Zeolite

Zeolite sorbents consist of crystals, in the size range of 1-9 µm, which are pelletized with a small amount of binder. Diffusion rate is controlled by the crystals in the zeolite sorbents. The mass balance within a spherical crystal is described as

$$\frac{\partial q_i}{\partial t} = \frac{D_e}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial q_i}{\partial r}\right). \qquad \text{eq 2.38}$$

The rate of change of $\bar{q}_i$, the volume average q, is

$$\frac{\partial \bar{q}_i}{\partial t} = \frac{3}{R_p^3}\int\frac{\partial q_i}{\partial t}r_p{}^2 dr \qquad \text{eq 2.39}$$

which results in

$$\frac{\partial \bar{q}_i}{\partial t} = \frac{3D_e}{R_p}\left(\frac{\partial q_i}{\partial r_p}\right)_{r_p=R_p}. \qquad \text{eq 2.40}$$

As before, assuming a parabolic distribution

$$q_i = E_i + F_i r_p{}^2 \qquad \text{eq 2.41}$$

and

$$q_i = q_i^s \text{ at } r_p = R_p, \qquad \text{eq 2.42}$$

using surface condition

$$E_i = q_i^s - F_i R_p^2 \qquad \text{eq 2.43}$$

by integrating equation 2.41 and using the volume average

$$F_i = (q_i^s - \bar{q}_i)\frac{5}{2R_p^2} \qquad \text{eq 2.44}$$

obtain

$$\left(\frac{\partial q_i}{\partial r}\right)_{r=R_p} = (q_i^s - \bar{q}_i)5/R_p, \qquad \text{eq 2.45}$$

and from equation 4.45 and 2.40,

$$\left(\frac{\partial \bar{q}_i}{\partial t}\right) = 15D_e/R_p{}^2(q_i^s - \bar{q}_i) \qquad \text{eq. 2.46}$$

Assuming the mass transfer resistance between the bulk flow of the bed and the solid is not the limiting resistance of gas diffusion into the solid, q_i^s can be replaced by $\dot{q}_i$ which is in equilibrium with the bulk flow concentration. Thus,

$$\frac{\partial \bar{q}_i}{\partial t} = 15D_e/R_p{}^2(q_i^{\bullet} - \bar{q}_i). \qquad \text{eq 2.47}$$

This equation is what Glueckauf (1955) suggested. Eq 4.47 is the basis for the Linear Driving Force approximation model,

$$\frac{\partial \bar{q}_i}{\partial t} = k_{ef} a_s (q_i^* - \bar{q}_i) \qquad \text{eq 2.48}$$

where k_{ef} may be obtained by experimental procedure and a_s is the interfacial surface area.

2.2.5 Two Dimensional Adsorption Mathematical Model

For a packed bed the porosity varies with distance from the wall. Near the wall the porosity is higher than the bulk of the bed. This increases the permeability. A few particles away from the wall, the porosity equals the free stream value (Benenati, 1962; Roblee, 1958). As a consequence of the porosity increase in the vicinity of the wall, the velocity of the flow parallel to the wall increases as the wall is approached and goes through a maximum before it decreases to zero (to satisfy the no-slip condition). In general this leads to a net increase in flux, i.e., to the phenomenon called channeling (Nield and Bejan, 1992). During the experimental portion of this study (see Sect. 4.3.1), the effect of channeling was found to be significant for the intended final application of this model. As a result, the effect on momentum, energy, and material balances was considered to be important enough to require the development of a two-dimensional packed bed model.

2.2.5.1 Mass Equation

$$\frac{\partial C_i}{\partial t} = D_{eff,i,x}\frac{\partial^2 C_i}{\partial x^2} - \frac{\partial (uC_i)}{\partial x} + D_{eff,i,r}\frac{\partial}{r\partial r}\left(\frac{r\partial C_i}{\partial r}\right) - \frac{(1-\varepsilon)}{\varepsilon} r_i$$

Boundary / Initial Conditions

$$\text{for } t<0 \quad C_i = C_{i,0} \quad \text{for } 0 \le x \le L \text{ and } 0 \le r \le R$$

$$\text{for } t \ge 0 \quad C_i = C_{i,x0} \text{ at } x=0 \text{ and } 0 \le r \le R$$

$$\text{for } t \ge 0 \quad \frac{\partial C_i}{\partial x} = 0 \text{ at } x = L \text{ and } 0 \le r \le R \qquad \text{eq 2.49}$$

$$\text{for } t \ge 0 \quad \frac{\partial C_i}{\partial r} = 0 \text{ at } r = 0 \text{ and } r = R$$

2.2.5.2 Gas Phase Energy Equation

$$\rho_g C_{pg}\frac{\partial T_g}{\partial t} = k_{f,x}\frac{\partial^2 T_g}{\partial x^2} - \frac{\partial u T_g}{\partial x} + \frac{k_{f,r}}{r}\frac{\partial}{\partial r}\left(\frac{r\partial T_g}{\partial r}\right) - \frac{1-\varepsilon}{\varepsilon}h_s a_s (T_g - T_s)$$

Boundary / Initial Conditions

at $t < 0$, $T_g = T_{g,0}$ for $0 \le x \le L$ and $0 \le r \le R$

at $t \ge 0$, $T_g = T_{0,x}$ for $x = 0$ and $0 \le r \le R$

at $t \ge 0$, $\frac{\partial T_g}{\partial x} = 0$ for $x = L$ and $0 \le r \le R$

at $t \ge 0$, $\frac{\partial T_g}{\partial r} = 0$ at $r = 0$ for $0 \le x \le L$ eq 2.50

at $t \ge 0$, $k_{f,r}\frac{\partial T_g}{\partial r} = h_w(T_w - T_g)$ at $r = R$ for $0 \le x \le L$

2.2.5.3 Solid Phase Energy Equation

$$\rho_s C_{ps}\frac{\partial T_s}{\partial t} = k_{s,x}\frac{\partial^2 T_s}{\partial x^2} + \frac{k_{s,r}}{r}\frac{\partial}{\partial r}\left(\frac{r\partial T_s}{\partial r}\right) + h_s a_s (T_g - T_s) + \sum_{i=1}^{n}\Delta H_i \frac{\partial \bar{q}_i}{\partial t}$$

Boundry / Initial Condition

at $t < 0$, $T_s = T_{s,0}$ for $0 \le x \le L$ and $0 \le r \le R$

at $t \ge 0$, $T_s = T_{0,x}$ for $x = 0$ and $0 \le r \le R$

at $t \ge 0$, $\frac{\partial T_s}{\partial x} = 0$ for $x = L$ and $0 \le r \le R$ eq 2.51

at $t \ge 0$, $\frac{\partial T_s}{\partial r} = 0$ at $r = 0$ and $r = R$ for $0 \le x \le L$

Rate of adsorption, $\partial \bar{q}_i / \partial t$, can be substituted in the above equation either by 2.36 or 2.41. It depends on the adsorbent material. If it is monodisperse or bidisperse type of pellet equation 2.36 or 2.41 will be used, respectively.

2.2.5.4 Column Wall Energy Equation

$$\rho_w C_{pw}\frac{\partial T_w}{\partial t} = 2\pi R_i h_w (T_g - T_w) - 2\pi R_o h_o (T_w - T_o)$$

Initial Condition

at $t < 0$, $T_w = T_{w,0}$ eq 2.52

2.2.5.5 Bed Energy Equation Based On Effective Conductivity

$$(\varepsilon\rho_g c_{pg} + (1-\varepsilon)\rho_s c_{ps})\frac{\partial T}{\partial t} = \varepsilon k_{eff,x}\frac{\partial^2 T}{\partial x^2} + \varepsilon\frac{k_{eff,r}}{r}\frac{\partial}{\partial r}\left(\frac{r\partial T}{\partial r}\right) - \varepsilon u\rho_g c_{pg}\frac{\partial T}{\partial x}$$

$$+ (1-\varepsilon)\sum_{i=1}^{n}\Delta H_i\frac{\partial \bar{q}_i}{\partial t}$$

Boundary / Initial Condition

$$\text{at } t < 0,\ T = T_0 \text{ for } 0 \le x \le L \text{ and } 0 \le r \le R$$

$$\text{at } t \ge 0,\ T = T_{0,x} \text{ for } x = 0 \text{ and } 0 \le r \le R$$

$$\text{at } t \ge 0,\ \frac{\partial T}{\partial x} = 0 \text{ for } x = L \text{ and } 0 \le r \le R$$

$$\text{at } t \ge 0,\ \frac{\partial T}{\partial r} = 0 \text{ at } r = 0 \text{ for } 0 \le x \le L \qquad \text{eq 2.53}$$

$$\text{at } t \ge 0,\ k_{eff,r}\frac{\partial T}{\partial r} = h_w(T_w - T|_{r=R}) \text{ for } r = R$$

2.2.5.6 Momentum Equation

The governing momentum equation for fully developed flow in cylindrical beds is, from Vafai and Tien (1982).

$$\frac{\partial P}{\partial x} = -\rho_g C u^2 - \frac{\mu}{K}u + \frac{1}{r}\frac{\mu}{\varepsilon}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right),$$

Boundary Conditions

$$u = -\frac{K_\infty}{m}\frac{\partial P}{\partial x} \text{ at } r = 0 \text{ and } u = 0 \text{ at } r = R, \qquad \text{eq 2.54}$$

where ε is the bed porosity, and K and C are the permeability and inertial coefficient which depend on the porosity and the type of porous materials. In the above equation, the second term is the inertial effect which accounts for additional pressure drop resulting from interpore mixing found at higher Reynolds numbers (Vafai, 1981; Ergun, 1952). The third term is the Darcian force representing the pressure loss due to the presence of solid particles. The last term is the viscous shear force representing the resistance to the flow caused by sheer stress along the solid boundary. This term accounts for the no-slip boundary condition at the solid boundary. In this study the entrance effect is not considered since the flow is fully developed after one to two particle distance from the entrance (Vafai and Tien, 1981).

2.2.6 Porosity Variation

$$\varepsilon = \varepsilon_\infty[1 + a\exp(-by/d_p)] \qquad \text{eq 2.55}$$

where ε_∞ is the free-stream porosity, y is the distance from the wall, d is the particle diameter, and a is taken to be 1.4 (Nield and Bejan, 1992). b is an experimental parameter that depends on packing and particle size and it varies from 2 to 8.

The empirical coefficients K and C are given by the relations developed by Ergun (1952) for flow in a packed bed:

$$K = \frac{d_p^{\,2}\varepsilon^3}{150(1-\varepsilon)^2} \qquad \text{eq 2.56}$$

$$C = \frac{1.75(1-\varepsilon)}{d_p\varepsilon^3} \qquad \text{eq 2.57}$$

The variables C and K are both functions of the bed porosity and particle diameter, d_p. The porosity in a packed bed increases from the center of bed, free-stream porosity, to a maximum of one at the bed-wall boundary . This increase is confined to a few particle diameters from the wall (Benenati, 1962; Roblee, 1958).

In the above two dimensional equations, the term which represents the radial diffusion is

$$\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{r\partial C}{\partial r}\right), \qquad \text{eq 2.58}$$

where C varies with r. Carrying out the derivative

$$\frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial r^2} \qquad \text{eq 2.59}$$

At the center where r=0, the first term is not finite. But

$$\operatorname*{limit}_{r\to 0}\left(\frac{1}{r}\frac{\partial C}{\partial r}\right) = \frac{\partial^2 C}{\partial r^2} \qquad \text{eq 2.60}$$

by L'Hospital's rule. Therefore the term, eq 2.58, in 2-dimensional form for the center point is replaced by

$$\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{r\partial C}{\partial r}\right) = 2\frac{\partial^2 C}{\partial r^2} \qquad \text{eq 2.61}$$

Therefore, the diffusional term in the discretized forms of two dimensional PDE's at center grid is replaced by eq 2.61.

2.2.7 Calculation Of Thermal Conductivity For 2-D Flow

In this study two different equations were used to calculate the effective conductivity in the packed bed. One is based on the works of Kunii and Smith (1960), and the other one is based on the experimental work of Fahien (1954).

The effective thermal conductivity in the axial and radial direction, $k_{eff,x}k_{eff,r}$, are related as,

$$k_{eff,x} = k_o + k_{f,x}$$

$$k_{eff,r} = k_o + k_{f,r} \qquad \text{eq 2.62}$$

These conductivity's can be calculated by the theoretical equations presented below.

2.2.7.1 Stagnant Conductivity k_o

Kunii and Smith (1960) presented theoretical equations for estimating the stagnant conductivity, k_o. The stagnant conductivity can be found if k_s, solid thermal conductivity, is given .

$$\frac{k_e}{k_f} = \varepsilon + (1-\varepsilon)/\left[\phi + (2/3)(k_f/k_s)\right] \qquad \text{eq 2.63}$$

where k_f is the thermal conductivity of fluid and ϕ is the contribution of solid to solid heat transfer through fluid film around a contacting point of neighboring particles. ϕ is given by

$$\begin{aligned}
&\phi = \phi_2 + (\phi_1 - \phi_2)\left[(\varepsilon - .26)/.216\right] \\
&\text{for } .476 \geq \varepsilon \geq .26 \\
&\phi = \phi_1 \\
&\text{for } \varepsilon > .476 \qquad \text{eq 2.64} \\
&\phi = \phi_2 \\
&\text{for } \varepsilon < .26
\end{aligned}$$

where ϕ_1 and ϕ_2 are given in a schematic form, and are being interpolated linearly in tabular form in the computer program.

2.2.7.2 Effective Radial Conductivity kf,r

Packed bed thermal conductivity in the radial direction is given by Baron (1952) as

$$C_{pf}\rho_f u / k_{f,r} = N_{Pe_H} = 8 - 10 \qquad \text{eq 2.65}$$

where N_{Pe_H} is Peclet number, therefore the effective thermal conductivity in the radial direction would be (Yagi and Kunii, 1952).

$$\begin{aligned}
&\frac{k_{eff,r}}{k_f} = \frac{k_e}{k_f} + (\alpha\beta) N_{Re_p} N_{Pr} \\
&\text{where } (\alpha\beta) = 1/N_{Pe_H} = .1 \text{ to } .125 \qquad \text{eq 2.66}
\end{aligned}$$

2.2.7.3 Effective Axial Conductivity keff,x

A similar equation can be derived for effective thermal conductivity in the axial direction, (Yagi, Kunii, and Wakao, 1960).

$$\begin{aligned}
&\frac{k_{eff,x}}{k_f} = \frac{k_e}{k_f} + \lambda N_{Re_p} N_{Pr} \\
&\text{where } \lambda = .5 \text{ to } 1.0 \qquad \text{eq 2.67}
\end{aligned}$$

In the above equations, the radial and axial conductivity are the combination of two terms. The first term is the stagnation conductivity, which varies from a bulk conductivity to fluid conductivity with distance from the center to the column wall. Therefore, it depends on the porosity variation which also is a function of bed parameters. The second term is due to the dynamic or dispersion conductivity which incorporates the mixing caused by flow through the particles. Incorporating the effects of porosity variation into the effective conductivity, the effective conductivity reduces to (Hunt, 1987)

$$\frac{k_{eff,r}}{k_f} = (1 + a' \exp(-bR))\frac{k_o}{k_f} + (\alpha\beta)\frac{l(R)}{d} N_{Re_p} N_{Pr}$$

where $(\alpha\beta) = 1 / N_{Pe_H} = .1 \text{ to } .15$ eq 2.68

and in the axial direction

$$\frac{k_{eff,x}}{k_f} = (1 + a' \exp(-bR))\frac{k_o}{k_f} + \lambda\frac{l(R)}{d} N_{Re_p} N_{Pr}$$

where $\lambda = .5 \text{ to } 1.0$ eq 2.69

where a' is chosen such that k_o/k_f equals one at the wall as velocity becomes zero. The variation of dispersion, mixing length, is

$$\frac{l(R)}{d} = R_t / R_t - r / R_t \text{ for } R_t - r \le 1$$ eq 2.70

where R_t is tube diameter. An expression similar to porosity variation was used to predict the mixing length variation (Hunt, 1987).

2.2.8 Calculation Of Mass Diffusivity For 2-D Flow

A similar theoretical approach can be taken for the calculation of mass diffusivity in the radial and axial direction as that used for thermal diffusivity in the previous section.

2.2.8.1 Effective Diffusivity

Effective diffusivity follows the same expression as in thermal conductivity:

$$D_{eff,x} = D_o + D_{f,x}$$
$$D_{eff,r} = D_o + D_{f,r}$$ eq 2.71

2.2.8.2 Effective Radial Diffusivity $D_{f,r}$

The effective diffusivity in the radial direction by analogy to heat transfer is:

$$\frac{D_{eff,r}}{D_f} = \frac{D_o}{D_f} + (\alpha\beta) N_{Re_p} N_{SC},$$

where $(\alpha\beta) = 1 / N_{Pe_H} = .1 \text{ to } .125.$ eq 2.72

2.2.8.3 Effective Axial diffusivity $D_{eff,x}$

A similar equation can be derived for effective diffusivity in the axial direction;

$$\frac{D_{eff,x}}{D_f} = \frac{D_o}{D_f} + \lambda N_{Re_p} N_{SC},$$

where $\lambda = .5 \text{ to } 1.0.$ eq 2.73

Incorporating the effects of porosity variation into the effective diffusivity, the effective diffusivity reduces to (Hunt, 1987),

$$\frac{D_{eff,r}}{D_f} = (1 + a' \exp(-bR))\frac{D_{\cdot}}{D_f} + (\alpha\beta)\frac{l(R)}{d} N_{Re_p} N_{Sc},$$

where $(\alpha\beta) = 1 / N_{Pe_R} = .1$ to $.125$. eq 2.74

and in the axial direction,

$$\frac{D_{eff,x}}{D_f} = (1 + a' \exp(-bR))\frac{D_{\cdot}}{D_f} + \lambda\frac{l(R)}{d} N_{Re_p} N_{Sc},$$

where $\lambda = .5$ to 1.0. eq 2.75

2.2.9 Effective Radial Thermal Conductivity Based on Fahien Equations

In contrast to weak effect of mass diffusion on radial mass fraction, the thermal conductivity profile has a strong effect on both temperature and mass adsorption. A thermal conductivity profile for different ratio of d_p/d_t was approximated by Fahien (1954),

$$k_{eff,r} = < k > (k_0^{\cdot} + 3(k_M^{\cdot} - k_0^{\cdot})\frac{r^2}{{r_M}^2} + 2(k_M^{\cdot} - k_0^{\cdot})\frac{r^3}{{r_M}^3}) \quad 0 \le r \le r_M$$

$$k_{eff,r} = < k > (k_M^{\cdot} - (k_M^{\cdot} - k_W^{\cdot})\frac{r - r_M}{1 - r_M}) \quad r > r_M$$

$$k_M^{\cdot} = \frac{3.0 - .9 k_0^{\cdot} {r_M}^2 - k_W^{\cdot} \frac{{r_M}^3 - 3{r_M}^2 + 2}{1 - r_M}}{1 + r_M + .1{r_M}^2}$$

eq 2.76

$$r_M = 1 - \frac{2}{\alpha}$$

$$\alpha = d_t / d_p$$

eq 2.77

where $k_0^{\cdot}$ is the effective radial conductivity at the center of the column wall, $k_M^{\cdot}$ is the maximum effective thermal conductivity, $k_W^{\cdot}$ is the effective thermal conductivity near the wall, $< k >$ is the average effective thermal conductivity, and r_M is the location of maximum in conductivity profile. These conductivity's are obtained by Argo and Smith (1953) equation using the void fraction values as a function of radial position. According to Argo and Smith

$$k_{eff} = \varepsilon\left[k_g + \frac{d_p C_{pg} G}{N_{Pe}\varepsilon} + 4\left(\frac{\sigma}{2 - \sigma}\right)d_p(0.173)\left({T_a}^3 / 100^4\right)\right] + (1 - \varepsilon)\frac{hk_s d_p}{2k_s + hd_p}$$

eq 2.78

where σ is the emissivity of the solid particle, T_a is the average temperature. In the above equation,

$$h = h_c + h_r + h_p$$

eq 2.79

$$h_c = 1.95 C_{pg} G N_{Pr}^{-2/3} N_{Re}^{-.51} \qquad N_{Re} < 350$$

$$h_c = 1.06 C_{pg} G N_{Pr}^{-2/3} N_{Re}^{-.41} \qquad N_{Re} > 350 \qquad \text{eq 2.80}$$

$$h_r = \frac{k_r\left(2k_s + hd_p\right)}{d_p k_s}$$

$$k_r = 4\left(\frac{\sigma}{2-\sigma}\right) d_p(.173)\left(\frac{T_s^{\,3}}{100^4}\right) \qquad \text{eq 2.81}$$

$$h_p = \frac{k_p\left(2k_s + hd_p\right)}{d_p k_s}$$

$$\log_{10} k_p = 1.76 + .0129 k_s / \sigma \qquad \text{eq 2.82}$$

2.2.10 Effective Radial Diffusivity Based on Fahien Equations

$$D_{eff,r} = < D > (D_0^* + 3\left(D_M^* - D_0^*\right)\frac{r^2}{r_M^{\,2}} + 2\left(D_M^* - D_0^*\right)\frac{r^3}{r_M^{\,3}}) \quad 0 \le r \le r_M$$

$$D_{eff,r} = < D > (D_M^* \frac{r - r_M}{1 - r_M}) \quad r > r_M \qquad \text{eq 2.83}$$

$$D_M^* = \frac{3.0 - .9 D_0^* r_M^{\,2}}{1 + r_M + .1 r_M^{\,2}}$$

D_0^* is obtained from work of Fahien and Smith (1955) to be

$$D_0^* = 9/8 V_0\left(1 + 4.85\alpha^{-2}\right) \qquad \text{eq 2.84}$$

where V_0 is velocity at the center of packed bed.

2.2.11 Heat Transfer Coefficient Between Wall Surface and Packed Bed, h_w

The effective thermal conductivity in the wall layer of thickness Rp=dp/2, k_{ew}, is defined and h_w is considered as a correction factor based on the difference $k_{eff,r}$ and k_{ew} (Kunii and Suzuki, 1966).

$$\frac{k_{ew}}{k_f} = \frac{k_{ew0}}{k_f} + \frac{1}{\dfrac{1}{\alpha_w N_{Pr} N_{Re}} + \dfrac{1}{h_w^* d_p / k_f}} \qquad \text{eq 2.85}$$

where α_w denotes the contribution of fluid mixing in the wall layer and is taken as 0.2. h_w^* represents the heat transfer coefficient of the thermal boundary layer which develops on the wall surface. This becomes dominant at high $N_{Re,p}$ and is given by Blasius type equation as (Suzuki, 1990),

$$h^{\bullet}_{w} d_p / k_f = C\left(N_{Pr}{}^{1/3} N_{Re,p}{}^{3/4}\right) \qquad \text{eq 2.86}$$

where C is an experimental coefficient with value of 0.1 to 0.2 (Kunii, Suzuki and Ono, 1968). k_{ew0} is obtained by the following equation similarly to eq (2.65).

$$\frac{k_{ew0}}{k_f} = \varepsilon_w + (1-\varepsilon_w)/\left[\phi_w + (2/3)(k_f / k_s)\right] \qquad \text{eq 2.87}$$

where ε_w denotes void fraction in the wall layer of about 0.7.

SECTION 3

NUMERICAL ANALYSIS

The solution of nonequilibria adsorption/desorption problems must be solved numerically. For an *n* component mixture, the numerical model would involve the solution of several coupled differential equations: *n-1* mass balances equations, n mass balances of rate equations, n equilibrium isotherms, one total mass balance, one momentum equation, one heat balance of fluid flow, one solid phase heat balance, and one heat balance for the wall.

The finite difference technique, which is mostly used for processes with varying boundary and initial conditions, is a convenient method for solving the coupled partial differential equations (PDE's).

In this work, the PDE's were discretized by first or second order differences in time and spatial dimensions. The set of discretized finite difference equations were solved simultaneously by an implicit method. At first, based on the stiffness and the sharpness of the momentum equation and the mass and heat transfer fronts, two different implicit numerical methods were used to simulate the dynamic behavior of the packed bed. For adsorption and flow-through desorption of one and two dimensions, the implicit-iterative method of under relaxation was used. For vacuum desorption, since the convergence of equations are difficult to achieve because of the steep gradient of pressure drop in the packed bed, a more rigorous method had to be used. The implicit method of Newman (1968, 1967) was used to simulate the vacuum desorption process. Later, during the development of the numerical computation, it appeared that the use of Newman method for all the processes reduced the required CPU time and resulted in more stable discretized PDE's.

3.1 The Discrete Numerical Equations

The spatial dimension was discretized using first or second order backward, forward or central differencing depending on the stability of PDE. For instance, having C as a variable, the spatial finite differences for three cases are:

$$
\begin{aligned}
\frac{\partial C^n}{\partial x} &= \frac{C_i^n - C_{i-1}^n}{\Delta x} + O(\Delta x) \quad \textbf{Backward difference} \\
\frac{\partial C^n}{\partial x} &= \frac{C_{i+1}^n - C_i^n}{\Delta x} + O(\Delta x) \quad \textbf{Forward difference} \\
\frac{\partial C^n}{\partial x} &= \frac{C_{i+1}^n - C_{i-1}^n}{2\Delta x} + O(\Delta x^2) \quad \textbf{Central difference} \\
\frac{\partial^2 C^n}{\partial x^2} &= \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta x^2} + O(\Delta x^2) \quad \textbf{Central difference}
\end{aligned}
\qquad \text{eq 3.1}
$$

The backward finite difference was used in time:

$$
\frac{\partial C_i^{n+1}}{\partial t} = \frac{C_i^{n+1} - C_i^n}{\Delta t} + O(\Delta t) \qquad \text{eq 3.2}
$$

Where

n =the old time step
n+1 =the current time step being solved for
i =the node position

3.1.1 Discretized Spatial Finite Difference With Unequal Grid Size

Figure 3.1 shows a situation in which adjacent grids are not the same size. This type of grid spacing was used out in the radial direction in solving the PDE's. To establish the finite difference for the first and second derivatives, Taylor series expansion in powers of ($f\Delta x$) and ($-\Delta x$) were used, giving

$$C_A = C_{i,j} + (f\Delta x)\left.\frac{\partial C}{\partial x}\right|_{i,j} + \frac{(f\Delta x)^2}{2!}\left.\frac{\partial^2 C}{\partial x^2}\right|_{i,j} + O(f\Delta x)^3$$

$$C_{i-1,j} = C_{i,j} - (\Delta x)\left.\frac{\partial C}{\partial x}\right|_{i,j} + \frac{(\Delta x)^2}{2!}\left.\frac{\partial^2 C}{\partial x^2}\right|_{i,j} + O(\Delta x)^3 \qquad \text{eq 3.3}$$

Eliminating $\left.\frac{\partial C}{\partial x}\right|_{i,j}$ between these equations,

$$\frac{\partial^2 C}{\partial x^2} = \frac{2}{(\Delta x)^2}\left[\frac{C_A}{f(1+f)} + \frac{C_{i-1,j}}{(1+f)} - \frac{C_{i,j}}{f}\right] \qquad \text{eq 3.4}$$

If $\left.\frac{\partial T}{\partial x}\right|_{i,j}$ is eliminated between eqs 3.3 and 3.4 then

$$\frac{\partial C}{\partial x} = \frac{1}{(\Delta x)}\left[\frac{C_A}{f(1+f)} - \frac{f}{(1+f)}C_{i-1,j} - \frac{(1-f)}{f}C_{i,j}\right] \qquad \text{eq 3.5}$$

with an error of the order of $(\Delta x)^2$.

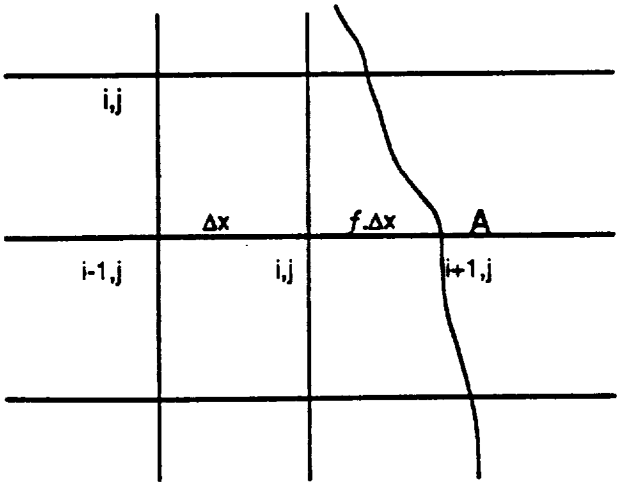


FIGURE 3-1 SHOWS THE GRIDS WITH DIFFERENT SIZES

3.2 Discretized Partial Differential Equations (PDE's)

Using the approximated values of first and second order derivatives in PDE's, the discretized forms of PDE's for one-dimensional flow will be generated for numerical solution. Theses approximated equations will be solved by the methods mentioned above.

3.2.1 Discretized Component Mass Balance:

$$\frac{\partial C_i}{\partial t} = D_i \frac{\partial^2 C_i}{\partial x^2} - u\frac{\partial C_i}{\partial x} - \frac{1-\varepsilon}{\varepsilon}\frac{\partial \bar{q}}{\partial t} \qquad \text{eq 2.9}$$

Inserting the finite difference approximations:

$$\frac{C_{i,j}^{n+1} - C_{i,j}^{n}}{\Delta t} = D_i \frac{C_{i,j+1}^{n+1} - 2C_{i,j}^{n+1} + C_{i,j-1}^{n+1}}{(\Delta x)^2} - u_j^{n+1}\frac{C_{i,j}^{n+1} - C_{i,j-1}^{n+1}}{\Delta x} - \frac{1-\varepsilon}{\varepsilon} r_{i,j}^{n+1}$$

collecting $C_{i,j}^{n+1}$ terms

$$A = \frac{1}{\Delta t} + \frac{2D_i}{\Delta x^2} + \frac{u_j^{n+1}}{\Delta x}$$

solving for $C_{i,j}^{n+1}$ and applying under - relaxation method

$$C_{i,j}^{n+1} = C_{i,j}^{n} + \omega\left(-C_{i,j}^{n} + \frac{1}{A}\left(\frac{C_{i,j}^{n}}{\Delta t} + D_i\frac{C_{i,j+1}^{n+1} + C_{i,j-1}^{n+1}}{(\Delta x)^2} + u_j^{n+1}\frac{C_{i,j-1}^{n+1}}{\Delta x} - \frac{1-\varepsilon}{\varepsilon} r_{i,j}^{n+1}\right)\right) \qquad \text{eq 3.6}$$

3.2.2 Discretized Gas Phase Energy

$$\rho_g c_{pg}\frac{\partial T_g}{\partial t} = k_f\frac{\partial^2 T_g}{\partial x^2} - u\rho_g c_{pg}\frac{\partial T_g}{\partial x} + \frac{1-\varepsilon}{\varepsilon}h_s a_s(T_s - T_g) - \frac{4h_w}{\varepsilon d}(T_g - T_w) \qquad \text{eq 2.11}$$

Inserting the finite difference approximation

$$\rho_g c_{pg}\frac{T_{g,j}^{n+1} - T_{g,j}^{n}}{\Delta t} = k_f\frac{T_{g,j+1}^{n+1} - 2T_{g,j}^{n+1} + T_{g,j-1}^{n+1}}{(\Delta x)^2} - u_j^{n+1}\rho_g c_{pg}\frac{T_{g,j}^{n+1} - T_{g,j-1}^{n+1}}{\Delta x} +$$

$$\frac{1-\varepsilon}{\varepsilon}h_s a_s(T_{s,j}^{n+1} - T_{g,j}^{n+1}) - \frac{4h_w}{\varepsilon d}(T_{g,j}^{n+1} - T_{w,j}^{n+1}) \qquad \text{eq 3.7}$$

collecting $T_{g,j}^{n+1}$ terms

$$A = \frac{\rho_g c_{pg}}{\Delta t} + \frac{2k_f}{(\Delta x)^2} + \frac{u_j^{n+1}\rho_g c_{pg}}{\Delta x} + h_s a_s\frac{1-\varepsilon}{\varepsilon}T_{g,j}^{n+1} + \frac{4h_w}{\varepsilon d}T_{g,j}^{n+1}$$

solving for $C_{i,j}^{n+1}$ and applying under - relaxation method

$$T_{g,j}^{n+1} = T_{g,j}^{n} + \omega\left(-T_{g,j}^{n} + \frac{1}{A}\left(\begin{array}{c}\rho_g c_{pg}\dfrac{T_{g,j}^{n}}{\Delta t} + k_f\dfrac{T_{g,j+1}^{n+1} + T_{g,j-1}^{n+1}}{(\Delta x)^2} + u_j^{n+1}\rho_g c_{pg}\dfrac{T_{g,j-1}^{n+1}}{\Delta x} + \\ h_s a_s\dfrac{1-\varepsilon}{\varepsilon}T_{s,j}^{n+1} + \dfrac{4h_w}{\varepsilon d}T_{w,j}^{n+1}\end{array}\right)\right)$$

3.2.3 Discretized Solid Phase Energy Equation

$$\rho_s c_{ps}\frac{\partial T_s}{\partial t} = k_s\frac{\partial^2 T_s}{\partial x^2} + h_s a_s (T_g - T_s) - \sum_{i=1}^{n}\Delta H_i \frac{\partial \bar{q}_i}{\partial t}$$

eq 2.12

$$\rho_s c_{ps}\frac{T_{s,j}^{n+1} - T_{s,j}^{n}}{\Delta t} = k_s\frac{T_{s,j+1}^{n+1} - 2T_{s,j}^{n+1} + T_{s,j-1}^{n+1}}{(\Delta x)^2} + h_s a_s (T_{g,j}^{n+1} - T_{s,j}^{n+1}) - \sum_{i=1}^{n}\Delta H_i r_i^{n+1}$$

collecting $T_{s,j}^{n+1}$ terms

$$A = \frac{\rho_s c_{ps}}{\Delta t} + \frac{2k_s}{(\Delta x)^2} + h_s a_s T_{s,j}^{n+1}$$

solving for $C_{i,j}^{n+1}$ and applying under - relaxation method

eq 3.8

$$T_{s,j}^{n+1} = T_{s,j}^{n} + \omega\left(-T_{s,j}^{n} + \frac{1}{A}\left(\rho_s c_{ps}\frac{T_{s,j}^{n}}{\Delta t} + k_s\frac{T_{s,j+1}^{n+1} + T_{s,j-1}^{n+1}}{(\Delta x)^2} + h_s a_s T_{g,j}^{n+1} - \sum_{i=1}^{n}\Delta H_i r_i^{n+1}\right)\right)$$

3.2.4 Discretized Column Wall Equation

$$\rho_w c_{pw}\frac{\partial T_w}{\partial t} = 2\pi R_i h_w (T_g - T_w) - 2\pi R.h.(T_w - T.)$$

eq 2.13

$$\rho_w c_{pw}\frac{T_{w,j}^{n+1} - T_{w,j}^{n}}{\Delta t} = 2\pi R_i h_w (T_{g,j}^{n+1} - T_{w,j}^{n+1}) - 2\pi R.h.(T_{w,j}^{n+1} - T_{.,j}^{n+1})$$

Collecting $T_{w,j}^{n+1}$ terms

$$A = \frac{\rho_w c_{pw}}{\Delta t} + 2\pi R_i h_w T_{w,j}^{n+1} + 2\pi R.h.T_{w,j}^{n+1}$$

$$T_{w,j}^{n+1} = T_{w,j}^{n} + \omega\left(-T_{w,j}^{n} + \frac{1}{A}\left(\rho_w c_{pw}\frac{T_{w,j}^{n}}{\Delta t} + 2\pi R_i h_w T_{g,j}^{n+1} + 2\pi R.h.T_{.,j}^{n+1}\right)\right)$$

eq 3.9

3.2.5 Discretized Total Material Equation

The total pressure equation was solved numerically using Newman's method.

$$\frac{\partial P}{\partial t} = D_l\frac{\partial^2 P}{\partial x^2} - u\frac{\partial P}{\partial x} - P\frac{\partial u}{\partial x} + \frac{P}{T_g}\left(\frac{\partial T}{\partial t} - D_l\frac{\partial^2 T}{\partial x^2} + u\frac{\partial T}{\partial x}\right) - RT\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n}\frac{\partial \bar{q}_i}{\partial t}$$

eq 2.10

$$F_1 = -\frac{P_j^{n+1} - P_j^{n}}{\Delta t} + D_1\frac{P_{j+1}^{n+1} - 2P_j^{n+1} + P_{j-1}^{n+1}}{(\Delta x)^2} - u\frac{P_j^{n+1} - P_{j-1}^{n+1}}{\Delta x} - P\frac{u_j^{n+1} - u_{j-1}^{n+1}}{\Delta x} +$$

eq 3.10

$$\frac{p_j^{n+1}}{T_{g,j}^{n+1}}\left(\frac{T_{g,j}^{n+1} - P_{g,j}^{n}}{\Delta t} - D_1\frac{T_{g,j+1}^{n+1} - 2T_{g,j}^{n+1} + T_{g,j-1}^{n+1}}{(\Delta x)^2} + u\frac{T_{g,j}^{n+1} - T_{g,j-1}^{n+1}}{\Delta x}\right) - RT_{g,j}^{n+1}\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n} r_i^{n+1}$$

3.2.6 Discretized Momentum Equation

The momentum equation was solved numerically by Newman's method.

$$\frac{dP}{dx} = -\frac{\mu}{K} u_D - \rho C u^2 \qquad \text{eq 2.15}$$

$$F_2 = \frac{P_j^{n+1} - P_j^n}{\Delta t} + \frac{\mu}{K} u_j^{n+1} + \rho C u_j^{n+1^2} \qquad \text{eq. 3.11}$$

3.2.7 Discretized Boundary Condition

In this study most of the boundary conditions are the homogeneous boundary type. In this type of boundary condition the normal derivative of variables such as temperature, velocity, and concentration at the boundary surface vanishes,

$$\frac{\partial C}{\partial x} = 0 \qquad \text{eq 3.13}$$

where C is any variable.

The boundary conditions for all the PDE's were discretized by extending the region at boundaries a distance Δx so that there is an imaginary y node at j-1 or at j+1 at the inlet or outlet of the column, respectively. Equation 3.13 can be discretized as,

$$\frac{C_{j+1} - C_{j-1}}{2\Delta x} = 0 \qquad \text{eq 3.14}$$

This equation can be solved for C_{j+1} or C_{j-1} and to be substituted in the original PDE equation for the imaginary node.

A second type boundary condition, where there is a linear combination of the variable and its normal derivative at the boundary surface, such as boundary surface at the column wall,

$$k\frac{\partial T}{\partial x} + hT = hT_\infty \qquad \text{eq 3.15}$$

where k is conductivity, T is fluid temperature, h is connective coefficient, and T_∞ is the wall temperature. Eq 3.15 can be discretized as,

$$\frac{T_{j+1} - T_{j-1}}{2\Delta x} + hT_j = hT_\infty \qquad \text{eq 3.16}$$

This equation can be solved for T_{j-1} and to be substituted in the original PDE equation for the imaginary node.

3.3 Numerical Approach

For each interior grid point, a set of algebraic equations as given above results when finite differences are used to approximate the PDE's. These system of simultaneous equations may be written in a general matrix form as

$$[\mathbf{A}][\phi]=[\mathbf{F}]$$

eq 3.17

where [A] is a square matrix, [ϕ] is a column vector containing unknown nodal values of ϕ, and [F] is the column vector containing the source and the prescribed boundary conditions. From linear algebra, we know that the unique solutions of the above equation exist when the determinant of [A] is nonzero.

An approach to the solution of eq. 3.17 is by using iterative methods. Iterative methods start from a guessed initial field, and sequentially improve the field by using successive iterations until eq 3.13 is satisfied. When the most recent values of the iterate are used, the Gauss-Seidel method results. Iteration can be enhanced by extrapolating the Gauss-Seidel improvement. This technique is called relaxation method (Gerald and Wheatley, 1984), for the case of a simple steady-state one-dimensional diffusion equation:

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\dot{Q}}{k}=0$$

the finite difference approximation will be

$$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}+\frac{\dot{Q}_{i,j}}{k}=0$$

using the Guess - Siedel method

$$\phi_i^{n+1}=\frac{1}{2}(\phi_{i+1}^{n}-\phi_{i-1}^{n+1}+\Delta x^2\frac{\dot{Q}_{i,j}}{k})$$

implimenting the relaxation method the above equation becomes

eq 3.18

$$\phi_i^{n+1}=\frac{\omega}{2}(\phi_{i+1}^{n}-\phi_{i-1}^{n+1}+\Delta x^2\frac{\dot{Q}_{i,j}}{k})+(1-\omega)\phi_i^{n}$$

where n indicates successive iteration, and ω is called the relaxation factor. Iterative convergence is achieved with $0< \omega <2$; $\omega =1$ correspond to Gauss-Seidel iteration; $0< \omega <1$; to successive under-relaxation (SUR); and $1< \omega <2$; to successive over-relaxation (SOR). A best, or optimum, value of ω can be found from numerical test or from analysis. In this study, since the equations are coupled and non linear, the value of ω was found by numerical test to be between .2 to .5 depending on the stiffness of the PDE.

In the case of vacuum desorption process, because of steep gradients in the pressure front and the velocity equation, the SOR method was not a suitable choice for solving the PDE's system. Instead, a different numerical method which is more stable and where the radius of convergence is higher was used. This method is lengthy and explained in detail by Newman (1968, 1967). In short, the set of PDE's are linearized by Newton method resulting in a system of tridiagonol matrix results which can be solved by any ordinary technique. The time of convergence for

Newman method is longer for a given time step but the grid points are three times larger and the time step is up to a order of magnitude larger than SOR.

3.4 Convergence And Stability Of Numerical Techniques

The examination of convergence or the rate of convergence of the numerical methods used in this study relied on actual testing of the iterative methods. Since the set of equations are coupled and nonlinear, the analytical procedure for determination of convergence can not be used. By extensive trial and error of SOR method, the approximated equations were stable for ω up to 0.5. The component mass balance equations were stable for a wide range of ω close to 0.5, except for the PDE for H_2O component because of the high affinity of zeolite for water. The heat balance equations were also stable for a wide range of ω close to 0.5. Keeping ω constant at these values, the number of grids for a bed 20 inches long bed were 75 for CO_2 and 150 for water in single component adsorption. The time step ranged from 7 to 18 seconds. In multi-component adsorption the limiting value is determined by what the H_2O component was in the single component adsorption case.

In Newman's method, the number of grids could be reduced to 50 and time step increased to as large as 60 seconds. In both methods the iteration was stopped if $(C^{n+1}-C^{n})/C^{n+1}$ was less than $1.0E^{-4}$ for each grid point. This error criteria was chosen because the difference in two consecutive iteration was less than 1.0e-5 for a gas phase concentration of 1.0e-10. For the first few steps, the number of iterations is large but it decreases to a minimum of 20 to 50 after that.

The stability and the rate of convergence in these PDE's are mostly affected by the rate of adsorption which is related to the isotherms. The second most important criteria is the diffusion coefficients. The implementation of diffusion coefficients makes the PDE's parabolic and in turn causes the equations to be more stable, preventing any initial error from propagating in time.

SECTION 4

COMPUTER MODEL VERIFICATION

4.1 Test Apparatus

Verification of the numerical models is accomplished with test data from the Molecular Sieve Material Bench Test (MSMBT), a Marshall Space Flight Center apparatus. The MSMBT was designed to capture the dynamic adsorption and desorption characteristics of sorbent materials to be used in the International Space Station 4BMS for a range of CO_2 and H_2O partial pressures. The test apparatus consists of a small packed column 2 inches in diameter and 10 or 20 inches long, depending on the configuration. Instrumentation of the column includes temperature probes and sampling tubes for measurement at sorbent material endpoints and intermediate points. Continuous measurements of the exit CO_2 and H_2O partial pressures, and all temperatures, are recorded. A gas chromatograph is used to capture the breakthrough at interior bed locations. The MSMBT is fully described elsewhere (Knox, 1992; Mohamadinejad, 1994). Any significant changes made in the test configuration will be discussed in the following sections as appropriate.

The MSMBT is used to obtain data empirically that is not available otherwise or not reliable in the open literature, such as heat of adsorption and lumped mass transfer coefficients. Testing to obtain empirical data is performed in such a way as to isolate the phenomenon of interest as much as possible. Heat transfer coefficients, (not available otherwise due to the use of unique NASA foam insulation for superior adiabatic conditions) for example, were obtained by flowing dry, heated nitrogen through a sorbent bed desorbed in the presence of nitrogen.

The second use of the MSMBT is to verify, following determination of empirical values, the accuracy of the model under conditions similar to those expected in the ISSA. This consists of comparing a series of MSMBT runs at varying conditions to the results of the computer model ran at the same conditions. Of special interest is the capability of the model to predict transient bed temperature and partial pressure for conditions not used to obtain the empirical values.

4.2 Thermal Characterization with Inert Gas Heating

In order to approach adiabatic conditions during adsorption and strip desorption testing, three layers of high performance insulation are used on the column. The inner layers are commercially available felt insulation. A third layer of spray-on foam specifically designed for the Space Transportation System External Tank was added to improve heat retention. However, thermal characteristics were not available for the foam insulation. In order to obtain coefficients for the transfer of heat from the outside (ambient air) through the three layers of insulation to the canister wall (H_OW) and from the carrier gas, or fluid, to the canister wall (H_FW), a test using heated nitrogen gas was conducted.

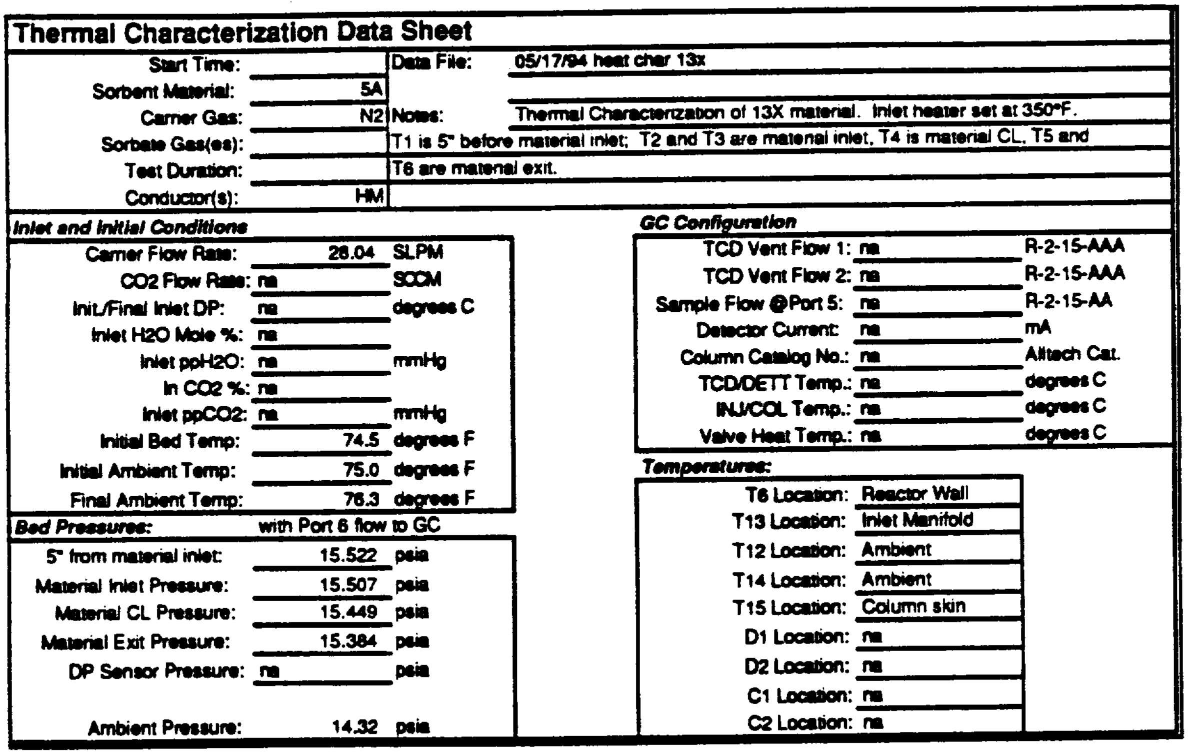
To simulate the inert gas heating test, the bed inlet temperature measured during the test was curve fit and applied as an input to the computer model. Bed inlet volumetric flow rate was calculated based on the ideal gas law.

4.2.1 Molecular Sieve 5A

Specifications for the thermal characterization test with molecular sieve 5A are shown in Figure 4-1. Volumetric flow rate is required as a model input. This is calculated based on the flow controller rate in standard liters per minute (conditions at point 1) and bed inlet conditions (point 2):

$$G_2 = \frac{P_1 T_2 G_1}{P_2 T_1} = \frac{14.696}{15.47}\frac{28.04}{535}T_2 \qquad \text{eq 4.1}$$

P_2 (15.47 psia) is the average inlet pressure calculated from test data. Pressure varied less than 0.5% from P_2 during the test. T_2 is the test inlet temperature which has been curve fit for use in the model. G_1 is the volumetric flowrate.



Thermal Characterization Data Sheet

Start Time:		Data File:	05/17/94 heat char 13x
Sorbent Material:	5A		
Carrier Gas:	N2	Notes:	Thermal Characterization of 13X material. Inlet heater set at 350°F.
Sorbate Gas(es):		T1 is 5" before material inlet; T2 and T3 are material inlet, T4 is material CL, T5 and	
Test Duration:		T6 are material exit.	
Conductor(s):	HM		

Inlet and Initial Conditions

Carrier Flow Rate:	28.04	SLPM
CO2 Flow Rate:	na	SCCM
Init/Final Inlet DP:	na	degrees C
Inlet H2O Mole %:	na	
Inlet ppH2O:	na	mmHg
In CO2 %:	na	
Inlet ppCO2:	na	mmHg
Initial Bed Temp:	74.5	degrees F
Initial Ambient Temp:	75.0	degrees F
Final Ambient Temp:	76.3	degrees F

Bed Pressures: with Port 6 flow to GC

5" from material inlet:	15.522	psia
Material Inlet Pressure:	15.507	psia
Material CL Pressure:	15.449	psia
Material Exit Pressure:	15.384	psia
DP Sensor Pressure:	na	psia
Ambient Pressure:	14.32	psia

GC Configuration

TCD Vent Flow 1:	na	R-2-15-AAA
TCD Vent Flow 2:	na	R-2-15-AAA
Sample Flow @Port 5:	na	R-2-15-AA
Detector Current:	na	mA
Column Catalog No.:	na	Alltech Cat.
TCD/DETT Temp.:	na	degrees C
INJ/COL Temp.:	na	degrees C
Valve Heat Temp.:	na	degrees C

Temperatures:

T6 Location:	Reactor Wall
T13 Location:	Inlet Manifold
T12 Location:	Ambient
T14 Location:	Ambient
T15 Location:	Column skin
D1 Location:	na
D2 Location:	na
C1 Location:	na
C2 Location:	na

FIGURE 4-1 5A THERMAL CHARACTERIZATION TEST SPECIFICATIONS

4.2.1.1 FLOW1MOL Model Thermal Verification

The determination of heat transfer coefficients and verification of the single material flow-through adsorption and desorption model, FLOW1MOL, is described in this section.

Variations of the heat transfer coefficients, H_OW and H_FW, were studied with the computer model. Sensitivity to changes in H_OW is shown in Figure 4-2. Sensitivity to changes in H_FW is shown in Figure 4-3. The test data for the bed outlet is shown as "T5 [°F]", while the model output for variations in the heat transfer coefficient is labeled "T gas@5".

The final values selected for H_FW and H_OW were 2.5 and 0.25, respectively. The comparison with test data is shown in Figure 4-4. The test temperatures for the bed inlet, center, and outlet are labeled based on the measuring thermocouple as "T3 [°F]", "T4 [°F]", and "T5

[°F]" respectively. The corresponding model data is labeled based on node point as "gas T 5", "gas T 50", and "gas T 101" respectively.

As shown in Figure 4-4, the FLOW1MOL thermal model can simulate test data very closely with the adjustment of the heat transfer coefficients. To test the model's predictive capabilities, another nitrogen heating test with different flow rate and heater set point should be run. Independent simulation with FLOW1MOL and comparison of results would indicate predictive accuracy.

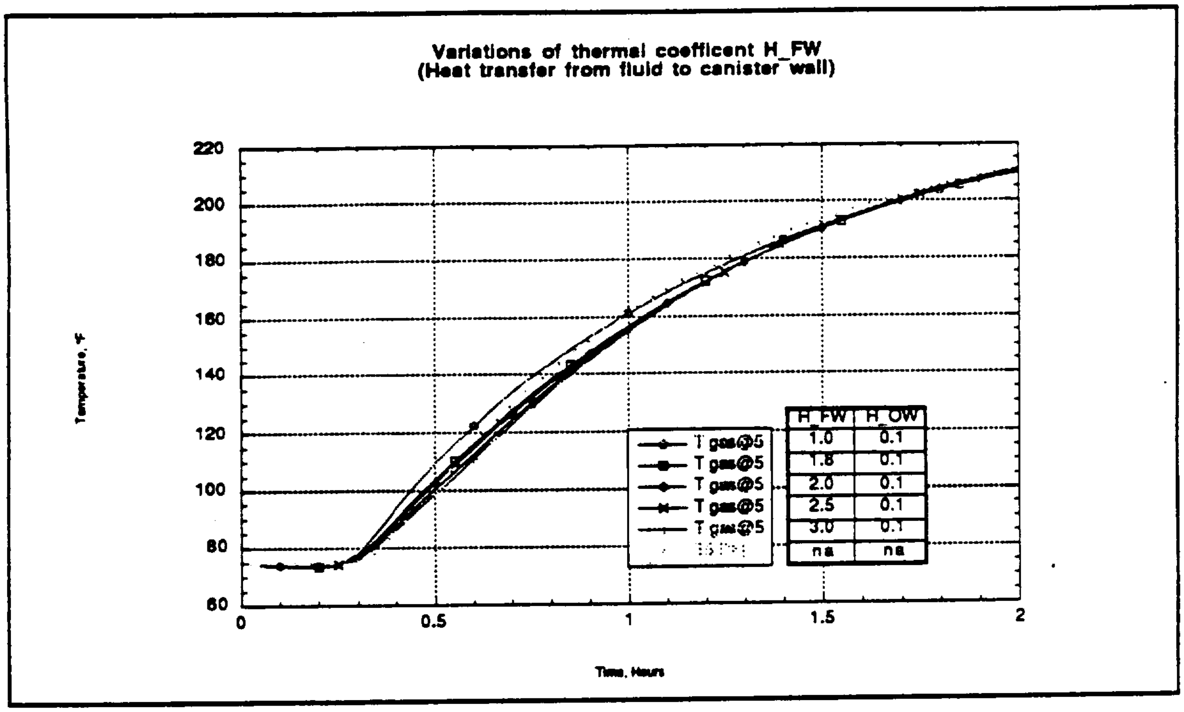


FIGURE 4-2 VARIATIONS OF H_FW FOR FLOW1MOL

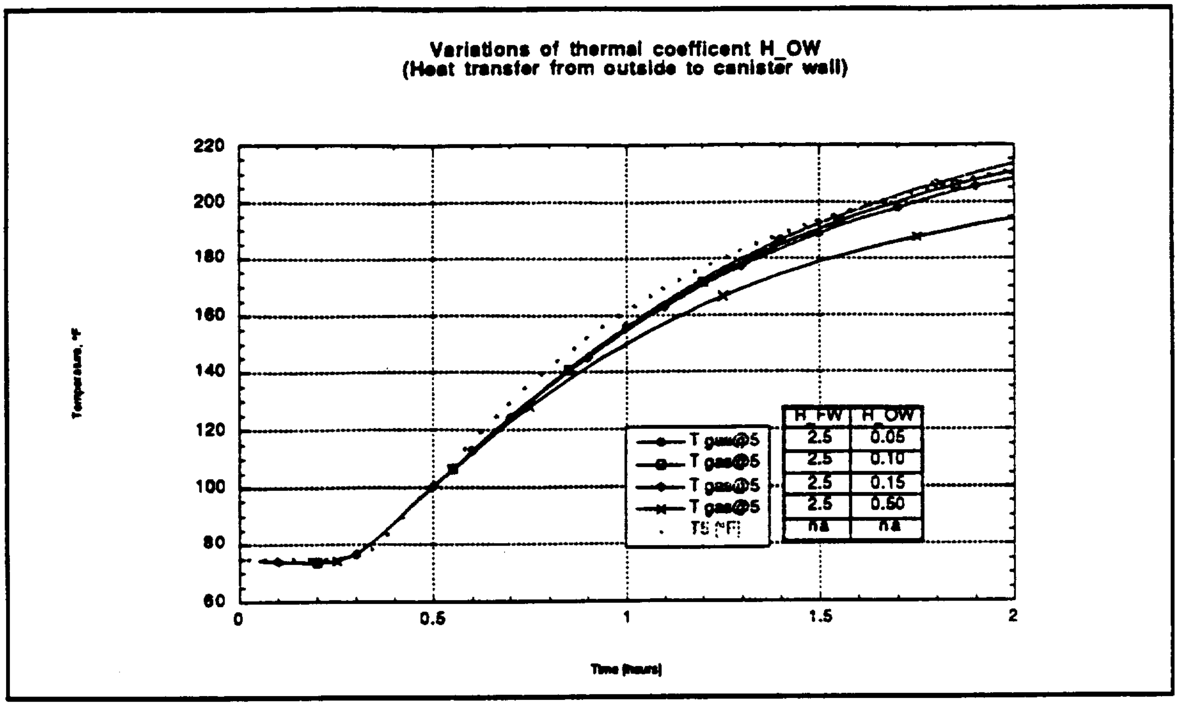


FIGURE 4-3 VARIATIONS OF H_OW FOR FLOW1MOL

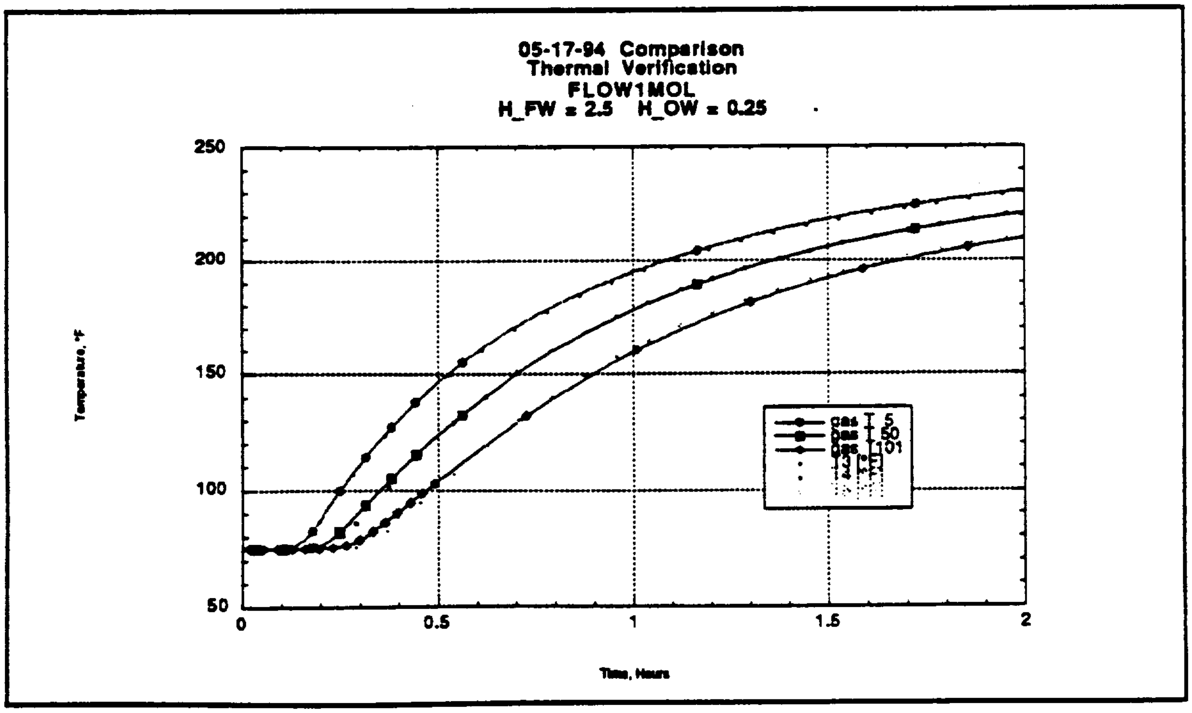


FIGURE 4-4 FINAL THERMAL VERIFICATION OF FLOW1MOL FOR 5A

4.2.1.2 2DMOL Model Thermal Verification

The determination of heat transfer coefficients and verification of the two-dimensional adsorption and desorption model, 2DMOL, is described in this section.
Variations of the heat transfer coefficients, H_OW and H_FW, were studied with the computer model. Sensitivity to changes in H_OW is shown in Figure 4-5. Sensitivity to changes in H_FW is shown in Figure 4-6. The test data for the bed outlet is shown as "T5 [°F]", while the model output for variations in the heat transfer coefficient is labeled "T gas@5". Note that the sensitivity of the 2-dimensional model is greatly increased over that of the 1-dimensional model as shown in the previous section. This can be expected due to the existence of radial nodes in this model, allowing a radial gradient in temperature. Since the H_FW is a function of temperature, it changes along the axial direction of the packed column. Simulation of model with the actual test shows that the prediction of H_FW by theoretical equation as it described in Section 2. fits the temperature profile very well.

The final values selected for H_FW and H_OW were 0.3 and 0.25, respectively. The comparison with test data is shown in Figure 4-7. The test temperatures for the bed inlet, center, and outlet are labeled based on the measuring thermocouple as "T3 [°F]", "T4 [°F]", and "T5 [°F]" respectively. The corresponding model data is labeled based on node point as "gas T 5", "gas T 50", and "gas T 101" respectively.

As shown in Figure 4-7, the 2DMOL thermal model can simulate test data very closely with the adjustment of the heat transfer coefficients. The final value selected for H_FW, 0.3, however, is an order of magnitude less than the theoretical value (4.0), calculated in Equation 2.86. This is expected since in the actual adsorption test the bed temperature is around 100° F. Since the heat transfer coefficient is a function of Reynolds number, it would be best to run several experiments with different Reynolds number and fit the obtained results in a empirical equation.

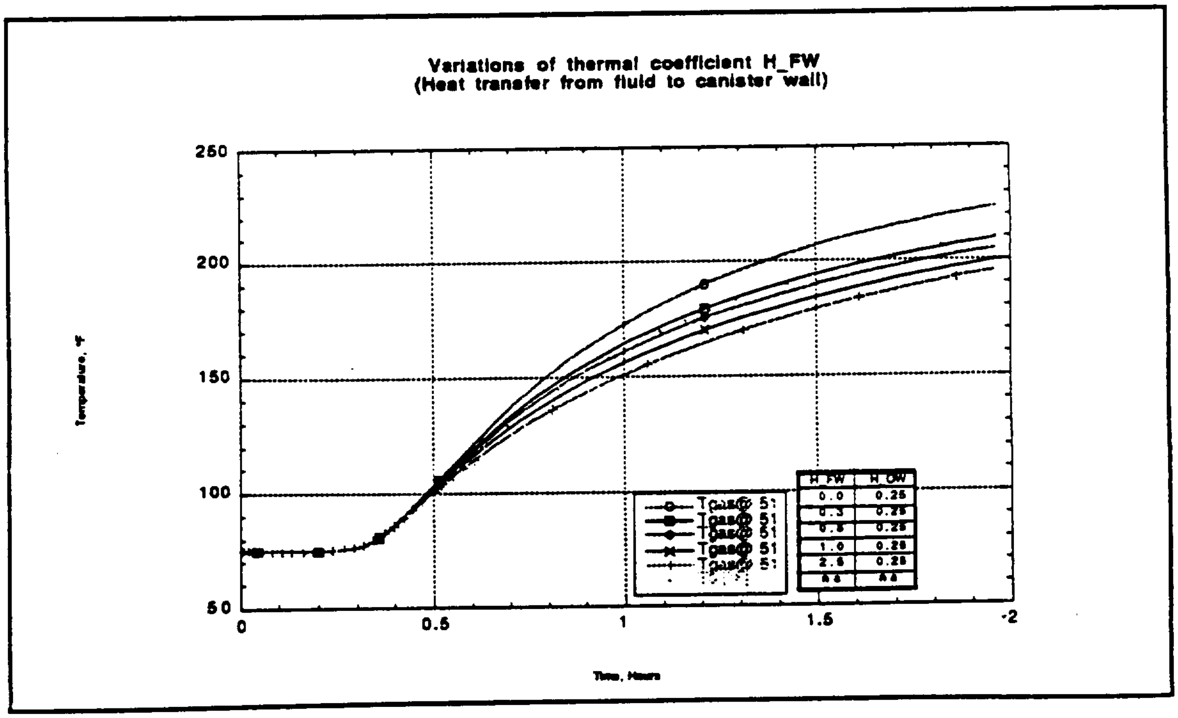


FIGURE 4-5 VARIATIONS OF H_FW FOR 2DMOL

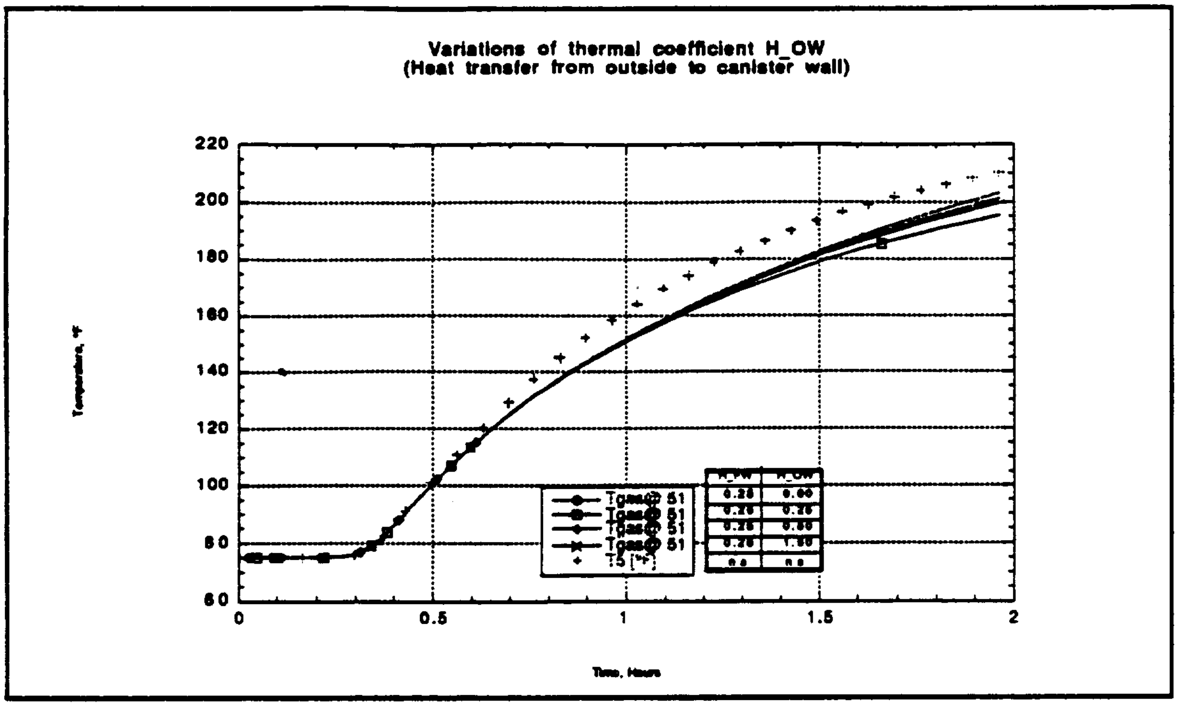


FIGURE 4-6 VARIATIONS OF H_OW FOR 2DMOL

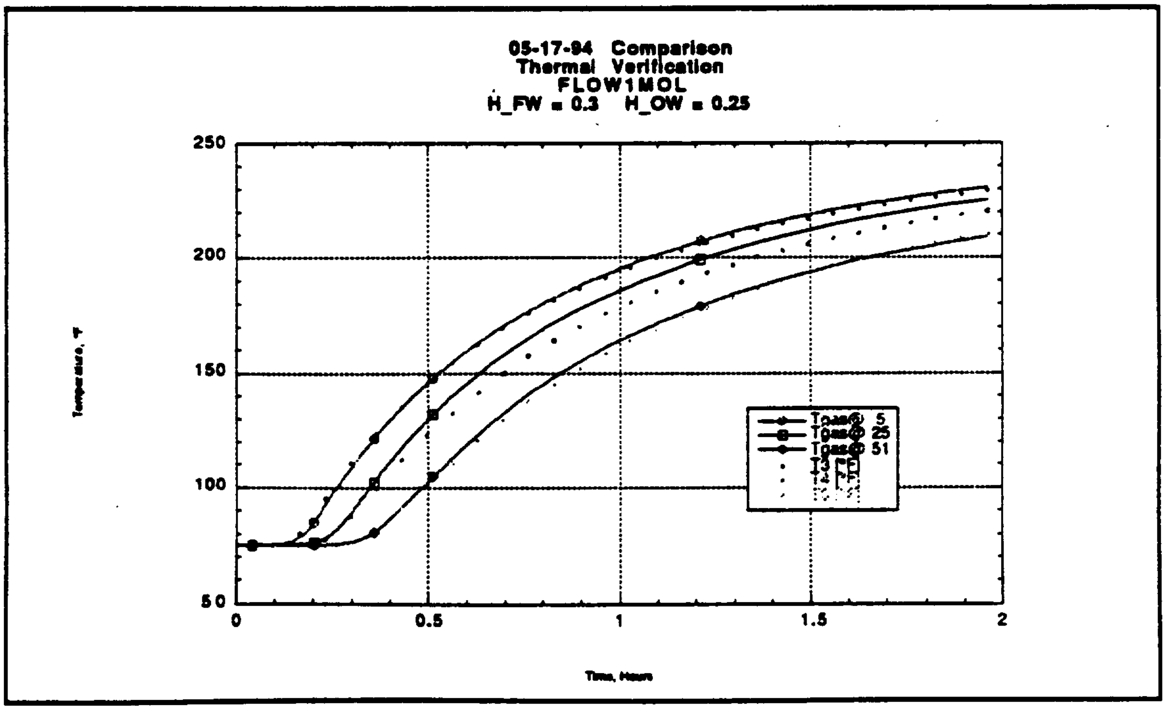


FIGURE 4-7 FINAL THERMAL VERIFICATION OF 2DMOL FOR 5A

4.3 Carbon Dioxide Adsorption

4.3.1 Molecular Sieve 5A

The test ran on 01-26-94 will be used to verify the FLOW1MOL and 2DMOL bed models for carbon dioxide adsorption. Test specifications are shown in Figure 4-8.

Two phenomenon were observed to occur during this adsorption test which are not generally accounted for in mathematical models of this type.

The first phenomenon is the significance of bed channeling in the 1.875 inch ID cylindrical column with sorbent pellets of 8-12 mesh (appx. 2 mm in diameter). The existence of significant bed channeling is evident in Figure 4-9, which shows the breakthrough of CO_2 in the column. Note the discrepancy in the breakthrough at the material exit centerline (labeled "Outlet") from a point 5 inches downstream of the material exit ("Outlet Beads"). The downstream point measures all gas exiting the column, mixed via turbulence through 5 inches of 3 mm glass beads. Consistent with the assumption of flow channeling at the column wall, the breakthrough is earlier for the mixed gas than at the column centerline. The two-dimensional model was developed to account for the channeling, and to aid in derivation of a technique to intelligently apply a channeling factor to the more CPU efficient single dimension models.

It is important to note that the presence of channeling will be much more pronounced in the 5A sorbent bed heater core. The heater core is configured such that a number of approximately 1/2 inch square channels contain the sorbent material. Channeling occurs at the walls of the packed bed, and the sorbent bed channels have a very high ratio of pellet diameter to wall diameter, which is the critical parameter for flow channeling. The two-dimensional model can be used to simulate the actual 4BMS as long as a good estimate of 5A sorbent bed reactor core channeling is available.

The second phenomenon observed is the importance of including nitrogen co-adsorption for accurate modeling of the carbon dioxide breakthrough with nitrogen as the carrier gas. Figures 4-10 and 4-11 show the test data compared with modeling results with and without coadsorption of N2. As seen in Figure 4-10, the initial adsorption of N2 on a empty bed creates a fast rise in temperature over the entire bed. The higher bed temperature reduces bed CO_2 adsorption capacity. N2 coadsorption also marginally reduces the bed capacity for CO_2. As a result, the earlier breakthrough curves shown in Figure 4-11 reflect the lower bed capacity for modeling with N2 coadsorption. These results also correspond well to the actual test data in both figures.

4.3.1.1 FLOW1MOL Model Verification

The determination of input values, mass transfer coefficients, and verification of the single material flow-through adsorption and desorption model, FLOW1MOL, is described in this section.

Model inputs are shown in Figure 4-12. Determination of appropriate values for each non-obvious input parameter are discussed below.

Inside Diameter: Based on 2 inch OD and wall thickness of 1/16 inch.

Wall Density: For canister wall, stainless steel

Void Fraction: Based on value given in Wright et. al. for Davison 5A

Interfacial Surface Area: Based on Davison Chemical Co. supply.

Pellet Conductivity: Based on Davison Chemical Co. supply.

Pellet Specific Heat: Based on Davison Chemical Co. supply.

Heat Transfer Fluid to Wall: Determined via thermal characterization test (see Sect. 4-2)

Adsorb Data Sheet

Start Time:	10:00	**Data File:**	01/26/94 CO2 on 13x
Sorbent Material:	5A		
Carrier Gas:	N2	**Notes:**	Another run like 01/19/94 to validate column repeatability
Sorbate Gas(es):	CO2		
Test Duration:	3 HRS		
Conductor(s):	JG, JK		

Inlet and Initial Conditions

Carrier Flow Rate:	28.04	SLPM
CO2 Flow Rate:	223.9	SCCM
Init./Final Inlet DP:		degrees C
Inlet H2O Mole %:		
Inlet ppH2O:		mmHg
In CO2 %:	0.774	
Inlet ppCO2:	6.14	mmHg
Initial Bed Temp:	77.7	degrees F
Initial Ambient Temp:	77	degrees F
Final Ambient Temp:	77	degrees F

GC Configuration

TCD Vent Flow 1:	8
TCD Vent Flow 2:	9.7
Sample Flow @Port 5:	11
Detector Current:	150
Column Catalog No.:	C-5000
TCD/DETT Temp.:	120/130
INJ/COL Temp.:	150/100
Valve Heat Temp.:	117

Temperatures:

T6 Location:	Matl. exit opp. scre
T11 Location:	Exit in Beads
T12 Location:	Ambient
T14 Location:	
T15 Location:	Insulation Skin
D1 Location:	Column exit
D2 Location:	Column exit
C1 Location:	Column exit
C2 Location:	Column exit

Bed Pressures: with Port 6 flow to GC

Inlet Manifold Pressure:	15.587	psia
Material Inlet Pressure:	15.348	psia
Material CL Pressure:	15.312	psia
Material Exit Pressure:	15.265	psia
DP Sensor Pressure:	15.22	psia
Ambient Pressure:		psia

CO2 Calibration:

Cal. Gas CO2 % Vol: 0.39992% in N2

Low Pressure Calibration: 14.9 psia

	H2O Area	N2 Area
Sample 1:	12036	2739386
Sample 2:	12048	2739601

High Pressure Calibration: 15.3 psia

	H2O Area	N2 Area
Sample 1:	12309	2797320

FIGURE 4-8 SPECIFICATIONS FOR 01-26-94 CO_2 ADSORPTION TESTING

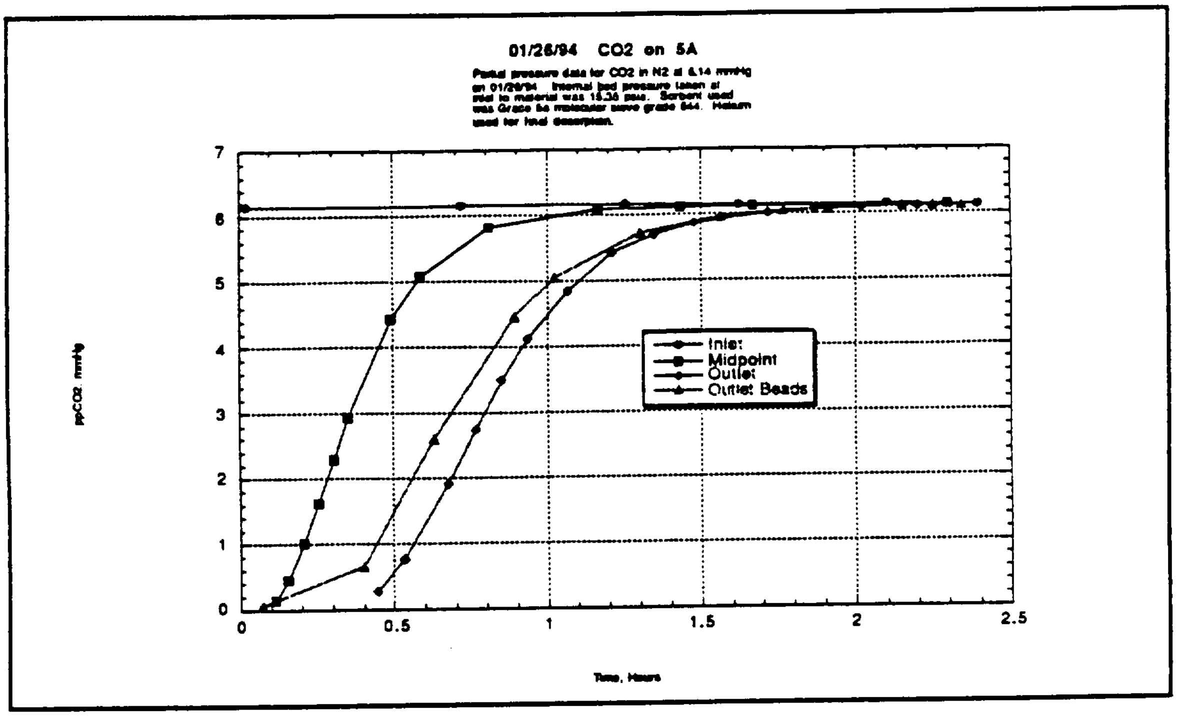


FIGURE 4-9 CO_2 PARTIAL PRESSURES FOR 01-26-94 TEST

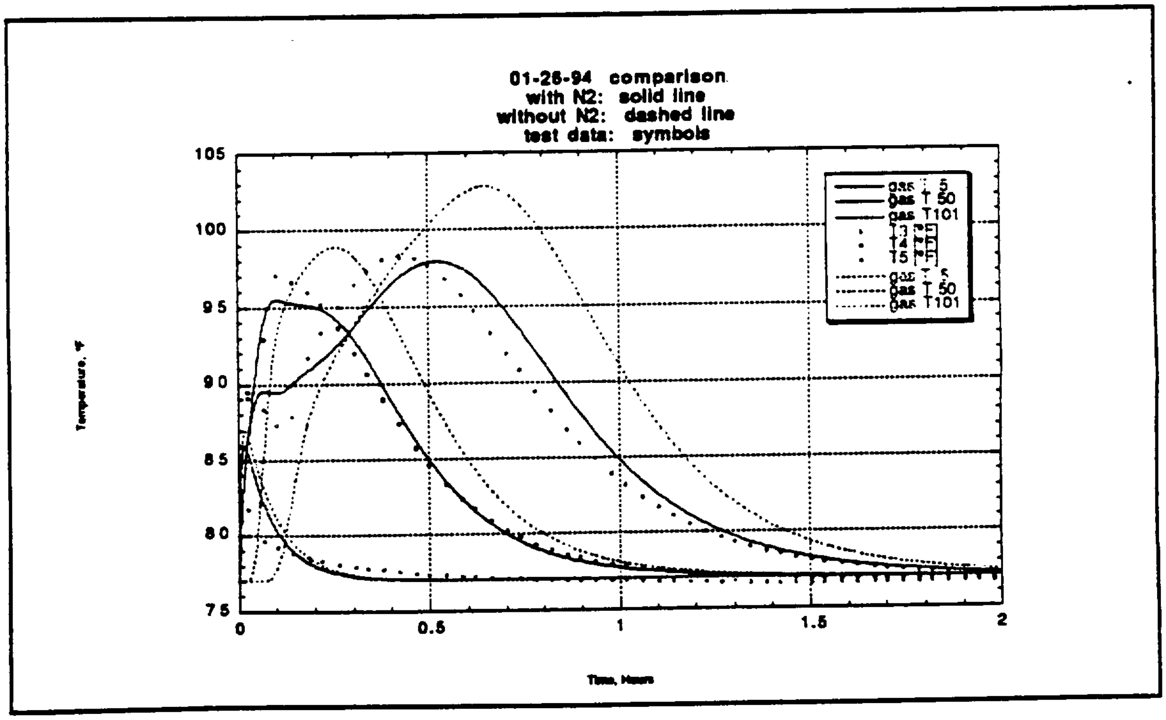


FIGURE 4-10 CO_2 MODEL TEMPERATURE COMPARISONS WITH AND WITHOUT NITROGEN COADSORPTION

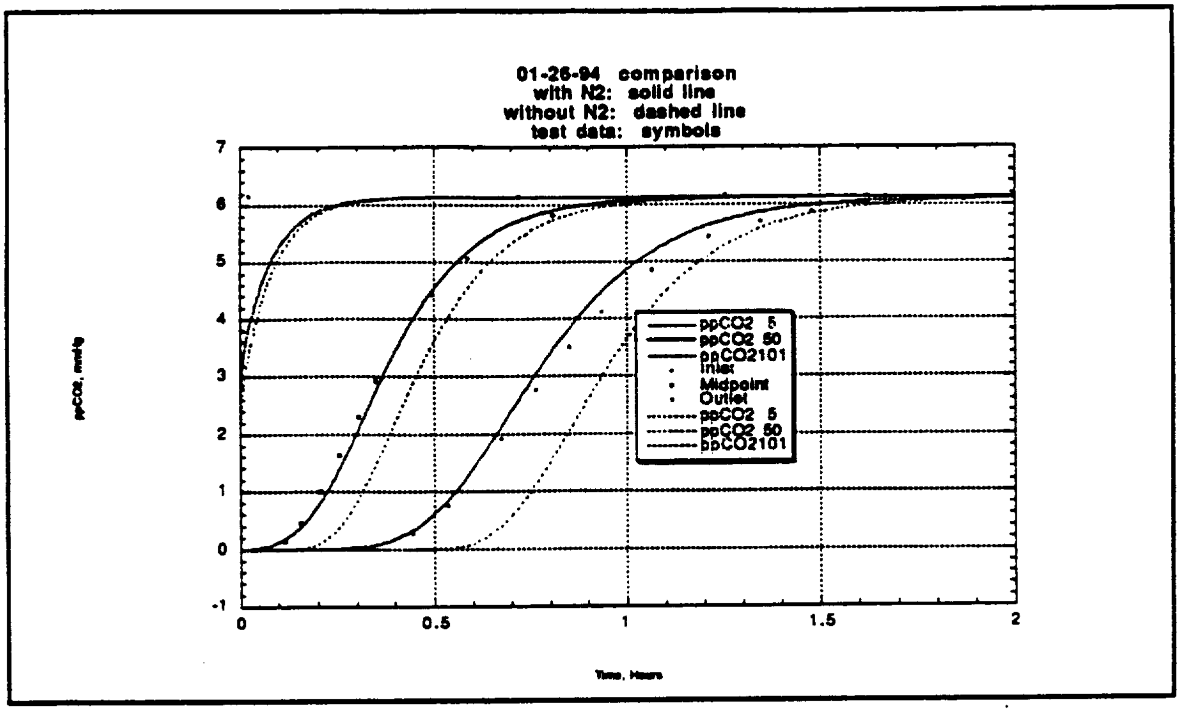


FIGURE 4-11 CO_2 MODEL BREAKTHROUGH COMPARISONS WITH AND WITHOUT NITROGEN COADSORPTION

Heat Transfer Outside to Wall: Determined via thermal characterization test (see Section 4-2)

Pellet Density: Based on Davison Chemical Co. supply.

Heat of Adsorption: Based on Davison Chemical Co. supply.

Mass Transfer Coefficients: Determined empirically based on comparison of test data to model results

Results of the model comparison are shown by the solid lines in Figures 4-10 and 4-11. As is evident, the comparison is very favorable. Note in particular the excellent agreement of coadsorption temperature effects in Figure 4-10. The immediate temperature increase caused by N2 adsorption is properly reflected by the model. In addition, although not exact in magnitude, the subsequent dramatic drop at both the midpoint and outlet positions is also properly modeled.

Also note the proper change in slope of the breakthrough curve in Figure 4-12 from the midpoint to outlet positions. This change is due to increasing bed temperatures and the subsequent broadening of the mass transfer zone which is an indication of a nonequilibrium conditions between the gas and solid phase during the adsorption of CO_2 on 5A material. The comparison of the experimental data with models predicts the mass transfer coefficient of CO_2 to be .017 ft/hr.

Variable	*Description*	*Units*	*Current Value*
END	final time	hours	3
DELT1	time step	hours	0.00100
NJ	Number of nodes	n/a	101
DELZ1	distance between nodes	feet	0.0083333
TO	Ambient temperature	°R	537
G F	Volumetric flow rate	CFH	55.5
T O	Inlet temperature	°R	537
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P TOT	Total Inlet Pressure	psia	15.89
BC C(1)	Inlet partial pressure	psia	0.12
BC C(2)	Inlet partial pressure	psia	0.00
BC C(3)	Inlet partial pressure	psia	15.77
BC C(4)	Inlet partial pressure	psia	0.00
Z	Bed length	feet	0.83333333
D E	External Diameter	feet	0.1666667
D I	Inside Diameter	feet	0.15625
RO WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CON S	Pellet (solid) conductivity	BTU/°F-	0.1
CP S	Pellet (solid) specific heat	BTU/°F-lb	0.2
H FW	Heat transfer fluid to wall		2.5
H OW	Heat transfer outside to wall		0.1
RO S	Pellet density	lb/ft^3	75
M W(1)	Molecular Weight	lb/lb-mol	44
M W(2)	Molecular Weight	lb/lb-mol	18
M W(3)	Molecular Weight	lb/lb-mol	28
M W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-20000
HEAT(2)	heat of adsorption	BTU/lb-mole	-30000
HEAT(3)	heat of adsorption	BTU/lb-mole	-9600
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K F(1)	Mass Transfer Coefficent	ft/hr	0.02
K F(2)	Mass Transfer Coefficent	ft/hr	0.003
K F(3)	Mass Transfer Coefficent	ft/hr	0.05
K F(4)	Mass Transfer Coefficent	ft/hr	0
IND(1)	Indicator for CO2 present		1
IND(2)	Indicator for H2O present		0
IND(3)	Indicator for N2 present		1
IND(4)	Indicator for O2 present		0

FIGURE 4-12 INPUT DATA SET "A" FOR FLOW1MOL COMPARISON WITH 01-26-94 TEST

Variable	Description	Units	Current Value
LABEL(1)	Output Labels		ppCO2
LABEL(2)			pp N2
LABEL(3)			gas T
LABEL(4)			p tot
AOUT(1)	C array indices to output		1
AOUT(2)			2
AOUT(3)			5
AOUT(4)			9
IOUT(1)	Node locations to be output		5
IOUT(2)			25
IOUT(3)			50
IOUT(4)			75
IOUT(5)			101
S_B	Cross sectional surface area	ft^3	0.019175
CON_WA	Wall thermal conductivity	Btu/hr/ft/R	32.875
CON_WI_Q			0.29
CON_WI_K			0.19
CP_WA			0.109848
CP_WI_Q			0.21
CP_WI_K			0.21
RO_WI_Q			3.5
RO_WI_K			16
X_WA	Canister Wall Thickness	feet	0.00541667
X_WI_Q	Q-felt insultation thickness	feet	0.04166667
X_WI_K	K-felt insulation thickness	feet	0.02083333
RA			0.0045893
RAV			0.0091785
EPSIN			0.317
RHOS			43
ALPHA1			575
R_P			0.00472441
D_P			0.00944882
BC_L1	boundary temperature for fluid		537
BC_L2	boundary temperature for solid		537
BC_L3	boundary temperature for canister wall		537
BC_L5	fluid pressure at each grid(?)		15.886
BC_L6			15.886
IN_L1	inlet temperature for fluid		537
IN_L2	inlet temperature for solid		537
IN_L3	inlet temperature for canister wall		537
IN_L5	fluid pressure at each grid(?)		15.886
IN_L6			15.886
R	Gas Constant		10.73
PI			3.141593
TEST			0

FIGURE 4-12 INPUT DATA SET "B" FOR FLOW1MOL COMPARISON WITH 01-26-94 TEST

4.3.1.2 2DMOL Model Verification

The determination of input values, mass transfer coefficients, and verification of the two-dimensional single material flow-through adsorption and desorption model, 2DMOL, is described in this section.

Model inputs are shown in Figure 4-13 and 4-14.

Variable	*Description*	*Units*	*Current Value*
END	final time	hours	2
DELT	time step	hours	0.00250
NJ	Number of axial nodes	n/a	51
NJR	Number of radial nodes	n/a	20
DELZ	distance between nodes	feet	0.01666666
TAMB	Ambient temperature	°R	537
G_F	Volumetric flow rate	CFH	55.6
TO	Inlet temperature	°R	537
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P_TOT	Total Inlet Pressure	mmHg	789.43
P_C(1)	Inlet partial pressure	mmHg	6.13
P_C(2)	Inlet partial pressure	mmHg	0.00
P_C(3)	Inlet partial pressure	mmHg	783.30
P_C(4)	Inlet partial pressure	mmHg	0.00
Z	Bed length	feet	0.83333333
D_E	External Diameter	feet	0.1666667
D_I	Inside Diameter	feet	0.155833
RO_WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.25
RO_S	Pellet density	lb/ft^3	75
M_W(1)	Molecular Weight	lb/lb-mol	44
M_W(2)	Molecular Weight	lb/lb-mol	18
M_W(3)	Molecular Weight	lb/lb-mol	28
M_W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.02
K_F(2)	Mass Transfer Coefficent	ft/hr	0.0035
K_F(3)	Mass Transfer Coefficent	ft/hr	0.1
K_F(4)	Mass Transfer Coefficent	ft/hr	0.1

FIGURE 4-13 INPUT DATA SET "A" FOR 2DMOL COMPARISON WITH 01-26-94 TEST

Variable	Description	Units	Current Value
S_B	Cross sectional surface area	ft^3	0.019069
CP_WA			0.109848
RA			0.00472441
RAV			0.0058
RHOS			75
ALPHA1			635
R_P			0.0047244
D_P			0.0094488
R	Gas Constant		10.73
PI			3.141593
UINS			0.354
HWALL			12
NC	Number of components		2
T_FO	same as TO		537
GC			416975040
CON_S1			1
NP			125
PT	same as PTOT		789.42569
LEN	same as Z		1.66667
VOID_B	same as EPSEX		0.35
IND(1)			1
IND(2)			0
IND(3)			1
IND(4)			0
IDES1			0
ISAT			0

FIGURE 4-14 INPUT DATA SET "B" FOR 2DMOL COMPARISON WITH 01-26-94 TEST

Results of the model comparison are shown in Figures 4-15 and 4-16. As is evident, the comparison is very favorable. Note that two sets of model data are presented; averaged data and centerline data. The centerline data for node 51 is the central node radially and the last bed material node axially. Averaged data for node 51 is also at the last bed node axially, but is an average of all the radial nodes. The averaged data is thus representative of gas after mixing in the glass beads, or the test data labeled "Outlet Beads". Averaging of data was done by two methods, Cubic Spline and Simpson integration (Gerald and Wheatley, 1984). It was found that the Simpson integration method gives better results than the Simpson method.

Based on the above discussion, it is expected that centerline data - the line with filled markers in Figure 4-15 - will compare with the small circular markers. As seen from the figure, this comparison is indeed favorable. Average data should be compared with the small triangular markers, and once again, this is a favorable comparison. As a result, it can be inferred that the two-dimensional model does correctly model the channeling observed in this test.

As seen from Figure 4-16, the temperature comparison is also favorable. The increase in fidelity of the two-dimensional model is evident by comparison of Figure 4-16 with the results from the one-dimensional model in Figure 4-10. The 2DMOL simulation more closely follows the actual temperature peak, both in time and in magnitude.

The short time of breakthrough allows it to use IAST, even though it takes more CPU time than using the Langmuir isotherm. A mass transfer coefficient of .017 ft/hr was used in predicting the CO_2 breakthrough curve.

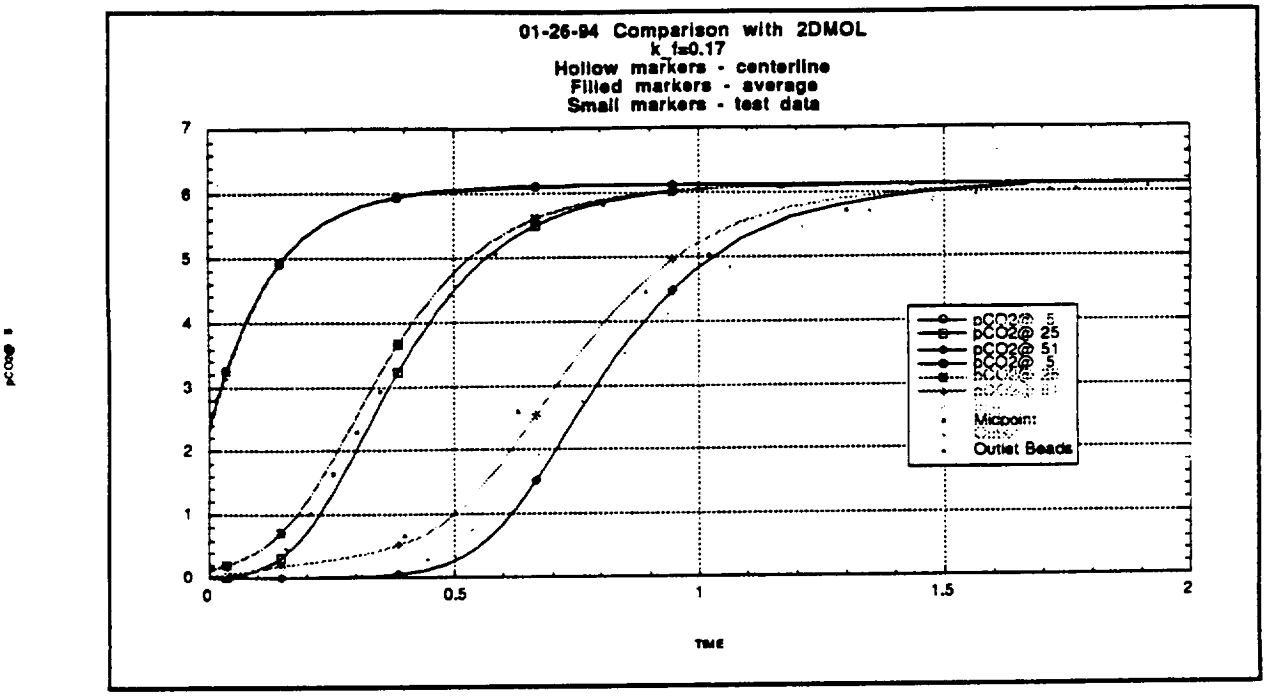


FIGURE 4-15 CO_2 MODEL BREAKTHROUGH COMPARISONS FOR 01-26-94 TEST WITH 2DMOL RESULTS

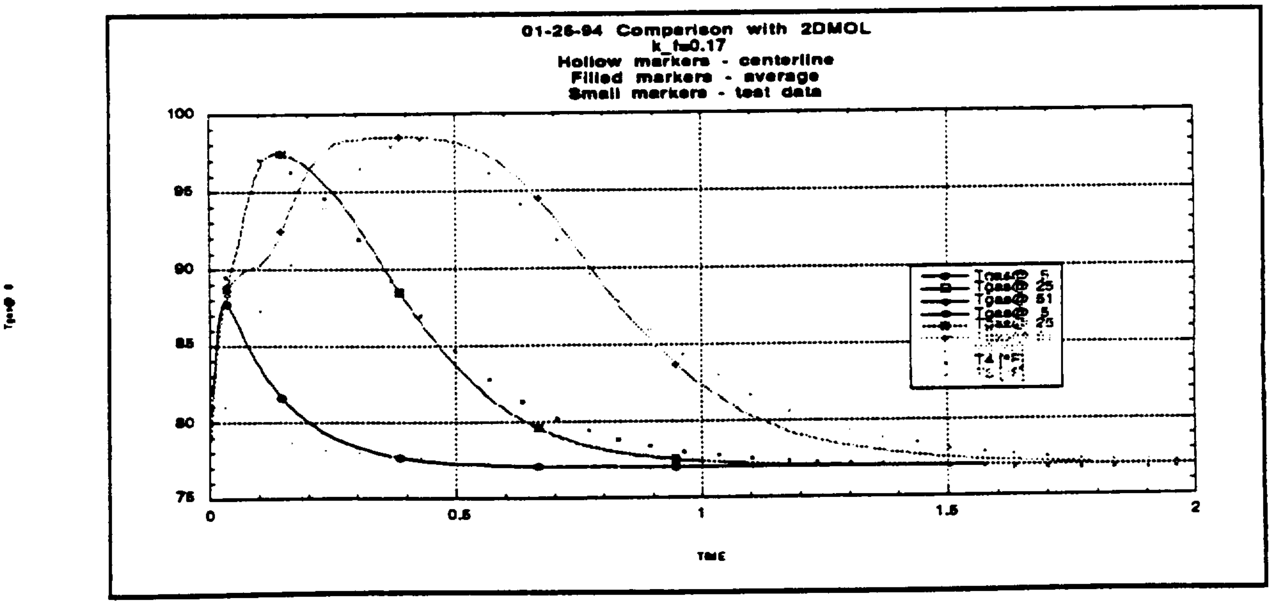
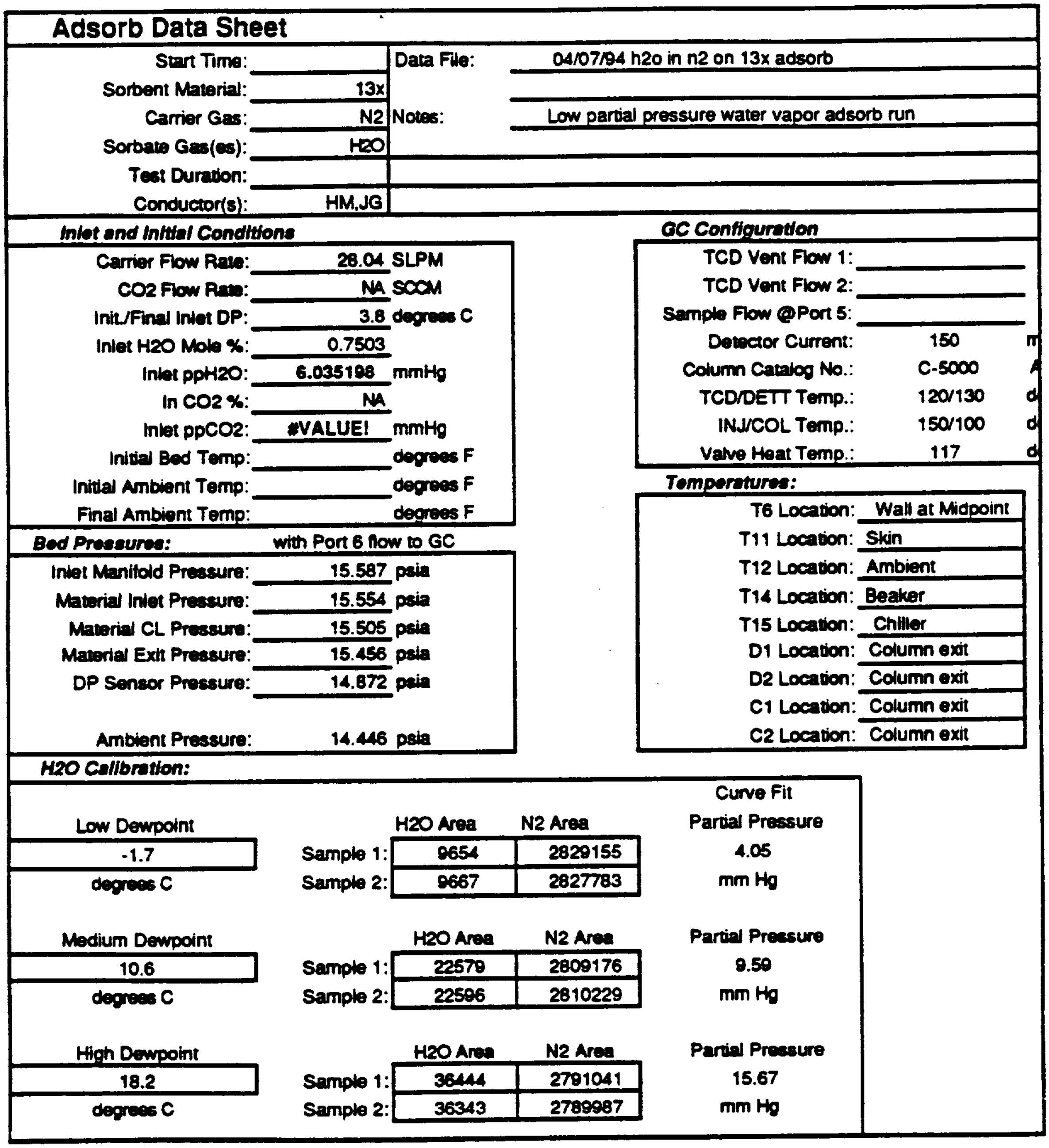


FIGURE 4-16 CO_2 TEMPERATURE COMPARISONS FOR 01-26-94 TEST WITH 2DMOL RESULTS

4.4 Water Adsorption

4.4.1 Molecular Sieve 5A



Adsorb Data Sheet

Start Time:		Data File:	04/07/94 h2o in n2 on 13x adsorb
Sorbent Material:	13x		
Carrier Gas:	N2	Notes:	Low partial pressure water vapor adsorb run
Sorbate Gas(es):	H2O		
Test Duration:			
Conductor(s):	HM,JG		

Inlet and Initial Conditions

Carrier Flow Rate:	28.04	SLPM
CO2 Flow Rate:	NA	SCCM
Init./Final Inlet DP:	3.8	degrees C
Inlet H2O Mole %:	0.7503	
Inlet ppH2O:	6.035198	mmHg
In CO2 %:	NA	
Inlet ppCO2:	#VALUE!	mmHg
Initial Bed Temp:		degrees F
Initial Ambient Temp:		degrees F
Final Ambient Temp:		degrees F

GC Configuration

TCD Vent Flow 1:		
TCD Vent Flow 2:		
Sample Flow @Port 5:		
Detector Current:	150	m
Column Catalog No.:	C-5000	A
TCD/DETT Temp.:	120/130	d
INJ/COL Temp.:	150/100	d
Valve Heat Temp.:	117	d

Bed Pressures: with Port 6 flow to GC

Inlet Manifold Pressure:	15.587	psia
Material Inlet Pressure:	15.554	psia
Material CL Pressure:	15.505	psia
Material Exit Pressure:	15.456	psia
DP Sensor Pressure:	14.872	psia
Ambient Pressure:	14.446	psia

Temperatures:

T6 Location:	Wall at Midpoint
T11 Location:	Skin
T12 Location:	Ambient
T14 Location:	Beaker
T15 Location:	Chiller
D1 Location:	Column exit
D2 Location:	Column exit
C1 Location:	Column exit
C2 Location:	Column exit

H2O Calibration:

Dewpoint		H2O Area	N2 Area	Curve Fit Partial Pressure
Low Dewpoint -1.7 degrees C	Sample 1:	9654	2829155	4.05 mm Hg
	Sample 2:	9667	2827783	
Medium Dewpoint 10.6 degrees C	Sample 1:	22579	2809176	9.59 mm Hg
	Sample 2:	22596	2810229	
High Dewpoint 18.2 degrees C	Sample 1:	36444	2791041	15.67 mm Hg
	Sample 2:	36343	2789987	

FIGURE 4-17 SPECIFICATIONS FOR 01-26-94 H_2O ADSORPTION TESTING

The test ran on 01-26-94 will be used to verify the FLOW1MOL and 2DMOL bed models for water adsorption. Test specifications are shown in Figure 4-17. Breakthrough for water is shown in Figure 4-18. Note, as in the test of CO_2 adsorption, the discrepancy between breakthrough at the material exit measured at the centerline and after gas mixing (marked "Exit" and "Mixed" respectively). The extremely high affinity of zeolite 5A is also evident in the time required for breakthrough.

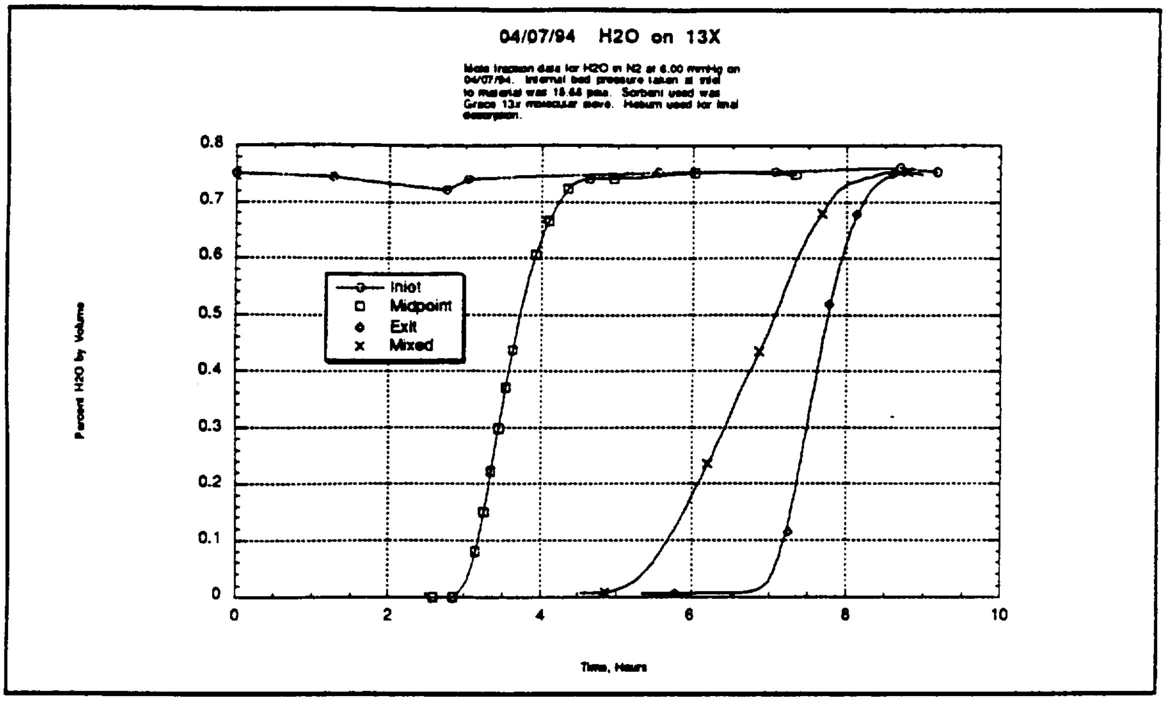


FIGURE 4-18 H_2O BREAKTHROUGH FOR 04-07-94 TEST

4.4.1.1 FLOW1MOL Model Verification

The determination of mass transfer coefficients and verification of the single material flow-through adsorption and desorption model, FLOW1MOL, is described in this section.

Model inputs are shown in Figure 4-19 and 4-20. Results of the model comparison are shown by the solid lines in Figures 4-21 and 4-22. For breakthrough, the FLOW1MOL model compares well with test data at the midpoint and fairly well at material exit, except as the breakthrough curve approaches the inlet partial pressure. Deviations from the test data may be explained by the lack of channeling modeled by FLOW1MOL. Channeling appears to have a greater influence in water adsorption than CO_2, perhaps due to the much longer duration of the adsorption run. For temperatures, the comparison is also fairly good, especially during the heating of the column.

Variable	Description	Units	Current Value
END	final time	hours	4
DELT1	time step	hours	0.00300
NJ	Number of nodes	n/a	61
DELZ1	distance between nodes	feet	0.01388888
TO	Ambient temperature	°R	532
G_F	Volumetric flow rate	CFH	56.76
T_O	Inlet temperature	°R	532
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P_TOT	Total Inlet Pressure	psia	15.68
BC_C(1)	Inlet partial pressure	psia	0
BC_C(2)	Inlet partial pressure	psia	0.116021
BC_C(3)	Inlet partial pressure	psia	15.56397
BC_C(4)	Inlet partial pressure	psia	0
Z	Bed length	feet	0.83333333
D_E	External Diameter	feet	0.1666667
D_I	Inside Diameter	feet	0.15625
RO_WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CON_S	Pellet (solid) conductivity	BTU/hr ft °R	0.1
CP_S	Pellet (solid) specific heat	BTU/hr lb °R	0.2
H_FW	Heat transfer fluid to wall		2.5
H_OW	Heat transfer outside to wall		0.1
RO_S	Pellet density	lb/ft^3	75
M_W(1)	Molecular Weight	lb/lb-mol	44
M_W(2)	Molecular Weight	lb/lb-mol	18
M_W(3)	Molecular Weight	lb/lb-mol	28
M_W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-19000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.02
K_F(2)	Mass Transfer Coefficent	ft/hr	0.0025
K_F(3)	Mass Transfer Coefficent	ft/hr	0.1
K_F(4)	Mass Transfer Coefficent	ft/hr	0
IND(1)	Indicator for CO2 present		0
IND(2)	Indicator for H2O present		1
IND(3)	Indicator for N2 present		1
IND(4)	Indicator for O2 present		0

FIGURE 4-19 INPUT DATA SET "A" FOR FLOW1MOL COMPARISON WITH 04-07-94 TEST

LABEL(1)	Output Labels		"ppH2O"
LABEL(2)			"pp N2"
LABEL(3)			"gas T"
LABEL(4)			"tot P"
AOUT(1)	C array indices to output		1
AOUT(2)			2
AOUT(3)			5
AOUT(4)			10
IOUT(1)	Node locations to be output		2
IOUT(2)			15
IOUT(3)			30
IOUT(4)			45
IOUT(5)			61
S_B	Cross sectional surface area	ft^3	0.019175
CON_WA	Wall conductivity	BTU/hr ft °R	32.875
CON_WI_Q	Conductivity for Q-fiber	BTU/hr ft °R	0.29
CON_WI_K	Conductivity for Min-K	BTU/hr ft °R	0.19
CP_WA	Specific Heat for wall	BTU/hr lb °R	0.109848
CP_WI_Q	Specific Heat for Q-fiber	BTU/hr lb °R	0.21
CP_WI_K	Specific Heat for Min-K	BTU/hr lb °R	0.21
RO_WI_Q	Density for Q-fiber	lb/ft^3	3.5
RO_WI_K	Density for Min-K	lb/ft^3	16
X_WA	Canister Wall Thickness	feet	0.00541667
X_WI_Q	Q-fiber insultation thickness	feet	0.04166667
X_WI_K	Min-K insulation thickness	feet	0.02083333
RA	not used		0.00472441
RAV	not used		0.00472441
EPSIN	Pellet macropore porosity		0.317
RHOS	Pellet density	lb/ft^3	43
ALPHA1	not used		575
R_P	Pellet radius	feet	0.00472441
D_P	Pellet diameter	feet	0.00944882
BC_L1	boundary temperature for fluid	°R	532
BC_L2	boundary temperature for solid	°R	532
BC_L3	boundary temperature for canister wall	°R	532
BC_L5	fluid pressure at inlet	psia	15.68
BC_L6	fluid pressure at inlet	psia	15.68
IN_L1	initial temperature for fluid	°R	532
IN_L2	initial temperature for solid	°R	532
IN_L3	initial temperature for canister wall	°R	532
IN_L5	initial pressure at each grid	°R	15.68
IN_L6	initial pressure at each grid	°R	15.68
R	Gas Constant		10.73
PI			3.141593
ITEST	flag for calculation of flow properties		0

FIGURE 4-20 INPUT DATA SET "B" FOR FLOW1MOL COMPARISON WITH 04-07-94 TEST

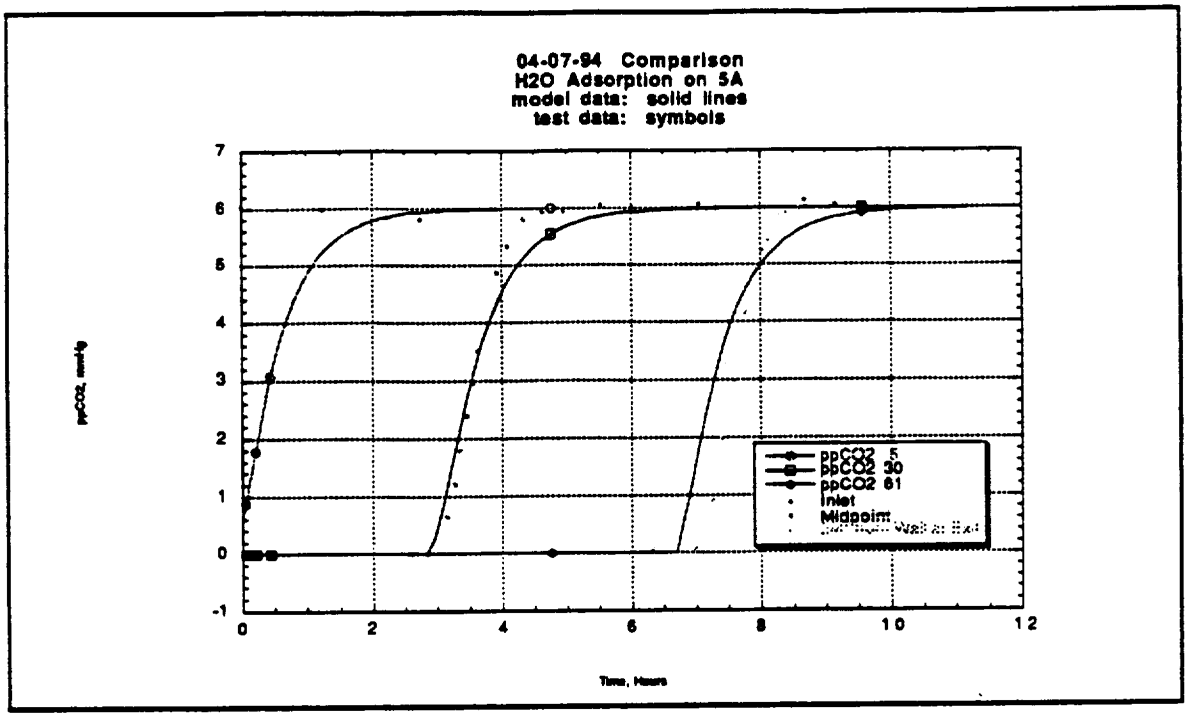


FIGURE 4-21 H_2O BREAKTHROUGH COMPARISONS FOR 04-07-94 TEST WITH FLOW1MOL RESULTS

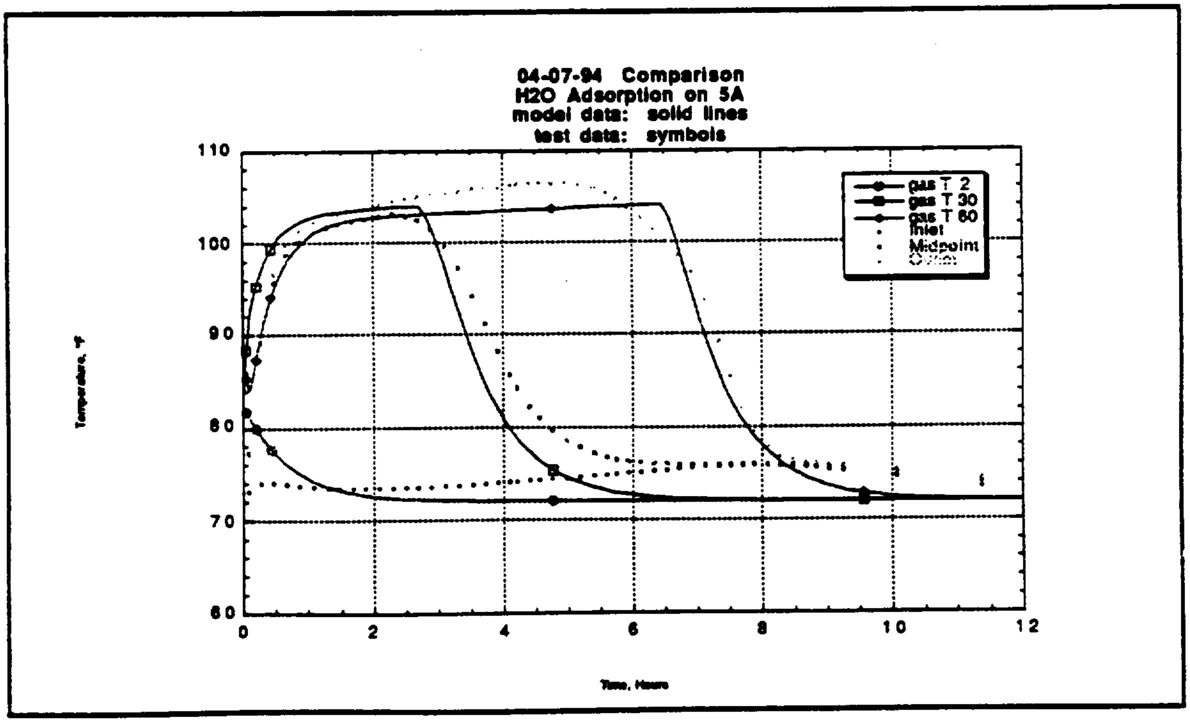


FIGURE 4-22 H_2O TEMPERATURE COMPARISONS FOR 04-07-94 TEST WITH FLOW1MOL RESULTS

4.4.1.2 2DMOL Model Verification

The determination of mass transfer coefficients and verification of the two-dimensional adsorption and desorption model, 2DMOL, is described in this section.

Input data for the 2DMOL model for the 04-07-94 test are shown in Figures 4-23 and 4-24. The results of the model comparison are shown by the solid lines in Figures 4-25 and 4-26. As expected the center line breakthrough and temperature profile match the one-dimensional flow model. The average breakthrough at the midpoint and the outlet of the bed are shown. The model midpoint average is reasonably good, since the dispersion is moderate compared to the centerline breakthrough. There was no test data point for the average component partial pressure at the midpoint. This can not be done without the distortion of flow regime in the rest of the bed.

The result of the model for the average breakthrough at the outlet shows an early breakthrough compared to test data for breakthrough of H_2O/N_2. This can be attributed to several factors. The equilibrium isotherm, the porosity, and the radial diffusivity of H_2O. This is also apparent from an early breakthrough at the centerline of the bed. Any small deviation in isotherm measurement causes an early/later breakthrough. The porosity variation effect is also a large contributor to the early average breakthrough. In this study, The wall porosity was taken as .85 as it was suggested in theoretical section. Monudin (1976) used a value of .54 in a packed reactor and had good model result with test data. Therefore, it is possible that a smaller porosity value should be taken. The third parameter that affects the average breakthrough is the radial diffusion of H_2O in the bed. Two different equations were used in this work to describe the radial diffusion as it described earlier. The equation by Fahien (1954) predicts the average breakthrough better and was used in this study. Even though this equation gave good result for the adsorption of CO_2/N_2, it is possible that it underestimates the water radial diffusivity.

A mass transfer coefficient of .0035 was used to predict the breakthrough curve. Langmuir isotherm was used to predict H_2O/N_2 isotherm.

Variable	Description	Units	Current Value
END	final time	hours	5
DELT	time step	hours	0.00250
NJ	Number of axial nodes	n/a	51
NJR	Number of radial nodes	n/a	20
DELZ	distance between nodes	feet	0.01666666
TAMB	Ambient temperature	°R	532
G_F	Volumetric flow rate	CFH	56.12
TO	Inlet temperature	°R	532
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P_TOT	Total Inlet Pressure	mmHg	810.89
P_C(1)	Inlet partial pressure	mmHg	0.00
P_C(2)	Inlet partial pressure	mmHg	6.04
P_C(3)	Inlet partial pressure	mmHg	804.85
P_C(4)	Inlet partial pressure	mmHg	0.00
Z	Bed length	feet	0.83333333
D_E	External Diameter	feet	0.1666667
D_I	Inside Diameter	feet	0.155833
RO_WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.25
RO_S	Pellet density	lb/ft^3	75
M_W(1)	Molecular Weight	lb/lb-mol	44
M_W(2)	Molecular Weight	lb/lb-mol	18
M_W(3)	Molecular Weight	lb/lb-mol	28
M_W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.017
K_F(2)	Mass Transfer Coefficent	ft/hr	0.0035
K_F(3)	Mass Transfer Coefficent	ft/hr	0.05
K_F(4)	Mass Transfer Coefficent	ft/hr	0.1
ERROR1			0.00000001
ERROR2			1E-09

FIGURE 4-23 INPUT DATA SET "A" FOR 2DMOL COMPARISON WITH 04-07-94 TEST

Variable	*Description*	*Units*	*Current Value*
S_B	Cross sectional surface area	ft^3	0.019069
CP_WA	Wall specific heat		0.109848
RA	not used	feet	0.00472441
RAV	not used	feet	0.0058
RHOS	not used		75
ALPHA1	not used		635
R_P	Pellet radius	feet	0.0047244
D_P	Pellet diameter	feet	0.0094488
R	Gas Constant		10.73
PI			3.141593
UINS			0.354
HWALL			12
NC	Number of components		2
T_FO	same as TO		537
GC			416975040
CON_S1			1
NP			125
PT	same as PTOT		789.42569
LEN	same as Z		1.66667
VOID_B	same as EPSEX		0.35
IND(1)			1
IND(2)			0
IND(3)			1
IND(4)			0
IDES1			0
ISAT			0

FIGURE 4-24 INPUT DATA SET "B" FOR 2DMOL COMPARISON WITH 04-07-94 TEST

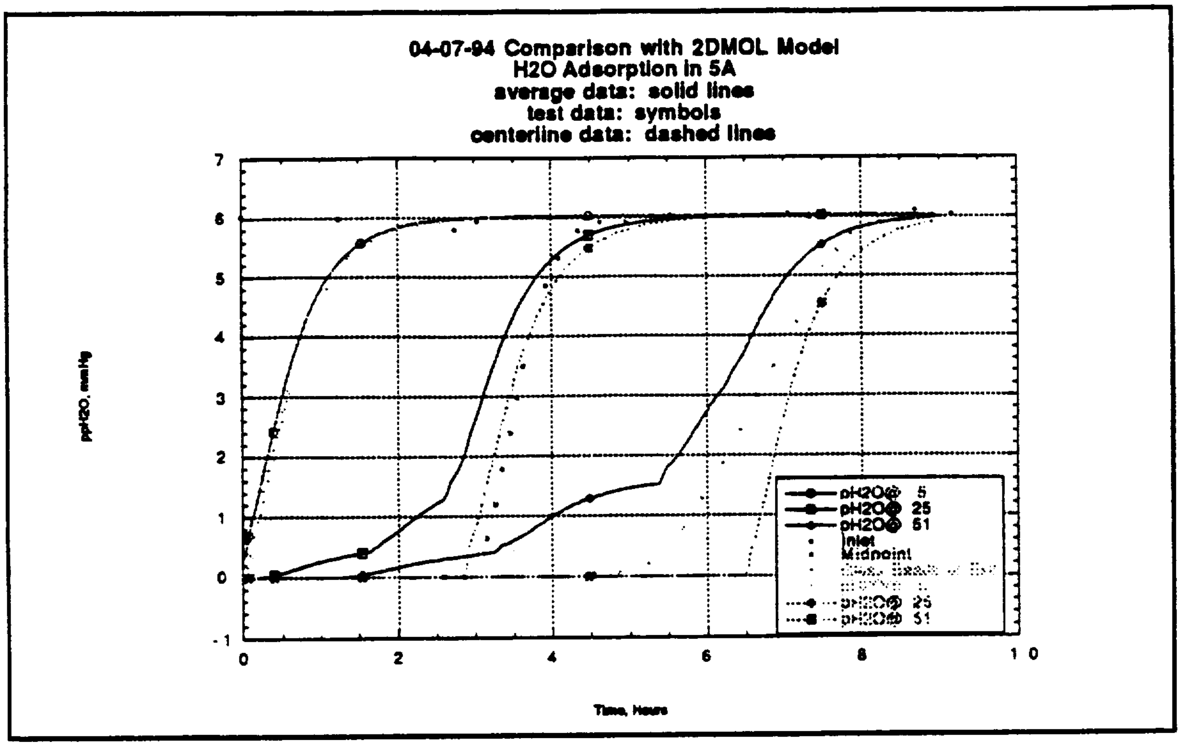


FIGURE 4-25 H_2O BREAKTHROUGH COMPARISONS FOR 04-07-94 TEST WITH 2DMOL RESULTS

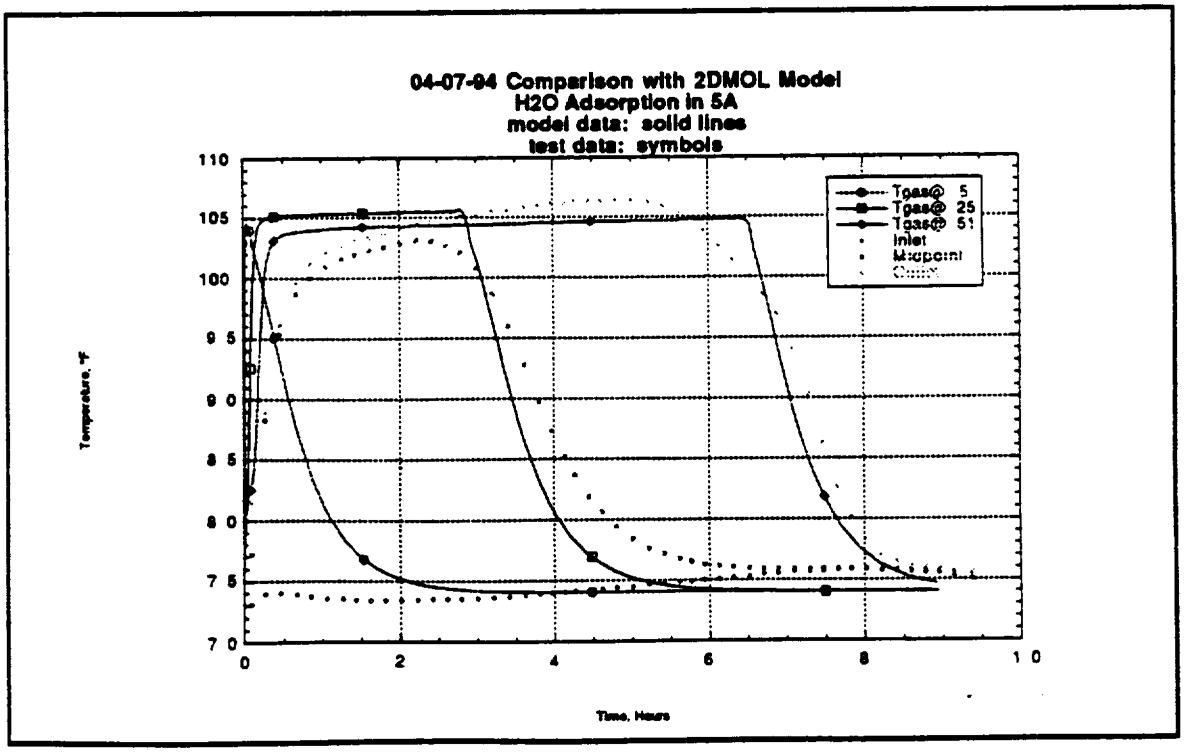
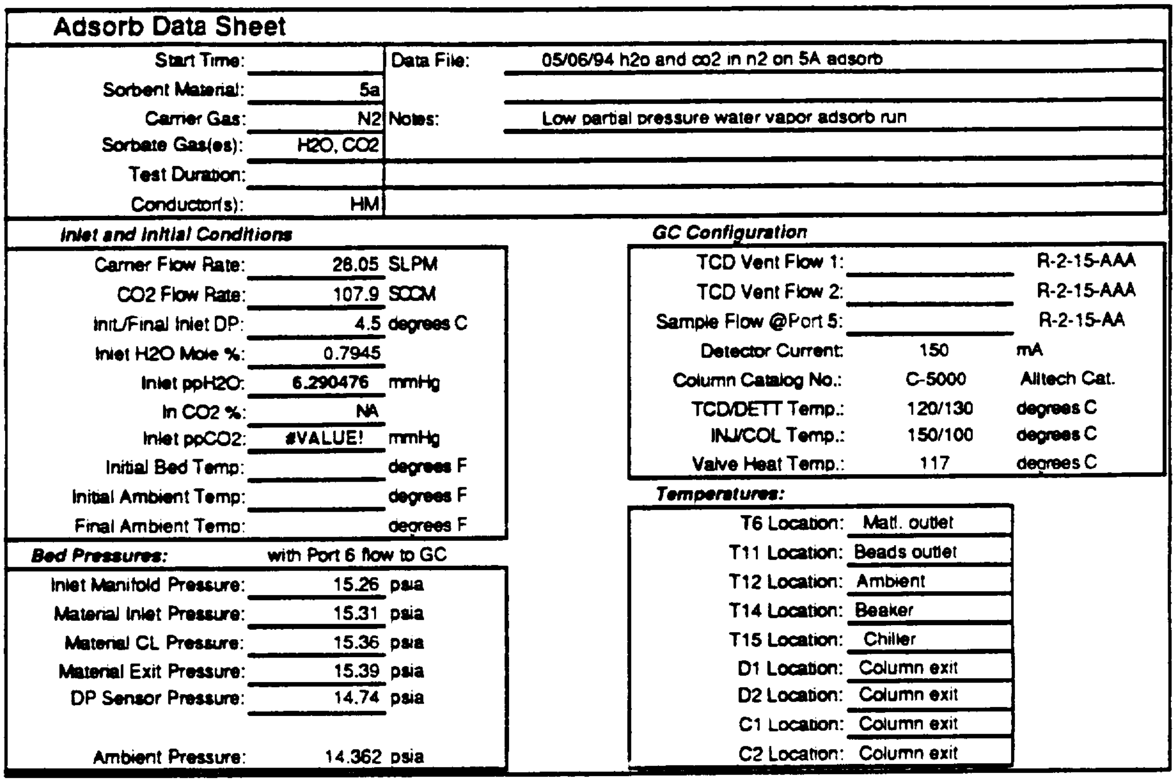


FIGURE 4-26 H_2O TEMPERATURE COMPARISONS FOR 04-07-94 TEST WITH 2DMOL RESULTS

4.5 Water and Carbon Dioxide Adsorption

4.5.1 Molecular Sieve 5A



Adsorb Data Sheet

Start Time:		Data File:	05/06/94 h2o and co2 in n2 on 5A adsorb
Sorbent Material:	5a		
Carrier Gas:	N2	Notes:	Low partial pressure water vapor adsorb run
Sorbate Gas(es):	H2O, CO2		
Test Duration:			
Conductor(s):	HM		

Inlet and Initial Conditions

Carrier Flow Rate:	28.05	SLPM
CO2 Flow Rate:	107.9	SCCM
Init/Final Inlet DP:	4.5	degrees C
Inlet H2O Mole %:	0.7945	
Inlet ppH2O:	6.290476	mmHg
In CO2 %:	NA	
Inlet ppCO2:	#VALUE!	mmHg
Initial Bed Temp:		degrees F
Initial Ambient Temp:		degrees F
Final Ambient Temp:		degrees F

Bed Pressures: with Port 6 flow to GC

Inlet Manifold Pressure:	15.26	psia
Material Inlet Pressure:	15.31	psia
Material CL Pressure:	15.36	psia
Material Exit Pressure:	15.39	psia
DP Sensor Pressure:	14.74	psia
Ambient Pressure:	14.362	psia

GC Configuration

TCD Vent Flow 1:		R-2-15-AAA
TCD Vent Flow 2:		R-2-15-AAA
Sample Flow @Port 5:		R-2-15-AA
Detector Current:	150	mA
Column Catalog No.:	C-5000	Alltech Cat.
TCD/DETT Temp.:	120/130	degrees C
INJ/COL Temp.:	150/100	degrees C
Valve Heat Temp.:	117	degrees C

Temperatures:

T6 Location:	Matl. outlet
T11 Location:	Beads outlet
T12 Location:	Ambient
T14 Location:	Beaker
T15 Location:	Chiller
D1 Location:	Column exit
D2 Location:	Column exit
C1 Location:	Column exit
C2 Location:	Column exit

FIGURE 4-27 SPECIFICATIONS FOR 05-06-94 H_2O/CO_2 ADSORPTION TESTING

The test ran on 05-06-94 will be used to verify the FLOW1MOL and 2DMOL bed models for water and carbon dioxide adsorption. Test specifications are shown in Figure 4-27. Breakthrough for water is shown in Figure 4-28. Note, as in the previous tests, the discrepancy between breakthrough at the material exit measured at the centerline and after gas mixing (marked "matl outlet" and "column outlet" respectively). The extremely high affinity of zeolite 5A is also evident in the time required for breakthrough. It is also evidenced by the fact that the presence of CO_2 is not evident by comparison of this figure with Figure 4-18, which shows water breakthrough for a similar test, except without CO_2.

Figure 4-29 shows the centerline bed temperatures for the test. Note the early peaks resulting from CO_2 adsorption around 0.5 hours.

Figure 4-30 shows the carbon dioxide breakthrough. The roll-up phenomenon is evident in this graph, wherein the weaker adsorbate (CO_2) is driven off the bed by the stronger (H_2O), resulting in an outlet partial pressure greater than at the inlet.

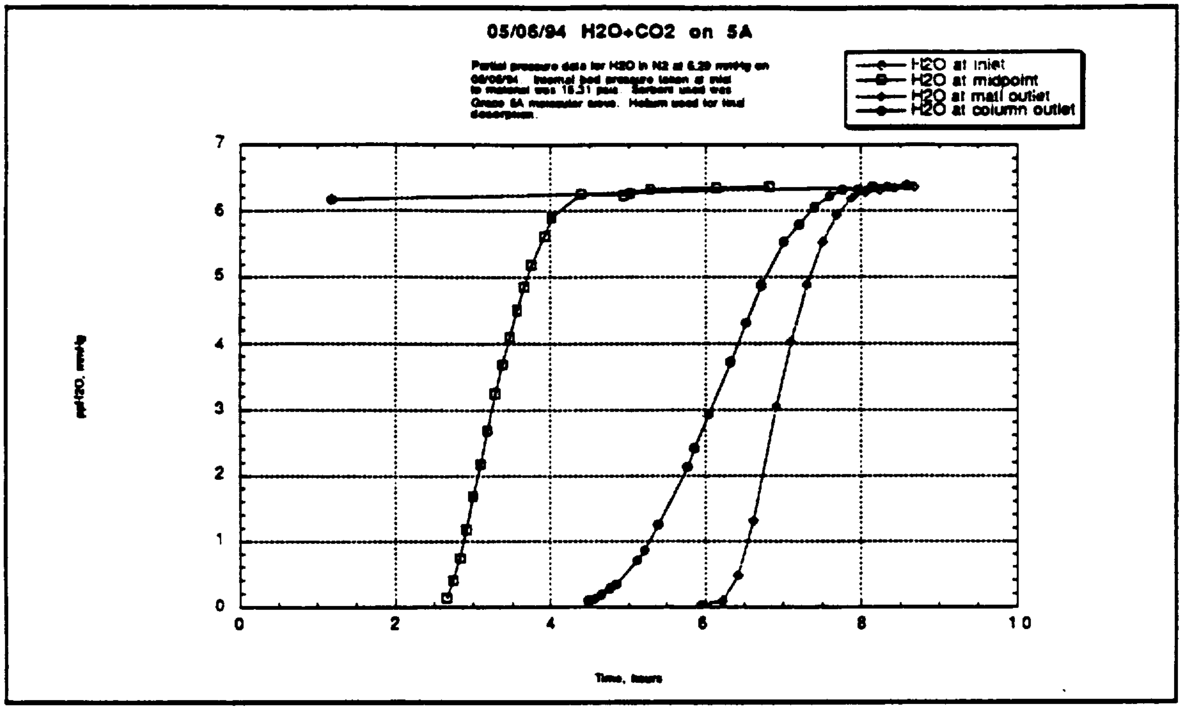


FIGURE 4-27 H_2O BREAKTHROUGH FOR 05-06-94 TEST

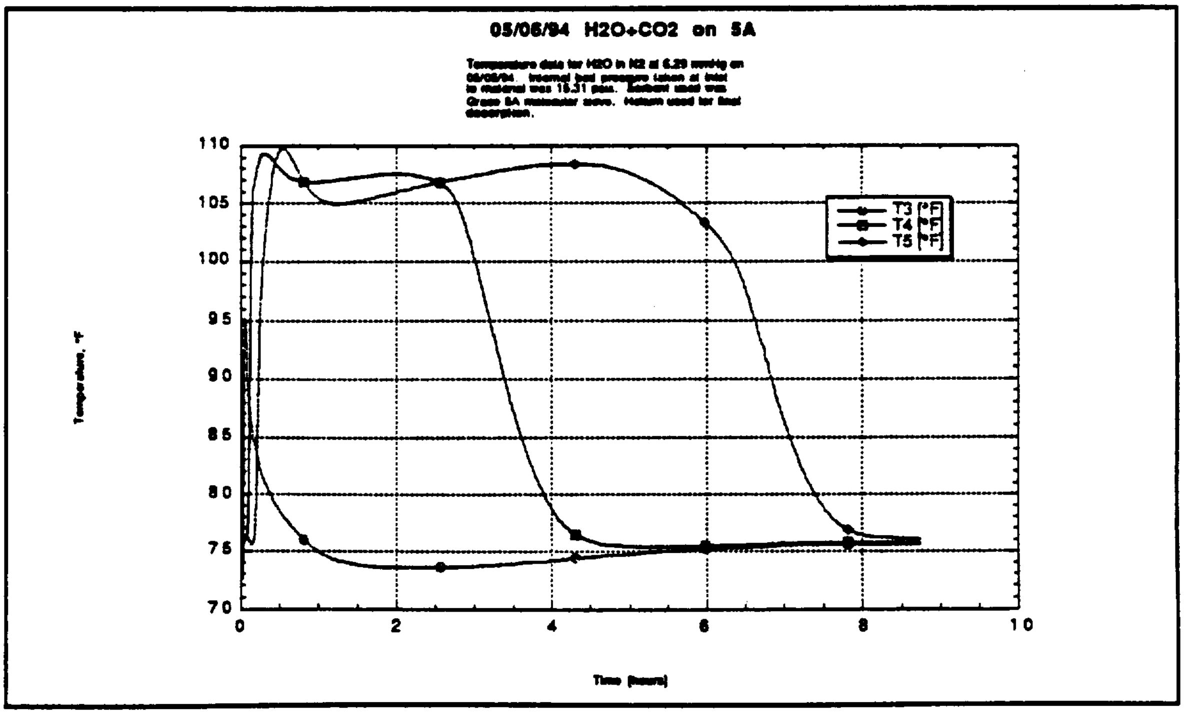


FIGURE 4-28 $H_2O/CO2$ TEMPERATURES FOR 05-06-94 TEST

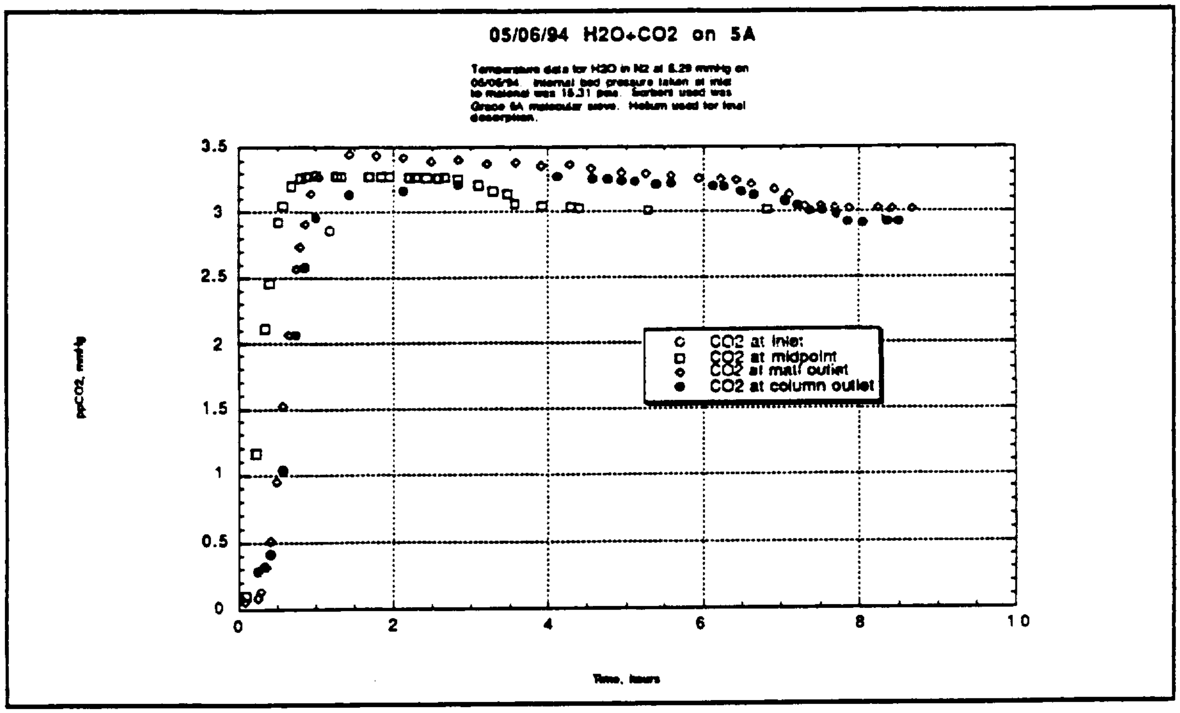


FIGURE 4-29 CO_2 BREAKTHROUGH FOR 05-06-94 TEST

4.5.1.1 FLOW1MOL Model Verification

The determination of mass transfer coefficients and verification of the single material flow-through adsorption and desorption model, FLOW1MOL, is described in this section.

Model inputs are shown in Figure 4-30 and 4-31. Determination of appropriate values for each non-obvious input parameter is discussed below (values discussed in previous sections are neglected).

Results of the model comparison are shown by the solid lines in Figures 4-32, 4-33, and 4-34. For breakthrough, the FLOW1MOL model compares well with test data at the midpoint. At the outlet, the model data (representing plug flow) falls, as expected, between the test data for the outlet at the centerline and the outlet following mixing.

For temperatures, the comparison is also fairly good, especially during the heating of the column. As noted for N_2/CO_2 adsorption, an early temperature peak corresponding to CO2 adsorption is reflected by the model. In contrast to CO_2 adsorption, the temperature profile remains steady for a large time and suddenly decreases. The period of steady state temperature profile indicates the balance between the heat generated by heat of adsorption of H_2O and the heat of desorption of CO_2 and N_2 and the heat transferring to the wall of the packed bed. The sudden decrease of temperature or the sharp breakthrough of H_2O is the reflection of high affinity of water on 5A material. This causes the adsorption of H_2O to occur in a short length of the bed, with little dispersion taking place. It is possible that in the case of water adsorption the assumption of equilibrium between the gas phase and solid phase is valid to a large extend. Mass transfer coefficients of .0035 ft/hr for H_2O and .017 ft/hr for CO_2 were used to predict the breakthrough curves.

For carbon dioxide breakthrough, the comparison of model results and test data show that FLOW1MOL properly models the trends exhibited in coadsorption, that is, the roll-up phenomenon observed earlier. Differences in magnitude are most likely due to the lack of accuracy in the equilibrium isotherms and the approximate nature of the multi-component isotherm equations. The accuracy is expected to improve as more precise isotherm data, particularly on multi-component adsorption, is received. The Langmuir isotherm for mixture of CO_2 and H_2O was used which underestimates the loading of CO_2 and H_2O on 5A. The IAST predicts this loading much closer to the test data. Since the solution to the IAST isotherm is not analytical it is costly in terms of CPU usage.

Variable	*Description*	*Units*	*Default Value*	*Current Value*
END	final time	hours	3	4
DELT1	time step	hours	0.00100	0.00300
NJ	Number of nodes	n/a	101	61
DELZ1	distance between nodes	feet	0.0083333	0.01388888
TO	Ambient temperature	°R	532.5	534.4
G F	Volumetric flow rate	CFH	62.82	57.9
T O	Inlet temperature	°R	532.5	534.4
NC	Number of components		2	3
INERT	Inert gas (1=N2; 2=He)	n/a	1	1
P TOT	Total inlet Pressure	psia	15.413	15.26
BC_C(1)	CO2 Inlet partial pressure	psia	0.059	0.05647969
BC C(2)	H2O Inlet partial pressure	psia	15.354	0.12224371
BC C(3)	N2 Inlet partial pressure	psia	0	15.0812766
BC_C(4)	O2 Inlet partial pressure	psia	0	0
Z	Bed length	feet	0.83333333	0.83333333
D E	External Diameter	feet	0.1666667	0.1666667
D I	Inside Diameter	feet	0.15625	0.15625
RO WA	Wall Density	lb/ft^3	489	489
EPSEX	void fraction (eta)	dimless	0.373	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635	635
CON S	Pellet (solid) conductivity	BTU/°F-	0.1	0.1
CP S	Pellet (solid) specific heat	BTU/°F-lb	0.2	0.2
H_FW	Heat transfer fluid to wall		2.5	2.5
H OW	Heat transfer outside to wall		0.1	0.1
RO S	Pellet density	lb/ft^3	70	75
M_W(1)	Molecular Weight	lb/lb-mol	44	44
M W(2)	Molecular Weight	lb/lb-mol	28	18
M W(3)	Molecular Weight	lb/lb-mol	18	28
M W(4)	Molecular Weight	lb/lb-mol	32	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-17000	-19000
HEAT(2)	heat of adsorption	BTU/lb-mole	-8988	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.025	0.017
K F(2)	Mass Transfer Coefficent	ft/hr	0.05	0.0035
K F(3)	Mass Transfer Coefficent	ft/hr	0	0.1
K_F(4)	Mass Transfer Coefficent	ft/hr	0	0
IND(1)	Indicator for CO2 present			1
IND(2)	Indicator for H2O present			1
IND(3)	Indicator for N2 present			1
IND(4)	Indicator for O2 present			0

FIGURE 4-30 INPUT DATA SET "A" FOR FLOW1MOL COMPARISON WITH 05-06-94 TEST

LABEL(1)	Output Labels		"ppH2O"	"ppCO2"
LABEL(2)			"pp N2"	"ppH2O"
LABEL(3)			"gas T"	"ppN2"
LABEL(4)			"tot P"	"gas T"
AOUT(1)	C array indices to output		1	1
AOUT(2)			2	2
AOUT(3)			6	3
AOUT(4)			11	7
IOUT(1)	Node locations to be output		5	2
IOUT(2)			10	15
IOUT(3)			30	30
IOUT(4)			35	45
IOUT(5)			60	61
S_B	Cross sectional surface area	ft^3	0.019175	0.019175
CON_WA			32.875	32.875
CON_WI_Q			0.29	0.29
CON_WI_K			0.19	0.19
CP_WA			0.109848	0.109848
CP_WI_Q			0.21	0.21
CP_WI_K			0.21	0.21
RO_WI_Q			3.5	3.5
RO_WI_K			16	16
X_WA	Canister Wall Thickness	feet	0.00541667	0.00541667
X_WI_Q	Q-felt insultation thickness	feet	0.04166667	0.04166667
X_WI_K	K-felt insulation thickness	feet	0.02083333	0.02083333
RA			0.00472441	0.00472441
RAV			0.00472441	0.00472441
EPSIN			0.317	0.317
RHOS			43	43
ALPHA1			575	575
R_P			0.00472441	0.00472441
D_P			0.00944882	0.00944882
BC_L1	boundary temperature for fluid		532.5	532
BC_L2	boundary temperature for solid		532.5	532
BC_L3	boundary temperature for canister wall		532.5	532
BC_L5	fluid pressure at each grid(?)		15.413	15.68
BC_L6			15.413	15.68
IN_L1	inlet temperature for fluid		532.5	532
IN_L2	inlet temperature for solid		532.5	532
IN_L3	inlet temperature for canister wall		532.5	532
IN_L5	fluid pressure at each grid(?)		15.413	15.68
IN_L6			15.413	15.68
R	Gas Constant		10.73	10.73
PI			3.141593	3.141593
ITEST			0	0

FIGURE 4-31 INPUT DATA SET "B" FOR FLOW1MOL COMPARISON WITH 05-06-94 TEST

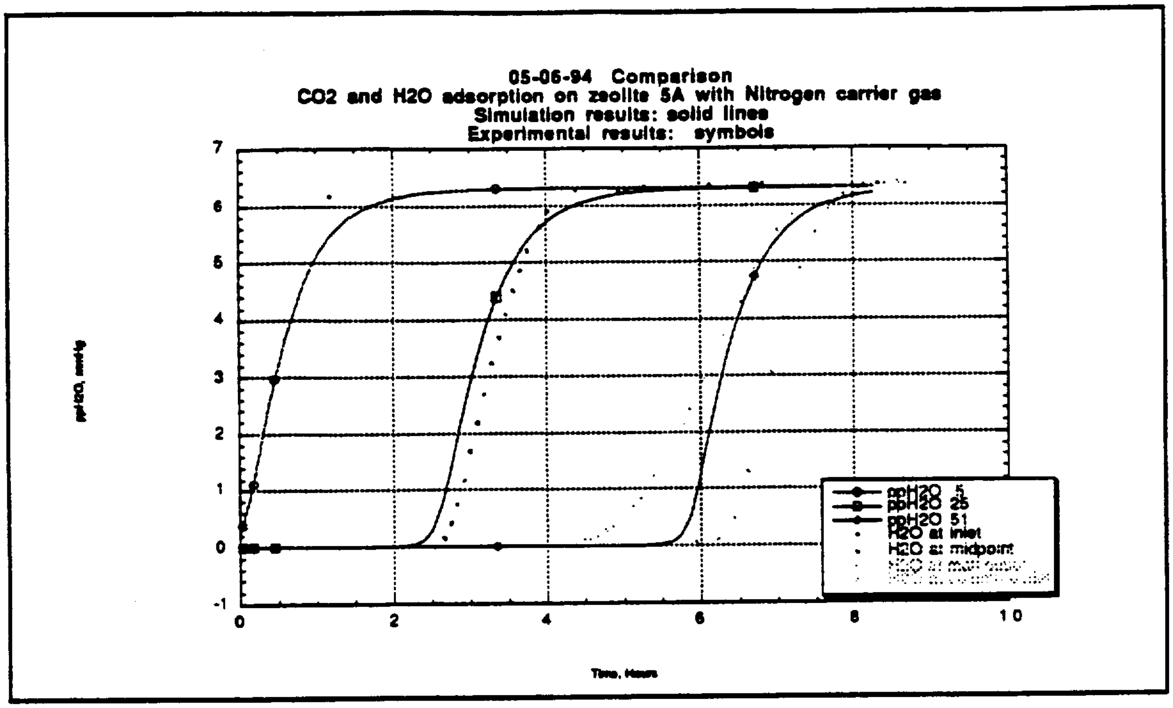


FIGURE 4-32 H_2O BREAKTHROUGH COMPARISONS FOR 05-06-94 TEST WITH FLOW1MOL RESULTS

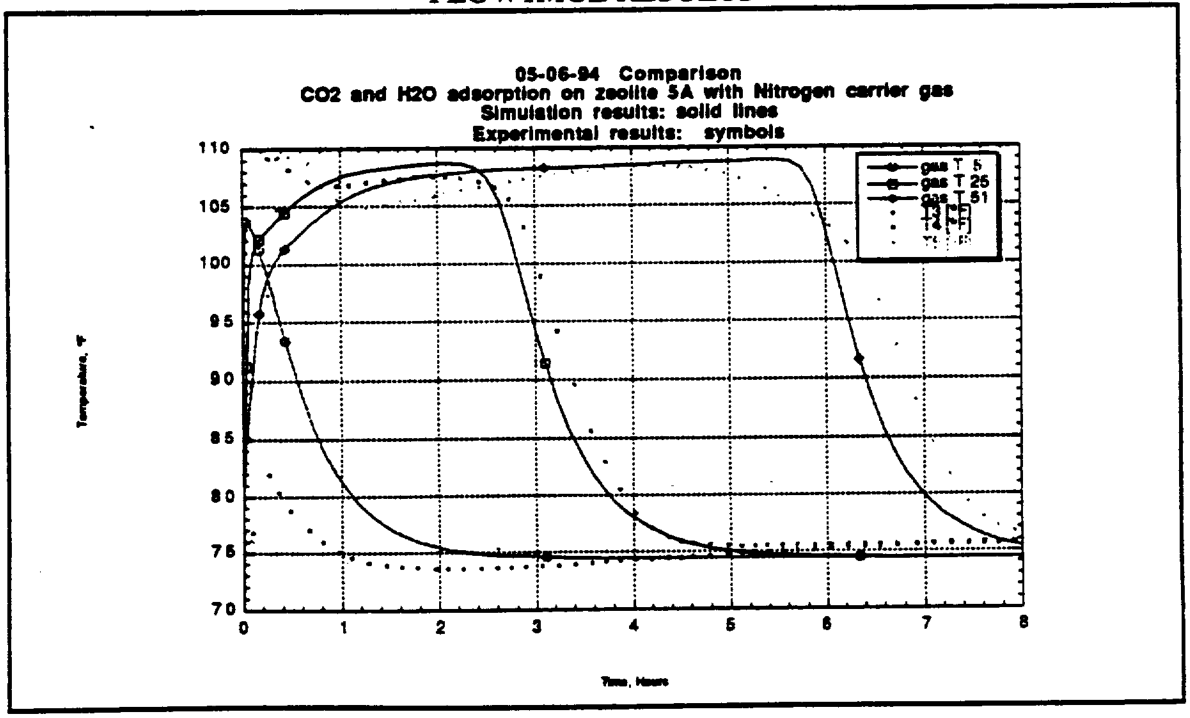


FIGURE 4-33 H_2O/CO_2 TEMPERATURE COMPARISONS FOR 05-06-94 TEST WITH FLOW1MOL RESULTS

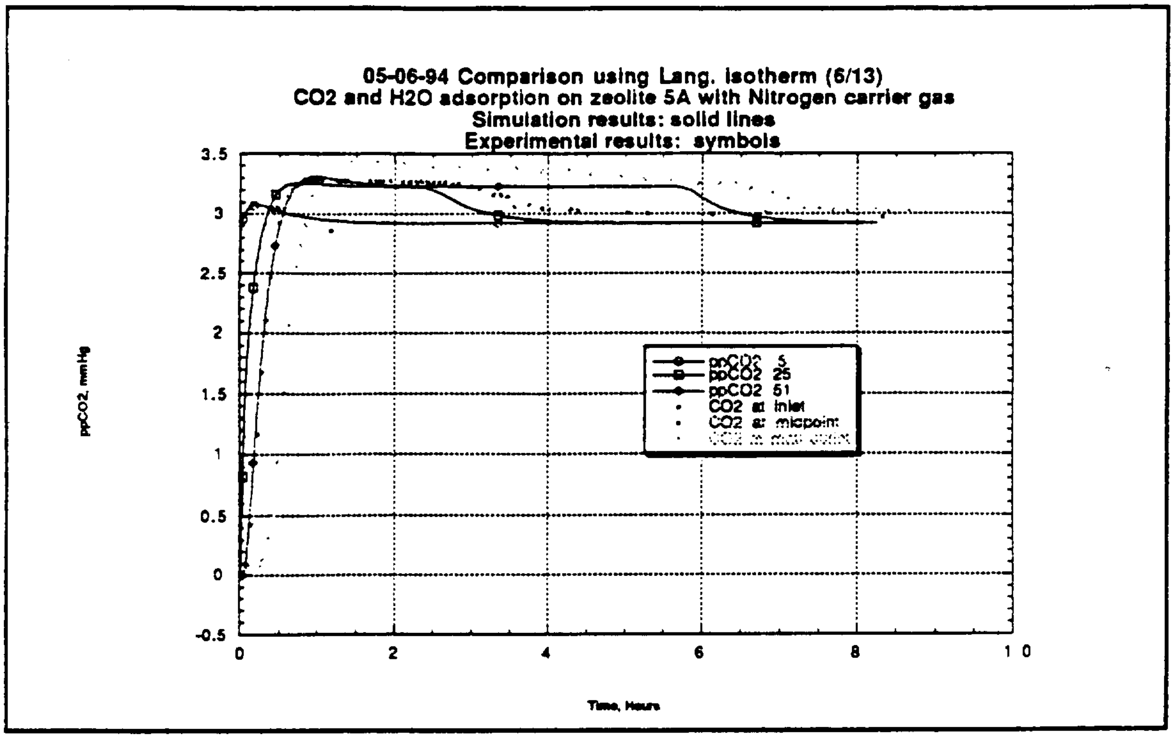


FIGURE 4-34 CO_2 BREAKTHROUGH COMPARISONS FOR 05-06-94 TEST WITH FLOW1MOL RESULTS

4.5.1.2 2DMOL Model Verification

The determination of mass transfer coefficients and verification of the two-dimensional adsorption and desorption model, 2DMOL, is described in this section.

Input data for the 2DMOL model for the 05-06-94 test are shown in Figures 4-35 and 4-36. Determination of appropriate values for each non-obvious input parameter is discussed below (values discussed in previous sections are neglected).

Results of the model comparison are shown by the solid lines in Figures 4-37 and 4-38. As expected the center line breakthrough and temperature profile match the one dimensional flow model. The average breakthrough at the midpoint and the outlet of the bed are shown. The model midpoint average is reasonably good, since the dispersion is moderate compare to the centerline breakthrough. There was no test data point for the average component partial pressure at the midpoint.

The result of the model for the average breakthrough at the outlet shows an early breakthrough in compared to the test data for breakthrough of H_2O/N_2. This can be attributed to several factors: the equilibrium isotherm, the porosity, and the radial diffusivity of H_2O as were discussed in H_2O/N_2 adsorption.

The Langmuir isotherm was used to predict loading of the H_2O/CO_2 mixture on 5A material. Langmuir isotherm for mixture of CO_2 and H_2O underestimates the loading of CO_2 and H_2O on 5A. The IAST predicts this loading much closer to the test data. Since the solution to the IAST isotherm is not analytical it is costly in terms of CPU usage.

Variable	*Description*	*Units*	*Current Value*
END	final time	hours	2
DELT	time step	hours	0.00250
NJ	Number of axial nodes	n/a	51
NJR	Number of radial nodes	n/a	20
DELZ	distance between nodes	feet	0.01666666
TAMB	Ambient temperature	°R	534.4
G_F	Volumetric flow rate	CFH	57.9
TO	Inlet temperature	°R	534.4
NC	Number of components		3
INERT	Inert gas (1=N2; 2=He)	n/a	1
P_TOT	Total Inlet Pressure	mmHg	789.26
P_C(1)	Inlet partial pressure	mmHg	2.92
P_C(2)	Inlet partial pressure	mmHg	6.31
P_C(3)	Inlet partial pressure	mmHg	779.94
P_C(4)	Inlet partial pressure	mmHg	0.00
Z	Bed length	feet	0.83333333
D_E	External Diameter	feet	0.1666667
D_I	Inside Diameter	feet	0.155833
RO_WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.25
RO_S	Pellet density	lb/ft^3	75
M_W(1)	Molecular Weight	lb/lb-mol	44
M_W(2)	Molecular Weight	lb/lb-mol	18
M_W(3)	Molecular Weight	lb/lb-mol	28
M_W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.017
K_F(2)	Mass Transfer Coefficent	ft/hr	0.0035
K_F(3)	Mass Transfer Coefficent	ft/hr	0.05
K_F(4)	Mass Transfer Coefficent	ft/hr	0.1
ERROR1			0.00000001
ERROR2			1E-09

FIGURE 4-35 INPUT DATA SET "A" FOR 2DMOL COMPARISON WITH 05-06-94 TEST

Variable	*Description*	*Units*	*Current Value*
S_B	Cross sectional surface area	ft^3	0.019069
CP_WA	Wall specific heat		0.109848
RA	not used	feet	0.00472441
RAV	not used	feet	0.0058
RHOS	not used		75
ALPHA1	not used		635
R_P	Pellet radius	feet	0.0047244
D_P	Pellet diameter	feet	0.0094488
R	Gas Constant		10.73
PI			3.141593
UINS			0.354
HWALL			12
NC	Number of components		3
T_FO	same as TO		534.4
GC			416975040
CON_S1			1
NP			125
PT	same as PTOT		789.42569
LEN	same as Z		1.66667
VOID_B	same as EPSEX		0.35
IND(1)			1
IND(2)			1
IND(3)			1
IND(4)			0
IDES1			0
ISAT			0

FIGURE 4-36 INPUT DATA SET "B" FOR 2DMOL COMPARISON WITH 05-06-94 TEST

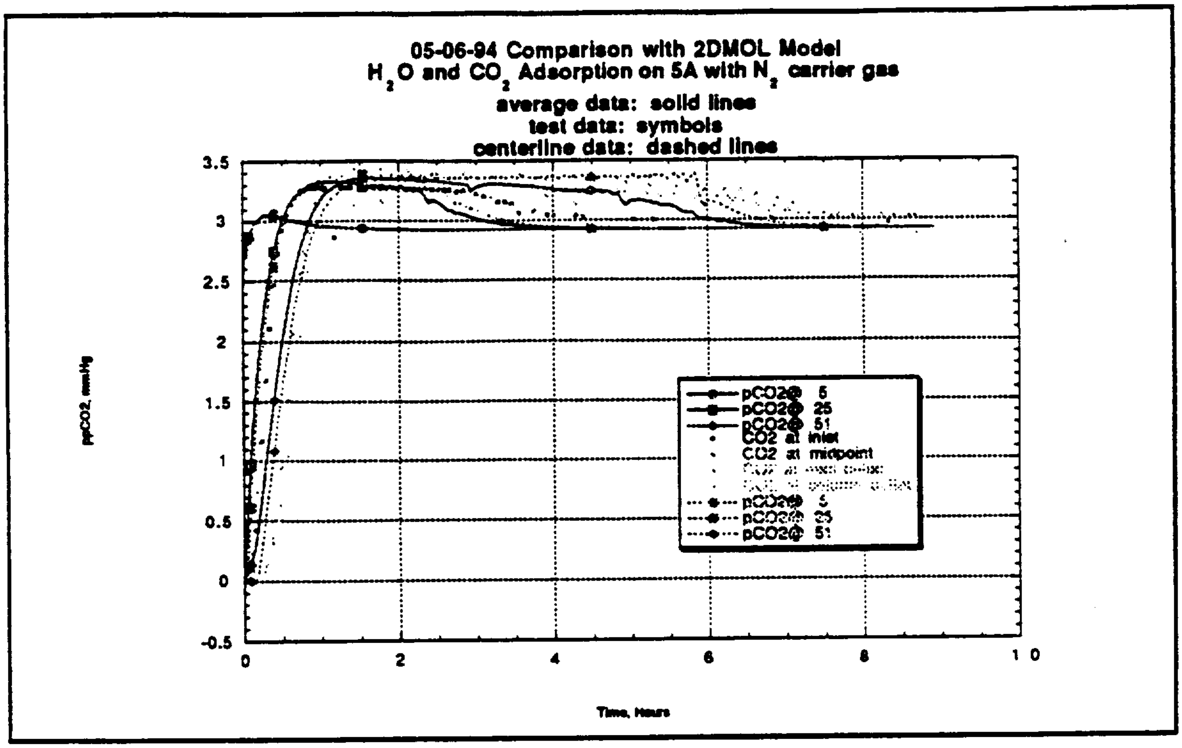


FIGURE 4-37 CO_2 BREAKTHROUGH COMPARISONS FOR 05-06-94 TEST WITH 2DMOL RESULTS

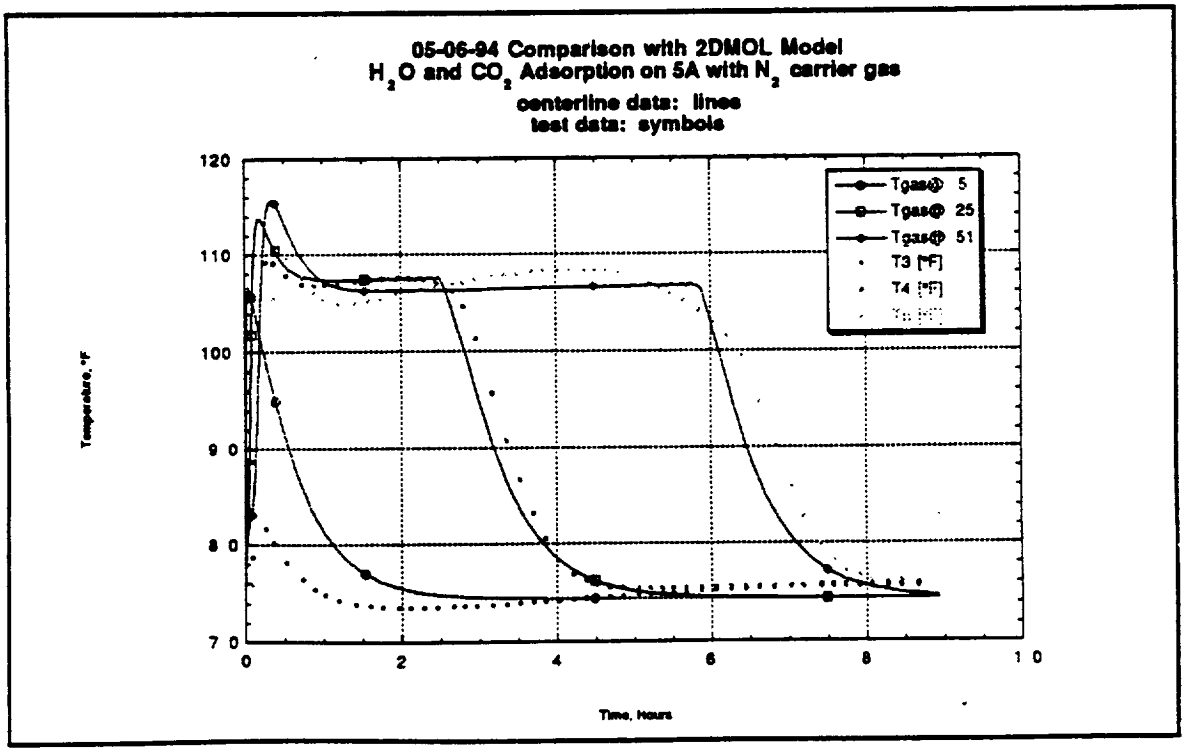


FIGURE 4-38 H_2O/CO_2 TEMPERATURE COMPARISONS FOR 05-06-94 TEST WITH 2DMOL RESULTS

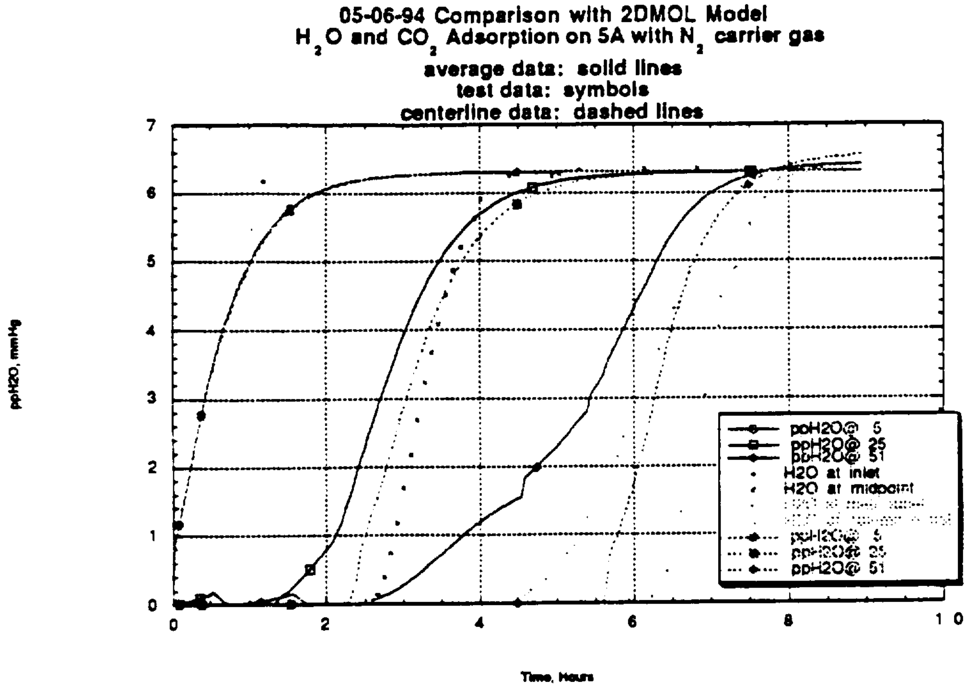
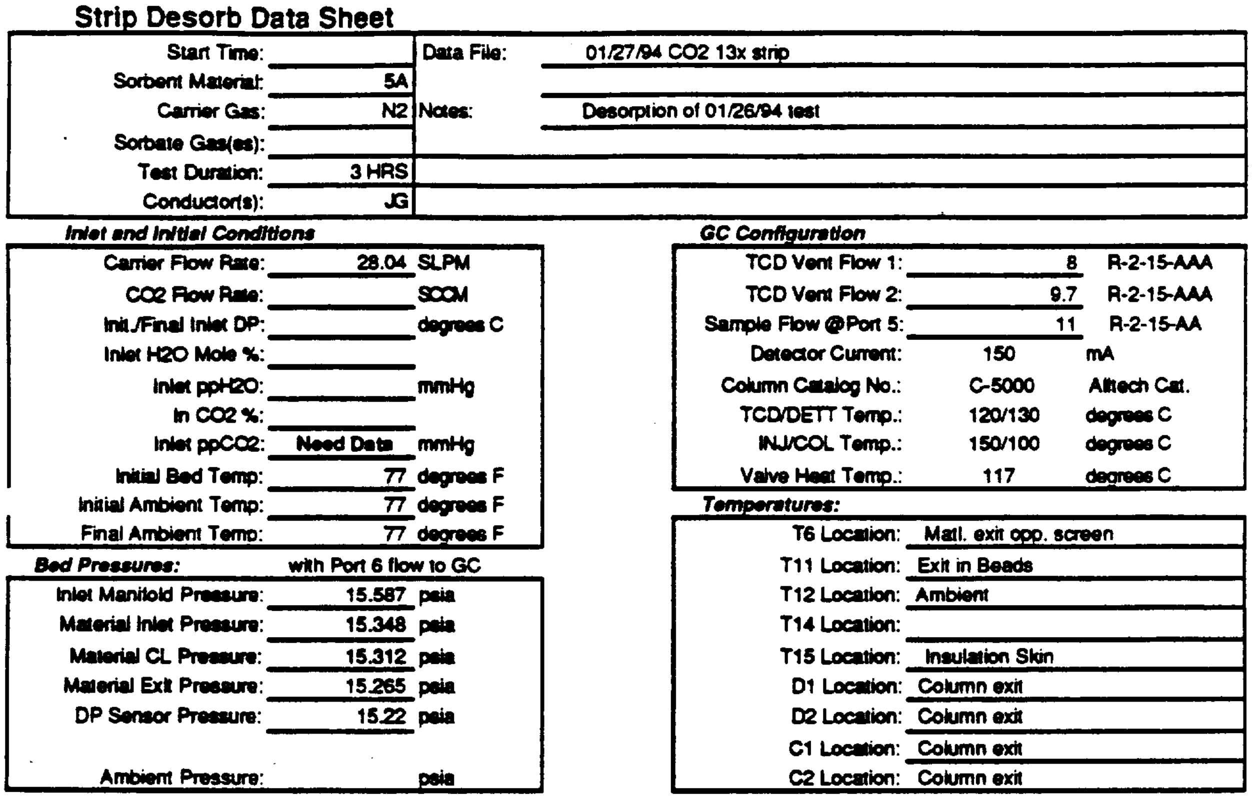


FIGURE 4-39 H_2O/CO_2 BREAKTHROUGH COMPARISONS FOR 05-06-94 TEST WITH 2DMOL RESULTS

4.6 Carbon Dioxide Desorption

4.6.1 Molecular Sieve 5A



Strip Desorb Data Sheet

Start Time:		Data File:	01/27/94 CO2 13x strip
Sorbent Material:	5A		
Carrier Gas:	N2	Notes:	Desorption of 01/26/94 test
Sorbate Gas(es):			
Test Duration:	3 HRS		
Conductor(s):	JG		

Inlet and Initial Conditions

Carrier Flow Rate:	28.04	SLPM
CO2 Flow Rate:		SCCM
Init./Final Inlet DP:		degrees C
Inlet H2O Mole %:		
Inlet ppH2O:		mmHg
In CO2 %:		
Inlet ppCO2:	Need Data	mmHg
Initial Bed Temp:	77	degrees F
Initial Ambient Temp:	77	degrees F
Final Ambient Temp:	77	degrees F

Bed Pressures: with Port 6 flow to GC

Inlet Manifold Pressure:	15.587	psia
Material Inlet Pressure:	15.348	psia
Material CL Pressure:	15.312	psia
Material Exit Pressure:	15.265	psia
DP Sensor Pressure:	15.22	psia
Ambient Pressure:		psia

GC Configuration

TCD Vent Flow 1:	8	R-2-15-AAA
TCD Vent Flow 2:	9.7	R-2-15-AAA
Sample Flow @Port 5:	11	R-2-15-AA
Detector Current:	150	mA
Column Catalog No.:	C-5000	Alltech Cat.
TCD/DETT Temp.:	120/130	degrees C
INJ/COL Temp.:	150/100	degrees C
Valve Heat Temp.:	117	degrees C

Temperatures:

T6 Location:	Matl. exit opp. screen
T11 Location:	Exit in Beads
T12 Location:	Ambient
T14 Location:	
T15 Location:	Insulation Skin
D1 Location:	Column exit
D2 Location:	Column exit
C1 Location:	Column exit
C2 Location:	Column exit

FIGURE 4-40 SPECIFICATIONS FOR 01-27-94 CO_2 DESORPTION TESTING

The test ran on 01-27-94 will be used to verify the FLOW1MOL and 2DMOL bed models for carbon dioxide desorption. Test specifications are shown in Figure 4-40. Breakthrough for CO_2 and temperatures will be shown in the comparative plots in the following sections.

4.6.1.1 FLOW1MOL Model Verification

The determination of mass transfer coefficients and verification of the single material flow-through desorption and desorption model, FLOW1MOL, is described in this section.

Model inputs are shown in Figure 4-41 and 4-42.

Results of the model comparison are shown by the solid lines in Figures 4-43 and 4-44. After the bed was saturated with CO_2, the regeneration process was started by using N_2 as the purge gas. As it is shown the effluent CO_2 concentration has a sharp drop in the first few minutes and the slope of the breakthrough flattens out as time goes by. The initial drop in temperature is due to heat of desorption and finally reaches the inlet condition when there is no depletion of CO_2 from the bed. The same mass transfer coefficient of .017 was used. Model predictions of temperature and breakthrough agree well with test data. The IAST was used to predict the mixture isotherm of CO_2/N_2.

Variable	Description	Units	Default Value	Current Value
END	final time	hours	3	2
DELT1	time step	hours	0.00100	0.00300
NJ	Number of nodes	n/a	101	51
DELZ1	distance between nodes	feet	0.0083333	0.016666
TO	Ambient temperature	°R	532.5	537
G_F	Volumetric flow rate	CFH	62.82	55.26
T_O	Inlet temperature	°R	532.5	537
NC	Number of components		2	2
INERT	Inert gas (1=N2; 2=He)	n/a	1	1
P_TOT	Total Inlet Pressure	psia	15.413	15.265
BC_C(1)	CO2 Inlet partial pressure	psia	0.059	0.1185348
BC_C(2)	H2O Inlet partial pressure	psia	15.354	0
BC_C(3)	N2 Inlet partial pressure	psia	0	15.265
BC_C(4)	O2 Inlet partial pressure	psia	0	0
Z	Bed length	feet	0.83333333	0.83333333
D_E	External Diameter	feet	0.1666667	0.1666667
D_I	Inside Diameter	feet	0.15625	0.15625
RO_WA	Wall Density	lb/ft^3	489	489
EPSEX	void fraction (eta)	dimless	0.373	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635	635
CON_S	Pellet (solid) conductivity	BTU/°F-	0.1	0.1
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.2	0.25
H_FW	Heat transfer fluid to wall		2.5	2
H_OW	Heat transfer outside to wall		0.1	0.1
RO_S	Pellet density	lb/ft^3	70	75
M_W(1)	Molecular Weight	lb/lb-mol	44	44
M_W(2)	Molecular Weight	lb/lb-mol	28	18
M_W(3)	Molecular Weight	lb/lb-mol	18	28
M_W(4)	Molecular Weight	lb/lb-mol	32	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-17000	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-8988	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.025	0.017
K_F(2)	Mass Transfer Coefficent	ft/hr	0.05	0.0035
K_F(3)	Mass Transfer Coefficent	ft/hr	0	0.05
K_F(4)	Mass Transfer Coefficent	ft/hr	0	0
IND(1)	Indicator for CO2 present			1
IND(2)	Indicator for H2O present			0
IND(3)	Indicator for N2 present			1
IND(4)	Indicator for O2 present			0
IDES1	Flag to indicate if bed ungoing desorption			0
ISAT	Flag to indicate prior saturation of bed			1

FIGURE 4-41 INPUT DATA SET "A" FOR FLOW1MOL COMPARISON WITH 01-27-94 TEST

LABEL(1)	Output Labels		"ppH2O"	'ppCO2'
LABEL(2)			"pp N2"	'ppN2'
LABEL(3)			"gas T"	'ppN2'
LABEL(4)			"tot P"	'gas T'
AOUT(1)	C array indices to output		1	1
AOUT(2)			2	2
AOUT(3)			6	5
AOUT(4)			11	6
IOUT(1)	Node locations to be output		5	5
IOUT(2)			10	15
IOUT(3)			30	25
IOUT(4)			35	35
IOUT(5)			60	51
S_B	Cross sectional surface area	ft^3	0.019175	0.019175
CON_WA			32.875	32.875
CON_WI_Q			0.29	0.29
CON_WI_K			0.19	0.19
CP_WA			0.109848	0.109848
CP_WI_Q			0.21	0.21
CP_WI_K			0.21	0.21
RO_WI_Q			3.5	3.5
RO_WI_K			16	16
X_WA	Canister Wall Thickness	feet	0.00541667	0.00541667
X_WI_Q	Q-felt insultation thickness	feet	0.04166667	0.04166667
X_WI_K	K-felt insulation thickness	feet	0.02083333	0.02083333
RA			0.00472441	0.0045893
RAV			0.00472441	0.0091785
EPSIN			0.317	0.317
RHOS			43	43
ALPHA1			575	575
R_P			0.00472441	0.00472441
D_P			0.00944882	0.00944882
BC_L1	boundary temperature for fluid		532.5	537
BC_L2	boundary temperature for solid		532.5	537
BC_L3	boundary temperature for canister wall		532.5	532
BC_L5	fluid pressure at each grid(?)		15.413	15.265
BC_L6			15.413	15.265
IN_L1	inlet temperature for fluid		532.5	537
IN_L2	inlet temperature for solid		532.5	537
IN_L3	inlet temperature for canister wall		532.5	537
IN_L5	fluid pressure at each grid(?)		15.413	15.265
IN_L6			15.413	15.265
R	Gas Constant		10.73	10.73
PI			3.141593	3.141593
ITEST			0	0

FIGURE 4-42 INPUT DATA SET "B" FOR FLOW1MOL COMPARISON WITH 01-27-94 TEST

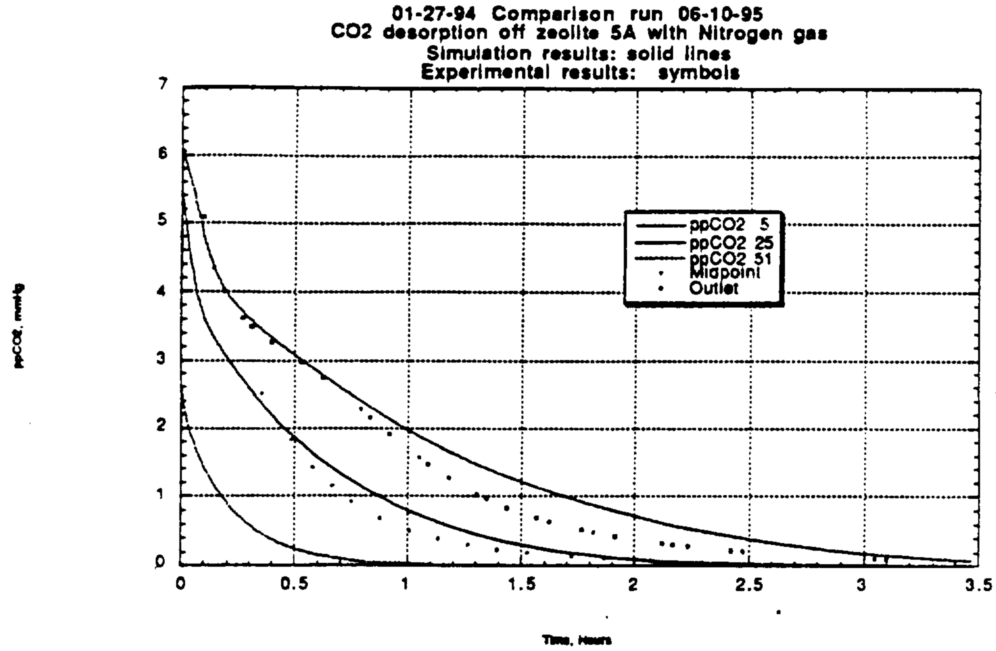


FIGURE 4-43 CO_2 BREAKTHROUGH COMPARISONS FOR 01-27-94 TEST WITH FLOW1MOL RESULTS

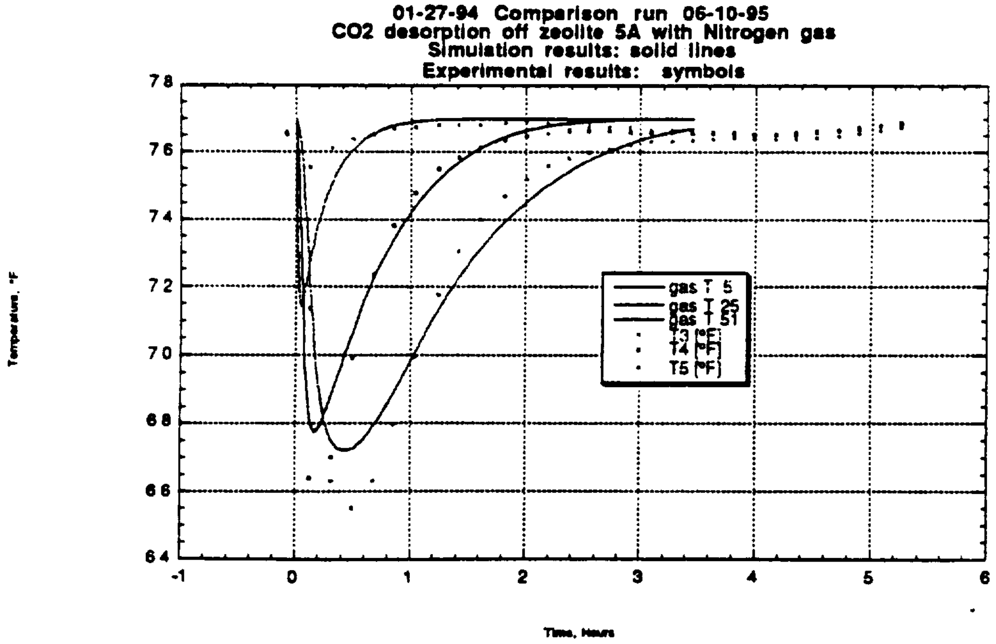


FIGURE 4-44 CO_2 TEMPERATURE COMPARISONS FOR 01-27-94 TEST WITH FLOW1MOL RESULTS

4.6.1.2 2DMOL Model Verification

The determination of mass transfer coefficients and verification of the two-dimensional desorption model, 2DMOL, is described in this section.

Input data for the 2DMOL model for the 01-27-94 test are shown in Figures 45 and 46.

Results of the model comparison are shown by the solid lines in Figures 4-47 and 4-48. The model prediction of centerline and average breakthrough matches the obtained experimental data. The temperature profile result of the two dimensional model also estimates the experimental data fairly well. The few degree discrepancy between the model and experimental data is largely due to thermal conductivity profile estimation which has a strong effect on radial temperature profile. Trial and error must be used in order to obtain a corrected thermal conductivity equation. The IAST was used to predict the CO_2/N_2 mixture isotherm. A mass transfer coefficient of .017 ft/hr was used for the prediction of breakthrough curve.

Variable	*Description*	*Units*	*Current Value*
END	final time	hours	2
DELT	time step	hours	0.00250
NJ	Number of axial nodes	n/a	51
NJR	Number of radial nodes	n/a	20
DELZ	distance between nodes	feet	0.01666666
TAMB	Ambient temperature	°R	537
G_F	Volumetric flow rate	CFH	55.6
TO	Inlet temperature	°R	537
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P_TOT	Total Inlet Pressure	mmHg	789.43
P_C(1)	Inlet partial pressure	mmHg	6.13
P_C(2)	Inlet partial pressure	mmHg	0.00
P_C(3)	Inlet partial pressure	mmHg	783.30
P_C(4)	Inlet partial pressure	mmHg	0.00
Z	Bed length	feet	0.83333333
D_E	External Diameter	feet	0.1666667
D_I	Inside Diameter	feet	0.155833
RO_WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.25
RO_S	Pellet density	lb/ft^3	75
M_W(1)	Molecular Weight	lb/lb-mol	44
M_W(2)	Molecular Weight	lb/lb-mol	18
M_W(3)	Molecular Weight	lb/lb-mol	28
M_W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.017
K_F(2)	Mass Transfer Coefficent	ft/hr	0.0035
K_F(3)	Mass Transfer Coefficent	ft/hr	0.05
K_F(4)	Mass Transfer Coefficent	ft/hr	0.1
ERROR1			0.00000001
ERROR2			1E-09

FIGURE 4-45 INPUT DATA SET "A" FOR 2DMOL COMPARISON WITH 01-27-94 TEST

Variable	*Description*	*Units*	*Current Value*
S_B	Cross sectional surface area	ft^3	0.019069
CP_WA	Wall specific heat		0.109848
RA	not used	feet	0.00472441
RAV	not used	feet	0.0058
RHOS	not used		75
ALPHA1	not used		635
R_P	Pellet radius	feet	0.0047244
D_P	Pellet diameter	feet	0.0094488
R	Gas Constant		10.73
PI			3.141593
UINS			0.354
HWALL			12
NC	Number of components		2
T_FO	same as TO		537
GC			416975040
CON_S1			1
NP			125
PT	same as PTOT		789.42569
LEN	same as Z		1.66667
VOID_B	same as EPSEX		0.35
IND(1)			1
IND(2)			0
IND(3)			1
IND(4)			0
IDES1			0
ISAT			1

FIGURE 4-46 INPUT DATA SET "B" FOR 2DMOL COMPARISON WITH 01-27-94 TEST

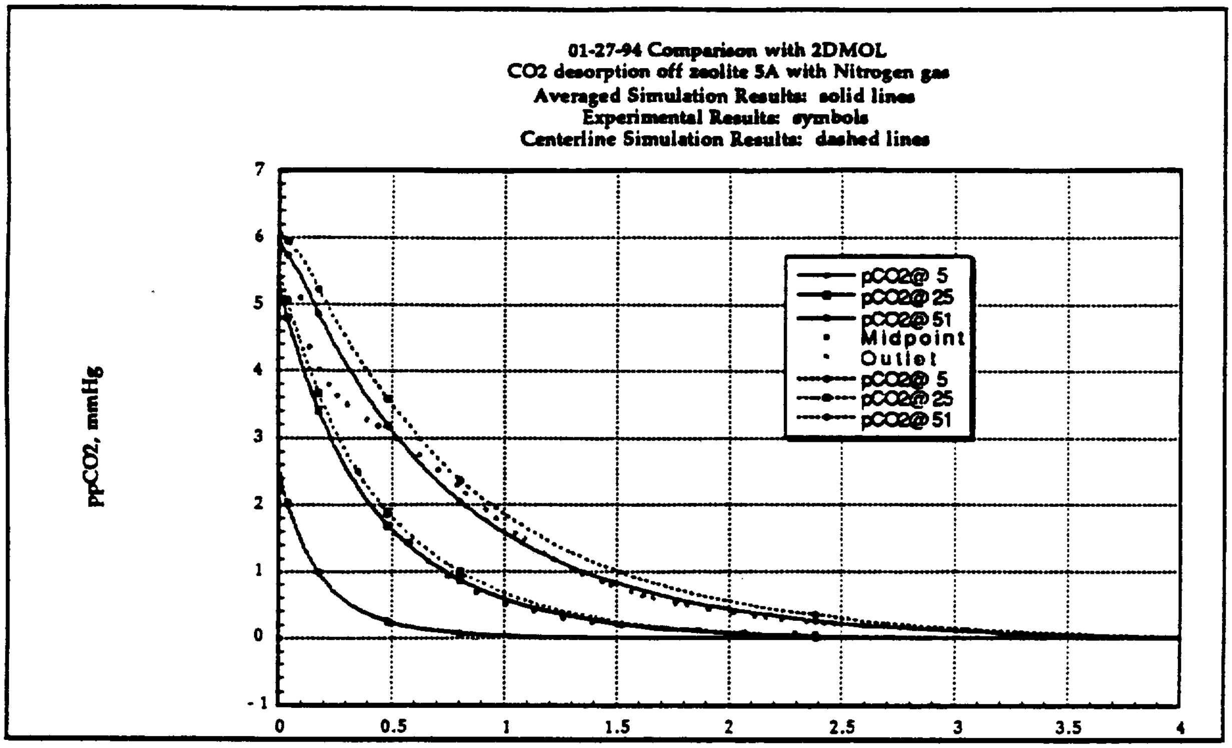


FIGURE 4-47 CO_2 BREAKTHROUGH COMPARISONS FOR 01-27-94 TEST WITH 2DMOL RESULTS

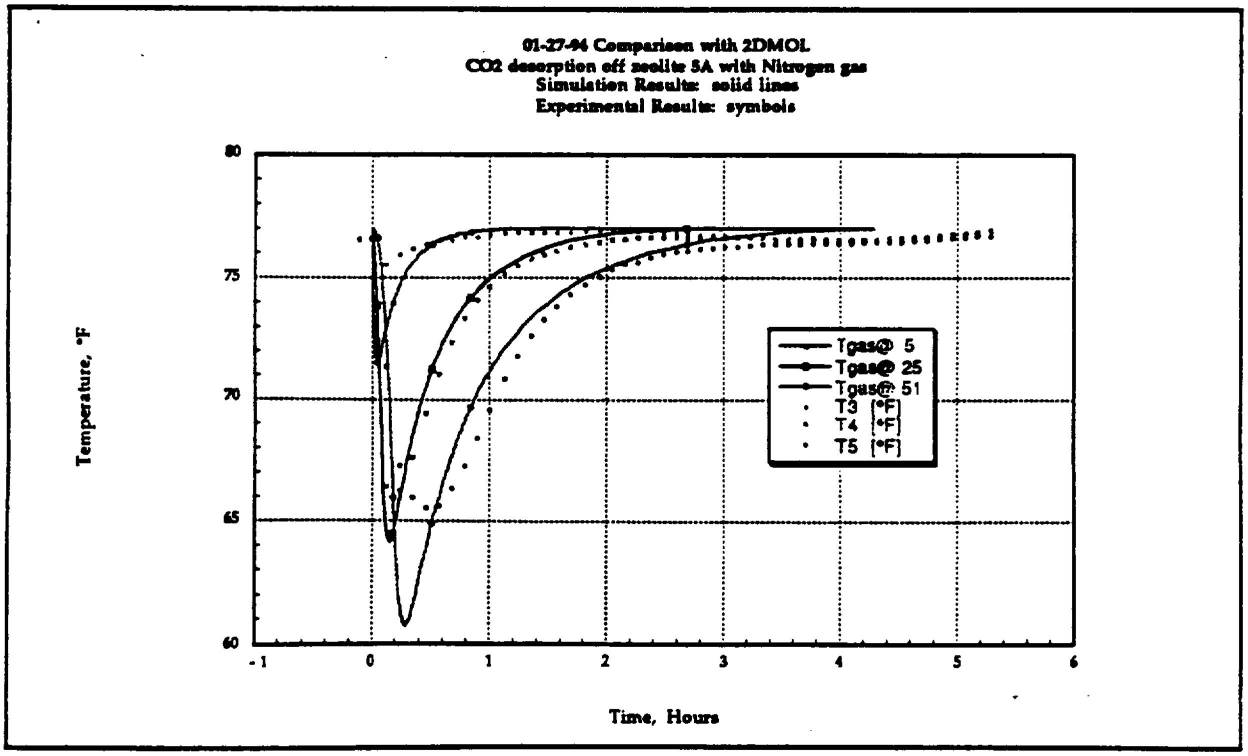
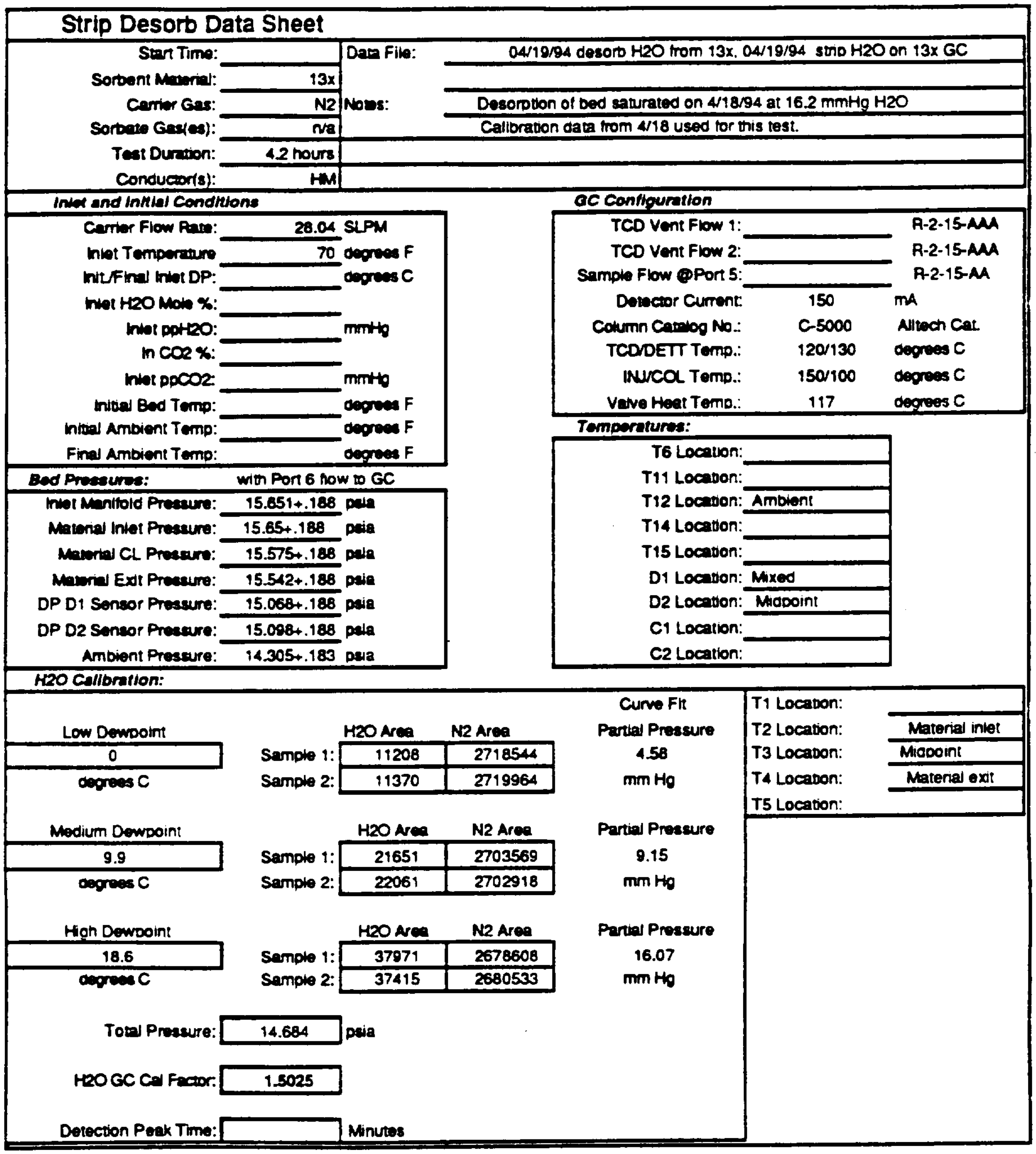


FIGURE 4-48 CO_2 TEMPERATURE COMPARISONS FOR 01-27-94 TEST WITH 2DMOL RESULTS

4.7 Water Desorption

4.7.1 Molecular Sieve 5A



Strip Desorb Data Sheet

Start Time:
Sorbent Material: 13x
Carrier Gas: N2
Sorbate Gas(es): n/a
Test Duration: 4.2 hours
Conductor(s): HM

Data File: 04/19/94 desorb H2O from 13x, 04/19/94 strip H2O on 13x GC
Notes: Desorption of bed saturated on 4/18/94 at 16.2 mmHg H2O
Calibration data from 4/18 used for this test.

Inlet and Initial Conditions

Carrier Flow Rate: 28.04 SLPM
Inlet Temperature 70 degrees F
Init./Final Inlet DP: degrees C
Inlet H2O Mole %:
Inlet ppH2O: mmHg
In CO2 %:
Inlet ppCO2: mmHg
Initial Bed Temp: degrees F
Initial Ambient Temp: degrees F
Final Ambient Temp: degrees F

Bed Pressures: with Port 6 flow to GC

Inlet Manifold Pressure: 15.651+.188 psia
Material Inlet Pressure: 15.65+.188 psia
Material CL Pressure: 15.575+.188 psia
Material Exit Pressure: 15.542+.188 psia
DP D1 Sensor Pressure: 15.068+.188 psia
DP D2 Sensor Pressure: 15.098+.188 psia
Ambient Pressure: 14.305+.183 psia

GC Configuration

TCD Vent Flow 1: R-2-15-AAA
TCD Vent Flow 2: R-2-15-AAA
Sample Flow @Port 5: R-2-15-AA
Detector Current: 150 mA
Column Catalog No.: C-5000 Alltech Cat.
TCD/DETT Temp.: 120/130 degrees C
INJ/COL Temp.: 150/100 degrees C
Valve Heat Temp.: 117 degrees C

Temperatures:

T6 Location:
T11 Location:
T12 Location: Ambient
T14 Location:
T15 Location:
D1 Location: Mixed
D2 Location: Midpoint
C1 Location:
C2 Location:

H2O Calibration:

		H2O Area	N2 Area	Curve Fit Partial Pressure
Low Dewpoint 0 degrees C	Sample 1:	11208	2718544	4.58
	Sample 2:	11370	2719964	mm Hg
Medium Dewpoint 9.9 degrees C	Sample 1:	21651	2703569	9.15
	Sample 2:	22061	2702918	mm Hg
High Dewpoint 18.6 degrees C	Sample 1:	37971	2678608	16.07
	Sample 2:	37415	2680533	mm Hg

Total Pressure: 14.684 psia
H2O GC Cal Factor: 1.5025
Detection Peak Time: Minutes

T1 Location:
T2 Location: Material inlet
T3 Location: Midpoint
T4 Location: Material exit
T5 Location:

FIGURE 4-49 SPECIFICATIONS FOR 04-19-94 H2O DESORPTION TESTING

The test ran on 04-19-94 will be used to verify the FLOW1MOL and 2DMOL bed models for water desorption. Test specifications are shown in Figure 4-49. Breakthrough for H2O and temperatures will be shown in the comparative plots in the following sections.

4.7.1.1 FLOW1MOL Model Verification

The determination of mass transfer coefficients and verification of the single material flow-through desorption and desorption model, FLOW1MOL, is described in this section.

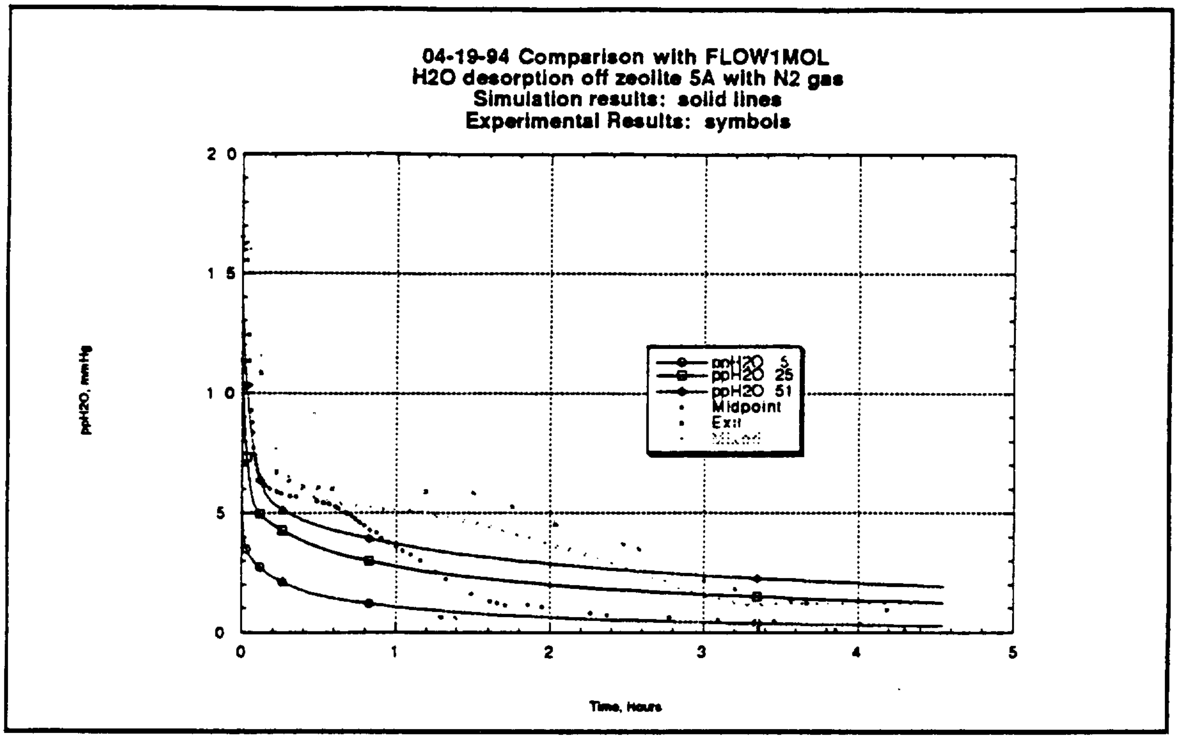
Model inputs are shown in Figures 4-50 and 4-51. Results of the model comparison are shown by the solid lines in Figures 4-52 and 4-53. The results of the partial pressure of H_2O and temperature profile of 1-D desorption model are not in good agreement with the test data. The model shows a fast reduction of H_2O partial pressure in the gas phase. A mass transfer coefficient as large as .04 ft/hr, in contrast with .0035 ft/hr in the case of H_2O adsorption, was used. However, there shouldn't be such a large difference mass transfer coefficients in adsorption versus desorption process for the same component. Even with this large mass transfer coefficient , the desorption of H_2O from the bed was not enough to increase the H_2O partial pressure in the gas phase. In contrast with adsorption process, any small discrepancy of H_2O partial pressure with test data will remain as an error throughout the completion of the test. In adsorption any small error at any point in the bed, if it caused by the isotherm at some partial pressure of the feed will be eliminated at a later time because of the correct value of isotherm at a larger partial pressure of the feed. This can be seen from the early breakthrough of most works done by previous researchers, (Hirpyuki, et al 1982, Rice, 1982). The obvious reason is that the equilibrium isotherm at low partial pressures are being underestimated. It is also possible that the adsorption/desorption of H_2O on 5A material could be modeled based on the assumption of equilibrium, where there is no resistance between the two phases. The other possible source of error is how the H_2O partial pressure was measured. Perhaps a more sensitive instrument should have been used in the test.

Variable	Description	Units	Default Value	Current Value
END	final time	hours	3	2
DELT1	time step	hours	0.00100	0.00300
NJ	Number of nodes	n/a	101	51
DELZ1	distance between nodes	feet	0.0083333	0.016666
TO	Ambient temperature	°R	532.5	531
G_F	Volumetric flow rate	CFH	62.82	55.26
T_O	Inlet temperature	°R	532.5	531
NC	Number of components		2	2
INERT	Inert gas (1=N2; 2=He)	n/a	1	1
P_TOT	Total Inlet Pressure	psia	15.413	15.55
BC_C(1)	CO2 Inlet partial pressure	psia	0.059	0
BC_C(2)	H2O Inlet partial pressure	psia	15.354	0.3132568
BC_C(3)	N2 Inlet partial pressure	psia	0	15.55
BC_C(4)	O2 Inlet partial pressure	psia	0	0
Z	Bed length	feet	0.83333333	0.83333333
D_E	External Diameter	feet	0.1666667	0.1666667
D_I	Inside Diameter	feet	0.15625	0.15625
RO_WA	Wall Density	lb/ft^3	489	489
EPSEX	void fraction (eta)	dimless	0.373	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635	635
CON_S	Pellet (solid) conductivity	BTU/°F-	0.1	0.1
CP_S	Pellet (solid) specific heat	BTU/°F-lb	0.2	0.25
H_FW	Heat transfer fluid to wall		2.5	2
H_OW	Heat transfer outside to wall		0.1	0.1
RO_S	Pellet density	lb/ft^3	70	75
M_W(1)	Molecular Weight	lb/lb-mol	44	44
M_W(2)	Molecular Weight	lb/lb-mol	28	18
M_W(3)	Molecular Weight	lb/lb-mol	18	28
M_W(4)	Molecular Weight	lb/lb-mol	32	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-17000	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-8988	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988	-8988
K_F(1)	Mass Transfer Coefficent	ft/hr	0.025	0.017
K_F(2)	Mass Transfer Coefficent	ft/hr	0.05	0.02
K_F(3)	Mass Transfer Coefficent	ft/hr	0	0.05
K_F(4)	Mass Transfer Coefficent	ft/hr	0	0
IND(1)	Indicator for CO2 present			0
IND(2)	Indicator for H2O present			1
IND(3)	Indicator for N2 present			1
IND(4)	Indicator for O2 present			0
IDES1	Flag to indicate if bed ungoing desorption			0
ISAT	Flag to indicate prior saturation of bed			1

FIGURE 4-50 INPUT DATA SET "A" FOR FLOW1MOL COMPARISON WITH 04-19-94 TEST

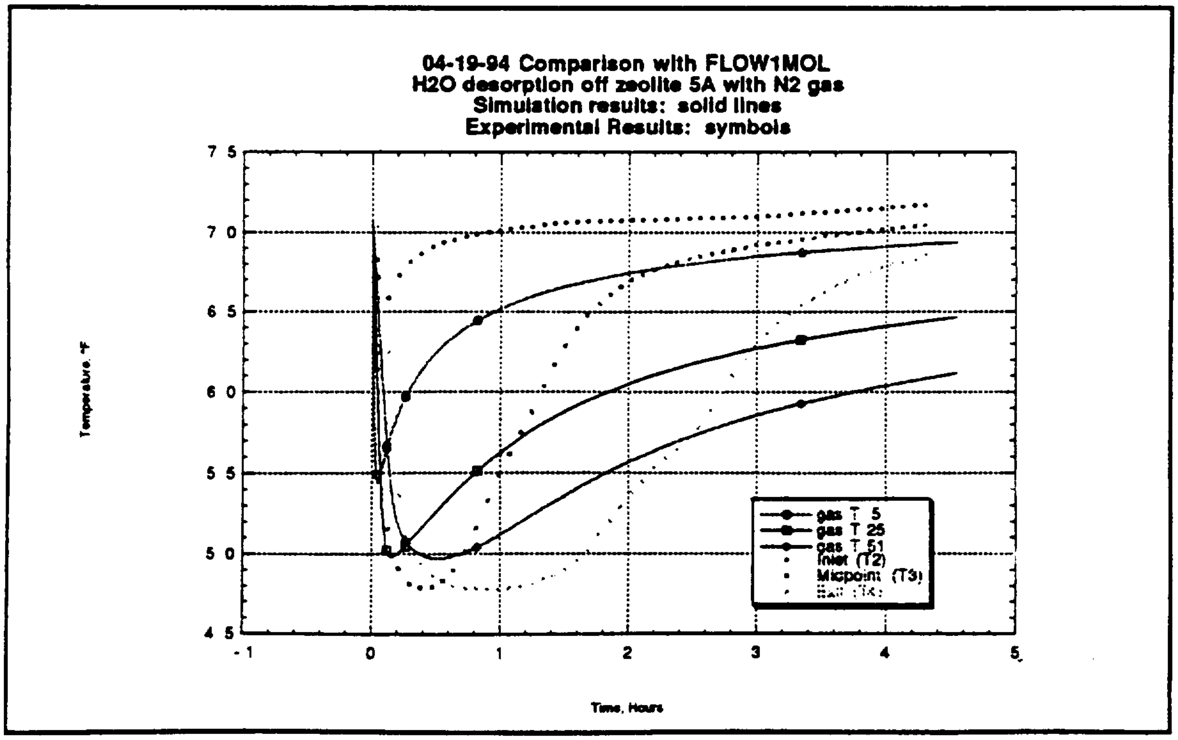
LABEL(1)	Output Labels		"ppH2O"	'ppCO2'
LABEL(2)			"pp N2"	'ppN2'
LABEL(3)			"gas T"	'ppN2'
LABEL(4)			"tot P"	'gas T'
AOUT(1)	C array indices to output		1	1
AOUT(2)			2	2
AOUT(3)			6	5
AOUT(4)			11	6
IOUT(1)	Node locations to be output		5	5
IOUT(2)			10	15
IOUT(3)			30	25
IOUT(4)			35	35
IOUT(5)			60	51
S_B	Cross sectional surface area	ft^3	0.019175	0.019175
CON_WA			32.875	32.875
CON_WI_Q			0.29	0.29
CON_WI_K			0.19	0.19
CP_WA			0.109848	0.109848
CP_WI_Q			0.21	0.21
CP_WI_K			0.21	0.21
RO_WI_Q			3.5	3.5
RO_WI_K			16	16
X_WA	Canister Wall Thickness	feet	0.00541667	0.00541667
X_WI_Q	Q-felt insultation thickness	feet	0.04166667	0.04166667
X_WI_K	K-felt insulation thickness	feet	0.02083333	0.02083333
RA			0.00472441	0.0045893
RAV			0.00472441	0.0091785
EPSIN			0.317	0.317
RHOS			43	43
ALPHA1			575	575
R_P			0.00472441	0.00472441
D_P			0.00944882	0.00944882
BC_L1	boundary temperature for fluid		532.5	531
BC_L2	boundary temperature for solid		532.5	531
BC_L3	boundary temperature for canister wall		532.5	531
BC_L5	fluid pressure at each grid(?)		15.413	15.55
BC_L6			15.413	15.55
IN_L1	inlet temperature for fluid		532.5	531
IN_L2	inlet temperature for solid		532.5	531
IN_L3	inlet temperature for canister wall		532.5	531
IN_L5	fluid pressure at each grid(?)		15.413	15.55
IN_L6			15.413	15.55
R	Gas Constant		10.73	10.73
PI			3.141593	3.141593
ITEST2			0	0
ITEST1				1

FIGURE 4-51 INPUT DATA SET "B" FOR FLOW1MOL COMPARISON WITH 04-19-94 TEST



04-19-94 Comparison with FLOW1MOL
H2O desorption off zeolite 5A with N2 gas
Simulation results: solid lines
Experimental Results: symbols

FIGURE 4-52 H2O BREAKTHROUGH COMPARISONS FOR 04-19-94 TEST WITH FLOW1MOL RESULTS



04-19-94 Comparison with FLOW1MOL
H2O desorption off zeolite 5A with N2 gas
Simulation results: solid lines
Experimental Results: symbols

FIGURE 4-53 TEMPERATURE COMPARISONS FOR 04-19-94 TEST WITH FLOW1MOL RESULTS

4.7.1.2 2DMOL Model Verification

The determination of mass transfer coefficients and verification of the two-dimensional desorption and desorption model, 2DMOL, is described in this section.

Input data for the 2DMOL model for the 04-19-94 test are shown in Figures 4-54 and 4-55. Results of the model comparison are shown by the solid lines in Figures 4-56 and 4-57. The results of the partial pressure of H_2O and temperature profile of 2-D desorption model are better than the 1-D model. The model shows a fast reduction of H_2O partial pressure in gas phase as in 1-D model, but much closer to the test data. A mass transfer coefficient of .0035 ft/hr was chosen as in the case of H_2O adsorption. The same argument can be made as in 1-D on the discrepancies between model results and the test data. However, since the 2-D results are in better agreement with the test data, one can conclude the significant effect radial dispersion has on the adsorption/desorption of H_2O on the 5A material. More study and accurate testing are needed in the case of water desorption on 5A material.

Variable	*Description*	*Units*	*Current Value*
END	final time	hours	5
DELT	time step	hours	0.00250
NJ	Number of axial nodes	n/a	51
NJR	Number of radial nodes	n/a	20
DELZ	distance between nodes	feet	0.01666666
TAMB	Ambient temperature	°R	532
G F	Volumetric flow rate	CFH	56.12
TO	Inlet temperature	°R	532
NC	Number of components		2
INERT	Inert gas (1=N2; 2=He)	n/a	1
P TOT	Total Inlet Pressure	mmHg	837.78
P C(1)	Inlet partial pressure	mmHg	0.00
P C(2)	Inlet partial pressure	mmHg	16.20
P C(3)	Inlet partial pressure	mmHg	821.58
P C(4)	Inlet partial pressure	mmHg	0.00
Z	Bed length	feet	0.83333333
D E	External Diameter	feet	0.1666667
D I	Inside Diameter	feet	0.155833
RO WA	Wall Density	lb/ft^3	489
EPSEX	void fraction (eta)	dimless	0.35
AINT	Interfacial Surface Area	ft^2/ft^3	635
CP S	Pellet (solid) specific heat	BTU/°F-lb	0.25
RO S	Pellet density	lb/ft^3	75
M W(1)	Molecular Weight	lb/lb-mol	44
M W(2)	Molecular Weight	lb/lb-mol	18
M W(3)	Molecular Weight	lb/lb-mol	28
M W(4)	Molecular Weight	lb/lb-mol	32
HEAT(1)	heat of adsorption	BTU/lb-mole	-18000
HEAT(2)	heat of adsorption	BTU/lb-mole	-28000
HEAT(3)	heat of adsorption	BTU/lb-mole	-8988
HEAT(4)	heat of adsorption	BTU/lb-mole	-8988
K F(1)	Mass Transfer Coefficent	ft/hr	0.017
K F(2)	Mass Transfer Coefficent	ft/hr	0.004
K F(3)	Mass Transfer Coefficent	ft/hr	0.05
K F(4)	Mass Transfer Coefficent	ft/hr	0.1
ERROR1			0.00000001
ERROR2			1E-09

FIGURE 4-54 INPUT DATA SET "A" FOR 2DMOL COMPARISON WITH 04-19-94 TEST

Variable	*Description*	*Units*	*Current Value*
S_B	Cross sectional surface area	ft^3	0.019069
CP_WA	Wall specific heat		0.109848
RA	not used	feet	0.00472441
RAV	not used	feet	0.0058
RHOS	not used		65
ALPHA1	not used		635
R_P	Pellet radius	feet	0.0047244
D_P	Pellet diameter	feet	0.0094488
R	Gas Constant		10.73
PI			3.141593
UINS			0.354
HWALL			12
NC	Number of components		2
T_FO	same as TO		532
GC			416975040
CON_S1			1
NP			125
PT	same as PTOT		837.7789
LEN	same as Z		1.66667
VOID_B	same as EPSEX		0.35
IND(1)			0
IND(2)			1
IND(3)			1
IND(4)			0
IDES1			0
ISAT			1

FIGURE 4-55INPUT DATA SET "B" FOR 2DMOL COMPARISON WITH 04-19-94 TEST

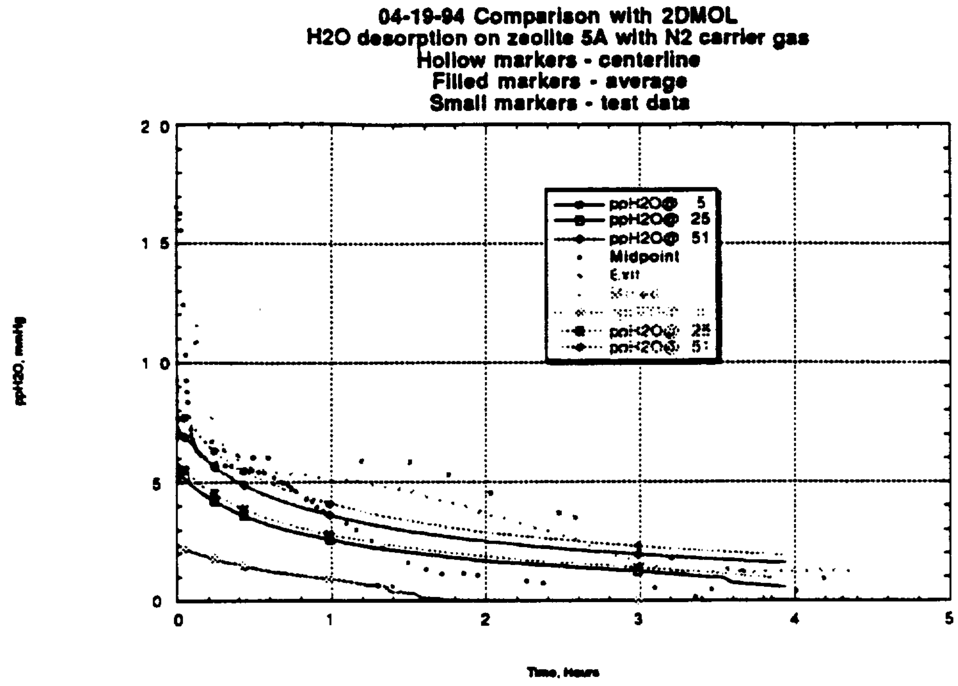


FIGURE 4-56 BREAKTHROUGH COMPARISONS FOR 04-19-94 TEST WITH 2DMOL RESULTS

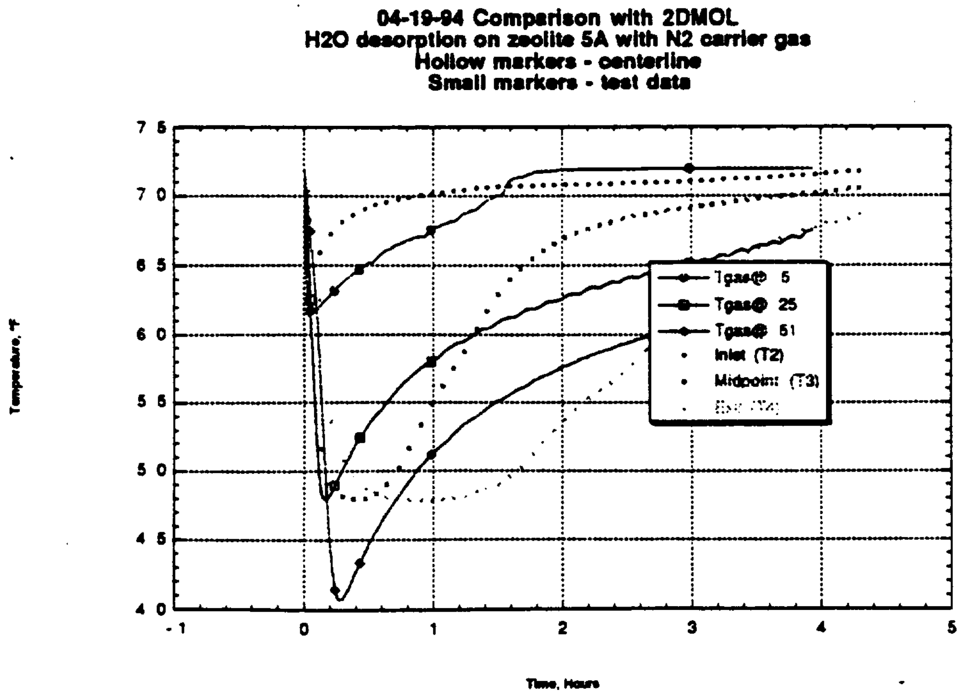


FIGURE 4-57 TEMPERATURE COMPARISONS FOR 04-19-94 TEST WITH 2DMOL RESULTS

4.8 FLOW3MOL MODEL RESULTS

The FLOW3MOL model represents the 4BMS desiccant beds, with the capability to model three different layers of sorbent material stacked in a single packed bed. The FLOW3MOL model was run for demonstration purposes configured for a packed column of the approximate size of the MSMBT apparatus. No reliable test data is currently available to compare these results with, so they are presented without comparison.

Figure 4-58 shows results from the FLOW3MOL model run for CO_2 breakthrough. The location of the breakthrough points are at the end of 13X, middle of Silica gel, end of Silica gel, middle of 13X, and end of 13X materials. As can be seen the CO_2 breakthrough occurs very fast for the first section of bed because of presence of water. No CO_2 adsorption takes place on Silica gel material. CO_2 is desorbed on the last section of the bed, 13X. Since for some period of time all of the H_2O would be adsorbed on Silica gel, flow passes to the last section of the bed without water. This gives the opportunity for CO_2 being adsorbed there. As silica gel is saturated with H_2O, water will pass to the 13X and this cause the roll-up of CO_2.

Figure 4-59 shows results from the FLOW3MOL model run for H_2O breakthrough. It can be seen the H_2O breakthrough at the exit of Silica Gel material is sharp at low partial pressure and disperse at higher partial pressure of H_2O. This is caused by the Silica gel equilibrium isotherm which dose not have high affinity for H_2O at low partial pressure in compare to 13X. However it has more capacity for adsorption at higher partial pressures of H_2O.

Figure 4-60 shows results from the FLOW3MOL model run for temperature. The figure shows the sharp increase and decrease of temperature at the beginning of the bed as it expected and a steady temperature for sometime on 13X section which is the same profile as in 13X-single bed.

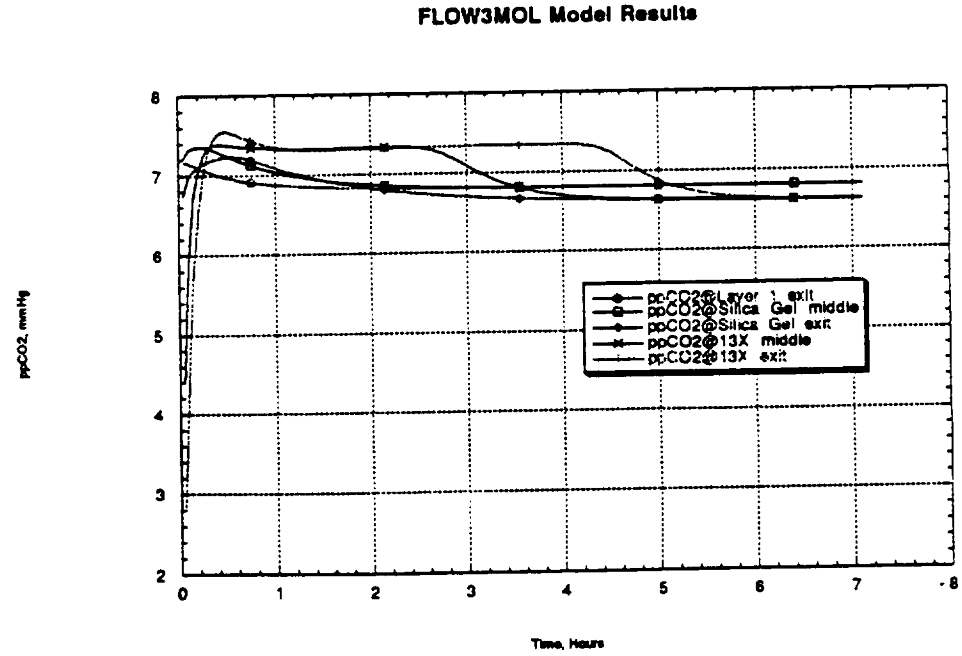


FIGURE 4-58 CARBON DIOXIDE BREAKTHROUGH RESULTS FOR FLOW3MOL

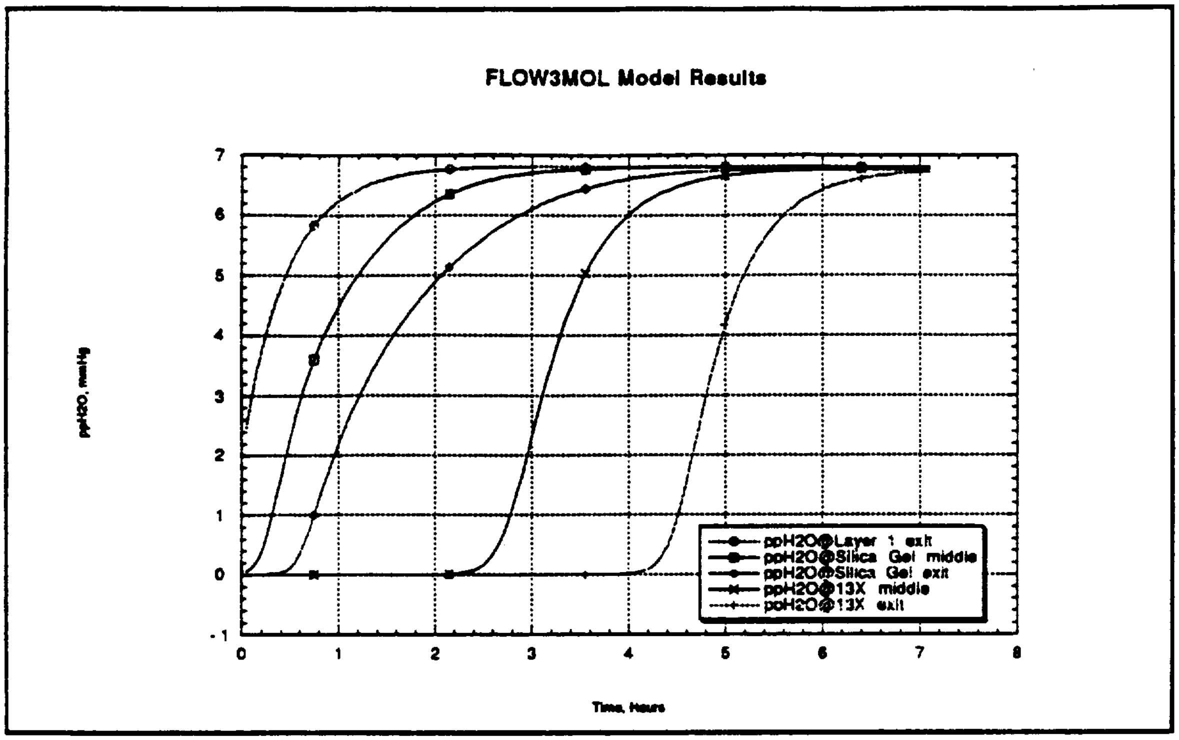


FIGURE 4-59 WATER BREAKTHROUGH RESULTS FOR FLOW3MOL

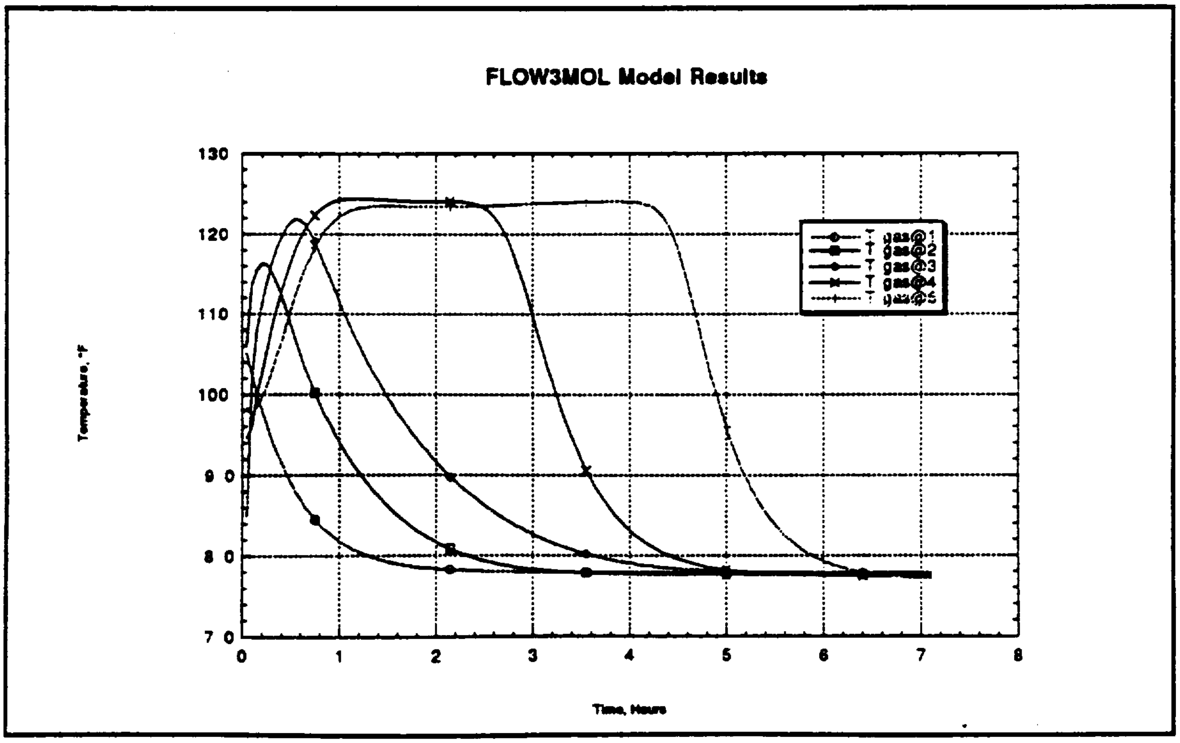


FIGURE 4-60 TEMPERATURE RESULTS FOR FLOW3MOL

4.9 VACMOL Model Results

The VACMOL model represents the 4BMS 5A sorbent beds, with the capability to model thermal/vacuum desorption for a packed bed. The VACMOL model was run for demonstration purposes configured for a packed column of the approximate size of the 5A sorbent beds in the Life Test 4BMS. No reliable test data is currently available to compare these results with, so they are presented without comparison.

Figure 4-61 shows temperature prediction results from the VACMOL model. The constant heat input into the vacuum chamber causes a steady rise of temperature as time progresses.

Figure 4-62 shows results from the VACMOL model run for H_2O and CO_2 partial pressures. Initially the bed is saturated with CO_2, H_2O, and N_2. at atmospheric pressure and temperature. In order to simulate the actual bed performance a pressure boundary condition of 25 mm Hg was chosen. The sudden decrease in pressure is caused by outflow of N_2. The partial pressure of CO_2 and H_2O start rising as heat being input to the vacuum chamber. Since CO_2 is less adsorbent than H_2O, it starts coming off the bed sooner and depleted off the bed in very short time.

Figure 4-63 shows results from the VACMOL model run for total pressure. A sudden decrease in total pressure to the boundary condition value is an indication that pressure is being transferred along the bed in very short time. After a minima, the pressure start rising because of desorption of CO_2 and H_2O from the bed as temperature increases.

The model is written in a such a way that any velocity, caused by differential pressure, will not exceed the choke velocity. As shown in Figures 4-64 and 4-65, the velocity profile starts from a very high value, because of large pressure gradient along the bed, to a minima after a very short time. The minima corresponds to a minima of pressure gradient in the bed. The velocity starts rising as CO_2 and H_2O are desorbed off the bed. This makes the pressure increase inside of the bed, creating a differential pressure with the boundary pressure. Given the profile of pressure and velocity along the bed, it is reasonable to say that variation of these two variables along the bed is not significant. For simulation of the actual 4BMS a lumped velocity and pressure could be used.

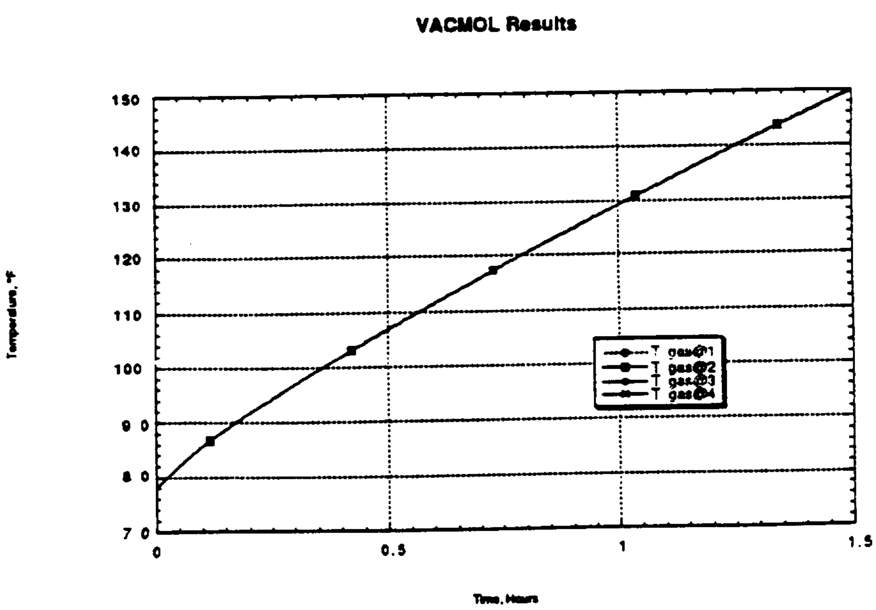


FIGURE 4-61 TEMPERATURE RESULTS FOR VACMOL

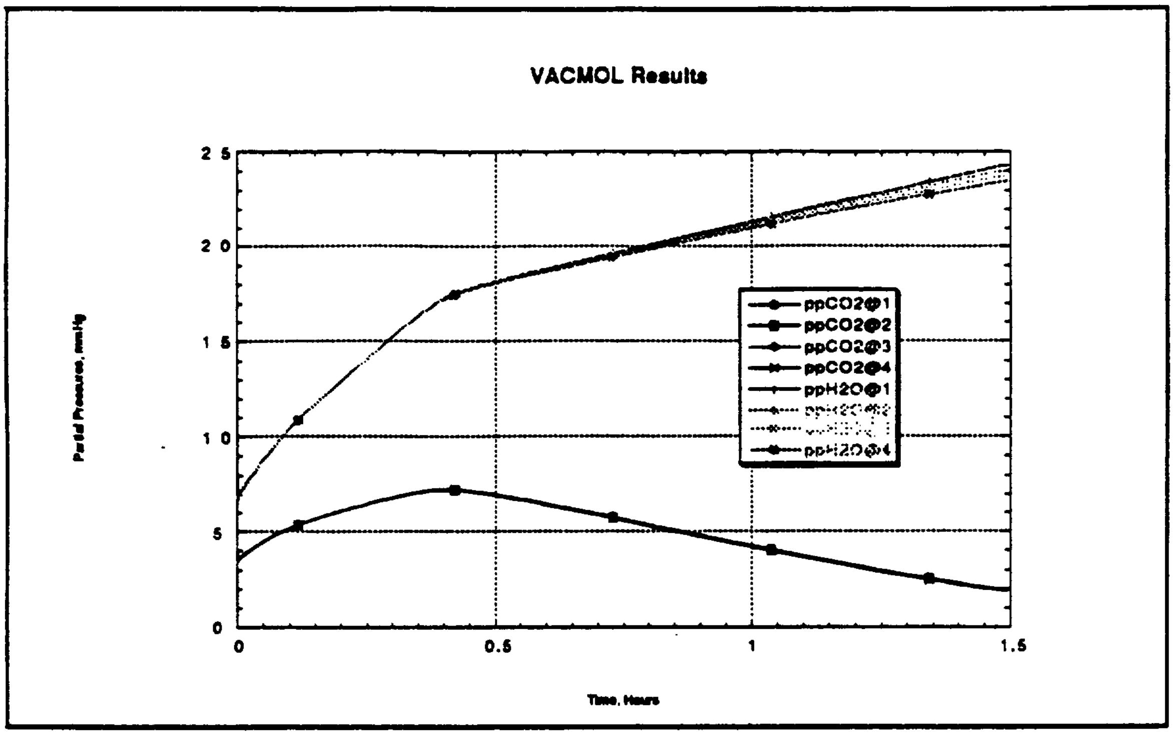


FIGURE 4-62 PARTIAL PRESSURE RESULTS FOR VACMOL

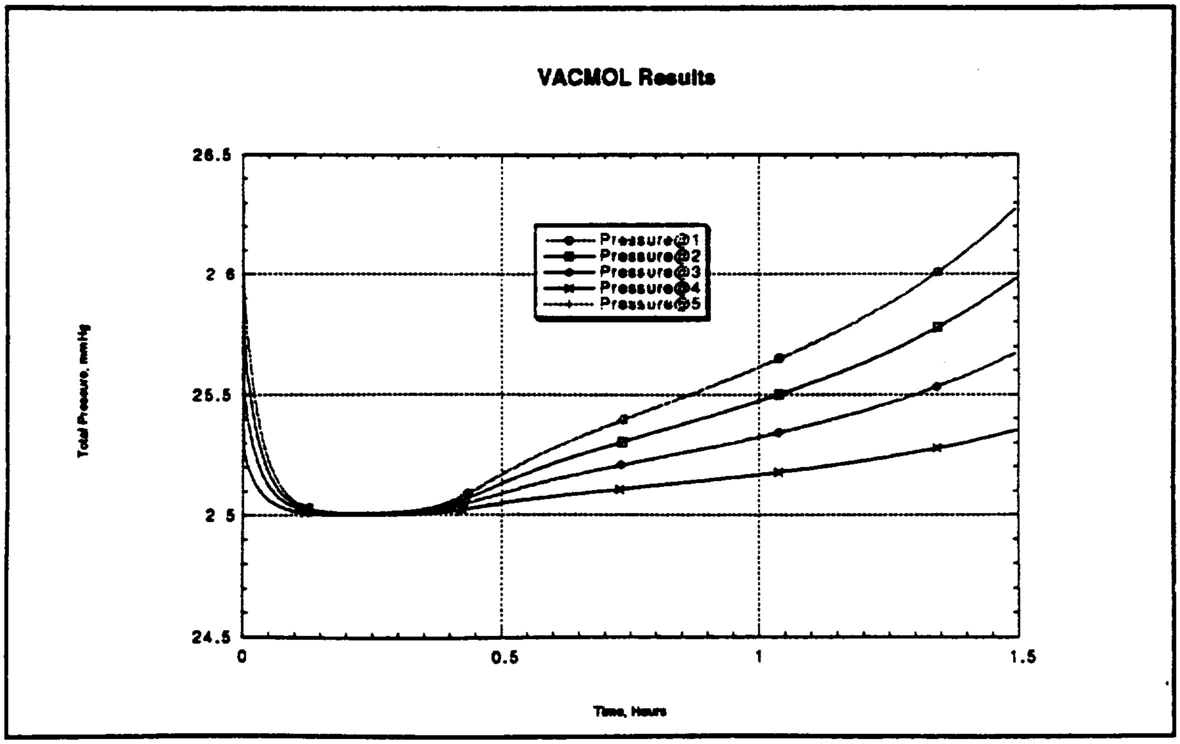


FIGURE 4-63 TOTAL PRESSURE RESULTS FOR VACMOL

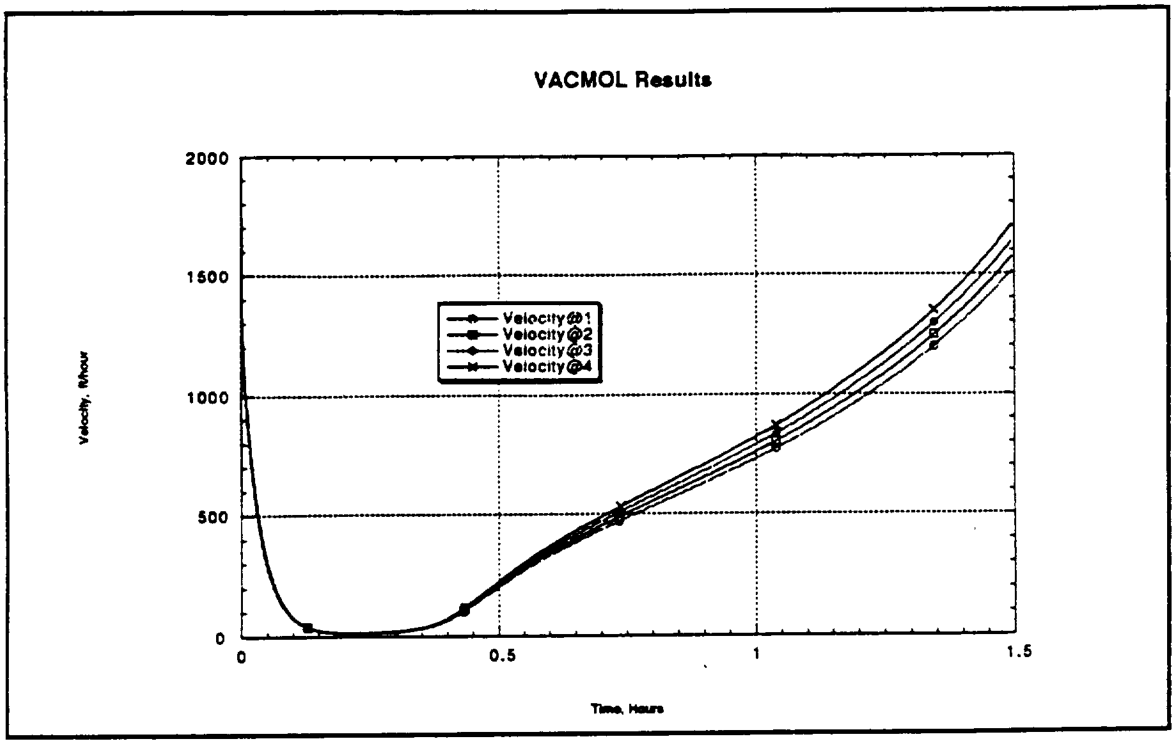


FIGURE 4-64 VELOCITY RESULTS FOR VACMOL

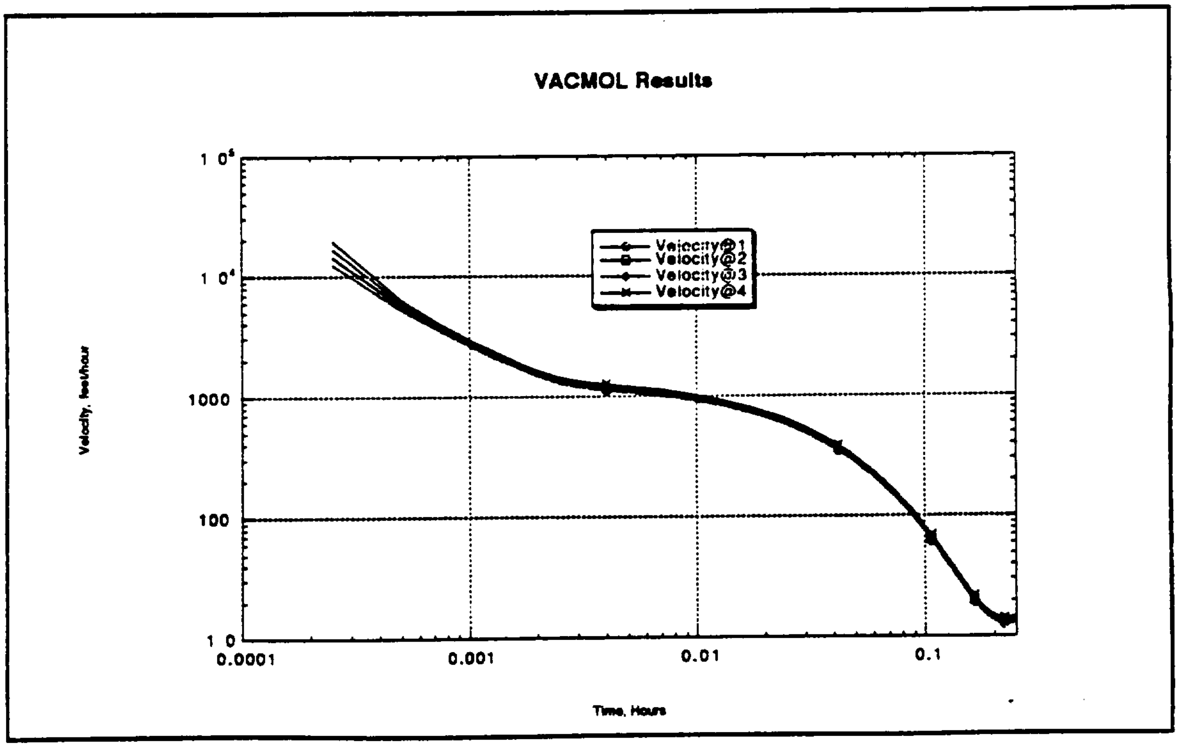


FIGURE 4-65 VELOCITY RESULTS FOR VACMOL (FIRST 0.25 HOURS)

SECTION 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

Based on analytical and experimental investigation of convective flows in porous media the following conclusions are drawn:

1. The experimental results from the laboratory scale-fixed bed adsorber are quantitatively consistent with the one-dimensional model at the column center. The average concentration of cross sectional bed obtained by test result deviates from the column center concentration appreciably. This indicates the strong effects of porosity variation along the radial direction of column bed on the temperature, concentration, and velocity field. These observations resulted in the decision to model the dynamic behavior of the column in two-dimensions.

2. The two-dimensional model demonstrates the importance of non-darcian momentum and porosity variation on the flow distribution, concentration, and temperature in the radial direction of the bed. The two-dimensional model not only predicts the concentration and temperature profile at the column center well, but it also predicts the average concentration reasonably well.

3. A linear driving force mass transfer model provides a reasonable fit to the experimental adsorption and desorption.

4. The concentration of a key component, CO_2, is affected by the presence of the non-key component, N_2, in CO_2/N_2 adsorption. CO_2 effluent concentration overshoots its inlet concentration because of H_2O displacement (more easily adsorbed) component. The height of this roll-up is increased with the inlet concentration of H_2O component.

5. Comparison of model results with test data shows the created models can be accurately used for modeling of the actual 4BMS.

6. The two-dimensional model can be used to simulate the actual 5A bed of 4BMS, which contains an array of approximately 1/2 inch square channels, for porosity variation.

7. The IAST equilibrium predicts the breakthrough curve much closer to the test data than the Lanmuir-Fredlitch theory for mixtures of CO_2/N_2 and CO_2/H_2O.

5.2 Recommendations

1. Since the PDE's of the two-dimensional model are a strong function of porosity, research into the velocity variation is obviously important.

2. Because of transient behavior of adsorption processes, the CPU usage of the two-dimensional model is very large. Attempts should be made to convert the two-dimensional model to a one dimensional model by choosing the appropriate functions to describe the concentration, temperature, and velocity field in the radial direction of the column.

3. The solution to IAST functions relies on a trial and error type. This costs a large amount of CPU time. Further study should be done to eliminate this barrier.

4. Thermal vacuum description is the regenerative mode of the actual 4BMS 5A sorbent beds. Experimental data is needed to verify the model.

5. Literature shows that a pressure swing process which only uses two beds is more efficient than any other adsorption/desorption process. Research should be done to convert the 4BMS to a 2BMS.

6. More recent isotherm data, obtained from the same sorbent that is expected to be used for the actual flight 4BMS system, has recently become available. This sorbent data should be incorporated into the models developed in this study.

7. The models developed during this study have been shown to model the processes present in the 4BMS to a very reasonable degree of accuracy. Therefore, they should be incorporated into an integrated 4BMS for the purpose of analyzing and optimizing the 4BMS system.

SECTION 6

REFERENCES

Argo, W. B., and Smith, J. M., Chem. Eng. Prog., 49, 443, (1953).

Baron, T., *Chem. Eng. Progr.*, 48, 118 (1952).

Beavers, G.S. and Sparrow, E.M., "Non-Darcy Flow through Fibrous Porous Media," *J. Applied Mech.*, 36, 711-714, (1969).

Benenati, R.F. and C. B. Brosilow, "Void Fraction Distribution in Beds of Spheres," AICHE J., 8, 359-361, (1962).

Bird, R. B., W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, Jon Wiley & Sons, Inc., New York (1960).

Darcy, H., *Les Fontains Publiques de la ville de Dijon*, Damont, Paris. (1956).

"Davison Molecular Sieves Adsorption Equilibria", W. R. Grace & Co., Davison Chemical Division.

Ergun, S., "Fluid Flow Through Packed Column," *Chem. Eng. Prog.*, 48, 89-94 (1952).

Fahien, R.W., Ph.D Thesis, Purdue University, (1954).

Fahien, R.W., and J.M. Smith, *AICHE J.*, 1,28, (1955).

Glueckauf. E., "Theory of Chromatography-Part 10," Trans. Faraday. Soc., 51, 1540, (1955)

Gerald, C. F and P. O. Wheatley, Applied Numerical Analysis, Addison-Wesley Publishing Company, U S A (1984).

Hiroyuki. Y. and D. Ruthven, " Dynamic Behavior of an Adiabatic Adsorption Column-I, " Chemical Eng Sci, 36, 6, 877-884, (1982).

Hunt. M.L., " Non-Darcian Convection in Packed-Sphere and Fibrous Media," Ph.D. thesis, Univ. of Calif., Berkeley, 1987.

Kaviani, M., *Principle of Heat Transfer in Porous Media*, Pringer-Verlag, New York (1991).

Knox, J.C., Finite Difference Modeling and Experimental Investigation of Carbon Dioxide Adsorption on a Molecular Sieve Sorbent Material Used in Spacecraft Carbon Dioxide Removal Systems, MSE Thesis, University of Alabama in Huntsville, 1992.

Kunii, D. and J. M. Smith., "Heat Transfer Characteristics of Porous Rocks," *AICHE J*, 6, 97, (1960).

Kunii, D. and M. Smith., 3rd Int. Dev. on Heat Transfer, IV, 344, (1966).

Kunii, D., Suzuki, M., and One, N., "Heat Transfer from Wall Surface to Packed Beds at High Reynolds Number," J. Chem.Eng.Japan, 1, 21-26, (1968).

Langmuir, I., " The Adsorption of Gases on Plane Surface of Glass, Mica, Platinum," *J. Amer. Chem. Soc., 40,* 1361 (1918).

Moinuddin, A., "Design of Steady State Packed Bed Tubular Reactor" Ph.D. Thesis, University of Florida, (1976).

Myers, A. L. , and J. M., Prausnitz, "Thermodynamics of Mixed-Gas Adsorption," *AICHE J.*, 11,121 (1965).

Mohamadinejad. H, "Development and Verification Testing of a Hardware-Independent Molecular Sieve Model," 2ed International Conference on Life Support and Biosphere Sciences, Feb 22-23, 123-134, (1994).

Newman, J., "Numerical Solution of Coupled, Ordinary Differential Equations," Ind and Eng. Chem. Fund, 7, 514-517 (1968).

Newman, J., Numerical Solution Of Coupled, Ordinary Differential Equations (UCRL-17739), Lawrence Radiation Laboratory, University of California, Berkeley, August (1967).

Nield, D.A., and A. Bejan., *Convection in Porous Media*, Springer-Verlag, New York (1992).

Petrovic, L. J., and G. Thodos, "Mass Transfer in the Flow of Gases through Packed Bed," *Ind. Eng. Chem. Fundam.*, 7, 2, 274 (1968).

Rice, R., " Approximate Solution for Bacth, Packed Tube and Radial Flow Adsorbers-Compersion With Experiment," Chemical Eng Sci, 37, 1, 83-91, (1982).

Roblee, L.H.S., R. M. Baird, and J. W. Tierney., " Radial porosity Variation in Packed Beds," AICHE J., 4, 460-464, (1958).

Ruthven, D. M., Principle of Adsorption and Adsorption Processes, John Wiley & Sons, Inc., New York, (1984).

Satterfield, C. N., *Hetrogenous Catalysis in Practice*, McGraw-Hill Book Company, New York (1980).

Sips,R., *J. Chem.Phys.*, 490 (1948).

Slattery, J. C., R. B. Bird, *AICHE J.*, 4,137-142 (1958).

Smith, J. M., Chemical Engineering Kinetics, 3rd ed., McGrow-Hill Book Company, New York (1981).

Suzuki. M., Adsorption Engineering, Kodansha Ltd, Japan (1990).

Tsai, M. C., S. S Wang, and R. T. Yang, "Pore Diffusion Model for Cyclic Separation," AICHE J., 29, 966 (1983).

Vafai, K. and C. L. Tien., "Boundary and Inertia Effect on Convective Mass Transfer in Porous Media," *Int. J. Heat Mass Transfer*, 24, 195-203, (1981).

Wright R.M., Ruder, J.M., Dunn, V.B., and Hwang, K.C., "Development of Design Information for Molecular-Sieve Type Regenerative CO2 Removal Systems", NASA Contractor Report CR-2277, AiResearch Manufacturing Company, Los Angeles, 1973

Yagi, S. and D. Kunii., *AICHE J*, 3, 373 (1957)

Yagi, S., D. Kunii., and N. Wakao., *AICHE J*, 6, 543 (1960).

Yang, R. T., *Gas Separation by Adsorption Process*, Butterwotths, Boston (1987).

APPENDIX A

DERIVATION OF IAST EQUATION (EQ 2.8)

Knowing that the spreading pressure is related to temperature and the amount adsorbed on the solid by

$$\frac{\pi(P_i^\circ)A}{RT} = \int_0^{P} \frac{q_i^\circ}{P_i^\circ}\,dP_i^\circ \qquad \text{eq 2.4}$$

and assuming the adsorbed phase obeying Langmuir-Freundlich isotherm

$$q_i^\circ = \frac{q_{mi}B_iP_i^{n_i}}{1+B_iP_i^{n_i}} \qquad \text{eq 2.2}$$

the spreading pressure can be obtained as

$$\frac{\pi(p_i^\circ)A}{RT} = \int_0^{p_i^\circ} \frac{q_i}{p_i}\,dP_i = \int_0^{p_i^\circ} \frac{q_iB_iP_i^{n_i}}{p_i(1+B_iP_i^{n_i})}\,dP_i \qquad \text{eq A.1}$$

therefore π is

$$\pi = \frac{RT}{A}\frac{q_{mi}}{n_i}\ln\left(1+B_iP_i^{\circ n_i}\right) \qquad \text{eq A.2}$$

solving for P_i

$$P_i^\circ = \left(\frac{\exp\left(\frac{\pi n_i}{q_{mi}}\frac{A}{RT}\right)-1}{B_i}\right)^{\frac{1}{n_i}} \qquad \text{eq A.3}$$

knowing the fact that

$$P_i = Py_i = p_i^\circ(\pi)x_i \qquad \text{eq 2.6}$$

substitution of eq 2.23 into 2.6, and solving for x_i

$$x_i = \frac{Py_i}{\left(\frac{\exp\left(\frac{\pi n_i}{q_{mi}}\frac{A}{RT}\right)-1}{B_i}\right)^{\frac{1}{n_i}}} \qquad \text{eq A.4}$$

knowing that

$$\sum x_i = \sum y_i = 1 \qquad \text{eq 2.7}$$

equation 2.8 will result

$$\sum_{i=1}^{i=n} \frac{Py_i}{\left(\frac{\exp\left(\frac{\pi A n_i}{RTq_{mi}} \right) - 1}{B_i} \right)^{\frac{1}{a_i}}} - 1 = 0 \qquad \text{eq 2.8}$$

APPENDIX B

DERIVATION OF MOMENTUM, ENERGY, AND MOLAR CONCENTRATION BALANCES FOR TWO DIMENSIONS

In order to derive the mass, energy, and momentum balances for adsorption packed column in cylindrical coordinates, the conservation laws were set over a thin "shell" of fluid. In all of conservation laws the balances are:

$$\begin{pmatrix}\textbf{time rate of}\\ \textbf{change of i}\end{pmatrix} = (\textbf{rate of i in}) - (\textbf{rate of i out}) + \begin{pmatrix}\textbf{rate production or}\\ \textbf{dissipation of i}\end{pmatrix} \qquad \text{eq B.1}$$

where i could be momentum, thermal heat, or molar mass of component i. After a balance is made on a shell of finite thickness, the limit is taken as the spatial dimensions approach zero. As a result of this, a differential equation is generated which describes the time rate of change of a particular variable.

In this study, the diffusion of momentum, heat and molar mass was taken into account not only in the axial direction but also in the radial direction.

For developing the conservative equations the following assumptions were made:

1. The gas phase is dilute and follows ideal gas law
2. Axial and radial conduction within column wall is neglected
3. The adsorbent pellet can be modeled as sphere
4. The temperature within the pellet is uniform
5. The radial convection is negligible

B.1 Conservation Of Molar Mass Of Component i In The Mixture

In developing the molar mass balance of component i in the mixtures which is being adsorbed by pellet particles within the packed bed, the fluxes of component i take place by diffusion and convection in axial direction and only by diffusion in radial direction, since the radial convection is negligible. The rate of disappearance of component i in the mixture is by diffusion of i at the surface and into the pore and the solid part of the particles. Writing the molar mass conservation of component i

$$\begin{aligned}\varepsilon 2\pi r\Delta r\Delta x\frac{\partial C_i}{\partial t} &= \varepsilon 2\pi r\Delta r N_i\big|_x - \varepsilon 2\pi r\Delta r N_i\big|_{x+\Delta x} + \varepsilon 2\pi\Delta x r N_i\big|_r - \varepsilon 2\pi\Delta x r N_i\big|_{r+\Delta r} - \\ &\quad (1-\varepsilon)2\pi r\Delta r\Delta x r_i\end{aligned} \qquad \text{eq B.2}$$

Collecting the terms and dividing by $\varepsilon 2\pi r\Delta r\Delta x$

$$\frac{\partial C_i}{\partial t} = -\frac{\left(N_i\big|_{x+\Delta x} - N_i\big|_x\right)}{\Delta x} - \frac{\left(rN_i\big|_{r+\Delta r} - rN_i\big|_r\right)}{r\Delta r} - \frac{(1-\varepsilon)}{\varepsilon}r_i \qquad \text{eq B.3}$$

Take the limit as Δx and Δr go to zero

$$\frac{\partial C_i}{\partial t} = -\frac{\partial N_{i,x}}{\Delta x} - \frac{\partial (rN_{i,r})}{r\Delta r} - \frac{(1-\varepsilon)}{\varepsilon}r_i \qquad \text{eq B.4}$$

Knowing that

$$N_{i,x} = -D_{eff,i,x}\frac{\partial C_i}{\partial x} + u_x C_i$$

$$N_{i,r} = -D_{eff,i,r}\frac{\partial C_i}{\partial r} + u_r C_i \qquad \text{eq B.5}$$

where the second term, the convection term in the radial direction, is negligible. Substitution of fluxes into the above equation

$$\frac{\partial C_i}{\partial t} = D_{eff,i,x}\frac{\partial^2 C_i}{\partial x^2} - \frac{\partial (uC_i)}{\partial x} + D_{eff,i,r}\frac{\partial}{r\partial r}\left(\frac{r\partial C_i}{\partial r}\right) - \frac{(1-\varepsilon)}{\varepsilon}r_i \qquad \text{eq B.6}$$

The derivation for one dimensional molar mass balance is the same as above except that the third term on the right hand side, flux due to radial diffusion, is not considered.

B.2 Conservation Of Energy Of Fluid Flow

Energy transfer in fluid phase is by convection in the axial direction and by diffusion in axial and radial directions. Energy is also being produced by heat of adsorption of components in the particle. Energy is being transferred to the wall of column by an overall heat transfer coefficient. Writing the energy balance for the gas phase

$$\begin{aligned} \varepsilon 2\pi r\Delta r\Delta x\rho_g c_{pg}\frac{\partial T_g}{\partial t} = {} & \varepsilon 2\pi r\Delta r(q-\rho_g c_{pg}u_x T_g)\Big|_x - \varepsilon 2\pi r\Delta r(q-\rho_g c_{pg}u_x T_g)\Big|_{x+\Delta x} + \\ & \varepsilon 2\pi r\Delta x(q-\rho_g c_{pg}u_r T_g)\Big|_r - \varepsilon 2\pi r\Delta x(q-\rho_g c_{pg}u_r T_g)\Big|_{r+\Delta r} - \\ & (1-\varepsilon)2\pi r\Delta r\Delta x h_s a_s(T_g - T_s) \end{aligned} \qquad \text{eq B.7}$$

Collecting the terms and dividing by $\varepsilon 2\pi r\Delta r\Delta x$

$$\begin{aligned} \rho_g c_{pg}\frac{\partial T_g}{\partial t} = {} & -\frac{(q+\rho_g c_{pg}u_x T_g)\Big|_{x+\Delta x} - (q+\rho_g c_{pg}u_x T_g)\Big|_x}{\Delta x} - \\ & \frac{r(q+\rho_g c_{pg}u_r T_g)\Big|_{r+\Delta r} - r(q+\rho_g c_{pg}u_r T_g)\Big|_r}{r\Delta r} - \\ & \frac{(1-\varepsilon)}{\varepsilon}h_s a_s(T_g - T_s) \end{aligned} \qquad \text{eq B.8}$$

Take the limit as Δx and Δr go to zero and ignoring the radial convection

$$\rho_g c_{pg}\frac{\partial T_g}{\partial t} = -\frac{\partial q_x}{\partial x} - \rho_g c_{pg}\frac{\partial (u_x T_g)}{\partial x} - \frac{\partial (rq_r)}{r\partial r} - \frac{(1-\varepsilon)}{\varepsilon}h_s a_s(T_g - T_s) \qquad \text{eq B.9}$$

Knowing the fluxes are

$$q_x = -k_{f,x}\frac{\partial T_g}{\partial x}$$

$$q_r = -k_{f,r}\frac{\partial T_g}{\partial r} \qquad \text{eq B.10}$$

Substitution of fluxes into the above equation

$$\rho_g c_{pg}\frac{\partial T_g}{\partial t} = k_{f,x}\frac{\partial^2 T_g}{\partial x^2} - \rho_g c_{pg}\frac{\partial (uT_g)}{\partial x} + k_{f,r}\frac{\partial}{r\partial}(r\frac{\partial T_g}{\partial r}) - \frac{(1-\varepsilon)}{\varepsilon}h_s a_s (T_g - T_s) \qquad \text{eq B.11}$$

The derivation for one dimensional heat balance is the same as above except that the third term on the right hand side, flux due to radial diffusion, is not being considered. And, a dissipation rate of energy to the column wall must be added to the above equation.

B.3 Conservation Of Energy Of Solid Phase

Energy transfers in solid phase by diffusion in axial and radial directions. Energy is also being produced by heat of adsorption of components in the particle. Energy is being transferred to the fluid phase by an overall heat transfer coefficient. Writing the energy balance for the solid phase

$$2\pi r\Delta r\Delta x\rho_s c_{ps}\frac{\partial T_s}{\partial t} = 2\pi r\Delta r(q)|_x - 2\pi r\Delta r(q)|_{x+\Delta x} + 2\pi r\Delta x(q)|_r - 2\pi r\Delta x(q)|_{r+\Delta r} -$$

$$2\pi r\Delta r\Delta x h_s a_s (T_s - T_g) - 2\pi r\Delta r\Delta x\sum_{i=1}^{n}\Delta H_i r_i \qquad \text{eq B.13}$$

Collecting the terms and dividing by $\varepsilon 2\pi r\Delta r\Delta x$

$$\rho_s c_{ps}\frac{\partial T_s}{\partial t} = -\frac{(q)|_{x+\Delta x} - (q+)|_x}{\Delta x} - \frac{r(q)|_{r+\Delta r} - r(q)|_r}{r\Delta r} - h_s a_s (T_s - T_g) - \sum_{i=1}^{n}\Delta H_i r_i \qquad \text{eq 4.14}$$

knowing the fluxes

$$q_x = -k_{s,x}\frac{\partial T_s}{\partial x}$$

$$q_r = -k_{s,r}\frac{\partial T_s}{\partial r} \qquad \text{eq B.14}$$

inserting fluxes in equation

$$\rho_s c_{ps}\frac{\partial T_s}{\partial t} = k_{s,x}\frac{\partial^2 T_s}{\partial x^2} + k_{s,r}\frac{\partial}{r\partial}(r\frac{\partial T_s}{\partial r}) - h_s a_s (T_s - T_g) - \sum_{i=1}^{n}\Delta H_i r_i \qquad \text{eq B.15}$$

The derivation for one dimensional heat balance for the solid phase is the same as above except that the second term on the right hand side, flux due to radial diffusion, is not being considered

B.4 Conservation Of Energy Based On An Effective Conductivity

When the temperature difference between fluid and particle phases is neglected, column temperature can be derived by summation of eqs B-11 and 4.15.

$$(\varepsilon\rho_g c_{pg} + (1-\varepsilon)\rho_s c_{ps})\frac{\partial T_g}{\partial t} = \varepsilon k_{eff,x}\frac{\partial^2 T}{\partial x^2} - \varepsilon\rho_g c_{pg}\frac{\partial(uT_g)}{\partial x} + \varepsilon k_{eff,r}\frac{\partial}{r\partial}(r\frac{\partial T}{\partial r}) - (1-\varepsilon)\sum_{i=1}^{n}\Delta H_i r_i \qquad \text{eq B.16}$$

where $k_{x,eff}$ and $k_{r,eff}$ are the effective thermal conductivity in the axial and radial direction in the packed column, which are

$$k_{eff,x} = k_{s,x} + k_{f,x}$$

$$k_{eff,r} = k_{s,r} + k_{f,r} \qquad \text{eq B.17}$$

The derivation for one dimensional-homogeneous heat balance is the same as above except that the third term on the right hand side, flux due to radial diffusion, is not considered. And, a dissipation rate of energy to the column wall must be added to the above equation.

B.5 Total Mass Balance

Assuming ideal gas law $C_i = {}^{P_i}\!/_{RT}$ and knowing $\sum P_i = p$, the component mass balance equation can be recast into an overall mass balance equation, substitution of C_i into eq 2.10 and carrying out the derivatives yields to

$$-\frac{Dl}{RTg}$$

$$-\frac{D_i}{RT_g}\frac{\partial^2 P_i}{\partial x^2} - \frac{D_i P_i}{R}\left(2T_g^{-3}\left(\frac{\partial T_g}{\partial x}\right)^2 - T_g^{-2}\frac{\partial^2 T_g}{\partial x^2}\right) + \frac{1}{RT_g}\frac{\partial P_i}{\partial t} - \frac{P_i}{RT_g^2}\frac{\partial T_g}{\partial t}$$

$$+\frac{u}{RT_g}\frac{\partial P_i}{\partial x} - \frac{uP_i}{RT_g^2}\frac{\partial T_g}{\partial x} + \frac{P_i}{RT_g}\frac{\partial u}{\partial x} + \frac{1-\varepsilon}{\varepsilon}\frac{\partial \bar{q}_i}{\partial t} = 0 \qquad \text{eq B.18}$$

neglecting the second term and multiplying by RT_g, the above equation reduces to

$$\frac{\partial P_i}{\partial t} = D_i\frac{\partial^2 P_i}{\partial x^2} - u\frac{\partial P_i}{\partial x} - P_i\frac{\partial u}{\partial x} + \frac{P_i}{T_g}\left(\frac{\partial T_g}{\partial t} - D_i\frac{\partial^2 T_g}{\partial x^2} + u\frac{\partial T_g}{\partial x}\right) - RT_g\frac{1-\varepsilon}{\varepsilon}\frac{\partial \bar{q}_i}{\partial t} \qquad \text{eq B.19}$$

summation of the above equation over all the components and knowing that $\sum P_i = p$, the equation for total pressure is

$$\frac{\partial P}{\partial t} = D_i\frac{\partial^2 P}{\partial x^2} - u\frac{\partial P}{\partial x} - P\frac{\partial u}{\partial x} + \frac{P}{T_g}\left(\frac{\partial T}{\partial t} - D_i\frac{\partial^2 T}{\partial x^2} + u\frac{\partial T}{\partial x}\right) - RT\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n}\frac{\partial \bar{q}_i}{\partial t} \qquad \text{eq B.20}$$

APPENDIX C

DESCRIPTION OF COMPUTER PROGRAMS

Four FORTRAN computer programs are presented in this report. The first is a two-dimensional model of flow adsorption/desorption in a packed bed. The second is a one-dimensional model of flow adsorption/desorption in a packed bed. The third is a model of the thermal vacuum desorption. And the last is a tri-sectional packed bed with two different sorbent materials. The programs are capable of simulating up to four gas constituents for each process. If it is necessary to simulate more constituents, minor changes are needed in a few subroutines.

C.1 Two Dimensional FORTRAN Program

A flow chart of the two-dimensional modeling of flow adsorption/desorption is shown in Figures C-1 and C-2. This program simulates the two-dimensional adsorption and desorption of a fixed-bed column 10 inches long and 1.87 inches in diameter. The model simulates four constituents with variations in radial velocity and porosity. The users supply initial and boundary condition in the INITIAL subroutine. If it is necessary to run the model with a larger diameter or length, the number of grids must be specified in the INITIAL subroutine.

For each time step the temperature dependent parameters (density, conductivity, viscosity, etc.) are recalculated. If the temperature change is not significant in the packed bed during the simulation, the user can specify that these parameters will not be recalculated.

Solid-gas equilibrium for the pure components are predicted by the Langmuir-Freundlich isotherm. For mixtures, two types of solid-gas equilibrium are implemented in the program. One is based on the Langmuir-Freundlich isotherm for mixtures. The second prediction uses the IAST isotherm. The IAST model has an iterative solution and is CPU time intensive. The user can switch an indicator to chose the desired isotherm method.

The energy balance includes the heat lost to the surroundings. For an adiabatic simulation of a packed bed, the heat transfer coefficient between the wall and the surroundings equal is set to zero.

Heat transfer is assumed between the solid, fluid flow, and the wall of the column. The program is also capable of simulating the heat transfer bed based on an effective conductivity. In this case a thermal equilibrium between solid and fluid flow is assumed. Axial and radial conductivity is included in the model. These parameters vary along the radial and axial direction based on the porosity variation. The program is also capable of making these parameters constant by switching the proper indicator.

Mass transfer is simulated with a lumped resistance model. This resistance is obtained by matching experimental breakthrough curves with the results of the model.

This model uses the UNDER-RELAXATION method for calculation of all variables in the PDE's, except the velocity and the pressure which are computed by the NEWMAN method. The total numbers of equations involved for four components is 13.

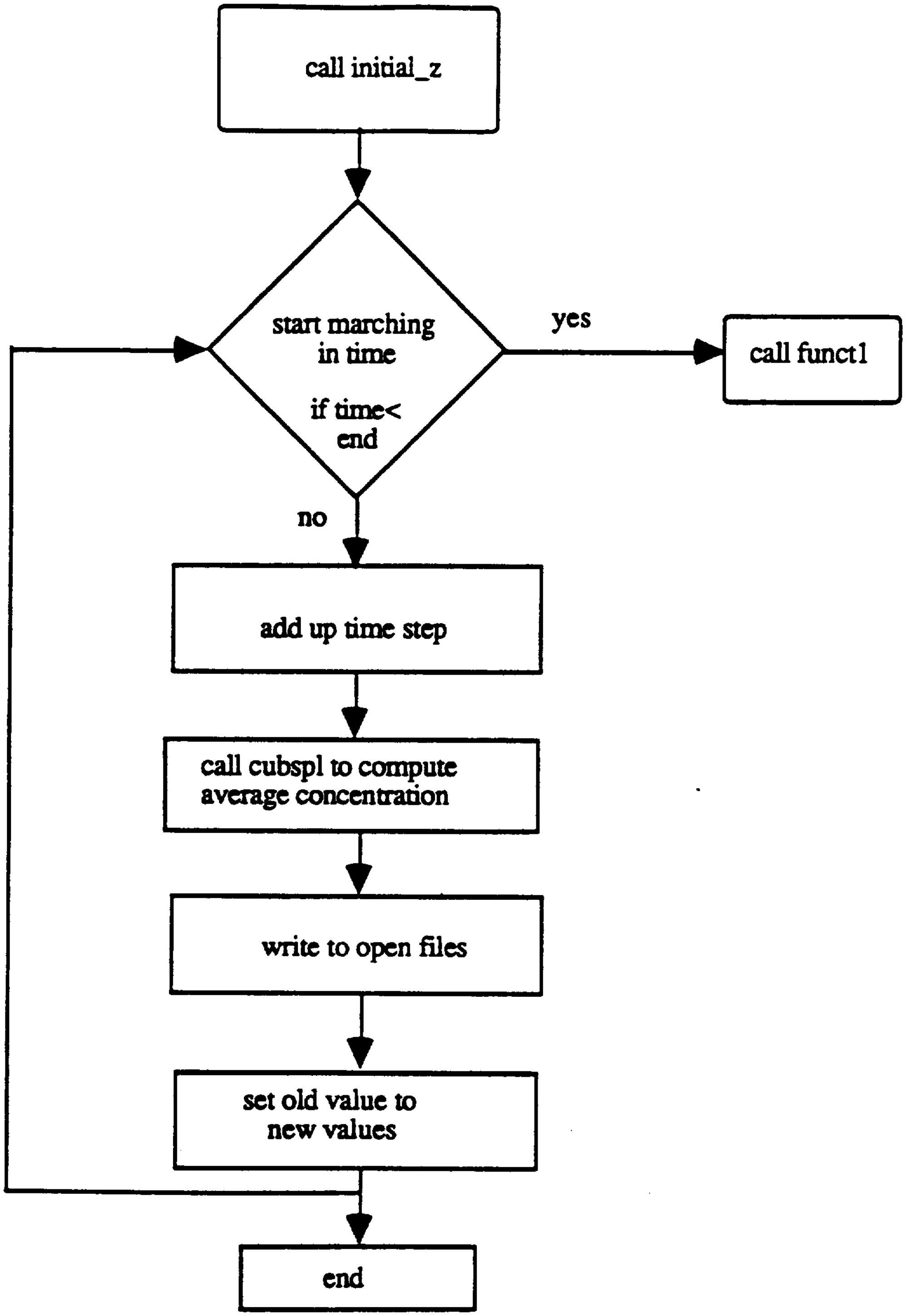


FIGURE C-1 THE FLOW CHART OF THE MAIN PROGRAM FOR THE 2-D MODEL

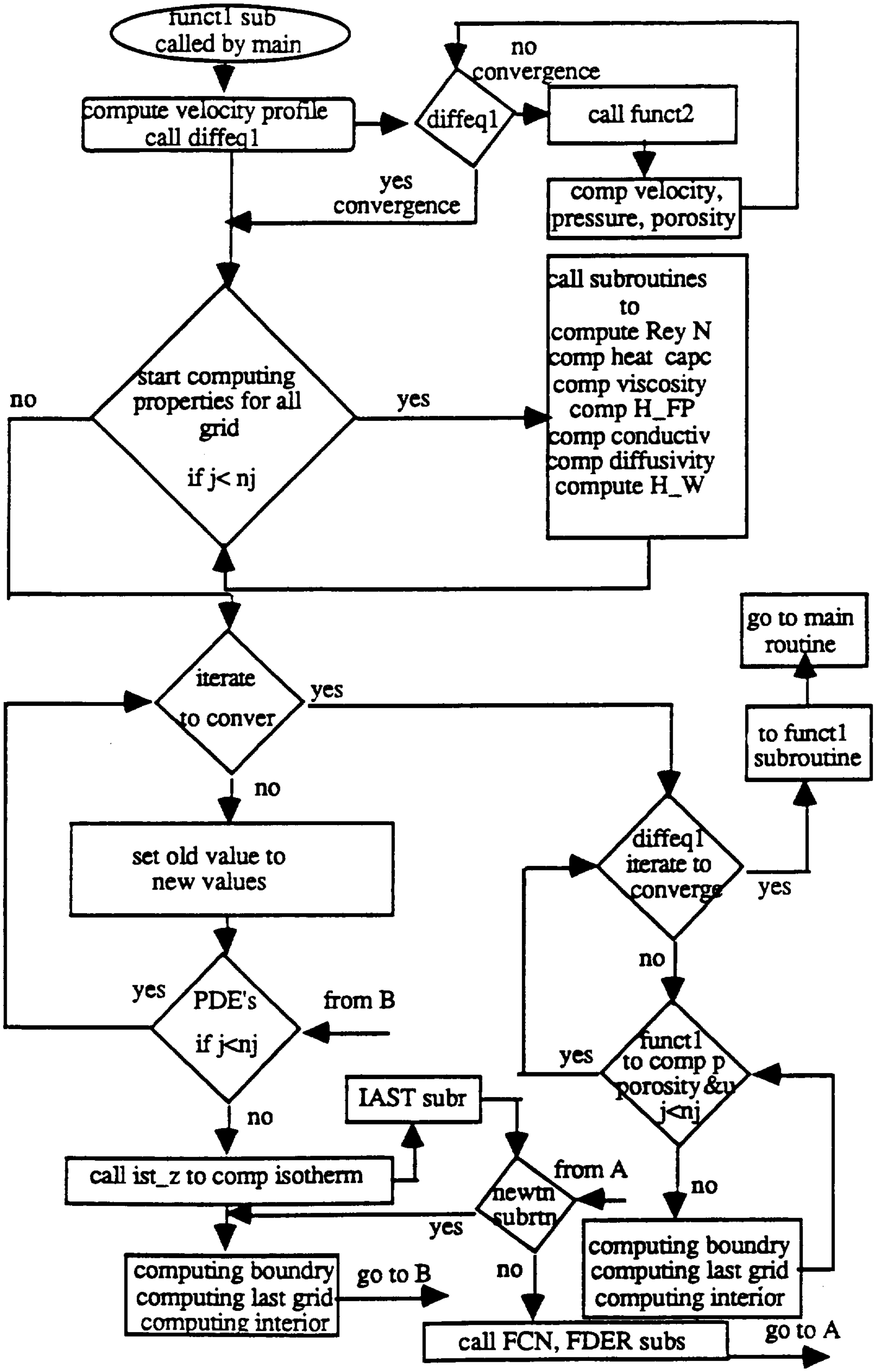


FIGURE C-2 THE FLOW CHART FOR THE SUBROUTINE PROGRAMS FOR THE 2-D MODEL

A list of program variables is given below with a brief description.

Variable	Description	Units	Type
LL	Time step index	none	Int
T_Time	current time	minutes	real
S(i,j,k)	matrix variable; i=variable ID; j=axial grid number; k=radial grid number	various	real*13
P(i,j,k).	Old of S array	various	real*13
L1	gas temperature	R	real
L2	solid temperature	R	real
L3	wall temperature	R	real
L4	velocity	ft/hr	real
L5	pressure	psia	real
I6	pressure	psia	real
PT	pressure	mmHG	real
NCOMP	number of components	dimensionless	int
INERT	type of carrier gas 1=n2; 2=He	dimensionless	real
EPSEX	porosity	sq ft/cu ft	real
TAMB	outside temperature	R	real
RHOS	particle density	lbs/cu ft	real
Z	bed length	ft	real
VOID_B	porosity	sq ft/cu ft	real
D_I	inside bed diameter	ft	real
D_E	outside bed diameter	ft	real
S_B	cross sectional area of bed	ft	real
CP_WA	heat capacity of wall	BTU/F/LB	real
RO_WA	density of wall	lb/cu ft	real
CP_S	heat capacity of particle	Btu/lb-F	real
RO_S	density of particle	lb/cu ft	real
R_P	particle radius	ft	real
D_P	particle diameter	ft	real
heat(i)	heat of adsorption for each component	BTU/lb mole	real
NC	number of components	dimension less	int
G_F	volumetric flow rate	cu ft/min	real
T_FO	inlet temperature	R	real
P_C	inlet component partial pressure	psia	real
TO	initial temperature	R	real
R	ideal gas law constant	lbf/sq ft .lb mol/cu ft/R	real
PI	constant	3.14	real
M_W	molecular weight	lb/mole	real
GC	gravitational constant	in hr	real
K_F	mass transfer coefficient	ft/hr	real
NJ	grids number in axial direction	dimensionless	int
NJR	grids number in radial direction	dimensionless	int
DELZ	grid size in axial direction	ft	real
DELT	time step	hr	real
CON_s1	pellet conductivity	BTU/hr/ft/F	real
ERROR	convergence criteria	various	real

A list of program variables is given below with a brief description (continued).

Variable	Description	Units	Type
N	number of equation	dimensionless	int
DELR1	grid size in radial direction	ft	real
DELR2	grid size in radial direction	ft	real
Y_F	component mole fraction	mole/mole	real
P_TOT	Total Pressure	psia	real
BC_L1 to BC_L6	boundary condition for L1 to L6	various	real
W0	under relaxation coeff	dimensionless	real
I_SOL_FLO	indicator, no equilibrium between the gas and solid temperature	indicator	int
DP	particle diameter	ft	real
H_OW	heat transfer coefficient outside of the wall	Btu/sq ft-min-r	real
U(i,j)	velocity/Darcy velocity, center line velocity	dimensionless	real
UMAX	max velocity	ft/hr	real
NUMAX	the grid where max velocity is	ft/hr	int
EPS(i)	porosity	sq ft/cu ft	real
REY	Reynolds number	dimensionless	real
CP_P	specific heat	BTU/F/cu ft	real
VISC_P	viscosity	lb/ft/hr	real
H_FP	heat transfer coefficient fluid-particle	Btu/hr/sq ft/F	real
CON_Z	fluid axial conductivity	Btu/hr/ft/F	real
CON_R	fluid radial conductivity	Btu/hr/ft/F	real
CON_S	particle conductivity	Btu/hr/ft/F	real
EFF_CON_Z1	fluid axial effective conductivity	Btu/hr/ft/F	real
EFF_CON_R1	fluid radial effective conductivity	Btu/hr/ft/F	real
DIF	molecular diffusivity	sq ft/hr	real
SC_N	Schmidt number	dimensionless	real
Eff_DIFF_R	fluid radial effective diffusivity	sq ft/hr	real
EFF_DIFF_Z	fluid axial effective diffusivity	sq ft/hr	real
H_W	heat transfer coefficient fluid-wall	Btu/hr/sq ft/F	real
V(i)	Langmuir constant	lb mol/cu ft	real
B(i)	Langmuir constant	dimensionless	real
PO(i)	Langmuir constant	dimensionless	real
PP	component partial Pressure	mm Hg	real
Q1	adsorbed concentration	lb mol/solid lb	real

C.2 One-Dimensional Flow Adsorption/Desorption FORTRAN Program for Column Packed with One Section Material

A flow chart of the one-dimensional modeling of flow adsorption/desorption is shown in Figure C-3 and C-4. This program simulates the one-dimensional adsorption and desorption of a fixed-bed column 10 inches long and 1.87 inches in diameter. Users supply initial and boundary condition in the INITIAL subroutine. If it is necessary to run the model with larger diameter or length, the number of grids must be specified in the INITIAL subroutine.

For each time step the temperature dependent parameters (density, conductivity, viscosity, etc.) are recalculated. If the temperature change is not significant in the packed bed during the simulation, the user can specify that these parameters will not be recalculated.

Solid-gas equilibrium for the pure components are predicted by the Langmuir-Freundlich isotherm. For mixtures, two types of solid-gas equilibrium are implemented in the program. One is based on the Langmuir-Freundlich isotherm for mixtures. The second prediction uses the IAST isotherm. The IAST model has an iterative solution and is CPU time intensive. The user can switch an indicator to chose the desired isotherm method.

The energy balance includes the heat lost to the surroundings. For an adiabatic simulation of a packed bed, the heat transfer coefficient between the wall and the surroundings equal is set to zero.

Heat transfer is assumed between the solid, fluid flow, and the wall of the column.

Mass transfer is simulated with a lumped resistance model. This resistance is obtained by matching experimental breakthrough curves with the results of the model.

This model uses the UNDER-RELAXATION method for calculation of all variables in the PDE's, except the velocity and the pressure which are computed by the NEWMAN method. The total numbers of equations involved for four components is 13.

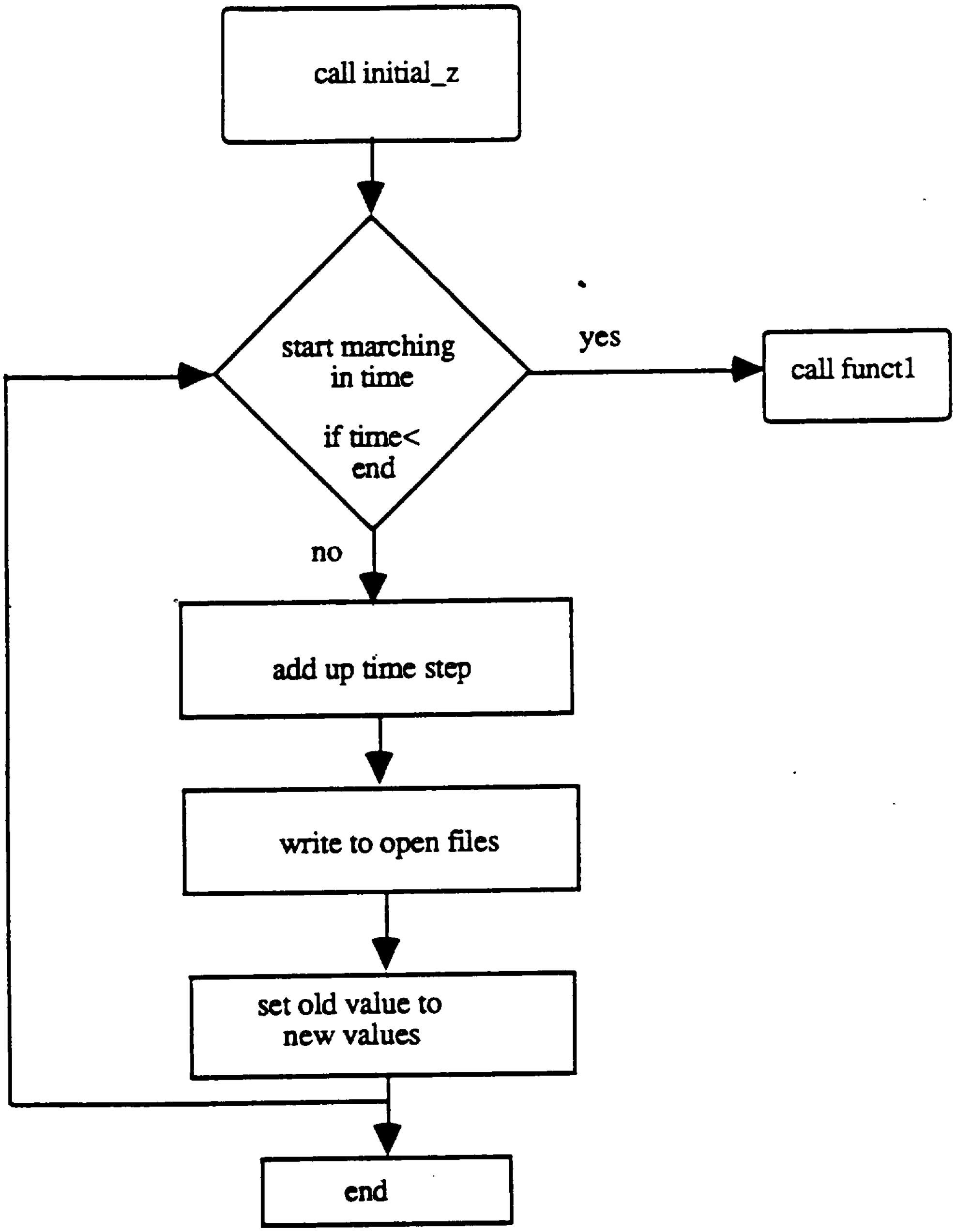


FIGURE C-3 THE FLOW CHART OF THE MAIN PROGRAM FOR THE 1-D MODEL

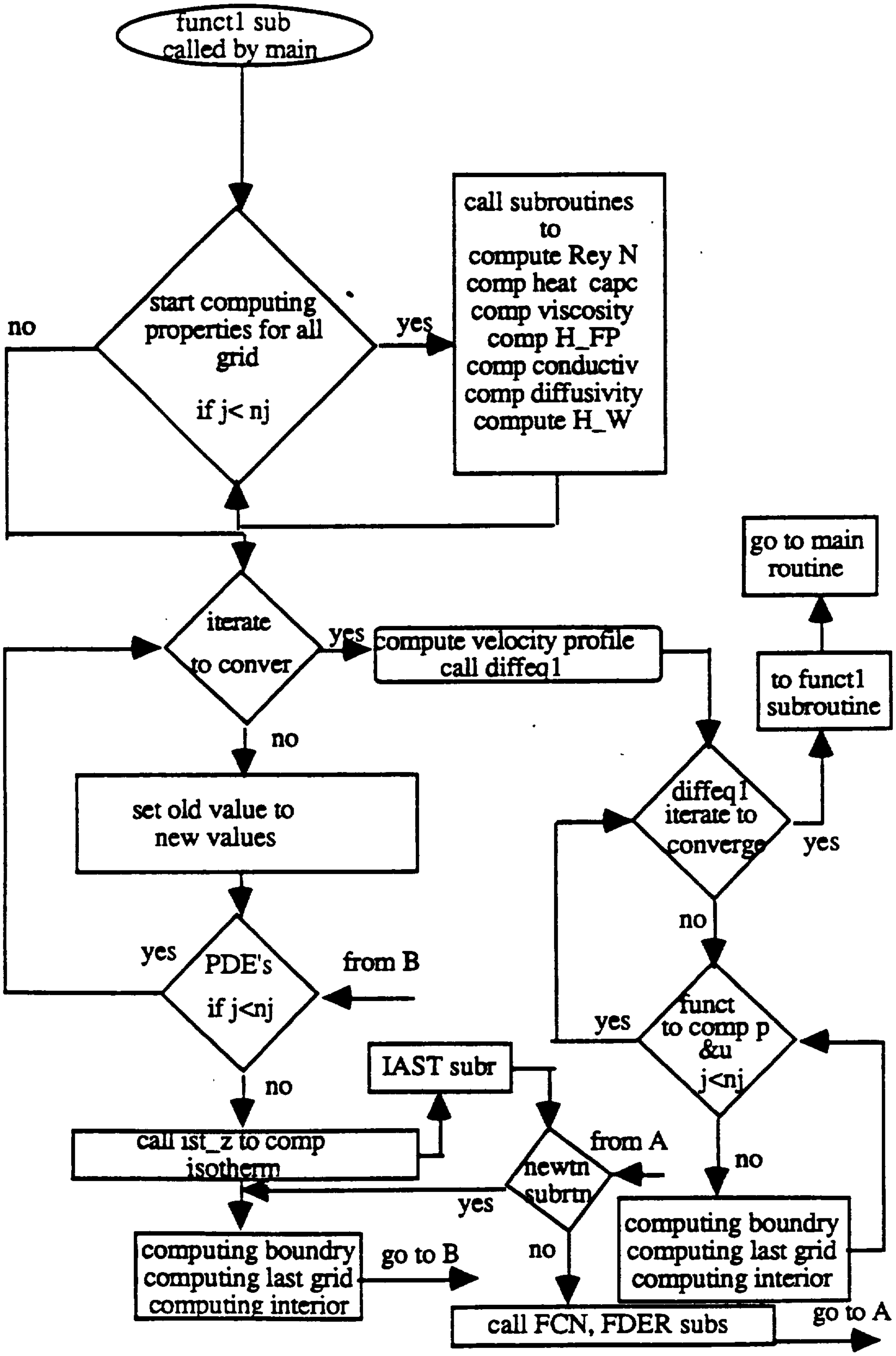


FIGURE C-4 THE FLOW CHART OF THE SUBROUTINES PROGRAM FOR THE 1-D MODEL

A list of program variables is given below with a brief description.

Variable	Description	Units	Type
LL	Time step index	none	Int
T_Time	current time	minutes	real
C(i,j)	matrix variable; i=variable ID; j=axial grid number; k=radial grid number	various	real*13
P(i,j).	Old of C array	various	real*13
L1	gas temperature	R	real
L2	solid temperature	R	real
L3	wall temperature	R	real
L4	velocity	ft/hr	real
L5	pressure	psia	real
I6	pressure	psia	real
PT	pressure	mm HG	real
NCOMP	number of component	non dimensional	int
INERT	type of carrier gas	non dimensional	real
EPSEX	porosity	cu ft/cu ft	real
RHOS	particle density	lbs/cu ft	real
Z	bed length	ft	real
VOID_B	porosity	cu ft/cu ft	real
D_I	inside bed diameter	ft	real
D_E	outside bed diameter	ft	real
S_B	cross sectional area of bed	ft	real
CP_WA	heat capacity of wall	BTU/F/LB	real
RO_WA	density of wall	lb/cu ft	real
CP_S	heat capacity of particle	real	
RO_S	density of particle	lb/cu ft	real
R_P	particle radius	ft	real
D_P	particle diameter	ft	real
heat(i)	heat of adsorption for each component	BTU/lb mole	real
NC	number of component	non dimension	int
G_F	volumetric flow rate	cu ft/min	real
TO	initial temperature	R	real
R	ideal gas law constant	lbf/sq ft .lb mol/cu ft/R	real
PI	constant	3.14	real
M_W	Molecular weight	real	
GC	constant		real
K_F	mass transfer coefficient	ft/hr	real
NJ	grids number in axial direction	non dimension	int
DELZ	grid size in axial direction	ft	real
DELT	time step	hr	real
CON_S1	pellet conductivity	BTU/hr/ft/F	real
ERROR	convergence criteria	various	real
N	number of equation	non dimension	int
Y_F	component mole fraction	lb mole i/lb mole	real

A list of program variables is given below with a brief description (continued).

Variable	Description	Units	Type
P_TOT	Total Pressure	psia	real
BC_L1 to BC_L6	boundary condition for L1 to L6	various	real
W0	under relaxation coeff	non dimension	real
H_OW	heat transfer coefficient outside of the wall	Btu/hr/F/sq ft	real
REY	Reynolds number	non dimension	real
CP_P	specific heat	BTU/F/cu ft	real
VISC_P	viscosity	lb/ft/hr	real
H_FP	heat transfer coefficient fluid-particle	Btu/hr/sq ft/F	real
CON_L	fluid axial conductivity	Btu/hr/ft/F	real
CON_S	particle conductivity	Btu/hr/ft/F	real
H_W	heat transfer coefficient fluid-wall	Btu/hr/sq ft/F	real
V(i)	Langmuir constant	lb mole/lb solid	real
B(i)	Langmuir constant	non dimension	real
PO(i)	Langmuir constant	non dimension	real
PP	component partial Pressure	mm Hg	real
Q1	adsorbed concentration	lb mol/ lb solid	real
BC_C1(i)	boundary condition of molar concentration	lb mole/cu ft	real
W0-W3	under relaxation for each component and temperature	non dimension	real
F3-F8	PDE's coefficient	various	real
PE_N_M	mass Peclet number	non-dimension	real
PE_N_H	heat Peclet number	nonedimension	real
AINT	interfacial area	sq ft/cu ft	real
U_F1	velocity	ft/hr	real

C.3 One-Dimensional Flow Adsorption/Desorption FORTRAN Program for Column Packed with Three Sections of Different Material

A flow chart of the one-dimensional modeling of flow adsorption/desorption for column packed with three sections of different materials is shown in Figure C-5 and C-6. This program simulates the one-dimensional adsorption and desorption of a fixed-bed column 10 inches long and 1.87 inches in diameter. Users supply initial and boundary condition in the INITIAL subroutine. If it is necessary to run the model with larger diameter or length, the number of grids must be specified in the INITIAL subroutine.

The first section of the column, about 3 inches, contains the 13X material, followed by about 10 inches of silica gel, and finally the rest of the bed is filled with 13X material.

For each time step the temperature dependent parameters (density, conductivity, viscosity, etc.) are recalculated. If the temperature change is not significant in the packed bed during the simulation, the user can specify that these parameters will not be recalculated.

Solid-gas equilibrium for the pure component is predicted by Langmuir-Freundlich isotherm for the 13X material. Silica gel is depicted from the literature as an exponential function of temperature and pressure. For mixtures, two types of solid-gas equilibrium are implemented in the program. One is based on the Langmuir-Freundlich isotherm for mixtures. The second prediction uses the IAST isotherm. The IAST model has an iterative solution and is CPU time intensive. The user can switch an indicator to choose the desired isotherm method.

The energy balance includes the heat lost to the surroundings. For an adiabatic simulation of a packed bed, the heat transfer coefficient between the wall and the surroundings equal is set to zero.

Heat transfer is assumed between the solid, fluid flow, and the wall of the column. .

Mass transfer is presented as lumped resistance model for the 13X material. For silica gel, the flux of the adsorbate is considered to be a combination of solid and pore diffusion. This resistance is obtained by matching experimental breakthrough curves with the results of model.

The model uses UNDER-RELAXATION method for calculation of all variables in the PDE's, except the velocity and the pressure which are computed by NEWMAN method. The total numbers of equations involve for four components is 13.

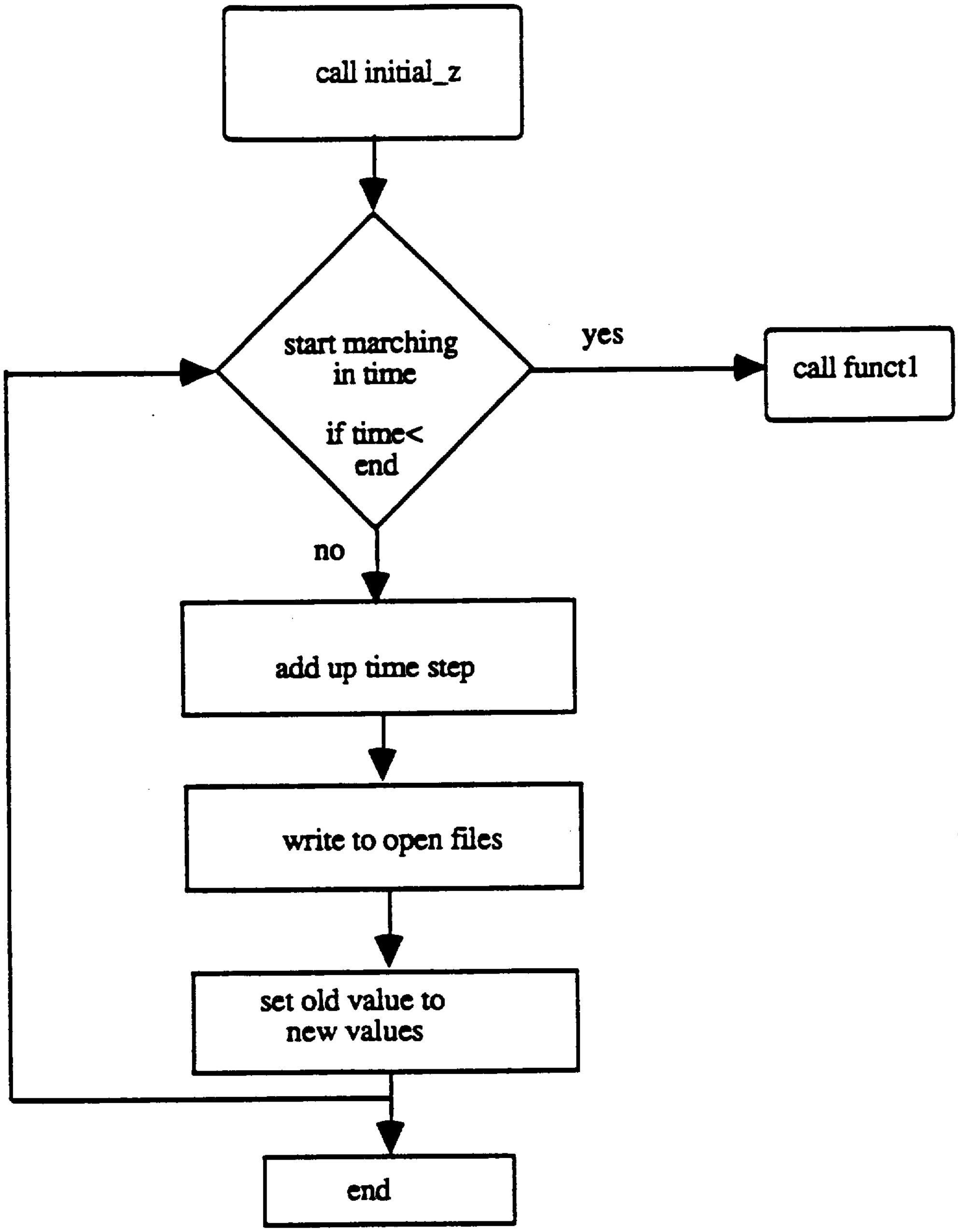


FIGURE C-5 THE FLOW CHART OF THE MAIN PROGRAM FOR THE 1-D MODEL FOR BED PACKED WITH THREE DIFFERENT MATERIALS

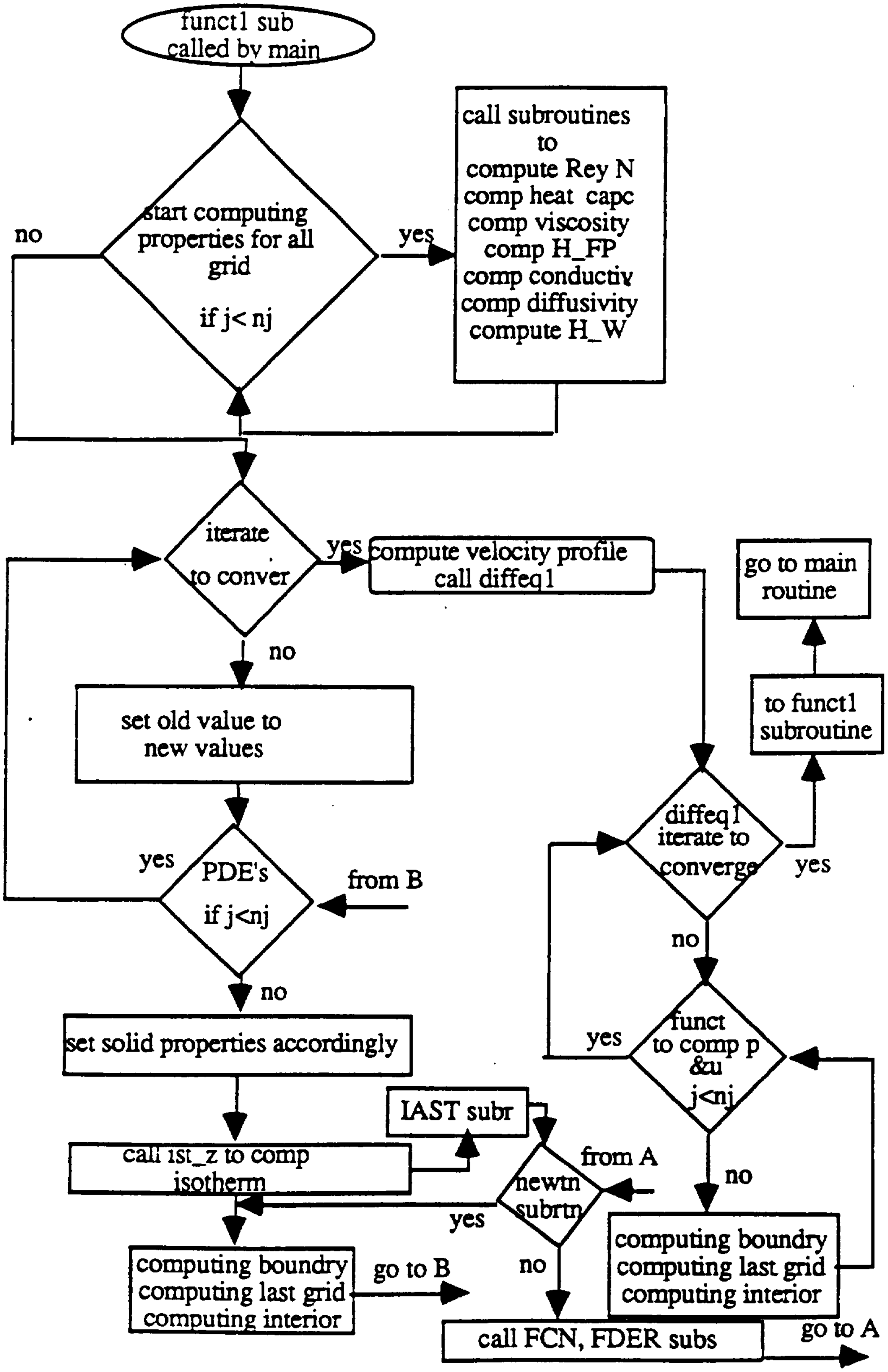


FIGURE C-6 THE FLOW CHART OF THE SUBROUTINES PROGRAM FOR THE 2-D MODEL FOR BED PACKED WITH THREE DIFFERENT MATERIALS

A list of program variables is given below with a brief description.

Variable	Description	Units	Type
LL	Time step index	none	Int
T_Time	current time	minutes	real
C(i,j)	matrix variable; i=variable ID; j=axial grid number; k=radial grid number	various	real*13
P(i,j).	Old of S array	various	real*13
L1	gas temperature	R	real
L2	solid temperature	R	real
L3	wall temperature	R	real
L4	velocity	ft/hr	real
L5	pressure	psia	real
I6	pressure	psia	real
PT	pressure	mm HG	real
NCOMP	number of component	nonedimension	int
INERT	type of carrier gas	nonedimension	real
EPSEX1,2	porosity, material type 1,2	cu ft/cu ft	real
RHOS1,2	particle density, material type 1, 2	lbs/cu ft	real
Z	bed length	ft	real
VOID_B	porosity	cu ft/cu ft	real
D_I	inside bed diameter	ft	real
D_E	outside bed diameter	ft	real
S_B	cross sectional area of bed	ft	real
CP_WA	heat capacity of wall	BTU/F/LB	real
RO_WA	density of wall	lb/cu · ft	real
CP_S1,2	heat capacity of particle type 1,2	real	
RO_S1,2	density of particle	lb/cu ft	real
R_P1,2	particle radius of type 1, 2	ft	real
D_P1,2	particle diameter type 1 1,2	ft	real
heat1,2(i)	heat of adsorption for each component of type 1,2	BTU/lb mole	real
NC	number of component	nonedimension	int
G_F	volumetric flow rate	cu ft/min	real
TO	initial temperature	R	real
R	ideal gas law constant	lbf/sq ft .lb mol/cu ft/R	real
PI	constant	3.14	
M_W	Molecular weight	lb /mole	real
GC	constant	in hr, ...	real
K_F	mass transfer coefficient	ft/hr	real
NJ	grids number in axial direction	nonedimension	int
DELZ	grid size in axial direction	ft	real
DELT	time step	hr	real
CON_S1	pellet conductivity	BTU/hr/ft/F	real
ERROR	convergence criteria	various	real
N	number of equation	non-dimension	int
Y_F	component mole fraction	lb mole/lb mol	real

A list of program variables is given below with a brief description (continued).

Variable	Description	Units	Type
P_TOT	Total Pressure	psia	real
BC_L1 to BC_L6	boundary condition for L1 to L6	various	real
W0	under relaxation coeff	nonedimension	real
H_OW	heat transfer coefficient outside of the wall	Btu/hr/F/sq ft	real
REY	Reynolds number	nonedimension	real
CP_P	specific heat	BTU/F/cu ft	real
VISC_P	viscosity	lb/ft/hr	real
H_FP	heat transfer coefficient fluid-particle	Btu/hr/sq ft/F	real
CON_L	fluid axial conductivity	Btu/hr/ft/F	real
CON_S1,2	particle conductivity of type 1 and 2	Btu/hr/ft/F	real
H_W	heat transfer coefficient fluid-wall	Btu/hr/sq ft/F	real
V(i)	Langmuir constant	lb mole/lb solid	real
B(i)	Langmuir constant	nonedimension	real
PO(i)	Langmuir constant	nonedimension	real
PP	component partial Pressure	mm Hg	real
Q1	adsorbed concentration	lb mol/solid lb	real
BC_C1(i)	boundary condition of molar concentration	lb mole/cu ft	real
W0-W3	under relaxation for each component and temperature	nonedimension	real
F(i)	PDE's variable		real
AINT	interfacial area	sq ft/cu ft	real
U_F1	velocity	ft/hr	real
LAYER1,2	number of grids for first section, for the last section	nonedimension	int
EPS	particle porosity	cu ft/cu ft	real

C.4 One Dimensional Thermal Vacuum Desorption FORTRAN Program

A flow chart of the one-dimensional modeling of thermal vacuum desorption is shown in Figures C-7 and C-8. This program simulates a one-dimensional desorption of a fixed-bed column 10 inches long and 1.87 inches in diameter . The users supply initial and boundary conditions in INITIAL subroutine. If it is necessary to run the model with a larger diameter or length, the number of grids must be specified in the INITIAL subroutine.

For each time step the temperature dependent parameters (density, conductivity, viscosity, etc.) are recalculated. If the temperature change is not significant in the packed bed during the simulation, the user can specify that these parameters will not be recalculated.

Solid-gas equilibrium for the pure component is predicted by Langmuir-Freundlich isotherm for the 13X material. For mixtures, two types of solid-gas equilibrium are implemented in the program. One is based on the Langmuir-Freundlich isotherm for mixtures. The second prediction uses the IAST isotherm. The IAST model has an iterative solution and is CPU time intensive. The user can switch an indicator to choose the desired isotherm method.

The energy balance includes the heat lost to the surroundings. Adiabatic simulation of a packed bed can be done by setting the heat transfer coefficient between the wall and the surroundings equal zero.

Heat transfer is assumed between the solid, fluid flow, and the wall of the column. Axial conductivity is included in the model. These parameters vary along axial direction inside the bed. The program is also capable of making these parameters constant by switching the proper indicator.

Mass transfer is presented as lumped resistance model. This resistance is obtained by matching the experimental breakthrough curve with the result of model.

In this model the material balances are based on the partial pressure. The pressure and velocity gradient inside the column are steep. Because of this, the model uses NEWMAN method for calculation of all variables in the PDE's. The total numbers of equations involve for four components is 13.

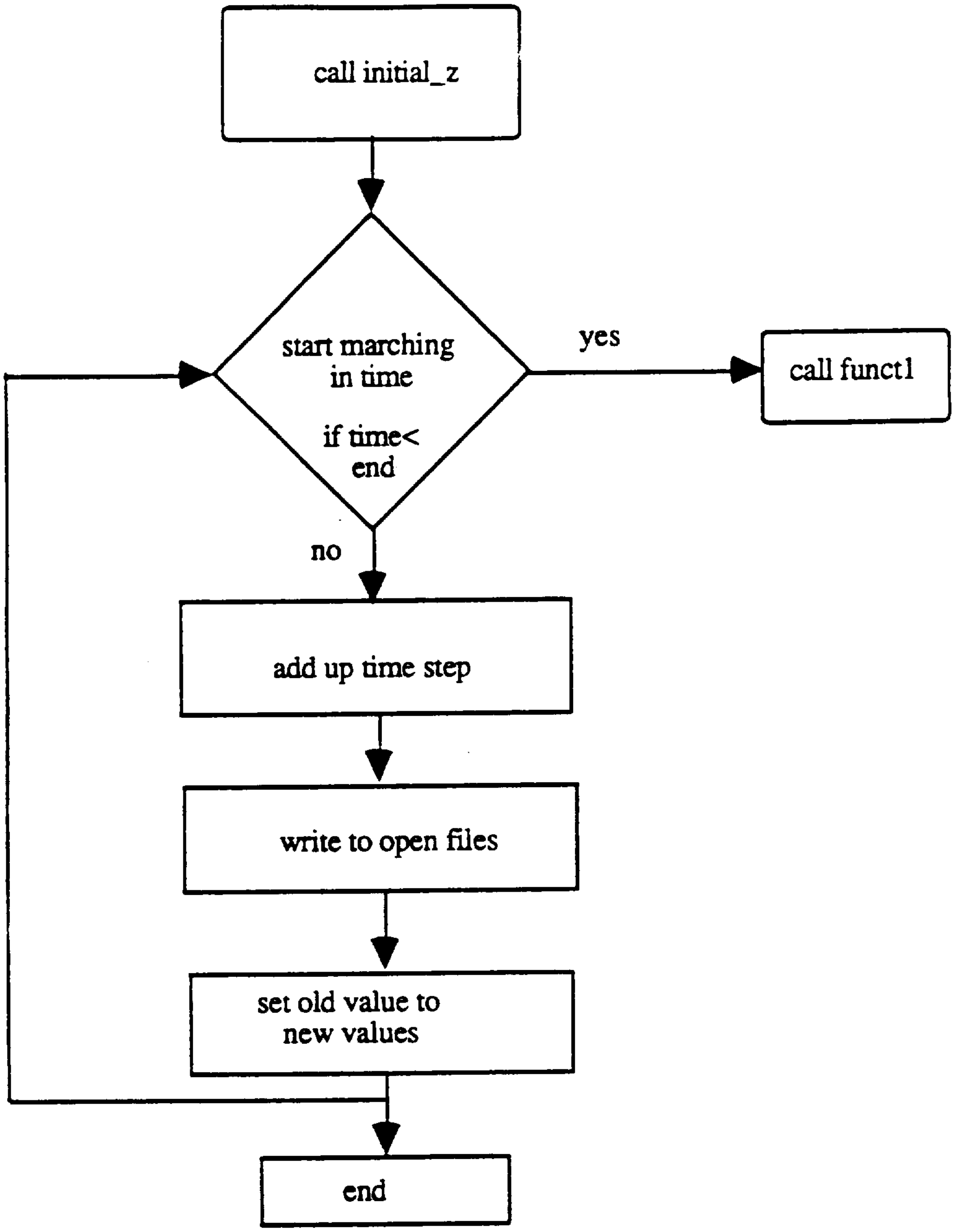


FIGURE C-7 THE FLOW CHART OF THE MAIN PROGRAM FOR THE THERMAL VACUUM DESORPTION

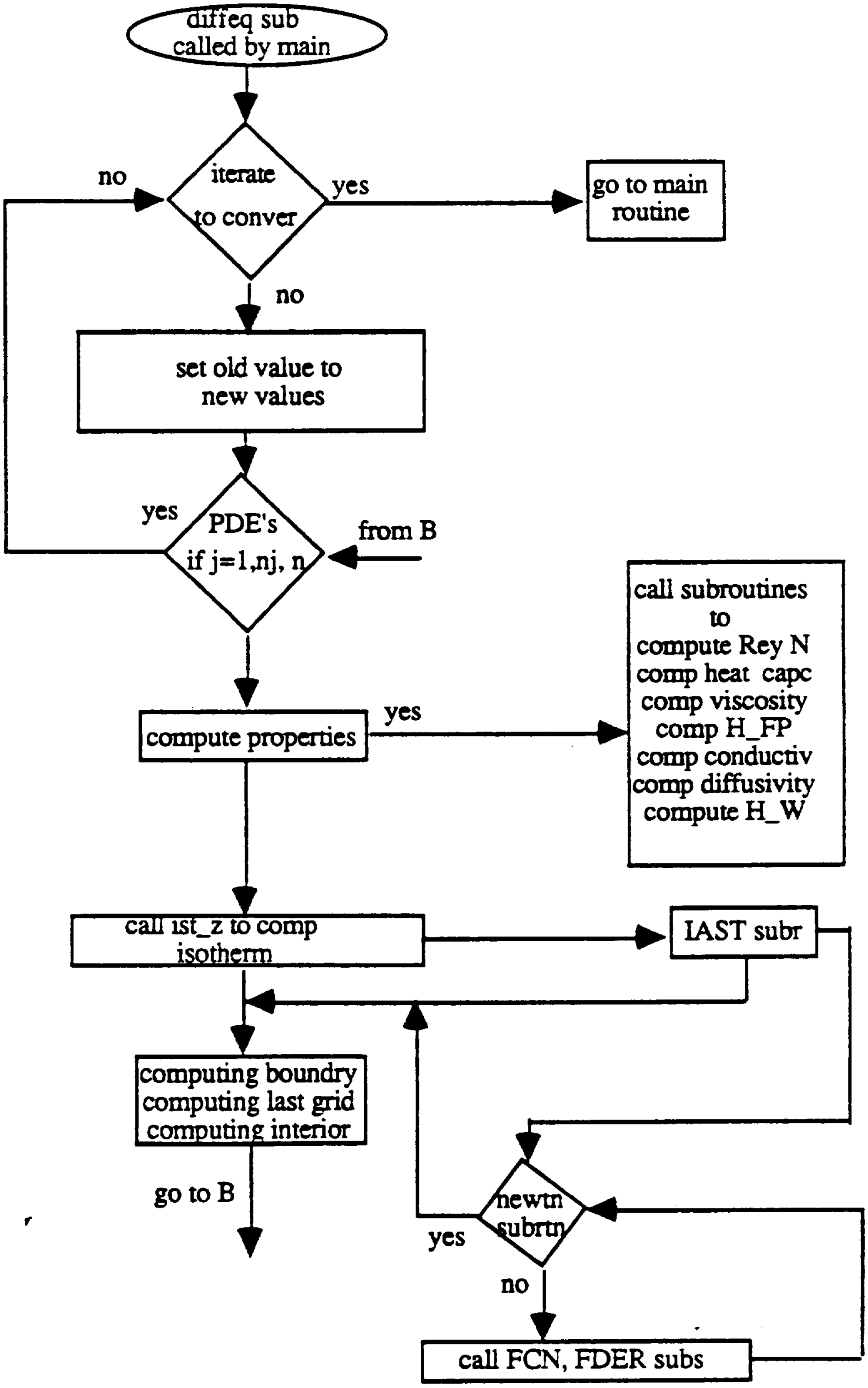


FIGURE C-8 THE FLOW CHART OF THE SUBROUTINES PROGRAM FOR THE THERMAL VACUUM DESORPTION

A list of program variables is given below with a brief description.

Variable	Description	Units	Type
LL	Time step index	none	Int
T_Time	current time	minutes	real
C(i,j)	matrix variable; i=variable ID; j=axial grid number; k=radial grid number	various	real*13
P(i,j).	Old of C array	various	real*13
L1	gas temperature	R	real
L2	solid temperature	R	real
L3	wall temperature	R	real
L4	velocity	ft/hr	real
L5	pressure	psia	real
l6	pressure	psia	real
PT	pressure	mm HG	real
NCOMP	number of component	non-dimensional	int
INERT	type of carrier gas	non-dimensional	real
EPSEX	porosity	cu ft/cu ft	real
RHOS	particle density	lbs/cu ft	real
Z	bed length	ft	real
VOID_B	porosity	cu ft/cu ft	real
D_I	inside bed diameter	ft	real
D_E	outside bed diameter	ft	real
S_B	cross sectional area of bed	ft	real
CP_WA	heat capacity of wall	BTU/F/LB	real
RO_WA	density of wall	lb/cu ft	real
CP_S	heat capacity of particle	real	
RO_S	density of particle	lb/cu ft	real
R_P	particle radius	ft	real
D_P	particle diameter	ft	real
heat(i)	heat of adsorption for each component	BTU/lb mole	real
NC	number of component	nonedimension	int
G_F	volumetric flow rate	cu ft/min	real
TO	initial temperature	R	real
R	ideal gas law constant	lbf/sq ft .lb mol/cu ft/R	real
PI	constant	3.14	real
M_W	Molecular weight	real	
GC	constant		real
K_F	mass transfer coefficient	ft/hr	real
NJ	grids number in axial direction	nonedimension	int
DELZ	grid size in axial direction	ft	real
DELT	time step	hr	real
CON_S1	pellet conductivity	BTU/hr/ft/F	real
ERROR	convergence criteria	various	real
N	number of equation	nonedimension	int
Y_F	component mole fraction	lb mole i/lb mole	real

A list of program variables is given below with a brief description (continued).

Variable	Description	Units	Type
P_TOT	Total Pressure	psia	real
BC_L1 to BC_L6	boundary condition for L1 to L6	various	real
W0	under relaxation coeff	nonedimension	real
H_OW	heat transfer coefficient outside of the wall	Btu/hr/F/sq ft	real
REY	Reynolds number	nonedimension	real
CP_P	specific heat	BTU/F/cu ft	real
VISC_P	viscosity	lb/ft/hr	real
H_FP	heat transfer coefficient fluid-particle	Btu/hr/sq ft/F	real
CON_L	fluid axial conductivity	Btu/hr/ft/F	real
CON_S	particle conductivity	Btu/hr/ft/F	real
H_W	heat transfer coefficient fluid-wall	Btu/hr/sq ft/F	real
V(i)	Langmuir constant	lb mole/lb solid	real
B(i)	Langmuir constant	nonedimension	real
PO(i)	Langmuir constant	nonedimension	real
PP	component partial Pressure	mm Hg	real
Q1	adsorbed concentration	lb mol/ lb solid	real
BC_C1(i)	boundary condition of molar concentration	lb mole/cu ft	real
W0-W3	under relaxation for each component and temperature	nonedimension	real
F3-F8	PDE's coefficient	various	real
PE_N_M	mass Peclet number	nonedimension	real
PE_N_H	heat Peclet number	nonedimension	real
AINT	interfacial area	sq ft/cu ft	real
U_F1	velocity	ft/hr	real

APPENDIX D

FLOW1MOL FORTRAN CODE

```
C
C this is the main routin. the initial values of matix C and parameters are
C called by "INITIAL_Z" subroutine. the main routine then calls the "FUNCT2_Z"
C subroutine to solve the discretized partial diffential equations (pde's).
C after convergance, the routine calls the "DIFFEQ1" subroutine, which is
C the maine routine for solving the momentum and the pressure equations,
C if it is desire to solve these two equations as the time progresses.
C
    IMPLICIT REAL*8(A-H,O-Z)
    INTEGER IOUT(5),AOUT(4)
    CHARACTER*5 LABEL(4)
    CHARACTER*8 DATE
    CHARACTER*12 OUTFILE,VISFILE
    REAL*4 PL(11,301),QP(14,301),T_TIME,K_F(4),END
    REAL*8 C1(14,301),P1(14,301),C11(14,301),P11(14,301),C12(14,301)
    COMMON/DATE/DATE
    COMMON/PLOT/ISAT,LABEL,AOUT,IOUT
    COMMON/BND_Z/C(14,301),P(14,301),N,NJ
    COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,U_F1,Z,END,NC,II,LL,K_F
    COMMON/INDIC/L1,L2,L3,L4,L5,L6,TIME,PERERR,MT
    COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
    COMMON/PRO_IND/ITEST2,ITEST1
    COMMON/NUM_OF_ITR/ITCNT1
    COMMON/INDIC1/IND(4)
        WRITE(*,*)'DESIGNATE A DATE FOR THE INPUT/OUTPUT FILES'
        WRITE(*,*)'FORMAT 00-00-00 WITH DOUBLE QUOTES'

   READ*,DATE
   OUTFILE = DATE // '.OUT'
   VISFILE = DATE // '.VIS'
   OPEN (50,FILE=OUTFILE,STATUS='NEW',RECL=32766)
   OPEN (66,FILE=VISFILE,STATUS='NEW',RECL=32766)

C IOUT  designates the 5 grid locations to be printed/plotted
C LABEL designates the 4 labels for printed/plotted data

C    DATA IOUT/2,15,30,45,61/
C    DATA LABEL/'ppH2O','pp N2','pp N2','gas T'/

C Initialize the c array with the initial guesses of the solution

     CALL INITIAL_Z
     T_TIME=0.
        TIME=0.
C Write header data to ACSII file

   WRITE(50,47)((LABEL(J),IOUT(I),I=1,5),J=1,4)

CJK     ICOT=1 not used again!
      T_TIME=0
         NLOOPS = END/DELT1
      IF(IND(2).EQ.0 .OR. ISAT.EQ.1) THEN
       IND0=0
       NEQ=2*NC+3
       NA=NEQ+2
      ELSE
       IND0=1
       NEQ=2*NC
       NA=NEQ+5
      END IF
```

```
C Begin looping through each time step;
C Call main routine for calculation of state conditions at current time step
     DO 301 LL=1,NLOOPS
     ITRAT=0
c call diffeq subroutine co compute the pressure and the velocity
 121       IF(ITRAT.GT.5)GO TO 201
      DO 30 I1=1,NA
        DO 30 I2=1,NJ
          C1(I1,I2)=C(I1,I2)
                  C12(I1,I2)=C(I1,I2)
          P1(I1,I2)=P(I1,I2)
30      CONTINUE
             ITRAT=ITRAT+1
      MT=1
      IF(ITRAT.EQ.1 .AND. LL.EQ.1)THEN
               ITCNT0=15
             ELSE IF(ITRAT.EQ.1) THEN
               ITCNT0=7
             ELSE
               ITCNT0=7
      END IF
      IF(IND0.EQ.1)THEN
        CALL DIFFEQ(C1,P1,NJ,NEQ,1,ITCNT0)
      ELSE
        CALL DIFFEQ0(C1,P1,NJ,NEQ,1)
      END IF
      IF(IND0.EQ.1)THEN
        DO 35 I1=1,NEQ
          DO 35 I2=1,NJ
            C11(I1,I2)=C1(I1,I2)
            P11(I1,I2)=P1(I1,I2)
 35       CONTINUE
        L13=2*NC+1
        DO I1=1,3
          DO I2=1,NJ
            C1(I1,I2)=C(L13,I2)
            P1(I1,I2)=P(L13,I2)
          END DO
          L13=L13+1
               END DO
        MT=0
        CALL DIFFEQ(C1,P1,NJ,3,3,4)
        DO 31 I1=1,NC*2
          DO 31 I2=1,NJ
            C(I1,I2)=C11(I1,I2)
            P(I1,I2)=P11(I1,I2)
 31       CONTINUE
        L13=2*NC+1
        DO I1=1,3
          DO I2=1,NJ
            C(L13,I2)=C1(I1,I2)
            P(L13,I2)=P1(I1,I2)
                 END DO
          L13=L13+1
               END DO
               DO MM1=1,NC
                    DO MM2=1,NJ
                      IF (C(MM1,MM2).GT.1.0E-15)THEN
            DIF=ABS(C(MM1,MM2)-C12(MM1,MM2))/C(MM1,MM2)
                        IF(DIF.GT.1.0E-2)THEN
                  GO TO 121
            END IF
           END IF
                END DO
```

```
             END DO
        ELSE
         DO I1=1,NEQ
          DO I2=1,NJ
           C(I1,I2)=C1(I1,I2)
           P(I1,I2)=P1(I1,I2)
          END DO
             END DO
        END IF
        ITEST1=1
 201      IF(ISAT.EQ.1)GO TO 303
        DO 300 I2=1,NJ
         C1(1,I2)=C(L4,I2)
         P1(1,I2)=P(L4,I2)
         C1(2,I2)=C(L5,I2)*760./14.696
         P1(2,I2)=P(L5,I2)*760./14.696
         C1(3,I2)=C(L6,I2)*760./14.696
         P1(3,I2)=P(L6,I2)*760./14.696
         C1(4,I2)=C(L1,I2)
         P1(4,I2)=P(L1,I2)
 300      CONTINUE
        IF(IND0.EQ.1)THEN
         CALL DIFFEQ(C1,P1,NJ,3,2,5)
        ELSE
         CALL DIFFEQ0(C1,P1,NJ,3,2)
        END IF
        DO 400 I2=1,NJ
         C(L4,I2)=C1(1,I2)
         C(L5,I2)=C1(2,I2)*14.696/760.
         C(L6,I2)=C1(3,I2)*14.696/760.
 400      CONTINUE

C Add time step (converted from dimless to minutes)

 303      T_TIME = T_TIME+DELT*Z/U_F1*60.
        TIME=T_TIME/60.
        ITEST1=0
CJK Do we want to common DELT or DELT1 (both not needed)?
        IF(LL.EQ.25)THEN
         DELT1=DELT1*2
         DELT=DELT*2
        END IF
        IF(LL.EQ.50)THEN
         DELT1=DELT1*2
         DELT=DELT*2
        END IF
CJK Eliminate above?

C Convert to units used for plotting (psia to mmHg, R to F, R*8 to R*4)

         DO 74 M2=1,NJ
          DO I1=1,NC
           QP(I1,M2)=C(I1,M2)*10.73D0*C(L1,M2)*760/14.696
          END DO
          DO I1=NC+1,2*NC
           QP(I1,M2)=C(I1,M2)
          END DO
          QP(L1,M2)=C(L1,M2)-460.
          QP(L2,M2)=C(L2,M2)-460.
          QP(L3,M2)=C(L3,M2)-460.
          QP(L4,M2)=C(L4,M2)
          QP(L5,M2)=C(L5,M2)
          QP(L6,M2)=C(L6,M2)
 74      CONTINUE
```

```
C ISAMP is the sampling frequency, where "10" plots every 10th time step data

    ISAMP=10
    IF(T_TIME.LT.30) ISAMP=1

C  Print runtime output headers every 4th and data every time step

    IF((LL/(ISAMP*4))*ISAMP*4.EQ.LL.OR.LL.EQ.1) THEN
     WRITE(6,49)
     WRITE(6,41)'       ',('  GRID ',IOUT(I),'  ',I=1,5)
    ENDIF
    WRITE(6,49)
    WRITE(6,40)'**** TIME = ',T_TIME,'LOOPS = ',ITCNT1,'  ***'
    WRITE(6,42)LABEL(1),(QP(AOUT(1),IOUT(I)),I=1,5)
    WRITE(6,42)LABEL(2),(QP(AOUT(2),IOUT(I)),I=1,5)
    WRITE(6,43)LABEL(3),(QP(AOUT(3),IOUT(I)),I=1,5)
    WRITE(6,43)LABEL(4),(QP(AOUT(4),IOUT(I)),I=1,5)

C  Print time slice data 1/10 less often

    IF((LL/(ISAMP*10))*ISAMP*10.EQ.LL) THEN
      WRITE(66,44)(LABEL(I),I=1,4),T_TIME
      WRITE(66,45)((QP(AOUT(I),IOUT(J)),I=1,4),J=1,5)
    ENDIF

C  Print pp, temp output data every ISAMP

    IF((LL/ISAMP)*ISAMP.EQ.LL) THEN
     WRITE(50,46)T_TIME,((QP(AOUT(J),IOUT(I)),I=1,5),J=1,4)
    ENDIF

    DO 78 I1=1,N
     DO 78 I2=1,NJ
      P(I1,I2)=C(I1,I2)
 78   CONTINUE
 301  CONTINUE

 40   FORMAT(1X,A12,G10.2,32X,A8,I4,A5,/)
 41   FORMAT(1X,A7,5(A7,I3,A2))
 42   FORMAT(1X,A5,5(2X,G10.2))
 43   FORMAT(1X,A5,5(2X,F10.2))
 44   FORMAT(1X,/,4(A5,','),E15.3)
 45   FORMAT(1X,20(4(E15.5,','),/))
 46   FORMAT(1X,21(E15.5,','))
 47   FORMAT(1X,'TIME,',20(A5,I3,','))
 49   FORMAT(1X,72('-'))
     STOP
     END
 C
 C this subroutine is being called by maine routine once to get the intial
 c values and the neccessary parameters
 C
      SUBROUTINE INITIAL_Z
       IMPLICIT REAL*8(A-H,O-Z)
       INTEGER IOUT(5),AOUT(4)
       REAL*4 K_F(4),END
       REAL*8 M_AVE,M_W(4),C_FO(4),IN_L1,IN_L2,IN_L3,IN_L4,
     . IN_L5,IN_L6,ss(4),q2(4)
       CHARACTER*5 LABEL(4)
       CHARACTER*8 DATE
       CHARACTER*14 INFILEA,INFILEB
       COMMON/DATE/DATE
       COMMON/PLOT/ISAT,LABEL,AOUT,IOUT
```

```
     COMMON/BND_Z/C(14,301),P(14,301),N,NJ
     COMMON/PROP_B_Z/D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,
   . CP_WA,CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,
   . X_WI_K,D_LMI,D_LMA,H_FW,H_OW
     COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4),M_W
     COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,U_F1,Z,END,NC,II,LL,K_F
     COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
     COMMON/PRO_IND/ITEST2,ITEST1
     COMMON/INDIC/L1,L2,L3,L4,L5,L6,TIME
     COMMON/INDIC1/IND(4)
     COMMON/NCY/NC1
     COMMON/GAS/INERT,NCOMP
     COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
C     WRITE(*,*)"SELECT FILE CONTAINING INPUT SET A"
    INFILEA = DATE // '_A.DAT'
         OPEN (UNIT=40,FILE=INFILEA,STATUS='OLD')
C     WRITE(*,*)'SELECT FILE CONTAINING INPUT SET B'
    INFILEB = DATE // '_B.DAT'
         OPEN (UNIT=41,FILE=INFILEB,STATUS='OLD')
C
C INITIALIZATION
C
    READ(UNIT=40,FMT=*)END,DELT1,NJ,DELZ1,TO,G_F,T_O,NC,INERT,P_TOT,
   . BC_C(1),BC_C(2),BC_C(3),BC_C(4),Z,D_E,D_I,RO_WA,EPSEX,
   . AINT,CON_S,CP_S,H_FW,H_OW,RO_S,M_W(1),M_W(2),M_W(3),M_W(4),
   . HEAT(1),HEAT(2),HEAT(3),HEAT(4),K_F(1),K_F(2),K_F(3),K_F(4),
   . ind(1),ind(2),ind(3),ind(4),IDES1,ISAT
    READ(UNIT=41,FMT=*)
   . LABEL(1),LABEL(2),LABEL(3),LABEL(4),AOUT(1),AOUT(2),
   . AOUT(3),AOUT(4),IOUT(1),IOUT(2),IOUT(3),IOUT(4),IOUT(5),
   . S_B,CON_WA,CON_WI_Q,CON_WI_K,CP_WA,CP_WI_Q,CP_WI_K,
   . RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,X_WI_K,RA,RAV,EPSIN,RHOS,
   . ALPHA1,R_P,D_P,BC_L1,BC_L2,BC_L3,BC_L5,BC_L6,
   . IN_L1,IN_L2,IN_L3,IN_L5,IN_L6,R,PI,ITEST2,ITEST1
c number of component
     NC1=NC
c number of equation
     N=NC*2+6
     ALF=(1.0D0-EPSEX)/EPSEX
     PT=P_TOT*760./14.696
    N1=1
    DO I1=1,4
     IF(IND(I1).EQ.1)THEN
      IF(I1.EQ.3)N2=N1
      M_W(N1)=M_W(I1)
      BC_C(N1)=BC_C(I1)
      HEAT(N1)=HEAT(I1)
      K_F(N1)=K_F(I1)
      IF(I1.EQ.3)THEN
       IDES=N1
       IND3=N1
      ELSE IF(I1.EQ.4)THEN
       IND4=N1
      END IF
      N1=N1+1
     END IF
    END DO
     SUM_Y=0
     M_AVE=0.
     YO=0
c compute the initial mole fraction and average molecular weight
     DO 10 I=1,NC
      Y_F=BC_C(I)/P_TOT
      YO=Y_F+YO
```

```
        SUM_Y=SUM_Y+Y_F
        BC_C(I)=P_TOT*Y_F/R/BC_L1
        M_AVE=M_AVE+Y_F*M_W(I)
        C_FO(I)=BC_C(I)
 10     CONTINUE
      M_AVE=M_AVE+(1-YO)*28.0
      RO_FO=P_TOT/(BC_L1*R)
      GN=G_F*RO_FO/S_B/60.
      U_F1=G_F/(S_B*EPSEX)
      BC_L4=U_F1
c make the grid size and the time step dimensionless
      DELZ=DELZ1/Z
      DELT=U_F1*DELT1/Z
c
c  L1 is the gas temp, L2 is the solid temp, L3 is the wall temp, L4 is the
c velocity, L5 is the total pressure, and L6 is the total pressure witout
c thermal effect
c
      L1=2*NC+1
      L2=L1+1
      L3=L2+1
      L4=L3+1
      L5=L4+1
      L6=L5+1
      DO 20 I1=1,N
       DO 20 I2=1,NJ
        P(I1,I2)=0.0D0
        C(I1,I2)=0.0D0
 20     CONTINUE
c
c set the first grid equal to the initial condition
c
      DO 22 I1=1,NC
       C(I1,1)=C_FO(I1)
 22     CONTINUE
      DO 24 I1=1,NC
       DO 24 I2=2,NJ
        C(I1,I2)=0.0
 24     CONTINUE
      DO 30 I=1,NJ
       P(L1,I)=IN_L1
       P(L2,I)=IN_L2
       P(L3,I)=IN_L3
       P(L4,I)=BC_L4
       P(L5,I)=BC_L5
       P(L6,I)=BC_L6
       C(L1,I)=BC_L1
       C(L2,I)=BC_L2
       C(L3,I)=BC_L3
       C(L4,I)=BC_L4
       C(L5,I)=BC_L5
       C(L6,I)=BC_L6
30     CONTINUE
     IF(IND(3).EQ.1)THEN
      DO I1=1,NJ
       C(N2,I1)=C_FO(N2)
      END DO
     END IF
      IF(IDES1.EQ.1)THEN
       SS(IDES)=C(IDES,1)
       CALL IST_Z(1,C(L2,1),BC_L1,SS,Q2)
       DO I=1,NC
        Q2(I)=Q2(I)*RO_S
       END DO
```

```
        DO I=1,NJ
          C(IDES,I)=C_FO(IDES)
          P(IDES,I)=C_FO(IDES)
          C(IDES+NC,I)=Q2(IDES)
          P(IDES+NC,I)=Q2(IDES)
        END DO
       END IF
      IF(ISAT.EQ.1)THEN
        DO I1=1,NC
          SS(I1)=BC_C(I1)
        END DO
        CALL IST_Z(1,C(L2,1),BC_L1,SS,Q2)
        DO I=1,NC
          Q2(I)=Q2(I)*RO_S
        END DO
        DO I1=1,NC
          DO I2=1,NJ
            C(I1,I2)=BC_C(I1)
            C(I1+NC,I2)=Q2(I1)
            P(I1,I2)=BC_C(I1)
            P(I1+NC,I2)=Q2(I1)
          END DO
        END DO
        DO I1=1,NC
          BC_C(I1)=0
        END DO
        BC_C(IND3)=c_fo(IND3)
      end if
      RETURN
      END
C
C
c this subroutine is being called by maine routine to compute the variable in
c C matrix. in this routine first velocity profile is being determined. the
c conductivity, diffusivity,pososity....... are being calculated in this
c routine by calling the appropeniate subroutine. the routine obtained the
c C matix in axial and radial directions. it itrates till it conveges to the
c allowable error
c
      SUBROUTINE FUNCT1(J)
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*4 K_F(4),END
       REAL*8 M_AVE,M_W(4),KGAS,K_FP(4,301),C1(14,301),
     . P1(14,301),REY_P(301),KEFF,D_L(4),H(4),YO(4),SS(4)
       DIMENSION Q2(4),C2(14,301),PE_N_M(4),RATE_C2(301)
       COMMON/PR_OLD_Z/ RO_P(301),CP_P(301),CON_FPP(301),
     . CON_LPP(301),D_LPP(4,301),H_FP(301),H_TTP(301),H_IIP(301),
     . H_OWPP(301),H_FWPP(301),Q(4,301),VISC_P(301),RATE_C1(301)
       COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,301)
       COMMON/PROP_B_Z/D_L,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,
     . CP_WA,CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,
     . X_WI_K,D_LMI,D_LMA,H_FW,H_OW
       COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4),M_W
       COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,U_F1,Z,END,NC,II,LL,K_F
       COMMON/INDIC/L1,L2,L3,L4,L5,L6,TIME,PERERR,MT
       COMMON/INDIC1/IND(4)
       COMMON/PRO_IND/ITEST2,ITEST1
       COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
       COMMON/GAS/INERT,NCOMP
       COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
       DATA R,PI,ICOM/10.73D0,3.141593D0,1/

C       IF(LL.EQ.ICOM)THEN
```

```
C       BC_L1=TEMPIN(TIME)
C       ICOM=LL+1
C      END IF
    IF(ITEST1.EQ.1)THEN
     D_T=D_E+D_I
     X_W=D_E-D_I
c calculate the area for the heat transfer, inside and outside
     A_C=2*D_I/((D_I+D_E)*X_W)
     A_INS=2*D_E/((D_I+D_E)*X_W)
     RO_F=C(L5,1)/(C(L1,1)*R)
c calculate superfictial mass velocity
     GN2=G_F*RO_F/S_B/60.
     IF(ITEST1.EQ.1 .OR. ITEST2.EQ.1)THEN
      DO 199 J1=1,NJ
       RO_F=C(L5,J1)/(C(L1,J1)*R)
       YTOT=0.
c calculate the mole fraction
       DO I1=1,NC-1
        YO(I1)=C(I1,J1)/RO_F
        YTOT=YO(I1)+YTOT
       END DO
       YO(NC)=1-YTOT
       M_AVE=0.
c calculate the average molecular weight
       DO I1=1,NC
        M_AVE=M_AVE+M_W(I1)*YO(I1)
       END DO
       M_AVE=M_AVE+(1-YTOT)*28.0
       TEMP=C(L1,J1)
       GN=GN2
c calcilate the viscosity
       VISC_P(J1)=VIS(TEMP)*60
c calculate reynold's number
       REY=RE(GN,TEMP)
c calculate the heat capacity of the fluid flow
       CP_P(J1)=CPGAS(TEMP,YO)
c calculate the film coefficient between the particle and fluid
       H_FP(J1)=HFILM(M_AVE,TEMP,REY,CP_P(J1))*60.
c chane the unit on pressure
       PP=C(L5,J1)*760./14.77
c calculate the axial effective conductivity
       CON_LPP(J1)=EFFK(GN,TEMP,CP_P(J1),RO_F,PP,YO)*60.
c calculate axial mass dispersion for each component
       DO I1=1,NC
        D_LPP(I1,J1)=EFFD(I1,GN,TEMP,RO_F,PP,YO)*60.
        IF(D_LPP(I1,J1).LT.0) THEN
         D_LPP(I1,J1)=D_LPP(I1,J1-1)
        END IF
       END DO
 199   CONTINUE
     END IF
     ITEST1=0
    ENDIF
c start itration for each time step
    IF(MT.EQ.1)THEN
     DO I1=1,NC
      D_L(I1)=D_LPP(I1,J)
     END DO
     DO I1=1,NC
      SS(I1)=C(I1,J)
     END DO
c calculate the equlibrum 2=IST theory; 1=Langmuir theory
     CALL IST_Z(1,C(L2,J),BC_L1,SS,Q2)
     DO 40 I=1,NC
```

```
       Q(I,J)=Q2(I)*RO_S
 40   CONTINUE
c compute the C array for the first grid
     IF(J.EQ.1)THEN
c compute molar concentration of each component
      DO 85 M1=1,NC
       PE_N_M(M1)=U_F1*Z/D_L(M1)
85     CONTINUE
          C_TOT=0
      DO 90 M1=1,NC-1
       F2=ALF*K_F(M1)*AINT*Z/U_F1
       F(M1)=1.0D0/PE_N_M(M1)/DELZ**2*(BC_C(M1)-2*C(M1,J)+
   .   C(M1,J+1))-1.0D0/U_F1/DELZ*(C(M1,J)-BC_C(M1))*C(L4,J)-
   .   F2*(Q(M1,J)-C(NC+M1,J))-(C(M1,J)-P(M1,J))/DELT
           C_TOT=C_TOT+C(M1,J)
 90    CONTINUE
          F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute amount adsorbed on the pellet
      DO 70 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       F(NC+M1)=F1*(Q(M1,J)-C(M1+NC,J))-(C(M1+NC,J)-P(NC+M1,J))/DELT
 70    CONTINUE
      FLUX1=0
      FLUX2=0
      DO 100 M1=1,NC
c compute total amount adsorbed
           FLUX0=AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
       FLUX2=FLUX2+FLUX0
       FLUX1=FLUX1+FLUX0*HEAT(M1)
 100    CONTINUE
      RATE_C1(J)=FLUX2
      RATE_C2(J)=FLUX1
C
C--------------------------------------------------------------
C THE LAST ROW
C
c compute the C array for the last grid
     ELSE IF(J .EQ. NJ) THEN
      DO 125 M1=1,NC
       PE_N_M(M1)=U_F1*Z/D_L(M1)
 125    CONTINUE
          C_TOT=0
c compute the molar concentraion of each component
      DO 130 M1=1,NC-1
       F2=ALF*K_F(M1)*AINT*Z/U_F1
       F(M1)=1.0D0/PE_N_M(M1)/DELZ**2*(2*C(M1,J-1)-2*C(M1,J))-
   .   1.0D0/U_F1/DELZ*(C(M1,J)-C(M1,J-1))*C(L4,J)-F2*(Q(M1,J)-
   .   C(NC+M1,J))-(C(M1,J)-P(M1,J))/DELT
           C_TOT=C_TOT+C(M1,J)
 130    CONTINUE
          F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute the amount adsorbed on the bed
      DO 110 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       F(NC+M1)=F1*(Q(M1,J)-C(M1+NC,J))-(C(M1+NC,J)-P(NC+M1,J))/DELT
 110    CONTINUE
      FLUX1=0.
      FLUX2=0.
      DO 140 M1=1,NC
c compute total amount adsorbed on the particle
           FLUX0=AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
       FLUX2=FLUX2+FLUX0
       FLUX1=FLUX1+FLUX0*HEAT(M1)
 140    CONTINUE
```

```
      RATE_C1(J)=FLUX2
      RATE_C2(J)=FLUX1
C
C--------------------------------------------------------------
C
C INTERIOR ROWS
C computation of C aray for interior grids
     ELSE
      DO 165 M1=1,NC
       PE_N_M(M1)=U_F1*Z/D_L(M1)
 165   CONTINUE
          C_TOT=0.
c compute the molar concentration
      DO 170 M1=1,NC-1
       F2=ALF*K_F(M1)*AINT*Z/U_F1
       F(M1)=1.0D0/PE_N_M(M1)/DELZ**2*(C(M1,J-1)-2*C(M1,J)+
     . C(M1,J+1))-1.0D0/U_F1/DELZ*(C(M1,J)-C(M1,J-1))*C(L4,J)-
     . F2*(Q(M1,J)-C(NC+M1,J))-(C(M1,J)-P(M1,J))/DELT
          C_TOT=C_TOT+C(M1,J)
 170   CONTINUE
          F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute the amount adsorbed on
      DO 150 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       F(NC+M1)=F1*(Q(M1,J)-C(M1+NC,J))-(C(M1+NC,J)-P(NC+M1,J))/DELT
 150   CONTINUE
      FLUX1=0.
      FLUX2=0.
      DO 180 M1=1,NC
c compute toal amount adsorbed
           FLUX0=AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
       FLUX2=FLUX2+FLUX0
       FLUX1=FLUX1+FLUX0*HEAT(M1)
 180   CONTINUE
      RATE_C1(J)=FLUX2
      RATE_C2(J)=FLUX1
C--------------------------------------------------------------
      END IF

     ELSE
c compute the C array for the first grid
C
      RO_F=C(L5,J)/(C(L1,J)*R)
      CP_F=CP_P(J)
      H_FS=H_FP(J)
      CON_L=CON_LPP(J)
c Peclet number for the solid
      PE_N_S=U_F1*RO_S*CP_S*Z/CON_S
c Peclet number for the heat
      PE_N_H=U_F1*RO_F*CP_F*Z/CON_L
c these are coefficient the Discretized PDE's
      F3=ALF*H_FS*Z*AINT/(RO_F*CP_F*U_F1)
      F4=H_FW*Z*4/(U_F1*D_I*EPSEX*RO_F*CP_F)
      F5=H_FS*AINT*Z/(U_F1*RO_S*CP_S)
      F6=Z/(RO_S*CP_S*U_F1)
      F7=Z*H_FW*A_C/U_F1/RO_WA/CP_WA
      F8=Z*H_OW*A_INS/U_F1/RO_WA/CP_WA
      IF(J.EQ.1)THEN
c compute gas temperature
       F(1)=1.0D0/PE_N_H/DELZ**2*(BC_L1-2*C(1,J)+C(1,J+1))-
     . 1.0D0/(DELZ*U_F1)*(C(1,J)-BC_L1)*C(L4,J)-F3*(C(1,J)-C(2,J))
     . -F4*(C(1,J)-C(3,J))-(C(1,J)-P(1,J))/DELT
c compute the solid temperature
       F(2)=1.0D0/PE_N_S/DELZ**2*(BC_L1-2*C(2,J)+C(2,J+1))+
```

```
     . F5*(C(1,J)-C(2,J))-F6*RATE_C2(J)-(C(2,J)-P(2,J))/DELT
c compute the wall temperature
       F(3)=F7*(C(1,J)-C(3,J))-F8*(C(3,J)-TO)-(C(3,J)-
     . P(3,J))/DELT
C
C---------------------------------------------------------------
C THE LAST ROW
C
      ELSE IF(J.EQ.NJ)THEN
c compute the gas temperature
       F(1)=1.0D0/PE_N_H/DELZ**2*(C(1,J-1)-2.*C(1,J)+
     . C(1,J-1))-1.0D0/(DELZ*U_F1)*(C(1,J)-C(1,J-1))*C(14,J)-
     . F3*(C(1,J)-C(2,J))-F4*(C(1,J)-C(3,J))-(C(1,J)-P(1,J))/
     . DELT
c compute the solid temperature
       F(2)=1.0D0/PE_N_S/DELZ**2*(C(2,J-1)-2*C(2,J)+C(2,J-1))+
     . F5*(C(1,J)-C(2,J))-F6*RATE_C2(J)-(C(2,J)-P(2,J))/DELT
c compute the wall temperature
       F(3)=F7*(C(1,J)-C(3,J))-F8*(C(3,J)-TO)-(C(3,J)-
     . P(3,J))/DELT
      ELSE
C---------------------------------------------------------------
C THE INTERIOR ROW
C
c compute gas temperature
       F(1)=1.0D0/PE_N_H/DELZ**2*(C(1,J-1)-2*C(1,J)+C(1,J+1))-
     . 1.0D0/(DELZ*U_F1)*(C(1,J)-C(1,J-1))*C(14,J)-F3*
     . (C(1,J)-C(2,J))-F4*(C(1,J)-C(3,J))-(C(1,J)-P(1,J))/DELT
c compute solid temperature
       F(2)=1.0D0/PE_N_S/DELZ**2*(C(2,J-1)-2*C(2,J)+C(2,J+1))+
     . F5*(C(1,J)-C(2,J))-F6*RATE_C2(J)-(C(2,J)-P(2,J))/DELT
c compute wall temperature
       F(3)=F7*(C(1,J)-C(3,J))-F8*(C(3,J)-TO)-(C(3,J)-
     . P(3,J))/DELT
       END IF
      END IF
 210  RETURN
      END

C
C
c this subroutine is being called by maine routine to compute the variable in
c C matrix. in this routine first velocity profile is being determined. the
c conductivity, diffusivity,pososity....... are being calculated in this
c routine by calling the approperiate subroutine. the routine obtained the
c C matix in axial and radial directions. it itrates till it conveges to the
c allowable error
c
      SUBROUTINE FUNCT0(J)
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*4 K_F(4),END
       REAL*8 M_AVE,M_W(4),KGAS,K_FP(4,301),C1(14,301),
     . P1(14,301),REY_P(301),KEFF,D_L(4),H(4),YO(4),SS(4)
       DIMENSION Q2(4),C2(14,301),PE_N_M(4)
       COMMON/PR_OLD_Z/ RO_P(301),CP_P(301),CON_FPP(301),
     . CON_LPP(301),D_LPP(4,301),H_FP(301),H_TTP(301),H_IIP(301),
     . H_OWPP(301),H_FWPP(301),Q(4,301),VISC_P(301),RATE_C1(301)
       COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,301)
       COMMON/PROP_B_Z/D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,
     . CP_WA,CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,
     . X_WI_K,D_LMI,D_LMA,H_FW,H_OW
       COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4),M_W
       COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,U_F1,Z,END,NC,II,LL,K_F
```

```
      COMMON/INDIC/L1,L2,L3,L4,L5,L6,TIME,PERERR
      COMMON/INDIC1/IND(4)
      COMMON/PRO_IND/ITEST2,ITEST1
      COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
      COMMON/GAS/INERT,NCOMP
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA R,PI,ICOM/10.73D0,3.141593D0,1/

C
C       IF(LL.EQ.ICOM)THEN
C        BC_L1=TEMPIN(TIME)
C        ICOM=LL+1
C       END IF
     IF(ITEST1.EQ.1)THEN
      D_T=D_E+D_I
      X_W=D_E-D_I
c calculate the area for the heat transfer, inside and outside
      A_C=2*D_I/((D_I+D_E)*X_W)
      A_INS=2*D_E/((D_I+D_E)*X_W)
      RO_F=C(L5,1)/(C(L1,1)*R)
c calculate superficial mass velocity
      GN2=G_F*RO_F/S_B/60.
      IF(ITEST1.EQ.1 .OR. ITEST2.EQ.1)THEN
       DO 199 J1=1,NJ
        RO_F=C(L5,J1)/(C(L1,J1)*R)
        YTOT=0.
c calculate the mole fraction
        DO I1=1,NC-1
         YO(I1)=C(I1,J1)/RO_F
         YTOT=YO(I1)+YTOT
        END DO
        YO(NC)=1-YTOT
        M_AVE=0.
c calculate the average molecular weight
        DO I1=1,NC
         M_AVE=M_AVE+M_W(I1)*YO(I1)
        END DO
        M_AVE=M_AVE+(1-YTOT)*28.0
        TEMP=C(L1,J1)
        GN=GN2
c calcilate the viscosity
        VISC_P(J1)=VIS(TEMP)*60
c calculate reynold's number
        REY=RE(GN,TEMP)
c calculate the heat capacity of the fluid flow
        CP_P(J1)=CPGAS(TEMP,YO)
c calculate the film coefficient between the particle and fluid
        H_FP(J1)=HFILM(M_AVE,TEMP,REY,CP_P(J1))*60.
c chane the unit on pressure
        PP=C(L5,J1)*760./14.77
c calculate the axial effective conductivity
        CON_LPP(J1)=EFFK(GN,TEMP,CP_P(J1),RO_F,PP,YO)*60.
c calculate axial mass dispersion for each component
        DO I1=1,NC
         D_LPP(I1,J1)=EFFD(I1,GN,TEMP,RO_F,PP,YO)*60.
         IF(D_LPP(I1,J1).LT.0) THEN
          D_LPP(I1,J1)=D_LPP(I1,J1-1)
         END IF
        END DO
 199    CONTINUE
      END IF
      ITEST1=0
     ENDIF
c start itration for each time step
```

```
      RO_F=C(L5,J)/(C(L1,J)*R)
      CP_F=CP_P(J)
      H_FS=H_FP(J)
      CON_L=CON_LPP(J)
      DO I1=1,NC
       D_L(I1)=D_LPP(I1,J)
      END DO
c Peclet number for the solid
      PE_N_S=U_F1*RO_S*CP_S*Z/CON_S
c Peclet number for the heat
      PE_N_H=U_F1*RO_F*CP_F*Z/CON_L
c these are coefficient the Discretized PDE's
      F3=ALF*H_FS*Z*AINT/(RO_F*CP_F*U_F1)
      F4=H_FW*Z*4/(U_F1*D_I*EPSEX*RO_F*CP_F)
      F5=H_FS*AINT*Z/(U_F1*RO_S*CP_S)
      F6=AINT*Z/(RO_S*CP_S*U_F1)
      F7=Z*H_FW*A_C/U_F1/RO_WA/CP_WA
      F8=Z*H_OW*A_INS/U_F1/RO_WA/CP_WA
      DO I1=1,NC
       SS(I1)=C(I1,J)
      END DO
c calculate the equlibrum 2=IST theory; 1=Langmuir theory
      CALL IST_Z(1,C(L2,J),BC_L1,SS,Q2)
      DO 40 I=1,NC
       Q(I,J)=Q2(I)*RO_S
 40   CONTINUE
c compute the C array for the first grid
     IF(J.EQ.1)THEN
c compute molar concentration of each component
      DO 85 M1=1,NC
       PE_N_M(M1)=U_F1*Z/D_L(M1)
       H(M1)=1.0D0/DELT+2/PE_N_M(M1)/DELZ**2+1/U_F1/DELZ*C(L4,J)
85     CONTINUE
          C_TOT=0
      DO 90 M1=1,NC-1
       F2=ALF*K_F(M1)*AINT*Z/U_F1
       F(M1)=-C(M1,J)+1./H(M1)*(1.0D0/PE_N_M(M1)/DELZ**2*(BC_C(M1)+
   .   C(M1,J+1))-1.0D0/U_F1/DELZ*(-BC_C(M1)*C(L4,J))-F2*(Q(M1,J)-
   .   C(NC+M1,J))+P(M1,J)/DELT)
          C_TOT=C_TOT+C(M1,J)
 90    CONTINUE
          F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute amount adsorbed on the pellet
      DO 70 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       A2=1/DELT+F1
       F(NC+M1)=-C(NC+M1,J)+1/A2*(F1*Q(M1,J)+P(NC+M1,J)/DELT)
 70    CONTINUE
      FLUX1=0
      FLUX2=0
      DO 100 M1=1,NC
c compute total heat of adsorption
       FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
c compute total amount adsorbed
       FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
 100   CONTINUE
      RATE_C1(J)=FLUX2
c compute gas temperature
      AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
      F(L1)=-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*(BC_L1+C(L1,J+1))-
   .  1.0D0/(DELZ*U_F1)*(-BC_L1*C(L4,J))-F3*(-C(L2,J))-F4*(-C(L3,J))+
   .  P(L1,J)/DELT)
c compute solid temperatyre
      F(L2)=-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2*(BC_L1-2*C(L2,J)+
```

```
   . C(L2,J+1))+F5*(C(L1,J)-C(L2,J))-F6*FLUX1)+P(L2,J)
c compute wall temperature
     F(L3)=-C(L3,J)+DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+
   . P(L3,J)
C
C------------------------------------------------
C THE LAST ROW
C
c compute the C array for the last grid
    ELSE IF(J .EQ. NJ) THEN
     DO 125 M1=1,NC
      PE_N_M(M1)=U_F1*Z/D_L(M1)
      H(M1)=1.D0/DELT+2.D0/PE_N_M(M1)/DELZ**2+1/U_F1/DELZ*C(L4,J)
 125  CONTINUE
         C_TOT=0.
c compute the molar concentraion of each component
     DO 130 M1=1,NC-1
      F2=ALF*K_F(M1)*AINT*Z/U_F1
      F(M1)=-C(M1,J)+1.D0/H(M1)*(1.0D0/PE_N_M(M1)/DELZ**2*
   .  (C(M1,J-1)+C(M1,J-1))-1.0D0/U_F1/DELZ*(-C(M1,J-1)*
   .  C(L4,J))-F2*(Q(M1,J)-C(NC+M1,J))+P(M1,J)/DELT)
      TOT_C=C(M1,J)+TOT_C
          C_TOT=C_TOT+C(M1,J)
 130  CONTINUE
         F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute the amount adsorbed on the bed
     DO 110 M1=1,NC
      F1=K_F(M1)*AINT*Z/U_F1
      A2=1/DELT+F1
      F(NC+M1)=-C(NC+M1,J)+1/A2*(F1*Q(M1,J)+P(NC+M1,J)/DELT)
 110  CONTINUE
     FLUX1=0.
     FLUX2=0.
     DO 140 M1=1,NC
c compute total heat of adsorption
      FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
c compute total amount adsorbed on the particle
      FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
 140  CONTINUE
     RATE_C1(J)=FLUX2
c compute the gas temperature
     AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
     F(L1)=-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*(C(L1,J-1)+
   . C(L1,J-1))-1.0D0/(DELZ*U_F1)*(-C(L1,J-1)*C(L4,J))-
   . F3*(-C(L2,J))-F4*(-C(L3,J))+P(L1,J)/DELT)
c compute the solid temperature
     F(L2)=-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2*(C(L2,J-1)-
   . 2*C(L2,J)+C(L2,J-1))+F5*(C(L1,J)-C(L2,J))-F6*FLUX1)+P(L2,J)
c compute the wall temperature
     F(L3)=-C(L3,J)+DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+
   . P(L3,J)

C
C------------------------------------------------
C
C INTERIOR ROWS
C computation of C aray for interior grids
    ELSE
     DO 165 M1=1,NC
      PE_N_M(M1)=U_F1*Z/D_L(M1)
      H(M1)=1.D0/DELT+2.D0/PE_N_M(M1)/DELZ**2+1.D0/U_F1/DELZ*
   .  C(L4,J)
 165  CONTINUE
         C_TOT=0.
```

```
c compute the molar concentration
      DO 170 M1=1,NC-1
       F2=ALF*K_F(M1)*AINT*Z/U_F1
       F(M1)=-C(M1,J)+1.D0/H(M1)*(1.0D0/PE_N_M(M1)/DELZ**2*
     . (C(M1,J-1)+C(M1,J+1))-1.0D0/U_F1/DELZ*(-C(M1,J-1)*
     . C(L4,J))-F2*(Q(M1,J)-C(NC+M1,J))+P(M1,J)/DELT)
           C_TOT=C_TOT+C(M1,J)
 170   CONTINUE
         F(NC)=(C(L5,J)-C_TOT*C(L1,J)*R)/(C(L1,J)*R)-C(NC,J)
c compute the amount adsorbed on
      DO 150 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       A2=1/DELT+F1
       F(NC+M1)=-C(NC+M1,J)+1.D0/A2*(F1*Q(M1,J)+P(NC+M1,J)/DELT)
 150   CONTINUE
      FLUX1=0.
      FLUX2=0.
      DO 180 M1=1,NC
c compute total heat of adsorption
       FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
c compute toal amount adsorbed
       FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
 180   CONTINUE
      RATE_C1(J)=FLUX2
c compute gas temperature
      AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
      F(L1)=-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*(C(L1,J-1)+
     . C(L1,J+1))-1.0D0/(DELZ*U_F1)*(-C(L1,J-1)*C(L4,J))-F3*
     . (-C(L2,J))-F4*(-C(L3,J))+P(L1,J)/DELT)
c compute solid temperature
      F(L2)=-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2*(C(L2,J-1)-2*
     . C(L2,J)+C(L2,J+1))+F5*(C(L1,J)-C(L2,J))-F6*FLUX1)+P(L2,J)
c compute wall temperature
      F(L3)=-C(L3,J)+DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+
     . P(L3,J)
     END IF
 210 RETURN
    END

C
C-----------------------------------------------------------------
C
C THIS SUBROUTINE IS BEING CALLED BY DIFFEQ1 WHICH IS ALSO BEING CALLED BY
C SUBROUTINE FUNCT2_Z TO COMPUTE THE VELOCITY AND THE PRESSURE DROP IN THE BED.
C THE EQUATIONS ARE BEING SOLVED BY NEWMAN'S METHOD.
C
      SUBROUTINE FUNCT2(J)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 D_L(4),M_AVE
      COMMON/OLD/ AA(14),SUM(14),COLD(14,301)
      COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,301)
      COMMON/PP_OLD_Z/ RO_P(301),CP_P(301),CON_FPP(301),
     . CON_LPP(301),D_LPP(4,301),H_FP(301),H_TTP(301),H_IIP(301),
     . H_OWPP(301),H_FWPP(301),Q(4,301),VISC_P(301),RATE_C1(301)
      COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4)
      COMMON/PROP_D_Z/DELZ1,DELT1
      COMMON/NCY/NC1
      DATA L1,L4,L5,L6,R1,CONV/4,1,2,3,555.0,51.714752314/
      DATA FAC,GC,M_AVE/2.78450526316,416975040.0,28.0/
C
C
```

```
C MASS TRANSFER FROM THE BULK OF GAS SRTEAM TO THE SURFACE OF ABSORBENT
C
C ESTABLISHED COEFFICIENT MATRIX
C
C THE FIRST ROW
C
c set parameters, total adsorbed, viscocity, density
   NC=NC1
   RATE_C=RATE_C1(J)
   VISC_F=VISC_P(J)
   KK=0
   SUM1=0.
   DO I=1,NC
    D_L(I)=D_LPP(I,J)
    IF(D_L(I).GT.0)THEN
     KK=KK+1
     SUM1=SUM1+D_L(I)
    END IF
   END DO
   D_L_AVE=SUM1/KK
   RO_F=C(L5,J)/(C(L1,J)*R1)
   IF(J.EQ.1)THEN
c compute the velocity for the first grid
    F(L4)=FAC*(C(L5,J)-BC_L5*CONV)/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
   . (C(L4,J)-BC_L4)/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/
   . D_P/D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/
   . D_P/EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c compute the presure without the heat effect
    F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(BC_L5*CONV-
   . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-BC_L5*CONV)/DELZ1+
   . C(L5,J)*(C(L4,J)-BC_L4)/DELZ1+R1*C(L1,J)*ALF*RATE_C
c compute the pressure with heat effect
    F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(BC_L6*CONV-
   . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-BC_L6*CONV)/DELZ1+
   . C(L6,J)*(C(L4,J)-BC_L4)/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
   . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J+1)-2*C(L1,J)+BC_L1)+
   . C(L4,J)*(C(L1,J+1)-BC_L1)/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

C
C-----------------------------------------------------
C the last grid
    ELSE IF(J.EQ.NJ)THEN
c compute the velocity
    F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
   . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
   . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
   . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c compute the pressure without the heat effect
    F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J-1)-
   . 2*C(L5,J)+C(L5,J-1))+C(L5,J)*(C(L5,J)-C(L5,J-1))/DELZ1+
   . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1+R1*C(L1,J)*ALF*RATE_C
c cmpute the presure with the heat effect
    F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(C(L6,J-1)-
   . 2*C(L6,J)+C(L6,J-1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ1+
   . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
   . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J-1))+
   . C(L4,J)*(C(L1,J-1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

C
C-----------------------------------------------------
C
c interior grids

    ELSE
```

```
c compute velocity
     F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
   . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
   . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
   . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c compute pressure without the effect
     F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J-1)-
   . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-C(L5,J-1))/DELZ1+
   . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1+R1*C(L1,J)*ALF*RATE_C
c compute the presure with heat effect
     F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(C(L6,J-1)-
   . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ1+
   . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
   . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J-1))+
   . C(L4,J)*(C(L1,J+1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

    END IF

C-----------------------------------------------------------
 210 RETURN
    END
C
C
    SUBROUTINE DIFFEQ0(C1,P1,NJ1,N1,IND1)
    IMPLICIT REAL*8(A-H,O-Z)
C*********************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C*********************************************************************
C
C    CALLED BY:  MAIN CALLING PROGRAM
C
C*********************************************************************
C
C    SUBROUTINES CALLED:
C         WRTOUT (FOR DATA OUTPUT)
C         BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C         FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C              FOR A VARIABLE)
C
C*********************************************************************
C
C    LIST OF IMPORTANT VARIABLES:
C
C         A     A coefficient described in Newman, Appendix C
C         AA     first, AA is F(WORKC*CU).  Later AA is the
C              value of the derivative df/dc used in Newton's
C              method
C         B     B coefficient described in Newman, Appendix C
C         C     variables to be solved for
C         CD     multiplication factor used in obtaining
C              numerical derivatives
C         COLD   value of C from previous iteration
C         CU     2.0 - CD
C         D     D coefficient described in Newman, Appendix C
C         ERR    convergence criterion
C         F     value of function f(C), calculated in FUNCT
C         G     residual of f(C) calculated with updated C value
C         I     index used for equation number
C         ITCNT   index for iteration number
C         ITPRT   flag used for determining whether intermediate
C              calculations are output; for ITPRT=0,only
C              converged results are output; for
C              ITPRT=1,results of each iteration are printed.
```

```
C          J     index for node number
C          K      index for equation number
C          M      index used in working through nodes used to
C               calculate numerical derivatives
C          MM      used to determine starting node (in relation to
C               J) for estimation of numerical derivatives
C          N      number of equations (no. of variables)
C          NJ     number of node points
C          SAVEC   saved value of C
C          SUM     intermediate value used in calculating G
C          TINIER  criterion used to avoid working with small numbers
C          TINY    criterion used to avoid working with small numbers
C          TNIEST  criterion used to avoid working with small numbers
C          WORKC   saved value of C; modified when C less than Tinier
C          X      X value described in Newman, Appendix C
C          Y      Y value described in Newman, Appendix C
C***********************************************************************
C
C DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C WHATEVER YOO¿ppp
   REAL*8 C1(14,301),P1(14,301)
   COMMON/OLD/ AA(14),SUM(14),COLD(14,301)
   COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
  .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,301)
   COMMON/NUM_OF_ITR/ITCNT1
   DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-20,1.0D-4/
   DATA CU,CD,ITCNT0/1.0001,.9999,15/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
   ITPRT=0
   NJ=NJ1
   N=N1
   IF (ITPRT.GT.0) CALL WRTOUT2
   IF(IND1.EQ.1)THEN
    DO 2 I1=1,N1+2
     DO 2 I2=1,NJ1
      C(I1,I2)=C1(I1,I2)
      P(I1,I2)=P1(I1,I2)
 2   CONTINUE
   ELSE IF(IND1.EQ.2)THEN
    DO 5 I1=1,N1+1
     DO 5 I2=1,NJ1
      C(I1,I2)=C1(I1,I2)
      P(I1,I2)=P1(I1,I2)
 5   CONTINUE
   END IF
C
C LOOP BEGUN FOR ITERATIONS
C
   IF(ITCNT0.GT.7)ITCNT0=ITCNT0-1
   DO 75 ITCNT=1,ITCNT0
C
C COLD ARRAY SET UP
C
     ITCNT1=ITCNT
     DO 10 K=1,N
       DO 10 J=1,NJ
         COLD(K,J)=C(K,J)
```

```
10    CONTINUE
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
      DO 60 J=1,NJ
       IF(IND1.EQ.1)THEN
        CALL FUNCT0(J)
       ELSE
        CALL FUNCT2(J)
       END IF
        DO 15 I=1,N
          SUM(I)=0.0
          G(I)=-F(I)
15       CONTINUE
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
        IF (J.EQ.1) THEN
          MM=0
        ELSE IF (J.LT.NJ) THEN
          MM=-1
        ELSE
          MM=-2
        END IF
        DO 50 M=MM,MM+2
C
C ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
          DO 50 K=1,N
            SAVEC=C(K,J+M)
            WORKC=SAVEC
C
C DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
            IF (ABS(WORKC).LT.TINY) THEN
              IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
              C(K,J+M)=1.2*WORKC
              IF(IND1.EQ.1)THEN
               CALL FUNCT0(J)
              ELSE
               CALL FUNCT2(J)
              END IF
              DO 20 I=1,N
                AA(I)=-F(I)
20             CONTINUE
              C(K,J+M)=1.1*WORKC
              IF(IND1.EQ.1)THEN
               CALL FUNCT0(J)
              ELSE
               CALL FUNCT2(J)
              END IF
              DO 25 I=1,N
                AA(I)=AA(I)+4.0*F(I)
25             CONTINUE
              C(K,J+M)=WORKC
              IF(IND1.EQ.1)THEN
               CALL FUNCT0(J)
              ELSE
               CALL FUNCT2(J)
              END IF
              DO 30 I=1,N
```

```
              AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
30           CONTINUE
C
C  DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
          ELSE
            C(K,J+M)=WORKC*CU
            IF(IND1.EQ.1)THEN
             CALL FUNCT0(J)
            ELSE
             CALL FUNCT2(J)
            END IF
            DO 35 I=1,N
              AA(I)=F(I)
35           CONTINUE
            C(K,J+M)=WORKC*CD
            IF(IND1.EQ.1)THEN
             CALL FUNCT0(J)
            ELSE
             CALL FUNCT2(J)
            END IF
            DO 40 I=1,N
              AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
40           CONTINUE
          ENDIF
          C(K,J+M)=SAVEC
C
C  VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
          DO 45 I=1,N
            SUM(I)=SUM(I)+AA(I)*C(K,J+M)
            IF (M.EQ.-2) Y(I,K)=AA(I)
            IF (M.EQ.-1) A(I,K)=AA(I)
            IF (M.EQ.0) B(I,K)=AA(I)
            IF (M.EQ.1) D(I,K)=AA(I)
            IF (M.EQ.2) X(I,K)=AA(I)
45         CONTINUE
50       CONTINUE
        DO 55 I=1,N
         G(I)=G(I)+SUM(I)
55       CONTINUE
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
        CALL BAND2(J)
60     CONTINUE
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C  ITERATION IS BEGUN
C
      DO 65 K=1,N
        DO 65 J=1,NJ
         IF(DABS(C(K,J)).GT.TNIEST) THEN
           IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
         ENDIF
65     CONTINUE
     GO TO 80
70   IF(ITPRT .GT. 0) CALL WRTOUT2
75   CONTINUE
80   CONTINUE
     DO 95 I1=1,N1
      DO 95 I2=1,NJ1
       C1(I1,I2)=C(I1,I2)
```

```
95  CONTINUE
    RETURN
    END


    SUBROUTINE DIFFEQ(C1,P1,NJ1,N1,IND1,ITCNT0)
    IMPLICIT REAL*8(A-H,O-Z)
C*********************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C*********************************************************************
C
C    CALLED BY:  MAIN CALLING PROGRAM
C
C*********************************************************************
C
C    SUBROUTINES CALLED:
C         WRTOUT (FOR DATA OUTPUT)
C         BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C         FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C              FOR A VARIABLE)
C
C*********************************************************************
C
C    LIST OF IMPORTANT VARIABLES:
C
C         A      A coefficient described in Newman, Appendix C
C         AA     first, AA is F(WORKC*CU).  Later AA is the
C              value of the derivative df/dc used in Newton's
C              method
C         B      B coefficient described in Newman, Appendix C
C         C      variables to be solved for
C         CD     multiplication factor used in obtaining
C              numerical derivatives
C         COLD   value of C from previous iteration
C         CU     2.0 - CD
C         D      D coefficient described in Newman, Appendix C
C         ERR    convergence criterion
C         F      value of function f(C), calculated in FUNCT
C         G      residual of f(C) calculated with updated C value
C         I      index used for equation number
C         ITCNT  index for iteration number
C         ITPRT  flag used for determining whether intermediate
C              calculations are output; for ITPRT=0,only
C              converged results are output; for
C              ITPRT=1,results of each iteration are printed.
C         J      index for node number
C         K      index for equation number
C         M      index used in working through nodes used to
C              calculate numerical derivatives
C         MM     used to determine starting node (in relation to
C              J) for estimation of numerical derivatives
C         N      number of equations (no. of variables)
C         NJ     number of node points
C         SAVEC  saved value of C
C         SUM    intermediate value used in calculating G
C         TINIER criterion used to avoid working with small numbers
C         TINY   criterion used to avoid working with small numbers
C         TNIEST criterion used to avoid working with small numbers
C         WORKC  saved value of C; modified when C less than Tinier
C         X      X value described in Newman, Appendix C
C         Y      Y value described in Newman, Appendix C
C*********************************************************************
```

```
C
C  DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C  IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C  IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C  DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C  NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C  WHATEVER YOO¿ppp
      REAL*8 C1(14,301),P1(14,301)
      COMMON/INDIC/L1,L2,L3,L4,L5,L6
      COMMON/OLD/ AA(14),SUM(14),COLD(14,301)
      COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
     .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,301)
      COMMON/NUM_OF_ITR/ITCNT1
      COMMON/NCY/NC1
      DATA TINY0,TINIER0,TNIEST0,ERR/1.0D-5,1.0D-10,1.0D-15,1.0D-3/
      DATA TINY1,TINIER1,TNIEST1,ERR/1.0D-10,1.0D-15,1.0D-20,1.0D-3/
      DATA TINY2,TINIER2,TNIEST2,ERR/1.0D-1,1.0D-2,1.0D-3,1.0D-3/
      DATA CU,CD/1.0001,.9999/
C
C  ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C  IF ITPRT=1.
C
      ITPRT=0
      NJ=NJ1
      N=N1
      IF (ITPRT.GT.0) CALL WRTOUT2
      IF(IND1.EQ.1)THEN
       DO 2 I1=1,N1+5
        DO 2 I2=1,NJ1
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
 2     CONTINUE
      ELSE IF(IND1.EQ.2)THEN
       DO 5 I1=1,N1+1
        DO 5 I2=1,NJ1
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
 5     CONTINUE
      ELSE IF(IND1.EQ.3)THEN
       ITR=5+NC1*2
       DO 7 I1=1,ITR
        DO 7 I2=1,NJ1
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
 7     CONTINUE
      END IF
C
C  LOOP BEGUN FOR ITERATIONS
C
C     IF(ITCNT0.GT.7)ITCNT0=ITCNT0-1
      DO 75 ITCNT=1,ITCNT0
C
C  COLD ARRAY SET UP
C
        ITCNT1=ITCNT
        DO 10 K=1,N
          DO 10 J=1,NJ
            COLD(K,J)=C(K,J)
10      CONTINUE
C
C  LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C  PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
        DO 60 J=1,NJ
```

```
      IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
       CALL FUNCT1(J)
      ELSE
       CALL FUNCT2(J)
      END IF
       DO 15 I=1,N
         SUM(I)=0.0
         G(I)=-F(I)
15      CONTINUE
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
       IF (J.EQ.1) THEN
         MM=0
       ELSE IF (J.LT.NJ) THEN
         MM=-1
       ELSE
         MM=-2
       END IF
       DO 50 M=MM,MM+2
C
C ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
         DO 50 K=1,N
           SAVEC=C(K,J+M)
           WORKC=SAVEC
C
C DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
           IF(N.EQ.1)THEN
            TINY=TINY0
            TINIER=TINIER0
            TNIEST=TNIEST0
           ELSE IF(N.GE.L1)THEN
            TINY=TINY2
            TINIER=TINIER2
            TNIEST=TNIEST2
           ELSE
            TINY=TINY1
            TINIER=TINIER1
            TNIEST=TNIEST1
           END IF
           IF (ABS(WORKC).LT.TINY) THEN
             IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
             C(K,J+M)=1.2*WORKC
             IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
              CALL FUNCT1(J)
             ELSE
              CALL FUNCT2(J)
             END IF
             DO 20 I=1,N
               AA(I)=-F(I)
20            CONTINUE
             C(K,J+M)=1.1*WORKC
             IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
              CALL FUNCT1(J)
             ELSE
              CALL FUNCT2(J)
             END IF
             DO 25 I=1,N
               AA(I)=AA(I)+4.0*F(I)
25            CONTINUE
```

```
            C(K,J+M)=WORKC
            IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
             CALL FUNCT1(J)
            ELSE
             CALL FUNCT2(J)
            END IF
            DO 30 I=1,N
              AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
30           CONTINUE
C
C DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
          ELSE
            C(K,J+M)=WORKC*CU
            IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
             CALL FUNCT1(J)
            ELSE
             CALL FUNCT2(J)
            END IF
            DO 35 I=1,N
              AA(I)=F(I)
35           CONTINUE
            C(K,J+M)=WORKC*CD
            IF(IND1.EQ.1 .OR. IND1.EQ.3)THEN
             CALL FUNCT1(J)
            ELSE
             CALL FUNCT2(J)
            END IF
            DO 40 I=1,N
              AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
40           CONTINUE
           ENDIF
           C(K,J+M)=SAVEC
C
C VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
          DO 45 I=1,N
            SUM(I)=SUM(I)+AA(I)*C(K,J+M)
            IF (M.EQ.-2) Y(I,K)=AA(I)
            IF (M.EQ.-1) A(I,K)=AA(I)
            IF (M.EQ.0) B(I,K)=AA(I)
            IF (M.EQ.1) D(I,K)=AA(I)
            IF (M.EQ.2) X(I,K)=AA(I)
45          CONTINUE
50      CONTINUE
       DO 55 I=1,N
         G(I)=G(I)+SUM(I)
55      CONTINUE
C
C BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
       CALL BAND2(J)
60     CONTINUE
C
C CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C ITERATION IS BEGUN
C
      DO 65 K=1,N
        DO 65 J=1,NJ
         IF(DABS(C(K,J)).GT.TNIEST) THEN
           IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
         ENDIF
65     CONTINUE
```

```
   GO TO 80
70  IF(ITPRT .GT. 0) CALL WRTOUT2
75  CONTINUE
80  CONTINUE
   DO 95 I1=1,N1
    DO 95 I2=1,NJ1
     C1(I1,I2)=C(I1,I2)
95  CONTINUE
   RETURN
   END

   SUBROUTINE WRTOUT2
C************************************************************************
   IMPLICIT REAL*8(A-H,O-Z)
   COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),
   .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14)
   IF (ITCNT.NE.0) WRITE (*,99)ITCNT
   WRITE (*,100)
   DO 1 K=1,NJ,2
     WRITE(*,101)K,(C(I,K),I=1,N)
1   CONTINUE
99  FORMAT(' ITCNT=',I2)
100  FORMAT('  J       C1            C2            C3            '
  &,' C4            C5            C6'/)
101   FORMAT(1X,I3,6(1PE16.8))
   RETURN
   END
C  BLOCK TRIDIAGONAL MATRIX SUBROUTINE
   SUBROUTINE BAND2(J)
   IMPLICIT REAL*8(A-H,O-Z)
   DIMENSION E(14,14,301)
   COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29),X(14,14),Y(14,14),
   .G(14),N,NJ,ITPRT,ITCNT,F(14)
 101 FORMAT('0DETERM=0 AT J=',I4)
   IF (J.EQ.1) THEN
     NP1=N+1
     DO 2 I=1,N
       D(I,2*N+1)=G(I)
       DO 2 L=1,N
        D(I,L+N)=X(I,L)
 2   CONTINUE
     CALL MATINV2(N,2*N+1,DETERM)
     IF(DETERM.EQ.0) WRITE (2,101) J
     DO 5 K=1,N
       E(K,NP1,1)=D(K,2*N+1)
       DO 5 L=1,N
         E(K,L,1)=-D(K,L)
         X(K,L)=-D(K,L+N)
 5   CONTINUE
     RETURN
    ELSE IF(J.EQ.2)THEN
     DO 7 I=1,N
       DO 7 K=1,N
         DO 7 L=1,N
          D(I,K)=D(I,K)+A(I,L)*X(L,K)
 7   CONTINUE
    ELSE IF(J.EQ.NJ) THEN
     DO 10 I=1,N
       DO 10 L=1,N
         G(I)=G(I)-Y(I,L)*E(L,NP1,J-2)
         DO 10 M=1,N
           A(I,L)=A(I,L)+Y(I,M)*E(M,L,J-2)
 10   CONTINUE
    ENDIF
```

```
   DO 12 I=1,N
     D(I,NP1)=-G(I)
     DO 12 L=1,N
       D(I,NP1)=D(I,NP1)+A(I,L)*E(L,NP1,J-1)
       DO 12 K=1,N
         B(I,K)=B(I,K)+A(I,L)*E(L,K,J-1)
 12 CONTINUE
   CALL MATINV2(N,NP1,DETERM)
   IF(DETERM.EQ.0) WRITE(2,101) J
   DO 15 K=1,N
     DO 15 M=1,NP1
       E(K,M,J)=-D(K,M)
 15 CONTINUE
   IF(J.EQ.NJ)THEN
     DO 17 K=1,N
       C(K,J)=E(K,NP1,J)
 17   CONTINUE
     DO 18 M=NJ-1,1,-1
       DO 18 K=1,N
         C(K,M)=E(K,NP1,M)
         DO 18 L=1,N
           C(K,M)=C(K,M)+E(K,L,M)*C(L,M+1)
 18   CONTINUE
     DO 19 L=1,N
       DO 19 K=1,N
         C(K,1)=C(K,1)+X(K,L)*C(L,3)
 19   CONTINUE
   ENDIF
   RETURN
   END
C
C  MATRIX INVERSION SUBROUTINE
   SUBROUTINE MATINV2(N,M,DETERM)
   IMPLICIT REAL*8(A-H,O-Z)
   DIMENSION ID(14)
   COMMON/BND/A(14,14),B(14,14),C(14,301),D(14,29)
   DETERM=1.0
   DO 1 I=1,N
 1   ID(I)=0
   DO 18 NN=1,N
     BMAX=0.0
     DO 6 I=1,N
       IF(ID(I).EQ.0)THEN
         DO 5 J=1,N
           IF(ID(J).EQ.0)THEN
             IF(DABS(B(I,J)).GT.BMAX) THEN
               BMAX=DABS(B(I,J))
               IROW=I
               JCOL=J
             ENDIF
           ENDIF
 5        CONTINUE
       ENDIF
 6   CONTINUE
     IF(BMAX.EQ.0.0)THEN
       DETERM=0.0
       RETURN
     ENDIF
     ID(JCOL)=1
     IF(JCOL.NE.IROW) THEN
 9      DO 10 J=1,N
         SAVE=B(IROW,J)
         B(IROW,J)=B(JCOL,J)
         B(JCOL,J)=SAVE
```

```
 10      CONTINUE
        DO 11 K=1,M
          SAVE=D(IROW,K)
          D(IROW,K)=D(JCOL,K)
          D(JCOL,K)=SAVE
 11      CONTINUE
      ENDIF
      FF = 1.0/B(JCOL,JCOL)
      DO 13 J=1,N
 13      B(JCOL,J)=B(JCOL,J)*FF
      DO 14 K=1,M
 14      D(JCOL,K)=D(JCOL,K)*FF
      DO 18 I=1,N
        IF(I.NE.JCOL)THEN
 15        FF = B(I,JCOL)
          DO 16 J=1,N
 16          B(I,J) = B(I,J) -FF*B(JCOL,J)
          DO 17 K=1,M
 17          D(I,K) = D(I,K) -FF*D(JCOL,K)
        ENDIF
 18  CONTINUE
    RETURN
    END
C
C-------------------------------------------------------
C
C
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is by Forythe,
c Computer Methods for Mathematical Computation. it is an bisect method with
c quadratic convergence.
c
c B,V,PO arrays are sinle equlibrum constant
      SUBROUTINE IST_Z1(METHOD,T,SS,Q1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/NCY/NC1
      COMMON/INDIC1/IND(4)
      REAL*8 Q1(4),SS(4)
      INTEGER I,NLIM
      EXTERNAL FCN1
      DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
      DATA R/555/
      T_G=T
      NC=NC1
c the partial pressure
      DO I1=1,NC
        Y1(I1)=SS(I1)
          END DO
      DO I=1,NC
        PP(I)=Y1(I)*T_G*R
      END DO
c no mole fration return
      Z=1.0E-32
      IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
        DO I=1,NC
          Q1(I)=0.
        END DO
        RETURN
      END IF
      N1=1
```

```
c set the constant as a function of temperature
      IF(IND(1).EQ.1)THEN
C        V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
       V(N1)=17.0/44/100
C        B(N1)=6864.9000131*EXP(-.019625791466*T)
       B(N1)=5.323235056e-6*T**(-.5)*EXP(13948.544244/1.987/T)
           PO(N1)=.8
       N1=N1+1
      END IF
      IF(IND(2).EQ.1)THEN
       V(N1)=1./69.035
       B(N1)=1.879094E-4*EXP(5467.4817024/T)
       IF(T.LT.610.) THEN
        B(N1)=4.5597278759E-7*EXP(9628.9655743/T)
       ELSE
        B(N1)=5.8089066684E-7*EXP(9115.734593/T)
       END IF
       N1=N1+1
           PO(N1)=1.0
      END IF
      IF(IND(3).EQ.1)THEN
       V(N1)=-1.637879912E-5*T+.00961297026
       IF(T.LE.532)THEN
        B(N1)=3.2694515539E-7*T+4.59988799E-4
       ELSE
        B(N1)=7.90864008E-5*T-4.14400420E-2
       END IF
           PO(N1)=1.0
       N1=N1+1
      END IF
      IF(IND(4).EQ.1)THEN
       V(N1)=-1.637879912E-5*T+.00961297026
       IF(T.LE.532)THEN
        B(N1)=3.2694515539E-7*T+4.59988799E-4
       ELSE
        B(N1)=7.90864008E-5*T-4.14400420E-2
       END IF
           PO(N1)=1.0
      END IF
      IF(NC.EQ.1)THEN
       Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
       RETURN
      ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
       Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
       Q1(1)=0
       RETURN
      ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
       Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
       Q1(2)=0
       RETURN
      END IF
      TERM1=0.
c calculate the equilibruim isthotherm by Langmuir method
      DO I=1,NC
       IF(PP(I).GT.0)THEN
        TERM1=B(I)*PP(I)**PO(I)+TERM1
       END IF
      END DO
      SUM=0.
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
```

```
        SUM=SUM+Q1(I)
       END IF
      END DO
c if the Langmuir method is asked for then returned
      IF(METHOD.EQ.1)THEN
       RETURN
      END IF
c if not, take the result as the first guess for IST theory
      DO I=1,NC
       X1(I)=Q1(I)/SUM
       IF(X1(I).GT.0)THEN
        PI(I)=PP(I)/X1(I)
       ELSE
        PI(I)=0.
       END IF
      END DO
c calculate the spread pressure
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        X1(I)=0
       ELSE
        X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
        IF=I
       END IF
      END DO
      X=X1(IF)
      DO I=1,NC
       IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
        X=X1(I)
       END IF
      END DO
      AX=X
      DELX=AX
      BX=AX
c call Zeroin subroutine to find the rout to the IST equation
      DO I=1,100
       BX=BX+DELX
       FUN=FCN1(BX)
       IF(FUN.LT.0)GO TO 113
      END DO
113   X=ZEROIN(FCN1,AX,BX,TOL)
c rout was found
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        PP(I)=1.0E-32
       END IF
c calculate the fraction in the solid phase
       PART1=PO(I)*X/V(I)
       IF(PART1.GT.73)PART1=73
       PI(I)=(EXP(PART1)-1)/B(I)
       X1(I)=PP(I)/PI(I)
      END DO
      TOT_Q=0
c calculate the total amount adsorbed
      DO I=1,NC
       Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
       TOT_Q=TOT_Q+X1(I)/Q1(I)
      END DO
c calculate the amount adsorbed for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80   RETURN
      END
```

```
c IST function
    REAL FUNCTION FCN1(X)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
    COMMON/NCY/NC1
    NC=NC1
    SUM=0.
    DO I=1,NC
     IF(PP(I).GT.0) THEN
      PART1=PO(I)*X/V(I)
      IF(PART1.GT.73)PART1=73
      SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
     END IF

C      IF(PP(I).LE.0)PP(I)=1.0E-32
C      PART1=PO(I)*X/V(I)
C      IF(PART1.GT.73)PART1=73
C      SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
    END DO
    FCN1=SUM-1
    RETURN
    END
c
c-----------------------------------------------------------
c
c subroutine to find the root of equation by bisect method

     REAL FUNCTION ZEROIN(FCN1,AX,BX,TOL)
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 AX,BX,FCN1,TOL
     REAL*8 A,B,C,D,E,EPS,FA,FB,FC,TOL1,XM,P,Q,R,S
     EPS=1.0
10    EPS=EPS/2.
     TOL1=1.0+EPS
     IF(TOL1.GT.1.0) GO TO 10
c initialization
     A=AX
     B=BX
     FA=FCN1(A)
     FB=FCN1(B)
c begin step
20    C=A
     FC=FA
     D=B-A
     E=D
30    IF(ABS(FC) .GE. ABS(FB)) GO TO 40
     A=B
     B=C
     C=A
     FA=FB
     FB=FC
     FC=FA
c convergence test
40    TOL1=2.*EPS*ABS(B)+.5*TOL
     XM=.5*(C-B)
     IF(ABS(XM).LE.TOL1)GO TO 90
     IF(FB.EQ.0.0) GO TO 90
c is bisection necessary
     IF(ABS(E).LE.TOL1)GO TO 70
     IF(ABS(FA).LE.ABS(FB)) GO TO 70
c is quaratic interpolatation possible
     IF(A.NE.C)GO TO 50
c linear interpolation
     S=FB/FA
```

```
      P=2.0*XM*S
      Q=1.0-S
      GO TO 60
c inverse quadratic interpoltation
50    Q=FA/FC
      R=FB/FC
      S=FB/FA
      P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.0))
      Q=(Q-1.0)*(R-1.0)*(S-1.0)
c adjust signs
60    IF(P.GT.0.0)Q=-Q
      P=ABS(P)
c is interpolation acceptable
      IF((2.0*P).GE.(3.*XM*Q-ABS(TOL1*Q))) GO TO 70
      IF(P.GE.ABS(.5*E*Q)) GO TO 70
      E=D
      D=P/Q
      GO TO 80
c bisection
70    D=XM
      E=D
c complete step
80    A=B
      FA=FB
      IF(ABS(D).GT.TOL1)B=B+D
      IF(ABS(D).LE.TOL1)B=B+SIGN(TOL1,XM)
      FB=FCN(B)
      IF((FB*(FC/ABS(FC))).GT.0)GO TO 20
      GO TO 30
c done
90    ZEROIN=B
      RETURN
      END
C
C-------------------------------------------------------------
C
C
c the second method. this method is faster but the initial guess must be near
c the root of the equation.
c
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is Newton,
c
c B,V,PO arrays are sinle equlibrum constant
      SUBROUTINE IST_Z(METHOD,T,TT,SS,Q1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/NCY/NC1
      COMMON/INDIC1/IND(4)
      REAL*8 Q1(4),SS(4)
      INTEGER I,NLIM
      EXTERNAL FCN,FDER
      DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
      DATA R/555.0/
      T_G=T
      NC=NC1
      DO I1=1,NC
        Y1(I1)=SS(I1)
            END DO
      DO I=1,NC
        PP(I)=Y1(I)*T_G*R
```

```
      END DO
c no mole fration return
      Z=1.0E-32
      IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
       DO I=1,NC
        Q1(I)=0.
       END DO
       RETURN
      END IF
      N1=1
c set the constant as a function of temperature
     IF(IND(1).EQ.1)THEN
c 13x by l, grace
c      point=-1567.1205874+9.00256903*t-.0172347*t*t+1.100806e-5*t*t*t
c      if(pp(n1).gt.point .or. t.gt.627)then
c       V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
c       B(N1)=6864.9000131*EXP(-.019625791466*T)
c       po(n1)=1.0
c      else
c 13x by l_f,grace
C       v(n1)=23.32228/44/100
C       b(n1)=4.9639763e-4*T**(-.5)*exp(9233.33778/1.987/T)
c       po(n1)=-2.9138991288+.017181761178*T-2.8549083257e-5*T*T+
c   .   1.622511757e-8*T*T*T
c      end if
c from Finn data 5A
c      if(pp(n1).lt.1.0) then
C        b(N1)=2.37022397e-6*t**(-.5)*exp(14907.6535/1.987/t)
C        v(N1)=16.6/44/100
C        po(N1)=.80
c Finn by langmuir
c      else
c      v(n1)=(25.97135008-6.084518e-3*T-2.1799516646e-5*T*T)/44/100
c      b(n1)=1.88863393e-5*T**(-.5)*exp(12170.875335/1.987/T)
c      po(n1)=1.
c      end if
C 5A BY GRACE, BY L_F
c      V(N1)=17.0/100/44
c      B(N1)=2.341477E-4*T**(-.5)*EXP(10257.166145/1.987/T)
c      PO(N1)=3.830450111E-4*T**1.188379596
C 5A BY GRACE, BY L
C       V(N1)=(-4.27886889+.087218022*T-9.1010715804E-5*T*T)/44/100
C       B(N1)=9.2533309123E-5*T**(-.5)*EXP(10719.0/1.987/T)
C       PO(N1)=1.
C 5A BY GRACE; BY L_F BETWEEN 0-75 C
      V(N1)=(399.3942-1.938428896*T+3.2540515E-3*T*T-
    . 1.8226211899E-6*T*T*T)/44/100
      B(N1)=35.358072159-.17016733*T+2.7458762E-4*T*T-
    . 1.484116035E-7*T**3
      PO(N1)=-29.3861079+.1538898497*T-2.61225877E-4*T*T+
    . 1.478184694E-7*T**3
c linde 5a
c       v(n1)=.005
c       b(n1)=1.129015193e-5*exp(5055.015089/t)
c       po(n1)=.6
       N1=N1+1
      END IF
c h2o on 5a by grace l,l
      IF(IND(2).EQ.1)THEN
       IF(IND(1).EQ.1)THEN
        if(pp(n1).lt. .4)then
         if(t.le.564.)then
          v(n1)=(39.914452-8.87103e-2*t+6.839502987e-5*t*t)/100/18
         else
```

```
           v(n1)=(873.44464-3.867834937*t+5.80375049e-3*t*t-
     .     2.9346685e-6*t*t*t)/100/18
          end if
          b(n1)=29690.66923-137.837129*T+.214456126*t*t
          po(n1)=1
         else
          b(n1)=48.566639771-2.2620239975e-1*t+3.9101703168e-4*t*t-
     .    2.379098497e-7*t*t*t
          v(n1)=(95.093824733-.21725775358*t+1.4628603e-4*t*t)/100/18
          po(n1)=1
         end if
        ELSE
C h2o on 5a grace by L.F
         b(n1)=24.573259-5.8385278e-2*t+3.473682438e-5*t*t
         v(n1)=(66.62520167-.12299345*t+7.1897316221e-5*t*t)/100/18
         if(t.le.600)then
          po(n1)=.8219916486-3.3519986e-3*t+4.679276479e-6*t*t
         else
          po(n1)=-6.0926707634+2.043004e-2*t-2.924858439e-5*t*t+
     .    1.1746734108e-8*t*t*t
         end if
        END IF
        b(n1)=.2247292664*t**(-.5)*exp(7858.45996/1.987/t)
        v(n1)=(172.54468-.66643613*t+9.758449e-4*t*t-4.986888e-
     . 7*t*t*t)/18/100
        po(n1)=1.
C        V(N1)=22.0/18./100.
C            B(N1)=78.33485-.142551*T
C            PO(N1)=1.
        N1=N1+1
       END IF
       IF(IND(3).EQ.1)THEN
        V(N1)=-1.637879912E-5*T+.00961297026
        IF(T.LE.532)THEN
         B(N1)=3.2694515539E-7*T+4.59988799E-4
        ELSE
         B(N1)=7.90864008E-5*T-4.14400420E-2
        END IF
             PO(N1)=1.0
        N1=N1+1
       END IF
       IF(IND(4).EQ.1)THEN
        V(N1)=-1.637879912E-5*T+.00961297026
        IF(T.LE.532)THEN
         B(N1)=3.2694515539E-7*T+4.59988799E-4
        ELSE
         B(N1)=7.90864008E-5*T-4.14400420E-2
        END IF
             PO(N1)=1.0
       END IF
       IF(NC.EQ.1)THEN
        Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
        RETURN
       ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
        Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
        Q1(1)=0
        RETURN
       ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
        Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
        Q1(2)=0
        RETURN
       ELSE IF(NC.EQ.3)THEN
        if(pp(1).le.0 .and. pp(2).le.0)then
         Q1(3)=V(3)*B(3)*PP(3)**PO(3)/(1+B(3)*PP(3)**PO(3))
```

```
        Q1(1)=0
        Q1(2)=0
        return
       else if(pp(1).le.0 .and. pp(3).le.0)then
        Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
        Q1(1)=0
        Q1(3)=0
        return
       else if(pp(2).le.0 .and. pp(3).le.0)then
        Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
        Q1(2)=0
        Q1(3)=0
        return
       end if
      END IF
      TERM1=0.
c calculate the equilibruim by Langmuir isotherm
      DO I=1,NC
       IF(PP(I).GT.0)THEN
        TERM1=B(I)*PP(I)**PO(I)+TERM1
       END IF
      END DO
      SUM=0.
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
        SUM=SUM+Q1(I)
       END IF
      END DO
c if the Langmuir method is asked foR, then return. If not use it as first guess
c for IST theory
      IF(METHOD.EQ.1)THEN
       RETURN
      END IF
      DO I=1,NC
       X1(I)=Q1(I)/SUM
       IF(X1(I).GT.0) THEN
        PI(I)=PP(I)/X1(I)
       ELSE
        PI(I)=0.
       END IF
      END DO
c set the initial guess for the spreading pressure
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        X1(I)=0
       ELSE
        X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
        IF=I
       END IF
      END DO
      X=X1(IF)
      DO I=2,NC
       IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
        X=X1(I)
       END IF
      END DO
c call newton method to the rout to IST equation
      IF(X .LE. 1.0E-17) GO TO 80
      CALL NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
c rout was found, calculate the fraction in the solid phase
      DO I=1,NC
```

```
       IF(PP(I).LE.0) THEN
        X1(I)=0.
       ELSE
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        PI(I)=(EXP(PART1)-1)/B(I)
        X1(I)=PP(I)/PI(I)
       END IF
      END DO
c calculate the total amound in the solid phase
      TOT_Q=0
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
        TOT_Q=TOT_Q+X1(I)/Q1(I)
       END IF
      END DO
c calculate the amount in equilibruim with the gas phase for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80    RETURN
      END
c
c IST final equation
c
      REAL FUNCTION FCN(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
       END IF
      END DO
      FCN=SUM-1
      RETURN
      END
c
c IST final derivitive
c
      REAL FUNCTION FDER(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART3=PO(I)*X/V(I)
        IF(PART3.LT.35)THEN
         PART1=-PP(I)/B(I)/V(I)*EXP(PART3)
         PART2=((EXP(PART3)-1)/B(I))**((1+PO(I))/PO(I))
         SUM=SUM+PART1/PART2
        ELSE IF(PART3/PO(I).LT.73)THEN
         SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .   EXP(-PART3/PO(I))
        ELSE
```

```
          PART3=73
          SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .    EXP(-PART3)
         END IF
        END IF
       END DO
       FDER=SUM
       RETURN
       END

c
c------------------------------------------------------------
c this subroutine finds the rout to a nonlinear equation using the Newton
c method
       SUBROUTINE NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,J)
       IMPLICIT REAL*8(A-H,O-Z)
       INTEGER NLIM,I,J
       FX=FCN(X)
       X1=X
       DO J=1,NLIM
        DELX=FX/FDER(X)
        X=X-DELX
        FX=FCN(X)
        IF (ABS(X-X1)/X.LE.XTOL)THEN
         RETURN
        END IF
        IF(FX.NE.0)THEN
         IF(ABS(FX-FX1)/FX.LE.FTOL) THEN
          RETURN
         END IF
        END IF
        X1=X
        FX1=FX
       END DO
       I=-1
       PRINT 200, NLIM,X,FX
 200   FORMAT(/'TOLERANCE NOT MET  ',I4,' ITERATIONS X= ',
     2 E12.5,' F(X)=',E12.5)
       RETURN
       END

       FUNCTION CPGAS(TEM,Y)
C THIS FUNCTION CALCULATES THE HEAT CAPACITIES OF CMRIER GAS IN BTU/#MOLE/R
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 A(4),B(4),C(4),D(4),Y(4)
       COMMON/GAS/INERT,NCOMP
       COMMON/INDIC1/IND(4)
       COMMON/NCY/NC1
C CO2
       DATA A/36.11,33.46,29.,29.1/
C H2O
       DATA B/4.233E-2,.6880E-2,.2199E-2,1.158E-2/
C N2
       DATA C/-2.887E-5,.7604E-5,.5723E-5,-.6076E-5/
C O2
       DATA D/7.464E-9,-3.593E-9,-2.871E-9,1.311E-9/
       NC=NC1
       CPGAS=0.
       YO=0.
       N1=1
       T=TEM/1.8-273.15
       IF (INERT.EQ.1) THEN
        DO I=1,4
         IF(IND(I).EQ.1)THEN
```

```
        CP=A(I)+B(I)*T+C(I)*T*T+D(I)*T*T*T
        CPGAS=CP*Y(N1)/4.18669+CPGAS
        YO=Y(N1)+YO
        N1=N1+1
       END IF
      END DO
      CPGAS=CPGAS+(A(3)+B(3)*T+C(3)*T*T+D(3)*T*T*T)*(1-YO)/4.18669
     ELSE
      CPGAS=4.97
     ENDIF
     RETURN
     END

     FUNCTION CPS(TEMP)
     IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES TNE HEAT CAPACITY
C BTU/LB/R.
     CPS=0.2
     RETURN
     END

     REAL FUNCTION KGAS (TEMP)
C THIS FUNCTION CALCULATES THE THERMAL CONDUCTIVITY OF CARRIER GAS ASSUMING
C A LINEAR FUNCTION OF TEMPERATURE AND BASED ON VALUES AT 492 AND 672 R.
C BTU/MIN/FT/R
     IMPLICIT REAL*8(A-H,O-Z)
     COMMON/GAS/INERT,NCOMP
     IF (INERT.EQ.1) THEN
      KGAS=3.88E-7*TEMP+0.4052E-4
     ELSE
      KGAS=1.667E-6*TEMP+6.1E-4
     ENDIF
     RETURN
     END

     FUNCTION RE (G,TEMP)
C TNIS FUNCTION CALCULATES THE PARICLE REYNOLDS NUMBER ASSUMINC THAT
C THE MOLECULAR WEIGHT OF TNE GAS IS EQUAL TO THAT OF CARRIER GAS.
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 LEN,MW
     COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
     COMMON/GAS/INERT,NCOMP
     DATA IFLAG/0/
     IF (INERT.EQ.1) MW=28.0
     IF (INERT.EQ.2) MW=4.
     IF (IFLAG.EQ.0)THEN
      CONST=2.*RA
      IFLAG=1
     ENDIF
     RE=CONST*MW*ABS(G)/VIS(TEMP)
     RETURN
     END

     FUNCTION VIS(TEMP)
     IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE VISCOSITY OF CARRIER GAS AS A LINEAR FUNCTION
C OF TEMPERATURE (LB/MIN/FT).
     COMMON/GAS/INERT,NCOMP
     IF (INERT.EQ.1) THEN
C       VIS=1.0E-6*TEMP+1.65E-4
      VIS=-.0102007812+5.384663937E-5*TEMP-8.7973727E-8*TEMP*TEMP+
    2      4.811387495E-11*TEMP*TEMP*TEMP
     ELSE
      VIS=0.9444E-6*TEMP+2.863E-4
```

```
      ENDIF
      RETURN
      END

      FUNCTION RHOG(TR,P)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
C THIS FUNCTION CALCULATES THE MOLAR DENSITY OF AN IDEAL GAS IN
C LBMOLES/CV FS.
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA R/555./
      RHOG=P/R/TR
      RETURN
      END

      FUNCTION EFFD(ICOMP,G,TEMP,RHO,PT,YO)
C THIS FUNCTION CALCVLATES TNE EFFECTIVE AXIAL DIFFUSIVITIES IN A PACXED BED
C USING THE EDWARDS AND RICHARDSON CORROLATION. (FT*FT/MIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA1,LAMBDA2,LEN,YO(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA IFLAG/0/,LAMBDA1/0.73/
      IF(IFLAG.EQ.0)THEN
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF
      DIF=DIFF(ICOMP,TEMP,PT,YO)
      EFFD=LAMBDA1*DIF+CONST*ABS(G)*LAMBDA2(G,DIF,RHO)/RHO
      RETURN
      END

      FUNCTION EFFK(G,T,CP,RHO,P,YO)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE EFFECTIVE AXIAL THERMAL CONDUCTIVITY OF A
C PACKED BED IN BTU/FT/MIN/F.
      COMMON/NCY/NC1
      REAL*8 YO(4)
      NC=NC1
      EFFK=0.
      KK=0
      DO I1=1,NC
       SUM=CP*RHO*EFFD(I1,G,T,RHO,P,YO)
       IF(SUM.GT.0)KK=KK+1
       EFFK=SUM+EFFK
      END DO
       EFFK=EFFK/KK
      RETURN
      END

      REAL FUNCTION LAMBDA2(G,DIF,RHO)
C THIS FUNCTION CALCULATES THE DIMENSTONLESS PARAMETER OF THE EDWARDS AND
C RICHARDS CORROLATION MODIFIED FOR SMALL DIAMETER PARTICLES.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      DATA IFLAG/0/
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      IF(IFLAG.EQ.0)THEN
       IF(RA.GT.0.00492)THEN
         PE=2.
       ELSE
         PE=406.4*RA
       ENDIF
       CONST=2.*RA/EPSEX
       IFLAG=1
```

```
      ENDIF

      LAMBDA2=1./PE/(1.+9.5*DIF/CONST/ABS(G)*RHO)
      RETURN
      END


      FUNCTION DIFF(ICOMP,TEM,P1,YO)
C  THIS FUNCTION CALCULATES THE MOLECULAR  DIFFUSION COEFFICIENTS BY CORRECTING
C  FOR PRESSURE AND TEMPERATURE (FT*FT/MIN).
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,YO(4),V(4),V1(4),MW1(4),MW(4),DIF(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
      COMMON/INDIC1/IND(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA V1/26.9,12.7,17.9,16.6/
      DATA MW1/44.01,18.016,28.013,31.999/
      NC=NC1
      K=ICOMP
      P=P1/760.
      T=TEM/1.8
      N1=1
      DO I1=1,4
       IF(IND(I1).EQ.1)THEN
         MW(N1)=MW1(I1)
         V(N1)=V1(I1)
         N1=N1+1
       END IF
      END DO
      DO I=1,NC
       IF(NC.EQ.1) THEN
         MW(I)=MW(K)
         V(I)=V(K)
       END IF
       PART1=((MW(K)+MW(I))/MW(K)/MW(I))**.5
       PART2=(V(K)**(1./3)+V(I)**(1./3))**2
       DIF(I)=PART1/PART2*T**1.75/P*1.0E-3
      END DO
      IF(NC.EQ.1)THEN
       DIFF=DIF(1)*3.8745
       RETURN
      END IF
      IF(NC.EQ.2)THEN
       IF(K.EQ.1)DIFF=DIF(2)*3.8745
       IF(K.EQ.2)DIFF=DIF(1)*3.8745
       RETURN
      END IF
      SUM=0.
      DO I=1,NC
       IF(I.NE.K)THEN
         SUM=SUM+YO(I)/DIF(I)
       END IF
      END DO
      if(sum.ne.0) then
       DIFF=(1-YO(K))/SUM*3.8745
      else
       diff=1
      endif
      RETURN
      END
C
C
      FUNCTION HFILM(MW,TEMP,REY,CP)
```

```
C THIS FUNCTION CALCULATES THE FLUID FILM HEAT TRANSFER COEFFICIENT
C IN BTU/SQ FT/MIN/R USING THE CORRELATION OF PETROY AND THODOS.
   IMPLICIT REAL*8(A-H,O-Z)
   REAL*8 ID,KGAS,KG,LEN,MW
   COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
   COMMON/GAS/INERT,NCOMP
   KG=KGAS(TEMP)
   PR=CP/MW*VIS(TEMP)/KG
   HFILM=0.357/EPSEX*REY**0.64*PR**0.33*KG*0.5/RA
   RETURN
   END

C    FUNCTION DHCAR(TEMP,P)
C THIS SUBROUTINE CALCULATES HEAT OF ADSORPTION OF CARRIER GAS ONTO THE
C ACTIVATED CARBON TIMES THE TEMPERATURE DERIVATIVE OF THE SOLID PHASE
C CARRIER GAS CONCENTRATION AT THE BED PRESSURE (BTU/# SOLID/R).
C    IMPLICIT REAL*8(A-H,O-Z)
C    REAL*8 LEN
C    COMMON/GAS/INERT,NCOMP
C    COMMON/PRIME/GN2,RA,RAV,EPSEX,ALF,RHOS,ALPHA1
C    DATA R/1.9872/,A/1.28E-9/,B/3151./
C
C    IF(INERT.EQ.1) THEN
C      DH=R*B
C      DHN2=-DH*A*B*EXP(B/TEMP)*P/TEMP/TEMP
C    ELSE
C      DHN2=0.
C    ENDIF
C    RETURN
C    END

    FUNCTION DHADS(ICOMP,T)
C THIS FUNCTION CALCULATES THE ISOTERIC HEAT OF ADSORPTION. (BTU/#MOLE)
    IMPLICIT REAL*8(A-H,O-Z)
    REAL*8 LEN
    COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
    IF(ICOMP.EQ.2) THEN
     DHADS=20400.0
    ELSE
     DHADS=20000.0
    END IF
    RETURN
    END
C
    REAL*8 FUNCTION TEMPIN(TIME)
    IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES CURRENT INLET TEMPERATURE
C BASED ON THE TIME FOR A TEMPERATURE CHARACTERIZATION
C COMPARSION RUN
     IF(TIME.LT.0.2) THEN
         T0 = 71.776
         T1 = 13.611
         T2 = -25.775
         T3 = -7529.2
         T4 = 84821.0
         T5 = -1.9364E+5
     ELSE
         T0 = 74.358
         T1 = 154.04
         T2 = -77.999
         T3 = 20.167
         T4 = -2.5737
         T5 = .12789
     ENDIF
```

```
X = TIME
TEMPIN = T0+T1*X+T2*X**2.+T3*X**3.+T4*X**4.+T5*X**5.
TEMPIN = TEMPIN+460.
RETURN
END
```

APPENDIX E

2DMOL FORTRAN CODE

```
C this is the main routin. the initial values of matix S and parameters are
C called by "INITIAL_Z" subroutine. the main routine then calls the "FUNCT1"
C subroutine to solve the discretized partial diffential equations (pde's).
C after convergance, the routine calls the "DIFFEQ1" subroutine, which is
C the maine routine for solving the momentum and the pressure equations,
C if it is desire to solve these two equations as the time progresses.
C
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 C2(14,201,101),C1(14,201),P1(14,201),RP(101),RP1(101),
    . Y1(33),Y2(33),Y3(33)
     REAL*4 QP(14,201,101),C_AVE(2,101),P_TIME,aa,bb,cc,dd
     INTEGER IOUT(5)
     CHARACTER*5 COUT(3)
     CHARACTER*5 LABEL(4)
     CHARACTER*8 DATE
     CHARACTER*16 RADFILE,AVGFILE,CTLFILE
     COMMON/DATE/DATE,ISAT
     COMMON/NEW_GRID/DELR0(101),ALF1(101),NGR1,NGR2
     COMMON/BND3/S(14,201,101),P(14,201,101)
     COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT,NJ1
     COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
     COMMON/INDIC/L1,L2,L3,L4,L5,L6,LL,M3,II,JJ,IVEL_1,MT
     COMMON/PROP_WAL/CP_WA,RO_WA,ERROR0,ERROR1
     COMMON/INDIC1/IND(4)
     DATA IVEL_1,ICU_SIM/0,1/
     COMMON/RADIAL/RP

          WRITE(*,*)'DESIGNATE A FILE DATE FOR THE OUTPUT FILES'

    READ*,DATE
    RADFILE = DATE // '_RAD.OUT'
    AVGFILE = DATE // '_AVG.OUT'
    CTLFILE = DATE // '_CTL.OUT'
     OPEN (50,FILE=RADFILE,STATUS='NEW',RECL=32766)
     OPEN (60,FILE=AVGFILE,STATUS='NEW',RECL=32766)
     OPEN (70,FILE=CTLFILE,STATUS='NEW',RECL=32766)
C      DATA IOUT/5,25,50,75,101/
     DATA IOUT/5,15,25,35,51/
     DATA COUT/'pCO2@',' pN2@','Tgas@'/

C WRITE HEADER DATA TO ACSII FILE

     WRITE(50,48)((COUT(I),IOUT(J),J=1,5),I=1,3)
     WRITE(60,47)((COUT(I),IOUT(J),J=1,5),I=1,3)
     WRITE(70,47)((COUT(I),IOUT(J),J=1,5),I=1,3)

C
C       INITIALIZE THE C ARRAY WITH THE INITIAL GUESSES OF THE SOLUTION
C
      CALL INITIAL_Z
      T_TIME=0.
      TIME=0.
C strat marching through time, LL is the number of time steps
      IF(IND(2).EQ.0 .OR. ISAT.EQ.1)THEN
       IND0=0
       NEQ=2*NC+3
       NA=NEQ
          ELSE
       IND0=1
       NEQ=2*NC
```

```
       NA=NEQ+3
      END IF
      NEQ1=2*NC+5
      IVEL=1
      NJ_OR_1=1
      LL2=15
      DO LL=1,1800
       IF(LL2.GT.5)THEN
        LL2=LL2-1
       END IF
       DO 20 LL1=1,LL2
        DO I1=1,NEQ1
                 DO I2=1,NJ1
                  DO I3=1,NJR
                   C2(I1,I2,I3)=S(I1,I2,I3)
                  END DO
                 END DO
            END DO
        DO 5 I1=1,NJ1
         IF(IVEL.EQ.1)THEN
          CALL VEL_POR(I1,NJ_OR_1,IVEL_1)
          IVEL=2
         END IF
          DO  I2=1,NA
           DO  I3=1,NJR
            C1(I2,I3)=S(I2,I1,I3)
            P1(I2,I3)=P(I2,I1,I3)
           END DO
                  END DO
         JJ=I1
         MT=1
         IF(LL1.EQ.1 .AND. LL2.EQ.1)THEN
                  ITCNT0=13
                 ELSE IF(LL2.EQ.1)THEN
                  ITCNT0=7
                 ELSE
                  ITCNT0=5
                 END IF
         IF(IND0.EQ.1)THEN
          CALL DIFFEQ2(C1,P1,NEQ,NJR,1,ITCNT0)
         ELSE
          CALL DIFFEQ1(C1,P1,NEQ,NJR,1,itcnt0)
         END IF
         DO M1=1,NEQ
          DO M2=1,NJR
           S(M1,I1,M2)=C1(M1,M2)
          END DO
         END DO
5        CONTINUE
         DO 10 I1=1,NC
          DO 10 I2=1,NJ1
           DO 10 I3=1,NJR
            IF(S(I1,I2,I3).GE.1.0E-15)THEN
             IF(ABS(S(I1,I2,I3)-C2(I1,I2,I3))/
              S(I1,I2,I3).GT.1.0E-2)THEN
              GO TO 25
             END IF
            END IF
10        CONTINUE
         IF(LL1.GE.2)GO TO 205
25       IF(IND0.EQ.1)THEN
         DO 35 LLL=1,5
          DO 15 I1=1,NJ1
           JJ=I1
```

```
           L13=2*NC+1
           DO M1=1,3
            DO M2=1,NJR
             C1(M1,M2)=S(L13,I1,M2)
             P1(M1,M2)=P(L13,I1,M2)
            END DO
            L13=L13+1
           END DO
           MT=0
           CALL DIFFEQ2(C1,P1,3,NJR,2,4)
           L13=2*NC+1
           L31=L13
           DO M1=1,3
            DO M2=1,NJR
             S(L13,I1,M2)=C1(M1,M2)
            END DO
            L13=L13+1
           END DO
 15        CONTINUE
          DO 17 I1=L31,L31+3
           DO 17 I2=1,NJ1
            DO 17 I3=1,NJR
             IF(S(I1,I2,I3).GE.1.0E-15)THEN
              IF(ABS(S(I1,I2,I3)-C2(I1,I2,I3))/
               S(I1,I2,I3).GT.1.0E-2)THEN
               GO TO 35
              END IF
             END IF
 17        CONTINUE
          IF(LLL.GE.2)GO TO 20
 35       CONTINUE
        END IF
 20     CONTINUE
c if M3=0 the pressure and velocity will be computed once, which is the best
c choice since the change of these two vaiables are infintidimal in Z
c direction
205      M3=0
        IF(M3.EQ.0)GO TO 433
c change the name of gas temperature, velocity, and pressure for INE DIFFdiffeq1 subr
        DO I2=1,NJ1
         C1(1,I2)=S(L4,I2,1)
         P1(1,I2)=P(L4,I2,1)
         C1(2,I2)=S(L5,I2,1)
         P1(2,I2)=P(L5,I2,1)
         C1(3,I2)=S(L6,I2,1)
         P1(3,I2)=P(L6,I2,1)
         C1(4,I2)=S(L1,I2,1)
         P1(4,I2)=P(L1,I2,1)
        END DO
c call DIFFEQ1 to solve the velocity and the pressure equations
        CALL DIFFEQ1(C1,P1,3,NJ1,2,5)
c rename the velocity and pressure variables
        DO I2=1,NJ1
         DO I1=1,NJR
          P(L5,I2,I1)=S(L5,I2,I1)
          S(L5,I2,I1)=C1(2,I2)
          P(L6,I2,I1)=S(L6,I2,I1)
          S(L6,I2,I1)=C1(3,I2)
         END DO
          P(L4,I2,1)=S(L4,I2,1)
          S(L4,I2,1)=C1(1,I2)
        END DO
c comulating time step
 433     T_TIME=T_TIME+DELT*60.
```

```
      TIME=TIME+DELT
c increase the time step after some initial time steps, since the equations
c are more stable
      IF(LL.EQ.25)DELT=DELT*2
C       IF(LL.EQ.50)DELT=DELT*2
c convert the varibale for printing the data
      DO 74 M1=1,NJ1
       DO 74 M2=1,NJR
        DO I1=1,NC
         QP(I1,M1,M2)=S(I1,M1,M2)*10.73D0*S(L1,M1,M2)*760/14.696
        END DO
        DO I1=NC+1,2*NC
         QP(I1,M1,M2)=S(I1,M1,M2)*10.73D0*S(L2,M1,M2)*760/14.696/75
        END DO
        QP(L1,M1,M2)=S(L1,M1,M2)-460.
        QP(L2,M1,M2)=S(L2,M1,M2)-460.
        QP(L3,M1,M2)=S(L3,M1,M2)-460.
        QP(L4,M1,M2)=S(L4,M1,M2)
        QP(L5,M1,M2)=S(L5,M1,M2)
        QP(L6,M1,M2)=S(L6,M1,M2)
 74    CONTINUE
c compute the inverse of bed radios, the RP varibles are calculted in FUNCT1
   IF(LL.EQ.1)THEN
    DO K1=1,NJR
     IF(K1.NE.1)THEN
      RP1(K1)=1./RP(K1)
     ELSE
      RP1(1)=0.
     END IF
    END DO
   END IF
c the average concentration is being computed by numerical integration for
c every "ISAMP" time steps
   ISAMP=10
   IF(LL.LT.130)ISAMP=1
   IF((LL/ISAMP)*ISAMP.EQ.LL) THEN
c the averge is taken at 5 points in the axial direction, KK is the grid point
c in axial direction, k1 is comonent number
    N_AXIAL=5
    DO K1=1,NC
C      KK=10
     KK=5
     DO K2=1,N_AXIAL
c the integration is done numerically by cubic spline; or by Simpson method
      IF(ICU_SIM.EQ.1)THEN
       DO K3=1,NJR
        Y1(K3)=S(K1,KK,K3)*S(L1,KK,K3)*555.*RP1(K3)
       END DO
       CALL CUBSPL(RP1,Y1,NJR,1,C_AVE(K1,KK))
       C_AVE(K1,KK)=C_AVE(K1,KK)*2/RP1(NJR)/RP1(NJR)
          ELSE
       DO K3=1,5
        Y2(K3)=S(K1,KK,K3)*S(L1,KK,K3)*555.*RP1(K3)
       END DO
       CALL SIMPS(Y2,5,DELR0(1),RESULT1)
       PART1=RESULT1*2
       DO K3=5,NJR
        Y3(K3-4)=S(K1,KK,K3)*S(L1,KK,K3)*555.*RP1(K3)
       END DO
       CALL SIMPS(Y3,NH,DELR0(NJR),RESULT2)
       PART2=RESULT2*2
       C_AVE(K1,KK)=(PART1+PART2)/RP1(NJR)/RP1(NJR)
      END IF
      IF(K2.EQ.1)THEN
```

```
C         KK=25
         KK=15
        ELSE IF(K2.EQ.4)THEN
C         KK=101
         KK=51
        ELSE
C         KK=KK+25
         KK=KK+10
        END IF
       END DO
      END DO
c printing the variables
      P_TIME=T_TIME
      IF((LL/(ISAMP*4))*ISAMP*4.EQ.LL.OR.LL.EQ.1) THEN
       WRITE(6,49)
       WRITE(6,41)'          ',(' GRID ',IOUT(I),'  ',I=1,5)
      ENDIF
      AA=S(1,25,1)*S(L1,25,1)*555
      BB=S(1,51,1)*S(L1,51,1)*555
      cc=S(2,25,1)*S(L1,25,1)*555
      dd=S(2,51,1)*S(L1,51,1)*555
      WRITE(*,*)AA,BB,cc,dd
      WRITE(6,49)
      WRITE(6,40)'**** TIME = ',P_TIME,'LOOPS = ',LL1,'  ***'
      WRITE(6,43)'PPCO2    ',(C_AVE(1,IOUT(I)),I=1,5)
      WRITE(6,44)'PPN2     ',(C_AVE(2,IOUT(I)),I=1,5)
      WRITE(6,44)'GAS TEMPS ',(QP(L1,IOUT(I),1),I=1,5)

      IF((LL/(ISAMP*10))*ISAMP*10.EQ.LL) THEN
       WRITE(50,46)(P_TIME,RP1(I),
     & ((QP(J,IOUT(K),I),K=1,5),J=1,3),I=1,33,2)
      ENDIF
      WRITE(60,45)P_TIME,((C_AVE(I,IOUT(J)),J=1,5),I=1,2),
     &  (QP(L1,IOUT(I),1),I=1,5)
      WRITE(70,45)P_TIME,((QP(J,IOUT(I),1),I=1,5),J=1,2),
     &  (QP(L1,IOUT(I),1),I=1,5)

      ENDIF
          DO I1=1,NEQ+2
           DO I2=1,NJ1
            DO I3=1,NJR
             P(I1,I2,I3)=S(I1,I2,I3)
            END DO
           END DO
          END DO
        END DO

 40    FORMAT(1X,A12,G10.2,32X,A8,I4,A5,/)
 41    FORMAT(1X,A12,5(A7,I3,A2))
 42    FORMAT(1X,6(A12))
 43    FORMAT(1X,A10,5(2X,G10.2))
 44    FORMAT(1X,A10,5(2X,F10.2))
 45    FORMAT(1X,16(E15.5,','))
 46    FORMAT(1X,17(E15.5,','))
 47    FORMAT(1X,'TIME',15(',',A5,I3))
 48    FORMAT(1X,'TIME,R INCHES',15(',',A5,I3))
 49    FORMAT(1X,70('-'))
       STOP
       END
C
C this subroutine is being called by maine routine once to get the intial
c values and the neccessary parameters
c
      SUBROUTINE INITIAL_Z
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K_F(4),K_FO(4),MOL(4),M_AVE,M_W(4),BC_1(4),MW,Q1(4),
     . SS(4),Q2(4)
      CHARACTER*8 DATE
      CHARACTER*14 INFILEA,INFILEB
      COMMON/DATE/DATE,ISAT
      COMMON/BND3/S(14,201,101),P(14,201,101)
      COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT,NJ1
      COMMON/PROP_B_Z/REC,VOID_B,D_I,D_E,S_B,ALF,GC,Z,M_W
      COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1,HEAT(4)
      COMMON/PROP_WAL/CP_WA,RO_WA,ERROR0,ERROR1
      COMMON/INIT_Z/RO_FO,U_FO,U_F1,CP_FO,T_FO,TO,C_FO(4),K_F,P_C(4)
      COMMON/MISC_Z/R,G_F,PI,P_TOT,BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_1
      COMMON/INDIC/L1,L2,L3,L4,L5,L6,LL
      COMMON/INDIC1/IND(4)
      COMMON/GAS/INERT,NCOMP,MW
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
C
C INITIALIZATION
C
C  SELECT FILE CONTAINING INPUT SET A
     INFILEA = DATE // '_A.DAT'
          OPEN (UNIT=40,FILE=INFILEA,STATUS='OLD')

C  SELECT FILE CONTAINING INPUT SET B
     INFILEB = DATE // '_B.DAT'
          OPEN (UNIT=41,FILE=INFILEB,STATUS='OLD')

C Initialization


     READ(UNIT=40,FMT=*)END,DELT,NJ,NJR,DELZ,TAMB,G_F,TO,NCOMP,INERT,
     . P_TOT,P_C(1),P_C(2),P_C(3),P_C(4),Z,D_E,D_I,RO_WA,EPSEX,
     . AINT,CP_S,RO_S,M_W(1),M_W(2),M_W(3),M_W(4),HEAT(1),HEAT(2),
     . HEAT(3),HEAT(4),K_F(1),K_F(2),K_F(3),K_F(4),ERROR0,ERROR1
     READ(UNIT=41,FMT=*)S_B,CP_WA,RA,RAV,RHOS,ALPHA1,R_P,D_P,R,PI,
     . UINS,HWALL,NC,T_FO,GC,CON_S1,NP,PT,LEN,VOID_B,IND(1),IND(2),
     . IND(3),IND(4),IDES1,ISAT
c the number of component
      NJ1=NJ
      NC1=NC
c number of equations
      N=NC*2+3
c number of grids
      NJ2=NJR
c number of grids in radial directions, variable grids were used for efficincy
      A1=21
      A2=104
c 90 percent of the radious was used for the first 49 grids and 10 percent
c for 71 remainder
      DELR1= .93*D_I/2/(A1-1)
      DELR2= .07*D_I/2/(A2)
c volume of solid/volum of porosity
      ALF=(1.0D0-VOID_B)/VOID_B
      N1=1
     IND3=0
     IND4=0
     DO I1=1,4
      IF(IND(I1).EQ.1)THEN
        M_W(N1)=M_W(I1)
        P_C(N1)=P_C(I1)
        HEAT(N1)=HEAT(I1)
```

```
        K_F(N1)=K_F(I1)
        IF(I1.EQ.3)THEN
         IDES=N1
             IND3=N1
        ELSE IF(I1.EQ.4)THEN
         IND4=N1
        END IF
        N1=N1+1
       END IF
      END DO
      SUM_Y=0
c find the partial pressures and average molecular weight
       DO I=1,NC
        Y_F=P_C(I)/P_TOT
        SUM_Y=SUM_Y+Y_F
        M_AVE=M_AVE+Y_F*M_W(I)
       END DO
       M_AVE=M_AVE+(1-SUM_Y)*28.
       MW=M_AVE
c L1 is the gas temp, L2 is the solid temp, L3 is the wall temp, L4 is the
c velocity, L5 is the total pressure, and L6 is the total pressure witout
c thermal effect
c
       L1=2*NC+1
       L2=L1+1
       L3=L2+1
       L4=L3+1
       L5=L4+1
       L6=L5+1
C
C BOUNDRY CONDITIONS
C
       DO I1=1,NC
        BC_1(I1)=P_C(I1)/555.0/TO
       END DO
       BC_L1=TO
       BC_L2=TO
       BC_L3=TO
       BC_L4=G_F/D_I/D_I/PI*4./VOID_B
       BC_L5=P_TOT
       BC_L6=BC_L5
C
C INITIALIZATION OF VARIABLES
C
       DO I=1,NJ
        DO J=1,NJR
         P(L1,I,J)=BC_L1
         P(L2,I,J)=BC_L2
         P(L3,I,J)=BC_L3
         P(L4,I,J)=BC_L4
         P(L5,I,J)=BC_L5
         P(L6,I,J)=BC_L6
         S(L1,I,J)=BC_L1
         S(L2,I,J)=BC_L2
         S(L3,I,J)=BC_L3
         S(L4,I,J)=BC_L4
         S(L5,I,J)=BC_L5
         S(L6,I,J)=BC_L6
        END DO
       END DO
       DO I1=1,NC
         DO I2=1,NJ
              DO I3=1,NJR
           P(I1,I2,I3)=0.
```

```
    S(I1,I2,I3)=0.
   END DO
  END DO
END DO
DO I=1,NC
  DO J=1,NJR
   S(L,1,J)=BC_1(I)
  END DO
END DO
IF(IND3.NE.0)THEN
      DO I2=1,NJ
        DO I3=1,NJR
    S(IND3,I2,I3)=BC_1(IND3)
   END DO
  END DO
END IF
IF(IDES1.EQ.1)THEN
  SS(IDES)=S(IDES,1,1)
  CALL IST_Z(1,S(L2,1,1),SS,Q2)
  DO I=1,NC
   Q2(I)=Q2(I)*RO_S
  END DO
  DO I1=1,NJ
   DO I2=1,NJR
    S(IDES,I1,I2)=BC_1(IDES)
    P(IDES,I1,I2)=BC_1(IDES)
    S(NC+IDES,I1,I2)=Q2(IDES)
    P(NC+IDES,I1,I2)=Q2(IDES)
   END DO
  END DO
END IF
IF(IND4.NE.0)THEN
  DO I1=1,NJ
   DO I2=1,NJR
    S(IDES+1,I1,I2)=BC_1(IDES+1)
    P(IDES+1,I1,I2)=BC_1(IDES+1)
    S(NC+IDES+1,I1,I2)=Q2(IDES+1)
    P(NC+IDES+1,I1,I2)=Q2(IDES+1)
   END DO
  END DO
END IF
IF(ISAT.EQ.1)THEN
  DO I1=1,NC
   SS(I1)=BC_1(I1)
  END DO
  CALL IST_Z(1,S(L2,1,1),SS,Q2)
  DO I=1,NC
   Q2(I)=Q2(I)*RO_S
  END DO
  DO I1=1,NC
   DO I2=1,NJ
        DO I3=1,NJR
             S(I1,I2,I3)=BC_1(I1)
     S(I1+NC,I2,I3)=Q2(I1)
             P(I1,I2,I3)=BC_1(I1)
     P(I1+NC,I2,I3)=Q2(I1)
        END DO
   END DO
  END DO
  DO I1=1,NC
       BC_1(I1)=0
  END DO
  BC_1(IND3)=P_C(ind3)/555.0/TO
end if
```

```
      RETURN
      END
C
C
      SUBROUTINE VEL_POR(JJ,NJ_OR_1,IVEL_1)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 U(14,201),EPS(1000),U1(101,201),A1(14,201),V(30),
     . VS(30),RP(30),EPS1(1000)
      COMMON/UG/U_GUESS,EPS
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201,101)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),F(14),P(14,201)
      COMMON/NEW_GRID/DELR0(101),ALF1(101),NGR1,NGR2
      COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT,NJ1
      COMMON/BND3/S(14,201,101)
      COMMON/PROP_B_Z/REC,VOID_B,D_I,D_E,S_B,ALF,GC,Z,M_W
      COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1,HEAT(4)
      COMMON/INDIC/L1,L2,L3,L4,L5,L6,LL
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      DATA R1/555./
c a number to make the division of grids correctly
      IF(IVEL_1.EQ.0)THEN
       III=2
c there is a temperature difference between the gas and solid phases
       EC=1./VOID_B-1.
       DP=R_P*2.
c grid number for velocity dietribution
       NP1=125
       TR=D_I/DP
c grid fraction for almost-constant velocity section in the radial direction
       FRAC1=.93
c grid fraction for variable velocity section in the radial direction
       FRAC2=.07
c number of grids in the bulk section
       K1=21
c number of grids close to the wall
       K2=104
       K11=K1
       K22=K2
       IR=5
c start computing the velocity profile
c guess the velocity at the center of the column
         U_GUESS=1.0
 133       IF(U_GUESS.EQ.1.0)THEN
c set the radial grid velcities to guess one if it is the first itration
          DO I1=1,NP-1
           U(JJ,I1)=U_GUESS
          END DO
         ELSE
c if not set it to the  most recent one
          DO I1=2,NP-1
           U(JJ,I1)=C(1,I1)
          END DO
          U(1,1)=U_GUESS
         END IF
c at wall no-slip flow, therefore velocity is zero
         U(JJ,NP)=0.
         RO_F=S(L5,JJ,1)/S(L1,JJ,1)/R1
         IF(U_GUESS.EQ.1)THEN
          GN=S(L4,JJ,1)*RO_F/60.*VOID_B
         ELSE
          GN=UD*RO_F
```

```
        END IF
        T_GAS=S(L1,JJ,1)
        REY_D=RE(GN,T_GAS)
        REC= 1.75/150.*REY_D/(1-VOID_B)
        US=((1.+4.*REC)**.5-1.)/(2*REC)
c "DIFFEQ1" is the main routine to compute the velocity profile
        CALL DIFFEQ1(U,A1,1,NP,3,5)
c if converged go out
        IF(ABS(U_GUESS-C(1,2)).GT.1.0E-7)THEN
         U_GUESS=C(1,2)
         UD=S(L4,JJ,1)*VOID_B/C(1,2)/60.
         GO TO 133
        END IF
c the velocity obtained is a intrinisic velocity multiplying by porosity
c gives seepage velocity, here c(1,I1) is array of variable velocity which
c is being calculated in one of the subroutine of DIFFEQ1 routine
        DO I1=1,NP
         U(JJ,I1)=C(1,I1)*UD*60.
         U1(JJ,I1)=C(1,I1)
        END DO
        DO I=1,NP
         EPS1(I1)=EPS(I1)
        END DO
        idelr_in1=5
        idelr_in2=7
        n1=1
        delr0(n1)=delr1*5
        eps(n1)=eps(n1)
        alf1(n1)=(1-eps(n1))/eps(n1)
        if(nj_or_1.eq.1)then
         do i6=1,nj
          s(l4,i6,n1)=u(jj,1)/eps(n1)
         end do
        else
         s(l4,i6,n1)=u(jj,1)/eps(n1)
        end if
        do i8=5,20,5
         n1=n1+1
         delr0(n1)=delr1*5
         eps(n1)=eps(i8)
         alf1(n1)=(1-eps(n1))/eps(n1)
         if(nj_or_1.eq.1)then
          do i6=1,nj
           s(l4,i6,n1)=u(jj,i8)/eps(n1)
          end do
         else
          s(l4,i6,n1)=u(jj,i8)/eps(n1)
         end if
        end do
        ngr1=n1
        do i8=27,125,7
         n1=n1+1
         delr0(n1)=delr2*7
         eps(n1)=eps(i8)
         alf1(n1)=(1-eps(n1))/eps(n1)
         if(nj_or_1.eq.1)then
          do i6=1,nj
           s(l4,i6,n1)=u(jj,i8)/eps(n1)
          end do
         else
          s(l4,i6,n1)=u(jj,i8)/eps(n1)
         end if
        end do
        if(nj_or_1.eq.1)then
```

```
        do i6=1,nj
          s(l4,i6,N1)=u(jj,124)/eps(124)
        end do
       else
        s(l4,i6,N1)=u(jj,124)/eps(124)
       end if
       ngr2=n1-ngr1
      ELSE IF(IVEL_1.EQ.1)THEN
       C1=1.4
       NJR1=6
       NJR2=NJR-10
       DELR11=.8*D_I/(2*(NJR1-1))
       DELR12=.2*D_I/(2*(NJR-NJR1))
       DELR0(1)=DELR11
       DELR0(NJR)=DELR12
       EPS(1)=VOID_B
       EPS(NJR)=VOID_B*(1+C1)
       ALF1(1)=(1-EPS(1))/EPS(1)
       ALF1(NJR)=(1-EPS(NJR))/EPS(NJR)
       I1=2
       DO K=NJR-2,1,-1
        IF(I1.LE.NJR1)THEN
         Y1=(NJR1-1)*DELR11+NJR2*DELR12
         DELR0(I1)=DELR11
        ELSE
         Y1=DELR12*K
         DELR0(I1)=DELR12
        END IF
        CALL COEF2(Y1,VOID_B,D_P,C1,POR)
        EPS(I1)=POR
        ALF1(I1)=(1-EPS(I1))/EPS(I1)
        I1=I1+1
      END DO
      RP(1)=0.
      RP(NJR)=1.
      DO K=2,NJR-1
       RP(K)=2*DELR0(K)/D_I+RP(K-1)
      END DO
      VAV=S(L4,1,1)*EPS(1)
          NJJ=NJR
          RMS=1.-2*D_P/D_I
          BO=.16*(D_P/(D_I))**(-3./2.)
          AO=1./(BO+2.)-RMS/(BO+1)
          A2=RMS/(BO+1.)
          A3=1./(BO+2.)
          D1=AO/2.+A2/(BO+3.)-A3/(BO+4.)
          DO I=1,NJJ
           VS(I)=(AO+A2*(RP(I)**(BO+1.))-A3*(RP(I)**(BO+2.)))/
   .   (2.*D1)
          END DO
          DO I=1,NJJ
           V(I)=VS(I)*VAV
       do i6=1,nj
        s(l4,i6,i)=v(i)/eps(i)
       end do
      END DO
     END IF
    RETURN
    END
C
C
c this subroutine is being called by maine routine to compute the variable in
c C matrix. in this routine first velocity profile is being determined. the
c conductivity, diffusivity,pososity....... are being calculated in this
```

```
c routine by calling the approperiate subroutine. the routine obtained the
c C matix in axial and radial directions. it itrates till it conveges to the
c alloweble error
c
      SUBROUTINE FUNCT4(J1)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K_F(4),K_FO(4),MOL(4),M_AVE,M_W(4),NU_NP,NU_NW,KGAS,
     . K_FP(4,201,201),BC_1(4),D_L(4),Q2(4),MW,A1(14,201),
     . C1(14,201,101),D_RF(4),U(14,201),EPS(1000),E(30),AKT(30),
     . C2(14,201,101),F01(101),F11(101),F21(101),F02(101),
     . F12(101),F22(101),RP(101),DR(101),RR0(101),D_LP_R(4,201,101),
     . D_LP_Z(4,201,101),U1(101,201),YO(4),SS(4),RATE_H1(201,33)
      COMMON/UG/U_GUESS,EPS
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201,101)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),F(14),P(14,201)
      COMMON/BND3/S(14,201;101)
      COMMON/NEW_GRID/DELR0(101),ALF1(101),NGR1,NGR2
      COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT,NJ1
      COMMON/PR_OLD_Z/CON_Z(201,101),CON_R(201,101),CON_S(201,101),
     . H_W(201,101),CP_P(201,101),H_FP(201,101),Q(4,201,101),
     . VISC_P(201,101),D_LPP(4,201),RATE_C1(201,101)
      COMMON/PROP_WAL/CP_WA,RO_WA,ERROR0,ERROR1
      COMMON/PROP_B_Z/REC,VOID_B,D_I,D_E,S_B,ALF,GC,Z,M_W
      COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1,HEAT(4)
      COMMON/INIT_Z/RO_FO,U_FO,U_F1,CP_FO,T_FO,TO,C_FO(4),K_F,P_C(4)
      COMMON/MISC_Z/R,G_F,PLP_TOT,BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_1
      COMMON/INDIC/L1,L2,L3,L4,L5,L6,LL,M3,II,JJ,IVEL_1,MT
      COMMON/INDIC1/IND(4)
      COMMON/RADIAL/RP
      COMMON/GAS/INERT,NCOMP,MW
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      DATA EMIS,R1,IONCE,JONCE,ICOUNT,ICOND1,ICOM/.9,555.,0,0,0,1,1/
c there is a temperature difference between the gas and solid phases
c      IF(LL.EQ.ICOM)THEN
c       BC_L1=TEMPIN(TIME)
c           ICOM=LL+1
c      END IF
      IF(J1.EQ.1 .AND. IONCE.EQ.0)THEN
       IONCE=1
       NJ_OR_1=1
       I_SOL_FLO=0
c an indication number for type of gas to compute the gas conductivity,
c 1=N2, 2=air, 3=co2
       IGAS=1
       DP=R_P*2.
       IF(IVEL_1.EQ.1)THEN
        IR=10
       ELSE
        IR=5
       END IF
c calculating the area for heat transfer to the wall or from the wall
       D_T=D_E+D_I
       X_W=D_E-D_I
       A_C=2*D_I/((D_I+D_E)*X_W)
       A_INS=2*D_E/((D_I+D_E)*X_W)*1.4
c heat transfer coefficient outside of the wall
       H_OW=0.1
      END IF
c call vel_por to compute velosity and porosity profile
      IF(J1.EQ.1 .AND. JONCE.EQ.0)THEN
       JONCE=1
```

```
c compute the ratio of distance from the center to each grid over the pellet
c diameter
     RR=0.
     RR0(NJR)=0.
     DO I5=NJR-1,1,-1
      RR=RR+DELR0(I5+1)/D_P
      RR0(I5)=RR
     END DO
c this section calculates the coefficients to the discretized PDE's
c
c compute the inverse of distance from the center to each grid
     RR=0.
     DO I5=1,NJR
      IF(I5.NE.1)THEN
       RR=RR+DELR0(I5)
       RP(I5)=1./RR
      ELSE
       RP(I5 =0.
      END IF
      IF(I5.LT.IR) THEN
       DELR=DELR0(1)
       DR(I5)=DELR
      ELSE IF(I5.eq.IR)THEN
       DELR1=DELR0(IR)
       DELR2=DELR0(IR+1)
       DELR=DELR1
       DR(I5)=DELR
       FR=DELR2/DELR1
c these variables are bein computed at the pont where two adjant grids are
c not the same size
       F01(I5)=FR/(1+FR)
       F11(I5)=(1.-FR)/FR
       F21(I5)=1./((1.+FR)*FR)
       F02(I5)=1./(1+FR)
       F12(I5)=1./FR
       F22(I5)=1./((1.+FR)*FR)
      ELSE
       DELR=DELR0(IR+1)
       DR(I5)=DELR
      END IF
     END DO
    END IF
c this section calls the proper subroutine to compute the parameters such as
c diffusivity, conductivity.....
   IF(MT.EQ.1)THEN
    IF(J1.EQ.1 .AND. ICOUNT.LT.JJ)THEN
     I5=JJ
     ICOUNT=ICOUNT+1
      DO I6=1,NJR
       DO M1=1,NC
        Y0(M1)=C(M1,I6)*R1*C(L1,I6)/S(L5,I5,I6)
       END DO
       TEMP=C(L1,I6)
       M_AVE=0
       SUM_Y=0.
       DO M1=1,NC
        M_AVE=M_AVE+Y0(M1)*M_W(M1)
        SUM_Y=SUM_Y+Y0(M1)
       END DO
       M_AVE=M_AVE+(1-SUM_Y)*28.
       RO_F=S(L5,I5,I6)/(C(L1,I6)*R1)
       IF(RO_F.LE.0)RO_F=P_TOT/(C(L1,I6)*R1)
       GN=S(L4,I5,I6)*RO_F/60.*VOID_B
c calculate Reynolds number
```

```
      REY=RE(GN,TEMP)
c calculate specific heat of gas phase
      CP_P(I5,I6)=CPGAS(TEMP,YO)
c calc viscosity
      VISC_P(I5,I6)=VIS(TEMP)*60.
c calc heat transfer coeff
      H_FP(I5,I6)=HFILM(MW,TEMP,REY,CP_P(I5,I6))*60.
      itest11=0
      IF(ITEST11.EQ.0) MN=2
c this section compute the conductivity of gas and solid in axial and
cradial directions
      CALL PHI_COND(IGAS,TEMP,CON_S1,EPS(1),PHI,PHIW)
      CALL CONDU(MN,IGAS,PHI,D_P,CON_S1,RR0(I6),U1(1,I6),REY,
     .      TEMP,EPS(I6),EPS(1),EFF_CON_R1,CON_R1,EFF_CON_Z1,
     .      CON_Z1,CON_P1,NJR,NGR1,NGR2,I6)
c if there is a tepmperature difference between the gas and solid
      IF(I_SOL_FLO.EQ.1)THEN
       CON_Z(I5,I6)=CON_Z1
       CON_S(I5,I6)=CON_P1
c if there is no tepmperature difference between the gas and solid
      ELSE
       CON_Z(I5,I6)=EFF_CON_Z1
      END IF
c computation for number of component
      DO M1=1,NC
c calc the molecular diffusivity of eac componet in the mixture
       DIF=DIFF(M1,TEMP,S(L5,I5,I6),YO)*60
c calc Schmidt number
       SC_N=VISC_P(I5,I6)/DIF/RO_F/M_AVE
c calc axial diffusion
       CALL AXDIFF_R_Z(MN,IGAS,PHI,D_P,DIF,RR0(I6),U1(1,I6),
     .       REY,SC_N,TEMP,EPS(I6),EPS(1),EFF_DIFF_R,EFF_DIFF_Z,
     .       njr,ngr1,NGR2,I6)
C         D_LP_R(M1,I5,I6)=EFF_DIFF_R
C         D_LP_Z(M1,I5,I6)=EFF_DIFF_Z
        D_LP_Z(M1,I5,I6)=CON_Z(I5,I6)/RO_F/CP_P(I5,I6)
      END DO
c calc the heat transfer coefficint between the gas phase and the wall
      IF(I6.EQ.NJR) THEN
       IF(GN.EQ.0)GN=S(L4,1,1)*RO_F/60.*VOID_B
       REY=RE(GN,TEMP)
       CALL HEAT_WALL(IGAS,PHIW,D_P,CON_S1,RR0(I6),U1(1,I6),
     .       REY,TEMP,VOID_B,H_W1)
       H_W(I5,NJR)=H_W1
      END IF
      IF(ICOND1.EQ.1)THEN
       VAV=G_F/D_I/D_I/PI*4
       GA=G_F/D_I/D_I/PI*4.*RO_F
       TAV=S(L1,1,1)/1.8
       RO=S(L5,1,1)/S(L1,1,1)/R1
       CP=CP_P(1,1)
       AVIS=VISC_P(1,1)
       AKC=.000358895+3.0026706379E-5*S(L1,1,1)-5.3528942E-9*
     1  S(L1,1,1)*S(L1,1,1)
       CALL CONDUC1(EMIS,CON_S1,D_P,D_I,GA,VAV,TAV,RO,CP,AVIS,
     1  AKC,NJR,EPSEX,EPS(1),EPS(NJR),DELR0,E,AKT)
       ICOND1=0
      END IF
      CON_R(I5,I6)=AKT(I6)
      IF(E(I6).LE.0)E(I6)=E(I6-1)
      DO M1=1,NC
C         D_LP_R(M1,I5,I6)=AKT(I6)/RO/CP
       D_LP_R(M1,I5,I6)=E(I6)
      END DO
```

```
          END DO
          IF(CON_Z(I5,NJR).EQ.0)THEN
           CON_Z(I5,NJR)=CON_Z(I5,NJR-1)
           DO M1=1,NC
             D_LP_Z(M1,I5,NJR)=CON_Z(I5,NJR)/RO_F/CP_P(I5,NJR)
           END DO
          END IF
          DO I1=1,NJ
           DO M1=1,NC
            D_LPP(M1,J1)=D_LP_Z(M1,I1,1)
           END DO
          END DO
         END IF
c set the parameters to  new variables
        J=JJ
        RO_F=S(L5,J,J1)/(C(L1,J1)*R1)
        DO M1=1,NC
         D_L(M1)=D_LP_Z(M1,J,J1)
        END DO
        DO M1=1,NC
         D_RF(M1)=D_LP_R(M1,J,J1)
        END DO
        DO I1=1,NC
         SS(I1)=C(I1,J1)
        END DO
c call the isotherm to compute the equilibruim concentration
        IF(I_SOL_FLO.EQ.1)THEN
         CALL IST_Z(1,C(L2,J1),SS,Q2)
        ELSE
         CALL IST_Z(1,C(L1,J1),SS,Q2)
        END IF
c based the equilibruim per unit volume of pellet
        DO I=1,NC
         Q(I,J,J1)=Q2(I)*RO_S
        END DO
c this is the boundry, the first grids in axial direction
         IF(J.EQ.1)THEN
          RATE_C=0
c compute the total amount is being adsorbed
          DO M1=1,NC
           RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
          END DO
          RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
          RATE_H=0
c compute the total heat of adsorption
C          HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C     .    C(3,J1)**2/RO_S/RO_S)*44
          DO M1=1,NC
           RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
     .     C(M1+NC,J1))
          END DO
          RATE_H1(J,J1)=RATE_H
c this "if" calculates the first grid in radial direction, center of the bed
          IF(J1.EQ.1) THEN
c computation of molar density component
          TOT_C=0.
          DO M1=1,NC-1
           A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
     .     DR(J1)/DR(J1)+1./DELT)
           P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
           P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
           P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
           P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
           P5=-(-P(M1,J1))/DELT
```

```
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
   .  C(NC,J1)
c computation of molar adsorption for each component
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
   .   P(M1+NC,J1)/DELT)
      END DO
c the same computation as above except it is now for the grid at the wall
         ELSE IF(J1.EQ.NJR)THEN
         TOT_C=0.
c comp of molar density of component
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
   .   DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
       P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
       P6=P(M1,J1)/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
   .  C(NC,J1)
c comp the molar  adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
   .   P(M1+NC,J1)/DELT)
      END DO
         ELSE
c the same computation as above except it is now for the grids from the center
c to the pont where the velocity start to chane
      IF(J1.NE.JR)THEN
c comp molar density of each component
         TOT_C=0.
      DO  M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/DR(J1)
   .   /DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
       P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/
   .   DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
   .  C(NC,J1)
c compute molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
   .   P(M1+NC,J1)/DELT)
      END DO
c the same computation as above except it is now for the grids from the point
c where the velocity starts to change to the wall
```

```
      ELSE
          TOT_C=0.
c compute molar density of each component
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
     . RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(S(M1,J+1,J1)+BC_1(M1))
       P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
     . C(M1,J1-1))/(DR(J1))
       P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
     . /DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute molar  adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
     END IF
        ENDIF
C
C-----------------------------------------------------------
C this section compute the S matrix variables for grid at z=L, the outflow
c boundry
c
c
     ELSE IF(J.EQ.NJ1)THEN
c compute the total amount adsorbed
      RATE_C=0
      DO M1=1,NC
       RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
      END DO
c compute the total heat of adsorption
      RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
      RATE_H=0
C      HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C    . C(3,J1)**2/RO_S/RO_S)*44
      DO M1=1,NC
       RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
     . C(M1+NC,J1))
      END DO
      RATE_H1(J,J1)=RATE_H
c this section compute the s matrix at outflow boundry at the ceter of bed
      IF(J1.EQ.1) THEN
      TOT_C=0.
c compute the component molar density
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
     . DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
       P5=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
       TOT_C=TOT_C+C(M1,J1)
      END DO
```

```
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute the molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
c this section is for r at the wall
     ELSE IF(J1.EQ.NJR) THEN
     TOT_C=0.
c compute molar density
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
     . DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
       P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
     ELSE
c this section computes S matrix for those grids with constant grid siz
      IF(J1.NE.IR)THEN
      TOT_C=0.
c compute molar density
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
     . DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
       P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
c this section compute s for variable grid size
      ELSE
      TOT_C=0.
c compute the molar density
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
     . RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
```

```
         P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
         P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
         P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
         P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
     .   C(M1,J1-1))/(DR(J1))
         P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
     .   /DR(J1)/DR(J1)
         P6=-(-P(M1,J1))/DELT
         F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
         TOT_C=TOT_C+C(M1,J1)
        END DO
        F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .  C(NC,J1)
c compute molar adsorption
        DO M1=1,NC
         A2=1./(K_F(M1)*AINT+1./DELT)
         F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .   P(M1+NC,J1)/DELT)
        END DO
       END IF
      END IF

C
C-------------------------------------------------------------
C
c this section computes the s matrix for grids inside of the column
c
      ELSE
c comp total molar adsorption
        RATE_C=0
        DO M1=1,NC
         RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
        END DO
c compute total heat of adsorption
        RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
        RATE_H=0
C         HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C   .    C(3,J1)**2/RO_S/RO_S)*44
        DO M1=1,NC
         RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
     .   C(M1+NC,J1))
       END DO
       RATE_H1(J,J1)=RATE_H
c this section copm s for the grids at the center of bed
      IF(J1.EQ.1) THEN
        TOT_C=0.
c comp molar density
        DO M1=1,NC-1
         A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
     .   DR(J1)/DR(J1)+1./DELT)
         P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
         P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
         P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
         P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
         P5=-(-P(M1,J1))/DELT
         F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
         TOT_C=TOT_C+C(M1,J1)
        END DO
        F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .  C(NC,J1)
c comp molar adsorption
        DO M1=1,NC
         A2=1./(K_F(M1)*AINT+1./DELT)
         F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
```

```
     .    P(M1+NC,J1)/DELT)
         END DO
c comp s for the grd at wall
        ELSE IF(J1.EQ.NJR) THEN
         TOT_C=0.
c comp molar density
         DO M1=1,NC-1
          A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/DR(J1)
     .    /DR(J1)+1./DELT)
          P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
          P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
          P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
          P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
          P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
          P6=-(-P(M1,J1))/DELT
          F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
          TOT_C=TOT_C+C(M1,J1)
         END DO
         F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .   C(NC,J1)
c comp molar adsorption
         DO M1=1,NC
          A2=1./(K_F(M1)*AINT+1./DELT)
          F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .    P(M1+NC,J1)/DELT)
         END DO
c this section computes s for grids of constant size
       ELSE
        IF(J1.NE.IR)THEN
         TOT_C=0.
c comp the molar density
         DO M1=1,NC-1
          A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2.*D_RF(M1)/
     .    DR(J1)/DR(J1)+1./DELT)
          P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
          P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
          P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
          P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
          P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/DR(J1)/DR(J1)
          P6=-(-P(M1,J1))/DELT
          F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
          TOT_C=TOT_C+C(M1,J1)
         END DO
         F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .   C(NC,J1)
c comp the molar adsorption
         DO M1=1,NC
          A2=1./(K_F(M1)*AINT+1./DELT)
          F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .    P(M1+NC,J1)/DELT)
         END DO
c this section computes s  for variale grid size
        ELSE
         TOT_C=0.
c comp molar density
         DO M1=1,NC-1
          A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
     .    RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
          P1=D_L(M1)/DELZ**2*(S(M1,J+1,J1)+S(M1,J-1,J1))
          P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
          P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
          P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
     .    C(M1,J1-1))/(DR(J1))
          P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
```

```
     .     /DR(J1)/DR(J1)
           P6=-(-P(M1,J1))/DELT
           F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
           TOT_C=TOT_C+C(M1,J1)
          END DO
          F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .    C(NC,J1)
c comp molar adsorption
          DO M1=1,NC
           A2=1./(K_F(M1)*AINT+1./DELT)
           F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .     P(M1+NC,J1)/DELT)
          END DO
         END IF
        END IF
       END IF
C FOR TEMPERATURE
C-----------------------------------------------------------
      ELSE
c set the parameters to  new variables
       H_FW=H_W(J,NJR)
       RO_F=S(L5,I5,J1)/(C(1,J1)*R1)
       CP_F=CP_P(J,J1)
       H_FS=H_FP(J,J1)
       CON_L=CON_Z(J,J1)
       CON_RF=CON_R(J,J1)
       CON_P=CON_S(J,J1)
       VISC_F=VISC_P(J,J1)
       J=JJ
       KK=0
       IF(J.EQ.1)THEN
        IF(J1.EQ.1)THEN
c computation of gas temperature
          IF(I_SOL_FLO .EQ.1)THEN
           A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .     1./DELT)+4*CON_RF/DR(J1)/DR(J1))
           P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
           P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
           P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
           P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .     S(L4,J,J1))/(DELZ)
           P5=4*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
           P6=-RO_F*CP_F*(-P(1,J1))/DELT
           F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c computation of solid temperature
           A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     .     4*CON_P/DR(J1)/DR(J1))
           P1=AINT*H_FS*(C(1,J1)-C(2,J1))
           P2=-RATE_H1(J,J1)
           P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
           P4=4*CON_P*(C(2,J1+1))/DR(J1)/DR(J1)
           P5=-RO_S*CP_S*(-P(2,J1))/DELT
           F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5)
c computation of effective temperature
          ELSE
           A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .     DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+4*
     .     EPS(J1)*CON_RF/DR(J1)/DR(J1))
           P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
           P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
           P3=-(1-EPS(J1))*RATE_H1(J,J1)
           P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .     S(L4,J,J1))/(DELZ)
           P5=EPS(J1)*4*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
```

```
          P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)

          F(2)=0.
         END IF
c the same computation as above except it is now for the grid at the wall
             ELSE IF(J1.EQ.NJR)THEN
c comp of gas phase temperature
         IF(I_SOL_FLO .EQ.1)THEN
          A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .    (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
          P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
          P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4,J,J1))/(DELZ)
          P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .    C(1,J1)))/(2*DR(J1))
          P6=CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .    C(1,J1)))/DR(J1)/DR(J1)
          P7=-RO_F*CP_F*(-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c comp of solid phase temperature
          A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+2*CON_P/DR(J1)
     .    /DR(J1))
          P1=AINT*H_FS*(C(1,J1)-C(2,J1))
          P2=-RATE_H1(J,J1)
          P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
          P4=CON_P*RP(J1)*(C(2,J1-1)-C(2,J1-1))/(2*DR(J1))
          P5=CON_P*(2*C(2,J1-1))/DR(J1)/DR(J1)
          P6=-RO_S*CP_S*(-P(2,J1))/DELT
          F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
c comp of effective temperature
         ELSE
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    2.*EPS(J1)*CON_RF/DR(J1)/DR(J1))
          P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
          P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
          P3=-(1-EPS(J1))*RATE_H1(J,J1)
          P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4,J,J1))/(DELZ)
          P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .    C(1,J1)))/(2*DR(J1))
          P6=EPS(J1)*CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/
     .    CON_RF*(C(3,J1)-C(1,J1)))/DR(J1)/DR(J1)
          P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

          F(2)=0.
         END IF
          F(3)=-C(3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(1,J1)-
     .    C(3,J1))-A_INS*H_OW*(C(3,J1)-TO))-(-P(3,J1))
             ELSE
c to the pont where the velocity start to change
         IF(J1.NE.IR)THEN
c compute gas temperature
         IF(I_SOL_FLO .EQ.1)THEN
          A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .    (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
```

```
      P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
      P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J,J1))/(DELZ)
      P5=CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
      P6=CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
      P7=-RO_F*CP_F*(-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
      A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+2*CON_P/
     .   DR(J1)/DR(J1))
      P1=AINT*H_FS*(C(1,J1)-C(2,J1))
      P2=-RATE_H1(J,J1)
      P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
      P4=CON_P*RP(J1)*(C(2,J1+1)-C(2,J1-1))/(2*DR(J1))
      P5=CON_P*(C(2,J1+1)+C(2,J1-1))/DR(J1)/DR(J1)
      P6=-RO_S*CP_S*(-P(2,J1))/DELT
      F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
c compute effective  temperature
     ELSE
      A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .   DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
     .   EPS(J1)*CON_RF/DR(J1)/DR(J1))
      P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
      P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
      P3=-(1-EPS(J1))*RATE_H1(J,J1)
      P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-S(L4,J,J1))/
     .   (DELZ)
      P5=EPS(J1)*CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
      P6=EPS(J1)*CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
      P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .   (-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

      F(2)=0.
     END IF
c the same computation as above except it is now for the grids from the point
c where the velocity starts to change to the wall
     ELSE
c compute the gas temperature
      IF(I_SOL_FLO .EQ.1)THEN
      A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .   1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*F12(J1)/
     .   DR(J1)/DR(J1))
      P1=CON_L/DELZ**2*(S(L1,J+1,J1)+BC_L1)
      P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
      P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
      P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J,J1))/(DELZ)
      P5=CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-
     .   F01(J1)*C(1,J1-1))/(DR(J1))
      P6=2*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*C(1,J1-1))/
     .   DR(J1)/DR(J1)
      P7=-RO_F*CP_F*(-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute the solid temperature
      A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+CON_P*RP(J1)*
     .   F11(J1)/DR(J1)+2*CON_P*F12(J1)/DR(J1)/DR(J1))
      P1=AINT*H_FS*(C(1,J1)-C(2,J1))
      P2=-RATE_H1(J,J1)
      P3=CON_P/DELZ/DELZ*(S(L2,J+1,J1)+S(L2,J+1,J1))
      P4=CON_P*RP(J1)*(F21(J1)*C(2,J1+1)-F01(J1)*
     .   C(2,J1-1))/(DR(J1))
      P5=2*CON_P*(F22(J1)*C(2,J1+1)+F02(J1)*C(2,J1-1))/
     .   DR(J1)/DR(J1)
```

```
       P6=-RO_S*CP_S*(-P(2,J1))/DELT
       F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
      ELSE
c compute the effective temperature
       A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     . DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     . EPS(J1)*CON_RF*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/
     . DR(J1)/DR(J1)))
       P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J+1,J1)+BC_L1)
       P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
       P3=-(1-EPS(J1))*RATE_H1(J,J1)
       P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     . S(L4,J,J1))/(DELZ)
       P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-F01(J1)*
     . C(1,J1-1))/(DR(J1))
       P6=2*EPS(J1)*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*
     . C(1,J1-1))/DR(J1)/DR(J1)
       P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     . (-P(1,J1))/DELT
       F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

       F(2)=0.
      END IF
     END IF
        END IF
C
C-----------------------------------------------------------
C this section compute the S matrix variables for grid at z=L, the outflow
c boundry
c
c
     ELSE IF(J.EQ.NJ1)THEN
      IF(J1.EQ.1)THEN
c compute the gas temperature
      IF(I_SOL_FLO .EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     . (RO_F*CP_F)/DELT+4*CON_RF/DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
       P4=-RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
       P5=4*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
       P6=-RO_F*CP_F*(-P(1,J1))/DELT
       F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c compute the solid temperaure
       A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+4*CON_P/
     . DR(J1)/DR(J1))
       P1=AINT*H_FS*(C(1,J1)-C(2,J1))
       P2=-RATE_H1(J,J1)
       P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
       P4=4*CON_P*C(2,J1+1)/DR(J1)/DR(J1)
       P5=-RO_S*CP_S*(-P(2,J1))/DELT
       F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5)
      ELSE
c compute the effective temperature
       A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     . DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     . 4*EPS(J1)*CON_RF/DR(J1)/DR(J1))
       P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
       P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))*RATE_H1(J,J1)
       P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
```

```
        P5=4*EPS(J1)*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
        P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .  (-P(1,J1))/DELT
        F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)

        F(2)=0.
       END IF
c this section is for r at the wall
      ELSE IF(J1.EQ.NJR) THEN
c compute gas temperature
       IF(I_SOL_FLO .EQ.1)THEN
        A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+(RO_F*
     .  CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
        P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
        P4=-RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-S(L4,J-1,J1))/
     .  (2*DELZ)
        P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .  C(1,J1)))/(2*DR(J1))
        P6=CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .  C(1,J1)))/DR(J1)/DR(J1)
        P7=-RO_F*CP_F*(-P(1,J1))/DELT
        F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
        A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+2*CON_P/DR(J1)/
     .  DR(J1))
        P1=AINT*H_FS*(C(1,J1)-C(2,J1))
        P2=-RATE_H1(J,J1)
        P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
        P4=CON_P*RP(J1)*(C(2,J1-1)-C(2,J1-1))/(2*DR(J1))
        P5=CON_P*(2*C(2,J1-1))/DR(J1)/DR(J1)
        P6=-RO_S*CP_S*(-P(2,J1))/DELT
        F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
       ELSE
c compute effective temperature
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .  2*EPS(J1)*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))*RATE_H1(J,J1)
        P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     .  S(L4,J-1,J1))/(2*DELZ)
        P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     .  C(1,J1)))/(2*DR(J1))
        P6=EPS(J1)*CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/CON_RF*
     .  (C(3,J1)-C(1,J1)))/DR(J1)/DR(J1)
        P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .  (-P(1,J1))/DELT
        F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

        F(2)=0.
       END IF
c compute the wall temperature
        F(3)=-C(3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(1,J1)-
     .  C(3,J1))-A_INS*H_OW*(C(3,J1)-TO))-(-P(3,J1))
      ELSE
c this section computes S matrix for those grids with constant grid siz
       IF(J1.NE.IR)THEN
c compute gas temperature
       IF(I_SOL_FLO .EQ.1)THEN
        A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .  (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
```

```
      P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
      P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
      P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
      P4=-RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     .     S(L4,J-1,J1))/(2*DELZ)
      P5=CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
      P6=CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
      P7=-RO_F*CP_F*(-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
      A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+2*CON_P/
     .     DR(J1)/DR(J1))
      P1=AINT*H_FS*(C(1,J1)-C(2,J1))
      P2=-RATE_H1(J,J1)
      P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
      P4=CON_P*RP(J1)*(C(2,J1+1)-C(2,J1-1))/(2*DR(J1))
      P5=CON_P*(C(2,J1+1)+C(2,J1-1))/DR(J1)/DR(J1)
      P6=-RO_S*CP_S*(-P(2,J1))/DELT
      F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
     ELSE
c compute effective temperature
      A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .     DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
     .     EPS(J1)*CON_RF/DR(J1)/DR(J1))
      P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
      P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
      P3=-(1-EPS(J1))*RATE_H1(J,J1)
      P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     .     S(L4,J-1,J1))/(2*DELZ)
      P5=EPS(J1)*CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
      P6=EPS(J1)*CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
      P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .     (-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

      F(2)=0.
     END IF
c this section compute s for variable grid size
     ELSE
c compute the gas temperature
     IF(I_SOL_FLO.EQ.1)THEN
      A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .     1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*
     .     F12(J1)/DR(J1)/DR(J1))
      P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
      P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
      P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
      P4=-RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-S(L4,J-1,J1))/
     .     (2*DELZ)
      P5=CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-F01(J1)*C(1,J1-1))/
     .     (DR(J1))
      P6=2*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*C(1,J1-1))/DR(J1)
     .     /DR(J1)
      P7=-RO_F*CP_F*(-P(1,J1))/DELT
      F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute the solid temperature
      A2=1./(2*CON_P/DELZ**2+RO_S*CP_S/DELT+CON_P*RP(J1)*
     .     F11(J1)/DR(J1)+2*CON_P*F12(J1)/DR(J1)/DR(J1))
      P1=AINT*H_FS*(C(1,J1)-C(2,J1))
      P2=-RATE_H1(J,J1)
      P3=CON_P/DELZ/DELZ*(S(L2,J-1,J1)+S(L2,J-1,J1))
      P4=CON_P*RP(J1)*(F21(J1)*C(2,J1+1)-F01(J1)*C(2,J1-1))
     .     /(DR(J1))
      P5=2*CON_P*(F22(J1)*C(2,J1+1)+F02(J1)*C(2,J1-1))/
```

```
     .    DR(J1)/DR(J1)
          P6=-RO_S*CP_S*(-P(2,J1))/DELT
          F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
         ELSE
c compute the effective temperature
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    EPS(J1)*CON_RF*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/DR(J1)/
     .    DR(J1)))
          P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
          P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))*RATE_H1(J,J1)
          P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J-1,J1)-
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-F01(J1)*
     .    C(1,J1-1))/(DR(J1))
          P6=2*EPS(J1)*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*
     .    C(1,J1-1))/DR(J1)/DR(J1)
          P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

          F(2)=0.
         END IF
        END IF
       END IF

C
C-------------------------------------------------------------
C
c this section computes the s matrix for grids inside of the column
c
       ELSE
        IF(J1.EQ.1) THEN
c comp gas temperature
         IF(I_SOL_FLO.EQ.1)THEN
          A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .    (RO_F*CP_F)/DELT+4*CON_RF/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
          P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-S(L4,J-1,J1))/
     .    (2*DELZ)
          P5=4*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
          P6=-RO_F*CP_F*(-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c comp solid temperature
          A2=1./(2*CON_P/DELZ/DELZ+4*CON_P/DR(J1)/DR(J1)+
     .    RO_S*CP_S/DELT)
          P1=AINT*H_FS*(C(1,J1)-C(2,J1))
          P2=-RATE_H1(J,J1)
          P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
          P4=4*CON_P*(C(2,J1+1))/DR(J1)/DR(J1)
          P5=-RO_S*CP_S*(-P(2,J1))/DELT
          F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5)
         ELSE
c comp effective temperature
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    4*EPS(J1)*CON_RF/DR(J1)/DR(J1))
          P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
          P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))*RATE_H1(J,J1)
          P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
```

```
     . S(L4,J-1,J1))/(2*DELZ)
       P5=4*EPS(J1)*CON_RF*(C(1,J1+1))/DR(J1)/DR(J1)
       P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     . (-P(1,J1))/DELT
       F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6)

       F(2)=0.
      END IF
c comp s for the grd at wall
      ELSE IF(J1.EQ.NJR) THEN
c comp gas temperature
       IF(I_SOL_FLO.EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     . (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
       P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     . S(L4,J,J1))/(DELZ)
       P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     . C(1,J1)))/(2*DR(J1))
       P6=CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(3,J1)
     . -C(1,J1)))/DR(J1)/DR(J1)
       P7=-RO_F*CP_F*(-P(1,J1))/DELT
       F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
       A2=1./(2*CON_P/DELZ/DELZ+2*CON_P/DR(J1)/DR(J1)+
     . RO_S*CP_S/DELT)
       P1=AINT*H_FS*(C(1,J1)-C(2,J1))
       P2=-RATE_H1(J,J1)
       P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
       P4=CON_P*RP(J1)*(C(2,J1-1)-C(2,J1-1))/(2*DR(J1))
       P5=CON_P*(2*C(2,J1-1))/DR(J1)/DR(J1)
       P6=-RO_S*CP_S*(-P(2,J1))/DELT
       F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
      ELSE
c compute effective temperaure
       A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     . DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     . EPS(J1)*2*CON_RF/DR(J1)/DR(J1))
       P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
       P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))*RATE_H1(J,J1)
       P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     . S(L4,J,J1))/(DELZ)
       P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(3,J1)-
     . C(1,J1)))/(2*DR(J1))
       P6=EPS(J1)*CON_RF*(2*C(1,J1-1)+2*H_FW*DR(J1)/CON_RF*
     . (C(3,J1)-C(1,J1)))/DR(J1)/DR(J1)
       P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     . (-P(1,J1))/DELT
       F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

       F(2)=0.
      ENDIF
c compute wall temperature
       F(3)=-C(3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(1,J1)-
     . C(3,J1))-A_INS*H_OW*(C(3,J1)-TO))-(-P(3,J1))
c this section computes s for grids of constant size
      ELSE
      IF(J1.NE.IR)THEN
c comp gas temperature
      IF(I_SOL_FLO.EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
```

```
     .    (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
          P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
          P6=CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
          P7=-RO_F*CP_F*(-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c comp solid temperature
          A2=1./(2*CON_P/DELZ/DELZ+2*CON_P/DR(J1)/DR(J1)+
     .    RO_S*CP_S/DELT)
          P1=AINT*H_FS*(C(1,J1)-C(2,J1))
          P2=-RATE_H1(J,J1)
          P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
          P4=CON_P*RP(J1)*(C(2,J1+1)-C(2,J1-1))/(2*DR(J1))
          P5=CON_P*(C(2,J1+1)+C(2,J1-1))/DR(J1)/DR(J1)
          P6=-RO_S*CP_S*(-P(2,J1))/DELT
          F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
         ELSE
c compute effective temperature
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    2*EPS(J1)*CON_RF/DR(J1)/DR(J1))
          P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
          P2=-EPS(J1)*RO_P*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))*RATE_H1(J,J1)
          P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=EPS(J1)*CON_RF*RP(J1)*(C(1,J1+1)-C(1,J1-1))/(2*DR(J1))
          P6=EPS(J1)*CON_RF*(C(1,J1+1)+C(1,J1-1))/DR(J1)/DR(J1)
          P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

          F(2)=0.
         END IF
c this section computes s  for variale grid size
        ELSE
c compute gas temperature
         IF(I_SOL_FLO.EQ.1)THEN
          A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .    1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*
     .    F12(J1)/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(S(L1,J+1,J1)+S(L1,J-1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(1,J1)-C(2,J1))
          P4=-RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4;J-1,J1))/(2*DELZ)
          P5=CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-F01(J1)*
     .    C(1,J1-1))/(DR(J1))
          P6=2*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*C(1,J1-1))/
     .    DR(J1)/DR(J1)
          P7=-RO_F*CP_F*(-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
          A2=1./(2*CON_P/DELZ/DELZ+CON_P*RP(J1)*F11(J1)/DR(J1)+
     .    2*CON_P*F12(J1)/DR(J1)/DR(J1)+RO_S*CP_S/DELT)
          P1=AINT*H_FS*(C(1,J1)-C(2,J1))
          P2=-RATE_H1(J,J1)
          P3=CON_P/DELZ/DELZ*(S(L2,J-1,J1)+S(L2,J+1,J1))
          P4=CON_P*RP(J1)*(F21(J1)*C(2,J1+1)-
     .    F01(J1)*C(2,J1-1))/(DR(J1))
```

```
          P5=2*CON_P*(F22(J1)*C(2,J1+1)+F02(J1)
     .    *C(2,J1-1))/DR(J1)/DR(J1)
          P6=-RO_S*CP_S*(-P(2,J1))/DELT
          F(2)=-C(2,J1)+A2*(P1+P2+P3+P4+P5+P6)
         ELSE
c compute effective temperature
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    EPS(J1)*CON_RF*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/
     .    DR(J1)/DR(J1)))
          P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J+1,J1)+S(L1,J-1,J1))
          P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))*RATE_H1(J,J1)
          P4=-EPS(J1)*RO_F*CP_F*C(1,J1)*(S(L4,J+1,J1)-
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(1,J1+1)-F01(J1)*
     .    C(1,J1-1))/(DR(J1))
          P6=2*EPS(J1)*CON_RF*(F22(J1)*C(1,J1+1)+F02(J1)*
     .    C(1,J1-1))/DR(J1)/DR(J1)
          P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(1,J1))/DELT
          F(1)=-C(1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

          F(2)=0.
         END IF
        END IF
       END IF
      END IF
     END IF
     RETURN
     END
C
C-----------------------------------------------------
C
     SUBROUTINE FUNCT1(J1)
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 K_F(4),K_FO(4),MOL(4),M_AVE,M_W(4),NU_NP,NU_NW,KGAS,
    . K_FP(4,201,201),BC_1(4),D_L(4),Q2(4),MW,A1(14,201),
    . C1(14,201,101),D_RF(4),U(14,201),EPS(1000),E(30),AKT(30),
    . C2(14,201,101),F01(101),F11(101),F21(101),F02(101),
    . F12(101),F22(101),RP(101),DR(101),RR0(101),D_LP_R(4,201,101),
    . D_LP_Z(4,201,101),U1(101,201),YO(4),SS(4)
     COMMON/UG/U_GUESS,EPS
     COMMON/OLD/ AA(14),SUM(14),COLD(14,201,101)
     COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
    . Y(14,14),G(14),F(14),P(14,201)
     COMMON/BND3/S(14,201,101)
     COMMON/NEW_GRID/DELR0(101),ALF1(101),NGR1,NGR2
     COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT,NJ1
     COMMON/PR_OLD_Z/CON_Z(201,101),CON_R(201,101),CON_S(201,101),
    . H_W(201,101),CP_P(201,101),H_FP(201,101),Q(4,201,101),
    . VISC_P(201,101),D_LPP(4,201),RATE_C1(201,101)
     COMMON/PROP_WAL/CP_WA,RO_WA,ERROR0,ERROR1
     COMMON/PROP_B_Z/REC,VOID_B,D_I,D_E,S_B,ALF,GC,Z,M_W
     COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
     COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1,HEAT(4)
     COMMON/INIT_Z/RO_FO,U_FO,U_F1,CP_FO,T_FO,TO,C_FO(4),K_F,P_C(4)
     COMMON/MISC_Z/R,G_F,PI,P_TOT,BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_1
     COMMON/INDIC/L1,L2,L3,L4,L5,L6,LL,M3,II,JJ,IVEL_1
     COMMON/INDIC1/IND(4)
     COMMON/RADIAL/RP
     COMMON/GAS/INERT,NCOMP,MW
     COMMON/NCY/NC1
     COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
```

```
      DATA EMIS,R1,IONCE,JONCE,ICOUNT,ICOND1/.9,555.,0,0,0,1/
c there is a temperature difference between the gas and solid phases
c      IF(LL.EQ.ICOM)THEN
c       BC_L1=TEMPIN(TIME)
c          ICOM=LL+1
c      END IF
      IF(J1.EQ.1 .AND. IONCE.EQ.0)THEN
       IONCE=1
       NJ_OR_1=1
       I_SOL_FLO=0
c an indication number for type of gas to compute the gas conductivity,
c 1=N2, 2=air, 3=co2
       IGAS=1
       DP=R_P*2.
       IF(IVEL_1.EQ.1)THEN
        IR=10
       ELSE
        IR=5
       END IF
c calculating the area for heat transfer to the wall or from the wall
       D_T=D_E+D_I
       X_W=D_E-D_I
       A_C=2*D_I/((D_I+D_E)*X_W)
       A_INS=2*D_E/((D_I+D_E)*X_W)*1.4
c heat transfer coefficient outside of the wall
       H_OW=0.1
      END IF
c call vel_por to compute velosity and porosity profile
      IF(J1.EQ.1 .AND. JONCE.EQ.0)THEN
       JONCE=1
c compute the ratio of distance from the center to each grid over the pellet
c diameter
       RR=0.
       RR0(NJR)=0.
       DO I5=NJR-1,1,-1
        RR=RR+DELR0(I5+1)/D_P
        RR0(I5)=RR
       END DO
c this section calculates the coefficients to the discretized PDE's
c
c compute the inverse of distance from the center to each grid
       RR=0.
       DO I5=1,NJR
        IF(I5.NE.1)THEN
         RR=RR+DELR0(I5)
         RP(I5)=1./RR
        ELSE
         RP(I5)=0.
        END IF
        IF(I5.LT.IR) THEN
         DELR=DELR0(1)
         DR(I5)=DELR
        ELSE IF(I5.eq.IR)THEN
         DELR1=DELR0(IR)
         DELR2=DELR0(IR+1)
         DELR=DELR1
         DR(I5)=DELR
         FR=DELR2/DELR1
c these variables are bein computed at the pont where two adjant grids are
c not the same size
         F01(I5)=FR/(1+FR)
         F11(I5)=(1.-FR)/FR
         F21(I5)=1./((1.+FR)*FR)
         F02(I5)=1./(1+FR)
```

```
         F12(I5)=1./FR
         F22(I5)=1./((1.+FR)*FR)
        ELSE
         DELR=DELR0(IR+1)
         DR(I5)=DELR
        END IF
       END DO
      END IF
c this section calls the proper subroutine to compute the parameters such as
c diffusivity, conductivity.....
      IF(J1.EQ.1 .AND. ICOUNT.LT.JJ)THEN
       I5=JJ
       ICOUNT=ICOUNT+1
        DO I6=1,NJR
         DO M1=1,NC
          YO(M1)=C(M1,I6)*R1*C(L1,I6)/S(L5,I5,I6)
         END DO
         TEMP=C(L1,I6)
         M_AVE=0
         SUM_Y=0.
         DO M1=1,NC
          M_AVE=M_AVE+YO(M1)*M_W(M1)
          SUM_Y=SUM_Y+YO(M1)
         END DO
         M_AVE=M_AVE+(1-SUM_Y)*28.
         RO_F=S(L5,I5,I6)/(C(L1,I6)*R1)
         IF(RO_F.LE.0)RO_F=P_TOT/(C(L1,I6)*R1)
         GN2=GN
         GN=S(L4,I5,I6)*RO_F/60.*VOID_B
         IF(GN.EQ.0)GN=GN2
c calculate Reynolds number
         REY=RE(GN,TEMP)
c calculate specific heat of gas phase
         CP_P(I5,I6)=CPGAS(TEMP,YO)
c calc viscosity
         VISC_P(I5,I6)=VIS(TEMP)*60.
c calc heat transfer coeff
         H_FP(I5,I6)=HFILM(MW,TEMP,REY,CP_P(I5,I6))*60.
         itest11=0
         IF(ITEST11.EQ.0) MN=2
c this section compute the conductivity of gas and solid in axial and
cradial directions
         CALL PHI_COND(IGAS,TEMP,CON_S1,EPS(1),PHI,PHIW)
         CALL CONDU(MN,IGAS,PHI,D_P,CON_S1,RR0(I6),U1(1,I6),REY,
     .   TEMP,EPS(I6),EPS(1),EFF_CON_R1,CON_R1,EFF_CON_Z1,
     .   CON_Z1,CON_P1,NJR,NGR1,NGR2,I6)
c if there is a tepmperature difference between the gas and solid
         IF(I_SOL_FLO.EQ.1)THEN
          CON_Z(I5,I6)=CON_Z1
          CON_S(I5,I6)=CON_P1
c if there is no tepmperature difference between the gas and solid
         ELSE
          CON_Z(I5,I6)=EFF_CON_Z1
         END IF
c computation for number of component
         DO M1=1,NC
c calc the molecular diffusivity of eac componet in the mixture
          DIF=DIFF(M1,TEMP,S(L5,I5,I6),YO)*60
c calc Schmidt number
          SC_N=VISC_P(I5,I6)/DIF/RO_F/M_AVE
c calc axial diffusion
C           CALL AXDIFF_R_Z(MN,IGAS,PHI,D_P,DIF,RR0(I6),U1(1,I6),
C   .       REY,SC_N,TEMP,EPS(I6),EPS(1),EFF_DIFF_R,EFF_DIFF_Z,
C   .       njr,ngr1,NGR2,I6)
```

```
C           D_LP_R(M1,I5,I6)=EFF_DIFF_R
C           D_LP_Z(M1,I5,I6)=EFF_DIFF_Z
          D_LP_Z(M1,I5,I6)=CON_Z(I5,I6)/RO_F/CP_P(I5,I6)
         END DO
c calc the heat transfer coefficint between the gas phase and the wall
         IF(I6.EQ.NJR) THEN
          IF(GN.EQ.0)GN=S(L4,1,1)*RO_F/60.*VOID_B
          REY=RE(GN,TEMP)
          CALL HEAT_WALL(IGAS,PHIW,D_P,CON_S1,RR0(I6),U1(1,I6),
     .    REY,TEMP,VOID_B,H_W1)
          H_W(I5,NJR)=H_W1
         END IF
         IF(ICOND1.EQ.1)THEN
          VAV=G_F/D_I/D_I/PI*4
          GA=G_F/D_I/D_I/PI*4.*RO_F
          TAV=S(L1,1,1)/1.8
          RO=S(L5,1,1)/S(L1,1,1)/R1
          CP=CP_P(1,1)
          AVIS=VISC_P(1,1)
          AKC=.000358895+3.0026706379E-5*S(L1,1,1)-5.3528942E-9*
     1     S(L1,1,1)*S(L1,1,1)
          CALL CONDUC1(EMIS,CON_S1,D_P,D_I,GA,VAV,TAV,RO,CP,AVIS,
     1     AKC,NJR,EPSEX,EPS(1),EPS(NJR),DELR0,E,AKT)
          ICOND1=0
         END IF
         CON_R(I5,I6)=AKT(I6)
         IF(E(I6).LE.0)E(I6)=E(I6-1)
         DO M1=1,NC
C           D_LP_R(M1,I5,I6)=AKT(I6)/RO/CP
          D_LP_R(M1,I5,I6)=E(I6)
         END DO
        END DO
        IF(CON_Z(I5,NJR).EQ.0)THEN
         CON_Z(I5,NJR)=CON_Z(I5,NJR-1)
         DO M1=1,NC
           D_LP_Z(M1,I5,NJR)=CON_Z(I5,NJR)/RO_F/CP_P(I5,NJR)
         END DO
        END IF
        DO I1=1,NJ
         DO M1=1,NC
          D_LPP(M1,I1)=D_LP_Z(M1,I1,1)
         END DO
        END DO
      END IF
c set the parameters to  new variables
      J=JJ
      H_FW=H_W(J,NJR)
      RO_F=S(L5,J,J1)/(C(L1,J1)*R1)
      CP_F=CP_P(J,J1)
      H_FS=H_FP(J,J1)
      CON_L=CON_Z(J,J1)
      DO M1=1,NC
       D_L(M1)=D_LP_Z(M1,J,J1)
      END DO
      DO M1=1,NC
       D_RF(M1)=D_LP_R(M1,J,J1)
      END DO
      CON_RF=CON_R(J,J1)
      CON_P=CON_S(J,J1)
      KK=0
      D_L_AVE=0.
      DO M1=1,NC
       IF(D_L(M1).GT.0)THEN
        D_L_AVE=D_L_AVE+D_L(M1)
```

```
          KK=KK+1
        END IF
      END DO
      D_L_AVE=D_L_AVE/KK
      VISC_F=VISC_P(J,J1)
      DO I1=1,NC
        SS(I1)=C(I1,J1)
      END DO
c call the isotherm to compute the equilibruim concentration
      CALL IST_Z(1,C(L2,J1),SS,Q2)
c based the equilibruim per unit volume of pellet
      DO I=1,NC
        Q(I,J,J1)=Q2(I)*RO_S
      END DO
c this is the boundry, the first grids in axial direction
        IF(J.EQ.1)THEN
          RATE_C=0
c compute the total amount is being adsorbed
          DO M1=1,NC
            RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
          END DO
          RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
          RATE_H=0
c compute the total heat of adsorption
C           HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C   .       C(3,J1)**2/RO_S/RO_S)*44
          DO M1=1,NC
            RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
   .        C(M1+NC,J1))
          END DO
c this "if" calculates the first grid in radial direction, center of the bed
         IF(J1.EQ.1) THEN
          TOT_C=0.
c computation of molar density component
          DO M1=1,NC-1
           A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
   .       DR(J1)/DR(J1)+1./DELT)
           P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
           P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
           P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
           P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
           P5=-(-P(M1,J1))/DELT
           F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
           TOT_C=TOT_C+C(M1,J1)
          END DO
          F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
   .      C(NC,J1)
c computation of molar adsorption for each component
          DO M1=1,NC
           A2=1./(K_F(M1)*AINT+1./DELT)
           F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
   .       P(M1+NC,J1)/DELT)
          END DO
c computation of gas temperature
          IF(I_SOL_FLO .EQ.1)THEN
           A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
   .       1./DELT)+4*CON_RF/DR(J1)/DR(J1))
           P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
           P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
           P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
           P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
   .       S(L4,J,J1))/(DELZ)
           P5=4*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
           P6=-RO_F*CP_F*(-P(L1,J1))/DELT
```

```
      F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c computation of solid temperature
      A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     .      4*CON_P/DR(J1)/DR(J1))
      P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
      P2=-RATE_H
      P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
      P4=4*CON_P*(C(L2,J1+1))/DR(J1)/DR(J1)
      P5=-RO_S*CP_S*(-P(L2,J1))/DELT
      F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5)
c computation of effective temperature
      ELSE
      A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .      DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .      4*EPS(J1)*CON_RF/DR(J1)/DR(J1))
      P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
      P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
      P3=-(1-EPS(J1))*RATE_H
      P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .      S(L4,J,J1))/(DELZ)
      P5=4*EPS(J1)*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
      P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .      (-P(L1,J1))/DELT
      F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)

      F(L2)=C(L2,J1)-BC_L1
      END IF
c the same computation as above except it is now for the grid at the wall
          ELSE IF(J1.EQ.NJR)THEN
      TOT_C=0.
c comp of molar density of component
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
     .      DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
       P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
       P6=P(M1,J1)/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .      C(NC,J1)
c comp the molar  adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .       P(M1+NC,J1)/DELT)
      END DO
c comp of gas phase temperature
      IF(I_SOL_FLO .EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .      (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
       P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
       P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .      S(L4,J,J1))/(DELZ)
       P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(L3,J1)-
     .      C(L1,J1)))/(2*DR(J1))
       P6=CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(L3,J1)-
     .      C(L1,J1)))/DR(J1)/DR(J1)
```

```
        P7=-RO_F*CP_F*(-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c comp of solid phase temperature
        A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     .  2*CON_P/DR(J1)/DR(J1))
        P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
        P2=-RATE_H
        P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
        P4=CON_P*RP(J1)*(C(L2,J1-1)-C(L2,J1-1))/(2*DR(J1))
        P5=CON_P*(2*C(L2,J1-1))/DR(J1)/DR(J1)
        P6=-RO_S*CP_S*(-P(L2,J1))/DELT
        F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
c comp of effective temperature
       ELSE
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2.*
     .  EPS(J1)*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
        P3=-(1-EPS(J1))*RATE_H
        P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .  S(L4,J,J1))/(DELZ)
        P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*
     .  (C(L3,J1)-C(L1,J1)))/(2*DR(J1))
        P6=EPS(J1)*CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/
     .  CON_RF*(C(L3,J1)-C(L1,J1)))/DR(J1)/DR(J1)
        P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .  (-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

        F(L2)=C(L2,J1)-BC_L1
       END IF
        F(L3)=-C(L3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(L1,J1)-
     .  C(L3,J1))-A_INS*H_OW*(C(L3,J1)-TO))-(-P(L3,J1))
           ELSE
c the same computation as above except it is now for the grids from the center
c to the pont where the velocity start to chane
       IF(J1.NE.IR)THEN
       TOT_C=0.
c comp molar density of each component
       DO  M1=1,NC-1
        A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/DR(J1)
     .  /DR(J1)+1./DELT)
        P1=D_L(M1)/DELZ**2*(BC_1(M1)+S(M1,J+1,J1))
        P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
        P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
        P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
        P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/
     .  DR(J1)/DR(J1)
        P6=-(-P(M1,J1))/DELT
        F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
        TOT_C=TOT_C+C(M1,J1)
       END DO
       F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute molar adsorption
       DO M1=1,NC
        A2=1./(K_F(M1)*AINT+1./DELT)
        F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .  P(M1+NC,J1)/DELT)
       END DO
c compute gas temperature
       IF(I_SOL_FLO .EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
```

```
   .    (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
        P1=CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
        P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
        P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
        P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
   .    S(L4,J,J1))/(DELZ)
        P5=CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/(2*DR(J1))
        P6=CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/DR(J1)
        P7=-RO_F*CP_F*(-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
        A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
   .    2*CON_P/DR(J1)/DR(J1))
        P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
        P2=-RATE_H
        P3=CON_P*(S(L2,J+1,J1)+S(L2,J+1,J1))/DELZ/DELZ
        P4=CON_P*RP(J1)*(C(L2,J1+1)-C(L2,J1-1))/(2*DR(J1))
        P5=CON_P*(C(L2,J1+1)+C(L2,J1-1))/DR(J1)/DR(J1)
        P6=-RO_S*CP_S*(-P(L2,J1))/DELT
        F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
c compute effective  temperature
        ELSE
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
   .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
   .    EPS(J1)*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(BC_L1+S(L1,J+1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
        P3=-(1-EPS(J1))*RATE_H
        P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-S(L4,J,J1))/
   .    (DELZ)
        P5=EPS(J1)*CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/
   .    (2*DR(J1))
        P6=EPS(J1)*CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/DR(J1)
        P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
   .    (-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

        F(L2)=C(L2,J1)-BC_L1
        END IF
c the same computation as above except it is now for the grids from the point
c where the velocity starts to change to the wall
        ELSE
        TOT_C=0.
c compute molar density of each component
        DO M1=1,NC-1
         A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
   .     RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
         P1=D_L(M1)/DELZ**2*(S(M1,J+1,J1)+BC_1(M1))
         P2=-S(L4,J,J1)/(DELZ)*(-BC_1(M1))
         P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
         P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
   .     C(M1,J1-1))/(DR(J1))
         P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
   .     /DR(J1)/DR(J1)
         P6=-(-P(M1,J1))/DELT
         F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
         TOT_C=TOT_C+C(M1,J1)
        END DO
        F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
   .    C(NC,J1)
c compute molar  adsorption
        DO M1=1,NC
         A2=1./(K_F(M1)*AINT+1./DELT)
         F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
```

```
    .  P(M1+NC,J1)/DELT)
      END DO
c compute the gas temperature
      IF(I_SOL_FLO .EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
    .  1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*F12(J1)/
    .  DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(S(L1,J+1,J1)+BC_L1)
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
       P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
       P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
    .  S(L4,J,J1))/(DELZ)
       P5=CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-
    .  F01(J1)*C(L1,J1-1))/(DR(J1))
       P6=2*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*C(L1,J1-1))/
    .  DR(J1)/DR(J1)
       P7=-RO_F*CP_F*(-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute the solid temperature
       A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
    .  CON_P*RP(J1)*F11(J1)/DR(J1)+2*CON_P*F12(J1)/DR(J1)/DR(J1))
       P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
       P2=-RATE_H
       P3=CON_P/DELZ/DELZ*(S(L2,J+1,J1)+S(L2,J+1,J1))
       P4=CON_P*RP(J1)*(F21(J1)*C(L2,J1+1)-F01(J1)*
    .  C(L2,J1-1))/(DR(J1))
       P5=2*CON_P*(F22(J1)*C(L2,J1+1)+F02(J1)*C(L2,J1-1))/
    .  DR(J1)/DR(J1)
       P6=-RO_S*CP_S*(-P(L2,J1))/DELT
       F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
      ELSE
c compute the effective temperature
       A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
    .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
    .  EPS(J1)*CON_RF*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/DR(J1)
    .  /DR(J1)))
       P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J+1,J1)+BC_L1)
       P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-BC_L1)
       P3=-(1-EPS(J1))*RATE_H
       P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
    .  S(L4,J,J1))/(DELZ)
       P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-F01(J1)*
    .  C(L1,J1-1))/(DR(J1))
       P6=2*EPS(J1)*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*
    .  C(L1,J1-1))/DR(J1)/DR(J1)
       P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
    .  (-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

       F(L2)=C(L2,J1)-BC_L1
      END IF
      END IF

          ENDIF
C
C-----------------------------------------------------------
C this section compute the S matrix variables for grid at z=L, the outflow
c boundry
c
c
     ELSE IF(J.EQ.NJ1)THEN
c compute the total amount adsorbed
      RATE_C=0
      DO M1=1,NC
```

```
        RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
       END DO
c compute the total heat of adsorption
       RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
       RATE_H=0
C        HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C    .   C(3,J1)**2/RO_S/RO_S)*44
       DO M1=1,NC
        RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
     .  C(M1+NC,J1))
      END DO
c this section compute the s matrix at outflow boundry at the ceter of bed
      IF(J1.EQ.1) THEN
       TOT_C=0.
c compute the component molar density
       DO M1=1,NC-1
        A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
     .  DR(J1)/DR(J1)+1./DELT)
        P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
        P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
        P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
        P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
        P5=-(-P(M1,J1))/DELT
        F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
        TOT_C=TOT_C+C(M1,J1)
       END DO
       F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute the molar adsorption
       DO M1=1,NC
        A2=1./(K_F(M1)*AINT+1./DELT)
        F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .  P(M1+NC,J1)/DELT)
       END DO
c compute the gas temperature
       IF(I_SOL_FLO .EQ.1)THEN
        A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .  (RO_F*CP_F)/DELT+4*CON_RF/DR(J1)/DR(J1))
        P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
        P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     .  S(L4,J-1,J1))/(2*DELZ)
        P5=4*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
        P6=-RO_F*CP_F*(-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c compute the solid temperaure
        A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     .  4*CON_P/DR(J1)/DR(J1))
        P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
        P2=-RATE_H
        P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
        P4=4*CON_P*C(L2,J1+1)/DR(J1)/DR(J1)
        P5=-RO_S*CP_S*(-P(L2,J1))/DELT
        F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5)
       ELSE
c compute the effective temperature
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .  EPS(J1)*4*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))*RATE_H
        P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
```

```
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=4*EPS(J1)*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
          P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(L1,J1))/DELT
          F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)

          F(L2)=C(L2,J1)-BC_L1
         END IF
c this section is for r at the wall
        ELSE IF(J1.EQ.NJR) THEN
         TOT_C=0.
c compute molar density
         DO M1=1,NC-1
           A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
     .     DR(J1)/DR(J1)+1./DELT)
           P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
           P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
           P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
           P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
           P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
           P6=-(-P(M1,J1))/DELT
           F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
           TOT_C=TOT_C+C(M1,J1)
         END DO
         F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .   C(NC,J1)
c compute molar adsorption
         DO M1=1,NC
           A2=1./(K_F(M1)*AINT+1./DELT)
           F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .     P(M1+NC,J1)/DELT)
         END DO
c compute gas temperature
         IF(I_SOL_FLO .EQ.1)THEN
          A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .    (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
          P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
          P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
          P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
          P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     .    S(L4,J-1,J1))/(2*DELZ)
          P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(L3,J1)-
     .    C(L1,J1)))/(2*DR(J1))
          P6=CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(L3,J1)-
     .    C(L1,J1)))/DR(J1)/DR(J1)
          P7=-RO_F*CP_F*(-P(L1,J1))/DELT
          F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
          A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     .    2*CON_P/DR(J1)/DR(J1))
          P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
          P2=-RATE_H
          P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
          P4=CON_P*RP(J1)*(C(L2,J1-1)-C(L2,J1-1))/(2*DR(J1))
          P5=CON_P*(2*C(L2,J1-1))/DR(J1)/DR(J1)
          P6=-RO_S*CP_S*(-P(L2,J1))/DELT
          F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
         ELSE
c compute effective temperature
          A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .    DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .    2*EPS(J1)*CON_RF/DR(J1)/DR(J1))
          P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
          P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
```

```
       P3=-(1-EPS(J1))*RATE_H
       P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
       P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*
     . (C(L3,J1)-C(L1,J1)))/(2*DR(J1))
       P6=EPS(J1)*CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/CON_RF*
     . (C(L3,J1)-C(L1,J1)))/DR(J1)/DR(J1)
       P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     . (-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

       F(L2)=C(L2,J1)-BC_L1
      END IF
c compute the wall temperature
       F(L3)=-C(L3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(L1,J1)-
     . C(L3,J1))-A_INS*H_OW*(C(L3,J1)-TO))-(-P(L3,J1))
      ELSE
c this section computes S matrix for those grids with constant grid siz
      IF(J1.NE.IR)THEN
      TOT_C=0.
c compute molar density
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/
     . DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
       P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J
c compute molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
c compute gas temperature
      IF(I_SOL_FLO .EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     . (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
       P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
       P5=CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/(2*DR(J1))
       P6=CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/
     . DR(J1)
       P7=-RO_F*CP_F*(-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
       A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
     . 2*CON_P/DR(J1)/DR(J1))
       P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
       P2=-RATE_H
       P3=CON_P*(S(L2,J-1,J1)+S(L2,J-1,J1))/DELZ/DELZ
       P4=CON_P*RP(J1)*(C(L2,J1+1)-C(L2,J1-1))/(2*DR(J1))
       P5=CON_P*(C(L2,J1+1)+C(L2,J1-1))/DR(J1)/DR(J1)
       P6=-RO_S*CP_S*(-P(L2,J1))/DELT
```

```
        F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
       ELSE
c compute effective temperature
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
     .  EPS(J1)*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))*RATE_H
        P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     .  S(L4,J-1,J1))/(2*DELZ)
        P5=EPS(J1)*CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/(2*
     .  DR(J1))
        P6=EPS(J1)*CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/DR(J1)
        P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .  (-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

        F(L2)=C(L2,J1)-BC_L1
       END IF
c this section compute s for variable grid size
       ELSE
       TOT_C=0.
c compute the molar density
       DO M1=1,NC-1
        A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
     .  RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
        P1=D_L(M1)/DELZ**2*(2*S(M1,J-1,J1))
        P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
        P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
        P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
     .  C(M1,J1-1))/(DR(J1))
        P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
    . . /DR(J1)/DR(J1)
        P6=-(-P(M1,J1))/DELT
        F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
        TOT_C=TOT_C+C(M1,J1)
       END DO
       F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c compute molar adsorption
       DO M1=1,NC
        A2=1./(K_F(M1)*AINT+1./DELT)
        F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .  P(M1+NC,J1)/DELT)
       END DO
c compute the gas temperature
       IF(I_SOL_FLO.EQ.1)THEN
        A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .  1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*
     .  F12(J1)/DR(J1)/DR(J1))
        P1=CON_L/DELZ**2*(2*S(L1,J-1,J1))
        P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
        P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     .  S(L4,J-1,J1))/(2*DELZ)
        P5=CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-F01(J1)*
     .  C(L1,J1-1))/(DR(J1))
        P6=2*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*C(L1,J1-1))/
     .  DR(J1)/DR(J1)
        P7=-RO_F*CP_F*(-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute the solid temperature
        A2=1./(2*CON_P/DELZ**2+(RO_S*CP_S)/DELT+
```

```
     .   CON_P*RP(J1)*F11(J1)/DR(J1)+2*CON_P*F12(J1)/DR(J1)/DR(J1))
         P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
         P2=-RATE_H
         P3=CON_P/DELZ/DELZ*(S(L2,J-1,J1)+S(L2,J-1,J1))
         P4=CON_P*RP(J1)*(F21(J1)*C(L2,J1+1)-F01(J1)*C(L2,J1-1))
     .   /(DR(J1))
         P5=2*CON_P*(F22(J1)*C(L2,J1+1)+F02(J1)*C(L2,J1-1))/
     .   DR(J1)/DR(J1)
         P6=-RO_S*CP_S*(-P(L2,J1))/DELT
         F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
        ELSE
c compute the effective temperature
         A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .   DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .   CON_RF*EPS(J1)*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/DR(J1)/
     .   DR(J1)))
         P1=EPS(J1)*CON_L/DELZ**2*(2*S(L1,J-1,J1))
         P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
         P3=-(1-EPS(J1))*RATE_H
         P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J-1,J1)-
     .   S(L4,J-1,J1))/(2*DELZ)
         P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-F01(J1)*
     .   C(L1,J1-1))/(DR(J1))
         P6=2*EPS(J1)*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*
     .   C(L1,J1-1))/DR(J1)/DR(J1)
         P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .   (-P(L1,J1))/DELT
         F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

         F(L2)=C(L2,J1)-BC_L1
        END IF
        END IF
       END IF

C
C---------------------------------------------------------
C
c this section computes the s matrix for grids inside of the column
c
       ELSE
c comp total molar adsorption
         RATE_C=0
         DO M1=1,NC
          RATE_C=RATE_C+K_F(M1)*AINT*(Q(M1,J,J1)-C(M1+NC,J1))
         END DO
c compute total heat of adsorption
         RATE_C1(J,J1)=RATE_C*(1-EPS(J1))
         RATE_H=0
C         HEAT(2)=-(411.93585-57470.2733*C(3,J1)/RO_S+3971259.2258*
C    .    C(3,J1)**2/RO_S/RO_S)*44
         DO M1=1,NC
          RATE_H=RATE_H+HEAT(M1)*K_F(M1)*AINT*(Q(M1,J,J1)-
     .    C(M1+NC,J1))
        END DO
c this section copm s for the grids at the center of bed
        IF(J1.EQ.1) THEN
         TOT_C=0.
c comp molar density
         DO M1=1,NC-1
          A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+4*D_RF(M1)/
     .    DR(J1)/DR(J1)+1./DELT)
          P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
          P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
          P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
```

```
         P4=4*D_RF(M1)*(C(M1,J1+1))/DR(J1)/DR(J1)
         P5=-(-P(M1,J1))/DELT
         F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5)
         TOT_C=TOT_C+C(M1,J1)
        END DO
        F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .  C(NC,J1)
c comp molar adsorption
        DO M1=1,NC
         A2=1./(K_F(M1)*AINT+1./DELT)
         F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .   P(M1+NC,J1)/DELT)
        END DO
c comp gas temperature
        IF(I_SOL_FLO.EQ.1)THEN
         A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .   (RO_F*CP_F)/DELT+4*CON_RF/DR(J1)/DR(J1))
         P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
         P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
         P3=-(1-EPS(J1))/EPS(J1)**H_FS*AINT*(C(L1,J1)-C(L2,J1))
         P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J-1,J1))/(2*DELZ)
         P5=4*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
         P6=-RO_F*CP_F*(-P(L1,J1))/DELT
         F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)
c comp solid temperature
         A2=1./(2*CON_P/DELZ/DELZ+4*CON_P/DR(J1)/DR(J1)+
     .   RO_S*CP_S/DELT)
         P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
         P2=-RATE_H
         P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
         P4=4*CON_P*(C(L2,J1+1))/DR(J1)/DR(J1)
         P5=-RO_S*CP_S*(-P(L2,J1))/DELT
         F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5)
        ELSE
c comp effective temperature
         A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .   DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .   4*EPS(J1)*CON_RF/DR(J1)/DR(J1))
         P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
         P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
         P3=-(1-EPS(J1))*RATE_H
         P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J-1,J1))/(2*DELZ)
         P5=4*EPS(J1)*CON_RF*(C(L1,J1+1))/DR(J1)/DR(J1)
         P6=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .   (-P(L1,J1))/DELT
         F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6)

         F(L2)=C(L2,J1)-BC_L1
        END IF
c comp s for the grd at wall
       ELSE IF(J1.EQ.NJR) THEN
        TOT_C=0.
c comp molar density
        DO M1=1,NC-1
         A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2*D_RF(M1)/DR(J1)
     .   /DR(J1)+1./DELT)
         P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
         P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
         P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
         P4=D_RF(M1)*RP(J1)*(C(M1,J1-1)-C(M1,J1-1))/(2*DR(J1))
         P5=2*D_RF(M1)*(C(M1,J1-1))/DR(J1)/DR(J1)
         P6=-(-P(M1,J1))/DELT
```

```
        F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
        TOT_C=TOT_C+C(M1,J1)
       END DO
       F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c comp molar adsorption
       DO M1=1,NC
        A2=1./(K_F(M1)*AINT+1./DELT)
        F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .  P(M1+NC,J1)/DELT)
       END DO
c comp gas temperature
       IF(I_SOL_FLO.EQ.1)THEN
        A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     .  (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
        P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
        P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
        P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .  S(L4,J,J1))/(DELZ)
        P5=CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*(C(L3,J1)-
     .  C(L1,J1)))/(2*DR(J1))
        P6=CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/CON_RF*(C(L3,J1)
     .  -C(L1,J1)))/DR(J1)/DR(J1)
        P7=-RO_F*CP_F*(-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
        A2=1./(2*CON_P/DELZ/DELZ+2*CON_P/DR(J1)/DR(J1)+
     .  RO_S*CP_S/DELT)
        P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
        P2=-RATE_H
        P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
        P4=CON_P*RP(J1)*(C(L2,J1-1)-C(L2,J1-1))/(2*DR(J1))
        P5=CON_P*(2*C(L2,J1-1))/DR(J1)/DR(J1)
        P6=-RO_S*CP_S*(-P(L2,J1))/DELT
        F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
       ELSE
c compute effective temperaure
        A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .  DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
     .  EPS(J1)*CON_RF/DR(J1)/DR(J1))
        P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
        P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
        P3=-(1-EPS(J1))*RATE_H
        P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .  S(L4,J,J1))/(DELZ)
        P5=EPS(J1)*CON_RF*RP(J1)*(2.*H_FW*DR(J1)/CON_RF*
     .  (C(L3,J1)-C(L1,J1)))/(2*DR(J1))
        P6=EPS(J1)*CON_RF*(2*C(L1,J1-1)+2*H_FW*DR(J1)/CON_RF*
     .  (C(L3,J1)-C(L1,J1)))/DR(J1)/DR(J1)
        P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .  (-P(L1,J1))/DELT
        F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

        F(L2)=C(L2,J1)-BC_L1
       ENDIF
c compute wall temperature
        F(L3)=-C(L3,J1)+DELT/RO_WA/CP_WA*(A_C*H_FW*(C(L1,J1)-
     .  C(L3,J1))-A_INS*H_OW*(C(L3,J1)-TO))-(-P(L3,J1))
c this section computes s for grids of constant size
      ELSE
      IF(J1.NE.IR)THEN
       TOT_C=0.
c comp the molar density
```

```
      DO M1=1,NC-1
       A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+2.*D_RF(M1)/
     . DR(J1)/DR(J1)+1./DELT)
       P1=D_L(M1)/DELZ**2*(S(M1,J-1,J1)+S(M1,J+1,J1))
       P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
       P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
       P4=D_RF(M1)*RP(J1)*(C(M1,J1+1)-C(M1,J1-1))/(2*DR(J1))
       P5=D_RF(M1)*(C(M1,J1+1)+C(M1,J1-1))/DR(J1)/DR(J1)
       P6=-(-P(M1,J1))/DELT
       F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
       TOT_C=TOT_C+C(M1,J1)
      END DO
      F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     . C(NC,J1)
c comp the molar adsorption
      DO M1=1,NC
       A2=1./(K_F(M1)*AINT+1./DELT)
       F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     . P(M1+NC,J1)/DELT)
      END DO
c comp gas temperature
      IF(I_SOL_FLO.EQ.1)THEN
       A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/DELZ+
     . (RO_F*CP_F)/DELT+2*CON_RF/DR(J1)/DR(J1))
       P1=CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
       P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))/EPS(J1)**H_FS*AINT*(C(L1,J1)-C(L2,J1))
       P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
       P5=CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/(2*DR(J1))
       P6=CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/DR(J1)
       P7=-RO_F*CP_F*(-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c comp solid temperature.
       A2=1./(2*CON_P/DELZ/DELZ+2*CON_P/DR(J1)/DR(J1)+
     . RO_S*CP_S/DELT)
       P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
       P2=-RATE_H
       P3=CON_P*(S(L2,J-1,J1)+S(L2,J+1,J1))/DELZ/DELZ
       P4=CON_P*RP(J1)*(C(L2,J1+1)-C(L2,J1-1))/(2*DR(J1))
       P5=CON_P*(C(L2,J1+1)+C(L2,J1-1))/DR(J1)/DR(J1)
       P6=-RO_S*CP_S*(-P(L2,J1))/DELT
       F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
      ELSE
c compute effective temperature
       A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     . DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+2*
     . EPS(J1)*CON_RF/DR(J1)/DR(J1))
       P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J-1,J1)+S(L1,J+1,J1))
       P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
       P3=-(1-EPS(J1))*RATE_H
       P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     . S(L4,J-1,J1))/(2*DELZ)
       P5=EPS(J1)*CON_RF*RP(J1)*(C(L1,J1+1)-C(L1,J1-1))/(2*
     . DR(J1))
       P6=EPS(J1)*CON_RF*(C(L1,J1+1)+C(L1,J1-1))/DR(J1)/DR(J1)
       P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     . (-P(L1,J1))/DELT
       F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

       F(L2)=C(L2,J1)-BC_L1
      END IF
c this section computes s  for variale grid size
      ELSE
```

```
        TOT_C=0.
c comp molar density
        DO M1=1,NC-1
         A2=1./(2*D_L(M1)/DELZ**2+S(L4,J,J1)/DELZ+F11(J1)*D_RF(M1)*
     .   RP(J1)/DR(J1)+2*D_RF(M1)*F12(J1)/DR(J1)/DR(J1)+1./DELT)
         P1=D_L(M1)/DELZ**2*(S(M1,J+1,J1)+S(M1,J-1,J1))
         P2=-S(L4,J,J1)/(DELZ)*(-S(M1,J-1,J1))
         P3=-ALF1(J1)*AINT*K_F(M1)*(Q(M1,J,J1)-C(M1+NC,J1))
         P4=D_RF(M1)*RP(J1)*(F21(J1)*C(M1,J1+1)-F01(J1)*
     .   C(M1,J1-1))/(DR(J1))
         P5=2*D_RF(M1)*(F22(J1)*C(M1,J1+1)+F02(J1)*C(M1,J1-1))
     .   /DR(J1)/DR(J1)
         P6=-(-P(M1,J1))/DELT
         F(M1)=-C(M1,J1)+A2*(P1+P2+P3+P4+P5+P6)
         TOT_C=TOT_C+C(M1,J1)
        END DO
        F(NC)=(S(L5,J,J1)-TOT_C*R1*S(L1,J,J1))/R1/S(L1,J,J1)-
     .  C(NC,J1)
c comp molar adsorption
        DO M1=1,NC
         A2=1./(K_F(M1)*AINT+1./DELT)
         F(M1+NC)=-C(M1+NC,J1)+A2*(K_F(M1)*AINT*(Q(M1,J,J1))+
     .   P(M1+NC,J1)/DELT)
        END DO
c compute gas temperature
        IF(I_SOL_FLO.EQ.1)THEN
         A2=1./(2*CON_L/DELZ**2+RO_F*CP_F*(S(L4,J,J1)/DELZ+
     .   1./DELT)+CON_RF*RP(J1)*F11(J1)/DR(J1)+2*CON_RF*
     .   F12(J1)/DR(J1)/DR(J1))
         P1=CON_L/DELZ**2*(S(L1,J+1,J1)+S(L1,J-1,J1))
         P2=-RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
         P3=-(1-EPS(J1))/EPS(J1)*H_FS*AINT*(C(L1,J1)-C(L2,J1))
         P4=-RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J-1,J1))/(2*DELZ)
         P5=CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-F01(J1)*
     .   C(L1,J1-1))/(DR(J1))
         P6=2*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*C(L1,J1-1))/
     .   DR(J1)/DR(J1)
         P7=-RO_F*CP_F*(-P(L1,J1))/DELT
         F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)
c compute solid temperature
         A2=1./(2*CON_P/DELZ/DELZ+CON_P*RP(J1)*F11(J1)/DR(J1)+
     .   2*CON_P*F12(J1)/DR(J1)/DR(J1)+RO_S*CP_S/DELT)
         P1=AINT*H_FS*(C(L1,J1)-C(L2,J1))
         P2=-RATE_H
         P3=CON_P/DELZ/DELZ*(S(L2,J-1,J1)+S(L2,J+1,J1))
         P4=CON_P*RP(J1)*(F21(J1)*C(L2,J1+1)-
     .   F01(J1)*C(L2,J1-1))/(DR(J1))
         P5=2*CON_P*(F22(J1)*C(L2,J1+1)+F02(J1)
     .   *C(L2,J1-1))/DR(J1)/DR(J1)
         P6=-RO_S*CP_S*(-P(L2,J1))/DELT
         F(L2)=-C(L2,J1)+A2*(P1+P2+P3+P4+P5+P6)
        ELSE
c compute effective temperature
         A2=1./(EPS(J1)*(2*CON_L/DELZ**2+RO_F*CP_F*S(L4,J,J1)/
     .   DELZ)+(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)/DELT+
     .   EPS(J1)*CON_RF*(RP(J1)*F11(J1)/DR(J1)+2*F12(J1)/DR(J1)
     .   /DR(J1)))
         P1=EPS(J1)*CON_L/DELZ**2*(S(L1,J+1,J1)+S(L1,J-1,J1))
         P2=-EPS(J1)*RO_F*CP_F*S(L4,J,J1)/(DELZ)*(-S(L1,J-1,J1))
         P3=-(1-EPS(J1))*RATE_H
         P4=-EPS(J1)*RO_F*CP_F*C(L1,J1)*(S(L4,J+1,J1)-
     .   S(L4,J-1,J1))/(2*DELZ)
         P5=EPS(J1)*CON_RF*RP(J1)*(F21(J1)*C(L1,J1+1)-F01(J1)*
```

```
     .    C(L1,J1-1))/(DR(J1))
          P6=2*EPS(J1)*CON_RF*(F22(J1)*C(L1,J1+1)+F02(J1)*
     .    C(L1,J1-1))/DR(J1)/DR(J1)
          P7=-(EPS(J1)*RO_F*CP_F+(1-EPS(J1))*RO_S*CP_S)*
     .    (-P(L1,J1))/DELT
          F(L1)=-C(L1,J1)+A2*(P1+P2+P3+P4+P5+P6+P7)

         F(L2)=C(L2,J1)-BC_L1
         END IF
        END IF
        END IF
       END IF
      RETURN
      END
C
C-----------------------------------------------------------------
C
C
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is by Forythe,
c Computer Methods for Mathematical Computation. it is an bisect method with
c quadratic convergence.
c
c B,V,PO arrays are sinle equlibrum constant
      SUBROUTINE IST_Z1(METHOD,T,SS,Q1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/INDIC1/IND(4)
      COMMON/NCY/NC1
      REAL*8 Q1(4),SS(4)
      INTEGER I,NLIM
      EXTERNAL FCN1
      DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
      DATA R/555/
      T_G=T
      NC=NC1
c the partial pressure
      DO I1=1,NC
       Y1(I1)=SS(I1)
           END DO
      DO I=1,NC
       PP(I)=Y1(I)*T_G*R
      END DO
c no mole fration return
      Z=1.0E-32
      IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
       DO I=1,NC
        Q1(I)=0.
       END DO
       RETURN
      END IF
      N1=1
c set the constant as a function of temperature
     IF(IND(1).EQ.1)THEN
C      V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
      V(N1)=17.0/44/100
C      B(N1)=6864.9000131*EXP(-.019625791466*T)
      B(N1)=5.323235056e-6*T**(-.5)*EXP(13948.544244/1.987/T)
          PO(N1)=.8
      N1=N1+1
     END IF
```

```
      IF(IND(2).EQ.1)THEN
       V(N1)=1./69.035
       B(N1)=1.879094E-4*EXP(5467.4817024/T)
       IF(T.LT.610.) THEN
        B(N1)=4.5597278759E-7*EXP(9628.9655743/T)
       ELSE
        B(N1)=5.8089066684E-7*EXP(9115.734593/T)
       END IF
       N1=N1+1
           PO(N1)=1.0
      END IF
      IF(IND(3).EQ.1)THEN
       V(N1)=-1.637879912E-5*T+.00961297026
       IF(T.LE.532)THEN
        B(N1)=3.2694515539E-7*T+4.59988799E-4
       ELSE
        B(N1)=7.90864008E-5*T-4.14400420E-2
       END IF
           PO(N1)=1.0
       N1=N1+1
      END IF
      IF(IND(4).EQ.1)THEN
       V(N1)=-1.637879912E-5*T+.00961297026
       IF(T.LE.532)THEN
        B(N1)=3.2694515539E-7*T+4.59988799E-4
       ELSE
        B(N1)=7.90864008E-5*T-4.14400420E-2
       END IF
           PO(N1)=1.0
      END IF
c calculate the equilibruim isthotherm by Langmuir method
      IF(NC.EQ.1)THEN
       Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
       RETURN
      ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
       Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
       Q1(1)=0
       RETURN
      ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
       Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
       Q1(2)=0
       RETURN
      END IF
      TERM1=0.
      DO I=1,NC
       IF(PP(I).GT.0)THEN
        TERM1=B(I)*PP(I)**PO(I)+TERM1
       END IF
      END DO
      SUM=0.
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
        SUM=SUM+Q1(I)
       END IF
      END DO
c if the Langmuir method is asked for then returned
      IF(METHOD.EQ.1)THEN
       RETURN
      END IF
c if not, take the result as the first guess for IST theory
      DO I=1,NC
```

```
       X1(I)=Q1(I)/SUM
       IF(X1(I).GT.0) THEN
        PI(I)=PP(I)/X1(I)
       ELSE
        PI(I)=0.
       END IF
      END DO
c calculate the spread pressure
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        X1(I)=0
       ELSE
        X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
        IF=I
       END IF
      END DO
      X=X1(IF)
      DO I=1,NC
       IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
        X=X1(I)
       END IF
      END DO
      AX=X
      DELX=AX
      BX=AX
c call Zeroin subroutine to find the rout to the IST equation
      DO I=1,100
       BX=BX+DELX
       FUN=FCN1(BX)
       IF(FUN.LT.0)GO TO 113
      END DO
113   X=ZEROIN(FCN1,AX,BX,TOL)
c rout was found
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        PP(I)=1.0E-32
       END IF
c calculate the fraction in the solid phase
       PART1=PO(I)*X/V(I)
       IF(PART1.GT.73)PART1=73
       PI(I)=(EXP(PART1)-1)/B(I)
       X1(I)=PP(I)/PI(I)
      END DO
      TOT_Q=0
c calculate the total amount adsorbed
      DO I=1,NC
       Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
       TOT_Q=TOT_Q+X1(I)/Q1(I)
      END DO
c calculate the amount adsorbed for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80   RETURN
      END
c IST function
      REAL FUNCTION FCN1(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
```

```
          PART1=PO(I)*X/V(I)
          IF(PART1.GT.73)PART1=73
          SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
         END IF

C         IF(PP(I).LE.0)PP(I)=1.0E-32
C         PART1=PO(I)*X/V(I)
C         IF(PART1.GT.73)PART1=73
C         SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
        END DO
        FCN1=SUM-1
        RETURN
        END
c
c-------------------------------------------------------------
c
c subroutine to find the root of equation by bisect method

      REAL FUNCTION ZEROIN(FCN1,AX,BX,TOL)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 AX,BX,FCN1,TOL
      REAL*8 A,B,C,D,E,EPS,FA,FB,FC,TOL1,XM,P,Q,R,S
      EPS=1.0
10     EPS=EPS/2.
      TOL1=1.0+EPS
      IF(TOL1.GT.1.0) GO TO 10
c initialization
      A=AX
      B=BX
      FA=FCN1(A)
      FB=FCN1(B)
c begin step
20     C=A
      FC=FA
      D=B-A
      E=D
30     IF(ABS(FC) .GE. ABS(FB)) GO TO 40
      A=B
      B=C
      C=A
      FA=FB
      FB=FC
      FC=FA
c convergence test
40     TOL1=2.*EPS*ABS(B)+.5*TOL
      XM=.5*(C-B)
      IF(ABS(XM).LE.TOL1)GO TO 90
      IF(FB.EQ.0.0) GO TO 90
c is bisection necessary
      IF(ABS(E).LE.TOL1)GO TO 70
      IF(ABS(FA).LE.ABS(FB)) GO TO 70
c is quaratic interpolatation possible
      IF(A.NE.C)GO TO 50
c linear interpolation
      S=FB/FA
      P=2.0*XM*S
      Q=1.0-S
      GO TO 60
c inverse quadratic interpoltation
50     Q=FA/FC
      R=FB/FC
      S=FB/FA
      P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.0))
      Q=(Q-1.0)*(R-1.0)*(S-1.0)
```

```
c adjust signs
60    IF(P.GT.0.0)Q=-Q
      P=ABS(P)
c is interpolation acceptable
      IF((2.0*P).GE.(3.*XM*Q-ABS(TOL1*Q))) GO TO 70
      IF(P.GE.ABS(.5*E*Q)) GO TO 70
      E=D
      D=P/Q
      GO TO 80
c bisection
70    D=XM
      E=D
c complete step
80    A=B
      FA=FB
      IF(ABS(D).GT.TOL1)B=B+D
      IF(ABS(D).LE.TOL1)B=B+SIGN(TOL1,XM)
      FB=FCN(B)
      IF((FB*(FC/ABS(FC))).GT.0)GO TO 20
      GO TO 30
c done
90    ZEROIN=B
      RETURN
      END
C
C-------------------------------------------------------------
C
C
c the second method. this method is faster but the initial guess must be near
c the root of the equation.
c
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is Newton,
c
c B,V,PO arrays are sinle equlibrum constant
      SUBROUTINE IST_Z(METHOD,T,SS,Q1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/NCY/NC1
      COMMON/INDIC1/IND(4)
      REAL*8 Q1(4),SS(4)
      INTEGER I,NLIM
      EXTERNAL FCN,FDER
      DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
      DATA R/555.0/
      T_G=T
      NC=NC1
      DO I1=1,NC
       Y1(I1)=SS(I1)
          END DO
      DO I=1,NC
       PP(I)=Y1(I)*T_G*R
      END DO
c no mole fration return
      Z=1.0E-32
      IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
       DO I=1,NC
        Q1(I)=0.
       END DO
       RETURN
      END IF
```

```
      N1=1
c set the constant as a function of temperature
      IF(IND(1).EQ.1)THEN
c 13x by l, grace
c        point=-1567.1205874+9.00256903*t-.0172347*t*t+1.100806e-5*t*t*t
c        if(pp(n1).gt.point .or. t.gt.627)then
c         V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
c         B(N1)=6864.9000131*EXP(-.019625791466*T)
c         po(n1)=1.0
c        else
c 13x by l_f,grace
c         v(n1)=23.32228/44/100
c         b(n1)=4.9639763e-4*T**(-.5)*exp(9233.33778/1.987/T)
c         po(n1)=-2.9138991288+.017181761178*T-2.8549083257e-5*T*T+
c     .   1.622511757e-8*T*T*T
c        end if
c from Firm data 5A
c         if(pp(n1).lt.1.0) then
C            b(N1)=2.37022397e-6*t**(-.5)*exp(14907.6535/1.987/t)
C            v(N1)=16.6/44/100
C            po(N1)=.80
c Firm by langmuir
c         else
c         v(n1)=(25.97135008-6.084518e-3*T-2.1799516646e-5*T*T)/44/100
c         b(n1)=1.88863393e-5*T**(-.5)*exp(12170.875335/1.987/T)
c         po(n1)=1.
c         end if
C 5A BY GRACE, BY L_F
c         V(N1)=17.0/100/44
c         B(N1)=2.341477E-4*T**(-.5)*EXP(10257.166145/1.987/T)
c         PO(N1)=3.830450111E-4*T**1.188379596
C 5A BY GRACE, BY L
C         V(N1)=(-4.27886889+.087218022*T-9.1010715804E-5*T*T)/44/100
C         B(N1)=9.2533309123E-5*T**(-.5)*EXP(10719.0/1.987/T)
C         PO(N1)=1.
C 5A BY GRACE; BY L_F BETWEEN 0-75 C
        V(N1)=(399.3942-1.938428896*T+3.2540515E-3*T*T-
     .  1.8226211899E-6*T*T*T)/44/100
        B(N1)=35.358072159-.17016733*T+2.7458762E-4*T*T-
     .  1.484116035E-7*T**3
        PO(N1)=-29.3861079+.1538898497*T-2.61225877E-4*T*T+
     .  1.478184694E-7*T**3
c linde 5a
c         v(n1)=.005
c         b(n1)=1.129015193e-5*exp(5055.015089/t)
c         po(n1)=.6
         N1=N1+1
      END IF
c h2o on 5a by grace l,l
      IF(IND(2).EQ.1)THEN
C IND(1).EQ.1 FOR CO2+H2O+N2; IND(1).EQ.0 FOR H2O+N2
C         IF(IND(1).EQ.1)THEN
        if(pp(n1).lt. .4)then
         if(t.le.564.)then
          v(n1)=(39.914452-8.87103e-2*t+6.839502987e-5*t*t)/100/18
         else
          v(n1)=(873.44464-3.867834937*t+5.80375049e-3*t*t-
     .    2.9346685e-6*t*t*t)/100/18
         end if
         b(n1)=29690.66923-137.837129*T+.214456126*t*t
        po(n1)=1
       else
         b(n1)=48.566639771-2.2620239975e-1*t+3.9101703168e-4*t*t-
     .   2.379098497e-7*t*t*t
```

```
        v(n1)=(95.093824733-.21725775358*t+1.4628603e-4*t*t)/100/18
        po(n1)=1
       end if
C       ELSE
CC h2o on 5a grace by L.F
C        b(n1)=24.573259-5.8385278e-2*t+3.473682438e-5*t*t
C        v(n1)=(66.62520167-.12299345*t+7.1897316221e-5*t*t)/100/18
C        if(t.le.600)then
C          po(n1)=.8219916486-3.3519986e-3*t+4.679276479e-6*t*t
C        else
C          po(n1)=-6.0926707634+2.043004e-2*t-2.924858439e-5*t*t+
C     .    1.1746734108e-8*t*t*t
C       end if
C       END IF
C       b(n1)=.2247292664*t**(-.5)*exp(7858.45996/1.987/t)
C       v(n1)=(172.54468-.66643613*t+9.758449e-4*t*t-4.986888e-
C     . 7*t*t*t)/18/100
C       po(n1)=1.
      N1=N1+1
     END IF
     IF(IND(3).EQ.1)THEN
      V(N1)=-1.637879912E-5*T+.00961297026
C       IF(T.LE.532)THEN
        B(N1)=3.2694515539E-7*T+4.59988799E-4
C       ELSE
C         B(N1)=7.90864008E-5*T-4.14400420E-2
C       END IF
            PO(N1)=1.0
      N1=N1+1
     END IF
     IF(IND(4).EQ.1)THEN
      V(N1)=-1.637879912E-5*T+.00961297026
      IF(T.LE.532)THEN
     . B(N1)=3.2694515539E-7*T+4.59988799E-4
      ELSE
        B(N1)=7.90864008E-5*T-4.14400420E-2
      END IF
            PO(N1)=1.0
     END IF
     IF(NC.EQ.1)THEN
      Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
      RETURN
     ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
      Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
      Q1(1)=0
      RETURN
     ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
      Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
      Q1(2)=0
      RETURN
     ELSE IF(NC.EQ.3)THEN
      if(pp(1).le.0 .and. pp(2).le.0)then
        Q1(3)=V(3)*B(3)*PP(3)**PO(3)/(1+B(3)*PP(3)**PO(3))
        Q1(1)=0
        Q1(2)=0
        return
      else if(pp(1).le.0 .and. pp(3).le.0)then
        Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
        Q1(1)=0
        Q1(3)=0
        return
      else if(pp(2).le.0 .and. pp(3).le.0)then
        Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
        Q1(2)=0
```

```
         Q1(3)=0
         return
        end if
       END IF
       TERM1=0.
c calculate the equilibruim by Langmuir isotherm
       DO I=1,NC
        IF(PP(I).GT.0)THEN
         TERM1=B(I)*PP(I)**PO(I)+TERM1
        END IF
       END DO
       SUM=0.
       DO I=1,NC
        IF(PP(I).LE.0.) THEN
         Q1(I)=0.
        ELSE
         Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
         SUM=SUM+Q1(I)
        END IF
       END DO
c if the Langmuir method is asked foR, then return. If not use it as first guess
c for IST theory
       IF(METHOD.EQ.1)THEN
        RETURN
       END IF
       DO I=1,NC
        X1(I)=Q1(I)/SUM
        IF(X1(I).GT.0) THEN
         PI(I)=PP(I)/X1(I)
        ELSE
         PI(I)=0.
        END IF
       END DO
c set the initial guess for the spreading pressure
       DO I=1,NC
        IF(PP(I).LE.0.) THEN
         X1(I)=0
        ELSE
         X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
         IF=I
        END IF
       END DO
       X=X1(IF)
       DO I=2,NC
        IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
         X=X1(I)
        END IF
       END DO
c call newton method to the rout to IST equation
       IF(X .LE. 1.0E-17) GO TO 80
       CALL NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
c rout was found, calculate the fraction in the solid phase
       DO I=1,NC
        IF(PP(I).LE.0) THEN
         X1(I)=0.
        ELSE
         PART1=PO(I)*X/V(I)
         IF(PART1.GT.73)PART1=73
         PI(I)=(EXP(PART1)-1)/B(I)
         X1(I)=PP(I)/PI(I)
        END IF
       END DO
c calculate the total amound in the solid phase
       TOT_Q=0
```

```
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
        TOT_Q=TOT_Q+X1(I)/Q1(I)
       END IF
      END DO
c calculate the amount in equilibruim with the gas phase for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80    RETURN
      END
c
c IST final equation
c
      REAL FUNCTION FCN(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
       END IF
      END DO
      FCN=SUM-1
      RETURN
      END
c
c IST final derivitive
c
      REAL FUNCTION FDER(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART3=PO(I)*X/V(I)
        IF(PART3.LT.35)THEN
         PART1=-PP(I)/B(I)/V(I)*EXP(PART3)
         PART2=((EXP(PART3)-1)/B(I))**((1+PO(I))/PO(I))
         SUM=SUM+PART1/PART2
        ELSE IF(PART3/PO(I).LT.73)THEN
         SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .   EXP(-PART3/PO(I))
        ELSE
         PART3=73
         SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .   EXP(-PART3)
        END IF
       END IF
      END DO
      FDER=SUM
      RETURN
      END

c
```

```
c--------------------------------------------------------
c this subroutine finds the rout to a nonlinear equation using the Newton
c method
     SUBROUTINE NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
     IMPLICIT REAL*8(A-H,O-Z)
     INTEGER NLIM,I,J
     FX=FCN(X)
     X1=X
     DO J=1,NLIM
      DELX=FX/FDER(X)
      X=X-DELX
      FX=FCN(X)
      IF (ABS(X-X1)/X.LE.XTOL)THEN
       RETURN
      END IF
      IF(FX.NE.0)THEN
       IF(ABS(FX-FX1)/FX.LE.FTOL) THEN
        RETURN
       END IF
      END IF
      X1=X
      FX1=FX
     END DO
     I=-1
     PRINT 200, NLIM,X,FX
 200  FORMAT(/'TOLERANCE NOT MET ',I4,' ITERATIONS X= ',
   2 E12.5,' F(X)=',E12.5)
     RETURN
     END


    FUNCTION CPGAS(TEM,Y)
C THIS FUNCTION CALCULATES THE HEAT CAPACITIES OF CMRIER GAS IN BTU/#MOLE/R
    IMPLICIT REAL*8(A-H,O-Z)
    REAL*8 A(4),B(4),C(4),D(4),Y(4)
    COMMON/GAS/INERT,NCOMP
    COMMON/NCY/NC1
    COMMON/INDIC1/IND(4)
C CO2
    DATA A/36.11,33.46,29.,29.1/
C H2O
    DATA B/4.233E-2,.6880E-2,.2199E-2,1.158E-2/
C N2
    DATA C/-2.887E-5,.7604E-5,.5723E-5,-.6076E-5/
C O2
    DATA D/7.464E-9,-3.593E-9,-2.871E-9,1.311E-9/
    NC=NC1
    T=TEM/1.8-273.15
    IF (INERT.EQ.1) THEN
     CPGAS=0.
     YO=0.
     N1=1
     DO I=1,4
      IF(IND(I).EQ.1)THEN
       CP=A(I)+B(I)*T+C(I)*T*T+D(I)*T*T*T
       CPGAS=CP*Y(N1)/4.18669+CPGAS
       YO=Y(N1)+YO
       N1=N1+1
      END IF
     END DO
     CPGAS=CPGAS+(A(3)+B(3)*T+C(3)*T*T+D(3)*T*T*T)*(1-YO)/4.18669
    ELSE
     CPGAS=4.97
    ENDIF
```

```
      RETURN
      END

      FUNCTION CPS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES TNE HEAT CAPACITY
C BTU/LB/R.
      CPS=1.0
      RETURN
      END

      REAL FUNCTION KGAS (TEMP)
C THIS FUNCTION CALCULATES THE THERMAL CONDUCTIVITY OF CARRIER.
C BTU/MIN/FT/R
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
        KGAS=3.88E-7*TEMP+0.4052E-4
      ELSE
        KGAS=1.667E-6*TEMP+6.1E-4
      ENDIF
      RETURN
      END

      FUNCTION RE (G,TEMP)
C TNIS FUNCTION CALCULATES THE PARICLE REYNOLDS NUMBER ASSUMINC THAT
C THE MOLECULAR WEIGHT OF TNE GAS IS EQUAL TO THAT OF CARRIER GAS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,MW
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
C       COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/GAS/INERT,NCOMP
      DATA IFLAG/0/
      IF (INERT.EQ.1) MW=28.0
      IF (INERT.EQ.2) MW=4.
      IF (IFLAG.EQ.0)THEN
        CONST=2.*RA
        IFLAG=1
      ENDIF
      RE=CONST*MW*ABS(G)/VIS(TEMP)
      RETURN
      END

      FUNCTION VIS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE VISCOSITY OF CARRIER GAS AS A  FUNCTION
C OF TEMPERATURE (LB/MIN/FT).
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
C         VIS=1.0E-6*TEMP+1.65E-4
        VIS=-.0102007812+5.384663937E-5*TEMP-8.7973727E-8*TEMP*TEMP+
     2      4.811387495E-11*TEMP*TEMP*TEMP
      ELSE
        VIS=0.9444E-6*TEMP+2.863E-4
      ENDIF
      RETURN
      END

      FUNCTION RHOG(TR,P)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
C THIS FUNCTION CALCULATES THE MOLAR DENSITY OF AN IDEAL GAS IN
C LBMOLES/CV FS.
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
```

```
      DATA R/555./
      RHOG=P/R/TR
      RETURN
      END

      FUNCTION EFFD(ICOMP,G,TEMP,RHO,PT,YO)
C THIS FUNCTION CALCVLATES TNE EFFECTIVE AXIAL DIFFUSIVITIES IN A PACXED BED
C USING THE EDWARDS AND RICHARDSON CORROLATION. (FT*FT/MIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA1,LAMBDA2,LEN,YO(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      DATA IFLAG/0/,LAMBDA1/0.73/
      IF(IFLAG.EQ.0)THEN
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF
      DIF=DIFF(ICOMP,TEMP,PT,YO)
      EFFD=LAMBDA1*DIF+CONST*ABS(G)*LAMBDA2(G,DIF,RHO)/RHO
      RETURN
      END

      FUNCTION EFFK(G,T,CP,RHO,P,YO)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE EFFECTIVE AXIAL THERMAL CONDUCTIVITY OF A
C PACKED BED IN BTU/FT/MIN/F.
      COMMON/NCY/NC1
      REAL*8 YO(4)
      NC=NC1
      EFFK=0.
      KK=0
      DO I1=1,NC
       SUM=CP*RHO*EFFD(I1,G,T,RHO,P,YO)
       IF(SUM.GT.0)KK=KK+1
       EFFK=SUM+EFFK
      END DO
       EFFK=EFFK/KK
      RETURN
      END

      REAL FUNCTION LAMBDA2(G,DIF,RHO)
C THIS FUNCTION CALCULATES THE DIMENSTONLESS PARAMETER OF THE EDWARDS AND
C RICHARDS CORROLATION MODIFIED FOR SMALL DIAMETER PARTICLES.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      DATA IFLAG/0/
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      IF(IFLAG.EQ.0)THEN
       IF(RA.GT.0.00492)THEN
         PE=2.
       ELSE
        PE=406.4*RA
       ENDIF
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF

      LAMBDA2=1./PE/(1.+9.5*DIF/CONST/ABS(G)*RHO)
      RETURN
      END


      FUNCTION DIFF(ICOMP,TEM,P1,YO)
C  THIS FUNCTION CALCULATES THE MOLECULAR  DIFFUSION COEFFICIENTS BY CORRECTING
C  FOR PRESSURE AND TEMPERATURE (FT*FT/MIN).
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,YO(4),V1(4),MW1(4),V(4),MW(4),DIF(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
      COMMON/INDIC1/IND(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      DATA V1/26.9,12.7,17.9,16.6/
      DATA MW1/44.01,18.016,28.013,31.999/
      NC=NC1
      K=ICOMP
      P=P1/760.
      T=TEM/1.8
      N1=1
      DO I1=1,4
       IF(IND(I1).EQ.1)THEN
        MW(N1)=MW1(I1)
        V(N1)=V1(I1)
        N1=N1+1
       END IF
      END DO
      DO I=1,NC
       IF(NC.EQ.1) THEN
        MW(I)=MW(K)
        V(I)=V(K)
       END IF
       PART1=((MW(K)+MW(I))/MW(K)/MW(I))**.5
       PART2=(V(K)**(1./3)+V(I)**(1./3))**2
       DIF(I)=PART1/PART2*T**1.75/P*1.0E-3
      END DO
      IF(NC.EQ.1)THEN
       DIFF=DIF(1)*3.8745
       RETURN
      END IF
      IF(NC.EQ.2)THEN
       IF(K.EQ.1)DIFF=DIF(2)*3.8745
       IF(K.EQ.2)DIFF=DIF(1)*3.8745
       RETURN
      END IF
      DO I=1,NC
       IF(I.NE.K)THEN
        SUM=SUM+YO(I)/DIF(I)
       END IF
      END DO
      if(sum.ne.0) then
       DIFF=(1-YO(K))/SUM*3.8745
      else
       diff=1
      endif
      RETURN
      END


      FUNCTION HFILM(MW,TEMP,REY,CP)
C THIS FUNCTION CALCULATES THE FLUID FILM HEAT TRANSFER COEFFICIENT
C IN BTU/SQ FT/MIN/R USING THE CORRELATION OF PETROY AND THODOS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 ID,KGAS,KG,LEN,MW
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      COMMON/GAS/INERT,NCOMP
      KG=KGAS(TEMP)
      PR=CP/MW*VIS(TEMP)/KG
      HFILM=0.357/EPSEX*REY**0.64*PR**0.33*KG*0.5/RA
      RETURN
      END
```

```
      FUNCTION DHADS(ICOMP,T)
C THIS FUNCTION CALCULATES THE ISOTERIC HEAT OF ADSORPTION. (BTU/#MOLE)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1,UINS,HWALL,LEN
      IF(ICOMP.EQ.2) THEN
        DHADS=20400.0
      ELSE
        DHADS=20000.0
      END IF
      RETURN
      END
C THIS FUNCTION CALCULATE THE CONDUCTIVTY OF FLUID IN AXIAL AND RADIAL
C DIRECTION.

        SUBROUTINE CONDUC1(EMIS,AKS,D_P,D_I,GA,VAV,TAV,RO,CP,AVIS,
     1  AKC,NJR,DELAV,DELTA1,DELTA2,DELR0,E,AKT)
        IMPLICIT REAL*8(A-H,O-Z)
             DIMENSION V(30),VS(30),RP(30),AK(30),AKST(30),DELTA(30),
     1       E(30),ES(30),DER(30),DKR(30),W(30),AKT(30),DELR0(30)
        DELTA(1)=DELTA1
        DELTA(NJR)=DELTA2
        RP(1)=0.
        RP(NJR)=1.
        DO K=2,NJR
          RP(K)=2*DELR0(K)/D_I+RP(K-1)
        END DO
C
C VELOCITY CALCULATION
C
             NJJ=NJR
             RMS=1.-2*D_P/D_I
             BO=.16*(D_P/(D_I))**(-3./2.)
             AO=1./(BO+2.)-RMS/(BO+1)
             A2=RMS/(BO+1.)
             A3=1./(BO+2.)
             D=AO/2.+A2/(BO+3.)-A3/(BO+4.)
             DO I=1,NJJ
              VS(I)=(AO+A2*(RP(I)**(BO+1.))-A3*(RP(I)**(BO+2.)))/(2.*D)
             END DO
             DO I=1,NJJ
              V(I)=VS(I)*VAV
             END DO
             DPDT=D_P/(D_I)
             RW=1.-DPDT
             VW=(AO+A2*(RW**(BO+1.))-A3*(RW**(BO+2.)))/(4.*D)
C
C CALC OF EAV, EO,AND EW
C
             R1=1.-4.*DPDT
             EO=VS(1)*(1.+19.4*DPDT**2)*9./8.
             EAV=D_P*VAV/(9.*(1.+19.4*DPDT**2))
             CCC=(1.+R1+0.1*R1**2)/6.
             EM=(.5-.15*EO*R1**2)/CCC
             EW=EM*(1.-RW)/(1.-R1)/2.
C
C CALC OF AKAV, OK, WK
C
             ANPR=CP*AVIS/AKC
             E(1)=EO*EAV
             E(NJJ)=EW*EAV
             V(NJJ)=  VW*VAV
             DO J=1,3
```

```
        IF(J.EQ.3)THEN
         I=3
         V(3)=VAV
         DELTA(3)=DELAV
         E(3)=EAV
        ELSE IF(J.EQ.1)THEN
         I=1
        ELSE IF(J.EQ.2)THEN
         I=NJJ
        END IF
        ANRE=D_P*RO*V(I)/AVIS
        IF((ANRE-350).LE.0)THEN
         AJH=1.95*ANRE**(-.51)
        ELSE
         AJH=1.06*ANRE**(-.41)
        END IF
        HC=AJH*CP*RO*V(I)*(ANPR**(-2./3.))
        AKR=4.*EMIS*D_P*.173*((TAV*1.8)**3.)/((100.**4.)*(2.-EMIS))
        AKP=EXP((-1.76+.0129*AKS/DELTA(I))/.434)
        H=HC+300.
        DO K=1,50
         HR=AKR*(2.*AKS+H*D_P)/(D_P*AKS)
         HP=AKP*(2.*AKS+H*D_P)/(D_P*AKS)
         H1=HC+HR+HP
         IF(ABS((H1-H)/H).LE.0.01) GO TO 60
         H=H1
        END DO
60      AK(I)=DELTA(I)*(AKC+CP*RO*E(I)/DELTA(I)+AKR)+
   1    (1.-DELTA(I))*(H*AKS*D_P/(2.*AKS+H*D_P))
       END DO
       AKAV=AK(3)
       OK=AK(1)/AK(3)
       WK=AK(NJR)/AK(3)
C
C CALC OF ES, AKST, DER, DKR
C
       AKM=(.5-.15*OK*R1**2-WK*(R1**3-3.*R1+2.)/(6.*(1-R1)))/CCC
       DO I=1,NJJ
        IF(RP(I).GE.R1)THEN
         ES(I)=EM*(1.-RP(I))/(1.-R1)
         AKST(I)=AKM-(AKM-WK)*(RP(I)-R1)/(1-R1)
         DER(I)=-EM/(1.-R1)
         DKR(I)=(WK-AKM)/(1.-R1)
        ELSE
         ES(I)=EO+3.*(EM-EO)*(RP(I)/R1)**2+(EO-EM)*2.*(RP(I)/R1)**3
         AKST(I)=OK+3.*(AKM-OK)*(RP(I)/R1)**2+(OK-AKM)*2.*(RP(I)/R1)
   1     **3
         DKR(I)=6.*(AKM-OK)*RP(I)/R1**2+6.*(OK-AKM)*RP(I)**2/R1**3
         DER(I)=6.*(EM-EO)*RP(I)/R1**2+6.*(EO-EM)*RP(I)**2/R1**3
        END IF
       END DO
       DO I=1,NJJ
        E(I)=ES(I)*EAV
        AKT(I)=AKST(I)*AKAV
       END DO
    RETURN
       END

    SUBROUTINE CONDU(ITEST,IGAS,PHI,D_P,CON_P,RR,U0,RE_N,TEM,EPS,
  . EPS_INF,EFF_CON_R,CON_R,EFF_CON_Z,CON_Z,CON_S,NJR,NGR1,NGR2,NN)
    IMPLICIT REAL*8(A-H,O-Z)
       DATA ALF_BET1,ALF_BET2,PE_N,EPS_W,CO/.13,.8,9.,.7,.15/
C
```

```
C NITROGEN
C
      IF(IGAS.EQ.1)THEN
        CON_F=.000358895+3.0026706379E-5*TEM-5.3528942E-9*TEM*TEM
        PR_N=.8485-.0003845398*TEM+2.76374798E-7*TEM*TEM-
     .  5.62898861E-11*TEM*TEM*TEM
C
C AIR
C
       ELSE IF(IGAS.EQ.2)THEN
        CON_F=.0002153778+3.0066679595E-5*TEM-4.8615223803E-9*TEM*TEM
             PR_N=.822301567-.000315649*TEM+2.155594425E-7*TEM*TEM-
     .  4.22896146E-11*TEM*TEM*TEM
C
C CO2
C
       ELSE
        CON_F=-.00146886721+1.652016966E-5*TEM+7.3392609982E-9*TEM*TEM
             PR_N=1.038870689-.000842560218*TEM*TEM+8.338446183E-7*TEM*TEM-
     .  3.453641637E-10*TEM*TEM*TEM
       END IF
       IF(ITEST.EQ.1) THEN
             CON_0=EPS+(1-EPS)/(PHI+2.*CON_F/CON_P/3.)
        CON_01=ALF_BET1*PR_N*RE_N/EPS
        EFF_CON_1=CON_0+CON_01
        EFF_CON_R=EFF_CON_1*CON_F
        CON_R=EFF_CON_R-CON_F*CON_0
        EFF_CON_Z=CON_F*(CON_0+CON_01*ALF_BET2/ALF_BET1)
        CON_Z=EFF_CON_Z-CON_F*CON_0
        CON_S=CON_F*CON_0
       ELSE
             CON_0=EPS_INF+(1-EPS_INF)/(PHI+2.*CON_F/CON_P/3.)
        CONST=1./CON_0-1.
        TEXP=EXP(-5.*RR)
        IF(NN.GT.NGR1)THEN
          CON_03=RR*ALF_BET1*PR_N*RE_N/EPS
          CON_02=CON_0*(1.+CONST*TEXP)
        ELSE
          CON_03=ALF_BET1*PR_N*RE_N/EPS
          CON_02=CON_0
        END IF
        EFF_CON_1=CON_02+CON_03
        EFF_CON_R=EFF_CON_1*CON_F
        CON_R=EFF_CON_R-CON_F*CON_02
        EFF_CON_Z=CON_F*(CON_02+CON_03*ALF_BET2/ALF_BET1)
        CON_Z=EFF_CON_Z-CON_F*CON_02
        CON_S=CON_F*CON_02
       END IF
            RETURN
       END
C
C THIS SUBROUTINE COMPUTE THE DIFFUSIVITY IN AXIAL AND RADIAL DIRECTION FOR
C TWO DIM
C

C THIS SUBROUTINE COMPUTE THE DIFFUSIVITY IN AXIAL AND RADIAL DIRECTION FOR
C TWO DIM
C
      SUBROUTINE AXDIFF_R_Z(ITEST,IGAS,PHI,D_P,DIFF_F,RR,U0,RE_N,SC_N,
     . TEM,EPS,EPS_INF,EFF_DIFF_R,EFF_DIFF_Z,njr,ngr1,NGR2,nn)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA ALF_BET1,ALF_BET2,PE_N,EPS_W,CO/.1,1.0,10,.7,.15/
C
C NITROGEN
```

```
C
      IF(ITEST.EQ.1) THEN
       DIFF_0=1-(1.-EPS)**.5
       DIFF_01=ALF_BET1*SC_N*RE_N/EPS
       EFF_DIFF_1=DIFF_0+DIFF_01
       EFF_DIFF_R=EFF_DIFF_1*DIFF_F
       EFF_DIFF_Z=DIFF_F*(DIFF_0+DIFF_01*ALF_BET2/ALF_BET1)
      ELSE
       DIFF_0=1-(1.-EPS_INF)**.5
       CONST=1./DIFF_0-1.
       TEXP=EXP(-8.*RR)
       IF(NN.GT.NGR1)THEN
c         DIFF_03=(1-TEXP)*ALF_BET1*SC_N*RE_N/EPS
c         DIFF_02=DIFF_0*(1.+CONST*TEXP)
        DIFF_03=RR*ALF_BET1*SC_N*RE_N
        DIFF_02=DIFF_0*(1.+CONST*TEXP)
       ELSE
         DIFF_03=ALF_BET1*SC_N*RE_N
         DIFF_02=DIFF_0
       END IF
       EFF_DIFF_1=DIFF_02+DIFF_03
       EFF_DIFF_R=EFF_DIFF_1*DIFF_F
       EFF_DIFF_Z=DIFF_F*(DIFF_02+DIFF_03*ALF_BET2/ALF_BET1)
      END IF
      RETURN
      END
C
C THIS SUBROUTINE COMPUTE THE HEAT TRANSFER COEFFICIENT FOR THE GAS AND WALL
C
      SUBROUTINE HEAT_WALL(IGAS,PHIW,D_P,CON_P,RR,U0,RE_D,TEM,
     . EPS,H_W)
      IMPLICIT REAL*8(A-H,O-Z)
          DATA ALF_BET1,ALF_BET2,PE_N,EPS_W,CO/.13,.8,9.,.7,.15/
C
C NITROGEN
C
      IF(IGAS.EQ.1)THEN
        CON_F=.000358895+3.0026706379E-5*TEM-5.3528942E-9*TEM*TEM
        PR_N=.8485-.0003845398*TEM+2.76374798E-7*TEM*TEM-
     .  5.62898861E-11*TEM*TEM*TEM
C
C AIR
C
       ELSE IF(IGAS.EQ.2)THEN
        CON_F=.0002153778+3.0066679595E-5*TEM-4.8615223803E-9*TEM*TEM
            PR_N=.822301567-.000315649*TEM+2.155594425E-7*TEM*TEM-
     .  4.22896146E-11*TEM*TEM*TEM
C
C CO2
C
       ELSE
        CON_F=-.00146886721+1.652016966E-5*TEM+7.3392609982E-9*TEM*TEM
            PR_N=1.038870689-.000842560218*TEM*TEM+8.338446183E-7*TEM*TEM-
     .  3.453641637E-10*TEM*TEM*TEM
       END IF
           CON_0=EPS+(1-EPS)/(PHIW+2.*CON_F/CON_P/3.)
       CON_01=ALF_BET1*PR_N*RE_D
       EFF_CON_1=CON_0+CON_01
           CON_W1=EPS_W+(1-EPS_W)/(PHIW+2.*CON_F/CON_P/3.)
       H_W0=CO*PR_N**(1./3.)*RE_D**(3./4)*CON_F/D_P
       CON_W2=CON_W1+1./(1./(.2*PR_N*RE_D)+1./(H_W0*PR_N*D_P/2./CON_F))
       H_W1=1./CON_W2-1./EFF_CON_1
       H_W=1.*CON_F/D_P/2./H_W1
           RETURN
```

```
      END
C
C THIS SUBROUTINE COMPUTE A CONSTANT FOR CALCULATION OF THERMAL CONDUCTIVITY
C
      SUBROUTINE PHI_COND(IGAS,TEM,CON_P,EPS,PHI,PHIW)
      IMPLICIT REAL*8(A-H,O-Z)
          REAL*8 X1(11),Y1(11),X2(11),Y2(11),X3(11),Y3(11),XINT
      DATA X1/20000.,4000.,1000.,200.,80.,8.,.6,.2,.07,.02,.01/
      DATA Y1/.04,.05,.06,.08,.1,.2,.41,.8,2.,6.,12./
      DATA X2/20000.,400.,200.,40.,8.,2.,.8,.2,.08,.04,.01/
      DATA Y2/.01,.02,.023,.04,.08,.2,.4,.8,4.,8.,28./
      DATA X3/10000.,2000.,400.,20.,2.,0.,0.,0.,0.,0.,0/
      DATA Y3/.06,.08,.1,.2,.34,.0,.0,.0,.0,.0,.0/
C
C NITROGEN
C
     IF(IGAS.EQ.1)THEN
       CON_F=.000358895+3.0026706379E-5*TEM-5.3528942E-9*TEM*TEM
C
C AIR
C
      ELSE IF(IGAS.EQ.2)THEN
       CON_F=.0002153778+3.0066679595E-5*TEM-4.8615223803E-9*TEM*TEM
C
C CO2
C
      ELSE
       CON_F=-.00146886721+1.652016966E-5*TEM+7.3392609982E-9*TEM*TEM
      END IF
      N=11
      XMAX1=CON_P/CON_F
      ITEST1=0
      ITEST2=0
      ITEST3=0
      DO I=1,N
       IF(X1(I).LT.XMAX1 .AND. ITEST1.EQ.0) THEN
        SLOPE=(Y1(I-1)-Y1(I))/(X1(I-1)-X1(I))
        PHI1=SLOPE*(XMAX1-X1(I-1))+Y1(I-1)
        ITEST1=1
       END IF
       IF(X2(I).LT.XMAX1 .AND. ITEST2.EQ.0) THEN
        SLOPE=(Y2(I-1)-Y2(I))/(X2(I-1)-X2(I))
        PHI2=SLOPE*(XMAX1-X2(I-1))+Y2(I-1)
        ITEST2=1
       END IF
       IF(X3(I).LT.XMAX1 .AND. ITEST3.EQ.0) THEN
        SLOPE=(Y3(I-1)-Y3(I))/(X3(I-1)-X3(I))
        PHIW=SLOPE*(XMAX1-X3(I-1))+Y3(I-1)
        ITEST3=1
       END IF
       IF(ITEST1.EQ.1 .AND. ITEST2.EQ.1 .AND. ITEST3.EQ.1)GO TO 20
     END DO
 20   IF(EPS.LT.0.26) THEN
       PHI=PHI2
     ELSE IF(EPS.GT.0.476) THEN
       PHI=PHI1
     ELSE
       PHI=PHI2+(PHI1-PHI2)*(EPS-.26)/.216
     END IF
     RETURN
     END


C
```

```
C SUBROUTINE TO CALCULATE THE POROUSITY VARIATION
C

      SUBROUTINE COEF2(Y1,VOID_B,D_P,C1,POR)
      IMPLICIT REAL*8(A-H,O-Z)
      C2=6.0
      POR=VOID_B*(1+C1*EXP(-C2*Y1/D_P))
      RETURN
      END

           SUBROUTINE COEF1(E,EPI, Y,C)
      IMPLICIT REAL*8(A-H,O-Z)
C B=2
C      S1= -.16329931237*Y*EPI/(1-EPI)
C B=3
c     S1= -.244048968*Y*EPI/(1-EPI)
C B=4
C      S1= -.326598632*Y*EPI/(1-EPI)
C B=5
C       S1= -.4082482808*Y*EPI/(1-EPI)
C B=6
       S1= -.489897937*Y*EPI/(1-EPI)
C B=8
C      S1= -.653197264*Y*EPI/(1-EPI)
      E= EPI*(1.+1.4*EXP(S1))
           E= EPI*(1.+C*EXP(S1))
           RETURN
           END
C
C-------------------------------------------------------------
C
C THIS SUBROUTINE IS BEING CALLED BY DIFFEQ1 WHICH IS ALSO BEING CALLED BY
C SUBROUTINE FUNCT TO COMPUTE THE VELOSTU AND THE PRESSURE DROP IN THE BED.
C THE EQUATIONS ARE BEING SOLVED BY NEWMAN'S METHOD.
C
      SUBROUTINE FUNCT2(J)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 D_L(2),M_AVE
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),F(14),P(14,201)
      COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
      COMMON/PR_OLD_Z/CON_Z(201,101),CON_R(201,101),CON_S(201,101),
     . H_W(201,101),CP_P(201,101),H_FP(201,101),Q(4,201,101),
     . VISC_P(201,101),D_LPP(4,201),RATE_C1(201,101)
      COMMON/MISC_Z/R,G_F,PI,P_TOT,BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,RHOS,TAMB,ALPHA1
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1
      COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
      DATA L1,L4,L5,L6,R1,CONV/4,1,2,3,555.0,51.714752314/
      DATA FAC,GC,M_AVE/2.78450526316,416975040.0,28.0/
C
C THE FIRST ROW
C
     ALF=(1-EPSEX)/EPSEX
     RATE_C=RATE_C1(J,1)
     VISC_F=VISC_P(J,1)
     DO M1=1,NC
      D_L(M1)=D_LPP(M1,J)
     END DO
     KK=0
     D_L_AVE=0
     DO M1=1,NC
      IF(D_L(M1).GT.0)THEN
```

```
          D_L_AVE=D_L_AVE+D_L(M1)
          KK=KK+1
        END IF
      END DO
      RO_F=C(L5,J)/(C(L1,J)*R1)
      IF(J.EQ.1)THEN
c equation for velocity
        F(L4)=FAC*(C(L5,J)-BC_L5*CONV)/DELZ+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-BC_L4)/DELZ+150*(1-EPSEX)**2*VISC_F*C(L4,J)/
     . D_P/D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/
     . D_P/EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT)
c equation for pressure not considering the pressure drop due to temperature
c change, dynamic pressure
        F(L5)=(C(L5,J)-P(L5,J))/DELT-D_L_AVE/DELZ**2*(BC_L5*CONV-
     . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-BC_L5*CONV)/DELZ+
     . C(L5,J)*(C(L4,J)-BC_L4)/DELZ+R1*C(L1,J)*ALF*RATE_C
c total pressure
        F(L6)=(C(L6,J)-P(L6,J))/DELT-D_L_AVE/DELZ**2*(BC_L5*CONV-
     . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-BC_L5*CONV)/DELZ+
     . C(L6,J)*(C(L4,J)-BC_L4)/DELZ-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT-D_L_AVE/DELZ**2*(C(L1,J+1)-2*C(L1,J)+BC_L1)+
     . C(L4,J)*(C(L1,J+1)-BC_L1)/(2*DELZ))+R1*C(L1,J)*ALF*RATE_C

C
C------------------------------------------------
C the last grid
c
c equation for velocity
       ELSE IF(J.EQ.NJ)THEN
        F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-C(L4,J-1))/DELZ+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
     . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
     . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT)
c equation for pressure not considerin the pressure drop due to temperature
c change, dynamic pressure
        F(L5)=(C(L5,J)-P(L5,J))/DELT-D_L_AVE/DELZ**2*(C(L5,J-1)-
     . 2*C(L5,J)+C(L5,J-1))+C(L5,J)*(C(L5,J)-C(L5,J-1))/DELZ+
     . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ+R1*C(L1,J)*ALF*RATE_C
c total pressur
        F(L6)=(C(L6,J)-P(L6,J))/DELT-D_L_AVE/DELZ**2*(C(L6,J-1)-
     . 2*C(L6,J)+C(L6,J-1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ+
     . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT-D_L_AVE/DELZ**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J-1))+
     . C(L4,J)*(C(L1,J-1)-C(L1,J-1))/(2*DELZ))+R1*C(L1,J)*ALF*RATE_C
C
C------------------------------------------------
C
c grids inside the column
       ELSE
c the velocity
        F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-C(L4,J-1))/DELZ+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
     . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
     . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT)
c equation for pressure not considering the pressure drop due to temperature
c change, dynamic pressure
        F(L5)=(C(L5,J)-P(L5,J))/DELT-D_L_AVE/DELZ**2*(C(L5,J-1)-
     . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-C(L5,J-1))/DELZ+
     . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ+R1*C(L1,J)*ALF*RATE_C
c total pressure
        F(L6)=(C(L6,J)-P(L6,J))/DELT-D_L_AVE/DELZ**2*(C(L6,J-1)-
     . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ+
     . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT-D_L_AVE/DELZ**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J-1))+
```

```
     . C(L4,J)*(C(L1,J+1)-C(L1,J-1))/(2*DELZ))+R1*C(L1,J)*ALF*RATE_C
      END IF
 210  RETURN
      END
C
C--------------------------------------------------------------------
C
c this subroutine is being called by diffeq1 which is also being called by
c funct subroutine. it compute the velocity profile in the radial direction
c it uses the Newman method.
        SUBROUTINE FUNCT3(J)
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 DR(200),EPS(200),C_K1(200),C_K2(200)
        COMMON/UG/U_GUESS,EPS
        COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
        COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     .  Y(14,14),G(14),F(14),P(14,201)
        COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
        COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S1
        COMMON/PROP_B_Z/REC,VOID_B,D_I
        COMMON/PROP_D_Z/DELZ,DELT,TIME,DELR1,DELR2,JZ,JT,NC,NJR,NP
        COMMON/MISC_Z/R,G_F,PI,P_TOT,BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_1
C
       IF(J.EQ.1 .AND. M1.EQ.0) THEN
c radial fraction of constant velocity
         FRAC1=.93
c radial fraction of variable velocity
         FRAC2=.07
c number of grids
         K1=21
         K2=99
c dimensionless constant
                FAC= (1-VOID_B)/VOID_B*12.247449
c dimensionless radial size
                DELR11= FRAC1*D_I/2/D_P*FAC/(K1-1)
                DELR22= FRAC2*D_I/2/D_P*FAC/(K2)
         EC=1./VOID_B-1.
         EC=1.4
c compute the porosity variation
                DO I=1, NP-1
                 R= (DELR22*(K2)+DELR11*(K1-I))
                 IF(I .GT. K1) THEN
                  R= DELR22*(NP-I)
                 END IF
                 CALL COEF1(POR, VOID_B, R, EC)
           EPS(I)=POR
c compute C and K constant in MOmentum equation
          C_K1(I)=D_P*D_P*EPS(I)*EPS(I)*EPS(I)/150./(1-EPS(I))/
     .    (1-EPS(I))
          C_K2(I)=1.75*(1-EPS(I))/D_P/EPS(I)/EPS(I)/EPS(I)
         END DO
c calculate the constant coefficient in PDE's
         C_K11=C_K1(1)
         C_K22=C_K2(1)
         BETA=C_K11/D_P/D_P
         EPS(NP)=EPS(1)*(1+1.4)
         DELR11=DELR11/FAC
         DELR22=DELR22/FAC
         FR=DELR22/DELR11
         F01=FR/(1+FR)
         F11=(1.-FR)/FR
         F21=1./((1.+FR)*FR)
         F02=1./(1+FR)
         F12=1./FR
```

```
        F22=1./((1.+FR)*FR)
        M1=1
      END IF
c compute the radios from the center to the grid
      IF(J.LE.K1 .AND. J.NE.1) THEN
        DELR=DELR11
        RP=(J-1)*DELR
        RP=1./RP
      ELSE IF(J.GT.K1)THEN
        DELR=DELR22
        RP=(K1-1)*DELR11+(J-K1)*DELR22
        RP=1./RP
      END IF
c compute the velocity equation for the center grid
      IF(J.EQ.1)THEN
       F(1)=U_GUESS-C(1,J)
C-----------------------------------------------------
C
c compute the velocity the the wall boundry
      ELSE IF(J.EQ.NJ)THEN
       F(1)=0-C(1,J)
C
C-----------------------------------------------------
C
c compute the velocity for grids in radial directions
      ELSE
        IF(J.NE.K1) THEN
         P1=1.-C_K11/C_K1(J)*C(1,J)
         P2=-C_K2(J)/C_K22*REC*C(1,J)*C(1,J)
         P3=RP*(C(1,J+1)-C(1,J-1))/(2*DELR)
         P4=(C(1,J+1)-2*C(1,J)+C(1,J-1))/DELR/DELR
         F(1)=P1+P2+BETA/EPS(J)*(P3+P4)
        ELSE
         P1=1.-C_K11/C_K1(J)*C(1,J)
         P2=-C_K2(J)/C_K22*REC*C(1,J)*C(1,J)
         P3=RP*(F21*C(1,J+1)-F11*C(1,J)-F01*C(1,J-1))/(DELR)
         P4=2*(F22*C(1,J+1)-F12*C(1,J)+F02*C(1,J-1))/DELR/DELR
         F(1)=P1+P2+BETA/EPS(J)*(P3+P4)
        END IF
      END IF
 210  RETURN
      END
C
C-----------------------------------------------------
C This subroutine is the main routine for Newman method for solutio of PDE's.
C
C************************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C************************************************************************
C
C     CALLED BY:  MAIN CALLING PROGRAM
C
C************************************************************************
C
C     SUBROUTINES CALLED:
C          WRTOUT (FOR DATA OUTPUT)
C          BAND   (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C          FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C               FOR A VARIABLE)
C
C************************************************************************
C
C     LIST OF IMPORTANT VARIABLES:
```

```
C
C          A     A coefficient described in Newman, Appendix C
C          AA     first, AA is F(WORKC*CU). Later AA is the
C               value of the derivative df/dc used in Newton's
C               method
C          B     B coefficient described in Newman, Appendix C
C          C     variables to be solved for
C          CD     multiplication factor used in obtaining
C               numerical derivatives
C          COLD   value of C from previous iteration
C          CU     2.0 - CD
C          D     D coefficient described in Newman, Appendix C
C          ERR    convergence criterion
C          F     value of function f(C), calculated in FUNCT
C          G     residual of f(C) calculated with updated C value
C          I    index used for equation number
C          ITCNT   index for iteration number
C          ITPRT  flag used for determining whether intermediate
C               calculations are output; for ITPRT=0,only
C               converged results are output; for
C               ITPRT=1,results of each iteration are printed.
C          J     index for node number
C          K     index for equation number
C          M     index used in working through nodes used to
C               calculate numerical derivatives
C          MM     used to determine starting node (in relation to
C               J) for estimation of numerical derivatives
C          N     number of equations (no. of variables)
C          NJ    number of node points
C          SAVEC  saved value of C
C          SUM    intermediate value used in calculating G
C          TINIER  criterion used to avoid working with small numbers
C          TINY   criterion used to avoid working with small numbers
C          TNIEST  criterion used to avoid working with small numbers
C          WORKC  saved value of C; modified when C less than Tinier
C          X     X value described in Newman, Appendix C
C          Y     Y value described in Newman, Appendix C
C**************************************************************************
C
C DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C WHATEVER YOU WANT TO.
   SUBROUTINE DIFFEQ1(C1,P1,N1,NP2,IND,itrat)
   IMPLICIT REAL*8(A-H,O-Z)
   REAL*8 C1(14,201),P1(14,201)
   COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
   COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
  .Y(14,14),G(14),F(14),P(14,201)
   COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
   DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-15,1.0D-4/
   DATA CU,CD/1.0001,.9999/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
   ITPRT=0
   L1=N
   L2=NJ
   L3=NJ2
   NJ2=NP2
   N=N1
```

```
      NJ=NP2
      IF (ITPRT.GT.0) CALL WRTOUT2
      IF(IND.EQ.1) THEN
       DO I1=1,N1
        DO I2=1,NP2
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
        END DO
       END DO
      ELSE IF(IND.EQ.2) THEN
       DO I1=1,N1+1
        DO I2=1,NP2
         C(I1,I2)=C1(I1,I2)
        END DO
       END DO
      ELSE IF(IND.EQ.3) THEN
       DO I1=1,N1
        DO I2=1,NP2
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
        END DO
       END DO
      END IF
C
C LOOP BEGUN FOR ITERATIONS
C
      DO ITCNT=1,ITRAT
C
C COLD ARRAY SET UP
C
        DO K=1,N
         DO J=1,NJ
          COLD(K,J)=C(K,J)
         END DO
        END DO
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
        DO J=1,NJ
         IF(IND.EQ.1)THEN
          CALL FUNCT1(J)
         ELSE IF (IND.EQ.2)THEN
          CALL FUNCT2(J)
         ELSE IF(IND.EQ.3)THEN
          CALL FUNCT3(J)
         END IF
         DO I=1,N
          SUM(I)=0.0
          G(I)=-F(I)
         END DO
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
         IF (J.EQ.1) THEN
          MM=0
         ELSE IF (J.LT.NJ) THEN
          MM=-1
         ELSE
          MM=-2
         END IF
         DO M=MM,MM+2
```

```
C
C ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
      DO K=1,N
        SAVEC=C(K,J+M)
        WORKC=SAVEC
C
C DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
        IF (ABS(WORKC).LT.TINY) THEN
          IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
          C(K,J+M)=1.2*WORKC
          IF(IND.EQ.1)THEN
            CALL FUNCT1(J)
          ELSE IF(IND.EQ.2)THEN
            CALL FUNCT2(J)
          ELSE IF(IND.EQ.3)THEN
            CALL FUNCT3(J)
          END IF
          DO I=1,N
            AA(I)=-F(I)
          END DO
          C(K,J+M)=1.1*WORKC
          IF(IND.EQ.1)THEN
            CALL FUNCT1(J)
          ELSE IF(IND.EQ.2)THEN
            CALL FUNCT2(J)
          ELSE IF(IND.EQ.3)THEN
            CALL FUNCT3(J)
          END IF
          DO I=1,N
            AA(I)=AA(I)+4.0*F(I)
          END DO
          C(K,J+M)=WORKC
          IF(IND.EQ.1)THEN
            CALL FUNCT1(J)
          ELSE IF(IND.EQ.2)THEN
            CALL FUNCT2(J)
          ELSE IF(IND.EQ.3)THEN
            CALL FUNCT3(J)
          END IF
          DO I=1,N
            AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
          END DO
C
C DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
        ELSE
          C(K,J+M)=WORKC*CU
          IF(IND.EQ.1)THEN
            CALL FUNCT1(J)
          ELSE IF(IND.EQ.2)THEN
            CALL FUNCT2(J)
          ELSE IF(IND.EQ.3)THEN
            CALL FUNCT3(J)
          END IF
          DO I=1,N
            AA(I)=F(I)
          END DO
          C(K,J+M)=WORKC*CD
          IF(IND.EQ.1)THEN
            CALL FUNCT1(J)
          ELSE IF(IND.EQ.2)THEN
            CALL FUNCT2(J)
```

```
             ELSE IF(IND.EQ.3)THEN
              CALL FUNCT3(J)
             END IF
             DO I=1,N
               AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
             END DO
            ENDIF
            C(K,J+M)=SAVEC
C
C  VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
           DO I=1,N
             SUM(I)=SUM(I)+AA(I)*C(K,J+M)
             IF (M.EQ.-2) Y(I,K)=AA(I)
             IF (M.EQ.-1) A(I,K)=AA(I)
             IF (M.EQ.0) B(I,K)=AA(I)
             IF (M.EQ.1) D(I,K)=AA(I)
             IF (M.EQ.2) X(I,K)=AA(I)
           END DO
          END DO
         END DO
         DO I=1,N
          G(I)=G(I)+SUM(I)
         END DO
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
         CALL BAND2(J)
       END DO
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C  ITERATION IS BEGUN
C
       DO K=1,N
         DO J=1,NJ
          IF(DABS(C(K,J)).GT.TNIEST) THEN
           IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
          ENDIF
         END DO
       END DO
      GO TO 80
70     IF(ITPRT .GT. 1) CALL WRTOUT2
      END DO
80     CONTINUE
      DO I1=1,N
       DO I2=1,NJ
        C1(I1,I2)=C(I1,I2)
       END DO
      END DO
      N=L1
      NJ=L2
      NJ2=L3
      RETURN
      END
C
C-------------------------------------------------------
C This subroutine is the main routine for Newman method for solutio of PDE's.
C
C**********************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C**********************************************************************
C
```

```
C     CALLED BY:  MAIN CALLING PROGRAM
C
C************************************************************************
C
C     SUBROUTINES CALLED:
C           WRTOUT (FOR DATA OUTPUT)
C           BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C           FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C                 FOR A VARIABLE)
C
C************************************************************************
C
C     LIST OF IMPORTANT VARIABLES:
C
C           A      A coefficient described in Newman, Appendix C
C           AA      first, AA is F(WORKC*CU).  Later AA is the
C                 value of the derivative df/dc used in Newton's
C                 method
C           B      B coefficient described in Newman, Appendix C
C           C      variables to be solved for
C           CD      multiplication factor used in obtaining
C                 numerical derivatives
C           COLD    value of C from previous iteration
C           CU     2.0 - CD
C           D      D coefficient described in Newman, Appendix C
C           ERR    convergence criterion
C           F      value of function f(C), calculated in FUNCT
C           G      residual of f(C) calculated with updated C value
C           I      index used for equation number
C           ITCNT   index for iteration number
C           ITPRT   flag used for determining whether intermediate
C                 calculations are output; for ITPRT=0,only
C                 converged results are output; for
C                 ITPRT=1,results of each iteration are printed.
C           J      index for node number
C           K      index for equation number
C           M      index used in working through nodes used to
C                 calculate numerical derivatives
C           MM     used to determine starting node (in relation to
C                 J) for estimation of numerical derivatives
C           N      number of equations (no. of variables)
C           NJ     number of node points
C           SAVEC   saved value of C
C           SUM    intermediate value used in calculating G
C           TINIER  criterion used to avoid working with small numbers
C           TINY   criterion used to avoid working with small numbers
C           TNIEST  criterion used to avoid working with small numbers
C           WORKC   saved value of C; modified when C less than Tinier
C           X      X value described in Newman, Appendix C
C           Y      Y value described in Newman, Appendix C
C************************************************************************
C
C  DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C  IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C  IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C  DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C  NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C  WHATEVER YOU WANT TO.
      SUBROUTINE DIFFEQ2(C1,P1,N1,NP2,IND,ITCNT0)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 C1(14,201),P1(14,201)
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     .Y(14,14),G(14),F(14),P(14,201)
```

```
      COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
      COMMON/NCY/NC1
      DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-15,1.0D-4/
      DATA CU,CD/1.0001,.9999/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
      ITPRT=0
      ITCNT=0
      L1=N
      L2=NJ
      L3=NJ2
      NJ2=NP2
      N=N1
      NJ=NP2
      IF (ITPRT.GT.0) CALL WRTOUT2
      IF(IND.EQ.1) THEN
       DO I1=1,N1+3
        DO I2=1,NP2
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
        END DO
       END DO
      ELSE IF(IND.EQ.2) THEN
       DO I1=1,N1
        DO I2=1,NP2
         C(I1,I2)=C1(I1,I2)
         P(I1,I2)=P1(I1,I2)
        END DO
       END DO
      END IF
C
C LOOP BEGUN FOR ITERATIONS
C
      DO ITCNT=1,ITCNT0
C
C COLD ARRAY SET UP
C
        DO K=1,N
          DO J=1,NJ
            COLD(K,J)=C(K,J)
          END DO
        END DO
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
        DO J=1,NJ
          CALL FUNCT4(J)
          DO I=1,N
            SUM(I)=0.0
            G(I)=-F(I)
          END DO
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
          IF (J.EQ.1) THEN
            MM=0
          ELSE IF (J.LT.NJ) THEN
            MM=-1
          ELSE
```

```
        MM=-2
      END IF
      DO M=MM,MM+2
C
C  ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
        DO K=1,N
          SAVEC=C(K,J+M)
          WORKC=SAVEC
C
C  DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
          IF (ABS(WORKC).LT.TINY) THEN
            IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
            C(K,J+M)=1.2*WORKC
            CALL FUNCT4(J)
            DO I=1,N
              AA(I)=-F(I)
            END DO
            C(K,J+M)=1.1*WORKC
            CALL FUNCT4(J)
            DO I=1,N
              AA(I)=AA(I)+4.0*F(I)
            END DO
            C(K,J+M)=WORKC
            CALL FUNCT4(J)
            DO I=1,N
              AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
            END DO
C
C  DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
          ELSE
            C(K,J+M)=WORKC*CU
            CALL FUNCT4(J)
            DO I=1,N
              AA(I)=F(I)
            END DO
            C(K,J+M)=WORKC*CD
            CALL FUNCT4(J)
            DO I=1,N
              AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
            END DO
          ENDIF
          C(K,J+M)=SAVEC
C
C  VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
          DO I=1,N
            SUM(I)=SUM(I)+AA(I)*C(K,J+M)
            IF (M.EQ.-2) Y(I,K)=AA(I)
            IF (M.EQ.-1) A(I,K)=AA(I)
            IF (M.EQ.0) B(I,K)=AA(I)
            IF (M.EQ.1) D(I,K)=AA(I)
            IF (M.EQ.2) X(I,K)=AA(I)
          END DO
        END DO
      END DO
      DO I=1,N
        G(I)=G(I)+SUM(I)
      END DO
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
```

```
      CALL BAND2(J)
     END DO
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C  ITERATION IS BEGUN
C
     DO K=1,N
       DO J=1,NJ
        IF(DABS(C(K,J)).GT.TNIEST) THEN
          IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
        ENDIF
       END DO
     END DO
    GO TO 80
70   IF(ITPRT .GT. 0) CALL WRTOUT2
    END DO
80   CONTINUE
    DO I1=1,N
     DO I2=1,NJ
      C1(I1,I2)=C(I1,I2)
     END DO
    END DO
    N=L1
    NJ=L2
    NJ2=L3
    RETURN
    END

    SUBROUTINE WRTOUT2
C***********************************************************************
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
   .Y(14,14),G(14),F(14),P(14,201)
    COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
    IF (ITCNT.NE.0) WRITE (*,99)ITCNT
    WRITE (*,100)
    DO K=1,NJ2,2
      WRITE(*,101)K,(C(I,K),I=1,N)
    END DO
99   FORMAT(' ITCNT=',I2)
100  FORMAT('  J      C1           C2           C3          '
   &,' C4          C5          C6'/)
101   FORMAT(1X,I3,6(1PE16.8))
    RETURN
    END
C   BLOCK TRIDIAGONAL MATRIX SUBROUTINE
    SUBROUTINE BAND2(J)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION E(14,14,201)
    COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
   .Y(14,14),G(14),F(14),P(14,201)
    COMMON/BND2/N,NJ2,NJ,ITPRT,ITCNT
  101 FORMAT('0DETERM=0 AT J=',I4)
    IF (J.EQ.1) THEN
      NP1=N+1
      DO I=1,N
        D(I,2*N+1)=G(I)
        DO L=1,N
         D(I,L+N)=X(I,L)
        END DO
      END DO
      CALL MATINV2(N,2*N+1,DETERM)
C       IF(DETERM.EQ.0) WRITE (2,101) J
```

```
        DO K=1,N
          E(K,NP1,1)=D(K,2*N+1)
          DO L=1,N
            E(K,L,1)=-D(K,L)
            X(K,L)=-D(K,L+N)
          END DO
        END DO
        RETURN
      ELSE IF(J.EQ.2)THEN
        DO I=1,N
          DO K=1,N
            DO L=1,N
              D(I,K)=D(I,K)+A(I,L)*X(L,K)
            END DO
          END DO
        END DO
      ELSE IF(J.EQ.NJ2) THEN
        DO I=1,N
          DO L=1,N
            G(I)=G(I)-Y(I,L)*E(L,NP1,J-2)
            DO M=1,N
              A(I,L)=A(I,L)+Y(I,M)*E(M,L,J-2)
            END DO
          END DO
        END DO
      ENDIF
      DO I=1,N
        D(I,NP1)=-G(I)
        DO L=1,N
          D(I,NP1)=D(I,NP1)+A(I,L)*E(L,NP1,J-1)
          DO K=1,N
            B(I,K)=B(I,K)+A(I,L)*E(L,K,J-1)
          END DO
        END DO
      END DO
      CALL MATINV2(N,NP1,DETERM)
C       IF(DETERM.EQ.0) WRITE(2,101) J
      DO K=1,N
        DO M=1,NP1
          E(K,M,J)=-D(K,M)
        END DO
      END DO
      IF(J.EQ.NJ2)THEN
        DO K=1,N
          C(K,J)=E(K,NP1,J)
        END DO
        DO M=NJ2-1,1,-1
          DO K=1,N
            C(K,M)=E(K,NP1,M)
            DO L=1,N
              C(K,M)=C(K,M)+E(K,L,M)*C(L,M+1)
            END DO
          END DO
        END DO
        DO L=1,N
          DO K=1,N
            C(K,1)=C(K,1)+X(K,L)*C(L,3)
          END DO
        END DO
      ENDIF
      RETURN
      END
C
C     MATRIX INVERSION SUBROUTINE
```

```
      SUBROUTINE MATINV2(N,M,DETERM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ID(14)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29)
      DETERM=1.0
      DO I=1,N
       ID(I)=0
      END DO
      DO NN=1,N
        BMAX=0.0
        DO I=1,N
          IF(ID(I).EQ.0)THEN
            DO J=1,N
              IF(ID(J).EQ.0)THEN
                IF(DABS(B(I,J)).GT.BMAX) THEN
                  BMAX=DABS(B(I,J))
                  IROW=I
                  JCOL=J
                ENDIF
              ENDIF
            END DO
          ENDIF
        END DO
        IF(BMAX.EQ.0.0)THEN
          DETERM=0.0
          RETURN
        ENDIF
        ID(JCOL)=1
        IF(JCOL.NE.IROW) THEN
    9      DO J=1,N
            SAVE=B(IROW,J)
            B(IROW,J)=B(JCOL,J)
            B(JCOL,J)=SAVE
          END DO
          DO K=1,M
            SAVE=D(IROW,K)
            D(IROW,K)=D(JCOL,K)
            D(JCOL,K)=SAVE
          END DO
        ENDIF
        FF = 1.0/B(JCOL,JCOL)
        DO J=1,N
          B(JCOL,J)=B(JCOL,J)*FF
        END DO
        DO K=1,M
          D(JCOL,K)=D(JCOL,K)*FF
        END DO
        DO I=1,N
          IF(I.NE.JCOL)THEN
   15         FF = B(I,JCOL)
            DO J=1,N
             B(I,J) = B(I,J) -FF*B(JCOL,J)
            END DO
            DO K=1,M
              D(I,K) = D(I,K) -FF*D(JCOL,K)
            END DO
          ENDIF
        END DO
       END DO
       RETURN
       END
C
C------------------------------------------------
C
```

```
        SUBROUTINE CUBSPL(X,Y,N,IEND,SUM1)
    IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------
C THIS SUBROUTINE COMPUTE THE MATRIX FOR FINDING THE COEFFICIENTS OF A
C CUBIC SPLINE THROUGH A SET OF DATA. THE SYSTEM THEN IS SOLVED THE
C SECOND DERIVTITVE VALUE.
C
C-----------------------------------------------------------
C
C X,Y   ARRAY OF X AND Y VALUES TO BE FITTED
C S     ARRAY OF SECOND DERIVTITVE VALUSES AT THE POINTS
C N     NUMBER OF POINTS
C IEND  TYPE OF END CONDITION TO BE USED
C       IEND=1, LINEAR ENDS, S(1)=S(N)=0.
C       IEND=2, PARABOLIC ENDS, S(1)=S(2)=, S(N)=S(N-1)
C       IEND=3, CUBIC ENDS, S(1),S(N) ARE EXTRAPOLATED
C F     AUGMENTED MATRIX OF COEFFICIENTS AND R.H.S FOR FINDING S
C
C-----------------------------------------------------------
C
    REAL*4 SUM1
        REAL*8 X(N),Y(N),S(101),F(101,4),A(100),B(100),C(100),D(100),
   1 DX1,DY1,DX2,DY2,DXN1,DXN2
        INTEGER N,IEND,NM1,NM2,I,J
C
C-----------------------------------------------------------
C
C COMPUTE FOR THE N-2 ROWES
C
        NM2=N-2
        NM1=N-1
        DX1=X(2)-X(1)
        DY1=(Y(2)-Y(1))/DX1*6.0
        DO I=1,NM2
         DX2=X(I+2)-X(I+1)
         DY2=(Y(I+2)-Y(I+1))/DX2*6.0
         F(I,1)=DX1
         F(I,2)=2.0*(DX1+DX2)
         F(I,3)=DX2
         F(I,4)=DY2-DY1
         DX1=DX2
         DY1=DY2
        END DO
C
C-----------------------------------------------------------
C ADJUST FIRST AND LAST ROWS APPROPRIATE TO END CONDITION.
C
        GO TO (20,50,80), IEND
C
C FOR IEND=1, NO CHANGE IS NEEDED
C
 20     GO TO 100
C
C FOR IEND=2
C
 50     F(1,2)=F(1,2)+X(2)-X(1)
        F(NM2,2)=F(NM2,2)+X(N)-X(NM1)
        GO TO 100
C
C FOR IEND=3
C
 80     DX1=X(2)-X(1)
        DX2=X(3)-X(2)
```

```
          F(1,2)=(DX1+DX2)*(DX1+2*DX2)/DX2
          F(1,3)=(DX2*DX2-DX1*DX1)/DX2
          DXN2=X(NM1)-X(NM2)
          DXN1=X(N)-X(NM1)
          F(NM2,1)= (DXN2*DXN2-DXN1*DXN1)/DXN2
          F(NM2,2)=(DXN1+DXN2)*(DXN1+2.*DXN2)/DXN2
      GO TO 100
C
C-----------------------------------------------------------
C
 100      DO I=2,NM2
           F(I,2)=F(I,2)-F(I,1)/F(I-1,2)*F(I-1,3)
           F(I,4)=F(I,4)-F(I,1)/F(I-1,2)*F(I-1,4)
          END DO
C
C BACK SUBSTITUTION
C
          F(NM2,4)=F(NM2,4)/F(NM2,2)
          DO I=2,NM2
           J=NM1-I
           F(J,4)=(F(J,4)-F(J,3)*F(J+1,4))/F(J,2)
          END DO
C
C-----------------------------------------------------------
C
C NOW PUT THE VALUES INTO THE S VECTORS
          DO I=1,NM2
           S(I+1)=F(I,4)
          END DO
C
C-----------------------------------------------------------
C
C GET S(1) AND S(N)
C
          GO TO (150,160,170),IEND
C
C-----------------------------------------------------------
C
 150      S(1)=0.
          S(N)=0.
      GO TO 200
C
C-----------------------------------------------------------
C
 160      S(1)=S(2)
          S(N)=S(N-1)
          GO TO 200
C
C-----------------------------------------------------------
C
 170      S(1)=((DX1+DX2)*S(2)-DX1*S(3))/DX2
          S(N)=((DXN2+DXN1)*S(NM1)-DXN1*S(NM2))/DXN2
C
C-----------------------------------------------------------
C
C FIND THE INTEGRATION
C
C FIND THE COEFFICIENTS
C
 200  CONTINUE
          DO I=1,N-1
           A(I)=(S(I+1)-S(I))/(6*(X(I+1)-X(I)))
           B(I)=S(I)/2
           C(I)=(Y(I+1)-Y(I))/(X(I+1)-X(I))-
```

```
     1  (2*(X(I+1)-X(I))*S(I)+(X(I+1)-X(I))*S(I+1))/6
          D(I)=Y(I)
        END DO
      SUM=0.
      DO I=1,N-1
          PART1=A(I)/4*(X(I+1)-X(I))**4
          PART2=B(I)/3*(X(I+1)-X(I))**3
          PART3=C(I)/2*(X(I+1)-X(I))**2
          PART4=D(I)*(X(I+1)-X(I))
       SUM=SUM+PART1+PART2+PART3+PART4
        END DO
       SUM1=SUM
      RETURN
      END
C
C INTEGRATION BY CUBIC SPLINE
C
        SUBROUTINE SIMPS(F,N,H,RESULT)
      IMPLICIT REAL*8(A-H,O-Z)
C
C-----------------------------------------------------------
C       SUBROUTINE SIMPS:
C                      THIS SUBROUTINE PERFORMS SIMPSON'S RULE INTEGRATION
C       OF A FUNCTION DEFINED BY A TABLE OF EQUISPACED VALUES.
C
C-----------------------------------------------------------
C
C       PARAMETERS ARE:
C                F        -ARRAY OF VALUES OF THE FUNCTION
C                N        -NUMER OF PONTS
C                H        -THE UNIFORM SPACING BETWEEN X VALUES
C                RESULT          -ESTIMATE OF THE INTEGRAL
C-----------------------------------------------------------
        REAL*8 F(N),H,RESULT
        INTEGER N,NPANNEL,NHALF,NBEGIN,NEND
C
C-----------------------------------------------------------
C       CHECK TO SEE IF NUMBER OF PANELS IS EVEN. NUMBER OF PANELS IS N-1.
C
        NPANEL=N-1
        NHALF=NPANEL/2.
        NBEGIN=1
        RESULT=0.
        IF((NPANEL-2*NHALF) .NE. 0) THEN
C
C NUMBER OF PANEL IS ODD. USE 3/8 RULE ON FIRST THREE, 1/3 RULE ON REST.
C
         RESULT=3.0*H/8.*(F(1)+3.*F(2)+3.*F(3)+F(4))
         NBEGIN=4
         IF(N.EQ.4)RETURN
        END IF
C
C-----------------------------------------------------------
C
C APPLY 1/3 RULE- ADD IN FIRST, SECOND AND LAST VALUES.
C
        RESULT=RESULT+H/3.0*(F(NBEGIN)+4.0*F(NBEGIN+1)+F(N))
        NBEGIN=NBEGIN+2
        IF(NBEGIN.EQ.N)RETURN
C
C-----------------------------------------------------------
C
C THE PATTERN AFTER NBEGIN+2 IS REPETITIVE. GET NEND, THE PLACE TO STOP
C
```

```
        NEND=N-2
        DO 10 I=NBEGIN,NEND,2
10        RESULT=RESULT+H/3.0*(2.0*F(I)+4.*F(I+1))
        RETURN
        END

    REAL*8 FUNCTION TEMPIN(TIME)
    IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES CURRENT INLET TEMPERATURE
C BASED ON THE TIME FOR A TEMPERATURE CHARACTERIZATION
C COMPARSION RUN
      IF(TIME.LT.0.2) THEN
        T0 = 74.713
        T1 = 33.11
        T2 = -775.68
        T3 = 5027.2
        T4 = 0.0
        T5 = 0.0
      ELSE
        T0 = 25.789
        T1 = 396.87
        T2 = -403.93
        T3 = 251.85
        T4 = -85.612
        T5 = 11.979
      ENDIF
      X = TIME
      TEMPIN = T0+T1*X+T2*X**2.+T3*X**3.+T4*X**4.+T5*X**5.
      TEMPIN = TEMPIN+460.
      RETURN
      END
```

APPENDIX F

VACMOL FORTRAN CODE

```
C
C  RELAXATION METHOD
C
C  PROGRAM COND
C  DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 201 NODE POINTS; IF
C  IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C  IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C  DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C  NODE POINTS ARE DESIRED, CHANGE THE 201 IN THE DIMENSION STATEMENTS TO
C  WHATEVER YOU WANT.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PL(11,201),QP(14,201),PL_TIME,T_TIME
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201),Z,U_F1
      COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,II,LL,ITEST1,K_F
      COMMON/INDIC/L1,L2,L3,L4,L5,L6
      COMMON/PR_OLD_Z/VISC_P(201),RO_P(201),CP_P(201),CON_FPP(201),
     . CON_LPP(201),D_LPP(4,201),H_FP(201),H_TTP(201),H_IIP(201),
     . H_OWPP(201),H_FWPP(201),Q(4,201),RATE_C1(201),AMOL_W(201)
      COMMON/Q_SOURCE/DELH
      COMMON/BC_PRE/BC_L55,TIME
      DATA DELH0/270.0/
      OPEN (50,FILE='ASCII.DAT',STATUS='NEW',RECL=32766)
      OPEN (66,FILE='CON_CO2.DAT',STATUS='NEW')
      OPEN (88,FILE='CON_H2O.DAT',STATUS='NEW')
      OPEN (99,FILE='TEM.DAT',STATUS='NEW')
C WRITE HEADER DATA TO ACSII FILE
      WRITE(50,42)
C
C        INITIALIZE THE C ARRAY WITH THE INITIAL GUESSES OF THE SOLUTION
C
       CALL INITIAL_Z
       ICOT=1
       T_TIME=0
       DO 301 LL=1,20000
        IF(C(L1,1).LT.(400.+460.))THEN
         DELH=DELH0
        ELSE
         DELH=0.
        END IF
        CALL DIFFEQ
        ITEST1=0
        T_TIME=T_TIME+DELT*Z/U_F1*60.
        TIME=T_TIME
        IF(LL.EQ.200)THEN
         DELT1=DELT1*2
         DELT=DELT*2
        END IF
        IF(LL.EQ.400)THEN
         DELT1=DELT1*2
         DELT=DELT*2
        END IF
        IF(LL.EQ.600)THEN
         DELT1=DELT1*2
         DELT=DELT*2
        END IF
        IF(LL.EQ.800)THEN
         DELT1=DELT1*2
         DELT=DELT*2
```

```
      END IF
      IF(LL.EQ.1000)THEN
       DELT1=.0001
       DELT=U_F1*DELT1/Z
      END IF
      DO 74 M2=1,NJ
        QP(1,M2)=C(1,M2)*C(L5,M2)
        QP(2,M2)=C(2,M2)*C(L5,M2)
        QP(3,M2)=C(L1,M2)-460.
        QP(4,M2)=C(L2,M2)-460.
        QP(5,M2)=C(L3,M2)-460.
        QP(6,M2)=C(L4,M2)
        QP(7,M2)=C(L5,M2)
        QP(8,M2)=C(3,M2)
        QP(9,M2)=C(4,M2)
74       CONTINUE
     WRITE(*,*)LL,II,T_TIME
     WRITE(*,*)QP(1,1),QP(1,42),QP(1,84),QP(1,126),QP(1,168)
     WRITE(*,*)QP(2,1),QP(2,42),QP(2,84),QP(2,126),QP(2,168)
     WRITE(*,*)QP(3,1),QP(3,42),QP(3,84),QP(3,126),QP(3,168)
     WRITE(*,*)QP(4,1),QP(4,42),QP(4,84),QP(4,126),QP(4,168)
     WRITE(*,*)QP(5,1),QP(5,42),QP(5,84),QP(5,126),QP(5,168)
     WRITE(*,*)QP(6,1),QP(6,42),QP(6,84),QP(6,126),QP(6,168)
     WRITE(*,*)QP(7,1),QP(7,42),QP(7,84),QP(7,126),QP(7,168)
     WRITE(*,*)QP(8,1),QP(8,42),QP(8,84),QP(8,126),QP(8,168)
     WRITE(*,*)QP(9,1),QP(9,42),QP(9,84),QP(9,126),QP(9,168)
     DO 520 J=1,151
      PL(1,J)=QP(1,J)
      PL(2,J)=QP(2,J)
      PL(3,J)=QP(4,J)
      PL(4,J)=QP(6,J)
520    PL(5,J)=QP(7,J)
    PL_TIME=T_TIME
    print*,pl_time
    ISAMP=25
    IF((LL/ISAMP)*ISAMP.EQ.LL) THEN
    WRITE(50,43)PL_TIME,PL(1,1),PL(1,42),PL(1,84),PL(1,126),PL(1,168)
   &            ,PL(2,1),PL(2,42),PL(2,84),PL(2,126),PL(2,168)
   &            ,PL(3,1),PL(3,42),PL(3,84),PL(3,126),PL(3,168)
   &            ,PL(4,1),PL(4,42),PL(4,84),PL(4,126),PL(4,168)
   &            ,PL(5,1),PL(5,42),PL(5,84),PL(5,126),PL(5,168)
    ENDIF
       DO 78 I1=1,N
        DO 78 I2=1,NJ
         P(I1,I2)=C(I1,I2)
78      CONTINUE
200    CONTINUE
301     CONTINUE
42    FORMAT(1X,'TIME,ppCO2@1,ppCO2@2,ppCO2@3,ppCO2@4,ppCO2@5',
   &          ',ppH2O@1,ppH2O@2,ppH2O@3,ppH2O@4,ppH2O@5',
   &          ',T gas@1,T gas@2,T gas@3,T gas@4,T gas@5',
   &          ',T bed@1,T bed@2,T bed@3,T bed@4,T bed@5',
   &          ',Twall@1,Twall@2,Twall@3,Twall@4,Twall@5')
43    FORMAT(1X,26(E15.5,','))
44    FORMAT(/5X,5(E15.5,3X))
45    FORMAT(/5X,5(E15.5,3X))
46    FORMAT(/5X,5(E15.3,3X))
     STOP
     END
C
C
    SUBROUTINE INITIAL_Z
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 K_F(4),M_AVE,M_W(4),C_FO(4),IN_L1,IN_L2,IN_L3,IN_L4,
```

```
     . IN_L5,IN_L6,Q2(4)
       REAL*8 C1(14,201),P1(14,201)
       COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201),Z,U_F1
       COMMON/PROP_B_Z/D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,
     . CP_WA,CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,
     . X_WI_K,D_LMI,D_LMA
       COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4),M_W
       COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,II,LL,ITEST1,K_F
       COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
       COMMON/PR_OLD_Z/ VISC_P(201)
       COMMON/INDIC/L1,L2,L3,L4,L5,L6
       COMMON/NCY/NC1
       COMMON/BC_PRE/BC_L55,TIME
       COMMON/GAS/INERT,NCOMP
       COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
       DATA INERT,NCOMP/1,3/
C
C INITIALIZATION
C
       DATA Z,D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,CP_WA,
     . CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,X_WI_K
     . /1.4,0.5412D0,0.5620D0,0.46D0,32.875D0,
     . .29D0,.19D0,.109848D0,.21D0,.21D0,489.0D0,3.5D0,16.0D0,
     . 5.41666667D-3,4.1666667D-2,2.083333D-2/
       DATA PT,RA,RAV,EPSEX,EPSIN,RHOS,ALPHA1
     . /790.2014,4.72441E-3,4.72441E-3,.373,.317,43.,.575/
       DATA AINT,CP_S,RO_S,R_P,D_P,CON_S,HEAT
     . /635.D0,0.2D0,70.0,4.724410D-3,9.44882D-3,.1,-19000.D0,
     . -28000.D0,-8988.0D0,-8988./
       DATA G_F,P_TOT,TO
     . /58.6,15.28,538.0D0/
       DATA BC_C/.17,0.3,14.81,0.0/
       DATA BC_L1,BC_L2,BC_L3,BC_L5,BC_L6
     . /538.,538.,538.,25.,25./
       DATA IN_L1,IN_L2,IN_L3,IN_L4,IN_L5,IN_L6
     . /538.,538.,538.,0.0,790.20415,790.20415/
       DATA M_W/44.,18.,28.,32/
       DATA NC,NJ,DELZ1,DELT1/4,169,.00833333,1.0E-5/
C        DATA NC,NJ,DELZ1,DELT1/4,51,.01666667,.001/
       DATA R,PI/555.0,3.141593D0/
       DATA K_F/.017,.0035,.05,.05/
       NC1=NC
       N1=NC*2+5
       ALF=(1.0D0-EPSEX)/EPSEX
       PT=P_TOT
       SUM_Y=0
       DO 10 I=1,NC
        Y_F=BC_C(I)/P_TOT
        SUM_Y=SUM_Y+Y_F
        BC_C(I)=Y_F
        C_FO(I)=BC_C(I)
 10     CONTINUE
       L1=2*NC+1
       L2=L1+1
       L3=L2+1
       L4=L3+1
       L5=L4+1
       L6=L5+1
        BC_L55=BC_L5
        VISC_F=VIS(IN_L1)*60
        R_CRT=BC_L5/IN_L5
        IF(R_CRT.LT.0.53) THEN
         BC_L5=.53*IN_L5
```

```
       END IF
       DO 413 I2=1,NJ
        C1(1,I2)=0.0
        P1(1,I2)=0.0
        C1(2,I2)=BC_L5
        P1(2,I2)=IN_L5
        C1(3,I2)=IN_L1
        P1(3,I2)=IN_L1
        VISC_P(I2)=VISC_F
413     CONTINUE
       CALL DIFFEQ1(C1,P1,NJ,2)
       N=N1
       DO 423 I2=1,NJ
        C(L4,I2)=C1(1,I2)
        C(L5,I2)=C1(2,I2)
423     CONTINUE
       U_F1=C(L4,1)
       DELZ=DELZ1/Z
       DELT=U_F1*DELT1/Z
      DO 22 I1=1,NC
       C(I1,1)=C_FO(I1)
22     CONTINUE
      DO 24 I1=1,NC
       DO 24 I2=1,NJ
        C(I1,I2)=C_FO(I1)
        P(I1,I2)=C_FO(I1)
24     CONTINUE
      DO 30 I=1,NJ
       P(L1,I)=IN_L1
       P(L2,I)=IN_L2
       P(L3,I)=IN_L3
       P(L4,I)=IN_L4
       P(L5,I)=IN_L5
       P(L6,I)=IN_L6
       C(L1,I)=BC_L1
       C(L2,I)=BC_L2
       C(L3,I)=BC_L3
       C(L6,I)=BC_L6
30    CONTINUE
     CALL IST_Z(1,C(L2,1),P(L5,1),C(1,1),C(2,1),C(3,1),C(4,1),Q2)
     DO 40 I=NC+1,NC+NC
      DO 40 I1=1,NJ
       C(I,I1)=Q2(I-NC)*RO_S
       P(I,I1)=C(I,I1)
40   CONTINUE
     RETURN
     END
C
C
     SUBROUTINE FUNCT(J)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K_F(4),M_AVE,M_W(5),KGAS,K_FP(4,201),C1(14,201),
    . P1(14,201),REY_P(201),KEFF,D_L(4),YO(4)
      DIMENSION Q2(4),C2(14,201),PE_N_M(4)
      COMMON/PR_OLD_Z/ VISC_P(201),RO_P(201),CP_P(201),CON_FPP(201),
    . CON_LPP(201),D_LPP(4,201),H_FP(201),H_TTP(201),H_IIP(201),
    . H_OWPP(201),H_FWPP(201),Q(4,201),RATE_C1(201),AMOL_W(201),
    . D_LAV(201)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
    . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201),Z,U_F1
      COMMON/PROP_B_Z/D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,CP_WA,
    . CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,X_WI_K,
    . D_LMI,D_LMA
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4),M_W
```

```
      COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,II,LL,ITEST1,K_F
      COMMON/INDIC/L1,L2,L3,L4,L5,L6
      COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
      COMMON/Q_SOURCE/DELH
      COMMON/GAS/INERT,NCOMP
      COMMON/BC_PRE/BC_L55,TIME
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA R,PI/555.0,3.141593D0/
      DATA R1,CONV/555.0,1/
      DATA FAC,GC/2.78450526316,416975040.0/

C
C
C MASS TRANSFER FROM THE BULK OF GAS SRTEAM TO THE SURFACE OF ABSORBENT
C
C ESTABLISHED COEFFICIENT MATRIX
     IF(TIME.GE.6.0)GO TO 501
     IF(ITEST.EQ.0)THEN
       D_T=D_E+D_I
       X_W=D_E-D_I
       A_C=2*D_I/((D_I+D_E)*X_W)
       A_INS=2*D_E/((D_I+D_E)*X_W)
       ITEST=1
       U_F1=C(L4,1)
       H_FW=6.0
       H_OW=0.1
     ENDIF
     IF(J.EQ.1 .AND. ITEST1.NE.ITCNT)THEN
       RO_F=C(L5,1)/(C(L1,1)*R)
       DO 10 I=1,NJ
         DO I1=1,NC
           YO(I1)=C(I1,J)
         END DO
         M_AVE=0.
         DO I1=1,NC
          M_AVE=M_AVE+M_W(I1)*YO(I1)
         END DO
         AMOL_W(I)=M_AVE
         RO_F=C(L5,J)/(C(L1,J)*R)
         TEMP=C(L1,J)
         GN=C(L4,J)*RO_F/60.
         VISC_P(I)=VIS(TEMP)*60
         REY=RE(GN,TEMP,AMOL_W(I))
         IF(REY.LT.1.0) GO TO 10
         CP_P(I)=CPGAS(TEMP,YO)
         H_FP(I)=HFILM(AMOL_W,TEMP,REY,CP_P(I))*60.
         PP=C(L5,J)
         CON_LPP(I)=EFFK(GN,TEMP,CP_P(J),RO_F,PP,YO)*60.
         DO I1=1,NC
           D_LPP(I1,I)=EFFD(I1,GN,TEMP,RO_F,PP,YO)*60.
           IF(D_LPP(I1,J).LT.0) THEN
             D_LPP(I1,I)=D_LPP(I1,J-1)
           END IF
         END DO
 10      CONTINUE
       ITEST1=ITCNT
      END IF
      GO TO 502
 501  DO I1=1,NC
       YO(I1)=C(I1,J)
      END DO
      M_AVE=0.
      DO I1=1,NC
```

```
     M_AVE=M_AVE+M_W(I1)*YO(I1)
    END DO
    AMOL_W(I)=M_AVE
502 R_CRT=BC_L55/C(L5,NJ)
    IF(R_CRT.LT.0.53) THEN
     BC_L5=.53*C(L5,NJ)
    ELSE
     BC_L5=BC_L55
    END IF
    RO_F=C(L5,J)/(C(L1,J)*R)
    CP_F=CP_P(J)
    H_FS=H_FP(J)
    CON_L=CON_LPP(J)
    D_L_AVE=0.
    SUM=0
    DO I1=1,NC
     D_L(I1)=D_LPP(I1,J)
     IF(D_L(I1).NE.0)THEN
       SUM=SUM+1
       D_L_AVE=D_L_AVE+D_L(I1)
     END IF
    END DO
    D_L_AVE=D_L_AVE/SUM
    D_LAV(J)=D_L_AVE
    VISC_F=VISC_P(J)
    PE_N_S=U_F1*RO_S*CP_S*Z/CON_S
    PE_N_H=U_F1*RO_F*CP_F*Z/CON_L
    F3=ALF*H_FS*Z*AINT/(RO_F*CP_F*U_F1)
    F4=H_FW*Z*4/(U_F1*D_I*EPSEX*RO_F*CP_F)
    F5=H_FS*AINT*Z/(U_F1*RO_S*CP_S)
    F6=AINT*Z/(RO_S*CP_S*U_F1)
    F7=Z*H_FW*A_C/U_F1/RO_WA/CP_WA
    F8=Z*H_OW*A_INS/U_F1/RO_WA/CP_WA
    F9=Z/((1-EPSEX)*RO_S*CP_S*U_F1*PI*D_I/2*D_I/2*Z)
    CALL IST_Z(1,C(L2,J),C(L5,J),C(1,J),C(2,J),C(3,J),C(4,J),Q2)
    DO 20 I=1,NC
     Q(I,J)=Q2(I)*RO_S
20   CONTINUE
    C_K=2.7845/(150*(1-EPSEX)**2*VISC_P(J)/D_P/D_P/
   .EPSEX/EPSEX/GC)
    IF(J.EQ.1)THEN
      DO 50 M1=1,NC
        F1=K_F(M1)*AINT*Z/U_F1
        F(NC+M1)=F1*(Q(M1,J)-C(NC+M1,J))-(C(NC+M1,J)-P(NC+M1,J))/DELT
50      CONTINUE
      FLUX1=0
      FLUX2=0
      DO 60 M1=1,NC
        FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
        FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
60      CONTINUE
      RATE_C1(J)=FLUX2
      TOT_C=0.
      DO 40 M1=1,NC-1
       F(M1)=(-D_L(M1)/DELZ**2*(C(M1,J+1)-2*C(M1,J)+C(M1,J+1))+
   .   C(L4,J)/(2.0*DELZ)*(C(M1,J+1)-C(M1,J+1))+1.0/DELT*(C(M1,J)-
   .   P(M1,J)))-ALF/RO_F*(C(M1,J)*RATE_C1(J)-AINT*K_F(M1)*(Q(M1,J)-
   .   C(NC+M1,J)))
       TOT_C=TOT_C+C(M1,J)
40      CONTINUE
      F(NC)=1.-TOT_C-C(NC,J)

      F(L1)=1.0D0/PE_N_H/DELZ**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J+1))-
   .  1.0D0/(DELZ*U_F1)*(C(L1,J)-C(L1,J))*C(L4,J)-F3*(C(L1,J)-
```

```
    . C(L2,J))-F4*(C(L1,J)-C(L3,J))-(C(L1,J)-P(L1,J))/DELT

      F(L2)=1.0D0/PE_N_S/DELZ**2*(C(L2,J+1)-2*C(L2,J)+C(L2,J+1))+
    . F5*(C(L1,J)-C(L2,J))-F6*FLUX1+DELH*F9-(C(L2,J)-
    . P(L2,J))/DELT

      F(L3)=F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO)-(C(L3,J)-
    . P(L3,J))/DELT

     F(L4)=(C(L5,J+1)-C(L5,J))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
    . (C(L4,J)-C(L4,J))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
    . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
    . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)

     F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J+1)-
    . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J+1)-C(L5,J))/DELZ1+
    . C(L5,J)*(C(L4,J)-C(L4,J))/DELZ1-C(L5,J)/C(L1,J)*((C(L1,J)-
    . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J+1))+
    . C(L4,J)*(C(L1,J+1)-C(L1,J+1))/(2*DELZ1))+R1*C(L1,J)*ALF*
    . RATE_C1(J)


C
C-----------------------------------------------------------
C THE LAST ROW
C
     ELSE IF(J .EQ. NJ) THEN
      DO 90 M1=1,NC
       F1=K_F(M1)*AINT*Z/U_F1
       F(NC+M1)=F1*(Q(M1,J)-C(NC+M1,J))-(C(NC+M1,J)-P(NC+M1,J))/DELT
90     CONTINUE
      FLUX1=0
      FLUX2=0
      DO 100 M1=1,NC
       FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
       FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
100     CONTINUE
      RATE_C1(J)=FLUX2
      TOT_C=0.
      DO 80 M1=1,NC
       F(M1)=(-D_L(M1)/DELZ**2*(C(M1,J-1)-2*C(M1,J)+C(M1,J-1))+
    .  C(L4,J)/(2.0*DELZ)*(C(M1,J-1)-C(M1,J-1))+1.0/DELT*(C(M1,J)-
    .  P(M1,J)))-ALF/RO_F*(C(M1,J)*RATE_C1(J)-AINT*K_F(M1)*(Q(M1,J)-
    .  C(NC+M1,J)))
       TOT_C=TOT_C+C(M1,J)
80     CONTINUE
      F(NC)=1.-TOT_C-C(NC,J)

      F(L1)=1.0D0/PE_N_H/DELZ**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J-1))-
    . 1.0D0/(DELZ*U_F1)*(C(L1,J)-C(L1,J-1))*C(L4,J)-F3*(C(L1,J)-
    . C(L2,J))-F4*(C(L1,J)-C(L3,J))-(C(L1,J)-P(L1,J))/DELT

      F(L2)=1.0D0/PE_N_S/DELZ**2*(C(L2,J-1)-2*C(L2,J)+C(L2,J-1))+
    . F5*(C(L1,J)-C(L2,J))-F6*FLUX1+DELH*F9-(C(L2,J)-
    . P(L2,J))/DELT

      F(L3)=F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO)-(C(L3,J)-
    . P(L3,J))/DELT


     F(L4)=(BC_L5-C(L5,J))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
    . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
    . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
    . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
```

```
    F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(BC_L5-
  . 2*C(L5,J)+C(L5,J-1))+C(L4,J)*(BC_L5-C(L5,J))/DELZ1+
  . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L5,J)/C(L1,J)*((C(L1,J)-
  . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J-1))+
  . C(L4,J)*(C(L1,J-1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*
  . RATE_C1(J)

C
C-------------------------------------------------------------
C
C INTERIOR ROWS
C
    ELSE
     DO 130 M1=1,NC
      F1=K_F(M1)*AINT*Z/U_F1
      F(NC+M1)=F1*(Q(M1,J)-C(NC+M1,J))-(C(NC+M1,J)-P(NC+M1,J))/DELT
 130   CONTINUE
     FLUX1=0
     FLUX2=0
     DO 140 M1=1,NC
      FLUX1=FLUX1+HEAT(M1)*(K_F(M1)*(Q(M1,J)-C(NC+M1,J)))
      FLUX2=FLUX2+AINT*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
 140   CONTINUE
     RATE_C1(J)=FLUX2
     TOT_C=0.
     DO 120 M1=1,NC
      F(M1)=(-D_L(M1)/DELZ**2*(C(M1,J+1)-2*C(M1,J)+C(M1,J-1))+
  .   C(L4,J)/(2.0*DELZ)*(C(M1,J+1)-C(M1,J-1))+1.0/DELT*(C(M1,J)-
  .   P(M1,J)))-ALF/RO_F*(C(M1,J)*RATE_C1(J)-AINT*K_F(M1)*(Q(M1,J)-
  .   C(NC+M1,J)))
      TOT_C=TOT_C+C(M1,J)
 120   CONTINUE
     F(NC)=1.-TOT_C-C(NC,J)

     F(L1)=1.0D0/PE_N_H/DELZ**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J-1))-
  .  1.0D0/(DELZ*U_F1)*(C(L1,J)-C(L1,J-1))*C(L4,J)-F3*(C(L1,J)-
  .  C(L2,J))-F4*(C(L1,J)-C(L3,J))-(C(L1,J)-P(L1,J))/DELT

     F(L2)=1.0D0/PE_N_S/DELZ**2*(C(L2,J+1)-2*C(L2,J)+C(L2,J-1))+
  .  F5*(C(L1,J)-C(L2,J))-F6*FLUX1+DELH*F9-(C(L2,J)-
  .  P(L2,J))/DELT

     F(L3)=F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO)-(C(L3,J)-
  .  P(L3,J))/DELT


     F(L4)=(C(L5,J+1)-C(L5,J))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
  . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
  . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
  . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)

     F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J+1)-
  . 2*C(L5,J)+C(L5,J-1))+C(L4,J)*(C(L5,J+1)-C(L5,J))/DELZ1+
  . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L5,J)/C(L1,J)*((C(L1,J)-
  . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J+1))+
  . C(L4,J)*(C(L1,J+1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*
  . RATE_C1(J)
 C-------------------------------------------------------------
     END IF
     RETURN
     END

     SUBROUTINE DIFFEQ
```

```
      IMPLICIT REAL*8(A-H,O-Z)
C**************************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C**************************************************************************
C
C    CALLED BY:  MAIN CALLING PROGRAM
C
C**************************************************************************
C
C    SUBROUTINES CALLED:
C         WRTOUT (FOR DATA OUTPUT)
C         BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C         FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C              FOR A VARIABLE)
C
C**************************************************************************
C
C    LIST OF IMPORTANT VARIABLES:
C
C         A      A coefficient described in Newman, Appendix C
C         AA      first, AA is F(WORKC*CU).  Later AA is the
C              value of the derivative df/dc used in Newton's
C              method
C         B      B coefficient described in Newman, Appendix C
C         C      variables to be solved for
C         CD      multiplication factor used in obtaining
C              numerical derivatives
C         COLD    value of C from previous iteration
C         CU      2.0 - CD
C         D      D coefficient described in Newman, Appendix C
C         ERR     convergence criterion
C         F      value of function f(C), calculated in FUNCT
C         G      residual of f(C) calculated with updated C value
C         I     index used for equation number
C         ITCNT   index for iteration number
C         ITPRT   flag used for determining whether intermediate
C              calculations are output; for ITPRT=0,only
C              converged results are output; for
C              ITPRT=1,results of each iteration are printed.
C         J      index for node number
C         K       index for equation number
C         M       index used in workin: through nodes used to
C              calculate numerical derivatives
C         MM      used to determine starting node (in relation to
C              J) for estimation of numerical derivatives
C         N       number of equations (no. of variables)
C         NJ      number of node points
C         SAVEC   saved value of C
C         SUM     intermediate value used in calculating G
C         TINIER  criterion used to avoid working with small numbers
C         TINY    criterion used to avoid working with small numbers
C         TNIEST  criterion used to avoid working with small numbers
C         WORKC   saved value of C; modified when C less than Tinier
C         X       X value described in Newman, Appendix C
C         Y       Y value described in Newman, Appendix C
C**************************************************************************
C
C  DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C  IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C  IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C  DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C  NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C  WHATEVER YOO¿ppp
```

```
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201)
      DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-20,1.0D-4/
      DATA CU,CD/1.0001,.9999/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
      ITPRT=0
      IF (ITPRT.GT.0) CALL WRTOUT2
C
C LOOP BEGUN FOR ITERATIONS
C
      DO 75 ITCNT=1,10
C
C COLD ARRAY SET UP
C
        DO 10 K=1,N
          DO 10 J=1,NJ
            COLD(K,J)=C(K,J)
10         CONTINUE
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
        DO 60 J=1,NJ
          CALL FUNCT(J)
          DO 15 I=1,N
            SUM(I)=0.0
            G(I)=-F(I)
15          CONTINUE
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
          IF (J.EQ.1) THEN
            MM=0
          ELSE IF (J.LT.NJ) THEN
            MM=-1
          ELSE
            MM=-2
          END IF
          DO 50 M=MM,MM+2
C
C ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
            DO 50 K=1,N
              SAVEC=C(K,J+M)
              WORKC=SAVEC
C
C DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
              IF (ABS(WORKC).LT.TINY) THEN
                IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
                C(K,J+M)=1.2*WORKC
                CALL FUNCT(J)
                DO 20 I=1,N
                  AA(I)=-F(I)
20               CONTINUE
                C(K,J+M)=1.1*WORKC
                CALL FUNCT(J)
                DO 25 I=1,N
```

```
              AA(I)=AA(I)+4.0*F(I)
25            CONTINUE
            C(K,J+M)=WORKC
            CALL FUNCT(J)
            DO 30 I=1,N
              AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
30            CONTINUE
C
C  DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
          ELSE
            C(K,J+M)=WORKC*CU
            CALL FUNCT(J)
            DO 35 I=1,N
              AA(I)=F(I)
35            CONTINUE
            C(K,J+M)=WORKC*CD
            CALL FUNCT(J)
            DO 40 I=1,N
              AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
40            CONTINUE
          ENDIF
          C(K,J+M)=SAVEC
C
C  VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
          DO 45 I=1,N
            SUM(I)=SUM(I)+AA(I)*C(K,J+M)
            IF (M.EQ.-2) Y(I,K)=AA(I)
            IF (M.EQ.-1) A(I,K)=AA(I)
            IF (M.EQ.0) B(I,K)=AA(I)
            IF (M.EQ.1) D(I,K)=AA(I)
            IF (M.EQ.2) X(I,K)=AA(I)
45          CONTINUE
50        CONTINUE
        DO 55 I=1,N
          G(I)=G(I)+SUM(I)
55        CONTINUE
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
        CALL BAND2(J)
60      CONTINUE
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C  ITERATION IS BEGUN
C
      DO 65 K=1,4,2
        DO 65 J=1,NJ
         IF(DABS(C(K,J)).GT.TNIEST) THEN
           IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
         ENDIF
65      CONTINUE
     GO TO 80
70   IF(ITPRT .GT. 0) CALL WRTOUT2
75   CONTINUE
80   CONTINUE
     RETURN
     END

      SUBROUTINE FUNCT1(J)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 D_L(2),M_AVE
```

```
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
     . Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201)
      COMMON/PR_OLD_Z/ VISC_P(201),RO_P(201),CP_P(201),CON_FPP(201),
     . CON_LPP(201),D_LPP(4,201),H_FP(201),H_TTP(201),H_IIP(201),
     . H_OWPP(201),H_FWPP(201),Q(4,201),RATE_C1(201),AMOL_W(201),
     . D_LAV(201)
      COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4)
      COMMON/PROP_D_Z/DELZ1,DELT1
      DATA L1,L4,L5,R1,CONV/3,1,2,555.0,1/
      DATA FAC,GC/2.78450526316,416975040.0/
C
C
C MASS TRANSFER FROM THE BULK OF GAS SRTEAM TO THE SURFACE OF ABSORBENT
C
C ESTABLISHED COEFFICIENT MATRIX
C
C THE FIRST ROW
C
      RATE_C=RATE_C1(J)
      VISC_F=VISC_P(J)
      C_K=2.7845/(150*(1-EPSEX)**2*VISC_F/D_P/D_P/
     .EPSEX/EPSEX/GC)
      D_L_AVE=D_LAV(J)

      IF(J.EQ.1)THEN
       F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J+1)-
     . 2*C(L5,J)+C(L5,J+1))-C_K*(C(L5,J+1)-C(L5,J))**2/DELZ1/DELZ1-
     . C_K*C(L5,J)*(C(L5,J+1)-2*C(L5,J)+C(L5,J+1))/DELZ1/DELZ1-
     . C(L5,J)/C(L1,J)*((C(L1,J)-P(L1,J))/DELT1-D_L_AVE/DELZ1**2*
     . (C(L1,J+1)-2*C(L1,J)+C(L1,J+1))+C(L4,J)*(C(L1,J+1)-C(L1,J+1))/
     . (2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

       F(L4)=(C(L5,J+1)-C(L5,J))/DELZ1+(150*(1-EPSEX)**2*VISC_F*
     . C(L4,J)/D_P/D_P/EPSEX/EPSEX/GC/2.7845)

C
C-----------------------------------------------------
C
      ELSE IF(J.EQ.NJ)THEN

       F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J-1)-
     . 2*C(L5,J)+BC_L5*CONV)-C_K*(BC_L5*CONV-C(L5,J))**2/DELZ1/DELZ1-
     . C_K*C(L5,J)*(C(L5,J-1)-2*C(L5,J)+BC_L5*CONV)/DELZ1/DELZ1-
     . C(L5,J)/C(L1,J)*((C(L1,J)-P(L1,J))/DELT1-D_L_AVE/DELZ1**2*
     . (C(L1,J-1)-2*C(L1,J)+C(L1,J-1))+C(L4,J)*(C(L1,J-1)-
     . C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

       F(L4)=(BC_L5*CONV-C(L5,J))/DELZ1+(150*(1-EPSEX)**2*VISC_F*
     . C(L4,J)/D_P/D_P/EPSEX/EPSEX/GC/2.7845)
C
C------------------------------------------------------
C
      ELSE

       F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J+1)-
     . 2*C(L5,J)+C(L5,J-1))-C_K*(C(L5,J+1)-C(L5,J))**2/DELZ1/DELZ1-
     . C_K*C(L5,J)*(C(L5,J+1)-2*C(L5,J)+C(L5,J-1))/DELZ1/DELZ1-
     . C(L5,J)/C(L1,J)*((C(L1,J)-P(L1,J))/DELT1-D_L_AVE/DELZ1**2*
     . (C(L1,J+1)-2*C(L1,J)+C(L1,J-1))+C(L4,J)*(C(L1,J+1)-
     . C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C
```

```
      F(L4)=(C(L5,J+1)-C(L5,J))/DELZ1+(150*(1-EPSEX)**2*VISC_F*
     . C(L4,J)/D_P/D_P/EPSEX/EPSEX/GC/2.7845)
      END IF

C-------------------------------------------------------
 210  RETURN
      END

      SUBROUTINE DIFFEQ1(C1,P1,NJ1,N1)
      IMPLICIT REAL*8(A-H,O-Z)
C**********************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C**********************************************************************
C
C     CALLED BY:  MAIN CALLING PROGRAM
C
C**********************************************************************
C
C     SUBROUTINES CALLED:
C          WRTOUT (FOR DATA OUTPUT)
C          BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C          FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C               FOR A VARIABLE)
C
C**********************************************************************
C
C     LIST OF IMPORTANT VARIABLES:
C
C          A     A coefficient described in Newman, Appendix C
C          AA     first, AA is F(WORKC*CU). Later AA is the
C               value of the derivative df/dc used in Newton's
C               method
C          B     B coefficient described in Newman, Appendix C
C          C     variables to be solved for
C          CD     multiplication factor used in obtaining
C               numerical derivatives
C          COLD   value of C from previous iteration
C          CU     2.0 - CD
C          D     D coefficient described in Newman, Appendix C
C          ERR    convergence criterion
C          F     value of function f(C), calculated in FUNCT
C          G     residual of f(C) calculated with updated C value
C          I     index used for equation number
C          ITCNT   index for iteration number
C          ITPRT   flag used for determining whether intermediate
C               calculations are output; for ITPRT=0,only
C               converged results are output; for
C               ITPRT=1,results of each iteration are printed.
C          J     index for node number
C          K     index for equation number
C          M     index used in working through nodes used to
C               calculate numerical derivatives
C          MM     used to determine starting node (in relation to
C               J) for estimation of numerical derivatives
C          N     number of equations (no. of variables)
C          NJ     number of node points
C          SAVEC   saved value of C
C          SUM    intermediate value used in calculating G
C          TINIER  criterion used to avoid working with small numbers
C          TINY   criterion used to avoid working with small numbers
C          TNIEST  criterion used to avoid working with small numbers
C          WORKC   saved value of C; modified when C less than Tinier
C          X     X value described in Newman, Appendix C
```

```
C         Y      Y value described in Newman, Appendix C
C***************************************************************************
C
C DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C WHATEVER YOO¿ppp
   REAL*8 C1(14,201),P1(14,201)
   COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
   COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
  .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201)
   DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-20,1.0D-8/
   DATA CU,CD/1.0001,.9999/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
   ITPRT=0
   ITCNT=0
   NJ=NJ1
   N=N1
   IF (ITPRT.GT.0) CALL WRTOUT2
   DO 5 I1=1,N1+1
    DO 5 I2=1,NJ1
     C(I1,I2)=C1(I1,I2)
     P(I1,I2)=P1(I1,I2)
5   CONTINUE
C
C LOOP BEGUN FOR ITERATIONS
C
   DO 75 ITCNT=1,10
C
C COLD ARRAY SET UP
C
     DO 10 K=1,N
      DO 10 J=1,NJ
        COLD(K,J)=C(K,J)
10    CONTINUE
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
     DO 60 J=1,NJ
      CALL FUNCT1(J)
      DO 15 I=1,N
        SUM(I)=0.0
        G(I)=-F(I)
15      CONTINUE
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
      IF (J.EQ.1) THEN
        MM=0
      ELSE IF (J.LT.NJ) THEN
        MM=-1
      ELSE
        MM=-2
      END IF
      DO 50 M=MM,MM+2
C
```

```
C  ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
          DO 50 K=1,N
            SAVEC=C(K,J+M)
            WORKC=SAVEC
C
C  DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
            IF (ABS(WORKC).LT.TINY) THEN
              IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
              C(K,J+M)=1.2*WORKC
              CALL FUNCT1(J)
              DO 20 I=1,N
                AA(I)=-F(I)
20             CONTINUE
              C(K,J+M)=1.1*WORKC
              CALL FUNCT1(J)
              DO 25 I=1,N
                AA(I)=AA(I)+4.0*F(I)
25             CONTINUE
              C(K,J+M)=WORKC
              CALL FUNCT1(J)
              DO 30 I=1,N
                AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
30             CONTINUE
C
C  DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
            ELSE
              C(K,J+M)=WORKC*CU
              CALL FUNCT1(J)
              DO 35 I=1,N
                AA(I)=F(I)
35             CONTINUE
              C(K,J+M)=WORKC*CD
              CALL FUNCT1(J)
              DO 40 I=1,N
                AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
40             CONTINUE
             ENDIF
             C(K,J+M)=SAVEC
C
C  VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
            DO 45 I=1,N
              SUM(I)=SUM(I)+AA(I)*C(K,J+M)
              IF (M.EQ.-2) Y(I,K)=AA(I)
              IF (M.EQ.-1) A(I,K)=AA(I)
              IF (M.EQ.0) B(I,K)=AA(I)
              IF (M.EQ.1) D(I,K)=AA(I)
              IF (M.EQ.2) X(I,K)=AA(I)
45           CONTINUE
50        CONTINUE
         DO 55 I=1,N
           G(I)=G(I)+SUM(I)
55        CONTINUE
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
         CALL BAND2(J)
60      CONTINUE
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
```

```
C  ITERATION IS BEGUN
C
       DO 65 K=1,N
         DO 65 J=1,NJ
           IF(DABS(C(K,J)).GT.TNIEST) THEN
             IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
           ENDIF
65       CONTINUE
     GO TO 80
70   IF(ITPRT .GT. 0) CALL WRTOUT2
75   CONTINUE
80   CONTINUE
     DO 95 I1=1,N1
      DO 95 I2=1,NJ1
       C1(I1,I2)=C(I1,I2)
95   CONTINUE
     RETURN
     END

     SUBROUTINE WRTOUT2
C*************************************************************************
     IMPLICIT REAL*8(A-H,O-Z)
     COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),
    .Y(14,14),G(14),N,NJ,ITPRT,ITCNT,F(14)
     IF (ITCNT.NE.0) WRITE (*,99)ITCNT
     WRITE (*,100)
     DO 1 K=1,NJ,2
       WRITE(*,101)K,(C(I,K),I=1,N)
1    CONTINUE
99   FORMAT(' ITCNT=',I2)
100  FORMAT(' J       C1           C2           C3          '
    &,' C4          C5           C6'/)
101  FORMAT(1X,I3,6(1PE16.8))
     RETURN
     END
C  BLOCK TRIDIAGONAL MATRIX SUBROUTINE
     SUBROUTINE BAND2(J)
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION E(14,14,201)
     COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29),X(14,14),Y(14,14),
    .G(14),N,NJ,ITPRT,ITCNT,F(14)
 101 FORMAT('0DETERM=0 AT J=',I4)
     IF (J.EQ.1) THEN
       NP1=N+1
       DO 2 I=1,N
         D(I,2*N+1)=G(I)
         DO 2 L=1,N
          D(I,L+N)=X(I,L)
  2    CONTINUE
       CALL MATINV2(N,2*N+1,DETERM)
       IF(DETERM.EQ.0) WRITE (2,101) J
       DO 5 K=1,N
         E(K,NP1,1)=D(K,2*N+1)
         DO 5 L=1,N
           E(K,L,1)=-D(K,L)
           X(K,L)=-D(K,L+N)
  5    CONTINUE
       RETURN
     ELSE IF(J.EQ.2)THEN
       DO 7 I=1,N
         DO 7 K=1,N
           DO 7 L=1,N
             D(I,K)=D(I,K)+A(I,L)*X(L,K)
  7    CONTINUE
```

```
      ELSE IF(J.EQ.NJ) THEN
        DO 10 I=1,N
          DO 10 L=1,N
            G(I)=G(I)-Y(I,L)*E(L,NP1,J-2)
            DO 10 M=1,N
              A(I,L)=A(I,L)+Y(I,M)*E(M,L,J-2)
 10     CONTINUE
      ENDIF
      DO 12 I=1,N
        D(I,NP1)=-G(I)
        DO 12 L=1,N
          D(I,NP1)=D(I,NP1)+A(I,L)*E(L,NP1,J-1)
          DO 12 K=1,N
            B(I,K)=B(I,K)+A(I,L)*E(L,K,J-1)
 12   CONTINUE
      CALL MATINV2(N,NP1,DETERM)
      IF(DETERM.EQ.0) WRITE(2,101) J
      DO 15 K=1,N
        DO 15 M=1,NP1
          E(K,M,J)=-D(K,M)
 15   CONTINUE
      IF(J.EQ.NJ)THEN
        DO 17 K=1,N
          C(K,J)=E(K,NP1,J)
 17     CONTINUE
        DO 18 M=NJ-1,1,-1
          DO 18 K=1,N
            C(K,M)=E(K,NP1,M)
            DO 18 L=1,N
              C(K,M)=C(K,M)+E(K,L,M)*C(L,M+1)
 18     CONTINUE
        DO 19 L=1,N
          DO 19 K=1,N
            C(K,1)=C(K,1)+X(K,L)*C(L,3)
 19     CONTINUE
      ENDIF
      RETURN
      END
C
C     MATRIX INVERSION SUBROUTINE
      SUBROUTINE MATINV2(N,M,DETERM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ID(14)
      COMMON/BND/A(14,14),B(14,14),C(14,201),D(14,29)
      DETERM=1.0
      DO 1 I=1,N
 1    ID(I)=0
      DO 18 NN=1,N
        BMAX=0.0
        DO 6 I=1,N
          IF(ID(I).EQ.0)THEN
            DO 5 J=1,N
              IF(ID(J).EQ.0)THEN
                IF(DABS(B(I,J)).GT.BMAX) THEN
                  BMAX=DABS(B(I,J))
                  IROW=I
                  JCOL=J
                ENDIF
              ENDIF
 5          CONTINUE
          ENDIF
 6      CONTINUE
        IF(BMAX.EQ.0.0)THEN
          DETERM=0.0
```

```
        RETURN
      ENDIF
      ID(JCOL)=1
      IF(JCOL.NE.IROW) THEN
 9        DO 10 J=1,N
            SAVE=B(IROW,J)
            B(IROW,J)=B(JCOL,J)
            B(JCOL,J)=SAVE
 10        CONTINUE
         DO 11 K=1,M
            SAVE=D(IROW,K)
            D(IROW,K)=D(JCOL,K)
            D(JCOL,K)=SAVE
 11        CONTINUE
      ENDIF
      FF = 1.0/B(JCOL,JCOL)
      DO 13 J=1,N
 13     B(JCOL,J)=B(JCOL,J)*FF
      DO 14 K=1,M
 14     D(JCOL,K)=D(JCOL,K)*FF
      DO 18 I=1,N
        IF(I.NE.JCOL)THEN
 15        FF = B(I,JCOL)
          DO 16 J=1,N
 16          B(I,J) = B(I,J) -FF*B(JCOL,J)
          DO 17 K=1,M
 17          D(I,K) = D(I,K) -FF*D(JCOL,K)
        ENDIF
 18  CONTINUE
     RETURN
     END


     FUNCTION RE (G,TEMP,MW)
C TNIS FUNCTION CALCULATES THE PARICLE REYNOLDS NUMBER ASSUMINC THAT
C THE MOLECULAR WEIGHT OF TNE GAS IS EQUAL TO THAT OF CARRIER GAS.
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 LEN,MW
     COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
     COMMON/GAS/INERT,NCOMP
     DATA IFLAG/0/
     IF (IFLAG.EQ.0)THEN
      CONST=2.*RA
      IFLAG=1
     ENDIF
     RE=CONST*MW*ABS(G)/VIS(TEMP)
     RETURN
     END
      SUBROUTINE IST_Z(METHOD,T,PTOT,Y11,Y12,Y13,Y14,Q1)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/NCY/NC1
      REAL*8 Q1(4)
      INTEGER I,NLIM
      EXTERNAL FCN,FDER
      DATA XTOL,FTOL,I,NLIM/1.0E-12,1.0E-10,0,50/
      DATA PO/1.,1.,1.,1./
      DATA R/555/
      T_G=T
      NC=NC1
      Y1(1)=Y11
      Y1(2)=Y12
      Y1(3)=Y13
      Y1(4)=Y14
```

```
DO I=1,NC
 PP(I)=Y1(I)*PTOT
END DO
Z=1.0E-32
IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
 DO I=1,NC
  Q1(I)=0.
 END DO
 RETURN
END IF
V(1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
V(2)=1./69.035
V(3)=-1.637879912E-5*T+.00961297026
V(4)=V(3)
B(1)=6864.9000131*EXP(-.019625791466*T)
IF(T.LT.610.) THEN
 B(2)=4.5597278759E-7*EXP(9628.9655743/T)
ELSE
 B(2)=5.8089066684E-7*EXP(9115.734593/T)
END IF
IF(T.LE.532)THEN
 B(3)=3.2694515539E-7*T+4.59988799E-4
ELSE
 B(3)=7.90864008E-5*T-4.14400420E-2
END IF
B(4)=B(3)
TERM1=0.
DO I=1,NC
 IF(PP(I).GT.0)THEN
  TERM1=B(I)*PP(I)**PO(I)+TERM1
 END IF
END DO
SUM=0.
DO I=1,NC
 IF(PP(I).LE.0.) THEN
  Q1(I)=0.
 ELSE
  Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
  SUM=SUM+Q1(I)
 END IF
END DO
IF(METHOD.EQ.1)THEN
 RETURN
END IF
DO I=1,NC
 XI(I)=Q1(I)/SUM
 IF(XI(I).GT.0) THEN
  PI(I)=PP(I)/XI(I)
 ELSE
  PI(I)=0.
 END IF
END DO
DO I=1,NC
 IF(PP(I).LE.0.) THEN
  X1(I)=0
 ELSE
  X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
 END IF
END DO
X=X1(1)
DO I=2,NC
 IF(X1(I).LT.X)THEN
  X=X1(I)
 END IF
```

```
      END DO
      IF(X .LE. 1.0E-17) GO TO 80
      CALL NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        XI(I)=0.
       ELSE
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        PI(I)=(EXP(PART1)-1)/B(I)
        XI(I)=PP(I)/PI(I)
       END IF
      END DO
      TOT_Q=0
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
        TOT_Q=TOT_Q+XI(I)/Q1(I)
       END IF
      END DO
      DO I=1,NC
       Q1(I)=1./TOT_Q*XI(I)
      END DO
80     RETURN
      END

      REAL FUNCTION FCN(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
       END IF
      END DO
      FCN=SUM-1
      RETURN
      END

      REAL FUNCTION FDER(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART3=PO(I)*X/V(I)
        IF(PART3.LT.32)THEN
         PART1=-PP(I)/B(I)/V(I)*EXP(PART3)
         PART2=((EXP(PART3)-1)/B(I))**((1+PO(I))/PO(I))
         SUM=SUM+PART1/PART2
        END IF
       END IF
      END DO
      FDER=SUM
      RETURN
      END
```

```
      SUBROUTINE NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER NLIM,I,J
      FX=FCN(X)
      DO J=1,NLIM
       DELX=FX/FDER(X)
       X=X-DELX
       FX=FCN(X)
       IF (ABS(DELX).LE.XTOL)THEN
        RETURN
       END IF
       IF(ABS(FX).LE.FTOL) THEN
        RETURN
       END IF
      END DO
      I=-1
      PRINT 200, NLIM,X,FX
 200  FORMAT(/'TOLERANCE NOT MET ',I4,' ITERATIONS X= ',
     2 E12.5,' F(X)=',E12.5)
      RETURN
      END

      FUNCTION CPGAS(TEM,Y)
C THIS FUNCTION CALCULATES THE HEAT CAPACITIES OF CMRIER GAS IN BTU/#MOLE/R
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 A(4),B(4),C(4),D(4),Y(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
C CO2
      DATA A/36.11,33.46,29.,29.1/
C H2O
      DATA B/4.233E-2,.6880E-2,.2199E-2,1.158E-2/
C N2
      DATA C/-2.887E-5,.7604E-5,.5723E-5,-.6076E-5/
C O2
      DATA D/7.464E-9,-3.593E-9,-2.871E-9,1.311E-9/
      NC=NC1
      T=TEM/1.8-273.15
      IF (INERT.EQ.1) THEN
       CPGAS=0.
       DO I=1,NC
        CP=A(I)+B(I)*T+C(I)*T*T+D(I)*T*T*T
        CPGAS=CP*Y(I)/4.18669+CPGAS
       END DO
      ELSE
       CPGAS=4.97
      ENDIF
      RETURN
      END

      FUNCTION CPS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES TNE HEAT CAPACITY
C BTU/LB/R.
      CPS=0.2
      RETURN
      END

      REAL FUNCTION KGAS (TEMP)
C THIS FUNCTION CALCULATES THE THERMAL CONDUCTIVITY OF CARRIER GAS ASSUMING
C A LINEAR FUNCTION OF TEMPERATURE AND BASED ON VALUES AT 492 AND 672 R.
C BTU/MIN/FT/R
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
       KGAS=3.88E-7*TEMP+0.4052E-4
      ELSE
       KGAS=1.667E-6*TEMP+6.1E-4
      ENDIF
      RETURN
      END

      FUNCTION VIS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE VISCOSITY OF CARRIER GAS AS A LINEAR FUNCTION
C OF TEMPERATURE (LB/MIN/FT).
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
C       VIS=1.0E-6*TEMP+1.65E-4
       VIS=-.0102007812+5.384663937E-5*TEMP-8.7973727E-8*TEMP*TEMP+
     2     4.811387495E-11*TEMP*TEMP*TEMP
      ELSE
       VIS=0.9444E-6*TEMP+2.863E-4
      ENDIF
      RETURN
      END

      FUNCTION RHOG(TR,P)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
C THIS FUNCTION CALCULATES THE MOLAR DENSITY OF AN IDEAL GAS IN
C LBMOLES/CV FS.
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA R/555./
      RHOG=P/R/TR
      RETURN
      END

      FUNCTION EFFD(ICOMP,G,TEMP,RHO,PT,YO)
C THIS FUNCTION CALCVLATES TNE EFFECTIVE AXIAL DIFFUSIVITIES IN A PACXED BED
C USING THE EDWARDS AND RICHARDSON CORROLATION. (FT*FT/MIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA1,LAMBDA2,LEN,YO(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA IFLAG/0/,LAMBDA1/0.73/
      IF(IFLAG.EQ.0)THEN
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF
      DIF=DIFF(ICOMP,TEMP,PT,YO)
      EFFD=LAMBDA1*DIF+CONST*ABS(G)*LAMBDA2(G,DIF,RHO)/RHO
      RETURN
      END

      FUNCTION EFFK(G,T,CP,RHO,P,YO)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE EFFECTIVE AXIAL THERMAL CONDUCTIVITY OF A
C PACKED BED IN BTU/FT/MIN/F.
      COMMON/NCY/NC1
      REAL*8 YO(4)
      NC=NC1
      EFFK=0.
      KK=0
      DO I1=1,NC
       SUM=CP*RHO*EFFD(I1,G,T,RHO,P,YO)
       IF(SUM.GT.0)KK=KK+1
       EFFK=SUM+EFFK
```

```
      END DO
       EFFK=EFFK/KK
      RETURN
      END

      REAL FUNCTION LAMBDA2(G,DIF,RHO)
C THIS FUNCTION CALCULATES THE DIMENSTONLESS PARAMETER OF THE EDWARDS AND
C RICHARDS CORROLATION MODIFIED FOR SMALL DIAMETER PARTICLES.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      DATA IFLAG/0/
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      IF(IFLAG.EQ.0)THEN
       IF(RA.GT.0.00492)THEN
         PE=2.
       ELSE
         PE=406.4*RA
       ENDIF
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF

      LAMBDA2=1./PE/(1.+9.5*DIF/CONST/ABS(G)*RHO)
      RETURN
      END


      FUNCTION DIFF(ICOMP,TEM,P1,YO)
C  THIS FUNCTION CALCULATES THE MOLECULAR  DIFFUSION COEFFICIENTS BY CORRECTING
C  FOR PRESSURE AND TEMPERATURE (FT*FT/MIN).
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,YO(4),V(4),MW(4),DIF(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA V/26.9,12.7,17.9,16.6/
      DATA MW/44.01,18.016,28.013,31.999/
      NC=NC1
      K=ICOMP
      P=P1/760.
      T=TEM/1.8
      DO I=1,NC
       PART1=((MW(K)+MW(I))/MW(K)/MW(I))**.5
       PART2=(V(K)**(1./3)+V(I)**(1./3))**2
       DIF(I)=PART1/PART2*T**1.75/P*1.0E-3
      END DO
      SUM=0.
      DO I=1,NC
       IF(I.NE.K)THEN
         SUM=SUM+YO(I)/DIF(I)
       END IF
      END DO
      DIFF=(1-YO(K))/SUM*3.8745/60.
      RETURN
      END


      FUNCTION HFILM(MW,TEMP,REY,CP)
C THIS FUNCTION CALCULATES THE FLUID FILM HEAT TRANSFER COEFFICIENT
C IN BTU/SQ FT/MIN/R USING THE CORRELATION OF PETROY AND THODOS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 ID,KGAS,KG,LEN,MW
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/GAS/INERT,NCOMP
```

```
      KG=KGAS(TEMP)
      PR=CP/MW*VIS(TEMP)/KG
      HFILM=0.357/EPSEX*REY**0.64*PR**0.33*KG*0.5/RA
      RETURN
      END

C     FUNCTION DHCAR(TEMP,P)
C THIS SUBROUTINE CALCULATES HEAT OF ADSORPTION OF CARRIER GAS ONTO THE
C ACTIVATED CARBON TIMES THE TEMPERATURE DERIVATIVE OF THE SOLID PHASE
C CARRIER GAS CONCENTRATION AT THE BED PRESSURE (BTU/# SOLID/R).
C     IMPLICIT REAL*8(A-H,O-Z)
C     REAL*8 LEN
C     COMMON/GAS/INERT,NCOMP
C     COMMON/PRIME/GN2,RA,RAV,EPSEX,ALF,RHOS,ALPHA1
C     DATA R/1.9872/,A/1.28E-9/,B/3151./
C
C     IF(INERT.EQ.1) THEN
C       DH=R*B
C       DHN2=-DH*A*B*EXP(B/TEMP)*P/TEMP/TEMP
C     ELSE
C       DHN2=0.
C     ENDIF
C     RETURN
C     END

      FUNCTION DHADS(ICOMP,T)
C THIS FUNCTION CALCULATES THE ISOTERIC HEAT OF ADSORPTION. (BTU/#MOLE)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      IF(ICOMP.EQ.2) THEN
       DHADS=20400.0
      ELSE
       DHADS=20000.0
      END IF
      RETURN
      END
```

APPENDIX G

FLOW3MOL FORTRAN CODE

```
c
c A BED WITH THREE DIFFERENT MATERIALS. IN THIS CASE, 13X,SILICA GEL, AND 13X
c
C this is the main routin. the initial values of matix C and parameters are
C called by "INITIAL_Z" subroutine. the main routine then calls the "FUNCT2_Z"
C subroutine to solve the discretized partial diffential equations (pde's).
C after convergance, the routine calls the "DIFFEQ1" subroutine, which is
C the maine routine for solving the momentum and the pressure equations,
C if it is desire to solve these two equations as the time progresses.C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PL(11,201),QP(14,201),PL_TIME,T_TIME
      COMMON/BND_Z/P(14,201),C(14,201),U_F1,Z
      COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,N,NJ,II,LL
      COMMON/INDIC/L1,L2,L3,L4,L5,L6
      COMMON/IND1/IND(4)
      OPEN (50,FILE='ASCII.DAT',STATUS='NEW',RECL=32766)
      OPEN (66,FILE='CON_CO2.DAT',STATUS='NEW')
      OPEN (88,FILE='CON_H2O.DAT',STATUS='NEW')
      OPEN (99,FILE='TEM.DAT',STATUS='NEW')
C WRITE HEADER DATA TO ACSII FILE
      WRITE(50,42)
C
C INITIALIZE THE C ARRAY WITH THE INITIAL GUESSES OF THE SOLUTION
C
        CALL INITIAL_Z
       ICOT=1
       T_TIME=0
c start marching through time
       DO 301 LL=1,9000
c call funct2_z subroutine to solve the PDE's by numerial method
         CALL FUNCT2_Z
c accumulate the time steps
         T_TIME=T_TIME+DELT*Z/U_F1*60.
          DO 74 M2=1,NJ
           QP(1,M2)=C(1,M2)*10.73D0*C(L1,M2)*760/14.696
           QP(2,M2)=C(2,M2)*10.73D0*C(L1,M2)*760/14.696
           QP(3,M2)=C(L1,M2)-460.
           QP(4,M2)=C(L2,M2)-460.
           QP(5,M2)=C(L3,M2)-460.
 74       CONTINUE
       WRITE(*,*)W0,LL,II,T_TIME
       WRITE(*,*)QP(1,6),QP(1,25),QP(1,50),QP(1,75),QP(1,101)
       WRITE(*,*)QP(2,10),QP(2,25),QP(2,50),QP(2,75),QP(2,101)
       WRITE(*,*)QP(3,10),QP(3,25),QP(3,50),QP(3,75),QP(3,101)
       WRITE(*,*)QP(4,10),QP(4,25),QP(4,50),QP(4,75),QP(4,101)
       WRITE(*,*)W0,LL,II,T_TIME
C        WRITE(*,*)II,T_TIME
C        WRITE(*,*)QP(1,3),QP(1,37),QP(1,75),QP(1,112),QP(1,150)
C        WRITE(*,*)QP(2,3),QP(2,37),QP(2,75),QP(2,112),QP(2,150)
C        WRITE(*,*)QP(3,3),QP(3,37),QP(3,75),QP(3,112),QP(3,150)
      DO 520 J=1,151
        PL(1,J)=QP(1,J)
        PL(2,J)=QP(2,J)
        PL(3,J)=QP(3,J)
        PL(4,J)=QP(4,J)
 520    PL(5,J)=QP(5,J)
      PL_TIME=T_TIME
      ISAMP=25
      IF((LL/ISAMP)*ISAMP.EQ.LL) THEN
```

```
   WRITE(50,43)PL_TIME,PL(1,6),PL(1,25),PL(1,50),PL(1,75),PL(1,101)
  &              ,PL(2,6),PL(2,25),PL(2,50),PL(2,75),PL(2,101)
  &              ,PL(3,6),PL(3,25),PL(3,50),PL(3,75),PL(3,101)
  &              ,PL(4,6),PL(4,25),PL(4,50),PL(4,75),PL(4,101)
  &              ,PL(5,6),PL(5,25),PL(5,50),PL(5,75),PL(5,101)
   ENDIF
      DO 78 I1=1,N
       DO 78 I2=1,NJ
        P(I1,I2)=C(I1,I2)
78     CONTINUE
200   CONTINUE
301    CONTINUE
42    FORMAT(1X,'TIME,ppCO2@1,ppCO2@2,ppCO2@3,ppCO2@4,ppCO2@5',
  &             ',ppH2O@1,ppH2O@2,ppH2O@3,ppH2O@4,ppH2O@5',
  &             ',T gas@1,T gas@2,T gas@3,T gas@4,T gas@5',
  &             ',T bed@1,T bed@2,T bed@3,T bed@4,T bed@5',
  &             ',Twall@1,Twall@2,Twall@3,Twall@4,Twall@5')
43    FORMAT(1X,26(E15.5,','))
44    FORMAT(/5X,5(E15.5,3X))
45    FORMAT(/5X,5(E15.5,3X))
46    FORMAT(/5X,5(E15.3,3X))
     STOP
     END
C--------------------------------------------------------
C this subroutine is being called by maine routine once to get the intial
c values and the neccessary parameters
C
    SUBROUTINE INITIAL_Z
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 K_F(4),M_AVE,M_W(4),C_FO(4),YO(4)
     COMMON/BND_Z/P(14,201),C(14,201),U_F1,Z
     COMMON/PROP_B_Z/D_L,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,
   . CP_WA,CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,
   . X_WI_K,D_LMI,D_LMA
     COMMON/PROP_S_Z_1/CP_S1,RO_S1,AINT1,R_P1,D_P1,CON_S1,
   . G_F1,HEAT1(4),M_W
     COMMON/PROP_S_Z_2/CP_S2,RO_S2,AINT2,R_P2,D_P2,CON_S2,
   . G_F2,HEAT2(4)
     COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,N,NJ,IL,LL
     COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
     COMMON/INDIC/L1,L2,L3,L4,L5,L6
     COMMON/NCY/NC1
     COMMON/GAS/INERT,NCOMP
     COMMON/IND1/IND(4)
     COMMON/PRIME_1/GN1,RA1,RAV1,EPSEX1,EPSIN1,ALF1,RHOS1,ALPHA11
     COMMON/PRIME_2/GN2,RA2,RAV2,EPSEX2,EPSIN2,ALF2,RHOS2,ALPHA12
     DATA INERT,NCOMP/1,3/
C
C INITIALIZATION
C
     DATA Z,D_L,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,CP_WA,
   . CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,X_WI_K
   . /1.666666667,.155833D0,.1666667D0,1.90690D-2,32.875D0,
   . .29D0,.19D0,.109848D0,.21D0,.21D0,489.0D0,3.5D0,16.0D0,
   . 5.41666667D-3,4.1666667D-2,2.083333D-2/
     DATA PT,RA1,RAV1,EPSEX1,EPSIN1,RHOS1,ALPHA11
   . /800.2,4.72441E-3,4.72441E-3,.373,.317,43.,635./
     DATA RA2,RAV2,EPSEX2,EPSIN2,RHOS2,ALPHA12
   . /5.2174E-3,5.2174E-3,.373,.317,43.,575./
     DATA AINT1,CP_S1,RO_S1,R_P1,D_P1,CON_S1,HEAT1
   . /635.D0,0.22D0,65.0,4.72441D-03,9.44882D-3,.1,-20340.D0,
   . -32000.D0,-8988.0D0,-8988./
     DATA AINT2,CP_S2,RO_S2,R_P2,D_P2,CON_S2,HEAT2
   . /575.D0,0.22D0,75.0,5.2174D-03,1.02548D-2,.1,-17500.D0,
```

```
     . -30000.D0,-8988.0D0,-8988./
       DATA G_F,P_TOT,TO
     . /63.53,15.47,538.0D0/
       DATA BC_C/.13187,.13187,2.01,13.3281/
       DATA BC_L1,BC_L2,BC_L3,BC_L5,BC_L6
     . /538.,538.,538.,15.47,15.47/
       DATA M_W/44.,18.,28.,32./
       DATA NC,NJ,DELZ1,DELT1/4,101,.008333,.002/
       DATA R,PI/10.73D0,3.141593D0/
       DATA IND/1,1,1,1/
c number of component
       NC1=NC
c number of equations
       N=NC*2+5
       ALF1=(1.0D0-EPSEX1)/EPSEX1
       ALF2=(1.0D0-EPSEX2)/EPSEX2
       PT=P_TOT*760./14.696
       SUM_Y=0
       G_F1=G_F
       G_F2=G_F
c calculate the mole fraction
       DO 10 I=1,NC
         Y_F=BC_C(I)/P_TOT
         SUM_Y=SUM_Y+Y_F
         BC_C(I)=P_TOT*Y_F/R/BC_L1
 10    CONTINUE
c calculate fluid density
       RO_FO=P_TOT/(BC_L1*R)
c calc molar superficial velocity
       GN=G_F*RO_FO/S_B/60.
       U_F1=G_F/(S_B*EPSEX1)
       BC_L4=U_F1
c grid size in dimensionless
       DELZ=DELZ1/Z
c time step in dimensionless
       DELT=U_F1*DELT1/Z
c  L1 is the gas temp, L2 is the solid temp, L3 is the wall temp, L4 is the
c velocity, L5 is the total pressure, and L6 is the total pressure witout
c thermal effect
c
       L1=2*NC+1
       L2=L1+1
       L3=L2+1
       L4=L3+1
       L5=L4+1
       L6=L5+1
c set the initial guess and the old value of C array; P
       DO 20 I1=1,N
        DO 20 I2=1,NJ
         P(I1,I2)=0.0D0
         C(I1,I2)=0.0D0
 20    CONTINUE
c set the the first grid to the boundry
       DO 22 I1=1,NC
        C(I1,1)=BC_C(I1)
 22    CONTINUE
       DO 24 I1=1,2
        DO 24 I2=2,NJ
         C(I1,I2)=0.
         C(I1+2,I2)=BC_C(I1+2)
 24    CONTINUE
c set the initial temperature and pressure
       DO 30 I=1,NJ
        P(L1,I)=BC_L1
```

```
      P(L2,I)=BC_L2
      P(L3,I)=BC_L3
      P(L4,I)=U_F1
      P(L5,I)=BC_L5
      P(L6,I)=BC_L6
      C(L1,I)=BC_L1
      C(L2,I)=BC_L2
      C(L3,I)=BC_L3
      C(L4,I)=U_F1
      C(L5,I)=BC_L5
      C(L6,I)=BC_L6
30    CONTINUE
     RETURN
     END
c---------------------------------------------------------------------------
c this subroutine is being called by maine routine to compute the variable in
c C matrix. in this routine first velocity profile is being determined. the
c conductivity, diffusivity,pososity....... are being calculated in this
c routine by calling the approperiate subroutine. the routine obtained the
c C matix in axial and radial directions. it itrates till it conveges to the
c allowable error
c
    SUBROUTINE FUNCT2_Z
     IMPLICIT REAL*8(A-H,O-Z)
     REAL*8 K_F(4),M_AVE,M_W(5),KGAS,K_FP(4,201),C1(14,201),
   . P1(14,201),REY_P(201),KEFF_SIL,KEFF(4),D_L(4),A(4),YO(4),SS(4)
     DIMENSION Q2(4),C2(14,201),PE_N_M(4),DWDX(4),DWDT(4)
     COMMON/PRIME_1/GN1,RA1,RAV1,EPSEX1,EPSIN1,ALF1,RHOS1,ALPHA11
     COMMON/PRIME_2/GN2,RA2,RAV2,EPSEX2,EPSIN2,ALF2,RHOS2,ALPHA12
     COMMON/PROP_S_Z_1/CP_S1,RO_S1,AINT1,R_P1,D_P1,CON_S1,
   . G_F1,HEAT1(4),M_W
     COMMON/PROP_S_Z_2/CP_S2,RO_S2,AINT2,R_P2,D_P2,CON_S2,
   . G_F2,HEAT2(4)
     COMMON/PR_OLD_Z/ RO_P(201),CP_P(201),CON_FPP(201),
   . CON_LPP(201),D_LPP(4,201),H_FP(201),H_TTP(201),H_IIP(201),
   . H_OWPP(201),H_FWPP(201),Q(4,201),VISC_P(201),RATE_C1(201),
   . WM_AVE1(201)
     COMMON/BND_Z/P(14,201),C(14,201),U_F1,Z
     COMMON/PROP_B_Z/D_I,D_E,S_B,CON_WA,CON_WI_Q,CON_WI_K,CP_WA,
   . CP_WI_Q,CP_WI_K,RO_WA,RO_WI_Q,RO_WI_K,X_WA,X_WI_Q,X_WI_K,
   . D_LMI,D_LMA
     COMMON/PROP_S_Z/CP_S,RO_S,AINT,R_P,D_P,CON_S,G_F,HEAT(4)
     COMMON/PROP_D_Z/DELZ1,DELT1,DELT,DELZ,NC,N,NI,II,LL
     COMMON/INDIC/L1,L2,L3,L4,L5,L6
     COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4),TO
     COMMON/DELZ_DIFF/D01,D11,D21,D02,D12,D22,FR
     COMMON/LAYERS/LAYER1,LAYER2
     COMMON/GAS/INERT,NCOMP
     COMMON/NCY/NC1
     COMMON/IND1/IND(4)
     COMMON/PRIME/GN,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
     DATA R,PI/10.73D0,3.141593D0/
      IIO=100
c set the number of grids for different material in the bed
    IF(LL.EQ.1)THEN
     IPORE=0
     LAYER1=6
     LAYER2=50
     EPSP=.52
     DELZ_OLD=DELZ
c set the under-relaxation coeff
     W0=.2
     W1=.2
     W2=.2
```

```
       W3=.2
      ENDIF
      IF(W0.LE.0.5)W0=W0+.002
      IF(W1.LE.0.5)W1=W1+.002
      IF(W2.LE.0.5)W2=W2+.002
      IF(W3.LE.0.5)W3=W3+.002
      ITEST=0
      ICOUNT=1
      ICONV=0
c calculate the surface area for the heat transfer
      ERROR=1.0E-3
      D_T=D_E+D_I
      X_W=D_E-D_I
      A_C=2*D_I/((D_I+D_E)*X_W)
      A_INS=2*D_E/((D_I+D_E)*X_W)
      RO_F=C(L5,1)/(C(L1,1)*R)
      G_F=G_F1
      GN=G_F*RO_F/S_B/60
c compute parameters for all the grids
      DO 199 J=1,NJ
c set parameters for middle material, silica gel
        IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
         RA=RA2
         EPSEX=EPSEX2
         EPSIN=EPSIN2
         R_P=R_P2
         D_P=D_P2
         DO 211 I1=1,NC
          HEAT(I1)=HEAT2(I1)
 211      CONTINUE
c set parameters for ends material, in this case 13X on bouth ends
        ELSE
         RA=RA1
         EPSEX=EPSEX1
         EPSIN=EPSIN1
         AINT=AINT1
         R_P=R_P1
         D_P=D_P1
         DO 213 I1=1,NC
          HEAT(I1)=HEAT1(I1)
 213      CONTINUE
        END IF
C         IF(LL.EQ.1) THEN
        RO_F=C(L5,J)/(C(L1,J)*R)
        YTOT=0.
c calculate the mole fraction
        DO I1=1,NC-1
         YO(I1)=C(I1,J)/RO_F
         YTOT=YO(I1)+YTOT
        END DO
        YO(NC)=1-YTOT
c ccalculate the average molecular weight
        M_AVE=0.
        DO I1=1,NC
         M_AVE=M_AVE+M_W(I1)*YO(I1)
        END DO
        WM_AVE1(J)=M_AVE
        TEMP=C(L1,J)
        GN=G_F*RO_F/S_B/60.
c calculate viscocity
        VISC_P(J)=VIS(TEMP)*60
c calculate reynolds number
        REY=RE(GN,TEMP)
        REY_P(J)=REY
```

```
c calculate heat capacity
      CP_P(J)=CPGAS(TEMP,YO)
c calculate heat transfer coefficient between the pellet and fluid
      H_FP(J)=HFILM(M_AVE,TEMP,REY,CP_P(J))*60.
c conver pressure
      PP=C(L5,J)*760./14.696
c calculate the thermal dispersion
      CON_LPP(J)=EFFK(GN,TEMP,CP_P(J),RO_F,PP,YO)*60.
c calculate mass diffusivity
      DO I1=1,NC
       D_LPP(I1,J)=EFFD(I1,GN,TEMP,RO_F,PP,YO)*60.
       IF(D_LPP(I1,J).LT.0) THEN
        D_LPP(I1,J)=D_LPP(I1,J-1)
       END IF
      END DO
C     END IF
 199  CONTINUE
c heat transfer coefficient between the fluid and the wall
     H_FW=4.0
c heat transfer coeff between the waal and outside
     H_OW=0.05
c start itration
     IF(LL.EQ.1)II_OLD=3000
     DO 220 II=1,II_OLD
c set the most recent computed C array to C2 array for check of conversion
 20    DO 30 I1=1,N
       DO 30 I2=1,NJ
        C2(I1,I2)=C(I1,I2)
 30    CONTINUE
 34   DO 190 J=1,NJ
c since there are differnt sizes of grids at the boundries of two adjsent
c materials, the equations must revised.
      IF(J.EQ.LAYER1)THEN
       ITESTJ=1
c fraction of actual grid size
       FR=3.750015
       FR=1.0
c PDE's coefficints
       D01=FR/(1+FR)
       D11=(1-FR)/FR
       D21=1./FR/(1+FR)
       D02=1./(1+FR)
       D12=1./FR
       D22=1./FR/(1+FR)
      ELSE IF(J.EQ.LAYER2)THEN
       ITESTJ=1
       FR=.2666666
       FR=1.0
       D01=FR/(1+FR)
       D11=(1-FR)/FR
       D21=1./FR/(1+FR)
       D02=1./(1+FR)
       D12=1./FR
       D22=1./FR/(1+FR)
      ELSE
       ITESTJ=0
      END IF
c fix the grid size
      IF(J.LE.LAYER1)THEN
       DELZ=.2666666*DELZ_OLD
       DELZ=DELZ_OLD
      ELSE IF(J.GT.LAYER2)THEN
       DELZ=.2666666*DELZ_OLD
       DELZ=DELZ_OLD
```

```
        ELSE
         DELZ=DELZ_OLD
        END IF
c set bed and pellet parameter for each material
        IF(J.GE.LAYER1 .AND. J.LT.LAYER2)THEN
         RA=RA2
         RAV=RAV2
         EPSEX=EPSEX2
         EPSIN=EPSIN2
         ALF=ALF2
         RHOS=RHOS2
         ALPHA1=ALPHA12
         CP_S=CP_S2
         RO_S=RO_S2
         AINT=AINT2
         R_P=R_P2
         D_P=D_P2
         CON_S=CON_S2
         DO 111 I1=1,NC
          HEAT(I1)=HEAT2(I1)
111       CONTINUE
        ELSE
         RA=RA1
         RAV=RAV1
         EPSEX=EPSEX1
         EPSIN=EPSIN1
         ALF=ALF1
         RHOS=RHOS1
         ALPHA1=ALPHA11
         CP_S=CP_S1
         RO_S=RO_S1
         AINT=AINT1
         R_P=R_P1
         D_P=D_P1
         CON_S=CON_S1
         DO 113 I1=1,NC
          HEAT(I1)=HEAT1(I1)
113       CONTINUE
        END IF
c set fluid parameters
        RO_F=C(L5,J)/(C(L1,J)*R)
        CP_F=CP_P(J)
        H_FS=H_FP(J)
        CON_L=CON_LPP(J)
        DO I1=1,NC
         D_L(I1)=D_LPP(I1,J)
        END DO
c peclet number for solid
        PE_N_S=U_F1*RO_S*CP_S*Z/CON_S
c peclet number for haet in gas
        PE_N_H=U_F1*RO_F*CP_F*Z/CON_L
c coefficinet in PDE's after making them dimensionless
        F3=ALF*H_FS*Z*AINT/(RO_F*CP_F*U_F1)
        F4=H_FW*Z*4/(U_F1*D_I*EPSEX*RO_F*CP_F)
        F5=H_FS*AINT*Z/(U_F1*RO_S*CP_S)
        F6=Z/(RO_S*CP_S*U_F1)
        F7=Z*H_FW*A_C/U_F1/RO_WA/CP_WA
        F8=Z*H_OW*A_INS/U_F1/RO_WA/CP_WA
c co2 partial pressure
        PCO2=C(5,J)*R*C(L2,J)
c h2o partial pressure
        PH2O=C(6,J)*R*C(L2,J)
c set dynamic parameters for each section
        IF(J.GE.LAYER1 .AND. J.LT.LAYER2)THEN
```

```
        CALL DWDXDT_SIL(PCO2,PH2O,C(L1,J),C(L2,J),C(L5,J),
     .  REY_P(J),DWDX(2),DWDT(2))
        DWDX(1)=0.
        DWDT(1)=0.
        K_F(1)=0.
        K_F(2)=.3
        K_F(3)=0.
        K_F(4)=0.
       ELSE
        IF(IPORE.EQ.1)THEN
c call to compute the solid amount in equilibruim with gas phase for silica
         CALL DWDXDT_13X(PCO2,PH2O,C(L1,J),C(L2,J),C(L5,J),
     .   REY_P(J),DWDX,DWDT)
         K_F(1)=.3
         K_F(2)=.3
        ELSE
        DO I1=1,NC
         SS(I1)=C(I1,J)
        END DO
c call to compute the solid amount in equilibruim with gas phase for 13X
        CALL IST_Z(1,C(L2,J),SS,Q2)
         DO 3 I=1,NC
          Q(I,J)=Q2(I)*RO_S
 3        CONTINUE
        END IF
        K_F(1)=.017
        K_F(2)=.0035
        K_F(3)=.05
        K_F(4)=.05
       END IF
c compute the C array for the first grid
      IF(J.EQ.1) THEN
       FLUX1=0.
       FLUX2=0.
       DO 70 M1=1,NC
        IF(M1.EQ.1) THEN
         IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
          W00=0
         ELSE
          W00=W0
         END IF
        ELSE
         W00=W1
        ENDIF
c compute the amount adsorbed in pore of pellet for silica gel
        IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
         F1=15.*K_F(M1)/R_P/R_P*Z/U_F1
         A1=1./(EPSP/DELT+F1+RO_S*DWDX(M1)/DELT)
         C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+A1*(F1*C(M1,J)+
     .   (DWDX(M1)*RO_S+EPSP)*P(NC+M1,J)/DELT-DWDT(M1)*(C(L1,J)-
     .   P(L1,J))/DELT))
c compute total heat of adsorption
         FLUX1=FLUX1+HEAT(M1)*F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
c compute total amount adsorbed
         FLUX2=FLUX2+F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
        ELSE
c compute the amount adsorbed in solid phase for 13X
         F1=K_F(M1)*AINT*Z/U_F1
         A2=1/DELT+F1
         C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+1/A2*(F1*Q(M1,J)+
     .   P(NC+M1,J)/DELT))
c compute total heat of adsorption
         FLUX1=FLUX1+HEAT(M1)*3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
c compute the amount adsorbed in solid phase for 13X
```

```fortran
          FLUX2=FLUX2+3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
         END IF
 70      CONTINUE
        RATE_C1(J)=FLUX2
        DO 85 M1=1,NC
         PE_N_M(M1)=U_F1*Z/D_L(M1)
         A(M1)=1.0D0/DELT+2/PE_N_M(M1)/DELZ**2+1/U_F1/DELZ*C(L4,J)
 85      CONTINUE
c compute molar concentration in the gas phase
        DO 90 M1=1,NC
         IF(M1.EQ.1)THEN
          W11=W0
         ELSE
          W11=W1
         ENDIF
         IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
          F2=15.*ALF*K_F(M1)/R_P/R_P*Z/U_F1
          PART1=F2*(C(M1,J)-C(M1+NC,J))
         ELSE
          F2=ALF*K_F(M1)*3./R_P*Z/U_F1
          PART1=F2*(Q(M1,J)-C(NC+M1,J))
         END IF
         C(M1,J)=C(M1,J)+W11*(-C(M1,J)+1.D0/A(M1)*(1.0D0/PE_N_M(M1)/
     .   DELZ**2*(BC_C(M1)+C(M1,J+1))-1.0D0/U_F1/DELZ*(-BC_C(M1)*
     .   BC_L4)-PART1+P(M1,J)/DELT))
 90      CONTINUE
c compute gas temperature
        AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
        C(L1,J)=C(L1,J)+W2*(-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*
     .  (BC_L1+C(L1,J+1))-1.0D0/(DELZ*U_F1)*(-BC_L1*BC_L4)-F3*
     .  (-C(L2,J))-F4*(-C(L3,J))+P(L1,J)/DELT))
c compute solid temperature
        C(L2,J)=C(L2,J)+W3*(-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2
     .  *(BC_L2-2*C(L2,J)+C(L2,J+1))+F5*(C(L1,J)-C(L2,J))-F6*FLUX1)
     .  +P(L2,J))
c compute wall temperature
        C(L3,J)=DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+P(L3,J)
C
C-----------------------------------------------------------------
C THE LAST ROW
C computaion of C array for the last grid
c
       ELSE IF(J .EQ. NJ) THEN
        FLUX1=0
        FLUX2=0
        DO 87 M1=1,NC
         IF(M1.EQ.1) THEN
          IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
           W00=0
          ELSE
           W00=W0
          END IF
         ELSE
          W00=W1
         ENDIF
c compute amount adsorbed in the pore of pellet, silical gel
         IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
          F1=15.*K_F(M1)/R_P/R_P*Z/U_F1
          A1=1./(EPSP/DELT+F1+RO_S*DWDX(M1)/DELT)
          C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+A1*(F1*C(M1,J)+
     .    (DWDX(M1)*RO_S+EPSP)*P(NC+M1,J)/DELT-DWDT(M1)*(C(L1,J)-
     .    P(L1,J))/DELT))
c compute total heat of adsorption, silical gel
          FLUX1=FLUX1+HEAT(M1)*F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
```

```
c compute total amount adsorbed, silica gel
        FLUX2=FLUX2+F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
       ELSE
c compute amount adsorbed in the solid of pellet, 13X
        F1=K_F(M1)*AINT*Z/U_F1
        A2=1/DELT+F1
        C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+1/A2*(F1*Q(M1,J)+
   .    P(NC+M1,J)/DELT))
c compute total heat of adsorption, 13X
        FLUX1=FLUX1+HEAT(M1)*3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
c compute total amount adsorbed, 13X
        FLUX2=FLUX2+3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
       END IF
 87    CONTINUE
      RATE_C1(J)=FLUX2
c compute molar concentration of each component
      DO 125 M1=1,NC
       PE_N_M(M1)=U_F1*Z/D_L(M1)
       A(M1)=1.D0/DELT+2.D0/PE_N_M(M1)/DELZ**2+1/U_F1/DELZ*C(L4,J)
 125   CONTINUE
      DO 130 M1=1,NC
       IF(M1.EQ.1)THEN
        W11=W0
       ELSE
        W11=W1
       ENDIF
       IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
        F2=15.*ALF*K_F(M1)/R_P/R_P*Z/U_F1
        PART1=F2*(C(M1,J)-C(M1+NC,J))
       ELSE
        F2=ALF*K_F(M1)*3./R_P*Z/U_F1
        PART1=F2*(Q(M1,J)-C(NC+M1,J))
       END IF
       C(M1,J)=C(M1,J)+W11*(-C(M1,J)+1.D0/A(M1)*(1.0D0/PE_N_M(M1)/
   .   DELZ**2*(C(M1,J-1)+C(M1,J-1))-1.0D0/U_F1/DELZ*(-C(M1,J-1)*
   .   C(L4,J-1))-PART1+P(M1,J)/DELT))
 130   CONTINUE
c gas temperature
      AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
      C(L1,J)=C(L1,J)+W2*(-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*
   . (C(L1,J-1)+C(L1,J-1))-1.0D0/(DELZ*U_F1)*(-C(L1,J-1)*C(L4,J-1))-
   . F3*(-C(L2,J))-F4*(-C(L3,J))+P(L1,J)/DELT))
c solid temperature
      C(L2,J)=C(L2,J)+W3*(-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2
   . *(C(L2,J-1)-2*C(L2,J)+C(L2,J-1))+F5*(C(L1,J)-C(L2,J))
   . -F6*FLUX1)+P(L2,J))
c wall temperatur
      C(L3,J)=DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+P(L3,J)

C
C-----------------------------------------------
C computation of C array for interior grids
C INTERIOR ROWS
C
      ELSE
       FLUX1=0
       FLUX2=0
       DO 198 M1=1,NC
        IF(M1.EQ.1) THEN
         IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
          W00=0
         ELSE
          W00=W0
         END IF
```

```
        ELSE
         W00=W1
        ENDIF
c compute amount adsorbed in pore of pellet, silica gel
        IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
         F1=15.*K_F(M1)/R_P/R_P*Z/U_F1
         A1=1./(EPSP/DELT+F1+RO_S*DWDX(M1)/DELT)
         C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+A1*(F1*C(M1,J)+
     .   (DWDX(M1)*RO_S+EPSP)*P(NC+M1,J)/DELT-DWDT(M1)*(C(L1,J)-
     .   P(L1,J))/DELT))
c compte toal heat of adsorption
         FLUX1=FLUX1+HEAT(M1)*F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
c compute toal amount adsorbed
         FLUX2=FLUX2+F1*U_F1/Z*(C(M1,J)-C(NC+M1,J))
c compute amount adsorbed in pore of pellet, 13X
        ELSE
         F1=K_F(M1)*AINT*Z/U_F1
         A2=1/DELT+F1
         C(NC+M1,J)=C(NC+M1,J)+W00*(-C(NC+M1,J)+1/A2*(F1*Q(M1,J)+
     .   P(NC+M1,J)/DELT))
c compte toal heat of adsorption
         FLUX1=FLUX1+HEAT(M1)*3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
c compute toal amount adsorbed
         FLUX2=FLUX2+3.0/R_P*K_F(M1)*(Q(M1,J)-C(NC+M1,J))
        END IF
 198    CONTINUE
       RATE_C1(J)=FLUX2
c if the computaion is where two different sizes of grids are adjsent then
      IF(ITESTJ.EQ.1) THEN
       DELZ=DELZ_OLD
       DO 165 M1=1,NC
        PE_N_M(M1)=U_F1*Z/D_L(M1)
        A(M1)=1.D0/DELT+2.D0/PE_N_M(M1)/DELZ**2*D12+1.D0/U_F1/DELZ*
     .  C(L4,J)*D11*0
 165    CONTINUE
       DO 170 M1=1,NC
        IF(M1.EQ.1)THEN
         W11=W0
        ELSE
         W11=W1
        ENDIF
        IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
         F2=15.*ALF*K_F(M1)/R_P/R_P*Z/U_F1
         PART1=F2*(C(M1,J)-C(M1+NC,J))
        ELSE
         F2=ALF*K_F(M1)*3./R_P*Z/U_F1
         PART1=F2*(Q(M1,J)-C(NC+M1,J))
c compute molar concentration of gas
        END IF
        C(M1,J)=C(M1,J)+W11*(-C(M1,J)+1.D0/A(M1)*(2.0D0/PE_N_M(M1)/
     .  DELZ**2*(C(M1,J-1)*D02+C(M1,J+1)*D22)-1.0D0/U_F1/DELZ*
     .  (-C(M1,J-1)*D01+C(M1,J+1)*D21-C(M1,J)*D11)*C(L4,J-1)-PART1+
     .  P(M1,J)/DELT))
 170    CONTINUE
       AL1=1./DELT+2./PE_N_H/DELZ**2*D12+C(L4,J)/DELZ/U_F1*D11*0+
     . F3+F4
c gas temperature
       C(L1,J)=C(L1,J)+W2*(-C(L1,J)+1./AL1*(2.0D0/PE_N_H/DELZ**2*
     . (C(L1,J-1)*D02+C(L1,J+1)*D22)-1.0D0/(DELZ*U_F1)*(-C(L1,J-1)*
     . D01+C(L1,J+1)*D21-D11*C(L1,J))*C(L4,J-1)-F3*(-C(L2,J))-
     . F4*(-C(L3,J))+P(L1,J)/DELT))
c solid temperature
       C(L2,J)=C(L2,J)+W3*(-C(L2,J)+DELT*(2.0D0/PE_N_S/DELZ**2
     . *(C(L2,J-1)*D02-C(L2,J)*D12+C(L2,J+1)*D22)+F5*(C(L1,J)-
```

```
     . C(L2,J))-F6*FLUX1)+P(L2,J))
c if the adjsent grids are the same sizes
      ELSE
       DO 161 M1=1,NC
        PE_N_M(M1)=U_F1*Z/D_L(M1)
        A(M1)=1.D0/DELT+2.D0/PE_N_M(M1)/DELZ**2+1.D0/U_F1/DELZ*
     .  C(L4,J)
 161    CONTINUE
       DO 171 M1=1,NC
        IF(M1.EQ.1)THEN
         W11=W0
        ELSE
         W11=W1
        ENDIF
        IF(J.GE.LAYER1 .AND. J.LE.LAYER2)THEN
         F2=15.*ALF*K_F(M1)/R_P/R_P*Z/U_F1
         PART1=F2*(C(M1,J)-C(M1+NC,J))
        ELSE
         F2=ALF*K_F(M1)*3./R_P*Z/U_F1
         PART1=F2*(Q(M1,J)-C(NC+M1,J))
        END IF
c molar concentration in the gas phase
        C(M1,J)=C(M1,J)+W11*(-C(M1,J)+1.D0/A(M1)*(1.0D0/PE_N_M(M1)/
     .  DELZ**2*(C(M1,J-1)+C(M1,J+1))-1.0D0/U_F1/DELZ*(-C(M1,J-1)*
     .  C(L4,J-1))-PART1+P(M1,J)/DELT))
 171    CONTINUE
c gas temperature
       AL1=1./DELT+2./PE_N_H/DELZ**2+C(L4,J)/DELZ/U_F1+F3+F4
       C(L1,J)=C(L1,J)+W2*(-C(L1,J)+1./AL1*(1.0D0/PE_N_H/DELZ**2*
     . (C(L1,J-1)+C(L1,J+1))-1.0D0/(DELZ*U_F1)*(-C(L1,J-1)*C(L4,J-1))-
     . F3*(-C(L2,J))-F4*(-C(L3,J))+P(L1,J)/DELT))
c solid temperature
       C(L2,J)=C(L2,J)+W3*(-C(L2,J)+DELT*(1.0D0/PE_N_S/DELZ**2
     . *(C(L2,J-1)-2*C(L2,J)+C(L2,J+1))+F5*(C(L1,J)-C(L2,J))
     . -F6*FLUX1)+P(L2,J))
      END IF
c wall temperature
       C(L3,J)=DELT*(F7*(C(L1,J)-C(L3,J))-F8*(C(L3,J)-TO))+P(L3,J)
C-----------------------------------------------------------------
      END IF
 190   CONTINUE
      IF (IL.LT.IIO) THEN
       GO TO 220
      END IF
c check for convergance
      DO 200 I1=1,NC
       DO 200 I2=1,NJ
        IF(C(I1,I2).NE.0) THEN
         IF(ABS(C(I1,I2)-C2(I1,I2))/C(I1,I2).GT.ERROR) GO TO 220
        END IF
 200   CONTINUE
      IF (IL.LT.100)GO TO 220
c call diffeq and compute the total pressure and velosity
     DO 300 I2=1,NJ
      C1(1,I2)=C(L4,I2)
      P1(1,I2)=P(L4,I2)
      C1(2,I2)=C(L5,I2)*760./14.696
      P1(2,I2)=P(L5,I2)*760./14.696
      C1(3,I2)=C(L6,I2)*760./14.696
      P1(3,I2)=P(L6,I2)*760./14.696
      C1(4,I2)=C(L1,I2)
      P1(4,I2)=P(L1,I2)
 300  CONTINUE
     CALL DIFFEQ(C1,P1,NJ,3)
```

```
      DO 400 I2=1,NJ
       C(L4,I2)=C1(1,I2)
       C(L5,I2)=C1(2,I2)*14.696/760.
       C(L6,I2)=C1(3,I2)*14.696/760.
400   CONTINUE
      GO TO 210
220   CONTINUE
210   RETURN
      END
c
c--------------------------------------------------------------
c this subroutine compute the equilibruim amount adsorbed in the solid phase
c of pellet, for silica gel
       SUBROUTINE DWDXDT_SIL(PCO2,PH2O,TG,TS,P,REY,DWDX2,DWDT3)
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 KF,LEN,MW
       COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
       COMMON/GAS/INERT,NCOMP
       COMMON/P0/PH2O1,XP
       COMMON/IND1/IND(4)
       DATA MW/18/
       DWDX1=DWDX_SIL(PH2O,P,TS,TG)
       DWDX2=DWDX1*1545.12*TS/MW
       DWDT3=DWDX1*PH2O1/TS*XP/TS
       RETURN
       END
c
c--------------------------------------------------------------
c this subroutine calculate the derivitive of pre concentration with respect,
c to temperature and other componet,silica gel
c
       FUNCTION DWDX_SIL(PH2O2,P,TS,TG)
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON/IND1/IND(4)
       COMMON/P0/PH2O1,XP
       DATA R/1545.0/
       DATA C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13/-5674.5359,
     . 6.3925247,-9.677843E-3,6.2215701E-7,2.0747825E-9,9.484024E-13,
     . 4.1635019,-5800.2206,1.3914993,-4.860239E-2,4.1764768E-5,
     . -1.4452093E-8,6.5459673/
       DATA R/1545.0/
       ITEST=1
       T1=TS/1.8
       IF(T1.LE.273.15)THEN
        P0=EXP(C1/T1+C2+C3*T1+C4*T1**2+C5*T1**3+C6*T1**4+C7*LOG(T1))
       ELSE
        P0=EXP(C8/T1+C9+C10*T1+C11*T1**2+C12*T1**3+C13*LOG(T1))
       END IF
       P0=P0/47.880
       PH2O=PH2O2*144.0
       IF(ITEST.EQ.1)THEN
        PH2O1=PH2O
        IF(PH2O.GE.1.0D-10)THEN
         XP=TS*(LOG(P0/PH2O))
        ELSE
         PH2O1=1.0D-10
         XP=TS*(LOG(P0/PH2O1))
        END IF
        PART1=1.+(XP/1200.)*(XP/1200.)
        PART2=-.177*ATAN(XP/1200.)
        PART3=-8.7025E-7*XP*PART1**(-2.77)*EXP(PART2)
        PART4=-5.08875E-5*1./PART1*EXP(PART2)*PART1**(-1.77)
        DWDX_SIL=-(PART3+PART4)*TS/PH2O1
       ELSE
```

```
        PH2O1=PH2O
        IF(PH2O1.LE.1.0E-10) PH2O1=1.0E-10
        XP=TS*(LOG(P0/PH2O1))
        W=.354*(1.0+(XP/1200)**2)**(-1.77)*
     .  EXP(-.177*ATAN(XP/1200))
        XP1=XP+10.0
        W1=.354*(1.0+(XP1/1200)**2)**(-1.77)*
     .  EXP(-.177*ATAN(XP1/1200))
        DWDX_SIL=-TS*(W1-W)/(XP1-XP)/PH2O1
      END IF
      RETURN
      END
c
c-------------------------------------------------------------------
c this subroutine compute the equilibruim amount adsorbed in the solid phase
c of pellet, for 13x if it being treated as monodisperse pore
      SUBROUTINE DWDXDT_13X(PCO2,PH2O,TG,TS,P,REY,DWDX,DWDT)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 KF,LEN,MW(4),DWDX(4),DWDT(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/GAS/INERT,NCOMP
      COMMON/IND1/IND(4)
      DATA NC,MW/2,44.,18.,0.,0./
      CALL DWDX_13X(PCO2,PH2O,P,TS,TG,DWDX,DWDT)
      DO 10 I1=1,NC
       DWDX(I1)=DWDX(I1)*555.0*TS
 10    CONTINUE
      RETURN
      END
c
c-------------------------------------------------------------------
c this subroutine calculate the derivitive of pre concentration with respect
c to temperature and other componet,13x
      SUBROUTINE DWDX_13X(PCO2,PH2O,P,TSO,TG,DWDX,DWDT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IND1/IND(4)
      REAL*8 Q0(4),B(4),N(4),P1(4),W(4),DWDX(4),TERM(4),DWDT(4),
     . Q2(4),Q1(4)
      DATA R,NC/555.,2/
      ITEST=0
 10    Q0(1)=(-.0541863937*TSO+56.6418249)/100
      B(1)=28.19709466*(1+(TSO/9588.8566167)**1.4188024497)**
     . -331.23533278
      N(1)=-4.24259381+.02210824511*TSO-3.5706432963D-5*TSO*TSO+
     . 2.0564581055D-8*TSO*TSO*TSO
      Q0(2)=.015
      B(2)=1.879094E-4*EXP(5467.4817024/TSO)
      N(2)=.5
      P1(1)=PCO2/14.696*760.0
      P1(2)=PH2O/14.696*760.0
      IF(P1(1).LT.1.0E-10) P1(1)=1.0E-10
      IF(P1(2).LT.1.0E-10) P1(2)=1.0E-10
      TERM1=0.
      DO 20 I=1,NC
       TERM(I)=B(I)*P1(I)**N(I)
       TERM1=TERM1+TERM(I)
 20    CONTINUE
      IF(ITEST.EQ.0) THEN
        DWDX(1)=Q0(1)*B(1)*N(1)*P1(1)**(N(1)-1.)*
     .  (1+TERM(2))/(1+TERM1)**2
        DWDX(2)=Q0(2)*B(2)*N(2)*P1(2)**(N(2)-1.)*
     .  (1+TERM(1))/(1+TERM1)**2
        DO 30 I1=1,NC
          Q1(I1)=Q0(I1)*B(I1)*P1(I1)**N(I1)/TERM1
```

```
30     CONTINUE
        TSO1=TSO
        TSO=TSO+1.0
        ITEST=1
        GO TO 10
       ELSE
        DO 40 I1=1,NC
         Q2(I1)=Q0(I1)*B(I1)*P1(I1)**N(I1)/TERM1
40       CONTINUE
       END IF
       DO 50 I1=1,NC
        DWDT(I1)=(Q2(I1)-Q1(I1))/(TSO-TSO1)
50      CONTINUE
       RETURN
       END

C-----------------------------------------------------------
C
C
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is by Forythe,
c Computer Methods for Mathematical Computation. it is an bisect method with
c quadratic convergence.
c
c B,V,PO arrays are sinle equlibrum constant
       SUBROUTINE IST_Z1(METHOD,T,SS,Q1)
       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
       COMMON/NCY/NC1
       COMMON/IND1/IND(4)
       REAL*8 Q1(4),SS(4)
       INTEGER I,NLIM
       EXTERNAL FCN1
       DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
       DATA R/555/
       T_G=T
       NC=NC1
c the partial pressure
       DO I1=1,NC
        Y1(I1)=SS(I1)
            END DO
       DO I=1,NC
        PP(I)=Y1(I)*T_G*R
       END DO
c no mole fration return
       Z=1.0E-32
       IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
        DO I=1,NC
         Q1(I)=0.
        END DO
        RETURN
       END IF
       N1=1
c set the constant as a function of temperature
      IF(IND(1).EQ.1)THEN
C        V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
        V(N1)=17.0/44/100
C        B(N1)=6864.9000131*EXP(-.019625791466*T)
        B(N1)=5.323235056e-6*T**(-.5)*EXP(13948.544244/1.987/T)
            PO(N1)=.8
        N1=N1+1
```

```
    END IF
    IF(IND(2).EQ.1)THEN
     V(N1)=1./69.035
     B(N1)=1.879094E-4*EXP(5467.4817024/T)
     IF(T.LT.610.) THEN
      B(N1)=4.5597278759E-7*EXP(9628.9655743/T)
     ELSE
      B(N1)=5.8089066684E-7*EXP(9115.734593/T)
     END IF
     N1=N1+1
         PO(N1)=1.0
    END IF
    IF(IND(3).EQ.1)THEN
     V(N1)=-1.637879912E-5*T+.00961297026
     IF(T.LE.532)THEN
      B(N1)=3.2694515539E-7*T+4.59988799E-4
     ELSE
      B(N1)=7.90864008E-5*T-4.14400420E-2
     END IF
         PO(N1)=1.0
     N1=N1+1
    END IF
    IF(IND(4).EQ.1)THEN
     V(N1)=-1.637879912E-5*T+.00961297026
     IF(T.LE.532)THEN
      B(N1)=3.2694515539E-7*T+4.59988799E-4
     ELSE
      B(N1)=7.90864008E-5*T-4.14400420E-2
     END IF
         PO(N1)=1.0
    END IF
    IF(NC.EQ.1)THEN
     Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
     RETURN
    ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
     Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
     Q1(1)=0
     RETURN
    ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
     Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
     Q1(2)=0
     RETURN
    END IF
    TERM1=0.
c calculate the equilibruim isthotherm by Langmuir method
    DO I=1,NC
     IF(PP(I).GT.0)THEN
      TERM1=B(I)*PP(I)**PO(I)+TERM1
     END IF
    END DO
    SUM=0.
    DO I=1,NC
     IF(PP(I).LE.0.) THEN
      Q1(I)=0.
     ELSE
      Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
      SUM=SUM+Q1(I)
     END IF
    END DO
c if the Langmuir method is asked for then returned
    IF(METHOD.EQ.1)THEN
     RETURN
    END IF
c if not, take the result as the first guess for IST theory
```

```
      DO I=1,NC
       X1(I)=Q1(I)/SUM
       IF(X1(I).GT.0) THEN
        PI(I)=PP(I)/X1(I)
       ELSE
        PI(I)=0.
       END IF
      END DO
c calculate the spread pressure
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        X1(I)=0
       ELSE
        X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
        IF=I
       END IF
      END DO
      X=X1(IF)
      DO I=1,NC
       IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
        X=X1(I)
       END IF
      END DO
      AX=X
      DELX=AX
      BX=AX
c call Zeroin subroutine to find the rout to the IST equation
      DO I=1,100
       BX=BX+DELX
       FUN=FCN1(BX)
       IF(FUN.LT.0)GO TO 113
      END DO
113   X=ZEROIN(FCN1,AX,BX,TOL)
c rout was found
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        PP(I)=1.0E-32
       END IF
c calculate the fraction in the solid phase
       PART1=PO(I)*X/V(I)
       IF(PART1.GT.73)PART1=73
       PI(I)=(EXP(PART1)-1)/B(I)
       X1(I)=PP(I)/PI(I)
      END DO
      TOT_Q=0
c calculate the total amount adsorbed
      DO I=1,NC
       Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
       TOT_Q=TOT_Q+X1(I)/Q1(I)
      END DO
c calculate the amount adsorbed for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80   RETURN
      END
c IST function
      REAL FUNCTION FCN1(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
```

```
      IF(PP(I).GT.0) THEN
       PART1=PO(I)*X/V(I)
       IF(PART1.GT.73)PART1=73
       SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
      END IF

C      IF(PP(I).LE.0)PP(I)=1.0E-32
C      PART1=PO(I)*X/V(I)
C      IF(PART1.GT.73)PART1=73
C      SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
     END DO
     FCN1=SUM-1
     RETURN
     END
c
c------------------------------------------------------------
c
c subroutine to find the root of equation by bisect method

      REAL FUNCTION ZEROIN(FCN1,AX,BX,TOL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IND1/IND(4)
      REAL*8 AX,BX,FCN1,TOL
      REAL*8 A,B,C,D,E,EPS,FA,FB,FC,TOL1,XM,P,Q,R,S
      EPS=1.0
10     EPS=EPS/2.
      TOL1=1.0+EPS
      IF(TOL1.GT.1.0) GO TO 10
c initialization
      A=AX
      B=BX
      FA=FCN1(A)
      FB=FCN1(B)
c begin step
20     C=A
      FC=FA
      D=B-A
      E=D
30     IF(ABS(FC) .GE. ABS(FB)) GO TO 40
      A=B
      B=C
      C=A
      FA=FB
      FB=FC
      FC=FA
c convergence test
40     TOL1=2.*EPS*ABS(B)+.5*TOL
      XM=.5*(C-B)
      IF(ABS(XM).LE.TOL1)GO TO 90
      IF(FB.EQ.0.0) GO TO 90
c is bisection necessary
      IF(ABS(E).LE.TOL1)GO TO 70
      IF(ABS(FA).LE.ABS(FB)) GO TO 70
c is quaratic interpolatation possible
      IF(A.NE.C)GO TO 50
c linear interpolation
      S=FB/FA
      P=2.0*XM*S
      Q=1.0-S
      GO TO 60
c inverse quadratic interpoltation
50     Q=FA/FC
      R=FB/FC
      S=FB/FA
```

```
      P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.0))
      Q=(Q-1.0)*(R-1.0)*(S-1.0)
c adjust signs
60    IF(P.GT.0.0)Q=-Q
      P=ABS(P)
c is interpolation acceptable
      IF((2.0*P).GE.(3.*XM*Q-ABS(TOL1*Q))) GO TO 70
      IF(P.GE.ABS(.5*E*Q)) GO TO 70
      E=D
      D=P/Q
      GO TO 80
c bisection
70     D=XM
      E=D
c complete step
80     A=B
      FA=FB
      IF(ABS(D).GT.TOL1)B=B+D
      IF(ABS(D).LE.TOL1)B=B+SIGN(TOL1,XM)
      FB=FCN(B)
      IF((FB*(FC/ABS(FC))).GT.0)GO TO 20
      GO TO 30
c done
90     ZEROIN=B
      RETURN
      END
C
C-----------------------------------------------------------------
C
C
c the second method. this method is faster but the initial guess must be near
c the root of the equation.
c
c this subroutine compute the amount of adsorbed gas in equilibrium with gas
c molar density. for single componet uses Langmuir-fredrich isotherm. the
c computaion  of equlibruim for multi component uses the Ideal Solution Theory
c (IST). since the equations are none linear and implicit,a numeriacl method
c was used to compute the adsorbed equlibruim amount. the method is Newton,
c
c B,V,PO arrays are sinle equlibrum constant
      SUBROUTINE IST_Z(METHOD,T,SS,Q1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4),PI(4),XI(4),X1(4),Y1(4)
      COMMON/NCY/NC1
      COMMON/IND1/IND(4)
      REAL*8 Q1(4),SS(4)
      INTEGER I,NLIM
      EXTERNAL FCN,FDER
      DATA XTOL,FTOL,I,NLIM/1.0E-5,1.0E-5,0,50/
      DATA R/555.0/
      T_G=T
      NC=NC1
      DO I1=1,NC
        Y1(I1)=SS(I1)
           END DO
      DO I=1,NC
        PP(I)=Y1(I)*T_G*R
      END DO
 c no mole fration return
      Z=1.0E-32
      IF(PP(1).LE.Z.AND.PP(2).LE.Z.AND.PP(3).LE.Z.AND.PP(4).LE.Z)THEN
        DO I=1,NC
          Q1(I)=0.
        END DO
```

```
        RETURN
       END IF
       N1=1
c set the constant as a function of temperature
      IF(IND(1).EQ.1)THEN
c 13x by l, grace
c        point=-1567.1205874+9.00256903*t-.0172347*t*t+1.100806e-5*t*t*t
c        if(pp(n1).gt.point .or. t.gt.627)then
c         V(N1)=(81.2983138-.21118560043*T+1.587732293D-4*T*T)/44/100
c         B(N1)=6864.9000131*EXP(-.019625791466*T)
c         po(n1)=1.0
c        else
c 13x by l_f,grace
        v(n1)=23.32228/44/100
        b(n1)=4.9639763e-4*T**(-.5)*exp(9233.33778/1.987/T)
        po(n1)=-2.9138991288+.017181761178*T-2.8549083257e-5*T*T+
     .  1.622511757e-8*T*T*T
c        end if
c from Finn data 5A
c         if(pp(n1).lt.1.0) then
C           b(N1)=2.37022397e-6*t**(-.5)*exp(14907.6535/1.987/t)
C           v(N1)=16.6/44/100
C           po(N1)=.80
c Finn by langmuir
c         else
c         v(n1)=(25.97135008-6.084518e-3*T-2.1799516646e-5*T*T)/44/100
c         b(n1)=1.88863393e-5*T**(-.5)*exp(12170.875335/1.987/T)
c         po(n1)=1.
c         end if
C 5A BY GRACE, BY L_F
c         V(N1)=17.0/100/44
c         B(N1)=2.341477E-4*T**(-.5)*EXP(10257.166145/1.987/T)
c         PO(N1)=3.830450111E-4*T**1.188379596
C 5A BY GRACE, BY L
C         V(N1)=(-4.27886889+.087218022*T-9.1010715804E-5*T*T)/44/100
C         B(N1)=9.2533309123E-5*T**(-.5)*EXP(10719.0/1.987/T)
C         PO(N1)=1.
C 5A BY GRACE; BY L_F BETWEEN 0-75 C
C         V(N1)=(399.3942-1.938428896*T+3.2540515E-3*T*T-
C     .   1.8226211899E-6*T*T*T)/44/100
C         B(N1)=35.358072159-.17016733*T+2.7458762E-4*T*T-
C     .   1.484116035E-7*T**3
C         PO(N1)=-29.3861079+.1538898497*T-2.61225877E-4*T*T+
C     .   1.478184694E-7*T**3
c linde 5a
c         v(n1)=.005
c         b(n1)=1.129015193e-5*exp(5055.015089/t)
c         po(n1)=.6
        N1=N1+1
       END IF
c h2o on 5a by grace l,l
       IF(IND(2).EQ.1)THEN
        if(pp(n1).lt. .46)then
         if(t.le.564.)then
           v(n1)=(39.914452-8.87103e-2*t+6.839502987e-5*t*t)/100/18
         else
           v(n1)=(873.44464-3.867834937*t+5.80375049e-3*t*t-
     .     2.9346685e-6*t*t*t)/100/18
         end if
         b(n1)=29690.66923-137.837129*T+.214456126*t*t
         po(n1)=1
        else
         b(n1)=48.566639771-2.2620239975e-1*t+3.9101703168e-4*t*t-
     .   2.379098497e-7*t*t*t
```

```
          v(n1)=(95.093824733-.21725775358*t+1.4628603e-4*t*t)/100/18
          po(n1)=1
         end if
         IF(T.LT.610.) THEN
          B(2)=4.5597278759E-7*EXP(9628.9655743/T)
         ELSE
          B(2)=5.8089066684E-7*EXP(9115.734593/T)
         END IF
         V(2)=.015
         PO(2)=1.0
c h2o on 5a grace by L.F
C         b(n1)=24.573259-5.8385278e-2*t+3.473682438e-5*t*t
C         v(n1)=(66.62520167-.12299345*t+7.1897316221e-5*t*t)/100/18
C         if(t.le.600)then
C           po(n1)=.8219916486-3.3519986e-3*t+4.679276479e-6*t*t
C         else
C           po(n1)=-6.0926707634+2.043004e-2*t-2.924858439e-5*t*t+
C     .     1.1746734108e-8*t*t*t
C         end if
         N1=N1+1
        END IF
        IF(IND(3).EQ.1)THEN
         V(N1)=-1.637879912E-5*T+.00961297026
         IF(T.LE.532)THEN
          B(N1)=3.2694515539E-7*T+4.59988799E-4
         ELSE
          B(N1)=7.90864008E-5*T-4.14400420E-2
         END IF
             PO(N1)=1.0
         N1=N1+1
        END IF
        IF(IND(4).EQ.1)THEN
         V(N1)=-1.637879912E-5*T+.00961297026
         IF(T.LE.532)THEN
          B(N1)=3.2694515539E-7*T+4.59988799E-4
         ELSE
          B(N1)=7.90864008E-5*T-4.14400420E-2
         END IF
             PO(N1)=1.0
        END IF
        IF(NC.EQ.1)THEN
         Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
         RETURN
        ELSE IF(NC.EQ.2 .AND. PP(1).EQ.0)THEN
         Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
         Q1(1)=0
         RETURN
        ELSE IF(NC.EQ.2 .AND. PP(2).EQ.0)THEN
         Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
         Q1(2)=0
         RETURN
        ELSE IF(NC.EQ.3)THEN
         if(pp(1).le.0 .and. pp(2).le.0)then
          Q1(3)=V(3)*B(3)*PP(3)**PO(3)/(1+B(3)*PP(3)**PO(3))
          Q1(1)=0
          Q1(2)=0
          return
         else if(pp(1).le.0 .and. pp(3).le.0)then
          Q1(2)=V(2)*B(2)*PP(2)**PO(2)/(1+B(2)*PP(2)**PO(2))
          Q1(1)=0
          Q1(3)=0
          return
         else if(pp(2).le.0 .and. pp(3).le.0)then
          Q1(1)=V(1)*B(1)*PP(1)**PO(1)/(1+B(1)*PP(1)**PO(1))
```

```
        Q1(2)=0
        Q1(3)=0
        return
       end if
      END IF
      TERM1=0.
c calculate the equilibruim by Langmuir isotherm
      DO I=1,NC
       IF(PP(I).GT.0)THEN
        TERM1=B(I)*PP(I)**PO(I)+TERM1
       END IF
      END DO
      SUM=0.
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PP(I)**PO(I)/(1+TERM1)
        SUM=SUM+Q1(I)
       END IF
      END DO
c if the Langmuir method is asked foR, then return. If not use it as first guess
c for IST theory
      IF(METHOD.EQ.1)THEN
       RETURN
      END IF
      DO I=1,NC
       X1(I)=Q1(I)/SUM
       IF(X1(I).GT.0) THEN
        PI(I)=PP(I)/X1(I)
       ELSE
        PI(I)=0.
       END IF
      END DO
c set the initial guess for the spreading pressure
      DO I=1,NC
       IF(PP(I).LE.0.) THEN
        X1(I)=0
       ELSE
        X1(I)=V(I)/PO(I)*LOG(1+B(I)*PI(I)**PO(I))
        IF=I
       END IF
      END DO
      X=X1(IF)
      DO I=2,NC
       IF(X1(I).LT.X .AND. X1(I).GT.0)THEN
        X=X1(I)
       END IF
      END DO
c call newton method to the rout to IST equation
      IF(X .LE. 1.0E-17) GO TO 80
      CALL NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
c rout was found, calculate the fraction in the solid phase
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        X1(I)=0.
       ELSE
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        PI(I)=(EXP(PART1)-1)/B(I)
        X1(I)=PP(I)/PI(I)
       END IF
      END DO
c calculate the total amound in the solid phase
```

```
      TOT_Q=0
      DO I=1,NC
       IF(PP(I).LE.0) THEN
        Q1(I)=0.
       ELSE
        Q1(I)=V(I)*B(I)*PI(I)**PO(I)/(1+B(I)*PI(I)**PO(I))
        TOT_Q=TOT_Q+X1(I)/Q1(I)
       END IF
      END DO
c calculate the amount in equilibruim with the gas phase for each component
      DO I=1,NC
       Q1(I)=1./TOT_Q*X1(I)
      END DO
 80   RETURN
      END
c
c IST final equation
c
      REAL FUNCTION FCN(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/NCY/NC1
      COMMON/IND1/IND(4)
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART1=PO(I)*X/V(I)
        IF(PART1.GT.73)PART1=73
        SUM=SUM+PP(I)/((EXP(PART1)-1)/B(I))**(1./PO(I))
       END IF
      END DO
      FCN=SUM-1
      RETURN
      END
c
c IST final derivitive
c
      REAL FUNCTION FDER(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EQLIB/PP(4),B(4),V(4),PO(4)
      COMMON/IND1/IND(4)
      COMMON/NCY/NC1
      NC=NC1
      SUM=0.
      DO I=1,NC
       IF(PP(I).GT.0) THEN
        PART3=PO(I)*X/V(I)
        IF(PART3.LT.35)THEN
         PART1=-PP(I)/B(I)/V(I)*EXP(PART3)
         PART2=((EXP(PART3)-1)/B(I))**((1+PO(I))/PO(I))
         SUM=SUM+PART1/PART2
        ELSE IF(PART3/PO(I).LT.73)THEN
         SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .   EXP(-PART3/PO(I))
        ELSE
         PART3=73
         SUM=SUM-B(I)**((1+PO(I))/PO(I))*PP(I)/B(I)/V(I)*
     .   EXP(-PART3)
        END IF
       END IF
      END DO
      FDER=SUM
      RETURN
```

```
      END
c
c-----------------------------------------------------------
c this subroutine finds the rout to a nonlinear equation using the Newton
c method
      SUBROUTINE NEWTN(FCN,FDER,X,XTOL,FTOL,NLIM,I)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IND1/IND(4)
      INTEGER NLIM,I,J
      FX=FCN(X)
      X1=X
      DO J=1,NLIM
       DELX=FX/FDER(X)
       X=X-DELX
       FX=FCN(X)
       IF (ABS(X-X1)/X.LE.XTOL)THEN
        RETURN
       END IF
       IF(FX.NE.0)THEN
        IF(ABS(FX-FX1)/FX.LE.FTOL) THEN
         RETURN
        END IF
       END IF
       X1=X
       FX1=FX
      END DO
      I=-1
      PRINT 200, NLIM,X,FX
 200  FORMAT(/'TOLERANCE NOT MET  ',I4,' ITERATIONS X= ',
     2 E12.5,' F(X)=',E12.5)
      RETURN
      END
C
C
C
      FUNCTION CPGAS(TEM,Y)
C THIS FUNCTION CALCULATES THE HEAT CAPACITIES OF CMRIER GAS IN BTU/#MOLE/R
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 A(4),B(4),C(4),D(4),Y(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
       COMMON/IND1/IND(4)
C CO2
      DATA A/36.11,33.46,29.,29.1/
C H2O
      DATA B/4.233E-2,.6880E-2,.2199E-2,1.158E-2/
C N2
      DATA C/-2.887E-5,.7604E-5,.5723E-5,-.6076E-5/
C O2
      DATA D/7.464E-9,-3.593E-9,-2.871E-9,1.311E-9/
      NC=NC1
      T=TEM/1.8-273.15
      IF (INERT.EQ.1) THEN
       CPGAS=0.
       DO I=1,NC
        CP=A(I)+B(I)*T+C(I)*T*T+D(I)*T*T*T
        CPGAS=CP*Y(I)/4.18669+CPGAS
       END DO
      ELSE
       CPGAS=4.97
      ENDIF
      RETURN
      END
```

```
c
c-----------------------------------------------------------------------
c
      FUNCTION CPS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES TNE HEAT CAPACITY
C BTU/LB/R.
      CPS=0.2
      RETURN
      END
c
c-----------------------------------------------------------------------
c
      REAL FUNCTION KGAS (TEMP)
C THIS FUNCTION CALCULATES THE THERMAL CONDUCTIVITY OF CARRIER GAS ASSUMING
C A LINEAR FUNCTION OF TEMPERATURE AND BASED ON VALUES AT 492 AND 672 R.
C BTU/MIN/FT/R
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
        KGAS=3.88E-7*TEMP+0.4052E-4
      ELSE
        KGAS=1.667E-6*TEMP+6.1E-4
      ENDIF
      RETURN
      END
c
c-----------------------------------------------------------------------
c
      FUNCTION RE (G,TEMP)
C TNIS FUNCTION CALCULATES THE PARICLE REYNOLDS NUMBER ASSUMINC THAT
C THE MOLECULAR WEIGHT OF TNE GAS IS EQUAL TO THAT OF CARRIER GAS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,MW
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/GAS/INERT,NCOMP
      DATA IFLAG/0/
      IF (INERT.EQ.1) MW=28.0
      IF (INERT.EQ.2) MW=4.
      IF (IFLAG.EQ.0)THEN
        CONST=2.*RA
        IFLAG=1
      ENDIF
      RE=CONST*MW*ABS(G)/VIS(TEMP)
      RETURN
      END
c
c-----------------------------------------------------------------------
c
      FUNCTION VIS(TEMP)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE VISCOSITY OF CARRIER GAS AS A LINEAR FUNCTION
C OF TEMPERATURE (LB/MIN/FT).
      COMMON/GAS/INERT,NCOMP
      IF (INERT.EQ.1) THEN
C       VIS=1.0E-6*TEMP+1.65E-4
        VIS=-.0102007812+5.384663937E-5*TEMP-8.7973727E-8*TEMP*TEMP+
     2     4.811387495E-11*TEMP*TEMP*TEMP
      ELSE
        VIS=0.9444E-6*TEMP+2.863E-4
      ENDIF
      RETURN
      END
c
```

```
c--------------------------------------------------------------
c
      FUNCTION RHOG(TR,P)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
C THIS FUNCTION CALCULATES THE MOLAR DENSITY OF AN IDEAL GAS IN
C LBMOLES/CV FS.
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      DATA R/555./
      RHOG=P/R/TR
      RETURN
      END
c
c--------------------------------------------------------------
c
      FUNCTION EFFD(ICOMP,G,TEMP,RHO,PT,YO)
C THIS FUNCTION CALCVLATES TNE EFFECTIVE AXIAL DIFFUSIVITIES IN A PACXED BED
C USING THE EDWARDS AND RICHARDSON CORROLATION. (FT*FT/MIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA1,LAMBDA2,LEN,YO(4)
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/IND1/IND(4)
      DATA IFLAG/0/,LAMBDA1/0.73/
      IF(IFLAG.EQ.0)THEN
       CONST=2.*RA/EPSEX
       IFLAG=1
      ENDIF
      DIF=DIFF(ICOMP,TEMP,PT,YO)
      EFFD=LAMBDA1*DIF+CONST*ABS(G)*LAMBDA2(G,DIF,RHO)/RHO
      RETURN
      END
c
c--------------------------------------------------------------
c
      FUNCTION EFFK(G,T,CP,RHO,P,YO)
      IMPLICIT REAL*8(A-H,O-Z)
C THIS FUNCTION CALCULATES THE EFFECTIVE AXIAL THERMAL CONDUCTIVITY OF A
C PACKED BED IN BTU/FT/MIN/F.
      COMMON/NCY/NC1
       COMMON/IND1/IND(4)
      REAL*8 YO(4)
      NC=NC1
      EFFK=0.
      KK=0
      DO I1=1,NC
       SUM=CP*RHO*EFFD(I1,G,T,RHO,P,YO)
       IF(SUM.GT.0)KK=KK+1
       EFFK=SUM+EFFK
      END DO
       EFFK=EFFK/KK
      RETURN
      END
c
c--------------------------------------------------------------
c
      REAL FUNCTION LAMBDA2(G,DIF,RHO)
C THIS FUNCTION CALCULATES THE DIMENSTONLESS PARAMETER OF THE EDWARDS AND
C RICHARDS CORROLATION MODIFIED FOR SMALL DIAMETER PARTICLES.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN
      DATA IFLAG/0/
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      IF(IFLAG.EQ.0)THEN
       IF(RA.GT.0.00492)THEN
```

```
          PE=2.
        ELSE
          PE=406.4*RA
        ENDIF
        CONST=2.*RA/EPSEX
        IFLAG=1
      ENDIF
      LAMBDA2=1./PE/(1.+9.5*DIF/CONST/ABS(G)*RHO)
      RETURN
      END
c
c-----------------------------------------------------------------
c
      FUNCTION DIFF(ICOMP,TEM,P1,YO)
C  THIS FUNCTION CALCULATES THE MOLECULAR  DIFFUSION COEFFICIENTS BY CORRECTING
C  FOR PRESSURE AND TEMPERATURE (FT*FT/MIN).
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEN,YO(4),V(4),MW(4),DIF(4)
      COMMON/GAS/INERT,NCOMP
      COMMON/NCY/NC1
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/IND1/IND(4)
      DATA V/26.9,12.7,17.9,16.6/
      DATA MW/44.01,18.016,28.013,31.999/
      NC=NC1
      K=ICOMP
      P=P1/760.
      T=TEM/1.8
      DO I=1,NC
        PART1=((MW(K)+MW(I))/MW(K)/MW(I))**.5
        PART2=(V(K)**(1./3)+V(I)**(1./3))**2
        DIF(I)=PART1/PART2*T**1.75/P*1.0E-3
      END DO
      SUM=0.
      DO I=1,NC
        IF(I.NE.K)THEN
          SUM=SUM+YO(I)/DIF(I)
        END IF
      END DO
      DIFF=(1-YO(K))/SUM*3.8745/60.
      RETURN
      END
c
c-----------------------------------------------------------------
c
      FUNCTION HFILM(MW,TEMP,REY,CP)
C THIS FUNCTION CALCULATES THE FLUID FILM HEAT TRANSFER COEFFICIENT
C IN BTU/SQ FT/MIN/R USING THE CORRELATION OF PETROY AND THODOS.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 ID,KGAS,KG,LEN,MW
      COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
      COMMON/GAS/INERT,NCOMP
       COMMON/IND1/IND(4)
      KG=KGAS(TEMP)
      PR=CP/MW*VIS(TEMP)/KG
      HFILM=0.357/EPSEX*REY**0.64*PR**0.33*KG*0.5/RA
      RETURN
      END
c
c-----------------------------------------------------------------
c
C     FUNCTION DHCAR(TEMP,P)
C THIS SUBROUTINE CALCULATES HEAT OF ADSORPTION OF CARRIER GAS ONTO THE
C ACTIVATED CARBON TIMES THE TEMPERATURE DERIVATIVE OF THE SOLID PHASE
```

```
C CARRIER GAS CONCENTRATION AT THE BED PRESSURE (BTU/# SOLID/R).
C    IMPLICTT REAL*8(A-H,O-Z)
C    REAL*8 LEN
C    COMMON/GAS/INERT,NCOMP
C    COMMON/PRIME/GN2,RA,RAV,EPSEX,ALF,RHOS,ALPHA1
C    DATA R/1.9872/,A/1.28E-9/,B/3151./
C
C    IF(INERT.EQ.1) THEN
C      DH=R*B
C      DHN2=-DH*A*B*EXP(B/TEMP)*P/TEMP/TEMP
C    ELSE
C      DHN2=0.
C    ENDIF
C    RETURN
C    END
c
c---------------------------------------------------------
c
   FUNCTION DHADS(ICOMP,T)
C THIS FUNCTION CALCULATES THE ISOTERIC HEAT OF ADSORPTION. (BTU/#MOLE)
   IMPLICTT REAL*8(A-H,O-Z)
   REAL*8 LEN
   COMMON/PRIME/GN2,RA,RAV,EPSEX,EPSIN,ALF,RHOS,ALPHA1
   IF(ICOMP.EQ.2) THEN
    DHADS=20400.0
   ELSE
    DHADS=20000.0
   END IF
   RETURN
   END
c
c---------------------------------------------------------
C THIS SUBROUTINE IS BEING CALLED BY DIFFEQ1 WHICH IS ALSO BEING CALLED BY
C SUBROUTINE FUNCT2_Z TO COMPUTE THE VELOCITY AND THE PRESSURE DROP IN THE BED.
C THE EQUATIONS ARE BEING SOLVED BY NEWMAN'S METHOD.
c
      SUBROUTINE FUNCT1(J)
      IMPLICTT REAL*8(A-H,O-Z)
      REAL*8 D_L(2),M_AVE
      COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
      COMMON/BND/A(14,13),B(14,13),C(14,201),D(14,29),X(14,13),
     . Y(14,13),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201)
      COMMON/PR_OLD_Z/ RO_P(201),CP_P(201),CON_FPP(201),
     . CON_LPP(201),D_LPP(4,201),H_FP(201),H_TTP(201),H_IIP(201),
     . H_OWPP(201),H_FWPP(201),Q(4,201),VISC_P(201),RATE_C1(201),
     . WM_AVE1(201)
      COMMON/BOUN_CON/BC_L1,BC_L2,BC_L3,BC_L4,BC_L5,BC_L6,BC_C(4)
      COMMON/PROP_D_Z/DELZ1,DELT1
      COMMON/PROP_S_Z_1/CP_S1,RO_S1,AINT1,R_P1,D_P1,CON_S1,
     : G_F1,HEAT1(4)
      COMMON/PROP_S_Z_2/CP_S2,RO_S2,AINT2,R_P2,D_P2,CON_S2,
     . G_F2,HEAT2(4)
      COMMON/DELZ_DIFF/D01,D11,D21,D02,D12,D22,FR
      COMMON/NCY/NC1
      COMMON/IND1/IND(4)
      COMMON/PRIME_1/GN1,RA1,RAV1,EPSEX1,EPSIN1,ALF1,RHOS1,ALPHA11
      COMMON/PRIME_2/GN2,RA2,RAV2,EPSEX2,EPSIN2,ALF2,RHOS2,ALPHA12
      COMMON/LAYERS/LAYER1,LAYER2
      DATA L1,L4,L5,L6,R1,CONV/4,1,2,3,555.0,51.714752314/
      DATA FAC,GC/2.78450526316,416975040.0/
C
C THE FIRST ROW
C
c set variable values calculted in functe2_z subroutine
```

```
      NC=NC1
      RATE_C=RATE_C1(J)
      VISC_F=VISC_P(J)
      M_AVE=WM_AVE1(J)
      KK=0
      SUM1=0.
      DO I=1,NC
       D_L(I)=D_LPP(I,J)
       IF(D_L(I).GT.0)THEN
        KK=KK+1
        SUM1=SUM1+D_L(I)
       END IF
      END DO
      D_L_AVE=SUM1/KK
      RO_F=C(L5,J)/(C(L1,J)*R1)
c set constant for material in between
       IF(J.GE.LAYER1 .AND. J.LT.LAYER2)THEN
        EPSEX=EPSEX2
        D_P=D_P2
        ALF=ALF2
       ELSE
c set constant for material at two ends
        EPSEX=EPSEX1
        D_P=D_P1
        ALF=ALF1
       END IF
c solve the pde's for the first grid
      IF(J.EQ.1)THEN
c velosity
       F(L4)=FAC*(C(L5,J)-BC_L5*CONV)/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-BC_L4)/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/
     . D_P/D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/
     . D_P/EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c total pressure witout the heat
       F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(BC_L5*CONV-
     . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-BC_L5*CONV)/DELZ1+
     . C(L5,J)*(C(L4,J)-BC_L4)/DELZ1+R1*C(L1,J)*ALF*RATE_C
c total pressure with heat
       F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(BC_L6*CONV-
     . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-BC_L6*CONV)/DELZ1+
     . C(L6,J)*(C(L4,J)-BC_L4)/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J+1)-2*C(L1,J)+BC_L1)+
     . C(L4,J)*(C(L1,J+1)-BC_L1)/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C

C
C-----------------------------------------------------
C last grid
       ELSE IF(J.EQ.NJ)THEN
c velosity
       F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
     . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
     . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c total pressure without heat
       F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J-1)-
     . 2*C(L5,J)+C(L5,J-1))+C(L5,J)*(C(L5,J)-C(L5,J-1))/DELZ1+
     . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1+R1*C(L1,J)*ALF*RATE_C
c total pressure with heat
       F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(C(L6,J-1)-
     . 2*C(L6,J)+C(L6,J-1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ1+
     . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J-1)-2*C(L1,J)+C(L1,J-1))+
     . C(L4,J)*(C(L1,J-1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C
```

```
C
C--------------------------------------------------------
C interior grids
     ELSE
c velocity
      F(L4)=FAC*(C(L5,J)-C(L5,J-1))/DELZ1+1./GC*(RO_F*M_AVE*C(L4,J)*
     . (C(L4,J)-C(L4,J-1))/DELZ1+150*(1-EPSEX)**2*VISC_F*C(L4,J)/D_P/
     . D_P/EPSEX/EPSEX+1.75*(1-EPSEX)*RO_F*M_AVE*C(L4,J)**2/D_P/
     . EPSEX+RO_F*M_AVE*(C(L4,J)-P(L4,J))/DELT1)
c total pressure without heat
      F(L5)=(C(L5,J)-P(L5,J))/DELT1-D_L_AVE/DELZ1**2*(C(L5,J-1)-
     . 2*C(L5,J)+C(L5,J+1))+C(L4,J)*(C(L5,J)-C(L5,J-1))/DELZ1+
     . C(L5,J)*(C(L4,J)-C(L4,J-1))/DELZ1+R1*C(L1,J)*ALF*RATE_C
c total pressure with heat
      F(L6)=(C(L6,J)-P(L6,J))/DELT1-D_L_AVE/DELZ1**2*(C(L6,J-1)-
     . 2*C(L6,J)+C(L6,J+1))+C(L4,J)*(C(L6,J)-C(L6,J-1))/DELZ1+
     . C(L6,J)*(C(L4,J)-C(L4,J-1))/DELZ1-C(L6,J)/C(L1,J)*((C(L1,J)-
     . P(L1,J))/DELT1-D_L_AVE/DELZ1**2*(C(L1,J+1)-2*C(L1,J)+C(L1,J-1))+
     . C(L4,J)*(C(L1,J+1)-C(L1,J-1))/(2*DELZ1))+R1*C(L1,J)*ALF*RATE_C
     END IF

C--------------------------------------------------------
 210 RETURN
     END

C
     SUBROUTINE DIFFEQ(C1,P1,NJ1,N1)
     IMPLICIT REAL*8(A-H,O-Z)
C***********************************************************************
C GENERALIZED CALLING PROGRAM FOR BAND(J) TO SOLVE DIFFERENTIAL
C EQUATIONS TAKING PARTIAL DERIVATIVES NUMERICALLY
C***********************************************************************
C
C    CALLED BY:  MAIN CALLING PROGRAM
C
C***********************************************************************
C
C    SUBROUTINES CALLED:
C         WRTOUT (FOR DATA OUTPUT)
C         BAND  (TO SOLVE BANDED TRIDIAGONAL COEFFICIENT MATRIX)
C         FUNCT  (TO OBTAIN VALUE FOR FUNCTION FOR A GIVEN VALUE
C              FOR A VARIABLE)
C
C***********************************************************************
C
C    LIST OF IMPORTANT VARIABLES:
C
C         A      A coefficient described in Newman, Appendix C
C         AA      first, AA is F(WORKC*CU). Later AA is the
C              value of the derivative df/dc used in Newton's
C              method
C         B      B coefficient described in Newman, Appendix C
C         C      variables to be solved for
C         CD      multiplication factor used in obtaining
C              numerical derivatives
C         COLD    value of C from previous iteration
C         CU      2.0 - CD
C         D      D coefficient described in Newman, Appendix C
C         ERR     convergence criterion
C         F      value of function f(C), calculated in FUNCT
C         G      residual of f(C) calculated with updated C value
C         I     index used for equation number
C         ITCNT   index for iteration number
C         ITPRT   flag used for determining whether intermediate
```

```
C               calculations are output; for ITPRT=0,only
C               converged results are output; for
C               ITPRT=1,results of each iteration are printed.
C          J     index for node number
C          K      index for equation number
C          M      index used in working through nodes used to
C               calculate numerical derivatives
C          MM     used to determine starting node (in relation to
C               J) for estimation of numerical derivatives
C          N     number of equations (no. of variables)
C          NJ    number of node points
C          SAVEC  saved value of C
C          SUM    intermediate value used in calculating G
C          TINIER  criterion used to avoid working with small numbers
C          TINY   criterion used to avoid working with small numbers
C          TNIEST  criterion used to avoid working with small numbers
C          WORKC   saved value of C; modified when C less than Tinier
C          X      X value described in Newman, Appendix C
C          Y      Y value described in Newman, Appendix C
C************************************************************************
C
C DIMENSIONS HAVE BEEN SET FOR 6 EQUATIONS AND 101 NODE POINTS; IF
C IT IS DESIRABLE TO INCREASE THE NUMBER OF EQUATIONS, CHANGE THE 6
C IN THE DIMENSIONS TO WHATEVER NUMBER YOU WANT, AND CHANGE THE SECOND
C DIMENSION OF D TO 2N+1, WHERE N IS THE NUMBER OF EQUATIONS.  IF MORE
C NODE POINTS ARE DESIRED, CHANGE THE 101 IN THE DIMENSION STATEMENTS TO
C WHATEVER YOO¿ppp
   REAL*8 C1(14,201),P1(14,201)
   COMMON/OLD/ AA(14),SUM(14),COLD(14,201)
   COMMON/BND/A(14,13),B(14,13),C(14,201),D(14,29),X(14,13),
  .Y(14,13),G(14),N,NJ,ITPRT,ITCNT,F(14),P(14,201)
   COMMON/IND1/IND(4)
   DATA TINY,TINIER,TNIEST,ERR/1.0D-10,1.0D-15,1.0D-20,1.0D-6/
   DATA CU,CD/1.0001,.9999/
C
C ITCNT INITIALIZED HERE, AND INITIAL VALUES FOR VARIABLES PRINTED OUT
C IF ITPRT=1.
C
   ITPRT=0
   ITCNT=0
   NJ=NJ1
   N=N1
   IF (ITPRT.GT.0) CALL WRTOUT2
   DO 5 I1=1,N1+1
    DO 5 I2=1,NJ1
     C(I1,I2)=C1(I1,I2)
     P(I1,I2)=P1(I1,I2)
 5  CONTINUE
C
C LOOP BEGUN FOR ITERATIONS
C
   DO 75 ITCNT=1,50
C
C COLD ARRAY SET UP
C
     DO 10 K=1,N
       DO 10 J=1,NJ
         COLD(K,J)=C(K,J)
10     CONTINUE
C
C LOOP BEGUN TO MARCH THROUGH EACH NODE POINT.  SUBSEQUENT CALCULATIONS
C PERFORMED AT EACH NODE POINT.  G VALUES ALSO SET AT -F(I).
C
     DO 60 J=1,NJ
```

```
      CALL FUNCT1(J)
      DO 15 I=1,N
        SUM(I)=0.0
        G(I)=-F(I)
15     CONTINUE
C
C THIS IS THE PLACE THE DERIVATIVES ARE CALCULATED.  MM IS SET
C TO INDICATE THE INTERVAL WHERE THE DERIVATIVES ARE TO BE CALCULATED
C (IN RELATION TO J).
C
      IF (J.EQ.1) THEN
        MM=0
      ELSE IF (J.LT.NJ) THEN
        MM=-1
      ELSE
        MM=-2
      END IF
      DO 50 M=MM,MM+2
C
C ORIGINAL VALUES OF C STORED IN SAVEC AND WORKC
C
        DO 50 K=1,N
          SAVEC=C(K,J+M)
          WORKC=SAVEC
C
C DERIVATIVES CALCULATED FOR SMALL VALUES OF C (LESS THAN TINY)
C
          IF (ABS(WORKC).LT.TINY) THEN
            IF(ABS(WORKC).LT.TINIER) WORKC=SIGN(TINIER,WORKC)
            C(K,J+M)=1.2*WORKC
            CALL FUNCT1(J)
            DO 20 I=1,N
              AA(I)=-F(I)
20           CONTINUE
            C(K,J+M)=1.1*WORKC
            CALL FUNCT1(J)
            DO 25 I=1,N
              AA(I)=AA(I)+4.0*F(I)
25           CONTINUE
            C(K,J+M)=WORKC
            CALL FUNCT1(J)
            DO 30 I=1,N
              AA(I)=(AA(I)-3.0*F(I))/(0.2*WORKC)
30           CONTINUE
C
C DERIVATIVES CALCULATED FOR LARGER VALUES OF C (GREATER THAN TINY)
C
          ELSE
            C(K,J+M)=WORKC*CU
            CALL FUNCT1(J)
            DO 35 I=1,N
              AA(I)=F(I)
35           CONTINUE
            C(K,J+M)=WORKC*CD
            CALL FUNCT1(J)
            DO 40 I=1,N
              AA(I)=(AA(I)-F(I))/((CU-CD)*WORKC)
40           CONTINUE
          ENDIF
          C(K,J+M)=SAVEC
C
C VALUES FOR A,B,D,X AND Y GIVEN.  SUM IS ALSO INCREMENTED
C
          DO 45 I=1,N
```

```
              SUM(I)=SUM(I)+AA(I)*C(K,J+M)
              IF (M.EQ.-2) Y(I,K)=AA(I)
              IF (M.EQ.-1) A(I,K)=AA(I)
              IF (M.EQ.0) B(I,K)=AA(I)
              IF (M.EQ.1) D(I,K)=AA(I)
              IF (M.EQ.2) X(I,K)=AA(I)
45            CONTINUE
50        CONTINUE
        DO 55 I=1,N
          G(I)=G(I)+SUM(I)
55        CONTINUE
C
C  BAND CALLED TO SOLVE THE BLOCK TRIDIAGONAL MATRIX
C
        CALL BAND2(J)
60      CONTINUE
C
C  CONVERGENCE CRITERION CHECKED.  IF ANY VALUE OF RELATIVE CONVERGENCE
C  FOR ANY VARIABLE IS GREATER THAN THE CONVERGENCE CRITERION, A NEW
C  ITERATION IS BEGUN
C
      DO 65 K=1,N
        DO 65 J=1,NJ
          IF(DABS(C(K,J)).GT.TNIEST) THEN
            IF(DABS((C(K,J)-COLD(K,J))/C(K,J)).GT.ERR) GO TO 70
          ENDIF
65      CONTINUE
    GO TO 80
70   IF(ITPRT .GT. 0) CALL WRTOUT2
75   CONTINUE
80   CONTINUE
    DO 95 I1=1,N1
      DO 95 I2=1,NJ1
        C1(I1,I2)=C(I1,I2)
95   CONTINUE
    RETURN
    END

    SUBROUTINE WRTOUT2
C************************************************************************
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/BND/A(14,13),B(14,13),C(14,201),D(14,29),X(14,13),
   .Y(14,13),G(14),N,NJ,ITPRT,ITCNT,F(14)
    IF (ITCNT.NE.0) WRITE (*,99)ITCNT
    WRITE (*,100)
    DO 1 K=1,NJ,2
      WRITE(*,101)K,(C(I,K),I=1,N)
1    CONTINUE
99   FORMAT(' ITCNT=',I2)
100  FORMAT('  J       C1            C2            C3             '
   &,' C4         C5           C6'/)
101   FORMAT(1X,I3,6(1PE16.8))
    RETURN
    END
C   BLOCK TRIDIAGONAL MATRIX SUBROUTINE
    SUBROUTINE BAND2(J)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION E(14,14,201)
    COMMON/BND/A(14,13),B(14,13),C(14,201),D(14,29),X(14,13),Y(14,13),
   .G(14),N,NJ,ITPRT,ITCNT,F(14)
  101  FORMAT('0DETERM=0 AT J=',I4)
    IF (J.EQ.1) THEN
      NP1=N+1
      DO 2 I=1,N
```

```
        D(I,2*N+1)=G(I)
        DO 2 L=1,N
         D(I,L+N)=X(I,L)
 2    CONTINUE
      CALL MATINV2(N,2*N+1,DETERM)
      IF(DETERM.EQ.0) WRITE (2,101) J
      DO 5 K=1,N
        E(K,NP1,1)=D(K,2*N+1)
        DO 5 L=1,N
          E(K,L,1)=-D(K,L)
          X(K,L)=-D(K,L+N)
 5    CONTINUE
      RETURN
    ELSE IF(J.EQ.2)THEN
      DO 7 I=1,N
        DO 7 K=1,N
          DO 7 L=1,N
           D(I,K)=D(I,K)+A(I,L)*X(L,K)
 7    CONTINUE
    ELSE IF(J.EQ.NJ) THEN
      DO 10 I=1,N
        DO 10 L=1,N
          G(I)=G(I)-Y(I,L)*E(L,NP1,J-2)
          DO 10 M=1,N
            A(I,L)=A(I,L)+Y(I,M)*E(M,L,J-2)
10    CONTINUE
    ENDIF
    DO 12 I=1,N
      D(I,NP1)=-G(I)
      DO 12 L=1,N
        D(I,NP1)=D(I,NP1)+A(I,L)*E(L,NP1,J-1)
        DO 12 K=1,N
          B(I,K)=B(I,K)+A(I,L)*E(L,K,J-1)
12  CONTINUE
    CALL MATINV2(N,NP1,DETERM)
    IF(DETERM.EQ.0) WRITE(2,101) J
    DO 15 K=1,N
      DO 15 M=1,NP1
        E(K,M,J)=-D(K,M)
15  CONTINUE
    IF(J.EQ.NJ)THEN
      DO 17 K=1,N
        C(K,J)=E(K,NP1,J)
17    CONTINUE
      DO 18 M=NJ-1,1,-1
        DO 18 K=1,N
          C(K,M)=E(K,NP1,M)
          DO 18 L=1,N
            C(K,M)=C(K,M)+E(K,L,M)*C(L,M+1)
18    CONTINUE
      DO 19 L=1,N
        DO 19 K=1,N
          C(K,1)=C(K,1)+X(K,L)*C(L,3)
19    CONTINUE
    ENDIF
    RETURN
    END
C
C   MATRIX INVERSION SUBROUTINE
    SUBROUTINE MATINV2(N,M,DETERM)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION ID(14)
    COMMON/BND/A(14,13),B(14,13),C(14,201),D(14,29)
    DETERM=1.0
```

```
  DO 1 I=1,N
1   ID(I)=0
  DO 18 NN=1,N
    BMAX=0.0
    DO 6 I=1,N
      IF(ID(I).EQ.0)THEN
        DO 5 J=1,N
          IF(ID(J).EQ.0)THEN
            IF(DABS(B(I,J)).GT.BMAX) THEN
              BMAX=DABS(B(I,J))
              IROW=I
              JCOL=J
            ENDIF
          ENDIF
5         CONTINUE
      ENDIF
6    CONTINUE
    IF(BMAX.EQ.0.0)THEN
      DETERM=0.0
      RETURN
    ENDIF
    ID(JCOL)=1
    IF(JCOL.NE.IROW) THEN
9       DO 10 J=1,N
          SAVE=B(IROW,J)
          B(IROW,J)=B(JCOL,J)
          B(JCOL,J)=SAVE
10       CONTINUE
      DO 11 K=1,M
          SAVE=D(IROW,K)
          D(IROW,K)=D(JCOL,K)
          D(JCOL,K)=SAVE
11       CONTINUE
    ENDIF
    FF = 1.0/B(JCOL,JCOL)
    DO 13 J=1,N
13      B(JCOL,J)=B(JCOL,J)*FF
    DO 14 K=1,M
14      D(JCOL,K)=D(JCOL,K)*FF
    DO 18 I=1,N
      IF(I.NE.JCOL)THEN
15        FF = B(I,JCOL)
        DO 16 J=1,N
16          B(I,J) = B(I,J) -FF*B(JCOL,J)
        DO 17 K=1,M
17          D(I,K) = D(I,K) -FF*D(JCOL,K)
      ENDIF
18 CONTINUE
   RETURN
   END
```