EGG-FSP--10315

DE92 018044

TMAP4 User's Manual

G. R. Longhurst
D. F. Holland
J. L. Jones
B. J. Merrill

June 12, 1992

Idaho National Engineering Laboratory
P. O. Box 1625
Idaho Falls, ID 83415-3523

Prepared for the
U. S. Department of Energy
Office of Energy Research
Under DOE Idaho Field Office
Contract DE-AC07-76ID10570

ABSTRACT

The Tritium Migration Analysis Progam, Version 4 (TMAP4) has been developed by the Fusion Safety Program at the Idaho National Engineering Laboratory (INEL) as a safety analysis code, mainly to analyze tritium retention and loss in fusion reactor structures and systems during normal operation and accident conditions. TMAP4 incorporates one-dimensional thermal- and mass-diffusive transport and trapping calculations through structures and zero dimensional fluid transport between enclosures and across the interface between enclosures and structures. A key feature is the ability to input problem definition parameters as constants, interpolation tables, or FORTRAN equations. The code is specifically intended for use under a DOS operating system on PC-type mini-computers, but it has also been run successfully on workstations and mainframe computer systems. Use of the equation-input feature requires access to a FORTRAN-77 compiler and a linker program.

CONTENTS

CONTENTS Continued

FIGURES

TMAP4 *USER'S MANUAL*

1.0 INTRODUCTION

TMAP (Tritium Migration Analysis Program) was developed by EG&G Idaho, Inc., to dynamically analyze dissolved gas movement through structures, between structures and adjoining enclosures, and among enclosures. Historically, it was initiated to assist in evaluation of tritium losses from fusion reactor systems during normal and accident conditions. However, it became evident that TMAP has application to a much wider variety of problems. TMAP incorporates a one-dimensional diffusion capability that determines the thermal response of structures, solves equations for solute atom movement through surfaces and in bulk materials, and also includes zero-dimensional equations for flows between and chemical reactions within defined control volumes. Now TMAP4 (TMAP Version 4) has been produced to correct deficiencies found in earlier versions, add some flexibilities that did not previously exist, and undergo rigorous quality certification to Quality Assurance Level A. It is written primarily for use with modern desk-top computers (PCs), but it has been used successfully on larger systems (work-stations and main-frame environments).

2.0 MODEL DEVELOPMENT

TMAP calculates the time-dependent responses of a system of solid structures (may be a layered composite), and a related system of gas-filled volumes or enclosures, with respect to:

1. Movement of solute species across structure surfaces. This movement may be governed by dissociation/recombination or by a solution law, such as Sieverts' or Henry's laws, or inhibited from crossing the surface

2. Movement by Fick's-law diffusion in the bulk of a structure with optional trapping within a structure

3. Thermal response of structures to applied heat or boundary temperature loadings conditions

4. Chemical reactions within enclosures

5. Convective flow between enclosures.

These capabilities are discussed in detail in the following sections. The code was developed to address hydrogen isotopic species in metals and surrounding gas-filled or evacuated volumes. For that reason, the following discussion specifically addresses that class of problems, but it may be generalized to other solute species and solid-material environments.

2.1 Movement Across Surfaces

Gas movement across metal surfaces is often governed by molecular dissociation to solid solution and the reverse process of molecular recombination to the gas phase at the surface of the metal. For many situations with metals, this process is sufficiently close to equilibrium that a solution law such as Sieverts' law applies. For non-metals, solution usually follows Henry's law, and it is molecules that are absorbed. Adsorption and release of molecules from a surface not in equilibrium with the surrounding gas may be rate-limited by chemisorption which is characterized by an activation energy. TMAP4 has the capability of simulating these processes.

When dissociation and recombination govern, the net flow of atoms of species "*s*" into the surface is given by

$$J_s = \sum_{m=1}^{k} a_{m_s} J_m \tag{1}$$

$$J_m = K_{d_m} P_m - \sum_{i,j} K_{r_{ij}} C_i C_j \tag{2}$$

where

J_s = atom flux of atomic species "*s*" into the surface

a_{m_s} = number of atoms of species "*s*" in molecule of species "*m*"

J_m = molecular flux of molecular species "*m*" into the surface

K_{d_m} = dissociation coefficient for molecular species "*m*" at the surface

P_m = partial pressure of molecular species "*m*"

$K_{r_{ij}}$ = recombination coefficient for molecular species "*m*" which consists of atomic species "*i*" and "*j*" (e.g., $s_i + s_j \rightarrow m_{ij}$)

C_i = surface concentration of atomic species "*i*"

C_j = surface concentration of atomic species "*j*"

Note in Eq.(2) that where two different atoms comprise the molecule, it is necessary to count species "*i*" combining with species "*j*" as well as the reverse even though the recombination coefficient will be the same for both combinations.

For cases where movement through the surface approaches equilibrium, surface concentrations are determined by appropriate solubility relationships. For metals, Sieverts' law relates the equilibrium concentration of solute atoms in the metal to the hydrogen partial pressure above the surface of the metal. If the molecule is homonuclear (two atoms of the same isotope), and only one molecular species containing the atomic species "*s*" is present, then that relationship is

$$C_s = K_s \, P_m^{1/2} \tag{3}$$

where

C_s = concentration of mobile atomic species "*s*" at the surface

K_s = Sieverts' law solubility coefficient

and P_m is as defined previously. When solutes in surfaces approach equilibrium with the gas mixture above the surface, the net molecular flow to or from the surface is small relative to the individual dissociation and recombination terms in Eq.(2) For such a case, it may be shown from Eqs.(2) and (3) that Sieverts' solubility constant and the surface dissociation and recombination coefficients are related by

$$K_{d_m} = K_{r_{ii}} \, K_s^2 \tag{4}$$

This relationship is frequently assumed to hold even when conditions are moderately far from equilibrium at the surface.

It is inconsistent to think of a single gas of heteronuclear (different-isotope) molecules in equilibrium with concentrations of the different atomic species, C_i and C_j, at the surface. In addition to the heteronuclear recombination of the two atomic species, the two homogeneous recombination reactions will also take place. For example, if the heterogeneous molecule reaction at the surface was

$$HT(g) \rightleftharpoons H(s) + T(s) \tag{5}$$

then the reactions

$$2T(s) \rightleftharpoons T_2(g) \tag{6}$$

and

$$2H(s) \rightleftharpoons H_2(g) \tag{7}$$

must also be taking place. Those reactions cannot be in equilibrium unless the homonuclear molecular species (e.g., T_2 and H_2) exist in the gas over the surface in the correct proportions for the atomic species concentrations (T and H) in the surface. Conversely, for two or more different homonuclear molecular gases to be in equilibrium with a surface it must be assumed that the correct partial pressures of the heteronuclear products of the various atomic species are also present. For that reason, Sieverts' law should only be invoked when there is a single, homonuclear molecular gas over a surface. However, as a computational artifice, the dissociation/recombination law may be invoked with very high rate constants to achieve the effect of a solution law at the surface. For diatomic gas molecules composed of different atomic species "i" and "j" (heteronuclear), these coefficients must be related by

$$\frac{K_{d_{ij}}}{K_{r_{ij}}} = \frac{2K_{s_{ij}}^{\;2}}{\sqrt{K_{s_{ii}}K_{s_{jj}}}} \tag{8}$$

where it has been assumed that

$$C_i = K_{s_{ij}} P_{m_{ij}} \tag{9}$$

The movement of gas molecules across the surface of nonmetals usually does not depend on dissociation and recombination. As a rule, molecular species absorbed in a nonmetal structure are the same species as exist in the gas over the surface of the structure. When movement of gas molecules into and out from the surface is in equilibrium, Henry's law applies:

$$C_m = K_h P_m \tag{10}$$

where

C_m = concentration of the mobile molecular species "*m*" at the surface

K_h = Henry's law solubility coefficient.

If the surface processes are not in equilibrium, then it may be shown that the molecular flux to the surface is given by

$$J_m = J_{m_{in}} - J_{m_{out}} = \nu_o K_{h_o} \exp\left(-\frac{E_c}{kT}\right) P_m - C_m \nu_o \exp\left(\frac{E_h - E_c}{kT}\right) \tag{11}$$

where

ν_o = Debye frequency ($\approx 10^{13}$ s^{-1})

K_{h_o} = Henry's law solubility pre-exponential factor

E_c = surface activation energy (see Figure 1)

k = Boltzmann's constant

T = temperature of the gas and the surface (assumed equal).

E_h = Henry's law enthalpy of solution.

The potentials involved in this process are diagramed in Figure 1. Two different assumptions on the relationship of the various energies are shown there.

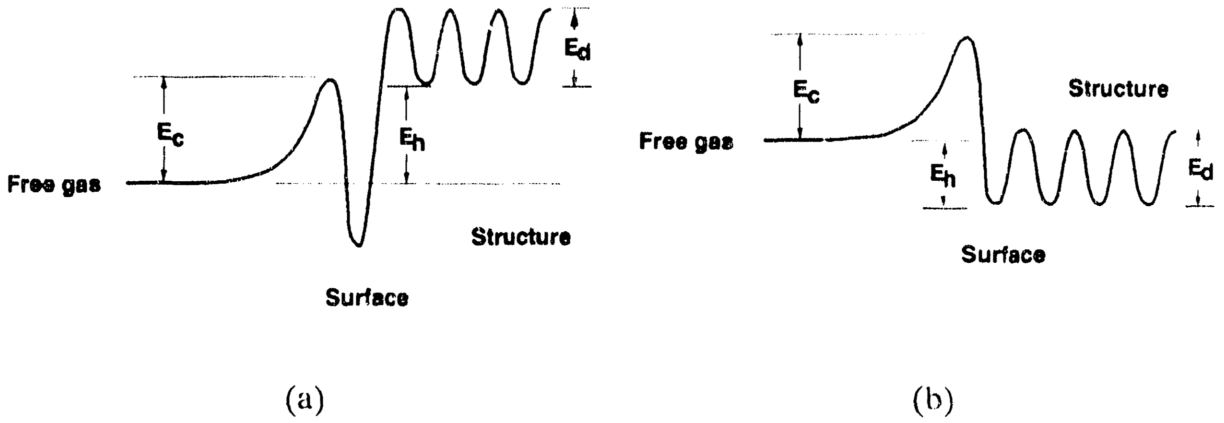


Figure 1. Two different assumptions on surface potentials for Henry's law, each of which involves a surface binding energy.

2.2 Movement Through Structures

Movement of solute gas atoms in a structure is governed by the conservation of atoms of species "*s*", described by

$$\frac{\partial C_s}{\partial t} + \nabla \cdot J_s = S_s - \frac{\partial C_s^t}{\partial t} \tag{12}$$

where

C_s = concentration of atomic species "*s*" in the structure

J_s = diffusive flux of species "*s*" atoms

S_s = local source or production rate of species "*s*" atoms

C_s^t = concentration of trapped atoms of species "*s*".

The most general statement regarding flows in thermodynamic systems is[1]
where

$$J_k = \sum_j L_{jk} F_j \tag{13}$$

J_k = a generalized flow involving some extensive thermodynamic variable

L_{jk} = a generalized kinetic coefficient

F_j = a generalized force, usually the gradient of an intensive thermodynamic variable.

When the intensive property is moles of dissolved gas, the conjugate ("*j*" = "*k*" in Eq.(13)) force is the concentration gradient, the kinetic coefficient is the negative of the diffusivity, and the extensive flow is the diffusive flux. However, since gradients in other intensive properties are also likely to exist, there will be more terms than one in the sum. TMAP4 considers temperature gradients in addition to the concentration gradients since kinetic coefficients for most other forces (e.g., gradients in electric potential and magnetization) generally will be small. Thus, the flux of dissolved gas atoms is given by

$$J_s = -D_s \left(\nabla C_s + \frac{Q_s^* C_s}{RT^2} \nabla T \right) \tag{14}$$

where

D_s = diffusivity of atomic species "*s*" in the structure material

Q_s^* = heat of transport or Ludwig-Soret coefficient for atomic species "*s*"

R = universal gas constant

T = local temperature

The inventory of solute gas atoms (or molecules) can be affected by trapping sites where the energy required for an atom to move to another nearby site is greater than that required for ordinary diffusion. Trapping sites can result from impurities and structural irregularities arising from cold work, precipitation of alternate phases, or neutron displacement damage.

Considering a solute particle (atom or molecule) concentration, C_j, the governing equations are

$$\frac{\partial C_j^t}{\partial t} = \frac{\alpha_{t_j} C_t^e}{N} C_j - \alpha_{r_j} C_j^t \tag{15}$$

$$C_t^e = C_t^o - \sum_j C_j^t \tag{16}$$

$$\alpha_{t_j} = \frac{D_j}{\lambda^2} \tag{17}$$

$$\alpha_{r_j} = \nu_o \exp\left(-\frac{E_t}{RT}\right) \tag{18}$$

where

C_j^t = concentration of trapped particles of species "*j*"

C_t^o = concentration of trapping sites

C_t^e = concentration of empty trapping sites

D_j = diffusivity of species "*j*" in the material

λ = jump distance (usually the lattice constant)

ν_o = release attempt frequency, usually the Debye frequency ($\approx 10^{13}$ s^{-1})

E_t = trapping energy (diffusion activation plus binding energies)

N = atomic number density of the host material

α_{t_j} = trapping rate coefficient for species "*j*"

α_{r_j} = trap release rate coefficient for species "*j*"

In addition to diffusion and trapping as defined by the above equations, TMAP4 features the ability to *link* or connect structures together (composite materials) such that solute atoms coming from one structure segment go directly into the connecting structure segment. It is essential that when two structure segments are linked together, the diffusing species in one are the same as the diffusing species in the other. The chemical potentials of the diffusing species in the structures are characterized by the solubilities, either Sieverts' law or Henry's law, depending on the materials. TMAP assumes equilibrium between the chemical potentials of the diffusing species. Hence, for linked segments "*A*" and "*B*" where Sieverts' law applies

$$C_s^B = \frac{K_s^B}{K_s^A} C_s^A \tag{19}$$

where K_s^A , K_s^B are the Sieverts' law solubilities in materials of segments "*A*" and "*B*", respectively. A similar equation applies using the corresponding Henry's law solubilities when that law is applicable.

2.3 Enclosure Dynamics

Partial pressures of gases in volumes or *enclosures*, as they will be referred to in TMAP4, are determined by species conservation equations that account for convective flow between enclosures, diffusion into enclosures from structures, and conversion between molecular species by chemical reactions. The basic conservation equation for enclosure molecular species having concentration C_m is

$$\frac{\partial C_m}{\partial t} + \nabla \cdot \left(C_m u_m + J_m \right) = S_{chem\ m} \tag{20}$$

where

$\boldsymbol{u}_m$ = velocity of molecular species "*m*"

$\boldsymbol{J}_m$ = diffusive flux of molecular species "*m*"

S_{chem_m} = volumetric source of molecular species "*m*" due to chemical reactions.

Integrating this equation over the volume of the "*ith*" enclosure by assuming a uniform concentration and applying Gauss' theorem to the convective terms leads to

$$V_i \frac{\partial C_{m_i}}{\partial t} = -C_{m_i} \sum_j Q_{ij} + \sum_j C_{m_j} Q_{ji} - \sum_k A_k J_{m_k} + V_i S_{chem_m} \qquad (21)$$

where

V_i = volume of enclosure "*i*"

C_{m_i} = concentration of molecular species "*m*" in enclosure "*i*"

j = index over all enclosures

Q_{ij} = volumetric flow rate of gas from enclosure "*i*" to enclosure(s) "*j*"

A_k = surface area of structure "*k*" providing part of the boundary for enclosure "*i*"

J_{m_k} = net flux of molecular species "*m*" going into wall structure "*k*"

Note that in metals the molecular species "*m*" going into a surface is different from the atomic species diffusing in the material. The partial pressure of molecular gas "*m*" in the enclosure is determined from this time-dependent species concentration and the ideal gas law.

The chemical reaction calculations of TMAP4 require the user to define specific chemical reaction equations and to provide forward reaction rate expressions for these reactions. An example is the oxidation of tritium

$$T_2 + \frac{1}{2} O_2 = T_2O \qquad (22)$$

Often a chemical reaction is assumed to proceed in the forward direction at a rate that is proportional to the gradient in chemical potential of the system with respect to molar fractions of the reactants and products. Hence, a rate expression may involve a constant, the concentrations of the reactants and products, and the partial molar Gibbs free energies of the various species. However, for the case of oxidation of tritium cited above, the inherent radioactive decay of the tritium catalyzes the reaction, so the reaction rate is proportional to the concentration of tritium present in all forms. We would write for the oxidation of T_2 in a system involving only H and T isotopes of hydrogen, for example,

$$R_c = K_c C_{T_2}\left(2C_{T_2} + C_{HT} + C_{HTO} + 2C_{T_2O}\right) \tag{23}$$

where K_c is the chemical conversion rate constant (m^3/molecule ·s).

The chemical source term, S_{chem_m}, in Eq.(21) is the sum over all such reactions of the product of the reaction rate and the stoichiometric coefficients for the production of that species. Specifically,

$$S_{chem_m} = \sum_j \left(-a_{m_j} + b_{m_j}\right) R_{c_j} \tag{24}$$

where

a_{m_j} = the reactant- and product-stoichiometric coefficients for molecular species "m"

b_{m_j} in reaction "j"

R_{c_j} = the forward reaction rate.

Suppose, for example, an enclosure has T_2, HT, H_2, and O_2 that react to form the additional molecular species T_2O, H_2O, and HTO. The reactions that could contribute to the HTO formation rate are

$$HT + \frac{1}{2}O_2 \rightleftharpoons HTO \tag{25}$$

$$H_2O + T_2O \rightleftharpoons 2HTO \tag{26}$$

$$T_2 + H_2O \rightleftharpoons HTO + HT \tag{27}$$

$$H_2 + T_2O \rightleftharpoons HTO + HT \tag{28}$$

$$HT + H_2O \rightleftharpoons HTO + H_2 \tag{29}$$

and

$$HT + T_2O \rightleftharpoons HTO + T_2 \tag{30}$$

Each of these has an associated rate constant, and, as written, each has an HTO product stoichiometric coefficient, b_{m_j}, of one or two, but the reactant stoichiometric coefficients are zero. All these reactions need to be included in the sum of Eq.(24).

2.4 Structural Thermal Response

The processes that govern the permeation of solute gas atoms in structures are highly temperature dependent. Therefore, TMAP4 was developed with the capability of solving the time-dependent heat conduction equation to determine the thermal response of a structure. This equation is

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) + S_h \tag{31}$$

where

ρ = material mass density

C_p = material specific heat

T = temperature

κ = material thermal conductivity

S_h = local volumetric heating rate.

This equation is solved in conjunction with the boundary conditions of either surface convection, an imposed surface heat flux, a specified temperature, or an adiabatic surface.

The thermal response calculations are extended for a linked-segment (composite material) structure by the use of a gap or surface-contact conductance equation at the interface. The interface equation for heat flow across the interface between the two structures is

$$q = h_{gap}\left(T_{s_2} - T_{s_1}\right) \tag{32}$$

where

q = interface heat flux

h_{gap} = user defined gap thermal conductance

T_{s_2} = surface temperature of the segment more positive in the distance coordinate

T_{s_1} = surface temperature of the segment more negative in the distance coordinate.

Here, segment "2" is positioned further in the positive "x" direction than in segment "1".

3.0 DIFFERENCE EQUATIONS

The sections in this chapter describe the development of finite difference equations solved by TMAP4. These are based on the governing equations for the various processes described in Chapter 2. The nomenclature used is the same as used previously and is summarized as follows.

A_k = surface area of the "*kth*" diffusion segment

C_m = gas molecular concentration in an enclosure

C_s = solute gas atom concentration in a diffusion segment

C_t^e = concentration of empty traps

C_t^o = total trap concentration

D = diffusion coefficient

h = convective heat transfer coefficient

i = index for spatial location

J_m = molecular species flux

J_s = atomic species flux

K_d = molecular dissociation coefficient

K_h = Henry's law solubility coefficient

K_r = molecular recombination coefficient

K_s = Sieverts' law solubility coefficient

n, o = superscripts denoting new or old information with respect to the current time step

P_m = partial pressure of molecular gas species

Q = volumetric flow rate

Q^* = heat of transport or Ludwig-Soret coefficient

R = universal gas constant

R_c = forward reaction rate constant for chemical reaction

S_h = source term for heat addition

$S_{m,s}$ = source term for molecular or atomic species addition

t = time

T = temperature

V = volume of enclosure

x = spatial distance

α_r = trap release rate coefficient

α_t = trapping rate coefficient

Δ = incremental change (Δt = time step, Δx = node thickness)

κ = thermal conductivity

Section 3.1 discusses the derived difference equations for movement of solute gas atoms through a diffusion structure. Section 3.2 gives the equations for calculating the thermal response of the structure. Section 3.3 presents the equations of change for the dynamic response of enclosures. In each section, the solution scheme employed to solve the various derived equations is also discussed.

3.1 Atom Movement Through Materials

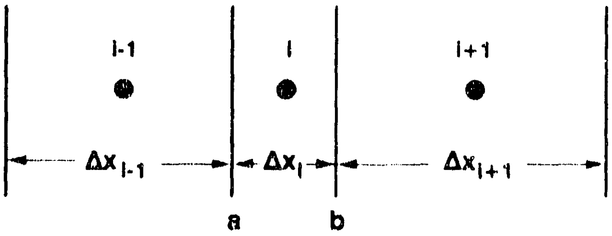


Figure 2. Representation of general finite difference model used in TMAP4.

Movement of dissolved gas atoms in solids is governed by the conservation law of Eq.(12), the diffusion flux, Eq.(14), and the action of trapping described by Eqs.(15) to (18). These are reduced to one spatial dimension and descretized for TMAP4. Boundary conditions for this movement involve the solubility of the atoms in the material or the various rate constants for dissociation and recombination. For reference, a representation of the nodal system used for gas atom transport calculations is given in Figure 2. Note that the spacing between the nodes is arbitrary.

3.1.1 Diffusion and Trapping

TMAP4 uses space-centered, implicit (time-backward) difference approximations for derivatives. The atom flux of Eq.(14) at location "b" of Figure 2 is thus given by

$$J_s|_b = -\left(\frac{D_{i+1} + D_i}{\Delta x_{i+1} + \Delta x_i}\right)\left[C_{s_{i+1}}^n - C_{s_i}^n + \frac{\left(Q_{i+1}^* + Q_i^*\right)}{R\left(T_{i+1} + T_i\right)^2}\left(T_{i+1} - T_i\right)\left(C_{s_{i+1}}^n + C_{s_i}^n\right)\right] \quad . \tag{33}$$

A similar expression may also be written at location "a". These expressions are used to approximate the divergence of atomic species flux at location "i" as

$$\nabla \cdot J_s = \frac{J_s|_b - J_s|_a}{\Delta x_i} \quad . \tag{34}$$

Trapping, if any, is described by Eqs.(15) to (18) The finite difference approximation to Eq.(15) is

$$\frac{C_{s_i}^{t\,n} - C_{s_i}^{t\,o}}{\Delta t} = \frac{\alpha_t}{N} C_{t_i}^{e\,n} C_{s_i}^n - \alpha_r C_{s_i}^{t\,n} \quad . \tag{35}$$

Solving for $C_{s_i}^{t\,n}$ results in

$$C_{s_i}^{t\,n} = C_{s_i}^{t\,o} + \frac{\frac{\alpha_t}{N} C_t^{e\,n} C_{s_i}^n \Delta t}{1 + \alpha_r \Delta t} \quad . \tag{36}$$

Combining Eqs.(32) through (35) with the conservation law of Eq.(12) yields the general tri-diagonal equation that is solved at each time step for each internal (non-surface) node in the diffusion structure segment. At node "i", this equation is

$$\begin{aligned}
&-\left\{\left(\frac{D_{i+1} + D_i}{\Delta x_{i+1} + \Delta x_i}\right)\times\left[1 + \frac{Q_{i+1}^{*} + Q_i^{*}}{R(T_{i+1} + T_i)^2}\left(T_{i+1} - T_i\right)\right]\right\} C_{s_{i+1}}^{n} \\
&+\left\{\frac{\Delta x_i}{\Delta t} + \left(\frac{D_{i+1} + D_i}{\Delta x_{i+1} + \Delta x_i}\right)\times\left[1 - \frac{Q_{i+1}^{*} + Q_i^{*}}{R(T_{i+1} + T_i)^2}\left(T_{i+1} - T_i\right)\right] + \left(\frac{D_i + D_{i-1}}{\Delta x_i + \Delta x_{i-1}}\right)\times\right. \\
&\qquad\left.\left[1 + \frac{Q_i^{*} + Q_{i-1}^{*}}{R(T_i + T_{i-1})^2}\left(T_i - T_{i-1}\right)\right] + \frac{\alpha_t\, C_{t_i}^{e\,n}}{N(1 + \alpha_r \Delta t)}\right\} C_{s_i}^{n} \\
&-\left\{\left(\frac{D_i + D_{i-1}}{\Delta x_i + \Delta x_{i-1}}\right)\times\left[1 - \frac{Q_i^{*} + Q_{i-1}^{*}}{R(T_i + T_{i-1})^2}\left(T_i - T_{i-1}\right)\right]\right\} C_{s_{i-1}}^{n} \\
&\qquad = \frac{C_{s_i}^{\prime}\,\Delta x_i}{\Delta t} + S_{s_i} + \frac{\alpha_r\, C_{s_i}^{t\,o}\,\Delta x_i}{1 + \alpha_r \Delta t}
\end{aligned} \tag{37}$$

It is evident here that the tri-diagonal quantities, $C_{s_{i+1}}^{n}$, $C_{s_i}^{n}$, and $C_{s_{i-1}}^{n}$ are the only unknowns when trapping is neglected, but when trapping is included, an additional unknown is introduced. That additional unknown is the concentration of empty traps, $C_{t_i}^{e\,n}$. TMAP4 uses an implicit Newton-Rhapson iteration technique to determine this quantity. By forming the finite difference of the time derivative of Eq.(16) and combining that with Eq.(34), the resulting expression for the next iterate of empty trap concentration becomes

$$C_{t_i}^{e\ n} = C_{t_i}^{e\ o} - \Delta t \left(\frac{\alpha_t}{N} C_{t_i}^{e\ n} \sum_j C_{j_i}^{n} - \alpha_r \sum_j C_{j_i}^{t\ n} \right) \qquad (38)$$

where for this equation

j = all mobile species that compete for trap sites (TMAP4 assumes all species present compete).

Iteration on these equations continues until convergence is achieved to within the user specified tolerance.

3.1.2 Surface Boundary Conditions

Solute gas atom or molecule movement at a surface is described by one of four different boundary conditions: particle-flux-rate-dependent, solution-law-uependent, specified-surface-concentration, or non-flow. We now consider these in detail.

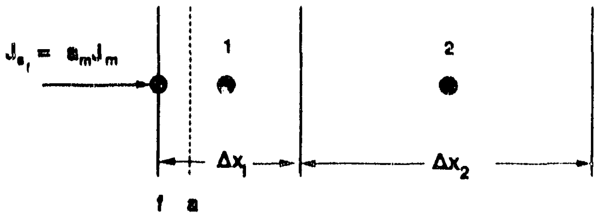


Figure 3. Notation for surface flux discretization.

The particle-flux-rate-dependent boundary condition, referred to here as ***ratedep***, is accommodated in the discretization by constructing a surface node of zero thickness. This arrangement is illustrated in Figure 3 where we denote the surface node by the subscript "f". A surface flux, J_{s_f} enters the surface of the structure at the surface node. The surface node has no associated volume, so the requirement for conservation is that the particle flux into the surface must be exactly balanced by diffusive transport of the gas atoms away from the surface. In equation form this relation is

$$J_{s_f} = -D \left(\nabla C_s + \frac{C_s Q_s^*}{R T^2} \nabla T \right) \qquad (39)$$

where the right side is to be evaluated at location "a" where the best values of the driving potential gradients are obtained. This equation, written in tridiagonal form, is

$$\begin{aligned} & C_{s_f}^{n}\left[\frac{Q_{s_f}^{*}+Q_{s_1}^{*}}{R\left(T_f+T_1\right)^2}\left(T_1-T_f\right)-1\right]\left(D_{s_f}+D_{s_1}\right) \\ & \qquad +C_{s_1}^{n1}\left[\frac{Q_{s_f}^{*}+Q_{s_1}^{*}}{R\left(T_f+T_1\right)^2}\left(T_1-T_f\right)+1\right]\left(D_{s_f}+D_{s_1}\right)=J_{s_f}\Delta x_1 \end{aligned} \tag{40}$$

and defines the surface concentration, $C_{s_f}^{n}$. Now for the ***ratedep*** boundary condition,

$$J_{s_f}=\sum_{m} K_{d_m} P_m-\sum_{i} \sum_{j} K_{r_{i,j}} C_{i_f}^{n} C_{j_f}^{n} \tag{41}$$

This is amenable to solution as part of the tridiagonal matrix system. The non-linearity introduced by the product of the surface concentrations in Eq.(41) is handled by allowing one concentration to be a variable of iteration for the Newton-Rhapson iteration scheme. The other one is solved for from the tri-diagonal matrix system between iterations, so it converges as the overall solution converges.

The solution-law-dependent boundary condition, called ***lawdep*** here, relates solute gas atom or molecule concentrations in surfaces to corresponding molecular gas pressures in the adjacent enclosure using Sieverts' law, Eq.(3), Henry's law, Eq.(10), or some similar one specified by the user. Since these concentrations are assumed to be known from conditions external to the diffusion problem (e.g., enclosure partial pressures), the terms involving surface-node concentrations (C_{s_i} - C_{s_f}) in the tridiagonal system of Eq.(37) are moved to the right-hand side for solution as boundary conditions. Because enclosure concentrations ar also being iteratively updated to convergence (see Section 3.3), the effect is to simultaneously converge surface concentrations and enclosure species partial pressures when either *ratedep* or *lawdep* conditions hold.

Specified surface concentration, called ***sconc*** here, assumes that the user has specified the concentration in the surface nodes, possibly as a function of time. This may be done by specifying an equation or table to define that concentration. The same approach is followed for computing as for the *lawdep* condition, but here the **sconc** values really do remain constant during the iterative convergence.

For the ***nonflow*** boundary condition, the concentration at the surface node is made equal to that at the adjacent node, thus forcing the concentration gradient to be zero at the surface. This creates a plane of symmetry at that surface.

3.1.3 Composite Surface Treatment

The adopted composite structure model illustrated in Figure 4 is an approximation to the condition that the chemical potentials of a diffusing species crossing the interface between two different materials will be equal. That chemical potential is effectively the ratio of concentration to the solubility of that species in the surface. That is, whether the diffusing species is either atomic or molecular, the ratio of the concentrations of that species on either side of the interface between to segments of the diffusion structure is the same as the corresponding ratio of solubilities. For Sieverts' law solution at the interface of materials "*A*" and "*B*" Eq.(19) relates the solute concentrations on either side of the interface, and a corresponding relationship holds if Henry's law applies.

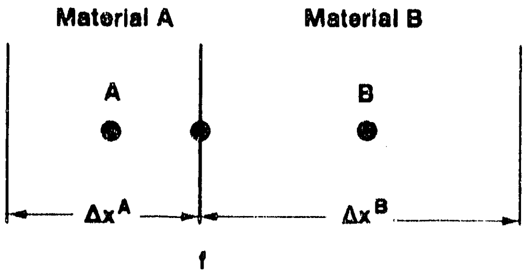


Figure 4. Notation for modelling flux across the interface of two dissimilar materials.

The transport of solute atoms from one material to another at an interface may be written in a form that will fit into a tridiagonal equation. This may be visualized with the aid of Figure 4. The flux, as given by Eq.(14) for the interface location, "f" in the figure, must be the same if written for one material or the other. That is, transport of the solute species across the interface must be continuous. Atoms or molecules passing through the interface must be conserved. The discretized equation that describes that continuity is

$$
\begin{aligned}
& C_s^A \left\{ \frac{D_s^A}{\Delta x^A} \left[1 - \frac{Q_{s_f}^{\bullet A} + Q_s^{\bullet A}}{R\left(T_f^A + T^A\right)^2} \left(T_A - T_f^A\right) \right] \right\} \\
& + C_{s_f}^A \left\{ - \frac{D_s^A}{\Delta x^A} \left[1 + \frac{Q_{s_f}^{\bullet A} + Q_s^{\bullet A}}{R\left(T_f^A + T^A\right)^2} \left(T_f^A - T^A\right) \right] + \frac{D_s^B K_s^B}{K_s^A \Delta x_B} \left[1 - \frac{Q_{s_f}^{\bullet B} + Q_s^{\bullet B}}{R\left(T_f^{B^2} + T^B\right)^2} \left(T^B - T_f^B\right) \right] \right\} \\
& + C_{s_f}^B \left\{ - \frac{D_s^B}{\Delta x_B} \left[1 + \frac{Q_{s_f}^{\bullet B} + Q_s^{\bullet B}}{R\left(T_s^B + T^B\right)^2} \left(T^B - T_f^B\right) \right] \right\} = 0
\end{aligned}
\tag{42}
$$

where the subscript "f" refers to values at the interface, and the superscripts "A" and "B" refer to the two different materials. T^A and T^B refer to the first node away from the interface. Note that the temperatures "T_f^A" and "T_f^B" on the two sides of the interface may be different, as will be explained in Section 3.2.3. Equation (42) is tridiagonal and is incorporated into the overall tridiagonal matrix for solution.

3.2 Thermal Response of Materials

The thermal response of a structure is governed by Eq.(31). The discretization of that equation and the corresponding boundary condition definitions are now presented.

3.2.1 Structure Bulk Temperature

With the aid of Figure 2 in Section 3.1, the finite difference form of Eq.(31) after collection of terms involving temperature becomes

$$
\begin{aligned}
&-\left(\frac{\kappa_{i+1}+\kappa_i}{\Delta x_{i+1}+\Delta x_i}\right)T^n_{i+1} + \left[\rho C_p\frac{\Delta x_i}{\Delta t} + \left(\frac{\kappa_{i+1}+\kappa_i}{\Delta x_{i+1}+\Delta x_i}\right) + \left(\frac{\kappa_i+\kappa_{i-1}}{\Delta x_i+\Delta x_{i-1}}\right)\right]T^n_i \\
&\qquad -\left(\frac{\kappa_i+\kappa_{i-1}}{\Delta x_i+\Delta x_{i-1}}\right)T^n_{i-1} = S_{h_i}\Delta x_i + \rho C_p\frac{\Delta x_i}{\Delta T}T^o_i
\end{aligned}
\qquad (43)
$$

This system, also tridiagonal, is solved once per time step, just prior to converging species concentrations.

3.2.2 Surface Boundary Conditions

TMAP4 provides four types of thermal boundary conditions: ***adiabatic, convective, surface-temperature,*** and ***surface-heat-flux***. Each condition incorporates a different equation into the thermal tridiagonal matrix equation set. For the adiabatic (***adiab***) boundary condition, the temperature at the surface of the structural segment is made equal to the temperature at the first node inside the surface. That forces the thermal gradient to zero, thereby preventing heat flux.

The convective (***convec***) boundary condition equates the conductive flux at a surface with the convective heat transfer coefficient, h, and a temperature difference, $T_f - T_e$, where T_f is the surface temperature and T_e is the adjacent enclosure temperature. Referring to Figure 3, section 3.1.2, the finite difference equation for the left boundary is then

$$\left[h + \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1}\right)\right] T_f'' - \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1}\right) T_1'' = h\,T_e \tag{44}$$

with a similar expression for the right boundary.

The user has the option of a set-temperature (***stemp***) boundary condition in which the temperature of the surface node of the segment is specified. The temperature specification may be in the form of a constant, an equation, or a look-up table. Those temperature variables are simply shifted to the right side of the tridiagonal as boundary conditions for the matrix inversion.

For the specified-heat-flux or ***sflux*** boundary condition, an expression is provided by the user for the heat flux, q_h, at the surface of the segment. This expression may be a constant, an equation, or a time-dependent look-up table. The equ..tion to be solved for the left-hand boundary is

$$\left(\frac{\kappa_1 + \kappa_f}{\Delta x_1}\right) T_f'' - \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1}\right) T_1'' = q_\epsilon \tag{45}$$

and a corresponding equation may be written for the structure right-hand surface.

3.2.3 Composite Structure Treatment

When two elements of a composite structure are in thermal contact, there is frequently an insulating layer, such as an oxide layer, that inhibits heat transfer, which in metals is often dominated by electron conductivity.[2] Also, there may be an actual gap between the segments for which it is possible to specify a thermal conductivity. In either of these cases, the temperature at one side of the thermal interface will not necessarily be the same as the temperature at the other side of that interface. However, of necessity the heat flux at the composite interface must be conserved. The expression for that continuity is

$$-\kappa \frac{\partial T}{\partial x}\Big|_{f^A} = -\kappa \frac{\partial T}{\partial x}\Big|_{f^B} = h_{gap}\left(T_f^A - T_f^B\right) \tag{46}$$

The relationship is illustrated in Figure 4. Again, the subscript "*f*" refers to values at the surface of the structural segment. The superscripts "*A*" and "*B*" refer to segments of the different materials on the two sides of the gap. The resulting equation in tri-diagonal form for surface "*A*" is

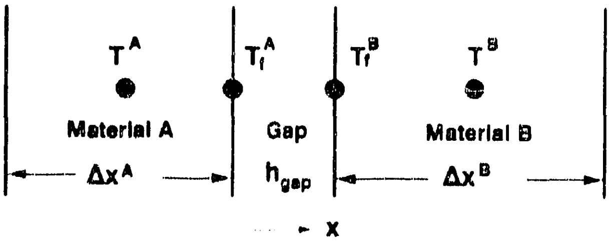


Figure 5. Relationship of quantities for heat flux at the interface of a composite material.

$$-\left(\frac{\kappa_f^A + \kappa^A}{\Delta x^A}\right)T^A + \left[\left(\frac{\kappa_s^A + \kappa^A}{\Delta x^A}\right) + h_{gap}\right]T_f^A - h_{gap}T_f^B = 0 \tag{47}$$

and there is a corresponding relation for the "*B*" surface.

3.3 Enclosure Response

TMAP4 allows two types of user-defined enclosures, "*functional*" and "*boundary*" types. Enclosure dynamics for *functional* type enclosures are governed by Eq.(21). If we consider the "*ith*" enclosure and apply the finite differencing technique to Eq.(21), there results

$$V_i \frac{C^n_{m_i} - C^o_{m_i}}{\Delta t} = \sum_j C^n_{m_j} Q_{ji} - C^n_{m_i} \sum_j Q_{ij} - \sum_k A_k J_{m_k} + V_i S_{chem_m} \quad . \tag{48}$$

This says that the rate of increase of gas molecular species "*m*" in enclosure "*i*" is the sum of flows of that species into that enclosure from other connected enclosures, less the flow of that species from the "*ith*" enclosure to the other enclosures, less that flowing into the adjacent "*k*" surfaces, plus that generated in the enclosure volume by chemical reactions. Per Eq.(24) that source term includes all pertinent reactions.

As with other state variable equations, this equation is solved using the Newton-Rhapson iteration technique. That is

$$C^{\nu+1}_{m_i} = C^{\nu}_{m_i} - \omega \frac{F}{\dfrac{\partial F}{\partial C_m}} \tag{49}$$

where ω is a relaxation parameter to optimize the speed of convergence. For this application, the function "*F*" is defined as

$$F = V_i \left(\frac{C^{\nu}_{m_i} - C^o_{m_i}}{\Delta t} \right) - \sum_j C^{\nu}_{m_j} Q_{ji} + C^{\nu}_{m_i} \sum_j Q_{ij} + \sum_k A_k J_{m_k} - V_i \, S_{chem_i} \tag{50}$$

and

$$\frac{\partial F}{\partial C_m} = \frac{V_i}{\Delta t} + \sum_j Q_{ij} - \sum_k A_k \frac{\partial J_{m_k}}{\partial C_m} - V_i \frac{\partial S_{chem_m}}{\partial C_m} \quad . \tag{51}$$

For the chemical rate derivative we have discretized the chemical derivative term as

$$\frac{\partial S_{chem_m}}{\partial C_{m_i}} = \sum_j \left(-a_{m_j} + b_{m_j}\right)\left(\frac{\bar{R}_j - R_j^{\,n}}{\bar{C}_{m_i} - C_{m_i}^n}\right) \tag{52}$$

where $\bar{C}_{m_i} = 1.05\ C_{m_i}$ is an arbitrary concentration offset defined solely for the purpose of generating approximations to derivatives. $\bar{R}_{m_i}$ is the forward reaction rate evaluated when $C_{m_i}^n = \bar{C}_{m_i}$ The subscript "m" denotes the different molecular species present. After convergence, terms with the "$\nu + 1$" superscript become the new time values with superscript, "n".

The relations for the molecular fluxes into surfaces and their derivatives depend on the type of boundary condition imposed for the diffusion segments.

For ***ratedep*** boundary conditions we have from Lq.(2) for molecular flux into the "kth" surface

$$J_{m_k} = K_{d_m} P_m - \sum_{i=1}^{n} K_{r_{ij}} C_{i_k} C_{j_k} \tag{53}$$

Then with the relation

$$P_{m_i} = C_{m_i} k_B T_{e_i} \tag{54}$$

where

k_B = Boltzmann's constant

T_{e_i} = temperature of enclosure "i"

we see that

$$\frac{\partial J_{m_k}}{\partial C_{m_i}} = K_{d_{m_k}} k_B T_{e_i} \tag{55}$$

If the ***lawdep*** boundary condition applies, where the solute concentration in the surface is related to the partial pressure of the gas above the surface through a solubility law such as Sieverts' law or Henry's law, then Eqs.(39) and (40) apply in defining the flux to the surface. Consider a generalized solubility concentration at the surface,

$$C''_{s_i} = K_\lambda P^\epsilon \tag{56}$$

where

K_λ = solubility pre-exponential coefficient

ϵ = the pressure exponent (0.5 for Sieverts' law, 1.0 for Henry's law).

Rewriting Eq.(40) with the following simplifications:

$$g_2 = \frac{D_{s_f} + D_{s_1}}{\Delta x_1} \tag{57}$$

$$g_{t2} = \frac{Q^*_{s_f} + Q^*_{s_1}}{R\left(T_f + T_1\right)^2}\left(T_1 - T_f\right) g_2 \tag{58}$$

when we substitute the solubility of Eq.(56) for the surface concentration into Eq.(40) with these simplifications we get

$$J_{s_f} = K_\lambda \left(C''_m k T_e\right)^\epsilon \left(g_{t2} - g_2\right) + C''_{s_1}\left(g_{t2} + g_2\right) \tag{59}$$

For the Newton-Rhapson iteration derivative of Eq.(51), we find

$$\frac{\partial J_{s_k}}{\partial C_m} = \epsilon K_\lambda (k T_e)^\epsilon C_m^{\epsilon - 1} (g_{t2} - g_2) \quad . \tag{60}$$

If we specify the surface concentration directly in the ***sconc*** boundary condition, we assume no coupling between the molecular gas species in the enclosure and any flux of atoms to the surface of the material. Hence the "J_{m_k}" term drops out of the Newton-Rhapson function "F". The same thing holds if the ***nonflow*** boundary condition is applied to the surface(s) bounding the enclosure.

4.0 CODE USAGE

In this chapter we present the requirements for installation and use of TMAP4. It is intended that this code be capable of running on a PC with as little as 640 K of random access memory. However, it is also suitable for installation on "mini" computers such as the IBM RISC System/6000™ or the Sun Workstation, or on mainframe systems. Here we present the technique for installing and preparing the code to run. Then in Chapter 5 we set forth the requirements for preparing the input file needed for problem execution.

4.1 Installation

4.1.1 General

We assume in what follows that the user has a good working knowledge of the computer system on which TMAP4 is to be used and of its operating system (DOS 5.0 for example). TMAP4 is furnished on a diskette that contains several files. These include FORTRAN-77 files for the preprocessor, computational and data extraction modules, compiled (.OBJ) files for these modules, DOS batch files, and ASCII files of sample programs and helpful information. The files and their purposes are as follows:

EQU.OBJ A dummy linkable equation object file. It is linked if no equations are used in the input file and no **TAPE7.OBJ** module is produced by the preprocessor.

EXTRACT.EXE An executable file for the code that generates x-y pairs for plotting from the **PLTDATA** file.

EXTRACT.FOR The FORTRAN-77 source file for **EXTRACT.EXE**.

READ.ME An ASCII file with installation instructions and supplementary data.

SAMP1.INP
SAMP2CHG.INP
SAMP2HT.INP
SAMP3.INP
Sample input data files for example problems provided in Appendix A.

SAMP1.OUT
SAMP2CHG.OUT
SAMP2HT.OUT
SAMP3.OUT
Output files generated when the corresponding input (***.INP**) files were run with TMAP4.

T.BAT A DOS batch utility file to compile and link a FORTRAN-77 file specified by the user using the F77L Lehey FORTRAN compiler.

TMAP4.BAT A DOS batch file to compile and run a problem using TMAP4. It automatically adds suffixes of **.INP** for the input file, **.OUT** for the output file, and **.PLT** for the **PLTDATA** file (used to generate x-y pairs for plotting.) A listing of **TMAP4.BAT** is provided in Appendix D for reference.

TMAPC4.FOR FORTRAN-77 source code for the main computational program module. This is the part of the code that performs the calculations.

TMAPC4.OBJ The compiled version of **TMAPC4.FOR** linked to form the executable code that performs the calculations. If other than the Lehey FORTRAN-77 compiler is available, this file must be regenerated by compiling **TMAPC4.FOR** before running the code. Do not link this module. Linking is done by **TMAP4.BAT** during execution.

TMAPP4.EXE The compiled and linked preprocessor module that assembles the input data in a form useable by **TMAPC4.FOR**. If other than the Lehey FORTRAN-77 compiler is available, this file must be regenerated by compiling **TMAPP4.FOR**, using the new compiler, and linking it.

TMAPP4.FOR The FORTRAN-77 source code for **TAMPP4.EXE.**

The general concept of TMAP4 operation is that the user generates an input data file that contains the specification of the problem (the topic of Chapter 5 of this manual). That input file is first copied by **TMAP4.BAT** into a file called **TMAPINP**, which is called for by the preprocessor (**TMAPP4**) leaving the original input file untouched for future reference. If in other than a DOS environment, the functions of the **TMAP4.BAT** file will need to be accomplished manually or by an equivalent script file. The **TMAPINP** file is processed by the preprocessor into an object file (called **TAPE7.OBJ**) that is linked together with the computational module (**TMAPC4.OBJ**) to form an executable binary file (**TMAPE.EXE**). Execution takes place, producing two ASCII output files and an ASCII summary data file. One of the output files, called **PREPOUT**, contains the listing of the input file and any diagnostic comments generated. If errors are detected by the preprocessor, the code does not execute. The second output file, **CODEOUT**, is only generated if the computational code actually executes. It contains the information produced to document the intermediate results of the calculations at each user-specified time step. These are combined by **TMAP4.BAT** into an ASCII output file (user specified root name with the **.OUT** extension). Again, if in other than a DOS environment, this combining of files will need to be accomplished manually or by another script file. The summary data or **PLTDATA** file (user specified root name with the **.PLT** extension) contains tabulated data from which x-y data pairs for plotting may be extracted using **EXTRACT.EXE**. Plotting may be done with a user-supplied program such as a spread sheet or other graphics package.

If equations are to be included in the input file, such as for specifying transport parameters or state-variable histories, the preprocessor generates a subroutine module called **EQU.OBJ** that contains the compiled equations. A FORTRAN-77 compiler is required in the same directory as the TMAP4 files to accomplish that task. If no equations are used in the input file, then the FORTRAN-77 compiler is not needed, and the dummy file, **EQU.OBJ** is linked in place of **TAPE7.OBJ**. Hence, the need for the compiler is dependent on the desire of the user to make use of the equations feature in the input. The code calculates numerical values in double precision arithmetic.

4.1.2 Setup

With this background, the first task in installation is to copy the files on the supplied disk into the working directory on a hard drive (a single floppy disk will not hold both the compiler and TMAP4). That directory must also contain the FORTRAN-77 compiler if needed and a linker program.

If other than the Lehey FORTRAN-77 compiler is to be used, it will be necessary to recompile the **TMAPC4.FOR**, **TMAPP4.FOR**, and **EXTRACT.FOR** files using the alternate compiler. The **TMAPP4.OBJ** and **EXTRACT.OBJ** files produced in that operation will also need to be individually linked. As a check to see that the installation is correct, try running one of the sample problems. The same procedures are applicable for running user-defined problems, once the problem has been specified in the ***.INP** file.

4.2 Operation

To run TMAP4 make the active directory the one in which TMAP4 has been loaded. For example type

```
C:\>cd\tmap4
```

if **C:\TMAP4** is the directory where TMAP4 is located. Then start the problem. If **SAMPLE-1.INP** is to be run, type

```
C:\TMAP4>tmap4 sample-1
```

The program will automatically specify that the output file should be called **SAMPLE-1.OUT** and the **PLTDATA** file should be renamed **SAMPLE-1.PLT**. It will first copy **SAMPLE-1.INP** into **TMAPINP**, write the input data FORTRAN-77 file called **TAPE7.FOR** and compile it into **TAPE7.OBJ**. Then it will link that or the dummy file **EQU.OBJ** with **TMAPC4.OBJ** to create the executable file, **TMAPE.EXE**, and execute the problem. If the preprocessing operation is successful, the screen will display the message "TMAP4 PROBLEM IN PROGRESS", and a display at the bottom of the screen will indicate progress through the problem. The output file will contain a copy of the input file, listings of intermediate calculational results, and a statement at the end of the output file indicating

```
****** NORMAL PROBLEM TERMINATION REACHED ******
```

and the screen will display the message

```
*****************PROBLEM EXECUTION COMPLETE*****************
```

The operator may wish to rename the furnished **SAMPLE-1.INP** file to some other name (keep the "**.INP**" extension) before running the trial calculation to prevent overwriting the output and plot files and thus facilitate comparison with results after the calculation is complete. When the problem has finished, the output file(s) may be edited with any text editor. Note that **TMAP4.BAT** deletes **PREPOUT** and **CODEOUT** after combining them into the ***.OUT** file and renames **PLTDATA** to the ***.PLT** file.

It would also be worthwhile to exercise the **EXTRACT.EXE** code to verify its functioning. If the **SAMPLE-1.PLT** file was produced, this may be done by entering

```
C:\TMAP4>extract
```

to start execution. The program will then request a name for its output file and present a number of choices for output, depending on the *plot input* data block parameters. The output file consists of two columns of numbers. The first is the time at which the values apply. The second is the parameter selected by the user from the available choices. These will be explained in detail in Section 5.2.9.

4.2 Restart Capability

TMAP4 includes a restart capability to allow for alterations of equations, tables, or any *control input* parameter (see Section 5.2.8 for a discussion of these parameters). The code will create a *restart* file called **RSTOUT** at the completion of the problem, when the specified problem end time has been reached. This file, if present, is overwritten during the execution of TMAP4, so if later restarting is contemplated, the **RSTOUT** file must be saved under a different name before running TMAP4 for another problem.

Restarting a problem involves several steps:

1. Ensure the presence of the previously generated **RSTOUT** file with the local file name changed to **RSTIN**.

2. Revise the input file (see Chapter 5 for input file definition) as follows:

   a. The first non-comment card of the input deck should be

   restart

   b. Remove the *title input, main input, enclosure input, thermal input, diffusion input*, and *plot input* data blocks, leaving only *equation input, table input*, and *control input* data blocks and the **end of data** declaration.

   c. Be sure that the number and sequence of equations in the *equation input* block and the number and sequence of tables in the *table input* block remain the same in the restart file as in the original input file.

   d. Use of the **time** statement in the *control input* block is optional when restarting a problem. If included, the new value of starting time will override the elapsed time calculated in the previous execution. If left out, the starting time for the restart execution will be the ending time of the previous execution. All other *control input* parameter statements are required. Note that the **timend** value is measured with respect to the total problem time and must exceed the new starting time if any calculations are to be accomplished.

3. Provide the restart ***.INP** file with a modified root name to prevent overwriting of the already written ***.OUT** and ***PLT** files by TMAP4 during the continued execution. For example, if the original input file was named **PROB.INP**, the user

may wish to name the restart input file **PROB-1.INP**. Note that to operate under the DOS batch executive program, **TMAP4.BAT**, the input file must have the "**.INP**" filename extension. An alternative would be to rename the previously generated ***.OUT** and ***PLT** files to prevent overwriting. Similar precautions should be taken if operating outside of **TMAP4.BAT** control.

4. Initiate execution by entering

```
C:\TMAP4>tmap4 <filename>
```

where <*filename*> is the root name (with no extensions) of the restart input file just generated. (Here C:\TMAP4> is assumed to be the DOS prompt.)

Sample Problem No. 2 in Appendix B is an example of a problem that made use of the restart capability. Input files for both the original and the restarted problem are given there.

5.0 THE INPUT FILE

In this chapter we present the requirements and details for constructing or editing an input file for use with TMAP4. This file is normally assembled using a text editor of the user's choice. It must exist as an ASCII file before starting TMAP4, and it must have the filename extension of "**.INP**" if TMAP4 is used with the DOS batch file, **TMAP4.BAT**. While the structure of the input file features several "user friendly" characteristics, there are some firm requirements that must be adhered to. We first discuss those requirements and then the various sections of the input file.

5.1 General Requi.ements

TMAP4 is capable of running complex problems that involve a number of diffusion segments that may be connected to each other or to enclosure volumes. These volumes may be connected to each other and/or to one or more diffusion segments. Specific restrictions on the structuring of problems for TMAP4 include the following.

- Maximum number of structures for thermal/solute diffusion is 20 with no more than 200 total nodes. These will be referred to as ***thermsegs*** and/or ***diffsegs*** in the code. They are the same structures for both kinds of calculations. Each end of a *thermseg/diffseg* may interact with only one enclosure.

- Linking of structures (*thermsegs/diffsegs*) together as composite elements is accommodated.

- Maximum number of defined diffusion species is 10.

- Thermal calculations in individual *thermsegs/diffsegs* are optional.

- Trapping calculations in individual *thermsegs/diffsegs* are optional.

- Maximum number of defined ***enclosures*** is 40. Each *enclosure* may interact with more than one *thermseg/diffseg*.

- Maximum number of defined flow paths for convective flow between *enclosures* is 100.

- Maximum number of defined enclosure species is 10.

- Maximum number of chemical reactions that may be defined for operation in *enclosures* is 50.

- Maximum number of user-defined equations is 50.

- Maximum number of tables is 10 with an aggregate of 240 data pairs.

- Of the nine input data blocks, eight are required to be present in the proper sequence, even if empty.

- Individual entries within the data blocks are order-independent though most are required to be present.

There are nine input data blocks which must appear in the proper sequence. Each must begin and end with a data block declaration statement. There is also a required terminal declaration statement. Many of these blocks have required parameter statements that must be entered. Others may be left empty, but the block declaration statements given below must be present in the input file. These blocks with their declaration statements are as follows.

Block Name	Required Declarations
1. Title	**title input** ⋮ **end of title input**
2. Main	**main input** ⋮ **end of main input**
3. Enclosure	**enclosure input** ⋮ **end of enclosure input**
4. Thermal	**thermal input** ⋮ **end of thermal input**
5. Diffusion	**diffusion input** ⋮ **end of diffusion input**

6. Equation	**equation input** ⋮ **end of equation input**	May be empty
7. Table	**table input** ⋮ **end of table input**	May be empty
8. Control	**control input** ⋮ **end of control input**	
9. Plot	**plot input** ⋮ **end of plot input**	May be empty
Terminal Declaration	**end of data**	

With the exception of these block declaration statements and the *title input* data, nearly every input data statement must terminate with ",**end**" to let the preprocessor know that the data statement is concluded. Input data may be in any of the first 80 columns of the line and may be continued to succeeding lines provided that the last character in the line to be continued is not a delimiter (, . + - * /). Note, too, that there is no space allowed between the comma and the word "end" in the statement termination. The *title input* data consists of one or two lines of character data, and any ASCII characters may be included.

Comment lines may be included by placing a "**$**" in the first column of the line. These comment lines must not come in the middle of a multi-line input statement. Partial-line comments may be included by placing the "**$**" in a line after the "**,end**" statement terminator.

No blank lines are permitted in the input file.

During preprocessing of the input file, the preprocessor makes checks for internal consistency, completeness, and correctness of the input data. When errors are detected, diagnostic/warning messages are inserted into the **PREPOUT** file, which contains the listing of the input data, to indicate where the error was detected and the nature of the error. Preprocessing then continues in the next data block. If errors have been detected during the preprocessing, an error flag is set that prevents execution of the computational module.

We now proceed to present detailed requirements and formats for the various input data blocks. These will be discussed in the order in which these blocks must appear as indicated previously. In what follows, two kinds of data are indicated. Some parameters are keywords that must be entered exactly as indicated in the format and example statements. These keywords will be shown as follows:

dspcnme

Other data are user-supplied names or quantities. These must be entered either as literals, integers or real numbers. When these data are referred to in the examples that follow, they will be shown as in the following examples:

<*dspc*(α)$_2$>	The name ("α" implies literal or alphabetic) of a diffusion species
<*nbr(i)*>	The number ("i" means integer) of enclosures to be specified
<*temp(r)*>	Temperature value ("r" indicates real number).

The subscripts if any indicate the position in a sequence or series. Brackets around a variable name or statement indicate it is optional and need not be included.

5.2 Detailed Requirements

In this section we present the detailed requirements for the input file for TMAP4. In each subsection there will be a general statement as to the function of the data in that block, the format for that information, a statement of the required information, and an example from a typical input deck that shows the features described.

5.2.1 Title Input

The *title input* information is used to identify the problem being simulated. It is a user-specified block of text that may be at most two lines of 80 characters. This information is copied by **TMAPC4** to the output file at the beginning of the output data for each time step. Any ASCII characters may be used in the *title input* data.

Format:

title input
[<*title text data*>]
end of title input

Example:

```
title input
SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA
end of title input
```

5.2.2 Main Input

In this section, the user supplies the main problem formulation information to define the general structure of the problem. Information such as the diffusion and enclosure species names, the number of enclosure volumes, the number of *thermsegs/diffsegs* and the number of nodes in each, and if any segments will be linked. Five statements are required in this block. The order in which they are entered is not important, but all five must be supplied. Four of the five statements are required in this data block. The **linksegs** statement is optional. Except for the terminal "**,end**", spacing within the statements is optional.

Format:

dspcnme = *<dspc(α)$_1$>,<dspc(α)$_2$>,...,<dspc(α)$_n$>*,**end**

The *<dspc(α)$_i$>*, i=1,2,...,n, are the user-defined names of the species that will be used in *diffsegs*. This list must be all-inclusive and contain the names of all species that will be considered in any of the problem *diffsegs*. The names are alphabetic (lower case only) or numeric, but they may not contain delimiters. A maximum of 10 diffusion species names may be defined.

espcnme = *<espc(α)$_1$>,<espc(α)$_2$>,...,<espc(α)$_n$>*,**end**

The *<espc(α)$_i$>*, i=1,2,...,n, are the names of the species that will be used in *enclosures*. This list must be all-inclusive and contain the names of all species that will be considered in any of the problem *enclosures*. The names are alphabetic (lower case only) or numeric, but they may not contain delimiters. Further, they may not duplicate any of the *<dspc(α)$_i$>* defined in the **dspcnme** statement. A maximum of 10 different enclosure species may be defined. All enclosure species must be defined in each enclosure.

segnds = *<nodes(i)$_1$>,<nodes(i)$_2$>,...,<nodes(i)$_n$>*,**end**

The *<nodes(i)$_j$>*, j=1,2,...,n, are the numbers of nodes in each of the "*n*" *thermsegs/diffsegs*. One integer value must appear for each *thermseg/diffseg* in the problem. At least one but no more than 20 *thermsegs/diffsegs* must be defined. The number of entries in this statement defines the number of *thermseg/diffseg* structures in the problem. The order in which these *<nodes(i)$_i$>* are specified establishes the segment relative sequence number for each segment so specified.

The number of nodes in each *thermseg/diffseg* must be two greater than the number of diffusion nodes desired per segment to give one node with zero segment thickness at each surface of the *thermseg/diffseg*. The total number of all nodes for all defined *thermsegs/diffsegs* must not exceed 200, and no more than 100 nodes may be defined in any single *thermseg/diffseg*.

nbrencl = <*nbr(i)*>,end

This statement specifies the number of enclosures <*nbr(i)*> to be included in the problem. The quantity is an integer. The maximum number of enclosures allowed is 40.

[linksegs = <*seg(i)*$_1$>,...,<*seg(i)*$_{n1}$>[,also,<*seg(i)*$_{n1+1}$>,...,<*seg(i)*$_{n2}$>],end]

This statement is optional. If omitted, none of the *thermsegs/diffsegs* will be considered to be linked together. If included, then the segments will be linked together in the sequence in which they are listed. The entries, <*seg(i)*$_i$> are integers indicating segment relative sequence numbers established in the **segnds** statement in this *main input* data block. The right side (maximum x node) of the segment defined in the <*seg(i)*$_1$> position of the **segnds** statement will be linked together with the left side (minimum x node) of the segment identified in the <*seg(i)*$_2$> position. The right side of the <*seg(i)*$_2$> segment will be linked with the left side of the <*seg(i)*$_3$> segment, and so on. Hence, the segments must be identified in each series in the order in which the linking takes place. Multiple separate composite linkages are specified with the optional "**,also,**" conjunction followed by the second linkage series. The number of times this additional linkage-definition process may be repeated in the **linksegs** statement is arbitrary provided that (1) only those segments defined in the **segnds** statement may be referred to here, and (2) each segment may be linked to only one other segment at each end. The left end of the first segment in each linked series and the right end of the last segment in that series are assumed unlinked and therefore interact only with enclosures.

Example:

```
main input
  dspcnme=t,d,end
  espcnme=t2,dt,d2,end
  segnds=18,12,end
  nbrencl=2,end
  linksegs=1,2,end
end of main input
```

This block defines a system consisting of two *thermsegs/diffsegs* that are linked together to form a composite structure. The first segment has 18 nodes while the second has only 12. Two enclosures are also defined, but there is no informatio n in this block as to the physical relationship between the two enclosures and the two *thermsegs/diffsegs*. The species that diffuse in the structures are identified here as "t" and "d" while in the enclosures there are different species, "d2", "t2", and "dt".

Example:

```
main input
  dspcnme=t,d,end
  espcnme=t2,dt,d2,end
  segnds =15,12,15,12,15,12,15,12,15,12,15,12,15,12,15,12,end
  nbrencl=2,end
  linksegs=1,2,also,3,4,also,5,6,also,7,8,also,9,10,also,11,12
           also,13,14,also,15,16,end
end of main input
```

This block is similar to the previous one, but instead of two *thermsegs/diffsegs* there are 16 defined. Odd-numbered segments have 15 nodes, and even-numbered ones have 12. These are linked together in eight pairs, each pair containing an odd-numbered segment and an even-numbered one. Note that there is no comma at the end of the first **linksegs** line. Linked segments need not be consecutively numbered.

5.2.3 Enclosure Input

In this section the user provides TMAP4 specific data required to define the *enclosure* volumes. There are two types of enclosures. One is called ***functional***, and the other is a ***boundary*** enclosure. *Functional* enclosures allow dynamic pressure and chemical reaction calculations to be included during the running of the problem. There may also be convective flows from one *functional* enclosure to another. On the other hand, *boundary*-type enclosures are associated with pre-determined conditions that do not depend on thermal effects or mass diffusion. Fluid flows are allowed to and from *boundary* type enclosures, but these do not influence the static properties in them.

Each enclosure is identified with a separate sub-block of data. The sequence in which these definitions are made determines the enclosure relative sequence number, however, an arbitrary enclosure reference number must also be defined by the user. There are two requirements on these definitions: (1) all *functional* enclosures must be defined before any *boundary* enclosures are defined; and (2) the user-defined reference numbers must be integers lying in the range between 1 and the total number of enclosures.

Each sub-block begins with a **start** declaration statement, but there is no terminal declaration for the sub-block. There are order-independent parameter statements after the **start** statement for each sub-block of *enclosure* inputs. The number of these statements depends on whether the enclosure is *functional* or *boundary* in nature. We first consider *functional* enclosures.

5.2.3.1 Functional Enclosures

There are five parameter statements following the "**start**" declaration in specifying *functional enclosures*. Of these one is optional, the specification of fluid flows between enclosures.

Format:

start func,<*enclnbr(i)*> [,end]

> The keyword "**func**" in this statement identifies this as a *functional* type enclosure. These must be defined first, before *boundary*-type enclosures. The first enclosure defined would have associated with it an enclosure relative sequence number of 1, the second would be 2, and so forth. The integer identified by <*enclnbr(i)*> is a user-specified *enclosure reference number* that may take on any value between 1 and the total number of enclosures identified in the **nbrencl** statement in the *main input*. This is a rare instance where the terminal "**,end**" is optional.

etemp = <*temp(r)*>,end

> This statement identifies the temperature of the enclosure. The value of that temperature, <*temp(r)*>, is a real constant. There is at present no provision in TMAP4 for variation in enclosure temperatures. This temperature is the one that relates pressure and enclosure species concentration.

espres = <$espc(\alpha)_1$>,<$value(r)$>,...,<$espc(\alpha)_n$>,<$value(r)$>,**end**

For *functional* enclosures this statement defines the initial *enclosure* species partial pressures in units of pascals. There must be an entry for each enclosure species defined in the **espcnme** statement in the *main input* block. If in this particular *enclosure* the partial pressure of a given species is zero, it must be so stated.

Each entry module consists of the name, <$espc(\alpha)_i$>, of the enclosure species together with a real number, <*value(r)*>, for the initial partial pressure of that species.

reaction = **nequ**,<*neq(i)*>,
rateequ,<*equnbr(i)*>,
nreact,<*nbrreact(i)*>,<$espc(\alpha)_1$>,<$stoich(r)_1$>,...,
<$espc(\alpha)_n$>,<$stoich(r)_n$>,
nprod,<*nbrprod(i)*>,<$espc(\alpha)_1$>,<$stoich(r)_1$>,...,
<$espc(\alpha)_n$>,<$stoich(r)_n$>,

.

.

.

[**ratequ**,<*equnbr(i)*>,
nreact,<*nbrreact(i)*>,<$espc(\alpha)_1$>,<$stoich(r)_1$>,...,
<$espc(\alpha)_n$>,<$stoich(r)_n$>,
nprod,<*nbrprod(i)*>,<$espc(\alpha)_1$>,<$stoich(r)_1$>,...,
<$espc(\alpha)_n$>,<$stoich(r)_n$>,]
end

This rather involved statement defines the chemical reactions that are taking place within the *enclosure* being defined. The keyword **nequ** indicates that the next number, <*nequ(i)*>, is the number of chemical reactions that are taking place in that *enclosure*. The next keyword, **ratequ**, begins a module that must be defined for each of the <*nequ(i)*> reactions. The parameter <*equnbr(i)*> identifies the *equation* number where the reaction rate constant, R_c, is found. That equation may have "time", "temp" (enclosure temperature), and "conc(i)" or *enclosure* species concentrations as arguments. The subscript on concentrations in the equation is the relative sequence number of the enclosure species. The keyword **nreact** indicates that the next parameter, <*nbrreact(i)*>, is the number of reactants in the reaction. The keyword, **nprod** together with <*nbrprod(i)*> indicate the number of products in the reaction.

Consider for example an *enclosure* in which the two reactions

$$T_2 + H_2O \rightleftharpoons HTO + HT \qquad (61)$$

$$HT + H_2O \rightleftharpoons HTO + H_2 \qquad (62)$$

are taking place. In each equation there are two reactants and two products. In the first reaction the reactants may be labelled (<*espc(α)*$_i$>) t2 and h2o and the stoichiometric coefficient (<*stoich(r)*$_i$>) for each reactant is the real number 1.0. The products may be labelled hto and ht, and again each has a stoichiometric coefficient of 1.0. Suppose that the forward reaction rate expressions for these two reactions are defined in the third and fourth equations in the *equation input* block.

Example:

```
reaction = nequ,2,ratequ,3,nreact,2,t2,1.0,h2o,1.0,nprod,2,hto,1.0,ht,1.0
                      ratequ,4,nreact,2,ht,1.0,h2o,1.0,nprod,2,hto,1.0,h2,1.0,end
```

Again notice that there is no comma at the end of the first line.

evol = <*enclv(r)*>,end

This statement defines the volume of the *functional enclosure*. The value, <*enclv(r)*>, is a real number with units of m^3. TMAP4 only provides for constant enclosure volumes.

Example:

```
evol=1.38e+04,end
```

[outflow = nbrflwp,<*npath(i)*>

$$\textbf{qflow,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}\textbf{,rencl,}<\mathit{value(i)}>$$

.

.

.

$$[\textbf{qflow},\left\{\begin{array}{l}\textbf{const},<\mathit{value}(r)>\\ \textbf{equ},<\mathit{value}(i)>\\ \textbf{tabl},<\mathit{value}(i)>\end{array}\right\},\textbf{rencl},<\mathit{value}(i)>],\textbf{end}]$$

This optional statement defines the flow paths **from** the enclosure being defined. The number of such flow paths equals <*npath(i)*>. For each of these paths there must be a **qflow** statement that includes two parameters. The first, coming just after the **qflow** keyword, is the volumetric flow rate in units of m^3/s. It may be specified as a real constant (**const**) having the value <*value(r)*>; as an equation (**equ**), holding position <*value(i)*> in the equation list, that is a function of "time"; or as a lookup table (**tabl**) holding position <*value(i)*> in the list of tables, also a function of "time". The second parameter, following the **rencl** keyword, is the user- defined reference number of the *enclosure* receiving the flow from the one currently being defined. Note that the total number of flow paths <*npath(i)*> must not exceed 100.

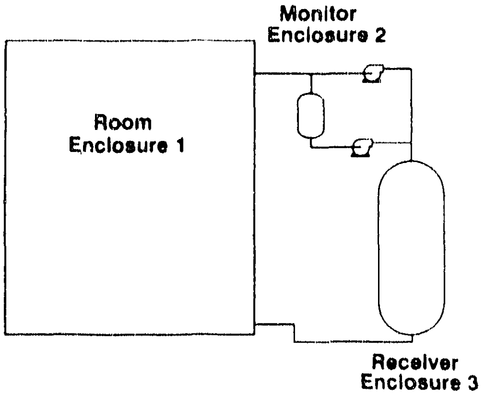


Figure 6. Schematic of system consisting of a room, a monitor, and an air-cleanup unit.

For an example, suppose we consider a system that involves a room with a volume of 500 m^3 into which 100 Ci of tritium in the form of T_2 has been released. The system also includes a tritium monitor and an air cleanup unit. These components are shown schematically in Figure 6. We define each of the chambers as a *functional*-type enclosure and assign the reference number 1 to the room. Air carrying the tritium species is drawn from the room by circulation pumps through two paths. One path with a flow of 20 liters per minute goes into the 20-liter ion chamber, which we label enclosure 2. From there it is carried to the 20-m^3 receiver tank for the air-cleanup unit, enclosure 3. The second flow path from enclosure 1 goes directly to enclosure 3. It has a time history that involves stops and starts. We can best define that with a table, in this case *table 1*. The air-cleanup system has a flow back to the room that is the sum of the flows from enclosure 2 and enclosure 1. We model the tritium removal in the cleanup unit as a series of chemical reactions, in *equations* 3 - 6, with only reactants (no products) and a forward rate coefficient that corresponds with the molecule arrival rate at a surface. We are interested in monitoring HTO concentrations in enclosures 1 through 3 as a function of time. We assume that the two chemical reactions described previously produce that tritiated water in all the enclosures. Forward reaction rate coefficients for HTO-producing reactions are in the *equation input* block as equations 1 and 2. The *enclosure* definition block would have the following structure.

Example:

```
enclosure input
start func,1,end                         $ This is the room
  etemp=303.0,end
  esppres=t2,8.686e-3,ht,0.0,h2o,714.0,hto,0.0,end
  reaction=nequ,2,ratequ,1,nreact,2,t2,1.0,h2o,1.0,nprod,2,hto,1.0,ht,1.0
                  ratequ,2,nreact,2,ht,1.0,h2o,1.0,nprod,1,hto,1.0,end
  evol=500.0,end
  outflow=nbrflwp,2,qflow,const,3.33e-4,rencl,2
                    qflow,tabl,1,rencl,3,end
start func,2,end                         $ This is the ion chamber
  etemp,303.0,end
  esppres=t2,0.0,ht,0.0,h2o,714.0,hto,0.0,end
  reaction=nequ,2,ratequ,1,nreact,2,t2,1.0,h2o,1.0,nprod,2,hto,1.0,ht,1.0
                  ratequ,2,nreact,2,ht,1.0,h2o,1.0,nprod,1,hto,1.0,end
  evol=0.02,end
  outflow=nbrflwp,1,qflow,const,3.33e-4,rencl,3,end
start func,3,end                         $ This is the cleanup chamber
  etemp,303.0,end
  esppres=t2,0.0,ht,0.0,h2o,714.0,hto,0.0,end
  reaction=nequ,6,ratequ,1,nreact,2,t2,1.0,h2o,1.0,nprod,2,hto,1.0,ht,1.0
                  ratequ,2,nreact,2,ht,1.0,h2o,1.0,nprod,1,hto,1.0
                  ratequ,3,nreact,1,t2,1.0,nprod,0
                  ratequ,4,nreact,1,ht,1.0,nprod,0
                  ratequ,5,nreact,1,h2o,1.0,nprod,0
                  ratequ,6,nreact,1,hto,1.0,nprod,0,end
  evol=20.0,end
  outflow=nbrflwp,1,qflow,tabl,2,rencl,1,end
end of enclosure input
```

5.2.3.2 Boundary Enclosures

Boundary enclosures represent static-condition environments that are not altered by heat transfer or diffusion effects. Three parameter statements are required for *boundary enclosures* following the "**start**" declaration for the sub-block.

Format:

start bdry,<*enclnbr(i)*>**[,end]**

The keyword "**bdry**" in this statement identifies this as a *boundary*-type enclosure. These may only be defined after any *functional enclosures* have been defined. The first *boundary enclosure* defined would have associated with it an enclosure relative sequence number 1 greater than that of the last *functional enclosure* defined. The integer identified by *<enclnbr(i)>* is a user-specified *enclosure* reference number that may take on any value between 1 and the total number of enclosures identified in the **nbrencl** statement in the *main input*. However, no two enclosures may share the same reference number. Again this is an instance where the terminal "**,end**" is optional.

etemp = *<temp(r)>*,end

This statement identifies the temperature of the enclosure. The value, *<temp(r)>* is a real constant. There is no provision at present in TMAP4 for variation in enclosure temperatures. This temperature is the one that relates pressure and enclosure species concentration.

$$\textbf{espres} = <espc(\alpha)_1>, \left\{ \begin{array}{l} <value(r)> \\ \textbf{const,}<value(r)> \\ \textbf{equ,}<value(i)> \\ \textbf{tabl,}<value(i)> \end{array} \right\}, \ldots,$$

$$<espc(\alpha)n> \left\{ \begin{array}{l} <value(r)> \\ \textbf{const,}<value(r)> \\ \textbf{equ,}<value(i)> \\ \textbf{tabl,}<value(i)> \end{array} \right\} \textbf{,end}$$

For *boundary enclosures* this statement defines the species partial pressures in units of pascals. There must be an entry module for each enclosure species defined in the **espcnme** statement in the *main input* block. If in this particular *enclosure* the partial pressure of a given species is zero, it must be so stated.

Each entry module consists of the name, *<espc(α)$_i$>*, of the enclosure species together with a statement indicating how the partial pressure of that species is to be computed. That specification has four options. The first two, *<value(r)>* and **const,***<value(r)>* have exactly the same effect of setting the partial pressure of the stated species to a constant value throughout the entire problem. The third option, **equ,***<value(i)>*, refers the computation of the partial pressure for that *enclosure* species according to the equation appearing in position *<value(i)>* in the list of equations in the *equation input* section to be discussed later in section 5.2.6. That equation may use only "time", the elapsed time from the start of the problem, as an argument. The last option, **tabl,***<value(i)>* causes the pressure to be determined from a look-up table in the *table input* block. Tables are entered as "time" - "espres" pairs, and the value returned is linearly interpolated from the table values.

[outflow = nbrflwp,<*npath(i)*>

$$\textbf{qflow,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}\textbf{,rencl,}<\mathit{value(i)}>$$

.

.

.

$$\textbf{[qflow,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}\textbf{,rencl,}<\mathit{value(i)}>\textbf{],end]}$$

This optional statement defines the flow paths **from** the enclosure being defined. The number of such flow paths is <*npath(i)*>. For each of these paths there must be a **qflow** module that includes two parameters. The first, coming just after the **qflow** keyword, is the volumetric flow rate in units of m^3/s. It may be specified as a real constant (**const**) having the value <*value(r)*>; as an equation (**equ**), holding position <*value(i)*> in the equation list, that is a function of "time"; or as a lookup table (**tabl**) holding position <*value(i)*> in the list of tables, also a function of "time". The second parameter, following the **rencl** keyword is the user- defined reference number of the *enclosure* receiving the flow from the one currently being defined. Note that the total number of flow paths <*npath(i)*> for the entire problem must not exceed 100.

For an example, we consider two *enclosures* that exist, one on either side of a divertor plate in a tokamak fusion reactor. Because of the pulsed nature of tokamak operation, the pressure of gasses on the plasma side fluctuates in a periodic manner. We assume that those gases consist of the two homonuclear species, D_2 and T_2 and the heteronuclear species, DT. The fluctuations are defined by *equations* 1 and 2, one for the two homonuclear species in the plasma-side *enclosure* and one for the heteronuclear species whose pressure will be about twice that of the homonuclear species. In this case, no flows are defined between either *enclosure* and any other.

Example:

```
enclosure input
```

```
start bdry,1                                 $ Plasma side
  etemp=673.,end
  esppres=t2,equ,1,dt,equ,2,d2,equ,1,end     $ Burn cycles in Eqs.1 & 2
start bdry,2                                 $ Coolant channel
  etemp=323.2,end
  esppres=t2,0.0,dt,0.0,d2,0.0,end
end of enclosure input
```

5.2.4 Thermal Input

In this data block the user provides the required input for determining or specifying the temperature profile for *thermsegs/diffsegs* or *linked* segment groups. A maximum of 20 *thermsegs/diffsegs* is allowed including those that exist as *linked* segments. A segment's temperature distribution can be calculated by TMAP4 from user-supplied input using the conduction equation and user-specified temperature or heat-flux boundary conditions. Temperatures may be specified at boundaries as functions of time by equations or tables or as constants. Although heat transfer calculations are arbitrary, TMAP4 requires that if thermal calculations are performed for one segment of a *linked segment* structure, they must be performed for all segments that comprise that *linked segment*.

Note that at least one *thermseg/diffseg* must be defined in every problem, even if it takes no active part. It may be a dummy segment, but its definition must conform with the requirements for active segments. The order in which the various *thermsegs/diffsegs* are defined gives them a relative sequence number. It is the same sequence in which they were identified in the "**segnds**" statement in the *main input* block. This is their only identification in TMAP4, and all references to them make use of this program-assigned identifier.

Within this block of data there are sub-blocks that define the individual structures. Each sub-block begins with a "**start**" declaration statement. There are eight order-independent statements following the "**start**" declaration statement. The first two of these are required for all *thermsegs/diffsegs*. The last six are optional. If statements 3 - 7 are omitted, TMAP4 assumes that no thermal calculations are required, and none are attempted. The last one ("**hgap**") is only needed for *linked* segments in which temperature calculations are performed. Aside from "**hgap**" if any of the last six are included, they all must be included.

Format:

start thermseg [,end]

There must be a *thermseg* sub-block entry for each individual diffusion/heat-transfer element, whether or not it is linked. It is also the place where the geometric structure of the *thermseg/diffseg* is defined. A maximum of 20 such elements may be defined. This statement begins each sub-block defining heat transfer conditions. The "**,end**" closure is optional here. The order of statements within the sub-block is optional. The order in which the various *thermsegs/diffsegs* are defined gives them a relative sequence number. Their order in this block must agree with the order in which they were specified in the "**segnds**" statement in the *main input* block. It is their only identification in TMAP4, and all references to them make use of this program-assigned identifier. Partial comment lines, beginning with "$" after the "**end**" statement, or full comment lines may be used to clarify the ıdentity of each element for the user.

delx = 0.0,$<dx(r)_1>,<dx(r)_2>,\ldots,<dx(r)_n>$**,0.0,end**

This required statement defines the nodal separations (Δx_i) for this *thermseg/diffseg*. The real-number entries, $<dx(r)_i>$, with units of meters, are identified with the nodes in left-to-right order, that is, in increasin'; distance, x. Both left- and right-surface nodes are required to have zero thickness, even if the segments are linked. There must be an entry for each node, and the number of entries, including one for each surface node, must correspond to the numbers defined in the "**segnds**" statement in the *main input* data block.

Arbitrary subsets of the input data having the same numerical value may be combined in a shorthand input format:

...*<numnds(i)>*<dx(r)>*...

where *<numnds(i)>* is an integer indicating how many nodes are to be specified in this cluster, and *<dx(r)>* defines the uniform width of each of those segments.

$$\textbf{dtemp} = \left\{ \begin{array}{l} <\mathit{temp}(r)_1>,\ldots,[<\mathit{numnds}(i)> * <\mathit{temp}(r)>],\ldots,<\mathit{temp}(r)_n> \\ \textbf{const}, <\mathit{value}(r)> \end{array} \right\}$$

end

This required statement establishes the initial temperature for this *thermseg/diffseg*. Two options are possible. The first shows entries for each node in the segment, some or all of which may use the shorthand entry for multiple nodes in each segment that have the same temperature. Use this form when there is an initial temperature profile in the segment. The second option indicates by the keyword, **const**, that all of the nodes in this segment have the same uniform temperature given by *<value(r)>*. For *thermsegs/diffsegs* in which no thermal calculations are performed (statements following this one are omitted), the values defined in this statement are the node temperatures throughout the entire calculation.

For an example of the simplest kind of *thermseg*, consider a segment with 12 nodes, the middle ten of which have uniform spacing, that is at the constant and uniform temperature of 303 K.

Example:

```
thermal input
start thermseg
  delx=0.0,10*1.5e-4,0.0,end
  dtemp=12*303.,end
end of thermal input
```

The remaining statements for *thermseg* sub-blocks are optional as a group.

$$[\textbf{tcon} = \left\{ \begin{array}{l} \textbf{const}, <\mathit{value}(r)> \\ \textbf{equ}, <\mathit{value}(i)> \\ \textbf{tabl}, <\mathit{value}(i)> \end{array} \right\} \textbf{,end}]$$

This statement is optional but must be included if material properties within the segment are to vary with time and/or temperature. It defines the thermal conductivity in W/m ·K for the entire segment. There is no provision for this material property to vary spatially within an individual segment except as temperature changes. The value is specified either as a constant with value *<value(r)>*, as an equation which may be a function of "time" and/or "temp", which appears as equation *<value(i)>* in the *equation input* block, or as a lookup table, again in position *<value(i)>* in the *table input* block, in which the first of each number pair is temperature while the second is thermal conductivity.

$$\textbf{[rhocp} = \left\{ \begin{array}{l} \textbf{const,}<\textit{value}(r)> \\ \textbf{equ,}<\textit{value}(i)> \\ \textbf{tabl,}<\textit{value}(i)> \end{array} \right\} \textbf{,end]}$$

This statement is optional, but must be included if properties within the segment are to vary with temperature. It defines the thermal capacity (product of material density and specific heat) for the segment in units of J/m^3 ·K. Again, there is no provision for this material property to change spatially within an individual segment except with changes in temperature. The value is specified either as a constant with value <*value(r)*>, as equation <*value(i)*> which may be a function of "time" and/or "temp", or as a lookup table, again in position <*value(i)*> in the *table input* block, in which the first of each number pair is temperature while the second is thermal capacity.

$$\textbf{[hsrc} = \left\{ \begin{array}{l} \textbf{const,}<\textit{value}(r)> \\ \textbf{equ,}<\textit{value}(i)> \\ \textbf{tabl,}<\textit{value}(i)> \end{array} \right\} \textbf{,srcpf,}<\textit{value}(r)_1>,\ldots,<\textit{value}(r)_n>\textbf{,end]}$$

This statement is optional, but must be included if properties within the segment are to vary with time and/or temperature. It defines the local heat generation rate in units of W/m^3 for each of the node segments in the *thermseg/diffseg*. The specification consists of two parts. The first or temporal part identifies the magnitude and time history of the local heat generation rate. It may be specified as a constant with a magnitude <*value(r)*>, as an equation which is a function of "time", or as a time-dependent lookup table. Conventions for identifying which equation or table is referred to are the same as with other entries in this block. The keyword **srcpf** identifies the following data as the spatial distribution or source peaking factor profile. An entry must be made in this sequence for each node in the *thermseg/diffseg*. Those for the two surface nodes should be zero because they have no associated volume. The shorthand entry format,...<*numnds(i)*>* <*value(r)*>..., may be used for multiple nodes where the **srcpf** has the same value.

The local heat generation rate is calculated as the product of the temporal part (applies to the entire segment) and the spatial part (an entry for each node). Hence, the magnitude of the heat generation rate may be associated with either part or divided between the parts, at the users discretion.

$$\left\{\begin{matrix}\textbf{htrbcl}\\ \textbf{htrbcr}\end{matrix}\right\} = \left[\begin{matrix} \left\{\begin{matrix}\textbf{adiab}\\ \textbf{link}\end{matrix}\right\}\textbf{,end} \\ \left\{\begin{matrix}\textbf{stemp}\\ \textbf{sflux}\end{matrix}\right\}, \left\{\begin{matrix}\textbf{const,}<\mathit{value}(r)>\\ \textbf{equ,}<\mathit{value}(i)>\\ \textbf{tabl,}<\mathit{value}(i)>\end{matrix}\right\}\textbf{,end} \\ \textbf{convec,}\left\{\begin{matrix}\textbf{const,}<\mathit{value}(r)>\\ \textbf{equ,}<\mathit{value}(i)>\\ \textbf{tabl,}<\mathit{value}(i)>\end{matrix}\right\}\textbf{,encl,}<\mathit{enclnbr}(i)>\textbf{,end}\end{matrix}\right]$$

This represents two statements, both of which are optional, but both must be included if properties within the segment are to vary with time and/or temperature. They define the boundary conditions on the left (**htrbcl**) and right (**htrbcr**) ends of the *thermseg/diffseg*. Separate statements are needed for each boundary, but they have the same five mutually exclusive format alternatives.

The first option, **adiab**, is for an adiabatic boundary. In that option the temperature gradient is forced to zero at the boundary, and there is no heat flux. This option like the *linked-segment* option discussed next requires no further data in this statement.

The next option, **link**, is for a linked-segment boundary. If that option is chosen, TMAP4 adjusts the node temperatures such that the heat flux from the left segment of the linked structure is identically equal to that entering the right segment (see Sections 2.4 and 3.2.3 for a discussion of heat flux at a linked boundary). Also if selected, the **hgap** statement described below is required.

The **stemp** boundary condition fixes the temperature at the left or right boundary to be specified by either a constant (*<value(r)>*), an equation (the one in position *<value(i)>*) which may only be a function of "time", or a lookup table (the one in position *<value(i)>*) which is a sequence of number pairs, the first being time and the second being temperature in kelvins. With this option, the temperature at the edge node is fixed, and the temperatures at adjacent nodes are determined by the thermal diffusivity of the material. This option only applies to a free (not *linked*) boundary.

The **sflux** boundary condition fixes the heat flux at the left or right boundary to be specified by either a constant (*<value(r)>*), an equation (the one in position *<value(i)>*) which may be a function of "time" or "temp", or a lookup table (the one in position *<value(i)>*) which is a sequence of number pairs, the first being time and the second being heat flux in the positive *x* direction at the surface. A positive heat flux at the left edge would be into the segment, whereas at the right edge it would be out of the segment. With this option, the temperatures are calculated dynamically in accordance with the heat conduction equation and other boundary conditions. This option only applies to a free (not *linked*) boundary.

The last option, **convec**, is for a thermal convective free (not *linked*) boundary. The first parameter entered is the thermal convective heat transfer coefficient, h, in W/m^2·K (see Section 3.2.2). The value for h may be specified by either a constant (<*value(r)*>), an equation (the one in position <*value(i)*>) which may be a function of "time" or "temp", or a lookup table (the one in position <*value(i)*>) which is a sequence of number pairs, the first being time and the second being the value of h. The keyword **encl** identifies the user-defined reference number (integer), <*value(i)*>, of the *enclosure* in which this convection is taking place. This is necessary because the temperature of that *enclosure* is required to calculate the rate of heat transfer. Again, positive heat flux is in the positive x direction.

$$[\textbf{hgap} = \left\{\begin{array}{l}\textbf{const,}<value(r)>\\ \textbf{equ,}<value(i)>\\ \textbf{tabl,}<value(i)>\end{array}\right\}\textbf{,end}]$$

This optional statement is only required if there is time-varying temperature and *linked-segment* structures have been defined. It specifies the gap conductance, h_{gap} (see Section 3.2.3) that characterizes heat flows between *linked* segments. The value for h_{gap} may be specified by either a constant (<*value(r)*>), an equation (the one in position <*value(i)*>) which may be a function of "time" and/or "temp", or a lookup table (the one in position <*value(i)*>) which is a sequence of number pairs, the first being **temperature** and the second being the value of h_{gap}. Note that only the left member of each *linked-segment* pair requires the **hgap** specification. The same value will automatically be applied to the left surface of the right member.

An example of a more complex *thermseg/diffseg* definition is for a pair of linked segments that represent half of a symmetric plate problem in which the surface coating is subject to volumetric heating in the outer (left-most) nodes. The left surface is convectively cooled by the fluid in enclosure 1. The right side of the first segment is linked to the left side of the second segment which represents half of the substrate. It has an adiabatic right boundary corresponding to the plane of symmetry for the problem. The node spacing in the first segment illustrates the advised practice of not making adjacent node spacings too disparate in size. Rather, when small surface nodes are needed near the surface, for example to model ion implantation, use a number of nodes with graded spacings.

Example:

```
thermal input
$ Segment 1
start thermseg,end
   delx=0.0,2*1.0e-8,2*1.0e-7,2*1.0e-6,2*1.0e-5,20*9.8880e-5,0.0,end
   dtemp=30*432.5,end$ The 30 nodes all start at 432.5 K.
   tcon=const,19.0,end
   rhocp=const,3.5958e6,end
   hsrc=tabl,4,srcpf,0.0,2*1.0,0.999,0.98,0.6,0.3,0.1,22*0.0,end
   htrbcl=convec,const,5930.,encl,1,end
   htrbcr=link,end
   hgap=const,1.0e6,end$ Set high for good thermal contact
$ Segment 2
start thermseg,end
   delx=0.0,10*1.2e-4,0.0,end$ This structure is 1.2 mm thick
   dtemp=12*432.5,end$ with 12 nodes all starting at 432.5K
   tcon=tabl,5,end$ Thermal conductivity is in Table 5
   rhocp=equ,3,end$ Thermal capacity is in Eq.3
   hsrc=const,0.0,srcpf,12*0.0,end
   htrbcl=link,end
   htrbcr=adiab,end$ This is the symmetry plane
end of thermal input
```

5.2.5 Diffusion Input

In this data block, the user specifies all additional information required to solve the mass-diffusion part of the problem. This information must be provided for each *thermseg/diffseg* identified (number of nodes and relative sequence number) in the **segnds** statement of the *main input* block and by the **thermseg** sub-blocks in the *thermal input* block. There must be a one-for one correspondence between the elements in these definitions. Each *thermseg/diffseg* is represented in the *diffusion input* data block by a separate sub-block of data.

Beside the **start** declaration statement for each sub-block, there are nine order-independent data statements of which two are optional. The format for the statements in these sub-blocks is as follows.

Format:

start diffseg [,end]

> Each sub-block begins with this declaration. Additional identification for each *thermseg/diffseg* specified here may be provided with partial-line comments (begins with "$") in this statement or full-line comments. The "**,end**" is optional in this statement.

dconc = *<dspc(α)$_1$>,<value(r)$_1$>,...,<value(r)$_{n1}$>*
<dspc(α)$_2$>,<value(r)$_1$>,...,<value(r)$_{n2}$>
⋮
<dspc(α)$_m$>,<value(r)$_1$>,...,<value(r)$_{nm}$>,**end**

or alternatively

dconc = *<dspc(α)$_1$>*,**const,***<value(r)$_1$>,<dspc(α)$_2$>*,**const,***<value(r)$_1$>*
...,*<dspc(α)$_m$>,<value(r)$_1$>,...,<value(r)$_{nm}$>*,**end**

> This statement defines the initial concentration (**dconc**) of mobile solute atom species in the diffusion segment. There must be an entry module, *<dspc(α)$_i$>*, *<value(r)$_1$>,...,<value(r)$_{ni}$>* or *<dspc(α)$_i$>*,**const,***<value(r)$_i$>*, for each diffusion species identified in the **dspcnme** statement of the *main input* data block. These modules may be entered in any order of user-designated diffusion species names, *<dspc(α)$_i$>*. TMAP4 checks to make sure there is an entry for each one. Within each module there must be a concentration value, *<value(r)$_i$>*, for each node in the diffusion segment including the two surface nodes. The shorthand entry format, ...*<numnds(i)>* <value(r)>*..., may be used for multiple nodes where **dconc** has the same value. The alternative formulation, **,const,***<value(r)$_i$>*, is equivalent if all the nodes in the *thermseg/diffseg* have the same initial concentration of that diffusion species.

ctrap = *<dspc(α)$_1$>,<value(r)$_1$>,...,<value(r)$_{n1}$>*
<dspc(α)$_2$>,<value(r)$_1$>,...,<value(r)$_{n2}$>
⋮
<dspc(α)$_m$>,<value(r)$_1$>,...,<value(r)$_{nm}$>,**end**

or alternatively

[**ctrap** = *<dspc(α)$_1$>*,**const,***<value(r)$_1$>,<dspc(α)$_2$>*,**const,***<value(r)$_1$>*
...,*<dspc(α)$_m$>,<value(r)$_1$>,...,<value(r)$_{nm}$>*,**end**]

This optional statement defines the initial concentration (**ctrap**) of trapped solute atom species in the diffusion segment. If it is omitted, the **trapping** statement below must also be omitted, and TMAP4 assumes that no trapping is operative. If included, then there must be an entry module, $<dspc(\alpha)_i>$, $<value(r)_1>$,..., $<value(r)_{ni}>$ or $<dspc(\alpha)_i>$,**const**,$<value(r)_i>$, for each diffusion species identified in the **dspcnme** statement of the *main input* data block. These modules may be entered in any order of user-designated diffusion species names, $<dspc(\alpha)_i>$. TMAP4 checks to make sure there is an entry for each one.

Within each module there must be a concentration value, $<value(r)_j>$, for each node in the diffusion segment including the two surface nodes. However, because the two surface nodes have zero volume, the value of **ctrap** for those nodes should be zero. If they are specified to be non-zero, TMAP4 changes them to zero. The shorthand entry format,...*<numnds(i)>* <value(r)>*..., may be used for multiple nodes where **dconc** has the same value. The alternative formulation, **,const,**$<value(r)_i>$, may also be used if all the nodes in the *thermseg/diffseg* have the same initial trapped atom concentration of that diffusion species.

[trapping = cetrpi,<*value(r)*>**,nbrden,**<*value(r)*>

$$<dspc(\alpha)_1>,\mathbf{alpht},\left\{\begin{array}{l}\mathbf{const},<value(r)>\\ \mathbf{equ},<value(i)>\\ \mathbf{tabl},<value(i)>\end{array}\right\},\mathbf{alphr},\left\{\begin{array}{l}\mathbf{const},<value(r)>\\ \mathbf{equ},<value(i)>\\ \mathbf{tabl},<value(i)>\end{array}\right\}$$

⋮

$$<dspc(\alpha)_n>,\mathbf{alpht},\left\{\begin{array}{l}\mathbf{const},<value(r)>\\ \mathbf{equ},<value(i)>\\ \mathbf{tabl},<value(i)>\end{array}\right\},\mathbf{alphr},\left\{\begin{array}{l}\mathbf{const},<value(r)>\\ \mathbf{equ},<value(i)>\\ \mathbf{tabl},<value(i)>\end{array}\right\}$$

end]

This is an optional statement that specifies the rate at which mobile solute atoms become trapped and trapped solute atoms are released to become mobile. If it is omitted, the **ctrap** statement above must also be omitted, and TMAP4 assumes that trapping is not operative. If included, then the **ctrap** statement must also be included.

The keyword **cetrpi** indicates that the following parameter, *<value(r)>*, is the fraction of the material lattice sites that are trapping sites. The next keyword, **nbrden**, is followed by the atom number density (atom/m^3) of the material of the *thermseg/diffseg*. Following that, there must be an entry module for each diffusion species identified in the **dspcnme** statement of the *main input* block. Each module consists of the user-specified diffusion species name, $<dspc(\alpha)_i>$; the keyword **alpht**, indicating that the following specification defines the frequency (s^{-1}) at which atoms of that species are trapped; and the keyword **alphr**, indicating that the next value is the frequency (s^{-1}) at which trapped atoms are released (see Eqs.(18) and (19) for a discussion of these parameters). In each case the value for **alpht** or **alphr** may be specified by either a constant (*<value(r)>*), an equation (the one in position *<value(i)>*) which may be a function of "time" or "temp", or a lookup table (the one in position *<value(i)>*) which is a sequence of number pairs, the first being **temperature** and the second being the value of **alpht** or **alphr**. The statement terminates with **,end**.

$$\mathbf{dcoef} = <\mathit{dspc}(\alpha)_1>, \left\{\begin{array}{l}\textbf{const},<\mathit{value}(r)>\\ \textbf{equ},<\mathit{value}(i)>\\ \textbf{tabl},<\mathit{value}(i)>\end{array}\right\},\ldots,$$

$$<\mathit{dspc}(\alpha)_n>, \left\{\begin{array}{l}\textbf{const},<\mathit{value}(r)>\\ \textbf{equ},<\mathit{value}(i)>\\ \textbf{tabl},<\mathit{value}(i)>\end{array}\right\}\textbf{,end}$$

This required statement specifies the diffusivity in m^2/s for each diffusion species, *<dspc(α)$_i$>* in the *thermseg/diffseg*. There must be an entry in this statement for each of the diffusion species identified in the **dspcnme** statement in the *main input* data block. The value for **dcoef** may be specified by either a constant (*<value(r)>*), an equation (the one in position *<value(i)>*) which may be a function of "time" or "temp", or a lookup table (the one in position *<value(i)>*) which is a sequence of number pairs, the first being **temperature** and the second being the value of diffusivity. While the specification applies to the entire *thermseg/diffseg*, the diffusivity may vary from node to node, if either an equation or a table is used to define it, because of temperature differences.

$$\mathbf{qstrdr} = <\mathit{dspc}(\alpha)_1>, \left\{\begin{array}{l}\textbf{const},<\mathit{value}(r)>\\ \textbf{equ},<\mathit{value}(i)>\\ \textbf{tabl},<\mathit{value}(i)>\end{array}\right\},\ldots,$$

$$<\mathit{dspc}(\alpha)_n>, \left\{\begin{array}{l}\textbf{const},<\mathit{value}(r)>\\ \textbf{equ},<\mathit{value}(i)>\\ \textbf{tabl},<\mathit{value}(i)>\end{array}\right\}\textbf{,end}$$

This required statement specifies the thermal transport parameter, Q^*/R, in kelvins for each diffusion species, *<dspc(α)$_i$>* in the *thermseg/diffseg*. There must be an entry in this statement for each of the diffusion species identified in the **dspcnme** statement in the *main input* data block. The value for **qstrdr** may be specified by either a constant (*<value(r)>*), an equation (the one in position

<*value(i)*>) which may be a function of "time" or local "temp", or a lookup table (the one in position <*value(i)*>) which is a sequence of number pairs, the first being local **temperature** and the second being the value of Q^*/R.

$$\textbf{spcsrc} = \langle dspc(\alpha)_1 \rangle \left\{ \begin{array}{l} \textbf{const},\langle value(r) \rangle \\ \textbf{equ},\langle value(i) \rangle \\ \textbf{tabl},\langle value(i) \rangle \end{array} \right\} \textbf{,srcpf},\langle value(r)_1 \rangle$$

$$,\ldots,\langle value(r)_{n1} \rangle$$

$$\vdots$$

$$\langle dspc(\alpha)_m \rangle \left\{ \begin{array}{l} \textbf{const},\langle value(r) \rangle \\ \textbf{equ},\langle value(i) \rangle \\ \textbf{tabl},\langle value(i) \rangle \end{array} \right\} \textbf{,srcpf},\langle value(r)_1 \rangle$$

$$,\ldots,\langle value(r)_{nm} \rangle$$

end

This required statement defines the rate at which solute atoms of each species are introduced (e.g., by implantation or transmutation) within each node segment. The specification consists of a data module for each of the diffusion species identified in the **dspcnme** statement of the *main input* block. There must be a module for each species. Each module consists of two parts. The first or temporal part identifies the species (<*dspc(α)$_i$*>) and the magnitude and time history of the local species production rate. It may be specified as a constant with a magnitude <*value(r)*>, an equation which is a function of "time", or as a lookup table. Conventions for identifying which equation or table is referred to are the same as with other entries in this block. In the second part, the keyword "**srcpf**" identifies the following data as the spatial distribution or source peaking factor profile. An entry must be made in this sequence for each node in the *thermseg/diffseg*. The shorthand entry format,...<*numnds(i)*>* <*value(r)*>..., may be used for multiple nodes where the **srcpf** has the same value.

The local species generation rate is calculated as the product of the temporal part (applies to the entire *thermseg/diffseg*) and the spatial part (an entry for each node). Hence, the magnitude of the species generation rate could be associated with either part or divided between the parts, at the users discretion.

$$\left\{ \begin{array}{l} \textbf{difbcl} \\ \textbf{difbcr} \end{array} \right\} = \left\{ \begin{array}{l} \textbf{ratedep},(\textit{ratedep parameters})\textbf{,end} \\ \textbf{lawdep},(\textit{lawdep parameters})\textbf{,end} \\ \textbf{sconc},(\textit{sconc parameters})\textbf{,end} \\ \textbf{link},(\textit{link parameters})\textbf{,end} \\ \textbf{nonflow,end} \end{array} \right\}$$

This pair of statements, **difbcl** and **difbcr** define the left and right diffusion boundary conditions. For each there is a selection of five different options that may be selected. Each option consists of a keyword and an associated parameter list. We now discuss these parameter lists in detail.

ratedep boundary:

$$\left\{\begin{array}{l}\textbf{difbcl}\\ \textbf{difbcr}\end{array}\right\} = \textbf{ratedep,encl,}<\mathit{enclnbr(i)}>$$

$$<\mathit{dspc(\alpha)_1}>,<\mathit{espc(\alpha)_1}>$$

$$\textbf{ksubd,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\},<\mathit{dspc(\alpha)_1}>,\textbf{ksubr}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}$$

$$\vdots$$

$$<\mathit{espc(\alpha)_m}>$$

$$\textbf{ksubd,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\},<\mathit{dspc(\alpha)_m}>,\textbf{ksubr}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}$$

$$\star\quad\star\quad\star$$

$$<\mathit{dspc(\alpha)_n}>,<\mathit{espc(\alpha)_1}>$$

$$\textbf{ksubd,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\},<\mathit{dspc(\alpha)_1}>,\textbf{ksubr}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}$$

$$\vdots$$

$$<\mathit{espc(\alpha)_m}>$$

$$\textbf{ksubd,}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\},<\mathit{dspc(\alpha)_m}>,\textbf{ksubr}\left\{\begin{array}{l}\textbf{const,}<\mathit{value(r)}>\\ \textbf{equ,}<\mathit{value(i)}>\\ \textbf{tabl,}<\mathit{value(i)}>\end{array}\right\}$$

end

This complex statement provides the specification for a *ratedep* boundary condition, one in which dissociation and recombination of solute atoms controls the rate of atom entry into the left or right surface of the *thermseg/diffseg*. The keyword, **encl**, after **ratedep** indicates that the next parameter value, $<\mathit{enclnbr(i)_l}>$, will be the user-defined integer reference number for the enclosure with which this *thermseg/diffseg* exchanges solute atoms. Because in general there will be a number of different diffusion species and enclosure species, it is necessary to identify for TMAP4 which ones are participating in the dissociation and recombination and how fast that process takes place.

An entry module is required for each diffusion species/enclosure species combination so participating. Hence, $<dspc(\alpha)_i>$ is the name of the "*ith*" diffusion species undergoing recombination, and $<espc(\alpha)_j>$ is the name of the "*jth*" enclosure species undergoing dissociation. If there are several molecular species in the enclosure that may dissociate to form the diffusion species (for example HTO, T_2, and HT may all dissociate to form T atoms), then there must be an entry module for each combination (HTO-T, T_2-T, and HT-T in this instance). A similar series of modules must be included for each of the *n* different diffusion species (H and T in the present example).

Within each module, the first entry is the enclosure species name, $<espc(\alpha)_j>$. It is followed by the keyword, **ksubd**, which indicates that the next parameter is the definition of the dissociation rate coefficient (see section 2.1) in units of molecules/m^2·s·Pa. It may be in either constant, equation, or table format. The equation may have "temp" and/or "time" as arguments, and the table lists **ksubd** as a function of temperature. This parameter is followed by the name, $<dspc(\alpha)_i>$, of the diffusion species that recombines with the one identified in the series header. It may be the same diffusion species name for homonuclear recombination, or it may be a different name for heteronuclear recombination. The keyword, **ksubr**, indicates that the following parameter will be the specification of the recombination coefficient, in units of molecules·m^4/atom^2·s. The same options exist for specifying **ksubr** as for **ksubd**.

Once a series for each $<dspc(\alpha)_i>$ is defined with a module in each series for each $<espc(\alpha)_j>$, the statement is terminated with the usual "**,end**".

Example:

```
difbcl=ratedep,encl,1
        t
        t2,ksubd,equ,23,t,ksubr,equ,24
        dt,ksubd,equ,25,d,ksubr,equ,26
        d
        d2,ksubd,equ,27,d,ksubr,equ,28
        dt,ksubd,equ,25,t,ksubr,equ,26,end
```

In this example the left face of the *thermseg/diffseg* exchanges solute atoms with enclosure 1. There are two diffusion species, "t" and "d" and three enclosure species, "t2", "d2", and "dt". The "t" recombines with itself and with "d" to form "t2" and "dt", respectively, while the solute "d" recombines with itself and with "t" to form "d2" and "dt". Notice that the recombination coefficient, **ksubr**, is the same for both "dt" formation reactions, and it is found in equation 26. The dissociation coefficients, **ksubd**, are also identical. However, because of the different isotopic masses, the two homonuclear dissociation and recombination reactions have different rate coefficients.

lawdep boundary:

$$\left\{\begin{matrix}\textbf{difbcl}\\ \textbf{difbcr}\end{matrix}\right\} = \textbf{lawdep,encl,}<\mathit{enclnbr(i)}>$$

$$<\mathit{dspc}(\alpha)_l>,<\mathit{espc}(\alpha)_l>,\textbf{pexp},<\mathit{value(r)}>,\textbf{solcon}\left\{\begin{matrix}\textbf{const},<\mathit{value(r)}>\\ \textbf{equ},<\mathit{value(i)}>\\ \textbf{tabl},<\mathit{value(i)}>\end{matrix}\right\}$$

⋮

$$<\mathit{dspc}(\alpha)_m>,<\mathit{espc}(\alpha)_m>,\textbf{pexp},<\mathit{value(r)}>,\textbf{solcon}\left\{\begin{matrix}\textbf{const},<\mathit{value(r)}>\\ \textbf{equ},<\mathit{value(i)}>\\ \textbf{tabl},<\mathit{value(i)}>\end{matrix}\right\}$$

end

In this option, the concentration of each diffusion species in the surface node of the *thermseg/diffseg* is controlled by a solution law of the form

$$C_{<\mathrm{dspc}(\alpha)>} = K_s\, P_{<\mathrm{espc}(\alpha)>}{}^{\mathrm{pexp}}$$

where $C_{<\mathrm{dspc}(\alpha)>}$ is the concentration of diffusion species *<dspc(α)>* in the structure, and $P_{<\mathrm{espc}(\alpha)>}$is the partial pressure of the *<espc(α)>* gas in enclosure *<enclnbr(i)>* (the user-specified enclosure reference number). A module, corresponding to one line in the above given format, must be provided for each diffusion species in this *thermseg/diffseg*. The keyword, **pexp**, identifies the following real value as the exponent of the solution law. The keyword, **solcon**, identifies the following specification as that for the solubility constant, K_s in the solution law.

WARNING...results will be unpredictable when more than one enclosure species contributes to a given diffusion species using this *lawdep* boundary condition. For that reason, it is advisable not to have more than one entry module if **pexp** ≠ 1.0.

The specification of **solcon** may be as a constant, as an equation, or as a lookup table. The equation may be a function of temperature (used as "temp" in the equation). The table consists of temperature -**solcon** pairs. Units for **solcon** are atoms/m^3 ·Pa$^{\mathrm{pexp}}$.

Example:

```
difbcl=lawdep,encl,2,t2d,t2,pexp,1.0,solcon,const,6.0e+19
              htd,ht,pexp,1.0,solcon,const,6.0e+19
              htod,hto,pexp,1.0,solcon,const,2.0e23
              h2od,h2o,pexp,1.0,solcon,const,2.0e23,end
```

In this example the *thermseg/diffseg* communicates on the left with enclosure 2. Because Henry's law applies, and all the diffusing species (indicated here with a terminal "d" in the diffusion species names) are related to the partial pressures of the corresponding gasses over the surface to the power 1.0, we can consider several simultaneous solubilities. In this case we consider the solution of "t2", "ht", "hto", and "h2o". Here all the solubilities are considered to be constants, one for the molecular hydrogen gas and another for the water species.

sconc boundary:

$$\left\{\begin{array}{l}\textbf{difbcl}\\ \textbf{difbcr}\end{array}\right\} = \textbf{sconc,encl,}<\textit{enclnbr(i)}>$$

$$<\textit{dspc}(\alpha)_l>,<\textit{espc}(\alpha)_l>\textbf{,nsurfs,}<\textit{value(i)}>\textbf{,conc,}\left\{\begin{array}{l}\textbf{const,}<\textit{value(r)}>\\ \textbf{equ,}<\textit{value(i)}>\\ \textbf{tabl,}<\textit{value(i)}>\end{array}\right\}$$

$$\vdots$$

$$<\textit{dspc}(\alpha)_m>,<\textit{espc}(\alpha)_m>\textbf{,nsurfs,}<\textit{value(i)}>\textbf{,conc,}\left\{\begin{array}{l}\textbf{const,}<\textit{value(r)}>\\ \textbf{equ,}<\textit{value(i)}>\\ \textbf{tabl,}<\textit{value(i)}>\end{array}\right\}$$

end

The keyword, **sconc**, indicates this boundary condition is for user-specified surface concentrations. Even though the diffusion species concentration is arbitrarily defined at this surface, the diffusion flux at the *thermseg/diffseg* surface results in a net flow of $<dspc(\alpha)_i>$ atoms as molecules of $<espc(\alpha)_j>$ into or out from the enclosure identified by the integer following the keyword, **encl**. The enclosure number, *<enclnbr(i)>*, is the user-specified reference number. The number of diffusion species atoms entering or leaving the enclosure per enclosure species molecule is specified by the integer, *<value(i)>*, following the keyword, **nsurfs**. For example, two T atoms would leave for each T_2 molecule formed. The keyword, **conc**, indicates that the next parameter is the specification for the concentration at the surface in units of atoms/m^3. That specification may be a constant, an equation that is a function of "time" and/or "temp", or a table of "time" - **conc** pairs.

There must be an entry module for each diffusion species in the *thermseg/diffseg*.

Example:

```
difbcr=sconc,encl,6
          t2d,t2,nsurfs,1,conc,tabl,3
          htd,ht,nsurfs,1,conc,tabl,3
          htod,hto,nsurfs,1,conc,equ,4
          h2od,h2o,nsurfs,1,conc,equ,4
          t,t2,nsurfs,2,conc,equ,5
          h,h2,nsurfs,2,conc,equ,5,end
```

In this example, the right side of the *thermseg/diffseg* interacts with enclosure 6. The diffusion species are "t2d", "htd", "htod", "h2od", "t", and "d". The corresponding enclosure species are "t2", "ht", "hto", "h2o", and "h2". Notice that the "t2" enclosure species contributes to two different diffusion species. Only the last two modules in which homonuclear enclosure molecules dissociate to form two identical atoms have the **nsurfs** parameter at 2 rather than 1. The **conc**, or concentration of the diffusion species at the right surface node, in the first two modules is specified by table 3 while the rest are defined by various equations.

Note here, too, that all the species that may be logically inferred to be present need not be included in the model. Obviously there could be some "h2" in the structure and some "t2o" in the enclosure and structure, but they have been neglected in the present problem.

link boundary:

$$\left\{\begin{matrix}\textbf{difbcl}\\ \textbf{difbcr}\end{matrix}\right\} = \textbf{link},<dspc(\alpha)_1>,\textbf{solcon},\left\{\begin{matrix}\textbf{const},<value(r)>\\ \textbf{equ},<value(i)>\\ \textbf{tabl},<value(i)>\end{matrix}\right\}$$

$$\vdots$$

$$<dspc(\alpha)_m>,\textbf{solcon},\left\{\begin{matrix}\textbf{const},<value(r)>\\ \textbf{equ},<value(i)>\\ \textbf{tabl},<value(i)>\end{matrix}\right\},\textbf{end}$$

The **link** specification relates the concentrations in two materials that are linked together in proportion to the ratio of their solubilities as described in Section 2.2. TMAP4 requires that the solubility constant, **solcon**, be defined for each diffusion species, $<dspc(\alpha)_i>$. A module, giving the diffusion species name, the keyword, **solcon**, and the solubility specification must be provided for each diffusion species in this *thermseg/diffseg*.

This specification is only used on *linked segment* boundaries as defined in the *main input* data block. It must be given in the appropriate sub-blocks for both sides of the linked boundary. For example, suppose *thermsegs/diffsegs* 1 and 2 were linked together in that order. Then in the *diffseg* sub-block for segment 1, the **difbcr** statement would use the **link** specification whereas in the sub-block for segment 2 it would be the **difbcl** statement that would use the **link** specification.

The units on **solcon** are assumed to be atom/m^3 ·Pa$^\epsilon$ where ϵ is an arbitrary value that is assumed to be the same on both sides of the *linked segment* interface. For this application only, because the ratio of the two solubilities is taken, the actual units may be different, provided they are the same on both sides.

Example:

```
difbcr=link
        t,solcon,equ,29
        d,solcon,equ,29,end
```

Here the solubilities for the two diffusion species "t" and "d" are the same. They are specified in equation 29.

nonflow boundary:

$$\left\{\begin{array}{l}\textbf{difbcl}\\ \textbf{difbcr}\end{array}\right\} = \textbf{nonflow,end}$$

This specification is the simplest of all the diffusion boundary condition specifications. No parameters other than the keyword, **nonflow**, are required.

surfa = <*value(r)*>,end

This statement defines the surface area in m^2 associated with each segment. *This value is used in calculating atom flows into enclosures.*

5.2.6 Equation Input

One of the very useful features of TMAP4 is the ability to define material properties, pressures, temperatures, etc. using algebraic equations. In this block, which must be included even if no equations are defined, the user may define up to 100 equations. Each equation must be in the form

y = *f* [time, temp, conc(<*species(i)*>)],end

where the "**y**" and the "**=**" are mandatory, *f* denotes a user-specified function, and the equation must be terminated by "**,end**" with no space after the comma. Arguments in the functions must be in the form shown, that is "**time**" (elapsed problem time in seconds), "**temp**" (local temperature in kelvins) or "**conc(*i*)**" (species concentration in particles/m^3) where "*i*" is the species relative sequence number). The species relative sequence number is the one associated with the order in which the species were defined in the **espcnme/dspcnme** statement in the *main input* data block.

Note that not all equations will accept all arguments. The applicability of the arguments to the various equation types is discussed in previous sections where this method of parameter specification was presented.

The order in which equations are listed determines the equation number. The first one listed is equation 1, the second equation 2, and so on.

Any FORTRAN-77 allowed equation structure is acceptable. All equation formulations may be continued on successive lines with no need of a continuation character, and the full 80-column width may be used. The "**,end**" terminates the equation definition. Partial line comments may be used after the "**,end**", or full-line comments may be used between equations to provide useful information about the equation to the user. Comments may not interrupt an equation defined on multiple lines.

Format:

equation input
(user defined equations and comments)
end of equation input

Example:

```
equation input
$ Pressure equations
 y = 2.76*(1-int(time-int(time/600.)*600)/400),end           $Eq.1 pres d2,t2
 y = 5.525*(1-int(time-int(time/600.)*600)/400),end          $Eq.2 pres dt
$ Thermal conductivity
 y = 47.985+(0.0275-1.2047E-5*temp)*temp,end                 $Eq.3 K for Nb
$ Temperature Equations
 y = 323.+2.2*(1-int(time-int(time/600.)*600)/400),end       $Eq.4 face 1 temp
 y = 323.+29.4*(1-int(time-int(time/600.)*600)/400),end      $Eq.5 face 3 temp
 y = 323.+63.3*(1-int(time-int(time/600.)*600)/400),end      $Eq.6 face 5 temp
 y = 323.+203.5*(1-int(time-int(time/600.)*600)/400),end     $Eq.7 face 7 temp
 y = 323.+151.4*(1-int(time-int(time/600.)*600)/400),end     $Eq.8 face 9 temp
 y = 323.+0.3*(1-int(time-int(time/600.)*600)/400),end       $Eq.9 face 11 temp
 y = 323.+0.3*(1-int(time-int(time/600.)*600)/400),end       $Eq.10 face 13 temp
 y = 323.+0.2*(1-int(time-int(time/600.)*600)/400),end       $Eq.11 face 15 temp
$ Diffusivity equations
 y = 4.1e-7*exp(-.39/(8.625e-5*temp))/sqrt(3.),end           $Eq.12 D for t in W
 y = 4.1e-7*exp(-.39/(8.625e-5*temp))/sqrt(2.),end           $Eq.13 D for d in W
 y = 8.5e-8*exp(-.15/(8.625e-5*temp))/sqrt(3.),end          $Eq.14 D for t in Nb
 y = 8.5e-8*exp(-.15/(8.625e-5*temp))/sqrt(2.),end          $Eq.15 D for d in Nb
$ Q* equations
 y = 27119.-19.39*temp,end                                 $Eq.16 Q* for t in Ta
 y = 24826.-16.39*temp,end                                 $Eq.17 Q* for d in Ta
 y = 3291.-2.405*temp,end                                  $Eq.18 Q* for t in Nb
 y = 2698.-2.405*temp,end                                  $Eq.19 Q* for d in Nb
$ Trap and release rate equations
 y = 6.491e12*exp(-.39/(8.625e-5*temp))/sqrt(2.5),end        $Eq.20 At for d/t in
W
 y = 1.e13*exp(-1.56/(8.625e-5*temp)),end                    $Eq.21 Ar for d/t in
W
$ Ion flux history equation
 y = 1.*(1-int(time-int(time/600.)*600)/400),end              $Eq.22 ion flux
$ Dissociation and recombination equations
 y = 1.0728e24/sqrt(temp)*exp(-16604.7/temp),end              $Eq.23 Kd t2/W
 y = 3.393697e-25*exp(7552.17/temp)/sqrt(temp),end            $Eq.24 Kr t/W
 y = 1.0728e24/sqrt(temp*5./6.)*exp(-16604.7/temp),end        $Eq.25 Kd dt/W
 y = 3.393697e-25*exp(7552.17/temp)/sqrt(temp*5./6.),end      $Eq.26 Kr dt/W
 y = 1.0728e24/sqrt(temp*4./6.)*exp(-16604.7/temp),end        $Eq.27 Kd d2/W
 y = 3.393697e-25*exp(7552.17/temp)/sqrt(temp*4./6.),end      $Eq.28 Kr d2/W
$ Solubility equations
 y = 1.76E24*exp(-1.04/(8.635E-5*temp)),end                   $Eq.29 S in W
 y = 6.3E24*exp(0.088/(8.635E-5*temp)),end                    $Eq.30 S in Nb
end of equation input
```

This rather extensive set of equations is associated with a problem where there are eight parallel diffusion structures each of which is a *linked segment* consisting of two *thermsegs/diffsegs*, one of tungsten and the other of niobium. Each left-hand segment has its own thermal history that is cyclic in nature at the left edge of the structure. The right edges are maintained at a constant temperature, but this information does not appear in this *equation input* data block. Pressure loadings are also cyclic and are different for two different enclosure species. In this case the pressure loadings are common to all the left faces of the *thermsegs/diffsegs*. Note that in the "Ion flux history equation" (for **spcsrc** statement), the number of equations was minimized by including only a unit value for intensity in the *equation input* block but including that data, which is different for each *thermseg/diffseg* in the **srcpf** part of the specification.

Example:

```
equation input        $Species indices 1-T2,2-HT,3-H2O,4-HTO
$Reaction rate equation for reaction  T2 + H2O -> HTO + HT  (Eq. 1)
  y=7.8e-29*conc(1)*(2.0*conc(1)+conc(2)+conc(4)),end
$reaction rate equation for reaction  HT + H2O -> HTO (+ H2)(Eq. 2)
  y=4.8e-29*conc(2)*(2.0*conc(1)+conc(2)+conc(4)),end
$tritium cleanup reaction  T2 -> HTO -> dryer               (Eq. 3)
  y=258.0*conc(1),end
$tritium cleanup reaction  HT -> HTO -> dryer               (Eq. 4)
  y=258.0*conc(2),end
$tritium cleanup reaction  H2O -> dryer                     (Eq. 5)
  y=258.0*conc(3),end
$tritium cleanup reaction  HTO -> dryer                     (Eq. 6)
  y=258.0*conc(4),end
end of equation input
```

This is an example using **conc(i)** concentrations as arguments in the equation specifications. All of these equations are reaction rate equations, the first two for oxidation reactions, and the last four for catalytic oxidation and/or desiccant absorption reactions.

5.2.7 Table Input

This section allows the user to supply any referenced table. Each table must have the form

$$x_1, y_1, x_2, y_2, \ldots, x_n, y_n, \mathrm{end}$$

where the x_i values correspond with the independent variable, usually "time" or "temp", and the y_i values are as defined in the statement that makes reference to the table. As with *equation input*, the order in which the tables are listed determines the table number for the referring statement. The first table listed is table 1, the second table 2, and so on. The independent values, the x_i, must be entered in ascending order and should not be duplicated. A linear interpolation subroutine is used to evaluate data between the tabulated points.

The maximum number of tables that may be defined is 10 with an aggregate of 240 data pairs.

Example:

```
table input
$  Flow from room into receiver  (Table 1)
  0.0,0.0,1800.0,0.0
          1801.0,0.02,7200.0,0.02
                         7200.1,0.0,9000.0,0.0
                                    9000.1,0.02,1.0e10,0.02,end
$  Flow from receiver into room  (Table 2)
  0.0,3.33e-4,1800.0,3.33e-4
          1801.0,0.020333,7200.0,0.020333
                         7200.1,3.33e-4,9000.0,3.33e-4
                                    9000.1,0.020333,1.0e10,0.020333,end
end of table input
```

In this table two **qflow** histories are defined. In each table, the first number of each pair is time in seconds. The second is the volumetric flow rate from the indicated enclosure in m^3/s. Notice that spacing within each table is arbitrary and that no comma separates data items on successive lines. Also, there may be no blank lines.

5.2.8 Control Input

This data block supplies TMAP4 with the input required to control problem execution. Six order-independent control parameter statements are required in addition to the starting and ending declarations.

Format:

time = <*tstart(r)*>,end

The starting time for the problem is given in seconds. Normally this will be 0.0.

tstep = *<timestep(r)>*,end

This specifies the increment for advancing time during calculations, also in seconds. Normally, small time steps give more accurate results, but because of the time-backward or implicit nature of the finite differencing, this is usually not critical for convergence or stability.

timend = *<endtime(r)>*,end

This is the problem time in seconds at which the user wants the calculations to stop.

nprint = *<ncyc(i)>*,end

This is the output edit controller. An output edit, consisting of all the calculated temperatures, mobile concentrations, trapped concentrations, enclosure species concentrations, etc., is generated in the output file after every *<ncyc(i)>* computational cycles. That means that at time intervals of *<ncyc(i)>** *<timestep(r)>* seconds, an output edit will be generated.

itermx = *<iter(i)>*,end

This parameter specifies the maximum number of iterations allowed on any computational cycle. If more than *<iter(i)>* iterations are required for convergence, a listing of the involved state variables, Newton-Rhapson functions, and their derivatives over the last three iterations is posted to the output file, and the problem stops.

delcmx = *<tol(r)>*,end

This parameter specifies the convergence tolerance to be used as a criterion for terminating the convergence iterations. It is a fractional number representing the acceptable change in a quantities magnitude from one cycle to the next divided by that magnitude.

Each of these parameters requires only a single input in integer or floating-point (real) format, as indicated by the symbol in parentheses in the input variable given in each statement.

5.2.9 Plot Input

When TMAP4 executes, it generates a special data file called **PLTDATA** (later changed by **TMAP4.BAT** to a user-specified file name with a **.PLT** suffix) that contains information that can be extracted using the **EXTRACT.EXE** program to generate tables for plotting. The determination of what data to record in this file and at what frequency during the calculations is provided in the *plot input* data block. There are seven statements in this block, but their order of entry is arbitrary.

Use of this block is optional; however, the block declaration statements, **plot input** and **end of plot input** are required, and if any parameter statements are included in the block, all seven must be included. Empty value lists are allowed on some statements if no inputs for that statement are desired.

Format:

[nplot = *<cyc(i)>*,**end]**

This statement controls the frequency at which data are recorded in the **PLTDATA** file. The entry, *<cyc(i)>*, is an integer specifying the number of computational cycles between file writes. Data is written at intervals of *<cyc(i)>****tstep**. If *<cyc(i)>* is zero no plot data will be generated.

[plotseg = [<*seg(i)*$_1$>,...,<*seg(i)*$_n$>,]**end]**

This statement lists integers identifying the *thermsegs/diffsegs* for which plot data should be recorded. These numbers are the relative sequence nu nbers corresponding to the order in which the segments were defined in the **segnds** statement of the *main input* data block and in the *diffusion input* data block. Data will be recorded in the **PLTDATA** file for each *thermseg/diffseg* identified here. If no segments are specified, no data will be generated for *thermsegs/diffsegs*. While the sequence of entering the relative sequence numbers identifying the segments is not important, the order of writing data to the **PLTDATA** file is in ascending order of the relative sequence numbers in this statement.

[plotencl = [<*encl(i)*$_1$>,...,<*encl(i)*$_n$>,]**end]**

This statement identifies *enclosures* for data recording in exactly the same way as **plotseg** identifies *thermsegs/diffsegs*.

[dname = [<*dspc*(α)$_1$>,...,<*dspc*(α)$_n$>,]end]

Here the diffusion species are identified for which plot data is required. Each <*dspc*(α)$_i$> is a character name assigned in the **dspcnme** statement of the *main input* data block. If no names are listed here, no data for diffusion species will be written to the **PLTDATA** file. Data is written only for the diffusion species names listed and only for the *thermsegs/diffsegs* identified in the **plotseg** statement.

[ename = [<*espc*(α)$_1$>,...,<*espc*(α)$_n$>,]end]

In this statement the enclosure species are identified for which plot data is required. Each <*espc*(α)$_i$> is a character name assigned in the **espcnme** statement of the *main input* data block. Again, if no names are listed here, no data for enclosure species will be written to the **PLTDATA** file. Data is written only for the enclosure species names listed and only for the *enclosures* identified in the **plotseg** statement.

[dplot = [stemp,][sconc,][moblinv,][trapinv,][sflux,]end]

Five different types of information concerning each *thermseg/diffseg* identified in the **plotseg** statement may be written to the **PLTDATA** file. These are:

stemp surface temperatures in kelvins (both left and right)

sconc surface concentrations (number/m^3) of the diffusion species identified in the **dname** statement (both left and right)

moblinv the mobile inventory of each diffusion species identified in the **dname** statement for the whole *thermseg/diffseg* in units of number/m^2

trapinv the trapped inventory of each diffusion species identified in the **dname** statement for the whole *thermseg/diffseg* in units of number/m^2

sflux diffusive fluxes across left and right surfaces of each diffusion species identified in the **dname** statement in units of number/m^2 ·s (positive direction is from left to right)

[eplot = [conc,][conv,][diff,]end]

Three different types of information concerning each *enclosure* identified in the **plotencl** statement may be written to the **PLTDATA** file. These are:

conc concentration (number/m^3) of each enclosure species identified in the **ename** statement

conv time-integrated flow (number of particles) of each enclosure species identified in the **ename** statement by convection from or to other enclosures

diff time-integrated flow (number of particles) of each enclosure species identified in the **ename** statement by diffusion into or out from all *thermsegs/diffsegs* connected to the enclosure

If no options are specified, no data is written to the **PLTDATA** file regarding the enclosure species.

5.2.10 Terminal Declaration

The last entry in the input file is the terminal declaration

end of data

6.0 EXTRACT UTILITY

EXTRACT is an interactive TMAP4 utility program that reads an existing **PLTDATA** file (renamed to ***.PLT** where * is a user-defined root name). It allows the user to individually extract parameters from that file as tables of "time" - parameter number pairs for plotting. While the filename furnished by the user for **EXTRACT** to operate on is optional within DOS environment requirements for filename definition, the file must be a **PLTDATA** file generated by TMAP4. Coded header statements in such a file allow **EXTRACT** to assess the amount and availability of TMAP4 data in the file, and they provide a means of interfacing with the user. Computational data in a **PLTDATA** file are selected on the basis of choices made by the user during **EXTRACT** execution. An output file with a user-defined name will be generated that consists of two columns of numbers. The first one is problem elapsed "time" in seconds. The second is the value of the parameter selected.

6.1 Program Usage

The **EXTRACT** program is executed by simply typing the word

```
extract
```

upon which the user will be asked to furnish the name of the **PLTDATA** or ***.PLT** file used as input for **EXTRACT.** Next, the user will be asked to specify the name of the plot data file to be generated. Any filename acceptable to DOS or the current operating system may be used. A number of options will then be presented for selection of the data desired for the specified plot data file. The options presented will depend on the parameters identified in the *plot input* block in the TMAP4 input file. Note that *thermseg/diffseg* and *enclosure* numbers as well as names of enclosure or diffusion species are selected by relative sequence number, not by name or user-defined reference number. These relative sequence numbers are listed on the screen together with the names or user-defined reference numbers at the time selection is to be made. When the selection is complete, the required data are written to the plot data file, and the program terminates.

6.2 PLTDATA File Structure

We provide here some information regarding the structure of the **PLTDATA** file that may be useful if it is necessary to work with a file of this type outside of **EXTRACT**. The file consists of several file header statements and sequential fixed-format data blocks. The file header statements consist of coded numbers, the original problem title, and user-specified species names which are used during an **EXTRACT** procedure. The following sequential fixed-format structure applies to all written data subject to its being requested in the *plot input* data block:

```
transient time
{loop over requested thermsegs/diffsegs}:
    left and right surface temperatures
    {loop over all requested diffusion species}:
        left and right surface concentrations
        mobile inventory
        left and right surface diffusive fluxes
        trapped inventory
{loop over requested enclosures}:
    {loop over all requested enclosure species}:
        species concentration
        integrated convective flow
        integrated diffusive flow
```

It is important to note that this order for data listing is assumed for all *thermsegs/diffsegs*, diffusion species, enclosures, and enclosure species. This overall order is based on the original user input order. For example, suppose four diffusion species names were defined in the **dspcnme** statement of the *main input* block in the problem input file in the order listed: T2, HTO, H2O, and HT. Yet, the user selected only HT and HTO, in that order, in the **dname** statement of the *plot input* data block (**dname = HT,HTO,end**). The order in which the data for these species would appear in the **PLTDATA** file is first

HTO and then HT, in ascending order of their relative sequence number. Other types of data follow the same rule.

REFERENCES

1. H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, John Wiley & Sons, New York, 1985, p.313.

2. F. J. Blatt, *Physics of Electronic Conduction in Solids*, McGraw-Hill, New York, 1968, pp. 202-207.

INDEX

APPENDIX A

SAMPLE PROBLEM No. 1

This problem is the simulation of experimental results obtained at the INEL in 1985 and published as, R. A. Anderl, D. F. Holland, D. A. Struttmann, G. R. Longhurst, and B. J. Merrill, "Tritium Permeation in Stainless Steel Structures Exposed to Plasma Ions," Proceedings, Eleventh Symposium on Fusion Engineering, November 18-22, 1985, Austin, TX, Vol I, pp. 644-649, IEEE Cat. No. CH2251-7. The experiment involved applying an ion beam to a 2.5-cm diameter, 0.5-mm thick sample of a modified 316 stainless steel called Primary Candidate Alloy (PCA). Details of the experiment and the means of evaluating the necessary transport parameters to get a good fit between TMAP results and the experimental data are given in the publication. The TRIM code was used to determine that the average implantation depth for the ions was 11 μm ± 5.4 μm. Reemission data from the TRIM calculation showed that only 75% of the incident flux remained in the metal. The other 25% was re-emitted.

The input code listing for the TMAP simulation is given below. One known non-physical feature in the modelling is that the cleanup of the upstream surface was modelled by a simple exponential in time rather than in ion fluence which was interrupted twice during the actual experiment. The pressures upstream and downstream could have been taken as zero and obtained essentially the same results.

SAMPLE PROBLEM No.1 INPUT FILE

```
title input
SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA
end of title input
$
main input
  dspcnme=d,end
  espcnme=d2,end
```

```
   segnds=21,end
   nbrencl=2,end
 end of main input
 $
 enclosure input
 $
 start bdry,1,end                                   $Implantation side
   etemp=703.,end
   esppres=d2,tabl,1,end
 start bdry,2,end                                   $Downstream side
   etemp=703.,end
   esppres=d2,const,2.e-6,end
 end of enclosure input
 $
 thermal input
 start thermseg,end
   delx=0.0,5*4.0e-9,1.0e-8,1.0e-7,1.0e-6,1.0e-5,10*4.88e-5,0.0,end
   dtemp=21*703.,end
 end of thermal input
 $
 diffusion input
 start diffseg,end
   dconc=d,21*0.0,end
   dcoef=d,const,3.0e-10,end                        $Diffusivity (m2/s)
   qstrdr=d,const,0.0,end                           $No temperature gradient
   spcsrc=d,tabl,2,srcpf,3*0.0,0.25,1.0,0.25,15*0.0,end
   difbcl=ratedep,encl,1,d
          d2,ksubd,equ,1,d,ksubr,equ,2,end
   difbcr=ratedep,encl,2,d
          d2,ksubd,const,1.7918e15,d,ksubr,const,2.0e-31,end
   surfa=1.0,end                                    $Answers will be d/m^2
 end of diffusion input
 $
 equation input
 $ Dissociation constant (d_2/m^2.s.Pa^1/2)
  y = 8.959e18*(1.0-0.9998*exp(-6.0e-5*time)),end $Eq.1
 $ Recombination constant (m^4/d_2.s)
  y = 1.0e-27*(1.0-0.9998*exp(-6.0e-5*time)),end $Eq.2
 end of equation input
 $
 table input
 $ Upstream enclosure pressure history
  0.0,4.0e-5,5850.0,4.0e-5,5850.1,9.0e-6,9010.0,9.0e-6,9010.1,4.0e-5
  12200.0,4.0e-5,12200.1,9.0e-6,14600.0,1.9e-6,14600.1,4.0e-5,17220.0
  4.0e-5,17220.1,9.0e-6,1.0e10,9.0e-6,end          $Table 1
 $ Implantation flux (d/m2s)
  0.0,4.9e19,5850.0,4.9e19,5850.1,0.0,9010.0,0.0,9010.1,4.9e19
  12200.0,4.9e19,12200.1,0.0,14600.0,0.0,14600.1,4.9e19,17220.0
  4.9e19,17220.1,0.0,1.0e10,0.0,end                $Table 2
 end of table input
 $
 control input
```

```
  time=0.0,end
  tstep=20.0,end                                          $ Seconds
  timend=19200.,end                                       $ 320 minutes for problem
  nprint=60,end                                           $ Print out every 20 min
  itermx=90,end
  delcmx=1.e-8,end
 end of control input
 $
 plot input
  nplot=3,end                                             $ Plot points every minute
  plotseg=1,end
  plotencl=1,2,end
  dname=d,end
  ename=d2,end
  dplot=moblinv,sflux,end
  eplot=end
 end of plot input
 end of data
```

PARTIAL OUTPUT DATA LISTING

```
 >>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
 SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA


   Problem Time=  0.0000E+00 seconds     Time step=  2.0000E+01 seconds   Iterations =     0
 Diffusion segment (  1)    Segment Iterations=    0  Left Enclosure =    1  Right Enclosure =    2
    Segment Temperatures (K)
          7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02
          7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02 7.0300E+02
          7.0300E+02 7.0300E+02 7.0300E+02
      Specie= d
         Mobile Concentration (number/m**3)
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00
         Mobile inventory =  0.0000E+00 number/m**2
         Flux: Left =  0.0000E+00 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)
 Enclosure (  1)
    Specie= d2              Concentration (number/m**3)=  0.0000E+00        Pressure (Pa)=  0.0000E+00
      Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
 Enclosure (  2)
    Specie= d2              Concentration (number/m**3)=  2.0616E+14        Pressure (Pa)=  2.0000E-06
      Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
1
 >>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
 SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA


   Problem Time=  1.2000E+03 seconds     Time step=  2.0000E+01 seconds   Iterations =     1
 Diffusion segment (  1)    Segment Iterations=    3  Left Enclosure =    1  Right Enclosure =    2
      Specie= d
         Mobile Concentration (number/m**3)
          7.2638E+23 7.2687E+23 7.2785E+23 7.2883E+23 7.2965E+23 7.2981E+23 7.2981E+23 7.2981E+23 7.2982E+23
          7.2988E+23 7.2994E+23 7.2762E+23 7.2281E+23 7.1548E+23 7.0560E+23 6.9319E+23 6.7827E+23 6.6093E+23
          6.4125E+23 6.1933E+23 6.0733E+23
```

```
        Mobile inventory =  3.4457E+20 number/m**2
        Flux: Left = -7.3504E+19 number/(s-m**2)        Right =  1.4754E+17 number/(s-m**2)
Enclosure (  1)
   Specie= d2              Concentration (number/m**3)=  4.1231E+15         Pressure (Pa)=  4.0000E-05
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  4.3802E+22
Enclosure (  2)
   Specie= d2              Concentration (number/m**3)=  2.0616E+14         Pressure (Pa)=  2.0000E-06
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  1.2619E+20
1
>>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA


  Problem Time=  2.4000E+03 seconds      Time step=  2.0000E+01 seconds    Iterations =      1
Diffusion segment (  1)    Segment Iterations=     3  Left Enclosure =     1  Right Enclosure =     2
     Specie= d
        Mobile Concentration (number/m**3)
          5.2303E+23 5.2352E+23 5.2450E+23 5.2548E+23 5.2630E+23 5.2646E+23 5.2646E+23 5.2646E+23 5.2640E+23
          5.2588E+23 5.2298E+23 5.1734E+23 5.1090E+23 5.0364E+23 4.9559E+23 4.8674E+23 4.7712E+23 4.6675E+23
          4.5565E+23 4.4385E+23 4.3762E+23
        Mobile inventory =  2.4403E+20 number/m**2
        Flux: Left = -7.3472E+19 number/(s-m**2)        Right =  7.6605E+16 number/(s-m**2)
Enclosure (  1)
   Specie= d2              Concentration (number/m**3)=  4.1231E+15         Pressure (Pa)=  4.0000E-05
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  8.7890E+22
Enclosure (  2)
   Specie= d2              Concentration (number/m**3)=  2.0616E+14         Pressure (Pa)=  2.0000E-06
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  1.8797E+20
1
>>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
SAMPLE PROBLEM No.1 - Plasma driven permeation of PCA


  Problem Time=  3.6000E+03 seconds      Time step=  2.0000E+01 seconds    Iterations =      1
Diffusion segment (  1)    Segment Iterations=     3  Left Enclosure =     1  Right Enclosure =     2
     Specie= d
        Mobile Concentration (number/m**3)
          4.3468E+23 4.3517E+23 4.3615E+23 4.3713E+23 4.3794E+23 4.3810E+23 4.3810E+23 4.3810E+23 4.3804E+23
          4.3751E+23 4.3459E+23 4.2930E+23 4.2360E+23 4.1747E+23 4.1093E+23 4.0398E+23 3.9664E+23 3.8891E+23
          3.8080E+23 3.7234E+23 3.6793E+23
        Mobile inventory =  2.0293E+20 number/m**2
        Flux: Left = -7.3471E+19 number/(s-m**2)        Right =  5.4150E+16 number/(s-m**2)
Enclosure (  1)
   Specie= d2              Concentration (number/m**3)=  4.1231E+15         Pressure (Pa)=  4.0000E-05
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  1.3197E+23
Enclosure (  2)
   Specie= d2              Concentration (number/m**3)=  2.0616E+14         Pressure (Pa)=  2.0000E-06
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  2.2617E+20
```

This problem had a **PLTDATA** file generated in which surface fluxes were among the quantities recorded for plotting. After running the **extract** utility, selecting diffusion data, and surface fluxes for diffusion species "**d**", the plot of Figure A-1 was generated. Actual experimental data are also shown on the figure. They were very closely approximated by the calculated permeation.

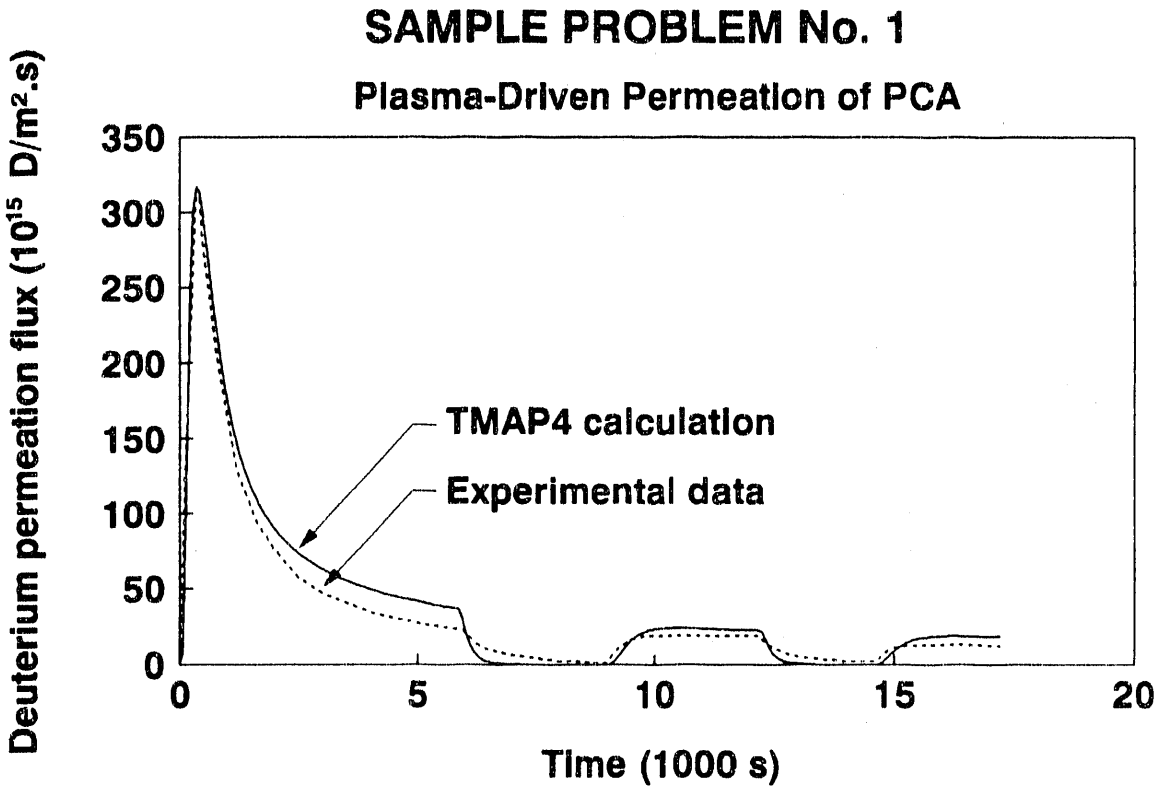


Figure 1. Computed permeation transient which is essentially a duplicate of experimental data.

APPENDIX B

SAMPLE PROBLEM No. 2

This problem is taken from work done by R. G. Macaulay-Newcombe at McMaster University. He and co-workers conducted thermal absorption and desorption experiments, as well as implantation experiments, on wafers of polished beryllium. The work was published as: R. G. Macaulay-Newcombe, D. A. Thompson, and W. W. Smeltzer, "Deuterium diffusion, trapping and release in ion-implanted beryllium," *Fusion Engineering and Design 8* (1991) 419. Of the several data sets presented, the one modelled here is that represented in Figure 2 (a) in that publication. The beryllium was 0.4-mm thick and had an area of 104 mm^2. It was polished to a mirror finish then exposed to 13.3 KPa of deuterium at 773 K for 50 minutes. It was quickly cooled under a vacuum of about 1 μPa. The cooling time constant for the apparatus was taken as 45 minutes. After removing the sample from the charging furnace, it was transferred in the air to a thermal desorption furnace where the temperature was increased from ambient (300 K) to 1073 K at the rate of 3 K/min. This was done under vacuum, and the pressure of the chamber was monitored by residual gas analysis and calibrated against standard leaks. In that way the emission rate from the sample could be measured as a function of temperature. Data from that measurement, given in Figure 2 (a) of their paper are reproduced in Figure B-1. From Rutherford backscattering measurements made on the samples before charging with deuterium they deduced that the thickness of the oxide film was 18 nm. This is typical for polished beryllium. The metal is so reactive in air that the film forms almost immediately after any surface oxide removal. On the other hand, it is relatively stable and would only grow slightly when exposed to air between charging and thermal desorption.

This experiment was modelled using a two-segment model in TMAP with the segments linked. The first was the BeO film, which was modelled using 18 equally spaced nodes of 1 nm each plus the two surface nodes. The second segment was a half-thickness wafer of beryllium with reflective boundary conditions at the midplane. It was made up of 15 segments of varying thickness to accommodate solution stiffness plus the two surface nodes. The solubility of deuterium in beryllium used was that given by K. L. Wilson, et al., "Beryllium--a better tokamak plasma-facing material?" *Journal of Vacuum Science and Technology A 8 (3)* (May/June-1990) 1750, based on work done by W. A. Swansiger, also of Sandia National Laboratory. The diffusivity of deuterium in beryllium was measured by E. Abramov, et al., "Deuterium Permeation and Diffusion in High Purity Beryllium," CFFTP-G-9013, May 1990. They made measurements on high-grade (99% pure) and extra grade (99.8% pure). The values for high-grade, consistent with Dr. Macaulay-Newcombe's measurements of the purity of his samples.

Deuterium transport properties for the BeO were more challenging. First, it is not clear in what state the deuterium exists in the BeO. However, it has been observed (G. R. Longhurst, "Tritium Behavior in ITER Beryllium," EGG-FSP-9304, October 1990) an activation energy of -78 kJ/mole (exothermic solution) for tritium coming out of neutron irradiated beryllium in work by D. L. Baldwin of Battelle Pacific Northwest Laboratory. The same energy has appeared in other results (can be inferred from Dr. Swansiger's work cited by Wilson, et al., above and by R. A. Causey, et al., "Tritium reter. .n and migration in beryllium," *Journal of Nuclear Materials*, 176 & 177 (1990) 654, among others), so one may be justified in using it. The solubility coefficient is not well known. Measurements reported by R. G. Macaulay-Newcombe, et al, "Thermal Absorption and Desorption of Deuterium in Beryllium and Beryllium Oxide," Fifth International Conference on Fusion Reactor Materials, Clearwater, FL, November 20-24, 1991, and in followup telephone conversations indicate about 200 appm of D in BeO after exposure to 13.3 kPa of D_2 at 773 K. That suggests a coefficient of only 1.88×10^{18} d/m^3.Pa$^{1/2}$. On the other hand, the integrated area under the curve in the referenced experimental data is 1.1×10^{14} d/mm^2 which implies a solubility coefficient of 2.85×10^{20} d/m^3.Pa$^{1/2}$. Since much of the deuterium in the oxide layer will get out during the cool-down process (and because it gives a good fit) the solubility coefficient was taken to be 5×10^{20} d/m^3.Pa$^{1/2}$.

Deuterium diffusion measurements in BeO were made by J. D. Fowler, et al., "Tritium Diffusion in Al_2O_3 and BeO," *Journal of the American Ceramic Society, 60 (3-4)* (Mar/Apr 1977) 155. They found a wide range of results for diffusivity in BeO, depending on the physical form of the material, having measured it for single-crystal, sintered, and powdered BeO. This model used one expression for the charging phase and another for the thermal desorption phase, believing that the surface film would change somewhat during the transfer between the two furnaces. For the charging phase the model used 20 times that for the sintered BeO. Thermal expansion mismatches tend to open up cracks and channels in the oxide layer, so this seems a reasonable value. The same activation energy of 48.5 kJ/mole, was retained, however. For the thermal desorption phase, the prefactor of the sintered material ($7x10^{-5}$ m^2/s) and an activation energy of 223.7 kJ/mole (53.45 kcal/mole) were used. These values gave good results and lie well within the scatter of Fowler's data. Exposure of the sample to air after heating should tend to make the oxide more like single crystal by healing the cracks that may have developed.

The model applies 13.3 kPa of D_2 for 50 hours followed by evacuation to 1 μPa and cool down with a 45 minute time constant for one hour. The deuterium concentrations in the sample are of a complex distribution that results from first charging the sample and then discharging it during the cooldown. This problem is then restarted with different equations to simulate thermal desorption in the 1-μPa environment. That begins at 300 K and goes to 1073 K. Again, the concentration profiles in both the substrate beryllium and the oxide film have a peculiar interaction because of the activation energies involved, but the flux exuding from the sample when doubled to account for the two sides of the specimen in the laboratory gives a good fit to the experimental data.

INPUT FILE FOR SPECIMEN CHARGING

```
title input
R. G. Macaulay-Newcombe's thermal charging problem for gas absorption into
```

```
     a wafer of polished beryllium with a thin oxide film.
end of title input
$
main input
  dspcnme=d,end
  espcnme=d2,end
  segnds =20,17,end                           $ Oxide has 20, Be has 17
  nbrencl=1,end
  linksegs=1,2,end                            $ The oxide and Be are joined
end of main input
$
enclosure input
$
start bdry,1,end                              $ Outside of sample
  etemp=773.00,end
  esppres=d2,equ,6,end                        $ Pressure history in Eq.6
end of enclosure input
$
thermal input
$ Segment 1 - BeO film
start thermseg,end
   delx=0.0,18*1.0e-9,0.0,end
   dtemp=20*773.0,end
   tcon=const,159.2,end
   rhocp=const,3.0e6,end
   hsrc=const,0.0,srcpf,20*0.0,end
   htrbcl=stemp,equ,1,end                    $ Temperature history in Eq.1
   htrbcr=link,end
   hgap=const,1.e6,end
$ Segment 2 - Be metal - half thick
start thermseg,end
   delx=0.0,1.0e-9,1.e-8,1.e-7,1.e-6,1.e-5,10*1.888e-5,0.0,end
   dtemp=17*773.0,end
   tcon=const,168.0,end
   rhocp=const,3.37e6,end
   hsrc=const,0.0,srcpf,17*0.0,end
   htrbcl=link,end
   htrbcr=adiab,end
end of thermal input
$
diffusion input
$ Seg No. 1  BeO film
start diffseg,end
  dconc=d,20*0.0,end
  dcoef=d,equ,2,end                             $ D for d in BeO in Eq.2
  qstrdr=d,const,0.0,end                        $ Q* of no consequence
  spcsrc=d,const,0.0,srcpf,20*0.0,end
  difbcl=lawdep,encl,1
         d
         d2,pexp,0.5,solcon,equ,3,end
  difbcr=link,d,solcon,equ,3,end                $ Sol. for D in BeO in Eq.3
  surfa=1.04e-4,end
```

```
$ Seg No. 2  Be foil - half thickness
start diffseg,end
  dconc=d,17*0.0,end
  dcoef=d,equ,4,end                            $ D for d in Be in Eq.4
  qstrdr=d,const,0.0,end                       $ Q* of no consequence
  spcsrc=d,const,0.0,srcpf,17*0.0,end
  difbcl=link,d,solcon,equ,5,end               $ Sol. for D in Be in Eq.5
  difbcr=nonflow,end                           $ Midplane of foil - no flow
  surfa=1.04e-4,end
end of diffusion input
$
equation input
$ Temperature history equation
 y = 773.-int(time/180000.)*(1-exp(-(time-180000.)/2700.))*475.,end $Eq.1 Temp
$ Diffusion and solubility equations
 y = 1.40e-4*exp(-24408./temp),end             $Eq.2 D of d in BeO (Fowler)
 y = 2.85e20*exp(9377.7/temp),end              $Eq.3 S for d in BeO
 y = 8.0e-9*exp(-4220./temp),end               $Eq.4 D of d in Be (Abramov)
 y = 7.156e27*exp(-11606./temp),end            $Eq.5 S for d in Be (Swansiger)
$ Pressure history equation
 y = 13300.0*(1-int(time/180010.))+1.0e-6,end  $Eq.6 Pressure history
end of equation input
$
table input
end of table input
$
control input
 time=0.0,end                                  $ Seconds
 tstep=60.0,end                                $ One minute sampling
 timend=182400.,end                            $ Time to 800 C
 nprint=90,end                                 $ Cycles
 itermx=90,end
 delcmx=1.e-8,end
end of control input
$
plot input
 nplot=1,end                                   $ Cycles
 plotseg=1,2,end
 plotencl=end
 dname=d,end
 ename=end
 dplot=moblinv,sflux,end
 eplot=end
end of plot input
end of data
```

INPUT FILE FOR THERMAL DESORPTION

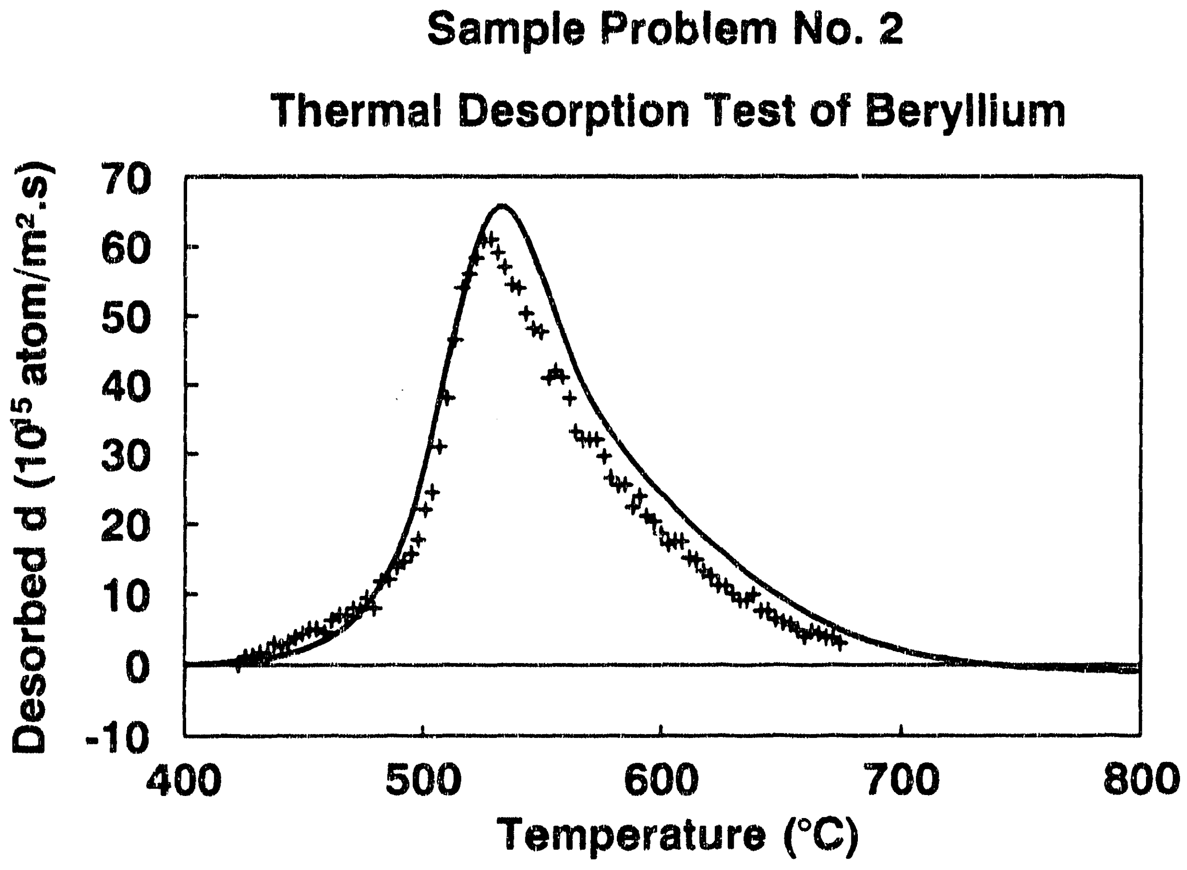
```
restart
$
equation input
$ Temperature history equation
 y = 300.0+0.05*time,end                       $Eq.1 Heat rate 3 K/min
$ Diffusion and solubility equations
 y = 7.00e-5*exp(-27000./temp),end             $Eq.2 D of d in BeO (Fowler 2)
 y = 5.00e20*exp(9377.7/temp),end              $Eq.3 S for d in BeO
 y = 8.0e-9*exp(-4220./temp),end               $Eq.4 D of d in Be (Abramov Be-2)
 y = 7.156e27*exp(-11606./temp),end            $Eq.5 S for d in Be (Swansiger)
$ Pressure history equation
 y = 0.001,end                                 $Eq.6 Pressure history
end of equation input
$
table input
end of table input
$
control input
 time=0.0,end
 tstep=60.0,end                                $ One minute sampling
 timend=15460.,end                             $ Time to 800 C
 nprint=10,end                                 $ Cycles
 itermx=90,end
 delcmx=1.e-8,end
end of control input
end of data
```

PARTIAL OUTPUT FILE LISTING

```
>>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
R. G. Macaulay-Newcombe's thermal charging problem for gas absorption into
     a wafer of polished beryllium with a thin oxide film.


  Problem Time=  0.0000E+00 seconds     Time step=  6.0000E+01 seconds    Iterations =      1
Diffusion Segment (  1)     Segment Iterations=     0  Left Enclosure =     1  Right Segment =     2
   Segment Temperatures (K)
          4.9328E+02 4.9328E+02 4 9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02
          4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02
          4.9328E+02 4.9328E+02
     Specie= d
        Mobile Concentration (number/m**3)
          9.0226E+25 1.3384E+26 3.9748E+26 6.6097E+26 9.2337E+26 1.1843E+27 1.4437E+27 1.7018E+27 1.9592E+27
          2.2172E+27 2.4779E+27 2.7453E+27 3.0253E+27 3.3284E+27 3.6727E+27 4.0994E+27 4.6976E+27 5.7890E+27
          1.1546E+28 1.1282E+31
        Mobile inventory =  5.2005E+19 number/m**2
        Flux: Left = -3.9568E+09 number/(s-m**2)         Right = -1.0226E+15 number/(s-m**2)
Diffusion Segment (  2)     Segment Iterations=     0  Left Segment =     1  Right Enclosure =     0
   Segment Temperatures (K)
          4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02
          4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02 4.9328E+02
     Specie= d
        Mobile Concentration (number/m**3)
```

```
          5.4141E+19 5.4473E+19 5.8124E+19 9.4630E+19 4.5969E+20 4.1101E+21 1.3656E+22 2.5832E+22 3.7434E+22
          4.8207E+22 5.7915E+22 6.6347E+22 7.3320E+22 7.8685E+22 8.2327E+22 8.4167E+22 8.4167E+22
        Mobile inventory =  1.0763E+19 number/m**2
        Flux: Left = -1.0226E+15 number/(s-m**2)        Right =  2.7381E+00 number/(s-m**2)
 Enclosure (  1)
   Specie= d2              Concentration (number/m**3)=  9.3744E+13         Pressure (Pa)=  1.0000E-06
     Integrated net flows          Convective (number)=  0.0000E+00     Diffusive (number)= -7.0100E+16
1
 >>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
 R. G. Macaulay-Newcombe's thermal charging problem for gas absorption into
      a wafer of polished beryllium with a thin oxide film.


   Problem Time=  6.0000E+02 seconds     Time step=  6.0000E+01 seconds   Iterations =      1
 Diffusion Segment (  1)    Segment Iterations=    0  Left Enclosure =    1  Right Segment =    2
   Segment Temperatures (K)
          3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02
          3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02
          3.3000E+02 3.3000E+02
     Specie= d
        Mobile Concentration (number/m**3)
          3.4708E+31 1.3384E+26 3.9748E+26 6.6097E+26 9.2337E+26 1.1843E+27 1.4437E+27 1.7018E+27 1.9592E+27
          2.2172E+27 2.4779E+27 2.7453E+27 3.0253E+27 3.3284E+27 3.6737E+27 4.0994E+27 4.6976E+27 5.7890E+27
          1.1546E+28 8.8835E+41
        Mobile inventory =  5.2005E+19 number/m**2
        Flux: Left =  1.4236E+01 number/(s-m**2)        Right = -3.6436E+11 number/(s-m**2)
 Diffusion Segment (  2)    Segment Iterations=    0  Left Segment =    1  Right Enclosure =    0
   Segment Temperatures (K)
          3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02
          3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02 3.3000E+02
     Specie= d
        Mobile Concentration (number/m**3)
          3.0819E+21 3.0819E+21 3.0820E+21 3.0830E+21 3.1035E+21 4.3412E+21 1.3652E+22 2.5819E+22 3.7415E+22
          4.8182E+22 5.7885E+22 6.6313E+22 7.3283E+22 7.8646E+22 8.2285E+22 8.4125E+22 8.4125E+22
        Mobile inventory =  1.0763E+19 number/m**2
        Flux: Left = -3.6436E+11 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)
 Enclosure (  1)
   Specie= d2              Concentration (number/m**3)=  9.3744E+16         Pressure (Pa)=  1.0000E-03
     Integrated net flows          Convective (number)=  0.0000E+00     Diffusive (number)= -7.0100E+16

                          *   *   *

 >>> TMAP MOD1/CY04 -- LATEST REVISION: 1/16/92  <<<
 R. G. Macaulay-Newcombe's thermal charging problem for gas absorption into
      a wafer of polished beryllium with a thin oxide film.


   Problem Time=  1.5000E+04 seconds     Time step=  6.0000E+01 seconds   Iterations =      1
 Diffusion Segment (  1)    Segment Iterations=    0  Left Enclosure =    1  Right Segment =    2
   Segment Temperatures (K)
          1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03
          1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03
          1.0500E+03 1.0500E+03
     Specie= d
        Mobile Concentration (number/m**3)
          1.1960E+23 1.1928E+23 1.1865E+23 1.1802E+23 1.1739E+23 1.1676E+23 1.1613E+23 1.1550E+23 1.1487E+23
          1.1424E+23 1.1361E+23 1.1298E+23 1.1235E+23 1.1171E+23 1.1108E+23 1.1045E+23 1.0982E+23 1.0919E+23
          1.0856E+23 1.0824E+23
        Mobile inventory =  2.0506E+15 number/m**2
        Flux: Left =  2.9973E+14 number/(s-m**2)        Right =  3.0064E+14 number/(s-m**2)
 Diffusion Segment (  2)    Segment Iterations=    0  Left Segment =    1  Right Enclosure =    0
   Segment Temperatures (K)
          1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03
          1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03 1.0500E+03
     Specie= d
        Mobile Concentration (number/m**3)
     3.2430E+21 3.2430E+21 3.2430E+21 3.2429E+21 3.2417E+21 3.2303E+21 3.2019E+21 3.1687E+21 3.1394E+21
     3.1138E+21 3.0920E+21 3.0740E+21 3.0595E+21 3.0487E+21 3.0416E+21 3.0380E+21 3.0380E+21
   Mobile inventory =  6.2076E+17 number/m**2
   Flux: Left =  3.0064E+14 number/(s-m**2)          Right =  0.0000E+00 number/(s-m**2)
Enclosure (  1)
  Specie= d2            Concentration (number/m**3)=  9.3744E+16          Pressure (Pa)=  1.0000E-03
    Integrated net flows         Convective (number)=  0.0000E+00     Diffusive (number)= -6.6263E+16
     ****** NORMAL PROBLEM TERMINATION REACHED ******
```

From the extracted diffusion species data on surface flux from the left side of *thermseg/diffseg* 1, the solid curve in Figure B-1 is constructed where it is compared with the experimental data.



Sample Problem No. 2

Thermal Desorption Test of Beryllium

Figure B-1. Comparison of calculated results (line) with experimental data (+++) for thermal desorption test of beryllium after charging for 50 hours.

APPENDIX C

SAMPLE PROBLEM No. 3

This is a probl?m that involves multiple enclosures and chemical reactions. It is an experiment that was conducted at the Tritium Systems Test Assembly (TSTA) at Los Alamos National Laboratory and documented in D. F. Holland and R. A. Jalbert, "A Model for Tritium Concentration Following Tritium Release into a Test Cell and Subsequent Operation of an Atmospheric Cleanup System," Proceedings, Eleventh Symposium on Fusion Engineering, November 18-22, 1985, Austin, TX, Vol I, pp. 638-643, IEEE Cat. No. CH2251-7. The main part of the experiment is an exposure chamber with a nominal volume of 1 m^3 which is lined with epoxy paint that is 0.16 mm thick. Tritium is admitted to the chamber as T_2 at the commencement of the experiment. Normally moist (20% R.H.) air is admitted to the chamber at the rate of 0.54 m^3/hr constantly throughout the test. Samples of glycol taken from a bubbler just downstream from the exposure chamber are taken at intervals and scintillation counted to determine the time averaged HTO concentration in the chamber as a histogram in time. Tritium and water are absorbed into the paint during the initial part of the test and re-emitted later. Chemical reactions described by the formulae

$$T_2 + H_2O \text{ --> } HTO + HT$$

$$HT + H_2O \text{ --> } HTO + H_2$$

take place within the exposure chamber, mainly as a consequence of the radioactivity of the tritium itself. Results from the measurement of the resulting HTO concentration in the exposure chamber following a 10 Ci initial injection (effectively instantaneously) while purging with room air is shown in Figure 3 of the referenced article.

Modelling consisted of three enclosures (1) the room from which air was drawn, (2) the exposure chamber, and (3) the tritium waste treatment system (TWT) to which the exhaust gases were directed. Only enclosure (2) was treated as "functional" or chemically active. The paint on the inside of the exposure chamber was treated as a diffusive segment with Henry's law solubility governing the concentration at its interface with enclosure (2) and non-flow conditions at the interface of the paint with the underlying aluminum foil. Experiments had previously demonstrated that there is virtually no transport of tritium into the aluminum foil. The technique for determining the constants and other information required to generate a model that gives reasonable results are given in the paper and are not duplicated here.

SAMPLE PROBLEM No. 3 INPUT FILE

```
title input
  SAMPLE PROBLEM 3 - HTO history in an exposure chamber at TSTA
end of title input
$
main input
  dspcnme=t2d,htd,htod,h2od,end
  espcnme=t2,ht,hto,h2o,end
  segnds=12,end
  nbrencl=3,end
end of main input
$
enclosure input
start func,2,end              $ Tritium exposure chamber
  etemp=303.,end
  esppres=t2,0.434,ht,0.,hto,0.,h2o,714.,end
  outflow=nbrflwp,1,qflow,const,1.5e-4,rencl,3,end
  reaction=nequ,2,ratequ,1
           nreact,2,t2,1.,h2o,1.,nprod,2,hto,1.,ht,1.
                  ratequ,2
           nreact,2,ht,1.,h2o,1.,nprod,1,hto,1.,end
  evol=0.96,end
start bdry,1,end              $ Source air from room
  etemp=303.,end
  esppres=t2,0.,ht,0.,hto,0.,h2o,714.,end
  outflow=nbrflwp,1,qflow,const,1.5e-4,rencl,2,end
start bdry,3,end              $ Sink, TWT system
  etemp=303.,end
  esppres=t2,0.,ht,0.,hto.0.,h2o,714.,end
end of enclosure input
$
```

```
thermal input
start thermseg                $ Paint inside the exposure chamber
  delx=0.,10*1.6e-5,0.,end
  dtemp=12*303.,end
end of thermal input
$
diffusion input
start diffseg
  dconc=t2d,12*0.,htd,12*0.,htod,12*0.,h2od,12*0.,end
  dcoef=t2d,const,4.e-12,htd,const,4.e-12
        htod,const,1.e-14,h2od,const,1.e-14,end
  qstrdr=t2d,const,0.,htd,const,0.,htod,const,0.,h2od,const,0.,end
  spcsrc=t2d,const,0.,srcpf,12*0.,htd,const,0.,srcpf,12*0.
         htod,const,0.,srcpf,12*0.,h2od,const,0.,srcpf,12*0.,end
  difbcl=lawdep,encl,2,t2d,t2,pexp,1.,solcon,const,4.e19
                       htd,ht,pexp,1.,solcon,const,4.e19
                     htod,hto,pexp,1.,solcon,const,6.e24
                     h2od,h2o,pexp,1.,solcon,const,6.e24,end
  difbcr=nonflow,end
  surfa=5.6,end
end of diffusion input
$
equation input
$ Reaction rate equations
$   Index for conc array is relative enclosure specie number
$     (i.e., t2=1, ht=2, hto=3, h2o=4)
  y = 2.0e-29*conc(1)*(2.*conc(1)+conc(2)+conc(3)),end  $ Eq.1
  y = 1.0e-29*conc(2)*(2.*conc(1)+conc(2)+conc(3)),end  $ Eq.2
end of equation input
$
table input
end of table input
$
control input
  time=0.,end
  tstep=60.,end
  timend=180000.,end
  nprint=60,end
  itermx=90,end
  delcmx=1.e-5,end
end of control input
$
plot input
  nplot=5,end
  plotseg=1,end
  plotencl=2,3,end
  dname=t2d,htd,htod,htod,end
  ename=t2,ht,hto,end
  dplot=moblinv,sflux,end
  eplot=conc,end
end of plot input
end of data
```

PARTIAL OUTPUT FILE LISTING

```
>>> TMAP MOD1/CY04 -- LATEST REVISION: 10/07/91 <<<
  SAMPLE PROBLEM 3 - HTO history in an exposure chamber at TSTA


  Problem Time=  0.0000E+00 seconds     Time step=  6.0000E+01 seconds   Iterations =      0


Diffusion segment (  1)    Segment Iterations=    0  Left Enclosure =    2  Right Enclosure =    0
   Segment Temperatures (K)
          3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02 3.0300E+02
          3.0300E+02 3.0300E+02 3.0300E+02


     Specie= t2d
        Mobile Concentration (number/m**3)
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00
        Mobile inventory =  0.0000E+00 number/m**2
        Flux: Left =  0.0000E+00 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= htd
        Mobile Concentration (number/m**3)
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00
        Mobile inventory =  0.0000E+00 number/m**2
        Flux: Left =  0.0000E+00 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= htod
        Mobile Concentration (number/m**3)
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00
        Mobile inventory =  0.0000E+00 number/m**2
        Flux: Left =  0.0000E+00 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= h2od
        Mobile Concentration (number/m**3)
          0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
          0.0000E+00 0.0000E+00 0.0000E+00
        Mobile inventory =  0.0000E+00 number/m**2
        Flux: Left =  0.0000E+00 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


Enclosure (  2)
   Specie= t2           Concentration (number/m**3)=  1.0379E+20         Pressure (Pa)=  4.3400E-01
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
   Specie= ht           Concentration (number/m**3)=  0.0000E+00         Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
   Specie= hto          Concentration (number/m**3)=  0.0000E+00         Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
   Specie= h2o          Concentration (number/m**3)=  1.7076E+23         Pressure (Pa)=  7.1400E+02
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00


Enclosure (  1)
   Specie= t2           Concentration (number/m**3)=  0.0000E+00         Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
   Specie= ht           Concentration (number/m**3)=  0.0000E+00         Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00    Diffusive (number)=  0.0000E+00
   Specie= hto          Concentration (number/m**3)=  0.0000E+00         Pressure (Pa)=  0.0000E+00
```

```
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00
   Specie= h2o             Concentration (number/m**3)=  1.7076E+23          Pressure (Pa)=  7.1400E+02
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00


Enclosure (  3)
   Specie= t2              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00
   Specie= ht              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00
   Specie= hto             Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00
   Specie= h2o             Concentration (number/m**3)=  1.7076E+23          Pressure (Pa)=  7.1400E+02
     Integrated net flows          Convective (number)=  0.0000E+00      Diffusive (number)=  0.0000E+00
1
>>> TMAP MOD1/CY04 -- LATEST REVISION: 10/07/91 <<<
  SAMPLE PROBLEM 3 - HTO history in an exposure chamber at TSTA


  Problem Time=  3.6000E+03 seconds     Time step=  6.0000E+01 seconds   Iterations =     3


Diffusion segment (  1)    Segment Iterations=    0  Left Enclosure =    2  Right Enclosure =    0


     Specie= t2d
        Mobile Concentration (number/m**3)
          9.9163E+18 9.9470E+18 9.9197E+18 9.8255E+18 9.6861E+18 9.5222E+18 9.3531E+18 9.1959E+18 9.0647E+18
          8.9709E+18 8.9221E+18 8.9221E+18
        Mobile inventory =  1.5105E+15 number/m**2
        Flux: Left = -1.5319E+10 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= htd
        Mobile Concentration (number/m**3)
          1.1190E+14 1.0673E+14 9.6902E+13 8.7758E+13 7.9461E+13 7.2141E+13 6.5900E+13 6.0821E+13 5.6962E+13
          5.4367E+13 5.3063E+13 5.3063E+13
        Mobile inventory =  1.1746E+10 number/m**2
        Flux: Left =  2.5825E+06 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= htod
        Mobile Concentration (number/m**3)
          1.2375E+19 2.2361E+18 1.2535E+17 5.0283E+15 1.5911E+14 4.1907E+12 9.5078E+10 1.9030E+09 3.4197E+07
          5.5926E+05 8.5306E+03 8.5306E+03
        Mobile inventory =  3.7865E+13 number/m**2
        Flux: Left =  1.2673E+10 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


     Specie= h2od
        Mobile Concentration (number/m**3)
          2.8594E+27 7.6557E+26 5.6839E+25 2.8077E+24 1.0481E+23 3.1650E+21 8.0658E+19 1.7856E+18 3.5068E+16
          6.2071E+14 1.0176E+13 1.0176E+13
        Mobile inventory =  1.3205E+22 number/m**2
        Flux: Left =  2.6173E+18 number/(s-m**2)        Right =  0.0000E+00 number/(s-m**2)


Enclosure (  2)
   Specie= t2              Concentration (number/m**3)=  5.9288E+19          Pressure (Pa)=  2.4791E-01
     Integrated net flows          Convective (number)= -4.2715E+19      Diffusive (number)= -8.4589E+15
   Specie= ht              Concentration (number/m**3)=  6.6903E+14          Pressure (Pa)=  2.7975E-06
     Integrated net flows          Convective (number)= -2.4513E+14      Diffusive (number)= -6.5776E+10
   Specie= hto             Concentration (number/m**3)=  4.9324E+14          Pressure (Pa)=  2.0624E-06
     Integrated net flows          Convective (number)= -2.0191E+14      Diffusive (number)= -2.1205E+14
   Specie= h2o             Concentration (number/m**3)=  1.1397E+23          Pressure (Pa)=  4.7657E+02
     Integrated net flows          Convective (number)=  1.9438E+22      Diffusive (number)= -7.3949E+22
```

```
Enclosure (  1)
   Specie= t2              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00     Diffusive (number)=  0.0000E+00
   Specie= ht              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00     Diffusive (number)=  0.0000E+00
   Specie= hto             Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  0.0000E+00     Diffusive (number)=  0.0000E+00
   Specie= h2o             Concentration (number/m**3)=  1.7076E+23          Pressure (Pa)=  7.1400E+02
     Integrated net flows          Convective (number)= -9.2208E+22     Diffusive (number)=  0.0000E+00

Enclosure (  3)
   Specie= t2              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  4.2715E+19     Diffusive (number)=  0.0000E+00
   Specie= ht              Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  2.4513E+14     Diffusive (number)=  0.0000E+00
   Specie= hto             Concentration (number/m**3)=  0.0000E+00          Pressure (Pa)=  0.0000E+00
     Integrated net flows          Convective (number)=  2.0191E+14     Diffusive (number)=  0.0000E+00
   Specie= h2o             Concentration (number/m**3)=  1.7076E+23          Pressure (Pa)=  7.1400E+02
     Integrated net flows          Convective (number)=  7.2771E+22     Diffusive (number)=  0.0000E+00
```

Data were extracted for the HTO concentration in the exposure chamber, *enclosure* 2. A solid curve representing these data is compared in Figure C-1 with measurements made in bubblers inline with the exposure chamber exhaust. The period over which the bubblers were active in collecting HTO from the exposure chamber is shown on the time scale. They were integrated measurements over the intervals shown.

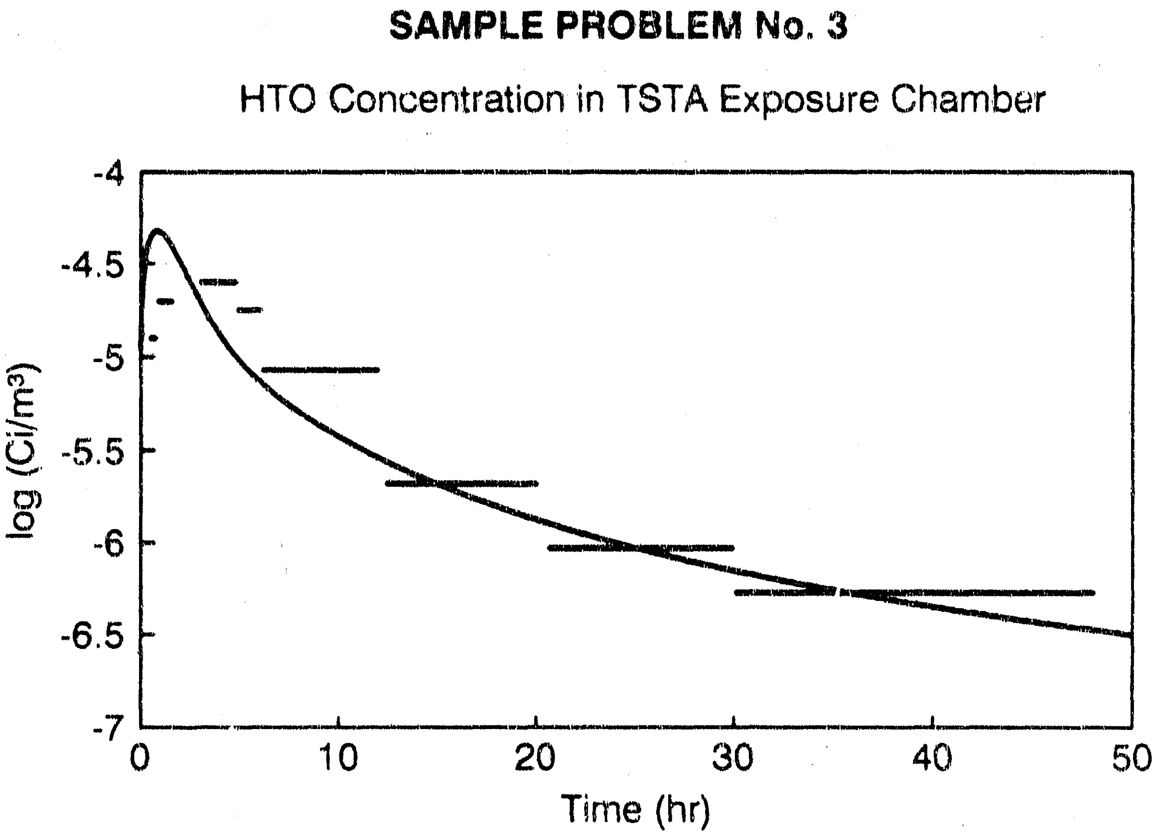


Figure C-1 Comparison of calculational result with actual experimental data.

APPENDIX D

TMAP4.BAT SCRIPT FILE LISTING

```
echo off
rem ************************************************
rem ****** PROCEDURE TO RUN THE TMAP CODE  **********
rem ******   ON A PC OPERATING UNDER DOS   **********
rem ******  USING THE LEHEY F77L COMPILER  **********
rem ************************************************
rem
rem --- assure any existing scratch files are deleted
if exist tape7.for  del tape7.for
if exist tape1      del tape1
rem --- check for possible output or pltdata file overwrite
if not exist %1.out goto x0
echo YOU WILL OVERWRITE %1.out...PRESS [Ctrl-C] TO STOP
pause
del %1.out
:x0
if not exist %1.plt goto x1
echo YOU WILL OVERWRITE %1.plt...PRESS [Ctrl-C] TO STOP
pause
del %1.plt
:x1
rem --- check if input filename supplied exists
if not exist %1.inp goto quit
rem --- start preprocessor execution
echo COPYING %1.inp INTO tmapinp
copy %1.inp tmapinp
rem
echo PREPARING tmapinp FOR COMPILATION
tmapp4
rem
rem --- perform check to see if FORTRAN compilation is required
rem     'tape7.for' is file name containing all user defined equations
if not exist tape7.for goto nocomp
rem
F77L tape7
rem --- check if compilation error detected
if errorlevel 1 goto quit
rem
link tape7+tmapc4,tmape,NUL,F77L;
del tape7.*
goto exec
:nocomp
rem
rem *********** link to execute without equations in input file *************
rem  ---- equ is just a dummy obj. file
```

```
link equ+tmapc4,tmape,NUL,f77l;
:exec
rem
rem ******* execute program ***********
tmape
rem
rem --- clearing out remaining files and form output file
del tapel
del tmape.exe
copy prepout+codeout %1.out
del prepout
del codeout
rem --- rename pltdata file if one was created
if not exist pltdata goto noplt
rename pltdata %1.plt
:noplt
echo *****************PROBLEM EXECUTION COMPLETE*****************
goto end
:quit
echo INPUT FILE NOT FOUND...EXECUTION ABORTED!
:end
```

END

DATE
FILMED
8 / 26 / 92