Idaho National Engineering and Environmental Laboratory

INEEL/EXT-2000-01337
October 2000

TMAP2000 Use

G. R. Longhurst
B. J. Merrill
J. L. Jones

BECHTEL BWXT IDAHO, LLC

INEEL/EXT-2000-01337

TMAP2000 Use

Glen R. Longhurst
Brad J. Merrill
James L. Jones

Published October 2000

Idaho National Engineering and Environmental Laboratory
Idaho Falls, Idaho 83415-3860

Prepared for the
U.S. Department of Energy
Office of Fusion Energy Sciences
Under DOE Idaho Operations Office
Contract DE-AC07-99ID13727

ABSTRACT

The TMAP Code was written in the late 1980s as a tool for safety analysis of systems involving tritium. Since then it was upgraded to TMAP4 and used in numerous applications including experiments supporting fusion safety, predictions for advanced systems such as the International Thermonuclear Experimental Reactor (ITER), and estimates involving tritium production technologies. Its further upgrade to TMAP2000 was accomplished in response to several needs. TMAP and TMAP4 had the capacity to deal with only a single trap for diffusing gaseous species in solid structures. TMAP2000 has been revised to include up to three separate traps and to keep track separately of each of up to 10 diffusing species in each of the traps. A difficulty in the original code dealing with heteronuclear molecule formation such as HD and DT has been removed. Under equilibrium boundary conditions such as Sieverts' law, TMAP2000 generates heteronuclear molecular partial pressures when solubilities and partial pressures of the homonuclear molecular species and the equilibrium stoichiometry are provided. A further sophistication is the addition of non-diffusing surface species and surface binding energy dynamics options. Atoms such as oxygen or nitrogen on metal surfaces are sometimes important in molecule formation with diffusing hydrogen isotopes but do not themselves diffuse appreciably in the material. TMAP2000 will accommodate up to 30 such surface species, allowing the user to specify relationships between those surface concentrations and populations of gaseous species above the surfaces. Additionally, TMAP2000 allows the user to include a surface binding energy and an adsorption barrier energy and includes asymmetrical diffusion between the surface sites and regular diffusion sites in the bulk. All of the previously existing features for heat transfer, flows between enclosures, and chemical reactions within the enclosures have been retained, but the allowed problem size and complexity have been significantly increased to take advantage of the greater memory and speed available on modern computers.

This report provides users of TMAP2000 with the specialized information they will need to properly construct the input files used with the code. It assumes the user has and is familiar with the TMAP4 Users Manual, and it focuses on changes from TMAP4 input file requirements.

FOREWORD

This report is intended to be used with the existing *TMAP4 User's Manual* (EGG-FSP-10315, June 12, 1992) and the *TMAP2000 Development* report (INEEL/EXT-2000-678, May 2000). It will allow users to implement the revised version of TMAP, TMAP2000, in analyzing transport of dissolved gases through structures and within enclosures surrounding those structures. The previous versions of the code were written for use within the limited capabilities of second generation desk-top computers such as the X286 class machines. These had typically only 640 kB of base memory available and operated with computational rates that were far below what is available now. The present code requires approximately 4 MB of available memory to run in and occupies nominally 16 MB of disk space, including that required to house the G77 Fortan system upon which the present upgrade is based. G77 Fortran was selected for several reasons. One is that it is freeware, available at no cost to the public (though donations to developers are solicited), and it can be distributed with the code. Another is that while not fully up-to-date, it is fully compliant with the Fortran 77 standard and supports many features of Fortran 90. All of the coding in TMAP2000 and its predecessors is sufficiently archaic in style that the Fortran 77 features are fully adequate. Finally, G77 works on UNIX platforms and in the Windows 95, 98, and NT environments. Therefore, there should be good portability for a variety of users. As of this writing, no effort has been made to adapt it to the Macintosh platforms, but that may be done in the future. For expediency in getting the updated code into the hands of users, a number of graphical interface features that could be adapted for simplifying examination of output have been deferred. Hence, for now, the plot data pairs continue to be written to the PLTDATA file, and the regular output data to the PREPOUT and CODEOUT files for subsequent evaluation. The EXTRACT utility has also been updated to allow retrieval of the expanded data available. The program continues to operate in two steps with a preprocessor and a computational module. The preprocessor stores the data from the TMAPINP file into the appropriate arrays stored in the unformatted TAPE1 file and generates the TAPE7.FOR file that is compiled into a callable subroutine for evaluation of equation type input. If no equations are used, a dummy file is substituted. The computational module then reads the data in TAPE1, performs the computations and generates the output files. Further refinements and enhancements are hoped for in the future.

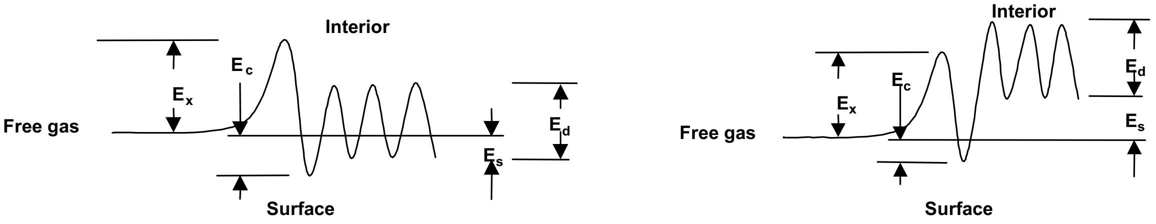
CONTENTS

TMAP2000 USE

INTRODUCTION

The theoretical and mathematical developments relevant to the TMAP2000 code were presented in a previous report (INEEL/EXT-2000-678, May 2000). The intent here is to provide detailed instructions on preparation of the input file, TMAPINP, upon which the preprocessor operates. It is assumed that the user has access to the TMAP4 User's Manual (EGG-FSP-10315, June 12, 1992) and is familiar with the operation of TMAP4. Here, selected items in the input file for TMAP2000, especially those that are different from those for TMAP4, are described and limitations on their use are set forth.

The major new aspect of TMAP2000, beyond the addition of multiple traps, deals with molecular formation, dissociation, capture, and release processes at the surface. As an aid to following the discussion of those processes later in the text, Figure 1 is provided that shows the relationship among the various energies involved. Figure 1(a) is for an exothermic absorber, one where the energy of the solute gas species in solid solution is lower than its free molecular gas energy. Figure 1(b) is an exothermic absorber where the reverse is true.



(a) **(b)**

Figure 1. Comparison of (a) exothermic, and (b) endothermic solubility potentials, each with a binding energy, E_c, and a capture threshold energy, E_x.

In the material that follows, instructions are provided for the various entries in the TMAPINP file used with TMAP2000. Words in **bold Arial** typeface are keywords that need to be entered in lower case as shown. Generally, spaces are optional. Items shown in angled brackets (**<>**) are input data. Those shown with ***(α)*** are alphanumeric constants, those with ***(i)*** are of type integer while those with ***(r)*** are real or floating-point numbers. Even though the code operates fully in double precision arithmetic for real numbers, the input preprocessor does not recognize "d" notation. Exponents in scientific notation should be entered with a lower-case "e".

Most statements are required unless otherwise indicated. Those that are optional are included in brackets (**[ ]**). Note that nearly all statements are terminated with "**,end**". This is a key to the preprocessor that the statement is concluded. Multiple line entries are thus permitted without any continuation indicator. Where multiple parameters in a statement are separated by commas in a statement requiring more than one line, no comma should be placed at the end of a continued line nor at the beginning of the next one. That function is provided by the carriage return marking the end of the line. Input file statements may fill the first 80 columns of any line. Comment statements must begin with the dollar sign ($). These may not appear in the middle of a multiple-line statement, though they may appear on the same line following the "**,end**" of a statement.

We now proceed to give the statement specifications, generally in the order they would be entered in the TMAPINP file.

Main Input

dspcnme = <$dspc(\alpha)_1$>,<$dspc(\alpha)_2$>,…,<$dspc(\alpha)_n$>,end

No change from TMAP4. This is where names are given to each of the diffusing species. This list does not include those species to be separately identified as surface-only species. The maximum number of diffusing species is still 10. Diffusion species names are limited to 8 characters.

espcnme = <$espc(\alpha)_1$>,<$espc(\alpha)_2$>,…,<$espc(\alpha)_n$>,end

No change in this statement except that the number of enclosure species allowed has been increased to 30. Enclosure species names are also limited to 8 characters. They may replicate diffusion species names, such as when Henry's law applies, but they are considered as separate species within the code, and repeating the names can lead to confusion.

[sspcnme = <$sspc(\alpha)_1$>,<$sspc(\alpha)_2$>,…,<$sspc(\alpha)_n$>,end]

This is a new optional statement that identifies surface-only species. These species were provided to allow expansion of calculational possibilities that may include such things as the formation of molecules that remain attached to the surface and interact with diffusing species at the surface without actually diffusing themselves. Examples may be water or methane and the complexes involved in their formation such as hydroxyl radicals. Rules for surface species interaction with enclosure species depend on the enclosure type and the diffusion boundary conditions at the surface. Surface-only species are only meaningful under *surfdep,* and *ratedep* diffusion boundary conditions. For *lawdep, sconc* and *nonflow* boundary conditions, surface species are meaningless since diffusion species concentrations do not depend on them. Further, for *lawdep* boundary conditions interacting with functional enclosures, concentrations of heteronuclear species in the enclosures must be computed using an equilibrium law. These various boundary condition types are summarized in Table 1.

Table 1. Summary of applicability and treatment of surface species for various combinations of enclosure type and diffusion boundary conditions.

Diffusion Boundary Condition	**Functional Enclosure**	**Boundary Enclosure**
Surfdep	Captured by species arrival from gas; returned to gas by escape from the surface; no species changes enroute, means there must be an enclosure species corresponding to each surface species that can escape or be captured; rates dynamically computed.	Captured by species arrival from gas; returned to gas by escape from the surface; no species changes enroute, means there must be an enclosure species corresponding to each surface species that can escape or be captured; rates of exchange are irrelevant because enclosure species concentrations are pre-specified..
Ratedep	Undergo dissociation and recombination reactions with automatic release to enclosure species (similar to diffusion species), but also surface formation, dissociation[a], and combination reactions; rates to enclosure are dynamically computed	Undergo dissociation and recombination reactions with automatic release to enclosure species (similar to diffusion species), but also surface formation, dissociation[a], and combination reactions; rates irrelevant because enclosure species concentrations are pre-specified.
Lawdep	Meaningless because surface concentrations of diffusion species are set by assigned solubility relations.	Meaningless because surface concentrations of diffusion species are set by assigned solubility relations.
Sconc	Meaningless because diffusion species concentrations are fixed, not subject to adjustment by surface interactions.	Meaningless because diffusion species concentrations are fixed, not subject to adjustment by surface interactions.
Nonflow	Meaningless because surface is not available for reactions.	Meaningless because surface is not available for reactions.

[a] Conventional dissociation/recombination reactions associated with the *ratedep* boundary condition are those relating surface concentrations of the species in question to the gas molecules above the surface. Surface formation and dissociation reactions addressed here involve other species at the surface. Formation means forming the surface species from surface constituents while dissociation is the reverse. Combination is the incorporation of the subject surface species with another one into a third.

For functional enclosures and *surfdep* diffusion boundary conditions, both surface-only and diffusion species may desorb if they can overcome the surface binding potential, (E_x - E_c in Figure 1), but for the most part they come off only by formation of a larger molecule having a lower surface binding energy. Surface-only species must have names different from those in the diffusion species name list. A maximum of 30 surface species names may be defined.

To illustrate the need for so many, if all three hydrogen isotopes were present as diffusion species, and we considered only oxygen additionally, we would need 3 diffusion species (H, D, T), 3 surface species (OH, OD, OT), and 13 enclosure species (6 forms of hydrogen gas, 6 forms of water and O_2). If ammonia formation were added, we would need 12 additional enclosure species and 10 additional surface species.

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

No change in the form here. However, the maximum number of *thermsegs/diffsegs* is increased from 20 to 50 with a maximum number of nodes, including surface nodes, of 1,000. Note that the product of the number of diffusing species and the number of nodes in any contiguously linked structure, which may include multiple diffusion segments, must be less than 400. The input processor provides warning messages if these limits are violated.

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

No change. The maximum number of enclosures is still 40.

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

No change. There is no arbitrary limit to the number of segments that can be linked together along any particular path, so long as the number of total segments does not exceed 50 and the number of nodes in a linked structure multiplied by the number of diffusing species does not exceed 400. Note that in TMAP2000 the segment linking need not be in numerical sequence.

Enclosure Input

In this section the user provides TMAP2000 specific data required to define the *enclosure* control 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 be diffusive flows of enclosure species to and from the surfaces. 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. Neither convective fluid flows nor diffusive enclosure species flows to and from boundary type enclosures are computed because properties in boundary enclosures are pre-specified. They may be used as sources and sinks for functional enclosures, however, where those flows are computed.

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, any enclosure reference number may be defined by the user provided that each is unique and that all the numbers between 1 and the number of enclosures defined are used. The sequence of entering enclosure types and the numbers assigned to the enclosures are otherwise arbitrary. This is a change from previous code versions.

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.

Functional Enclosures

There are five parameter statements following the "start" declaration in specifying *functional* enclosures. Of these, two are optional: the specification of fluid flows between enclosures and that for chemical reactions.

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

No change in this statement.

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

This statement identifies the temperature of the enclosure. The value of that temperature may be a constant, indicated by entering just the number or the keyword *const* and the number, or it may be defined in terms of an equation or a lookup table as a function of time. This temperature is used with the enclosure species concentrations to calculate partial pressures of the various enclosure species. In previous code versions, enclosure temperature was fixed. Temperatures are entered in kelvins.

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

This statement enters partial pressures for the various enclosure species. For functional enclosures, these are entries for starting pressure and there must be an entry for each of the enclosure species identified in the *main input* block. These are then changed by diffusive and convective fluid flows during the course of the problem. Pressures are entered in pascals.

When a functional enclosure faces a diffusion segment under *lawdep* boundary conditions, only one enclosure species can be linked to the concentration of any diffusion species. Normally, a hydrogenic diffusion species would be related to the partial pressure of its diatomic homonuclear molecule. The *lawdep* boundary condition implies chemical equilibrium at the surface, so if there is more than one diffusion species, the specified enclosure species may dissociate to atoms that combine to form heteronuclear molecules (e.g., HD). In these instances, heternouclear molecular pressures will need to be calculated automatically with the following statement.

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

<espc(α)$_2$>,<stoich(r)$_2$><espc(α)$_3$>,<stoich(r)$_3$>

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

<espc(α)$_2$>,<stoich(r)$_2$><espc(α)$_3$>,<stoich(r)$_3$>,…**],end]**

This new optional statement allows the user to specify which of the enclosure species combine together to make a third when at least one diffusion segment facing this *boundary* encolsure has the *lawdep* diffusion boundary condition. Examples are H_2 and D_2, which combine at the surface to make HD, or H_2 and O_2, which combine to form H_2O. Under these conditions, where homonuclear molecular gas pressures are specified, the code will automatically calculate corresponding heteronuclear molecular pressures according to

$$P_C = \eta P_A^{\nu_A} P_B^{\nu_B}$$

In this statement, *<espc(α)$_1$>* is the name of the sought-for (heteronuclear) enclosure species associated with P_C and similarly *<espc(α)$_2$>,<espc(α)$_3$>* are names of the specified (homonuclear) species P_A and P_B. The value following *<espc(α)$_1$>* is the equilibrium constant, η, which may be assigned values as a constant, or as equation or table lookup. The table is assumed to have temperature as the independent variable, while the equation form can specify either time or temperature or both as independent variables. The *<stoich(r)$_i$>* are the stoichiometric coefficients ν_i. Only those species identified in this statement will be automatically computed. Note that only one specification for each sought-for molecule so formed should be given, and only binary reactions are facilitated in the code.

This specification is for the current functional enclosure only. One must be provided for other enclosures where needed. If, in addition to at least one diffusion segment having the *lawdep* diffusion boundary condition, there are other diffusion segments facing this enclosure that have other diffusion boundary conditions, the sought-for (heteronuclear) molecular species will still be calculated and used.

[reaction = nequ,*<neq(i)>*

ratequ,*<eqnbr(i)>*

nreact,*<nbrreact(i)>,<espc(α)$_1$>,<stoich(r)$_1$>,…,*

<espc(α)$_n$>,<stoich(r)$_n$>

nprod,*<nbrprod(i)>,<espc(α)$_1$>,<stoich(r)$_1$>*,…,

<espc(α)$_n$>,<stoich(r)$_n$>

⋮

[ratequ,*<eqnbr(i)>*

nreact,*<nbrreact(i)>,<espc(α)$_1$>,<stoich(r)$_1$>*,…,

<espc(α)$_n$>,<stoich(r)$_n$>

nprod,*<nbrprod(i)>,<espc(α)$_1$>,<stoich(r)$_1$>*,…,

***<espc(α)$_n$>,<stoich(r)$_n$>*]**

end]

No change in this statement, which defines chemical reactions taking place in the enclosure volume. These are computed as if the enclosure were a static volume with no flows. This statement would not normally be used with a *lawdep* diffusion boundary condition unless it involves enclosure species that do not interact at the surface of the diffusion segment.

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

No change in this statement, which defines the volume of the functional enclosure. Units are cubic meters.

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

$$\textbf{qflow,}\left\{\begin{matrix} <\textbf{\textit{value(r)}}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{matrix}\right\}\textbf{,rencl,}<\textbf{\textit{valuer(i)}}>$$

⋮

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

No change in this statement. It defines the parameters of the optional flows from the present enclosure to other enclosures. Flow rates are entered in m^3/s.

Boundary Enclosures

Boundary enclosures represent pre-specified environments that are not altered by convective flows or diffusion effects. Two parameter statements are required for *boundary* enclosures following the "start" declaration for the sub-block, and one more is optional.

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

No change in this statement.

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

Temperatures are allowed to change in boundary enclosures as a function of time. Real values input alone or with the keyword *const* are the constant temperature over the duration of the problem. Integers entered after keywords *equ* or *tabl* are equation or table numbers, respectively, for evaluating the enclosure temperature, again, as a function of time. Temperatures must be entered in kelvins.

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

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

This statement provides a specification, entered in pascals, for the enclosure species pressures in *boundary* enclosures. If the diffusion boundary conditions for diffusion segments connected with this enclosure are *surfdep* or *ratedep*, then an entry should be made for each enclosure and surface-only species present. On the other hand, if the diffusion boundary conditions are *lawdep*, the user should only provide entries for one enclosure species per diffusion species, normally the homonuclear molecular species (e.g., H_2, D_2).

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

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

$$\vdots$$

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

No change in this statement.

Thermal Input

One change has been made in this section:

$$\textbf{temp = }\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_1>,\ldots,<\textbf{\textit{value(r)}}_n> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{ramp,}}<\textbf{\textit{value(r)}}_1>\textbf{,}<\textbf{\textit{value(r)}}_2> \\ \textbf{\textit{exp,}}<\textbf{\textit{value(r)}}_1>\textbf{,}<\textbf{\textit{value(r)}}_2>\textbf{,}<\textbf{\textit{value(r)}}_3> \\ \textbf{\textit{norm,}}<\textbf{\textit{value(r)}}_1>\textbf{,}<\textbf{\textit{value(r)}}_2>\textbf{,}<\textbf{\textit{value(r)}}_3>,<\textbf{\textit{value(r)}}_4> \end{array}\right\}\textbf{,end}$$

This new specification for initial temperature distribution in the structure has the added flexibility of applying functional distributions for temperature. A series of simple numerical values may be entered (one for each node or combined using the *<numnds(i)>*<value(r)>* format). Alternatively the following may be entered:

temp = const,*<value(r)*$_1$*>*

This sets the temperature value at every node equal to *<value(r)*$_1$*>.*

temp = ramp,*<value(r)*$_1$*>,<value(r)*$_2$*>*

This form provides the temperature at each node as a linear interpolation between *<value(r)*$_1$*>* at the left face and *<value(r)*$_2$*>* at the right face of the *diffseg.*

temp = exp,*<value(r)*$_1$*>,<value(r)*$_2$*>,<value(r)*$_3$*>*

This form calculates the initial temperature at each node as a decaying exponential beginning at the left face as *<value(r)*$_1$*>* and decaying to ,*<value(r)*$_2$*>* with a 1/e width of *<value(r)*$_3$*>* (m).

If *<value(r)*$_2$*>* is greater than *<value(r)*$_1$*>*, the function is a saturating exponential.

temp = norm,*<value(r)*$_1$*>,<value(r)*$_2$*>,<value(r)*$_3$*>,<value(r)*$_4$*>*

This form calculates the temperature as the sum of a normal distribution with a maximum value of *<value(r)*$_1$*>* centered at a distance x_o of *<value(r)*$_2$*>* (m) from the left face, and having a standard deviation of *<value(r)*$_3$*>* (m); combined with a constant background value of *<value(r)*$_4$*>*.

These functional forms may have greater or less usefulness for temperature, but they are also used for specifying initial concentration distributions, as discussed below.

Diffusion Input

start diffseg [,end]

Several changes have been implemented in this block. One is that the diffusion segment number density is required for all options, so there is an added required statement:

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

Also, initial concentrations may now be given spatial distributions automatically:

concd =

$$\textbf{<dspc(}\boldsymbol{\alpha}\textbf{)}_1\textbf{>,}\left\{\begin{array}{l} <\textbf{value(r)}_1>,\ldots,<\textbf{value(r)}_n> \\ \textbf{const,}<\textbf{value(r)}> \\ \textbf{ramp,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2> \\ \textbf{exp,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2>,<\textbf{value(r)}_3> \\ \textbf{norm,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2>,<\textbf{value(r)}_3>,<\textbf{value(r)}_4> \end{array}\right\}$$

$$\textbf{<dspc(}\boldsymbol{\alpha}\textbf{)}_2\textbf{>,}\left\{\begin{array}{l} <\textbf{value(r)}_1>,\ldots,<\textbf{value(r)}_n> \\ \textbf{const,}<\textbf{value(r)}> \\ \textbf{ramp,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2> \\ \textbf{exp,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2>,<\textbf{value(r)}_3> \\ \textbf{norm,}<\textbf{value(r)}_1>,<\textbf{value(r)}_2>,<\textbf{value(r)}_3>,<\textbf{value(r)}_4> \end{array}\right\}$$

$$\vdots$$

$$\boldsymbol{<dspc(\alpha)_m>,}\left\{\begin{array}{l} <\boldsymbol{value(r)_1}>,\ldots,<\boldsymbol{value(r)_n}> \\ \boldsymbol{const,}<\boldsymbol{value(r)}> \\ \boldsymbol{ramp,}<\boldsymbol{value(r)_1}>\boldsymbol{,}<\boldsymbol{value(r)_2}> \\ \boldsymbol{exp,}<\boldsymbol{value(r)_1}>\boldsymbol{,}<\boldsymbol{value(r)_2}>\boldsymbol{,}<\boldsymbol{value(r)_3}> \\ \boldsymbol{norm,}<\boldsymbol{value(r)_1}>\boldsymbol{,}<\boldsymbol{value(r)_2}>\boldsymbol{,}<\boldsymbol{value(r)_3}>,<\boldsymbol{value(r)_4}> \end{array}\right\}$$

end

This statement establishes the initial concentration values of the mobile diffusion species defined in the *Main Input* block at each of the nodes if those values are other than 0. Values of diffusion species will be stored beginning at the left surface node of each segment and progressing to the right face. Values given in excess of the number needed for the number of nodes specified for the *diffseg* will be ignored. If an insufficient number of values is provided, it will be assumed the remainder are zero. Note that the shorthand entry, *<numnds(i)>*<value(r)>* may also be used to input multiple values of a given number. Functional input forms are implemented the same way as in the *temp* distributions described previously except for the *norm* distribution where the first parameter, *<value(r)$_1$>*, is now the integrated number of atoms per square meter of surface area over the entire normal distribution (not counting the background value) rather than the maximum value. If the distribution function is truncated by the surface, as it frequently will be, *<value(r)$_1$>* is also adjusted so that it represents the number in the truncated distribution.

$$\textbf{[ssconc = <sspc(}\alpha\textbf{)}_1\textbf{>,}\left\{\begin{array}{l} <\boldsymbol{value(r)}> \\ \boldsymbol{link} \end{array}\right\}\textbf{,}\left\{\begin{array}{l} <\boldsymbol{value(r)}> \\ \boldsymbol{link} \end{array}\right\}$$

$$\textbf{[<sspc(}\alpha\textbf{)}_n\textbf{>,}\left\{\begin{array}{l} <\boldsymbol{value(r)}> \\ \boldsymbol{link} \end{array}\right\}\textbf{,}\left\{\begin{array}{l} <\boldsymbol{value(r)}> \\ \boldsymbol{link} \end{array}\right\}\textbf{,...],end]}$$

This new optional statement is for specifying initial concentration values of any surface-only species identified in the *maininp* block. Only those surface species identified here will be given non-zero initial values. The first entry following the surface species name is the initial value for the left face of the *diffseg*, while the second value is that for the right face. An entry of *link* for one face or the other should be entered if the *diffseg* is linked to another on that side. If no surface species are to be identified for this *diffseg* or if both faces are linked, the statement may be omitted.

$$\textbf{[trapping = ttyp,alphr,}\left\{\begin{array}{l} <\boldsymbol{value(r)_1}>,\ldots,<\boldsymbol{value(r)_n}> \\ \boldsymbol{const,}<\boldsymbol{value(r)}> \\ \boldsymbol{equ,}<\boldsymbol{value(i)}> \\ \boldsymbol{tabl,}<\boldsymbol{value(i)}> \end{array}\right\}$$

$$\textbf{tconc,}\left\{\begin{array}{l} <\textit{value(r)}_1>,\ldots,<\textit{value(r)}_n> \\ \textit{const,}<\textit{value(r)}> \\ \textit{ramp,}<\textit{value(r)}_1>,<\textit{value(r)}_2> \\ \textit{exp,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3> \\ \textit{norm,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3>,<\textit{value(r)}_4> \end{array}\right\}$$

$$\textbf{tspc,}<\textit{dspc}(\alpha)_1>\textbf{,alpht,}\left\{\begin{array}{l} <\textit{value(r)}_1>,\ldots,<\textit{value(r)}_n> \\ \textit{const,}<\textit{value(r)}> \\ \textit{equ,}<\textit{value(i)}> \\ \textit{tabl,}<\textit{value(i)}> \end{array}\right\}$$

$$\textbf{ctrap,}\left\{\begin{array}{l} <\textit{value(r)}_1>,\ldots,<\textit{value(r)}_n> \\ \textit{const,}<\textit{value(r)}> \\ \textit{ramp,}<\textit{value(r)}_1>,<\textit{value(r)}_2> \\ \textit{exp,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3> \\ \textit{norm,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3>,<\textit{value(r)}_4> \end{array}\right\}$$

$\vdots$

$$\textbf{[tspc,}<\textit{dspc}(\alpha)_n>\textbf{,alpht,}\left\{\begin{array}{l} <\textit{value(r)}_1>,\ldots,<\textit{value(r)}_n> \\ \textit{const,}<\textit{value(r)}> \\ \textit{equ,}<\textit{value(i)}> \\ \textit{tabl,}<\textit{value(i)}> \end{array}\right\}$$

$$\textbf{ctrap,}\left\{\begin{array}{l} <\textit{value(r)}_1>,\ldots,<\textit{value(r)}_n> \\ \textit{const,}<\textit{value(r)}> \\ \textit{ramp,}<\textit{value(r)}_1>,<\textit{value(r)}_2> \\ \textit{exp,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3> \\ \textit{norm,}<\textit{value(r)}_1>,<\textit{value(r)}_2>,<\textit{value(r)}_3>,<\textit{value(r)}_4> \end{array}\right\}\textbf{]}$$

$\vdots$

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

,end]

This is a new optional statement that replaces all the trapping associated entries in TMAP4. It sets the number of different trap types that will be active in this *diffseg* together with all the other relevant data pertaining to these traps. These data include for each trap type (*ttyp*): the trap release rate (*alphr*) (1/s), which depends on trap type but is assumed to be species independent, the distribution of trap concentration (*tconc*) (atom fraction) within the *diffseg*, the identities of the diffusing species (*tspc*) that will compete for that trap type, and for each trapped species: the trapping rate coefficient (*alpht*) (1/s), which is species dependent, and the concentration (*ctrap*) (fraction filled) of trapped atoms of that species initially in traps of that type. If trapping will be taking place in the problem, this statement must be included, at least one trap type must be specified, and at least one species must be trapped in each specified trap type. Not every *diffseg* need have trapping active, and if it is, only the trap types specified here will be assumed active in that segment. The maximum number of trap types in any *diffseg* is 3, but these need not be the same traps from one segment to the next. Only species identified in the main input block as diffusion species may be input following the *tspc* keyword.

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

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

No substantive change to this statement except that entries are required for each surface-only species as well as for each diffusion species. Both are indicated here by the entry, $<spc(\alpha)_i>$, and the order of entry is immaterial. In the case of surface-only species, the diffusivity is the lateral diffusivity, which can be taken as

$$D_{lat} = \frac{2}{3}\nu_o \lambda^2 \exp\left(-\frac{E_l}{kT}\right)$$

where ν_o is the Debye frequency and E_l is the activation energy for lateral movement. Like the standard diffusivity, this quantity may be entered as a constant, equation, or table.

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

$$\textbf{<}\textbf{\textit{dspc(}}\alpha\textbf{\textit{)}}_{n}\textbf{>,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_{1}>,\ldots,<\textbf{\textit{value(r)}}_{n}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{array}\right\}\textbf{,end}$$

No change to this statement. Note that entries are only needed for diffusion species and not for surface-only species.

$$\textbf{srcsd = <}\textbf{\textit{dspc(}}\alpha\textbf{\textit{)}}_{1}\textbf{>,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_{1}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{array}\right\}$$

$$\textbf{srcpf,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_{1}>,\ldots,<\textbf{\textit{value(r)}}_{n}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{ramp,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}> \\ \textbf{\textit{exp,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}>\textbf{,}<\textbf{\textit{value(r)}}_{3}> \\ \textbf{\textit{norm,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}>\textbf{,}<\textbf{\textit{value(r)}}_{3}<\textbf{\textit{value(r)}}_{4}> \end{array}\right\}$$

$$\vdots$$

$$\textbf{<}\textbf{\textit{dspc(}}\alpha\textbf{\textit{)}}_{n}\textbf{>,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_{1}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{array}\right\}$$

$$\textbf{srcpf,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}_{1}>,\ldots,<\textbf{\textit{value(r)}}_{n}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{ramp,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}> \\ \textbf{\textit{exp,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}>\textbf{,}<\textbf{\textit{value(r)}}_{3}> \\ \textbf{\textit{norm,}}<\textbf{\textit{value(r)}}_{1}>\textbf{,}<\textbf{\textit{value(r)}}_{2}>\textbf{,}<\textbf{\textit{value(r)}}_{3}<\textbf{\textit{value(r)}}_{4}> \end{array}\right\}\textbf{,end}$$

This statement defines the volumetric source rate for the various diffusion species, such as from implantation or neutronic transmutation. Temporal dependence is defined by the expression following the diffusion species name. The statement has been changed to allow functional inputs to the source peaking factor, which provides spatial distributions, and the initial keyword has been changed from *spcsrc* to *srcsd.* Note that because the species source rate is associated with volumetric production of a diffusion species, it must be specified as 0 for the two zero-volume surface nodes. For the functional forms, spatial distribution of diffusion species source rates will be accomplished automatically. Again in the *norm* distribution, the <value(r)$_1$>

parameter refers to the integrated value (atoms/m^{2}s) over the thickness of the segment, not counting the background (<value(r)$_4$>) value. No species declared as surface-only species should be included in this statement.

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

spc,<*spc(*α*)*$_1$>,nu,<*gnuz(r)*>,ec<*value(r)*$_1$>,[es,<*value(r)*$_2$>]

[exch,<*espc(*α*)*$_1$>,amu,<*value(r)*>,ex,<*value(r)*>,stk,<*value(r)*>]

[comb,<*spc(*α*)*>]

[form,<*spc*$_1$*(*α*)*>,<*spc*$_2$*(*α*)*>]

$$\textbf{[diss,}\left\{\begin{matrix}<\textbf{\textit{value(r)}}>\\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}>\\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}>\\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}>\end{matrix}\right\}\textbf{,prod,}<\textbf{\textit{spc}}_1\textbf{\textit{(}}\alpha\textbf{\textit{)}}>,<\textbf{\textit{spc}}_2\textbf{\textit{(}}\alpha\textbf{\textit{)}}>\textbf{]}$$

⋮

spc,<*spc(*α*)*$_n$>,<*gnuz(r)*>, . . .

end

This new option for the diffusion boundary conditions indicates that surface energies control the flows of species to and from the surface and the formation and destruction of the species at the surface. The first entry is the user-assigned number of the enclosure, <*enclnbr(i)*>, facing the surface.

A data sub-block should be entered for each species at the surface (surface-only or diffusion species) signified by the keyword spc. The species name is followed by the Debye frequency, <*gnuz(r)*>, used in determining enclosure-exchange surface reaction rates. Normally that will be the Debye frequency of the material making up the surface, though it may be assigned in other ways. Next comes the surface binding energy, E_c, denoted by the keyword ec, and entered in units of eV at <*value(r)*$_1$>. It may be followed by the solution enthalpy per atom, E_s, indicated by the keyword es. It is entered in units of eV at <*value(r)*$_2$>. These are used in establishing asymmetric diffusion from the surface to the subsurface sites. If the current surface species is a surface-only species, the E_s statement should be omitted because, by definition, surface-only species do not exist in the bulk. Following that are four different reaction types that may (but each need not) be specified. These are (1) *exch* or capture and escape of the surface species to the enclosure, (2) *comb* or the combination of the surface species with another species at the surface to form a different one, (3) *form* or the formation of the present surface species from two different ones, and (4) *diss* or the dissociation of the present surface species into two different ones. Any or all of these four sub-statements should be repeated for each reaction that involves the present surface species. These will now be discussed briefly.

The *exch* option, or escape to the enclosure, is a thermally activated process in which the surface species overcomes its binding potential and escapes in its current molecular state to the enclosure gas. Likewise, molecules in the enclosure overcome the barrier potential and attach to the surface. Required data include

- The identity of the enclosure species, <*espc(α)*$_1$>, by which it will be known in the enclosure domain (note that this may be the same name, but it is a separate entity from the diffusion or surface-only species name)
- The molecular mass in amu of the enclosure species
- The capture barrier energy, E_x, indicated by the keyword ex, of the surface species, entered in units of eV at <*value(r)*$_1$> (see Figure 1). Note that this value should not be less than zero.
- The sticking factor or probability that the arriving molecule will remain at the surface and not immediately return to the gas phase, following the keyword stk. This number will be unity or less.

The second or *comb* option is combination of the present surface species with another to form a third species, denoted by the keyword comb. The only required datum for this specification is the identity of the surface species, <*sspc(α)*>, with which the present species combines to form the new species. Because each reaction is defined and computed in terms of the present surface species, the identity of the species formed is not necessary at this point. This same reaction should appear as a formation reaction for that different species. Activation frequencies for the combination reaction are taken from the diffusivities entered in the *dcoef* statement.

The third option is the formation of the surface species from two different surface species, indicated by the keyword form . Required data here are only the identities of the two surface species that combine to form the present one. These names are entered at <*sspc*$_1$*(α)*> and <*sspc*$_2$*(α)*>. Activation frequencies for the combination reaction are taken from the diffusivities entered in the *dcoef* statement.

The final option for the surface reactions is the dissociation reaction, indicated by the keyword diss. Other data required here include the activation energy for the dissociation, indicated by the keyword eb and entered in units of eV at <*value(r>)*> and, after the keyword prod, the identities of the product species formed. Note that the reverse reaction should *not* be included in the *form* group since the product is formed from a single species, not as a combination.

There must be a spc entry for each species at the surface (maximum 40; 10 diffusion and 30 surface-only). Within each spc block, there must be a data set for each molecular transformation that takes place involving that surface species (max 40).

$$\left\{\begin{matrix}\textit{difbcl}\\ \textit{difbcr}\end{matrix}\right\} = \text{ratedep,encl,}<\textit{enclnbr(i)}>$$

$$\text{spc,}<\textit{sspc}(\alpha)_1>$$

$$\text{exch,}<\textit{espc}(\alpha)>\text{,ksubd,}\left\{\begin{matrix}<\textit{value(r)}>\\ \textit{const,}<\textit{value(r)}>\\ \textit{equ,}<\textit{value(i)}>\\ \textit{tabl,}<\textit{value(i)}>\end{matrix}\right\}$$

$$<\textit{spc}(\alpha)>\text{,ksubr,}\left\{\begin{matrix}<\textit{value(r)}>\\ \textit{const,}<\textit{value(r)}>\\ \textit{equ,}<\textit{value(i)}>\\ \textit{tabl,}<\textit{value(i)}>\end{matrix}\right\}$$

$$[\text{comb,}<\textit{sspc}(\alpha)>]$$

$$[\text{form,}<\textit{sspc}_1(\alpha)>,<\textit{sspc}_2(\alpha)>]$$

$$[\text{diss,}\left\{\begin{matrix}<\textit{value(r)}>\\ \textit{const,}<\textit{value(r)}>\\ \textit{equ,}<\textit{value(i)}>\\ \textit{tabl,}<\textit{value(i)}>\end{matrix}\right\}\text{,prod,}<\textit{sspc}_1(\alpha)>,<\textit{sspc}_2(\alpha)>]$$

$$\vdots$$

$$\text{spc,}<\textit{spc}(\alpha)_n>,\ldots$$

end

This diffusion boundary condition block is very similar to that for the *surfep* option. Several key differences should be pointed out. First, the *exch* sub-block is not optional. It is assumed that when the *ratedep* option for diffusion boundary conditions is selected, there will be conventional recombination and dissociation at the surface. However, as with the *surfdep* option, there may be surface conversion reactions that do not involve exchange with the surrounding gas going on as well. As with the *surfdep* case, it is necessary to enter a block for each surface-only and diffusion species defined in the problem, and a sub-block for each reaction involving that species. Next, instead of these reactions proceeding at rates characterized by the Debye frequency and some activation energies, the exchange recombination and dissociation processes must be given specifications in terms of constants, equations, or tables. Normally the equations and tables will be functions of temperature. For the *comb* and *form* options, rates are taken from diffusivities entered in the *dcoef* statement.

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

$$\textbf{dspc,}\textbf{\textit{<dspc(}}\alpha\textbf{\textit{)}}_{1}\textbf{\textit{>,<espc(}}\alpha\textbf{\textit{)>,}}\textbf{pexp,<value(r)>,solcon,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{array}\right\}$$

$$\vdots$$

$$\textbf{dspc,}\textbf{\textit{<dspc(}}\alpha\textbf{\textit{)}}_{1}\textbf{\textit{>,<espc(}}\alpha\textbf{\textit{)>,}}\textbf{pexp,<value(r)>}\textbf{\textit{,}}\textbf{solcon,}\left\{\begin{array}{l} <\textbf{\textit{value(r)}}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{array}\right\}$$

end

The form of this statement is essentially unchanged. The first entry is the enclosure number into which gas exchange is occurring. Following that, an entry block is made for each diffusion species. Surface-only species are not defined for *lawdep* boundary conditions. After giving the diffusing species name, the homonuclear enclosure species formed in the reaction (or corresponding identity enclosure species if Henry's law applies) with the current diffusion species is identified. Note that it is possible to treat just one part of a heteronuclear molecule as a Henry's law reaction when there is no homonuclear reaction available. Then, following the keyword pexp, the exponent from the solution law

$$C_{<sspc(\alpha)>} = solcon \times P_{<espc(\alpha)>}{}^{pexp}$$

is provided followed by the keyword solcon, and the solubility for the homonuclear (or identity) molecule of that species in the material. The exponent *pexp* is normally 0.5 for Sieverts' law solubility and 1.0 for Henry's law solubility. This solubility rule is used to establish surface concentrations of the atomic species corresponding to the specified homonuclear (or identity) molecules under the *lawdep* boundary condition. A *dspc* block must be entered for each diffusion species in the problem. Surface-only species are not entered because their reactions will have no effect on the surface concentrations of diffusion species.

$$\left\{\begin{array}{l}\textbf{\textit{difbcl}}\\ \textbf{\textit{difbcr}}\end{array}\right\} = \textbf{sconc}$$

$$\textbf{dspc,}<\textbf{\textit{dspc}}(\alpha)_1>\textbf{,conc,}\left\{\begin{matrix} <\textbf{\textit{value(r)}}> \\ \textbf{\textit{const,}}<\textbf{\textit{value(r)}}> \\ \textbf{\textit{equ,}}<\textbf{\textit{value(i)}}> \\ \textbf{\textit{tabl,}}<\textbf{\textit{value(i)}}> \end{matrix}\right\}$$

$$\vdots$$

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

This statement is modified from its earlier form. There must be a (d*spc*) entry block for each of the diffusion species defined in the problem, but no surface-only species are entered. Following the surface species name is the surface concentration specification for that species. That may be in constant, equation, or tabular form. Table look-ups assume time is the independent variable in the relation defined by the table while time or temperature may be independent in equations. A difference from the previous version is that neither the enclosure number nor the stoichiometry (keyword *nsurfs*) are needed because the *sconc* boundary condition is for setting the surface concentration of a diffusion species. Because it is strictly an unphysical boundary condition, provided only to examine certain hypothetical situations, no molecular flows to the surrounding enclosures are calculated. However, diffusive fluxes of diffusion species are determined and, if requested, recorded in the *pltdata* file.

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

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

There is no change in this statement

$$\left\{\begin{matrix} \textbf{\textit{difbcl}} \\ \textbf{\textit{difbcr}} \end{matrix}\right\} \textbf{ = nonflow,end}$$

There is no change in this statement.

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

There is no change in this statement.

Equation Input

No changes are incorporated into the *Equation Input* section.

Table Input

No changes are presently incorporated into this section.

Control Input

No changes are presently incorporated into this section.

Plot Input

Add the following optional statement to identify the surface species to be included in the *pltdata* file:

[sname = [<$sspc(\alpha)_1$>,...,<$sspc(\alpha)_n$>],end]

No other changes are presently incorporated in the statement form, but the *trapinv* specification under *dplot* will include trapping data for all of the traps defined for the diffusion segment.

Terminal Declaration

end of data

No change in this statement.

CONCLUSION

The changes made to the TMAP code to convert it to TMAP2000 are substantial and significant. They include allowing for multiple traps, allowing for surface species and surface-energy controlled surface kinetics, and increasing the size and complexity of problems that can be addressed. The added features will improve the functional usefulness of the code for a variety of applications where previous versions have not been able to provide an adequate representation.

Next steps for code development will include code verification and validation, whereas at this writing it is still in the test phase. Tasks further in the future may be to add time dependence to trap concentrations and graphical output, eliminating the need to manually extract data from the PLTDATA file.