SANDIA REPORT

SAND90-8003 • UC-401
Unlimited Release
Printed August 1990

Surface Chemkin (Version 3.7): A Fortran Package for Analyzing Heterogeneous Chemical Kinetics at a Solid-Surface—Gas-Phase Interface

M. E. Coltrin, R. J. Kee, F. M. Rupley

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94551
for the United States Department of Energy
under Contract DE-AC04-76DP00789

DO NOT MICROFILM
COVER

SF2900Q(8-81)

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from:

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge TN 37831

Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

NTIS price codes
Printed copy: *A07*
Microfiche copy: A0[illegible]

SAND--90-8003
DE91 010840

SAND90-8003
Unlimited Release
Printed August 1990

SURFACE CHEMKIN (VERSION 3.7): A FORTRAN PACKAGE FOR ANALYZING HETEROGENEOUS CHEMICAL KINETICS AT A SOLID-SURFACE – GAS-PHASE INTERFACE

Michael E. Coltrin
Surface Processing Sciences Division
Sandia National Laboratories
Albuquerque, New Mexico 87185

Robert J. Kee and Fran M. Rupley
Computational Mechanics Division
Sandia National Laboratories
Livermore, California 94551

ABSTRACT

This document is the user's manual for the Surface Chemkin package. Together with the Chemkin-II package, this software facilitates the formation, solution, and interpretation of problems involving elementary heterogeneous and gas-phase chemical kinetics in the presence of a solid surface. The package consists of two major software components: an Interpreter and a Surface Subroutine Library. The Interpreter is a program that reads a symbolic description of an elementary, user-specified chemical reaction mechanism. One output from the Interpreter is a data file that forms a link to the Surface Subroutine Library, which is a collection of about fifty modular Fortran subroutines that may be called from a user's application code to return information on chemical production rates and thermodynamic properties.

ACKNOWLEDGMENTS

We especially appreciate the efforts of our colleague Dr. Randy Creighton. Through frequent and extended discussions, in which he shared his keen insight on surface chemistry, he made many valuable contributions to the design and implementation of our surface-reaction formalism. Drs. Reggie Mitchell and Jim Miller have also influenced our understanding and formulation of the gas-phase and gas-surface interaction chemistry. We are most grateful for the comments and suggestions of our colleagues who have used the software during its development. Drs. Peter Glarborg, Joe Grcar, Bill Houf, and Rich Larson were actively writing applications codes to use Surface Chemkin at the same time we were continuing to modify the software to enhance its capabilities. Because the software underwent many significant changes, its use during development was often frustrating. Drs. Mark Allendorf, Bill Breiland, Greg Evans, Pauline Ho, and Reggie Mitchell were all actively using applications codes in their own research projects. They also had to live with periods of day-to-day changes in the software. We also thank Dr. Harry Moffat for his critical reading of this manual and a number of suggestions for improvement.

This work was supported primarily by the U. S. Department of Energy, Office of Basic Energy Sciences, Division of Materials Science and Division of Chemical Sciences. It was also supported by the Defense Advanced Research Projects Agency (DARPA) Materials Science Program.

CONTENTS

ILLUSTRATIONS

TABLES

NOMENCLATURE

Sym.	Meaning	CGS Units
a_i	Pre-exponential factor in sticking coefficient expression	none
a_k	Activity of k^{th} bulk-phase species	none
a_{mk}	Coefficients to fits of thermodynamic data	depends on m
a_k^o	Standard state specific Helmholtz free energy for the k^{th} species	ergs/g
A_k^o	Standard state Helmholtz free energy for the k^{th} species	ergs/mole
A_i	Pre-exponential factor in the rate constant of the i^{th} reaction	depends on reaction
b_i	Temperature exponent in sticking-coefficient expression	none
c_i	Activation energy in sticking-coefficient expression	[cal/mole]*
c_{p_k}	Specific heat at constant pressure of the k^{th} species	ergs/(g K)
$C_{p_k}^o$	Standard state specific heat at constant pressure of the k^{th} species	ergs/(mole K)
D_{kj}	Ordinary multicomponent diffusion coefficients	cm^2/sec
D_k^T	Thermal diffusion coefficient	g/(cm-sec)
E_i	Activation energy in the rate constant of the i^{th} reaction	[cal/mole]*
g_k^o	Standard state specific Gibbs free energy for the k^{th} species	ergs/g
G	Bulk growth rate	cm/sec
G_k^o	Standard state Gibbs free energy for the k^{th} species	ergs/mole
h_k	Specific enthalpy of the k^{th} species	ergs/g
H_k^o	Standard state enthalpy of the k^{th} species	ergs/mole
H_k	Enthalpy of the k^{th} species	ergs/mole
i	Reaction index	

---

*By default, Surface Chemkin uses activation energies in calories instead of ergs.

		CGS Units
I	Total number of reactions	
k	Species index	
k_{f_i}	Forward rate constant of the i^{th} reaction	depends on reaction
k_{r_i}	Reverse rate constant of the i^{th} reaction	depends on reaction
K	Total number of species	
K_b	Total number of bulk species	
K_b^f	Pointer to the index of the first bulk-phase species in the first bulk phase	
K_b^l	Pointer to the index of the last bulk-phase species in the last bulk phase	
K_g	Total number of gas-phase species	
K_g^f	Pointer to the index of the first gas-phase species	
K_g^l	Pointer to the index of the first gas-phase species	
K_s	Total number of surface species	
K_s^f	Pointer to the index of the first surface-phase species in the first surface phase	
K_s^l	Pointer to the index of the first surface-phase species in the last surface phase	
K_{phase}	Vector containing the number of species in each phase	
K_{c_i}	Equilibrium constant in concentration units for the i^{th} reaction	depends on reaction
K_{p_i}	Equilibrium constant in pressure units for the i^{th} reaction	depends on reaction
M	Number of coefficients in polynomial fits to C_p^o/R	
N_b	Total number of bulk mixtures	
N_b^f	Pointer to the index of the first bulk phase	
N_b^l	Pointer to the index of the last bulk phase	
N_s	Pointer to the index of the first surface phase	

		CGS Units
N_s^f	Pointer to the index of the last surface phase	
N_s^l	Total number of surface site types	
P	Pressure	dynes/cm^2
P_{atm}	Pressure of one standard atmosphere	dynes/cm^2
q_i	Rate of progress of the i^{th} reaction	moles/(cm^{2}sec)
R	Universal gas constant	ergs/(mole K)
R_c	Universal gas constant, in same units as activation energy E_i	[cal/(mole K)]*
$\dot{s}_k$	Production rate of a surface species	moles/(cm^{2}sec)
s_k^o	Standard state specific entropy of the k^{th} species	ergs/(g K)
S_k^o	Standard state entropy of the k^{th} species	ergs/(mole K)
t	Time	sec
T	Temperature	K
u	Convective velocity, Stefan flow velocity	cm/sec
u_k	Specific internal energy of the k^{th} species	ergs/g
U_k	Internal energy of the k^{th} species	ergs/mole
U_k^o	Standard state internal energy of the k^{th} species	ergs/mole
V	Volume	cm^3
V_k	Diffusion velocity	cm/sec
W_k	Molecular weight of k^{th} species	g/mole
$\overline{W}$	Mean molecular weight of a mixture	g/mole
X_k	Mole fraction of the k^{th} species	none
$[X_k]$	Molar concentration of k^{th} species	moles/cm^3
Y_k	Mass fraction of the k^{th} species	none
Z_k	Site fractions of the k^{th} species of a site	none

GREEK		CGS Units
β_i	Temperature exponent in the rate constant of the i^{th} reaction	none
Γ_n^o	Standard-state site density	moles/cm^2
Γ_n	Site density	moles/cm^2
$\dot{\Gamma}_n$	Production rate for a surface phase	moles/(cm^2 sec)
λ	Thermal conductivity	erg/(cm K sec)
ρ	Mass density	g/cm^3
ν_{ki}	Stoichiometric coefficients of the k^{th} reaction, $\nu_{ki} = \nu''_{ki} - \nu'_{ki}$	
ν'_{ki}	Stoichiometric coefficients of the k^{th} reactant species in the i^{th} reaction	
ν''_{ki}	Stoichiometric coefficients of the k^{th} product species in the i^{th} reaction	
$\dot{\omega}_k$	Chemical production rate of the k^{th} species	mole/(cm^2 sec)
σ_k	Number of surface sites a surface species occupies	
ϵ_{ki}	Coverage parameter	[cal/mole]cm^2/mole
η_{ki}	Coverage parameter	cm^2/mole
μ_{ki}	Coverage parameter	none
Φ	Dependent variable in an application code	
χ_k	Chemical symbol of the k^{th} species	

SURFACE CHEMKIN (VERSION 3.7)*: A FORTRAN PACKAGE FOR ANALYZING HETEROGENEOUS CHEMICAL KINETICS AT A SOLID-SURFACE – GAS-PHASE INTERFACE

I. INTRODUCTION

Heterogeneous reaction at the interface between a solid surface and adjacent gas is central to many chemical processes. Our purpose for developing the software package Surface Chemkin was motivated by our need to understand the complex surface chemistry in chemical vapor deposition systems involving silicon, silicon nitride, and gallium arsenide. However, we have developed the approach and implemented the software in a general setting. Thus, we expect it will find use in such diverse applications as chemical vapor deposition, chemical etching, combustion of solids, and catalytic processes, and for a wide range of chemical systems. We believe that it provides a powerful capability to help model, understand, and optimize important industrial and research chemical processes.

The Surface Chemkin software is designed to work in conjunction with the Chemkin-II[1] software, which handles the chemical kinetics in the gas phase. It may also be used in conjunction with the Transport Property Package,[2,3] which provides information about molecular diffusion. Thus, these three packages provide a foundation on which a user can build applications software to analyze gas-phase and heterogeneous chemistry in flowing systems.

These packages should not be considered "programs" in the ordinary sense. That is, they are not designed to accept input, solve a particular problem, and report the answer. Instead, they are software tools intended to help a user work efficiently with large systems of chemical reactions and develop Fortran representations of systems of equations that define a particular problem. It is up to the user to solve the problem and interpret the answer. A general discussion of this structured approach for simulating chemically reacting flow can be found in Kee and Miller.[4]

---

* This document describes the features in version 3.7 of Surface Chemkin. We expect that this software package will continue to evolve, and thus later versions of the software may render portions of this document out of date.

Structure and Use of Surface Chemkin

Using the Surface Chemkin package is analogous to using the Chemkin-II[1] package, and the Surface Chemkin package can only be used after the Chemkin Interpreter has been executed.* Therefore, it is necessary to be familiar with Chemkin-II before the Surface Chemkin package can be used effectively. The Chemkin Interpreter introduces the chemical elements that are used in either the gas-phase reaction mechanism or the surface-reaction mechanism. Gas-phase species (which can appear in surface reactions) are also introduced with the Chemkin Interpreter. Thus if a gas-phase species appears in the surface-reaction mechanism but not in the gas-phase mechanism, then the user must identify this species in the Chemkin Interpreter.

Like Chemkin-II, the Surface Chemkin package is composed of two blocks of Fortran code and two files:

- the Surface Interpreter (code)
- the Surface Subroutine Library (code)
- the Thermodynamic Database (file; may be same file used by Chemkin-II)
- the Surface Linking File (file).

To apply Surface Chemkin to a problem, the user first writes a Fortran program that describes his particular set of governing equations. To aid this programming effort, the user can call Chemkin-II and Surface Chemkin subroutines that define the terms in his equations relating to equation of state, chemical production rates, and thermodynamics, and then combine the results to define the problem.

After running the Chemkin Interpreter, the user runs the Surface Interpreter, which first reads the user's symbolic description of the surface-reaction mechanism and then extracts from the Thermodynamic Database the appropriate thermodynamic information for the species involved.[6] Chemkin-II and the Surface Chemkin can share a common database. The database has essentially the same format as that used by the NASA complex chemical equilibrium code of Gordon and McBride.[7] The output of the Surface Interpreter is the Surface Linking File, which contains all the pertinent information on the elements, species, and reactions in the surface reaction mechanism. Information on gas-phase species comes from the Chemkin Linking File, and thus is duplicated in both linking files.

The Surface Linking File is read by an initialization subroutine that is called from the user's code. The purpose of the initialization is to create three data arrays (one

---

* Caution: Surface Chemkin work only with the newer Chemkin-II, not the original Chemkin package[5].

integer, one floating point, and one character data type) for use internally by the other subroutines in the Surface Subroutine Library.

The Surface Subroutine Library has approximately fifty subroutines that return information on elements, species, reactions, thermodynamic properties, and chemical production rates. Generally, the input to these routines will be the state of gas and the surface—pressure, temperature, and species composition. The species composition is specified in terms of gas-phase mole fractions, surface site fractions, and bulk-phase activities.

Example

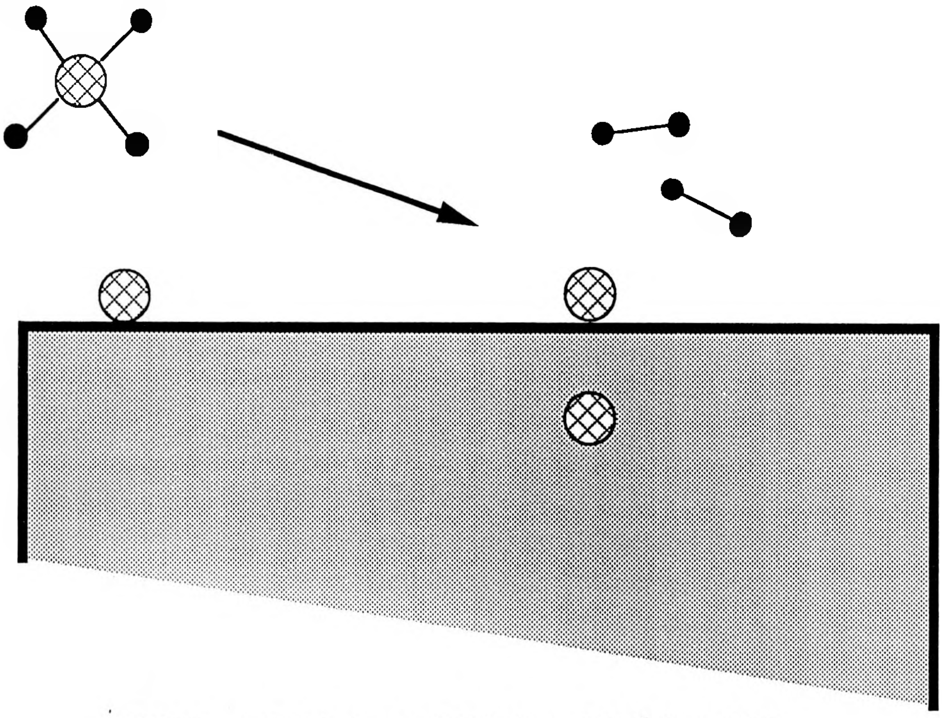
We illustrate the use of Surface Chemkin by a simple example involving deposition of silicon. The surface-reaction mechanism is shown in Fig. 1 as it appears for the input file to the Surface Interpreter. The first two lines identify a site type called "SILICON" that has a site density of 1.66×10^{-9} moles/cm^2. Only one species, SI(S), exists on this site type. The bulk material is identified as SI(B), and it has a mass density of 2.33 g/cm^3. This is a very simple example that has only one site type occupied by only one species and only one pure bulk material. In general, however, there could be many different site types, each of which could be occupied by a variety of species. Furthermore, there could be several bulk-phase mixtures that are composed of several species. Examples of all these possibilities are given later in the manual.

The reaction mechanism itself is listed next. The symbol => in each reaction expression indicates that all the reactions are irreversible. The three numbers following each reaction expression are its Arrhenius rate parameters (pre-exponential factor, temperature exponent, and activation energy).

All of the reactions in the mechanism have the same form: a gas-phase species reacting on a silicon site. The reaction of silane at the surface is illustrated in Fig. 2. Each silicon-containing gas-phase species can react on an atomic surface site, SI(S), to deposit a silicon atom as SI(B) and release hydrogen back into the gas phase. We have included SI(S) as both a reactant and a product to indicate that a "site" must be available at which the gas-phase species can react. In the example, however, the surface silicon SI(S) is distinguished from the bulk deposit SI(B) only by virtue of its position as the top-most atom at the surface. Therefore, each time a SI(S) is consumed by a reaction the bulk becomes one atom thicker and the silicon atom that just left the gas now forms the top-most surface layer, i.e., SI(S). For this mechanism, the SI(S) could have been just as well left out of the mechanism entirely. However, if other gas-phase species had been present (say phosphine carrying phosphorous as a dopant), these species could compete for the available silicon sites on the surface. Thus, by

```
SITE/SILICON/ / SDEN/1.66E-09/
    SI(S)
BULK      SI(B) /2.33/
REACTIONS
    SIH4 + SI(S) => SI(S) + SI(B) + 2H2               1.05E17       0.5 40000
    SI2H6 +2SI(S) => 2SI(S) + 2SI(B) + 3H2            4.55E26       0.5 40000
    SIH2 + SI(S) => SI(S) + SI(B) + H2                3.9933E11     0.5 0
    SI2H2 +2SI(S) => 2SI(S) + 2SI(B) + H2             1.7299E20     0.5 0
    2SI2H3 +4SI(S) => 4SI(S) + 4SI(B) + 3H2           6.2219E37     0.5 0
    H2SISIH2 +2SI(S) => 2SI(S) + 2SI(B) + 2H2         1.7007E20     0.5 0
    2SI2H5 +4SI(S) => 4SI(S) + 4SI(B) + 5H2           6.1186E37     0.5 0
    2SIH3 +2SI(S) => 2SI(S) + 2SI(B) + 3H2            2.3659E20     0.5 0
    2SIH +2SI(S) => 2SI(S) + 2SI(B) + H2              2.4465E20     0.5 0
    SI + SI(S) => SI(S) + SI(B)                       4.1341E11     0.5 0
    H3SISIH +2SI(S) => 2SI(S) + 2SI(B) + 2H2          1.7007E20     0.5 0
    SI2 +2SI(S) => 2SI(S) + 2SI(B)                    1.7607E20     0.5 0
    SI3 +3SI(S) => 3SI(S) + 3SI(B)                    8.6586E28     0.5 0
END
```

Figure 1. Sample Reaction Mechanism as Read by the Surface Chemkin Interpreter.



$$SiH_4 + Si(s) \rightarrow Si(s) + Si(b) + 2H_2$$

Figure 2. Illustration of Gas-Phase Silane Reacting at a Surface to Deposit a Silicon Atom and Release Two Hydrogen Molecules into the Gas Phase.

writing the reactions as we have, we have left open the possibility for other species to occupy surface sites and thus inhibit the deposition of silicon.

As an example of the full use of Surface Chemkin, assume that in an applications program we wish to evaluate a boundary condition, concerning the energy balance at a surface of an isothermal particle, which might take the following form:

$$\lambda \frac{\partial T}{\partial x} - \sum_{k=1}^{K_g} \rho Y_k \left(V_k + u\right) h_k = \sigma \epsilon \left(T^4 - T_0^4\right) + \sum_{k=1}^{K_s} \dot{s}_k W_k h_k$$

The dependent variables in this expression are the temperature T, gas-phase mass fractions Y_k, and convective velocity u. The surface site fractions and the bulk-species activities are also dependent variables, but do not appear explicitly in the expression. The first term describes thermal conduction to the surface from the gas phase. The thermal conductivity λ would be evaluated by a call to the Transport Library,[3] and the temperature gradient could be evaluated by finite differences. The second term concerns the diffusive and convective flux of energy by gas-phase species at the surface. The mass density ρ and the gas-phase enthalpies h_k would be evaluated by calls to the Chemkin Library. The gas-phase species diffusion velocities V_k would be evaluated in terms of diffusion coefficients that are obtained from the Transport Package and finite difference approximations to the species gradients. The first term on the right-hand side concerns the thermal radiation to or from the surface.

We now concentrate on the final term, which concerns the energy generated or consumed from surface reaction. The summation is over all surface and bulk species, and the factors in the summation are the production rate of surface species by surface reaction, $\dot{s}_k$, the species molecular weights, W_k, and the enthalpies of the surface and bulk species, h_k. The Fortran representation of this term begins with Surface Library subroutine calls (the output variables are underlined to help distinguish them):

```
CALL SKINIT(LSIWK, LSRWK, LSCWK, LINKSK, LOUT, ISKWRK, RSKWRK, CSKWRK)
CALL SKWT(RSKWRK, WT)
CALL SKHMS(T, ISKWRK, RSKWRK, HMS )
CALL SKRAT(P, T, ACT, SDEN, ISKWRK, RSKWRK, WDOT, SDOT)
```

The complete details for these calls are explained in later sections of this document, the object here being to illustrate the relative simplicity of a Surface Chemkin application. Briefly, the first call is to the initialization subroutine SKINIT, which reads the Surface Linking File created by the Surface Interpreter and creates the three work arrays. LSIWK, LSRWK, and LSCWK are the dimensions provided by the user for the data arrays ISKWRK, RSKWRK, and CSKWRK. LINKSK is the logical file number of the Surface Linking File, and LOUT is the logical file number for printed diagnostic and error messages. In the remaining calls, P and T are the pressure and temperature. The array ACT contains the gas-phase mole fractions, the surface site fractions, and the bulk species activities. The output variable arrays, HMS and WDOT, correspond to the factors in the summation, i.e., HMS = h_k, and WDOT = $\dot{s}_k$.

The Fortran representation of the summation in the last term, given by combining the results of the above subroutine calls, is simply

```
      SUM=0.0
      DO 100 K=FIRST_SURFACE_SPECIES, LAST_BULK_SPECIES
         SUM = SUM + WDOT(K)*WT(K)*HMS(K)
  100 CONTINUE
```

The species indices FIRST_SURFACE_SPECIES and LAST_BULK_SPECIES are also available from a call to the Surface Library, which Chapters VII and VIII explain in detail.

Transportability

The Surface Chemkin package was developed on VAX and Cray computers. However, we have not taken advantage of any special machine-dependent features. Written entirely in ANSI standard Fortran-77, the code is easily transportable to other computer systems. Since double-precision code is often required on small-word-length (i.e., 32-bit word) computers, we provide both single- and double-precision versions of the source code.

Organization of this Report

Chapter II introduces the formalism developed to describe surface chemistry behavior. Unlike for gas-phase chemistry, where much software has been written to analyze mass-action kinetics and chemically reacting flow, elementary heterogeneous reactions are nowhere else (to our knowledge) treated with the generality provided in this package. For the treatment we first had to define a systematic convention to translate heterogeneous reaction ideas into a form that was amenable to efficient computation.

In the spirit of Chemkin-II, Chapter III is a compendium of important equations in heterogeneous chemical kinetics. Many of the equations are simply definitions; but, in any case, derivations are either sketchy or not given. Although some readers will find many of the equations quite familiar, we find it useful to have them stated concisely in one document. For most equations, the package contains a subroutine that, when given the variables on the right-hand side, returns the variable on the left. Below some of the equation numbers is stated (in brackets) the name of the subroutine that provides information about that equation.

Using Chemkin-II and Surface Chemkin (and possibly the Transport Package) requires the manipulation of many programs and files. Chapter IV explains the mechanics of using these software packages and describes the job-control logic for running a typical problem.

Chapter V explains the Surface Chemkin Interpreter and how to set up the required symbolic input to define a reaction mechanism. We have allowed the possibility of including multiple site types and multiple mixtures of bulk species. Each site type and bulk mixture may contain several species. Therefore, the data structures needed to refer to the phases and the species can be complex. Chapter VI provides detailed information on the computational data structures that we use to refer to phases and species in each phase.

Chapters VII and VIII describe the Surface Subroutine Library, Chapter VII being composed of short descriptions for quick reference and Chapter VIII (an alphabetical listing) explaining the input and output in the call sequence. To demonstrate Surface Chemkin explicitly, Chapter IX goes through a sample problem in detail.

Appendix A defines the storage allocation of three work arrays that are created from the Linking File. With this information, it is possible for a user to create new subroutines for the library to suit a specialized need that was not anticipated in the current library.

II. DEVELOPMENT OF SURFACE FORMULATION

In this section we discuss the mathematical formalism developed to describe surface kinetics for events such as adsorption, desorption, surface reactions, and deposition. This formalism is essentially a set of rules for keeping track of surface species concentrations, conservation of mass and surface sites, mass-action kinetics, and rates (such as deposition rates).

For this discussion we define three types of species: gas-phase, surface, and bulk. The first is a species in the gas phase above the surface, which might be denoted in a reaction by (g). A surface species, perhaps denoted by (s), is defined to be the chemical species on the top-most layer of the solid, i.e., at the solid-gas interface. Each surface species occupies one or more "sites." (The total number of sites is often assumed to be conserved.) Any species in the solid below the top-most layer is defined to be a "bulk" species and could be denoted by (b). In writing elementary reactions for a surface mechanism in a kinetic model, mass, elemental composition, and charge are all conserved.

There can be more than one type of site on the surface. For example, one could specify that a surface consists of "ledge" sites and "plane" sites. The number of sites of each type would be characteristic of the crystal face. In our formalism there can be any number of site types. One may define a species that only resides on a certain type of site. For example, the properties of a hydrogen atom on a ledge site might be different from a hydrogen on a plane site, and they could be specified as different species (even though their elemental composition is the same). The population of different species occupying a given type of site is specified by site fractions. The sum of the site fractions of the species on a given site is 1. (Thus an "open site" is considered as a species.)

In the bulk there can be different types of bulk species. The simplest consists of a pure condensed species. There can be any number of pure bulk species. It is also possible to specify a bulk mixture with components A and B. The composition of the bulk phase may be input by the user by specifying the activities of each of the bulk-phase components.

The activity of a bulk species is defined in terms of the following equation for the chemical potential:

$$\mu_k(T, P, \mathbf{X}) = {\mu_k}^*(T) + RT \ln\left(a_k(T, P, \mathbf{X})\right),$$

where ${\mu_k}^*$ is the standard state chemical potential of species k at temperature T and at the standard pressure, 1 atm. The vector $\mathbf{X}$ represents the mole fractions of the species. Two conventions are normally used to complete the specification of the activity coefficient:

1. If the standard state is defined as being a pure bulk phase of k at temperature T and 1 atm, then a_k is further defined to approach 1 as X_k approaches 1 at 1 atm (Raoult's Law).

2. If the standard state is defined as being a hypothetical state of species k in infinite dilution in bulk-phase species j at temperature T and 1 atm, then a_k is further defined to approach 1 as X_k approaches 0 at 1 atm (Henry's Law).

Both conventions for the standard state work with Surface Chemkin, as do any other definitions that conform to the formalism expressed by the above equation for μ. $\mu_k{}^*(T)$ is specified through the entry for species k in the thermodynamics data file. The value of $a_k(T, P, \mathbf{X})$ is required as input to all Surface Chemkin subroutines that calculate bulk phase thermodynamic quantities. Therefore, users can construct their own subroutines to calculate $a_k(T, P, \mathbf{X})$, possibly incorporating models for non-ideality of the bulk phase, and can have the consequences properly incorporated into the surface kinetics mechanism. Although the activities of all components of an ideal solution must sum to 1, this condition is not enforced in Surface Chemkin.

Since Surface Chemkin allows for a number of different types of species (gas species, any number of types of surface sites, species residing on surface sites, pure bulk species, bulk mixtures, and species present in a bulk mixture), it is necessary to be able to keep track of them. We use the notion of different physical "phases" to group the chemical species in a problem. Our nomenclature corresponds to that of Eriksson,[8] which has been extended to account for surface sites. The order in which we discuss the phases is the order in which Surface Chemkin groups them.

Phase number 1 is the gas phase. Information about species in the gas phase is passed to Surface Chemkin from the gas-phase Chemkin-II interpreter. The mole fractions of the gas-phase species correspond to species activities, mentioned below.

We consider every type of surface site to be a distinct "phase." If there are N_s types of sites specified, then phases 2 through $N_s + 1$ are these sites. The user can specify the names of chemical species that exist only on a given site type. The site fractions of all the species that can exist on a given type of site (phase) sum to 1. The site fractions also correspond to activities.

The next type of phase is a bulk mixture. If a given problem has N_b different types of bulk mixtures, then these are considered to be phases $N_s + 2$ through $N_s + N_b + 1$. The user specifies the names of the species that can exist in a given bulk mixture. The amounts of these species are determined indirectly by their activities, which the user supplies. A limiting case is a pure bulk species, which is treated like a bulk mixture with only one chemical species, whose activity is unity if the chemical potential does not depend on pressure.

We now consider in more detail how to write chemical reactions involving surface and bulk species. A chemical species on the top-most layer of the solid, i.e., a surface species, occupies a site. For example, an arsine molecule adsorbed on a surface could occupy a site, and might be denoted $AsH_3(s)$. Another example might be a bare gallium atom, Ga(s), on top of a gallium arsenide crystal. What happens if another species, say a gas-phase AsH_3, lands on top of the Ga(s) (see Fig. 3)? In this case the gallium atom that was at the surface is covered up, so it is no longer a surface species. In our nomenclature it has become a bulk species. The adsorbed AsH_3 now occupies the topmost layer on this site, so it has become the surface species $AsH_3(s)$. In our formalism, we would write the adsorption reaction in Fig. 3 as

$$AsH_3(g) + Ga(s) \longrightarrow AsH_3(s) + Ga(b). \tag{1}$$

In this reaction, the number of sites included on the left-hand side of the reaction equals the number on the right-hand side; the reaction conserves sites.

Suppose that we had wanted to describe the reverse reaction, i.e., desorption of AsH_3 from the surface. We would then have to write the reaction as

$$AsH_3(s) + Ga(b) \longrightarrow AsH_3(g) + Ga(s). \tag{2}$$

Here, Ga(b) is included as a reactant in order to achieve site and elemental balance. We denote the formalism described in reactions (1) and (2) as the Atomic Site Formalism, i.e., Ga(s) is included explicitly either as a reactant or a product.

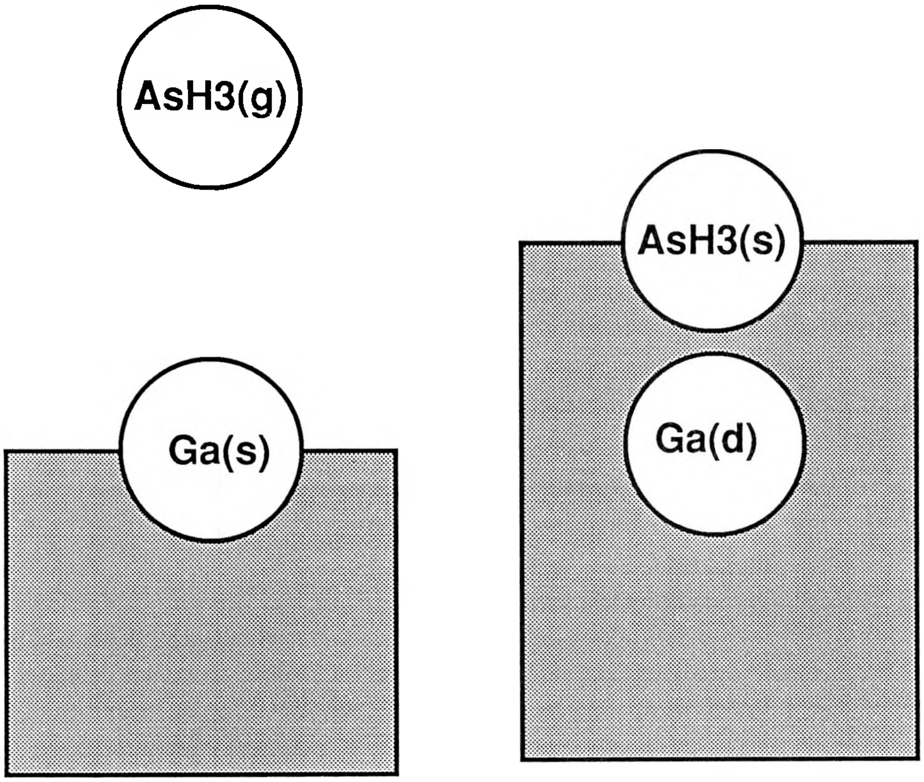


Figure 3. Illustration of an Adsorption Reaction using the Atomic Site Formalism.

An alternate way of posing the above example is to look at the situation on the left side of Fig. 3 not as having a surface gallium atom on a site, but to say that this is really an "open" site at which some event may take place (see Fig. 4). We would write the reaction of Fig. 4 as

$$\mathrm{O(s)} + \mathrm{AsH_3(g)} \longrightarrow \mathrm{AsH_3(s)}, \tag{3}$$

where O(s) denotes an open site. Since O(s) contains no elements (it is empty), this reaction conserves both sites and elements. We denote the formalism described in reaction (3) as the Open Site Formalism.

The Atomic Site and Open Site Formalisms are equally valid ways of stating these surface reactions, and either is allowed by the Surface Chemkin code. Personal preference or, perhaps, the nature of a particular problem would dictate one over the other. Note that an "open" site must be considered as a species.

What are the thermochemical implications of reactions such as (1) and (2)? In the Atomic Site Formalism, the interpretation is straightforward. In reaction (1) we have converted $\mathrm{AsH_3(g)}$ and Ga(s) into $\mathrm{AsH_3(s)}$ and Ga(b). Thus, the change in a thermochemical property, e.g., ΔH_{rx}, is just the difference in the heats of formation of the products and the reactants. What about in the Open Site Formalism? What are the properties of O(s), the open site? Because these two formalisms describe an identical physical event, it is evident the properties of the open site must be related to those of Ga(b) and Ga(s). For example, the heat of formation of this open site is just

$$\Delta H_f(\mathrm{O(s)}) = \Delta H_f(\mathrm{Ga(s)}) - \Delta H_f(\mathrm{Ga(b)}). \tag{4}$$

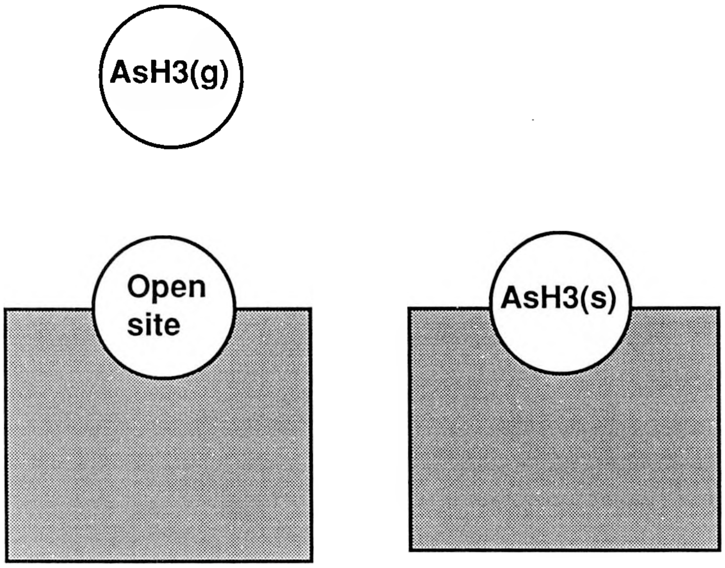


Figure 4. Illustration of an Adsorption Reaction using the Open Site Formalism.

III. CHEMICAL RATE AND THERMODYNAMIC EXPRESSIONS

This chapter lists expressions and equations that are useful in formulating chemically reacting flow problems. For many expressions and equations the subroutine that evaluates it is named.

Species can exist in the gas phase, on surface sites, or in bulk mixtures. In some cases it is desirable to refer to information about species without regard to the phases, and in other cases it is desirable to determine information about species in only one phase or group of phases. Therefore, before beginning to discuss our formalism in terms of mathematical expressions, we introduce a nomenclature that facilitates manipulating species information.

Information about a species (say a thermodynamic property) is presumed to be available in ordered arrays beginning with the first gas-phase species, continuing through the surface species, and ending with the last bulk species. In the expressions and equations below we presume that there are K species, and we use the index k to refer to a specific species. There are K_g gas-phase species, which, by convention, are always the first entries in the species arrays. The index of the first gas-phase species is K_g^f ($K_g^f = 1$ by our convention) and the last gas-phase species index is K_g^l ($K_g^l = K_g$). Thus the gas-phase species indices are $K_g^f \leq k \leq K_g^l$. In a similar way surface species indices are in the range $K_s^f \leq k \leq K_s^l$ and bulk species are in the range $K_b^f \leq k \leq K_b^l$. The surface species may be arranged on any number of sites, and the bulk species may exist in any number of bulk mixtures. Furthermore, situations can occur in which there are no surface species and/or no bulk species.

As discussed in Chapter II, the species are grouped in "phases." The first is the gas phase, whose index $n = 1$. The next N_s phases (if they are present) are the surface sites, whose phase indices are bounded by $N_s^f \leq n \leq N_s^l$. The final N_b phases are the bulk mixtures, whose indices are bounded by $N_b^f \leq n \leq N_b^l$. In each phase n there are $K_{\text{phase}}(n)$ species, and those species have indices in the range $K_{\text{phase}}^f(n) \leq k \leq K_{\text{phase}}^l(n)$.

Concentration Units

In a later section we discuss mass-action kinetics, where the rate of progress of reactions depends on molar concentrations either in the gas phase or on surface sites and activities in the bulk phases. However, for the purposes of formulating and solving the conservation equations that describe physical situations, it is often more natural to use gas-phase mass fractions and surface site fractions as dependent variables. Therefore, it is important to establish the rules for converting between the different ways to describe the composition of the gas and the surface.

For the gas-phase species the molar concentrations $[X_k]$ (in moles/cm^3) are written as

$$[X_k] = Y_k \rho / W_k \qquad (k = K_g^f, \ldots, K_g^l). \tag{5}$$

where the Y_k are the mass fractions, ρ is the gas-phase mass density, and the W_k are the molecular weights.

On the surface sites we can describe the composition in terms of an array of surface species site fractions Z_k. This array is of length K_s. It is composed of N_s subunits of the site fractions of each of the species on a given site n. The site fractions on each site are normalized:

$$\sum_{k=K^f(n)}^{K^l(n)} Z_k(n) = 1 \qquad (n = N_s^f, \ldots, N_s^l). \tag{6}$$

The sum runs from the first species in phase n to the last species in phase n. The surface molar concentration of a species is then

$$[X_k] = Z_k(n) \Gamma_n / \sigma_k(n), \tag{7}$$

where Γ_n is the density of sites of phase n (in moles/cm^2) and $\sigma_k(n)$ is the number of sites that each species k occupies.

In the bulk phases, the compositions of the mixtures are specified by the bulk species activities. The activities in a given bulk phase sum to 1. In the limiting case of a bulk phase consisting of only one species (a pure bulk phase) the activity of that species is 1. The activity a_k of bulk species k can appear in the chemical rate expression. For the sake of parallelism, we adopt the nomenclature for bulk species:

$$[X_k] = a_k, \qquad (k = K_b^f, \ldots, K_b^l) \tag{8}$$

and note that

$$\sum_{k=K^f(n)}^{K^l(n)} [X_k](n) = 1 \qquad (n = N_b^f, \ldots, N_b^l) \tag{9}$$

Surface Site Nonconservation

It is possible that a given surface reaction (or reactions) will not conserve the number of surface sites. In that case the density of sites Γ_n is not necessarily a constant. Therefore, one must take care in using an equality such as Eq. (7) when relating a site fraction and a surface molar concentration, that is, to ensure that the current (correct) value of $\Gamma_n(t)$ is used. It may be necessary to add equations to calculate the current value of the total site concentration of each surface phase. Because surface site nonconservation is an issue that can alter the basic system governing equations, we require that one acknowledge its use by adding a keyword on the REACTION line (discussed later). It is up to the user's application code to ensure that the current site concentrations are correct. Subroutines that return an array of species production rates also return an array of surface phase production rates, which would all be zero if sites are conserved in each elementary reaction.

Standard-State Thermodynamic Properties

Surface Chemkin presumes that the standard-state thermodynamic properties for all species (regardless of phase) are given in terms of polynomial fits to the specific heats at constant pressure:

$$\frac{C^o_{p_k}}{R} = \sum_{m=1}^{M} a_{mk} T^{(m-1)} \tag{10}$$

For the gas-phase species the superscript o refers to the standard state of an ideal gas at 1 atm. For perfect gases that we consider, however, the heat capacities are independent of pressure; the standard-state values are the actual values.

For surface species the standard state of species k refers to the case of a chemical potential for a surface of pure species k (i.e., $Z_k \rightarrow 1$) with a fixed site density Γ^o_n. Moreover, a perfect solution (i.e., noninteracting) is assumed for the surface phase, which is independent of the system pressure. Under these assumptions the chemical potential for surface species k on surface site n may be written as

$$\mu_k(T, P, \underline{Z}) = \mu^o_k(T) + RT \ln(\Gamma_n Z_k / \Gamma^o_n).$$

The standard state assumed by Surface Chemkin for bulk-phase species is discussed in the previous section.

Other thermodynamic properties are given in terms of integrals of the specific heats. First, the standard-state enthalpy is given by

$$H^o_k = \int_0^T C^o_{p_k} dT \tag{11}$$

so that

$$\frac{H_k^o}{RT} = \sum_{m=1}^{M} \frac{a_{mk}T^{(m-1)}}{m} + \frac{a_{M+1,k}}{T} \tag{12}$$

where the constant of integration $a_{M+1,k}R$ is the standard heat of formation at 0 K. Normally, however, this constant is evaluated from knowledge of the standard heat of formation at 298 K since the polynomial representations are usually not valid down to 0 K.

The standard-state entropy is written as

$$S_k^o = \int_0^T \frac{C_{p_k}^o}{T} dT \tag{13}$$

so that

$$\frac{S_k^o}{R} = a_{1k} \ln T + \sum_{m=2}^{M} \frac{a_{mk}T^{(m-1)}}{(m-1)} + a_{M+2,k} \tag{14}$$

where the constant of integration $a_{M+2,k}R$ is evaluated from knowledge of the standard-state entropy at 298 K.

The above equations are stated for an arbitrary-order (Mth order) polynomial, but Surface Chemkin is designed to work with thermodynamic data in the form used in the NASA chemical equilibrium code.[7] In this case, seven coefficients are needed for each of two temperature ranges.* These fits take the following form:

$$\frac{C_{p_k}^o}{R} = a_{1k} + a_{2k}T + a_{3k}T^2 + a_{4k}T^3 + a_{5k}T^4 \tag{15}$$

[SKCPOR]

$$\frac{H_k^o}{RT} = a_{1k} + \frac{a_{2k}}{2}T + \frac{a_{3k}}{3}T^2 + \frac{a_{4k}}{4}T^3 + \frac{a_{5k}}{5}T^4 + \frac{a_{6k}}{T} \tag{16}$$

[SKHORT]

$$\frac{S_k^o}{R} = a_{1k} \ln T + a_{2k}T + \frac{a_{3k}}{2}T^2 + \frac{a_{4k}}{3}T^3 + \frac{a_{5k}}{4}T^4 + a_{7k} \tag{17}$$

[SKSOR]

Other thermodynamic properties are easily given in terms of C_p^o, H^o, and S^o. The internal energy U is given as

$$U_k^o = H_k^o - RT, \tag{18}$$

[SKUML]

the standard-state Gibbs free energy G^o is written as

$$G_k^o = H_k^o - TS_k^o, \tag{19}$$

[SKGML]

---

* The Surface Chemkin Interpreter can be modified for additional temperature ranges, which would then require format changes to the thermodynamic data.

and the standard-state Helmholtz free energy A^o is defined to be

$$A_k^o = U_k^o - TS_k^o. \tag{20}$$

[SKAML]

For a perfect gas, the standard-state specific heats, enthalpies, and internal energies are also the actual values. Therefore, we drop the superscript o on those quantities.

Often, specific thermodynamic properties are needed in mass units (per gram) rather than in molar units (per mole). The conversion is made by dividing the property in molar units by the molecular weight. The specific properties are thus given as

$$c_{p_k} = \frac{C_{p_k}}{W_k} \tag{21}$$

[SKCPMS]

$$h_k = \frac{H_k}{W_k} \tag{22}$$

[SKHMS]

$$s_k^o = \frac{S_k^o}{W_k} \tag{23}$$

[SKSMS]

$$u_k = \frac{U_k}{W_k} \tag{24}$$

[SKUMS]

$$g_k^o = \frac{G_k^o}{W_k} \tag{25}$$

[SKGMS]

$$a_k^o = \frac{A_k^o}{W_k} \tag{26}$$

[SKAMS]

In addition to pure species properties, it is sometimes desirable to know mean properties for a mixture. The Chemkin[1] user's manual discusses this topic for gas-phase mixtures, and Chemkin provides subroutines to return mixture-average properties. At present, however, Surface Chemkin does not provide subroutines to return mixture-averaged properties for surface- or bulk-phase species. Thus, knowing the pure-species properties, the user must compute any averaged properties required in an application.

Chemical Reaction Rate Expressions

The I reversible (or irreversible) surface reactions involve K chemical species and can be represented in the general form

$$\sum_{k=1}^{K} \nu'_{ki}\chi_k \rightleftarrows \sum_{k=1}^{K} \nu''_{ki}\chi_k \qquad (i = 1, \ldots, I) \tag{27}$$

The stoichiometric coefficients ν_{ki} are integers* and χ_k is the chemical symbol for the kth species. Normally, an elementary reaction involves only three or four species; hence the ν_{ki} matrix is quite sparse for a large set of reactions.

The production rate $\dot{s}_k$ (in moles/cm^2 of surface /sec) for each of the K species (regardless of phase) is written as a sum over the rate-of-progress variables for all reactions involving the kth species:

$$\dot{s}_k = \sum_{i=1}^{I} \nu_{ki} q_i \qquad (k = 1, \ldots, K) \tag{28}$$

[SKRAT]

where

$$\nu_{ki} = (\nu''_{ki} - \nu'_{ki}). \tag{29}$$

[SKNU]

The rate-of-progress variable q_i for the ith reaction is given by the difference of the forward rates and the reverse rates:

$$q_i = k_{f_i} \prod_{k=1}^{K} [X_k]^{\nu'_{ki}} - k_{r_i} \prod_{k=1}^{K} [X_k]^{\nu''_{ki}}. \tag{30}$$

[SKROP]

It is not a requirement that the number of sites of type n balance in a given reaction. The production rate $\dot{\Gamma}_n$ (in moles/(cm^2 of surface)/sec) for each surface phase is

$$\dot{\Gamma}_n = \sum_{i=1}^{I} \Delta\sigma(n,i) q_i, \qquad (n = N_s^f, \ldots, N_s^l) \tag{31}$$

where

$$\Delta\sigma(n,i) = \sum_{k=K_s^f}^{K_s^l} \nu_{ki}\sigma_k(n). \tag{32}$$

---

* Global reactions are sometimes stated with non-integer stoichiometric coefficients. However, because we have designed Surface Chemkin to work exclusively with elementary reaction steps, we only consider integer stoichiometric coefficients.

The term $\Delta\sigma(n,i)$ is the net change in number of surface sites of type n for surface reaction i. As discussed above, the form of the concentrations $[X_k]$ depends upon whether the species is in the gas phase, on the surface, or in the bulk. Furthermore, the units on the rate constants will depend on the reactants and products in a particular reaction. The forward rate constants k_{f_i} for the I reactions are assumed to have the following Arrhenius temperature dependence:

$$k_{f_i} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{R_c T}\right) \tag{33}$$

[SKABE, SKRAEX]

where the pre-exponential factor A_i, the temperature exponent β_i, and the activation energy E_i are specified.* These three parameters are required input to the Surface Chemkin package for each reaction.

For reversible reactions, the reverse rate constants k_{r_i} are related to the forward rate constants through the equilibrium constants as

$$k_{r_i} = \frac{k_{f_i}}{K_{c_i}} \tag{34}$$

Although K_{c_i} is given in concentration units, the equilibrium constants are more easily determined from the thermodynamic properties in pressure units, K_{p_i}; they are related by

$$K_{c_i} = K_{p_i} \left(\frac{P_{\text{atm}}}{RT}\right)^{\sum_{k=1}^{K_g} \nu_{ki}} \prod_{n=N_s^f}^{N_s^l} (\Gamma_n^o)^{\Delta\sigma(n,i)} \tag{35}$$

[SKEQ]

where P_{atm} denotes a pressure of 1 atm, and Γ_n^o is the standard-state surface site density of site type n. The sum in the exponent runs only over the gas-phase species. The equilibrium constants K_{p_i} are obtained with

$$K_{p_i} = \exp\left(\frac{\Delta S_i^o}{R} - \frac{\Delta H_i^o}{RT}\right). \tag{36}$$

The Δ refers to the change that occurs in passing completely from reactants to products in the ith reaction. More specifically,

$$\frac{\Delta S_i^o}{R} = \sum_{k=1}^{K} \nu_{ki} \frac{S_k^o}{R} \tag{37}$$

$$\frac{\Delta H_i^o}{RT} = \sum_{k=1}^{K} \nu_{ki} \frac{H_k^o}{RT}. \tag{38}$$

---

* Two gas constants, R and R_c, are used throughout this report and the Surface Chemkin code. R_c is used only in conjunction with the activation energy E_i and has compatible units. The reason for the duality is that many users would rather use different units (say calories/mole) for the activation energies even though other units (say cgs or SI) are used elsewhere.

Surface-Coverage Modification of Rate Expression

In some cases there are experimental data that the Arrhenius expression for the rate constant, Eq. (33), is modified by the coverage (concentration) of some surface species. If optional coverage parameters are specified for species k and reaction i, then the rate constant for the forward reaction is modified as

$$k_{f_i} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{R_c T}\right) \prod_{k=K_s^f}^{K_s^l} 10^{\eta_{ki}[X_k]} [X_k]^{\mu_{ki}} \exp\left(\frac{-\epsilon_{ki}[X_k]}{R_c T}\right), \tag{39}$$

where the three coverage parameters are η_{ki}, μ_{ki}, and ϵ_{ki} for species k and reaction i. The product in Eq. (39) runs over only those species that have been specified as contributing to the coverage modification. Note that the term associated with μ_{ki} now gives the provision for the rate of progress of a reaction to be proportional to a *non-integer* power of a surface species concentration. Also, using this modified expression for k_{f_i}, the net pre-exponential factor is a function of coverage

$$\log_{10} A = \log_{10} A_i + \sum_{k=K_s^f}^{K_s^l} \eta_{ki}[X_k] \tag{40}$$

and the activation energy is a function of the coverage

$$E = E_i + \sum_{k=K_s^f}^{K_s^l} \epsilon_{ki}[X_k]. \tag{41}$$

For reactions with optional coverage dependence, the rate of progress is calculated using Eq. (30), with the forward rate coefficient from Eq. (39).

Sticking Coefficients

For some simple surface reaction mechanisms we have found it convenient to specify the surface reaction rate constant in terms of a "sticking coefficient" (probability). This option is allowed for the simple case of a surface reaction in which there is only one gas-phase reactant species, although there could be any number of surface species as reactants. (It is also a requirement that the stoichiometric coefficient of the gas-phase species must be 1.) The sticking coefficient's functional form is taken to be

$$\gamma_i = a_i T^{b_i} e^{-c_i/R_c T}. \tag{42}$$

In this case a_i and b_i are unitless and c_i has units compatible with R_c.

(CAUTION: Because γ_i is defined as a probability, it must lie between 0 and 1 to make physical sense. However, Surface Chemkin does not check the value of γ_i, and an unphysical value could lead to erroneous results, for example above a certain temperature. It is up to the user to ensure that the formula for the sticking coefficient is valid in a given application.)

As an example, to specify that $SiH_2(g)$ reacts with probability γ upon each collision with the surface, one could write the reaction

$$SiH_2(g) \longrightarrow Si(b) + H_2. \tag{43}$$

A somewhat more detailed way of using the sticking-coefficient specification would be to say that $SiH_2(g)$ reacts with probability γ upon each collision with a bare surface silicon atom, Si(s):

$$SiH_2(g) + Si(s) \longrightarrow Si(s) + Si(b) + H_2. \tag{44}$$

Internally the Surface Chemkin code converts rate constants given as sticking coefficients γ to the usual mass-action kinetic rate constants k by assuming that the sticking coefficient expression refers to the case of an open surface site fraction of 1 and by using

$$k_{f_i} = \gamma_i \sqrt{\frac{RT}{2\pi W}} / \Gamma_{\text{tot}}^n, \tag{45}$$

where R is the universal gas constant, W is the molecular weight of the gas-phase species, Γ_{tot} is the total number of moles of surface sites per cm^2, and n is the sum of all of the surface reactants' stoichiometric coefficients. For example, in the reaction

$$Si_2H_2(g) + 2Si(s) \longrightarrow 2Si(s) + 2Si(b) + H_2 \tag{46}$$

the value of n is 2, because there are two surface species appearing as reactants (2Si(s)).

The rate of progress is then calculated using Eq. (30) as usual. The sticking coefficient specification is only allowed for the forward reaction. If the reaction is written as reversible, the reverse reaction rate constant would be calculated from Eqs. (45) and (34).

There is a direct relationship between the Arrhenius coefficients A_i, β_i, and E_i in the rate expression from Eq. (33) and the sticking coefficient parameters a_i, b_i, and c_i. Surface Chemkin provides subroutine SKABE to return the Arrhenius coefficients as given by the following equations:

$$A_i = a_i \sqrt{\frac{R}{2\pi W}} / \Gamma_{\text{tot}}^{n} \qquad (47) \; [\text{SKABE}]$$

$$\beta_i = b_i + \frac{1}{2} \qquad (48) \; [\text{SKABE}]$$

$$E_i = c_i. \qquad (49) \; [\text{SKABE}]$$

Manipulation of Chemical Rate Sensitivity Coefficients

We have found sensitivity analysis to be a powerful tool in helping interpret the results of computational simulations. Sensitivity analysis is used to determine quantitatively the dependence of a solution on certain parameters that appear in a model's definition. The "raw" first-order sensitivity coefficient matrices $S_{li} = \partial \Phi_l / \partial \alpha_i$ report the partial derivatives of the dependent variable vector Φ (e.g., temperature, mass fractions, surface composition) with respect to a parameter vector α_i (e.g., reaction rate constants). Since there is much mathematical literature on sensitivity analysis and various methods to compute the sensitivity coefficients from the solution, we do not discuss the computation of S_{li} here.

However, given the sensitivity matrix it is possible to manipulate it further to obtain the sensitivities of species production rates with respect to the dependent variables:

$$\frac{d\dot{s}_k}{d\alpha_i} = \frac{\partial \dot{s}_k}{\partial \alpha_i} + \sum_l \frac{\partial \dot{s}_k}{\partial [X_l]} \frac{\partial [X_l]}{\partial \Phi_l} \frac{\partial \Phi_l}{\partial \alpha_i}, \qquad (50)$$

where the components of Φ are the mass fractions, site fractions, and activities for gas-phase, surface, and bulk species, respectively. The term $\partial [X_l] / \partial \Phi_l$ converts from concentration units to the units of Φ:

$$\frac{\partial [X_l]}{\partial \Phi_l} = \begin{cases} \left(\frac{P}{RT}\right) \left\{ \frac{\overline{W}}{W_l} - \frac{Y_l \overline{W}^2}{W_l^2} \right\}, & K_g^f \le l \le K_g^l \\ \Gamma_n / \sigma_k(n), & K_s^f \le l \le K_s^l \\ 1, & K_b^f \le l \le K_b^l \end{cases} \qquad (51)$$

We have included two subroutines in the Surface Library to facilitate calculation of these terms. The first gives the partial derivative of the production rate of species k with respect to the pre-exponential constant of the Arrhenius expression for surface reaction i:

$$\frac{\partial \dot{s}_k}{\partial \alpha_i} = \nu_{ki} q_i / \alpha_i. \tag{52}$$

[SKDRDA]

(Note that subroutine SKDRDA calculates the derivative with respect the Arrhenius pre-exponential even if the reaction was originally stated in the form of a sticking coefficient. The relationship between the Arrhenius and sticking coefficient pre-exponential is given by Eq. (47).)

The production rate of species k due to reaction i is

$$\dot{s}_{ki} = \nu_{ki} q_i. \tag{53}$$

Therefore, the dependence of $\dot{s}_{ki}$ upon the concentration of some species l is

$$\begin{aligned} \frac{\partial \dot{s}_{ki}}{\partial [X_l]} = \nu_{ki} q_i^f & \left[\frac{\nu'_{li}}{[X_l]} + \left\{ \eta_{li} \ln(10) + \frac{\mu_{li}}{[X_l]} - \frac{\epsilon_{li}}{R_c T} \right\} \right] \\ - \nu_{ki} q_i^r & \left[\frac{\nu''_{li}}{[X_l]} + \left\{ \eta_{li} \ln(10) + \frac{\mu_{li}}{[X_l]} - \frac{\epsilon_{li}}{R_c T} \right\} \right]. \end{aligned} \tag{54}$$

The terms inside the curled braces will only be present if species l modifies the rate of reaction i through coverage parameters, as in Eq. (39). The partial of the production rate of species k due to all reactions with respect to the concentration of species l is then

$$\frac{\partial \dot{s}_k}{\partial [X_l]} = \sum_{i=1}^{I} \frac{\partial \dot{s}_{ki}}{\partial [X_l]} \tag{55}$$

[SKDRDC]

These terms can all be combined to calculate the desired $\partial \dot{s}_k / \partial \alpha_i$ in the user's application code.

Flux-Matching Conditions at a Gas-Surface Interface

Heterogeneous reactions at a gas-surface interface affect the mass and energy balance at the interface, and thus have an important influence on the boundary conditions in a chemically reacting flow simulation. The convective and diffusive mass fluxes of gas-phase species at the surface are balanced by the production (or depletion) rates of gas-phase species by surface reactions. This relationship is

$$\mathbf{n} \cdot \left[\rho Y_k \left(\mathbf{V_k} + \mathbf{u}\right)\right] = \dot{s}_k W_k, \qquad (k = 1, \ldots, K_g) \tag{56}$$

where $\mathbf{n}$ is the unit outward-pointing normal vector to the surface, and the gas-phase diffusion velocities are related to the species and temperature gradients by

$$\mathbf{V_k} = \frac{1}{X_k \overline{W}} \sum_{j \neq k}^{K_g} W_j D_{kj} \nabla X_j - \frac{D_k^T}{\rho Y_k} \frac{\nabla T}{T} \qquad (k = 1, \ldots, K_g) \tag{57}$$

Here the X_k are the gas-phase mole fractions, the Y_k are the gas-phase mass fractions, $\overline{W}$ is the mean molecular weight, D_{kj} is the ordinary multicomponent diffusion coefficient matrix, and the D_k^T are the thermal diffusion coefficients. (Both types of diffusion coefficients can be evaluated by calls to the Transport Package[2,3]). In nonreacting fluid mechanics the fluid velocity normal to a solid wall is zero. However, if there are chemical reactions at the wall, then the velocity can be nonzero. This so-called Stefan flow velocity occurs when there is a net mass flux between the surface and the gas. The induced Stefan velocity is given by

$$\mathbf{n} \cdot \mathbf{u} = \frac{1}{\rho} \sum_{k=1}^{K_g} \dot{s}_k W_k. \tag{58}$$

The expression for the Stefan velocity is easily obtained from the interfacial mass balance (Eq. 56) by summing over all K_g species and noting that the mass fractions must sum to one, i.e.,

$$\sum_{k=1}^{K_g} Y_k = 1,$$

and that the sum of the diffusion fluxes must sum to zero, i.e.,

$$\sum_{k=1}^{K_g} Y_k V_k = 0.$$

Exothermicity (or endothermicity) of surface reactions contribute to the energy balance at an interface. Diffusive and convective fluxes in the gas phase are balanced by thermal radiative and chemical heat release at the surface. This balance is stated as

$$\mathbf{n}\cdot\lambda\nabla T\Big|_{\text{gas}} - \sum_{k=1}^{K_g}\mathbf{n}\cdot\rho Y_k\left(\mathbf{V_k}+\mathbf{u}\right)h_k = \sigma\epsilon\left(T^4-T_0^4\right)+\sum_{k=K_s^f}^{K_b^l}\dot{s}_k W_k h_k + \mathbf{n}\cdot\lambda_{\text{bulk}}\nabla T\Big|_{\text{bulk}}. \tag{59}$$

The summation on the right-hand side runs over all surface and bulk species. It is interesting to note that by substituting Eq. (56) into the flux term on the left-hand side, the energy balance can be written in a more compact form as

$$\mathbf{n}\cdot\lambda\nabla T\Big|_{\text{gas}} = \sigma\epsilon\left(T^4-T_0^4\right)+\sum_{k=1}^{K}\dot{s}_k W_k h_k + \mathbf{n}\cdot\lambda_{\text{bulk}}\nabla T\Big|_{\text{bulk}}. \tag{60}$$

Now the reaction-rate summation on the left-hand side runs over all species, including the gas-phase species.

The Surface Chemkin package allows the user to specify mass densities ρ_k for the bulk species. A possible use for the densities would be to convert surface reaction rate of production of a bulk species (in moles/cm^2/sec) into a growth rate G (in cm/sec). The needed relationship is

$$G = \sum_{k=K_b^f}^{K_b^l}\frac{\dot{s}_k W_k}{\rho_k} \tag{61}$$

IV. THE MECHANICS OF USING SURFACE CHEMKIN

Surface Chemkin is one component of a large body of software designed to facilitate the the computational modeling of chemical kinetics in flowing systems. An application program (say a chemical vapor deposition analysis code) can draw on any of three major software packages:

- Chemkin-II, which handles gas-phase equation-of-state, thermodynamic properties, and chemical kinetics;
- the Transport Package, which handles gas-phase molecular transport properties; and
- Surface Chemkin, which handles surface thermodynamics and chemical kinetics.

Each package consists of a symbolic preprocessor called an Interpreter, a database of either thermodynamic or transport properties, and a library of subroutines that can be called from the application code.

The software is highly structured and modular, which provides great flexibility in applying it to a wide variety of problems. However, this modularity also compels the user to manipulate a number of programs and files. The flow of information from the first input to the Chemkin Interpreter to the inclusion of a library subroutine in an application program is shown in Fig. 5.

We presume that all problems involving surface chemistry also involve one or more gas-phase species above the surface. Therefore, the first step in any problem involving surface chemistry is to run the Chemkin Interpreter, which reads the user's description of the gas-phase reaction mechanism. The Chemkin Interpreter also draws on a Thermodynamic Database containing polynomial fits to individual species specific heats, enthalpies, and entropies. In addition to printed output, the Chemkin Interpreter creates a Linking File containing all possible information regarding the particular gas-phase reaction mechanism. The Linking File is read by an initialization subroutine in the Gas-Phase Subroutine Library that makes the information available to all the other subroutines in the library.

The next step is to execute the Surface Chemkin Interpreter, which reads the user's symbolic description of the surface-reaction mechanism. Required thermodymanic data can come from the same Thermodynamic Database used by Chemkin or from a separate Thermodynamic Database compiled for surface species. Both Interpreters provide the capability to add to or override the data in the database by user input in the reaction description. The Surface Chemkin Interpreter extracts all needed information about gas-phase species from the Chemkin Linking File. (Thus the Chemkin Interpreter must be executed before the Surface Chemkin Interpreter.) Like the Chemkin Interpreter, the Surface Chemkin Interpreter also provides a printed output and a Linking File. Again, the Surface Linking File is read by an initialization subroutine in the Surface

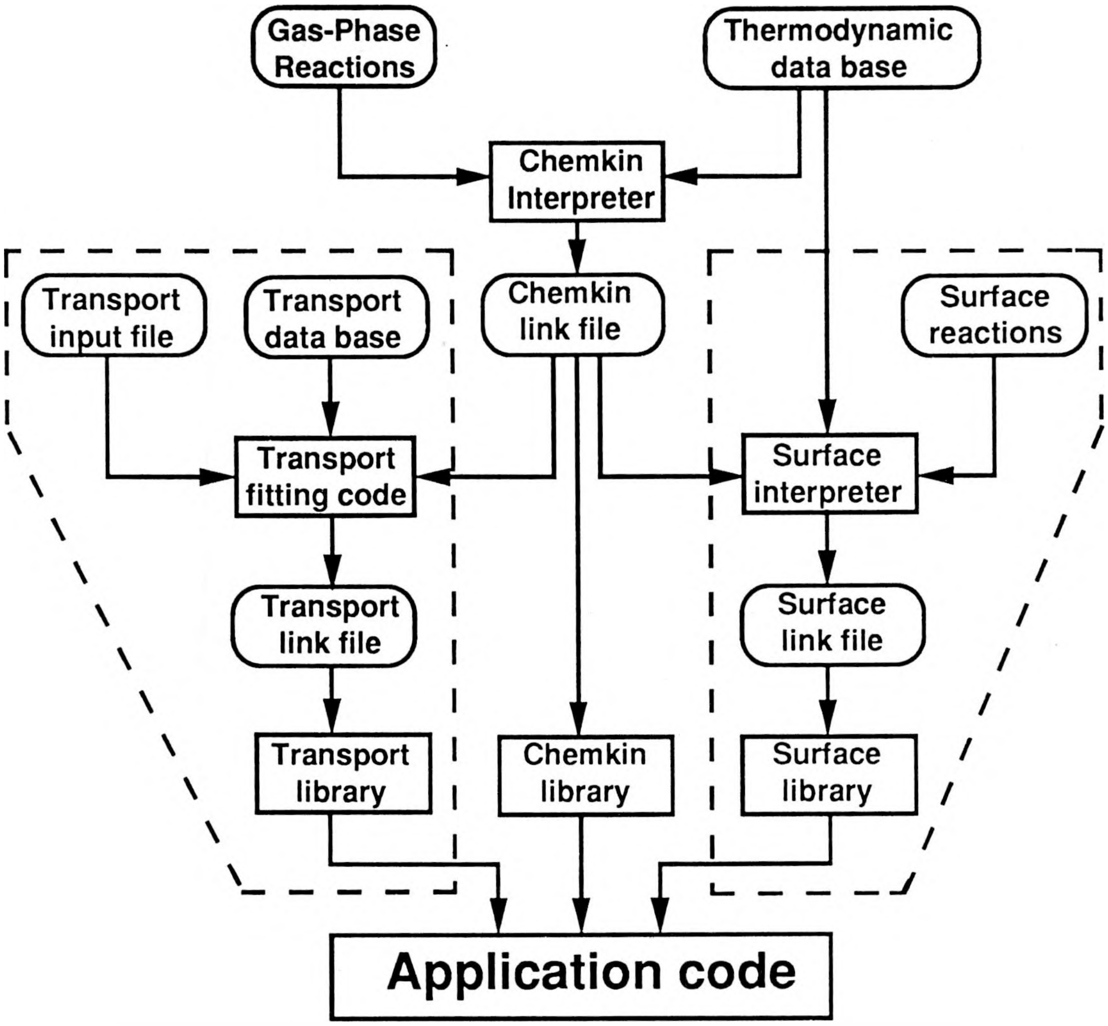


Figure 5. Relationships and Flow of Information between the Chemkin-II, Transport, and Surface Chemkin Packages, and a User's Application Program.

Subroutine Library that makes the surface-reaction mechanism information available to all other subroutines in the Library.

A third software package, which handles gas-phase molecular transport, may or may not be needed in a particular application. If it is used, the Transport Property Fitting Code reads the Chemkin Linking File and identifies all the gas-phase species that are present in the gas-phase reaction mechanism. Then, drawing on a database of molecular parameters, it computes polynomial fits to the temperature-dependent pure-species viscosities, thermal conductivities, and binary diffusion coefficients. As with the other packages it provides a Linking File that is read by an initialization subroutine in the Transport Property Subroutine Library. Subroutines from this library may be called by the application code to return transport properties for individual species or for multicomponent gas mixtures.

It is clear from the foregoing description that using these software packages requires the interaction of many programs and the manipulation of many input and output files. Therefore, while the modular structure of the software provides a powerful capability to formulate a wide variety of applications, it also requires that users be reasonably familiar with their computers' operating systems. The functional equivalent of the sample VAX/VMS command procedure shown in Fig. 6 is required on any computer system.

It may also be noted from the flow chart of Fig. 5 that the three software packages do not *solve* any problems—they simply make subroutines available to aid *formulation* of a problem. This structure provides maximum flexibility because the software does not need to be concerned with details of the limitless range of problems that a user may wish to pose and solve. At the same time, the burden is on the user to define the equations that describe his particular problem and to write an application code to solve these equations.

The schematic diagram in Fig. 7 shows how an applications code might interact with the low-level chemical kinetics and transport software packages. The boxes with the light borders indicate those blocks of software that are relegated to subroutine calls to one of the low-level packages, and the boxes with the heavy borders indicate those blocks of software that the user must write for a particular application. We show the problem-independent mathematical software as a box with a light border to indicate that many problems can solved with readily available, high-quality mathematical software. Certainly this is true for problems that can be formulated as systems of nonlinear algebraic equations or ordinary-differential-equation initial-value or boundary-value problems. However, for more complex problems involving systems of partial differential equations the user may have to write his own mathematical software.

VAX/VMS Commands			Meaning
$assign	GASMECH.DAT	FOR015	Assign the user's gas-phase reaction mechanism to Fortran unit 15. This is the input file for the Chemkin Interpreter.
$assign	CKINTERP.OUT	FOR016	Assign the output file for printed output from the Chemkin Interpreter.
$assign	CKTHERMO.DAT	FOR017	Assign the Thermodynamic Database to Fortran unit 17.
$assign	CKLINK.BIN	FOR025	Assign the Chemkin Linking file to Fortran unit 25.
$run	CKINTERP.EXE		Execute the Chemkin Interpreter.
$assign	SURFMECH.DAT	FOR015	Assign the user's surface reaction mechanism to Fortran unit 15. This is the input file for the Surface Chemkin Interpreter.
$assign	SKINTERP.OUT	FOR016	Assign the output file for printed output from the Surface Chemkin Interpreter.
$assign	SKTHERMO.DAT	FOR017	Assign the Surface Thermodynamic Database to Fortran unit 17. This could be the same database as used above for Chemkin.
$assign	SKLINK.BIN	FOR026	Assign the Surface Chemkin Linking file to Fortran unit 26.
$run	SKINTERP.EXE		Execute the Surface Chemkin Interpreter.
$assign	TRANSPORT.OUT	FOR06	Assign the output file for printed output from the Transport Package Fitting Code.
$assign	TRANSPORT.DAT	FOR031	Assign the Transport Database to Fortran unit 31.
$assign	TPLINK.BIN	FOR035	Assign the Transport Linking file to Fortran unit 35.
$run	TRANFIT.EXE		Execute the Transport Property Fitting code.
$for	SAMPLE.FOR		Compile the user's Fortran program.
$assign	SAMPLE.INP	FOR005	Assign a file containing any input required by the user's program to Fortran unit 5.
$assign	SAMPLE.OUT	FOR006	Assign a file to accept any printed output from the user's program to Fortran unit 6.
$link	SAMPLE.OBJ, CKLIB/LIB, SKLIB/LIB,TRANLIB/LIB		Link the user's program with the Chemkin Gas-Phase Subroutine Library, the Surface Library, and the Transport Library.
$run	SAMPLE		Execute the user's program.

Figure 6. Sample VAX/VMS Command Procedure, showing the Steps Required to run an Application Code using the Chemkin-II, Transport, and Surface Chemkin Packages.

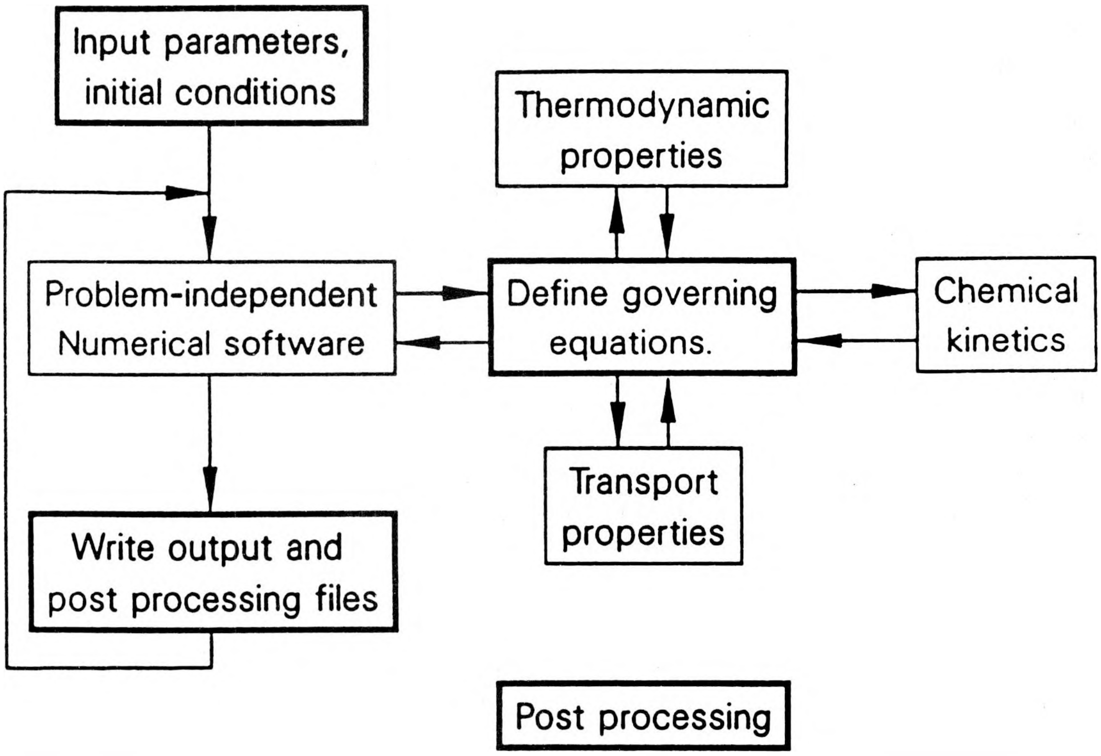


Figure 7. Schematic Representation of an "Ideal" Applications Program. The boxes with heavy borders represent functions that are problem specific, while the boxes with light borders represent functions that can be relegated to problem-independent software.

V. USING THE SURFACE CHEMKIN INTERPRETER

The Surface Chemkin Interpreter is used to read a symbolic description of an elementary surface-reaction mechanism (from unit number LIN) and numerical information about a gas-phase mechanism (from unit number LINKCK), and create a Linking File (unit number LINKSK) of pertinent information about the surface mechanism and the species it involves. The information in the Surface Chemkin Linking File is subsequently accessed by various subroutines in the Surface Chemkin Library to provide information on thermodynamic properties and chemical production rates.

The Surface Interpreter input includes information on surface species, bulk species, thermodynamic data, and the reaction mechanism. Input information on file LIN is given in 80-column card-image format. Site data are read first, followed by bulk data; optional thermodynamic data may follow, and optional reaction data are specified last. The thermodynamic data for the species may come from input (unit number LIN) and/or from a Thermodynamic Database (unit number LTHRM). The syntax for the four types of input is described below.

With the exception of the thermodynamic data, all input is format free. For the thermodynamic data, we have chosen to use the same format as in the NASA Chemical Equilibrium code of Gordon and McBride[7] and Chemkin-II.[1]

Site Data

Surface-phase species exist on sites, and a site and its species must be identified on one or more lines of site data. The first line in a set of site data must start with the word SITE; an optional name may be associated with a site if it immediately follows SITE and is delimited by slashes (/). If no name for the site is supplied, the interpreter substitutes the default name SITEn, where n is the number of the site (e.g., SITE2 for the second site type listed). Following SITE and/or the site name, the word SDEN and a slash-delimited density (the standard-state site density for this site, in moles/cm^2) for the site is required. The species that can reside on the site type are declared by a list of species symbols (names) on the same line or on additional lines. The name of a site species must not duplicate the name of a gas-phase species or a bulk species, and must be unique among the species names listed *for that site.* However, the same site species name may appear in the list for *another* site. Thus, a user can refer to a surface species uniquely by specifying both the name of the site and the species name. An optional slash-delimited site occupancy number may follow a species name, i.e., the number of individual sites that this species occupies. (For example, a large chemical species might cover two or more sites.) The default site occupancy for a surface species is 1. The sets of SITE data cards can continue for as many site types as are needed.

Any set of up to sixteen upper- or lower-case characters can be used as a site name or species symbol.* In addition, each species must be composed of elements that have been identified in the Chemkin Interpreter and thus contained in the Gas-phase Linking File (unit number LINKCK). One of the primary purposes of the site data is to identify the order in which Fortran arrays of site species information are referenced in the Surface Subroutine Library.

Any line starting with or portion of a line following an exclamation mark (!) is considered a comment and will be ignored. Blank lines are also ignored. Figure 8 shows sample site data. The rules for site data are summarized in Table I.

```
SITE / PLANE /     SDEN/1.04E-9/                                  ! PLANAR SITE

   ASH(V)                                             ! FIRST SPECIES ON PLANE SITE
   ASH2(V)     ASH3(V)     H(S)     CH3(V)     AS(V)     AS2(V)/2/
   V                                                         ! EMPTY PLANAR SITE
     END                                               ! an END statement is optional

SITE/LEDGE/      SDEN/1.66E-10/                                     ! LEDGE SITE

   GACH(L)                                               ! FIRST SPECIES ON LEDGE
   H(S)                                                 !THIS IS A DUPLICATE NAME
   DMG(L)/2/                                  ! THIS SPECIES OCCUPIES 2 LEDGE SITES
   L                                                          ! EMPTY LEDGE SITE
SITE     GA(S)                                         !SITE WITH ONLY ONE SPECIES
                                           !SITE NAME AND DENSITY NOT INCLUDED
```

Figure 8. Sample Site Data.

---

* Species symbols may not begin with a number, a plus sign (+), or an equality sign (=), have imbedded blanks, or include a slash (/). An ionic species may end with any number of plus or minus signs; an imbedded plus sign must be enclosed in parentheses.

TABLE I. SUMMARY OF THE RULES FOR SITE DATA

---

1. Site data must start with a line containing the word SITE, followed by an optional slash-delimited name (i.e., SITE/*name*/).
2. The standard-state site density is required as a slash-delimited number (in moles/cm^2) following the word SITE and/or the site name, and preceded by the word SDEN.
3. The site density is followed by one or more site species name declarations. Declaring a site with no site species is an error.
4. Site and species names are composed of up to sixteen upper- or lower-case character symbols. The names cannot begin with the characters +, =, or a number; an ionic species name may end with one or more +'s or −'s; an embedded plus sign must be enclosed in parenthesis (+). Names cannot include a slash (/).
5. All species names within a given site should be unique; duplicated species names will be ignored and a warning issued. A species name may not duplicate a name of a gas-phase or bulk species, but may duplicate the name of a species on a different site.
6. A site name must not duplicate the name of any other phase (gas, surface site, or bulk phase).
7. Each surface species that subsequently appears in a surface reaction must have been declared in this section.
8. A site species name may appear anywhere on the line.
9. A site species may have a slash-delimited site occupancy (the number of sites that this species occupies on the surface) following the species name.
10. A species name declaration that begins on one line may not continue to the next line (i.e., do not break species names into two lines.)
11. There may be more than one set of SITE data.
12. All characters on a line following an exclamation mark are considered comments.
13. SITE data are not required.

Bulk Data

A set of bulk data may consist of one or more condensed-phase species. The first line in a set of bulk data must start with the word BULK and may be followed by an optional slash-delimited name for the bulk phase. If no name is supplied for bulk phase *n*, then the name BULK*n* is supplied by the interpreter. Species are declared by a list of unique species symbols on the same line or on additional lines. An optional slash-delimited density (in g/cm^3) may follow a species name. If no density is supplied, the unphysical value of -1.0 is stored as a flag. The rules for bulk species symbols (names) are essentially the same as those for site species. Figure 9 shows sample bulk data. The rules for bulk data are summarized in Table II.

Thermodynamic Data

Any chemical species that appears in a problem must have thermodynamic data associated with it. The data may be extracted from a database (unit number LTHRM) and/or read from input (unit number LIN). If all the thermodynamic data are extracted from the database, then no thermodynamic data are required from the input file. However, if the user wishes to override information in the database or provide data on species not in the database, then Surface Interpreter thermodynamic input is needed. In any case the format for the information is the same.

The format (see Table III) is a minor modification of that used by Gordon and McBride[7] for the Thermodynamic Database in the NASA Chemical Equilibrium code. Our modification allows for a different midpoint temperature for the fits to the properties of each chemical species. We also allow a species to be composed of a maximum of five elements, not four. However, the formatting is such that the Chemkin Interpreter can use the NASA database directly without any modification.

As Table III indicates, the pertinent information includes the species name, the elemental composition of the species, and the temperature ranges over which the polynomial fits to thermodynamic data are valid. The fits to C_p^o/R, H^o/RT, and S^o/R consist of seven coefficients for each of two temperature ranges [see Eqs. (15 - 17)].* Further information about the fitting procedure and data for many species are in a report on the Chemkin Thermodynamic Database.[6]

---

* Additional temperature ranges and their fit coefficients may be accommodated by minor changes to the Interpreter and the Thermodynamic Database.

```
BULK / GA_RICH /    GA2AS(1)/3.0/    GA3AS(1)/3.0/     END
                                                         ! an END statement is optional
BULK / GA_RICH/
       GA2AS(1)/3.0/
       GA3AS(1)/3.0/
       GA2AS(1)/2.0/            !THIS NAME IS A DUPLICATE AND WILL BE IGNORED
BULK  AS(B)                                  !BULK PHASE WITH NO NAME SUPPLIED
                              !ONLY ONE BULK SPECIES AND NO DENSITY SUPPLIED
END
```

Figure 9. Sample Bulk Data.

TABLE II. SUMMARY OF THE RULES FOR BULK DATA

1. Bulk data must start with a line containing the word BULK, and may be followed by a slash-delimited name for the bulk phase (i.e., BULK/*name*/).
2. The BULK declaration and/or bulk name must be followed by one or more bulk species declarations. Declaring a bulk phase with no bulk species is an error.
3. Bulk and bulk species names are composed of up to sixteen upper- or lower-case character symbols. The names cannot begin with the +, =, or a number; an ionic species name may end with one or more +'s or −'s; an embedded plus sign must be enclosed in parenthesis (+). Names cannot include a slash (/).
4. Bulk species names must be unique within a given bulk phase; duplicated species names will be ignored and a warning issued. A species name may not duplicate a name used for a gas-phase or a site species, but may duplicate the name of a species in a different bulk phase.
5. A bulk phase name must not duplicate the name of any other phase (gas, surface site, or bulk phase).
6. Each bulk species that subsequently appears in a surface reaction must have been declared in this section.
7. A bulk species declaration may start anywhere on the line.
8. A bulk species name may be followed by an optional slash-delimited mass density (in g/cm^3).
9. A bulk species declaration that begins on one line may not continue to the next line (i.e., do not break species names into two lines.)
10. There may be more than one set of BULK data.
11. All characters on a line following an exclamation mark are considered comments and are ignored.
12. BULK data are not required.

When thermodynamic data input is required, it must immediately follow species data (SITE and BULK data). The first thermodynamic data line must start with the word THERMO (or THER). If all the thermodynamic data are input directly to the Interpreter, then the first line of this section must read THERMO ALL and the code will not expect a Thermodynamic Database from unit number LTHRM; for this option the next line must be line 2 of Table III. For either option, the subsequent thermodynamic data lines must be in the format of lines 3 – 6 of Table III. (For the THERMO option the default midpoint temperature is taken from the line 2 information already in the Thermodynamic Database.) As many species as needed can be included as THERMO input. Thermodynamic data for any species that are not declared will be ignored.

Figure 10 shows some examples of thermodynamic property input. In these examples for OH, OH+, and OH–, it is seen from columns 25 – 34 that the elemental composition of each molecule is one O atom and one H atom. Columns 35 – 39 indicate that two of the species, OH+ and OH–, are ionic since they contain –1 and +1 electrons (E), respectively. The G in column 45 indicates that all three species are gaseous. The 1000.00 in columns 66 – 73 for OH+ indicates the common temperature (in K) between the high- and low-temperature fits. If columns 66 – 73 are left blank, as they are for OH+ and OH–, then the common temperature is that given in columns 21 – 30 of line 2 in Table III, which in this example is in the Thermodynamic Database.

```
THERMO
OH                      O   1H   1          G  0300.00   5000.00  1000.00      1
 0.02882730E+02 0.10139743E-02-0.02276877E-05 0.02174683E-09-0.05126305E-14    2
 0.03886888E+05 0.05595712E+02 0.03637266E+02 0.01850910E-02-0.16761646E-05    3
 0.02387202E-07-0.08431442E-11 0.03606781E+05 0.13588605E+01                   4
OH+                     O   1H   1E  -1     G  0300.00   5000.00               1
 0.02719058E+02 0.15085714E-02-0.05029369E-05 0.08261951E-09-0.04947452E-13    2
 0.15763414E+06 0.06234536E+02 0.03326978E+02 0.13457859E-02-0.03777167E-04    3
 0.04687749E-07-0.01780982E-10 0.15740294E+06 0.02744042E+02                   4
OH-               1212860   1H   1E   1     G  0300.00   5000.00               1
 0.02846204E+02 0.10418347E-02-0.02416850E-05 0.02483215E-09-0.07775605E-14    2
-0.01807280E+06 0.04422712E+02 0.03390037E+02 0.07922381E-02-0.01943429E-04    3
 0.02001769E-07-0.05702087E-11-0.01830493E+06 0.12498923E+01                   4
END
```

Figure 10. Examples of Thermodynamic Data Input.

TABLE III. SUMMARY OF THE RULES FOR THERMO DATA

Line Number	Contents	Format	Column
1	THERMO (or THERMO ALL[a])	Free	Any
2[b]	Temperature ranges for 2 sets of coefficients: lowest T, common T, and highest T	3F10.0	1 to 30
3	Species name (must start in Column 1)	16A1	1 to16
	Date (not used in the code)	6A1	19 to 24
	Atomic symbols and formula	4(2A1,I3)	25 to 44
	Phase of species (S, L, or G for solid, liquid, or gas, respectively)	A1	45
	Low temperature	E10.0	46 to 55
	High temperature	E10.0	56 to 65
	Common temperature (if needed) (blank for default)	E8.0	66 to 73
	Atomic symbol and formula (if needed) (blank for default)	2A1,I3	74 to 78
	The integer 1	I1	80
4	Coefficients $a_1 - a_5$ in Eqs. (15 - 17), for upper temperature interval	5(E15.0)	1 to 75
	The integer 2	I1	80
5	Coefficients a_6, a_7 for upper temperature interval, and a_1, a_2, and a_3 for lower	5(E15.0)	1 to 75
	The integer 3	I1	80
6	Coefficients a_4, a_5, a_6, a_7 for lower temperature interval	4(E15.0)	1 to 60
	The integer 4	I1	80
...	Repeat lines 3 – 6 for each species.		
last	END (Optional, end of thermodynamic data.)	Free	Any

[a] Use only when all the thermodynamic data are to be taken from Surface Interpreter input.

[b] Include line 2 in the Interpreter input *only* with THERMO ALL. Line 2 does appear in the database itself.

The following cases summarize the possibilities for specifying thermodynamic data.

Case 1: All thermodynamic data from database only

1. Assign the database as file LTHRM (default Fortran unit 17).
2. No THERMO data required as input.

Case 2: Thermodynamic data from database and input

1. Assign the database as file LTHRM (default Fortran unit 17).
2. Include the following lines:
   THERMO
   Data in Table III format (lines 3 - 6 repeated) for species not in the database or to override species in database.
   END (optional)

Case 3: All thermodynamic data from input

1. Do not need to attach a database.
2. Include the following lines:
   THERMO ALL
   Line 2 of Table III format.
   Data in Table III format (lines 3 - 6 repeated) for at least all species named in the species data.
   END (optional)

Surface-Reaction Mechanism Description

The surface-reaction mechanism may consist of any number of chemical reactions involving the solid species named in the site and bulk data, as well as the gas-phase species declared in the gas-phase Chemkin Interpreter. A reaction may be reversible or irreversible.

Reaction data must start with the word REACTIONS (or REAC). On the same line the user may specify units of the Arrhenius rate coefficients to follow by including the words CAL/MOLE, KCAL/MOLE, JOULES/MOLE, KJOULES/MOLE, or KELVINS to indicate the units for E_i, and/or MOLES or MOLECULES to indicate the units for A_i [see Eq. (33)] . If MOLECULES is specified, then the units for A_i are cm-molecules-sec-K. If units are not specified, A_i and E_i are assumed to be in cm-mole-sec-K and cal/mole, respectively. The lines following the REACTION line contain reaction descriptions together with their Arrhenius rate coefficients. The reaction description is composed of reaction data and perhaps optional auxiliary information data.

The Interpreter normally considers any reaction that does not conserve the number of surface sites to be in error. If the user wishes to include such reactions (which may be perfectly valid), the word NONCON must appear on the REACTION line.

Reaction Data

Each reaction "line" is divided into two fields, where a "line" may take up two or more physical lines if it is more than 80 characters long. A reaction data line is continued on the next line using the special character &; any information following the & symbol on the same line is ignored. The first field in the reaction line contains the symbolic description of the reaction, while the second contains the Arrhenius rate coefficients. Both fields are format free, and blank spaces are ignored. All characters on a line following an exclamation mark (!) are considered comments and are ignored. Blank lines are also ignored.

The reaction description, given in the first field, must be composed of the species symbols, coefficients, and delimiters as summarized below.

Symbols: Each species in a reaction is described with a unique sequence of characters as they appear in the species data and the thermodynamic data. However, if a species name is not unique (because it is duplicated in another phase), the name must be modified by appending its slash-delimited phase name, i.e. as *name/phase/*.

Coefficients: A species symbol may be preceded by a positive integer coefficient. This coefficient is interpreted as the number of moles of the particular species present as either a reactant or a product; e.g., 2OH is equivalent to OH + OH. Noninteger coefficients are not allowed.

Delimiters:

+ A plus sign is the delimiter between all reactant species names and between all product species names.

= An equality sign is the delimiter between the last reactant and the first product in a reversible reaction.

<=> An equality sign enclosed by angle brackets can also used as the delimiter between the last reactant and the first product in a reversible reaction.

=> An equality sign with an angle bracket on the right is the delimiter between the last reactant and first product in an irreversible reaction.

The second field of the reaction line is used to define the Arrhenius rate coefficients A_i, β_i, and E_i in that order, as given by Eq. (33). At least one blank space must separate the first number and the last species name in the reaction. The three numbers must be separated by at least one blank space, be stated in either integer, floating point, or E format (e.g., 123 or 123.0 or 12.3E1), and have units associated with them. Unless modified by the REACTION line, the default units for A_i are cgs (cm, sec, K, mole), the exact units depending on the order of the reaction. The factor β_i is dimensionless. The default units for the activation energies are cal/mole.

The second field of the reaction line may optionally be used to specify the coefficients a_i, b_i, and c_i of Eq. (42) for a sticking coefficient. In order for the second field to apply to sticking coefficient parameters, the next line of input must contain the auxiliary information word STICK.

Examples of some reaction data are shown in Fig. 11. Table IV summarizes the reaction data rules.

```
REACTIONS                KCAL/MOLE                NONCON

  ASH3 + AS(P) <=> AS3(P) + AS(D)             4.0E11  0  25               ! Ref. 21
  ! ASH3 + AS(P) <=> AS3(P) + AS(D)    4.0E11  0  0    ! same as previous reaction,
  ASH <=> AS(D) + H(S)   1.0   0   0
  GA(CH3)3(L) + GA2AS(1) <=> AS + GA(CH3)(L) + 2GAME & continued on next line
        1.0E13 0 4000.
  STICK
```

Figure 11. Examples of Reaction Data.

Auxiliary Information Data

The format of an auxiliary information line is a character-string keyword followed by a slash-delimited (/) field containing an appropriate number of parameters (either integer, floating point, E format, or character).

For a reversible reaction, auxiliary information data may follow the reaction to specify Arrhenius parameters for the reverse-rate expression. Here, the three Arrhenius parameters (A_i, β_i, and E_i) for the reverse rate must follow the keyword REV. Using this option overrides the reverse rates that would normally be computed through the equilibrium constant, Eq. (30).

It sometimes happens that two or more reactions can involve the same set of reactants and products, but proceed through distinctly different processes. In these cases it may be appropriate to state a reaction mechanism that has two or more reactions that are the same, but have different rate parameters. However, duplicate reactions are normally considered errors by the Interpreter; if the user requires duplication (e.g., the same reactants and products with different Arrhenius parameters), an auxiliary information statement containing the keyword DUP (with no parameters) must follow the reaction line of each duplicate reaction (including the first occurrence of the reaction that is duplicated). For example, to specify different rate expressions for each of three identical reactions, there must be three occurrences of the keyword DUP, one following each of the reactions.

If the three coefficients given in the second field of the reaction line are to be interpreted as the parameters a_i, b_i, and c_i of Eq. (42)) for a sticking coefficient, then the keyword STICK (with no parameters) must follow the reaction line as auxiliary information.

To modify the expression for the forward rate constant by optional coverage parameters (see Eq. (39)) one uses the auxiliary information keyword COV followed by (slashed delimited) surface species name and the three parameters η_{ki}, μ_{ki}, and ϵ_{ki}. More than one set of COV data can appear for a given reaction, and these would be applied multiplicatively as in Eq. (39).

Any number of auxiliary information lines may follow a reaction line, in any order, and any number of keywords may appear on an auxiliary information line; however, a keyword and its parameter(s) must appear on the same line.

Examples of auxiliary information are shown in Fig. 12. The above rules are summarized in Table V.

TABLE IV. SUMMARY OF THE RULES FOR REACTION DATA

1. The first reaction line must start with the word REACTIONS (or REAC), and may be followed by units definition(s) or the word NONCON.
2. Valid unit declarations are KELVINS, MOLES, CAL/MOLE, KCAL/MOLE, JOULES/MOLE, KJOULES/MOLE, and MOLECULES.
3. The word NONCON is required on the first reaction line if any of the reactions do not conserve the number of sites of a given type.
4. The reaction description can begin anywhere on the line. All blank spaces, except those separating the Arrhenius coefficients, are ignored.
5. Each reaction description must have =, <=>, or => between the last reactant and the first product.
6. Each species in a reaction is described with a unique sequence of characters as they appear in the species data and the thermodynamic data. However, if a species name is not unique (because it is duplicated in another phase), the name must be modified by appending its slash-delimited phase name, i.e. as *name/phase/*.
7. A reaction description may be contained on more than one line. If a line contains the symbol &, all information following the & symbol will be ignored and the next line will be considered a continuation of the first.
8. Three Arrhenius coefficients must appear in order (A_i, β_i, and E_i) on each Reaction line, separated from each other and from the reaction description by at least one blank space; no blanks are allowed within a number.
9. There cannot be more than six reactants or six products in a reaction.
10. To specify a sticking coefficient rather than a rate constant the three numbers after the reaction description have the meaning a_i, b_i, and c_i (see Eq. (42)) and the auxiliary information word STICK must appear on the next line of input. To use this option the reaction must have only one gas-phase species as a reactant and its stoichiometric coefficient must be 1; the reaction may be reversible.
11. All characters on a line following an exclamation mark are comments.

```
    REACTIONS                              CAL/MOLE
    GA(CH3)3(L) + GA2AS(1) => AS + GA(CH3)(L) + 2GAME 1.0E13   0   3.0E4
                                                                 ! Example
                                          REV/5.0E12   0   1.5E4/

 ! The following two reactions are acceptable duplicates:
    GA(CH3)3(L)+GA3AS(1) = AS+GA(L) + 3GAME                1.0E13   0    40000
       DUPLICATE
    GA(CH3)3(L)+GA3AS(1) = AS+GA(L) + 3GAME                1.0E12   0    23000
       DUPLICATE
    AS + GA(S) <=> AS(S) + GA(D)                                0.5   0    2000.
       STICK   REV/1.0E13 0 40000/  COV/AS/LEDGE/-1.0E-15  -0.5  3E14/
```

Figure 12. Examples of Auxiliary Information Definitions.

TABLE V.

SUMMARY OF THE RULES FOR AUXILIARY INFORMATION DATA.

1. Auxiliary information lines may follow a reversible reaction to specify the reverse rate parameters explicitly; auxiliary information *must* follow any reactions that are duplicated.
2. Keyword declarations may appear anywhere on the line, in any order.
3. Any number of keywords may appear on a line and more than one line may be used, but a keyword and its parameter(s) must appear on the same line.
4. Keyword declarations that appear on the same line must be separated by at least one blank space.
5. Any blank spaces between a keyword and the first slash are ignored and any blanks between the slashes and parameter(s) are also ignored. However, no blank spaces are allowed within a keyword or a parameter.
6. All characters on a line following an exclamation mark are comments.
7. The keyword REV followed by three slash-delimited Arrhenius coefficients may be used to specify the reverse rate parameters.
8. The keyword DUPLICATE (or DUP) must follow every occurrence of a duplicated reaction.
9. The keyword STICK indicates that the three coefficients on the reaction line are to be interpreted as the parameters a_i, b_i, and c_i in Eq. (39).
10. The keyword COV is used to modify the forward rate constant by the expression in Eq. (39). The word COV is followed by a surface species name and the three coverage parameters η_{ki}, μ_{ki}, and ϵ_{ki}. The four entries after the word COV are slash-delimited.

Problems Having No Reactions

In some problems only information about the surface and bulk species is needed (e.g., chemical equilibrium computations). For these cases it is not necessary to include reaction data. The Interpreter will create the LINKSK file, but it will not contain any reaction information. Therefore, no subroutines in the Surface-Phase Subroutine Library that deal with chemical reactions (e.g., chemical production rates) should be used. (Although doing so would not generate an error, the production rates of all species would be returned as zero.)

Unit Conversion

The Surface Chemkin Interpreter provides the option of specifying units of the Arrhenius parameters. The parameters are always stored internally in the same way, i.e., activation energies in Kelvins, dimensionless temperature exponents, and pre-exponential factors consistent with moles-cm-sec-K. The program converts the user input activation energies in an obvious way. However, it is worthwhile to state explicitly the conversions for the Arrhenius pre-exponential constant and the surface coverage coefficients. In converting from "molecules" to "moles"

$$A_{\text{moles}} = A_{\text{molecules}}(6.023 \times 10^{23})^{n-1} \tag{62}$$

and

$$n = \sum_{k=1}^{K_s^I} \nu_k^I \tag{63}$$

Note that the ν_k^I are the stoichiometric coefficients for the gas-phase and surface reactant species (not product species or bulk-phase species).

Coverage parameters [Eq. (39)] are converted as

$$\eta_{k,\text{moles}} = \eta_{k,\text{molecules}} \times 6.023 \times 10^{23} \tag{64}$$

$$\epsilon_{k,\text{moles}} = \epsilon_{k,\text{molecules}} \times 6.023 \times 10^{23}. \tag{65}$$

The parameter μ_k itself is not affected by this unit change, but the pre-exponential constant A must be converted as

$$A_{\text{moles}} = A_{\text{molecules}} \times (6.023 \times 10^{23})^{n-1} \times \prod_{k=1}^{K_{\text{cov}}} (6.023 \times 10^{23})^{\mu_k}, \tag{66}$$

where K_{cov} is the number of species that have coverage parameters specified for this reaction.

Error Checks

The Interpreter checks each input line for proper syntax and writes self-explanatory diagnostic messages on logical file LOUT if errors are encountered. If an error occurs, the Interpreter continues to read and diagnose the input, but an error flag is written to the Linking file and Surface Library subroutine SKINIT will not initialize the work arrays. Therefore, the input must be error free before any of the Surface Chemkin subroutines can be called.

Possibilities for an error condition are as follows:

Site and Bulk Species Data

A species symbol duplicated within a phase is not considered a fatal error, but is eliminated from consideration and a diagnostic message is printed.

No density is found for a declared site.

No thermodynamic data are found for a declared species.

There are more species than the Interpreter is dimensioned for (500).*

A site or bulk phase name duplicates another phase name (gas, surface site, or bulk phase name).

A phase or species name contains an illegal character.

Thermodynamic Data

Thermodynamic Data are format sensitive and therefore provide possibilities for error if not formatted exactly as described by Table III.

An element in the thermodynamic data for a declared species is not included in the Chemkin element data.

With the THERMO ALL option, line 2 (Table III) is not found.

---

* This limit may be changed by a simple modification of a parameter statement in the Surface Chemkin Interpreter.

Reaction Data

A delimiter =>, <=>, or = between the reactants and the products is not found.

Three Arrhenius parameters are not found.

Reactants and/or products species names are not properly delineated by a plus sign (+).

A species as a reactant or product is not declared in the species data.

A reaction does not balance.

The number of sites in a reaction does not balance and the word NONCON was not included on the first REACTIONS line.

The charge of the reaction does not balance.

A reaction is a duplicate not declared by the auxiliary data keyword DUP.

There are more reactions than the Interpreter is dimensioned for (1000).

There are more than six reactants or six products in a reaction.

For a sticking-coefficient reaction, there is more that one gas-phase species, or the stoichiometric coefficient for the gas-phase species is not 1.

Auxiliary Data

An unknown or misspelled keyword occurs.

Parameters for a keyword are not enclosed in slashes.

There are the wrong number of parameters for a keyword.

REV is declared for an irreversible reaction.

Pre-exponential factor for a sticking coefficient is negative.

VI. DATA STRUCTURES IN SURFACE CHEMKIN

We have to consider species that may be in the gas phase, on various surface sites, or in various bulk mixtures, and the number of species, surface sites, and bulk mixtures change from problem to problem. The user may want to refer to a species by an integer species index (such as in a Fortran DO loop) or to refer to a species by a character string (such as in an input/output routine). Furthermore, in some circumstances it may be convenient to refer to a species in a long list of all species, and in other cases it may be convenient to refer only to those species in a particular phase. Thus it is important to have a data structure that is flexible enough to capture the required generality, yet sufficiently "friendly" to make it easy to use. In this chapter we use specific examples to illustrate the data structures and how to use them in an applications program.

Consider a system involving gallium-arsenide deposition where the gas-phase reaction mechanism is shown in Fig. 13 and the surface mechanism is shown in Fig. 14. (CAUTION: The mechanisms have been contrived to illustrate a variety of points about the data structures and should not be taken as a source of data for gallium-arsenide deposition processes.) The gas-phase mechanism includes fourteen species, even though only eleven of them appear directly in the gas-phase reaction mechanism. Arsine (ASH3 in our notation) is typically a feed gas in a gallium-arsenide deposition process. Monatomic arsenic (AS) and the arsenic dimer (AS2) have been included because the surface reaction mechanism includes desorption reactions, which introduce them into the gas. Therefore, because all gas-phase species are handled by Chemkin, these extra three species must be included as input to the Chemkin Interpreter. Of course, the gas-phase and surface-reaction mechanisms should not be considered entirely independently. If the surface mechanism generates a gas-phase species, then it is probably reasonable to ask if there are plausible gas-phase reactions that can consume it. For example, perhaps a gas-phase recombination reaction could be included to form an arsenic dimer.

The surface mechanism in the example has two surface sites, called "PLANE" and "LEDGE." There are five bulk "mixtures" although three of them are not actually mixtures, but pure compounds. The first three bulk mixtures have been assigned the names "GA-RICH," "GA-POOR," and "GA-STOICH," whereas the remaining two have not been given any specific names.

The data structure is illustrated in Fig. 15, where each column represents a "phase." The gas phase is *always* the first phase; the surface site phases are stored next, followed by the bulk phases. The species are stored sequentially beginning with the first gas-phase species and ending with the last bulk species. The gas-phase species are listed in the same order as they were presented to the Chemkin Interpreter. The species in the

```
ELEMENTS GA AS H C END
SPECIES
   ASH3 AS2 GAME3 GAME2 GAME GA CH3 CH4 H C2H6 C2H5 C2H4 H2
   AS
END

REACTIONS
   2CH3(+M) <=> C2H6(+M)           0.903E+17 -1.180    654.000
       LOW/ 3.18E41 -7.03 2762./
       TROE/ .6041 6927. 132./
       H2/2/
   CH3 + H(+M) <=> CH4(+M)         0.600E+17 -1.000      0.000
      LOW/8.0E26 -3. 0.0/
      H2/2/
   CH4 + H <=> CH3 + H2            0.220E+05  3.000   8750.000 ! CLARK$DOVE
   C2H6 + CH3 <=> C2H5 + CH4       0.550E+00  4.000   8300.000
   C2H6 + H <=> C2H5 + H2          0.540E+03  3.500   5210.000 ! CLARK$DOVE
   H + C2H4(+M) <=> C2H5(+M)       0.221E+14  0.000   2066.000 ! MICHAEL,GLAN$TROE
      LOW/6.37E27  -2.76  -54./
      H2/2/
   C2H5 + H <=> CH3 + CH3          1.0E+14    0.000      0.000
   H + H + M <=> H2 + M            0.100E+19 -1.000      0.000 ! D-L
   H + H + H2 <=> H2 + H2          0.920E+17 -0.600      0.000
   GAME3 <=> GAME2 + CH3           0.347E+16  0.000  59500.000 ! JACKO AND PRICE
   GAME2 <=> GAME + CH3            0.871E+08  0.000  35410.000 ! JACKO AND PRICE
END
```

Figure 13. Sample Gas-Phase Reaction Mechanism for a Gallium-Arsenide Sample Problem.

surface and bulk phases are in the same order as they appear in the Surface Interpreter input.

There are several indices and arrays of indices that are quite useful in referencing particular species and phases. They are returned from two subroutines, SKINDX and SKPKK, described in the following:

SUBROUTINE SKINDX (NELEM, KKGAS, KKSUR, KKBULK, KKTOT, NNPHAS, NNSURF, NFSURF, NLSURF, NNBULK, NFBULK, NLBULK, IISUR)

SUBROUTINE SKPKK(ISKWRK, KKPHAS, KFIRST, KLAST)

The input to SKPKK is the integer work space ISKWRK. On output all the indices will have values that pertain to the particular reaction mechanisms. For the example we are considering here, the indices have the following values:

```
                                             ! IPHASE = 1 IS THE GAS, NPHASE(1) = KK
SITE/PLANE/        SDEN/1.04E-9/             ! IPHASE = 2, NPHASE(2) = 3
   ASH3(P)
   ASH(P)
   AS(P)
END

SITE/LEDGE/        SDEN/1.04E-13/             ! IPHASE = 3, NPHASE(3) = 4
   GA(CH3)3(L)
   GA(CH3)2(L)
   GA(CH3)(L)
   GA(L)

BULK/GA_RICH/                                ! IPHASE = 4, NPHASE(4) = 2
    GA2AS(1)/3.00/                           ! THE DENSITY OF THE SPECIES IS
    GA3AS(1)/3.00/                           ! OPTIONAL

BULK/GA_POOR/  GAAS2(2)/3.15/ GAAS3(2)/3.15/ GAAS4(2)/3.15/ END
                                             ! IPHASE = 5, NPHASE(5) = 3

BULK/GA_STOICH/                              ! IPHASE = 6, NPHASE(6) = 1
    GAAS(3)/5.24/                            ! A BULK PHASE CAN HAVE JUST ONE SPECIES
END

BULK GA(B)/2.5581/                          ! IPHASE = 7, NPHASE(7) = 1
BULK AS(B)/2.7489/                          ! IPHASE = 8, NPHASE(8) = 1

REACTIONS
   ASH3  + AS(P) <=> ASH3(P)     + AS(B)               4.0E11  0    0.
   GAME3 + GA(L)  => GA(CH3)3(L) + GA(B)               1.0E11  0    0.
   GA(CH3)3(L) + GA2AS(1) <=> AS + GA(CH3)(L) + 2GAME  1.0E13  0    40000.
                                                   REV /1.0E11  0    25000./
   GA(CH3)3(L) + GA3AS(1) <=> AS + GA(L) + 3GAME       1.0E13  0    40000.
            DUPLICATE
   GA(CH3)3(L) + GA3AS(1) <=> AS + GA(L) + 3GAME       1.0E12  0.5  30000.
            DUPLICATE
   GA2AS(1) <=> GA(B) + GAAS(3)                        1.0E13  0    40000.
   GAAS2(2) <=> AS(B) + GAAS(3)                        1.0E13  0    40000.
   GAAS3(2) <=> AS2 + GAAS(3)                          1.0E13  0    40000.
   GAAS4(2) <=> AS(B) + AS2 + GAAS(3)                  1.0E13  0    40000.
END
```

Figure 14. Sample Surface-Reaction Mechanism for a Gallium-Arsenide Sample Problem.

NELEM, Number of elements—Here, NELEM=4. As presented on the first line of input to the Chemkin interpreter, the four elements are GA, AS, H, C.

KKGAS, Number of gas-phase species—Here KKGAS=14.

KKSUR, Number of surface species—There are seven surface species in this problem and they exist on two sites. Thus, KKSUR=7.

KKBULK, Total number of bulk species—There are eight bulk species that are arranged in five mixtures. Thus KKBULK=8.

KKTOT, Total number of species—KKTOT=KKGAS+KKSUR+KKBULK, so in this example KKTOT=29.

NNPHAS, Number of phases; gas + sites + bulk—Here there are eight phases, NNPHAS=8. The first is the gas phase, the second two are surface phases, and the final five are bulk phases.

NNSURF - Number of surface phases—Here there are two surface phases, NNSURF=2.

NFSURF, Pointer to the first surface phase—The first surface phase is the second phase overall, i.e., NFSURF=2. (Since the gas phase is *always* the first phase, the first surface phase will always be the second phase overall. However, there can be problems in which there are no surface phases, only bulk phases. In that case NFSURF=0.)

NLSURF, Pointer to the last surface phase—Here the last surface phase is the third phase overall. Thus NLSURF=3. (If a problem should have no surface phases then NLSURF=0).

NNBULK - Number of bulk phases—Here there are five bulk phases, i.e., NNBULK=5.

NFBULK, Pointer to the first bulk phase—Here the first bulk phase is the fourth phase overall, i.e., NFBULK=4. (In a problem that has no bulk phases NFBULK=0).

NLBULK, Pointer to the last bulk phase—In this problem, NLBULK=8. (In a problem that has no bulk phases, NLBULK=0).

IISUR, Number of surface reactions—Here there are 9 surface reactions, so IISUR=9.

GAS-PHASE	SURFACE SITES		BULK SOLIDS				
	"PLANE"	"LEDGE"	"GA-RICH"	"GA-POOR"	"GA-STOICH"		
1. ASH3	15. ASH3(P)	18. GA(CH3)3(L)	22. GA2AS(1)	24. GAAS2(2)	27. GAAS(3)	28. GA(B)	29. AS(B)
2. AS2	16. ASH(P)	19. GA(CH3)2(L)	23. GA3AS(1)	25. GAAS3(2)			
3. GAME3	17. AS(P)	20. GA(CH3)(L)		26. GAAS4(2)			
4. GAME2		21. GA(L)					
5. GAME							
6. GA							
7. CH3							
8. CH4							
9. H							
10. C2H6							
11. C2H5							
12. C2H4							
13. H2							
14. AS							

Figure 15. Schematic diagram of the Phase and Species Data Structure for a Gallium-Arsenide Sample Problem.

KKPHAS, Array of total number of species in the phases—In this example KKPHAS is an array of length 8, the total number of phases. The values of each element in the array indicate the number of species in the corresponding phase. Here, KKPHAS(1,...,8) = (14, 3, 4, 2, 3, 1, 1, 1), which shows that the gas phase has 14 species, the first surface phase has 3, the second surface phase has 4, and so on.

KFIRST, Array of starting species numbers of the phases—In this example KFIRST is an array of length 8, the total number of phases. The values of each element in the array indicate the species number of the first species in the corresponding phase. Here, KFIRST(1,...,8) = (1, 15, 18, 22, 24, 27, 28, 29), which shows that in a complete list of species, the first gas-phase species is number 1, the first species in the first surface phase is number 15, the first species in the second surface phase is number 18, and so on.

KLAST, Ending species numbers for the phases—In this example KLAST is an array of length 8, the total number of phases. The values of each element in the array indicate the species number of the last species in the corresponding phase. Here, KLAST(1,...,8) = (14, 17, 21, 23, 26, 27, 28, 29), which shows that in a complete list of species, the last gas-phase species is number 14, the last species in the first surface phase is number 17, the last species in the second surface phase is number 21, and so on.

Thermodynamic properties, molecular weights, and reaction rates, for example, have values for all species regardless of the phase in which they exist. In these cases the values are stored in long arrays that correspond to the data structure shown in Fig. 15. Other sets of variables or parameters do not exist for all species. For example, we may need the mass densities for the bulk species, but mass density makes no sense for surface species. Nevertheless, we maintain the same data structure. For example, the mass densities for all species can be retrieved by calling

SUBROUTINE SKDEN(P, T, ACT, SDEN, ISKWRK, RSKWRK, DEN).

Here, the input is the gas-phase pressure, the temperature, and the species activities (mole fractions for the gas phase, site fractions for the surface species, and activities for the bulk species) and the site densities of each site phase. The first segment of output array DEN contains the gas-phase species densities (in grams of species k per cm^3). The array elements corresponding to the surface species have no physical meaning and are set to the species densities (in moles/cm^2). The final elements of the DEN array contain the mass densities of the bulk species (in grams of species k per cm^3). If the user does not specify a mass density as input to the Surface Interpreter, then these are given a value of −1.

All the species can be identified by a character-string name, and the user can also optionally specify character-string names for the phases. The species names are those that were given as input to the Interpreters, and the application code can retreive the names by the calling following subroutine:

SUBROUTINE SKSYMS(CSKWRK, LOUT, KNAME, KERR)

The input is the character working space CSKWRK and LOUT, a logical file number on which to write diagnostic messages. The output is an array of character names KNAME and a logical error flag KERR. In the example we are considering here, the species names are shown in Fig. 15. As concrete illustrations,

KNAME(1) = "ASH3",
KNAME(KLAST(NLSURF)) = "GA(L)", or
KNAME(KFIRST(7)) = "GA(B)".

The phase names are optional and can be specified in the Surface Interpreter input. The names can be retrieved from an application code by the following subroutine call:

SUBROUTINE SKSYMP(CSKWRK, LOUT, PNAME, KERR)

Here the output consists of character-string names for the phases. The first phase is always the gas and its name is PNAM(1)="GAS". In the present example, PNAME(2) = "PLANE", PNAME(NFBULK) = "GA-RICH", and so on. If the user does not specify phase names, default names are supplied. Thus in this example, PNAME(8) = "BULK5".

VII. QUICK REFERENCE TO THE SURFACE SUBROUTINE LIBRARY

This chapter is arranged by topical area to provide a quick reference to each of the Surface Library Subroutines. In addition to the subroutine call list itself, the purpose of the subroutine is briefly described. The page number given for each subroutine refers a detailed description of the subroutine call in Chapter VIII.

Mnemonics

There are some rules of thumb for explaining the subroutine naming conventions. All subroutines names begin with the letters SK so that Surface Chemkin Subroutines are easily recognized and so that they are likely different from any user subroutine names. The four remaining letters identify the purpose of the subroutine.

Thermodynamic properties are referred to by CP (specific heat), H (enthalpy), S (entropy), U (internal energy), G (Gibbs free energy), and A (Helmholtz free energy). The thermodynamic property subroutines may be called to return properties in mass units, denoted by MS or S as the last letter(s), or in molar units, denoted by ML or L as the last letter(s).

The mnemonics for the variable names in the subroutine call lists are roughly the same as for the subroutine names. However, because six letters can be used (only four are available in the subroutine names because SK occupies two), the mnemonics can be more explicit.

1. INITIALIZATION

```
SUBROUTINE SKINDX (NELEM, KKGAS, KKSUR, KKBULK, KKTOT, NNPHAS,
                   NNSURF, NFSURF, NLSURF, NNBULK, NFBULK,
                   NLBULK, IISUR)
```
Returns a group of indices defining the size of the surface reaction mechanism.

```
SUBROUTINE SKINIT (LSIWK, LSRWK, LSCWK, LINSK, LOUT,
                   ISKWRK, RSKWRK, CSKWRK)
```
Reads the surface linking file and creates the internal work arrays ISKWRK, RSKWRK, and CSKWRK. SKINIT must be called before any other Surface Chemkin subroutine is called. The work arrays must then be made available as input to the other Surface Chemkin subroutines.

```
SUBROUTINE SKLEN  (LINSK, LOUT, LENI, LENR, LENC)
```
Reads the first record of the linking file to return the lengths required for the integer, real, and character work arrays.

2. INFORMATION ABOUT ELEMENTS

```
SUBROUTINE SKSYME (CSKWRK, LOUT, ENAM, KERR)
```
Returns a character array of element names.

3. INFORMATION ABOUT SPECIES

```
SUBROUTINE SKATCZ (P, T, ACT, SDEN, ISKWRK, RSKWRK, CZ)
```
Returns the concentrations of the species, given the pressure, temperature and activities.

```
SUBROUTINE SKCHRG (ISKWRK, RSKWRK, KCHARG)
```
Returns an array containing electronic charges of the species.

```
SUBROUTINE SKCOV  (ISKWRK, KOCC)
```
Returns an array of site occupancy numbers for the species.

```
SUBROUTINE SKDEN  (P, T, ACT, SDEN, ISKWRK, RSKWRK, DEN)
```
Returns a real array of species densities.

```
SUBROUTINE SKNCF (NELDIM, ISKWRK, NEL)
```
Returns the elemental composition of the species.

```
SUBROUTINE SKSYMS (CSKWRK, LOUT, KNAM, KERR)
```
Returns a character array of species names.

```
SUBROUTINE SKWT   (RSKWRK, WT)
```
Returns the molecular weights of the species.

4. INFORMATION ABOUT PHASES

SUBROUTINE SKPKK (ISKWRK, KKPHAS, KFIRST, KLAST)
Returns arrays of species pointers for the phases.

SUBROUTINE SKSDEN (RSKWRK, SDEN0)
Returns a real array of standard-state phase densities as given on input to the interpreter.

SUBROUTINE SKSYMP (CSKWRK, LOUT, PNAM, KERR)
Returns a character array of phase names.

5. INFORMATION ABOUT SURFACE REACTIONS

SUBROUTINE SKABE (ISKWRK, RSKWRK, RA, RB, RE)
Returns the Arrhenius coefficients of the surface reactions. If a reaction was specified in a sticking-coefficient form, then this subroutine converts the sticking coefficients to the equivalent Arrhenius coefficients.

SUBROUTINE SKICOV (IR, NDIM, ISKWRK, RSKWRK, NCOVI, KCOVI, CPARI)
Returns the coverage species index numbers and their coverage parameters for reaction IR.

SUBROUTINE SKISTK (IR, ISKWRK, ISTFL)
Returns an integer flag to indicate whether reaction IR uses sticking coefficients.

SUBROUTINE SKNCON (ISKWRK, NCON)
Returns the total number of surface reactions which do not conserve sites for each phase.

SUBROUTINE SKNU (IDIM, ISKWRK, KSTOIC, NSTOIC)
Returns the stoichiometric coefficients of the species and the net change in phases for all of the surface reactions in a mechanism.

SUBROUTINE SKRAEX (IR, RSKWRK, RA)
Returns the Pre-exponential rate constant (or sticking coefficient) of the IRth reaction, or changes its value, depending on the sign of IR.

SUBROUTINE SKSYMR (IR, LOUT, ISKWRK, RSKWRK, CSKWRK, LT, RNAM, KERR)
Returns the character string representation of reaction IR.

6. GAS CONSTANTS AND UNITS

```
SUBROUTINE SKRP   (ISKWRK, RSKWRK, RU, RUC, PATM)
   Returns universal gas constants and the pressure of one standard
   atmosphere.
```

7. THERMODYNAMIC PROPERTIES (NONDIMENSIONAL)

```
SUBROUTINE SKATHM (MAXTP, NDIM1, NDIM2, ISKWRK, RSKWRK, NT, TMP,
                   A)
   Returns the polynomial coefficients of the fits for
   thermodynamic properties of all of the species.

SUBROUTINE SKCPOR (T, ISKWRK, RSKWRK, CPOR)
   Returns an array of the nondimensional specific heats at constant
   pressure.

SUBROUTINE SKHORT (T, ISKWRK, RSKWRK, HORT)
   Returns an array of the nondimensional enthalpies.

SUBROUTINE SKSOR  (T, ISKWRK, RSKWRK, SOR)
   Returns an array of the nondimensional entropies.
```

8. THERMODYNAMIC PROPERTIES (MASS UNITS)

```
SUBROUTINE SKAMS  (T, ISKWRK, RSKWRK, AMS)
   Returns an array of the standard state Helmholtz free energies
   in mass units.

SUBROUTINE SKCPMS (T, ISKWRK, RSKWRK, CPMS)
   Returns an array of the specific heats at constant pressure
   in mass units.

SUBROUTINE SKGMS  (T, ISKWRK, RSKWRK, GMS)
   Returns an array of the standard state Gibbs free energies
   in mass units.

SUBROUTINE SKHMS  (T, ISKWRK, RSKWRK, HMS)
   Returns an array of the enthalpies in mass units.

SUBROUTINE SKSMS  (T, ISKWRK, RSKWRK, SMS)
   Returns an array of the standard state entropies in mass units.

SUBROUTINE SKUMS  (T, ISKWRK, RSKWRK, UMS)
   Returns an array of the internal energies in mass units.
```

9. THERMODYNAMIC PROPERTIES (MOLAR UNITS)

SUBROUTINE SKAML (T, ISKWRK, RSKWRK, AML)
Returns an array of the standard state Helmholtz free energies in molar units.

SUBROUTINE SKCPML (T, ISKWRK, RSKWRK, CPML)
Returns an array of the specific heats at constant pressure in molar units.

SUBROUTINE SKGML (T, ISKWRK, RSKWRK, GML)
Returns an array of the standard state Gibbs free energies in molar units.

SUBROUTINE SKHML (T, ISKWRK, RSKWRK, HML)
Returns an array of the enthalpies in molar units.

SUBROUTINE SKSML (T, ISKWRK, RSKWRK, SML)
Returns an array of the standard state entropies in molar units.

SUBROUTINE SKUML (T, ISKWRK, RSKWRK, UML)
Returns an array of the internal energies in molar units.

10. CHEMICAL PRODUCTION RATES

SUBROUTINE SKCONT (KSPEC, ROP, ISKWRK, RSKWRK, CIK)
Returns the contributions of each of the surface reactions to the molar production rate of species KSPEC.

SUBROUTINE SKDRDA (IR, P, T, ACT, SDEN, ISKWRK, RSKWRK, DKDAI)
Returns the partial of the rates of production for each of the species with respect to the pre-exponential constant of surface reaction IR.

SUBROUTINE SKDRDC (KSPEC, P, T, ACT, SDEN, ISKWRK, RSKWRK, DKDC)
Returns the partial derivative of the production rates for each of the species with respect to the concentration of species KSPEC.

SUBROUTINE SKRAT (P, T, ACT, SDEN, ISKWRK, RSKWRK, SDOT, SITDOT)
Returns production rates for the species and sites.

SUBROUTINE SKRATI (IR, ROP, ISKWRK, RSKWRK, SDOTI, SITDTI)
Returns rates of production for each of the species by surface reaction IR.

11. EQUILIBRIUM CONSTANTS AND RATE OF PROGRESS VARIABLES.

SUBROUTINE SKEQ (P, T, ACT, SDEN, ISKWRK, RSKWRK, EQKC)

Returns the equilibrium constants for the surface reactions given pressure, temperature, species activities, and the site densities.

SUBROUTINE SKROP (P, T, ACT, SDEN, ISKWRK, RSKWRK, ROP)

Returns rates of progress for the surface reactions.

SUBROUTINE SKRROP (ISKWRK, RSKWRK, IISUR, SMH, MAXSPR, RU, PATM, KKGAS, KKSUR, P, T, CZ, WT, NREAC, NRPP, NU, NUNK, NUSUMK, NSPAR, PAR, RPAR, NREV, IREV, NCOV, ICOV, KCOV, NDIM, CPAR, NSTK, ISTK, SDTOT, RKF, RKR, EQKC)

Returns forward and reverse rates of progress and equilibrium constants for the surface reactions.
It is not normally called by the user application code.

12. UTILITIES

SUBROUTINE SKCOMP (ISTR, IRAY, NN, IND, NT)

Given a character string vector IRAY of vector length NN, this subroutine will search for the first occurrence of the string ISTR. The position of ISTR in IRAY is returned in the integer argument NT. If the string ISTR is not found in IRAY, IND and NT are assigned a zero value.

Consider the following example,
IRAY = {"BOOK","BLUE","BEAR","BOOK"}
NN=4.
If ISTR="BLUE" then IND=2 and NT=1;
if ISTR="RED" then IND=0 and NT=0; and
if ISTR="BOOK",then IND=1 and NT=2.

SUBROUTINE SKPCMP (ISTR, IRAY, NN, SETS, NSETS, ISET, IND, NT)

This subroutine can do everything that the subroutine SKCOMP can do, and additionally, has the capabilities of separating the elements of IRAY into categories and then search IRAY by element and category. The categories that each element of IRAY will be assigned to are specified by the input character string vector SETS of vector length NSETS. Elements of each category in IRAY must be grouped congruously. The number of elements in each category within IRAY is specified by the input integer vector ISET. To search for the existence of an element within a category ISTR may additionally be composed of two substrings, ISTR="ELEMENT_NAME/CATEGORY_NAME/", where CATEGORY_NAME is one of the categories specified in SETS. In this case, IND will return the first position in IRAY where ELEMENT_NAME occurred within the category CATEGORY_NAME. NT will return the total number of times ELEMENT_NAME occurred within the category

CATEGORY_NAME. If ELEMENT_NAME is not found within the specified category, IND and NT are returned with a value of zero. If no category is specified within ISTR, IND and NT return with the same values as they would from subroutine SKCOMP.

Consider the following example,
IRAY = ("RED", "BLUE", "JADE", "RUBY", "TOPAZ", "JADE")
NN = 6
SETS = ("COLORS", "STONES"),
NSETS = 2
ISET = (4, 2).

This assumes that the elements of IRAY were grouped into two sets, consisting of 4 and 2 elements, respectively, and the following names
"COLORS" = ("RED", "BLUE", "JADE", "RUBY"), and
"STONES" = ("TOPAZ", "JADE").
If ISTR="BLUE" then IND=2 and NT=1;
if ISTR="PINK" then IND=0 and NT=0; and
if ISTR="JADE",then IND=3 and NT=2.
If ISTR="BLUE/COLORS/" then IND=2 and NT=1;
if ISTR="BLUE/STONES/" then IND=0 and NT=0;
if ISTR="JADE/GEMS/" then IND=0 and NT=0; and
if ISTR="JADE/STONES/",then IND=6 and NT=1.

SUBROUTINE SKPNT (LSAVE, LOUT, NPOINT, VERS, PREC, LENI, LENR, LENC, KERR)

Reads from a binary file information about a Surface Chemkin linking file, pointers for the Surface Chemkin Library, and returns lengths of work arrays.

SUBROUTINE SKSAVE (LINC, LOUT, LSAVE, ISKWRK, RSKWRK, CSKWRK)

Writes to a binary file information about a Surface Chemkin linking file, pointers for the Surface Chemkin Library, and Surface Chemkin work arrays.

SUBROUTINE SKSNUM (LINE, NEXP, LOUT, KNAM, KKTOT, PNAM, NNPHAS, KKPHAS, KNUM, NT, NVAL, RVAL, KERR)

This subroutine is used to read a format-free input line of combined alphanumerical data. It can be used to parse an input character string, LINE, which may be composed of several blank-delimited substrings. This subroutine assumes that the first substring in LINE is the name of a species in the Surface Chemkin mechanism. If the species name is not unique within the Surface Chemkin mechanism, the phase of the species should be input immediately after the species name, delimited by slashes. Upon return from the subroutine, KNUM returns the index position of the species within the Surface Chemkin linking file. If the species name is not unique, KNUM returns the first position and NT returns the number of the times the species occurs within the linking file. If the species name is not found, or there is a syntax error, on return, KNUM=0, NT=0, and KERR=.TRUE.

The substrings in LINE following the first are expected to represent numbers. They are converted into floating point values and stored in the output vector, RVAL(*). Upon input, NEXP is equal to the number of values expected to be found. If NEXP numbers are not found, KERR will be set to .TRUE. on return from the subroutine.

Example input:

LINE = GA(S)/BULK1/ 1.2
NEXP = 1, the number of values expected
LOUT = 6, a logical unit number on which to write diagnostic messages
KNAM(*) = Array of character species names
KKTOT = Total number of species
PNAM(*) = Array of character phase names
NNPHAS = Total number of phases
KKPHAS(*)= Index array of the number of species in the phases

Output:

KNUM = The index number of the species which has the name "GA(S)" and resides in phase "BULK1"
NT = 1, if there is only one species GA(S) in phase BULK1
NVAL = 1, the number of values found in LINE following the species name
RVAL(1) = 1.200E+00, the substring converted to a real number
KERR = .FALSE.

VIII. ALPHABETICAL LISTING OF THE SURFACE SUBROUTINE LIBRARY WITH DETAILED DESCRIPTIONS OF THE CALL LISTS

Each subroutine in the Surface Subroutine Library is described in this chapter, together with a detailed description of the variables in the call lists. For all arrays, information is given on the required dimensioning in the calling program. For all variables having units, the cgs units are stated.

```
SKABE      SKABE      SKABE      SKABE      SKABE      SKABE      SKABE
           ***********************************************************
                      *********************************
                                  ******

SUBROUTINE SKABE  (ISKWRK, RSKWRK, RA, RB, RE)
   Returns the Arrhenius coefficients of the surface reactions.
   If a reaction was specified in a sticking-coefficient form,
   then this subroutine converts the sticking coefficients to the
   equivalent Arrhenius coefficients.

INPUT
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real work space.
                 Data type - real array

OUTPUT
   RA     - Pre-exponential constants for the reactions.
                 cgs units - mole-cm-sec-K
                 Data type - real array
                 Dimension RA(*) at least IISUR, the total
                 number of surface reactions.
   RB     - Temperature dependence exponents for the reactions.
                 cgs units - none
                 Data type - real array
                 Dimension RB(*) at least IISUR, the total
                 number of surface reactions.
   RE     - Activation energies for the reactions.
                 cgs units - Kelvins
                 Data type - real array
                 Dimension RE(*) at least IISUR, the total
                 number of surface reactions.
```

SKAML SKAML SKAML SKAML SKAML SKAML SKAML

****

****

******

SUBROUTINE SKAML (T, ISKWRK, RSKWRK, AML)

Returns an array of the standard state Helmholtz free energies in molar units.

INPUT

T - Temperature.
cgs units - K
Data type - real scalar

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

AML - Standard state Helmholtz free energies in molar units for the species.
cgs units - ergs/mole
Data type - real array
Dimension AML(*) at least KKTOT, the total number of species.

SKAMS SKAMS SKAMS SKAMS SKAMS SKAMS SKAMS

****

****

******

SUBROUTINE SKAMS (T, ISKWRK, RSKWRK, AMS)

Returns an array of the standard state Helmholtz free energies in mass units.

INPUT

T - Temperature.
cgs units - K
Data type - real scalar

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

AMS - Standard state Helmholtz free energies in mass units for the species.
cgs units - ergs/gm
Data type - real array
Dimension AMS(*) at least KKTOT, the total number of species.

```
SKATCZ    SKATCZ    SKATCZ    SKATCZ    SKATCZ    SKATCZ    SKATCZ
          **********************************************
                    **************************
                              ******

SUBROUTINE SKATCZ (P, T, ACT, SDEN, ISKWRK, RSKWRK, CZ)
   Returns the concentrations of the species, given the pressure,
   temperature and activities.

INPUT
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                 cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                 cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
            a bulk phase should be 1.
                 cgs units - none
                 Data type - real array
                 Dimension ACT(*) at least KKTOT, the total number
                 of species.
   SDEN   - Site densities for the surface site types.  This
            vector may have an entry for each phase, including the
            gas phase, but the subroutine only uses entries for the
            surface site phases, NFSURF .LE. N .LE. NLSURF.
                 cgs units - moles/cm**2
                 Data type - real array
                 Dimension SDEN(*) at least NPHASE, the total
                 number of phases.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   CZ     - Matrix of the concentrations of the gas-phase and
            surface species in the problem, and the activities
            of the bulk species.  The first KKGAS entries of CZ
            are the gas-phase molar concentrations (moles/cm**3).
            The next KKSURF entries are the surface species molar
            concentrations (moles/cm**2).  The final KKBULK entries
            are the activities of the bulk species.
                 Data type - real array
                 Dimension CZ(*) at least KKTOT, the total
                 number of gas-phase + surface + bulk species.
```

SKATHM SKATHM SKATHM SKATHM SKATHM SKATHM SKATHM

****
****
******

SUBROUTINE SKATHM (MAXTP, NDIM1, NDIM2, ISKWRK, RSKWRK, NT, TMP, A)

Returns the polynomial coefficients of the fits for thermodynamic properties of all of the species.

INPUT

MAXTP - First dimension of an array of temperatures used in the thermodynamic fits for the species.
Data type - integer scalar

NDIM1 - First dimension of an array of thermodynamic fit coefficients; NDIM1 must be at least NCP2, the total number of coefficients for one temperature range.
Data type - integer scalar

NDIM2 - Second dimension of the array of thermodynamic fit coefficients; NDIM2 must be at least (MAXTP-1), the number of temperature ranges.
Data type - integer scalar

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

Where NT(K) is the number of temperatures used in fitting the thermodynamic properties of species K, TMP(N) is the Nth temperature, NT(K)-1 is the number of temperature ranges for which the polynomial coefficients are valid, then A (L, N, K) is the Lth polynomial coefficient, for the Nth temperature range, and the Kth species; i.e.,

```
         | <  N = 1  >. <N=2>  .                   .<  N = NT - 1>
   P  E  |    .        .       .                   .        .
   O  X  |    .        .       .                   .        .
   L  P  |    .        .       .                   .        .
   Y  R  |    .        .       .                   .        .
   N  E  |    .        .       .                   .        .
   O  S  |    .        .       .                   .        .
   M  S  |    .        .       .                   .        .
   I  I  |    .        .       .                   .        .
   A  O  |    .        .       .                   .        .
   L  N  |____.________._______.___________________.________._______
           TMP(1)   TMP(2) TMP(3) . . . . . TMP(NT-1)  TMP(NT)
```

NT - Number of temperatures used for fitting the thermodynamic properties of the species.
Data type - integer array
Dimension NT(*) at least KKTOT, the total number of species.

TMP - The temperatures which divide the temperature ranges over which the polynomial coefficients are valid.
cgs units - K
Data type - real array
Dimension TMP(*,*) exactly MAXTP (the maximum number of temperatures allowed) for the first dimension and at least KKTOT (the total number of species) for the second.

A - Three-dimensional array of fit coefficients to the thermodynamic data for the species.
The indicies in A(N,L,K) mean-
N = 1,NN represent polynomial coefficients in CP/R
CP/R(K)=A(1,L,K) + A(2,L,K)*T + A(3,L,K)*T**2 + ...
N = NN+1 is for the formation enthalpies, i.e.,
HO/R = A(NN+1,L,K)
N = NN+2 is for the formation entropies, i.e.,
SO/R = A(NN+2,L,K)
L = 1 is for temperature <= TMP(2,K)
L = 2 is for TMP(2,K) < temperature <= TMP(3)
:
L = (NTMP-1) is for TMP(NTMP-1) <= temperature;
K is the species index
Data type - real array
Dimension A(*,*,*) exactly NPCP2 (the total number of coefficients for each temperature range) for the first dimension, MXTP-1 (the maximum number of temperature ranges) for the second dimension, and at least KKTOT (the total number of species) for the third.

SKCHRG SKCHRG SKCHRG SKCHRG SKCHRG SKCHRG SKCHRG

****

****

******

SUBROUTINE SKCHRG (ISKWRK, RSKWRK, KCHARG)
Returns an array containing electronic charges of the species.

INPUT

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

KCHARG - Electronic charges of the species.
KCHARG(K)=-2 indicates that the Kth species has two excess electrons.
Data type - integer array
Dimension KCHARG(*) at least KKTOT, the total number of species.

SKCOMP SKCOMP SKCOMP SKCOMP SKCOMP SKCOMP SKCOMP

* * *

SUBROUTINE SKCOMP (ISTR, IRAY, NN, IND, NT)

Given a character string vector IRAY of vector length NN, this subroutine will search for the first occurrence of the string ISTR. The position of ISTR in IRAY is returned in the integer argument NT. If the string ISTR is not found in IRAY, IND and NT are assigned a zero value.

Consider the following example,

```
   IRAY = {"BOOK","BLUE","BEAR","BOOK"}
   NN=4.
If ISTR="BLUE" then IND=2 and NT=1;
if ISTR="RED"  then IND=0 and NT=0; and
if ISTR="BOOK",then IND=1 and NT=2.
```

INPUT

```
ISTR   - A reference character string.
IRAY   - Array of character strings.
              Data type - character array
              Dimension IRAY(*) at least NN.
NN     - Length of IRAY(*).
              Data type - integer scalar
```

OUTPUT

```
IND    - Index of the position in IRAY(*) containing ISTR.
         If ISTR is not in IRAY(*), IND=0.
              Data type - integer scalar
NT     - Total number of times ISTR occurs in IRAY.
              Data type - integer scalar
```

```
SKCONT     SKCONT     SKCONT     SKCONT     SKCONT     SKCONT     SKCONT
          **********************************************
                    **************************
                              ******

SUBROUTINE SKCONT (KSPEC, ROP, ISKWRK, RSKWRK, CIK)
   Returns the contributions of each of the surface reactions to
   the molar production rate of species KSPEC.

INPUT
   KSPEC  - Integer species number.
                 Data type - integer scalar
   ROP    - Rates of progress for the reactions.
                 cgs units - moles/(cm**2*sec)
                 Data type - real array
                 Dimension ROP(*) at least IISUR, the total number
                 of SURFACE reactions.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
                 Dimension ISKWRK(*) at least LENISK.
   RSKWRK - Array of real work space.
                 Data type - real array
                 Dimension RSKWRK(*) at least LENRSK.

OUTPUT
   CIK    - Contributions of the surface reactions to the molar
            production rate of species KSPEC.
                 cgs units - mole/(cm**2*sec)
                 Data type - real array
                 Dimension CIK(*) at least IISUR, the total number
                 of surface reactions.
```

```
SKCOV      SKCOV      SKCOV      SKCOV      SKCOV      SKCOV      SKCOV
          **********************************************
                    **************************
                              ******

SUBROUTINE SKCOV  (ISKWRK, KOCC)
   Returns an array of site occupancy numbers for the species.

INPUT
   ISKWRK - Array of integer workspace.
                 Data type - integer array

OUTPUT
   KOCC   - Site occupancy numbers for the species.
                 Data type - integer array
                 Dimension KOCC(*) at least KKTOT, the total
                 number of species.
```

```
SKCPML    SKCPML    SKCPML    SKCPML    SKCPML    SKCPML    SKCPML
          **********************************************
                    **************************
                              ******

SUBROUTINE SKCPML (T, ISKWRK, RSKWRK, CPML)
   Returns an array of the specific heats at constant pressure
   in molar units.

INPUT
   T      - Temperature.
                cgs units - K
                Data type - real scalar
   ISKWRK - Array of integer workspace.
                Data type - integer array
   RSKWRK - Array of real workspace.
                Data type - real array

OUTPUT
   CPML   - Specific heats at constant pressure in molar units
            for the species.
                cgs units - ergs/(mole*K)
                Data type - real array
                Dimension CPML(*) at least KKTOT, the total
                number of species.
```

```
SKCPMS    SKCPMS    SKCPMS    SKCPMS    SKCPMS    SKCPMS    SKCPMS
          **********************************************
                    **************************
                              ******

SUBROUTINE SKCPMS (T, ISKWRK, RSKWRK, CPMS)
   Returns an array of the specific heats at constant pressure
   in mass units.

INPUT
   T      - Temperature.
                cgs units - K
                Data type - real scalar
   ISKWRK - Array of integer workspace.
                Data type - integer array
   RSKWRK - Array of real workspace.
                Data type - real array

 OUTPUT
   CPMS   - Specific heats at constant pressure in mass units
            for the species.
                cgs units - ergs/(gm*K)
                Data type - real array
                Dimension CPMS(*) at least KKTOT, the total
                number of species.
```

```
SKCPOR    SKCPOR    SKCPOR    SKCPOR    SKCPOR    SKCPOR    SKCPOR
          **************************************************
                    ****************************
                              ******

SUBROUTINE SKCPOR (T, ISKWRK, RSKWRK, CPOR)
   Returns an array of the nondimensional specific heats at constant
   pressure.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

 OUTPUT
   CPOR   - Nondimensional specific heats at constant pressure
            for the species.
                 cgs units - none
                 Data type - real array
                 Dimension CPOR(*) at least KKTOT, the total
                 number of species.
```

```
SKDEN     SKDEN     SKDEN     SKDEN     SKDEN     SKDEN     SKDEN
          **************************************************
                    ****************************
                              ******

SUBROUTINE SKDEN  (P, T, ACT, SDEN, ISKWRK, RSKWRK, DEN)
   Returns a real array of species densities.

INPUT
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                 cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                 cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
```

a bulk phase should be 1.
cgs units - none
Data type - real array
Dimension ACT(*) at least KKTOT, the total number of species.

SDEN - Site densities for the surface site types. This vector may have an entry for each phase, including the gas phase, but the subroutine only uses entries for the surface site phases, NFSURF .LE. N .LE. NLSURF.
cgs units - moles/cm**2
Data type - real array
Dimension SDEN(*) at least NPHASE, the total number of phases.

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

DEN - Densities for the species.
NOTE: mass densities are not required to be input to the Interpreter for bulk-phase species. If they are input, they are returned by this subroutine. If not, DEN = -1.0 for the bulk species
cgs units - gm/(cm**3) for gas-phase phase species
gm/(cm**2) for surface species
gm/(cm**3) for bulk species
Data type - real array
Dimension DEN(*) at least KKTOT, the total number of species.

```
SKDRDA    SKDRDA    SKDRDA    SKDRDA    SKDRDA    SKDRDA    SKDRDA
          **************************************************
                    *****************************
                              *******

SUBROUTINE SKDRDA (IR, P, T, ACT, SDEN, ISKWRK, RSKWRK, DKDAI)
   Returns the partial of the rates of production for each of the
   species with respect to the pre-exponential constant of surface
   reaction IR.

INPUT
   IR     - Reaction index
                 Data type - integer scalar
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                 cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                 cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
            a bulk phase should be 1.
                 cgs units - none
                 Data type - real array
                 Dimension ACT(*) at least KKTOT, the total number
                 of species.
   SDEN   - Site densities for the surface site types.  This
            vector may have an entry for each phase, including the
            gas phase, but the subroutine only uses entries for the
            surface site phases, NFSURF .LE. N .LE. NLSURF.
                 cgs units - moles/cm**2
                 Data type - real array
                 Dimension SDEN(*) at least NPHASE, the total
                 number of phases.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   DKDAI  - Array of the partial of the production rates of the
            species with respect to the pre-exponential
            constant for reaction IR.
                 cgs units - moles/(cm**2*sec) / (units of A)
                 Data type - real array
                 Dimension DKDAI(*) at least KKTOT, the total
                 number of species.
```

```
SKDRDC    SKDRDC    SKDRDC    SKDRDC    SKDRDC    SKDRDC    SKDRDC
          **********************************************
                    **************************
                              ******

SUBROUTINE SKDRDC (KSPEC, P, T, ACT, SDEN, ISKWRK, RSKWRK, DKDC)
   Returns the partial derivative of the production rates for
   each of the species with respect to the concentration of species
   KSPEC.

INPUT
   KSPEC  - Species index
                 Data type - integer scalar
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                 cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                 cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
            a bulk phase should be 1.
                 cgs units - none
                 Data type - real array
                 Dimension ACT(*) at least KKTOT, the total number
                 of species.
   SDEN   - Site densities for the surface site types.  This
            vector may have an entry for each phase, including the
            gas phase, but the subroutine only uses entries for the
            surface site phases, NFSURF .LE. N .LE. NLSURF.
                 cgs units - moles/cm**2
                 Data type - real array
                 Dimension SDEN(*) at least NPHASE, the total
                 number of phases.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   DKDC   - Array of the partial of the production rates of the
            species with respect to the concentration
            of species KSPEC.
                 cgs units - moles/(cm**2*sec) / (units of KSPEC)
                 Data type - real array
                 Dimension DKDC(*) at least KKTOT, the total
                 number of species.
```

```
SKEQ      SKEQ      SKEQ      SKEQ      SKEQ      SKEQ      SKEQ
          **************************************************
                    ****************************
                              ******

SUBROUTINE SKEQ    (P, T, ACT, SDEN, ISKWRK, RSKWRK, EQKC)
   Returns the equilibrium constants for the surface reactions
   given pressure, temperature, species activities, and the site
   densities.

INPUT
   P      - Pressure.
                cgs units - dynes/cm**2
                Data type - real scalar
   T      - Temperature.
                cgs units - K
                Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
            a bulk phase should be 1.
                cgs units - none
                Data type - real array
                Dimension ACT(*) at least KKTOT, the total number
                of species.
   SDEN   - Site densities for the surface site types.  This
            vector may have an entry for each phase, including the
            gas phase, but the subroutine only uses entries for the
            surface site phases, NFSURF .LE. N .LE. NLSURF.
                cgs units - moles/cm**2
                Data type - real array
                Dimension SDEN(*) at least NPHASE, the total
                number of phases.
   ISKWRK - Array of integer workspace.
                Data type - integer array
                Dimension ISKWRK(*) at least LENISK.
   RSKWRK - Array of real work space.
                Data type - real array
                Dimension RSKWRK(*) at least LENRSK.

OUTPUT
   EQKC   - Equilibrium constants in concentration units
            for the reactions.
                cgs units - (moles,cm), depends on reaction
                Data type - real array
                Dimension EQKC(*) at least IISUR, the total
                number of surface reactions.
```

```
SKGML     SKGML     SKGML     SKGML     SKGML     SKGML     SKGML
          **********************************************
                    **************************
                              ******

SUBROUTINE SKGML  (T, ISKWRK, RSKWRK, GML)
   Returns an array of the standard state Gibbs free energies
   in molar units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   GML    - Standard state Gibbs free energies in molar units
            for the species.
                 cgs units - ergs/mole
                 Data type - real array
                 Dimension GML(*) at least KKTOT, the total
                 number of species.
```

```
SKGMS     SKGMS     SKGMS     SKGMS     SKGMS     SKGMS     SKGMS
          **********************************************
                    **************************
                              ******

SUBROUTINE SKGMS  (T, ISKWRK, RSKWRK, GMS)
   Returns an array of the standard state Gibbs free energies
   in mass units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   GMS    - Standard state Gibbs free energies in mass units
            for the species.
                 cgs units - ergs/gm
                 Data type - real array
                 Dimension GMS(*) at least KKTOT, the total
                 number of species.
```

```
SKHML     SKHML     SKHML     SKHML     SKHML     SKHML     SKHML
          **********************************************
                    **************************
                              ******

SUBROUTINE SKHML  (T, ISKWRK, RSKWRK, HML)
   Returns an array of the enthalpies in molar units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   HML    - Enthalpies in molar units for the species.
                 cgs units - ergs/mole
                 Data type - real array
                 Dimension HML(*) at least KKTOT, the total
                 number of species.


SKHMS     SKHMS     SKHMS     SKHMS     SKHMS     SKHMS     SKHMS
          **********************************************
                    **************************
                              ******

SUBROUTINE SKHMS  (T, ISKWRK, RSKWRK, HMS)
   Returns an array of the enthalpies in mass units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   HMS    - Enthalpies in mass units for the species.
                 cgs units - ergs/gm
                 Data type - real array
                 Dimension HMS(*) at least KKTOT, the total
                 number of species.
```

SKHORT SKHORT SKHORT SKHORT SKHORT SKHORT SKHORT

****

****

******

SUBROUTINE SKHORT (T, ISKWRK, RSKWRK, HORT)
Returns an array of the nondimensional enthalpies.

INPUT
T - Temperature.
cgs units - K
Data type - real scalar
ISKWRK - Array of integer workspace.
Data type - integer array
RSKWRK - Array of real workspace.
Data type - real array

OUTPUT
HORT - Nondimensional enthalpies for the species.
cgs units - none
Data type - real array
Dimension HORT(*) at least KKTOT, the total number of species.

```
SKICOV     SKICOV     SKICOV     SKICOV     SKICOV     SKICOV     SKICOV
           ************************************************
                      ***************************
                                 ******

SUBROUTINE SKICOV (IR, NDIM, ISKWRK, RSKWRK, NCOVI, KCOVI, CPARI)
   Returns the coverage species index numbers and their coverage
   parameters for reaction IR.

INPUT
   IR     - Surface reaction index number.
                 Data type - integer scalar
   NDIM   - Actual first dimension of CPAR.
                 Data type - integer scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   NCOVI  - Total number of species that modify the rate of
            reaction IR through coverage dependence.
                 Data type - integer scalar
   KCOVI  - Index numbers for the NCOVI species that modify the
            rate of reaction IR through coverage dependence.
                 Data type - integer array
                 Dimension KCOVI(*) at least KKTOT, the
                 total number of species.
   CPARI - Coverage parameters for the coverage species
            of reaction IR.
                 Data type - real array
                 Dimension CPARI(*,*) exactly NSCOV for the
                 first dimension, the total number of coverage
                 parameters required for each coverage species,
                 and at least KKTOT for the second dimension,
                 the total number of species.
```

SKINDX SKINDX SKINDX SKINDX SKINDX SKINDX SKINDX

****

****

******

```
SUBROUTINE SKINDX (NELEM, KKGAS, KKSUR, KKBULK, KKTOT, NNPHAS,
                   NNSURF, NFSURF, NLSURF, NNBULK, NFBULK,
                   NLBULK, IISUR)
```

Returns a group of indices defining the size of the surface reaction mechanism.

INPUT

NONE

OUTPUT

NELEM	- Number of elements. Data type - integer scalar
KKGAS	- Number of gas-phase species. Data type - integer scalar
KKSUR	- Number of surface species. Data type - integer scalar
KKBULK	- Total number of bulk species. Data type - integer scalar
KKTOT	- Total number of species. KKTOT=KKGAS+KKSUR+KKBULK Data type - integer scalar
NNPHAS	- Number of phases; gas + sites + bulk. Data type - integer scalar
NNSURF	- Number of surface phases. Data type - integer scalar
NFSURF	- Pointer to the first surface phase. Data type - integer scalar
NLSURF	- Pointer to the last surface phase. Data type - integer scalar
NNBULK	- Number of bulk phases. Data type - integer scalar
NFBULK	- Pointer to the first bulk phase. Data type - integer scalar
NLBULK	- Pointer to the last bulk phase. Data type - integer scalar
IISUR	- Number of surface reactions. Data type - integer scalar

```
SKINIT    SKINIT    SKINIT    SKINIT    SKINIT    SKINIT    SKINIT
          **********************************************
                    **************************
                              ******

SUBROUTINE SKINIT (LSIWK, LSRWK, LSCWK, LINSK, LOUT,
                   ISKWRK, RSKWRK, CSKWRK)
   Reads the surface linking file and creates the internal work
   arrays ISKWRK, RSKWRK, and CSKWRK.  SKINIT must be called before
   any other Surface Chemkin subroutine is called.  The work arrays
   must then be made available as input to the other Surface Chemkin
   subroutines.

INPUT
   LSIWK  - Dimension of ISKWRK
                 Data type - integer scalar
   LSRWK  - Dimension of RSKWRK
                 Data type - integer scalar
   LSCWK  - Dimension of CSKWRK
                 Data type - integer scalar
   LINSK  - Unit number assigned to linking file
                 Data type - integer scalar
   LOUT   - Unit number assigned for output
                 Data type - integer scalar

OUTPUT
   ISKWRK - Array of integer workspace containing integer data.
                 Data type - integer array
   RSKWRK - Array of real workspace containing real data.
                 Data type - real array
   CSKWRK - Array of character workspace containing character data.
                 Data type - CHARACTER*16 array
```

```
SKISTK    SKISTK    SKISTK    SKISTK    SKISTK    SKISTK    SKISTK
          **********************************************
                    **************************
                              ******

SUBROUTINE SKISTK (IR, ISKWRK, ISTFL)
   Returns an integer flag to indicate whether reaction IR uses
   sticking coefficients.

INPUT
   IR     - Surface reaction index number.
                 Data type - integer scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array

OUTPUT
   ISTFL  - 0 if reaction IR does not use sticking coefficients
            1 if reaction IR does use sticking coefficients
                 Data type - integer scalar
```

SKLEN SKLEN SKLEN SKLEN SKLEN SKLEN SKLEN

****

****

******

SUBROUTINE SKLEN (LINSK, LOUT, LENI, LENR, LENC)
Reads the first record of the linking file to return the lengths required for the integer, real, and character work arrays.

INPUT
LINSK - Unit number assigned to linking file
Data type - integer scalar
LOUT - Unit number assigned for output
Data type - integer scalar

OUTPUT
LENI - Dimension required for ISKWRK, the Surface Chemkin integer work array.
Data type - integer scalar
LENR - Dimension required for RSKWRK, the Surface Chemkin real work array.
Data type - integer scalar
LENC - Dimension required for CSKWRK, the Surface Chemkin character work array.
Data type - integer scalar

SKNCF SKNCF SKNCF SKNCF SKNCF SKNCF SKNCF

****

****

******

SUBROUTINE SKNCF (NELDIM, ISKWRK, NEL)
Returns the elemental composition of the species.

INPUT
NELDIM - First dimension of the matrix NEL.
Data type - integer scalar
ISKWRK - Array of integer workspace.
Data type - integer array

OUTPUT
NEL - Elemental compositions of the species.
NEL(M,K) is the number of atoms of element M in species K.
Data type - integer array
The first dimension of NEL(*,*) must be exactly NELDIM, which is at least NELM, the total number of elements in the problem, and the second dimension at least KKTOT, the total number of species.

```
SKNCON     SKNCON     SKNCON     SKNCON     SKNCON     SKNCON     SKNCON
           ***********************************************
                      ***************************
                                ******

SUBROUTINE SKNCON (ISKWRK, NCON)
   Returns the total number of surface reactions which do not
   conserve sites for each phase.

INPUT
   ISKWRK - Array of integer workspace.
                 Data type - integer array

OUTPUT
   NCON   - Number of surface reactions which do not conserve sites
            for each phase.
                 Date type - integer array
                 Dimension NCON(*) at least NPHASE,
                 the total number of phases
```

```
SKNU       SKNU       SKNU       SKNU       SKNU       SKNU       SKNU
           ***********************************************
                      ***************************
                                ******

SUBROUTINE SKNU   (IDIM, ISKWRK, KSTOIC, NSTOIC)
   Returns the stoichiometric coefficients of the species and the
   net change in phases for all of the surface reactions in a
   mechanism.

INPUT
   IDIM   - First dimension of the array NSTOIC.
                 Data type - integer scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array

OUTPUT
   KSTOIC - Stoichiometric coefficients for the surface reactions.
                 cgs units - none
                 Data type - integer array
                 The first dimension of NSTOIC(IDIM,*) must be
                 exactly IDIM, which is at least IISUR, the total
                 number of surface reactions, and at least KKTOT
                 for the second dimension, the total number of
                 species.
   NSTOIC - Net change of the phases for the surface reactions.
                 cgs units - none
                 Data type - integer array
                 The first dimension of INCF(IDIM,*) must be
                 exactly IDIM, which is at least IISUR, the total
                 number of surface reactions, and at least NPHASE
                 for the second dimension, the total number of
                 phases.
```

SKPCMP SKPCMP SKPCMP SKPCMP SKPCMP SKPCMP SKPCMP

****

****

******

SUBROUTINE SKPCMP (ISTR, IRAY, NN, SETS, NSETS, ISET, IND, NT)

This subroutine can do everything that the subroutine SKCOMP can do, and additionally, has the capabilities of separating the elements of IRAY into categories and then search IRAY by element and category. The categories that each element of IRAY will be assigned to are specified by the input character string vector SETS of vector length NSETS. Elements of each category in IRAY must be grouped congruously. The number of elements in each category within IRAY is specified by the input integer vector ISET. To search for the existence of an element within a category ISTR may additionally be composed of two substrings, ISTR="ELEMENT_NAME/CATEGORY_NAME/", where CATEGORY_NAME is one of the categories specified in SETS. In this case, IND will return the first position in IRAY where ELEMENT_NAME occurred within the category CATEGORY_NAME. NT will return the total number of times ELEMENT_NAME occurred within the category CATEGORY_NAME. If ELEMENT_NAME is not found within the specified category, IND and NT are returned with a value of zero. If no category is specified within ISTR, IND and NT return with the same values as they would from subroutine SKCOMP.

Consider the following example,

```
   IRAY = {"RED", "BLUE", "JADE", "RUBY", "TOPAZ", "JADE"}
   NN = 6
   SETS = {"COLORS", "STONES"},
   NSETS = 2
   ISET = {4, 2}.
```

This assumes that the elements of IRAY were grouped into two sets, consisting of 4 and 2 elements, respectively, and the following names

```
   "COLORS"  = {"RED", "BLUE", "JADE", "RUBY"}, and
   "STONES"  = {"TOPAZ", "JADE"}.
If ISTR="BLUE" then IND=2 and NT=1;
if ISTR="PINK" then IND=0 and NT=0; and
if ISTR="JADE",then IND=3 and NT=2.
If ISTR="BLUE/COLORS/" then IND=2 and NT=1;
if ISTR="BLUE/STONES/" then IND=0 and NT=0;
if ISTR="JADE/GEMS/"   then IND=0 and NT=0; and
if ISTR="JADE/STONES/",then IND=6 and NT=1.
```

INPUT

```
   ISTR     - A character string which may or may not end with a
              slash-delimited substring.
   IRAY(*)  - A reference array of character strings.
                  Data type - character array
                  Dimension IRAY(*) at least NN
   NN       - Number of entries in IRAY(*).
                  Data type - integer scalar
   SETS(*)  - A cross-reference array of character strings
              with which a subset of IRAY(*) is associated.
                  Data type - character array
                  Dimension SETS(*) at least NSETS
```

NSETS - Number of entries in SETS(*)
Data type - integer scalar
ISET(*) - Integer total number of entries in a subset of IRAY.
Data type - integer array
Dimension ISET(*) at least NSETS

OUTPUT
IND - Index of the position in IRAY(*) containing ISTR.
If ISTR is not in IRAY(*), IND = 0.
If the slash-delimited substring of ISTR is not in SETS(*), IND = 0.
If the slash-delimited substring of ISTR is in SETS(N), but the substring before the slash is not a member of the subset associated with SETS(N), IND = 0, whether or not the substring is in IRAY(*).
Data type - integer scalar
NT - Total number of times ISTR occurs in IRAY(*), or total number of times ISTR occurs in a subset of IRAY(*).

SKPKK SKPKK SKPKK SKPKK SKPKK SKPKK SKPKK
****
****
******

SUBROUTINE SKPKK (ISKWRK, KKPHAS, KFIRST, KLAST)
Returns arrays of species pointers for the phases.

INPUT
ISKWRK - Array of integer workspace.
Data type - integer array

OUTPUT
KKPHAS - The total number of species in each phase.
Data type - integer array
Dimension KKPHAS(*) at least NPHASE, the total number of phases
KFIRST - The index of the first species in each phase.
Data type - integer array
Dimension KFIRST(*) at least NPHASE, the total number of phases
KLAST - The index of the last species in each phase.
Data type - integer array
Dimension KLAST(*) at least NPHASE, the total number of phases

```
SKPNT     SKPNT     SKPNT     SKPNT     SKPNT     SKPNT     SKPNT
          **********************************************
                    **************************
                              ******

SUBROUTINE SKPNT  (LSAVE, LOUT, NPOINT, VERS, PREC, LENI, LENR,
                   LENC, KERR)
   Reads from a binary file information about a Surface Chemkin
   linking file, pointers for the Surface Chemkin Library, and
   returns lengths of work arrays.

INPUT
   LSAVE  - Integer input unit for binary data file.
                 Data type - integer scalar
   LOUT   - Integer output unit for printed diagnostics.
                 Data type - integer scalar

OUTPUT
   NPOINT - Total number of pointers.
                 Data type - integer scalar
   VERS   - Version number of the Surface Chemkin linking file.
                 Data type - real scalar
   PREC   - Machine precision of the Surface Chemkin linking file.
                 Data type - character string
   LENI   - Minimum length required for the integer work array.
                 Data type - integer scalar
   LENR   - Minimum length required for the real work array.
                 Data type - integer scalar
   LENC   - Minimum length required for the character work array.
                 Data type - integer scalar
   KERR   - Logical error flag.
```

```
SKRAEX    SKRAEX    SKRAEX    SKRAEX    SKRAEX    SKRAEX    SKRAEX
          **********************************************
                    **************************
                              ******

SUBROUTINE SKRAEX (IR, RSKWRK, RA)
   Returns the Pre-exponential rate constant
   (or sticking coefficient) of the IRth reaction, or changes its
   value, depending on the sign of IR.

INPUT
    IR     - Integer reaction number; IR>0 gets RA(I) from RSKWRK,
                                      IR<0 puts RA(I) into RSKWRK.
                  Data type - integer scalar
    RSKWRK - Array of real internal work space.
                  Data type - real array

OUTPUT
    RA     - Pre-exponential coefficient, or sticking coefficient,
             for IRth reaction
                  Data type - real scalar
                  cgs units:
                     mole-cm-sec-K for normal rate expressions
                     none for sticking coefficient reactions
```

```
SKRAT      SKRAT      SKRAT      SKRAT      SKRAT      SKRAT      SKRAT
          **********************************************
                    **************************
                              ******

SUBROUTINE SKRAT  (P, T, ACT, SDEN, ISKWRK, RSKWRK, SDOT, SITDOT)
   Returns production rates for the species and sites.

INPUT
   P       - Pressure.
                  cgs units - dynes/cm**2
   T       - Temperature.
                  cgs units - K
                  Data type - real scalar
   ACT     - Activities of the species, where
             for the first KKGAS species, ACT(*) are mole fractions
                  cgs units - none
             for the next KKSURF species, ACT(*) are site fractions,
             (species density normalized by the site density).
             The surface concentration in moles/cm**2 is:
             ACT(K)*SITE_DENSITY / # sites per species
                  cgs units - none
             for the next KKBULK species, ACT(*) are the bulk species
             activities.  A species in a bulk phase will have an
             activity from 0 to 1 and the sum of activities for
             a bulk phase should be 1.
                  cgs units - none
                  Data type - real array
                  Dimension ACT(*) at least KKTOT, the total number
                  of species.
   SDEN    - Site densities for the surface site types.  This
             vector may have an entry for each phase, including the
             gas phase, but the subroutine only uses entries for the
             surface site phases, NFSURF .LE. N .LE. NLSURF.
                  cgs units - moles/cm**2
                  Data type - real array
                  Dimension SDEN(*) at least NPHASE, the total
                  number of phases.
   ISKWRK - Array of integer workspace.
                  Data type - integer array
   RSKWRK - Array of real workspace.
                  Data type - real array

OUTPUT
   SDOT    - Production rates of the species.
             1) For K=1,KKGAS, SDOT(K) is the production rate of
                gas-phase species K in (moles/cm**2-sec).
             2) For K=KKGAS+1,KKGAS+KKSUR, SDOT(K) is the production
                rate of surface species K in (moles/cm**2-sec).
             3) For K=KKGAS+KKSUR+1,KKTOT, SDOT(K) is the production
                rate of bulk species K in (moles/cm**2-sec).
                  cgs units - moles/(cm**2*sec)
                  Data type - real array
                  Dimension SDOT(*) at least KKTOT, the total
                  number of species.
```

SITDOT - Production rates of the surface phases (subroutine calculates entries for the surface site phases only).
cgs units - moles/(cm**2*sec)
Data type - real array
Dimension SITDOT(*) at least NPHASE, the total number of phases.

```
SKROP     SKROP     SKROP     SKROP     SKROP     SKROP     SKROP
          **********************************************
                    **************************
                              ******

SUBROUTINE SKROP  (P, T, ACT, SDEN, ISKWRK, RSKWRK, ROP)
   Returns rates of progress for the surface reactions.

INPUT
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ACT    - Activities of the species, where
            for the first KKGAS species, ACT(*) are mole fractions
                 cgs units - none
            for the next KKSURF species, ACT(*) are site fractions,
            (species density normalized by the site density).
            The surface concentration in moles/cm**2 is:
            ACT(K)*SITE_DENSITY / # sites per species
                 cgs units - none
            for the next KKBULK species, ACT(*) are the bulk species
            activities.  A species in a bulk phase will have an
            activity from 0 to 1 and the sum of activities for
            a bulk phase should be 1.
                 cgs units - none
                 Data type - real array
                 Dimension ACT(*) at least KKTOT, the total number
                 of species.
   SDEN   - Site densities for the surface site types.  This
            vector may have an entry for each phase, including the
            gas phase, but the subroutine only uses entries for the
            surface site phases, NFSURF .LE. N .LE. NLSURF.
                 cgs units - moles/cm**2
                 Data type - real array
                 Dimension SDEN(*) at least NPHASE, the total
                 number of phases.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   ROP    - Rates of progress for the surface reactions.
                 cgs units - moles/(cm**2*sec).
                 Data type - real array
                 Dimension ROP(*) at least IISUR, the total number of
                 surface reactions.
```

```
SKRP      SKRP      SKRP      SKRP      SKRP      SKRP      SKRP
          ************************************************
                    ***************************
                              ******

SUBROUTINE SKRP   (ISKWRK, RSKWRK, RU, RUC, PATM)
   Returns universal gas constants and the pressure of one standard
   atmosphere.

INPUT
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real work space.
                 Data type - real array

OUTPUT
   RU     - Universal gas constant.
                 cgs units - 8.314E7 ergs/(mole*K)
                 Data type - real scalar
   RUC    - Universal gas constant used only in conjuction with
            activation energy.
                 preferred units - 1.987 cal/(mole*K)
                 Data type - real scalar
   PATM   - Pressure of one standard atmosphere.
                 cgs units - 1.01325E6 dynes/cm**2
                 Data type - real scalar
```

```
SKRROP    SKRROP    SKRROP    SKRROP    SKRROP    SKRROP    SKRROP
          ***********************************************
                    **************************
                              ******

SUBROUTINE SKRROP (ISKWRK, RSKWRK, IISUR, SMH, MAXSPR, RU, PATM,
                   KKGAS, KKSUR, P, T, CZ, WT, NREAC, NRPP, NU,
                   NUNK, NUSUMK, NSPAR, PAR, RPAR, NREV, IREV,
                   NCOV, ICOV, KCOV, NDIM, CPAR, NSTK, ISTK,
                   SDTOT, RKF, RKR, EQKC)
   Returns forward and reverse rates of progress and equilibrium
   constants for the surface reactions.
   It is not normally called by the user application code.

INPUT
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array
   IISUR  - Total number of surface reactions in the mechanism.
                 Data type - integer scalar
   SMH    - Entropies minus enthalpies for the species;
            SMH(K) = S(K)/R - H(K)/RT.
                 cgs units none
                 Data type - real array
                 Dimension SMH(*) at least KKTOT, the total
                 number of species.
   MAXSPR - Maximum number of species in a surface reaction.
                 Data type - integer scalar
   RU     - Universal gas constant.
                 cgs units - 8.314E7 ergs/(mole*K)
                 Data type - real scalar
   PATM   - Pressure of one standard atmosphere.
                 cgs units - 1.01325E6 dynes/cm**2
                 Data type - real scalar
   KKGAS  - Total number of gas-phase species.
                 Data type - integer scalar
   KKSUR  - Total number of site-phase species.
                 Data type - integer scalar
   P      - Pressure.
                 cgs units - dynes/cm**2
                 Data type - real scalar
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   CZ     - Matrix of the concentrations of the gas-phase and
            surface species in the problem, and the activities
            of the bulk species.  The first KKGAS entries of CZ
            are the gas-phase molar concentrations (moles/cm**3).
            The next KKSURF entries are the surface species molar
            concentrations (moles/cm**2).  The final KKBULK entries
            are the activities of the bulk species.
                 Data type - real array
                 Dimension CZ(*) at least KKTOT, the total
                 number of species.
   WT     - Array of molecular weights for the species.
                 Data type - real array
                 Dimension WT(*) at least KKTOT, the total
                 number of species
```

NREAC - Array of the number of reactants in the surface reactions.
Data type - integer array
Dimension NREAC(*) at least IISUR, the total number of surface reactions.

NRPP - Integer vector that indicates the reversibility and the total number of species (reactants plus products) of the IISUR surface reactions;
+NRPP = reversible surface reaction IR has NRPP reactants and products
-NRPP = irreversible reaction IR has ABS(NRPP) reactants and products
Data type - integer array
Dimension NRPP(*) at least IISUR, the total number of surface reactions.

NU - Matrix of stoichiometric coefficients for the KKTOT species in the IISUR surface reactions. NU(M,IR) is the stoichiometric coefficient of the Mth species in reaction IR. These coefficients are negative for reactants and positive for products. The species number for the Mth species is stored in NUNK.
Data type - integer array
Dimension NU(*,*) exactly MAXSPR for the first dimension and at least IISUR for the second, the total number of surface reactions.

NUNK - Matrix of species numbers corresponding to the stoichiometric coefficients in NU.
Data type - integer array
Dimension NUNK(*,*) exactly MAXSPR for the first dimension and at least IISUR for the second, the total number of surface reactions.

NUSUMK - The total of the coefficients of the gas-phase species in each surface reaction.
Data type - integer array
Dimension NUSUMK(*) at least IISUR, the total number of surface reactions.

NSPAR - Number of parameters in the rate expression. In the current formulation NSPAR=3 (see PAR(N,I) below).
Data type - integer scalar

PAR - Matrix of reaction rate parameters in the form:
K = A * T**b * EXP(-E/R*T)
1) PAR(1,I) are the pre-exponential factors, A.
2) PAR(2,I) are the temperature exponents, b.
3) PAR(3,I) are the activation energies, E.
Data type - real array
Dimension PAR(*,*) exactly NSPAR for the first dimension and at least IISUR for the second, the total number of surface reactions.

RPAR - Matrix of reverse Arrhenius parameters for the NREV reactions
Data type - real array
Dimension RPAR(*,*) exactly NSPAR for the first dimension and at least NREV for the second, the total number of reactions with reverse Arrhenius parameters defined.

NREV - Number of reactions which have reverse Arrhenius parameters defined.
Data type - integer scalar

IREV - Array of reaction numbers which have reverse Arrhenius parameters defined.
Data type - integer array
Dimension IREV(*) at least NREV, the total number of reactions with reverse Arrhenius parameters defined.

NCOV - Total number of site species coverage declarations.
Data type - integer scalar

ICOV - Reaction index numbers for the NCOV coverage declarations.
Data type - integer array
Dimension ICOV(*) at least NCOV, the total number of coverage declarations.

KCOV - Species index numbers for the NCOV coverage declarations.
Data type - integer array
Dimension KCOV(*) at least NCOV, the total number of coverage declarations.

NDIM - The first dimension of the matrix of coverage parameters for the NCOV coverage declarations.
Data type - integer scalar

CPAR - Matrix of coverage parameters for the NCOV coverage declarations.
Data type - real array
Dimension CPAR(*,*) exactly NSCOV for the first dimension, the number of coverage parameters allowed, and at least NCOV for the second dimension, the total number of coverage declarations.

NSTK - Number of reactions which have sticking coefficients.
Data type - integer scalar

ISTK - Array of reaction numbers for the NSTK reactions.
Data type - integer array
Dimension ISTK(*) at least NSTK, the total number of reactions with sticking coefficients.

SDTOT - The sum of the densities of the phases.
Data type - real scalar

OUTPUT

RKF - Forward rates of progress for the surface reactions.
cgs units - moles/(cm**2*sec)
Data type - real array
Dimension RKF(*) at least IISUR, the total number of surface reactions.

RKR - Reverse rates of progress for the surface reactions.
cgs units - moles/(cm**2*sec)
Data type - real array
Dimension RKR(*) at least IISUR, the total number of surface reactions.

EQKC - Equilibrium constants in concentration units for the surface reactions.
Data type - real array
Dimension EQKC(*) at least IISUR, the total number of surface reactions.

****
****
******

```
SUBROUTINE SKRATI (IR, ROP, ISKWRK, RSKWRK, SDOTI, SITDTI)
   Returns rates of production for each of the species by
   surface reaction IR.

INPUT
   IR     - Reaction index
                 Data type - integer scalar
   ROP    - Rates of progress for the surface reactions.
                 cgs units - moles/(cm**2*sec).
                 Data type - real array
                 Dimension ROP(*) at least IISUR, the total number
                 of surface reactions.
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   SDOTI  - Production rates of the species in reaction IR.
            1) For K=1,KKGAS, SDOTI(K) is the production rate of
               gas-phase species K in (moles/cm**2-sec).
            2) For K=KKGAS+1,KKGAS+KKSUR, SDOTI(K) is the production
               rate of surface species K in (moles/cm**2-sec).
            3) For K=KKGAS+KKSUR+1,KKTOT, SDOTI(K) is the production
               rate of bulk species K in (moles/cm**2-sec).
                 cgs units - moles/(cm**2*sec)
                 Data type - real array
                 Dimension SDOTI(*) at least KKTOT, the total
                 number of species.
  SITDTI  - Production rates of the surface phases due to reaction
            IR (subroutine calculates entries for the surface site
            phases only).
                 cgs units - moles/(cm**2*sec)
                 Data type - real array
                 Dimension SITDTI(*) at least NPHASE, the total
                 number of phases.
```

SKSAVE SKSAVE SKSAVE SKSAVE SKSAVE SKSAVE SKSAVE

****

****

******

SUBROUTINE SKSAVE (LINC, LOUT, LSAVE, ISKWRK, RSKWRK, CSKWRK)

Writes to a binary file information about a Surface Chemkin linking file, pointers for the Surface Chemkin Library, and Surface Chemkin work arrays.

INPUT

LINC - Logical file number for the linking file.
Data type - integer scalar

LOUT - Output file for printed diagnostics.
Data type - integer scalar

LSAVE - Integer output unit.
Data type - integer scalar

ISKWRK - Array of integer workspace containing integer data.
Data type - integer array

RSKWRK - Array of real workspace containing real data.
Data type - real array

CSKWRK - Array of character workspace containing character data.
Data type - CHARACTER*16 array

SKSDEN SKSDEN SKSDEN SKSDEN SKSDEN SKSDEN SKSDEN

****

****

******

SUBROUTINE SKSDEN (RSKWRK, SDEN0)

Returns a real array of standard-state phase densities as given on input to the interpreter.

INPUT

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

SDEN0 - Standard-state densities for the surface site types, AS READ BY THE INTERPRETER. The SDEN0 vector has an entry for each phase, including the gas phase, but this subroutine only writes into those entries related to the surface phases, i.e. NFSURF .LE. N .LE. NLSURF.
cgs units - moles/cm**2
Data type - real array
Dimension SDEN0(*) at least NPHASE, the total number of phases.

```
SKSMH     SKSMH     SKSMH     SKSMH     SKSMH     SKSMH     SKSMH
          **********************************************
                    **************************
                              ******

SUBROUTINE SKSMH  (T, ISKWRK, RSKWRK, SMH)
   Returns the array of dimensionless entropies minus enthalpies
   for the species.  It is normally not called directly by the user.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   SMH    - Dimensionless entropies minus enthalpies for the
            species;  SMH(K) = S(K)/R - H(K)/RT.
                 cgs units none
                 Data type - real array
                 Dimension SMH(*) at least KKTOT, the total
                 number of species.
```

```
SKSML     SKSML     SKSML     SKSML     SKSML     SKSML     SKSML
          **********************************************
                    **************************
                              ******

SUBROUTINE SKSML  (T, ISKWRK, RSKWRK, SML)
   Returns an array of the standard state entropies in molar units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   SML    - Standard state entropies in molar units for the species.
                 cgs units - ergs/(mole*K)
                 Data type - real array
                 Dimension SML(*) at least KKTOT, the total
                 number of species.
```

```
SKSMS      SKSMS      SKSMS      SKSMS      SKSMS      SKSMS      SKSMS
           **********************************************
                      **************************
                                ******

SUBROUTINE SKSMS  (T, ISKWRK, RSKWRK, SMS)
   Returns an array of the standard state entropies in mass units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   SMS    - Standard state entropies in mass units for the species.
                 cgs units - ergs/(gm*K)
                 Data type - real array
                 Dimension SMS(*) at least KKTOT, the total
                 number of species.
```

KNAM - Array of species names.
Data type - character array
Dimension KNAM(*) at least KKTOT, the total number of species.
KKTOT - Total number of species.
Data type - integer scalar
PNAM - Array of phase names.
Data type - character array
Dimension PNAM(*) at least NNPHAS, the total number of phases.
NNPHAS - Total number of phases.
Data type - integer scalar
KKPHAS - Array of the number of species in the phases.
Data type - integer array
Dimension KKPHAS(*) at least NNPHAS, the total number of phases.

OUTPUT
KNUM - Index number of species which corresponds to the species name in LINE.
Data type - integer scalar
NT - Number of times the species name occurs in the linking file.
Data type - integer scalar
NVAL - Number of real values found in LINE
Data type - integer scalar
RVAL - Array of real values found in LINE
Data type - real array
Dimension RVAL(*) at least NEXP
KERR - Error flag; syntax or dimensioning error, corresponding species not found, or total number of values found is not the number of values expected, will result in KERR = .TRUE.
Data type - logical

SKSOR SKSOR SKSOR SKSOR SKSOR SKSOR SKSOR

****

****

******

SUBROUTINE SKSOR (T, ISKWRK, RSKWRK, SOR)
Returns an array of the nondimensional entropies.

INPUT
T - Temperature.
cgs units - K
Data type - real scalar
ISKWRK - Array of integer workspace.
Data type - integer array
RSKWRK - Array of real workspace.
Data type - real array

OUTPUT
SOR - Nondimensional entropies for the species.
cgs units - none
Data type - real array
Dimension SOR(*) at least KKTOT, the total number of species.

SKSNUM SKSNUM SKSNUM SKSNUM SKSNUM SKSNUM SKSNUM

****

****

******

```
SUBROUTINE SKSNUM (LINE, NEXP, LOUT, KNAM, KKTOT, PNAM, NNPHAS,
                   KKPHAS, KNUM, NT, NVAL, RVAL, KERR)
```

This subroutine is used to read a format-free input line of combined alphanumerical data. It can be used to parse an input character string, LINE, which may be composed of several blank-delimited substrings. This subroutine assumes that the first substring in LINE is the name of a species in the Surface Chemkin mechanism. If the species name is not unique within the Surface Chemkin mechanism, the phase of the species should be input immediately after the species name, delimited by slashes.

Upon return from the subroutine, KNUM returns the index position of the species within the Surface Chemkin linking file. If the species name is not unique, KNUM returns the first position and NT returns the number of the times the species occurs within the linking file. If the species name is not found, or there is a syntax error, on return, KNUM=0, NT=0, and KERR=.TRUE.

The substrings in LINE following the first are expected to represent numbers. They are converted into floating point values and stored in the output vector, RVAL(*). Upon input, NEXP is equal to the number of values expected to be found. If NEXP numbers are not found, KERR will be set to .TRUE. on return from the subroutine.

Example input:

LINE = GA(S)/BULK1/ 1.2
NEXP = 1, the number of values expected
LOUT = 6, a logical unit number on which to write diagnostic messages
KNAM(*) = Array of character species names
KKTOT = Total number of species
PNAM(*) = Array of character phase names
NNPHAS = Total number of phases
KKPHAS(*)= Index array of the number of species in the phases

Output:

KNUM = The index number of the species which has the name "GA(S)" and resides in phase "BULK1"
NT = 1, if there is only one species GA(S) in phase BULK1
NVAL = 1, the number of values found in LINE following the species name
RVAL(1) = 1.200E+00, the substring converted to a real number
KERR = .FALSE.

INPUT

LINE - A character string.
Data type - CHARACTER*80

NEXP - Number of real values expected to be found in character string.
Data type - integer scalar

LOUT - Output unit for printed diagnostics.
Data type - integer scalar

```
SKSYME    SKSYME    SKSYME    SKSYME    SKSYME    SKSYME    SKSYME
          **************************************************
                    ****************************
                              ******

SUBROUTINE SKSYME (CSKWRK, LOUT, ENAM, KERR)
   Returns a character array of element names.

INPUT
   CSKWRK - Array of character workspace.
                 Data type - CHARACTER*16 array
   LOUT   - Output unit for printed diagnostics.
                 Data type - integer scalar

OUTPUT
   ENAM   - Element names.
                 Data type - CHARACTER*(*) array
                 Dimension ENAM(*) at least NELM,  the total
                 number of elements in the problem.
   KERR   - Error flag; character length errors will result in
            KERR = .TRUE.
                 Data type - logical
```

```
SKSYMP    SKSYMP    SKSYMP    SKSYMP    SKSYMP    SKSYMP    SKSYMP
          **************************************************
                    ****************************
                              ******

SUBROUTINE SKSYMP (CSKWRK, LOUT, PNAM, KERR)
   Returns a character array of phase names.

INPUT
   CSKWRK - Array of character workspace.
                 Data type - CHARACTER*16 array
   LOUT   - Output unit for printed diagnostics.
                 Data type - integer scalar

OUTPUT
   PNAM   - Phase names.
                 Data type - CHARACTER*(*) array
                 Dimension PNAM(*) at least NNPHAS, the total
                 number of sites.
   KERR   - Error flag; character length errors will result in
            KERR = .TRUE.
                 Data type - logical
```

SKSYMR SKSYMR SKSYMR SKSYMR SKSYMR SKSYMR SKSYMR

****

****

******

SUBROUTINE SKSYMR (IR, LOUT, ISKWRK, RSKWRK, CSKWRK, LT, RNAM, KERR)

Returns the character string representation of reaction IR.

INPUT

IR - Reaction index
Data type - integer scalar

LOUT - Output unit for printed diagnostics.
Data type - integer scalar

ISKWRK - Array of integer workspace.
Data type - integer array

RSKWRK - Array of real workspace.
Data type - real array

CSKWRK - Array of character workspace.
Data type - CHARACTER*16 array

OUTPUT

LT - Total number of non-blank characters in the reaction string.
Data type - integer scalar

RNAM - Reaction string.
Data type - CHARACTER*(*)

KERR - Error flag; character length errors will result in KERR = .TRUE.
Data type - logical

SKSYMS SKSYMS SKSYMS SKSYMS SKSYMS SKSYMS SKSYMS

****

****

******

SUBROUTINE SKSYMS (CSKWRK, LOUT, KNAM, KERR)

Returns a character array of species names.

INPUT

CSKWRK - Array of character workspace.
Data type - CHARACTER*16 array

LOUT - Output unit for printed diagnostics.
Data type - integer scalar

OUTPUT

KNAM - Species names.
Data type - CHARACTER*(*) array
Dimension KNAM(*) at least KKTOT, the total number of species.

KERR - Error flag; character length errors will result in KERR = .TRUE.
Data type - logical

```
SKUML     SKUML     SKUML     SKUML     SKUML     SKUML     SKUML
          **********************************************
                    **************************
                              ******

SUBROUTINE SKUML  (T, ISKWRK, RSKWRK, UML)
   Returns an array of the internal energies in molar units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   UML    - Internal energies in molar units for the species.
                 cgs units - ergs/mole
                 Data type - real array
                 Dimension UML(*) at least KKTOT, the total
                 number of species.
```

```
SKUMS     SKUMS     SKUMS     SKUMS     SKUMS     SKUMS     SKUMS
          **********************************************
                    **************************
                              ******

SUBROUTINE SKUMS  (T, ISKWRK, RSKWRK, UMS)
   Returns an array of the internal energies in mass units.

INPUT
   T      - Temperature.
                 cgs units - K
                 Data type - real scalar
   ISKWRK - Array of integer workspace.
                 Data type - integer array
   RSKWRK - Array of real workspace.
                 Data type - real array

OUTPUT
   UMS    - Internal energies in mass units for the species.
                 cgs units - ergs/gm
                 Data type - real array
                 Dimension UMS(*) at least KKTOT, the total
                 number of species.
```

SKWT SKWT SKWT SKWT SKWT SKWT SKWT

****

****

******

SUBROUTINE SKWT (RSKWRK, WT)

Returns the molecular weights of the species.

INPUT

RSKWRK - Array of real workspace.
Data type - real array

OUTPUT

WT - Molecular masses for the species.
cgs units - gm/mole
Data type - real array
Dimension WT(*) at least KKTOT, the total number of species.

IX. SAMPLE PROBLEM

The problem that we have chosen for an example involves the time-dependent deposition of a solid film on the surfaces of a fixed-volume, fixed-temperature container that was initially filled with a gas mixture. As with any new application for the Surface Chemkin package, one of the first tasks is to derive a system of equations that describe the process to be simulated. Here, the first equation involves the conservation of mass in the container:

$$\frac{dm_k}{dt} = V\dot{\omega}_k W_k + A\dot{s}_k W_k, \qquad (k = 1, \ldots, K_g) \tag{67}$$

where t is time, m_k is the mass of gas-phase species k in the container, V is the container volume, $\dot{\omega}_k$ is the molar production (destruction) rate of gas-phase species by gas-phase chemical reaction, W_k are the species molecular weights, A is the container-wall surface area, and $\dot{s}_k$ is the molar production rate of gas-phase species by surface reaction. After introducing the gas-phase mass density $\rho = m/V$ (where m is the total gas-phase mass) and the gas-phase species mass fractions $Y_k = m_k/m$, some manipulation leads to the following equation:

$$\frac{dY_k}{dt} = -\frac{Y_k}{\rho}\frac{d\rho}{dt} + \frac{\dot{\omega}_k W_k}{\rho} + \frac{1}{\rho}\frac{A}{V}\dot{s}_k W_k. \qquad (k = 1, \ldots, K_g) \tag{68}$$

The total mass in the gas phase depends on the production (destruction) of gas-phase species by surface reaction, as stated by

$$\frac{dm}{dt} = \sum_{k=1}^{K_g} A\dot{s}_k W_k. \tag{69}$$

We rewrite this equation slightly to make ρ a dependent variable and use the area-to-volume ratio A/V as a parameter:

$$\frac{d\rho}{dt} = \sum_{k=1}^{K_g} \frac{A}{V}\dot{s}_k W_k. \tag{70}$$

On the surface, the number of moles of species k is given by

$$N_k = Z_k(n)\Gamma_n A/\sigma_k(n), \tag{71}$$

where $Z_k(n)$ is the site fraction of species k on surface site n (fraction of sites occupied by species k in phase n), Γ_n is the density of sites in phase n (in moles/cm^2), and $\sigma_k(n)$

is the number of sites that species k occupies. The molar production (destruction) rate of surface species k by surface reaction is stated as

$$\frac{dN_k}{dt} = A\dot{s}_k \qquad (k = K_s^f, \ldots, K_s^l) \tag{72}$$

In terms of site fractions $Z_k(n)$, the equation governing the surface species is given as

$$\frac{dZ_k(n)}{dt} = \frac{\dot{s}_k \sigma_k(n)}{\Gamma_n} - \frac{Z_k(n)}{\Gamma_n}\frac{d\Gamma_n}{dt}, \tag{73}$$

where the last term can be dropped if the number of surface sites is fixed. When surface reactions create or destroy sites, then a conservation equation is included for the site densities of each surface phase n:

$$\frac{d\Gamma_n}{dt} = \dot{\Gamma}_n. \tag{74}$$

However, our sample mechanism conserves sites, so the time derivative in Eq. (74) is trivially zero.

The governing system of ordinary differential equations and accompanying initial conditions form an initial value problem. The equations will be solved using the code LSODE[9] written by Alan Hindmarsh. We find this code to be highly reliable for the solution of wide range of stiff initial-value problems.

The Fortran code for solution of the sample problem is given in Section 7 below. After initializing the gas-phase Chemkin Library and the Surface Library, the code reads the initial nonzero moles from input. It then repeatedly calls subroutine LSODE to obtain the solution at uniform print intervals. The governing equation formulation is found in SUBROUTINE FUN, which is called by LSODE.

The sections below present a VAX command procedure for the sample problem, gas-phase Chemkin Interpreter input and output, Surface Chemkin Interpreter input and output, the input to the sample problem, Fortran code, and output for the sample problem. The last section describes how to use LSODE.

Discussion of Sample Problem

We illustrate the input and output of the various Interpreters and example problem with an analysis of the deposition of Si_3N_4. The gas-phase reaction mechanism contains a detailed description of NH_3 decomposition (about which there is much published information), two reactions describing SiF_4 decomposition, and three cross-reactions. At the low pressures we consider, the gas-phase decomposition of reactants is slow. The surface reaction mechanism contains six steps describing the overall conversion of 3 SiF_4 and 4 NH_3 molecules to 3 Si(d) and 4 N(d) and 12 HF. (Note that the surface reaction mechanism is from a preliminary analysis at one temperature, and thus we have not supplied any activation energies. As such, this mechanism should only be considered as illustrative and not as a source of kinetic data on the Si_3N_4 system.)

The input to the sample problem gives the initial pressure as 2.63×10^{-3} atm (2 Torr) and temperature 1713 K. In this problem the temperature is fixed, but the pressure will increase as 12 moles of HF are produced for every 7 moles of reactant destroyed. The input gas-phase mole fractions represent a 6:1 ratio of NH_3 to SiF_4. Initial site fractions of the surface species came from a steady-state analysis of the system (not discussed here). The initial activities of the two bulk species are set to 1. The area to volume ratio is 6 (a cubic box).

The print-out from the sample problem shows the initial conditions followed by print-outs of the concentrations at subsequent stages of the deposition. The reactants SiF_4 and NH_3 are seen to be depleted and the product HF forms. In this fixed-volume system the pressure rises (discussed above). There is a net decrease in the gas density as the heavy Si and N atoms are lost from the gas into the bulk.

1. VAX Command Procedure

VAX/VMS Commands			Meaning
$assign	GASMECH.DAT	FOR015	Assign the user's gas-phase reaction mechanism to Fortran unit 15. This is the input file for the Chemkin Interpreter.
$assign	CKINTERP.OUT	FOR016	Assign the output file for printed output from the Chemkin Interpreter.
$assign	CKTHERMO.DAT	FOR017	Assign the Thermodynamic Database to Fortran unit 17.
$assign	CKLINK.BIN	FOR025	Assign the Chemkin Linking file to Fortran unit 25.
$run	CKINTERP.EXE		Execute the Chemkin Interpreter.
$assign	SURFMECH.DAT	FOR015	Assign the user's surface reaction mechanism to Fortran unit 15. This is the input file for the Surface Chemkin Interpreter.
$assign	SKINTERP.OUT	FOR016	Assign the output file for printed output from the Surface Chemkin Interpreter.
$assign	SKTHERMO.DAT	FOR017	Assign the Surface Thermodynamic Database to Fortran unit 17. This could be the same database as used above for Chemkin.
$assign	SKLINK.BIN	FOR026	Assign the Surface Chemkin Linking file to Fortran unit 26.
$run	SKINTERP.EXE		Execute the Surface Chemkin Interpreter.
$for	SAMPLE.FOR		Compile the user's Fortran program.
$assign	SAMPLE.INP	FOR005	Assign a file containing any input required by the user's program to Fortran unit 5.
$assign	SAMPLE.OUT	FOR006	Assign a file to accept any printed output from the user's program to Fortran unit 6.
$link	SAMPLE.OBJ, CKLIB/LIB, SKLIB/LIB, LSODE		Link the user's program with the Chemkin Gas-Phase Subroutine Library, the Surface Chemkin Library, and LSODE, the stiff ordinary differential equation solver.
$run	SAMPLE		Execute the user's program.

2. Input to Chemkin Interpreter

```
ELEMENTS H N HE SI F
END
SPECIES
H2 H N2 N NH NH2 NNH N2H2 N2H3 N2H4 NH3 HE
HF F SIF4 SIF3 SIHF3 SIF3NH2
END

REACTIONS
H+H+M=H2+M                    0.100E+19   -1.000         0.000    ! D-L
  H2/0.0/
H+H+H2=H2+H2                  0.920E+17   -0.600         0.000
NH+N=N2+H                     0.300E+14    0.000         0.000    ! JAM
NH+H=N+H2                     0.100E+15    0.000         0.000    ! NH3 CST
NH2+H=NH+H2                   0.692E+14    0.000      3650.000
NH3+H=NH2+H2                  0.636E+06    2.390     10171.000    ! MICHAEL
NNH=N2+H                      0.100E+05    0.000         0.000    ! JAM
NNH+H=N2+H2                   0.100E+15    0.000         0.000    ! JAM
NNH+NH2=N2+NH3                0.500E+14    0.000         0.000    ! JAM
NNH+NH=N2+NH2                 0.500E+14    0.000         0.000    ! JAM
NH2+NH=N2H2+H                 0.500E+14    0.000         0.000    ! NH3 CST
NH+NH=N2+H+H                  0.254E+14    0.000         0.000    ! NH3 CST
NH2+N=N2+H+H                  0.720E+14    0.000         0.000    ! PG
N2H2+M=NNH+H+M                0.500E+17    0.000     50000.000    ! NH3 CST
    N2/2/ H2/2/
N2H2+H=NNH+H2                 0.500E+14    0.000      1000.000    ! NH3 CST
N2H2+NH=NNH+NH2               0.100E+14    0.000      1000.000    ! NH3 CST
N2H2+NH2=NH3+NNH              0.100E+14    0.000      1000.000    ! NH3 CST
NH2+NH2=N2H2+H2               0.500E+12    0.000         0.000    ! NH3 CST
NH3+M=NH2+H+M                 0.140E+17    0.000     90600.000    ! MSGK
N2H3+H=NH2+NH2                1.60E+12     0.000         0.000    ! MSGK
N2H3+M=N2H2+H+M               3.50E+16     0.000     46000.000    ! MSGK
N2H3+NH=NH2+N2H2              2.00E+13     0.000         0.000    ! MSGK
NH2+NH2+M=N2H4+M              3.00E+20    -1.000         0.000    ! MSGK
H+N2H4=H2+N2H3                1.30E+13     0.000      2500.000    ! MSGK
NH2+N2H4=NH3+N2H3             3.90E+12     0.000      1500.000    ! MSGK
NH+H+M=NH2+M                  2.00E+16    -0.500         0.000    ! MSGK
NH2+NH2=NH3+NH                5.00E+12     0.000     10000.000    ! MSGK
F+NH3=NH2+HF                  4.27E+11     0.500       800.000    ! KONDRATIEV
SIF4=SIF3+F                   3.00E+12     0.000    147170.000    ! PHO&MEC
H+SIF4=HF+SIF3                1.00E+13     0.000     50000.000    ! PHO&MEC
NH2+SIF4=SIF3NH2+F            1.00E+11     0.000     40950.000    ! EST.
NH3+SIF3=SIF3NH2+H            1.00E+11     0.000      5000.000    ! EST.
NH3+SIF3=SIHF3+NH2            1.00E+11     0.000     10000.000    ! PHO&MEC
END
```

3. Output from Chemkin Interpreter

```
CHEMKIN INTERPRETER OUTPUT: CHEMKIN-II Version 1.9 May 1990
                             DOUBLE PRECISION

                                --------------------
                                ELEMENTS     ATOMIC
                                CONSIDERED   WEIGHT
                                --------------------
                                 1. H        1.00797
                                 2. N        14.0067
                                 3. HE       4.00260
                                 4. SI       28.0860
                                 5. F        18.9984
                                --------------------
-------------------------------------------------------------------------------
                      C
                   P  H
                   H  A
                   A  R
SPECIES            S  G  MOLECULAR   TEMPERATURE    ELEMENT COUNT
CONSIDERED         E  E  WEIGHT      LOW     HIGH   H  N  HE SI F
-------------------------------------------------------------------------------
  1. H2            G  0    2.01594   300.0  5000.0  2  0  0  0  0
  2. H             G  0    1.00797   300.0  5000.0  1  0  0  0  0
  3. N2            G  0   28.01340   300.0  5000.0  0  2  0  0  0
  4. N             G  0   14.00670   300.0  5000.0  0  1  0  0  0
  5. NH            G  0   15.01467   300.0  5000.0  1  1  0  0  0
  6. NH2           G  0   16.02264   300.0  5000.0  2  1  0  0  0
  7. NNH           G  0   29.02137   250.0  4000.0  1  2  0  0  0
  8. N2H2          G  0   30.02934   300.0  5000.0  2  2  0  0  0
  9. N2H3          G  0   31.03731   300.0  5000.0  3  2  0  0  0
 10. N2H4          G  0   32.04528   300.0  5000.0  4  2  0  0  0
 11. NH3           G  0   17.03061   300.0  5000.0  3  1  0  0  0
 12. HE            G  0    4.00260   300.0  5000.0  0  0  1  0  0
 13. HF            G  0   20.00637   300.0  5000.0  1  0  0  0  1
 14. F             G  0   18.99840   300.0  5000.0  0  0  0  0  1
 15. SIF4          G  0  104.07960   300.0  2000.0  0  0  0  1  4
 16. SIF3          G  0   85.08120   300.0  3000.0  0  0  0  1  3
 17. SIHF3         G  0   86.08917   300.0  3000.0  1  0  0  1  3
 18. SIF3NH2       G  0  101.10384   300.0  3000.0  2  1  0  1  3
-------------------------------------------------------------------------------
```

```
                                                          (k = A T**b exp(-E/RT))
      REACTIONS CONSIDERED                                   A          b          E

   1. H+H+M=H2+M                                         1.00E+18    -1.0         0.0
         H2            Enhanced by     0.000E+00
   2. H+H+H2=H2+H2                                       9.20E+16    -0.6         0.0
   3. NH+N=N2+H                                          3.00E+13     0.0         0.0
   4. NH+H=N+H2                                          1.00E+14     0.0         0.0
   5. NH2+H=NH+H2                                        6.92E+13     0.0      3650.0
   6. NH3+H=NH2+H2                                       6.36E+05     2.4     10171.0
   7. NNH=N2+H                                           1.00E+04     0.0         0.0
   8. NNH+H=N2+H2                                        1.00E+14     0.0         0.0
   9. NNH+NH2=N2+NH3                                     5.00E+13     0.0         0.0
  10. NNH+NH=N2+NH2                                      5.00E+13     0.0         0.0
  11. NH2+NH=N2H2+H                                      5.00E+13     0.0         0.0
  12. NH+NH=N2+H+H                                       2.54E+13     0.0         0.0
  13. NH2+N=N2+H+H                                       7.20E+13     0.0         0.0
  14. N2H2+M=NNH+H+M                                     5.00E+16     0.0     50000.0
         N2            Enhanced by     2.000E+00
         H2            Enhanced by     2.000E+00
  15. N2H2+H=NNH+H2                                      5.00E+13     0.0      1000.0
  16. N2H2+NH=NNH+NH2                                    1.00E+13     0.0      1000.0
  17. N2H2+NH2=NH3+NNH                                   1.00E+13     0.0      1000.0
  18. NH2+NH2=N2H2+H2                                    5.00E+11     0.0         0.0
  19. NH3+M=NH2+H+M                                      1.40E+16     0.0     90600.0
  20. N2H3+H=NH2+NH2                                     1.60E+12     0.0         0.0
  21. N2H3+M=N2H2+H+M                                    3.50E+16     0.0     46000.0
  22. N2H3+NH=NH2+N2H2                                   2.00E+13     0.0         0.0
  23. NH2+NH2+M=N2H4+M                                   3.00E+20    -1.0         0.0
  24. H+N2H4=H2+N2H3                                     1.30E+13     0.0      2500.0
  25. NH2+N2H4=NH3+N2H3                                  3.90E+12     0.0      1500.0
  26. NH+H+M=NH2+M                                       2.00E+16    -0.5         0.0
  27. NH2+NH2=NH3+NH                                     5.00E+12     0.0     10000.0
  28. F+NH3=NH2+HF                                       4.27E+11     0.5       800.0
  29. SIF4=SIF3+F                                        3.00E+12     0.0    147170.0
  30. H+SIF4=HF+SIF3                                     1.00E+13     0.0     50000.0
  31. NH2+SIF4=SIF3NH2+F                                 1.00E+11     0.0     40950.0
  32. NH3+SIF3=SIF3NH2+H                                 1.00E+11     0.0      5000.0
  33. NH3+SIF3=SIHF3+NH2                                 1.00E+11     0.0     10000.0

  NOTE:  A units mole-cm-sec-K, E units cal/mole


 NO ERRORS FOUND ON INPUT...CHEMKIN LINKING FILE WRITTEN.

 WORKING SPACE REQUIREMENTS ARE
    INTEGER:      679
    REAL:         695
    CHARACTER:     23
```

4. Input to Surface Chemkin Interpreter

```
SITE/SI3N4/    SDEN/4.17e-9/
   NHSIF(S)/2/
   SIF3NH2(S)/2/
   SIF2NH(S)/2/
   NH2SIFNH(S)/2/
   NHSIFNHSIFNH(S)/4/
   NHNH2(S)/2/
END
BULK SI(D)/2.066/
BULK N(D) /1.374/
END
THERMO ALL
  300.      600.     1685.
NHSIF(S)           J 3/67N   1H   1SI  1F   1S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
NHNH2(S)           J 3/67N   2H   3SI  0F   0S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
SIF3NH2(S)         J 3/67N   1H   2SI  1F   3S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
SIF2NH(S)          J 3/67N   1H   1SI  1F   2S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
NH2SIFNH(S)        J 3/67N   2H   3SI  1F   1S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
NHSIFNHSIFNH(S)    J 3/67N   3H   3SI  2F   2S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
SI(D)              J 3/67SI  100  000  000  0S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
N(D)               J 3/67N   100  000  000  0S   300.000  1685.000          1
 0.24753989E 01 0.88112187E-03-0.20939481E-06 0.42757187E-11 0.16006564E-13    2
-0.81255620E 03-0.12188747E 02 0.84197538E 00 0.83710416E-02-0.13077030E-04    3
 0.97593603E-08-0.27279380E-11-0.52486288E 03-0.45272678E 01                   4
END

REACTIONS
 NH3             + NHSIF(S)  => NHNH2(S)   + SI(D) + HF    6.3E08 0.5 0.0
 SIF4            + NHNH2(S)  => SIF3NH2(S) + N(D)  + HF    2.58E08 0.5 0.0
 SIF3NH2(S)                  => SIF2NH(S)          + HF    1.0E05 0.0 0.0
 NH3             + SIF2NH(S) => NH2SIFNH(S)        + HF    6.3E08 0.5 0.0
 NH2SIFNH(S)     + SIF2NH(S) => NHSIFNHSIFNH(S)    + HF    1.0E15 0.0 0.0
 NHSIFNHSIFNH(S) + SIF2NH(S) => 3NHSIF(S)  + N(D)  + HF    1.0E15 0.0 0.0
END
```

5. Output from Surface Chemkin Interpreter

```
SURFACE INTERPRETER OUTPUT: CHEMKIN-II Version 3.7  July 1990
                            DOUBLE PRECISION

 CKLIB:  Chemical Kinetics Library
         CHEMKIN-II Version 2.2, June 1990
         DOUBLE PRECISION
---------------------------------------------------------------------------
SPECIES                 MOLECULAR                           ELEMENT COUNT
CONSIDERED              WEIGHT       Density    Nsites       H  N  HE SI F
---------------------------------------------------------------------------

Gas phase species:
  1. H2                   2.01594                            2  0  0  0  0
  2. H                    1.00797                            1  0  0  0  0
  3. N2                  28.01340                            0  2  0  0  0
  4. N                   14.00670                            0  1  0  0  0
  5. NH                  15.01467                            1  1  0  0  0
  6. NH2                 16.02264                            2  1  0  0  0
  7. NNH                 29.02137                            1  2  0  0  0
  8. N2H2                30.02934                            2  2  0  0  0
  9. N2H3                31.03731                            3  2  0  0  0
 10. N2H4                32.04528                            4  2  0  0  0
 11. NH3                 17.03061                            3  1  0  0  0
 12. HE                   4.00260                            0  0  1  0  0
 13. HF                  20.00637                            1  0  0  0  1
 14. F                   18.99840                            0  0  0  0  1
 15. SIF4               104.07960                            0  0  0  1  4
 16. SIF3                85.08120                            0  0  0  1  3
 17. SIHF3               86.08917                            1  0  0  1  3
 18. SIF3NH2            101.10384                            2  1  0  1  3

 SITE: SI3N4                        0.417E-08 moles/cm**2
  19. NHSIF(S)           62.09907                       2    1  1  0  1  1
  20. SIF3NH2(S)        101.10384                       2    2  1  0  1  3
  21. SIF2NH(S)          81.09747                       2    1  1  0  1  2
  22. NH2SIFNH(S)        78.12171                       2    3  2  0  1  1
  23. NHSIFNHSIFNH(S)   139.21281                       4    3  3  0  2  2
  24. NHNH2(S)           31.03731                       2    3  2  0  0  0

 BULK: BULK1
  25. SI(D)              28.08600 0.207E+01 g/cm**3          0  0  0  1  0

 BULK: BULK2
  26. N(D)               14.00670 0.137E+01 g/cm**3          0  1  0  0  0
---------------------------------------------------------------------------
```

```
                                                        (k = A T**b exp(-E/RT))
    SURFACE REACTIONS CONSIDERED                           A        b        E

 1. NH3+NHSIF(S)=>NHNH2(S)+SI(D)+HF                    6.30E+08    0.5      0.0
 2. SIF4+NHNH2(S)=>SIF3NH2(S)+N(D)+HF                  2.58E+08    0.5      0.0
 3. SIF3NH2(S)=>SIF2NH(S)+HF                           1.00E+05    0.0      0.0
 4. NH3+SIF2NH(S)=>NH2SIFNH(S)+HF                      6.30E+08    0.5      0.0
 5. NH2SIFNH(S)+SIF2NH(S)                              1.00E+15    0.0      0.0
    =>NHSIFNHSIFNH(S)+HF
 6. NHSIFNHSIFNH(S)+SIF2NH(S)                          1.00E+15    0.0      0.0
    =>3NHSIF(S)+N(D)+HF

NOTE:   A units mole, E units cal/mole


NO ERRORS FOUND ON INPUT...SURFACE LINKING FILE WRITTEN.

WORKING SPACE REQUIREMENTS ARE
   INTEGER:      432
   REAL:         598
   CHARACTER:     35
```

6. Sample Problem Input

```
2.63e-3  1713
SIF4              0.14286
NH3               0.85714
NHSIF(S)          6.251E-2
NHNH2(S)          0.91587
SIF3NH2(S)        2.354E-4
SIF2NH(S)         2.0837E-2
NH2SIFNH(S)       1.806E-4
NHSIFNHSIFNH(S)   3.6127E-4
N(D)              1.0
SI(D)             1.0
END
6.0
5.0E-2   5.0E-3
```

7. User's Fortran Code

```
      PROGRAM SKSAMP
C
C       Gas-phase and surface reaction in a constant volume isothermal
C       container with fixed surface area.
C
C*****double precision
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER(I-N)
C*****END double precision
C*****single precision
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END single precision
C
      PARAMETER (LENIWK=4000, LENRWK=4000, LENCWK=500, LINCK=25,
     1           LENISK=4000, LENRSK=4000, LENCSK=500, LINSK=26, NK=5,
     2           NLMAX=55, LRW=1500, LIW=100, LIN=5, LOUT=6, KMAX=50)
C
      DIMENSION Z(KMAX), ELWRK(LRW), IELWRK(LIW)
C
      COMMON /PARAM/ ICKWRK(4000), RCKWRK(4000), KKGAS, T, AVRAT, RU,
     1               WT(50), WDOT(50), SDOT(50), X(50), ACT(50),
     2               ISKWRK(4000), RSKWRK(4000), KKTOT, NFSURF, NLSURF,
     3               KKSURF, NFBULK, NLBULK, KKBULK, KKPHAS(20),
     4               KFIRST(20), KLAST(20), SDEN0(20), SITDOT(20),
     5               KCOV(50), RCOV(50)
      CHARACTER*16 CCKWRK(LENCWK), CSKWRK(LENCSK), KSYM(KMAX),
     1             PNAM(20)
      CHARACTER*80 LINE
C
      LOGICAL KERR, IERR
      EXTERNAL FUN
      DATA KERR/.FALSE./, KSYM/KMAX*' '/
C
      DO 10 K = 1, KMAX
         X(K) = 0.0
   10 CONTINUE
C
C     Open the CHEMKIN LINK file
C
C*****vax
      OPEN (LINCK, STATUS='OLD', FORM='UNFORMATTED')
C*****END vax
C*****unix
C      OPEN (LINCK, FORM='UNFORMATTED', FILE='cklink')
C*****END unix
C
C     Initialize CHEMKIN
C
      CALL CKINIT (LENIWK, LENRWK, LENCWK, LINCK, LOUT, ICKWRK,
     1             RCKWRK, CCKWRK)
      CALL CKINDX (ICKWRK, RCKWRK, MM, KKGAS, II, NFIT)
```

```
C
C     Open the SURFACE LINK file
C
C*****vax
      OPEN (LINSK, STATUS='OLD', FORM='UNFORMATTED')
C*****END vax
C*****unix
C      OPEN (LINSK, FORM='UNFORMATTED', FILE='sklink')
C*****END unix
C
C     Initialize SURFCHEM
C
      CALL SKINIT (LENISK, LENRSK, LENCSK, LINSK, LOUT, ISKWRK,
     1             RSKWRK, CSKWRK)
      CALL SKINDX (NELEM, KKGAS, KKSUR, KKBULK, KKTOT, NNPHAS, NNSURF,
     1             NFSURF, NLSURF, NNBULK, NFBULK, NLBULK, IISUR)
      CALL SKPKK  (ISKWRK, KKPHAS, KFIRST, KLAST)
      CALL SKSDEN (RSKWRK, SDEN0)
      CALL SKCOV  (ISKWRK, KCOV)
      DO 30 K = KFIRST(NFSURF), KLAST(NLSURF)
C*****double precision
         RCOV(K) = DBLE(KCOV(K))
C*****END double precision
C*****single precision
C         RCOV(K) = REAL(KCOV(K))
C*****END single precision
   30 CONTINUE
C
      IF (KKTOT .GT. KMAX) THEN
         WRITE (LOUT, *)
     1   ' Species dimension too small...must be at least ',KKTOT
         STOP
      ENDIF
C
      CALL SKSYMS (CSKWRK, LOUT, KSYM, IERR)
      IF (IERR) KERR = .TRUE.
      CALL SKSYMP (CSKWRK, LOUT, PNAM, IERR)
      IF (IERR) KERR = .TRUE.
      CALL SKWT   (RSKWRK, WT)
      CALL SKRP   (ISKWRK, RSKWRK, RU, RUC, PATM)
C
C     Pressure and temperature
C
      WRITE (LOUT, '(/A)')
     1                 ' INPUT INITIAL PRESSURE(ATM) AND TEMPERATURE(K)'
      READ  (LIN,    *) PA, T
      WRITE (LOUT,7105) PA, T
      P = PA*PATM
```

```
C
C     Initial non-zero moles
C
   40 CONTINUE
      LINE = ' '
      WRITE (LOUT, '(/A)') ' INPUT INITIAL ACTIVITY OF NEXT SPECIES'
      READ  (LIN,  '(A)', END=45)   LINE
      WRITE (LOUT, '(X,A)') LINE
      ILEN = INDEX (LINE, '!')
      IF (ILEN .EQ. 1) GO TO 40
C
      ILEN = ILEN - 1
      IF (ILEN .LE. 0) ILEN = LEN(LINE)
      IF (INDEX(LINE(:ILEN), 'END') .EQ. 0) THEN
         IF (LINE(:ILEN) .NE. ' ') THEN
            CALL SKSNUM (LINE(:ILEN), 1, LOUT, KSYM, KKTOT, PNAM,
     1                   NNPHAS, KKPHAS, KNUM, NKF, NVAL, VAL, IERR)
            IF (IERR) THEN
               WRITE (LOUT,*) ' Error reading moles...'
               KERR = .TRUE.
            ELSE
               X(KNUM) = VAL
            ENDIF
         ENDIF
         GO TO 40
      ENDIF
C
   45 CONTINUE
C
C     Surface area to volume ratio
C
      WRITE (LOUT, '(/A)') ' INPUT SURFACE AREA TO VOLUME RATIO'
      READ  (LIN,     *) AVRAT
      WRITE (LOUT,7105) AVRAT
C
C     Final time and print interval
C
      WRITE (LOUT, '(/A)') ' INPUT FINAL TIME AND DT'
      READ  (LIN,     *) T2, DT
      WRITE (LOUT,7105) T2, DT
C
      IF (KERR) STOP
C
C     Normalize the gas-phase mole fractions
C
      XTOT = 0.0
      DO 50 K = 1, KKGAS
         XTOT = XTOT + X(K)
   50 CONTINUE
```

```
      IF (XTOT .EQ. 0.0) THEN
         WRITE (LOUT, *) ' ERROR... NO GAS-PHASE SPECIES WERE INPUT'
         STOP
      ENDIF
      DO 55 K = 1, KKGAS
         X(K) = X(K) / XTOT
   55 CONTINUE
C
C        Normalize the site fractions
C
      DO 80 N = NFSURF, NLSURF
         XTOT = 0.0
         DO 70 K = KFIRST(N), KLAST(N)
            XTOT = XTOT + X(K)
   70    CONTINUE
         IF (XTOT .EQ. 0.0) THEN
            WRITE (LOUT, *)
     1     ' ERROR... NO SURFACE SPECIES WERE INPUT FOR PHASE # ', N
            STOP
         ENDIF
         DO 75 K = KFIRST(N), KLAST(N)
            X(K) = X(K) / XTOT
   75    CONTINUE
   80 CONTINUE
C
C        Normalize the bulk activities
C
      DO 90 N = NFBULK, NLBULK
         XTOT = 0.0
         DO 84 K = KFIRST(N), KLAST(N)
            XTOT = XTOT + X(K)
   84    CONTINUE
         IF (XTOT .EQ. 0.0) THEN
            WRITE (LOUT, *)
     1     ' ERROR... NO BULK SPECIES WERE INPUT FOR PHASE # ', N
            STOP
         ENDIF
         DO 85 K = KFIRST(N), KLAST(N)
            X(K) = X(K) / XTOT
   85    CONTINUE
   90 CONTINUE
C
C     Initial conditions and gas-phase mass fractions
C
      TT1  = 0.0
      CALL CKXTY (X, ICKWRK, RCKWRK, Z)
C     Initial surface site fractions
      DO 110 K = KFIRST(NFSURF), KLAST(NLSURF)
         Z(K) = X(K)
  110 CONTINUE
```

```
C     Initial bulk deposit amounts
      DO 120 K = KFIRST(NFBULK), KLAST(NLBULK)
         Z(K) = 0.0
  120 CONTINUE
C     Initial gas-phase mass density
      CALL CKRHOY (P, T, Z, ICKWRK, RCKWRK, Z(KKTOT+1) )
C     Initial surface site densities
      DO 130 N = NFSURF, NLSURF
         Z(KKTOT+1+N-NFSURF+1) = SDENO(N)
  130 CONTINUE
C
C     Integration control parameters for LSODE
C
      TT2    = TT1
      NEQ    = KKTOT + 1 + NNSURF
      MF     = 22
      ITOL   = 1
      IOPT   = 0
      RTOL   = 1.0E-6
      ITASK  = 1
      ATOL   = 1.0E-12
      ISTATE= 1
      NLINES=NLMAX + 1
C
C     Integration loop
C
  250 CONTINUE
C
C           Print the solution
C
      CALL CKPY  (Z(KKTOT+1), T, Z(1), ICKWRK, RCKWRK, P)
      WRITE (LOUT,*) ' '
      WRITE (LOUT,*) ' TIME = ', TT2
      WRITE (LOUT, 7100) P, T, Z(KKTOT+1)
      WRITE (LOUT, *) ' GAS-PHASE MOLE FRACTIONS'
      CALL CKYTX (Z, ICKWRK, RCKWRK, X)
      CALL PRT1  (KKGAS, KSYM, LOUT, X)
      DO 190 N = NFSURF, NLSURF
         WRITE (LOUT, *) ' SURFACE SITE FRACTIONS ON PHASE (SITE) ', N
         CALL PRT1 (KKPHAS(N), KSYM(KFIRST(N)), LOUT, Z(KFIRST(N)))
         SUM = 0.0
         DO 185 K = KFIRST(NFSURF), KLAST(NLSURF)
            SUM = SUM + Z(K)
  185    CONTINUE
         WRITE (LOUT,*)'  SUM OF SURFACE SITE FRACTIONS', SUM
         WRITE (LOUT,*)'  SURFACE SITE DENSITY  ', Z(KKTOT+1+N-NFSURF+1)
  190 CONTINUE
      DO 195 N = NFBULK, NLBULK
         WRITE (LOUT, *) ' BULK DEPOSITION (GM/CM**2) IN PHASE ', N
         CALL PRT1 (KKPHAS(N), KSYM(KFIRST(N)), LOUT, Z(KFIRST(N)))
  195 CONTINUE
```

```
C
      IF (TT2 .GE. T2) STOP
      TT2 = MIN(TT2 + DT, T2)
C
C     Call the differential equation solver
C
  350 CONTINUE
      CALL LSODE (FUN, NEQ, Z, TT1, TT2, ITOL, RTOL, ATOL, ITASK,
     1            ISTATE, IOPT, ELWRK, LRW, IELWRK, LIW, JAC, MF)
C
      IF (ISTATE .LE. -1) THEN
         IF (ISTATE .EQ. -1) THEN
            ISTATE = 2
            GO TO 350
         ELSE
            WRITE (LOUT,*) ' ISTATE=',ISTATE
            STOP
         ENDIF
      ENDIF
      GO TO 250
C
C         FORMATS
C
 7003 FORMAT (1H1)
 7100 FORMAT (1H , ' GAS-PHASE STATE', /,
     1   '  P = ', 1PE12.4, ' T = ', 1PE12.4, ' DENSITY = ', 1PE12.4)
 7105 FORMAT (12E11.3)
 7110 FORMAT (26X, 5(1X,A10))
 7115 FORMAT (22X, 10E11.3)
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE FUN (NEQ, TIME, Z, ZP)
C
C*****double precision
      IMPLICIT DOUBLE PRECISION(A-H,O-Z), INTEGER(I-N)
C*****END double precision
C*****single precision
C      IMPLICIT REAL (A-H,O-Z), INTEGER(I-N)
C*****END single precision
C
      DIMENSION Z(NEQ), ZP(NEQ)
      COMMON /PARAM/ ICKWRK(4000), RCKWRK(4000), KKGAS, T, AVRAT, RU,
     1               WT(50), WDOT(50), SDOT(50), X(50), ACT(50),
     2               ISKWRK(4000), RSKWRK(4000), KKTOT, NFSURF, NLSURF,
     3               KKSURF, NFBULK, NLBULK, KKBULK, KKPHAS(20),
     4               KFIRST(20), KLAST(20), SDEN0(20), SITDOT(20),
     5               KCOV(50), RCOV(50)
```

```
C
C     Variables in Z are:  Z(K) = Y(K), K=1,KKGAS
C                          Z(K) = SURFACE SITE FRACTIONS,
C                                 K=KFIRST(NFSURF), KLAST(NLSURF)
C                          Z(K) = BULK SPECIES MASS,
C                                 K=KFIRST(NFBULK), KLAST(NLBULK)
C                          Z(K) = GAS-PHASE MASS DENSITY, K=KKTOT+1
C                          Z(K) = SURFACE SITE MOLAR DENSITIES,
C                                 K=KKTOT+2, KKTOT+1+NNSURF
C
C        Call CHEMKIN and SURFACE CHEMKIN subroutines
C
      CALL CKPY  (Z(KKTOT+1), T, Z(1), ICKWRK, RCKWRK, P)
      CALL CKWYP (P, T, Z(1), ICKWRK, RCKWRK, WDOT)
      CALL CKYTX (Z, ICKWRK, RCKWRK, ACT)
C
      DO 150 K = KFIRST(NFBULK), KLAST(NLBULK)
         ACT(K) = X(K)
150   CONTINUE
      DO 175 N = NFSURF, NLSURF
         SDEN0(N) = Z(KKTOT+1+N-NFSURF+1)
         DO 100 K = KFIRST(N), KLAST(N)
            ACT(K) = Z(K)
100      CONTINUE
175   CONTINUE
C
      CALL SKRAT (P, T, ACT, SDEN0, ISKWRK, RSKWRK, SDOT, SITDOT)
C
C        Form mass density equation
C
      SUM = 0.0
      DO 200 K = 1, KKGAS
         SUM = SUM + AVRAT * SDOT(K) * WT(K)
 200  CONTINUE
      ZP(KKTOT+1) = SUM
C
C        Form the gas-phase mass conservation equation
C
      DO 300 K = 1, KKGAS
         ZP(K) = ( - Z(K) * ZP(KKTOT+1) + WDOT(K) * WT(K)
     1             + AVRAT * SDOT(K) * WT(K) ) / Z(KKTOT+1)
 300  CONTINUE
C
C        Form the surface mass equations
C
      DO 400 N = NFSURF, NLSURF
         DO 400 K = KFIRST(N), KLAST(N)
             ZP(K) = (SDOT(K)*RCOV(K) - Z(K) * SITDOT(N)) / SDEN0(N)
  400 CONTINUE
```

```
C
C        Form the bulk mass equations
C
      DO 500 K = KFIRST(NFBULK), KLAST(NLBULK)
         ZP(K) = SDOT(K) * WT(K) * AVRAT
  500 CONTINUE
C
C        Form the surface site number-density equations
C
      DO 575 N = NFSURF, NLSURF
         ZP(KKTOT+1+N-NFSURF+1) = SITDOT(N)
  575 CONTINUE
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE PRT1 (KK, KSYM, LOUT, X)
C
C*****double precision
      IMPLICIT DOUBLE PRECISION (A-H, O-Z), INTEGER (I-N)
C*****END double precision
C*****single precision
C      IMPLICIT REAL (A-H,O-Z), INTEGER (I-N)
C*****END single precision
C
      DIMENSION X(KK)
      CHARACTER KSYM(KK)*(*)
C
      DO 10 K = 1, KK, 3
         WRITE (LOUT, 6010) (KSYM(L), X(L), L=K, MIN(K+2, KK))
   10 CONTINUE
 6010 FORMAT (3X, 3(A12,'=', 1PE10.3, 4X))
C
      RETURN
      END
```

8. Output from Fortran Code

```
 CKLIB:  Chemical Kinetics Library
         CHEMKIN-II Version 2.2, June 1990
         DOUBLE PRECISION

 SKLIB:  Surface kinetics library
         CHEMKIN-II Version 3.7, July 1990
         DOUBLE PRECISION

INPUT INITIAL PRESSURE(ATM) AND TEMPERATURE(K)
 0.263E-02  0.171E+04

INPUT INITIAL ACTIVITY OF NEXT SPECIES
SIF4             0.14286

INPUT INITIAL ACTIVITY OF NEXT SPECIES
NH3              0.85714

INPUT INITIAL ACTIVITY OF NEXT SPECIES
NHSIF(S)         6.251E-2

INPUT INITIAL ACTIVITY OF NEXT SPECIES
NHNH2(S)         0.91587

INPUT INITIAL ACTIVITY OF NEXT SPECIES
SIF3NH2(S)       2.354E-4

INPUT INITIAL ACTIVITY OF NEXT SPECIES
SIF2NH(S)        2.0837E-2

INPUT INITIAL ACTIVITY OF NEXT SPECIES
NH2SIFNH(S)      1.806E-4

INPUT INITIAL ACTIVITY OF NEXT SPECIES
NHSIFNHSIFNH(S)  3.6127E-4

INPUT INITIAL ACTIVITY OF NEXT SPECIES
N(D)             1.0

INPUT INITIAL ACTIVITY OF NEXT SPECIES
SI(D)            1.0

INPUT INITIAL ACTIVITY OF NEXT SPECIES
END

INPUT SURFACE AREA TO VOLUME RATIO
 0.600E+01

INPUT FINAL TIME AND DT
 0.500E-01  0.500E-02
```

```
TIME =    0.000000000000000E+000
GAS-PHASE STATE
P =    2.6648E+03 T =    1.7130E+03 DENSITY =    5.5136E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 0.000E+00    H            = 0.000E+00    N2           = 0.000E+00
 N            = 0.000E+00    NH           = 0.000E+00    NH2          = 0.000E+00
 NNH          = 0.000E+00    N2H2         = 0.000E+00    N2H3         = 0.000E+00
 N2H4         = 0.000E+00    NH3          = 8.571E-01    HE           = 0.000E+00
 HF           = 0.000E+00    F            = 0.000E+00    SIF4         = 1.429E-01
 SIF3         = 0.000E+00    SIHF3        = 0.000E+00    SIF3NH2      = 0.000E+00
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 6.251E-02    SIF3NH2(S)   = 2.354E-04    SIF2NH(S)    = 2.084E-02
 NH2SIFNH(S) = 1.806E-04    NHSIFNHSIFNH= 3.613E-04    NHNH2(S)     = 9.159E-01
 SUM OF SURFACE SITE FRACTIONS    1.00000000000000
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE             3
 SI(D)        = 0.000E+00
BULK DEPOSITION (GM/CM**2) IN PHASE             4
 N(D)         = 0.000E+00

TIME =    5.000000000000000E-003
GAS-PHASE STATE
P =    2.9649E+03 T =    1.7130E+03 DENSITY =    4.9573E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 2.742E-06    H            = 2.248E-08    N2           = 1.865E-10
 N            = 2.259E-14    NH           = 3.433E-10    NH2          = 5.504E-06
 NNH          = 1.425E-11    N2H2         = 3.457E-10    N2H3         = 1.258E-11
 N2H4         = 5.814E-13    NH3          = 6.799E-01    HE           = 0.000E+00
 HF           = 2.507E-01    F            = 8.650E-11    SIF4         = 6.933E-02
 SIF3         = 9.167E-11    SIHF3        = 2.789E-11    SIF3NH2      = 1.359E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 5.199E-02    SIF3NH2(S)   = 1.440E-04    SIF2NH(S)    = 1.420E-02
 NH2SIFNH(S) = 1.771E-04    NHSIFNHSIFNH= 3.545E-04    NHNH2(S)     = 9.331E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999998
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE             3
 SI(D)        = 4.060E-08
BULK DEPOSITION (GM/CM**2) IN PHASE             4
 N(D)         = 2.336E-08
```

```
TIME =   1.000000000000000E-002
GAS-PHASE STATE
P =   3.1321E+03 T =   1.7130E+03 DENSITY =   4.6601E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 5.142E-06    H            = 2.258E-08    N2           = 2.274E-09
 N            = 2.461E-13    NH           = 1.817E-09    NH2          = 1.028E-05
 NNH          = 8.046E-11    N2H2         = 1.792E-09    N2H3         = 5.380E-11
 N2H4         = 2.276E-12    NH3          = 5.888E-01    HE           = 0.000E+00
 HF           = 3.763E-01    F            = 2.798E-10    SIF4         = 3.492E-02
 SIF3         = 6.292E-11    SIHF3        = 5.372E-11    SIF3NH2      = 2.712E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 3.509E-02    SIF3NH2(S)   = 7.847E-05    SIF2NH(S)    = 8.636E-03
 NH2SIFNH(S) = 1.621E-04    NHSIFNHSIFNH= 3.244E-04    NHNH2(S)     = 9.557E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE            3
 SI(D)        = 6.750E-08
BULK DEPOSITION (GM/CM**2) IN PHASE            4
 N(D)         = 3.631E-08

TIME =   1.500000000000000E-002
GAS-PHASE STATE
P =   3.2231E+03 T =   1.7130E+03 DENSITY =   4.5026E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 7.410E-06    H            = 2.275E-08    N2           = 8.971E-09
 N            = 9.706E-13    NH           = 4.713E-09    NH2          = 1.476E-05
 NNH          = 2.013E-10    N2H2         = 4.312E-09    N2H3         = 1.183E-10
 N2H4         = 4.917E-12    NH3          = 5.406E-01    HE           = 0.000E+00
 HF           = 4.416E-01    F            = 5.134E-10    SIF4         = 1.781E-02
 SIF3         = 3.725E-11    SIHF3        = 6.840E-11    SIF3NH2      = 3.544E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 2.186E-02    SIF3NH2(S)   = 4.193E-05    SIF2NH(S)    = 4.975E-03
 NH2SIFNH(S) = 1.531E-04    NHSIFNHSIFNH= 3.065E-04    NHNH2(S)     = 9.727E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE            3
 SI(D)        = 8.371E-08
BULK DEPOSITION (GM/CM**2) IN PHASE            4
 N(D)         = 4.333E-08
```

```
TIME =   2.000000000000000E-002
GAS-PHASE STATE
P =   3.2718E+03 T =   1.7130E+03 DENSITY =   4.4204E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 9.647E-06    H            = 2.301E-08    N2           = 2.296E-08
 N            = 2.482E-12    NH           = 9.072E-09    NH2          = 1.913E-05
 NNH          = 3.744E-10    N2H2         = 7.861E-09    N2H3         = 2.047E-10
 N2H4         = 8.452E-12    NH3          = 5.148E-01    HE           = 0.000E+00
 HF           = 4.761E-01    F            = 7.532E-10    SIF4         = 9.128E-03
 SIF3         = 2.093E-11    SIHF3        = 7.636E-11    SIF3NH2      = 4.028E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 1.287E-02    SIF3NH2(S)   = 2.206E-05    SIF2NH(S)    = 2.745E-03
 NH2SIFNH(S) = 1.480E-04    NHSIFNHSIFNH= 2.963E-04    NHNH2(S)     = 9.839E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 9.310E-08
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 4.708E-08

TIME =   2.500000000000000E-002
GAS-PHASE STATE
P =   3.2974E+03 T =   1.7130E+03 DENSITY =   4.3779E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 1.190E-05    H            = 2.334E-08    N2           = 4.691E-08
 N            = 4.967E-12    NH           = 1.482E-08    NH2          = 2.345E-05
 NNH          = 6.003E-10    N2H2         = 1.247E-08    N2H3         = 3.133E-10
 N2H4         = 1.287E-11    NH3          = 5.010E-01    HE           = 0.000E+00
 HF           = 4.943E-01    F            = 9.855E-10    SIF4         = 4.683E-03
 SIF3         = 1.140E-11    SIHF3        = 8.062E-11    SIF3NH2      = 4.304E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 7.255E-03    SIF3NH2(S)   = 1.149E-05    SIF2NH(S)    = 1.470E-03
 NH2SIFNH(S) = 1.452E-04    NHSIFNHSIFNH= 2.907E-04    NHNH2(S)     = 9.908E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 9.837E-08
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 4.905E-08
```

```
TIME =   3.000000000000000E-002
GAS-PHASE STATE
P =   3.3107E+03 T =   1.7130E+03 DENSITY =   4.3561E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 1.419E-05    H            = 2.375E-08    N2           = 8.356E-08
 N            = 8.484E-12    NH           = 2.182E-08    NH2          = 2.777E-05
 NNH          = 8.806E-10    N2H2         = 1.817E-08    N2H3         = 4.444E-10
 N2H4         = 1.818E-11    NH3          = 4.937E-01    HE           = 0.000E+00
 HF           = 5.039E-01    F            = 1.207E-09    SIF4         = 2.403E-03
 SIF3         = 6.072E-12    SIHF3        = 8.287E-11    SIF3NH2      = 4.459E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 3.960E-03    SIF3NH2(S)   = 5.941E-06    SIF2NH(S)    = 7.729E-04
 NH2SIFNH(S) = 1.437E-04    NHSIFNHSIFNH= 2.876E-04    NHNH2(S)     = 9.948E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE            3
 SI(D)        = 1.012E-07
BULK DEPOSITION (GM/CM**2) IN PHASE            4
 N(D)         = 5.007E-08

TIME =   3.500000000000000E-002
GAS-PHASE STATE
P =   3.3176E+03 T =   1.7130E+03 DENSITY =   4.3449E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 1.653E-05    H            = 2.424E-08    N2           = 1.357E-07
 N            = 1.297E-11    NH           = 2.994E-08    NH2          = 3.210E-05
 NNH          = 1.217E-09    N2H2         = 2.500E-08    N2H3         = 5.982E-10
 N2H4         = 2.439E-11    NH3          = 4.898E-01    HE           = 0.000E+00
 HF           = 5.089E-01    F            = 1.420E-09    SIF4         = 1.233E-03
 SIF3         = 3.190E-12    SIHF3        = 8.405E-11    SIF3NH2      = 4.544E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 2.112E-03    SIF3NH2(S)   = 3.061E-06    SIF2NH(S)    = 4.018E-04
 NH2SIFNH(S) = 1.428E-04    NHSIFNHSIFNH= 2.860E-04    NHNH2(S)     = 9.971E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE            3
 SI(D)        = 1.028E-07
BULK DEPOSITION (GM/CM**2) IN PHASE            4
 N(D)         = 5.060E-08
```

```
TIME =    4.000000000000000E-002
GAS-PHASE STATE
P =    3.3212E+03 T =    1.7130E+03 DENSITY =    4.3392E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 1.893E-05    H            = 2.482E-08    N2           = 2.063E-07
 N            = 1.829E-11    NH           = 3.905E-08    NH2          = 3.642E-05
 NNH          = 1.611E-09    N2H2         = 3.300E-08    N2H3         = 7.750E-10
 N2H4         = 3.149E-11    NH3          = 4.878E-01    HE           = 0.000E+00
 HF           = 5.115E-01    F            = 1.627E-09    SIF4         = 6.324E-04
 SIF3         = 1.661E-12    SIHF3        = 8.466E-11    SIF3NH2      = 4.591E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 1.110E-03    SIF3NH2(S)   = 1.574E-06    SIF2NH(S)    = 2.076E-04
 NH2SIFNH(S) = 1.424E-04    NHSIFNHSIFNH= 2.851E-04    NHNH2(S)     = 9.983E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 1.036E-07
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 5.087E-08

TIME =    4.500000000000000E-002
GAS-PHASE STATE
P =    3.3230E+03 T =    1.7130E+03 DENSITY =    4.3363E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 2.140E-05    H            = 2.549E-08    N2           = 2.982E-07
 N            = 2.427E-11    NH           = 4.906E-08    NH2          = 4.075E-05
 NNH          = 2.063E-09    N2H2         = 4.217E-08    N2H3         = 9.747E-10
 N2H4         = 3.948E-11    NH3          = 4.868E-01    HE           = 0.000E+00
 HF           = 5.128E-01    F            = 1.829E-09    SIF4         = 3.244E-04
 SIF3         = 8.617E-13    SIHF3        = 8.497E-11    SIF3NH2      = 4.617E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)            2
 NHSIF(S)     = 5.774E-04    SIF3NH2(S)   = 8.084E-07    SIF2NH(S)    = 1.069E-04
 NH2SIFNH(S) = 1.422E-04    NHSIFNHSIFNH= 2.846E-04    NHNH2(S)     = 9.989E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 1.040E-07
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 5.101E-08
```

```
TIME =   5.000000000000000E-002
GAS-PHASE STATE
P =   3.3240E+03 T =   1.7130E+03 DENSITY =   4.3348E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 2.394E-05    H           = 2.624E-08    N2           = 4.144E-07
 N            = 3.073E-11    NH          = 5.987E-08    NH2          = 4.508E-05
 NNH          = 2.575E-09    N2H2        = 5.253E-08    N2H3         = 1.197E-09
 N2H4         = 4.838E-11    NH3         = 4.862E-01    HE           = 0.000E+00
 HF           = 5.135E-01    F           = 2.028E-09    SIF4         = 1.664E-04
 SIF3         = 4.471E-13    SIHF3       = 8.513E-11    SIF3NH2      = 4.631E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 2.986E-04    SIF3NH2(S)  = 4.149E-07    SIF2NH(S)    = 5.493E-05
 NH2SIFNH(S) = 1.421E-04    NHSIFNHSIFNH= 2.844E-04    NHNH2(S)     = 9.992E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 1.043E-07
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 5.108E-08
```

```
TIME =   5.000000000000000E-002
GAS-PHASE STATE
P =   3.3240E+03 T =   1.7130E+03 DENSITY =   4.3348E-07
GAS-PHASE MOLE FRACTIONS
 H2           = 2.394E-05    H           = 2.624E-08    N2           = 4.144E-07
 N            = 3.073E-11    NH          = 5.987E-08    NH2          = 4.508E-05
 NNH          = 2.575E-09    N2H2        = 5.253E-08    N2H3         = 1.197E-09
 N2H4         = 4.838E-11    NH3         = 4.862E-01    HE           = 0.000E+00
 HF           = 5.135E-01    F           = 2.028E-09    SIF4         = 1.664E-04
 SIF3         = 4.471E-13    SIHF3       = 8.513E-11    SIF3NH2      = 4.631E-10
SURFACE SITE FRACTIONS ON PHASE (SITE)           2
 NHSIF(S)     = 2.986E-04    SIF3NH2(S)  = 4.149E-07    SIF2NH(S)    = 5.493E-05
 NH2SIFNH(S) = 1.421E-04    NHSIFNHSIFNH= 2.844E-04    NHNH2(S)     = 9.992E-01
 SUM OF SURFACE SITE FRACTIONS  0.999999999999999
 SURFACE SITE DENSITY    4.170000000000000E-009
BULK DEPOSITION (GM/CM**2) IN PHASE           3
 SI(D)        = 1.043E-07
BULK DEPOSITION (GM/CM**2) IN PHASE           4
 N(D)         = 5.108E-08
```

9. LSODE Summary

```
      subroutine lsode (f, neq, y, t, tout, itol, rtol, atol, itask,
     1            istate, iopt, rwork, lrw, iwork, liw, jac, mf)
      external f, jac
      integer neq, itol, itask, istate, iopt, lrw, iwork, liw, mf
      double precision y, t, tout, rtol, atol, rwork
      dimension neq(1), y(1), rtol(1), atol(1), rwork(lrw), iwork(liw)
c-----------------------------------------------------------------------
c this is the march 30, 1987 version of
c lsode.. livermore solver for ordinary differential equations.
c this version is in double precision.
c
c lsode solves the initial value problem for stiff or nonstiff
c systems of first order ode-s,
c     dy/dt = f(t,y) ,  or, in component form,
c     dy(i)/dt = f(i) = f(i,t,y(1),y(2),...,y(neq)) (i = 1,...,neq).
c lsode is a package based on the gear and gearb packages, and on the
c october 23, 1978 version of the tentative odepack user interface
c standard, with minor modifications.
c-----------------------------------------------------------------------
c reference..
c     alan c. hindmarsh,  odepack, a systematized collection of ode
c     solvers, in scientific computing, r. s. stepleman et al. (eds.),
c     north-holland, amsterdam, 1983, pp. 55-64.
c-----------------------------------------------------------------------
c author and contact.. alan c. hindmarsh,
c                      computing and mathematics research div., l-316
c                      lawrence livermore national laboratory
c                      livermore, ca 94550.
c-----------------------------------------------------------------------
c summary of usage.
c
c communication between the user and the lsode package, for normal
c situations, is summarized here.  this summary describes only a subset
c of the full set of options available.  see the full description for
c details, including optional communication, nonstandard options,
c and instructions for special situations.  see also the example
c problem (with program and output) following this summary.
c
c a. first provide a subroutine of the form..
c               subroutine f (neq, t, y, ydot)
c               dimension y(neq), ydot(neq)
c which supplies the vector function f by loading ydot(i) with f(i).
c
c b. next determine (or guess) whether or not the problem is stiff.
c stiffness occurs when the jacobian matrix df/dy has an eigenvalue
c whose real part is negative and large in magnitude, compared to the
c reciprocal of the t span of interest.  if the problem is nonstiff,
c use a method flag mf = 10.  if it is stiff, there are four standard
c choices for mf, and lsode requires the jacobian matrix in some form.
c this matrix is regarded either as full (mf = 21 or 22),
c or banded (mf = 24 or 25).  in the banded case, lsode requires two
```

```
c half-bandwidth parameters ml and mu.  these are, respectively, the
c widths of the lower and upper parts of the band, excluding the main
c diagonal.  thus the band consists of the locations (i,j) with
c i-ml .le. j .le. i+mu, and the full bandwidth is ml+mu+1.
c
c c. if the problem is stiff, you are encouraged to supply the jacobian
c directly (mf = 21 or 24), but if this is not feasible, lsode will
c compute it internally by difference quotients (mf = 22 or 25).
c if you are supplying the jacobian, provide a subroutine of the form..
c               subroutine jac (neq, t, y, ml, mu, pd, nrowpd)
c               dimension y(neq), pd(nrowpd,neq)
c which supplies df/dy by loading pd as follows..
c     for a full jacobian (mf = 21), load pd(i,j) with df(i)/dy(j),
c the partial derivative of f(i) with respect to y(j).  (ignore the
c ml and mu arguments in this case.)
c     for a banded jacobian (mf = 24), load pd(i-j+mu+1,j) with
c df(i)/dy(j), i.e. load the diagonal lines of df/dy into the rows of
c pd from the top down.
c     in either case, only nonzero elements need be loaded.
c
c d. write a main program which calls subroutine lsode once for
c each point at which answers are desired.  this should also provide
c for possible use of logical unit 6 for output of error messages
c by lsode.  on the first call to lsode, supply arguments as follows..
c f      = name of subroutine for right-hand side vector f.
c          this name must be declared external in calling program.
c neq    = number of first order ode-s.
c y      = array of initial values, of length neq.
c t      = the initial value of the independent variable.
c tout   = first point where output is desired (.ne. t).
c itol   = 1 or 2 according as atol (below) is a scalar or array.
c rtol   = relative tolerance parameter (scalar).
c atol   = absolute tolerance parameter (scalar or array).
c          the estimated local error in y(i) will be controlled so as
c          to be roughly less (in magnitude) than
c             ewt(i) = rtol*abs(y(i)) + atol     if itol = 1, or
c             ewt(i) = rtol*abs(y(i)) + atol(i)  if itol = 2.
c          thus the local error test passes if, in each component,
c          either the absolute error is less than atol (or atol(i)),
c          or the relative error is less than rtol.
c          use rtol = 0.0 for pure absolute error control, and
c          use atol = 0.0 (or atol(i) = 0.0) for pure relative error
c          control.  caution.. actual (global) errors may exceed these
c          local tolerances, so choose them conservatively.
c itask  = 1 for normal computation of output values of y at t = tout.
c istate = integer flag (input and output).  set istate = 1.
c iopt   = 0 to indicate no optional inputs used.
c rwork  = real work array of length at least..
c             20 + 16*neq                    for mf = 10,
c             22 +  9*neq + neq**2           for mf = 21 or 22,
c             22 + 10*neq + (2*ml + mu)*neq  for mf = 24 or 25.
```

```
c lrw    = declared length of rwork (in user-s dimension).
c iwork  = integer work array of length at least..
c             20        for mf = 10,
c             20 + neq  for mf = 21, 22, 24, or 25.
c          if mf = 24 or 25, input in iwork(1),iwork(2) the lower
c          and upper half-bandwidths ml,mu.
c liw    = declared length of iwork (in user-s dimension).
c jac    = name of subroutine for jacobian matrix (mf = 21 or 24).
c          if used, this name must be declared external in calling
c          program.  if not used, pass a dummy name.
c mf     = method flag.  standard values are..
c          10 for nonstiff (adams) method, no jacobian used.
c          21 for stiff (bdf) method, user-supplied full jacobian.
c          22 for stiff method, internally generated full jacobian.
c          24 for stiff method, user-supplied banded jacobian.
c          25 for stiff method, internally generated banded jacobian.
c note that the main program must declare arrays y, rwork, iwork,
c and possibly atol.
c
c e. the output from the first call (or any call) is..
c      y = array of computed values of y(t) vector.
c      t = corresponding value of independent variable (normally tout).
c istate = 2  if lsode was successful, negative otherwise.
c          -1 means excess work done on this call (perhaps wrong mf).
c          -2 means excess accuracy requested (tolerances too small).
c          -3 means illegal input detected (see printed message).
c          -4 means repeated error test failures (check all inputs).
c          -5 means repeated convergence failures (perhaps bad jacobian
c             supplied or wrong choice of mf or tolerances).
c          -6 means error weight became zero during problem. (solution
c             component i vanished, and atol or atol(i) = 0.)
c
c f. to continue the integration after a successful return, simply
c reset tout and call lsode again.  no other parameters need be reset.
c
c-----------------------------------------------------------------------
```

APPENDIX A. STORAGE ALLOCATION FOR THE WORK ARRAYS

The work arrays contain all the pertinent information about the species and the reaction mechanism. They also contain some work space needed by various routines for internal manipulations. If a user wishes to modify an SKLIB subroutine or to write new routines, he will probably want to use the work arrays directly. The starting addresses for information stored in the work arrays are found in the labeled common block, COMMON /SKSTRT/, and are explained below.

```
 COMMON /SKSTRT/ MAXSPR, NELM,   NKKGAS, NKKSUR, NKKBLK, NKKTOT,
1                NPHASE, NFSUR,  NLSUR,  NNSUR,  NFBLK,  NLBLK,
2                NNBLK,  NIISUR, NSPAR,  NSCOV,  NIICOV, NIIREV,
3                NIISTK, NIICON, MXTP,   NCP,    NCP1,   NCP2,
4                NCP2T,  IiPKST, IiPKND, IiPTOT, IiKPHS, IiKCHG,
5                IiKCMP, IiNSCV, IiKNT,  IiNRPP, IiNREA, IiNUNK,
6                IiNU,   IiNCF,  IiNSUM, IiCOV,  IiKCV,  IiREV,
7                IiSTK,  IrPATM, IrRU,   IrRUC,  IrSDEN, IrKTMP,
8                IrKTHM, IrKDEN, IrAWT,  IrKWT,  IrPAR,  IrCOV,
9                IrREV,  IrKT1,  IrKT2,  IrPT1,  IrIT1,  IrIT2,
*                IrIT3,  IcENAM, IcKNAM, IcPNAM
```

INDEX CONSTANTS.

MAXSPR - Maximum number of species in any surface reaction. Unless changed in the interpreter MAXSPR=12.
NELM - Number of elements.
NKKGAS - Number of gas-phase species.
NKKSUR - Number of surface species.
NKKBLK - Number of bulk species.
NKKTOT - Total number of species. NKKTOT=NKKGAS+NKKSUR+NKKBLK
NPHASE - Number of phases; gas + sites + bulk.
NFSUR - Pointer to the first surface phase.
NLSUR - Pointer to the last surface phase.
NNSUR - Number of surface phases.
NFBLK - Pointer to the first bulk phase.
NLBLK - Pointer to the last bulk phase.
NNBLK - Number of bulk phases.
NIISUR - Number of surface reactions.
NSPAR - Number of parameters in the rate expression. In the current formulation NSPAR=3.
NSCOV - Number of parameters in a coverage expression.
NIICOV - Number of surface reactions with coverage parameters.
NIIREV - Number of surface reactions with reverse parameters.
NIISTK - Number of surface reactions with sticking coefficients.
NIICON - Number of surface reactions with non-conserved sites.
MXTP - Maximum number of temperatures used to fit thermodynamic properties of species; unless changed in the interpreter MXTP=3.

NCP - Number of polynomial coefficients to fits of CP/R. Unless the interpreter and the thermodynamic data base are changed NCP=5.
NCP1 - NCP + 1.
NCP2 - NCP + 2.
NCP2T - (MXTP-1) * NCP2. Total number of thermodynamic fit coefficients for (MXTP-1) temperature ranges. Unless changed NCP2T=14.

STARTING ADDRESSES FOR THE INTEGER WORK SPACE, ISKWRK.

IiPKST - Starting address of the starting species numbers of the NPHASE phases. ISKWRK(IiPKST+N-1) is the first species in the Nth phase.
IiPKND - Starting address of the final species numbers of the NPHASE phases. ISKWRK(IiPKND+N-1) is the last species in the Nth phase.
IiPTOT - Starting address of the total number of species in the NPHASE phases. ISKWRK(IiPTOT+N-1) is the total number of species in the Nth phase.
IiKPHS - Starting address of the phases of the NKKTOT species.
ISKWRK(IiKPHS+K-1) = -1, the Kth species is a solid
= 0, the Kth species is a gas
= +1, the Kth species is a liquid
IiKCHG - Starting address of the electronic charges of the NKKTOT species.
ISKWRK(IiKCHG+K-1) = -2, the Kth species has two excess electrons.
IiKCMP - Starting address of the elemental content of the NELM elements in the NKKTOT species.
ISKWRK(IiKCMP + (K-1)*NELM + M - 1) is the number of atoms of the Mth element in the Kth species.
IiNSCV - Starting address of site coverage for the NKKTOT species.
ISKWRK(IiNSCV+K-1) is site coverage for the Kth species.
IiKTMP - Starting address of the number of temperatures used to fit thermodynamic coefficients for the NKKTOT species.
ISKWRK(IiKTMP+K-1) = N, N temperatures were used in the fit for the Kth species.
IiNRPP - Starting address of the total number of participant species for the NIISUR reactions, and the reversibility of the reactions.
ISKWRK(IiNRPP+I-1) = +N, the Ith reaction is reversible and has N participant species (reactants + products)
= -N, the Ith reaction is irreversible and has N participant species (reactants + products)

IiNREA - Starting address of the number of reactants only for the NIISUR reactions. ISKWRK(IiNREA+I-1) is the total number of reactants (not including products) in the Ith reaction.

IiNUNK - Starting address of a matrix of species index numbers for the MAXSPR species in the NIISUR reactions. ISKWRK(IiNUNK+(I-1)*MAXSPR+N-1) = K, the species number of the Nth participant species in the Ith reaction.
IiNU - Starting address of a matrix of stoichiometric coefficients of the MAXSPR species in the NIISUR reactions. ISKWRK(IiNU+(I-1)*MAXSPR+N-1) is the coefficient of the Nth participant species in the Ith reaction.
IiNCF - Starting address of the net change in sites for the NPHASE phases.
IiNSUM - Starting address of the sum of the stoichiometric coefficients of the gas-phase reactants and products in the NIISUR surface reactions. ISKWRK(IiNSUM+I-1) is the sum of the stoichiometric coefficients of the gas-phase species in surface reaction I.
IiCOV - Starting address of the NIICOV reaction numbers for surface reactions with coverage parameters. ISKWRK(IiCOV+N-1) is the surface reaction index I of the Nth coverage declaration.
IiKCV - Starting address of the NIICOV species index numbers for surface reactions with coverage parameters. ISKWRK(IiKCV+N-1) is the species index K of the Nth coverage declaration; the reaction number is ISKWRK(IiCOV+N-1).
IiREV - Starting address of the NIIREV reaction numbers for surface reactions with reverse parameters. ISKWRK(IiREV+N-1) is the surface reaction index I of the Nth reaction with reverse parameters.
IiSTK - Starting address of the NIISTK reaction numbers for surface reactions with sticking coefficients. ISKWRK(IiSTK+N-1) is the surface reaction index I of the Nth reaction with sticking coefficients.

STARTING ADDRESSES FOR THE REAL WORK SPACE, RSKWRK.

IrPATM - RSKWRK(IrPATM) is the pressure of one standard atmosphere (dynes/cm**2).
IrRU - RSKWRK(IrRU) is the universal gas constant (ergs/mole-K).
IrRUC - RSKWRK(IrRUC) is the universal gas constant (cal/mole-K).
IrSDEN - Starting address of densities for the NPHASE phases. RSKWRK(IrSDEN+N-1) is the density of the Nth phase.
IrKTMP - Starting address of a matrix of the MXTP temperatures used in fits of thermodynamic properties for the NKKTOT species. RSKWRK(IrKTMP+(K-1)*MXTP+N-1) is the Nth temperature for the Kth species.

IrKTHM - Starting address of a three-dimensional array of coefficients for the NCP2 fits to the thermodynamic properties for the NKKTOT species, for (MXTP-1) temperature ranges.
RSKWRK(IrKTHM+(L-1)*NCP2+(K-1)*NCP2T+N-1) = A(N,L,K;
A(N,L,K),N=1,NCP2T = polynomial coefficients in the fits for the Kth species and the Lth temperature range, where the total number of temperature ranges for the Kth species is ISKWRK(IcNT+K-1) - 1.

IrKDEN - Starting address of the densities (gm/cm**3 for gas or bulk species, gm/cm**2 for surface species) of the NKKTOT species. RSKWRK(IrKDEN+K-1) is the density of the Kth species.

IrAWT - Starting address of the atomic weights for the NELM elements. RSKWRK(IrAWT+M-1) is the atomic weight of the Mth element.

IrKWT - Starting address of the molecular weights for the NKKTOT species. RSKWRK(IrKWT+K-1) is the molecular weight of the Kth species.

IrPAR - Starting address of a matrix of the NSPAR Arrhenius parameters for the NIISUR surface reactions.
RSKWRK(IrPAR+(I-1)*NSPAR+N-1) is the Nth parameter of the Ith surface reaction, where
N=1 is the pre-exponential factor (mole-cm-sec-K),
N=2 is the temperature exponent, and
N=3 is the activation energy (kelvins).

IrCOV - Starting address of a matrix of the NSCOV coverage parameters for the NIICOV coverage declarations.
RSKWRK(IrCOV+(N-1)*NSCOV+L-1) is the Lth coverage parameter for the Nth coverage declaration;
the reaction number is ISKWRK(IiCOV+N-1);
the species number is ISKWRK(IiKCV+N-1).

IrREV - Starting address of a matrix of the NSPAR reverse parameters for the NIIREV surface reactions with reverse parameters. RSKWRK(IrREV+(N-1)*NSPAR+L-1) is the Lth reverse parameter for the Nth reverse reaction; the reaction number is ISKWRK(IiREV+N-1).

IrKT1 - Starting addresses of arrays of internal work space
IrKT2 - of length NKKTOT.

IrIT1 - Starting addresses of arrays of internal work space
IrIT2 - of length NIISUR.
IrIT3 -

STARTING ADDRESSES FOR THE CHARACTER WORK SPACE, CSKWRK.

IcENAM - Starting address of the NELM element names.
CSKWRK(IcENAM+M-1) is the name of the Mth element.

IcKNAM - Starting address of the NKKTOT species names.
CSKWRK(IcKNAM+K-1) is the name of the Kth species.

IcPNAM - Starting address of the NPHASE phase names.
CSKWRK(IcPNAM+N-1) is the name of the Nth phase.

The linking file consists of the following binary records:

```
1) Information about the linking file:  VERS, PREC, KERR
   Where VERS   = character*16 string representing the version
                  number of the interpreter which created the
                  the linking file.
         PREC   = character*16 string representing the machine
                  precision of the linking file (SINGLE, DOUBLE).
         KERR   = logical which indicates whether or not
                  an error occurred in the interpreter input.

2) Index constants:
   LENISK, LENRSK, LENCSK, MAXSPR, MXTP, NCP, NELM,
   NKKGAS, NKKSUR, NKKBLK, NKKTOT, NPHASE, NFSUR, NLSUR,
   NNSUR, NFBLK, NLBLK, NNBLK, NIISUR, NSPAR, NSCOV, NIICOV,
   NIIREV, NIISTK, NIICOM
   Where LENISK = required length of ISKWRK.
         LENRSK = required length of RSKWRK.
         LENCSK = required length of CSKWRK.

3) Element information:
   (CSKWRK(IcENAM+M-1),                                !element names
    RSKWRK(IrAWT+M-1),                                 !atomic weights
    M=1,NELM)

4) Species information:
   (CSKWRK(IcKNAM+K-1),                                !species names
    RSKWRK(IrKWT+K-1),                                 !molecular weight
    ISKWRK(IiKPHS+K-1),                                !phase
    ISKWRK(IiKCHG+K-1),                                !charge
    ISKWRK(IiKNT+K-1),                                 !# of fit temps
    ISKWRK(IiNSCV+K-1),                                !site coverage
    RSKWRK(IrKDEN+K-1),                                !species densities
    (ISKWRK(IiKCMP+(K-1)*NELM+M-1),M=1,NELM),          !composition
    (RSKWRK(IrKTMP+(K-1)*MXTP+L-1), L=1,MXTP),         !array of temps
    ((RSKWRK(IrKTHM+(L-1)*NCP2+(K-1)*NCP2T+N-1),!fit coeff'nts
            N=1,NCP2), L=1,(MXTP-1)),
    K=1,NKKTOT),

5) Phase Information:
   (CSKWRK(IcPNAM+N-1),                                !phase names
    ISKWRK(IiPKST+N-1),                                !start species
    ISKWRK(IiPKND+N-1),                                !end species
    ISKWRK(IiPTOT+N-1),                                !# of species
    RSKWRK(IrSDEN+N-1),                                !phase densities
    N=1,NPHASE),
```

```
6) Reaction information (if NIISUR > 0)
   (ISKWRK(IiNRPP+I-1),                         !# of species
    ISKWRK(IiNREA+I-1),                         !# of reactants
    ISKWRK(IiNSUM+I-1),                         !sum gas-phase coeff.
    (RSKWRK(IrPAR+(I-1)*NSPAR+N-1),N=1,NSPAR),  !Arr. coefficients
    (ISKWRK(IiNUNK+(I-1)*MAXSPR+N-1),           !stoic coef
    ISKWRK(IiNU+(I-1)*MAXSPR+N-1),N=1,MAXSPR),  !species numbers
    I=1,NIISUR),
   (ISKWRK(IiCOV+I-1), ISKWRK(IiKCV+I-1),     !cov. reactions,species
   (RSKWRK(IrCOV+(I-1)*NSCOV+L-1),L=1,NSCOV), !cov. parameters
    I=1,NIICOV),
   (ISKWRK(IiREV+I-1),                          !rev. reactions
   (RSKWRK(IrREV+(I-1)*NSPAR+L-1),L=1,NSPAR),   !rev. parameters
    I=1,NIIREV),
   (ISKWRK(IiSTK+I-1),I=1,NIISTK)               !sticking reactions
```

REFERENCES

1. Kee, R. J., Rupley, F. M., and Miller, J. A., "Chemkin-II: A Fortran Chemical Kinetics Code Package for the Analysis of Gas-Phase Chemical Kinetics," Sandia National Laboratories Report, SAND89-8209 (1989).

2. Kee, R. J., Warnatz, J, and Miller, J. A., "A Fortran Computer Code Package for the Evaluation of Gas-Phase Viscosities, Conductivities, and Diffusion Coefficients," Sandia National Laboratories Report, SAND83-8209 (1983).

3. Kee, R. J., Dixon-Lewis, G., Warnatz, J., Coltrin, M. E., and Miller, J. A., "A Fortran Computer Package for the Evaluation of Gas-Phase, Multicomponent Transport Properties," Sandia National Laboratories report, in preparation.

4. Kee, R. J., and Miller, J. A., "A Structured Approach to the Computational Modeling of Chemical Kinetics and Molecular Transport in Flowing Systems," *Springer Series in Chemical Physics* **47**, 196 (1986). (Also available as Sandia National Laboratories Report SAND80-8003.)

5. Kee, R. J., Miller, J. A., and Jefferson, T. H., "Chemkin: A General-Purpose Problem-Independent, Transportable, Fortran Chemical Kinetics Code Package," Sandia National Laboratories Report, SAND80-8003 (1980).

6. Kee, R. J., Rupley, F. M., and Miller, J. A., "The Chemkin Thermodynamic Database," Sandia National Laboratories Report, SAND87-8215 (1987).

7. Gordon S. and McBride, B. J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks and Chapman-Jouguet Detonations," NASA SP-273 (1971).

8. Eriksson, G., *Acta Chem. Scand.* **25** 2651 (1971).

9. Hindmarsh, A. C., "ODEPACK, A Systemized Collection of ODE Solvers," in *Scientific Computing*, edited by R. S. Stephen et al., Vol 1, *IMACS Trans. on Scientific Computation* (North-Holland, Amsterdam), pp. 55–64 (1983).

UNLIMITED RELEASE

INITIAL DISTRIBUTION

1000	V. Narayanamurti
	Attn: J. P. Vandevender, 1200
1100	F. L. Vook
	Attn: S. T. Picraux, 1110
	P. S. Peercy, 1140
	G. A. Samara, 1150
	A. Owyoung, 1160
1120	J. B. Gerardo
1126	M. E. Bartram
1126	W. G. Breiland
1126	M. E. Coltrin (10)
1126	J. R. Creighton
1126	P. Ho
1126	A. W. Johnson
1126	H. K. Moffat
1128	R. T. McGrath
1141	J. Y. Tsao
1500	E. H. Barsis
1512	M. R. Baer
1553	W. L. Hermina
1600	W. Herrmann
1800	R. J. Eagan
1810	D. W. Schaefer
1812	R. J. Buss
1812	C. L. Renschler
1830	A. D. Romig, Jr
1834	A. K. Hays
1841	B. D. Kay

8000	J. C. Crawford
	Attn: E. E. Ives, 8100
	R. C. Wayne, 8400
	P. E. Brewer, 8500
8200	R. J. Detry
	Attn: W. D. Wilson, 8210
	D. L. Crawford
	R. C. Dougherty, 8270
	D. J. Havlik, 8280
8233	J. F. Grcar
8240	C. W. Robinson
	Attn: G. A. Benedetti, 8241
	M. R. Birnbaum, 8242
	M. L. Callabresi, 8243
	C. M. Hartwig, 8244
8244	R. S. Larson
8245	N. E. Bergan
8245	D. S. Dandy
8245	H. A. Dwyer
8245	G. H. Evans
8245	W. G. Houf
8245	R. J. Kee (100)
8245	A. E. Lutz
8245	E. Meeks
8245	F. M. Rupley

8300	P. L. Mattern
8300-A	J. S. Binkley
8300-A	R. Carling
8310	D. L. Lindner, acting
8316	R. E. Stoltz
8340	W. Bauer
8341	W. G. Wolfer
8342	G. D. Kubiak
8342	D. A. Outka
8342	R. H. Stulen
8343	G. J. Thomas
8347	W. L. Hsu
8351	J.-Y. Chen
8353	C. F. Melius
8353	J. A. Miller
8353	F. P. Tully, acting
8354	M. L. Koszykowski
8354	R. E. Palmer
8360	W. J. McLean
8361	M. D. Allendorf
8361	D. Hardesty
8361	R. E. Mitchell
8362	D. L. Siebers, acting

8535	Publications for OSTI (10)
8535	Publications/Central Technical Files, 8524-2
3141	Technical Library Processes Division (3)
8524-2	Central Technical Files (2)