JUN 2 2000

SANDIA REPORT
SAND2000-1202
Unlimited Release
Printed May 2000

RECEIVED
JUN 08 2000
OSTI

Surfkin: A Program to Solve Transient and Steady State Heterogeneous Reaction Kinetics

Michael E. Coltrin, Ryan R. Wixom, and David S. Dandy

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm



DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND2000-1202
Unlimited Release
Printed May 2000

Surfkin: A Program To Solve Transient And Steady State Heterogeneous Reaction Kinetics

Michael E. Coltrin and Ryan R. Wixom
Chemical Processing Sciences Department
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-0601

David S. Dandy
Chemical and Bioresource Engineering
Colorado State University
Fort Collins, CO 80523-1370

ABSTRACT

This report documents the Surfkin program, which simulates the kinetics of heterogeneous chemical reactions. The program is designed for use with the Chemkin and Surface Chemkin (heterogeneous chemistry) programs. It calculates time-dependent or steady state surface site fractions and bulk-species production / destruction rates. The surface temperature may be specified as a function of time to simulate a temperature-programmed desorption experiment, for example. This report serves as a user's manual for the program, explaining the required input and format of the output. Two detailed example problems are included to further illustrate the use of this program.

TABLE OF CONTENTS

I. INTRODUCTION 6

II. HETEROGENEOUS RATE EQUATIONS 7

III. PROGRAM STRUCTURE AND CONTROL 11
- Executing Chemkin Interpreters and Surfkin 11
- Reading the Surfkin binary output file 13
- Compiling and linking Surfkin 15
  - Sample Unix Makefile 15

IV. PROGRAM INPUT AND OUTPUT 16
- List of Keywords 17

V. Surfkin Example: Temperature Programmed Desorption 21
- Input files 22
  - Naming input and output files 22
  - Chemkin Interpreter input file: steam.chem 23
  - Chemkin Interpreter output file: steam.chem.out 23
  - Surface Chemkin Interpreter input file: steam.surf 24
  - Surface Chemkin Interpreter output file: steam.surf.out 25
  - Surfkin input file: steam.surfkin 26
- Running the simulation with script files 26
  - Script file: runsteam 27
  - Second script file: runsurfkin 27
  - Execution at the command prompt 28
  - Surfkin output file: steam.surfkin.out 29
  - Grep output: steam.txt 32
  - Analysis of the desorption rate 34

VI. Surfkin Example: Competitive Adsorption 35
- Theoretical background 35
- Input files 36
  - Chemkin Interpreter input file – coadsorb.chem 36
  - Chemkin Interpreter output file – coadsorb.chem.out 37
  - Surface Chemkin Interpreter input file – coadsorb.surf 38
  - Surface Chemkin Interpreter output file – coadsorb.surf.out 39
  - Surfkin input file – coadsorb.surfkin 40
  - Running the simulation 42
- Running the simulation with script files 42
  - Script file: runcoad 43
  - Surfkin output file: coadsorb.surfkin.out 44
  - Script file output: coadsorb.txt 45
  - Plot of the data in coadsorb.txt 46

REFERENCES 47

SURFKIN: A PROGRAM TO SOLVE TRANSIENT AND STEADY STATE HETEROGENEOUS REACTION KINETICS

I. INTRODUCTION

Heterogeneous chemical reactions occurring at a gas/surface interface are fundamental in a variety of important applications, such as combustion, catalysis, chemical vapor deposition and plasma processing. Detailed simulation of these processes may involve complex, coupled fluid flow, heat transfer, gas-phase chemistry, in addition to heterogeneous reaction chemistry. This report describes the Surfkin application code designed to aid in the analysis of the isolated heterogeneous reaction portion of the problem alone. It is designed to help develop and test surface reaction mechanisms easily, in the absence of the complications of fluid flow, etc., needed to solve the entire problem.

A heterogeneous reaction mechanism and rate expressions for each reaction are specified as input files to the Surface Chemkin Interpreter. The Surfkin program has access to the mechanism and kinetics information via the Surface Chemkin linking file. The modular approach of the Chemkin suite of software separates the specification of the reaction chemistry from the application code itself [1]. Thus, reaction mechanisms developed and tested using the simpler Surfkin code are directly transferable to more complex reacting flow simulations. Surfkin solves a much simpler problem, calculating the time-dependent or steady-state site fractions of surface species determined by the heterogeneous reaction mechanism, with the concentrations of any gas-phase species and the total pressure held fixed.

The surface temperature in the simulation may be held fixed or specified as a function of time. Temperature-programmed desorption (TPD) is often used to study the kinetics and thermochemistry of species adsorbed on surfaces. In this technique, the surface temperature is usually increased linearly with time from some low value, at which the surface is saturated with adsorbed species, to a much higher temperature, where most species have desorbed, leaving a bare surface. Analysis of desorption rates as a function of time (temperature) can yield surface kinetic rates and thermodynamic information. Thus, Surfkin can be used to simulate TPD experiments by specifying the appropriate surface temperature as a function of time.

This document describes the kinetic rate equations solved by Surfkin (Sec. II), the program structure and execution (Sec. III), and the input file format required to set up a Surfkin simulation (Sec. IV). Two detailed example problems are then presented. Section V shows an example using Surfkin to calculate the surface desorption rate of H_2O from a Pt surface as a function of time as the surface temperature is being ramped (increased) with time. Results are compared with the Redhead analysis [2] commonly used to extract rate information from temperature-programmed desorption experiments. The second example, in Sec. VI, uses Surfkin in a steady-state mode to analyze a competitive-adsorption mechanism.

II. HETEROGENEOUS RATE EQUATIONS

Surfkin solves the kinetic rate equations for arbitrarily complex heterogeneous reaction mechanisms. The program can solve a transient problem, in which the surface site fractions at time zero are specified, and then the differential equations describing the time-evolution are integrated forward in time. Surfkin can alternatively solve for the set of surface site fractions at steady state, which may also be of interest.

The surface reaction mechanism is handled through the Surface Chemkin interface. As such, details of the chemistry, including the number and names of chemical species, number of reactions, their stoichiometric coefficients, etc., are treated in a very general fashion. The nomenclature summarized next is explained in more detail in the Surface Chemkin user's manual [3].

Using a general notation, the surface reaction mechanism consists of I reactions (with index i, below) among K species with (index k). A general chemical reaction is written

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

where χ_k is the chemical name, e.g., $H_2O(s)$, of species k, and ν'_{ki} and ν''_{ki} are the forward and reverse stoichiometric coefficients, respectively, for species k in reaction i. The net stoichiometric coefficient is

$$\nu_{ki} = (\nu''_{ki} - \nu'_{ki}). \tag{2}$$

The rate-of-progress of reaction i (mol/cm^2/s for a heterogeneous reaction) is

$$q_i = k_{f_i} \prod_{k=1}^{K} [X_k]^{\nu'_{ki}} - k_{r_i} \prod_{k=1}^{K} [X_k]^{\nu''_{ki}}, \tag{3}$$

where k_{f_i} and k_{r_i} are the rate constants for the forward and reverse reactions, respectively. The "concentration" terms on the right-hand-side of Eq. (3) depend upon whether species k is a gas, surface, or bulk species. For an ideal gas-phase species, the molar concentration (mol/cm^3) is

$$[X_k] = \frac{P_k}{RT}, \tag{4}$$

where P_k is the partial pressure of species k, R is the gas constant, and T is temperature (K).

For a surface species,

$$[X_k] = Z_k(n)\Gamma_n/\sigma_k(n) \qquad (5)$$

is the molar concentration on the surface (mol/cm^2). In Eq. (5), $Z_k(n)$ is the site fraction of species k in surface phase n (if more than one surface phase exists [3]), Γ_n is the total density of surface sites (mol/cm^2) of surface phase n, and $\sigma_k(n)$ is a coverage number, i.e., the number of surface sites that each species k "covers" on the surface. For example, a large species like $Si_2H_5(s)$ may cover or occupy two adsorption sites in some reaction scheme. The site fractions in a given surface phase are assumed to sum to unity

$$\sum_{k=K_s^f(n)}^{K_s^l(n)} Z_k(n) = 1. \qquad \left(n = N_s^f, ..., N_s^l\right) \qquad (6)$$

The nomenclature in Eq. (6) gets rather complicated (because of the generality). The indices in the summation run from the index of the first surface species to the last surface species in phase n. In the general case of more than one surface phase, n ranges from the index of the first surface phase to the last surface phase.

In Eq. (3), if species k is a bulk-phase species, the generalized "concentration" appropriate for the rate-of-progress expression is the activity,

$$[X_k] = a_k. \qquad (7)$$

The Chemkin and Surface Chemkin suite of codes generally treat the gas and solid bulk phases as ideal solutions, in which case the activity of species k simply is set to its mole fraction. Thus, it is also assumed that the activities in each bulk phase also sum to unity

$$\sum_{k=K_b^f(n)}^{K_b^l(n)} a_k(n) = 1. \qquad \left(n = N_b^f, ..., N_b^l\right) \qquad (8)$$

The indices in the summation run from the index of the first bulk species to the last bulk species in bulk phase n. In the general case of more than one bulk phase, n ranges from the index of the first bulk phase to the last bulk phase.

The molar production/destruction rate (mol/cm^2/s) of species k is

$$\dot{s}_k = \sum_{i=1}^{I} \upsilon_{ki} q_i, \qquad (k = 1, ..., K), \qquad (9)$$

which includes the contributions from all I reactions in the mechanism.

The total number of surface sites in surface phase n may change in a given reaction mechanism (although this option is seldom used). For example, suppose a mechanism

defines surface phase 1 as "ledge sites" and surface phase 2 as "kink sites," and that some chemical reactions involve conversion of one type of site to the other. In this case, the total of number of ledge and kink sites changes with time as the kinetics evolve. In such cases, one must account for the production rate of surface site type n as

$$\dot{\Gamma}_n = \sum_{i=1}^{I} \Delta\sigma(n,i) q_i, \qquad \left(n = N_s^f, \ldots, N_s^l\right), \tag{10}$$

where $\Delta\sigma(n,i)$ is the number of sites of type n created or destroyed in reaction i:

$$\Delta\sigma(n,i) = \sum_{k=K_s^f(n)}^{K_s^l(n)} \upsilon_{ki}\sigma_k(n). \tag{11}$$

For a surface species, the molar time-rate-of-change, $\dot{s}_k$, can also be equated to the time derivative of Eq. (5)

$$\dot{s}_k \equiv \left[\dot{X}_k\right] = \frac{\dot{Z}_k(n)\Gamma_n}{\sigma_k(n)} + \frac{Z_k(n)\dot{\Gamma}_n}{\sigma_k(n)}. \tag{12}$$

Rearranging Eq. (12) yields an expression for the time derivative of the surface site fraction of species k on surface phase n

$$\dot{Z}_k(n) = \frac{\dot{s}_k\sigma_k}{\Gamma_n} - \frac{Z_k(n)\dot{\Gamma}_n}{\Gamma_n}. \tag{13}$$

In heterogeneous reaction mechanisms of interest, such as thin-film deposition or etching, there is often a net conversion between gas-phase and solid (bulk) material. The rate at which mass of bulk species k is created (or destroyed) is

$$\dot{M}_k = \dot{s}_k W_k, \tag{14}$$

where W_k is the molecular weight (g/mol) of species k.

In the transient simulation, Surfkin solves for the each of the surface site fractions $Z_k(n)$, the total site density Γ_n for each surface phase n (which may be constant), and the mass of bulk species k created (or destroyed) as a function of time. The initial (time equals zero) site fraction of each surface species is specified as input to the program. The initial site density of each surface phase is set to Γ_n^o, the standard state site density for surface phase n, which is defined in the Surface Chemkin Interpreter input file (reaction mechanism). At time $t = 0$, the mass of bulk species k created or destroyed M_k is set to zero for each bulk-phase species.

Equations (13), (10), and (14) are ordinary differential equations (ODEs) describing the time evolution of the dependent variables $Z_k(n)$, Γ_n, and M_k in Surfkin. Thus, the set of equations is an initial-value ODE problem, that is integrated forward in time using the VODE software [4]. The Surface Chemkin subroutine library evaluates $\dot{s}_k$, Γ_n, and other quantities needed to form the rate equations.

The gas-phase composition and the pressure are held constant in Surfkin, and the temperature of the gas is assumed to be the temperature of the surface T_s. However, the temperature of the surface can be specified to change linearly as a function of time. The surface temperature is assumed to be constant at T_i between time $t = 0$ and $t = t_i^r$, the initial time at which a linear temperature ramp is begun. The surface temperature is increased linearly with time to a value T_s, the final surface temperature, at time $t = t_f^r$. The temperature of the surface then remains constant at T_s until the end of the simulation at time t_e. The temperature ramp is optional, as are the initial and final periods in which the surface temperature remains constant. For example, if T_i is not input (TINI keyword, discussed in the next section) or $T_i = T_s$, then there would be no temperature ramp. Or, if $t_i^r = 0$ (RMPI keyword), then the temperature ramp would start immediately, with no initial period in which the temperature is held fixed.

After some long time, the surface site fractions will approach their equilibrium values, i.e., steady state. At steady state, the production and destruction rates exactly balance, and the time derivative is $\dot{s}_k = 0$ for every surface species in the mechanism. Although Surfkin could, in principle, determine the steady state surface coverage by simply integrating the kinetics in time until the site fractions no longer change, this is not the approach taken in the program. Surfkin solves for the steady state directly using a routine named "face" written by J. F. Grcar (unpublished), which in turn calls a non-linear equation solver twopnt [5]. The bulk species total mass creation M_k does not have a physical meaning in this limit of infinite time, and those components of the solution are reported as zero.

III. PROGRAM STRUCTURE AND CONTROL

Executing Chemkin Interpreters and Surfkin

In addition to direct user input, the Surfkin program depends on data and subroutines from the Chemkin [6] and Surface Chemkin [3] packages (Detailed instructions on using these software packages are found in their respective users' manuals.) Therefore, to use Surfkin, the user must be able to set up a control procedure that allows for the execution of the Chemkin and Surface Chemkin Interpreters, access to several databases, loading of subroutines from several libraries, and passing of files from one process to another. Figure 1 shows the relationships between these various components. A sample command procedure for running the Surfkin program under the Unix operating system is given in the first example, Section V. Analogous operations would have to be performed under any operating system.

The first step is to execute the Chemkin Interpreter. The Chemkin Interpreter reads user-supplied information about the species and chemical reactions for a particular gas-phase reaction mechanism from standard input (generally, a user-supplied text file in the proper format). It then extracts further information about the species' thermodynamic properties from a database in the file `therm.dat`. This information is stored on the Chemkin Linking File `chem.asc`, a file that is needed by the Surface Chemkin Interpreter, and later by the Chemkin subroutine library, which will be accessed by the Surfkin program. The printed output from the Chemkin Interpreter, including warning and error messages, is written to standard output. Although Surfkin ignores any gas-phase chemical kinetics specified in the Chemkin Interpreter input file, the Chemkin Interpreter execution step is still necessary.

The Surface Chemkin Interpreter must be executed after the Chemkin Interpreter has been run, because it relies on gas-phase species and element information in the Chemkin Linking file. The Surface Chemkin interpreter reads from standard input (another user-supplied text file) information about surface and bulk species names, site types, and surface reactions, and optional thermochemical information (in `therm.dat`). This information is combined into a Surface Chemkin linking file `surf.asc`, which is used by the Surface Chemkin subroutine library, called by the Surfkin program. The printed output from the Surface Chemkin Interpreter is written to standard output.

The input required to set up the Surfkin run is entered in a Keyword format (described in Section IV), and is read from standard input. The Surfkin program writes a descriptive version of the results of its simulation to standard output, and also writes a more detailed binary solution file to the file `save.dat`.

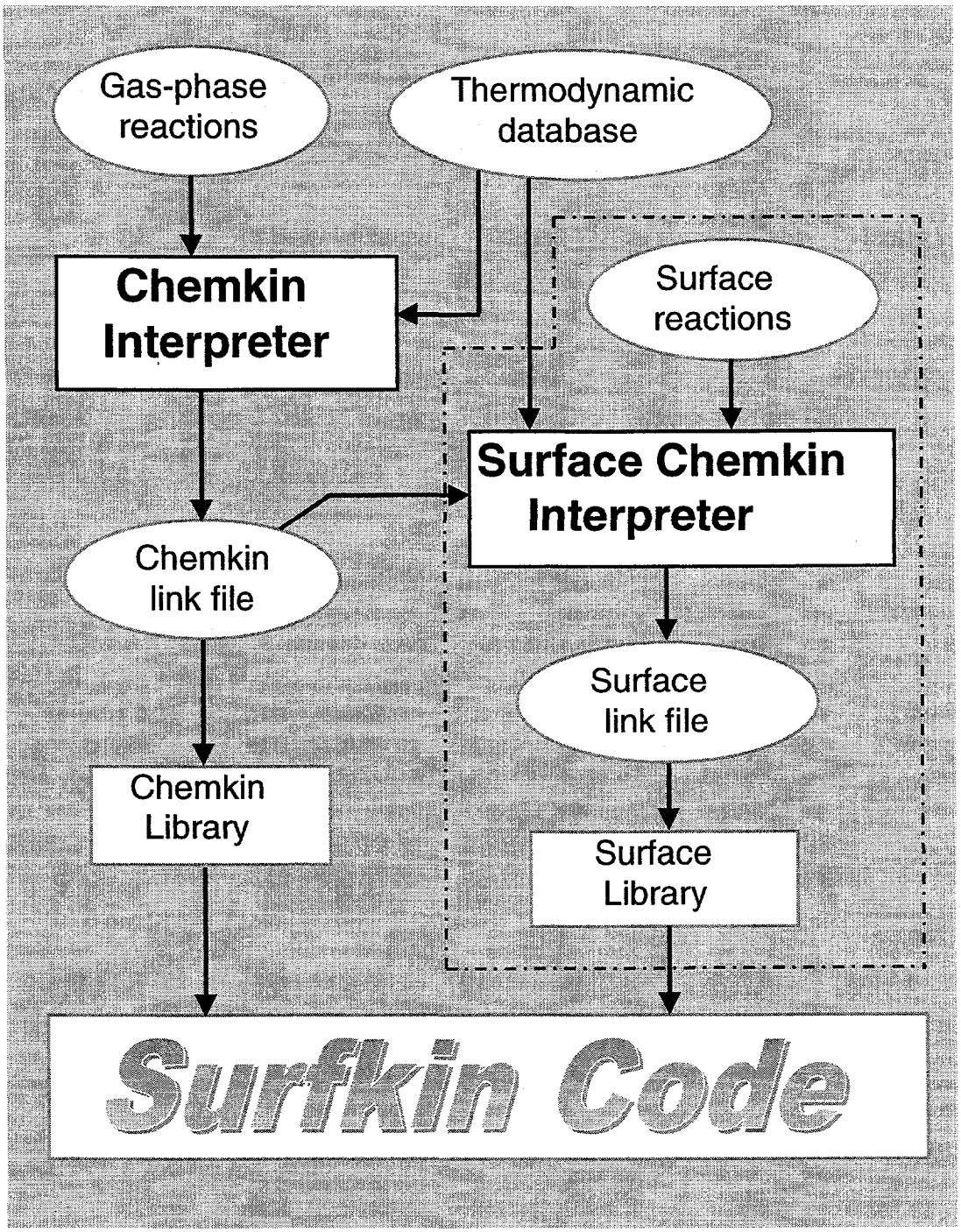


Figure 1. **Relationship of the Surfkin program to the Chemkin and Surface Chemkin preprocessors and the associated input and output files.**

Reading the Surfkin binary output file

Surfkin writes its solution to a data file in binary format, which can be read by a post-processor if desired. The format of the `save.dat` file is specified here.

Several arrays of information are written to the binary output file. Fortran statements to set-up the required arrays are, for example,

```
CHARACTER*16 SPNAME(KKTOT)
DIMENSION GASMOF(KKGAS), Z(NEQ)
```

The array SPNAME contains the names of each species in the Surface Chemkin mechanism. The array is KKTOT long, the total number of gas-phase species (KKGAS) plus surface species (KKSURF) plus bulk species (KKBULK). Each element is a `CHARACTER*16` variable.

The array GASMOF contains the mole fraction of each of the KKGAS species. The gas mole fractions remain constant throughout the simulation. This array and the rest of the data on the solution file are either all single precision or all double precision, depending on the version of the Surfkin. The precision of the program is listed at the beginning of the output file, in case there is any doubt.

The array Z contains each of the NEQ components of the solution. The first KKSURF elements are the surface site fractions of each surface species in the Surface Chemkin mechanism. The site fractions of the surface species on a given surface phase sum to one. The next KKBULK elements of the solution array contain the net change in mass of each bulk species in the mechanism since time $t = 0$ (described in the previous Section). These entries only make sense for a transient calculation. When the steady state option of Surfkin is selected, all KKBULK entries are set to zero. There are a total of NNSURF surface phases in the mechanism. The final NNSURF entries in the solution array contain the surface site densities (mol/cm^2) for each surface phase. Thus, the length of the solution array is NEQ = KKSURF + KKBULK + NNSURF. The nomenclature of surface site fractions, surface phases, etc., is explained fully in the Surface Chemkin documentation [3].

The first record in the file contains header information. It is generated by the line of code:

```
WRITE (LROUT) ' SURFKIN DATA FILE'
```

and this information is written only once, at the top of the file.

The next record of the file will be one of the three headers listed below. Depending upon the value read, the following several records will contain the information explained in each case.

Several Surfkin simulations may be run back-to-back using the CNTN keyword, described in the next section and illustrated in the second example, given in Sec. VI. At

the start of each simulation, the header information generated by the following lines of code is written:

```
WRITE (LROUT)' HEADER INFORMATION'
WRITE (LROUT) KKGAS, KKSURF, KKBULK, NNSURF
WRITE (LROUT) (GASMOF(K), K=1, KKGAS)
WRITE (LROUT) (SPNAME(K), K = 1, KKTOT)
```

Thus, a post-processor reading its next line as a character variable can check to see if the line contains the string ' HEADER INFORMATION', and if so, should read the information in the next three lines. These contain the numbers of gas, surface, and bulk species and the number of surface phases (first record), the mole fractions of the gas species (second record), and species names (third record). The set of header information is written just once at the start of each separate Surfkin simulation.

In a transient simulation, Surfkin writes the current solution at each output time. The header string below is the first record, indicating which block of information follows. The Fortran statements generating the records are:

```
WRITE (LROUT) ' TIME-DEPENDENT SOLUTION'
WRITE (LROUT) TIME, P, TEMPTR
WRITE (LROUT) (Z(K), K = 1, NEQ)
WRITE (LROUT) (SDOT(K), K = 1, KKTOT)
```

In this case TIME is the current simulation time in seconds, P is the total pressure in dyne/cm^2, and TEMPTR is the current temperature in K. The array Z contains the current values of the NEQ solution components, as explained above. The molar production/destruction rates (mol/cm^2/s) of the KKTOT species are in the array SDOT.

Finally, for a steady state calculation, the following header and records are written:

```
WRITE (LROUT) ' STEADY-STATE SOLUTION'
WRITE (LROUT) TIME, P, TEMPTR
WRITE (LROUT) (Z(K), K = 1, NEQ)
WRITE (LROUT) (SDOT(K), K = 1, KKTOT)
```

In this case, time has no meaning, and is set within the program to an arbitrarily large value. The pressure and surface temperature are in P and TEMPTR, as above. The steady state solution components are in the Z array. The array SDOT contains the steady state molar production/destruction rates of each species.

Compiling and linking Surfkin

The Surfkin user may have access to the source code and need to compile and link it to build an executable program. Three separate Fortran source files, as well as various Chemkin and utility libraries, are required to build Surfkin. The Surfkin code itself is actually a subroutine contained in the source `surfkin.f`. A short main program in `skin_main.f` opens various files, sets aside array space, and calls the Surfkin subroutine. The steady state calculation is done by the "face" subroutine, written by J. F. Grcar (unpublished).

Each Fortran source code needs to be compiled. The object codes are linked with several libraries: `chemkin.a` (Chemkin subroutine library), `surface_chemkin.a` (Surface Chemkin subroutine library), `chemkin_public.a` (containing the VODE integrator) [4]. The details of the compiling and linking the source codes and libraries will vary from one operating system to the next, but in all cases should be similar. An example Unix Makefile to build Surfkin on a HP workstation running the HP/UX operating system is shown below.

Sample Unix Makefile

```
# Makefile for Surfkin
#  (make the Surfkin executables,  store in ../bin)
#
#  usage:
#     make
#   or
#     make surfkin
#-----------------------------------------------------
#  vec and blas are HP math libraries.
#
CHEMKIN= /chemkin
CKLIB= $(CHEMKIN)/lib

# fortran compiler
F77     = f77

all: surfkin

clean:
        rm ../bin/surfkin

#/// surfkin

surfkin: surfkin.o skin_main.o face.o
        $(F77) $(FFLAGS) surfkin.o skin_main.o face.o    \
        $(CKLIB)/surface_chemkin.a $(CKLIB)/chemkin.a    \
        $(CKLIB)/chemkin_public.a  \
        -lblas -lvec -o ../bin/surfkin
        chmod +x ../bin/surfkin

#-----------------rules for object files-----------------

face.o: face.f
        $(F77) $(FFLAGS) -c face.f

skin_main.o: skin_main.f
        $(F77) $(FFLAGS) -c skin_main.f

surfkin.o:  surfkin.f
        $(F77) $(FFLAGS) -c surfkin.f
```

IV. PROGRAM INPUT AND OUTPUT

Input is given to the Surfkin program in keyword format, common to most Chemkin application codes. Each line of the input file contains a single keyword, and the keyword must be the first item on a line. For some keywords only the keyword itself is required, while for others additional information (such as a species name or a number) is needed. Some keywords have default values associated with them, and in such cases the Keyword line is optional. The order of the keyword inputs is generally unimportant, except that if the same keyword is given more than once (with conflicting values), the last one read will be operative. All keywords and associated modifiers are given in upper case. The rules governing the syntax of the keywords are listed below:

1. The first four columns of the line are reserved for the keyword, which must begin in the first column.
2. Any further input associated with the keyword can appear anywhere in columns 5 through 80. The specific column in which the information begins is unimportant.
3. When more than one piece of information is needed for a keyword, the order in which the information appears is important.
4. When numbers are required as input, they may be stated in integer, floating point, or E (scientific) format. The program converts the numbers to the proper type. The double precision specification is not recognized; however, if a double precision version of the program is being run, the double precision conversion is done internally.
5. When gas-phase species names or chemical reactions are given as input, they must appear exactly as they were specified in the Chemkin Interpreter input. Likewise, when surface or bulk species names are used as input, they must appear exactly as they were specified in the Surface Chemkin Interpreter input.
6. When more than one piece of information is given on a line, the pieces are delimited by one or more blank spaces.
7. If more information is input than required, then the last read inputs are used. For example, if the same keyword is encountered twice or if conflicting keywords are given, the last one read is used.
8. A period (.), forward slash (/), or exclamation point (!) in the first column means that particular line will be treated as a comment by Surfkin, that is, it will be ignored. Comment lines are echoed in the output file.
9. Any characters appearing on a line after an exclamation point (!) are considered to be comments and are not processed as keyword input. They are, however, echoed in the printed output file.

List of Keywords

PRES [value]
The total pressure in atmospheres

Default value: None.
PRES is a required keyword.
Example usage:
```
PRES        0.2
```

TEMP [value]
The surface temperature in Kelvin. If the temperature is to be ramped, TEMP is the value at the end of the temperature ramp.

Default value: None.
TEMP is a required keyword.
Example usage:
```
TEMP        723.15
```

TINI [value]
Value of the initial surface temperature (in Kelvin), when the surface temperature is to be ramped.

Default value: Set to TEMP (no temperature ramp).
Example usage:
```
TINI        100
```

RMPI [value]
Initial time of a temperature ramp in sec.

Default value: 0 (if TINI≠ TEMP, temperature ramp begins at = 0).
Example usage:
```
RMPI        1.0E-03
```

RMPF [value]
The final time of the temperature ramp in sec.

Default value: TEND (if TINI≠ TEMP, temperature ramp ends at TEND).
Example usage:
```
RMPF        10.5
```

REAC [gas phase species name] [gas phase mole fraction]
Sets the mole fractions for each of the gas phase species. The species name must appear exactly as it does in the Chemkin input file. This keyword can be used more than once if it is used with different species names. Species mole fractions should

sum to one. However, they are normalized to one by the program regardless of the input values.

Default value: 0 (at least one non-zero REAC value must be given)
Example usage:

```
REAC        H2    0.9
REAC        O2    0.1
```

SURF [surface species name] [Initial ($t = 0$) surface species site fraction]
Sets the initial values for the site fraction of each surface species. For a steady state calculation, the SURF keyword specifies the initial guess for the steady state site fractions to begin the iteration procedure. Surface site fractions should sum to one on each surface phase. However, they are normalized to one by the program regardless of the input values.

Default value: 0 (at least one non-zero SURF value must be given)
Example usage:

```
SURF        H2(S) 0.0
SURF        O2(S) 0.0
SURF        PT(S) 1.0
```

BULK [bulk species name] [mole fraction]
The mole fraction of each bulk species. Bulk mole fractions should sum to one in each bulk phase. However, they are normalized to one by the program regardless of the input values.

Default value: 0 (at least one non-zero BULK value must be given)
Example usage:

```
BULK        ALAS(D)     0.4
BULK        GAAS(D)     0.15
BULK        GAP(D)      0.35
```

TEND [value]
Final time for transient problem in sec.

Default value: None.
TEND is a required keyword, unless NOTD is specified.
Example usage:

```
TEND        10
```

DELT [value]
The solution will be written to the binary output file every DELT sec for a transient simulation. The solution will also be printed to the standard output every NPRT*DELT sec.

Default value: None.
DELT is a required keyword, unless NOTD is specified.

Example usage:

```
DELT        1.E-2
```

NOTD

Do not do a transient calculation.

Default value: NOTD is not in effect, i.e., the transient simulation is done.

Example usage:

```
NOTD
```

SDST

Find the steady state surface site fractions. If the transient simulation is also being done, the steady state calculation will be done afterward.

Default value: Steady state calculation is not performed.

Example usage:

```
TEND  1000.
DT    100
SDST
END
```

RTOL [value]

Sets the relative error tolerance for VODE, the differential equation integrator.

Default value: 1.0E-8

Example usage:

```
RTOL        1.0E-4
```

ATOL [value]

Sets the absolute error tolerance for VODE, the differential equation integrator.

Default value: 1.0E-14

Example usage:

```
ATOL        1.0E-8
```

NPRT [value]

For a transient simulation, the solution is saved to the binary output file every DELT sec. The printed output will be written to the standard output every NPRT*DELT. If the default value of NPRT = 1 is used, then the standard output will be printed each time the binary solution file is written.

Default value: 1

Example usage:

```
NPRT        5
```

LPRT

Print additional, detailed information about chemical reaction rates , i.e., this is an option for a longer printout.

Default value: Longer printout not given.

Example usage:

```
LPRT
```

SINP

The site fractions at the final time or the steady state site fractions should be written to the file `spin.inp`. The format of the output is compatible with other application codes, such as SPIN and AURORA. Thus, Surfkin can be used as a pre-processor to generate a self-consistent set of site-fractions for use as keyword input to SPIN, etc.

Default value: SPIN input file not created.

Example usage:

```
SINP
```

CNTN

Signals that another set of input data will follow. Many continuation data sets (and, thus, sets of simulations) can be entered in a single Surfkin input file. The last set of input data to be processed does not contain the CNTN keyword.

Default value: CNTN not in effect, i.e., no more input following the END

Example usage:

```
(Input keywords for the first simulation)
CNTN          ! indicates another calculation will follow
END           ! input is finished for the current calculation
!(Keywords for the next set of conditions)
CNTN          ! indicates another calculation will follow
END           ! input is finished for the current calculation
!(Keywords for the last calculation)
END           ! CNTN keyword is not included for the last run
```

END

Signals end of the current keyword input.

Default value: None.
The END keyword must be included.

Example usage:

```
END
```

V. Surfkin Example: Temperature Programmed Desorption

This section contains the first of two examples using the Surfkin program. This example problem uses a very simple, one-step, irreversible reaction: adsorbed water desorbing from a platinum surface, which initially is completely covered. The water desorption rate expression is taken from the literature [7]. The temperature of the surface is ramped linearly with time, desorbing the water. Surfkin calculates the surface coverage as a function of time. It also provides production/destruction rates for each species in the mechanism.

A typical temperature programmed desorption (TPD) experiment measures the desorption rate of some species being monitored, as surface temperature undergoes a linear ramp. Desorption rate is often measured with a mass spectrometer. It is possible to extract kinetic rate information by analyzing the desorption rate as a function of time. A very commonly used and simple analysis of TPD is due to Redhead [2]. The Redhead analysis allows one to estimate the activation energy for desorption (E_a) using the surface temperature at which the measured desorption rate is a maximum (T_m), and the surface temperature ramp rate (β)

$$E_a = RT_m\left(\ln\left[\frac{AT_m}{\beta}\right] - 3.64\right) \tag{15}$$

This formula assumes that the reaction pre-exponential constant (A) is known. Usually it is not, although pre-exponential factors for unimolecular reactions are often on the order of a molecular vibrational frequency. In analyzing experimental data, it is common to simply assume a pre-exponential of 10^{13} s^{-1}, and then to use the Redhead equation to estimate E_a.

This example gives sample input files necessary to set-up and run Surfkin, and extract the H_2O desorption rate as a function of time (and surface temperature). It also compares the activation energy specified in the Surface Chemkin reaction mechanism with the activation energy that one would extract from a Redhead analysis of the theoretical desorption curve. As should be the case, the two agree quite well, since the test is rather circular. It does, however, both verify that the kinetic simulation in Surfkin is accurate, and also illustrates that the Redhead analysis is very useful.

The simulation requires three input files—one input file for each of the following: the Chemkin Interpreter, the Surface Chemkin Interpreter, and Surfkin. Two Unix scripts to streamline execution of the problem are also given in this example. The general program flow is illustrated in Fig. 2.

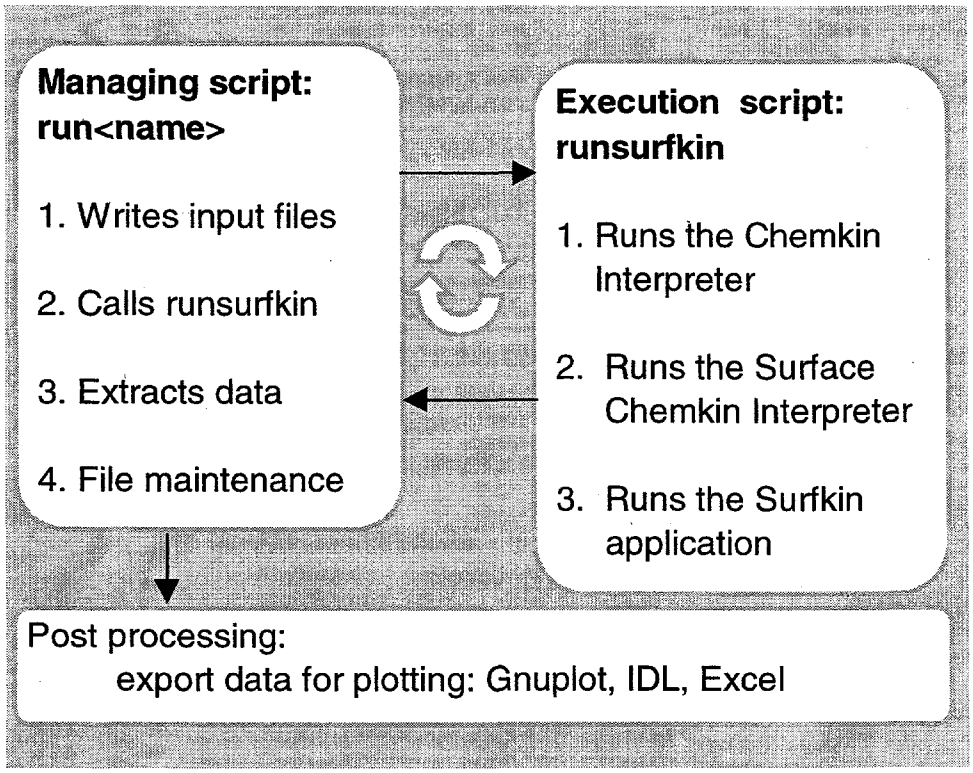


Figure 2. Flow of information and scripts to run the Surfkin example

Input files

Naming input and output files

The user has a great deal of flexibility in naming input and output files for Chemkin simulations. The following table illustrates the naming convention used for this sample simulation. In this example of water desorption, "steam" is used in the place of <name>.

Chemkin input file	<name>.chem
Chemkin output file	<name>.chem.out
Surface Chemkin input file	<name>.surf
Surface Chemkin output file	<name>.surf.out
Surfkin input file	<name>.surfkin
Surfkin output file	<name>.surfkin.out
Binary data files	<name>.dat

The Chemkin input file for this simulation is extremely simple. Chemkin input files can be much more complex, but since Surfkin does not consider gas-phase reactions, none are included in this mechanism. In this case, the Chemkin input file declares only the elements and gas-phase species involved in the mechanism. The thermodynamic data associated with each of these species are read by the Chemkin Interpreter directly from the database file `therm.dat`. Note that, although Pt doesn't appear as a gas-phase species in this system, it must be included as an element in the Chemkin Interpreter file because that information will be subsequently passed to the Surface Chemkin Interpreter. In short, all elements present in the system must be listed in the Chemkin Interpreter input file.

Chemkin Interpreter input file: steam.chem

```
ELEMENTS
H O PT
END
SPECIES
H2O
END
```

Chemkin Interpreter output file: steam.chem.out

```
 CHEMKIN-III GAS-PHASE MECHANISM INTERPRETER:
 DOUBLE PRECISION Vers. 6.17 1999/06/01
 Copyright 1995, Sandia Corporation.
 The U.S. Government retains a limited license in this software.


                          --------------------
                          ELEMENTS     ATOMIC
                          CONSIDERED   WEIGHT
                          --------------------
                           1. H       1.00797
                           2. O       15.9994
                           3. PT      195.090
                          --------------------
 ------------------------------------------------------------------------------
                           C
                         P H
                         H A
                         A R
 SPECIES                 S G  MOLECULAR  TEMPERATURE  ELEMENT COUNT
 CONSIDERED              E E  WEIGHT     LOW    HIGH  H  O  PT
 ------------------------------------------------------------------------------
   1. H2O                G 0   18.01534   300   5000   2  1  0
 ------------------------------------------------------------------------------


 NO ERRORS FOUND ON INPUT:
 ASCII Vers. 1.0 CHEMKIN linkfile chem.asc written.


 WORKING SPACE REQUIREMENTS ARE
    INTEGER:       10
    REAL:          28
    CHARACTER:      4
 Total CPUtime (sec):         1.11E-01
```

The Surface Chemkin input file is somewhat more complicated. The file contains information about the surface sites, surface species, thermodynamic data, and kinetics. The thermodynamic data are not really important in this example, because they are not actually used in any calculations. This is because the only surface reaction is specified to be irreversible. If it had been written as a reversible reaction, the rate constant in the reverse direction would have been based on the equilibrium constant, which in turn would be based on the reaction thermochemistry. In reality, the water desorption/adsorption reaction is reversible. It was written this way to ensure that we could make a simple comparison with the Redhead analysis by isolating only the reaction in the forward direction.

Surface Chemkin Interpreter input file: steam.surf

```
SITE/PT_SURFACE/     SDEN/2.7063E-9/
PT(S)
H2O(S)
END
THERMO ALL
   300.     1000.     3000.
PT(S)             0924910   0H   0PT  1     S   300.000  3000.000             1
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    2
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    3
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00                   4
H2O(S)            0924910   1H   2PT  1     I   300.000  3000.000             1
 0.25803051E+01 0.49570827E-02-0.46894056E-06-0.52633137E-09 0.11998322E-12    2
-0.38302234E+05-0.17406322E+02-0.27651553E+01 0.13315115E-01 0.10127695E-05    3
-0.71820083E-08 0.22813776E-11-0.36398055E+05 0.12098145E+02                   4
END
REACTIONS
 H2O(S)    =>     H2O  +   PT(S)                  1.0E+13   0.00  40300.0
END
```

Surface Chemkin Interpreter output file: steam.surf.out

```
CHEMKIN-III SURFACE MECHANISM INTERPRETER:
DOUBLE PRECISION Vers. 7.13 1999/06/01
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.


CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
DOUBLE PRECISION Vers. 5.18 1999/03/28
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.

--------------------------------------------------------------------------------
SPECIES                 MOLECULAR                          ELEMENT COUNT
CONSIDERED              WEIGHT       Density    Nsites     H  O  PT
--------------------------------------------------------------------------------

Gas phase species:
  1. H2O                 18.01534                          2  1  0

SITE: PT_SURFACE                  .271E-08 moles/cm**2
  2. PT(S)              195.09000                 1        0  0  1
  3. H2O(S)             213.10534                 1        2  1  1

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

  1. H2O(S)=>H2O+PT(S)                              1.00E+13      .0    40300.0

 NOTE:  A units moles, E units cal/mole


NO ERRORS FOUND ON INPUT:
ASCII Version 1.0 surface linkfile surf.asc written.

WORKING SPACE REQUIREMENTS ARE
   INTEGER:      144
   REAL:          87
   CHARACTER:      9
Total CPUtime (sec):        2.34E-02
```

The Surfkin input file contains keywords to control the simulation. For an additional explanation of each keyword used in this file, see Sec. III in this manual on keywords. The example input file tells Surfkin to do the following.

1. Set the initial temperature to 550 K (TINI)
2. At time = 10 seconds start a temperature ramp. (RMPI)
3. Ramp the temperature to reach 650 K at time = 310 seconds.
   (TEMP, TMPF, TEND keywords)
4. Set the pressure to 10^{-6} atmospheres (PRES)
5. Print data at intervals of 1 sec. (DELT)
6. Print out all available information. (LPRT)

Surfkin input file: steam.surfkin

```
LPRT
PRES              1.0E-06
REAC   H2O        1.0
SURF   H2O(S)     1.0
SURF   PT(S)      0.0
TINI              550
TEMP              650
TEND              310
DELT              1
RMPI              10
RMPF              310
END
```

Running the simulation with script files

Once these three input files have been specified, the simulation can be run using the executable commands for each program—the Chemkin Interpreter, Surface Chemkin Interpreter, and Surfkin. The following two files are Unix shell scripts that can be used to run the simulation. Script files generally simplify routine execution of Chemkin applications.

This first script is fairly simple. A more complicated script file that will accept parameters and actually writes the Surfkin input file is illustrated in the example in Section VI. If the program is being run under the Windows or DOS operating systems, one could create batch files to perform the analogous tasks. The first script calls a second script file named runsurfkin, and then extracts data from the output file with a Unix "grep" command. This grep command is included to shorten the amount of information that needs to be sorted. The current Surfkin simulation creates a rather large output file. The output of the grep command is redirected to a file steam.txt.

Script file: runsteam

```
# ! /bin/csh
runsurfkin steam.chem steam.surf steam.surfkin
grep -e "TIME =" -e "PT(S)       =" -e "H2O         =" -e "P = " steam.surfkin.out > \
steam.txt
```

The second script file--called from within the first--is named runsurfkin. This script file can be used on its own, or integrated into other scripts. It requires three arguments: the the input file names for the Chemkin Interpreter, Surface Chemkin Interpreter, and Surfkin, in that order. This script automatically concatenates the file extension '`.out`' to each input file as a name for the respective output files. It also renames the `.dat` file to correspond to the Surfkin output file. The directory name variables should be changed to the path names for the local Chemkin executables.

Second script file: runsurfkin

```
#! /bin/csh
#  This script (runsurfkin) takes three arguments:
#     <name>.chem, <name>.surf, and <name>.surfkin
#     These arguments are run through the chem, surf and surfkin exe's.
#

/bin/echo "beginning runsurfkin script"
/bin/date

##Set directories containing executables
## ***Modify to fit your Chemkin directories***
set CKBIN = '/chemkin/bin'
set LOCALCKBIN = '/home/rrwixom/bin'


# Call the Chemkin Interpreter
/bin/echo "running the Chemkin Interpreter"
"$CKBIN"/chem < $1 > "$1.out"

# Call the Surface Chemkin Interpreter
/bin/echo "running the Surface Chemkin Interpreter"
"$CKBIN"/surf < $2 > "$2.out"

# Run the Surfkin application code
/bin/echo "running the Surfkin application code"
"$LOCALCKBIN"/surfkin < $3 > "$3.out"

#save the binary data under new name
mv -f save.dat "$3.dat"

/bin/date
/bin/echo "Script ran successfully"
```

Execution at the command prompt

These tasks could also be run from the command prompt, instead of using the above scripts, by issuing general commands as:

% "executable" < "Input file" > "Output file"

This command line structure forces the "executable" program to treat standard input as the text file "Input file" and to use another text file "Output file" as standard output. So, to run the three applications would require:

% chem < steam.chem > steam.chem.out
% surf < steam.surf > steam.surf.out
% surfkin < steam.surfkin > steam.surfkin.out

The above commands or the script files generate the output files shown next. If the script files are used there will also be a shortened output file called steam.txt.

Surfkin output file: steam.surfkin.out

The Surfkin output file contains information at each of 310 time intervals. The following shows the output at only a few of these times.

```
                WORKING SPACE REQUIREMENTS
                 PROVIDED        REQUIRED
INTEGER             45000             696
REAL               400000           50328
CHARACTER            5000             314


SURFKIN: Surface kinetic rate equation integrator.
         Assumes fixed pressure and gas composition.
         Substrate temperature is constant, or varied
         linearly with time, or both.
         Version 2.8, July 1999
         DOUBLE PRECISION

CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
DOUBLE PRECISION Vers. 5.18 1999/03/28
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.


SKLIB: CHEMKIN-III SURFACE KINETICS LIBRARY,
DOUBLE PRECISION Vers. 7.13 1999/01/20
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.


          KEYWORD INPUT

         LPRT
         PRES            1.0E-06
         REAC  H2O       1.0
         SURF  H2O(S)    1.0
         SURF  PT(S)     0.0
         TINI            550
         TEMP            650
         TEND            310
         DELT            1
         RMPI            10
         RMPF            310
         END


NOTE: TEMPERATURE WILL BE HELD FIXED AT  550.0 K FOR
0 < T <  .1000E+02 S, THEN RAMPED TO  650.0 K
OVER THE INTERVAL  .1000E+02 < T <  .3100E+03 S,
THEN HELD FIXED AT  650.0 K FOR  .3100E+03 < T <  .3100E+03 S.

=====================================================
TIME =   .8000E+01
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5000E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  7.728E-03    H2O(S)      =  9.923E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:   .2706E-08
```

```
PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.604E-12

PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.604E-12    H2O(S)       = -2.604E-12

RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                          .26042E-11

CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                             PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                          .26042E-11  -.26042E-11
TOTAL (MOL/CM^2/S):                        .26042E-11  -.26042E-11

=====================================================
TIME =    .9000E+01
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5000E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  8.690E-03    H2O(S)       =  9.913E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:    .2706E-08

PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.602E-12

PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.602E-12    H2O(S)       = -2.602E-12

RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                          .26017E-11

CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                             PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                          .26017E-11  -.26017E-11
TOTAL (MOL/CM^2/S):                        .26017E-11  -.26017E-11

=====================================================
TIME =    .1000E+02
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5000E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  9.651E-03    H2O(S)       =  9.903E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:    .2706E-08

PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.599E-12

PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.599E-12    H2O(S)       = -2.599E-12
```

```
RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                          .25992E-11


CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                             PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                          .25992E-11  -.25992E-11
TOTAL (MOL/CM^2/S):                        .25992E-11  -.25992E-11


=====================================================
TIME =    .1100E+02
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5033E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  1.062E-02    H2O(S)       =  9.894E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:   .2706E-08


PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.655E-12


PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.655E-12    H2O(S)       = -2.655E-12


RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                          .26553E-11


CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                             PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                          .26553E-11  -.26553E-11
TOTAL (MOL/CM^2/S):                        .26553E-11  -.26553E-11


=====================================================
TIME =    .1200E+02
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5067E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  1.161E-02    H2O(S)       =  9.884E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:   .2706E-08


PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.712E-12


PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.712E-12    H2O(S)       = -2.712E-12


RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                          .27124E-11


CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                             PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                          .27124E-11  -.27124E-11
```

```
TOTAL (MOL/CM^2/S):                          .27124E-11  -.27124E-11


================================================
TIME =   .1300E+02
 GAS-PHASE STATE
 P =  1.0132E+00       T =  5.5100E+02

 GAS-PHASE MOLE FRACTIONS
  H2O          =  1.000E+00

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  PT(S)        =  1.263E-02    H2O(S)       =  9.874E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:   .2706E-08


PRODUCTION RATES OF GAS SPECIES SPECIES (MOL/CM^2/S):
  H2O          =  2.771E-12


PRODUCTION RATES OF SURFACE & BULK SPECIES (MOL/CM^2/S):
  PT(S)        =  2.771E-12    H2O(S)       = -2.771E-12


RATES OF PROGRESS OF THE SURFACE REACTIONS (MOL/CM^2/S):
H2O(S)=>H2O+PT(S)                            .27707E-11


CONTRIBUTIONS OF SURFACE REACTIONS TO MOLAR PRODUCTION RATES OF THE SPECIES IN PHASE  2

                                              PT(S)        H2O(S)
H2O(S)=>H2O+PT(S)                            .27707E-11  -.27707E-11
TOTAL (MOL/CM^2/S):                          .27707E-11  -.27707E-11


================================================
```

If the script files are run, the following text output is extracted using the "grep" command. Only a portion of the output from the file steam.txt is included here.

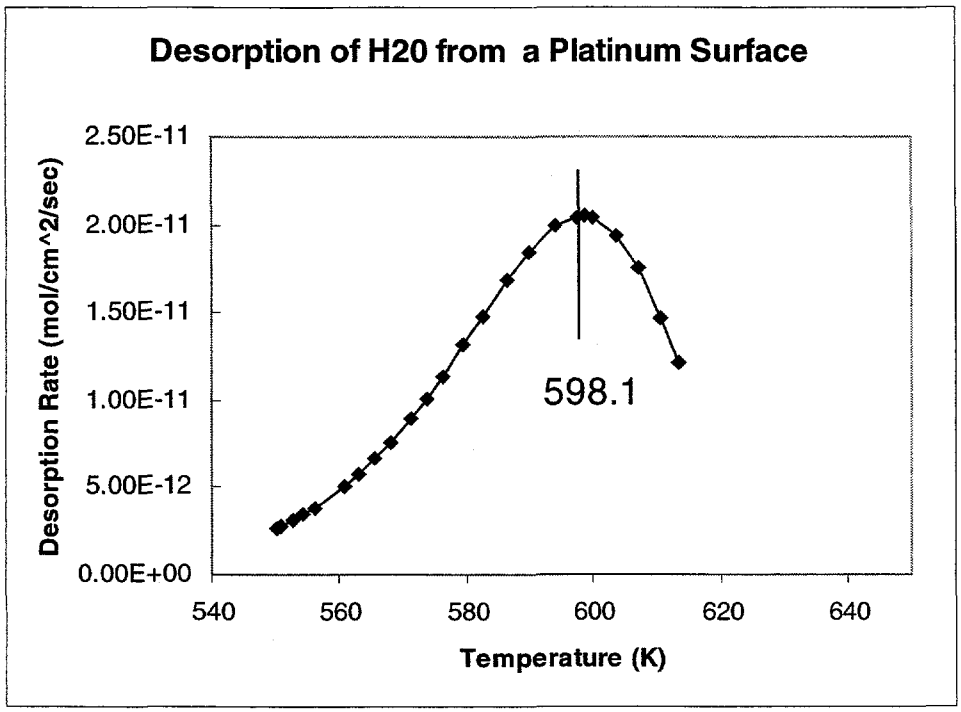
Grep output: steam.txt

```
TIME =   .0000E+00
  P =  1.0132E+00       T =  5.5000E+02
   H2O          =  1.000E+00
   PT(S)        =   .000E+00    H2O(S)       =  1.000E+00
   H2O          =  2.624E-12
   PT(S)        =  2.624E-12    H2O(S)       = -2.624E-12
 TIME =   .1000E+01
  P =  1.0132E+00       T =  5.5000E+02
   H2O          =  1.000E+00
   PT(S)        =  9.693E-04    H2O(S)       =  9.990E-01
   H2O          =  2.622E-12
   PT(S)        =  2.622E-12    H2O(S)       = -2.622E-12
 TIME =   .2000E+01
  P =  1.0132E+00       T =  5.5000E+02
   H2O          =  1.000E+00
   PT(S)        =  1.938E-03    H2O(S)       =  9.981E-01
   H2O          =  2.619E-12
   PT(S)        =  2.619E-12    H2O(S)       = -2.619E-12
 TIME =   .3000E+01
  P =  1.0132E+00       T =  5.5000E+02
   H2O          =  1.000E+00
   PT(S)        =  2.905E-03    H2O(S)       =  9.971E-01
   H2O          =  2.617E-12
   PT(S)        =  2.617E-12    H2O(S)       = -2.617E-12
```

```
TIME =   .4000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  3.872E-03    H2O(S)       =  9.961E-01
  H2O          =  2.614E-12
  PT(S)        =  2.614E-12    H2O(S)       = -2.614E-12
TIME =   .5000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  4.837E-03    H2O(S)       =  9.952E-01
  H2O          =  2.612E-12
  PT(S)        =  2.612E-12    H2O(S)       = -2.612E-12
TIME =   .6000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  5.802E-03    H2O(S)       =  9.942E-01
  H2O          =  2.609E-12
  PT(S)        =  2.609E-12    H2O(S)       = -2.609E-12
TIME =   .7000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  6.765E-03    H2O(S)       =  9.932E-01
  H2O          =  2.607E-12
  PT(S)        =  2.607E-12    H2O(S)       = -2.607E-12
TIME =   .8000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  7.728E-03    H2O(S)       =  9.923E-01
  H2O          =  2.604E-12
  PT(S)        =  2.604E-12    H2O(S)       = -2.604E-12
TIME =   .9000E+01
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  8.690E-03    H2O(S)       =  9.913E-01
  H2O          =  2.602E-12
  PT(S)        =  2.602E-12    H2O(S)       = -2.602E-12
TIME =   .1000E+02
 P =  1.0132E+00       T =  5.5000E+02
  H2O          =  1.000E+00
  PT(S)        =  9.651E-03    H2O(S)       =  9.903E-01
  H2O          =  2.599E-12
  PT(S)        =  2.599E-12    H2O(S)       = -2.599E-12
TIME =   .1100E+02
 P =  1.0132E+00       T =  5.5033E+02
  H2O          =  1.000E+00
  PT(S)        =  1.062E-02    H2O(S)       =  9.894E-01
  H2O          =  2.655E-12
  PT(S)        =  2.655E-12    H2O(S)       = -2.655E-12
TIME =   .1200E+02
 P =  1.0132E+00       T =  5.5067E+02
  H2O          =  1.000E+00
  PT(S)        =  1.161E-02    H2O(S)       =  9.884E-01
  H2O          =  2.712E-12
  PT(S)        =  2.712E-12    H2O(S)       = -2.712E-12
TIME =   .1300E+02
 P =  1.0132E+00       T =  5.5100E+02
  H2O          =  1.000E+00
  PT(S)        =  1.263E-02    H2O(S)       =  9.874E-01
  H2O          =  2.771E-12
  PT(S)        =  2.771E-12    H2O(S)       = -2.771E-12
TIME =   .1400E+02
 P =  1.0132E+00       T =  5.5133E+02
  H2O          =  1.000E+00
  PT(S)        =  1.366E-02    H2O(S)       =  9.863E-01
  H2O          =  2.830E-12
  PT(S)        =  2.830E-12    H2O(S)       = -2.830E-12
TIME =   .1500E+02
 P =  1.0132E+00       T =  5.5167E+02
  H2O          =  1.000E+00
  PT(S)        =  1.472E-02    H2O(S)       =  9.853E-01
  H2O          =  2.891E-12
  PT(S)        =  2.891E-12    H2O(S)       = -2.891E-12
```

Analysis of the desorption rate

If the entire output file is examined, it is easy to determine the temperature corresponding to the peak in the water desorption rate, which was $T_m = 598.1$ K. Using this peak desorption temperature, the ramp rate of $\beta = 0.333$ K/s, and $A = 10^{13}$ as the pre-exponential constant, the Redhead equation yields an activation energy of 40.152 kcal/mol. This is within 0.37% of the specified activation energy of 40.3 kcal/mol. The following graph created in Excel, is a plot of the calculated desorption rate, and illustrates the maximum in the desorption rate.



VI. Surfkin Example: Competitive Adsorption

This section contains a second example using Surfkin. In this illustration, Surfkin will be used to find the steady state site fractions of adsorbed species on the surface. The surface reaction mechanism models the competitive adsorption between hydrogen and oxygen on a platinum surface. This system is simple enough that we can compare the Surfkin output with an analytical solution for the steady state site fractions. This Section first discusses the closed-form solution. The various input files necessary to run this problem are presented, along with the resulting output files. There are also examples of script files that can be used to manage the problem in an efficient manner. The steady state site fractions as a function of the pressure of oxygen gas are calculated and compared to theory.

Theoretical background

Competitive adsorption is a simple, textbook example of heterogeneous reaction kinetics. An adsorption/desorption equilibrium exists between each of the two gases in the system, i.e., H_2 and O_2, and the Pt surface. In addition, the two gas species compete with one another for adsorption sites on the surface. The reversible heterogeneous reaction scheme is

$$H_2(g) + Pt(s) \leftrightarrow H_2(s)$$

$$O_2(g) + Pt(s) \leftrightarrow O_2(s).$$

Let the hydrogen adsorption and desorption rate constants be k_1 and k_{-1}, respectively, and the oxygen adsorption and desorption rate constants be k_2 and k_{-2}.

At steady state, the production and destruction rates of all surface species exactly balance, and so steady site fractions are constant. At steady state:

$$\frac{d[H_2(s)]}{dt} = k_1[H_2(g)][Pt(s)] - k_{-1}[H_2(s)] = 0 \qquad (15)$$

$$\frac{d[O_2(s)]}{dt} = k_2[O_2(g)][Pt(s)] - k_{-2}[O_2(s)] = 0. \qquad (16)$$

If we assume that the maximum surface coverage is one monolayer, then the total concentration of the three surface species must sum to a constant Γ, the total site density (mol/cm^2).

$$[Pt(s)] + [H_2(s)] + [O_2(s)] = \Gamma. \qquad (17)$$

The set of equations (15) – (17) can be solved to find the steady state site fractions.

$$\theta_{Pt} \equiv \frac{[Pt(s)]}{\Gamma} = \frac{1}{1 + K_1[H_2(g)] + K_2[O_2(g)]} \tag{18}$$

$$\theta_{H_2} \equiv \frac{[H_2(s)]}{\Gamma} = \frac{K_1[H_2(g)]}{1 + K_1[H_2(g)] + K_2[O_2(g)]} \tag{19}$$

$$\theta_{O_2} \equiv \frac{[O_2(s)]}{\Gamma} = \frac{K_2[O_2(g)]}{1 + K_1[H_2(g)] + K_2[O_2(g)]}, \tag{20}$$

where $K_i \equiv k_i / k_{-i}$, is the adsorption / desorption equilibrium constant for reaction i.

Input files

The Chemkin and Surface Chemkin Interpreter input files declare the chemical species and reactions in the mechanism. Because Surfkin does not treat gas-phase reactions, the Chemkin Interpreter input file can be quite simple. This input file, named `coadsorb.chem`, is shown below. The output from the Chemkin Interpreter is shown as the file `coadsorb.chem.out`.

Chemkin Interpreter input file – coadsorb.chem

```
ELEMENTS
H PT O
END
SPECIES
H2 O2
END
```

Chemkin Interpreter output file – coadsorb.chem.out

```
CHEMKIN-III GAS-PHASE MECHANISM INTERPRETER:
DOUBLE PRECISION Vers. 6.17 1999/06/01
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.

                          --------------------
                          ELEMENTS     ATOMIC
                          CONSIDERED   WEIGHT
                          --------------------
                           1. H       1.00797
                           2. PT      195.090
                           3. O       15.9994
                          --------------------
-----------------------------------------------------------------------
                          C
                       P  H
                       H  A
                       A  R
SPECIES                S  G  MOLECULAR  TEMPERATURE  ELEMENT COUNT
CONSIDERED             E  E  WEIGHT     LOW    HIGH  H  PT O
-----------------------------------------------------------------------
  1. H2                G  0    2.01594   300   5000   2  0  0
  2. O2                G  0   31.99880   300   5000   0  0  2
-----------------------------------------------------------------------


NO ERRORS FOUND ON INPUT:
ASCII Vers. 1.0 CHEMKIN linkfile chem.asc written.

WORKING SPACE REQUIREMENTS ARE
   INTEGER:        19
   REAL:           50
   CHARACTER:       5
Total CPUtime (sec):        1.13E-01
```

The Surface Chemkin Interpreter input file declares the name of the surface phase, and the three surface species in the mechanism. It lists thermodynamic information for each surface species, as required by the Interpreter. However, because the forward and reverse rate constants for both reactions are explicitly input, this thermodynamic information is not actually used anywhere in the calculation. The forward and reverse rate constants given in this mechanism are meant to be illustrative only. This made-up example should not be taken as a source for hydrogen and oxygen adsorption kinetics or surface thermochemical data.

Surface Chemkin Interpreter input file – coadsorb.surf

```
SITE/PT_SURFACE/     SDEN/1.08E-09/
H2(S)
O2(S)
PT(S)
END
THERMO ALL
   300.      1000.      3000.
PT(S)              092491O   0H    0PT  1      S   300.000  3000.000             1
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    2
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    3
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00                   4
O2(S)               92491O   2PT  1          I    300.00   3000.00 1000.00      1
 0.35989249E+01 0.20437732E-02-0.23878221E-06-0.22041054E-09 0.53299430E-13    2
-0.41095444E+04-0.21604582E+02-0.20174649E+01 0.14146218E-01-0.16376665E-05    3
-0.11264421E-07 0.60101386E-11-0.25084473E+04 0.79811935E+01                   4
H(S)                92491H   1PT  1          I    300.00   3000.00 1000.00      1
 0.10696996E+01 0.15432230E-02-0.15500922E-06-0.16573165E-09 0.38359347E-13    2
-0.50546128E+04-0.71555238E+01-0.13029877E+01 0.54173199E-02 0.31277972E-06    3
-0.32328533E-08 0.11362820E-11-0.42277075E+04 0.58743238E+01                   4
H2(S)              092491O   0H   2PT  1      I   300.000  3000.000             1
 0.15330955E+01 0.34586885E-02-0.32622225E-06-0.36824219E-09 0.83855205E-13    2
-0.36401533E+04-0.10822206E+02-0.21517782E+01 0.87039210E-02 0.11154106E-05    3
-0.42477102E-08 0.96133203E-12-0.22640681E+04 0.97397461E+01                   4
END
!!*****************************************************************************
!!********          H2 / O2 Co-Adsorption on Pt                    ***********
!!*****************************************************************************
REACTIONS         MWOFF MOLES KJOULES/MOLE
!    Rate Constant Corresponds to sticking coefficient of 0.023 at 1000K
 H2 + PT(S) = H2(S)                          1.186132E11   0.5    0.0
              REV/7.2749E+14 -2.86E-03 21.811/
 O2 + PT(S) = O2(S)                          1.3854E10     0.5    0.0
              REV/1.5728E+20 -0.9259   23.489/             ! Ljungstroem 89
END
```

Surface Chemkin Interpreter output file – coadsorb.surf.out

```
CHEMKIN-III SURFACE MECHANISM INTERPRETER:
DOUBLE PRECISION Vers. 7.13 1999/06/01
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.


CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
DOUBLE PRECISION Vers. 5.18 1999/03/28
Copyright 1995, Sandia Corporation.
The U.S. Government retains a limited license in this software.

-------------------------------------------------------------------------------
SPECIES                 MOLECULAR                          ELEMENT COUNT
CONSIDERED              WEIGHT       Density    Nsites     H  PT O
-------------------------------------------------------------------------------

Gas phase species:
  1. H2                   2.01594                          2  0  0
  2. O2                  31.99880                          0  0  2

SITE: PT_SURFACE                  .108E-08 moles/cm**2
  3. H2(S)              197.10594                 1        2  1  0
  4. O2(S)              227.08880                 1        0  1  2
  5. PT(S)              195.09000                 1        0  1  0

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

  1. H2+PT(S)<=>H2(S)                                 1.19E+11      .5        .0
     Reverse Arrhenius coefficients:                  7.27E+14      .0      21.8
  2. O2+PT(S)<=>O2(S)                                 1.39E+10      .5        .0
     Reverse Arrhenius coefficients:                  1.57E+20     -.9      23.5

 NOTE:  A units moles, E units Kjoule/mole


NO ERRORS FOUND ON INPUT:
ASCII Version 1.0 surface linkfile surf.asc written.

WORKING SPACE REQUIREMENTS ARE
   INTEGER:      188
   REAL:         150
   CHARACTER:     11
Total CPUtime (sec):        3.13E-02
```

This Surfkin example sets up the input to examine the steady state surface site fractions as a function of the H_2 and O_2 mole fractions in the gas, keeping the total pressure fixed. The Surfkin keyword CNTN (continuation) allows a series of calculations to be run back-to-back as the input mole fractions are varied. The NOTD keyword instructs Surfkin not to perform the transient calculation; the SDST keyword requests the steady state option. The high pressure (100 atm) and low surface temperature (200 K) were contrived simply to generate higher surface coverage and an interesting plot of the results.

Surfkin input file – coadsorb.surfkin

```
PRES                    100.0
TEMP                    200.0
REAC    H2              0.9
REAC    O2              0.1
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.8
REAC    O2              0.2
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.7
REAC    O2              0.3
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.6
REAC    O2              0.4
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.5
REAC    O2              0.5
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
```

```
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.4
REAC    O2              0.6
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.3
REAC    O2              0.7
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.2
REAC    O2              0.8
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
CNTN
END
!
PRES                    100.0
TEMP                    200.0
REAC    H2              0.1
REAC    O2              0.9
SURF    PT(S)           1.0
SURF    H2(S)           0.0
SURF    O2(S)           0.0
SDST
NOTD
END
```

Notice that the last set of input data does not contain the CNTN keyword, indicating that no more calculations are to follow.

Running the simulation

The input files and output files can have any name, but we illustrate a self-consistent naming convention. For this problem the Chemkin Interpreter input and output files are named `coadsorb.chem` and `coadsorb.chem.out`, respectively. The Surface Chemkin Interpreter input and output files are named `coadsorb.surf` and `coadsorb.surf.out`, respectively. The Surfkin input and output files are named `coadsorb.surfkin` and `coadsorb.surfkin.out`.

Once the three required input files have been created, the following commands will execute the Chemkin Interpreter, Surface Chemkin Interpreter, and Surfkin. The programs can be executed from a command line prompt, or from a script, as discussed in the last Section.

An example of running a program from the command line prompt is:

% "executable" < "Input file" > "Output file"

So, to run the three applications would require:

% chem < coadsorb.chem > coadsorb.chem.out
% surf < coadsorb.surf > coadsorb.surf.out
% surfkin < coadsorb.surfkin > coadsorb.surfkin.out

Running the simulation with script files

Script files can be a convenient and efficient method to set-up input files and to run Surfkin and the associated Interpreters. The runsurfkin script was already given in the example discussed in the previous Section. That script requires three arguments--the name of the Chemkin Interpreter input file, the name of the Surface Chemkin Interpreter input file and the name of the Surfkin input file. These must be entered in order.

Example of the runsurfkin script execution:

% runsurfkin coadsorb.chem coadsorb.surf coadsorb.surfkin

The script "runcoad," shown below is also useful. This script file was written specifically to set up the input for the current sample problem, but can be easily modified to fit other examples. The script, runcoad, will accept an arbitrarily long list of numbers as arguments. These numbers represent mole fractions of O_2 in the gas phase. However, because Unix script files cannot perform floating-point calculations, the mole fractions are entered as integers in the range one to ten. For example, a mole fraction of 0.2 is input as the integer 2. The runcoad script writes a Surfkin input file, and then calls the runsurfkin script to execute the Interpreters and Surfkin. It then rewrites the input file

with the next mole fraction, and re-runs the runsurfkin script. It does this for each mole fraction in the argument list.

Example of runcoad script execution:

```
% runcoad 1 2 3 4 5 6 7 8 9
```

The Unix "grep" command extracts lines of interest from the Surfkin output file, and places them in the file coadsorb.txt. This text file is much smaller than the long output file, and is readily exported for plotting.

Script file: runcoad

```
#! /bin/csh
#
#     Ryan Wixom 7-99
#
#     To execute: runcoad [ ** integer list of O2 gas phase mole
#                              fractions]
#
#               ** Because shell scripts do not support
#                  decimal numbers, the mole fractions
#                  are represented as integers between
#                  1 and 10. (10 being a mole fraction of 1.0)
#
#     This script will extract the lines that I am
#     interested in, using the grep command.
#     Then I can export the .txt file to a plotting program.
#####################################################################

#remove old coadsorb.txt
rm coadsorb.txt

#loop to handle each mole fraction argument
foreach O2gasFrac ($argv[*])

      rm coadsorb.surfkin
      rm coadsorb.surfkin.out

      #calculate H2 fraction,
      @ total = 10
      @ H2gasFrac = 0
      @ H2gasFrac = $total - $O2gasFrac

      #write input file for each argument
      echo "PRES            100" >> coadsorb.surfkin
      echo "TEMP            200" >> coadsorb.surfkin
      echo "REAC  H2       0."$H2gasFrac >> coadsorb.surfkin
      echo "REAC  O2       0."$O2gasFrac >> coadsorb.surfkin
      echo "SURF  PT(S)     1.0" >> coadsorb.surfkin
      echo "SURF  H2(S)     0.0" >> coadsorb.surfkin
      echo "SURF  O2(S)     0.0" >> coadsorb.surfkin
      echo "SDST" >> coadsorb.surfkin
      echo "NOTD" >> coadsorb.surfkin
      echo "END" >> coadsorb.surfkin
```

```
      #writes mol fraction to output text file
      echo "O2 gas-phase mol fraction: 0."$O2gasFrac >> coadsorb.txt

      #runs runsrufkin script (exe's) for each input file
      runsurfkin coadsorb.chem coadsorb.surf coadsorb.surfkin

      #extracts needed lines from output file and
      #directs these lines to the coadsorb.txt file
      grep -e "TIME =   " -e "P = " -e "H2           = " \
       -e "H2(S)        ="    coadsorb.surfkin.out >> coadsorb.txt
end
```

The following is a portion of the Surfkin output file `coadsorb.surfkin.out` generated by the above input. Because the output is quite long, only the output from one of the data sets is shown. The full output file contains similar information for each set of input mole fractions.

Surfkin output file: coadsorb.surfkin.out

(output from first set of input only)

```
                WORKING SPACE REQUIREMENTS
                 PROVIDED        REQUIRED
INTEGER             45000             749
REAL               400000           50413
CHARACTER            5000             317


 SURFKIN: Surface kinetic rate equation integrator.
          Assumes fixed pressure and gas composition.
          Substrate temperature is constant, or varied
          linearly with time, or both.
          Version 2.8, July 1999
          DOUBLE PRECISION

 CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
 DOUBLE PRECISION Vers. 5.18 1999/03/28
 Copyright 1995, Sandia Corporation.
 The U.S. Government retains a limited license in this software.

 SKLIB: CHEMKIN-III SURFACE KINETICS LIBRARY,
 DOUBLE PRECISION Vers. 7.13 1999/01/20
 Copyright 1995, Sandia Corporation.
 The U.S. Government retains a limited license in this software.


           KEYWORD INPUT

          PRES                    100.0
          TEMP                    200.0
          REAC   H2               0.9
          REAC   O2               0.1
          SURF   PT(S)            1.0
          SURF   H2(S)            0.0
          SURF   O2(S)            0.0
          SDST
          NOTD
          CNTN
          END


 &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

```
 RESULTS FROM STEADY-STATE ANALYSIS:


================================================
TIME =   .1000E+31
 GAS-PHASE STATE
 P =  1.0133E+08       T =  2.0000E+02

 GAS-PHASE MOLE FRACTIONS
  H2          =  9.000E-01    O2          =  1.000E-01

 SURFACE SITE FRACTIONS ON PHASE (SITE)  2
  H2(S)       =  8.645E-01    O2(S)       =  1.894E-05    PT(S)       =  1.355E-01

SUM OF SURFACE SITE FRACTIONS: 1.000000
SURFACE SITE DENSITY:   .1080E-08
```

Script file output: coadsorb.txt

The runcoad script creates a text file called coadsorb.txt. This file was exported to Excel and the data of interest was plotted. The following text contains an example of coadsorb.txt and a plot created from the data. The results of this simulation are in perfect agreement with the theoretical result shown at the beginning of this section.

```
O2 gas-phase mol fraction: 0.1
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  9.000E-01    O2          =  1.000E-01
   H2(S)       =  8.645E-01    O2(S)       =  1.894E-05    PT(S)       =
1.355E-01
O2 gas-phase mol fraction: 0.2
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  8.000E-01    O2          =  2.000E-01
   H2(S)       =  8.501E-01    O2(S)       =  4.190E-05    PT(S)       =
1.499E-01
O2 gas-phase mol fraction: 0.3
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  7.000E-01    O2          =  3.000E-01
   H2(S)       =  8.322E-01    O2(S)       =  7.031E-05    PT(S)       =
1.677E-01
O2 gas-phase mol fraction: 0.4
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  6.000E-01    O2          =  4.000E-01
   H2(S)       =  8.096E-01    O2(S)       =  1.064E-04    PT(S)       =
1.903E-01
O2 gas-phase mol fraction: 0.5
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  5.000E-01    O2          =  5.000E-01
   H2(S)       =  7.798E-01    O2(S)       =  1.537E-04    PT(S)       =
2.200E-01
O2 gas-phase mol fraction: 0.6
 TIME =   .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  4.000E-01    O2          =  6.000E-01
```

```
   H2(S)        =  7.391E-01    O2(S)        =  2.186E-04    PT(S)        =
2.607E-01
O2 gas-phase mol fraction: 0.7
 TIME =    .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  3.000E-01    O2           =  7.000E-01
   H2(S)       =  6.799E-01    O2(S)        =  3.128E-04    PT(S)        =
3.197E-01
O2 gas-phase mol fraction: 0.8
 TIME =    .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  2.000E-01    O2           =  8.000E-01
   H2(S)       =  5.861E-01    O2(S)        =  4.622E-04    PT(S)        =
4.134E-01
O2 gas-phase mol fraction: 0.9
 TIME =    .1000E+31
  P =  1.0133E+08       T =  2.0000E+02
   H2          =  1.000E-01    O2           =  9.000E-01
   H2(S)       =  4.145E-01    O2(S)        =  7.354E-04    PT(S)        =
5.848E-01
```

Plot of the data in coadsorb.txt

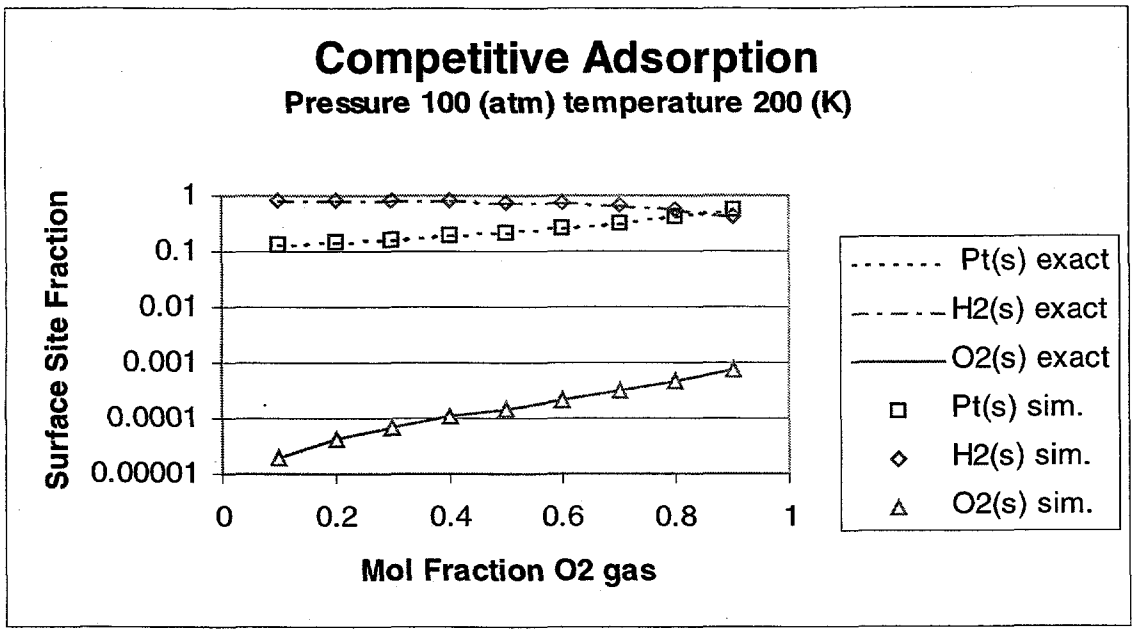


Figure 4. Comparison of steady state site fractions calculated by Surfkin and the exact theoretical result.

REFERENCES

1. R.J. Kee, F.M. Rupley, J.A. Miller, M.E. Coltrin, J.F. Grcar, E. Meeks, H.K. Moffat, A.E. Lutz, G. Dixon-Lewis, M.D. Smooke, J. Warnatz, G.H. Evans, R.S. Larson, R.E. Mitchell, L.R. Petzold, W.C. Reynolds, M. Caracotsios, W.E. Stewart, and P. Glarborg, *Chemkin Collection,* Reaction Design, Inc., San Diego, CA (1997).
2. P.A. Redhead, Vacuum, **12**, 203 (1962).
3. M.E. Coltrin, R.J. Kee, F.M. Rupley, and E. Meeks, *SURFACE CHEMKIN III: A Fortran Package for Analyzing Heterogeneous Chemical Kinetics at a Solid-Surface - Gas-Phase Interface,* Sandia National Laboratories (1996).
4. P.N. Brown, G.D. Byrne, and A.C. Hindmarsh, SIAM J. Sci. Stat. Comput., **10**, 1038-1051 (1989).
5. J.F. Grcar, *The Twopnt Program for Boundary Value Problems,* Sandia National Laboratories (1991).
6. R.J. Kee, F.M. Rupley, and E. Meeks, *Chemkin-III: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase and Plasma Kinetics,* Sandia National Laboratories (1996).
7. O. Deutschmann, R. Schmidt, F. Behrendt, and J. Warnatz, in *Twenty-sixth Symposium (International) on Combustion*: The Combustion Institute, Pittsburgh, PA (1996).

UNLIMITED RELEASE

DISTRIBUTION:

10		D. S. Dandy Chemical and Bioresource Engineering Colorado State University Ft. Collins, CO 80523-1370
1		E. Egan Reaction Design 6440 Lusk Boulevard Suite D-209 San Diego, CA 92121
1		R. J. Kee Division of Engineering Colorado School of Mines Golden, CO 80401
1		D. H. Klipstein Reaction Design 6440 Lusk Boulevard Suite D-209 San Diego, CA 92121
1		E. Meeks Reaction Design 6440 Lusk Boulevard Suite D-209 San Diego, CA 92121
1		F. M. Rupley Reaction Design 6440 Lusk Boulevard Suite D-209 San Diego, CA 92121
10		R. R. Wixom Materials Science and Engineering 122 S. Central Campus Dr. Room 304 Salt Lake City, UT 84112-0560
1	1100	S. T. Picraux, MS0360
1	1126	M. E. Bartram, MS0601
1	1126	W. G. Breiland, MS0601
35	1126	M. E. Coltrin, MS0601
1	1126	J. R. Creighton, MS0601
1	1126	J. Han, MS0601
1	1126	P. Ho, MS0601
1	1126	H. K. Moffat, MS0601

1	1126	C. C. Willan, MS0601
1	8356	J. F. Grcar, MS9051
1	8362	A. E. Lutz, MS9051
1	8728	G. H. Evans, MS9042
1	8728	W. G. Houf, MS9042
1	8728	C. D. Moen, MS9042
1	8728	M. W. Perra, MS9042
1	8728	P. A. Spense, MS9042
1	8728	A. Ting, MS9042
1	8728	W. S. Winters, MS9042
1	9111	J. E. Johannes, MS0834
2	9616	Technical Library, MS0899
1	9612	Review & Approval Desk, MS0612
1	8940-2	Central Technical Files, MS9018