UC-4

SAND87-8248 Revised
Unlimited Release
Printed November 1997
Supercedes SAND87-8248,
dated February 1988

SENKIN: A FORTRAN PROGRAM FOR PREDICTING HOMOGENEOUS GAS PHASE CHEMICAL KINETICS WITH SENSITIVITY ANALYSIS

Andrew E. Lutz, Robert J. Kee, and James A. Miller
MS-9051
Sandia National Laboratories, Livermore
Sandia National Laboratories
Livermore, CA 94551-0969

ABSTRACT

Senkin is a Fortran computer program that computes the time evolution of a homogeneous reacting gas mixture in a closed system. The model accounts for finite-rate elementary chemical reactions, and performs kinetic sensitivity analysis with respect to the reaction rates. The program considers six problem types:

A. an adiabatic system with constant pressure,
B. an adiabatic system with constant volume,
C. an adiabatic system with the volume a specified function of time,
D. a system where the pressure and temperature are constant, and
E. a system where the volume and temperature are constant, and
F. a system where the pressure and temperature are specified functions of time.

The program uses the DASAC software to solve both the nonlinear ordinary differential equations that describe the temperature and species mass fractions and the set of linear differential equations that describe the first-order sensitivity coefficients of temperature and species composition with respect to the individual reaction rates. The program runs in conjunction with the Chemkin package, which handles the chemical reaction mechanism.

SENKIN: A FORTRAN PROGRAM FOR PREDICTING HOMOGENEOUS GAS PHASE CHEMICAL KINETICS WITH SENSITIVITY ANALYSIS

INTRODUCTION

Senkin is a Fortran computer program that predicts the time-dependent chemical kinetics behavior of a homogeneous gas mixture in a closed system. In addition to predicting the species and temperature histories, the program can also compute the first-order sensitivity coefficients with respect to the elementary reaction rate parameters. Because of the improved sensitivity analysis capability, this program now supersedes our previous sensitivity analysis programs, CHEMSEN[1] and AIM.[2]

In nearly all models, the solutions depend both on initial and boundary conditions and on certain parameters that go into defining the model itself. Often the modeler is uncertain of the values of some of the parameters and thus he would like to know how sensitive the results of the model are to these parameters. Sensitivity analysis is a formal procedure to determine quantitatively how the solution to a model depends on certain parameters in the model formulation. In the cases considered here, the parameters are the elementary reaction rate constants. Thus, without solving the problem repetitively with different values for the rate constants, the sensitivity analysis allows one to understand how the model will respond to changes in the rate parameters. It also provides insight about how important certain reaction pathways are to the model's predictions.

Compared to repetitive running of the model, the sensitivity analysis is significantly more efficient. This relative efficiency increases as the number of parameters increases. The reason for the computational efficiency comes from taking advantage of the fact that the equations describing the sensitivity coefficients are linear, regardless of the nonlinearities in the model itself. Furthermore, when the model is solved by the implicit multistep methods that are especially efficient for stiff chemical kinetics problems, solution of the sensitivity equations can take effective advantage of information that is already available from solution of the model.

There are many possibilities for the chemical kinetics problems that one may need to solve for various applications. In this program we have chosen six problem types that we find useful. The distinction between the problems comes from the externally imposed thermodynamic conditions. The five problems are:

A. an adiabatic system with constant pressure,
B. an adiabatic system with constant volume,

C. an adiabatic system with the volume a specified function of time,
D. a system where the pressure and temperature are constant,
E. a system where the volume and temperature are constant, and
F. a system where the pressure and temperature is specified functions of time.

For our combustion applications, conditions A through C generally apply to spontaneous ignition problems. These options may find use for modeling combustion bombs, rapid compression machines, and perhaps reflected-wave shock tubes. The assumptions of constant pressure (case A) and constant volume (case B) are limiting conditions for a fixed mass of mixture that is reacting in an adiabatic system. For a closed, adiabatic system at constant pressure, the mixture is free to expand and the enthalpy of the system is constant. For a closed, adiabatic system with constant volume, no expansion work can be done on the surroundings and the internal energy of the mixture is constant. Case C considers a time-varying volume, and is intended for use in modeling configurations such as rapid compression machines.

Another general type of homogeneous kinetics problem is considered by problem types D and E. In these problems, heat release is not important, because either the reaction mechanism is not strongly exothermic, or the mixture contains such a large fraction of a diluent that the heat released per mass of mixture is relatively small. In these cases, the pressure or volume are held constant and the the energy equation is replaced by the condition that the temperature is known. Case F allows computation of mixture composition histories for specified temperature and pressure histories. This capability is useful for modeling situations such as a plug flow reactor, where the temperature history can be measured accurately.

The computational solution is accomplished by a code called DASAC which was written by Caracotsios and Stewart.[3] The software is a modification and extension of Petzold's differential/algebraic equation solver called DASSL.[4] DASAC handles the solution of the governing differential equations together with an efficient simultaneous computation of the first-order sensitivity coefficients. The numerical method is based on the backwards differentiation formulas and is especially well suited for solving the stiff equations that are common in chemical kinetics applications.

The following sections describe the governing equations for each of the cases handled by the program and provide instructions on how to operate the code. In the last section, an example problem illustrates the capabilities of the program.

GOVERNING EQUATIONS

In this section, the equations for mass and energy conservation are described for the five problem types considered by the program. The reacting mixture is treated as a closed system with no mass crossing the boundary, so the total mass of the mixture $m = \sum_{k=1}^{K} m_k$ is constant, and $dm/dt = 0$. Here m_k is the mass of the kth species and K is the total number of species in the mixture. The individual species are produced or destroyed according to

$$\frac{dm_k}{dt} = V\dot{\omega}_k W_k \qquad k = 1, \ldots, K \tag{1}$$

where t is time, $\dot{\omega}_k$ is the molar production rate of the kth species by elementary reaction, W_k is the molecular weight of the kth species, and V is the volume of the system, which may vary in time. Since the total mass is constant, this can be written in terms of the mass fractions as

$$\frac{dY_k}{dt} = v\dot{\omega}_k W_k \qquad k = 1, \ldots, K \tag{2}$$

where $Y_k = m_k/m$ is the mass fraction of the kth species and $v = V/m$ is the specific volume. The species equations (2) are the same in all cases, A through F. For cases D through F, the temperature is known, so the energy equation is unnecessary and the problem is completely defined by equations (2). For cases A through C, the energy equation must be derived in light of the specific constraints used in each case.

The first law of thermodynamics for a pure substance in an adiabatic, closed system states that

$$de + pdv = 0, \tag{3}$$

where e is the internal energy per mass and p is the pressure. This relation holds for an ideal mixture of gases, with the internal energy of the mixture given by

$$e = \sum_{k=1}^{K} e_k Y_k, \tag{4}$$

where e_k is the internal energy of the kth species. Differentiating the internal energy of the mixture leads to the expression

$$de = \sum_{k=1}^{K} Y_k de_k + \sum_{k=1}^{K} e_k dY_k. \tag{5}$$

Assuming calorically perfect gases, we write $de_k = c_{\mathrm{v},k} dT$, where T is the temperature of the mixture, and $c_{\mathrm{v},k}$ is the specific heat of the kth species evaluated at constant volume. Defining the mean specific heat of the mixture, $c_{\mathrm{v}} = \sum_{k=1}^{K} Y_k c_{\mathrm{v},k}$ and differentiating with respect to time, the energy equation becomes

$$c_{\mathrm{v}}\frac{dT}{dt} + \sum_{k=1}^{K} e_k \frac{dY_k}{dt} + p\frac{dv}{dt} = 0. \tag{6}$$

Substitution of equation (1) for the species production rate gives

$$c_{\mathrm{v}}\frac{dT}{dt}+p\frac{dv}{dt}+v\sum_{k=1}^{K}e_k\dot{\omega}_kW_k=0, \tag{7}$$

where $c_{\mathrm{v}}=\sum_{k=1}^{K}Y_kc_{\mathrm{v},k}$. The ideal gas equation of state is used to compute the pressure,

$$p=\frac{\rho RT}{\overline{W}}, \tag{8}$$

where R is the universal gas constant, $\overline{W}$ is the mean molecular weight of the mixture, and ρ is the mass density. In case C, we presume that the volume is provided as a function of time, so the specific volume and its rate of change are

$$v(t)=\frac{V(t)}{m} \tag{9}$$

and

$$\frac{dv}{dt}=\frac{1}{m}\frac{dV}{dt}. \tag{10}$$

The system of equations for case C consists of equation (7) for the energy, and the K equations (2) for the species mass fractions. In case B, the volume is held constant, so equation (7) reduces to

$$c_{\mathrm{v}}\frac{dT}{dt}+v\sum_{k=1}^{K}e_k\dot{\omega}_kW_k=0. \tag{11}$$

In case A, the first law of thermodynamics reduces to the condition that the enthalpy of the mixture is constant. The definition of enthalpy is, $h=e+pv$, which differentiated becomes

$$dh=de+vdp+pdv. \tag{12}$$

The pressure is constant, so the term involving dp drops out and the first law (equation 3) simplifies to the condition

$$dh=0. \tag{13}$$

The mixture enthalpy is

$$h=\sum_{k=1}^{K}Y_kh_k \tag{14}$$

where h_k is the specific enthalpy of the kth species. Proceeding as before, the energy equation for the constant pressure case becomes

$$c_{\mathrm{p}}\frac{dT}{dt}+v\sum_{k=1}^{K}h_k\dot{\omega}_kW_k=0, \tag{15}$$

where the mean specific heat of the mixture is $c_{\mathrm{p}} = \sum_{k=1}^{K} Y_k c_{\mathrm{p},k}$. The system of equations for case A consists of equation (15) for the energy, and the K equations (2) for the species mass fractions.

The net chemical production rate $\dot{\omega}_k$ of each species results from a competition between all the chemical reactions involving that species. Each reaction proceeds according to the law of mass action and the forward rate coefficients are in the modified Arrhenius form

$$k_f = AT^{\beta} \mathrm{exp}\left(\frac{-E}{RT}\right), \tag{16}$$

where the activation energy E, the temperature exponent β, and the pre-exponential constants A are parameters in the model formulation. The details of the chemical reaction equations and the thermochemical properties are found in the user's manual for Chemkin.[5]

The initial value problem for each of the different cases formulated above requires initial conditions for the temperature, pressure, and composition of the mixture. The initial density is computed from the equation of state. These are intensive variables, so the problem is independent of the absolute quantity of mixture in question. However, case C requires input of the system volume $V(t)$, which is an extensive variable. This forces the computation of another extensive variable, namely the mass of mixture, which is a constant during the solution. So in case C, the mass is computed from the initial density and volume, $m = \rho(0)V(0)$.

In many problems considered by case C, it is more convenient to specify the volume function of time in terms of a normalized volume. This is appropriate, because the solution really depends only on the density; the mass is just a constant that converts volume to density. If the volume is given in terms of a normalized function $\hat{V}(t)$, then the computed mass is just a different constant, say $\hat{m}$. The meaning and units of the constant $\hat{m}$ come from the normalization. For example, the user could supply the volume normalized by the initial volume, so that the progam sees a function $\hat{V}(t) = V(t)/V(0)$, which is one at time zero. The computed constant is then $\hat{m} = \rho(0)$, which has units of density, rather than mass. In summary, the user–supplied volume function can have any units or normalization; the computed mass will just be a constant other than the actual mass.

NUMERICAL SOLUTION METHOD

The system of ordinary differential equations described in the previous section is generally stiff, and thus is most efficiently solved by implicit techniques. A software package called DASAC[3] (Differential Algebraic Sensitivity Analysis Code) performs the time integration and first-order sensitivity analysis. The DASAC package is based on the differential/algebraic system solver DASSL[4], which performs the time integration using a backward differentiation formula (BDF). These BDF methods are in regular use for solving a wide range of stiff problems, including chemical kinetics problems. The notions of stiffness and implicit numerical methods are described elsewhere (see, for example, Kee, et al.[6]). The details of the numerical methods in DASSL are described by Petzold,[4] and the DASAC enhancements by Caracotsios and Stewart[3]. Therefore, in this document we only briefly outline some of the central features of the sensitivity methods.

Sensitivity Analysis

Sensitivity analysis is a powerful and systematic way to determine quantitatively the relationship between the solution to a model and the various parameters that appear in the model's definition. The review articles by Tilden, et al.[7] and Rabitz[8] are good surveys of sensitivity analysis; they discuss numerical methods and provide some insight on the application and value of such analysis. The value of sensitivity analysis was beginning to be appreciated in the late 1970s by several groups who began developing numerical methods to efficiently compute the sensitivity coefficients. Among these are the "Direct Method" by Dickinson and Gelinas,[9] the "Green's Function" method by Rabitz and coworkers[10] and the "Fourier Amplitude Sensitivity Test (FAST)" method by Cukier, et al.[11]

In 1981 Kramer et al.[2,12] improved the Green's function method with an approach they called the AIM method. Following that, at Sandia we developed a production code called CHEMSEN[1,13] that combined the Chemkin[5] chemical kinetics software with the AIM[2] software to model isothermal constant-volume chemical kinetics systems.

All the computationally efficient sensitivity analysis methods exploit the fact that the differential equations describing the sensitivity coefficients are linear, regardless of any nonlinearities in the model problem itself. Furthermore, some of the methods, including the one used here, take advantage of the fact that the sensitivity equations are described in terms of the Jacobian of the model problem. In stiff ODE software, such as DASSL[4] or LSODE,[14] which are based on BDF methods, the Jacobian is required for solution of the model problem, so it is available for the sensitivity computation.

Recently three groups recognized the efficiencies that can be achieved by implementing sensitivity analysis in conjunction with BDF-based software. Dunker's[15] implementation was for systems of ODEs using LSODE and Leis and Kramer's[16] was based on LSODI[14] for resticted systems of differential/algebraic equations. Caracotsios and Stewart's[17] implementation is for general systems of differential/algebraic equations (index less than one), but their work is based on DASSL. Their implementation, called DASAC, is a very well written and robust piece of production software. Therefore, we chose it for the current chemical kinetics code.

The system of ordinary differential equations that describe the physical problem are of the general form

$$\frac{dZ}{dt} = F(Z, t; a), \tag{17}$$

where, in our case, $Z = (T, Y_1, Y_2, \ldots, Y_K)^{\mathrm{t}}$ is the vector of temperature and mass fractions.

In versions of Senkin prior to version 1.8, the parameter vector a represents the pre-exponential constants for each of the elementary reactions: A in equation (16). However, since Chemkin-II[5b] added rate formulations for pressure-dependent reactions, it was necessary to include an extra parameter in the rate expression to allow perturbation of the rate. This extra parameter is normally one ($a_i = 1$), except when DASAC perturbs it in computing the sensitivity solution.

The first-order sensitivity coefficient matrix is defined as

$$w_{l,i} = \frac{\partial Z_l}{\partial a_i}, \tag{18}$$

where the indicies l and i refer to the dependent variables and reactions, respectively. Differentiating equation (17) with respect to the parameters a_i yields

$$\frac{dw_{l,i}}{dt} = \frac{\partial F}{\partial Z} \cdot w_{l,i} + \frac{\partial F_l}{\partial a_i} \tag{19}$$

Note that this equation for the sensitivity coefficients is linear, even though the model problem may be nonlinear. Of course, when coupled with a nonlinear model problem, the whole system is still nonlinear. Nevertheless, when solved via the same BDF method as the model problem, the sensitivity solution is efficient because of the linearity. The Newton iteration for the corrector step converges in one iteration.

The Jacobian matrix $\partial F/\partial Z$ that appears in equation (19) is exactly the one that is required by the BDF method in solving the original model problem, so it is readily available for the sensivity computation. Each column corresponds to the sensitivities with respect to one of the reaction pre-exponential constants. The solution proceeds column by column. Note that the Jacobian matrix is the same for each column of $w_{l,i}$. However, since the $\partial F_l/\partial a_i$ matrix describes the explicit dependence F on each of the reaction parameters a_i, each of its columns must be formed prior to solving for a column of $w_{l,i}$.

The main weakness with the method as we have implemented it is in the computation of the $\partial F/\partial Z$ and $\partial F_l/\partial a_i$ matrices. The Jacobian, as used for solving the model problem by the BDF method, is only an iteration matrix, and thus even though its accuracy may effect the convergence rate, its accuracy has no effect on the accuracy of the solution. The situation is quite different in the sensitivity equation, where these two matrices appear explicitly. We use finite difference methods to approximate the matrices, rather than derive analytic expressions for them. It is possible to derive the exact expressions, but it is a tedious and time consuming job. Thus, since the sensitivity equations themselves are defined in terms of approximate finite difference representations of the two coefficient matricies, we recognize that the accuracy in the predicted sensitivity coefficients is somewhat compromised. Nevertheless, since our principal application of the sensitivity coefficients is to provide insight about the behavior of the model, we believe that the approximations are justified.

PROGRAM STRUCTURE

Senkin requires subroutines from the Chemkin[5] and DASAC[3] software packages. To solve a problem using Senkin, the user must access the Chemkin thermodynamic data base, execute the Chemkin Interpreter, link Senkin with the Chemkin and DASAC libraries, pass input data to Senkin, and store the output data that is produced. Figure 1 shows the relationships between these various components.

The first step is to execute the Chemkin Interpreter, which reads user-supplied information about the species and chemical reactions for a particular reaction mechanism. It then extracts information about the species' thermodynamic properties from a data base. All this information is stored in the Chemkin linking file, which is needed by the Chemkin subroutine library. The user can refer to the Chemkin documentation for the details on that package.

The input that defines a particular problem and the parameters needed to solve it are read by Senkin from a text file. Unit numbers and default file names in Senkin are assigned by the driver program described below. The input data is read using a Keyword format that is described in the next section.

The output from Senkin is written in two forms: text and binary. Text output is written to provide a record of the run; it includes a summary of the initial conditions and printouts of the partial solution at a selected time interval. The full solution is written to the binary solution file after each successful integration step. The binary file is intended for use in graphics post-processing and/or providing initial data for restart calculations. It may be desirable to continue integration of a problem using a previously computed solution as the starting point. Senkin can read a previous solution file to restart new problems, as suggested by the dashed lines in Fig. 1. The formats of these binary files are discussed below.

Since graphics routines are not portable to all machines, Senkin does not contain any graphics or other interpretive outputs. Instead, we provide a post-processing routine, Psenk, that reads the binary solution file and print text files for use with your local graphics software.

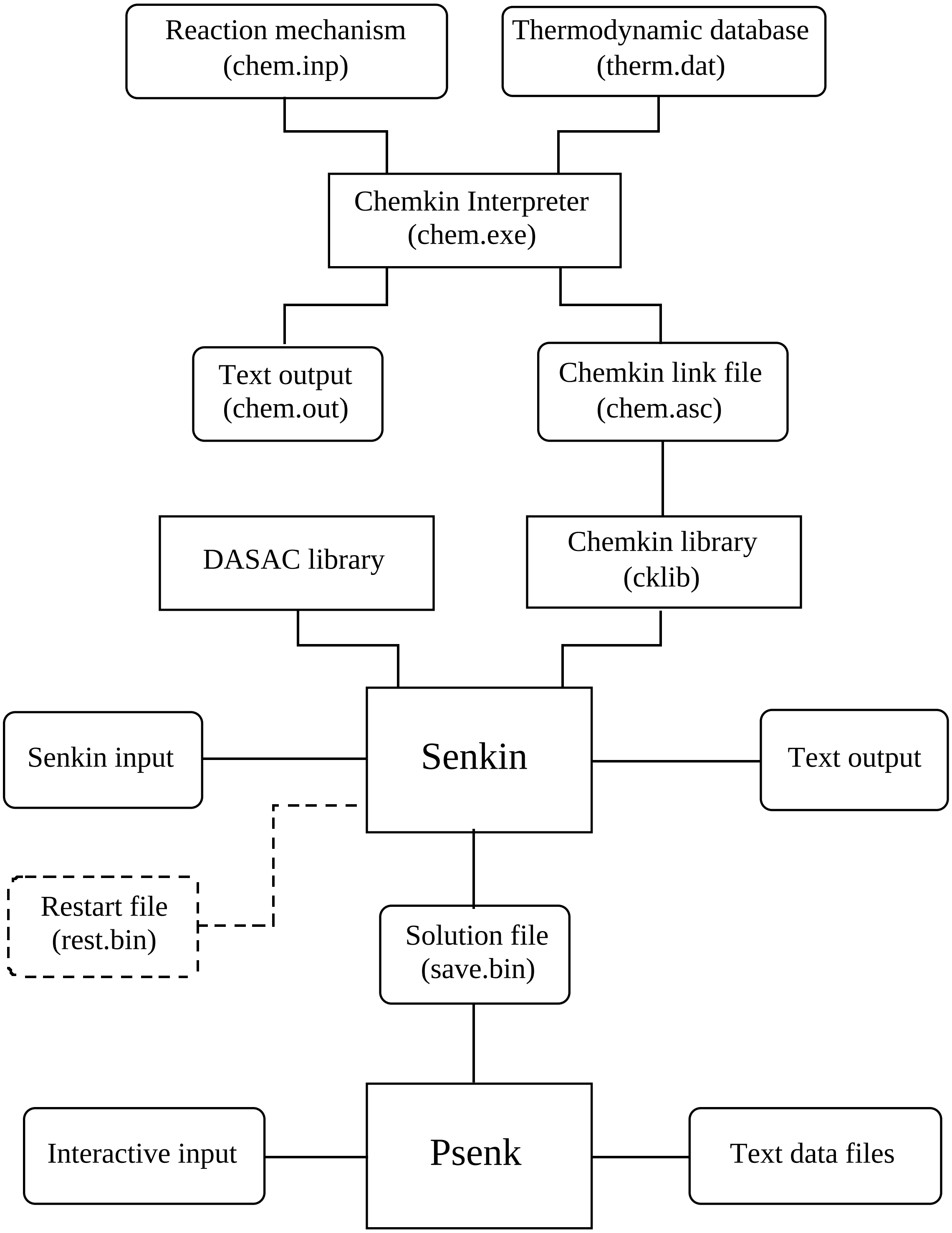

Figure 1. Relationship of Senkin to the Chemkin preprocessor and the associated input and output files.

Structure of the Binary Solution File

The binary file is written using the following Fortran statements. First, Senkin writes a copy of the Chemkin workspace:

```
WRITE (LSAVE) 'CKLINK'
WRITE (LSAVE) (ICKWRK(L), L = 1, LENI)
WRITE (LSAVE) (RCKWRK(L), L = 1, LENR)
WRITE (LSAVE) (CCKWRK(L), L = 1, LENC)
```

then at the start of a computation the general information:

```
WRITE (LSAVE) 'SENKIN SOLUTION'
WRITE (LSAVE) LSENS
WRITE (LSAVE) NSYS, KK, II
```

and after each timestep the solution:

```
WRITE (LSAVE) TIME, P, TEMP, (Y(K), K = 1, KK)
IF (LSENS) WRITE (LSAVE) (( W(L,I), L=1,NSYS), I = 1, II)
```

The variable `LSENS` is a logical variable which is `.TRUE.` if sensitivity coefficients are written to the file and `.FALSE.` otherwise. `NSYS` is an integer equal to the number of dependent variables, and `KK` is an integer equal to the number of species considered. For problem choices A through C, `NSYS` = `KK` + `1`, while for cases D and E, `NSYS` = `KK`. `TIME` is the time (seconds), `P` is the pressure (dynes per square centimeter), `TEMP` is the temperature (Kelvins), `Y(K)` are the species mass fractions, `W(L,I)` are the sensitivity coefficients for the `L`-th dependent variable with respect to the `I`-th reaction, and `II` is the number of reactions.

Restart

The binary solution file can be renamed and used to restart a new computation. The data read from the restart file will replace any initial conditions provided in the keyword input deck. The choices regarding problem type and whether or not sensitivity analysis is to be performed need not be consistent with the original problem that generated the restart data. To access the solution at any point in time, Senkin reads through the restart file searching for the time identified by the `TRES` keyword, and begins the integration as close to that time as possible.

User Modifications

Senkin is written as a subroutine, which is "driven" by a small main program that opens the appropriate files and allocates the working storage before passing control to Senkin. The integer, real, and character work space is allocated with the parameters `LENIWK`, `LENRWK`, and `LENCWK`. The first thing Senkin does is check to make sure that the driver has provided enough memory for the given problem. The first output from the program reports how much space is allocated and how much is required. If too little space is provided, the program stops at this point. Naturally, problems involving larger reaction mechanisms require more working space. The number of governing equations increases with the number of species, while the number of sensitivity equations increases with the product of the number of species and the number of reactions. Consequently, if sensitivity analysis is not requested, considerably less storage space is required.

The driver program may also be modified to change the default file names. The default names shown on the flowchart in Fig. 1 are set in the driver. The Senkin text input and output files use the unit numbers 5 and 6, which are standard input and output on most machines, but these may be redirected.

For problem choices C and F, user-defined subroutines are required. The driver program comes with empty subroutines to avoid compiler complaints of unknown modules, and these serve as templates for your specific problem. In problem choice C, where the volume is a function of time, a subroutine of the form

```
SUBROUTINE VOLT (TIME, VOL, DVDT)
```

must be provided to return the volume and its time derivative as a function of time. As explained in the previous section, the total mass of the system is computed from the volume at time zero. The mass is merely a constant during the integration, and it is needed only to convert the time-dependent volume to density. This means that the volume specified by subroutine `VOLT` may have any consistent units or normalization. The units of volume do not effect the calculation, except that the computed mass will be a constant that represents something other than mass. The units of `TIME` are seconds.

In problem choice F, temperature is a function of time and a subroutine of the form

```
SUBROUTINE TEMPT (TIME, TEMP, PA)
```

must be provided to return the temperature in Kelvins and the pressure in atmospheres. The functions for temperature or volume can in principle be any continuous functions of time; however, we suggest that analytic functions be used (functions that are continuous and differentiable at least once).

INPUT

Senkin reads text input in a keyword format. On each input statement an identifying keyword must appear first. For some input lines only the keyword itself is required, while for others, additional information is required. The order of the keyword inputs is unimportant, except for the first two keywords. The rules governing the syntax of the keyword images are listed below:

1. The first four columns of the card image are reserved for the keyword, and it 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 required, the order in which the information appears is important, and the pieces are delimited by one or more blank spaces.

4. When numbers are required as input, they may be stated in either integer, floating point, or E 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 species names are required as input, they must appear exactly as they are specified in the Chemkin input. They must appear in capital letters.

6. If more information is input than required, then the last read inputs are used. For example, if contradictory keywords are encountered, the last one read is taken.

7. A "comment" line can be inserted by placing either a period (.), a slash (/), or a exclamation mark (!) in the first column. Such a line is ignored by the code, but it is echoed back in the printed output. In addition, on any keyword line, any input that follows the required input and is enclosed in parentheses is taken as a comment.

8. The keyword `SENS` must appear as the first keyword if sensitivity analysis is desired.

9. One of the problem selection keywords (`CONP`, `CONV`, `VTIM`, `CONT`, `CNTV`, or `TTIM`) must either follow the `SENS` keyword, or be the first keyword if `SENS` is not present.

10. The keyword `END` must be the last input card.

KEYWORD DESCRIPTIONS

Sensitivity Option

`SENS` – Sensitivity analysis with respect to the reaction rate constants will be performed and sensitivity coefficients will be written to the binary file.
Default – No sensitivity analysis performed.

Problem Selection

`CONP` – The solution will be obtained with pressure held constant at the initial value. The equations solved are those of case A.
Default – None, a problem selection keyword must be present.

`CONV` – The solution will be obtained with the volume held constant. The equations solved are those of case B.
Default – None, a problem selection keyword must be present.

`VTIM` – The solution will be obtained with the volume as a function of time. Subroutine `VOLT ( TIME, VOL, DVDT )` must be provided to specify the volume. The equations solved are those of case C.
Default – None, a problem selection keyword must be present.

`CONT` – The solution will be obtained with the temperature and pressure held constant at the initial values. The equations solved are those of case D.
Default – None, a problem selection keyword must be present.

`COTV` – The solution will be obtained with the temperature and volume held constant at the initial values. The equations solved are those of case E.
Default – None, a problem selection keyword must be present.

`TTIM` – The solution will be obtained with pressure held constant and the temperature a function of time. Subroutine `TEMPT (TIME, TEMP, PA)` must be provided to specify the temperature. The equations solved are those of case F.
Default – None, a problem selection keyword must be present.

Initial Conditions

TEMP – The initial temperature of the gas mixture.
Units – K
Default – None, temperature is a required input.

PRES – The initial pressure fo the gas mixture.
Units – atmospheres
Default – None, pressure is a required input.

REAC – Moles of the reactant species in the initial mixture. This keyword must be followed by a species name and a numerical value on the same line. For example, `REAC H2 1.0`, enters one mole of hydrogen into the mixture. The moles fractions of the species will be normalized from the input mole quantities, so the absolute magnitudes of the input mole quantities are unimportant.
Units – moles or mole fraction.
Default – None, species input is required.

TLIM – In problems of type A, B, or C, the program computes an ignition time, which is defined by the time when the temperature first reaches a value equal to TLIM.
Units – K
Default – The initial temperature plus 400 K.

REST– The initial conditions will be read from a restart file. The data is read in binary form as described above. When this option is elected, keywords for TEMP, PRES, and REAC are no longer necessary and if present the data entered for these variables are ignored.
Default – Initial conditions taken from input keywords.

TRES – The initial time for a restart calculation.
Units – seconds
Default – The value of time read from the restart file.

Integration Control

`TIME` – The integration proceeds from time zero until this final time. Restart calculations may be given an initial time using `TRES`.
Units – seconds
Default – None, the integration time is a required input.

`DELT` – The time interval for solution printouts to the text output file.
Units – seconds
Default – None, the time interval is a required input.

`ATOL` – Absolute tolerance used by DASAC as an indicator of the accuracy desired in the physical solution. Typically `ATOL` should be less than the smallest meaningful value of the species mass fraction.
Default – 1.E-20

`RTOL` – Relative tolerance used by DASAC as an indicator of the accuracy desired in the physical solution.
Default – 1.E-8

`RTLS, ATLS` – Relative and absolute tolerances used by DASAC as an indicator of the accuracy desired in the solution for the sensitivity coefficients only. Generally, the sensitivity coefficients need not be solved to a great degree of accuracy, so these tolerances could be lower than the tolerances placed on the physical variables.
Default – 1.E-5

`END` – This Keyword must appear at the end of each set of input data.

POST PROCESSING

Since graphics routines are not portable to all machines, Senkin does not attempt to produce any graphical output; instead, it writes the solution at each integration step to a binary solution file. Then Psenk, a text post-processor, reads the solution and generates text files for plotting purposes. Like Senkin, Psenk must be compiled with the Chemkin library.

Psenk is intended to run interactively, prompting you for the name of the binary solution file, then a name for the text output file. The variables that go into the text file are controlled using a series of keywords, which use the same rules as the Senkin keywords.

Psenk keywords

`TEMP` – Requests the temperature be printed in the text file.

`TMIN` – Minimum time for selecting data from the solution file.
Default – 0.
Units – seconds

`TMAX` – Maximum time for selecting data from the solution file.
Default – None; required.
Units – seconds

`SPEC` *C*–Requests species fractions only for the list of species in character string *C* (separate species names by at least one blank space.
Default – All species printed.

`MOLE` – Print mole fractions.
Default – Mole fractions.

`MASS` – Print mass fractions.
Default – Mole fractions.

`TSEN` – Print sensitivity coefficients for temperature.
Default – Temperature sensitivity printed.

`XSEN` *C* – Print sensitivity coefficients for species in character list *C*.
Default – No species sensitivities printed.

`RATE` *C* – Print chemical production rates for species in character list *C*.
Default – No production rates printed.

Normalized Sensitivities

Senkin does not normalize the sensitivity solution, for two reasons: (1) it would eliminate the user's choice of normalization, and (2) our preferred normalization involves storing the entire time history of the solution, which would require too much space for most problems. The normalization scheme implemented in Psenk is described next.

The binary solution file contains

$$w_{l,i} = \frac{\partial Z_l}{\partial a_i}, \tag{18}$$

for the sensitivity of the l-th solution component to the i-th reaction. Psenk loops over the time solutions, reading $w_{j,l,i}$, where the additional index j is for time solution, storing only a single component of the sensitivities for the l fixed. Psenk processes the other components by rewinding the solution file and re-reading the sensitivities for the next component.

In general, Psenk normalizes the sensitivities as

$$\hat{w}_{j,l,i} = \frac{a_i}{\max_j\{Z_{j,l}\}} \cdot w_{j,l,i} \tag{20}$$

where $\max_j\{Z_{j,l}\}$ is the maximum value of $Z_{j,l}$ over the $j = 1, J$ time solutions. Taking the maximum value over the time history of the solution is especially important for the species, because many species are essentially zero during much of the time domain. For the mass fraction sensitivities, the normalization would be

$$\hat{w}_{j,k,i} = \frac{a_i}{\max_j\{Y_{j,k}\}} \cdot w_{j,k,i} \tag{21}$$

for the k-th species. If output is being presented in terms of mole fractions, the mass fraction sensitivities can be converted to mole fractions using

$$\tilde{w}_{j,k,i} = \frac{a_i}{\max_j\{X_{j,k}\}} \frac{\partial X_{j,k}}{\partial a_i} = \frac{a_i}{\max_j\{X_{j,k}\}} \left(\frac{\bar{W}_j}{W_k}\right) \left[w_{j,k,i} - Y_{j,k}\bar{W}_j \sum_{n=1}^{K} \frac{w_{j,k,i}}{W_n} \right] \tag{22}$$

We include the term a_i for completeness, and because Senkin versions earlier than version used the pre-exponential coefficient as the sensitivity parameter, $a_i = A_i$; some users using older version may need to be aware of this change.

EXAMPLE PROBLEM

The sample problem chosen for a demonstration of Senkin is the spontaneous ignition of a stoichiometric hydrogen-air mixture at constant pressure. The input reaction mechanism for the Chemkin interpreter is:

```
ELEMENTS
 H  O  N
END
SPECIES
 H2  H  O2  O  OH  HO2  H2O2  H2O   N2
END
REACTIONS
H+O2+M=HO2+M     3.61E17  -0.72       0.      !DIXON-LEWIS
  H2O/18.6/  H2/2.86/
H+H+M=H2+M       1.0E18   -1.0        0.      !D-L
H+H+H2=H2+H2     9.2E16   -0.6        0.
H+H+H2O=H2+H2O   6.0E19   -1.25       0.
H+OH+M=H2O+M     1.6E22   -2.0        0.      !D-L
  H2O/5/
H+O+M=OH+M       6.2E16   -0.6        0.      !D-L
  H2O/5/
O+O+M=O2+M       1.89E13   0.0    -1788.      !NBS
H2O2+M=OH+OH+M   1.3E17    0.0    45500.
H2+O2=2OH        1.7E13    0.0    47780.
OH+H2=H2O+H      1.17E9    1.3     3626.      !D-L$W
O+OH=O2+H        3.61E14  -0.5        0.      !JAM 1986
O+H2=OH+H        5.06E4    2.67    6290.      !KLEMM,ET AL 1986
OH+HO2=H2O+O2    7.5E12    0.0       0.0   !D-L
H+HO2=2OH        1.4E14    0.0     1073.      !D-L
O+HO2=O2+OH      1.4E13    0.0     1073.      !D-L
2OH=O+H2O        6.0E+8    1.3        0.      !COHEN-WEST.
H+HO2=H2+O2      1.25E13   0.0        0.      !D-L
HO2+HO2=H2O2+O2  2.0E12    0.0        0.
H2O2+H=HO2+H2    1.6E12    0.0     3800.
H2O2+OH=H2O+HO2  1.0E13    0.0     1800.
END
```

The keyword input for Senkin is:

```
SENS
CONP
PRES  1.
TEMP  1000.
REAC H2  2.
REAC O2  1.
REAC N2  3.76
TIME 2.E-4
DELT 1.E-4
DTSV 1.E-6
END
```

This input information is echoed to the output file, shown on the following pages. The program first verifies work space, then banners are printed by Chemkin. Next follows the initial conditions, for which the species molar input is renormalized into mole fractions. As the time integration proceeds, the solution is printed at a time interval of 100 μs. Since the TLIM Keyword is not specified, the temperature limit used to compute the ignition time will be the default value of 1400 K (the initial temperature plus 400 K). The ignition time is 173 μs. After the integration is completed, the program reports the number of time datasets that were printed to the binary solution file.

After using Psenk to post-process the solution, plots like those shown in figures 3 and 4 can be generated. These figures zoom in on the time window near the ignition event. Figure 4 shows production rates and sensitivities for H_2, which show a competition between the chain-branching reactions and a three-body reaction that competes for H-atom. The sensitivity coefficient is normalized as described in the post-processing section. As one might expect, the largest sensitivity occurs near the time of ignition when the most rapid change in temperature is taking place.

Text Output for Example Computation

```
 SENKIN:  Sensitity Analysis
          Author:  Andy Lutz
          CHEMKIN-III Version 3.8, 97/04/22
          DOUBLE PRECISION

 CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
 DOUBLE PRECISION Vers.  5.8 97/03/01
 Copyright 1995, Sandia Corporation.
 The U.S. Government retains a limited license in this software.




      Enter keyword:
          SENS

     Sensitivity analysis will be performed.


      Enter keyword:
          CONP

                Working Space Requirements
                 Provided        Required
 Integer             7500           877
 Real              60000           3354
 Character            500             22


 CKLIB: CHEMKIN-III GAS-PHASE CHEMICAL KINETICS LIBRARY,
 DOUBLE PRECISION Vers.  5.8 97/03/01
 Copyright 1995, Sandia Corporation.
 The U.S. Government retains a limited license in this software.



     Keyword input:

PRES  1.
TEMP  1000.
REAC H2  2.
REAC O2  1.
REAC N2  3.76
TIME 2.E-4
DELT 1.E-4
DTSV 1.E-6
END
```

```
    Pressure is constant.


 Initial Conditions:

 Pressure (atm)  =  1.0000E+00
 Temperature (K) =  1.0000E+03
 Density (gm/cc) =  2.5484E-04

 Mole Fractions:
H2         = 2.9586E-01
H          = 0.0000E+00
O2         = 1.4793E-01
O          = 0.0000E+00
OH         = 0.0000E+00
HO2        = 0.0000E+00
H2O2       = 0.0000E+00
H2O        = 0.0000E+00
N2         = 5.5621E-01


 Time Integration:


t(sec)= 0.0000E+00   P(atm)= 1.0000E+00   T(K)= 1.0000E+03
 H2        = 2.96E-01  H         = 0.00E+00  O2        = 1.48E-01
 O         = 0.00E+00  OH        = 0.00E+00  HO2       = 0.00E+00
 H2O2      = 0.00E+00  H2O       = 0.00E+00  N2        = 5.56E-01

t(sec)= 1.0029E-04   P(atm)= 1.0000E+00   T(K)= 1.0001E+03
 H2        = 2.96E-01  H         = 9.06E-06  O2        = 1.48E-01
 O         = 8.92E-07  OH        = 3.01E-07  HO2       = 9.79E-06
 H2O2      = 2.32E-08  H2O       = 2.12E-05  N2        = 5.56E-01

t(sec)= 2.0000E-04   P(atm)= 1.0000E+00   T(K)= 2.0896E+03
 H2        = 5.60E-02  H         = 5.08E-02  O2        = 2.68E-02
 O         = 1.55E-02  OH        = 2.45E-02  HO2       = 9.41E-06
 H2O2      = 6.35E-06  H2O       = 2.26E-01  N2        = 6.01E-01


  Integration completed:
  Ignition Time (sec) =   1.7265E-04
  Temp criteria (K) =   1.4000E+03

  Binary file has    201 time datasets.
Total CPUtime (sec):                3.96-316
```

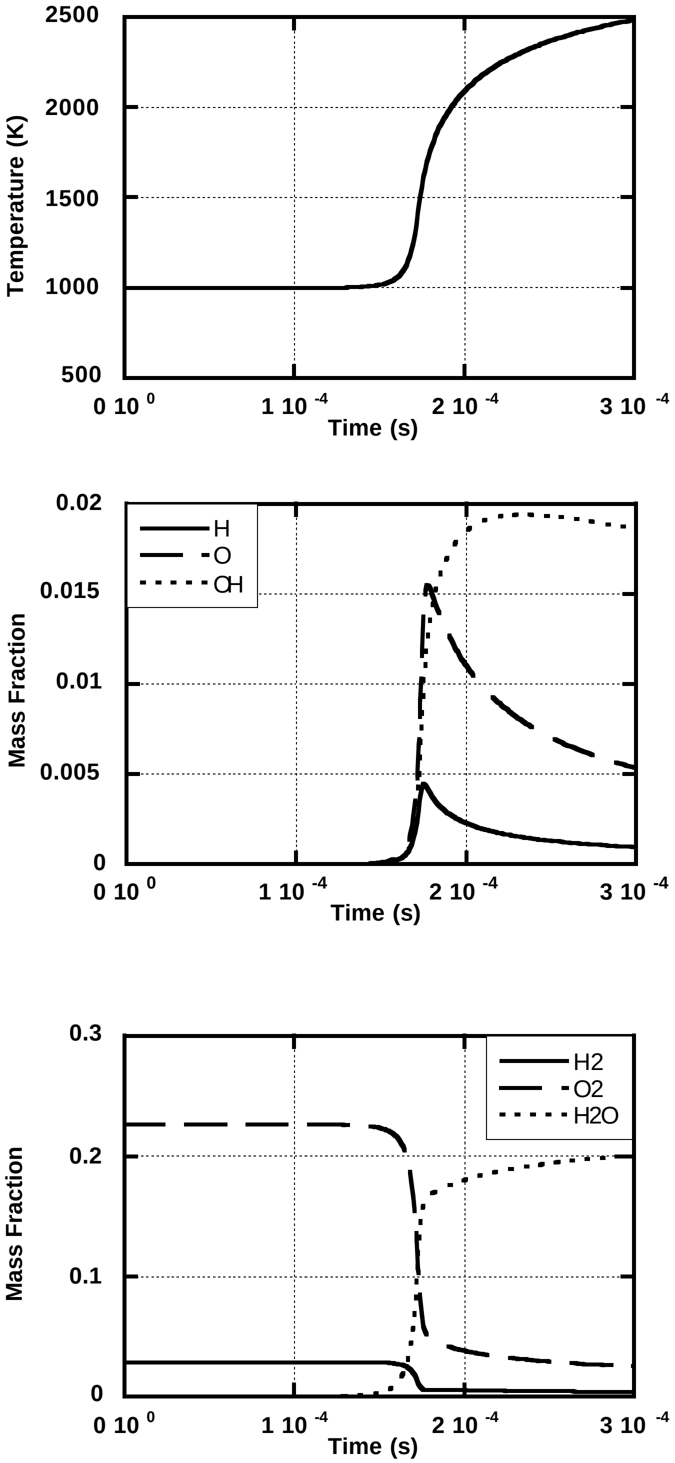

Figure 2. Temperature and mass fractions of some of the species in the hydrogen-air example.

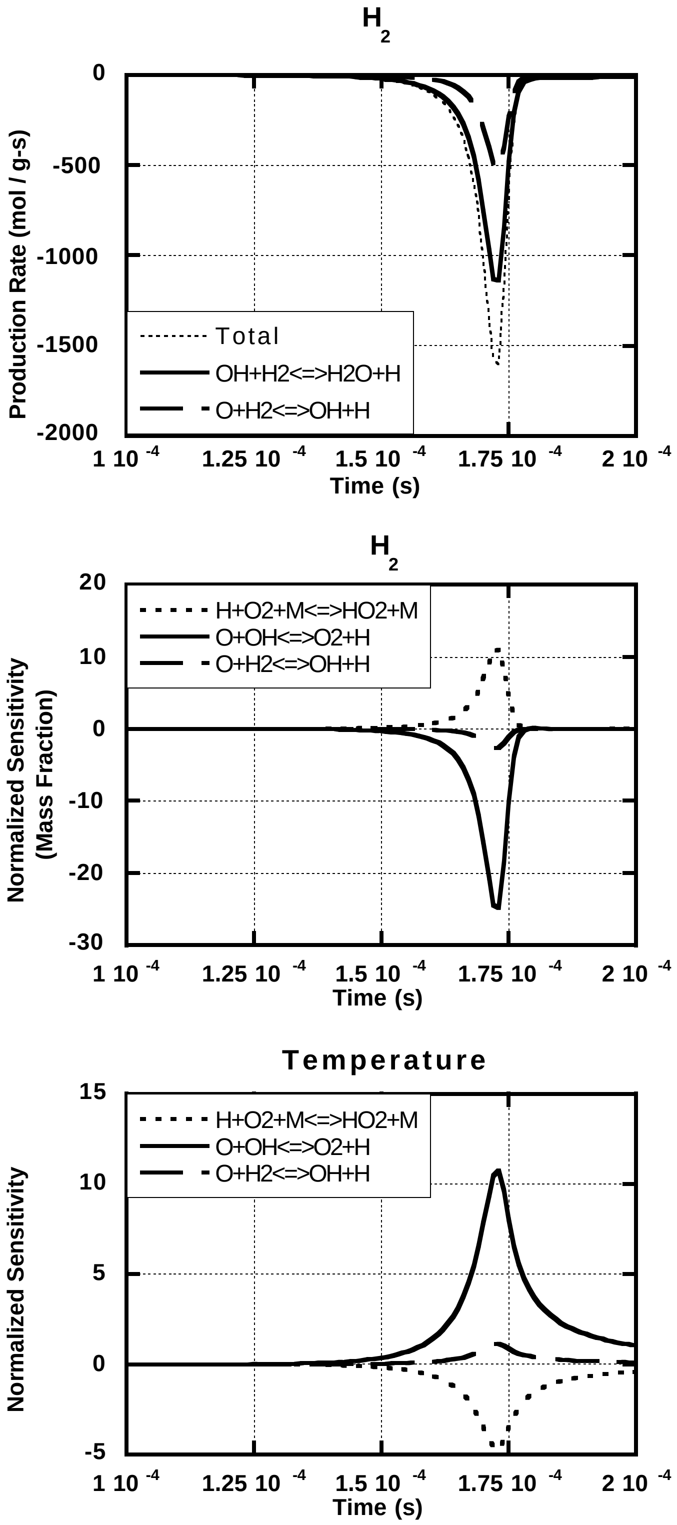

Figure 4. Production rate and normalized sensitivity for H_2, and sensitivity for temperature with respect to the important reaction rates for the example computation.

REFERENCES

1. Kramer, M. A., Kee, R. J. and Rabitz, H., "CHEMSEN: A Computer Code for Sensitivity Analysis of Elementary Chemical Reaction Models," Sandia National Laboratories Report SAND82-8230 (1982).

2. Kramer, M. A., Calo, J. M., Rabitz, H. and Kee, R. J., "AIM: The Analytically Integrated Magnus Method for Linear and Second Order Sensitivity Coefficients," Sandia National Laboratories Report SAND82-8231 (1982).

3. Caracotsios, M. and Stewart, W. E., "Sensitivity analysis of initial value problems including ODE's and algebraic equations," *Computers and Chemical Engineering* **9** No. 4, 359-365 (1985)

4. Petzold, L. R., " A description of DASSL: a differential/algebraic system solver," Sandia National Laboratories Report SAND82-8637, (1982).

5a. 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).

5b. Kee, R. J., Rupley, F. M., Miller, J. A., "Chemkin-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics," Sandia National Laboratories Report SAND89-8009, (1989).

6. Kee, R. J., Petzold, L. R., Smooke, M. D., Grcar, J. F., "Implicit Methods in Combustion and Chemical Kinetics Modeling," *Multiple Time Scales*, Academic Press (1985).

7. Tilden, J. W., Costanza, V, McRae, G. J., and Seinfeld, J. "Sensitivith Analysis of Chemically Reacting Systems," in *Modeling of Chemical Reaction Systems*, K. H. Ebert, P. Deuflhard, and W. Jaeger, Editors, Springer-Verlag, New York, p. 69-91 (1981).

8. Rabitz, H, Kramer, M. A., and Dacol, D., "Sensitivity Analysis in Chemical Kinetics," *Ann. Rev. Phys. Chem.* **34** 419-461 (1983).

9. Dickinson, R. P. and Gelinas, R. J., "Sensitivity Analysis of Ordinary Differential Equations – a Direct Method," *J. Comput. Phys.* **21** 123-143 (1976).

10. Dougherty, E. P., Hwang, J.-T., and Rabitz, H., "Further Developments and Applications of the Green's Function Method of Sensitivity Analysis in Chemical Kinetics," *J. Chem. Phys.* **71** 1794-1808 (1979).

11. Cukier, R. I., Levine, H. B., and Shuler, K. E., "Nonlinear Sensitivity Analysis for Multiparameter Model Systems," *J. Comput. Phys.* **26** 1-42 (1978)

12. Kramer, M. A., Calo, J. M., and Rabitz, H., "An Improved Computational Method for Sensivity Analysis: Green's Function Method with AIM," *Appl. Math. Modelling* **5** 432-441 (1981).
13. Kramer, M. A., Rabitz, H., Calo, J. M., and Kee, R. J., "Sensitivity Analysis in Chemical Kinetics: Recent Developments and Computational Comparisons," *Intl. J. Chem. Kinetics* **16** 559-578 (1984).
14. Hindmarsh, A. C. "LSODE and LSODI, Two Initial Value Ordinary Differential Equation Solvers," *ACM SIGNUM Newsletter* **15** 10-11 (1980).
15. Dunker, A. M., "The Direct Decoupled Method for Calculating Sensitivity Coefficients in Chemical Kinetics," *J. Chem. Phys.* **81** (5) 2385-2393 (1984).
16. Leis, J. R. and Kramer, M. A., "Sensitivity Analysis of Systems of Differential and Algebraic Equations," *Computers and Chemical Engineering* **9**, No. 1, 93-96 (1986).
17. Caracotsios, M. and Stewart, W. E., "Sensitivity Analysis of Initial Value Problems with Mixed ODEs and Algebraic Equations," *Computers and Chemical Engineering* **9**, No. 4, 359-365 (1986).