Manual for the Use of Soot Subroutines in Chemical Kinetics Applications

In a typical chemical kinetics code, species conservation is described by a set of nonlinear differential equations

$$\frac{\partial Y_k}{\partial t} = f\left((Y_k, \frac{\partial Y_k}{\partial x_i}, \dot{\omega}_{k,\ldots} \right), \; k = 1,\ldots,N_C, \tag{1}$$

where Y_k is the mass concentration of species *k*, t is time, x_i is the i*th* spatial coordinate, $\dot{\omega}_k$ is the rate for Y_k and N_C is the number of gas-phase species. Assuming the distribution of soot to be described by a set of moments, M_r, we have

$$\frac{\partial M_r}{\partial t} = f\left(M_r, \frac{\partial M_r}{\partial x_i}, Q_r, \ldots \right), \; r = 0,\ldots, N_M - 1 \;, \tag{2}$$

where Q_r is the "formation" rate of the *r*th moment and N_M is the number of moments. The indices of the moments conventionally begin with 0 rather than 1; M_0 is the total number of particles. It is the function of the soot subroutines described below to return the Q_r and, additionally, to return adjustments to the rates $\dot{\omega}_k$ for certain species in order to account for consumption or production of these species as a result of soot particle formation, coagulation, surface growth, and oxidation. If we write

$$\dot{\omega}'_k = \dot{\omega}_k + p_k, \quad k = 1,\ldots, N_C; \qquad \dot{\omega}'_{N_C+r+1} = Q_r, \quad r = 0,\ldots N_M - 1, \tag{3}$$

and

$$Y_{N_C+r+1} = M_r \;,\; r = 0,\ldots N_M - 1 \;, \tag{4}$$

where *p* is the array of soot production rates for chemical species and $\dot{\omega}'$ is the array of rates adjusted for soot production, then the problem may be described by equation 1 with $\dot{\omega}$ replaced by $\dot{\omega}'$ and with the index extending from 1 to N_C+N_M.

The user must provide the array *Y* and the indices within that array of the species of interest in soot formation, which include the species responsible for nucleation and those species that participate in surface reactions. Once the moment rates and production rates have been returned, he must set up the array $\dot{\omega}'$ for use in equation 1. At present, production rates are returned for the inception species and for C_2H_2, CO, H, H_2, H_2O, O_2, and OH, all of which must be present in the chemical mechanism. If the user wishes to add other species, he must modify the routine **surface**, as discussed below.

The module **soot.f** contains just two user-callable functions: integer function **soot,** which must be called each time the rates Q_r are needed, and subroutine **initSoot**, which must be called just once before any call to **soot**. The two routines are invoked in the following manner:

call initSoot (n_moments, iPAH, iC2H2, iCO, iH, iH2, iH$_2$O, iO2, iOH, iM0, freeMolOnly, M0_cutoff, PAH_carbon, PAH_diam)

integer **n_moments**	number of soot moments to be calculated; this is currently fixed at 6, i.e., M_0 through M_5
integer **iPAH**	the index of the inception species in the array Y
integer **iC2H2**	the index of C_2H_2
integer **iCO**	the index of CO
integer **iH**	the index of H
integer **iH2**	the index of H_2
integer **iH2O**	the index of H_2O
integer **iO2**	the index of O_2
integer **iOH**	the index of OH
integer **iM0**	the index of the first moment, M_0; this must follow all gaseous species and the other moments must follow in sequence
logical **freeMolOnly**	if true, coagulation rates are calculated for the free-molecular regime only; otherwise, for the transition regime
double precision **M0_cutoff**	below this value of the moment M0, only nucleation rates are calculated; the value must be equal to or greater than 0.0
integer **PAH_carbon**	number of carbon atoms in the inception PAH species
double precision **PAH_diam**	diameter of the inception PAH species (cm);

i = soot (Y, T, P, rate, nucRate, coagRate, surfRate, surfPAH, surfC2H2. SurfO2, surfOH, prodPAH, prodC2H2, prodCO, prodH, prodH2, prodH2O, prodO2, prodOH)

input:

double precision **Y(*)**	array of chemical species concentrations (mol cm^{-3}) and soot moments (cm^{-3}); moments must appear in sequence at end of array

double precision **T**	temperature (K)
double precision **P**	pressure (atm)

output:

double precision **rate(0:*)**	array of total rates for moments (cm^{-3} s^{-1})
double precision **nucRate(0:*)**	array of nucleation rates for moments (cm^{-3} s^{-1})
double precision **coagRate(0:*)**	array of coagulation rates for moments (cm^{-3} s^{-1})
double precision **surfRate(0:*)**	array of total rate of condensation, surface growth, and oxidation rates for moments (cm^{-3} s^{-1})
double precision **surfPAH(0:*)**	array of condensation rates (cm^{-3} s^{-1})
double precision **surfC2H2(0:*)**	array of surface growth/oxidation rates due to C_2H_2 (cm^{-3} s^{-1})
double precision **surfO2(0:*)**	array of surface growth/oxidation rates due to O_2 (cm^{-3} s^{-1})
double precision **surfOH(0:*)**	array of surface growth/oxidation rates due to OH (cm^{-3} s^{-1})
double precision **prodPAH**	surface production rate for the inception species (mol cm-3)
double precision **prodC2H2**	surface production rate for C_2H_2 (mol cm-3)
double precision **prodCO**	surface production rate for CO (mol cm-3)
double precision **prodH**	surface production rate for H (mol cm-3)
double precision **prodH2**	surface production rate for H_2 (mol cm-3)
double precision **prodH2O**	surface production rate for H_2O (mol cm-3)
double precision **prodO2**	surface production rate for O_2 (mol cm-3)
double precision **prodOH**	surface production rate for OH (mol cm-3)

return value:

0	normal return
-1	dispersion <= 1.0 (non-physical)
-2	moment $M_r <= M_{r-1}$ (non-physical)

Surface growth and oxidation are presently based on the following five equations:

$$\mathrm{C_{soot}-H+H=C_{soot}\bullet+H_2}$$
$$\mathrm{C_{soot}-H+OH=C_{soot}\bullet\ +\ H_2O}$$
$$\mathrm{C_{soot}\bullet+H\rightarrow C_{soot}}$$
$$\mathrm{C_{soot}\bullet+C_2H_2=C_{soot}\bullet+H}$$
$$\mathrm{C_{soot}\bullet+O_2\rightarrow products}$$
$$\mathrm{C_{soot}+OH\rightarrow products}$$

If the user wishes to add equations involving other species, he may do so by modifying the arguments to subroutines **initSoot** and **soot** and by modifying subroutine **surface.** He is discouraged from doing so.

Chemkin-II examples – PSR, SENKIN (PFR), and PREMIX

The Chemkin-II package comprises two interpreters and a set of library routines. The first (chemistry) interpreter reads a description of the user's chemical mechanism (**chem.inp**), adds thermodynamic information for each species from a data file (**therm.dat**), and writes a binary output file (**chem.bin**). The second (transport) interpreter reads **chem.bin** and **tran.dat**, a file that contains a set of transport properties for each species, and produces the binary output file **tplink**. The library contains a number of FORTRAN subroutines that provide species names, thermodynamic properties, reaction rates, etc., for the user's application, which characteristically involves the solution of a set of differential equations such as equation 1. The application reads the data in the binary file into a set of arrays, which are then passed to the library routines. These routines then return the values needed in the equations, which may then be solved.

Ideally, we should create an additional input file, which would combine the gas-phase chemistry and soot properties much in the manner of the surface chemistry input file used with Chemkin-III. Since the number of soot properties is small and since the reading and rewriting of **chem.bin** is cumbersome, we have decided on a simpler approach which requires only the modifications of the PSR application which would be necessary in any case, but which does not require an additional input file.

The first step, then, is to add to **chem.inp** a number of pseudo-species representing the moments of the soot distribution; this ensures that arrays required by PSR will be of the correct size. The pseudo-species should be the last species in the file and should be identified by a distinguishing character for later identification; we use the dollar sign, so that the soot moments M_0, M_1, … are named M0$, M1$, … We have modified the chemistry and transport interpreters so that thermodynamic and transport properties for the pseudo-species need not be added to **therm.dat** and **tran.dat**. We may now run the interpreters to create **chem.bin** and **tplink**. In the applications, the reading of **chem.bin** and, if necessary, **tplink**, and the subsequent initialization of Chemkin itself proceed as before. After the initialization, we must take the following steps:

1. Call a utility routine that examines the array of species name to determine the number of soot distribution moments, **nMoments** (by checking for the presence of the dollar sign) and the index in the array of the first moment. This routine returns an error if the moments are not in sequence at the end of the array. The value of **nMoments** returned is the maximum number that can be used, not necessarily the number that

will be used. **Note that neither this nor any of the other utility routines discussed below are part of the soot distribution.**

2. Reduce the size of the variable **nkk** in the Chemkin common block **ckstrt** by the number of moments. This ensures that calculations performed by Chemkin routines, such as the calculation of densities, will be based only on the gas-phase species.

We may now use the Chemkin library as needed, *e.g.*, to obtain the number of gas-phase reactions and the reaction rates.

PSR

One of the standard Chemkin-II applications is the perfectly-stirred reactor (PSR), a mathematical representation of the continuously-stirred tank reactor (CSTR) of chemical engineering texts (Glarborg P, Kee RJ, Grcar, JF, Miller, JA, "PSR: A FORTRAN Program for Modeling Well-Stirred Reactors," Sandia Report SAND86-8209, Sandia National Laboratories, Albuquerque, NM 87185, 1991). In this application, we find solution of the set of steady-state conservation equations

$$\frac{1}{\tau}\left(Y_k - Y_k^*\right) - \frac{\dot{\omega}k W_k}{\rho} = 0, k = 1, N_{gas}, \tag{5}$$

where τ is residence time (s), Y_k and Y^*_k are the inlet and outlet mass fractions, respectively, of the k*th* gas-phase species (dimensionless), $\dot{\omega}_k$ is the production rate of the k*th* species (cm^{-3} s^{-1}), W_k is the molecular weight of the k*th* species (g), ρ is the density of the mixture (g cm^{-3}), and N_{gas} is the number of gas-phase species. To these equations, we add the set

$$\frac{1}{\tau}\left(M_r - M_r^*\right) - Q_r = 0, r = 0, N_{moments}, \tag{6}$$

where M_r is the r*th* moment (cm^{-3}), Q_r is the formation rate of the r*th* moment (cm^{-3} s^{-1}), and $N_{moments}$ is the number of moments. (In the original application, it is also possible to obtain the temperature by solving the energy equation; here we consider only the fixed-temperature case). Because the moments are characteristically very large numbers, we choose to rewrite (6) as

$$\frac{1}{\tau}\left(1 - \frac{\exp\left(Y_k^*\right)}{\exp\left(Y_k\right)}\right) - \frac{Q_r}{\exp\left(Y_k\right)} = 0, k = N_{gas} + 1, N, \tag{7}$$

where N = Ngas + Nmoments and

$$Y_{r+1+N_{gas}} = \log(M_r). \tag{8}$$

The species names and properties are obtained from **chem.bin** and Chemkin is initialized, as in the original application. The the two steps described in the previous section are taken, so that the maximum number of moments is known and Chemkin deals only with the gas-phase species. The next step in the application is the reading of the input file **psr.inp**, which determines the reactor characteristics. We shall not discuss here the standard form of this file, which is described in the PSR manual (Glarborg P, Kee R, Grcar J, Miller J, "PSR:A FORTRAN Program for Modeling Well-Stirred Reactors," Sandia report SAND86-8209, Sandia National Laboratories, Albuquerque, NM 87185, 1991), but limit ourselves to those keywords which have been added to control soot-related behavior. These are:

- **NMOM <N>**
  Sets the number of moments to N (currently either 0 or 6). Use of this keyword allows the user to run the PSR without soot (**N**=0) without having to go to the trouble of changing **chem.bin**. In the future, it will allow the choice of a number of moments other than 6.

- **MOM <NAME> <VALUE>**
  Sets the natural logarithm of the initial value of the moment **NAME** to **VALUE**. Setting the initial value of one or more moments may increase the speed of convergence of the code. **NAME** must correspond to one of the moment species names used in **chem.inp**, *e.g.*, M0$. If a value is set for a given momen, the value of each subsequent moment is set by adding the logarithm of the dimer mass (*i.e.*, twice the number of carbon atoms in the inception species). Thus, setting the initial value of M_0 sets values for all other moments; however, these values may be overridden by subsequent uses of the **MOM** keyword. Only the value of the log of the zeroth moment is usually required; its default value is 1.0. The values set here are NOT assumed to be inlet moment values; see the keyword **SIN** below.

- **FM**
  For coagulation rates, use only the values for the free-molecular regime. The default is to use values for the transition regime.

- **MCUT <VALUE>**
  Set the cutoff value for M_0 to **VALUE**. When M0 is below this value, only nucleation rates are calculated. This may increase computational speed by avoiding the time-consuming calculation of coagulation and surface-growth rates until relatively large amounts of soot have developed. The default value is 0.0

- **SIN**
  Treat the initial values of the moments, set by keyword **MOM**, as inlet values. By default, the inlet values of all soot moments are 0.

- **LIST**
  Produce a list file of the final gas-phase mole-fractions and soot moments on FORTRAN unit 99. This is useful in creating input files for use where a series of reactors is involved.

- **TAB**
  Produce a tabular file of the final gas-phase mole-fractions and soot moments on FORTRAN unit 98. This is useful for spreadsheets.

At this point, the actual number of moments to be used in the calculation, **kMoments**, has been returned from **psrkey**. We set the number of species to be **kk-nMoments+kMoments** and then set **nMoments** = **kMoments**. If **nMoments** is not 0, we carry out the following steps:

1. Call a utility routine that identifies the indices in the array **ksym** of the gas-phase species whose concentrations are required or whose production rates are modified by the soot routines. At present these are the inception PAH species (*e.g.*, pyrene), C_2H_2, CO, H, H_2, O_2, and OH. These indices are used as input to subroutine **initSoot**. If chemical reactions involving species other than those listed were to be used in determining surface growth and oxidation, the appropriate indices must be found at this stage.

2. Call **initSoot** using the following arguments:

   - **nMoments**
   - The indices returned by the procedure of step 1 in correct order
   - The index of M_0, which is just **kk-nMoments+1**
   - The chosen values of **freeMolOnly** and **M0_cutoff**; in general, the values **.false**. and 0.0 may be used
   - The number of carbon atoms in the chosen inception PAH species and its diameter in cm.

The only further change in the original PSR application is to modify the subroutine **psrfun**, which must return the values of the left-hand sides of equations 5 and 7, so that the original calculations are used only for the gas-phase species and new ones for the moments; here **soot** must be called to obtain the Q_r. The values of $\log(M_r)$ and $\log(M_r^*)$ are included in the arrays **S** and **YIN**, respectively, as input to **psrfun**. The values of the Q_r are obtained from subroutine **soot**, which is called as indicated in the first part of this document. The inputs to **soot** are obtained in the following manner:

- The array **Y** comprises two parts: the first **kkgas** elements are the molar concentrations of the gas-phase species obtained from the array **S** by a call to the chemkin routine **ckytcr**; the remaining **nMoments** elements are identical to those in **S**.
- **T** and **P** are available in **psrfun**; the latter must be converted to atm.

The Q_r are returned from soot as **rate(0:nMoments-1)**. The several rate adjustments returned by **soot** must be added algebraically to the $\dot{\omega}_k$ before the values of equation 5 are obtained.

Once the solution has been obtained, we modify the subroutine **psprnt** so that a number of quantities relevant to soot are written to the output file. This is done through a call to the utility **printSoot**, which writes out the moments themselves, the average soot diameter, the total soot surface are, the fractional soot volume, the nucleation and coagulation rates of the 0th moment, the surface growth rates of the 1st moment and the adjusted and unadjusted rates for the inception species and those species involved in surface reactions.

PFR

The Chemkin application **SENKIN** (Lutz A, Kee R, Miller J, "SENKIN: A Fortran Program for Predicting Homogeneous Gas Phase Chemical Kinetics With Sensitivity Analysis," Sandia Report SAND87-8248, Sandia National Laboratories, Albuquerque, NM 87185, 1991) describes the time-dependent behavior of a homogeneous gas mixture in a closed system. A system of non-linear first-order differential equations must be solved:

$$\frac{dY_k}{dt} = \frac{\dot{\omega}_k W_k}{\rho}, \quad k = 1, N_{gas}, \tag{9}$$

where the symbols have the same meaning as those of the PSR above. This system may be said to represent a plug-flow reactor. As with PSR, we add additional equations representing the behavior of the soot distribution moments:

$$\frac{dM_r}{dt} = Q_r, \quad r = 0, N_{moments} - 1 \,. \tag{10}$$

Using the definition of equation 8, we have

$$\frac{dY_k}{dt} = \frac{Q_r}{M_r}, \quad k = N_{gas} + 1, N \,. \tag{11}$$

The modifications to the original application are similar to those for PSR. The initialization process and the use of the added keywords **NMOM**, **MOM**, **FM**, and **MCUT** are identical; the keyword **SIN** has no meaning in this application. **SENKIN** has several routines, representing different physical situations, for determining the RHS of equation 9. In three cases: constant temperature (keyword **CONT**), constant temperature and volume (**CNTV**), and user-defined temperature profile (**TTIM**), the energy equation is not used and the results may be considered correct. In the remaining cases, because we have not yet modified the energy equation, there may be errors, which are very likely to be small.

As with PSR, we have modified the output routine to print the average soot diameter (<d>), total surface area (S), fractional volume (Fv), nucleation (RN), coagulation (RC), and surface growth (SGA4, SGC2H2, SGO2, SGOH) rates, unadjusted production rates (e.g., Rate A4), and production rate adjustments (e.g., Adj A4) for each of the gas-phase species involved in soot creation.

PREMIX

The application **PREMIX** (Kee, RJ, Grcar, JF, Smooke, MD, Miller, JA, “A Fortran Program for Modeling Steady Laminar One-Dimensional Premixed Flames,” Sandia Report SAND85-8240, Sandia National Laboratories, Albuquerque, NM, 87185, 1995) describes the steady-state behavior of a one-dimensional flame. A system of non-linear first order differential equations must be solved:

$$\dot{M}\frac{dY_k}{dz} + \frac{d}{dz}\left(\rho A Y_k v_k\right) - A\dot{\omega}_k W_k = 0, \;\; k = 1, N_{gas}. \tag{12}$$

Here $\dot{M}$ is the mass flow rate (cm^3 s^{-1}), A the area (cm^2), V_k the diffusion velocity of the kth species (cm s^{-1}), and z the spatial coordinate (cm). The remaining symbols have been previously defined. To these equations must be added those for the soot moments M_r, which we state without deriving:

$$\frac{\partial}{\partial z}\left[-D_{i_0}\frac{\partial M_{r-2/3}}{\partial z} + v_T M_r\right] + \rho\, v \frac{\partial\left(M_r/\rho\right)}{\partial z} - Q_r = 0, \quad r = 1, N_{moments.} \tag{13}$$

The diffusion rate of the smallest soot particles, D_{i_0}, is given by

$$D_{i_0} = \frac{3}{2\rho}\sqrt{\frac{\overline{m}k_B T}{2\pi}}\left(1 + \frac{\pi\alpha_T}{8}\right)^{-1}\frac{1}{d_{i_0}^2}, \tag{14}$$

where $\overline{\mathrm{m}}$ is the mean mass of the gas (g), k_B is Boltzmann's constant (erg mol^{-1} K^{-1}), T the temperature (K), α_T the dimensionless thermal accommodation coefficient (≈1), and d_{i_0} the diameter of the smallest soot particle (cm). The thermal diffusion velocity, v_T, is given by

$$v_T = -\frac{3}{4}\left(1+\frac{\pi\alpha_T}{8}\right)^{-1}\frac{\mu}{\rho T}\frac{\partial T}{\partial z}, \tag{15}$$

where μ is the gas viscosity (g cm^{-1} s^{-1}). (The expressions for D_{i_0} and v_T apply only to the free-molecular regime; expressions for the continuum regime are not yet available). As with PSR and SENKIN, equation 13 is rewritten using the natural logarithms of the moments and combined with equation 12 to obtain a set of N(=N_{gas}+$N_{moments}$) equations.

We have added the keywords **NMOM**, **MOM**, **FM**, and **MCUT**, **LIST**, and **TAB**, the meanings of which are given above, under the PSR heading. If the keyword **NDIF** is used, diffusion of soot particles will be neglected; this greatly speeds the calculation with, so far as is now known, little effect on results. Please note that the use of **NDIF** has no effect on the gas-phase species.