NRL Memorandum Report 4091

CHEMEQ - A Subroutine for Solving Stiff Ordinary Differential Equations

T. R. Young, Jr.

Laboratory for Computational Physics

February 26, 1980

NAVAL RESEARCH LABORATORY
Washington, D.C.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4091	2. GOVT ACCESSION NO. AD-A083 545	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CHEMEQ — A SUBROUTINE FOR SOLVING STIFF ORDINARY DIFFERENTIAL EQUATIONS		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) T. R. Young, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Laboratory for Computational Physics Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NAVMAT/62543N ZF43-451-001 NRL Problem 62-0572-0-0
11. CONTROLLING OFFICE NAME AND ADDRESS CND Directed Funded Programs Naval Research Laboratory Washington, DC 20375		12. REPORT DATE February 26, 1980 13. NUMBER OF PAGES 49
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reactive Flow Problems
Stiff Ordinary Differential Equations
Selected Asymptotic Integration Method
Fortran Code and Variable Documentation and listing
Derivative Function E\aluator
Sample Test Problems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A FORTRAN subroutine (CHEMEQ) which employs the Selected Asymptotic Integration Method (SAIM) for solving the stiff ordinary differential equations associated with reactive flow problems is described and documented. The SAIM algorithm identifies the stiff equations for treatment with a stiffly stable method. The remaining equations are treated with a standard classical method. The algorithm is a very low-overhead method which is particularly efficient when coupled with fluid dynamic calculations. It also gives significant reductions in computational costs for model development. The algorithm, its implementation, the FORTRAN code, the internal

(Continues)

20. Abstract (Continued)

variables and the argument lists are discussed and documented. The application of CHEMEQ to a test problem is also provided as an example.

Contents

I. INTRODUCTION

This report documents a Fortran subroutine called CHEMEQ designed to solve sets of ordinary differential equations of the form:

$$\frac{dn_i}{dt} = F_i = Q_i - L_i n_i \qquad (1)$$

Here Q_i is the formation rate, $L_i n_i$ is the loss rate, and n_i is the density of the ith species. Often the time constants $1/L_i$ for the various species differ by many orders of magnitude and strong coupling between species may be present. If this is the case, the set of equations (1) is considered "stiff" and does not lend itself readily to numerical solution by classical methods.

Subroutine CHEMEQ was developed to apply a specialized numerical technique "The Selected Asymptotic Integration Method"(SAIM) to this class of equations. The method has a very low computational overhead associated with it and is particularly useful when combined with a transport algorithm such as the "Flux Corrected Transport"[2] module to form reactive flow models. In such applications computer memory is at a premium because copies of the chemical species variables are required at every grid point. Since CHEMEQ is a single-step algorithm, multiple copies of the data from several successive timesteps need not be saved. Further, since CHEMEQ is simple- and single-step, no start-up penalty such as evaluating a large Jacobian matrix is exacted at the beginning of an integration step. This is also very important because a reactive flow application requires millions of chemistry integration start-ups.

Note: Manuscript submitted August 14, 1979.

Whenever the hydrodynamic processes in the problem change the variables, the chemistry calculations must be reinitialized.

The efficiency of CHEMEQ is achieved by limiting the actual integration to second-order accuracy to minimize auxiliary storage and start-up expense. In reactive flow applications, however, the reaction rates are seldom known to better than 10% and the hydrodynamics calculations are seldom accurate to better than 1%. Thus integration of the chemistry to better than 1 part in 10^3 or 10^4 is an expensive folly. In this regime CHEMEQ seems to beat the classical methods by about a factor of 50-100 in speed on test problems where start-up is not a consideration. When a coupled hydro application on many grid points is attempted using a parallel processor, up to three orders of magnitude improvement seems possible.

The SAIM method has been applied successfully to such reactive flow problems as high altitude nuclear burst phenomena,[3,4] the solar-induced ionosphere,[5] laser-generated plasma interactions,[6] and the chemical kinetics associated with combustion[7] problems. It has also been used successfully for chemical model development, particle deposition in the ionosphere,[8] and other physical problems where stiff ordinary differential equations arise.

The subroutine CHEMEQ is written in simple standard Fortran but makes extensive use of the pipeline architecture of the NRL Texas Instruments, Advanced Scientific Computer (ASC). The subroutine is easily adapted to other machines without loss of efficiency.

A new subroutine is being prepared called VSAIM (Vectorized Selected Asymptotic Integration Method). This subroutine applies the asymptotic method used in CHEMEQ to several independent sets of equations (i.e., grid points) simultaneously, and thereby it takes full advantage of the parallel processing capability of the ASC. This subroutine is particularly useful for solving the chemical kinetics associated with hydrodynamic applications on computers that have parallel processing capabilities. VSAIM will be documented in a subsequent report.

Sections II and III describe the algorithm and its implementation. Appendix A describes the application of CHEMEQ to various problems. Appendix B gives the fortran listing of the subroutine together with tables of internal and argument list variable definitions. Appendix C gives the results of a sample atmospheric test problem using CHEMEQ together with program listings which illustrate the application of CHEMEQ.

II. ALGORITHM

CHEMEQ integrates a set of coupled ordinary differential equations (which may include "stiff" terms) of the form (1) by a one-step algorithm. The method has very low overhead since all that is required to start a new integration step are the current values of the variables and the derivatives. A second order predictor-corrector method, which takes special notice of those equations determined at the beginning of the step to be stiff is employed to continue the integration process.

The asymptotic integration method applied to the stiff equations best treats the situation where the solution is slowly changing or nearly asymptotic yet the time constants are prohibitively small.

This occurs when the formation rates and loss rates are large, nearly equal, and there is strong coupling between the equations. Thus the stiff equations are treated with a very stable method which damps out the small oscillations caused by the very small time constants. If, however, the formation rates and loss rates are small compared to the function size, the simple classical methods can be utilized for these equations to give the combined method.

The predictor-corrector algorithm provides enough information to choose the subsequent timestep size once convergence has been achieved. For efficiency an initial timestep is chosen which approximates the timestep that will be determined after convergence of the predictor-corrector scheme. This initial trial timestep is chosen independently of the stiffness criterion and is determined such that none of the variables will change by more than a prescribed amount. If the formation rate is much larger than the loss rate, it is reasonable to assume that Q_i and L_i will remain relatively constant for large changes in n_i. Often the initial change in n_i may be large enough to equilibrate the formation and loss rates. Thus the initial trial timestep is chosen in two ways as follows;

$$\delta t = \epsilon \min[n_i/\dot{n}_i, \text{ or } (\text{if } Q_i >> L_i n_i) 1/L_i] \qquad (2)$$

Here ϵ is a scale factor, the same value as the convergence criterion described in Eq. (6). The minimum is taken over the whole set of equations. The timestep chosen by Eq. (2) may be larger than some or all of the equilibration times, in which case the corresponding equations would be classified as stiff. Nevertheless, when solved by the asymptotic method, this timestep ensures that accuracy can be maintained. When a stiff equation is close to equilibrium, the changes in the functional values over the timestep will be small even though the adjustment rate toward equilibrium can be very much shorter than the timestep. When the stiff equation is far from a dynamic equilibrium, the timestep should be scaled down proportionally to the equilibration time to ensure that the transition to equilibrium will be followed accurately. This readjustment, because of the very fast rate, generally takes place very rapidly after which much longer timesteps may be taken.

After a timestep has been chosen, all of the equations are separated into two classes, stiff and normal, according to the criterion.

$$L_i \tau < 1 \qquad \text{Normal}$$

or (3)

$$L_i \tau \geq 1 \qquad \text{Stiff}$$

where the value of τ is problem-dependent and is chosen by the user to invoke asymptotic treatment as necessary. In addition, the user may force asymptotic treatment on any percentage of the set. Equation (3) is applied first. Then, beginning with the equations with the shortest characteristic time $(1/L_i)$ not already chosen by application of equation (3), additional equations are selected with increasing time constants until the percentage of the set specified is satisfied. If the equation is considered stiff at the start of the integration step, it is treated as stiff until the step has been completed. The two types of equations are then integrated by separate predictor-corrector schemes but using a simple asymptotic formula to replace the usual second-order corrector equation for all those equations which were determined to be stiff.

The predictor part of the step is performed as follows:

$$n_i(1) = n_i(0) + \delta t F_i(0) \qquad \text{(Normal)}$$

and (4)

$$n_i(1) = n_i(0) + \frac{\delta t F_i(0)}{1+\delta t L_i(0)} \qquad \text{(Stiff)}$$

where $F_i(0) = F_i[t(0), n_i(0)]$. Here we start at $t = t(0)$ and wish to find $n_i[t(0) + \delta t] \simeq n_i(1)$.

If we let the integer in the parentheses denote the iteration number then $n_i(k)$ is the kth iterated value of n_i, or an approximation to $n_i[t(0) + \delta t]$. The zeroth iteration, $n_i(0)$, is the initial value at $t(0)$ and $n_i(1)$ is the result of the predictor step. Also note that $F_i(k) \equiv F_i[t(0)+\delta t, n_i(k)]$ for the derivatives. The corrector formulas for the two types of equations are:

$$n_i(k+1) = n_i(0) + \frac{\delta t}{2}[F_i(0) + F_i(k)] \qquad \text{(Normal)}$$

and (5)

$$n_i(k+1) = n_i(0) + \frac{2\delta t[Q_i(k)-L_i(0)n_i(0)+F_i(0)]}{4 + \delta t[L_i(k)+L_i(0)]} \qquad \text{(Stiff)}$$

By comparing $n_i(k+1)$ with $n_i(k)$ on successive iterations using the relative error criterion ϵ to satisfy the following equation,

$$1 \geq \sigma = \max \left[\frac{|n_i(k+1) - n_i(k)|}{n_i(k+1)\epsilon} \right] , \tag{6}$$

the convergence of each of the individual equations can be determined. As applied in CHEMEQ, ϵ is typically $\sim 10^{-3}$ and if the formation and loss rates are nearly equal σ will be scaled down slightly. This allows quicker convergence for equations that are nearly in equilibrium.

In practice n_i is constrained by a minimum value when n_i is decaying exponentially toward zero. This lower bound is chosen by the user and must be carefully selected to insure that its value in no way affects the physics but yet decouples the equation from accurate integration. Decoupling is accomplished by skipping the application of equation (6) to all equations that have decayed to values corresponding to their lower bounds. Convergence for these equations is then trivial and the function no longer affects the size of the timestep.

For equations that are decaying exponentially to zero with time constants that are small enough to control the timestep,it is important for efficiency reasons to decouple these equations at the largest lower bound possible. However, it must be remembered that spurious results may occur in other equations sensitive to the limited equations if their lower bounds are too large. This results because the value of the function after decoupling is frozen at the lower bound for the duration of the integration process or until the total rate becomes positive. If there is any question, it is better to be on the conservative side by choosing the minimum values smaller than necessary. This may result in a little loss of efficiency but will reduce the possibility of erroneous results.

We have found that maximum speed is realized by keeping the allowed number of iterations on the corrector small. We typically use one or two. If satisfactory convergence of all equations has not been obtained before or during the last iteration, the step is started over with a smaller timestep. By keeping the maximum number of iterations small, a minimum amount of time is wasted on an unstable or nonconvergent step only to find out that the iteration procedure did not converge. By the same token, we have found it best to reduce the timestep sharply (a factor of 2 or 3) when nonconvergence is encountered rather than to reduce it slowly. Less time is wasted this way getting down to a sufficiently small step for convergence if the initial estimated step size is found to be too large. On the other hand, when increasing the timestep, as for example when convergence is achieved on the first or second iteration, we have found it best to only increase by 5-10% each step. During the integration of several successive steps, we use the appropriately modified timestep from one converged integration cycle as the trial timestep for the next integration cycle rather than using Eq. 2. The timestep modification is performed as follows

$$\delta t = \delta t \left[\frac{1}{\sqrt{J}} + .005\right] \qquad (7)$$

Using σ as the starting value, the $\sqrt{\sigma}$ is estimated with three iterations of Newton's method. This gives the desired asymmetrical property in that δt decreases faster than δt would increase for the inverse value of σ. In addition, δt is modified very little when σ is near 1.

Once convergence of all the equations is achieved, the new values of the $n_i(\delta t)$ are set equal to the values of $n_k(k+1)$. One can obtain convergence and completion of an integration step after only two derivative-function evaluations even when some or all of the equations are stiff.

III. HOW TO USE CHEMEQ

The Selected Asymptotic Integration Algorithm, as described in Section II, has been coded in Fortran which may be implemented on any digital computer of moderate size. It is intended as a very fast but moderately accurate integrator which can be used at each grid point of a large hydro- or magnetohydrodynamic calculation. Single point calculations are easily and efficiently accomplished by CHEMEQ as well.

CHEMEQ has four entries which are available to perform the various aspects of the integration. The main entry is used for normal operation. The others provide flexibility and optional controls. The variables in the argument lists and internal variables are documented in detail in Appendix B.

1. CHEMEQ (DTCHEM, DFE, N, F, FMIN) advances the equations the specified interval DTCHEM.

2. CHEMSP (EPSMN, EPSMX, DTMN, TNOT, PASS, TASS, PRT) resets the specified control parameters if the default values are not satisfactory.

3. CHEMCT (TMK) is for information purposes. This entry prints information which indicates how efficiently the integration process has been since either the last call to CHEMSP or the last call to CHEMCT.

4. CHEMPR is for diagnostic purposes. This entry may be called whenever an error occurs which can be attributed to the results of CHEMEQ. A partial set of the internal variables is printed as a diagnostic.

CHEMEQ is the main entry and is called to advance the equations as required. The initial values are passed in as arguments. After being advanced by the integration they are passed back in the same place. One of the arguments of CHEMEQ is the name of the derivative function subroutine DFE, utilizing a useful feature of Fortran which gives the user the option of specifying various configurations for the derivative functions within the confines of the same problem.

CHEMSP is called whenever any or all the default values of the control parameters in the argument list are not satisfactory. Variables such as the initial value of the independent variable, the absolute minimum timestep allowed, control parameters for convergence of the predictor-corrector combination, and the control parameters which affect the use of asymptotics may be reset here.

CHEMCT is called for diagnostic purposes. It displays information on the numbers of derivative function evaluator calls and the number of times asymptotics were employed. It also gives the number of times the integration step had to be restarted with a smaller timestep due to lack of convergence of the predictor-corrector scheme since the last call to CHEMCT, CHEMSP, or since the beginning. This information can be very helpful in determining the relative efficiency of the integration process.

CHEMPR is called for diagnostic purposes. If an error in the integration process is suspected, the user may call this entry to print out some of the internal variables. The current values of the production rates [C(I)], loss rates [D(I)], functions [F(I)], inverse time constants [RTAU(I)], total rates [CMD], estimated timestep required, from the previous step the total rates [DFS(I)], the functional values [FS(I)], the initial functional values [FO(I)] and the minimum values [FMIN(I)] are printed for diagnostic purposes.

Two subroutines are referenced from CHEMEQ.

1. DFE(F, C, D, T) The Derivative Function Evaluator which calculates the derivatives $\{\dot{n}_i\}$ as required.

2. CHEMER in the subroutine that is called whenever CHEMEQ determines that an error has occurred.

DFE, the derivative function evaluator, must be supplied by the user to provide on request by the integrator, the current derivatives $\{\dot{n}_i\}$. It is important to note that nearly all of the computer time spent in the integration process for most problems is spent in this user-supplied routine. It is therefore extremely valuable to put the extra effort into optimizing this routine, especially, when it will be incorporated into a large hydrodynamic code. Here are some suggestions for coding which may help produce efficient operation.

1. Avoid all unnecessary repetitive calculations. Quantities which can be calculated once should be stored for subsequent use. In particular, divisions and mathematical functions should be avoided since these are costly operations on most machines.

2. Replace all complicated functions with table look-ups whenever possible. This can be a very important economy measure.

3. Arrange the code in a fashion which takes advantage of your computer's optimization features. For example, the use of register to register or parallel processing capabilities.

4. The user may often take advantage of the structure of the problem he is working with.

For example, in a large atmospheric reactive-flow hydrodynamic code, the density values may vary drastically from the bottom of the grid to the top. Often at the top a much simpler reaction scheme will be sufficient to describe the chemistry than in the middle or lower portions of the mesh. Here the user may specify various configurations of the reaction scheme appropriate to the grid region and save a significant amount of computation. There are other ways to improve efficiency but they may not be as obvious as these listed. Often with a little imagination and persistence combined with a thorough knowledge of the problem area, significant improvements in computational efficiency can be realized.

CHEMER is called whenever CHEMEQ determines that a severe error has occurred. Currently the only error which can be identified by CHEMEQ is when the timestep becomes too small. CHEMEQ at this point provides output that may be useful and then calls CHEMER. The default version of CHEMER does nothing but print a message indicating that CHEMER has been called and then stops execution. However, the user may supply his own version of CHEMER which could provide printout of a much more complete set of diagnostics than CHEMEQ does or manipulate the data in such a fashion that the integration process might proceed.

The actual arguments and internal variables used in CHEMEQ, its entries and the associated subroutines will be described in detail in the appendix sections of this report.

IV. SUMMARY

CHEMEQ is intended to be a general purpose integrator for a *specific type* of equations. It employs a very low-overhead, moderately accurate, low-order technique. To obtain results for most physical models with an acceptable degree of accuracy, CHEMEQ can be extremely efficient. In many areas where problems are so computationally expensive they seem impossible to do by other methods, CHEMEQ gives accurate results in a reasonable amount of time. CHEMEQ can also be employed in the development of chemical or mathematical models when efficiency is not so important, but the user should not expect eight figures of accuracy. Two or three figures over a long integration is a more realistic estimate. CHEMEQ's forte lies in the solution of the stiff ordinary differential equations associated with chemically reactive flow problems. Here the reaction rates are split off from the hydrodynamic part of the equations and solved separately for each hydrodynamic timestep and at each grid point. The moderate accuracy of the methods used to solve the hydrodynamic equations suggest that the application of a more sophisticated technique, rather than a low-order, low-overhead method like CHEMEQ, would waste valuable computer time and could possibly render the problem so computationally inefficient that it would be impractical to pursue.

A potential user must be aware that CHEMEQ is not user-proof, problem-independent and can not always be used as a black box. The method is not identially conservative for arbitrarily large timesteps when asymptotics are employed and the minimum values should be chosen with some thought since they can become sources of spurious errors if not chosen small enough initially.

All methods, such as the selected asymptotic integration method. which do not conserve particle density or charge balance automatically may be forced to do so by at least two techniques. In one technique, conservation can be restored by adding the various concentrations to find the errors and then by distributing these errors throughout the densities in a number-conserving manner. The major fault with this is that a portion of the errors is incorporated into concentrations from which the errors may not have arisen. The second and better method is to reduce the frequency of the asymptotic treatment or decrease the timestep size to the point where errors due to nonconservation are within tolerable limits. Significant improvement in computational efficiency still results.

CHEMEQ is written in standard Fortran and should be easily adaptable to any computer that accepts Fortran. Although the present version is written in a fashion that promotes vectorization by the ASC, no special features of the ASC Fortran were incorporated into the code.

The storage requirements of CHEMEQ are proportional to the maximum number of equations for which storage has been reserved. For a maximum of 25 equations CHEMEQ requires about 2000 words of memory on the ASC.

Since CHEMEQ uses a convergence-dependent algorithm and an adaptive timestep, the overall timing will be strictly problem-dependent. It will depend on such things as the coupling between and relaxation times of the equations. As mentioned before, most of the integration time will be spent in the derivative function evaluations of which there are at least two required per CHEMEQ call. At least 50 μsec of ASC CPU time are required as integrator overhead per integration step per equation. This does not count the time required to evaluate the derivatives.

If CHEMEQ is applied as intended, the subroutine can solve large systems of stiff ordinary differential equations more efficiently than methods currently available. In some cases, its efficiency is unrivaled.

ACKNOWLEDGMENTS

I would like to acknowledge Jay Boris' contributions in the development of the selected asymptotic integration method. His penetrating insights into this problem have been invaluable in the development of this technique over the past few years.

This research was originally begun under Defense Nuclear Agency support and completed under CND Directed Funded programs through the Naval Material Command under Project ZF-43-451-001.

REFERENCES

1. T. R. Young and J. P. Boris,"A Numerical Technique for Solving Stiff Ordinary Differential Equations Associated with the Chemical Kinetics of Reactive-Flow Problems." J. Physical Chemistry, 81, 2424 (1977).

2. J. P. Boris,"Flux-Corrected Transport Modules for Solving Generalized Continuity Equations"and references therein, NRL Memorandum Report No. 3237 (1976).

3. J. P. Boris, B. E. McDonald, T. P. Coffey, and T. R. Young, Proceedings of the DNA High Altitude Nuclear Effects Symposium, Vol. 2 DASIAC SR-130 (DEC. 1971).

4. J. P. Boris, Proceedings of the 2nd European Conference on Computational Physics.

5. E. S. Oran, T. R. Young, D. V. Anderson, T. P. Coffey, and P. C. Kepple, A. W. Ali, and D. F. Strobel,"A Numerical Model of the Mid-Latitude Ionosphere," NRL Memorandum Report No. 2839 (1974).

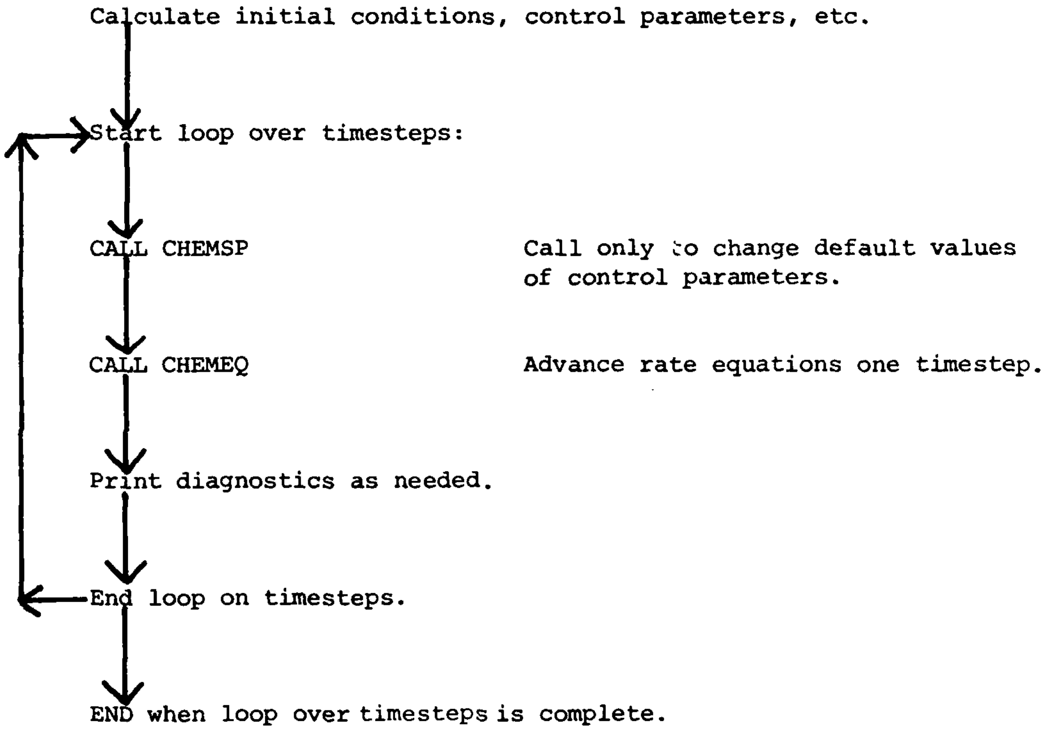
6. K. G. Whitney and J. Davis, "Hot-Spot Model of K-Line Emission from Laser-Heated Plasmas," J. Applied Physics, Vol. 45, No. 12, pp 5294-5302, (Dec. 1974).

7. Elaine Oran, Theodore Young, and Jay Boris, "Application of Time-Dependent Numerical Methods to the Description of Reactive Shocks," submitted for publication in the Seventeenth Symposium (International) on Combustion (Aug. 1978).

8. E. S. Oran, and D. J. Strickland, "Photo-Electron Flux in the Earth's Ionosphere," Planet, Space Sci. Vol. 26 (1978).

APPENDIX A

Table A1. Logical Sequence of Calls for Chemical Kinetics *Without Transport*



Calculate initial conditions, control parameters, etc.

Start loop over timesteps:

CALL CHEMSP — Call only to change default values of control parameters.

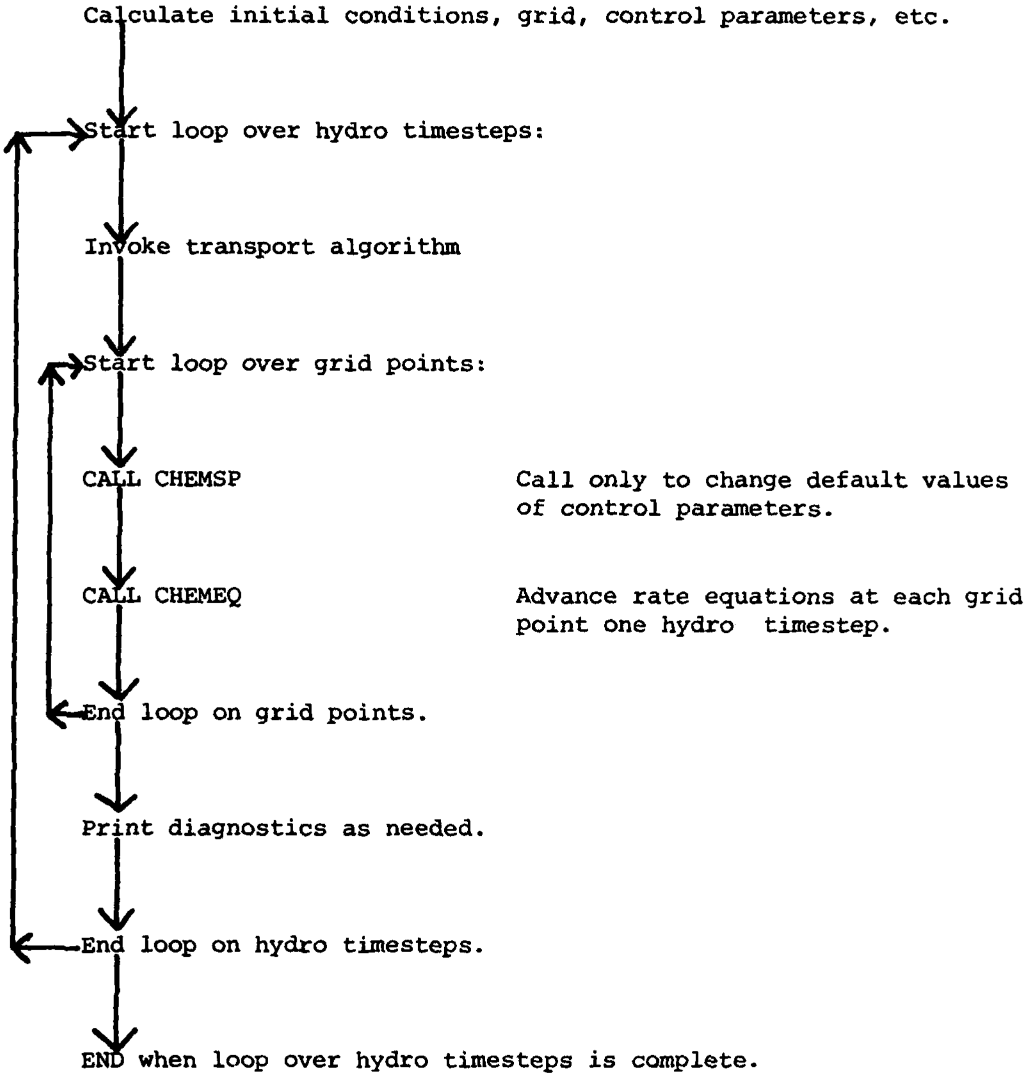
CALL CHEMEQ — Advance rate equations one timestep.

Print diagnostics as needed.

End loop on timesteps.

END when loop over timesteps is complete.

Table A2. Logical Sequence of Calls for Chemical Kinetics and Transport Combined



Calculate initial conditions, grid, control parameters, etc.

Start loop over hydro timesteps:

Invoke transport algorithm

Start loop over grid points:

CALL CHEMSP — Call only to change default values of control parameters.

CALL CHEMEQ — Advance rate equations at each grid point one hydro timestep.

End loop on grid points.

Print diagnostics as needed.

End loop on hydro timesteps.

END when loop over hydro timesteps is complete.

APPENDIX B

LISTING OF CHEMEQ

```
      SUBROUTINE CHEMEQ(DTCHEM, DFE, N, F, FMIN)
C
C
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    CHEMEQ (DTCHEM, DFE, N, F, FMIN)
CD
CD    ORIGINATORS:  T.R. YOUNG AND J.P. BORIS                    NRL 1971
CD
CD    DESCRIPTION:  CHEMEQ IS A SUBROUTINE WHICH SOLVES A CLASS OF
CD    ORDINARY DIFFERENTIAL EQUATIONS TERMED STIFF. THESE EQUATIONS
CD    CANNOT BE READILY SOLVED BY THE STANDARD CLASSICAL METHODS THUS
CD    THE SELECTED ASYMPTOTIC INTEGRATION METHOD IS EMPLOYED BY CHEMEQ.
CD    THE EQUATIONS ARE DIVIDED INTO TWO CATAGORIES BASED ON
CD    EQUILIBRATION TIMES AND ARE INTEGRATED BY EITHER A LOW ORDER
CD    CLASSICAL METHOD FOR THE EQUATIONS WHICH HAVE LONG EQUILIBRATION
CD    TIMES OR A VERY STABLE STEP-CENTERED METHOD WHICH HELPS
CD    PRESERVE THE ASYMPTOTIC NATURE OF THE SOLUTIONS WHEN
CD    EQUILIBRATION TIMES ARE VERY SHORT. AN ADAPTIVE STEPSIZE IS
CD    CHOSEN TO GIVE ACCURATE RESULTS FOR THE FASTEST CHANGING QUANTITY.
CD    THE ROUTINE ASSUMES THAT ALL OF THE INTEGRATED QUANTITES AND THE
CD    TIME STEP ARE POSITIVE.
CD
CD    ARGUMENT LIST DEFINITION:
CD    DTCHEM              REAL*4     THE INTERVAL OF INTEGRATION OR THE   I
CD                                   RANGE OF THE INDEPENDENT VARIABLE.
CD                                   0.0 <= T <= DTCHEM.
CD    DFE                 REAL*4     THE NAME OF THE DERIVITIVE FUNCTION  I
CD                                   EVALUATOR SUBROUTINE.
CD    N                   INTEGER    THE NUMBER OF EQUATIONS TO BE        I
CD                                   INTEGRATED. AN ERROR EXISIS IF N IS
CD                                   GREATER THAN NO SET BY THE PARAMETER
CD                                   STATEMENT.
CD    F(N)                REAL*4     THE INITIAL VALUES AT CALL TIME    I/O
CD                                   AND THE FINAL VALUES AT RETURN TIME.
CD    FMIN(N)             REAL*4     MINIMUM VALUES FOR EACH FUNCTION.    I
CD
CD    LANGUAGE AND LIMITATIONS:  ALTHOUGH THIS SUBROUTINE IS WRITTEN IN
CD    A FASHION WHICH PROMOTES VECTORIZATION BY THE ASC COMPILER, THE
CD    FORTRAN IS NEARLY STANDARD  AND SHOULD WORK WITH MINOR MODIFICAT-
CD    IONS ON ANY MACHINE.
CD
CD    ENTRY POINTS:  FOUR ENTRY POINTS ARE PROVIDED FOR FLEXIBILITY AND
CD    OPTIMUM CONTROL.
CD
CD    CHEMEQ:  ADVANCES THE EQUATIONS THE GIVEN INCREMENT 'DTCHEM'.
CD
```

```
CD     CHEMCT;  INFORMATIVE, PRINTS THE VALUES OF THE INDICATIVE
CD              COUNTERS LISTED BELOW;
CD              1.  THE NUMBER OF TIMES ASYMPTOTICS WERE USED.
CD              2.  THE NUMBER OF DERIVATIVE FUNCTION EVALUATIONS.
CD              3.  THE NUMBER TIMES THE INTEGRATION STEP WAS RESTARTED
CD                  DUE TO NONCONVERGENCE OF THE PREDICTOR-CORRECTOR
CD                  SCHEME.
CD
CD     CHEMSP;  PROVIDES THE USER WITH THE OPTION TO RESET THE MOST
CD              IMPORTANT CONTROL PARAMETERS.
CD
CD     CHEMPR;  INFORMATIVE, PRINTS OUT INTERNAL VARIABLES FOR DIAGNOSTIC
CD              PURPOSES.
CD
CD     SUBROUTINES REFERENCED;
CD
CD     DFE;     WHOSE ACTUAL NAME AND DEFINITION ARE SUPPLIED BY THE USER
CD              IS CALLED TO OBTAIN THE DERIVITIVE FUNCTIONS.
CD
CD     CALL DFE(F, C, D, T)
CD     ARGUMENT LIST TO DFE;
CD     F(N)                REAL*4    CURRENT VALUES OF THE DEPENDENT       I
CD                                   VARIABLE.
CD     C(N)                REAL*4    CALCULATED FORMATION RATES.           O
CD     D(N)                REAL*4    CALCULATED LOSS RATES.                O
CD     T                   REAL*4    CURRENT VALUE OF THE INDEPENDENT      I
CD                                   VARIABLE.
CD
CD     CHEMER; IS CALLED WHENEVER AN ERROR IS DETECTED. CURRENTLY THE
CD            ONLY ERROR RECOGNIZED IS A TIME STEP THAT IS TOO SMALL.
CD
CD     CALL CHEMER
CD     ARGUMENT LIST TO CHEMER;   NO ARGUMENTS.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       PARAMETER    ND = 150
C
       REAL*8       TS, TN
C
       REAL         F(N/ND/), FMIN(N/ND/), C(ND), D(ND), RTAUS(ND)
       REAL         FS(ND), DFS(ND), FC(ND), SCRA(ND), SCRB(ND), RTAU(ND)
       REAL         ASY(ND), COR(ND)
C
       INTEGER      FCOUNT, ACOUNT, RCOUNT, TFCNT, TACNT, TRCNT
C
       DATA         FCOUNT, ACOUNT, RCOUNT, TFCNT, TACNT, TRCNT/6*0/
```

```
      DATA          PASYI/0.00/, TCRASY/100.0/, NDD/ND/, EPSCL/100.0/
      DATA          TFD/Z41100008/, DTMIN/1.0E-15/, SQREPS/0.50/
      DATA          TSTART, DT, DTTEST/3*0.0/, TN/0.0D+00/, C/ND*0.0/
      DATA          EPSMAX/10.0/, LO/6/, EPSMIN/1.0E-02/, D/ND*0.0/
C
C     TEMPERARY FIX:  SEE DO LOOP 130 ! ! ! !
      DATA          SIGNM/Z80000000/
C
C     CHECK INPUT PARAMETERS.
          IF(N .LE. NDD)GO TO 110
          WRITE(LO, 1002) N, NDD
 1002     FORMAT(5(/),' FROM -CHEMEQ-  :  NO. OF EQ.S REQUESTED IS TOO',
         ' LARGE'/' REQUESTED (',I5,'), MAX. ALLOWED (',I5,')')
          STOP
C
C     INITIALIZE THE CONTROL PARAMETERS.
  110     TN = 0.0D+00
          DTTARG = DTCHEM
C
C     STORE AND LIMIT TO 'FMIN' THE INITIAL VALUES.
          DO 1 I = 1, N
          C(I) = 0.0
          D(I) = 0.0
          FO(I) = F(I)
    1     F(I) = AMAX1(F(I), FMIN(I))
C
C     EVALUATE THE DERIVITIVES OF THE INITIAL VALUES.
          CALL DFE(F, C, D, SNGL(TN + TSTART))
          FCOUNT = FCOUNT + 1
C
C     ESTIMATE THE INITIAL STEPSIZE.
C
C     STRONGLY INCREASING FUNCTIONS(C >>> D ASSUMED HERE) USE A STEP-
C     SIZE ESTIMATE PROPORTIONAL TO THE STEP NEEDED FOR THE FUNCTION TO
C     REACH EQUILIBRIUM WHERE AS FUNCTIONS DECREASING OR IN EQUILIBRIUM
C     USE A STEPSIZE ESTIMATE DIRECTLY PROPORTIONAL TO THE CHARACTER-
C     ISTIC STEPSIZE OF THE FUNCTION. CONVERGENCE OF THE INTEGRATION
C     SCHEME IS LIKELY SINCE THE SMALLEST ESTIMATE IS CHOSEN FOR THE
C     INITIAL STEPSIZE.
          SCRTCH = 1.0E-40
          DO 15 I = 1, N
          SCRA(I) = .1*EPSMIN*ABS(C(I)) - D(I)
          SCRB(I) = SIGN(1.0/F(I), SCRA(I))
          SCRA(I) = SCRB(I)*C(I)
          SCRB(I) = -ABS(ABS(C(I)) - D(I))*SCRB(I)
          SCRA(I) = AMAX1(SCRA(I), SCRB(I))
   15     SCRTCH = AMAX1(SCRA(I), SCRTCH)
```

```
          DT = SQREPS/SCRTCH
          IF(DT .LT. TCRASY)DT = SQRT(DT*TCRASY)
          IF(DT .GT. DTCHEM)DT = DTCHEM
C
C     THE STARTING VALUES ARE STORED.
  100     TS = TN
C
C     ASSIGN ASYMPTOTIC OR NORMAL TREATMENT FOR EACH EQUATION AT THE
C     BEGINING OF EACH STEP.
          NASY = ACOUNT
C
C     EQUATIONS WITH TOO SHORT A CHARACTERISTIC STEPSIZE ARE SELECTED
C     FOR ASYMPTOTIC TREATMENT.
          DO 130 I = 1,N
          RTAU(I) = C(I)/F(I)
          FS(I) = F(I)
          DFS(I) = C(I) - D(I)
          SCRA(I) = RTAU(I) - TCRASY
C
C     THE FOLLOWING TWO CARDS REPLACE THE THIRD WHICH DOES NOT COMPILE
C     PROPERLY ON NX - 5.027.139.
          SCRB(I) = AND(SIGNM, SCRA(I))
          ASY(I) = .5 + OR(.5, SCRB(I))
C         ASY(I) = .5 + SIGN(.5, SCRA(I))
          COR(I) = DFS(I) - D(I)*ASY(I)
          RTAUS(I) = RTAU(I)*ASY(I)
  130     ACOUNT = ACOUNT + ASY(I)
          NASY = PASYI*N - ACOUNT + NASY
          IF(NASY .LT. 1)GO TO 101
C
C     COMPLETE THE SELECTION OF EQUATIONS FOR ASYMPTOTIC TREATMENT UP TO
C     THE PERCENTAGE 'PASYI'. EQUATIONS WITH THE SHORTEST CHARACTERISTIC
C     STEPSIZES ARE CHOSEN FIRST.
          DO 20 I = 1,NASY
          RTMX = -1.0E+70
          DO 35 J = 1,N
          IF(ASY(I) .GT. 0.1)GO TO 35
          IF(RTMX .GT. RTAU(J))GO TO 35
          RTMX = RTAU(J)
          JS = J
   35     CONTINUE
          ASY(JS) = 1.0
          RTAUS(JS) = RTAU(JS)
          COR(JS) = COR(JS) - D(JS)
   20     ACOUNT = ACOUNT + 1
C
C     FIND THE PREDICTOR TERMS.
```

```
  101     DO 5 I = 1,N
C
C     THE FIRST ORDER PREDICTION FOR THE ASYMPTOTIC FUNCTIONS REDUCES TO
C     EULER'S METHOD FOR THE NONASYMPTOTIC FUNCTIONS IF "RTAU" = 0.0.
          SCRB(I) = DFS(I)/(1.0 + DT*RTAUS(I))
    5     CONTINUE
C
C     LIMIT DECREASING FUNCTIONS TO THEIR MINIMUM VALUES.
          DO 105 I = 1,N
          SCRB(I) = FS(I) + DT*SCRB(I)
  105     F(I) = AMAX1(SCRB(I), FMIN(I))
          TN = TS + DT
C
C     EVALUATE THE DERIVITIVES FOR THE CORRECTOR.
          CALL DFF(F, C, D, SNGL(TN + TSTART))
          FCOUNT = FCOUNT + 1
          EPS = 1.0E-10
          DO 4 I = 1,N
C
C     STEP CENTERED CORRECTOR FOR THE ASYMPTOTIC FUNCTIONS REDUCES TO
C     THE MODIFIED EULER METHOD FOR THE NONAYSMPTOTIC FUNCTIONS.
          SCRB(I) = ASY(I)*D(I)
          RTAU(I) = SCRB(I)/F(I)
    4     SCRB(I) = (COR(I) + C(I) + (SCRB(I) - D(I)))
     .               / (2.0 + .5*DT*(RTAU(I) + RTAUS(I)))
C
C     CALCULATE NEW F, CHECK FOR CONVERGENCE, AND LIMIT DECREASING
C     FUNCTIONS. THE ORDER OF THE OPPERATIONS IN THIS LOOP IS IMPORTANT.
          DO 6 I = 1,N
          SCRB(I) = AMAX1(FS(I) + DT*SCRB(I), 0.0)
          SCRA(I) = ABS(SCRB(I) - F(I))
          F(I) = AMAX1(SCRB(I), FMIN(I))
          SCRA(I) = SCRA(I)/F(I)
C
C     SCALE RELATIVE ERROR DOWN WHEN C & D ARE NEARLY EQUAL.
          SCRB(I) = ABS(C(I) - D(I))/(C(I) + D(I) + 1.0E-30)
          SCRB(I) = AMIN1(SCRB(I), SCRA(I))
          SCRA(I) = SCRA(I) + SCRB(I)
C
C     REMOVE RELATIVE ERROR CONTRIBUTION IF FUNCTION VALUE IS LESS THAN
C     THE MINIMUM VALUE.
          SCRB(I) = .25*(FS(I) + F(I)) - FMIN(I)
          SCRB(I) = .25 + SIGN(.25, SCRB(I))
          SCRA(I) = SCRB(I)*SCRA(I)
    6     EPS = AMAX1(SCRA(I), EPS)
          EPS = EPS*EPSCL
C
```

```
C     PRINT OUT DIANOSTICS IF STEPSIZE BECOMES TOO SMALL.
          IF(DT .GT. DTMIN + 1.0E-16*TN)GO TO 40
          WRITE(LO, 1003) DT, TN, DTMIN
          DO 25 L = 1,N
          CMD = C(L) - D(L)
          DTC = EPSMIN*F(L)/(ABS(CMD) + 1.0E-30)
   25     WRITE(LO, 1004) C(L), D(L), F(L), RTAU(L), CMD, DTC, DFS(L),
     .    FS(L), FO(L), FMIN(L)
 1003     FORMAT('1    CHEMEQ ERROR;   STEPSIZE TOO SMALL ! ! !', /,
     1    '     DT = ', 1PE10.3, ' TN = ', D25.15,
     2    ' DTMIN = ',E10.3, //, 14X, 'C', 9X, 'D', 9X, 'F', 6X, 'RTAU',
     3    5X, 'C - D       DTC        DFS', 8X, 'FS', 8X, 'FO      FMIN')
 1004     FORMAT(5X, 1P12E10.3)
          DT = DTCHEM - TS
          DT = AMIN1(DTMIN, ABS(DT))
C
C     CALL ERROR DIAGNOSTIC ROUTINE
          CALL CHEMER
C
C     CHECK FOR CONVERGENCE.
   40     IF(EPS .GT. EPSMAX)GO TO 30
C
C     END CHECK.
          DTTARG = DTCHEM - TN*TFD
          IF(DTTARG .GT. 0.0)GO TO 10
          RETURN
C
C     PERFORM STEPSIZE MODIFICATIONS.
   30     RCOUNT = RCOUNT + 1
          TN = TS
C
C     ESTIMATE SQRT(EPS) BY NEWTON ITERATION.
   10     RTEPS = .5*(EPS + 1.0)
          DO 50 J = 1,2
   50     RTEPS = .5*(RTEPS + EPS/RTEPS)
          DT = DT*(1.0/RTEPS + .005)
          DT = AMIN1(DT, SNGL(TFD*(DTCHEM - TN)))
C
C     BEGIN NEW STEP IF PREVIOUS STEP CONVERGED.
          IF(EPS .GT. EPSMAX)GO TO 101
          CALL DFE(F, C, D, SNGL(TN + TSTART))
          FCOUNT = FCOUNT + 1
          GO TO 100
C
C
      ENTRY CHEMCT (TMK)
C
```

```
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    CHEMCT (TMK)
CD    WRITE OUT THE VALUES OF THE VARIOUS INDICATIVE COUNTERS THAT THE
CD    PROGRAM KEEPS.
CD
CD    ARGUMENT LIST DEFINITION:
CD    TMK                 REAL*4    A FLOATING POINT NUMBER PRINTED      I
CD                                  TO IDENTIFY THE CALL.
CD
CD    OUTPUT VARIABLE DEFINITION:
CD    TMK                 REAL*4    FLOATING POINT IDENTIFIER.
CD    FCOUNT              INTEGER   NUMBER OF DERIVATIVE SUBROUTINE CALLS
CD                                  SINCE THE LAST CALL.
CD    ACOUNT              INTEGER   NUMBER OF TIMES THE ASYMPTOTIC TREAT-
CD                                  MENT WAS USED SINCE THE LAST CALL.
CD    RCOUNT              INTEGER   NUMBER OF TIMES STEPSIZE WAS REDUCED
CD                                  SINCE LAST CALL.
CD    TFCNT               INTEGER   TOTAL OF FCOUNT TO THIS CALL.
CD    TACNT               INTEGER   TOTAL OF ACOUNT TO THIS CALL.
CD    TRCNT               INTEGER   TOTAL OF RCOUNT TO THIS CALL.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      TFCNT = TFCNT + FCOUNT
      TACNT = TACNT + ACOUNT
      TRCNT = TRCNT + RCOUNT
C
C     PRINT OUT INDICATIVE COUNTERS.
      WRITE(LO, 1000) TMK, FCOUNT, ACOUNT, RCOUNT, TFCNT, TACNT,
     .      TRCNT
 1000 FORMAT(' CHEMEQ INDICES: TMK = ', 1PE10.3,
     .      ' FCOUNT, ACOUNT, RCOUNT = ', 3I7, ' TOTALS: ', 3I7)
C
C     RESET COUNTERS.
      FCOUNT = 0
      ACOUNT = 0
      RCOUNT = 0
      RETURN
C
C
      ENTRY CHEMSP(EPSMN, EPSMX, DTMN, TNOT, PASY, TASY, PRT)
C
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    CHEMSP(EPSMN, EPSMX, DTMN, TNOT, PASY, TASY, PRT)
CD
```

```
CD    RESET ANY LOCAL CONTROL PARAMETERS IF THEIR RESPECTIVE INPUT
CD    VALUES ARE GREATER THAN ZERO. DEFAULT VALUES ARE USED IF THE
CD    INPUT VALUES ARE ZERO OR LESS REPECTIVELY.
CD
CD    ARGUMENT LIST DEFINITION:
CD    EPSMN             REAL*4    THE MAXIMUM RELATIVE ERROR ALLOWED   I
CD                                FOR CONVERGENCE OF THE CORRECTOR STEP.
CD                                DEFAULT VALUE:  1.0E-02
CD    EPSMX             REAL*4    THIS NUMBER PROVIDES THE BASIS FOR   I
CD                                DECIDING WEATHER CONVERGENCE CAN BE
CD                                ACHIEVED WITH OUT ADDED STEPSIZE
CD                                REDUCTION. IF EPS/EPSMIN IS GREATER
CD                                THAN EPSMX FURTHER REDUCTION IS
CD                                APPLIED.
CD                                DEFAULT VALUE : 10.0
CD    DTMN              REAL*4    THE SMALLEST STEPSIZE ALLOWED.       I
CD                                DEFAULT VALUE:  1.0E-15
CD    TNOT              REAL*4    THE INITIAL VALUE OF THE INDEPENDENT I
CD                                VARIABLE T.
CD                                DEFAULT VALUE:  0.0
CD    PASY              REAL*4    THE PERCENTAGE OF THE EQUATIONS FOR  I
CD                                WHICH ASYMPTOTICS WILL ALWAYS BE
CD                                APPLIED. EQUATIONS WITH THE SMALLEST
CD                                CHARACTERISTIC STEPSIZE ARE CHOSEN
CD                                FIRST.
CD    TASY              REAL*4    ASYMPTOTICS ARE APPLIED IF THE CHAR- I
CD                                ACTERISTIC STEPSIZE OF AN EQUATION IS
CD                                LESS THAN TASY.
CD                                DEFAULT VALUE:  1.0E-02.
CD    PRT               REAL*4    CONTROLS THE OUTPUT OF CHEMSP.  ANY  I
CD                                NON ZERO VALUE SUPPRESSES ALL PRINT
CD                                OUTPUT FROM THIS ENTRY.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
          EPSMIN = 1.0E-02
          IF(EPSMN .GT. 0.0)EPSMIN = EPSMN
          IF(EPSMN .GT. 0.0)SQREPS = 5.0*SQRT(EPSMIN)
          EPSCL = 1.0/EPSMIN
          EPSMAX = 10.0
          IF(EPSMX .GT. 0.0)EPSMAX = EPSMX
          DTMIN = 1.0E-15
          IF(DTMN .GT. 0.0)DTMIN = DTMN
          TSTART = TNOT
          PASYI = 0.0
          IF(PASY .GT. 0.0)PASYI = .01*(PASY + .5)
          TCRASY = 100.0
          IF(TASY .GT. 0.0)TCRASY = 1.0/TASY
C
C     PRINT NEW VALUES OF CONTROL PARAMETERS.
          IF(PRT .EQ. 0.0)
     .    WRITE(LO, 1001) EPSMN, EPSMX, DTMN, TNOT, PASY, TASY
 1001     FORMAT(' INITALIZE "CHEMEQ" VIA "CHEMSP"', /,
     .    ' EPSMN, EPSMX, DTMN, TNOT, PASY, TASY = ', 1P6G10.3)
          RETURN
C
      ENTRY CHEMPR
C
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    CHEMPR MAY BE CALLED WHEN EVER AN ERROR OCCURS THAT CAN BE
CD    ATTRIBUTED TO THE RESULTS OF CHEMEQ. A PARTIAL SET OF THE INTERNAL
CD    VARIABLES IS PRINTED AS A DIAGNOSTIC.
CD
CD* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
          WRITE(LO, 1003) DT, TN, DTTEST
          DO 45 L = 1,N
          CMD = C(L) - D(L)
          DTC = EPSMIN*F(L)/(ABS(CMD) + 1.0E-30)
   45     WRITE(LO, 1004) C(L), D(L), F(L), RTAU(L), CMD, DTC, DFS(L),
     .    FS(L), FO(L), FMIN(L)
C
          RETURN
      END
```

LISTING OF CHEMER

```
      SUBROUTINE CHEMER
C
C     DIAGNOSTIC ROUTINE FOR STIFF O.D.E. SOLVER  -CHEMEQ-
C
          PRINT 1001
 1001     FORMAT(5(/), ' LIBRARY VERSION OF -CHEMER- CALLED.', /,
     .    ' USERS MAY SUPPLY THEIR OWN VERSION FOR DIAGNOSTICS.', /,
     .    ' NO ARGUMENTS ARE REQUIRED.', /,
     .    ' PROGRAM WILL CONTINUE RESETTING THE STEP SIZE TO MIN-', /,
     .    'IMUMS IF A NORMAL RETURN IS MADE.', //,
     .    ' (STOP 69) EXECUTED FROM LIBRARY VERSION OF -CHEMER-')
C
          STOP 69
      END
```

These subroutines may be punched onto cards directly from the program listing. On computers other than the TI ASC the PARAMETER statement sould be removed and occurrences of ND in the declarations should be replaced by a fixed point number at least as large as the largest set of equations to be integrated. This subroutine should be complied on the K level optimization on the ASC unless the number of equations expected is small. Then the J level compilation will give the most efficient code. No other compilation options are required on the ASC.

Table B1. Mnemonic Correspondence of Mathematical Variables and Fortran Notation in CHEMEQ

Variable Type Specification
D - DOUBLE PRECISION (Floating Point)
I - INTEGER (Standard length)
L - LOGICAL (Standard length)
R - SINGLE PRECISION (Floating Point)

Variable Origin
A - Argument
C - Common
L - Local

Subscripts indicate the variable is an array.

Entries and Arguments	Purpose
CHEMEQ(DTCHEM,DFE,N,F,FMIN)	Advance the rate equations the specified interval.
CHEMCT(T)	Print out indicative counters.
CHEMSP(EPSMN,EPSMX,DTMN,TNOT,PASS, TASS,PRT)	Change default values of control parameters as required.
CHEMPR	Print partial set of internal variables.

Subroutines Referenced and Arguments	Purpose
DFE(F, C, D, T)	Derivative Functions Evaluator
CHEMER	This routine is called when the timestep becomes too small. Default version should be replaced by the users version to provide diagnostics if error is persistent.

Fortran Variable	Type/Origin	Mathematical Variable	Comments
ACOUNT	I/L	Index	Counter; records the number times asymptotics were employed.
ASY(I)	R/L	Logical	Records the location of the equations selected for asymptotic treatment.
C(I)	R/L	Q_i	Current formation rates.
CMD	R/L	$Q_i - L_i n_i$	Current total rate (intermediate variable for printing).
COR(I)	R/L	Multiple Usage	Temporary storage array.
D(I)	R/L	$L_i n_i$	Current loss rates.
DFS(I)	R/L	$Q_i - L_i n_i$	Total rate saved from the beginning of the step.
DT	R/L	δt	Current timestep.
DTC	R/L	δt	Timestep suitable for stability estimate (intermediate variable for printing)
DTCHEM	R/A	$0 \leq t \leq t_{chem}$	Range of the independent variable t.
DTMIN	R/L	δt_{min}	Minimum timestep allowed Default value; 1.OE-15)
DTMN	R/A	δt_{min}	Minimum timestep. Replaces DTMIN if DTMN > 0.
DTTARG	R/L	$\delta t_{chem} - t$	Intermediate variable used for end check.
DTTEST	R/L	δt_{min}	Intermediate variable used to check DT for minimum value.
EPS	R/L	$\text{Max}(\sigma_i)$	The maximum value of the relative error. Used to check for convergence.
EPSCL	R/L	$1/\epsilon_{min}$	Intermediate variable used to avoid repeated divisions.

Fortran Variable	Type/Origin	Mathematical Variable	Comments
EPSMAX	R/L	$\epsilon_{max}/\epsilon_{min}$	If EPS is larger than this value the step is restarted. (Default value; 10.)
EPSMIN	R/L	ϵ_{min}	The convergence criterion. DT for following step will be scaled proportional to EPSMIN/EPS. (Default value; .01)
EPSMN	R/A	ϵ_{min}	The convergence criterion. Replaces EPSMIN if EPSMN > 0.
EPSMX	R/A	$\epsilon_{max}/\epsilon_{min}$	Step restart criterion. Replaces EPSMAX if EPSMX > 0.
F(I)	R/A	n_i	The current values of the solution to the set of equations being integrated.
F0(I)	R/L	$n_i(0)$	Initial values at t_o.
FCOUNT	I/L	Index	Counter, records the number derivative function calls.
FMIN(I)	R/A	$n_{i(min)}$	Minimum values for each equation.
FS(I)	R/L	$n_i(0)$	The values of the solution saved from the beginning of the current step.
I	I/L	Index	Subscript counter.
J	I/L	Index	DO loop subscript.
JS	I/L	Index	Save location for specific J.
L	I/L	Index	DO loop subscript.
LO	I/L		Numerical value for the logical unit for the printed output.

Fortran Variable	Type/Origin	Mathematical Variable	Comments
N	I/A		The number of equations to be advanced.
NASY	I/L		Intermediate used in the asymptotic selection process.
ND	I/L	Constant	Array size specification set by the PARAMETER Statement. This is an ASC Fortran feature.
NDD	I/L		Storage location for ND. This is ASC specific Fortran.
PASY	R/A	$0 \leq PASY \leq 100$	Percentage of equations to be treated by asymptotics. Replace PASYI if PASY > 0.
PASYI	R/L	%	The percentage value of the set of equations that will always be selected for asymptotics. (Default value; 0.)
PRT	R/A		Print control parameter. If non-zero printer output from entry CHEMSP sill be suppressed.
RCOUNT	I/L	Index	Counter, records the number of times integration process had to be restarted due to non-convergence.
RTAU(I)	R/L	L_i	The reciprocals of the characteristic times.
RTAUS(I)	R/L	L_i	The reciprocals of the characteristic times saved from the beginning of the current step.
RTEPS	R/L	$\sqrt{Max(\sigma_i)/\epsilon_{min}}$	Used to estimate new timesteps.
RTMX	R/L	$L_{i\ max}$	Intermediate variable used to store the maximum value from RTAUS.
SCRA(I)	R/L	Multiple usage	Temporary storage array.
SCRB(I)	R/L	Multiple usage	Temporary storage array.

Fortran Variable	Type/Origin	Mathematical Variable	Comments
SCREPS	R/L	$\sqrt{\epsilon_{min}}$	Square Root of EPSMIN.
SCRTCH	R/L		Intermediate variable used to estimate the initial timestep.
TACNT	I/L	Index	Counter total. Records the total each time ACOUNT is set to zero.
TASY	R/A	τ	Asymptotic treatment selection criterion. Replaces TCRASY if TASY > 0.
TCRASY	R/L	τ	Asymptotic treatment selection criterion. This parameter is problem-dependent and the value should be proportional to the overall characteristic timestep of the system of equations being solved. (Default value; 0.1. This value is often suitable for high altitude atmospherical and many combustion problems.) Often it is useful to vary this parameter as the solution progresses.
TFCNT	I/L	Index	Counter total. Records the total each time FCOUNT is set to zero.
TFD	R/L		Round-off parameter. Should have a 5 in the last significant figure for single precision floating point words.
TN	D/L	t	Current value of the independent variable t.
TNOT	R/A	t_o	Initial value of the independent variable t. Replaces TSTART if TNOT > 0.
TRCNT	I/L		Counter total. Records the total each time RCOUNT is set to zero.

Fortran Variable	Type/Origin	Mathematical Variable	Comments
TS	D/L	t_o	The value of the independent variable t saved from the beginning of the current step.
TSTART	R/L	t_o	Initial value of the independent variable t. (Default value; 0)
TMK	R/A		Floating point number (typically the value of Time) printed to identify the call to CHEMCT.

APPENDIX C

This example involves the integration of seven rate equations which describe the time evolution of an atmospheric chemical relaxation test problem with cesium and cesium ions. This particular set of rate equations which was originally suggested by D. Edelson of Bell Laboratories is considered stiff and not well suited for numerical integration by classical methods.

The sample program listed in this section is designed to determine the efficiency of various stiff ordinary differential equation solvers on this test problem. In this example CHEMEQ is used. Efficiency is determined by comparing the results at the end of the integration interval with known values and the computer time required to obtain these results or various values of the convergence parameter EPS.

Table C1. A List of the Seven Species Together with Their Initial and Accepted Final Concentrations for the Test Problem

		Number Densities	
i	Species	Initial y_i, cm^{-3}	Final $\bar{y}_i$, cm^{-3}
1	e^-	1.0×10^2	4.9657897283×10^4
2	O_2^-	5.2×10^2	2.5913949444×10^4
3	Cs^+	6.2×10^2	7.5571846728×10^4
4	Cs	1.0×10^{12}	1.5319405460×10^3
5	CsO_2	0	1.000×10^{12}
6	N_2	1.4×10^{15}	1.400×10^{15}
7	O_2	3.6×10^{14}	3.590×10^{14}

Table C2. A List of the Seven Reactions and Reaction Rates Through which the Seven Species of the Test Problem Interact

No.	Reaction	Rate constant or frequency
1	$O_2^- + Cs^+ \rightarrow Cs + O_2$	5×10^{-8} cm^3 s^{-1}
2	$Cs^+ + e^- \rightarrow Cs + hv$	1×10^{-12} cm^3 s^{-1}
3	$Cs + hv \rightarrow Cs^+ + e^-$	3.24×10^{-3} s^{-1}
4	$O_2^- + hv \rightarrow O_2 + e^-$	4×10^{-1} s^{-1}
5a	$O_2 + Cs + M \rightarrow CsO_2 + M$	1×10^{-31} cm^6 s^{-1}
6	$O_2 + e^- + O_2 \rightarrow O_2^- + O_2$	1.24×10^{-30} cm^6 s^{-1}
7	$O_2 + e^- + N_2 \rightarrow O_2^- + N_2$	1×10^{-31} cm^6 s^{-1}

[a] $M = [Cs] + [CsO_2] + [N_2] + [O_2]$.

In the following listing TACSR is the main program which provides the logic and overall control. Initialization and output of results takes place here. CSDFE is the derivative function evaluator for the text problem. Results for nine values of the convergence parameter EPS are printed at the end of the section.

```
      PROGRAM TACSR
C
C     THIS IS THE EXECUTIVE PROGRAM THAT PROVIDES THE LOGIC NESSICARY
C     TO ADVANCE A REACTIVE SEVEN SPECIES TEST PROBLEM FOR AN EVALUATION
C     OF THE INTEGRATION METHOD FOR VARIOUS VALUES OF THE CONVERGENCE
C     PARAMETER. IN THIS EXAMPLE "CHEMEQ" WILL BE EMPLOYED.
C
C             PROGRAM SPECIFICATIONS.
C             ----------------------
      REAL*8         DSEC
C
      REAL           Y(10), YF(10), YMIN(10), YI(10), EPSIL(10), EPS(15)
C
      INTEGER        SPSYM(10)
C
      EXTERNAL       CSCFE
C
      DATA           YMIN/10*1.0E-04/, MXCASE/9/
      DATA           SPSYM/'02-', 'CS+', 'CS', 'CS02', '02', 'N2', 'NE'/
      DATA           EPS/.1, .05, .01, .005, .001, .0005, .0001, .00005,
     .               .00001/
C
 1000     FORMAT('1CASE NO. ', I5, '    PARAMETERS;', /,
     .    ' CONVERGENCE PARAMETER EPS = ', 1PE10.3, /,
     .    ' INNER LOOP LENGTH;', I5)
 1001     FORMAT(/, '    SPECIE     Y - INITAL       Y - FINAL ',
     .    '  Y - SOLUTION   REL ERR')
 1002     FORMAT(5X, A4, 1P3E15.6, E10.3)
 1003     FORMAT(/, ' T - INITIAL = (', 1PE10.3, ') T - FINAL = (',
     .    E10.3, ')')
 1004     FORMAT(/' INTEGRATION STATISTICS;')
 1005     FORMAT(' CPU TIME USED FOR INTEGRATION;', 1PE10.3,
     .    ' SEC.,  CPU TIME NORMALIZED;', E10.3)
 1006     FORMAT(' SUM OF THE RELATIVE ERRORS SQUARED; ', 1PE10.3)
 1007     FORMAT(/)
C
C
C     INITIALIZE CONTROL PARAMETERS.
C
C     "TSCALE" IS A NORMILIZATION FACTOR USED TO COMPARE EFFICIENCY OF
C     INTEGRATION CODES FROM DIFFERENT COMPUTER INSTALATIONS. "TSCALE"
C     MAY BE DETERMINED BY TIMING A TEST CODE ON ALL INSTALATIONS
C     INVOLVED.
          TSCALE = 1.0
C
C     SET INNER LOOP LENGTH. SEE COMMENTS BELOW FOR DEFINITION.
          INLP = 1
```

```
C
C     SET THE TOTAL NUMBER OF SPECIES "NS" AND THE NUMBER TO BE
C     INTEGRATED "NA".
          NS = 7
          NA = 5
C
C     "TI"  -  INITIAL TIME,   "TF"  -  FINAL TIME.
          TI = 0.0
          TF = 1000.0
          DELTAT = (TF - TI)/INLP
C
C     STORE INITIAL(TI = 0.0) AND FINAL(TF = 1000.0) VALUES.
C
C     O2-
          YI(1) = 5.200E+02
          YF(1) = 2.59139492061D+04
C
C     CS+
          YI(2) = 6.200E+02
          YF(2) = 7.55718460300D+04
C
C     CS
          YI(3) = 1.000E+12
          YF(3) = 1.53194051722D+03
C
C     CSO2
          YI(4) = 1.000E-30
          YI(4) = 1.000E+04
          YF(4) = 9.99999923516D+11
C
C     O2
          YI(5) = 3.600E+14
          YF(5) = 3.59000000051D+14
C
C     N2
          YI(6) = 1.400E+15
          YF(6) = 1.40000000000D+15
C
C     NE
          YI(7) = 1.000E+02
          YF(7) = 4.96578968239D+04
C
C     LOOP OVER THE TEST CASES.
          DO 30 ICASE = 1,MXCASE
          PRINT 1000, ICASE, EPS(ICASE), INLP
          CALL CHEMSP(EPS(ICASE), 0., 0., TI, 0., 10.0, 0.)
          CPLT = 0.0
```

```
C
C     RESET "Y" TO INITIAL VALUES "YI".
          DO 35 I = 1,NS
   35     Y(I) = YI(I)
C
C     SET TIMER.
          CALL SECOND(1, DSEC)
C
C     INNER LOOP TO DETERMINE OVERHEAD OR RELATIVE STARTING EFFECIENCY
C     OF ITEGRATION SCHEME BEING TESTED.
          DO 5 ISTEP = 1,INLP
C
C     CALL INTEGRATOR.
          CALL CHEMEQ(DELTAT, CSDFE, NA, Y, YMIN)
    5     CONTINUE
C
C     CALCULATE CPU TIME USED IN THE INTEGRATION PROCESS.
          CALL SECOND(0, DSEC)
          CPUT = CPUT + DSEC
          TNORM = CPUT/TSCALE
C
C     RESET ELECTRON DENSITY.
          Y(7) = Y(2) - Y(1)
C
C     CALCULATE RELATIVE ERROR.
          DO 10 I = 1,NS
   10     EPSIL(I) = ABS(Y(I) - YF(I))/AMIN1(Y(I) , YF(I))
          SUM = 0.0
          DO 25 I = 1,NS
   25     SUM = SUM + EPSIL(I)**2
C
C     PRINT RESULTS.
          PRINT 1003, TI, TF
          PRINT 1001
          DO 15 I = 1,NS
   15     PRINT 1002, SPSYM(I), YI(I), YF(I), Y(I), EPSIL(I)
          PRINT 1004
          PRINT 1006, SUM
          PRINT 1005, CPUT, TNORM
          PRINT 1007
          CALL CHEMCT(TF)
   30     CONTINUE
          STOP 69
      END
      SUBROUTINE CSDFE(Y, C, D, T)
C
CD * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CD
CD    CSDFE(Y, C, D, T)
CD
CD    DESCRIPTION:
CD    DERIVATIVE FUNCTION EVALUATOR(DFE) FOR AN ATMOSPHERIC CHEMICAL
CD    RELAXATION TEST PROBLEM INVOLVING CESIUM AND CESIUM IONS. FORMAT-
CD    ION AND LOSS RATES ARE CALCULATED FOR THIS SET OF "STIFF ORDINARY
CD    DIFFERENTIAL EQUATIONS" THAT WAS SUGGESTED BY BY D. EDELSON OF
CD    BELL LABORATORIES.
CD
CD    ARGUMENT LIST DEFINITIONS:
CD    Y(I)           R*4  CURRENT VALUES OF THE FUNCTIONS PLUS THE     I/O
CD                        EXTRA DATA AT THE END OF THE ARRAY THAT MAY BE
CD                        PASSED BACK AND FORTH BETWEEN "CSDFE" AND THE
CD                        MAIN PROGRAM. LOCATIONS IN Y(I) WHICH REPRESENT
CD                        THE FUNCTIONS BEING ADVANCED SHOULD NOT BE
CD                        TAMPERED WITH HERE.
CD    C(I)           R*4  TOTAL FORMATION RATES.                         I
CD    D(I)           R*4  TOTAL LOSS RATES.                              I
CD    T              R*4  THE VALUE OF THE INDEPENDENT VARIABLE.         I
CD
CD * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C               LOCAL SPECIFICATIONS.
C               ---------------------
      REAL          NE, N2
      REAL          Y(1), C(1), D(1)
C
C     UTILIZE LOCAL STORAGE FOR VARIBLES.
          O2M  = Y(1)
          CSP  = Y(2)
          CS   = Y(3)
          CSO2 = Y(4)
          O2   = Y(5)
          N2   = Y(6)
C
C     CALCULATE ELECTRON DENSITY FOR LOCAL USE AND TRANSMISSION BACK TO
C     THE MAIN PROGRAM VIA Y(7). HOWEVER IN THIS CASE THIS VALUE SHOULD
C     NOT BE TRUSTED SINCE "CHEMGE" WILL NOT CALL THE "DFE" WITH THE
C     LATEST FUNCTION VALUES AFTER THE FINAL STEP HAS CONVERGED. Y(7)
C     WILL BE ONE ITERATION BEHIND IN THIS CASE. Y(7) AND Y(6) ARE
C     EXAMPLES THO, OF HOW DATA MAY BE TRANSFERED BETWEEN THE "DFE" AND
C     THE MAIN PROGRAM.
          NE = AMAX1(CSP - O2M, 0.0)
```

```
            Y(7) = NE
C
C     CALCLLATE REACTIØN RATES.
            CR1 = 5.00E-08*Ø2M*CSP
            CR2 = 1.00E-12*CSP*NE
            CR3 = 3.24E-03*CS
            CR4 = 4.00E-01*Ø2M
            CR5 = 1.00E-31*Ø2*CS*(CS + CSØ2 + N2 + Ø2)
            CR6 = 1.24E-30*Ø2*Ø2*NE
            CR7 = 1.00E-31*Ø2*N2*NE
C
C     CALCLLATE TØTAL FØRMATIØN RATES (C(I)) AND TØTAL LØSS RATES (D(I))
C     FØR EACH SPECIES.
C
C     Ø2M
            C(1) = CR6 + CR7
            D(1) = CR1 + CR4
C
C     CS+
            C(2) = CR3
            D(2) = CR1 + CR2
C
C     CS
            C(3) = CR1 + CR2
            D(3) = CR3 + CR5
C
C     CSØ2
            C(4) = CR5
C
C     Ø2
            C(5) = CR1 + CR4
            D(5) = CR5 + CR6 + CR7
C
            RETURN
      END
```

```
CASE NO.     1    PARAMETERS;
CONVERGENCE PARAMETER EPS =  1.000E-01
INNER LOOP LENGTH;    1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =      0.100     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.540664E 04 1.997E-02
    CS+     6.200000E 02   7.557181E 04   7.445481E 04 1.500E-02
    CS      1.000000E 12   1.531940E 03   1.616265E 03 5.504E-02
    CSO2    1.000000E 04   9.999999E 11   1.003010E 12 3.011E-03
    O2      3.600001E 14   3.589999E 14   3.589755E 14 6.805E-05
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965709E 04   4.904818E 04 1.243E-02

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  3.817E-03
CPU TIME LSEC FOR INTEGRATION; 3.112E-02 SEC.,  CPU TIME NORMALIZED; 3.112E-02


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =     231    165      1 TOTALS;     231    165      1




CASE NO.     2    PARAMETERS;
CONVERGENCE PARAMETER EPS =  5.000E-02
INNER LOOP LENGTH;    1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  5.000E-02     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.553557E 04 1.482E-02
    CS+     6.200000E 02   7.557181E 04   7.466781E 04 1.211E-02
    CS      1.000000E 12   1.531940E 03   1.556146E 03 1.580E-02
    CSO2    1.000000E 04   9.999999E 11   1.001245E 12 1.246E-03
    O2      3.600001E 14   3.589999E 14   3.589675E 14 9.048E-05
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965709E 04   4.913225E 04 1.070E-02

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  7.318E-04
CPU TIME LSEC FOR INTEGRATION; 5.699E-02 SEC.,  CPU TIME NORMALIZED; 5.699E-02


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =     422    305      6 TOTALS;     653    470      7
```

```
CASE NO.     3    PARAMETERS;
CONVERGENCE PARAMETER EPS =  1.000E-02
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  1.000E-02     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION    REL ERR
    O2-     5.200000E 02   2.591395E 04   2.579939E 04 4.440E-03
    CS+     6.200000E 02   7.557181E 04   7.538381E 04 2.494E-03
    CS      1.000000E 12   1.531940E 03   1.533065E 03 7.342E-04
    CSO2    1.000000E 04   9.999999E 11   1.000130E 12 1.306E-04
    O2      3.600001E 14   3.589999E 14   3.582676E 14 3.688E-04
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   4.958443E 04 1.482E-03

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  2.882E-05
CPU TIME LSEC FOR INTEGRATION; 1.779E-01 SEC.,  CPU TIME NORMALIZED; 1.779E-01


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =    1324    997      4 TOTALS:    1977   1467     11
```

```
CASE NO.     4    PARAMETERS;
CONVERGENCE PARAMETER EPS =  5.000E-03
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  5.000E-03     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION    REL ERR
    O2-     5.200000E 02   2.591395E 04   2.586108E 04 2.044E-03
    CS+     6.200000E 02   7.557181E 04   7.551887E 04 7.010E-04
    CS      1.000000E 12   1.531940E 03   1.533121E 03 7.704E-04
    CSO2    1.000000E 04   9.999999E 11   1.000013E 12 1.363E-05
    O2      3.600001E 14   3.589999E 14   3.587857E 14 5.970E-04
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   4.965780E 04 1.967E-06

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  5.621E-06
CPU TIME LSEC FOR INTEGRATION; 2.877E-01 SEC.,  CPU TIME NORMALIZED; 2.877E-01


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =    2143   1644      3 TOTALS:    4120   3111     14
```

```
CASE NO.     5    PARAMETERS;
CONVERGENCE PARAMETER EPS =  1.000E-03
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  1.000E-03     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.590206E 04 4.589E-04
    CS+     6.200000E 02   7.557181E 04   7.570156E 04 1.717E-03
    CS      1.000000E 12   1.531940E 03   1.537838E 03 3.850E-03
    CSO2    1.000000E 04   9.999999E 11   9.999150E 11 8.488E-05
    O2      3.600001E 14   3.589999E 14   3.583852E 14 1.715E-03
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   4.979950E 04 2.852E-03

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  2.906E-05
CPU TIME USEC FOR INTEGRATION; 8.119E-01 SEC.,  CPU TIME NORMALIZED; 8.119E-01


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =    6052   4815      2 TOTALS:   10172   7926     16
```

```
CASE NO.     6    PARAMETERS;
CONVERGENCE PARAMETER EPS =  5.000E-04
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  5.000E-04     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.590909E 04 1.874E-04
    CS+     6.200000E 02   7.557181E 04   7.580419E 04 3.075E-03
    CS      1.000000E 12   1.531940E 03   1.541755E 03 6.406E-03
    CSO2    1.000000E 04   9.999999E 11   9.998384E 11 1.614E-04
    O2      3.600001E 14   3.589999E 14   3.580228E 14 2.729E-03
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   4.989510E 04 4.777E-03

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  8.082E-05
CPU TIME USEC FOR INTEGRATION; 1.293E 00 SEC.,  CPU TIME NORMALIZED; 1.293E 00


CHEMEQ INDICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =    9636   7837      2 TOTALS:   19808  15763     18
```

```
CASE NO.      7    PARAMETERS;
CONVERGENCE PARAMETER EPS =  1.000E-04
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  1.000E-04     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 00) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.593362E 04 7.594E-04
    CS+     6.200000E 02   7.557181E 04   7.641800E 04 1.120E-02
    CS      1.000000E 12   1.531940E 03   1.566864E 03 2.280E-02
    CSO2    1.000000E 04   9.999999E 11   9.994252E 11 5.750E-04
    O2      3.600001E 14   3.589999E 14   3.558273E 14 8.916E-03
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   5.048437E 04 1.664E-02

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  1.002E-03
CPU TIME LSEC FOR INTEGRATION; 3.895E 00 SEC.,  CPU TIME NORMALIZED; 3.895E 00


CHEMEG INCICES; TMK =  1.000E 03 FCOUNT, ACCUNT, RCOUNT =   29039  24861      3 TOTALS:   48847  40624     21
```

```
CASE NO.      8    PARAMETERS;
CONVERGENCE PARAMETER EPS =  5.000E-05
INNER LOOP LENGTH;     1
INITALIZE "CHEMEQ" VIA "CHEMSP"
EPSMN, EPSMX, CTMN, TNOT, PASY, TASY =  5.000E-05     0.000     0.000     0.000     0.000     10.00

T - INITIAL = ( 0.000E 0C) T - FINAL = ( 1.000E 03)

   SPECIE     Y - INITAL       Y - FINAL    Y - SOLUTION   REL ERR
    O2-     5.200000E 02   2.591395E 04   2.596118E 04 1.823E-03
    CS+     6.200000E 02   7.557181E 04   7.703181E 04 1.932E-02
    CS      1.000000E 12   1.531940E 03   1.592337E 03 3.942E-02
    CSO2    1.000000E 04   9.999999E 11   9.990567E 11 9.440E-04
    O2      3.600001E 14   3.589999E 14   3.537241E 14 1.492E-02
    N2      1.400000E 15   1.400000E 15   1.400000E 15 0.000E 00
    NE      1.000000E 02   4.965789E 04   5.107063E 04 2.845E-02

INTEGRATION STATISTICS;
SUM OF THE RELATIVE ERRORS SQUARED;  2.964E-03
CPU TIME LSEC FOR INTEGRATION; 6.128E 00 SEC.,  CPU TIME NORMALIZED; 6.128E 00


CHEMEQ INCICES; TMK =  1.000E 03 FCOUNT, ACOUNT, RCOUNT =   45704  39739      2 TOTALS:   94551  80363     23
```