Computation of Chemical Equilibrium in a Constant Temperature and Pressure System

Jim Herzog

June 1986

UWFDM-695

Computation of Chemical Equilibrium in a Constant Temperature and Pressure System

Jim Herzog

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

June 1986

UWFDM-695

Computation of Chemical Equilibrium in a Constant Temperature and Pressure System

Jim Herzog

University of Wisconsin
Nuclear Engineering Department
1500 Johnson Drive
Madison, WI 53706

June 1986

UWFDM-695

Support for this work has been provided by the Department of Energy under DOE Contract DE-AC07-76ID01570.

i) Introduction

For multicomponent systems, at constant temperature and pressure, the calculation of chemical equilibrium of a system often requires the aid of a computer. This is especially true if the system contains several components and phases. A number of computational models have been developed of varying complexity to calculate the chemical equilibrium of multicomponent, multiphase systems[1]. The model chosen, in each instance, should depend upon the application. A system that contains many possible chemical species and phases requires a complex model. Unfortunately, the complex models tend to be application specific, each separate application requiring practically a total reformation of the model.

I have written a computer program that calculates the chemical equilibrium of modest size, multicomponent systems. For each application, the user is required to adjust a short subroutine and create a short input data file to employ the program. Thus the program is very general and easy to use. To achieve this, the program is limited to constant temperature and pressure systems. Also, the program is very inefficient in the calculation of the concentration of trace chemical species.

In this report I will describe, in detail, the mathematical derivation of the model, the general applicability of the model, and two specific applications of the model. In the first application of the model, it is used to predict the equilibrium composition of a propane/air system at a specified temperature and pressure. The results are then compared to published results of the same problem, to help determine the accuracy of the method. In the second application of the model, it is used to predict the equilibrium composition of a $Li_{17}Pb_{83}/H_2O$ system. This application is an attempt to model the recent HEDL large scale $Li_{17}Pb_{83}/H_2O$ experiments.

ii) Preliminaries

The model chosen is due to Van Zeggeren and Storey.[2] This model is also the basis for the chemical equilibrium subroutine in the core-melt program CORCON (the subroutine is named MLTREA). Unlike MLTREA, the program I have developed is intended to be easy to understand and employable to a wide range of problems. To achieve this, the program sacrifices efficiency. It has not been written to approach the solution quickly, especially if trace species are present, but to approach the solution at a relatively slow, methodical pace.

The Van Zeggeren and Storey model consists of a computational algorithm that finds the minimum Gibbs free energy of a system. This approach is taken because the minimum of a system's Gibbs free energy is equivalent to the chemical equilibrium of the system.[3] For a multicomponent system the Gibbs free energy is

$$G(p,T,n_i) = H(p,T,n_i) - S(p,T,n_i)\,T \qquad (1)$$

$$\Rightarrow \quad dG = \left(\frac{\delta G}{\delta T}\right)_{p,n_i} dT + \left(\frac{\delta G}{\delta p}\right)_{T,n_i} dp + \sum_i^N \left(\frac{\delta G}{\delta n_i}\right)_{p,T} dn_i \qquad (2)$$

but, since[4]

$$\left(\frac{\delta G}{\delta T}\right)_{p,n_i} = -S \quad \text{and} \quad \left(\frac{\delta G}{\delta p}\right)_{T,n_i} = V \qquad (3)$$

and defining

$$\left(\frac{\delta G}{\delta n_i}\right)_{p,T} = \mu_i \qquad (4)$$

where, μ_i = the chemical potential of species i,

$$\Rightarrow \quad dG = -S\,dT + V\,dp + \sum_i^N \mu_i\,dn_i \qquad (5)$$

iii) The Van Zeggeren and Storey Model

The Van Zeggeren and Storey model is based on a constant temperature and pressure system. Therefore the model starts with the differential form of the Gibbs free energy expessed as

$$dG = \sum_i^N \mu_i\,dn_i \qquad (6)$$

This equation is recast in the form

$$\delta G = \sum_i^N \mu_i\,\delta n_i \qquad (7)$$

This is done to mathematically separate the differential form of the Gibbs free energy from the form of the Gibbs free energy used in the program. To compute the minimum Gibbs free energy, the program approachs the solution by calculating successive sets of δn_i, with each set generating a more negative Gibbs free energy than the previous set. The set of n_i which forms the solution must also satisfy M conditions of the form

$$\sum_{i}^{N} a_{ie} n_i = B_e \tag{8}$$

These conditions arise from the fact that in the chemical system being modelled, mass is neither created nor destroyed, but only changes its chemical form. From the initial compostion of the system (the N initial chemical species masses), one obtains the masses of the M elements in the system. Thus the solution set of the masses of the chemical species must be such that the masses of the constituent elements are conserved.

Besides the mass balance constraints, the solution set n_i must satisfy one other obvious condition. Namely, the solution set must be such that

$$n_i \geq 0 \tag{9}$$

To insure that this condition is met, Van Zeggeren and Storey transform the equations by introducing a new variable ϕ, defined by

$$n_i = \exp(\phi_i) \tag{10}$$

This transformation insures that $n_i \geq 0$. The system equations

(7) and the mass balance constraints (8) are transformed to

$$\delta G = \sum_i^N \mu_i \, n_i \, \delta\phi_i \tag{11}$$

and

$$\sum_i^N a_{ie} \, n_i \, \delta\phi_i = \Delta B_e \tag{12}$$

where

$$\Delta B_e = B_e - b_e \tag{13}$$

Here the B_e are the constrained elemental abundances, and the b_e are the elemental abundances which are obtained from the current values of the masses of the chemical species.

Mathematically, the program causes the computer to follow the line of steepest slope down the Gibbs free energy surface $G(n_i)$, usually in fairly small steps, until the minimum Gibbs free energy is found. At any given point in the search, the direction of the steepest slope is that which makes δG in equation (11) an extremum, subject to conditions (12) and an additional condition

$$\sum_i^N (\delta\phi)^2 = \sigma^2 \tag{14}$$

The size of the step taken during the calculation is given by σ.

Next the method of Lagrangian multipliers is applied to the transformed system equations (11) and the transformed conditions (12,14). The Lagrangian for the system is

$$L = \sum_i^N \mu_i \, n_i \, \delta\phi_i - \sum_e^M \chi_e \left(\sum_i^N a_{ie} \, n_i \, \delta\phi_i - \Delta B_e \right) - \chi_\phi \left(\sum_i^N (\delta\phi_i)^2 - \sigma^2 \right) \tag{15}$$

Here, χ_e and χ_ϕ are the Lagrangian multipliers. The extremum of G is then given by the condition

$$\frac{\delta L}{\delta n_i} = 0 \tag{16}$$

$$\Rightarrow \quad \mu_i n_i - \chi_\phi \delta\phi_i - \sum_e^M \chi_e a_{ie} n_i = 0 \tag{17}$$

$$\Rightarrow \quad \delta\phi_i = \frac{n_i}{\chi_\phi} (\mu_i - \sum_e^M \chi_e a_{ie}) \tag{18}$$

Equations (18) can be used to find $\delta\phi_i$ in terms of both known quantities and the Lagrangian multipliers. It is thus necessary to express χ_e and χ_ϕ in terms of known quantities. This is accomplished by multiplying equations (18) by $a_{ie} n_i$ and summing over i. Using the mass balance equations (12) this leads to M equations

$$\sum_f^M (\sum_i^N a_{ie} a_{if} n_i^2) \chi_f = \sum_i^N a_{ie} \mu_i n_i^2 - \chi_\phi \Delta B_e \tag{19}$$

The Lagrangian multiplier χ_e can be written as

$$\chi_e = \varkappa_e + \chi_\phi \omega_e \tag{20}$$

The coefficients $\varkappa_e$ and ω_e are given by the equations

$$\sum_f^M (\sum_i^N a_{ie} a_{if} n_i^2) \varkappa_f = \sum_i a_{ie} \mu_i n_i^2 \tag{21}$$

$$\sum_f^M (\sum_i^N a_{ie} a_{if} n_i^2) \omega_f = -\Delta B_e \tag{22}$$

Equation (18) can now be rewritten as

$$\delta\phi_i = \frac{1}{\chi_\phi} (D_i - \chi_\phi E_i) \tag{23}$$

where
$$D_i = \mu_i n_i - \sum_e^M \omega_e a_{ie} n_i \tag{24}$$

$$E_i = \sum_e^M \omega_e a_{ie} n_i \tag{25}$$

Finally, by substituting for $\delta\phi_i$ from equation (23) into the subsidiary condition (14) and solving for χ_ϕ leads to

$$\chi_\phi = - \left[(\sum_i^N D_i^2) / (\sigma^2 - \sum_i^N E_i^2) \right]^{½} \tag{26}$$

The negative sign in this equation must be present for the procedure to minimize the Gibbs free energy.

iv) Computer Algorithm

The basic cycle for calculating improved estimates of the equilibrium composition consists of these steps:[5]

1). First the program parameters are initialized by a call to the input subroutine CHEMIN.

2). Next the species chemical potentials and the system Gibbs free energy are calculated. The species chemical potentials are given by

$$\mu_i = \mu_i^o(T) + R\,T\,\ln\left[\epsilon_i\, n_i \,/\, \left(\sum_j^N \alpha_j\, n_j\right)\right] \qquad (27)$$

where $\epsilon_i = \frac{P}{P_o}$, if n_i is a gas,

or $\epsilon_i = 1$, if n_i is a solid or liquid, (28)

and $\alpha_j = 1$, if species i and j are of the same phase,

or $\alpha_j = 0$, if species i and j are of different phases. (29)

For the applications I am considering, I use two definitions of the chemical potential at 1 atm. In the first application, the propane/air system, I use the standard definition:

$$\mu_i^o = (\Delta G_f^o(T))_i \qquad (30)$$

Since, for the $Li_{17}Pb_{83}$ compound, the Gibbs free energy of formation is not tabulated, I must use an alternate definition for the chemical potential of $Li_{17}Pb_{83}$ at 1 atm. Thus for the second application, the $Li_{17}Pb_{83}/H_2O$ system, I use the definition:

$$\mu_i^o = (G^o(T) - H^o(298))_i + (\Delta H_f^o(298))_i \qquad (31)$$

Both of these expressions are equally valid, because the actual values of μ_i^o are unimportant; only their differences are thermodynamically defined, and the selection of μ_i^o values is based on convention only.[6] Values of μ_i^o are obtained from data from the JANAF tables.[7]

3). Calculate the current values of ΔB_e, using equations (12) and (13).

4). Form the sums over i appearing in equations (21) and (22).

5). Solve equations (21) and (22) to determine the coefficients π_e and ω_e. These equations are solved using the matrix inversion subroutine MATINV.

6). Find D_i and E_i from equations (24) and (25).

7). Find χ_ϕ from equation (26).

8). Find the set of $\delta\phi_i$ values from equation (23).

9). Calculate new estimates of the equilibrium composition from

$$n_i = m_i \exp(\delta\phi_i) \qquad (32)$$

10). Next, two convergence criteria are checked to determine whether or not the set n_i corresponds to the equilibrium composition. The first condition is; has the step size decreased to the input cutoff value? Namely, is

$$\sigma \le \tau \qquad (33)$$

The second condition is, have the the current values of the element masses converged to the constrained element masses[8]? Namely, is

$$| B_e - b_e | \le \in B_e \qquad (34)$$

11). If either of the conditions (33) or (34) are false, then the step size is adjusted, depending upon the values of $\delta\phi_i$ from the current step and the previous step. The values of the program variables are then set to the new variable values, and the program returns to step 2).

12). If the conditions (33) and (34) are true, then the equilibrium composition of the system has been found. The program calls the output subroutine CHOUT, and then terminates.

v) Program Use

To employ the program, the user needs to create a short data file and subroutine that contain the problem specific data.

The subroutine that the user needs to create is called by the name CHEMPO. It must contain formulas for the chemical potentials, at 1 atm, of each species. Also, the subroutine must specify the phase of each species. The subroutine must be written in FORTRAN. Once compiled, the subroutine can then be linked with the object file that contains the compiled form of the rest of the program (CHEMEQ.OBJ). The interested reader should refer to the program listing appendix included at the end of this report to ascertain the required format of the CHEMPO subroutine. (This appendix contains the program listing of the two versions of CHEMPO that I have used in the applications discussed in this paper.)

The data file that the user must create contains problem specific data, in free format form, in the following configuration. The first line of the file must contain the number of species being considered (N), and then the number of constituent elements (M). The next N lines must contain the names of the species. Each record can contain up to 10 characters. The next M lines must contain the names of the elemnts. Again, each record can contain up to 10 characters. These N+M lines are not free format; the characters must be placed in the first 10 columns of each line. The order of the species and element names given in these lines will be the order maintained by the program. Namely, suppose the first record in this list is H_2O, then n_1, $\delta\phi_1$, and so forth, will refer to the species H_2O. Lastly, the final M lines must contain the array of stoichiometric coefficients (a_{ie}). Each of these lines must contain N numbers. As an example: Suppose that the first element in the element portion of the data file is H. Then suppose that there are 10 species being considered. If, say, only species 1, 3, and 9 contained hydrogen, and that these species were, $n_1 = H_2O$, $n_3 = HCl$, and $n_9 = H_2$; then the first line of the stoichiometric array portion of the data file will be

```
 2., 0., 1., 0., 0., 0., 0., 0., 2., 0.
```

The program will request that the user provide the rest of the needed input data. The program will first prompt the user to enter a program run title. It will then request that the user provide it with the system temperature and pressure for that run. The user will then be asked to give the initial mass (in mole) of the problem species. The program then reports the default values of the program conversion parameters, and asks whether or not the user wants to change these parameters. The parameters that the user has the power to adjust are: the initial step size, the minimum step size convergence parameter, the mass balance constraint convergence parameter, and the trace species cutoff parameter.

vi) The Propane/Air System Application

The model is first applied to a propane/air system. Specifically, the model is used to predict the results of the combustion of propane with air, to give decomposition products, at 2200 K and 40 atm. The only system parameter considered is the molar ratio R of air ($O_2 + 4N_2$) to propane (C_3H_8). The chemical equilibrium of this system is chosen because the Van Zeggeren and Storey used this problem to calibrate their models. The results of this analysis are tabulated on the next page. The data in this table are in mole fractions: $x_i = n_i / n$, where $n = \Sigma n_g$ (i.e., the sum of the molar masses of the gases in the equilibrium system). The first number in presented is taken from the literature.[9]

Comparison of the Results form the Model with the Literature

For the Reaction: C_3H_8 + R (O_2 + $4N_2$) => Products

	R = 1		R = 2		R = 5	
Products	Lit.	Model	Lit.	Model	Lit.	Model
CO_2	0.00002	1.75E-5	0.00989	0.00988	0.10795	0.10797
N_2	0.39996	0.39996	0.53322	0.53322	0.73874	0.73874
H_2O	0.00018	1.82E-4	0.05675	0.05675	0.14674	0.14674
CO	0.19976	0.19976	0.19006	0.19007	0.00294	2.93E-3
H_2	0.39953	0.39953	0.20966	0.20966	0.00077	7.65E-4
H	0.00056	5.61E-4	0.00041	4.06E-4	0.00002	2.45E-5
OH	0.00000	6.20E-8	0.00002	1.53E-5	0.00068	6.74E-4
O	0.00000	0.00000	0.00000	3.83E-7	0.00001	1.27E-5
NO	0.00000	2.15E-8	0.00000	1.64E-6	0.00097	9.75E-4
O_2	0.00000	0.00000	0.00000	0.00000	0.00119	0.00119
C	0.10020	0.10020	0.00000	0.00000	0.00000	0.00000

As the table shows, the program predicts the same equilibrium compostion as is given in the Van Zeggeren and Storey text. Thus one can conclude that the program is as effective as the program written by Van Zeggeren and Storey. As can be seen, the program produces data that is as accurate as the published data. But this is only because I experimented with the program parameters until the program would produce the desired accurate results. This means that the user has control over the accuracy of the program output. The parameter that provides the user with this ability is the trace species cutoff parameter.

The program is endowed with the ability to set the mass of certain trace species to zero, and thus increase the speed at which the program converges to the equilibrium composition. The program accomplishes this feat in the following manner. Early in the execution of the program, the maximum possible species mass, which is determined from the constained element masses, is calculated. If the ratio of the current species mass to the maximum species mass ever falls below the species cutoff parameter, then that species mass is set equal to zero. Thus if the user sets the species cutoff parameter to a relatively high value, the program will execute quickly. But, the user will then run the risk of having the program predict a final composition that is not near the true equilibrium composition. On the other hand, if the user sets the trace species cutoff parameter to a relatively small value, the program execution time may increase to a prohibitively high amount.

The best way to determine an adequate setting for the trace species cutoff parameter is to experiment. At first application of the program, the user should set the trace species cutoff parameter to a very low value. Since the program writes the species composition to the screen after every 10 iterations, one can monitor the action of the program. Typically, some species will converge quickly to relatively high masses. Other species will quickly reach ralatively small mass values, but will not converge to any value. They will instead, continue to decrease in value, but they may take a prohibitive amount of time to converge to some value. When this happens, the user can halt execution of the program and reexecute the program with a larger value of the species cutoff parameter. Using this method, the user will eventually find a solution that is adequate to his needs. As an example, the data for the R = 1 column in the above table took 2520 program iterations, with the trace species cutoff parameter set equal to $5 * 10^{-7}$, to determine. This took approximately 10 minutes of execution time on an IBM XT.

vii) The $Li_{17}Pb_{83}/H_2O$ System Application

Knowing that the program performs properly, one can now use it to determine the degree of equilibrium interaction exhibited in the data from an experiment. The data analyzed in this manner is taken from a series of large scale $Li_{17}Pb_{83}/H_2O$ experiments[10,11], carried out at the Hanford Engineering Development

Laboratory. The experiment consisted of the injection of high pressure steam into a liquid $Li_{17}Pb_{83}$ pool. The average temperature was approximately 1140K at atmospheric pressure. Through the course of the experiment, 90.18 mole of H_2O were injected into 1155 mole of alloy. With this data, one can run the program. The predicted equilibrium composition of the system is presented on the next page.

```
Li-17 Pb-83 / H2O  Reaction:  HEDL Large-scale test parameters
    For T =   1143.     K  and  P =  1.0135E+05 Pa
    The minimum Gibbs free energy = -5.6449E+07 J
    The number of iterations =   2403
                -----------------------------------
                ¦  Element     ¦ Element mass [MOLE]¦
                ¦--------------¦--------------------¦
                ¦    Li        ¦          196.350   ¦
                ¦    Pb        ¦          958.650   ¦
                ¦    H         ¦          180.360   ¦
                ¦    O         ¦          90.1800   ¦
                -----------------------------------
  --------------------------------------------------------------
  ¦  Species    ¦ Initial mass [MOLE] ¦ Equilibrium mass [MOLE]¦
  ¦-------------¦---------------------¦------------------------¦
  ¦   Li        ¦         68.4990     ¦            3.528763E-02 ¦
  ¦   Pb        ¦         862.785     ¦             622.127     ¦
  ¦   Li17Pb83  ¦         115.500     ¦             405.450     ¦
  ¦   H2        ¦         54.1080     ¦             63.6941     ¦
  ¦   H2O       ¦         9.01800     ¦            1.675298E-03 ¦
  ¦   Li2O      ¦         27.0540     ¦             37.2100     ¦
  ¦   LIOH      ¦         54.1080     ¦             52.9684     ¦
  --------------------------------------------------------------
*******************************************************************
   Li-17 Pb-83 / H2O  Reaction:  HEDL Large-scale test parameters

    For T =   1143.     K  and  P =  1.0135E+05 Pa
    The minimum Gibbs free energy = -5.6449E+07 J
    The number of iterations =   2413
                -----------------------------------
                ¦  Element     ¦ Element mass [MOLE]¦
                ¦--------------¦--------------------¦
                ¦    Li        ¦          196.348   ¦
                ¦    Pb        ¦          958.633   ¦
                ¦    H         ¦          180.358   ¦
                ¦    O         ¦          90.1790   ¦
                -----------------------------------
  --------------------------------------------------------------
  ¦  Species    ¦ Initial mass [MOLE] ¦ Equilibrium mass [MOLE]¦
  ¦-------------¦---------------------¦------------------------¦
  ¦   Li        ¦         26.5300     ¦            3.530048E-02 ¦
  ¦   Pb        ¦         910.700     ¦             622.116     ¦
  ¦   Li17Pb83  ¦         57.7500     ¦             405.441     ¦
  ¦   H2        ¦         80.0000     ¦             63.6936     ¦
  ¦   H2O       ¦         4.50900     ¦            1.680857E-03 ¦
  ¦   Li2O      ¦         74.3300     ¦             37.2098     ¦
  ¦   LIOH      ¦         11.3400     ¦             52.9675     ¦
  --------------------------------------------------------------
```

Before discussing the results of the analysis, the process of choosing an initial composition needs to be clarified. As shown on the last page, the program was executed twice, with only the initial composition varying between the two runs. If one compares the equilibrium composition of the two runs, one can see that the initial composition does not affect the program results. Although one may be led to believe that the program will perform properly regardless of the chosen initial composition, this is not the case. The user must supply the program with nonzero initial species masses. This may be a very complicated task if the number of species is much larger than the number of elements. In very simple terms, the process of choosing an initial composition can be explained in the following manner. The first step is to determine the constrained element masses. For the $Li_{17}Pb_{83}/H_2O$ system discussed above, since we know that the experiment consisted of the mixing of 90.18 mole of H_2O and 1155 mole of $Li_{17}Pb_{83}$, determining the constrained element masses is relatively easy. Next the user randomly sets a few of the initial species masses to nonzero values. He can then determine the remaining species masses by a process of elimination and random setting of values. In practice, the user may find the process of choosing a suitable initial composition a difficult task, but it can be accomplished with a little patience and experimentation.

Now turning to the results of the analysis, $Li_{17}Pb_{83}$ and H_2O react by two possible reaction paths:

$$Li_{17}Pb_{83} + .17\ H_2O \Rightarrow .17\ LiOH + .085\ H_2 + .83\ Pb \quad (35)$$
$$2\ Li_{17}Pb_{83} + .17\ H_2O \Rightarrow .17\ Li_2O + .17\ H_2 + 1.66\ Pb \quad (36)$$

Since the predicted equilibrium composition contains 52.97 mole of LiOH and 37.21 mole of Li_2O, and since practically all of the 90.18 mole of H_2O reacted, one can show that 41.26% of the H_2O reacted by the first route (equation (35)), and 58.74% of the H_2O reacted by the second route (equation(36)). But since a total of 99 mole of H_2 was collected during the experiment, all of the H_2O reacted to form one mole of H_2 per mole of H_2O reacted. Thus experimentally, the H_2O reacted almost exclusively by the second route (equation(36)). Analysis of the reaction products shows that, experimentally, about 68% of the lithium in the alloy pool reacted. This compares favorably to the program results, which show that 64.86% of the lithium would react in an equilibrium system. Thus the program correctly predicts the extent of the reaction. Therefore, one can conclude that the HEDL large scale experiments did not produce an equilibrium interaction, despite the practically instantaneous and complete reaction observed.

viii) Conclusion

The program presented here is a very effective and easy to use tool. The program can be applied to any modest sized system with little effort. Although the program may not execute efficiently with systems that contain many trace species, using the program is a much more simple task than writing an efficient program for the task at hand.

Nomenclature

a_{ie}	- The array of stoichiometric coefficients.
B_e	- The mass balanced constrained mass of element e.
b_e	- The current mass of element e.
D_i	- A factor in the $\delta\phi_i$ equation.
E_i	- A factor in the $\delta\phi_i$ equation.
G	- The system Gibbs free energy.
$G^o(T)$	- The Gibbs free energy of species i at 1 atm.
H	- The system enthalpy.
$H^o(298)$	- The enthalpy of species i at 1 atm. and 298K.
L	- The system Lagrangian.
M	- The number of elements in the system.
m_i	- The mass of species i from the previous step.
N	- the number of species in the system.
n_i	- The current mass of species i.
p	- The system pressure.
p_o	- Atmospheric pressure.
S	- The system entropy.
T	- The system temperature.
V	- The system volume.
$\Delta G_f^o(T)$	- The Gibbs free energy of formation at 1 atm.
$\Delta H_f^o(298)$	- The heat of formation at 1 atm. and 298K.
ϵ	- The mass balance convergence parameter.
π_e	- A factor for the second Lagrangian multiplier.
μ_i	- The chemical potential of species i.
μ_i^o	- The chemical potential of species i at 1 atm.

σ	- The program step size.
τ	- The minimum program step size.
ϕ_i	- The transformed form of n_i.
χ_i	- The first Lagrangian multiplier.
χ_ϕ	- The second Lagrangian multiplier.
ω	- A factor for the second Lagrangian multiplier.

References

1. F. Van Zeggeren and S. H. Storey, The Computation of Chemical Equilibria, Cambridge Press, (1970), Chap. 3.3.1.
2. Ibid.
3. Ibid., p. 9.
4. Ibid., p. 11.
5. F. Van Zeggeren and S. H. Storey, "Computation of Chemical Equilibrium Compositions II", The Can. J. of Chem. Engin., 18, (Oct. 1970), p. 592.
6. F. Van Zeggeren and S. H. Storey, The Computation of Chemical Equilibria, Cambridge Press, (1970), p. 31.
7. D. R. Stull, JANAF Thermochemical Tables.
8. "LWRSR Program Quarterly Report", 12, (April-June 1979), NUREG/CR-1112/, p. 26.
9. F. Van Zeggeren and S. H. Storey, The Computation of Chemical Equilibria, Cambridge Press, (1970), p. 31.
10. D Jeppson, et. al., "Fusion Safety Support Studies Progress Report", UCRL-53333 (Feb.-May 1984).
11. D Jeppson, et. al., "Fusion Safety Support Studies Progress Report", UCRL-53333 (June 1984).

Appendix

This appendix contains the listing of the computer program in three sections. The first section contains the main program listing. The next section contains the subroutine used in the propane/air application. The final section contains the subroutine used in the $Li_{17}Pb_{83}/H_2O$ application. The program is written in standard FORTRAN.

```
C                                                                       CHEMQ
C*********************************************************************** CHEMQ
C                                                                       CHEMQ
C     CHEMEQ Program                             By Jim Herzog 4/86 - 5/86 CHEMQ
C                                                                       CHEMQ
C      This program evaluates the chemical eqilibrium of a heterogeneousCHEMQ
C  mixture.  It assumes that the mixture consists of up to 3 homogeneousCHEMQ
C  phases ( liquid, solid, and gas ).                                   CHEMQ
C      The program is based on the fist-order steepest descent method of CHEMQ
C  Van Zeggeren and Storey.  The program is similar to the chemical     CHEMQ
C  equilibrium subroutine used in CORCON ( MLTREA subroutine ), but is  CHEMQ
C  more general and less powerful than MLTREA.                          CHEMQ
C      This program is designed to work with simple systems; number of  CHEMQ
C  species <= 20, number of elements <= 10, and seperated phases.  For  CHEMQ
C  the program to work with any system, all one needs to do is to adjustCHEMQ
C  the CHEMO subroutine and adjust the data file CHEMIN.DAT             CHEMQ
C                                                                       CHEMQ
C*********************************************************************** CHEMQ
C                                                                       CHEMQ
C     References:                                                       CHEMQ
C  1.  S. H. Storey and F. Van Zeggeren, 'The Computation of Chemical   CHEMQ
C      Equilibrium Compositions II', Can. Jour. of Chem. Eng., 48,      CHEMQ
C      Oct. 1970), p. 591-593                                           CHEMQ
C  2.  F. Van Zeggeren and S. H. Storey, 'The Computation of Chemical   CHEMQ
C      Equilibrium', Cambridge Univ. Press, Cambridge Ma. (1970).       CHEMQ
C                                                                       CHEMQ
C*********************************************************************** CHEMQ
C                                                                       CHEMQ
C  Definition of Variables                                              CHEMQ
C                                                                       CHEMQ
C     Let S denote species;  S = 1, NUMSPE                              CHEMQ
C     Let E denote elements; E = 1, NUMELE                              CHEMQ
C                                                                       CHEMQ
C     Input / Output:                                                   CHEMQ
C  P          = Pressure ( N/M**2 )                                     CHEMQ
C  T          = Temperature ( K )                                       CHEMQ
C  XMOL(S)    = Number of moles of the species ( MOLE )                 CHEMQ
C  CHEMPO(S)  = Chemical potential evaluated at T and 1 atm ( J/MOLE )  CHEMQ
C  ASC(E,S)   = Array of stoichiometric coefficients                    CHEMQ
C  PHASE(S)   = Phase of the species                                    CHEMQ
C  DSTEP      = Step size                                               CHEMQ
C  CONVRG     = Convergence parameter for minimum step size             CHEMQ
C  EPSILN     = Convergence Epsilon for element balance conversion      CHEMQ
C  TRACE      = Trace species cutoff parameter                          CHEMQ
C  NDEBUG     = Debug output parameter                                  CHEMQ
C  SPLIST(S)  = A character list of problem specific species            CHEMQ
C  ELLIST(E)  = A character list of problem specific elements           CHEMQ
C  PTITLE     = The problem run title                                   CHEMQ
C                                                                       CHEMQ
C     Internal Arrays:                                                  CHEMQ
C  XINT(S)    = Amount of species at start of current step              CHEMQ
C  XBEG(S)    = Initial amount of species                               CHEMQ
C  CHEMPO(S)  = Chemical potential of the species                       CHEMQ
C  BELE(E)    = Element balances for the problem                        CHEMQ
C  BDIF(E)    = Difference between the problem element balances and the CHEMQ
C               current element balance values                          CHEMQ
C  BNEW(E)    = Updated element balances                                CHEMQ
```

```
C  ETA(E)      = 1st Lagrangian multiplier factor                       CHEMQ
C  OMEGA(E)    = 1st Lagrangian multiplier factor                       CHEMQ
C  D(S)        = Factor in the delta phi equation                       CHEMQ
C  E(S)        = Factor in the delta phi equation                       CHEMQ
C  DELPHI(S)   = Delta phi                                              CHEMQ
C  DELPHO(S)   = Delta phi of previous step                             CHEMQ
C  DUMVEC(E)   = Iterative calculational vector                         CHEMQ
C  DUMRAY(E,E)= Iterative calculational array                           CHEMQ
C  XMAX(S)     = The maximum possible amount of each species            CHEMQ
C  XDUM(S)     = Adummy vector used to evaluate XMAX                    CHEMQ
C                                                                       CHEMQ
C     Internal Scalars:                                                 CHEMQ
C  GNEW        = Updated Gibb's free energy ( J )                       CHEMQ
C  GOLD        = Gibb's free energy of previous iteration               CHEMQ
C  XPHI        = 2nd Lagrangian multiplier                              CHEMQ
C  NUMELE      = The number of elements in the problem                  CHEMQ
C  NUMSPE      = The number of species in the problem                   CHEMQ
C  XMOLG       = The moles of gas in the system                         CHEMQ
C  XMOLL       = The moles of liquid in the system                      CHEMQ
C  XMOLS       = The moles of solid in the system                       CHEMQ
C  DSTEPO      = Step size at the beginning of the program              CHEMQ
C  DIREC       = The current direction of the Gibbs free energy surface CHEMQ
C                                                                       CHEMQ
C********************************************************************** CHEMQ
C                                                                       CHEMQ
      IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )                           CHEMQ
      IMPLICIT INTEGER  ( I-N )                                         CHEMQ
C                                                                       CHEMQ
      DIMENSION  XMOL(20), CHEMPO(20), ASC(10,20), XBEG(20), ETA(10),   CHEMQ
     1           XINT(20), CHEMPO(20), BELE(10), BDIF(10), DELPHI(20),  CHEMQ
     2           DUMVEC(10), DUMRAY(10,10), OMEGA(10), BNEW(10), D(20), CHEMQ
     3           E(20), DELPHO(20), XMAX(20), XDUM(20)                  CHEMQ
C                                                                       CHEMQ
      CHARACTER*1  PHASE(20)                                            CHEMQ
C                                                                       CHEMQ
      COMMON  / PARAM /                                                 CHEMQ
     1  DSTEP, CONVRG, EPSILN, TRACE                                    CHEMQ
      COMMON  / CHEM /                                                  CHEMQ
     1  NUMSPE, NUMELE,                                                 CHEMQ
     2  PHASE, ASC                                                      CHEMQ
C                                                                       CHEMQ
      IWRITE = 0                                                        CHEMQ
      IDEBUG = 11                                                       CHEMQ
      EPSMAT = .001                                                     CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     INPUT VARIABLES, GIBB'S FREE ENERGY, AND CHEMICAL POTENTIALS      CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     First we set needed input variables by calling the input subrout- CHEMQ
C  ine CHEMIN                                                           CHEMQ
      CALL CHEMIN  (                                                    CHEMQ
     1                                                                  CHEMQ
     2                                                                  CHEMQ
     3  T, P, XBEG, NDEBUG  )                                           CHEMQ
C                                                                       CHEMQ
      DSTEPO = DSTEP                                                    CHEMQ
      IF  ( NDEBUG .EQ. 1 )  OPEN ( 11, FILE = 'DEBUG.DAT', STATUS =    CHEMQ
     1      'NEW' )                                                     CHEMQ
C                                                                       CHEMQ
C     Next we call the subroutine CHEMO, which gives the temperature    CHEMQ
C  dependant variables CHEMPO(S)                                        CHEMQ
      CALL CHEMO  (                                                     CHEMQ
     1  T,                                                              CHEMQ
     2                                                                  CHEMQ
     3  CHEMPO  )                                                       CHEMQ
C                                                                       CHEMQ
C     Next we evaluate the starting values of the Gibb's free energy    CHEMQ
C  and the chemical potential of the species by calling subroutine GIBBSCHEMQ
      CALL GIBBS  (                                                     CHEMQ
     1  T, P, XBEG, CHEMPO,                                             CHEMQ
     2                                                                  CHEMQ
     3  GOLD, CHEMPO  )                                                 CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     ELEMENT ABUNDANCES, INITIAL SPECIES COMPOSITIONS AND SPECIES MAX. CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     The element abundances are set by mass balance constraints and    CHEMQ
C  must be held constant throughout the search for the equilibrium      CHEMQ
C  condition                                                            CHEMQ
      DO  100 J = 1, NUMELE                                             CHEMQ
        BELE(J) = 0.                                                    CHEMQ
        DO  105 I = 1, NUMSPE                                           CHEMQ
          BELE(J) = BELE(J)  +  ASC(J,I) * XBEG(I)                      CHEMQ
  105   CONTINUE                                                        CHEMQ
  100 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
      DO  150 I = 1, NUMSPE                                             CHEMQ
        XINT(I) = XBEG(I)                                               CHEMQ
  150 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
      DO  160 I = 1, NUMSPE                                             CHEMQ
        XMAX(I) = 1.D30                                                 CHEMQ
        DO  170 J = 1, NUMELE                                           CHEMQ
          XDUM(I) = ASC(J,I) / BELE(J)                                  CHEMQ
          IF  ( XDUM(I) .EQ. 0. )  GO TO 180                            CHEMQ
            XMAX(I) = DMIN1( XMAX(I), XDUM(I) )                         CHEMQ
  180     CONTINUE                                                      CHEMQ
  170   CONTINUE                                                        CHEMQ
  160 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
      IF  ( NDEBUG .EQ. 1 )  THEN                                       CHEMQ
        WRITE (IDEBUG,190)                                              CHEMQ
  190     FORMAT ( ' BELE ' )                                           CHEMQ
        WRITE (IDEBUG,372)  ( BELE(J), J = 1, NUMELE )                  CHEMQ
        WRITE (IDEBUG,191)                                              CHEMQ
  191     FORMAT ( ' XMAX ' )                                           CHEMQ
        WRITE (IDEBUG,462)  ( XMAX(I), I = 1, NUMSPE )                  CHEMQ
      END IF                                                            CHEMQ
C***********************                                                CHEMQ
C                                                                       CHEMQ
      NINT = 0                                                          CHEMQ
```

```
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     BEGINNING OF THE MAIN CALCULATIONAL LOOP                          CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
 1000 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
C     First we calculate the difference between the current element bal-CHEMQ
C  ances and the constrained balances                                   CHEMQ
         DO  200 J = 1, NUMELE                                          CHEMQ
           BDIF(J) = BELE(J)                                            CHEMQ
           DO  205 I = 1, NUMSPE                                        CHEMQ
             BDIF(J) = BDIF(J)  -  ASC(J,I) * XINT(I)                   CHEMQ
 205       CONTINUE                                                     CHEMQ
 200     CONTINUE                                                       CHEMQ
C                                                                       CHEMQ
C     Now we print intermediate values to the screen                    CHEMQ
         IF  ( DBLE( NINT / 10 ) .EQ. DBLE( NINT ) / 10. )  THEN        CHEMQ
           WRITE (IWRITE,215) NINT, GOLD                                CHEMQ
           WRITE (IWRITE,216)                                           CHEMQ
           WRITE (IWRITE,462)  ( XINT(I), I = 1, NUMSPE )               CHEMQ
         END IF                                                         CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
         IF  ( NDEBUG .EQ. 1 )  THEN                                    CHEMQ
           WRITE (IDEBUG,215) NINT, GOLD                                CHEMQ
 215         FORMAT ( ' --------------------------------------' /       CHEMQ
     1               ' ITERATION # = ', I5, ' G = ', 1P1G11.4 )         CHEMQ
           WRITE (IDEBUG,217)                                           CHEMQ
 217         FORMAT ( ' CHEMPO' )                                       CHEMQ
           WRITE (IDEBUG,462)  ( CHEMPO(I), I = 1, NUMSPE )             CHEMQ
           WRITE (IDEBUG,216)                                           CHEMQ
 216         FORMAT ( ' XINT ' )                                        CHEMQ
           WRITE (IDEBUG,462)  ( XINT(I), I = 1, NUMSPE )               CHEMQ
           WRITE (IDEBUG,220)                                           CHEMQ
 220         FORMAT ( ' BDIF ' )                                        CHEMQ
           WRITE (IDEBUG,372)  ( BDIF(J), J = 1, NUMELE )               CHEMQ
         END IF                                                         CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     SET UP AND SOLVE EQUATIONS FOR LAGRANGIAN MULTIPLIERS             CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     Set up array on left hand side of equations                       CHEMQ
         DO  300 J = 1, NUMELE                                          CHEMQ
           DO  310 L = J, NUMELE                                        CHEMQ
             SUM = 0.                                                   CHEMQ
             DO  320 I = 1, NUMSPE                                      CHEMQ
               SUM = SUM  +  ASC(J,I) * ASC(L,I) * XINT(I) * XINT(I)    CHEMQ
 320         CONTINUE                                                   CHEMQ
             DUMRAY(J,L) = SUM                                          CHEMQ
             DUMRAY(L,J) = SUM                                          CHEMQ
 310       CONTINUE                                                     CHEMQ
 300     CONTINUE                                                       CHEMQ
C                                                                       CHEMQ
C     Set up vector on right hand side of eta equation                  CHEMQ
```

```
      DO  350 J = 1, NUMELE                                             CHEMQ
        SUM = 0.                                                        CHEMQ
        DO  360 I = 1, NUMSPE                                           CHEMQ
          SUM = SUM  +  ASC(J,I) * CHEMPO(I) * XINT(I) * XINT(I)        CHEMQ
  360   CONTINUE                                                        CHEMQ
        DUMVEC(J) = SUM                                                 CHEMQ
  350 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
      IF  ( NDEBUG .EQ. 1 )  THEN                                       CHEMQ
        WRITE (IDEBUG,370)                                              CHEMQ
  370     FORMAT ( ' INTERMEDIATE ARRAY  ' )                            CHEMQ
        DO  371 J = 1, NUMELE                                           CHEMQ
          WRITE (IDEBUG,372)  ( DUMRAY(J,L), L = 1, NUMELE )            CHEMQ
  371   CONTINUE                                                        CHEMQ
        WRITE (IDEBUG,373)                                              CHEMQ
  373     FORMAT ( 10X, ' INTERMEDIATE VECTOR ' )                       CHEMQ
        WRITE (IDEBUG,372)  ( DUMVEC(J), J = 1, NUMELE )                CHEMQ
  372     FORMAT ( 5X, 1P5G11.4 / 6X, 1P5G11.4 )                        CHEMQ
      END IF                                                            CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C     Now to solve for the inverse of the intermediate array DUMRAY     CHEMQ
C  by calling the math subroutine MATINV                                CHEMQ
      CALL MATINV (                                                     CHEMQ
     1   NUMELE, EPSMAT, IWRITE,                                        CHEMQ
     2   DUMRAY,                                                        CHEMQ
     3   DETER  )                                                       CHEMQ
C                                                                       CHEMQ
C

C***** DEBUG DATA *****                                                 CHEMQ
      IF  ( NDEBUG .EQ. 1 )  THEN                                       CHEMQ
        WRITE (IDEBUG,375)                                              CHEMQ
  375     FORMAT ( ' INTERMEDIATE ARRAY DETERMINANT ' )                 CHEMQ
        WRITE (IDEBUG,*) DETER                                          CHEMQ
        WRITE (IDEBUG,376)                                              CHEMQ
  376     FORMAT ( ' INTERMEDIATE ARRAY INVERSE ' )                     CHEMQ
        DO  377 J = 1, NUMELE                                           CHEMQ
          WRITE (IDEBUG,372)  ( DUMRAY(J,L), L = 1, NUMELE )            CHEMQ
  377   CONTINUE                                                        CHEMQ
      END IF                                                            CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C     We can now solve for the additive factors in the first Lagrangian CHEMQ
C  multiplier ( ETA(NUMELE) and OMEGA(NUMELE) )                         CHEMQ
      DO  380 I = 1, NUMELE                                             CHEMQ
        ETA(I) = 0.                                                     CHEMQ
        OMEGA(I) = 0.                                                   CHEMQ
        DO  381 J = 1, NUMELE                                           CHEMQ
          ETA(I) = ETA(I)  +  DUMRAY(I,J) * DUMVEC(J)                   CHEMQ
          OMEGA(I) = OMEGA(I)  -  DUMRAY(I,J) * BDIF(J)                 CHEMQ
  381   CONTINUE                                                        CHEMQ
  380 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
      IF  ( NDEBUG .EQ. 1 )  THEN                                       CHEMQ
        WRITE (IDEBUG,390)                                              CHEMQ
```

```
 390        FORMAT ( ' ETA ' )                                          CHEMQ
          WRITE (IDEBUG,372)  ( ETA(J), J = 1, NUMELE )                 CHEMQ
          WRITE (IDEBUG,391)                                            CHEMQ
 391        FORMAT ( ' OMEGA ' )                                        CHEMQ
          WRITE (IDEBUG,372)  ( OMEGA(J), J = 1, NUMELE )               CHEMQ
        END IF                                                          CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     FIND THE FACTORS D AND E                                          CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
        DO  400 I = 1, NUMSPE                                           CHEMQ
          E(I) = 0.                                                     CHEMQ
          SUM = 0.                                                      CHEMQ
          DO  410 J = 1, NUMELE                                         CHEMQ
            SUM = SUM  +  ETA(J) * ASC(J,I)                             CHEMQ
            E(I) = E(I)  +  OMEGA(J) * ASC(J,I) * XINT(I)               CHEMQ
 410      CONTINUE                                                      CHEMQ
          D(I) = XINT(I) * ( CHEMPO(I) - SUM )                          CHEMQ
 400    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
        IF  ( NDEBUG .EQ. 1 )  THEN                                     CHEMQ
          WRITE (IDEBUG,415)                                            CHEMQ
 415        FORMAT ( ' D ' )                                            CHEMQ
          WRITE (IDEBUG,462)  ( D(I), I = 1, NUMSPE )                   CHEMQ
          WRITE (IDEBUG,416)                                            CHEMQ
 416        FORMAT ( ' E ' )                                            CHEMQ
          WRITE (IDEBUG,462)  ( E(I), I = 1, NUMSPE )                   CHEMQ
        END IF                                                          CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     SET UP AND SOLVE FOR 2ND LAGRANGIAN MULTIPLIER                    CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     First we calculate the the summation terms in the equation        CHEMQ
        SUMD = 0.                                                       CHEMQ
        SUME = 0.                                                       CHEMQ
        DO  420 I = 1, NUMSPE                                           CHEMQ
          SUMD = SUMD  +  D(I) * D(I)                                   CHEMQ
          SUME = SUME  +  E(I) * E(I)                                   CHEMQ
 420    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C     Now we can calculate the 2nd Lagrangian multiplier.  We must make CHEMQ
C  sure that DSTEP**2 <= SUME, if it isn't, we adjust the step size     CHEMQ
 2000   CONTINUE                                                        CHEMQ
        IF  ( ( DSTEP * DSTEP ) .LE. SUME )  THEN                       CHEMQ
          DSTEP = 1.5 * DSTEP                                           CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
          IF  ( NDEBUG .EQ. 1 )  THEN                                   CHEMQ
            WRITE (IDEBUG,430) DSTEP                                    CHEMQ
 430          FORMAT ( ' DSTEP IS TOO SMALL - NEW DSTEP = ', 1P1G11.4 ) CHEMQ
          END IF                                                        CHEMQ
C**********************                                                 CHEMQ
          GO TO 2000                                                    CHEMQ
        END IF                                                          CHEMQ
C                                                                       CHEMQ
        XPHI = -1. *  DSQRT( SUMD / ( DSTEP * DSTEP  -  SUME ) )        CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     CALCULATE DELPHI                                                  CHEMQ
C---------------------------------------------------------------------- CHEMQ
        DO  450 I = 1, NUMSPE                                           CHEMQ
          DELPHI(I) = ( D(I) / XPHI ) - E(I)                            CHEMQ
 450    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C***** DEBUG DATA *****                                                 CHEMQ
        IF  ( NDEBUG .EQ. 1 )  THEN                                     CHEMQ
          WRITE (IDEBUG,460) XPHI                                       CHEMQ
 460        FORMAT ( ' XPHI = ', 1P1G11.4 )                             CHEMQ
          WRITE (IDEBUG,461)                                            CHEMQ
 461        FORMAT ( ' DELPHI ' )                                       CHEMQ
          WRITE (IDEBUG,462)  ( DELPHI(I), I = 1, NUMSPE )              CHEMQ
 462        FORMAT ( 5X, 1P5G11.4 / 6X, 1P5G11.4 / 7X, 1P5G11.4 /       CHEMQ
     1                8X, 1P5G11.4 )                                    CHEMQ
        END IF                                                          CHEMQ
C**********************                                                 CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     CALCULATE XMOL AND GNEW                                           CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
        DO  500 I = 1, NUMSPE                                           CHEMQ
          XMOL(I) = XINT(I)  *  DEXP( DELPHI(I) )                       CHEMQ
 500    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C     Find the new Gibbs free energy and the chemical potential         CHEMQ
        CALL GIBBS  (                                                   CHEMQ
     1    T, P, XMOL, CHEMPO,                                           CHEMQ
     2                                                                  CHEMQ
     3    GNEW, CHEMPO  )                                               CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     CHECK CONVERGENCE CONDITIONS                                      CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     First we evaluate the new element balances                        CHEMQ
        DO  510 J = 1, NUMELE                                           CHEMQ
          BNEW(J) = 0.                                                  CHEMQ
          DO  515 I = 1, NUMSPE                                         CHEMQ
            BNEW(J) = BNEW(J)  +  ASC(J,I) * XMOL(I)                    CHEMQ
 515      CONTINUE                                                      CHEMQ
 510    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C     Now we check to see if the new element abundances have converged CHEMQ
C  to the element abundances set by mass balance constraints            CHEMQ
        IBCHCK = 0                                                      CHEMQ
        DO  520 J = 1, NUMELE                                           CHEMQ
          BCHECK = DABS( BELE(J) - BNEW(J) )  /  BELE(J)                CHEMQ
          IF  ( BCHECK .GT. EPSILN )  IBCHCK = 1                        CHEMQ
 520    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
```

```
C     Now we check to see if the step size has converged to the desired CHEMQ
C  minimum value                                                        CHEMQ
        ISCHCK = 0                                                      CHEMQ
        IF  ( DSTEP .GT. CONVRG )  ISCHCK = 1                           CHEMQ
C                                                                       CHEMQ
C     If both IBCHCK and ISCHCK equal 0, then the minimum Gibbs free    CHEMQ
C  energy has been found.                                               CHEMQ
        IF  ( ( IBCHCK .EQ. 0 ) .AND. ( ISCHCK .EQ. 0 ) )  GO TO 9000   CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     ELIMINATE TRACE SPECIES                                           CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
        DO  550 I = 1, NUMSPE                                           CHEMQ
          IF  ( XMOL(I) .LT. ( TRACE * XMAX(I) ) )  XMOL(I) = 0.        CHEMQ
 550    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     ADJUST STEP SIZE                                                  CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
C     First we determine the direction the program is traveling along   CHEMQ
C  the Gibbs free energy surface ( and a measure of the surface steep-  CHEMQ
C  ness ). The step size is then adjusted accordingly.                  CHEMQ
        IF  ( NINT .EQ. 0 )  THEN                                       CHEMQ
          DSTEP = .5 * DSTEP                                            CHEMQ
        ELSE                                                            CHEMQ
          SUM = 0.                                                      CHEMQ
          DO  600 I = 1, NUMSPE                                         CHEMQ
            SUM = SUM  +  DELPHI(I) * DELPHO(I)                         CHEMQ
 600      CONTINUE                                                      CHEMQ
          DIREC = SUM  / ( DSTEP * DSTEP )                              CHEMQ
          IF  ( DIREC .LT. 0. )  DSTEP = .5 * DSTEP                     CHEMQ
          IF  ( DIREC .GT. .7 )  DSTEP = DMIN1( DSTEP0, 2.*DSTEP )      CHEMQ
        END IF                                                          CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     RESET LOOP VARIABLES                                              CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
        DO  700 I = 1, NUMSPE                                           CHEMQ
          XINT(I) = XMOL(I)                                             CHEMQ
          DELPHO(I) = DELPHI(I)                                         CHEMQ
 700    CONTINUE                                                        CHEMQ
C                                                                       CHEMQ
        GOLD = GNEW                                                     CHEMQ
        NINT = NINT + 1                                                 CHEMQ
C                                                                       CHEMQ
C     We now return to the beginning of the loop                        CHEMQ
        GO TO 1000                                                      CHEMQ
C                                                                       CHEMQ
C---------------------------------------------------------------------- CHEMQ
C     MINIMUM GIBBS FREE ENERGY FOUND!                                  CHEMQ
C---------------------------------------------------------------------- CHEMQ
C                                                                       CHEMQ
 9000 CONTINUE                                                          CHEMQ
C                                                                       CHEMQ
```

```
C     We have found the minimum Gibb's free energy and can now call the CHEMQ
C  output subroutine (CHOUT)                                            CHEMQ
      CALL CHOUT  (                                                     CHEMQ
     1  T, P, XBEG, GNEW, XMOL, BELE, NINT                              CHEMQ
     2                                                                  CHEMQ
     3  )                                                               CHEMQ
C                                                                       CHEMQ
C     That's all folks!                                                 CHEMQ
      END                                                               CHEMQ
C                                                                       CHEMQ
C####################################################################### CHIN
C                                                                       CHIN
      SUBROUTINE CHEMIN  (                                              CHIN
     1                                                                  CHIN
     2                                                                  CHIN
     3  T, P, XMOL, NDEBUG  )                                            CHIN
C                                                                       CHIN
C     This subroutine is the user interface that sets the input         CHIN
C  variables used in the main program CHEMEQ                            CHIN
      IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )                           CHIN
      IMPLICIT INTEGER  ( I-N )                                         CHIN
C                                                                       CHIN
      DIMENSION  XMOL(20),  ASC(10,20)                                  CHIN
C                                                                       CHIN
      CHARACTER*1  PHASE(20), CPARAM, CDEBUG                            CHIN
      CHARACTER*10  SPLIST(20), ELLIST(10)                              CHIN
      CHARACTER*12  DATFIL                                              CHIN
      CHARACTER*80  PTITLE                                              CHIN
C                                                                       CHIN
      COMMON  / PARAM /                                                 CHIN
     1  DSTEP, CONVRG, EPSILN, TRACE                                    CHIN
      COMMON  / CHEM /                                                  CHIN
     1  NUMSPE, NUMELE,                                                 CHIN
     2  PHASE, ASC                                                      CHIN
      COMMON  / PRLST /                                                 CHIN
     1  SPLIST, ELLIST, PTITLE                                          CHIN
C                                                                       CHIN
      IWRITE = 0                                                        CHIN
      IREAD = 0                                                         CHIN
      IDATA = 12                                                        CHIN
C                                                                       CHIN
C---------------------------------------------------------------------- CHIN
C     READ IN PROBLEM SPECIFIC DATA FROM SPECIFIED DATA FILE            CHIN
C---------------------------------------------------------------------- CHIN
C                                                                       CHIN
C     We must first ask the user to specify the name of the input data  CHIN
C  file                                                                 CHIN
      WRITE (IWRITE,5)                                                  CHIN
 5      FORMAT ( 5X, ' Enter the name of the input data file [',        CHIN
     1            'filename.ext]' )                                     CHIN
      READ (IREAD,6) DATFIL                                             CHIN
 6      FORMAT ( 12A )                                                  CHIN
      OPEN ( IDATA, FILE = DATFIL )                                     CHIN
C                                                                       CHIN
C     First we read in the number of species and elements               CHIN
      READ (IDATA,*)  NUMSPE, NUMELE                                    CHIN
C                                                                       CHIN
```

```
C     Now we read in the character lists SPLIST and ELLIST              CHIN
      DO  10 I = 1, NUMSPE                                              CHIN
        READ (IDATA,11)  SPLIST(I)                                      CHIN
   11     FORMAT (10A)                                                  CHIN
   10 CONTINUE                                                          CHIN
C                                                                       CHIN
      DO  15 J = 1, NUMELE                                              CHIN
        READ (IDATA,11)  ELLIST(J)                                      CHIN
   15 CONTINUE                                                          CHIN
C                                                                       CHIN
C     Lastly, we read in the stoichiometric coefficient array ASC       CHIN
      DO  20 J = 1, NUMELE                                              CHIN
        READ (IDATA,*)  ( ASC(J,I), I = 1, NUMSPE )                     CHIN
   20 CONTINUE                                                          CHIN
C                                                                       CHIN
C---------------------------------------------------------------------- CHIN
C     PROMPT THE USER FOR PROBLEM PARAMETERS                            CHIN
C---------------------------------------------------------------------- CHIN
C                                                                       CHIN
C     First we prompt the user for the main problem variables; the      CHIN
C  system temperature and pressure, and the initial species mass        CHIN
      WRITE (IWRITE,100)                                                CHIN
  100   FORMAT ( ' Enter the problem title, up to 80 characters ' )     CHIN
      READ (IREAD,105)  PTITLE                                          CHIN
  105   FORMAT ( 80A )                                                  CHIN
C                                                                       CHIN
      WRITE (IWRITE,110)                                                CHIN
  110   FORMAT ( 5X,' Enter the system temperature [K] and the system', CHIN
     1           ' pressure [N/M**2] ' )                                CHIN
      READ (IREAD,*)  T, P                                              CHIN
C                                                                       CHIN
      WRITE (IWRITE,115)                                                CHIN
  115   FORMAT ( 5X,' Enter the initial mass [MOLE] of ' )              CHIN
      DO  120 I = 1, NUMSPE                                             CHIN
        WRITE (IWRITE,125)  I, SPLIST(I)                                CHIN
  125     FORMAT ( ' species number', I3, ': ', 10A )                   CHIN
        READ (IREAD,*)  XMOL(I)                                         CHIN
  120 CONTINUE                                                          CHIN
C                                                                       CHIN
C     Lastly, we prompt the user to change the problem parameters if he CHIN
C  wants to, and  ask him if he wants debug output                      CHIN
      DSTEP  = 2.5                                                      CHIN
      CONVRG = 1.D-4                                                    CHIN
      EPSILN = 1.D-4                                                    CHIN
      TRACE =  1.D-5                                                    CHIN
      WRITE (IWRITE,200)  DSTEP, CONVRG, EPSILN, TRACE                  CHIN
  200   FORMAT ( ' The initial value of the step size =      ',         CHIN
     1           1P1G11.4 /                                             CHIN
     2           ' The step size convergence parameter = ',             CHIN
     3           1P1G11.4 /                                             CHIN
     4           ' The mass balance convergence parameter = ',          CHIN
     5           1P1G11.4 /                                             CHIN
     6           ' The trace species cutoff parameter = ',              CHIN
     7           1P1G11.4 /                                             CHIN
     8           5X, ' Do you want to change these parameters',         CHIN
     9           ' [y/n] ?   '  )                                       CHIN
      READ (IREAD,210)  CPARAM                                          CHIN
  210   FORMAT ( A )                                                    CHIN
      IF  ( ( CPARAM .EQ. 'Y' ) .OR. ( CPARAM .EQ. 'y' ) )  THEN        CHIN
        WRITE (IWRITE,220)                                              CHIN
  220     FORMAT ( ' Enter the new initial step size ' )                CHIN
        READ (IREAD,*)  DELLAM                                          CHIN
        WRITE (IWRITE,221)                                              CHIN
  221     FORMAT ( ' Enter the new step size convergence parameter ' )  CHIN
        READ (IREAD,*)  CONVRG                                          CHIN
        WRITE (IWRITE,222)                                              CHIN
  222     FORMAT ( ' Enter the new mass balance convergence',           CHIN
     1              ' parameter  ' )                                    CHIN
        READ (IREAD,*)  EPSILN                                          CHIN
        WRITE (IWRITE,223)                                              CHIN
  223     FORMAT ( ' Enter the new trace species cutoff parameter ' )   CHIN
        READ (IREAD,*)  TRACE                                           CHIN
      END IF                                                            CHIN
C                                                                       CHIN
      WRITE (IWRITE,230)                                                CHIN
  230   FORMAT ( ' Do you want debug output [y/n] ? ' )                 CHIN
      READ (IREAD,210)  CDEBUG                                          CHIN
      IF  ( ( CDEBUG .EQ. 'Y' ) .OR. ( CDEBUG .EQ. 'y' ) )  THEN        CHIN
        NDEBUG = 1                                                      CHIN
      ELSE                                                              CHIN
        NDEBUG = 0                                                      CHIN
      END IF                                                            CHIN
C                                                                       CHIN
C     That's all folks!                                                 CHIN
      RETURN                                                            CHIN
      END                                                               CHIN
C                                                                       CHIN
C*********************************************************************** GIBBS
C                                                                       GIBBS
      SUBROUTINE GIBBS  (                                               GIBBS
     1  T, P, XMOL, CHEMPO,                                             GIBBS
     2                                                                  GIBBS
     3  G, CHEMPO )                                                     GIBBS
C                                                                       GIBBS
C     This subroutine evaluates the Gibb's free energy and the chemical GIBBS
C  potential of the species                                             GIBBS
      IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )                           GIBBS
      IMPLICIT INTEGER  ( I-N )                                         GIBBS
C                                                                       GIBBS
      DIMENSION  XMOL(20), CHEMPO(20), CHEMPO(20), ASC(10,20)           GIBBS
C                                                                       GIBBS
      CHARACTER*1  PHASE(20)                                            GIBBS
C                                                                       GIBBS
      COMMON  / CHEM /                                                  GIBBS
     1  NUMSPE, NUMELE,                                                 GIBBS
     2  PHASE, ASC                                                      GIBBS
C                                                                       GIBBS
C     First we evaluate XMOLG, XMOLL, and XMOLS                         GIBBS
      XMOLG = 0.                                                        GIBBS
      XMOLL = 0.                                                        GIBBS
      XMOLS = 0.                                                        GIBBS
      DO  100 I = 1, NUMSPE                                             GIBBS
        IF  ( PHASE(I) .EQ. 'G' )  XMOLG = XMOLG  +  XMOL(I)            GIBBS
        IF  ( PHASE(I) .EQ. 'L' )  XMOLL = XMOLL  +  XMOL(I)            GIBBS
        IF  ( PHASE(I) .EQ. 'S' )  XMOLS = XMOLS  +  XMOL(I)            GIBBS
 100  CONTINUE                                                          GIBBS
C                                                                       GIBBS
C     Now we can evaluate G and CHEMPO                                  GIBBS
      G = 0.                                                            GIBBS
      RT = 8.3143 * T                                                   GIBBS
      DO  110 I = 1, NUMSPE                                             GIBBS
        IF  ( XMOL(I) .EQ. 0. )  THEN                                   GIBBS
          CHEMPO(I) = CHEMPO(I)                                         GIBBS
        ELSE                                                            GIBBS
          IF  ( PHASE(I) .EQ. 'G' )  CHEMPO(I) = CHEMPO(I)  +  RT *     GIBBS
     1        DLOG( ( P * XMOL(I) ) / ( 1.013D5 * XMOLG ) )             GIBBS
          IF  ( PHASE(I) .EQ. 'L' )  CHEMPO(I) = CHEMPO(I)  +  RT *     GIBBS
     1        DLOG( XMOL(I) / XMOLL )                                   GIBBS
          IF  ( PHASE(I) .EQ. 'S' )  CHEMPO(I) = CHEMPO(I)  +  RT *     GIBBS
     1        DLOG( XMOL(I) / XMOLS )                                   GIBBS
        END IF                                                          GIBBS
C                                                                       GIBBS
        G = G  +  CHEMPO(I) * XMOL(I)                                   GIBBS
 110  CONTINUE                                                          GIBBS
C                                                                       GIBBS
C     That's all folks!                                                 GIBBS
      RETURN                                                            GIBBS
      END                                                               GIBBS
C                                                                       GIBBS
C*********************************************************************** CHOUT
C                                                                       CHOUT
      SUBROUTINE CHOUT  (                                               CHOUT
     1  T, P, XBEG, G, XMOL, BELE, NINT                                 CHOUT
     2                                                                  CHOUT
     3  )                                                               CHOUT
C                                                                       CHOUT
C     This subroutine writes the initial species mass, system tempera-  CHOUT
C  ture, system pressure, equilibrium species mass, and the minimum     CHOUT
C  Gibb's free energy                                                   CHOUT
      IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )                           CHOUT
      IMPLICIT INTEGER  ( I-N )                                         CHOUT
C                                                                       CHOUT
      DIMENSION  XMOL(20), XBEG(20),  ASC(10,20), BELE(10)              CHOUT
C                                                                       CHOUT
      CHARACTER*10  SPLIST(20), ELLIST(10)                              CHOUT
      CHARACTER*1  PHASE(20)                                            CHOUT
      CHARACTER*80  PTITLE                                              CHOUT
C                                                                       CHOUT
      COMMON  / CHEM /                                                  CHOUT
     1  NUMSPE, NUMELE,                                                 CHOUT
     2  PHASE, ASC                                                      CHOUT
      COMMON  / PRLST /                                                 CHOUT
     1  SPLIST, ELLIST, PTITLE                                          CHOUT
C                                                                       CHOUT
      OPEN ( 11, FILE = 'CHEMOUT.DAT', STATUS = 'NEW' )                 CHOUT
C                                                                       CHOUT
      IOUT = 11                                                         CHOUT
C                                                                       CHOUT
      WRITE (IOUT,10)  PTITLE                                           CHOUT
 10     FORMAT ( 80A  / / )                                             CHOUT
C                                                                       CHOUT
```

```
      WRITE (IOUT,100)  T, P, G, NINT                                   CHOUT
  100   FORMAT ( 5X, ' For T = ' , 1P1G11.4, ' K  and  P = ',           CHOUT
     1           1P1G11.4, ' Pa ' / 5X, ' The minimum Gibbs free',      CHOUT
     2           ' energy = ', 1P1G11.4, ' J ' / 5X, ' The number of',  CHOUT
     3           ' iterations = ', I6 )                                 CHOUT
C                                                                       CHOUT
      WRITE (IOUT,102)                                                  CHOUT
  102   FORMAT (// 15X, '-----------------------------------' )         CHOUT
      WRITE (IOUT,105)                                                  CHOUT
  105   FORMAT (   15X, '|  Element   | Element mass [MOLE]|' /       CHOUT
     1             15X, '|------------|--------------------|' )        CHOUT
      DO  106 I = 1, NUMELE                                             CHOUT
        WRITE (IOUT,107)  ELLIST(I), BELE(I)                            CHOUT
  107     FORMAT ( 15X, '|', 2X, A10, '|', 5X, 1P1G15.6, '|' )          CHOUT
  106 CONTINUE                                                          CHOUT
        WRITE (IOUT,108)                                                CHOUT
  108     FORMAT (   15X, '-----------------------------------' )       CHOUT
C                                                                       CHOUT
      WRITE (IOUT,109)                                                  CHOUT
  109   FORMAT(//15X,'-------------------------------------------------',CHOUT
     1                '-------------' )                                 CHOUT
      WRITE (IOUT,110)                                                  CHOUT
  110   FORMAT ( 15X,'|  Species   | Initial mass [MOLE] | Equilibrium',CHOUT
     1                ' mass [MOLE]|' )                                 CHOUT
      WRITE (IOUT,111)                                                  CHOUT
  111   FORMAT ( 15X,'|------------|---------------------|------------',CHOUT
     1                '------------|' )                                 CHOUT
      DO  200 I = 1, NUMSPE                                             CHOUT
        WRITE (IOUT,201)  SPLIST(I), XBEG(I), XMOL(I)                   CHOUT
  201     FORMAT ( 15X, '|', 2X, A10, '|', 5X, 1P1G15.6, 1X, '|',       CHOUT
     1             8X, 1P1G15.6, 1X, '|' )                              CHOUT
  200 CONTINUE                                                          CHOUT
      WRITE (IOUT,250)                                                  CHOUT
  250   FORMAT ( 15X,'-------------------------------------------------',CHOUT
     1                '-------------' )                                 CHOUT
C                                                                       CHOUT
C     That's all folks!                                                 CHOUT
      RETURN                                                            CHOUT
      END                                                               CHOUT
C                                                                       CHOUT
C#######################################################################
C
      SUBROUTINE MATINV  (
     1  N, EPS, IW,
     2  A,
     3  FDETER )
C
C  Subroutine MATINV computes the determinant and the inverse
C  matrix by Gauss-Jordan elimination using maximal pivoting.
C
C    N---------  NUMBER OF MATRIX EQUATIONS  [A] (X) = (B)
C                CURRENT MAXIMUM IS 50
C    EPS-------  CHECK VALUE FOR NEAR SINGULARITY OF PIVOT ELEMENT
C    A---------  'LEFT' HAND 2D MATRIX
C    FDETER----  DETERMINANT
C    IW--------  UNIT NUMBER FOR ERROR MESSAGE OUTPUT
C
```

```
C  Taken from Carnahan, Luther, Wilkes p.290  by J. J. Barry
C
C     DOUBLE PRECISION   REAL*8 (USING 8087 ARITHMETIC)
C
 IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
 IMPLICIT INTEGER ( I-M )
C
      DIMENSION  IROW(50),JCOL(50),JORD(50),Y(50)
      DIMENSION  A(10,10),B(10)
C
C  Error report if exceed allowable dimensions.
C
      IF (N .LE. 50) GO TO 5
      WRITE (IW,301)
 301  FORMAT(' ERROR IN MATINV - exceeded maximum dimension')
      FDETER = 0.
      RETURN
C
C  Begin elimination procedure.
C
    5 DETER = 1.
      DO 18  K=1,N
         KM1 = K - 1
C
C  Search for pivot element.
C
         PIVOT = 0.
         DO 11  I=1,N
         DO 11  J=1,N
C
C  Scan IROW and JCOL arrays for invalid pivot subscripts.
C
            IF (K .EQ. 1) GOTO 9
            DO 8  ISCAN=1,KM1
            DO 8  JSCAN=1,KM1
               IF (I .EQ. IROW(ISCAN)) GOTO 11
               IF (J .EQ. JCOL(JSCAN)) GOTO 11
    8       CONTINUE
    9       IF (DABS(A(I,J)) .LE. DABS(PIVOT)) GOTO 11
            PIVOT = A(I,J)
            IROW(K) = I
            JCOL(K) = J
   11    CONTINUE
C
C  Insure that selected pivot is larger than eps.
C
         IF (DABS(PIVOT) .GT. EPS) GOTO 13
         FDETER = 0.
         RETURN
C
C  Update the determinant value.
C
   13    IROWK = IROW(K)
         JCOLK = JCOL(K)
         DETER = DETER * PIVOT
C
C  Normalize pivot row elements.
```

```
C
         DO 14  J=1,N
            A(IROWK,J) = A(IROWK,J)/PIVOT
   14    CONTINUE
C
C  Carry out elimination and develop inverse.
C
         A(IROWK,JCOLK) = 1./PIVOT
         DO 18  I=1,N
            AIJCK = A(I,JCOLK)
            IF (I .EQ. IROWK) GOTO 18
            A(I,JCOLK) = -AIJCK/PIVOT
            DO 17  J=1,N
               IF (J .NE. JCOLK) A(I,J) = A(I,J) - AIJCK*A(IROWK,J)
   17       CONTINUE
   18 CONTINUE
C
C  Order solution values (if any) and create JORD array.
C
      DO 20  I=1,N
         IROWI = IROW(I)
         JCOLI = JCOL(I)
         JORD(IROWI) = JCOLI
   20 CONTINUE
C
C  Adjust sign of determinant.
C
      INTCH = 0
      NM1 = N - 1
      DO 22  I=1,NM1
         IP1 = I + 1
         DO 22  J = IP1,N
            IF (JORD(J) .GE. JORD(I)) GOTO 22
            JTEMP = JORD(J)
            JORD(J) = JORD(I)
            JORD(I) = JTEMP
            INTCH = INTCH + 1
   22 CONTINUE
      IF ((INTCH/2)*2 .NE. INTCH) DETER = -DETER
C
C    Unscramble the inverse.
C   - first by rows
C
   26 DO 28  J=1,N
         DO 27  I=1,N
            IROWI = IROW(I)
            JCOLI = JCOL(I)
            Y(JCOLI) = A(IROWI,J)
   27    CONTINUE
         DO 28  I=1,N
            A(I,J) = Y(I)
   28 CONTINUE
C
C  Then by columns.
C
      DO 30  I=1,N
         DO 29  J=1,N
```

```
            IROWJ = IROW(J)
            JCOLJ = JCOL(J)
            Y(IROWJ) = A(I,JCOLJ)
   29    CONTINUE
         DO 30  J=1,N
            A(I,J) = Y(J)
   30 CONTINUE
C
      FDETER = DETER
      RETURN
      END
      SUBROUTINE CHEMO  (                                               CHEMO
     1  T,                                                              CHEMO
     2                                                                  CHEMO
     3 CHEMPO  )                                                        CHEMO
C                                                                       CHEMO
C       This subroutine calculates the chemical potential and the phase CHEMO
C  of the species specified in this application.  Where necessary, the CHEMO
C  the chemical potential data from the JANAF tables has been curve     CHEMO
C  fitted to give CHEMPO(T) formulas.  Only temperatures between 600K   CHEMO
C  and 1400K are valid                                                  CHEMO
      IMPLICIT DOUBLE PRECISION  ( A-H, O-Z )                           CHEMO
      IMPLICIT INTEGER  ( I-N )                                         CHEMO
C                                                                       CHEMO
      DIMENSION  ASC(10,20), CHEMPO(20), GHSTAN(20), HOF(20)            CHEMO
C                                                                       CHEMO
      CHARACTER*1  PHASE(20)                                            CHEMO
C                                                                       CHEMO
      COMMON  / CHEM /                                                  CHEMO
     1  NUMSPE, NUMELE,                                                 CHEMO
     2  PHASE, ASC                                                      CHEMO
C                                                                       CHEMO
C       This subroutine uses the alternate definition of the standard   CHEMO
C  chemical potential.                                                  CHEMO
C       CHEMPO = ( GIBBSO - HSTAND ) + HOF                              CHEMO
C  where, GIBBSO = Gibb's free energy at 1atm and T                     CHEMO
C         HSTAND = enthalpy at 1atm and T                               CHEMO
C         HOF    = heat of formation at 1atm and 298K                   CHEMO
C                                                                       CHEMO
C  defining  GHSTAN = ( GIBBSO - HSTAND ) / T , this subroutine         CHEMO
C  evaluates GHSTAN(T) for a given T and thus CHEMPO(T).  This data is  CHEMO
C  drawn from the JANAF tables.                                         CHEMO
C                                                                       CHEMO
C       Species 1:  Li                                                  CHEMO
      GHSTAN(1) = -22.347 - .033 * T  +  6.0758D-6 * T * T              CHEMO
      HOF(1) = 2.381D3                                                  CHEMO
      PHASE(1) = 'L'                                                    CHEMO
C                                                                       CHEMO
C       Species 2:  Pb                                                  CHEMO
      GHSTAN(2) = -59.831  -  .03266 * T  +  5.8283D-6 * T * T          CHEMO
      HOF(2) = 4.289D3                                                  CHEMO
      PHASE(2) = 'L'                                                    CHEMO
C                                                                       CHEMO
C       Species 3:  Li17Pb83                                            CHEMO
C  The properties of this species are extrapolated from the properties CHEMO
C  of Li and Pb                                                         CHEMO
      GHSTAN(3) = .17 * GHSTAN(1)  +  .83 * GHSTAN(2)                   CHEMO
      HOF(3) = -7.83D3                                                  CHEMO
      PHASE(3) = 'L'                                                    CHEMO
C                                                                       CHEMO
C       Species 4:  H2                                                  CHEMO
      GHSTAN(4) = -119.44  -  .03121 * T  +  5.23D-6 * T * T            CHEMO
      HOF(4) = 0.                                                       CHEMO
      PHASE(4) = 'G'                                                    CHEMO
C                                                                       CHEMO
C       Species 5:  H2O                                                 CHEMO
      GHSTAN(5) = -175.66  -  3.5924D-2 * T  +  4.962D-6 * T * T        CHEMO
      HOF(5) = -2.4183D5                                                CHEMO
      PHASE(5) = 'G'                                                    CHEMO
C                                                                       CHEMO
C     Species 6:  Li2O                                                  CHEMO
      GHSTAN(6) = -12.34  -  .06745 * T  +  7.831D-6 * T * T            CHEMO
      HOF(6) = -5.9873D5                                                CHEMO
      PHASE(6) = 'S'                                                    CHEMO
C                                                                       CHEMO
C     Species 7:  LiOH                                                  CHEMO
C  Here we have to be careful, since there is a change of phase         CHEMO
      IF  ( T .LE. 744.3 )  THEN                                        CHEMO
        GHSTAN(7) = -23.145  +  .05125 * T                              CHEMO
        HOF(7) = -4.8467D5                                              CHEMO
        PHASE(7) = 'S'                                                  CHEMO
      ELSE                                                              CHEMO
        GHSTAN(7) = -15.226  -  9.3556D-2 * T  +  1.6268D-5 * T * T     CHEMO
        HOF(7) = -4.7389D5                                              CHEMO
        PHASE(7) = 'L'                                                  CHEMO
      END IF                                                            CHEMO
C                                                                       CHEMO
      DO  100 I = 1, NUMSPE                                             CHEMO
        CHEMPO(I) = GHSTAN(I) / T  +  HOF(I)                            CHEMO
  100 CONTINUE                                                          CHEMO
C                                                                       CHEMO
C     That's all folks!                                                 CHEMO
      RETURN                                                            CHEMO
      END                                                               CHEMO
```

```
$STORAGE:2
$NOFLOATCALLS
      SUBROUTINE CHEMO ( T, CHEMPO )
      IMPLICIT DOUBLE PRECISION ( A-H, O-Z )
      IMPLICIT INTEGER (I-N)
      DIMENSION  ASC(10,20), CHEMPO(20)
      CHARACTER*1  PHASE(20)
      COMMON / CHEM / NUMSPE, NUMELE, PHASE, ASC
C
C     CO2
        CHEMPO(1) = -3.964090D5
        PHASE(1) = 'G'
C
C     N2
        CHEMPO(2) = 0.
        PHASE(2) = 'G'
C
C     H2O
        CHEMPO(3) = -1.239340D5
        PHASE(3) = 'G'
C
C     CO
        CHEMPO(4) = -3.026496D5
        PHASE(4) = 'G'
C
C     H2
        CHEMPO(5) = 0.
        PHASE(5) = 'G'
C
C     H
        CHEMPO(6) = 9.480526D4
        PHASE(6) = 'G'
C
C     HO
        CHEMPO(7) = 6.953808D3
        PHASE(7) = 'G'
C
C     O
        CHEMPO(8) = 1.0829865D5
        PHASE(8) = 'G'
C
C     NO
        CHEMPO(9) = 6.251314D4
        PHASE(9) = 'G'
C
C     O2
        CHEMPO(10) = 0.
        PHASE(10) = 'G'
C
C     C
        CHEMPO(11) = 0.
        PHASE(11) = 'S'
C
      RETURN
      END
```