TECHNICAL REPORT NO. 643

ANALYSIS AND DESCRIPTION

OF AN IBM 7090/94 PROGRAM

TO COMPUTE EQUILIBRIUM CONDITIONS

FOR GASEOUS CHEMISTRY SYSTEMS

By Herbert H. Hopf

December 14, 1966
TECHNICAL REPORT NO. 643

ANALYSIS AND DESCRIPTION OF AN IBM 7090/94 PROGRAM

TO COMPUTE EQUILIBRIUM CONDITIONS FOR

GASEOUS CHEMISTRY SYSTEMS

* By Herbert H. Hopf

Prepared for

Advanced Research Projects Agency
Washington, D. C.

ARPA Order No. 396
Date of Contract: December 3, 1965
Contract Number DA-49-083 OSA-3135
Contract Expiration Date: January 1, 1967
Project Scientist: Dr. Hwachii Lien
516-333-6960
'Missile Phenomenology Studies"

Prepared by

General Applied Science Laboratories, Inc.
A Subsidiary of The Marquardt Corporation
Merrick and Stewart Avenues
Westbury, L. I., New York

*Sponsored by the Advanced Research
Projects Agency, Missile Phenomenology
Branch, ARPA Order No. 396

December 14, 1966
Approved by: Lewis Feldman, Manager
Applied Research Dept.
Mole Fractions vs. Temperature for a 28 Species Teflon-Air System (50% Teflon-Air by Volume) P = 0.264 Atmospheres
ABSTRACT

A general computer program for the calculation of the chemical equilibrium composition of a gaseous system was written for the IBM 7090/94. The program is based on the minimization of the Gibbs free energy of the system; the resulting non-linear equations are solved by a modified Newton-Raphson iteration scheme. The three options presently available for the two intensive variables necessary for the calculation of the equilibrium composition are pressure-temperature, pressure-enthalpy and pressure-entropy, although other options for any two intensive variables may be readily added.

A "program generator" that produces the source program cards in FORTRAN IV of the equilibrium program for a specific chemistry system was also written. In the process of generating the source program, the generator extracts from a library magnetic tape the necessary curve fits for the species enthalpy and entropy as functions of temperature for the specified chemistry system. This resulting source program may be utilized as a subroutine to serve the needs of each particular application.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>ANALYSIS</td>
<td>4</td>
</tr>
<tr>
<td>III</td>
<td>METHOD OF SOLUTION</td>
<td>13</td>
</tr>
<tr>
<td>IV</td>
<td>SOME REMARKS ABOUT CONVERGENCE</td>
<td>18</td>
</tr>
<tr>
<td>V</td>
<td>OTHER THERMODYNAMIC OPTIONS</td>
<td>20</td>
</tr>
<tr>
<td>VI</td>
<td>INSTRUCTIONS FOR USING THE GENERATED EQUILIBRIUM PROGRAM</td>
<td>22</td>
</tr>
<tr>
<td>VII</td>
<td>DESCRIPTION OF THE EQUILIBRIUM PROGRAM GENERATOR</td>
<td>28</td>
</tr>
<tr>
<td>VIII</td>
<td>SAMPLE APPLICATION</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>Appendix A</td>
<td>A-1/5</td>
</tr>
<tr>
<td></td>
<td>Appendix B</td>
<td>B-1/7</td>
</tr>
<tr>
<td></td>
<td>Appendix C</td>
<td>C-1</td>
</tr>
<tr>
<td></td>
<td>Appendix D</td>
<td>D-1/3</td>
</tr>
</tbody>
</table>
NOMENCLATURE

\( a_{ij} \)  
the number of atoms of element \( j \) in one molecule of species \( i \)

\( A \)  
the left-hand side of the element conservation matrix [Eq. (6)] consisting of the matrix elements \( a_{ij} \)

\( A_1 \)  
partitioned \( A \) matrix consisting of \( N_P \) rows and \( N_P \) columns

\( A_2 \)  
partitioned \( A \) matrix consisting of \((N_S - N_P)\) rows \( N_P \) columns

\( b_j \)  
element conservation constants from Eq. (5)

\( C_i \)  
\[ \frac{1}{R} \left[ \frac{h_i}{T} - s_i \right] + \ln P \] [Eq. (9)]

\( D_F \)  
the Jacobian \((\partial F_j / \partial \vec{n} X_i)\) defined by Eq. (31)

\( F_j \)  
residual vector defined by Eq. (30)

\( G \)  
Gibbs free energy, cal

\( g_i \)  
\[ - \left( \frac{1}{R} \left[ \frac{h_i}{T} - s_i \right] + \ln P + \ln X_i \right) \] [see Eq. (16)]

\( \bar{G} \)  
the row vector consisting of the \( g_i \) [see Eq. (18)]

\( \bar{G}_i \)  
the partitioned row vector \( \bar{G} \) consisting of \( N_P \) vector elements

\( \bar{G}_2 \)  
the partitioned row vector \( \bar{G} \) consisting of \((N_S - N_P)\) vector elements

\( H \)  
mixture enthalpy, \( \text{ft}^2/\text{sec}^2 \)

\( H \)  
mixture enthalpy, cal

\( h_i \)  
species enthalpy, cal/mole
\( h_i \) \hspace{1cm} \text{species enthalpy, ft}^2/\text{sec}^0 \\
NP \hspace{1cm} \text{number of elements (including charge, if applicable) of the chemistry system considered.} \\
NSP \hspace{1cm} \text{total number of species in the system} \\
P \hspace{1cm} \text{pressure, atmospheres} \\
R \hspace{1cm} \text{universal gas constant 1.98726 cal/mole } ^0\text{K or 89506 ft}^2/\text{sec}^2-^0\text{K lb/lb mole} \\
\hat{S} \hspace{1cm} \text{mixture entropy lb mole/lb} \\
S \hspace{1cm} \text{mixture entropy, cal/}^0\text{K} \\
\bar{S} \hspace{1cm} \text{mixture entropy, ft}^2/\text{sec}^2-^0\text{K} \\
s_i \hspace{1cm} \text{species entropy at standard state } (p = 1 \text{ atm}), \text{ cal/mole } ^0\text{K} \\
\bar{s}_i \hspace{1cm} \text{species entropy at standard state } (p = 1 \text{ atm}), \text{ ft}^2/\text{sec}^2-^0\text{K} \\
T \hspace{1cm} \text{temperature, } ^0\text{K} \\
X_i \hspace{1cm} \text{mole fraction of species } i \\
Y_i \hspace{1cm} \text{number of moles of species } i \\
\bar{Y} \hspace{1cm} \text{the total moles in the system } (\bar{Y} = \sum_{i=1}^{\text{NSP}} Y_i) \\
\text{Greek} \\
\Phi \hspace{1cm} \frac{G}{RT} + \sum_{j=1}^{\text{NP}} \left[ \sum_{i=1}^{\text{NP}} Y_i a_{ij} - b_j \right] \pi_j \text{ [see Eq. (8)]} \\
\pi_j \hspace{1cm} \text{Lagrange multipliers [see Eq. (8)]} \\
\beta_{ij} \hspace{1cm} \text{matrix elements resulting from the matrix multiplication } A_0 A_i^{-1} \text{ [see Eq. (26)]}
the column vector consisting of the modified Newton-Raphson correction terms [Eqs. (39)]

\[ \delta_{ij} \]  
Kronecker delta
TECHNICAL REPORT NO. 643

ANALYSIS AND DESCRIPTION OF AN IBM 7090/94 PROGRAM

TO COMPUTE EQUILIBRIUM CONDITIONS FOR

GASEOUS CHEMISTRY SYSTEMS

By Herbert H. Hopf

I. INTRODUCTION

The problem of computing the equilibrium composition of a gas has been the subject of many papers. An extensive search of the literature has revealed that two approaches have been used:

1. Utilization of equilibrium constants associated with the pertinent chemical reactions.


The equilibrium constant approach is the most common for small systems, where the system of equations is comprised of the element conservation equations and the laws of mass action. Where only a few species are involved, a set of reactions can be easily formulated, for which curve fits of the corresponding equilibrium constants as functions of temperature can be obtained. In recent years computer programs were written to
formulate the reactions for large chemistry systems. Curve fits of the species Gibbs Free Energy as a function of temperature are obtained, and the equilibrium constant is computed from the change in the Gibbs Free Energy across each reaction at a reference state.

For the second approach, the Gibbs Free Energy of the entire system is minimized subject to the constraint of charge and element conservation. A variety of techniques have been developed to solve the resultant system of non-linear equations; the numerical technique described in this report is a modified Newton Raphson iteration scheme.

The objectives of the effort described herein are two-fold:

1. Create an equilibrium program which may be used as a subroutine to generate the chemical equilibrium composition and desired thermodynamic properties for any specified gaseous chemistry system.

2. Write a "program generator" which automatically produces the source cards for this equilibrium program according to the chosen species, which are input for the program generator. This program generator
extracts the necessary thermodynamic data from a library magnetic tape.

Thus, a programming system has been created which provides the user with an analytical tool which may be easily linked to computer programs designed to solve a wide variety of fluid dynamics problems. The equilibrium program generator has produced equilibrium programs which have been successfully used for a variety of applications.

A typical application of such a generated program was for the computation of the equilibrium composition of an Air-Teflon mixture through a boundary layer adjacent to an ablating surface. A detailed description of a sample application is shown in Section VIII.
II. ANALYSIS

The Gibbs free energy of a mixture is defined as

$$ G = H - TS $$

where $H$ is the mixture enthalpy in calories

$S$ is the mixture entropy in calories/K

$T$ is the temperature in °K.

The enthalpy of a mixture of thermally perfect gas can be computed from the species enthalpy for a given temperature using the following equation

$$ H = \sum Y_i h_i(T) $$

where $Y_i$ is the number of moles of specie $i$, $h_i(T)$ is the enthalpy of specie $i$ in calories per mole.

The entropy of a mixture of thermally perfect gases can be computed from the species entropy at a given temperature and pressure using the following equation

$$ S - S_o = \sum Y_i [s_i(T) - R \ln X_i - R \ln P] $$

where $P$ is the mixture pressure in atmospheres

$R$ is the universal gas constant in cal./mole °K
$$s_i(T) \text{ is the entropy of specie } i, \text{ cal.}/\text{mole}^\circ\text{K}$$

$$X_i \text{ is the mole fraction of species } i$$

$$S_0 \text{ is the reference entropy taken as zero at } T = 0^\circ\text{K}, p = 1\text{ atm.}$$

The second term on the right-hand side of Eq. (3) is the contribution to the entropy due to the diffusion of the species.

The third term is the contribution to the entropy for an isothermal process where the pressure is changed from one atmosphere to a pressure $P$.

Substituting Eqs. (2) and (3) into Eq. (1),

$$\frac{G}{RT} = \sum_{i=1}^{\text{NSP}} Y_i \left\{ \frac{1}{R} \left[ \frac{h_i}{T} - s_i \right] + \ln X_i + \ln P \right\}$$

(4)

where $\text{NSP}$ is the total number of species in the system.

The determination of the equilibrium composition is equivalent to finding the set of $X_i$'s which minimizes (4), subject to the constraint of the element conservation equations

$$\sum_{i=1}^{\text{NSP}} Y_i a_{ij} = b_j \quad j=1,2,...,\text{NP}$$

(5)
where \( a_{ij} \) is the number of atoms of element \( j \) in one molecule of specie \( i \), \( b_j \) is the total number of atoms of element \( j \), and \( NP \) is the number of elements in the system.

Written in matrix form Eq. (5) becomes

\[
\begin{bmatrix}
Y_1, Y_2, \ldots, Y_{NSP}
\end{bmatrix}
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1, NP} \\
a_{21} & a_{22} & \cdots & a_{2, NP} \\
\vdots & \vdots & & \vdots \\
a_{NSP,1} & a_{NSP,2} & \cdots & a_{NSP, NP}
\end{bmatrix}
= (b_1, b_2, \ldots, b_{NP})
\]  

One additional equation which must be satisfied by the \( X_i \)'s is that

\[
\sum_{i=1}^{NSP} X_i = 1
\]  

Eq. (4) is minimized subject to the constraint given by Eq. (5) using Lagrange multipliers. Let

\[
\Phi = \frac{G}{RT} + \sum_{j=1}^{NP} \left\{ \sum_{i=1}^{NSP} Y_i a_{ij} - b_j \right\} \pi_j
\]
where $\pi_j$ are the Lagrange multipliers.

Let

$$C_i = \frac{1}{R} \left( \frac{h_i}{T} - s_i \right) + \ln P$$  \hspace{1cm} (9)

Then from Eq. (4) and (9)

$$\frac{G}{RT} = \sum_{i=1}^{\text{NSP}} Y_i \left[ C_i + \ln \left( \frac{Y_i}{\overline{Y}} \right) \right]$$  \hspace{1cm} (10)

and

$$\frac{\partial G/RT}{\partial Y_k} = \left[ C_k + \ln \left( \frac{Y_k}{\overline{Y}} \right) \right] + \sum_{i=1}^{\text{NSP}} Y_i \frac{\partial}{\partial Y_k} \left[ C_i + \ln \left( \frac{Y_i}{\overline{Y}} \right) \right]$$  \hspace{1cm} (11)

$$k = 1, 2, 3, \ldots, \text{NSP}$$

where $\overline{Y} = \sum_{i=1}^{\text{NSP}} Y_i$.

Consider the second term on the right-hand side of Eq. (11) and expand it as follows [dropping $C_k$, since $C_k = C_k(P, T)$ is independent of $Y_k$]:

$$Y_1 \frac{\partial}{\partial Y_k} \left( \ln \frac{Y_1}{\overline{Y}} \right) + Y_2 \frac{\partial}{\partial Y_k} \left( \ln \frac{Y_2}{\overline{Y}} \right) + \cdots + Y_{\text{NSP}} \frac{\partial}{\partial Y_k} \left( \ln \frac{Y_{\text{NSP}}}{\overline{Y}} \right)$$  \hspace{1cm} (12)
Note that:

\[
Y_t \frac{\partial}{\partial Y_k} \left( \ln \frac{Y_t}{Y} \right) = \frac{Y_t - Y_i}{Y} \quad \text{for } t = k
\]

\[
= -\frac{Y_i}{Y} \quad \text{for } t \neq k
\]  

(13)

From which it can be seen that

\[
\sum_i Y_i \frac{\partial}{\partial Y_k} \left[ C_k + \ln \left( \frac{Y_i}{Y} \right) \right] = 0
\]

(14)

If Eq. (14) is substituted into Eq. (11) and this result is substituted into the expression for the partial derivative of \( \Phi \) [as defined by Eq. (8)] with respect to \( Y_i \), then

\[
\frac{\partial \Phi}{\partial Y_i} = C_i + \ln X_i + \sum_{j=1}^{NP} a_{ij} \pi_j
\]

(15)

It can be shown that the solution \( \frac{\partial \Phi}{\partial Y_i} = 0 \) represents the minimum of the Gibbs free energy, which corresponds to the unique solution where the system is in chemical equilibrium (see Ref. 1).

There are NP element conservation equations given by (5) and NSP equations obtained through the minimization process.
given by (15). The unknowns in this system of equations are NSP species mole fractions and NP Lagrange multipliers.

The system of (NSP + NP) equations with the same number of unknowns will be reduced to a system of NSP equations with NSP unknowns.

Let

\[ g_i = C_i - \ln X_i = \left\{ \frac{1}{R} \left[ \frac{h_i}{T} - s_i \right] \right\} + \ln P + \ln X_i \]

then Eq. (15) can be written as

\[ \sum_{j=1}^{NP} a_{ij} \pi_j = g_i \quad i=1,2,\ldots,NSP \quad (17) \]

for the condition of the minimum Gibbs free energy (i.e. \( \frac{\partial \Phi}{\partial Y_i} = 0 \)).

If Eq. (17) is written in matrix form the result is

\[ \begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1,\text{NP}} \\
    a_{21} & a_{22} & \cdots & a_{2,\text{NP}} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{\text{NSP,1}} & a_{\text{NSP,2}} & \cdots & a_{\text{NSP,\text{NP}}} \\
\end{bmatrix}
\begin{bmatrix}
    \pi_1 \\
    \pi_2 \\
    \vdots \\
    \pi_{\text{NP}} \\
\end{bmatrix}
= \begin{bmatrix}
    g_1 \\
    g_2 \\
    \vdots \\
    g_{\text{NP}} \\
\end{bmatrix} \quad (18)\]
The set of equations represented by Eq. (18) can be written as one set of NP equations and another set of (NSP-NP) equations. If Eq. (19) is partitioned accordingly, the result is

\[
\begin{align*}
\mathbf{A} \mathbf{\Pi} &= \mathbf{G} \\
\mathbf{A}_1 \mathbf{\Pi} &= \mathbf{G}_1 \\
\mathbf{A}_2 \mathbf{\Pi} &= \mathbf{G}_2
\end{align*}
\]

Solving Eq. (21) for \( \mathbf{\Pi} = \mathbf{A}_1^{-1} \mathbf{G}_1 \) and substituting into Eq. (22), one obtains:

\[
\mathbf{A}_2 \mathbf{A}_1^{-1} \mathbf{G}_1 = \mathbf{G}_2
\]

If the expression for \( g_i \) given by (16) is substituted into (23) the result is

\[
\ln X_n = \sum_{j=1}^{NP} \left\{ \beta_{n-NP,j} (c_j + \ln X_j) \right\} - C_n
\]

\[
n = NP + 1, NP + 2, \ldots, NSP
\]
where

\[ C_i = -g_i - \ln X_i = \frac{1}{R} \left[ \frac{h_i}{T} - s_i \right] + \ln P \]  \hspace{1cm} (25)

\[
\begin{bmatrix}
\beta_{11} & \beta_{12} & \beta_{13} & \cdots & \beta_{1,NP} \\
\beta_{21} & \beta_{22} & \beta_{23} & \cdots & \beta_{2,NP} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\beta_{NP-NP,1} & \beta_{NP-NP,2} & \beta_{NP-NP,3} & \cdots & \beta_{NP-NP,NP}
\end{bmatrix} = [A_2 A^{-1}_1]
\]  \hspace{1cm} (26)

Eqs. (5) and (24) correspond to NSP equations in the NSP unknowns (mole fractions). In addition Eq. (7) can be used as a check equation. However, since the total number of moles in the system, may be of interest, this parameter is added to the set of unknowns and Eq. (7) is added to the system of equations.

This is accomplished by writing Eq. (5) in the following form

\[
\sum_{i=1}^{NSP} X_i a_{ij} = \frac{b_j}{Y} \hspace{1cm} j=1,\ldots,NP \]  \hspace{1cm} (27)
where \( \bar{Y} \) is the total number of moles in the system, i.e.

\[
\bar{Y} = \sum_{i=1}^{NSP} Y_i
\]  \hspace{1cm} (23)

\[
X_i = \frac{Y_i}{\bar{Y}}
\]  \hspace{1cm} (29)

The non-linear system of equations given by (7), (24) and (27) in the unknowns \( \bar{Y} \) and \( X_i \) \( (i=1,2,\ldots,NSP) \) is solved by using a modified Newton-Raphson technique.
III. METHOD OF SOLUTION

A residual vector, represented by $F_j$, is obtained from Eqs. (7) and (27) as follows:

$$
F_1 = \sum_{i=1}^{\text{NSP}} X_i - 1
$$

$$
F_j = \sum_{i=1}^{\text{NSP}} X_i a_{ij} - \frac{b_j}{Y}
$$

(30)

$j = 2, 3, \ldots, \text{NP}$

The Jacobian to be used in the Newton Raphson iterative procedure for $\text{NP}$ dimensions, $DF = (\frac{\partial F_j}{\partial \ln X_i})$ for $(1 \leq j \leq \text{NP})$, $(1 \leq i \leq \text{NP})$ is formed as follows:

$$
DF = 
\begin{bmatrix}
\frac{\partial F_1}{\partial \ln X_1} & \frac{\partial F_1}{\partial \ln X_2} & \ldots & \frac{\partial F_1}{\partial \ln X_{\text{NP}}} \\
\frac{\partial F_2}{\partial \ln X_1} & \frac{\partial F_2}{\partial \ln X_2} & \ldots & \frac{\partial F_2}{\partial \ln X_{\text{NP}}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_{\text{NP}}}{\partial \ln X_1} & \frac{\partial F_{\text{NP}}}{\partial \ln X_2} & \ldots & \frac{\partial F_{\text{NP}}}{\partial \ln X_{\text{NP}}}
\end{bmatrix}
$$

(31)
The following is required for the evaluation of the various terms \( \frac{\partial F_j}{\partial \ln X_t} \):

\[
\begin{align*}
\frac{\partial \ln X_j}{\partial \ln X_t} &= \delta_{jt} & (1 \leq j \leq NP) \\
\frac{\partial X_j}{\partial \ln X_t} &= x_j \delta_{jt} & (1 \leq t \leq NP)
\end{align*}
\]

(32)

where \( \delta_{jt} \) is the Kronecker delta.

\[
\delta_{jt} = \begin{cases} 
1 & \text{for } j = t \\
0 & \text{for } j \neq t
\end{cases}
\]

(33)

From Eq. (24) for \((NP + 1 \leq n \leq NSP), (1 \leq t \leq NP)\)

\[
\frac{\partial \ln X_n}{\partial \ln X_t} = \frac{\partial}{\partial \ln X_t} \left( \sum_{k=1}^{NP} (C_k + \ln X_k) \beta_{n-NP,k} - C_n \right) = \beta_{n-NP,t}
\]

and

\[
\frac{\partial X_n}{\partial \ln X_t} = x_n \beta_{n-NP,t}
\]

(34)

since \( C_i = C_i(P,T) \).

From Eq. (27)

\[
\frac{1}{Y} = \frac{1}{b_1} \sum_{i=1}^{NSP} x_i a_{ij}^{ij'}
\]

(35)
then

\[
\frac{\partial \left[ \frac{1}{Y} \right]}{\partial \ln X_{\ell}} = \frac{1}{b_1} \left[ X_{\ell} \alpha_{\ell 1} + \sum_{n=NP+1}^{NSP} X_{\ell} \alpha_n \beta_{n-NP, \ell} \right].
\] (36)

It should be noted that Eqs. (35) and (36) are valid only for \( b_1 \neq 0 \).

The partial derivatives that are required for DF as defined in Eq. (31) can now be evaluated as follows:

\[
\frac{\partial F_j}{\partial \ln X_{\ell}} = X_{\ell} \alpha_j + \sum_{n=NP+1}^{NSP} \left[ X_{\ell} \alpha_{nj} \beta_{n-NP, \ell} - \frac{\partial \left[ \frac{1}{Y} \right]}{\partial \ln X_{\ell}} \right] - b_j \frac{\partial \left[ \frac{1}{Y} \right]}{\partial \ln X_{\ell}}
\]

\[
\begin{align*}
\text{for } j &= 2, 3, \ldots, NP \\
\text{and } \ell &= 1, 2, \ldots, NP
\end{align*}
\] (37)

If the residual vector given by Eq. (30) is denoted by the column vector

\[
\mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{NP} \end{bmatrix}
\] (38)
and the set of correction terms denoted by the column vector

\[ \Delta = \begin{bmatrix}
\Delta \ln x_1 \\
\Delta \ln x_2 \\
\vdots \\
\Delta \ln x_N
\end{bmatrix} \tag{39} \]

and \( DF \) by Eq. (31), then in accordance with the Newton-Raphson procedure these correction terms are found by solving the following set of simultaneous linear equations

\[ (DF) \cdot (\Delta) = F \tag{40} \]

The next set of iterated values of the mole fractions of the "prime" species is given by the expression

\[ (\ln x_j)_N+1 = (\ln x_j)_N - \lambda (\Delta \ln x_j)_N \tag{41} \]

\[ j=1, \ldots, NP \]

where \( N \) denotes the iteration number and \( \lambda \) is a relaxation factor (\( 0 < \lambda \leq 1 \)).

The relaxation factor is chosen by the program so that it is as large as possible (but never larger than unity) and such that \( \max \{|F_j|\} \) decreases from iteration to iteration.
Initially $\lambda$ assumes a value of unity, but is subsequently reduced, if necessary, until $\max \{ F_j \}_N$ is smaller than $\max \{ F_j \}_{N-1}$. The relaxation factor $\lambda$ has been introduced to keep the iterations in the neighborhood of the solution whenever possible. If the solution is converging and $\lambda < 1$, then $\lambda$ is increased subject to the above constraint to accelerate convergence.

The mole fractions for $(NP + 1 \leq n \leq NSP)$ are obtained from Eq. (24), and the total moles in the system from (35).

The criteria for convergence of the iteration procedure are as follows:

1. $\left| F_j \right| < 10^{-8}, \quad j = 1, 2, \ldots, NP$

where the $F_j$ are defined by Eq. (30), and

2. $\left| \frac{(\ln X_i)^N - (\ln X_i)^{N-1}}{(\ln X_i)^N} \right| < 10^{-4}$

$i = 1, 2, 3, \ldots, NSP$

where $N$ is the iteration counter.

The first test ensures that the major species have converged, and the second test applies to all the species in the system including the trace species.
IV. SOME REMARKS ABOUT CONVERGENCE

Mathematical analysis (s.e Ref. 1) has shown that the Gibbs free energy function is strictly convex; hence, setting the derivatives of Gibbs free energy to zero results in a unique solution. However, since the system of equations cannot be solved in closed form, the question arises as to under what conditions will the program encounter convergence problems.

It is suspected that the majority of convergence problems will be attributable to either

(1) the estimates of the mole fractions $X_j$, $j=1, 2, \ldots, NP$, are too far from the solution,

(2) the species have not been properly ordered.

The user must consider the ordering of species when using the "program generator." The first NP species specified must be linearly independent. (Note that charge conservation is handled exactly like element conservation.) For example, if a seven species air ($N_2$, $N$, $O_3$, $O$, $NO$, $NO^+$, $e^-$) chemistry system is considered, $NP = 3$ (the three "elements" are $N$, $O$, $e^-$), $NSP = 7$ (there are 7 species) and the ordering of the first three species could be $N_2$, $O_3$, $e^-$ or $NO^+$, $NO$, $O_2$, as well as a number of other combinations. However, the choice of $N_2$,
N, e\(^-\) as the first three are incorrect, since \(N_2\) and \(N\) are not linearly independent.

The first NP species must be chosen such that each element (and charge, if applicable) be included among these species. This is a necessary condition for the system to be linearly independent; however, it is not sufficient to guarantee linear independence. Consider again the seven species air case, and let the first NP species be \(e^-\), NO and \(NO^+\). If one forms the \(A_1\) matrix as described in (20), it is apparent that this system is linearly dependent, since subtracting the second row from the first results in the third row. However, the rule described above will work in the large majority of chemistry systems.

Furthermore the program will work most efficiently if the first NP species, besides being linearly independent, are chosen so that their mole fractions are as large as possible. The user should also note that the species cannot be ordered such that electrons is the first species, since \(\bar{Y}\) (total moles) becomes singular as computed from Eq. (35).
V. OTHER THERMODYNAMIC OPTIONS

As described in the previous sections the program computes the equilibrium composition of a gaseous mixture for specified values of pressure and temperature. Other input options available are pressure-enthalpy and pressure-entropy. For these options, the program performs a series of pressure temperature calculations in an iterative manner until the specified values of enthalpy or entropy are satisfied within prescribed tolerances. Estimates of the corresponding values of temperature and species mole fractions must be transmitted to the equilibrium program through the calling sequence.

Utilizing the specified value of pressure, and the estimated values of temperature and the first NP species mole fractions, the program computes the corresponding equilibrium values of the species mole fractions \( X_i \), and of enthalpy, \( \bar{H}_1 \) (or entropy \( \bar{S}_1 \)). A second guess of the temperature is obtained (for enthalpy):

\[
T_2 = T_1 \pm 0.05 T_1 \quad \text{for} \quad \bar{H}_1 \leq \bar{H}
\]

where \( \bar{H} \) is the specified mixture enthalpy per unit mass, \( \text{ft}^2/\text{sec}^2 \)

\( \hat{S} \) is the entropy, \( \bar{S} = \frac{1}{K} \text{lb mole} \)

\( \bar{S} \) is the entropy per unit mass, \( \text{ft}^2/\text{sec}^2 - K \).
The program then calculates $\bar{H}_a$ corresponding to the specified pressure and $T_a$. The third estimate of temperature is calculated by the application of the "regula falsi" method (or method of chords), as follows:

$$T_3 = T_2 - \frac{f_2}{\left| \frac{df}{dT} \right|_{T_2}}$$

(43)

where $\left| \frac{df}{dT} \right|_{T_2} = \frac{f_2 - f_1}{T_2 - T_1}$

$$f_k = \bar{H}_k - \bar{H}.$$

All subsequent estimates of $T$ are calculated by expressing $f_k$ as a function of $T_k$ by means of a LaGrange second-order equation and evaluating this parabola at $f = 0$, to obtain the next estimate of $T_{k+1}$, where $k$ denotes the enthalpy iteration counter.

Convergence is assumed when $\left| \frac{\bar{H}_k - \bar{H}}{\bar{H}} \right| \leq 10^{-6}$

This procedure, of course, applies also to the pressure-entropy option. The program logic may be readily extended to include input options for any two intensive properties.
VI. **INSTRUCTIONS FOR USING THE GENERATED EQUILIBRIUM PROGRAM**

**Initializing Call Statement**

CALL GENL7X(X,PROP,PP, T, P, KOPT, BEE)

This CALL statement initializes the addresses of the formal parameters in the calling sequence and computes the element conservation constants $b_j$ described by Eq. (27), utilizing the values of the mole fractions stored in the X array, upon entry. The value of KOPT must be a positive integer. These values of the mole fractions must truly represent the gaseous system at some reference state; they cannot be approximations (or guesses) of the $X_i$.

The user also has the option of directly storing the element mole fractions in the BEE array rather than specifying the $X_i$'s. The option of specifying the values directly in the BEE array is particularly useful for applications where the element mole fractions are not constant, such as the case of a boundary layer adjacent to an ablating body. When the contents of the BEE array are specified directly, KOPT must be a negative integer upon entry. All subsequent calls (see below) must then specify KOPT < 0, and the BEE array must contain the current values of $b_j$ upon entry.
The formal parameters in the calling sequence are described below:

X is a one-dimensional array which contains the values of the mole fractions. The size of the array is equal to one more than the number of species in the system (NSP+1).

T is the temperature in °K.

P is the pressure in atmospheres.

KOPT is a code word which specifies the input option (see below).

BEE is a one-dimensional array, whose size is equal to the number of elements, NP, which contains the element conservation constants $b_j$ described by Eq. (27).

PROC and PR2 are dummy variables whose contents are identified by the code word, KOPT (see below).

**General Call Statement**

CALL GENR7X

This CALL statement may be executed repeatedly, once the initializing CALL statement has been executed. Note that the equilibrium subroutine operates upon those formal parameters which appear as arguments in the initializing CALL statement. Hence these arguments must contain the correct values of the input parameters when this general call statement is executed. It is...
suggested that these formal parameters ($X$, $PROP$, $PR2$, $T$, $P$, $KOPT$ and $BEE$) be included in COMMON storage so that transmission of their contents among all routines may be satisfactorily achieved.

**Pressure-Temperature (P-T) Option:** $KOPT = \pm 1$

Guesses of the first $NP$ species mole fractions ($X_j$, $j=1,2,...,NP$) must be specified in the $X$ array upon entry.

**Pressure-Enthalpy (P-H) Option:** $KOPT = \pm 2$

In addition to these guesses of $X_j$, $j=1,2,...,NP$, a guess of temperature must be specified in the $T$ parameter upon entry.

**Pressure-Entropy (P-S) Option:** $KOPT = \pm 3$

Same inputs as for the P-H option, above.

$KOPT = \pm 1$ for pressure-temperature input option

$\pm 2$ for pressure-enthalpy input option

$\pm 3$ for pressure-entropy input option.

If $KOPT > 0$ the contents of the $BEE$ array (values of $b_j$) are not altered.

If $KOPT < 0$ the user must specify the values of $b_j$ in the array $BEE$ upon entry.
1. The contents of PROP and PR2 are described in the table below. For KOPT = ± 1, the contents of PROP and PR2 are output properties; for KOPT = ± 2 and ± 3, the content of PROP is a specified input and the content of PR2 is an output property.

<table>
<thead>
<tr>
<th>KOPT</th>
<th>PROP</th>
<th>PR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>± 1</td>
<td>enthalpy, ft²/sec²</td>
<td>entropy, lb moles/lb</td>
</tr>
<tr>
<td>± 2</td>
<td>enthalpy, ft²/sec²</td>
<td>entropy, lb moles/lb</td>
</tr>
<tr>
<td>± 3</td>
<td>entropy, lb moles/lb</td>
<td>enthalpy, ft²/sec²</td>
</tr>
</tbody>
</table>

There are four situations under which the equilibrium program will call an error subroutine, and they are:

1. The temperature at which the species enthalpy and species entropy are to be evaluated from the curve fits, as shown in Eqs. (44), is higher than the upper bound of temperature range over which the fit is valid (no check is made for the lower bound of the temperature).

2. If more than ten iterations are required to satisfy the prescribed value enthalpy (or entropy) within a relative error of $10^{-6}$, when the P-H (or P-S) options are used.

3. If more than 30 iterations are required to converge to a solution for a P-T problem.
4. If between two successive iterations for a P-T problem, \( \max \left\{ |F_j| \right\} \) as defined by Eq. (30) does not decrease, the value of the relaxation factor \( \lambda \) used in Eq. (41) will be reduced; this reduction of \( \lambda \) will be repeated until \( \max \left\{ |F_j| \right\} \) decreases from iteration to iteration. However, the program calls the error subroutine if \( \lambda \) has been reduced for 10 consecutive P-T iterations.

For the first three situations it will probably be desirable to let the program continue after writing an error message. If the last situation occurs, an error message should be output after which the execution should be terminated. The run should be resubmitted, either with "better" guesses for the mole fractions, or possibly the species may have to be re-ordered.

A listing of the error subroutine which handles these situations as stated above appears in Appendix C.

Included in the equilibrium package is a subroutine that will solve a system of \( N \) simultaneous equations. The method employed is pivotal condensation, where the largest pivotal element available is utilized.

The user may CALL this subroutine in another part of his program by the following statement:
CALL CLEM (Al, DLNX, F, N, N+1, AT)

where the formal parameters are defined as follows:

Al is the matrix of coefficient
DLNX is the solution of vector
F is the forcing vector
N is the number of equations to be solved
AT is a working array used by the subroutine, but
must be dimensioned by the calling program to be
of size (N, N+i).

Streamline Calculation Using Equilibrium P-S Option

To perform calculations along streamlines using the P-S option of the equilibrium chemistry program, it is first necessary to execute either the P-T option or the P-H option at the initial point on the streamline in order to extract the value of entropy $\hat{S}$. All subsequent calculations along the streamline may then be executed using the P-S option where $\hat{S}$ is the value extracted at the initial point. This value of $\hat{S}$ is conserved along each streamline. To calculate the dimensionless value of entropy it is necessary to multiply $\hat{S}$ by the mixture molecular weight.
VII. DESCRIPTION OF THE EQUILIBRIUM PROGRAM GENERATOR

For each chemistry system used the user must first generate the FORTRAN IV source cards for the specific chemistry system requested. The generator program punches the source cards of the equilibrium subroutine, including all the required internal data, and the elements of the A matrix defined by Eq. (6). The curve fits are of the form:

\[
\overline{c}_{p_{i}} = a_{i} + b_{i}T
\]

\[
\overline{h}_{i} = \overline{h}_{i0} + \int_{0}^{T} c_{p_{i}} dT = a_{i}T + \frac{b_{i}T^{2}}{2} + d_{i}
\]

\[
\overline{s}_{i} = \overline{s}_{i0} + \int_{T_{0i}}^{T} \frac{c_{p_{i}}}{T} dT = a_{i} \ln T + b_{i}T + c_{i}
\]

Where
- \(\overline{c}_{p_{i}}\) is the species specific heat at constant pressure, \(ft^{2}/sec^{2} \cdot ^{\circ}K\)
- \(\overline{h}_{i}\) is the species enthalpy, \(ft^{2}/sec^{2}\) (note that species sensible enthalpy is taken to be zero at 0 \(^{\circ}K\))
- \(\overline{h}_{i0}\) is the species heat of formation at 0 \(^{\circ}K\), \(ft^{2}/sec^{2}\)
- \(\overline{s}_{i}\) is the species entropy, \(ft^{2}/sec^{2} \cdot ^{\circ}K\) (note \(\overline{s}_{i}\) is taken to be zero at 0 \(^{\circ}K\), \(p = 1\) atmosphere)
- \(T_{0i}\) is the lower bound of temperature for the curve fit of \(c_{p_{i}}\), \(^{\circ}K\).
The data for $c_{p_i}$ have been fit (in the least square sense) as a function of temperature, with first order equations; the curve fits are subdivided into several temperature ranges. The only constraint on these fits for the equilibrium program is that the temperature ranges for each fit ($c_{p_i}$, $\bar{h}_i$, $\bar{s}_i$) of a given species coincide. The curve fits now contained on the library tape used by the program generator satisfy this constraint. Presently a maximum of 6 temperature ranges per species is accommodated.
**Input Format to the Program Generator**

<table>
<thead>
<tr>
<th>CARD NO.</th>
<th>COLUMNS</th>
<th>DATA TO BE PUNCHED ON CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-8</td>
<td>Date (month/day/year)</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>43-45</td>
<td>(right adjusted) number of species in system</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4,5,6,...</td>
<td>6A12</td>
<td>The species symbols consistent with GASL library tape, with 12 columns per field for each species. The first NP (where NP is the total number of elements and charges, if applicable) species must be linearly independent, and the first of these species may not be electrons since Eq. (35) would become singular.</td>
</tr>
</tbody>
</table>

See Appendix D for sample inputs.
Operating Instructions for the Program Generator

When running the "program generator" on a 7094 IBM computer, the following operating instructions apply:

1. IBSYS system version 13 is to be used.

2. Mount a specified GASL Chemistry Library tape on unit A5.

3. Scratch tapes are to be mounted on units B5 and B6.

4. One file from output tape unit A6 is to be punched on FORTRAN cards; interpret columns 1-60. These cards are the source deck of the equilibrium program.

5. List under program control the output from unit B1.
Output Obtained from the Equilibrium Program Generator

The generator will output (on paper) a report detailing the coefficients of the least-square curve fits of the species enthalpy and species entropy, as functions of temperature as well as the upper and lower bounds of the temperature regions corresponding to each curve fit. Since the equilibrium program does not check for the lowest bound on temperature (i.e. it will automatically extrapolate to a temperature below the lowest bound), the user should check this output if he expects the program to encounter temperatures below the lowest bound, which for most species presently accommodated, is 200°K.

In addition, the source deck of the equilibrium subroutine, in FORTRAN IV, for the specified chemistry system is punched as output, a listing of which appears in the appendix. The user must add subroutine ERROR, as described in Appendix C.
VIII. SAMPLE APPLICATION

A typical application of a generated equilibrium program was for the computation of the equilibrium composition of an Air-Teflon mixture through a boundary layer adjacent to an ablating surface. An equilibrium subroutine comprised of 37 species was produced by the "program generator." A "user's" program was written, utilizing this subroutine, which traversed the boundary layer, calculating the equilibrium composition of the gas, and the mixture parameters, such as the temperature, corresponding to a specified variation of enthalpy, pressure, and element composition. The element composition varied from 90% teflon to air ratio, to 100% air, and the temperature varied from 1000 °K to 5500 °K. The frontispiece depicts the variation of the mole fractions with temperature for a preliminary chemical system consisting of 28 species which reflects a teflon air ratio of 50% by volume.
REFERENCES


APPENDIX A

FLOW CHARTS

OF THE

EQUILIBRIUM PROGRAM
FLOW CHART FOR SUBROUTINE GENL7X

ENTRY GENL7X

1. Test $KOPT < 0$
   - Store $BEE(I)$ in $A(NSPL, I)$, $1 \leq I \leq NP$
   - $KOPT = -KOPT$
2. Loop $I = 1, NP$
   - Test $X(I) \geq 0$
     - $X(I) = 0.001$
     - Store $\ln(XI) = XLN(I)$
   - $ZLAMDA = 1$
     - $lnp \rightarrow PLOG$
     - $IPH = 1$
     - Test $KOPT = 0$
       - Go To (B)
     - Go To (A)
Compute enthalpy

Compute entropy

Compute enthalpy

Test $\text{IPH-1}$

Test $\text{KOPT}$

Test $\text{PROP-ETCH}$

Test $\text{PROP}$

Test $\text{IPROP-ETHC}$

Go to

Compute $T$ by drawing Parabola thru last 3 Points (Müller's Method)

Compute $T$ by Regula Falsi method

$1$ and $= 0$

Test $\approx 3$

 imaginary

$> 10^{-6}$

$< 10^{-6}$

$\neq 2$

$\neq 3$

$= 3$

TOLD = TOLD

TOLD = TOLD

TOLD = TOLD

Told TN2, T = TOLD

Told TN2, T = TOLD

$T = 1.05T$

$T = .95T$

$T = 2$

$< 0$

$= 0$

Go to R

Go to R

Real

Imaginary

Go to R
Do, P, T problem (see separate flow chart starting at C)

Test KOPT

=1

Compute entropy, enthalpy

Return

≠1

IPH = IPH + 1

<10

Test IPH

≥10

CALL ERROR

Return
FLOW CHART FOR P,T PROBLEM

Compute \( C_i \) \( (1 \leq i \leq NSP) \) eq. 25
Compute \( \ln X_i \) \( (NP + 1 \leq i \leq NSP) \) eq. 24

Store
\[ XLN(I) \]
\[ -XLAST(I) \]

ITER = 0
KOUNT = 0
KDLJT1 = 0

Compute \( F_j \) \( (1 \leq j \leq NP) \) eq. 30
Store Max \[ |F_j| - F(NP + 1) \]
Store \( F_j - FOLD(I) \) \( (1 \leq j \leq NP + 1) \)

Compute Jacobian eq. 31, Solve for \( \Delta nX_1 \) eq. 40
Compute \( \ln X_i \) \( (1 \leq i \leq NSP) \) eqs. 41, 24 and store \( XLN(I) \)
Compute \( X_i \) \( (1 \leq i \leq NSP) \), Any \( X_i > 1 \) set \( \ln X_i = -.001 \)
Compute \( \bar{y} \) eq. 35, \( F_i \) \( (1 \leq i \leq NP + 1) \) eq. 30, store \( F(I) \)

Loop \( I = 1, NP \)

Test
\( F(I) - FOLD(I) \) \( \leq 0 \)

\( \lambda = \lambda / 2 \)
KOUNT1 = 0
FOLD(I) = F(I)
KOUNT = KOUNT + 1

CALL
ERROR
Return

Test
KOUNT \( \geq 10 \)

0
A-5

KOUNT1 = KOUNT1 + 1

<0

Test

KOUNT1 = 2

≥0

KOUNT1 = 0

λ = 2λ

≤0

Test

λ - 1

>0

λ = 1

Loop I = 1, NSP

Test

XLN(I) - XLAST(I)

-10^-4

XLN(I)

(all ≤ 0)

>0

<0

Test

F(NP+1) - 10^-6

≤ 0

Go To (P-T)

Problem Complete

ITER = ITER + 1

XLN(I) - XLAST(I)

F(I) - FOLD(I)

≤ 0

Test

ITER = 30

>0

CALL ERROR

Go To (Return)
APPENDIX B

LISTING OF A

GENERATED PROGRAM
$SUBTG GEN% FULIST
SUBROUTINE GENLTX(X,PROP,PR2,T,P,JOPT,BEE)
DIMENSION ALE(4,4),X(5),BETA(4,4),A(9,4),F(5),J(8)
LXLH(8),C(8),XLAST(8),ULN(4),PSI(4,4,4),
2GENOJ2,TEMP(4),TEPPL(4),AT(4,5),FOLD(5),
3M(8),EM(8),GPI(8,6,5),BEE(4)

C SPECIES ORDER
C 1  C2
C 2  N2
C 3  E-
C 4  AR
C 5  N
C 6  NG
C 7  NG+
C 8  C

DATA M /
1 2.798225CE C3, 3.1960308E O3, 1.6320951E C8, 2.416383E O3,
2 6.3920616E C3, 2.5838710E O3, 2.5838710E C8, 5.566250OE O3/

DATA EM /
1 3.200200CE C1, 2.801600CE 01, 5.4862000E-O4, 3.9944000E 01,
2 1.4CC8000CE C1, 3.00C6000CE 01, 3.00C6000CE 01, 1.60C6000CE C1/

DATA A /
1 2.0C6000CCE C0, o., 0., 0., 0.,
2 C., 1.60C6000CCE C0, 1.00C6000CCE C0, 1.00C6000CCE C0,
3 o., o., 2.00C6000CCE C0, 0.,
4 o., 1.00C6000CCE 00, 1.00C6000CCE 00, 1.00C6000CCE 00,
5 C., 0., C., 0.,
6 1.00C6000CCE C0, 0., 0., 0.,
7-1.00C6000CCE C0, 0., 0., 0.,
8 0., 0., 1.00C6000CCE 00, 0.,
9 0., 0., 0., 0.,

DATA BETA /
1.0-0., 5.0000000E-O1, 5.0000000E-O1, 5.0000000E-O1,
2 5.0000000E-O1, 5.0000000E-O1, 5.0000000E-O1, 0.,
3 0., 0., 0., 0.,
4 0., 0., 0.,

DATA A(CP 11,1,1),I=1,76)
1 9.371694CE C3, 1.0730000E 04, 4.757999E C8, 5.525999E C3,
2 1.5220000E C4, 1.0173000E 04, 1.0150000E 04, 1.0180000E C4,
3 1.1342000E C4, 1.3349000E 04, 4.757999E C8, 1.7240000E 03,
4 1.1822000E C4, 9.3989999E 03, 1.2823000E C4, 1.4540000E C4,
5 1.2575000E C4, 1.3870000E 04, 4.757999E C8, 1.2840000E 03,
6 2.0150000E C4, 1.2450000E 04, 6.8910000E 03, 1.3410000E C4,
7 1.6180000E 04, 6.2150000E 03, 4.757999E C8, 1.3090000E 04,
8-1.5180000E 04, 1.2587000E 04, 1.1969000E C3, 1.4480000E 03,
9 1.7652000E 04, 2.6283000E C3, 4.757999E C8, 2.4830000E C4,
1 2.2241000E C4, 1.6225000E 04, 2.2221000E 04, 1.8841000E 04,
2 2.2241000E C4, 1.2720000E 04, 4.757999E C8, 5.1110000E C4,
3 5.5615000E C4, 2.2960000E 04, 2.2831000E 04, 7.5617000E C3,
4 2.3085000E C0, 1.7330000E C0, 0., 2.7274000E-03,
5 5.1045000E C2, 1.3950000E 04, 1.5749000E C0, 4.1353000E C0,
6 6.6689599E-C1, 2.7776000E-01, 0., 3.5410000E-C1,
7 1.2950000E C0, 2.6811000E 00, 1.7045000E-01, 1.7283000E 02,
8 3.0589000E-C1, 1.3710000E-01, 0., 6.0010000E-01,
9 3.1748000E-01, 3.3385000E-01, 1.6190000E C0, 2.2711000E-C1,
1-2.1760000E-C2, 1.0375000E C0, 0., 1.4685000E 00/

DATA (CP (11,1,1),I=77,152)
1 2.5200000CE C0, 2.8819000E-01, 1.6171000E C0, 1.3810000E C0,
2-1.4887000CE-C1, 1.3752000E C0, 0., 2.2874000CE C0,
<table>
<thead>
<tr>
<th>DATA CP</th>
<th>1-1,1,1,1= 153, 228/</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1.15C1000 C6, -2.321100 E6, 0-</td>
<td>= 2.1278000E C7,</td>
</tr>
<tr>
<td>2 3.6854000 E8, 3.2467000 E8, 7-5.121000 E8, 0-</td>
<td>= 1.6640000E 08,</td>
</tr>
<tr>
<td>3-3.6614000 C6, -3.2845000 C6, 6-</td>
<td>= 3.9705000E C7,</td>
</tr>
<tr>
<td>4 3.3307000 C6, 3.0504000 C6, 6-3.7195000 E8, 0-</td>
<td>= 1.6732000E 08,</td>
</tr>
<tr>
<td>5-1.9227000 C7, 2.9248000 E7, 0-</td>
<td>= 1.1864000E C8,</td>
</tr>
<tr>
<td>6 6.1576999 E8, 3.0298000 E8, 7-3.5904000 E8, 0-</td>
<td>= 2.5559000E C8,</td>
</tr>
<tr>
<td>7-2.8499000 E7, 4.8081999 E7, 0-</td>
<td>= 2.2493000E 08,</td>
</tr>
<tr>
<td>8 1.5726000 C8, 2.5620000 E8, 6-3.5288000 E8, 0-</td>
<td>= 4.3566999E 08,</td>
</tr>
<tr>
<td>9-4.8845000 C8, 3.2656000 C7, 7-</td>
<td>= 4.1154000E C9,</td>
</tr>
<tr>
<td>1 2.5316000 C6, 6.4829000 E7, 2-3.8320000 E8, 0-</td>
<td>= 8.7229999E C7,</td>
</tr>
<tr>
<td>2 2.1250000 E7, 1.8000000 E8, 3-3.0000000 E8, 0-</td>
<td>= 1.1000000E C9,</td>
</tr>
<tr>
<td>3 3.3300000 E8, 6.0000000 E8, 2-2.0000000 E2, 0-</td>
<td>= 5.6000000C C8,</td>
</tr>
<tr>
<td>4 4.5000000 E8, 3.7000000 E8, 0-3.0000000 E4, 0-</td>
<td>= 1.2250000E C4,</td>
</tr>
<tr>
<td>5 8.5000000 E8, 2.0000000 E10, 3-7.5000000 E4, 0-</td>
<td>= 2.5000000E 03,</td>
</tr>
<tr>
<td>6 6.8000000 E6, 8.5000000 E8, 3-3.0000000 E4, 0-</td>
<td>= 1.3000000E C4,</td>
</tr>
<tr>
<td>7 1.6600000 E8, 3.0000000 E4, 0-9.5000000 E3, 0-</td>
<td>= 1.4750000E C4,</td>
</tr>
<tr>
<td>8 1.2250000 E4, 1.0500000 C4, 0-3.0000000 E4, 0-</td>
<td>= 1.4500000E C4,</td>
</tr>
<tr>
<td>9 2.4500000 C4, 1.5250000 C4, 1.2000000 C4, 0-1.7750000 E4, 0-</td>
<td>= 1.7750000E C4,</td>
</tr>
<tr>
<td>1 1.6000000 E4, 1.6000000 E3, 0-3.0000000 E4, 0-</td>
<td>= 1.5750000E C4/</td>
</tr>
</tbody>
</table>

**DATA PSI /**

| 1-0. | 5.0000000E-01, 5.0000000E-01, 5.0000000E-01, 5.0000000E-01 |
| 2-C. | 5.0000000E-01, 5.0000000E-01, 0. |
| 3-C. | 0. |
| 4-C. | 0. |
| 5-C. | 0. |
| 6 5.0000000E-01, 5.0000000E-01, 5.0000000E-01, 5.0000000E-01 |
| 7 C. | 0. |
| 8 C. | 0. |
| 9 C. | 0. |
| 10 C. | 0. |
| 11 C. | 0. |
| 12 C. | 0. |
| 13 C. | 0. |
| 14 C. | 0. |
| 15 C. | 0. |
| 16 C. | 0. |
| 17 C. | 0. |
MINIMIZE FREE ENERGY
NP ARE THE NUMBER OF ELEMENTS
NSP ARE THE NUMBER OF SPECIES
T IS A GIVEN FCR KCPT=2,3
KCPT=1 MEANS P,T GIVEN
KCPT=2 MEANS P,T GIVEN, PRCP IS ENTALPY
KCPT=3 MEANS P,T GIVEN, PRCP IS ENTHALPY
X(I) ARE MCLC FRACTIONS, YBAR=TOTAL INITIAL MCLES
X(NSPI)=1.
DC 23 I=1,AP
A(NSPI,1)=0.
23 A(NSPI,1)=A(NSPI,1)+A(I,1)*X(I)*X(NSPI)
RETURN
ENTRY GEAR
IF(KCPT.GT.0)GO TO 26
KCPT=-KCPT
GC 29 I=1,AP
29 A(NSPI,1)=BEE(1)
C 26 WRITE(6,11)(X(I),I=1,AP)
26 CONTINUE
DC 20 I=1,AP
IF(X(I).LE.0.1*X(I)=.GO1
20 XN(I)=ALLG(X(I))
IP=1
C 2 FCRMA(3ElG,6,110)
C WRITE(6,13)T,P,PRCP,KCPT
C 13 FCRMAT(3E20,8,110)
C INITIALIZE LAMDA
U=UDP=1.
PLCG=ALLG(P),
1995 GC TC(2007,2CGG,2CCG),KCPT
2000 IF(IPF.NE.1)GO TO 2007
LCPT=KCPT
C WILL RETURN IN GEAR THE MIXTURE ENTHALPY FCR KCPT
C I=2,ENTHALPY KCPT=3
GC TC 520C
C2164 WRITE(6,11)ETCF,T
2164 CONTINUE
IF(ABS((PRCP-ETCF)/PRCP).LT.0.000C01)GO TC 300
T=P=ETCF
IF(IPF.NE.2)GO TO 2002
IN2=TCG
TCL=T
IF(FNEW)=2CC3,300,20C4
2CC3 T=0.5*T
GC TC 2007
2CC4 T=1.0*T
GC TC 2007
2C02 IF(IPF.NE.3)GO TO 2008
3010 FPRIME=(FNEW-FOLD)/1(T-TCL)
TA=TCL
TCL=T-FNEW/FPRIME
GC TC 2007
2008 H=G=TCL
H=G=TCL-IN2
CH=FA/FA1
AM=CM*(FAHFA-1*CA)*FCLCH*CH*FN2)
BP=(2*CH+1.)*FAHFA-FCLCH*(1.1*CP)*2+CP*CLH*FN2
CM=FAHFA*(1.1+CP)
CLH=BM*BM-4.*AM*CH
IF(CCLH.LT.0.)GC TC 3C1C
CCLH=SQRT(CCA)
DENCM(1)=EP*CP
DENCM(2)=EP-CP
IF(ABS(DENCM(1))-ABS(DENCM(2)))3C32,3C03,3C04
3C03 BCTCM=DENCM(2)
3C04 BCTCM=DENCM(1)
3C05 TH2=TLC
TCLC=T
TH=TH-TH2.*CM/BCTCM
2CC7 LEPl=1
2CC8 GC TC 5407
2CC9 KKK=3
2CC10 GC TC 420C
2CC11 CONTINUE
C41C1 WRITE(0.,11)EICF,FAHFA,FPRIME,T,FCLCH,TLC,X
C41G1 DC 12 I=1,ASP
12 XLAST(I)=XLN(I)
4G ITER=C
26 KCLNT=C
27 KKK=1
28 64 KCLNT=0
29 LLL=3
3C GC TC 60C
3C GC 211 I=1,N
3C FCLC(I)=F(I)
3C 61 GC TC(4C17,4C2C),KKK
3C 41G2 LLL=2
3C GC TC 60C
C45G1 WRITE(0.,11)ZLAMCA,FCLC
45G1 CONTINUE
61 IF(F(I)-FCLC(I))7C,7C,71
61 71 IF(F(I),LT.,CGCC01)GC TO 85
6C ZLAMDA=ZLAMDA/2.
41 KCLNT=C
42 KUN=KUN+1
43 KKK=2
43 DC 75 I=1,10P
46 F(I)=FCLC(I)
46 75 XLN(I)=XLAST(I)
47 F(I)=FCLC(N)
47 81 KCLNT=KCLNT+1
48 IF(KCLNT-1C)61,61,72
49 72 CALL EMKCR(CA1,KRW,13)
5C 74 GC TC 305
5C 7C KKK=1
51 81 KCLNT=KCLNT+1
53 IF(KCLNT-21E5,50,50
54 9C ZLAMDA=2.*ZLAMDA
55 IF(ZLAMDA.GT.,1.*ZLAMDA=1.
56 KCLNT=0
57 85 GC 66 I=1,ASP
5C IF(ABS((XLN(I)-XLAST(I))/XLN(I))-00C1)86,86,91
5C 86 CONTINUE
B-5

IF(F(N) -.LCCCI195,95,95)

41 IF(=1,ERIC+1)

C WRITE(e,111)*

C SAVE LXA(II) I=1,NSP AND F(I) I=1,N

CC I 1=1,NSP

25 XLAST(I)=XLA(I)

CC 185 I=1,N

169 IF(=F(I))

1 IF(FIF10)=1,52,52

92 CALL ERKCH(GX1,KRLM+12)

95 CENTRAL

C 99 WRITE(e,111)CLMPCA,X,XLA,XLAST

IF(KCFI+EC.*10)GC TC 3CC

IP=IP+1

IF(1,FIF10)1555,1555,1555

1548 CALL ERKCH(GX1,KRLM+11)

11 FCPM AT(52GC,2)

3CC LLPI=KCP1+3

GC TC 52CC

3Cc PKCP=ETC

LC PI=7

GC TC 5407

3C7 LCPI=4

3C5 KCPI=LCPI-2

PH2=ETC

HEILKN

4C17 CC 4CC J=1,NSP

A(1,J)=O.

CC 4CC2 I=1,N

I=NX+1

4CC2 A(1,J)=A(1,J)*X(I)*BETA(I,J)

4CC1 A(1,J)=X(I)*A(1,J)

C LF MATRIX IN AI ARRAY

C GET L1/YEAR(I) /LN(L) IN TEMP(I)

CC 4CC4 L=1,NSP

TEMP(I)=0.

CC 4CC3 J=K,NSP

JJ=J-NP

4CC2 TEMP(I)=TFP(I)+PSI(JJ+1,L)*X(J)

4CC4 TEMP(I)=TMPF(I)+X(L)*A(I,L) /A(NSP+1)

C NLM FCPY CF(I)/CLNX(L) FCK I=2,NSP AND L=1,NSP

CC 4CC6 I=2,NSP

CC 4CC6 L=1,NSP

TEMPI(I)=G.

CC 4CC5 J=K,NSP

JJ=J-NP

4CC5 TEMPI(I)=TEMPI(I)+PSI(JJ,L)*X(I)

4CC6 A(1,L)=A(1,L)*X(I)+TEMPI(I)-A(NSP+1)*TEMP(I)

LLL=1

4CC7 CALL CLEM7(XA,CLNX,F,NSP,K,AT)

4C26 CC 4CC1 L=1,NSP

4CC1 XLA(I)=XLA(I)-2*CLMPA*CLNX(I)

4CC6 CC 4CC1 L=1,NSP

4CC1 TEMPI(L)=XLA(L)+CL(L)

CC 4CC5 L=K,NSP

XLA(I)=L.

JJ=J-NP

CC 4CC1 L=1,NSP

4CC1 XLA(I)=XLA(I)+TEMPI(I)*BETA(JJ,L)
4G25 XLN(I)=XLN(I)-C(I)
4G22 CONTINUE
C COMPLETE X(I) AND YBAR
X(NSPI)=0.
CC 4G0C J=1,NSP
4C X(J)=EXP(XLN(J))
4C IF(AES(XLN(J)).GT.68),X(J)=0.
4C GC TC(102,1G2,1G1),KKK
CLM=CLSổiES F FOR NEWTON RAPHSCHN
ALSO STICKS LARGEST F IN F(N)
6C CLM=C.
6C CC 6C1 I=1,NSP
6C1 CM=CUM+X(I)
F(I)=CLM-1.
CC 6C3 I=2,NSP
CC CLM=C.
6C CC 6C2 J=1,NSP
6C2 CLM=CUM+A(J,1)*X(J)
6C3 F(I)=CLM-A(NSPI,1)/X(NSPI)
FINKE THE LARGEST F(I)
F(N)=ABS(F(I))
CC 6C5 I=2,NSP
CC M=ABS(F(I))
CC IF(M=F(N))6C5,6C4,6C0
6C4 F(N)=E
6C5 CONTINUE
C WRITE(6,11)F
CC GC TC(4007,45CC,21C),LLL
C FIRST SUBSCRIPT DEFINES SPECIE NUMBER
C SECOND SUBSCRIPT IS FOR A RANGE IN TEMPERATURE
C THIRD SUBSCRIPT 1=A
C THIRD SUBSCRIPT 2=B
C THIRD SUBSCRIPT 3=C
C THIRD SUBSCRIPT 4=D
C THIRD SUBSCRIPT 5 UPPER T IN TEMP. RANGE
C S=A*B*T+B*T+C
C H=A*T+.5*B*T*T+C
CC CONVERT MCLE TC MASS FRACTIONS
C Z ARRAY FOR MCLE TC MASS FRACTIONS
52OC EMM=C.
CC 52C1 I=1,NSP
5201 EMM=EMM+X(I)*EMM(I)
CC 52C2 I=1,NSP
52C2 Z(I)=X(I)*EMM(I)/EMM
C WRITE(6,11)Z,EMM,EMM
5407 TLLG=ALLGT(1)
ETCZ=0.
CC 54C1 I=1,NSP
CC 54C2 JJ=1,ANAN
CC IF(CP(1,1,5)=T)54C2,54C3,54C3
54C2 CONTINUE
C CALL ERRCH(T,1,10)
54C3 GC TC(5406,5405,5404,5403,5403,5403,5403,5403),LLP
5405  HI=CP(1,JJ,1)+.5*CP(1,JJ,2)*T+CP(1,JJ,4)
C  WRITE6,1131,JJ,HI
ETCEN=ETCEN+4(I)*HI
GC TO 5401
5404  SI=CP(1,JJ,1)*TLOG+CP(1,JJ,2)*T+CP(1,JJ,3)
C  WRITE6,1131,JJ,SI
ETCEN=ETCEN+X(I)*SI/WM(I)+XLM(I)+PLOG)/EMM
GC TO 5401
5406  C(I)=CP(1,JJ,1)-CP(1,JJ,3)-.5*CP(1,JJ,2)+CP(1,JJ,4)/
     IT-CP(1,JJ,1)*TLOG
C(I)=C(I)/WM(I)+PLOG
C  WRITE6,1131,JJ,C(I)
5401  CONTINUE
C 113 FORMAT(211C,3E2G,8)
GC T0(163,2164,2164,365,365,365,365,365,365),LOPT
ENC
$1   SCLEM7
SUBCLUTINE SCLEM7(B,X,C,M,M1,AT)
DIMENSION B(M,M),X(M),D(M1),AT(M,M)
C  EQUATIONS ARE OF THE FORM BX=0
DC 17 I=1,M
17 X(I)=0.0
DO 200 I=1,M
200 AT(I,M1)=C(I)
DO 201 I=1,M
201 AT(I,J)=B(I,J)
DO 301 K=1,M
301 C=AT(K,K)
30 D=K=1,M
IF(ABS(AT(K,K))-ABS(C))9,5,8
9 C=AT(K,K)
10 IT=1
9 CONTINUE
IF(1-IT-7,7,7C
7C 71 J=K,M1
TEMP=AT(K,J)
AT(K,J)=AT(J,T)
71 AT(IT,J)=TEMP
7  CG 10 I=1,M1
10 AT(K,I)=AT(K,I)/D
IF(M-K)50,5C,18
18 R=K+1
CG 30 I=K1,M1
C=AT(I,I)
CG 30 I=K1,M1
30 AT(I,J)=AT(I,J)-AT(K,J)*C
32 CONTINUE
50 X(M)=AT(M,M1)
DO 65 K=2,M
X0=M+1-K
C=AT(KR,M+1)
DO 60 K=1,K
60 C=C-AT(KR,K)*X(I)
RETURN
ENC

412 CARCS PRINTED
APPENDIX C

LISTING OF SUGGESTED SUBROUTINE ERROR
C-1

SUBTC LIST  **CHECK.
  SUBROUTINE ERROR(C,K,I)
  WRITE(6,10)I
  10 FORMAT(6F30.6,C15.1,15X,1) FOR SPECIE NO. 14,4X,FACT VALU
  10 FORMAT(15X,1)
  GC IC 7
  2 WRITE(6,12)
  12 FORMAT(4X,50X) EQUILIBRIUM ITERATIONS EXCEED 10
  GC IC 7
  3 WRITE(6,13)
  13 FORMAT(4X,50X) PRESSURE TEMPERATURE ITERATIONS EXCEED 30
  GC IC 7
  4 WRITE(6,14)
  14 FORMAT(4X,50X) CLAMPAD CUT 1G TIMES
  7 RETURN
END

20 CARDs PRINTED
APPENDIX D

SAMPLE INPUT FOR THE

CHEMISTRY EQUILIBRIUM GENERATOR PROGRAM
<table>
<thead>
<tr>
<th>WORD 1</th>
<th>WORD 2</th>
<th>WORD 3</th>
<th>WORD 4</th>
<th>WORD 5</th>
<th>WORD 6</th>
<th>WORD 7</th>
<th>WORD 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

CARD NO.