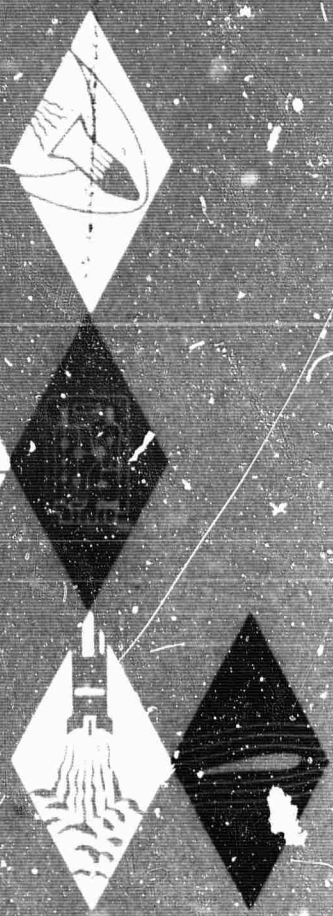


AD 644898



RESEARCH
ENGINEERING
PRODUCTION

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

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION			
Hardcopy	Microfiche		
\$ 3.00	\$.65	63 pp	AD
/ ARCHIVE COPY			

December 14, 1966

DDC
RECEIVED
JAN 12 1967
RECEIVED
C

GENERAL APPLIED SCIENCE LABORATORIES, INC.
MERRICK and STEWART AVENUES, WESTBURY, L. I., N. Y. (516) ED 3-6960

**BEST
AVAILABLE COPY**

Project 8050/0300

Total No. of Pages - vi & 50

Copy (9) of 150

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"

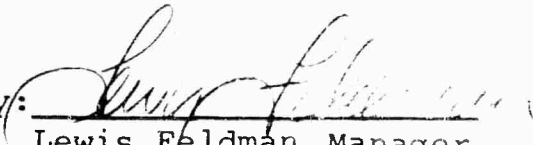
Prepared by

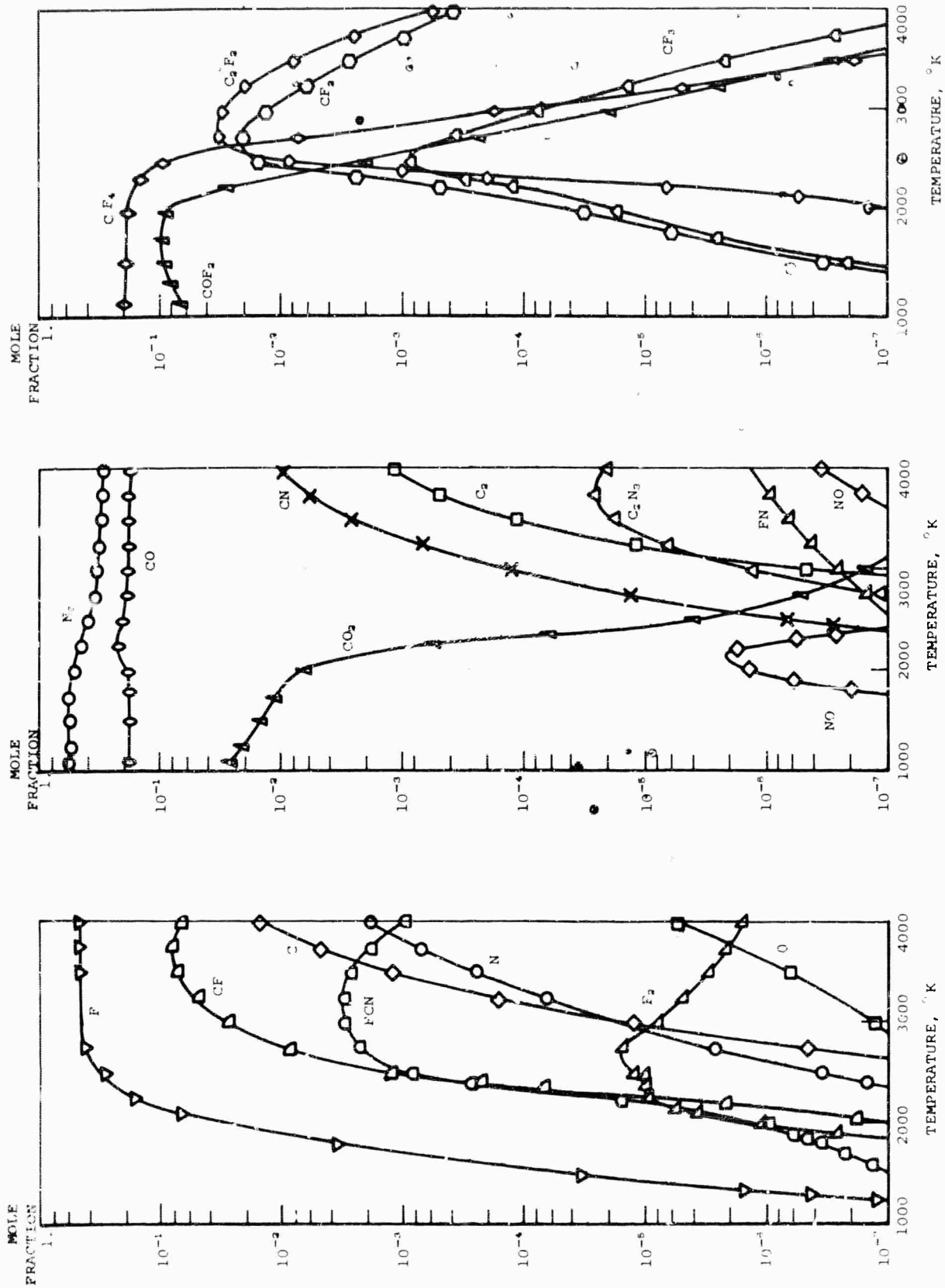
General Applied Science Laboratories, inc.
A Subsidiary of The Marquardt Corporation
Merrick and Stewart Avenues
Westbury, L. I., New York

*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 = .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.

BLANK PAGE

TABLE OF CONTENTS

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

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 NP rows and NP columns
A_2	partitioned A matrix consisting of $(NSP-NP)$ rows NP 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)]
DF	the Jacobian $(\partial F_j / \partial \ln X_\ell)$ 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}_1	the partitioned row vector \bar{G} consisting of NP vector elements
\bar{G}_2	the partitioned row vector \bar{G} consisting of $(NSP-NP)$ vector elements
\bar{H}	mixture enthalpy, ft^2/sec^2
H	mixture enthalpy, cal
h_i	species enthalpy, cal/mole

\bar{h}_i	species enthalpy, ft^2/sec^2
NP	number of elements (including charge, if applicable) of the chemistry system considered.
NSP	total number of species in the system
P	pressure, atmospheres
R	universal gas constant 1.98726 cal/mole $^\circ\text{K}$ or 89506 $\text{ft}^2/\text{sec}^2\text{-}^\circ\text{K}$ lb/lb mole
\hat{S}	mixture entropy lb mole/lb
S	mixture entropy, cal/ $^\circ\text{K}$
\bar{S}	mixture entropy, ft^2/sec^2 $^\circ\text{K}$
s_i	species entropy at standard state ($p = 1$ atm), cal/mole $^\circ\text{K}$
\bar{s}_i	species entropy at standard state ($p = 1$ atm), ft^2/sec^2 $^\circ\text{K}$
T	temperature, $^\circ\text{K}$
X_i	mole fraction of species i
Y_i	number of moles of species i
\bar{Y}	the total moles in the system $\left(\bar{Y} = \sum_{i=1}^{\text{NSP}} Y_i \right)$

Greek

Φ	$\frac{G}{RT} + \sum_{j=1}^{\text{NSP}} \left[\sum_{i=1}^{\text{NSP}} Y_i a_{ij} - b_j \right] \pi_j$ [see Eq. (8)]
π_j	Lagrange multipliers [see Eq. (8)]
β_{ij}	matrix elements resulting from the matrix multiplication $A_2 A_1^{-1}$ [see Eq. (26)]

Δ the column vector consisting of the modified
Newton-Raphson correction terms [Eqs. (39)]

δ_{ij} Kronecker delta

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.
2. Minimization of the Gibbs Free Energy of the system.

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 twofold:

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

$$G = H - TS \quad (1)$$

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

$$H = \sum_i Y_i h_i(T) \quad (2)$$

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

$$S - S_0 = \sum Y_i [s_i(T) - R \ln X_i - R \ln P] \quad (3)$$

where P is the mixture pressure in atmospheres

R is the universal gas constant in cal./mole °K

$s_i(T)$ is the entropy of specie i , cal./mole $^{\circ}K$

X_i is the mole fraction of species i

S_0 is the reference entropy taken as zero at

$T = 0^{\circ}K$, $p = 1$ 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),

$$\frac{G}{RT} = \sum_{i=1}^{NSP} Y_i \left\{ \frac{1}{R} \left[\frac{h_i}{T} - s_i \right] + \ln X_i + \ln P \right\} \quad (4)$$

where 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

$$\sum_{i=1}^{NSP} Y_i a_{ij} = b_j \quad j=1,2,\dots, NP \quad (5)$$

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

$$[Y_1, Y_2, \dots, Y_{NSP}] \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1, NP} \\ a_{21} & a_{22} & \dots & a_{2, NP} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{NSP, 1} & a_{NSP, 2} & \dots & a_{NSP, NP} \end{bmatrix} = (b_1, b_2, \dots, b_{NP}) \quad (6)$$

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

$$\sum_{i=1}^{NSP} X_i = 1 \quad (7)$$

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

$$\Phi = \frac{G}{RT} + \sum_{j=1}^{NP} \left\{ \left[\sum_{i=1}^{NSP} Y_i a_{ij} - b_j \right] \pi_j \right\} \quad (8)$$

where π_j are the Lagrange multipliers.

Let

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

Then from Eq. (4) and (9)

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

and

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

$$k=1, 2, 3, \dots, NSP$$

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

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]:

$$Y_1 \frac{\partial}{\partial Y_k} \left(\ln \frac{Y_1}{\bar{Y}} \right) + Y_2 \frac{\partial}{\partial Y_k} \left(\ln \frac{Y_2}{\bar{Y}} \right) + \dots + Y_{NSP} \frac{\partial}{\partial Y_k} \left(\ln \frac{Y_{NSP}}{\bar{Y}} \right) \quad (12)$$

Note that:

$$\begin{aligned}
 Y_l \frac{\partial}{\partial Y_k} \left(\ln \frac{Y_l}{\bar{Y}} \right) &= \frac{\bar{Y} - Y_l}{\bar{Y}} \quad \text{for } l = k \\
 &= - \frac{Y_l}{\bar{Y}} \quad \text{for } l \neq k
 \end{aligned}
 \tag{13}$$

From which it can be seen that

$$\sum_i Y_i \frac{\partial}{\partial Y_k} \left[C_k + \ln \left(\frac{Y_i}{\bar{Y}} \right) \right] = 0
 \tag{14}$$

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

$$\frac{\partial \Phi}{\partial Y_i} = C_i + \ln X_i + \sum_{j=1}^{NP} a_{ij} \pi_j
 \tag{15}$$

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

$$g_i = - C_i - \ln X_i = - \left\{ \frac{1}{R} \left[\frac{h_i}{T} - s_i \right] + \ln P + \ln X_i \right\} \quad (16)$$

then Eq. (15) can be written as

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

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

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1, NP} \\ a_{21} & a_{22} & \dots & a_{2, NP} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{NSP, 1} & a_{NSP, 2} & \dots & a_{NSP, NP} \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \cdot \\ \cdot \\ \pi_{NP} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \cdot \\ \cdot \\ g_{NSP} \end{bmatrix} \quad (18)$$

10

or

$$A \bar{\Pi} = \bar{G} \quad (19)$$

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

$$\begin{array}{c} \text{NP} \\ \updownarrow \\ \left[\begin{array}{c} A_1 \\ \text{---} \\ A_2 \end{array} \right] \bar{\Pi} = \left[\begin{array}{c} G_1 \\ \text{---} \\ G_2 \end{array} \right] \end{array} \quad \begin{array}{c} \text{NP} \\ \updownarrow \\ \text{NSP-NP} \end{array} \quad (20)$$

or

$$A_1 \bar{\Pi} = \bar{G}_1 \quad (21)$$

and

$$A_2 \bar{\Pi} = \bar{G}_2 \quad (22)$$

Solving Eq. (21) for $\bar{\Pi} = A_1^{-1} \bar{G}_1$ and substituting into Eq. (22), one obtains:

$$A_2 A_1^{-1} \bar{G}_1 = \bar{G}_2 \quad (23)$$

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

$$\ln X_n = \sum_{j=1}^{\text{NP}} \left\{ \beta_{\text{n-NP}, j} (C_j + \ln X_j) \right\} - C_n \quad (24)$$

$$n = \text{NP} + 1, \text{NP} + 2, \dots, \text{NSP}$$

where

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

$$[A_2 A_1^{-1}] = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \dots & \beta_{1, NP} \\ \beta_{21} & \beta_{22} & \beta_{23} & \dots & \beta_{2, NP} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \beta_{NSP-NP, 1} & \beta_{NSP-NP, 2} & \beta_{NSP-NP, 3} & \dots & \beta_{NSP-NP, NP} \end{bmatrix} \quad (26)$$

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

$$\sum_{i=1}^{NSP} X_i a_{ij} = \frac{b_j}{Y} \quad j=1, \dots, NP \quad (27)$$

where \bar{Y} is the total number of moles in the system, i.e.

$$\bar{Y} = \sum_{i=1}^{NSP} Y_i \quad (23)$$

$$X_i = \frac{Y_i}{\bar{Y}} \quad (29)$$

The non-linear system of equations given by (7), (24) and (27) in the unknowns \bar{Y} and X_i ($i=1,2,\dots,NSP$) 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:

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

$$F_j = \sum_{i=1}^{NSP} X_i a_{ij} - \frac{b_j}{Y} \quad (30)$$

$j=2, 3, \dots, NP$

The Jacobian to be used in the Newton Raphson iterative procedure for NP dimensions, $DF = (\partial F_j / \partial \ln X_\ell)$ for $(1 \leq j \leq NP)$, $(1 \leq \ell \leq NP)$ is formed as follows

$$DF = \begin{bmatrix} \frac{\partial F_1}{\partial \ln X_1} & \frac{\partial F_1}{\partial \ln X_2} & \dots & \frac{\partial F_1}{\partial \ln X_{NP}} \\ \frac{\partial F_2}{\partial \ln X_1} & \frac{\partial F_2}{\partial \ln X_2} & \dots & \frac{\partial F_2}{\partial \ln X_{NP}} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial F_{NP}}{\partial \ln X_1} & \frac{\partial F_{NP}}{\partial \ln X_2} & \dots & \frac{\partial F_{NP}}{\partial \ln X_{NP}} \end{bmatrix} \quad (31)$$

The following is required for the evaluation of the various terms $\frac{\partial F_j}{\partial \ln X_\ell}$:

$$\left. \begin{aligned} \frac{\partial \ln X_j}{\partial \ln X_\ell} &= \delta_{j\ell} \\ \frac{\partial X_j}{\partial \ln X_\ell} &= X_j \delta_{j\ell} \end{aligned} \right\} \begin{aligned} & \left(\begin{aligned} 1 \leq j \leq \text{NP} \\ 1 \leq \ell \leq \text{NP} \end{aligned} \right) \end{aligned} \quad (32)$$

where $\delta_{j\ell}$ is the Kronecker delta.

$$\delta_{j\ell} = \begin{cases} 1 & \text{for } j = \ell \\ 0 & \text{for } j \neq \ell \end{cases} \quad (33)$$

From Eq. (24) for $(\text{NP} + 1 \leq n \leq \text{NSP})$, $(1 \leq \ell \leq \text{NP})$

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

and

$$\frac{\partial X_n}{\partial \ln X_\ell} = X_n \beta_{n-\text{NP},\ell} \quad (34)$$

since $C_i = C_i(p, T)$.

From Eq. (27)

$$\frac{1}{\bar{Y}} = \frac{1}{b_1} \sum_{i=1}^{\text{NSP}} X_i a_{ij} \quad (35)$$

then

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

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:

$$\left. \begin{aligned} \frac{\partial F_1}{\partial \ln X_\ell} &= X_\ell + \sum_{n=NP+1}^{NSP} X_n \beta_{n-NP, \ell} \\ \frac{\partial F_j}{\partial \ln X_\ell} &= X_\ell^{a_{\ell j}} + \sum_{n=NP+1}^{NSP} [X_j^{a_{nj}} \beta_{n-NP, \ell}] - b_j \frac{\partial \left(\frac{1}{Y} \right)}{\partial \ln X_\ell} \end{aligned} \right\} (37)$$

$j=2, 3, \dots, NP$
 $\ell=1, 2, \dots, NP$

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

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ F_{NP} \end{bmatrix} \quad (38)$$

and the set of correction terms denoted by the column vector

$$\Delta = \begin{bmatrix} \Delta \ln X_1 \\ \Delta \ln X_2 \\ \cdot \\ \cdot \\ \Delta \ln X_{NP} \end{bmatrix} \quad (39)$$

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

$$(DF) \cdot (\Delta) = F \quad (40)$$

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

$$(\ln X_j)_{N+1} = (\ln X_j)_N - \lambda (\Delta \ln X_j)_N \quad (41)$$

$$j=1, \dots, NP$$

where N denotes the iteration number and λ 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 λ 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 λ has been introduced to keep the iterations in the neighborhood of the solution whenever possible. If the solution is converging and $\lambda < 1$, then λ 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)

$$|F_j| < 10^{-6}, \quad j=1,2,\dots, NP$$

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

(2)

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

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

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 (see 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 , $j=1,2,\dots, NP$, 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_2 , 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_2 , 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 $(X_i)_1$, and of enthalpy, \bar{H}_1 (or entropy \hat{S}_1). A second guess of the temperature is obtained (for enthalpy):

$$T_2 = T_1 \pm 0.05 T_1 \quad \text{for} \quad \bar{H}_1 \gtrless \bar{H}$$

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

\hat{S} is the entropy, $\frac{\bar{S}}{R}$, $\frac{\text{lb mole}}{\text{lb}}$

\bar{S} is the entropy per unit mass, $\frac{\text{ft}^2}{\text{sec}^2 \cdot \text{K}}$.

The program then calculates \bar{H}_2 corresponding to the specified pressure and T_2 . The third estimate of temperature is calculated by the application of the "regula falsi" method (or method of chords), as follows:

$$T_3 = T_2 - \frac{f_2}{\left(\frac{df}{dT}\right)_2} \quad (43)$$

where $\left(\frac{df}{dT}\right)_2 = \frac{f_2 - f_1}{T_2 - T_1}$

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

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 $\left| \frac{\bar{H}_k - \bar{H}}{\bar{H}} \right| \leq 10^{-8}$

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 PROGRAMInitializing Call Statement

```
CALL GENL7X (X, PROP, PR!, 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 $^{\circ}\text{K}$.

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).

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

General Call Statment

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 = ± 1

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

Pressure-Enthalpy (P- \bar{H}) Option: KOPT = ± 2

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

Pressure-Entropy (P- \hat{S}) Option: KOPT = ± 3

Same inputs as for the P- \bar{H} option, above.

KOPT = ± 1 for pressure-temperature input option

± 2 for pressure-enthalpy input option

± 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.

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.

<u>KOPT</u>	<u>PROP</u>	<u>PR2</u>
± 1	enthalpy, ft^2/sec^2	entropy, lb moles/lb
± 2	enthalpy, ft^2/sec^2	entropy, lb moles/lb
± 3	entropy, lb moles/lb	enthalpy, ft^2/sec^2

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, $\max \{ |F_j| \}$ as defined by Eq. (30) does not decrease, the value of the relaxation factor λ used in Eq. (41) will be reduced; this reduction of λ will be repeated until $\max \{ |F_j| \}$ decreases from iteration to iteration. However, the program calls the error subroutine if λ 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 (A1, DLNX, F, N, N+1, AT)

where the formal parameters are defined as follows:

A1 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+1).

Streamline Calculation Using Equilibrium P- \hat{S} Option

To perform calculations along streamlines using the P- \hat{S} option of the equilibrium chemistry program, it is first necessary to execute either the P-T option or the P- \bar{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- \hat{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:

$$\bar{c}_{p_i} = a_i + b_i T$$

$$\bar{h}_i = \bar{h}_{i_0} + \int_0^T c_{p_i} dT = a_i T + \frac{b_i T^2}{2} + d_i \quad (44)$$

$$\bar{s}_i = \bar{s}_{i_0} + \int_{T_{0i}}^T \frac{c_{p_i} dT}{T} = a_i \ln T + b_i T + c_i \left(T > T_{0i} \right)$$

- where \bar{c}_{p_i} is the species specific heat at constant pressure, $\text{ft}^2/\text{sec}^2 \text{ } ^\circ\text{K}$
- \bar{h}_i is the species enthalpy, ft^2/sec^2 (note that species sensible enthalpy is taken to be zero at $0 \text{ } ^\circ\text{K}$)
- \bar{h}_{i_0} is the species heat of formation at $0 \text{ } ^\circ\text{K}$, ft^2/sec^2
- \bar{s}_i is the species entropy, $\text{ft}^2/\text{sec}^2 \text{ } ^\circ\text{K}$ (note \bar{s}_i is taken to be zero at $0 \text{ } ^\circ\text{K}$, $p = 1$ atmosphere)
- T_{0i} is the lower bound of temperature for the curve fit of c_{p_i} , $^\circ\text{K}$.

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

<u>CARD NO.</u>	<u>COLUMNS</u>	<u>DATA TO BE PUNCHED ON CARD</u>
1	3-8	Date (month/day/year)
	45	1
	46	1
2	1	7
	43-45	(right adjusted) number of species in system
3	1	2
4,5,6,...	6A12	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.

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.

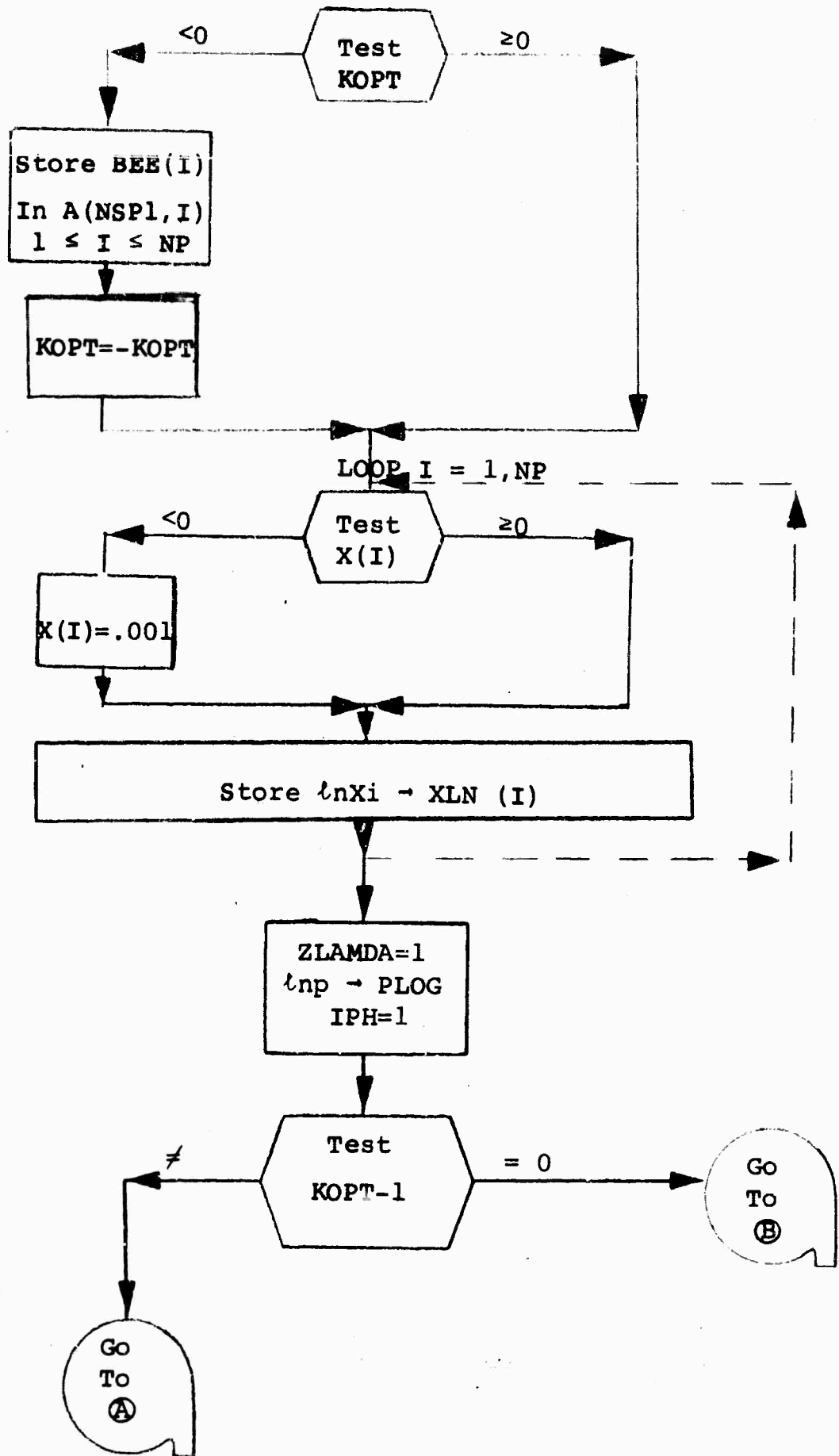
REFERENCES

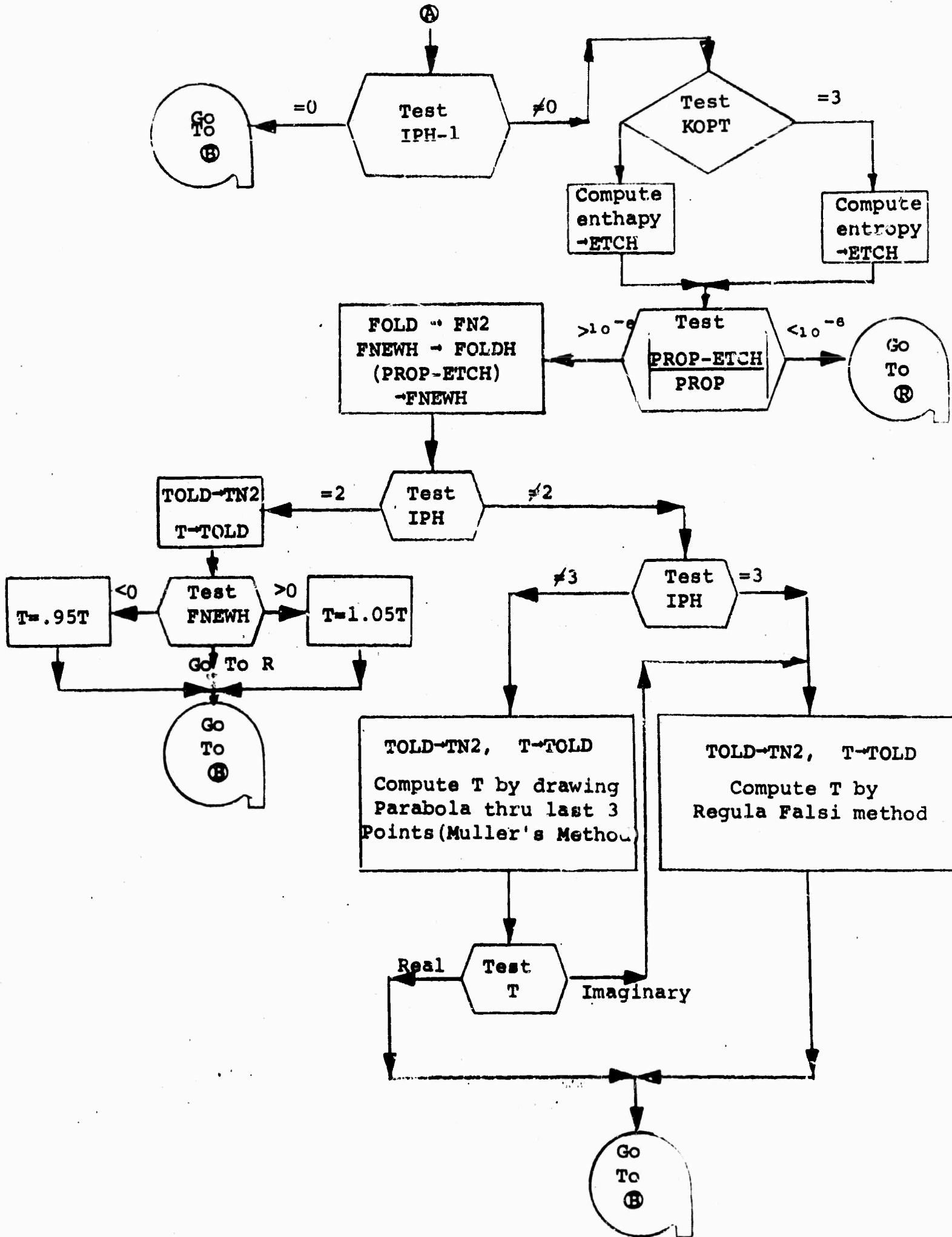
1. Kinetics, Equilibria and Performance of High Temperature Systems (Proceedings of the First Conference Western States Section, The Combustion Institute, November 1959) Edited by Bahn and Zukoski.
2. White, W. B., Johnson, S. M., and Dantzig, G. B., Chemical Equilibrium in Complex Mixtures, *J. Chem. Phys.* 28, No. 5, pp. 751-755 (May 1958).
3. Allbaugh, O. R., Equilibrium Ramjet Performance, 704 Computer Program #0446, Marquardt Report S-403 (December 1963).
4. Henrici, P., Elements of Numerical Analysis (John Wiley & Sons, New York, 1964).
5. Scarborough, J., Numerical Mathematical Analysis (The John Hopkins Press, Baltimore, 1958).
6. Hoffman, J. and Wecker, M., Program Generator for Finite-Rate Chemical Systems, GASL TR-575 (March 1966).

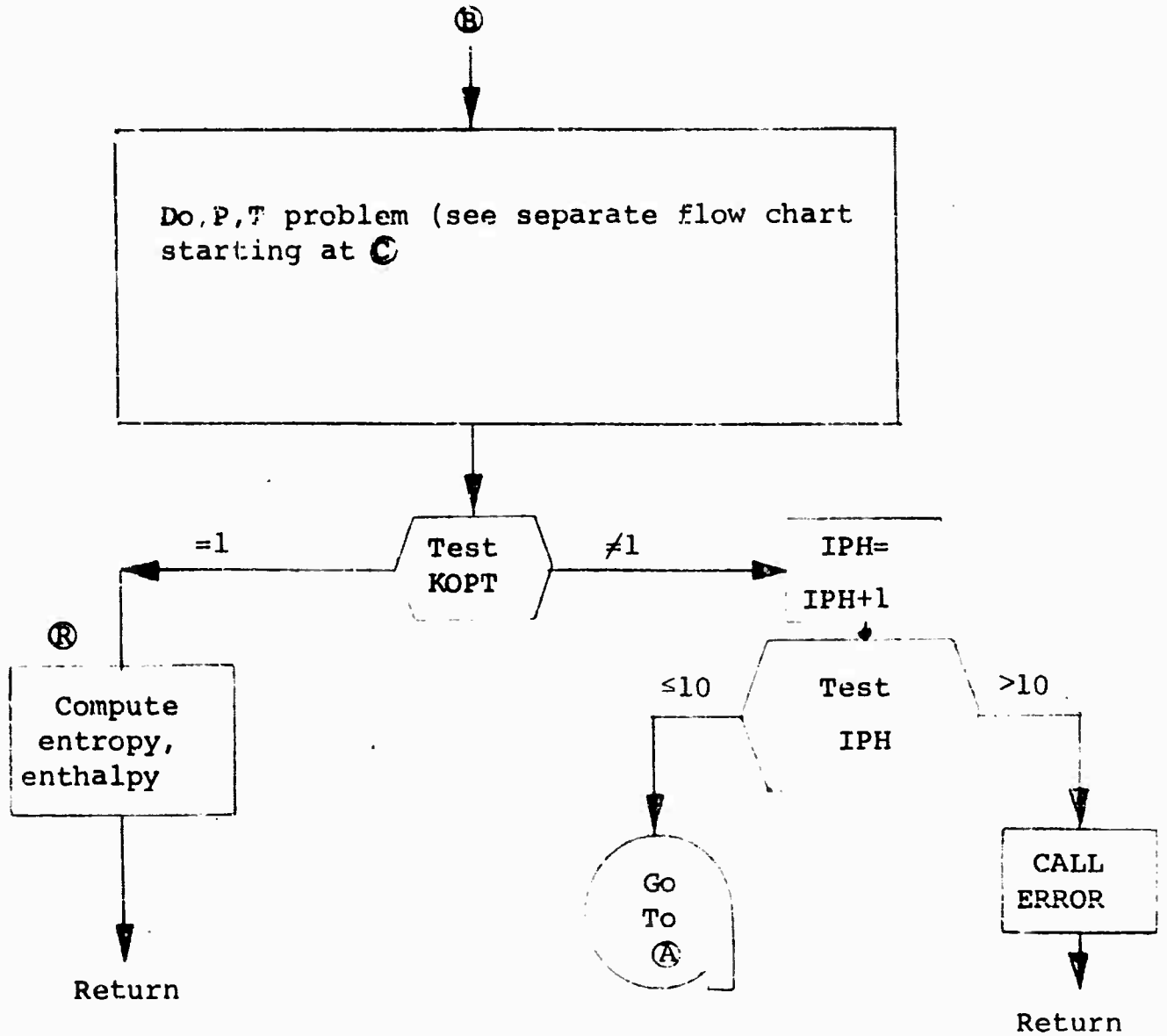
APPENDIX A
FLOW CHARTS
OF THE
EQUILIBRIUM PROGRAM

FLOW CHART FOR SUBROUTINE GENL7X

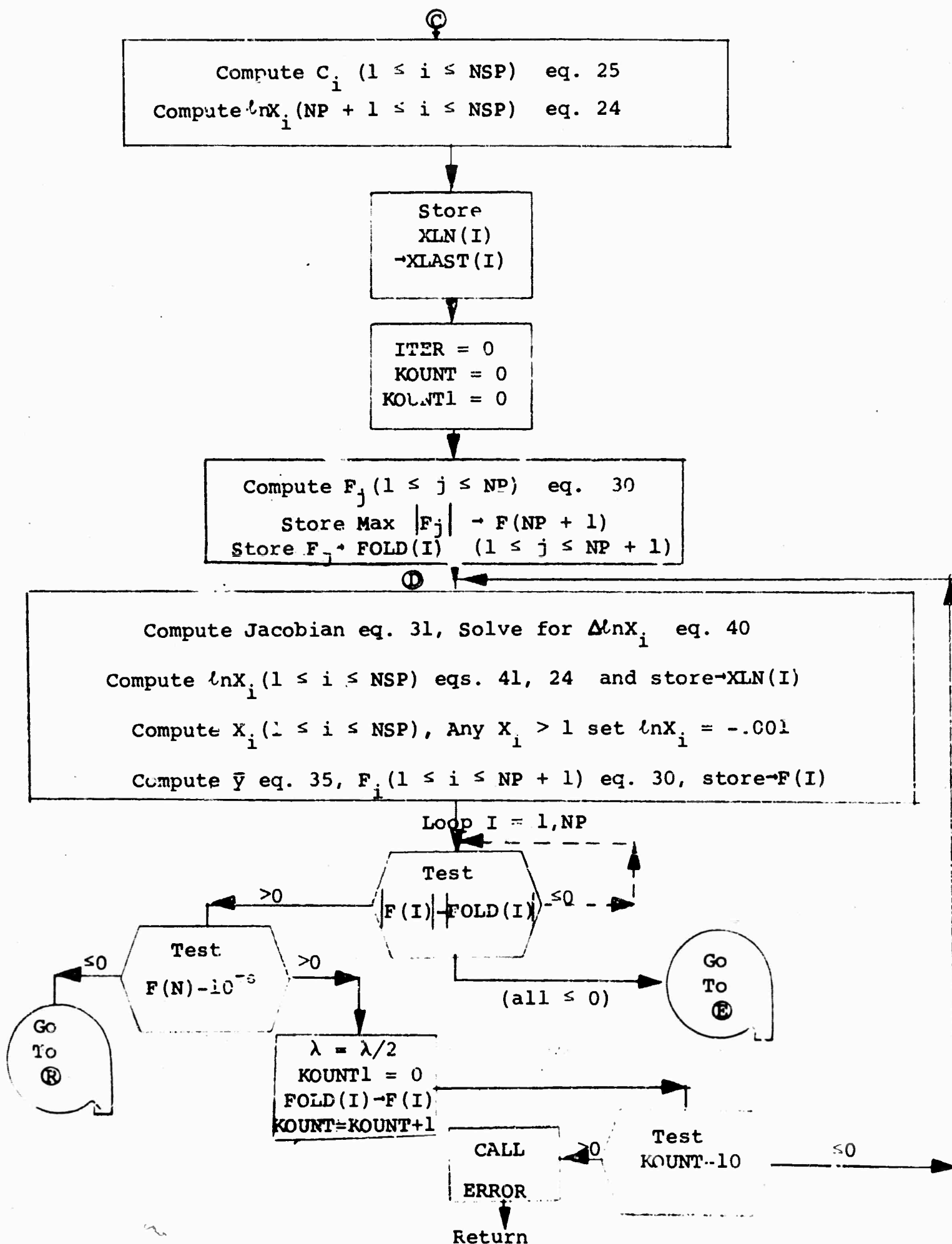
ENTRY GENR7X

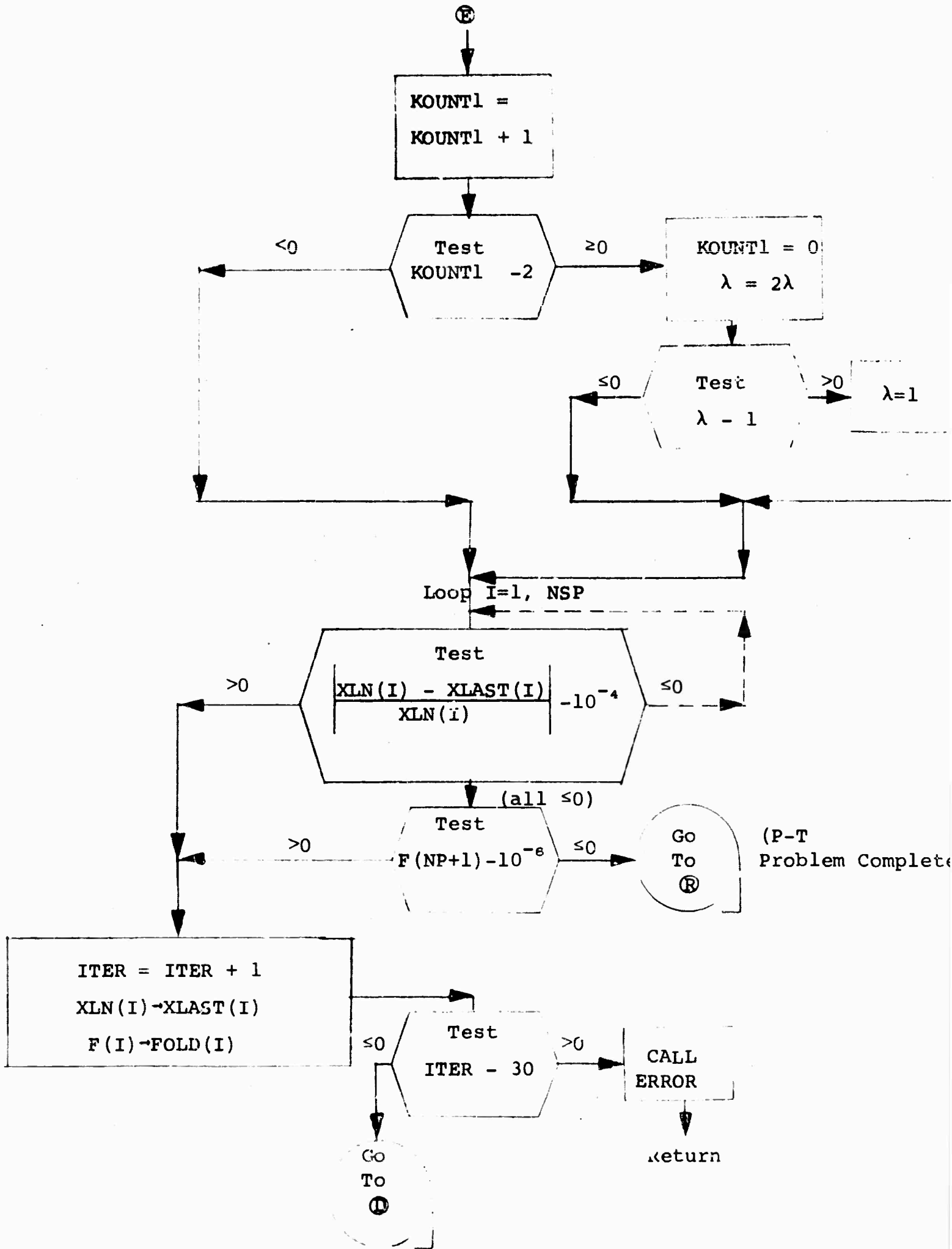






FLOW CHART FOR P,T PROBLEM





APPENDIX B

LISTING OF A
GENERATED PROGRAM



ete)

\$IBFTC GEN7X. FULIST

SUBROUTINE GEN7X(X,PROP,PR2,T,P,KOPT,BEE)
 DIMENSION AI(4, 4),X(9),BETA(4, 4),A(9, 4),F(5),Z(8),
 IXLN(8),C(8),XLAST(8),DLNX(4),PSI(4, 4, 4),
 ZDENOM(2),TEMP(4),TEMP1(4),AT(4, 5),FOLD(5),
 3W(8),EM(8),CP(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 NO
- C 7 NO+
- C 8 E

DATA W /

1 2.798125CE C3, 3.1960308E C3, 1.6320951E C8, 2.2416383E C3,
 2 6.3920616E C3, 2.983871CE C3, 2.5838710E C3, 5.5962500E C3/

DATA EM /

1 3.200000CE C1, 2.801600CE C1, 5.4862000E-04, 3.9944000E C1,
 2 1.400800CE C1, 3.000800CE C1, 3.0008000E C1, 1.6000000E C1/

DATA A /

1 2.000000CE C0, 0. , 0. , 0. ,
 2 0. , 1.000000CE C0, 1.0000000E C0, 1.0000000E C0,
 3 0. , 0. , 2.0000000E C0, 0. ,
 4 0. , 1.0000000E C0, 1.0000000E C0, 1.0000000E C0,
 5 0. , 0. , 0. , 0. ,
 6 1.000000CE C0, 0. , 0. , -0. ,
 7 -1.000000CE C0, 0. , 0. , 0. ,
 8 0. , 0. , 1.0000000E C0, 0. ,
 9 0. , 0. , 0. , 0. /

DATA BETA /

1-0. , 5.0000000E-01, 5.0000000E-01, 5.0000000E-01,
 2 5.0000000E-01, 5.0000000E-01, 5.0000000E-01, 0. ,
 3 0. , 0. , -1.0000000E C0, 0. ,
 4 0. , 0. , 0. , 0. /

DATA(CP (1,1,1),I= 1, 76)/

1 9.3716999E C3, 1.0730000E C4, 4.0757999E C8, 5.5925999E C3,
 2 1.5922000E C4, 1.0173000E C4, 1.0015000E C4, 1.6188000E C4,
 3 1.1342000E C4, 1.3349000E C4, 4.0757999E C8, 1.7240000E C3,
 4 1.1802000E C4, 9.3989999E C3, 1.2823000E C4, 1.4154000E C4,
 5 1.2575000E C4, 1.3870000E C4, 4.0757999E C8, -1.2845000E C3,
 6 2.0150000E C4, 1.2450000E C4, 6.8981000E C3, 1.3418000E C4,
 7 1.6108000E C4, 6.2157000E C3, 4.0757999E C8, -1.3009000E C4,
 8 -1.5187000E C4, 1.2587000E C4, 1.1969000E C3, 1.4487000E C3,
 9 1.7652000E C4, 2.6283000E C3, 4.0757999E C8, -2.4883000E C4,
 1 2.2241000E C4, 1.6225000E C4, 2.2221000E C3, -1.8841000E C4,
 2 2.0258000E C4, 1.2720000E C4, 4.0757999E C8, -5.1119000E C4,
 3 5.5651000E C4, 2.2964000E C4, 2.2831000E C4, 7.5617000E C3,
 4 2.3085000E C0, 1.7330000E C0, 0. , 2.7274000E-03,
 5 5.1045000E-02, 1.3905000E C0, 1.5749000E C0, -4.1353000E C0,
 6 6.6608999E-01, 2.7776000E-01, 0. , 3.5441000E-01,
 7 1.2995000E C0, 2.6811000E C0, 1.7045000E-01, -5.7238000E-02,
 8 3.8589000E-01, 1.3701000E-01, 0. , 6.0001000E-01,
 9 3.1748000E-01, 3.3385000E-01, 1.0169000E C0, 2.2711000E-01,
 1 -2.0176000E-02, 1.0375000E C0, 0. , 1.4685000E C0/

DATA(CP (1,1,1),I= 77, 152)/

1 2.526000CE C0, 2.8819000E-01, 1.6171000E C0, 1.0385000E C0,
 2 -1.4887000E-01, 1.3792000E C0, 0. , 2.2874000E C0,

3 9.9835000E-01, 4.9663000E-02, 1.5316000E 00, 2.1816000E 00,
 4-3.1303000E-C1, 7.4846000E-C1, 0. , 3.9532000E 00,
 5-2.3901000E-01,-2.8729000E-01, 1.5776000E-01, 1.1508000E 00,
 6 1.4875000E C4, 1.1777000E 04,-1.9130000E C9, 9.8261999E 03,
 7 2.6945000E C4, 1.7289000E C4, 1.3522000E C4, 1.7256000E 04,
 8 2.8742000E 03,-5.2383000E C3,-1.9130000E C9, 4.1957000E C4,
 9 5.6203000E C4, 2.1468000E 04,-5.0189000E 03, 2.7862000E C4,
 1-6.2367000E C3,-8.9963000E C3,-1.9130000E C9, 6.7269000E C4,
 2-1.0975000E 04, 2.6405000E 03, 4.1520000E C4, 3.2884000E C4,
 3-3.4750000E C4, 5.2605000E 04,-1.9130000E 09, 1.6705000E 05,
 4 2.9576000E C5, 1.6807000E C3, 8.8036000E 04, 1.3581000E 05,
 5-4.7711000E C4, 8.2233000E C4,-1.9130000E C9, 2.6895000E C5,
 6-4.5074000E 04,-2.9719000E 04, 7.9432999E 04, 3.1404000E 05,
 7-7.0111000E 04,-5.3676000E 03,-1.9130000E C9, 4.9628000E C5,
 8-3.5256000E C5,-8.9720999E 04,-9.8128999E 04, 6.8365000E 04,
 9 3.0346000E C4, 3.6566000E C4, 0. , 9.1752999E 02,
 1 3.6174000E C8, 3.2255000E 07, 3.5402000E C8, 1.6589000E C8/
 DATA(CP (1,1,1),I= 153, 228)/

1-1.1501000E C6,-2.3211000E C6, 0. , 2.1278000E C7,
 2 3.6854000E C8, 3.2487000E C7, 3.5121000E C8, 1.6640000E 08,
 3-3.8614000E C6,-3.2845000E C6, 0. , 3.9705000E C7,
 4 3.3307000E C8, 3.0504000E 07, 3.7195000E 08, 1.6732000E 08,
 5-1.9227000E C7, 2.9248000E C7, 0. , 1.1884000E C8,
 6 6.1576999E 08, 3.0298000E C7, 3.9904000E C8, 2.5559000E C8,
 7-2.8490000E C7, 4.8081999E 07, 0. , 2.0493000E 08,
 8 1.5726000E C8, 2.5620000E 06, 3.9288000E C8, 4.3566999E 08,
 9-4.8845000E C7,-3.2656000E C7, 0. , 4.1154000E C8,
 1-2.9376000E C8,-6.4829000E 07, 2.3832000E C8, 8.7229999E C7,
 2 1.2500000E 03, 1.8000000E 03, 3.0000000E C4, 1.1000000E C4,
 3 3.3000000E 03, 6.0000000E 02, 2.0000000E C3, 5.0000000E C2,
 4 4.5000000E 03, 3.7000000E 03, 3.0000000E C4, 1.2250000E C4,
 5 8.5000000E 03, 1.3000000E C3, 7.0000000E 03, 2.5000000E 03,
 6 8.8000000E C3, 8.5000000E 03, 3.0000000E 04, 1.3500000E C4,
 7 1.6000000E C4, 3.0000000E 03, 9.5000000E 03, 1.4750000E C4,
 8 1.2250000E C4, 1.0500000E 04, 3.0000000E C4, 1.4500000E C4,
 9 2.4500000E C4, 1.5250000E C4, 1.2000000E C4, 1.7750000E C4,
 1 1.6000000E 04, 1.6000000E 04, 3.0000000E 04, 1.5750000E C4/
 DATA(CP (1,1,1),I= 229, 240)/

1 2.7000000E C4, 2.0000000E C4, 1.5000000E 04, 2.6000000E C4,
 2 3.0000000E C4, 1.8000000E 04, 3.0000000E C4, 1.7500000E C4,
 3 3.0000000E C4, 3.0000000E 04, 1.7500000E C4, 3.0000000E C4/
 DATA NP,NSP,M,N,NSP1,NNN/ 4, 8, 4, 5, 9, 6/
 DATA PSI /

1-0. , 5.0000000E-01, 5.0000000E-01, 5.0000000E-C1,
 2-C. , 5.0000000E-01, 5.0000000E-01, 0. ,
 3-C. , 0. , -5.0000000E-01, 0. ,
 4-C. , 0. , 0. , 0. ,
 5 C. , 5.0000000E-01, 5.0000000E-01, 0. ,
 6 5.0000000E-01, 5.0000000E-01, 5.0000000E-C1, 0. ,
 7 C. , 0. , -5.0000000E-01, 0. ,
 8 C. , 0. , 0. , 0. ,
 9 C. , 0. , -1.0000000E 00, 0. ,
 1 C. , 0. , -1.0000000E 00, 0. ,
 2 0. , 0. , 1.0000000E 00, 0. ,
 3 C. , 0. , -0. , 0. ,
 4 0. , 0. , 0. , 0. ,
 5 0. , 0. , 0. , 0. ,
 6 C. , 0. , -0. , 0. ,
 7 C. , 0. , 0. , 0. ,

C	MINIMIZE FREE ENERGY	1
C	NP ARE THE NUMBER OF ELEMENTS	2
C	NSP ARE THE NUMBER OF SPECIES	3
C	T IS A GUESS FOR KCPT=2,3	4
C	KCPT=1 MEANS P,T GIVEN	5
C	KCPT=2 MEANS P,F GIVEN, PROP IS ENTHALPY	6
C	KCPT=3 MEANS P, S GIVEN, PROP IS ENTROPY	7
C	X(I) ARE MOLE FRACTIONS, YBAR=TOTAL INITIAL MOLES	8
	X(NSP1)=1.	9
	DO 23 I=1, NP	10
	A(NSP1, I)=0.	11
	DO 23 J=1, NSP	12
23	A(NSP1, I)=A(NSP1, I)+A(J, I)*X(J)*X(NSP1)	13
	RETURN	14
	ENTRY GEAR7X	15
	IF(KCPT.GT.0)GO TO 26	16
	KCPT=-KCPT	17
	DO 29 I=1, NP	18
29	A(NSP1, I)=BEE(I)	19
C	26 WRITE(6,11)(X(I), I=1, NP)	20
26	CONTINUE	21
	DO 20 I=1, NP	22
	IF(X(I).LE.0.)X(I)=.001	23
20	XLN(I)=ALOG(X(I))	24
	IPH=1	25
C	2 FORMAT(3E10.6, I10)	26
C	WRITE(6,13)T,P,PRCP,KCPT	27
C	13 FORMAT(3E20.8, I10)	28
C	INITIALIZE LAMDA	29
	ZLAMDA=1.	30
	PLCG=ALOG(P)	31
1999	GO TO(2007,2000,2000) ,KCPT	32
2000	IF(IPH.EQ.1)GO TO 2007	33
	LCPT=KCPT	34
C	WILL RETURN IN ETCH THE MIXTURE ENTHALPY FOR KCPT	35
C	I=2,ENTROPY KCPT=3	36
	GO TO 5200	37
C2164	WRITE(6,11)ETCH,T	38
2164	CONTINUE	39
	IF(ABS((PRCP-ETCH)/PRCP).LT..000001)GO TO 300	40
	FN2=FCLDH	41
	FCLDH=FNEW+	42
	FNEW=PRCP-ETCH	43
	IF(IPH.NE.2)GO TO 2002	44
	IN2=TCLD	45
	TCLD=T	46
	IF(FNEW)2003,300,2004	47
2003	T=.95*T	48
	GO TO 2007	49
2004	T=1.05*T	50
	GO TO 2007	51
2002	IF(IPH.NE.3)GO TO 2008	52
3010	FPRIME=(FNEW-FCLDH)/(T-TCLD)	53
	IN2=TCLD	54
	TCLD=T	55
	T=T-FNEW/FPRIME	56
	GO TO 2007	57
2008	HN=T-TCLD	58
	HN1=TCLD-IN2	59

1					60
2		GN=FN/HN1			61
3		AM=GN*(FNEW1-(1.+GN)*FOLDH+GN*FN2)			62
4		BM=(2.*GN+1.)*FNEW1-FOLDH*(1.+GN)**2+GN*GN*FN2			63
5		CM=FNEW1*(1.+GN)			64
6		CCN=BM*BM-4.*AM*CM			65
7		IF(CCN.LT.0.)GC TC 3010			66
8		CCN=SQRT(CCN)			67
9		DENCM(1)=BM+CCN			68
10		DENCM(2)=BM-CCN			69
11		IF(ABS(DENCM(1))-ABS(DENCM(2)))3002,3003,3004			70
12	3002	BCTTCM=DENCM(2)			71
13		GC TO 3005			72
14	3004	BCTTCM=DENCM(1)			73
15	3005	TN2=TCLC			74
16		TCLC=T			75
17		T=T-HN*2.*CM/BCTTCM			76
18	2007	LCPT=1			77
19		GC TC 5407			78
20	2163	KKK=3			79
21		GC TC 4200			80
22	4101	CONTINUE			81
23	C4101	WRITE(6,11)ETCF,FNEW1,FPRIME,T,FCLDH,TCLC,X			82
24	2001	DO 12 I=1,NSP			83
25	12	XLAST(I)=XLN(I)			84
26	40	ITER=C			85
27		KCUNT=C			86
28		KKK=1			87
29	64	KCUNT1=0			88
30		LLL=3			89
31		GC TC 600			90
32	210	DC 211 I=1,N			91
33	211	FCLD(I)=F(I)			92
34	61	GC TC(4017,4020),KKK			93
35	4102	LLL=2			94
36		GC TC 600			95
37	C4500	WRITE(6,11)ZLAMDA,FOLD			96
38	4500	CONTINUE			97
39		IF(F(N)-FCLD(N))70,70,71			98
40	71	IF(F(N).LT..00001)GC TO 85			99
41		ZLAMDA=ZLAMDA/2.			100
42		KCUNT1=C			101
43		KCUNT=KCUNT+1			102
44		KKK=2			103
45		DC 75 I=1,NP			104
46		F(I)=FCLD(I)			105
47	75	XLN(I)=XLAST(I)			106
48		F(N)=FCLD(N)			107
49		IF(KCUNT-10)61,61,72			108
50	72	CALL ERRCR(QX1,KRUM,13)			109
51	74	GC TC 305			110
52	70	KKK=1			111
53	81	KCUNT1=KCUNT1+1			112
54		IF(KCUNT1-2)85,90,90			113
55	90	ZLAMDA=2.*ZLAMDA			114
56		IF(ZLAMDA.GT.1.)ZLAMDA=1.			115
57		KCUNT1=0			116
58	85	DC 86 I=1,NSP			117
59		IF(ABS((XLN(I)-XLAST(I))/XLN(I))-0.0001)86,86,91			118
	86	CONTINUE			

```

      IF(F(N)-.000001)99,99,91
91 ITER=ITER+1
C WRITE(6,11)X
C SAVE LNX(I),I=1,NSP AND F(I),I=1,N
      DC 89 I=1,NSP
89 XLAST(I)=XLN(I)
      DC 189 I=1,N
189 FLLC(I)=F(I)
      IF(ITER-30)61,92,92
92 CALL ERRCR(QX1,KRUM,12)
99 CONTINUE
C 99 WRITE(6,11)ZLAMDA,X,XLN,XLAST
      IF(KCPT.EQ.1)GC TC 300
      IPF=IPF+1
      IF(IPF-10)1999,1999,1998
1998 CALL ERRCR(QX1,KRUM,11)
      11 FORMAT(5E20.8)
300 LCPT=KCPT+3
      GC TC 5200
300 PRCP=ETCH
      LCPT=7
      GC TC 5407
307 LCPT=4
305 KCPT=LCPT-3
      PR2=ETCH
      RETURN
4017 DC 4001 J=1,NP
      A1(1,J)=0.
      DC 4002 I=1,M
      II=NP+1
4002 A1(1,J)=A1(1,J)+X(II)*BETA(1,J)
4001 A1(1,J)=X(J)+A1(1,J)
C LF MATRIX IN A1 ARRAY
C GET L(1/YEAR)/DLNX(L) IN TEMP(L)
      DC 4004 L=1,NP
      TEMP(L)=0.
      DC 4003 J=N,NSP
      JJ=J-NP
4003 TEMP(L)=TEMP(L)+PSI(JJ,1,L)*X(J)
4004 TEMP(L)=(TEMP(L)+X(L)*A(L,1))/A(NSP1,1)
C NOW FORM DF(I)/DLNX(L) FOR I=2,NP AND L=1,NP
      DC 4006 I=2,NP
      DC 4006 L=1,NP
      TEMP1(L)=0.
      DC 4005 J=N,NSP
      JJ=J-NP
4005 TEMP1(L)=TEMP1(L)+PSI(JJ,1,L)*X(J)
4006 A1(1,L)=A(L,1)*X(L)+TEMP1(L)-A(NSP1,1)*TEMP(L)
      LLL=1
4007 CALL CLEM7X(A1,DLNX,F,NP,N,AT)
4020 DC 4021 I=1,NP
4021 XLN(I)=XLN(I)-ZLAMDA*DLNX(I)
4200 DC 4201 L=1,NP
4201 TEMP1(L)=XLN(L)+C(L)
      DC 4025 I=N,NSP
      XLN(I)=C.
      JJ=I-NP
      DC 4031 L=1,NP
4031 XLN(I)=XLN(I)+TEMP1(L)*BETA(JJ,L)

```


119	4025	XLN(I)=XLN(I)-C(I)	178
120		DC 4022 I=1,NSP	179
121		IF(XLN(I))4022,4022,4023	180
122	C	CCORRECT ANY MOLE FRACTIONS LARGER THAN 1	181
123	4023	XLN(I)=-.01	182
124	4022	CONTINUE	183
125	C	CCOMPUTE X(I) AND YBAR	184
126		X(NSP1)=0.	185
127		DC 4050 J=1,NSP	186
128		X(J)=EXP(XLN(J))	187
129		IF(ABS(XLN(J)).GT.68.)X(J)=0.	188
130	4050	X(NSP1)=X(NSP1)+A(J,1)*X(J)	189
131		X(NSP1)=A(NSP1,1)/X(NSP1)	190
132		GC TC(4102,4102,4101),KKK	191
133	C	CCOMPUTES F FOR NEWTON RAPHSON	192
134	C	ALSO STORES LARGEST F IN F(N)	193
135	600	DUM=C.	194
136		DC 601 I=1,NSP	195
137	601	DUM=DUM+X(I)	196
138		F(I)=DUM-1.	197
139		DC 603 I=2,NP	198
140		DUM=C.	199
141		DC 602 J=1,NSP	200
142	602	DUM=DUM+A(J,1)*X(J)	201
143	603	F(I)=DUM-A(NSP1,1)/X(NSP1)	202
144	C	FIND THE LARGEST F(I)	203
145		F(N)=ABS(F(1))	204
146		DC 605 I=2,NP	205
147		E=ABS(F(I))	206
148		IF(E-F(N))605,604,604	207
149	604	F(N)=E	208
150	605	CONTINUE	209
151	C	WRITE(6,11)F	210
152		GC TC(4007,4500,210),LLL	211
153	C	FIRST SUBSCRIPT DEFINES SPECIE NUMBER	212
154	C	SECOND SUBSCRIPT IS FOR A RANGE IN TEMPERATURE	213
155	C	THIRD SUBSCRIPT 1=A	214
156	C	THIRD SUBSCRIPT 2=B	215
157	C	THIRD SUBSCRIPT 3=C	216
158	C	THIRD SUBSCRIPT 4=D	217
159	C	THIRD SUBSCRIPT 5 UPPER T IN TEMP. RANGE	218
160	C	S=A*LN1+B*T+C	219
161	C	H=A*T+.5*B*T*T+C	220
162	C	CONVERT MOLE TO MASS FRACTIONS	221
163	C	Z ARRAY FOR MASS FRACTIONS	222
164	5200	EMM=C.	223
165		DC 5201 I=1,NSP	224
166	5201	EMM=EMM+X(I)*EM(I)	225
167		DC 5202 I=1,NSP	226
168	5202	Z(I)=X(I)*EM(I)/EMM	227
169	C	WRITE(6,11)Z,EM,EMM	228
170	5407	TLOG=ALOG(T)	229
171		ETCH=0.	230
172		DC 5401 I=1,NSP	231
173		DC 5402 JJ=1,NNNN	232
174		IF(CP(I,JJ,5)-T)5402,5403,5403	233
175	5402	CONTINUE	234
176		CALL ERRCHK(T,I,10)	235
177	5403	GC TC(5406,5405,5404,5405,5405,5404),LCPT	236

```

5405 HI=(CP(I,JJ,1)+.5*CP(I,JJ,2)*T)*T+CP(I,JJ,4)
C   WRITE(6,113)I,JJ,HI
    ETCH=ETCH+Z(I)*HI
    GC TC 5401
5404 SI=CP(I,JJ,1)*TLOG+CP(I,JJ,2)*T+CP(I,JJ,3)
C   WRITE(6,113)I,JJ,SI
    ETCH=ETCH+X(I)*(SI/W(I)-XLN(I)-PLCG)/EMM
    GC TC 5401
5406 C(I)=CP(I,JJ,1)-CP(I,JJ,3)-.5*CP(I,JJ,2)*T+CP(I,JJ,4)/
    IT-CP(I,JJ,1)*TLOG
    C(I)=C(I)/W(I)+PLCG
C   WRITE(6,113)I,JJ,C(I)
5401 CCNTINUE
C 113 FORMAT(2I10,3E20.8)
    GC TG(2163,2164,2164,306,305,305,307),LOPT
    END
$1  SCLEM7
    SUBROUTINE CLEM7X(B,X,C,M,M1,AT)
    DIMENSION B(M,M),X(M),D(M1),AT(M,M1)
C   EQLATIONS ARE OF THE FORM BX=D
    DC 17 I=1,M
    17 X(I)=0.0
    DO 200 I=1,M
    200 AT(I,M1)=D(I)
    DO 201 I=1,M
    DO 201 J=1,M
    201 AT(I,J)=B(I,J)
    DO 32 N=1,M
    C=AT(N,N)
    IT=C
    DO 5 I=N,M
    IF(ABS(AT(I,N))-ABS(C))9,9,8
    8 C=AT(I,N)
    IT=I
    9 CCNTINUE
    IF(IT-N)7,7,7C
    7C DO 71 J=N,M1
    TEMP=AT(N,J)
    AT(N,J)=AT(IT,J)
    71 AT(IT,J)=TEMP
    7 DO 10 I=1,M1
    10 AT(N,I)=AT(N,I)/O
    IF(M-N)50,50,18
    18 N1=N+1
    DO 30 I=N1,M
    C=AT(I,N)
    DO 30 J=N,M1
    30 AT(I,J)=AT(I,J)-AT(N,J)*C
    32 CCNTINUE
    50 X(M)=AT(M,M+1)
    DO 65 N=2,M
    NR=M+1-N
    C=AT(NR,M+1)
    DO 60 I=NR,M
    60 C=C-AT(NR,I)*X(I)
    65 X(NR)=C/AT(NR,NR)
    RETURN
    END

```

237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294

APPENDIX C

LISTING OF SUGGESTED

SUBROUTINE ERROR

C-1

```
318FTC LIST  ERROR.
      SUBROUTINE ERROR(Q,K,I)
      WRITE(6,10)I
10  FORMAT(6F10.3)
      J=I-9
      GO TO(1,2,3,4),J
      1  WRITE(6,11)Q,K
11  FORMAT(14F0TEMPERATURE =E15.7,19FFIT FOR SPECIE NO. 14,9FNCT VALIL
      1)
      GO TO 7
      2  WRITE(6,12)
12  FORMAT(48FOOCTER EQUILIBRIUM ITERATIONS EXCEED 10          )
      GO TO 7
      3  WRITE(6,13)
13  FORMAT(44HCPRESSURE TEMPERATURE ITERATIONS EXCEED 30      )
      GO TO 7
      4  WRITE(6,14)
14  FORMAT(19HCLAMDA CUT 10 TIMES                               )
      7  RETURN
      END
```

20 CARDS PRINTED

APPENDIX D

SAMPLE INPUT FOR THE
CHEMISTRY EQUILIBRIUM GENERATOR PROGRAM

118155										11																			
WORD 1		WORD 2		WORD 3		WORD 4		WORD 5		WORD 6		WORD 7		WORD 8															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	

STANDARD 8 WORD CARD

1

118155										11																	
WORD 1		WORD 2		WORD 3		WORD 4		WORD 5		WORD 6		WORD 7		WORD 8													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

STANDARD 8 WORD CARD

2

WORD 1		WORD 2		WORD 3		WORD 4		WORD 5		WORD 6		WORD 7		WORD 8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

TABCO 956424

STANDARD 8 WORD CARD