AD644898

AEROSPACE RESEARCH • AERODYNAMICS • PROPULSION • STRUCTURAL DYNAMICS • ELECTRONIC SYSTEMS AND INSTRUMENTS • COMPUTER MODULE

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	as

ARCHIVE COPY

December 14, 1966

DDC
RECEIVED
JAN 12 1967
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"

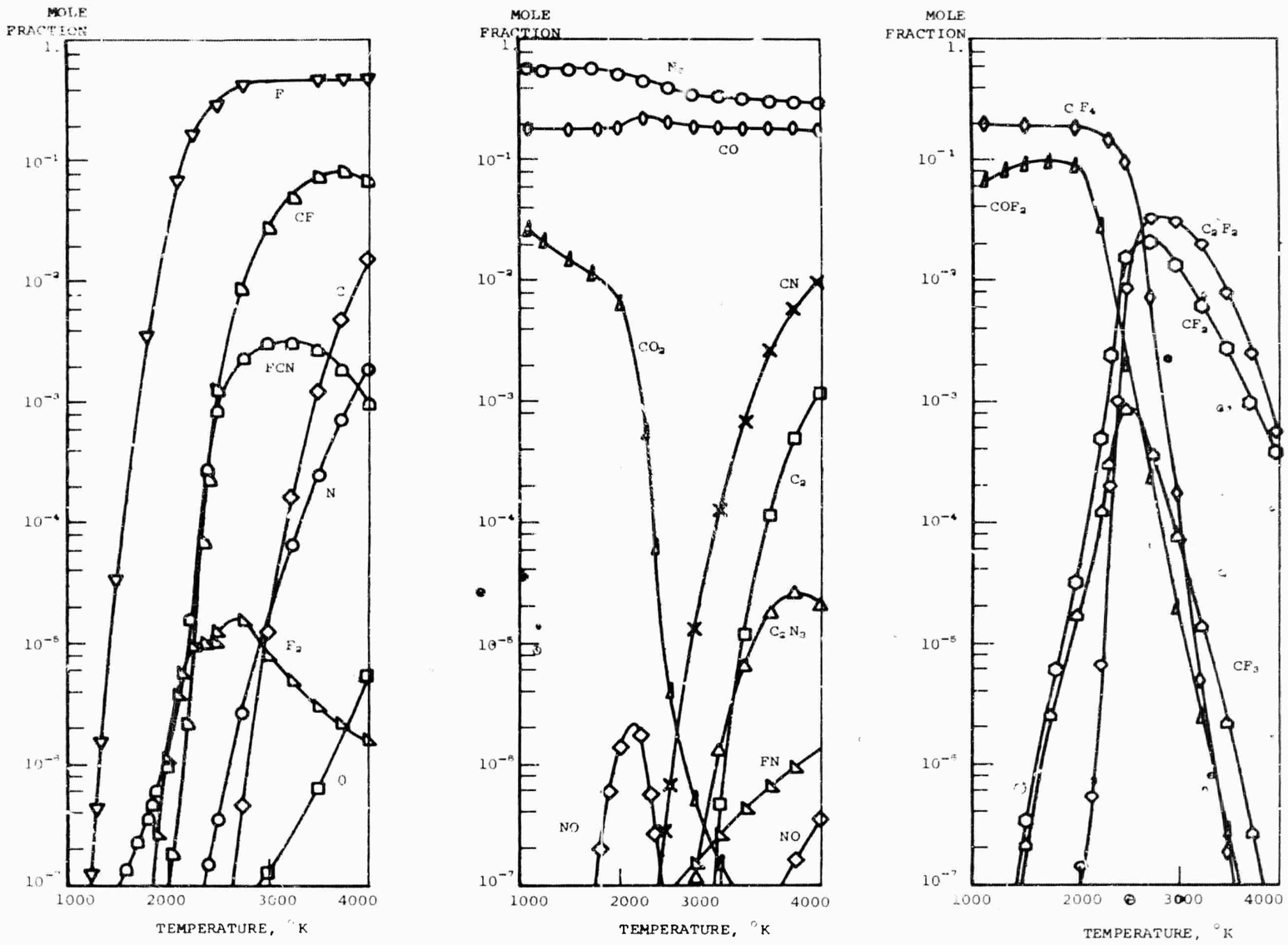
Prepared by

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
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

SECTION	TITLE	PAGE
I	INTRODUCTION	1
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 °K or 89506 ft^2/sec^2-°K lb/lb mole
$\hat{S}$	mixture entropy lb mole/lb
S	mixture entropy, cal/°K
$\bar{S}$	mixture entropy, ft^2/sec^2 °K
s_i	species entropy at standard state (p = 1 atm), cal/mole °K
$\bar{s}_i$	species entropy at standard state (p = 1 atm), ft^2/sec^2 °K
T	temperature, °K
X_i	mole fraction of species i
Y_i	number of moles of species i
$\bar{Y}$	the total moles in the system $\left(\bar{Y} = \sum_{i=1}^{NSP} Y_i\right)$

<u>Greek</u>

Φ	$\frac{G}{RT} + \sum_{j=1}^{NSP}\left[\sum_{i=1}^{NSP} Y_i a_{ij} - b_j\right]\pi_j$ [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

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.
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

$$G = H - TS \tag{1}$$

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

$$H = \sum_i Y_i h_i(T) \tag{2}$$

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

$$S - S_o = \Sigma Y_i[s_i(T) - R \ln X_i - R \ln P] \tag{3}$$

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 °K

X_i is the mole fraction of species i

S_o is the reference entropy taken as zero at

$T = 0$ °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),

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

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

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

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

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

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

$$\sum_{i=1}^{NSP} X_i = 1 \tag{7}$$

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

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

where π_j are the Lagrange multipliers.

Let

$$C_i = \frac{1}{R}\left[\frac{h_i}{T} - s_i\right] + \ell n\ P \tag{9}$$

Then from Eq. (4) and (9)

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

and

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

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

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]:

$$Y_1 \frac{\partial}{\partial Y_k}\left(\ell n \frac{Y_1}{\bar{Y}}\right) + Y_2 \frac{\partial}{\partial Y_k}\left(\ell n \frac{Y_2}{\bar{Y}}\right) + \cdots Y_{NSP} \frac{\partial}{Y_k}\left(\ell n \frac{Y_{NSP}}{\bar{Y}}\right) \tag{12}$$

Note that:

$$Y_\ell \frac{\partial}{\partial Y_k}\left(\ell n \frac{Y_\ell}{\bar{Y}}\right) = \frac{\bar{Y} - Y_\ell}{\bar{Y}} \quad \text{for } \ell = k$$

$$= -\frac{Y_\ell}{\bar{Y}} \quad \text{for } \ell \neq k \tag{13}$$

From which it can be seen that

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

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

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

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

$$g_i = - C_i - \ell n\ X_i = - \left\{\frac{1}{R}\left[\frac{h_i}{T} - s_i\right] + \ell n\ P + \ell n\ X_i\right\} \tag{16}$$

then Eq. (15) can be written as

$$\sum_{j=1}^{NP} a_{ij}\pi_j = g_i \qquad i=1,2,\ldots,NSP \tag{17}$$

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

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

or

$$A\,\overline{\Pi} = \overline{G} \tag{19}$$

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 partioned accordingly, the result is

$$\begin{array}{c} \text{NP} \\ \\ \text{NSP-NP} \end{array} \left\updownarrow \left[\begin{array}{c} A_1 \\ \hline A_2 \end{array}\right] [\overline{\Pi}] = \left[\begin{array}{c} G_1 \\ \hline G_2 \end{array}\right] \right\updownarrow \begin{array}{c} \text{NP} \\ \\ \text{NSP-NP} \end{array} \tag{20}$$

or

$$A_1\,\overline{\Pi} = \overline{G}_1 \tag{21}$$

and

$$A_2\,\overline{\Pi} = \overline{G}_2 \tag{22}$$

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

$$A_2 A_1^{-1}\overline{G}_1 = \overline{G}_2 \tag{23}$$

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

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

$$n = NP + 1, NP + 2, \ldots, NSP$$

where

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

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

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

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

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

$$\overline{Y} = \sum_{i=1}^{NSP} Y_i \tag{28}$$

$$X_i = \frac{Y_i}{\overline{Y}} \tag{29}$$

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

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

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

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

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

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

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

$$\left.\begin{aligned} \frac{\partial \ln X_j}{\partial \ln X_\ell} &= \delta_{j\ell} \\ \frac{\partial X_j}{\partial \ln X_\ell} &= X_j \delta_{j\ell} \end{aligned}\right\} \qquad \left\{\begin{aligned} 1 \le j \le NP \\ 1 \le \ell \le NP \end{aligned}\right\} \tag{32}$$

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

$$\delta_{j\ell} = \left\{\begin{aligned} 1 \text{ for } j = \ell \\ 0 \text{ for } j \neq \ell \end{aligned}\right\} \tag{33}$$

From Eq. (24) for $(NP + 1 \le n \le NSP)$, $(1 \le \ell \le NP)$

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

and

$$\frac{\partial X_n}{\partial \ln X_\ell} = X_n \beta_{n-NP,\ell} \tag{34}$$

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

From Eq. (27)

$$\frac{1}{\bar{Y}} = \frac{1}{b_1} \sum_{i=1}^{NSP} X_i a_{ij}, \tag{35}$$

then

$$\frac{\partial\left(\frac{1}{\bar{Y}}\right)}{\partial\ \ell n\ X_\ell} = \frac{1}{b_1}\left[X_\ell a_{\ell 1} + \sum_{n=NP+1}^{NSP} X_j a_{n1} \beta_{n-NP,\ell}\right] . \qquad (36)$$

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:

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

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

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

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

and the set of correction terms denoted by the column vector

$$\Delta = \begin{bmatrix} \Delta \ln X_1 \\ \Delta \ln X_2 \\ \vdots \\ \Delta \ln X_{NP} \end{bmatrix} \tag{39}$$

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

$$(\ln X_j)_{N+1} = (\ln X_j)_N - \lambda(\Delta \ln X_j)_N \qquad j=1,\ldots,NP \tag{41}$$

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)

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

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

(2)

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

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

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 (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, $j=1,2,\ldots,NP$, 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_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 $\overline{Y}$ (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, $\overline{H}_1$ (or entropy $\hat{S}_1$). A second guess of the temperature is obtained (for enthalpy):

$$T_2 = T_1 \pm 0.05\ T_1 \quad \text{for} \quad \overline{H}_1 \lessgtr \overline{H}$$

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

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

$\overline{S}$ is the entropy per unit mass, $\frac{ft^2}{sec^2 - {}^\circ K}$.

The program then calculates $\overline{H}_2$ 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:

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

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

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

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 $\left|\frac{\overline{H}_k - \overline{H}}{\overline{H}}\right| \leq 10^{-6}$

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,PR2,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 ^{o}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).

PROP 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,...,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,...,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.

KOPT	PROP	PR2
± 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-$\overline{H}$ (or P-$\hat{S}$) 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:

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

$$\bar{h}_i = \bar{h}_{i_o} + \int_o^T c_{p_i}\, dT = a_i T + \frac{b_i T^2}{2} + d_i \qquad (44)$$

$$\bar{s}_i = \bar{s}_{i_o} + \int_{T_{o_i}}^T \frac{c_{p_i}\, dT}{T} = a_i \ln T + b_i T + c_i \quad \left(T > T_{o_i}\right)$$

where $\bar{c}_{p_i}$ is the species specific heat at constant pressure, $ft^2/sec^2\ {}^\circ K$

$\bar{h}_i$ is the species enthalpy, ft^2/sec^2 (note that species sensible enthalpy is taken to be zero at $0\ {}^\circ K$)

$\bar{h}_{i_o}$ is the species heat of formation at $0\ {}^\circ K$, ft^2/sec^2

$\bar{s}_i$ is the species entropy, $ft^2/sec^2\ {}^\circ K$ (note $\bar{s}_i$ is taken to be zero at $0\ {}^\circ K$, $p = 1$ atmosphere)

T_{o_i} 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

CARD NO.	COLUMNS	DATA TO BE PUNCHED ON CARD
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 $200^{o}K$.

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

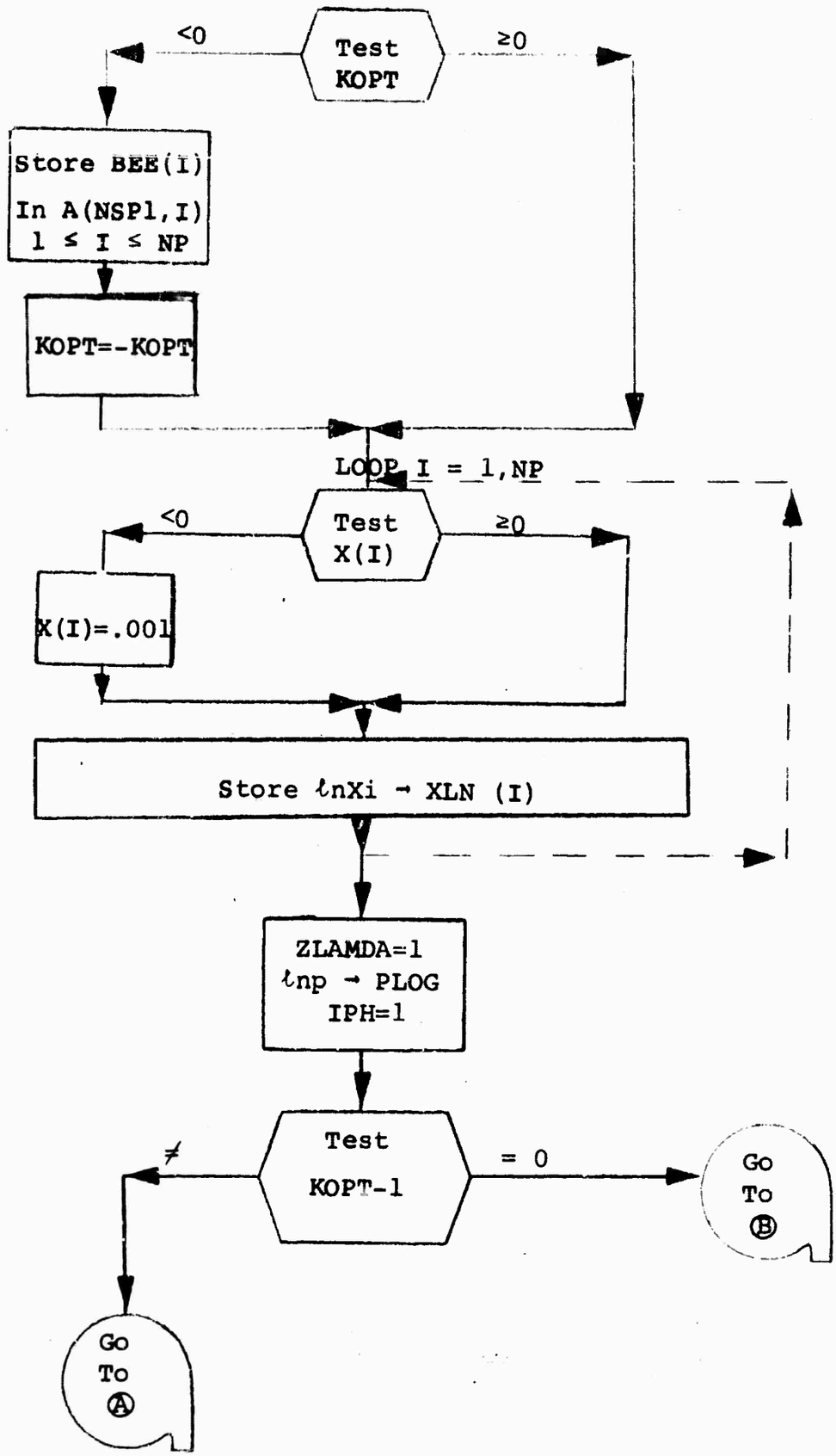
FLOW CHARTS

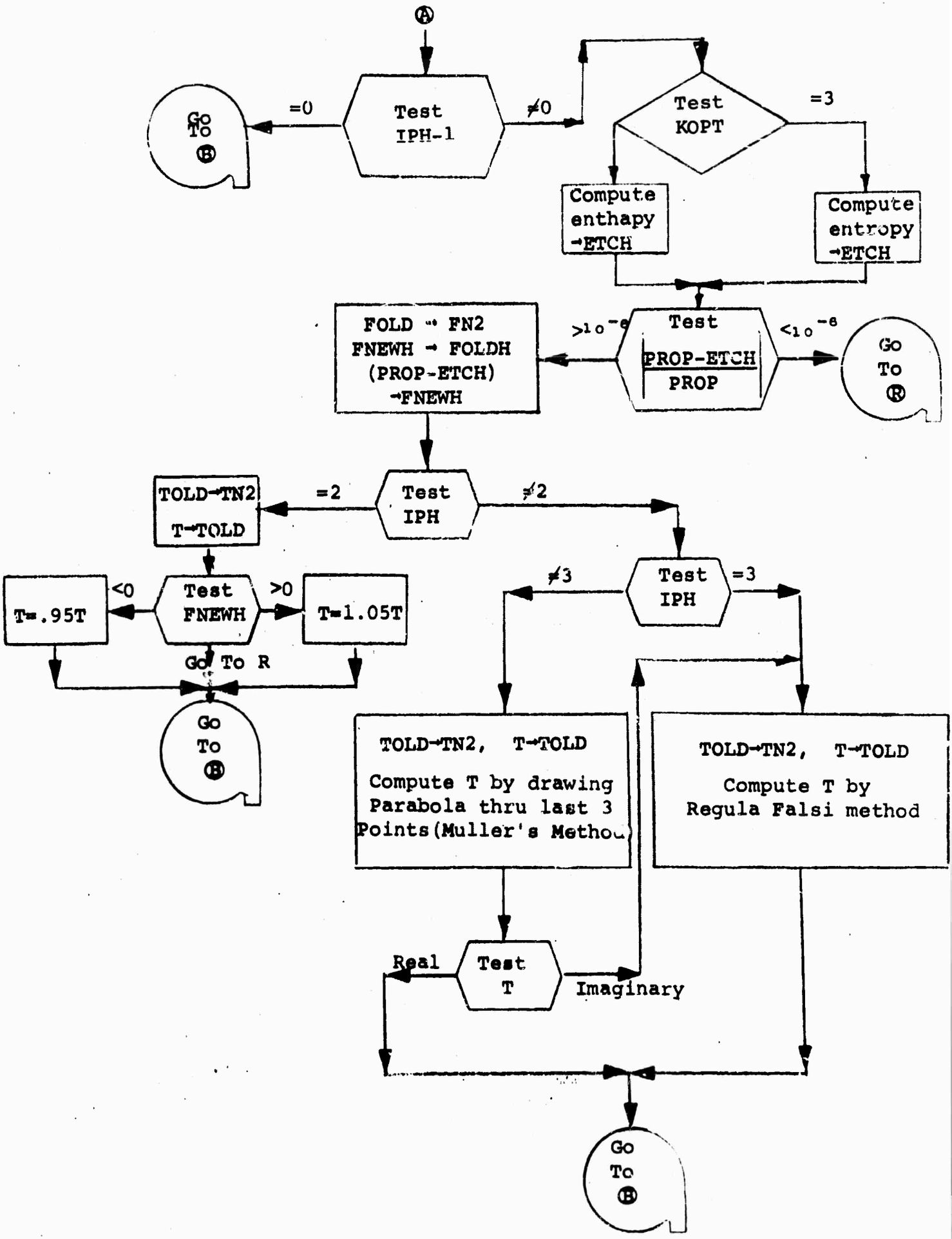
OF THE

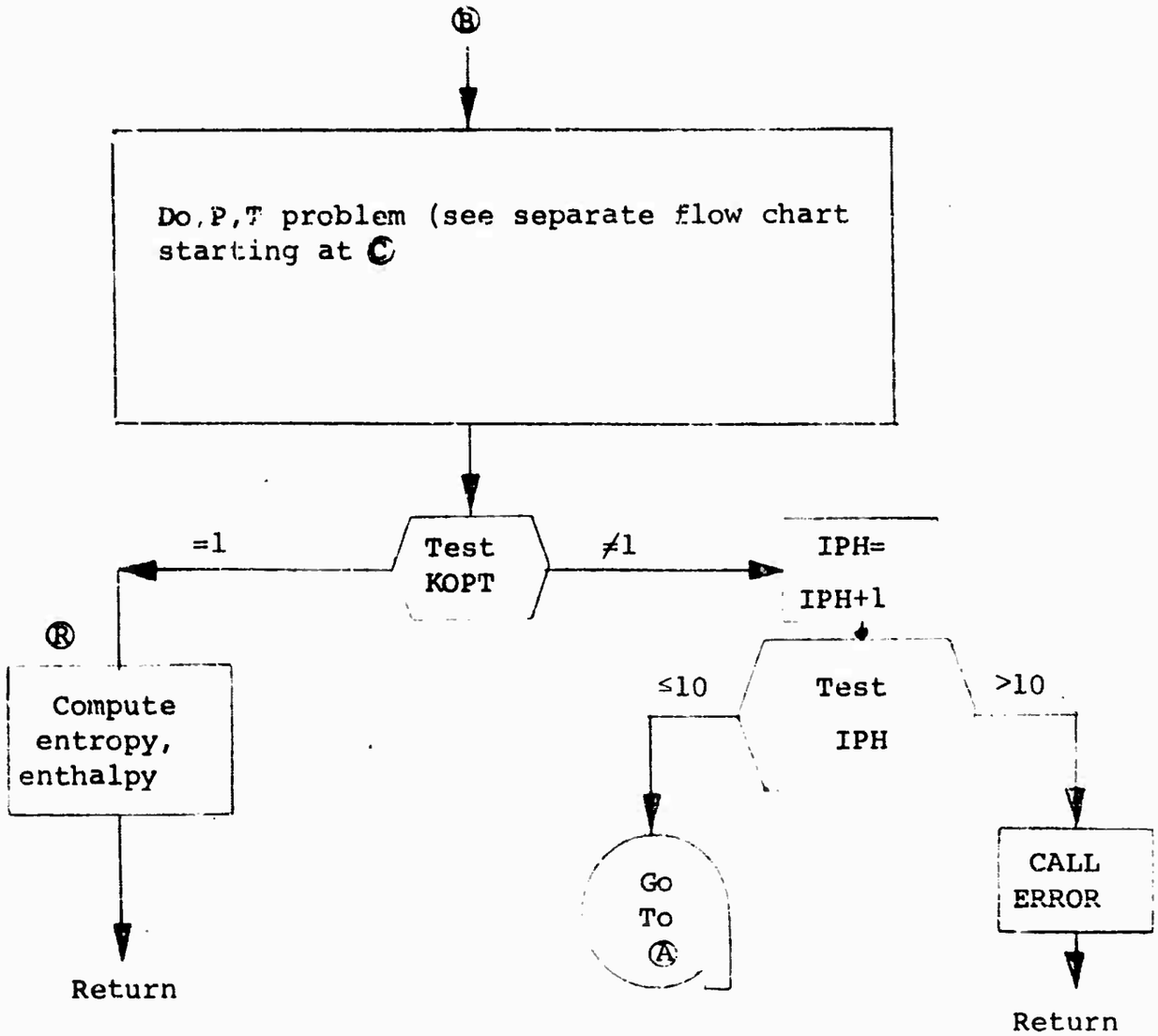
EQUILIBRIUM PROGRAM

FLOW CHART FOR SUBROUTINE GENL7X

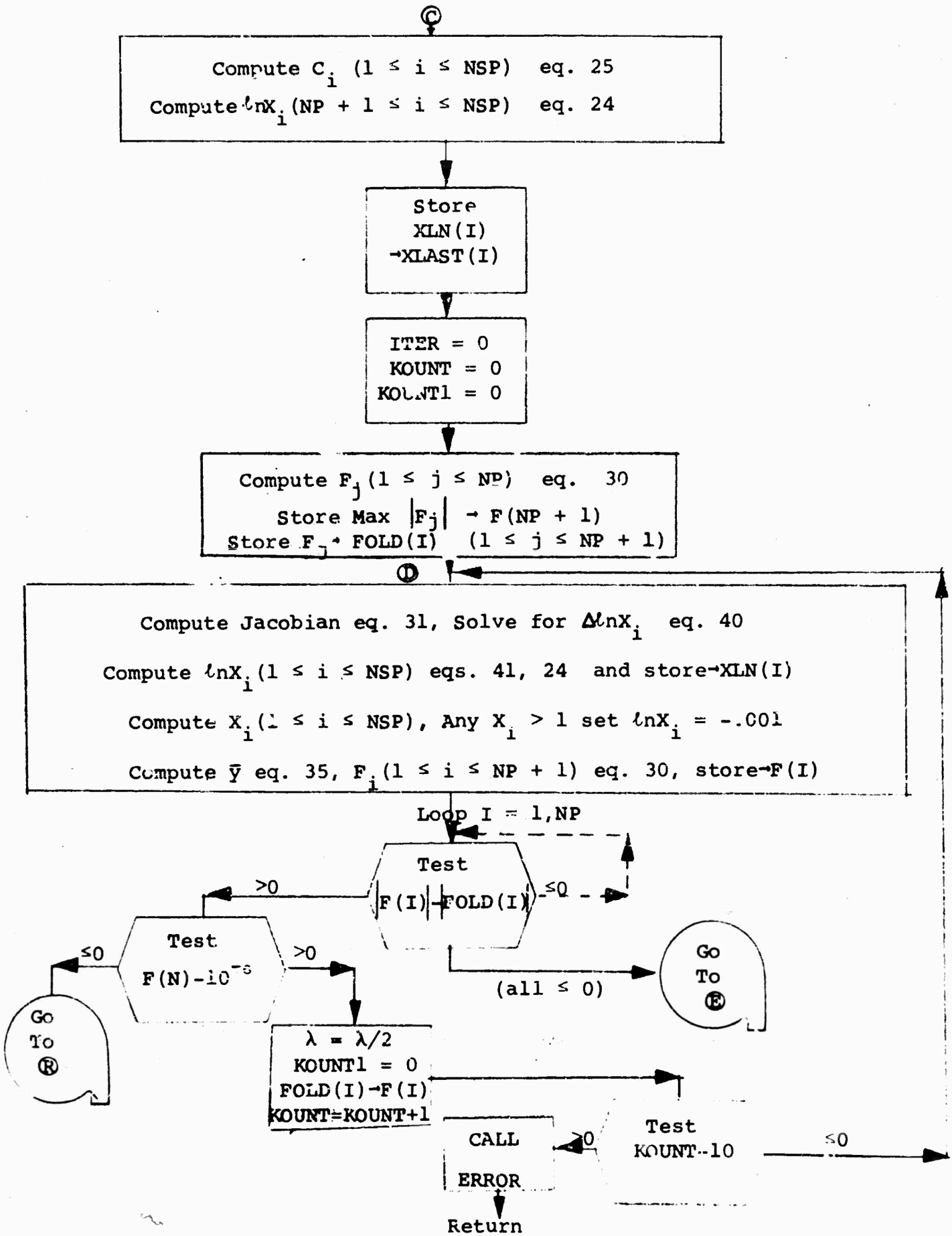
ENTRY GENR7X

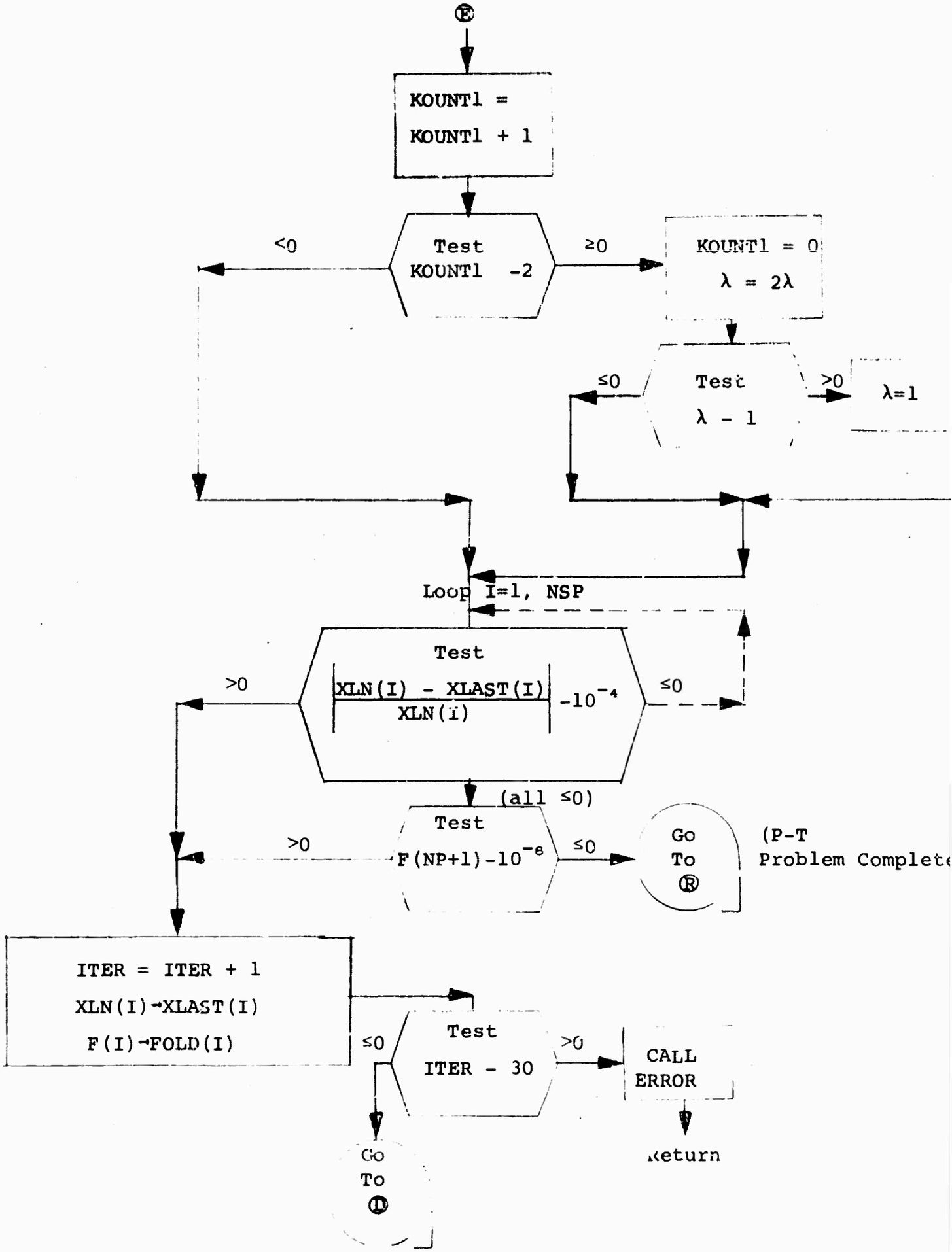



A
Test IPH-1
=0
Go To B
≠0
Test KOPT
=3
Compute enthapy →ETCH
Compute entropy →ETCH
Test PROP-ETCH / PROP
>10^-6
<10^-6
Go To B
FOLD → FN2
FNEWH → FOLDH
(PROP-ETCH)
→FNEWH
Test IPH
=2
≠2
TOLD→TN2
T→TOLD
Test FNEWH
<0
>0
T=.95T
T=1.05T
Go To R
Go To B
Test IPH
≠3
=3
TOLD→TN2, T→TOLD
Compute T by drawing Parabola thru last 3 Points (Muller's Method)
TOLD→TN2, T→TOLD
Compute T by Regula Falsi method
Test T
Real
Imaginary
Go To B


Ⓑ
Do,P,T problem (see separate flow chart starting at Ⓒ
Test KOPT
=1
≠1
IPH= IPH+1
Ⓡ
Compute entropy, enthalpy
Return
Test IPH
≤10
>10
Go To Ⓐ
CALL ERROR
Return

FLOW CHART FOR P,T PROBLEM




Ⓔ
KOUNT1 =
KOUNT1 + 1
Test
KOUNT1 -2
<0
≥0
KOUNT1 = 0
λ = 2λ
Test
λ - 1
≤0
>0
λ=1
Loop I=1, NSP
Test
XLN(I) - XLAST(I)
XLN(I)
-10^-4
>0
≤0
(all ≤0)
Test
F(NP+1)-10^-6
>0
≤0
Go
To
Ⓡ
(P-T
Problem Complete
ITER = ITER + 1
XLN(I)→XLAST(I)
F(I)→FOLD(I)
Test
ITER - 30
≤0
>0
CALL
ERROR
Return
Go
To
Ⓛ

APPENDIX B

LISTING OF A

GENERATED PROGRAM

ete)

```
$IBFTC GEN7X.  FULIST
      SUBROUTINE GENL7X(X,PROP,PR2,T,P,KOPT,BEE)
      DIMENSION A1( 4, 4),X(  9),BETA(  4, 4),A(  9, 4),F( 5),Z(  8),
     1XLN(  8),C(  8),XLAST(  8),DLNX( 4),PSI(  4, 4, 4),
     2DENOM(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        O
      DATA W      /
     1 2.798125CE 03, 3.1960308E 03, 1.6320951E 08, 2.2416383E 03,
     2 6.3920616E 03, 2.9838710E 03, 2.9838710E 03, 5.5962500E 03/
      DATA EM     /
     1 3.2000000E 01, 2.8016000E 01, 5.4862000E-04, 3.9944000E 01,
     2 1.4008000E 01, 3.0008000E 01, 3.0008000E 01, 1.6000000E 01/
      DATA   A    /
     1 2.0000000E 00, 0.            , 0.            , 0.            ,
     2 0.            , 1.0000000E 00, 1.0000000E 00, 1.0000000E 00,
     3 0.            , 0.            , 2.0000000E 00, 0.            ,
     4 0.            , 1.0000000E 00, 1.0000000E 00, 1.0000000E 00,
     5 0.            , 0.            , 0.            , 0.            ,
     6 1.0000000E 00, 0.            , 0.            , 0.            ,
     7-1.0000000E 00, 0.            , 0.            , 0.            ,
     8 0.            , 0.            , 1.0000000E 00, 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 00, 0.            ,
     4 0.            , 0.            , 0.            , 0.            /
      DATA(CP     (I,1,1),I=    1,   76)/
     1 9.3716999E 03, 1.0730000E 04, 4.0757999E 08, 5.5925999E 03,
     2 1.5922000E 04, 1.0173000E 04, 1.0015000E 04, 1.6188000E 04,
     3 1.1342000E 04, 1.3349000E 04, 4.0757999E 08, 1.7240000E 03,
     4 1.1802000E 04, 9.3989999E 03, 1.2823000E 04, 1.4154000E 04,
     5 1.2575000E 04, 1.3870000E 04, 4.0757999E 08,-1.2845000E 03,
     6 2.0150000E 04, 1.2450000E 04, 6.8981000E 03, 1.3418000E 04,
     7 1.6108000E 04, 6.2157000E 03, 4.0757999E 08,-1.3009000E 04,
     8-1.5187000E 04, 1.2587000E 04, 1.1969000E 03, 1.4487000E 03,
     9 1.7652000E 04, 2.6283000E 03, 4.0757999E 08,-2.4883000E 04,
     1 2.2241000E 04, 1.6225000E 04, 2.2221000E 03,-1.8841000E 04,
     2 2.0258000E 04, 1.2720000E 04, 4.0757999E 08,-5.1119000E 04,
     3 5.5651000E 04, 2.2964000E 04, 2.2831000E 04, 7.9617000E 03,
     4 2.3085000E 00, 1.7330000E 00, 0.            , 2.7274000E-03,
     5 5.1045000E-02, 1.3905000E 00, 1.5749000E 00,-4.1353000E 00,
     6 6.6608999E-01, 2.7776000E-01, 0.            , 3.5441000E-01,
     7 1.2995000E 00, 2.6811000E 00, 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 00, 2.2711000E-01,
     1-2.0176000E-02, 1.0375000E 00, 0.            , 1.4685000E 00/
      DATA(CP     (I,1,1),I=   77,  152)/
     1 2.5260000E 00, 2.8819000E-01, 1.6171000E 00, 1.0385000E 00,
     2-1.4887000E-01, 1.3792000E 00, 0.            , 2.2874000E 00,
```

```
3 9.9835000E-01, 4.9669000E-02, 1.5316000E 00, 2.1816000E 00,
4-3.1300000E-01, 7.4846000E-01, 0.            , 3.9532000E 00,
5-2.3901000E-01,-2.8729000E-01, 1.5776000E-01, 1.1508000E 00,
6 1.4875000E 04, 1.1777000E 04,-1.9130000E 09, 9.8261999E 03,
7 2.6945000E 04, 1.7289000E 04, 1.3522000E 04, 1.7256000E 04,
8 2.8742000E 03,-5.2383000E 03,-1.9130000E 09, 4.1957000E 04,
9 5.6203000E 04, 2.1468000E 04,-5.0189000E 03, 2.7862000E 04,
1-6.2367000E 03,-8.9963000E 03,-1.9130000E 09, 6.7269000E 04,
2-1.0975000E 04, 2.6450000E 03, 4.1520000E 04, 3.2884000E 04,
3-3.4750000E 04, 5.2650000E 04,-1.9130000E 09, 1.6705000E 05,
4 2.9576000E 05, 1.6807000E 03, 8.8036000E 04, 1.3581000E 05,
5-4.7711000E 04, 8.2233000E 04,-1.9130000E 09, 2.6895000E 05,
6-4.5074000E 04,-2.9719000E 04, 7.9432999E 04, 3.1404000E 05,
7-7.0111000E 04,-5.3676000E 03,-1.9130000E 09, 4.9628000E 05,
8-3.5256000E 05,-8.9720999E 04,-9.8128999E 04, 6.8365000E 04,
9 3.0346000E 04, 3.6566000E 04, 0.            , 9.1752999E 02,
1 3.6174000E 08, 3.2255000E 07, 3.5402000E 08, 1.6589000E 08/
 DATA(CP     (I,1,1),I=  153,  228)/
1-1.1501000E 06,-2.3211000E 06, 0.            , 2.1278000E 07,
2 3.6854000E 08, 3.2487000E 07, 3.5121000E 08, 1.6640000E 08,
3-3.8614000E 06,-3.2845000E 06, 0.            , 3.9705000E 07,
4 3.3307000E 08, 3.0504000E 07, 3.7195000E 08, 1.6732000E 08,
5-1.9227000E 07, 2.9248000E 07, 0.            , 1.1884000E 08,
6 6.1576999E 08, 3.0298000E 07, 3.9904000E 08, 2.5559000E 08,
7-2.8490000E 07, 4.8081999E 07, 0.            , 2.0493000E 08,
8 1.5726000E 08, 2.5620000E 06, 3.9288000E 08, 4.3566999E 08,
9-4.8845000E 07,-3.2656000E 07, 0.            , 4.1154000E 08,
1-2.9376000E 08,-6.4829000E 07, 2.3832000E 08, 8.7229999E 07,
2 1.2500000E 03, 1.8000000E 03, 3.0000000E 04, 1.1000000E 04,
3 3.3000000E 03, 6.0000000E 02, 2.0000000E 03, 5.0000000E 02,
4 4.5000000E 03, 3.7000000E 03, 3.0000000E 04, 1.2250000E 04,
5 8.5000000E 03, 1.3000000E 03, 7.0000000E 03, 2.5000000E 03,
6 8.8000000E 03, 8.5000000E 03, 3.0000000E 04, 1.3500000E 04,
7 1.6000000E 04, 3.0000000E 03, 9.5000000E 03, 1.4750000E 04,
8 1.2250000E 04, 1.0500000E 04, 3.0000000E 04, 1.4500000E 04,
9 2.4500000E 04, 1.5250000E 04, 1.2000000E 04, 1.7750000E 04,
1 1.6000000E 04, 1.6000000E 04, 3.0000000E 04, 1.5750000E 04/
 DATA(CP     (I,1,1),I=  229,  240)/
1 2.7000000E 04, 2.0000000E 04, 1.5000000E 04, 2.6000000E 04,
2 3.0000000E 04, 1.8000000E 04, 3.0000000E 04, 1.7500000E 04,
3 3.0000000E 04, 3.0000000E 04, 1.7500000E 04, 3.0000000E 04/
 DATA NP,NSP,M,N,NSP1,NNNN/  4,  8,  4,  5,  9,  6/
 DATA PSI    /
1-0.            , 5.0000000E-01, 5.0000000E-01, 5.0000000E-01,
2-0.            , 5.0000000E-01, 5.0000000E-01, 0.            ,
3-0.            , 0.            ,-5.0000000E-01, 0.            ,
4-0.            , 0.            , 0.            , 0.            ,
5 0.            , 5.0000000E-01, 5.0000000E-01, 0.            ,
6 5.0000000E-01, 5.0000000E-01, 5.0000000E-01, 0.            ,
7 0.            , 0.            ,-5.0000000E-01, 0.            ,
8 0.            , 0.            , 0.            , 0.            ,
9 0.            , 0.            ,-1.0000000E 00, 0.            ,
1 0.            , 0.            ,-1.0000000E 00, 0.            ,
2 0.            , 0.            , 1.0000000E 00, 0.            ,
3 0.            , 0.            ,-0.            , 0.            ,
4 0.            , 0.            , 0.            , 0.            ,
5 0.            , 0.            , 0.            , 0.            ,
6 0.            , 0.            ,-0.            , 0.            ,
7 0.            , 0.            , 0.            , 0.            /
```

```
C       MINIMIZE FREE ENERGY
C       NP ARE THE NUMBER OF ELEMENTS
C       NSP ARE THE NUMBER OF SPECIES
C       T IS AGUESS FOR KOPT=2,3
C       KOPT=1 MEANS P,T GIVEN
C       KOPT=2 MEANS P,H GIVEN,PROP IS ENTHALPY
C       KOPT=3 MEANS P, S GIVEN, PROP IS ENTROPY
C       X(I) ARE MOLE FRACTIONS,YBAR=TOTAL INITIAL MOLES
        X(NSP1)=1.
        DO 23 I=1,NP
        A(NSP1,I)=0.
        DO 23 J=1,NSP
   23   A(NSP1,I)=A(NSP1,I)+A(J,I)*X(J)*X(NSP1)
        RETURN
        ENTRY GENR7X
        IF(KOPT.GT.0)GO TO 26
        KOPT=-KOPT
        DO 29 I=1,NP
   29   A(NSP1,I)=BEE(I)
C  26   WRITE(6,11)(X(I),I=1,NP)
   26   CONTINUE
        DO 20 I=1,NP
        IF(X(I).LE.0.)X(I)=.001
   20   XLN(I)=ALOG(X(I))
        IPH=1
C    2  FORMAT(3E10.6,I10)
C       WRITE(6,13)T,P,PROP,KOPT
C   13  FORMAT(3E20.8,I10)
C       INITIALIZE LAMDA
        ZLAMDA=1.
        PLOG=ALOG(P)
 1999   GO TO(2007,2000,2000) ,KOPT
 2000   IF(IPH.EQ.1)GO TO 2007
        LOPT=KOPT
C       WILL RETURN IN ETCH THE MIXTURE ENTHALPY FOR KOPT
C      1=2,ENTROPY KOPT=3
        GO TO 5200
C2164   WRITE(6,11)ETCH,T
 2164   CONTINUE
        IF(ABS((PROP-ETCH)/PROP).LT..000001)GO TO 300
        FN2=FOLDH
        FOLDH=FNEWH
        FNEWH=PROP-ETCH
        IF(IPH.NE.2)GO TO 2002
        TN2=TOLD
        TOLD=T
        IF(FNEWH)2003,300,2004
 2003   T=.95*T
        GO TO 2007
 2004   T=1.05*T
        GO TO 2007
 2002   IF(IPH.NE.3)GO TO 2008
 3010   FPRIME=(FNEWH-FOLDH)/(T-TOLD)
        TN2=TOLD
        TOLD=T
        T=T-FNEWH/FPRIME
        GO TO 2007
 2008   HN=T-TOLD
        HN1=TOLD-TN2
```

```
      GN=HN/HN1
      AM=GN*(FNEWH-(1.+GN)*FOLDH+GN*FN2)
      BM=(2.*GN+1.)*FNEWH-FOLDH*(1.+GN)**2+GN*GN*FN2
      CM=FNEWH*(1.+GN)
      CDN=BM*BM-4.*AM*CM
      IF(CDN.LT.0.)GO TO 3010
      CDN=SQRT(CDN)
      DENOM(1)=BM+CDN
      DENOM(2)=BM-CDN
      IF(ABS(DENOM(1))-ABS(DENOM(2)))3002,3003,3004
 3003 BOTTOM=DENOM(2)
      GO TO 3005
 3004 BOTTOM=DENOM(1)
 3005 TN2=TOLD
      TOLD=T
      T=T-HN*2.*CM/BOTTOM
 2007 LOPT=1
      GO TO 5407
 2163 KKK=3
      GO TO 4200
 4101 CONTINUE
C4101 WRITE(6,11)ETOP,FNEWH,FPRIME,T,FOLDH,TOLD,X
 2001 DO 12 I=1,NSP
   12 XLAST(I)=XLN(I)
   40 ITER=0
      KOUNT=0
      KKK=1
   64 KOUNT1=0
      LLL=3
      GO TO 600
  210 DO 211 I=1,N
  211 FOLD(I)=F(I)
   61 GO TO(4017,4020),KKK
 4102 LLL=2
      GO TO 600
C4500 WRITE(6,11)ZLAMDA,FOLD
 4500 CONTINUE
      IF(F(N)-FOLD(N))70,70,71
   71 IF(F(N).LT..000001)GO TO 85
      ZLAMDA=ZLAMDA/2.
      KOUNT1=0
      KOUNT=KOUNT+1
      KKK=2
      DO 75 I=1,NP
      F(I)=FOLD(I)
   75 XLN(I)=XLAST(I)
      F(N)=FOLD(N)
      IF(KOUNT-10)61,61,72
   72 CALL ERROR(GX1,KRUN,13)
   74 GO TO 305
   70 KKK=1
   81 KOUNT1=KOUNT1+1
      IF(KOUNT1-2)85,90,90
   90 ZLAMDA=2.*ZLAMDA
      IF(ZLAMDA.GT.1.)ZLAMDA=1.
      KOUNT1=0
   85 DO 86 I=1,NSP
      IF(ABS((XLN(I)-XLAST(I))/XLN(I))-.0001)86,86,91
   86 CONTINUE
```

```
      IF(F(N)-.00000001)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
      DO 89 I=1,NSP
   89 XLAST(I)=XLN(I)
      DO 189 I=1,N
  189 FCLC(I)=F(I)
      IF(ITER-30)61,92,92
   92 CALL ERROR(GX1,KRUM,12)
   99 CONTINUE
C  99 WRITE(6,11)ZLAMDA,X,XLN,XLAST
      IF(KOPT.EQ.1)GO TO 300
      IPP=IPP+1
      IF(IPP-10)1999,1999,1998
 1998 CALL ERROR(GX1,KRUM,11)
   11 FORMAT(5E20.8)
  300 LOPT=KOPT+3
      GO TO 5200
  300 PROP=ETOP
      LOPT=7
      GO TO 5407
  307 LOPT=4
  305 KOPT=LOPT-3
      PR2=ETOP
      RETURN
 4017 DO 4001 J=1,NP
      A1(1,J)=0.
      DO 4002 I=1,M
      II=NP+I
 4002 A1(1,J)=A1(1,J)+X(II)*BETA(I,J)
 4001 A1(1,J)=X(J)+A1(1,J)
C     DF MATRIX IN A1 ARRAY
C     GET D(1/YBAR)/DLNX(L) IN TEMP(L)
      DO 4004 L=1,NP
      TEMP(L)=0.
      DO 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
      DO 4006 I=2,NP
      DO 4006 L=1,NP
      TEMP1(L)=0.
      DO 4005 J=N,NSP
      JJ=J-NP
 4005 TEMP1(L)=TEMP1(L)+PSI(JJ,I,L)*X(J)
 4006 A1(I,L)=A(L,I)*X(L)+TEMP1(L)-A(NSP1,1)*TEMP(L)
      LLL=1
 4007 CALL GLEM7X(A1,DLNX,F,NP,N,AT)
 4020 DO 4021 I=1,NP
 4021 XLN(I)=XLN(I)-ZLAMDA*DLNX(I)
 4200 DO 4201 L=1,NP
 4201 TEMP1(L)=XLN(L)+C(L)
      DO 4025 I=N,NSP
      XLN(I)=0.
      JJ=I-NP
      DO 4031 L=1,NP
 4031 XLN(I)=XLN(I)+TEMP1(L)*BETA(JJ,L)
```

```
 4025 XLN(I)=XLN(I)-C(I)
      DO 4022 I=1,NSP
      IF(XLN(I))4022,4022,4023
C     CORRECT ANY MOLE FRACTIONS LARGER THAN 1
 4023 XLN(I)=-.01
 4022 CONTINUE
C     COMPUTE X(I) AND YBAR
      X(NSP1)=0.
      DO 4050 J=1,NSP
      X(J)=EXP(XLN(J))
      IF(ABS(XLN(J)).GT.88.)X(J)=0.
 4050 X(NSP1)=X(NSP1)+A(J,1)*X(J)
      X(NSP1)=A(NSP1,1)/X(NSP1)
      GO TO(4102,4102,4101),KKK
C     COMPUTES F FOR NEWTON RAPHSON
C     ALSO STORES LARGEST F IN F(N)
  600 DUM=0.
      DO 601 I=1,NSP
  601 DUM=DUM+X(I)
      F(1)=DUM-1.
      DO 603 I=2,NP
      DUM=0.
      DO 602 J=1,NSP
  602 DUM=DUM+A(J,I)*X(J)
  603 F(I)=DUM-A(NSP1,I)/X(NSP1)
C     FIND THE LARGEST F(I)
      F(N)=ABS(F(1))
      DO 605 I=2,NP
      E=ABS(F(I))
      IF(E-F(N))605,604,604
  604 F(N)=E
  605 CONTINUE
C     WRITE(6,11)F
      GO TO(4007,4500,2100),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*LNT+B*T+C
C     H=A*T+.5*B*T*T+D
C     CONVERT MOLE TO MASS FRACTIONS
C     Z ARRAY FOR MASS FRACTIONS
 5200 EMM=0.
      DO 5201 I=1,NSP
 5201 EMM=EMM+X(I)*EM(I)
      DO 5202 I=1,NSP
 5202 Z(I)=X(I)*EM(I)/EMM
C     WRITE(6,11)Z,EM,EMM
 5407 TLOG=ALOG(T)
      ETOH=0.
      DO 5401 I=1,NSP
      DO 5402 JJ=1,NNNN
      IF(CP(I,JJ,5)-T)5402,5403,5403
 5402 CONTINUE
      CALL ERROR(T,I,10)
 5403 GO TO(5406,5405,5404,5405,5404,5405,5404),LOPT
```

```
 5405 HI=(CP(I,JJ,1)+.5*CP(I,JJ,2)*T)*T+CP(I,JJ,4)                2
C     WRITE(6,113)I,JJ,HI                                          2
      ETCH=ETCH+Z(I)*HI                                            2
      GO TO 5401                                                   2
 5404 SI=CP(I,JJ,1)*TLOG+CP(I,JJ,2)*T+CP(I,JJ,3)                   2
C     WRITE(6,113)I,JJ,SI                                          2
      ETCH=ETCH+X(I)*(SI/W(I)-XLN(I)-PLOG)/EMM                     2
      GO TO 5401                                                   2
 5406 C(I)=CP(I,JJ,1)-CP(I,JJ,3)-.5*CP(I,JJ,2)*T+CP(I,JJ,4)/      2
     1T-CP(I,JJ,1)*TLOG                                            2
      C(I)=C(I)/W(I)+PLOG                                          2
C     WRITE(6,113)I,JJ,C(I)                                        2
 5401 CONTINUE                                                     2
C 113 FORMAT(2I10,3E20.8)                                          2
      GO TO(2163,2164,2164,306,305,305,307),LOPT                   2
      END                                                          2
$1     SCLEM7                                                      2
      SUBROUTINE CLEM7X(B,X,D,M,M1,AT)                             2
      DIMENSION B(M,M),X(M),D(M1),AT(M,M1)                         2
C     EQUATIONS ARE OF THE FORM BX=D                               2
      DO 17 I=1,M                                                  2
   17 X(I)=0.0                                                     2
      DO 200 I=1,M                                                 2
  200 AT(I,M1)=D(I)                                                2
      DO 201 I=1,M                                                 2
      DO 201 J=1,M                                                 2
  201 AT(I,J)=B(I,J)                                               2
      DO 32 N=1,M                                                  2
      C=AT(N,N)                                                    2
      IT=0                                                         2
      DO 9 I=N,M                                                   2
      IF(ABS(AT(I,N))-ABS(C))9,9,8                                 2
    8 C=AT(I,N)                                                    2
      IT=I                                                         2
    9 CONTINUE                                                     2
      IF(IT-N)7,7,70                                               2
   70 DO 71 J=N,M1                                                 2
      TEMP=AT(N,J)                                                 2
      AT(N,J)=AT(IT,J)                                             2
   71 AT(IT,J)=TEMP                                                2
    7 DO 10 I=1,M1                                                 2
   10 AT(N,I)=AT(N,I)/C                                            2
      IF(M-N)50,50,18                                              2
   18 N1=N+1                                                       2
      DO 30 I=N1,M                                                 2
      C=AT(I,N)                                                    2
      DO 30 J=N,M1                                                 2
   30 AT(I,J)=AT(I,J)-AT(N,J)*C                                    2
   32 CONTINUE                                                     2
   50 X(M)=AT(M,M+1)                                               2
      DO 65 N=2,M                                                  2
      NR=M+1-N                                                     2
      C=AT(NR,M+1)                                                 2
      DO 60 I=NR,M                                                 2
   60 C=C-AT(NR,I)*X(I)                                            2
   65 X(NR)=C/AT(NR,NR)                                            2
      RETURN                                                       2
      END                                                          2
```

412 CARDS PRINTED

APPENDIX C

LISTING OF SUGGESTED

SUBROUTINE ERROR

```
$IBFTC LIST     ERROR.
      SUBROUTINE ERROR(G,K,I)
      WRITE(6,10)I
   10 FORMAT(6HCERROR,I3)
      J=I-9
      GO TO(1,2,3,4),J
    1 WRITE(6,11)G,K
   11 FORMAT(14HOTEMPERATURE =E15.7,19HFIT FOR SPECIE NO. I4,9HNOT VALID
     1)
      GO TO 7
    2 WRITE(6,12)
   12 FORMAT(43HOOUTER EQUILIBRIUM ITERATIONS EXCEED 10                )
      GO TO 7
    3 WRITE(6,13)
   13 FORMAT(44HOPRESSURE TEMPERATURE ITERATIONS EXCEED 30             )
      GO TO 7
    4 WRITE(6,14)
   14 FORMAT(19HOLAMDA CUT 10 TIMES                                    )
    7 RETURN
      END
```

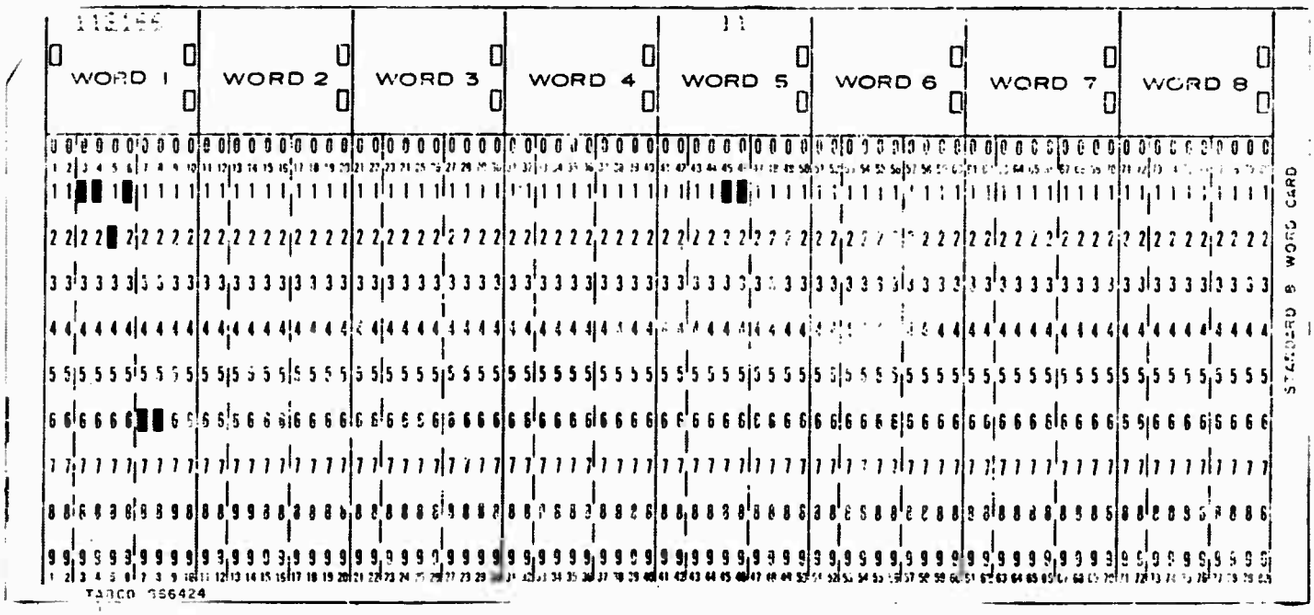
20 CARDS PRINTED

APPENDIX D

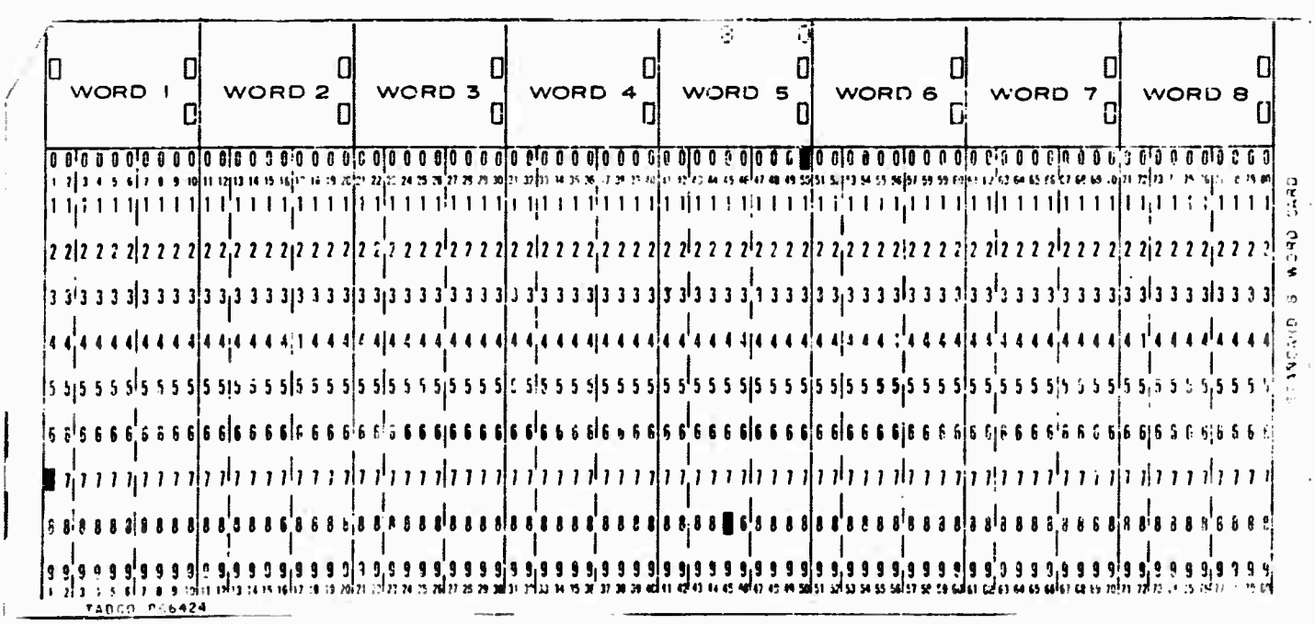
SAMPLE INPUT FOR THE

CHEMISTRY EQUILIBRIUM GENERATOR PROGRAM

CARD NO.

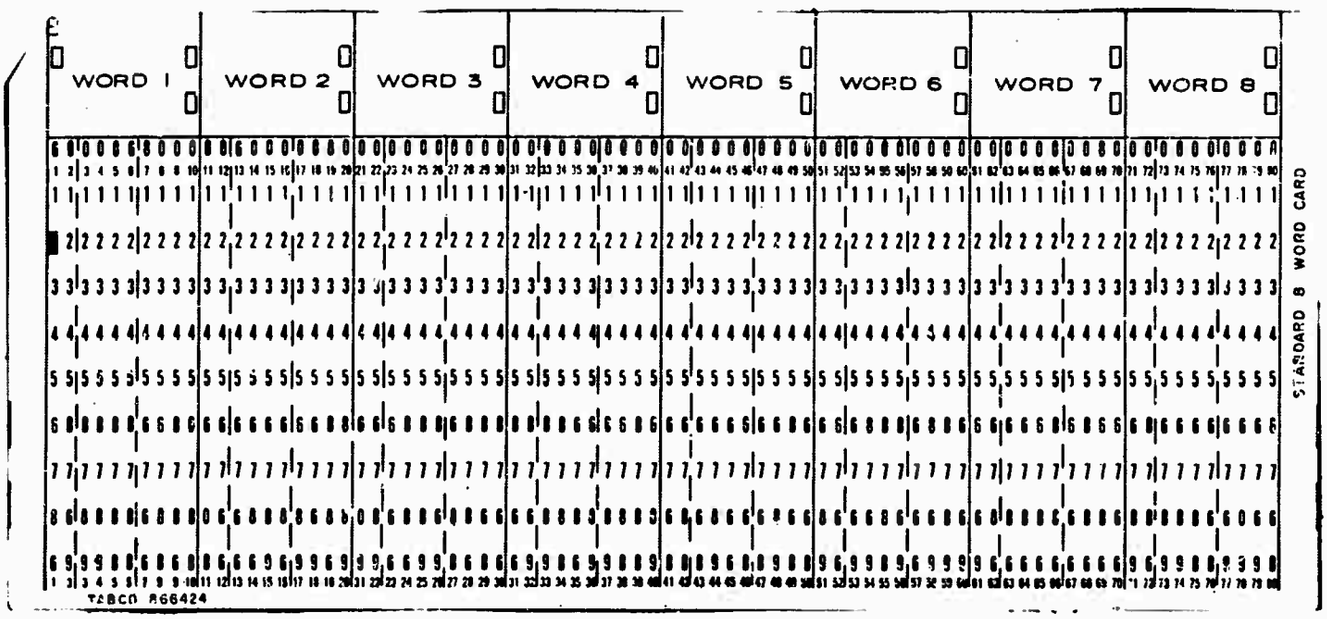

WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
STANDARD 8 WORD CARD
TABCO 056424

1

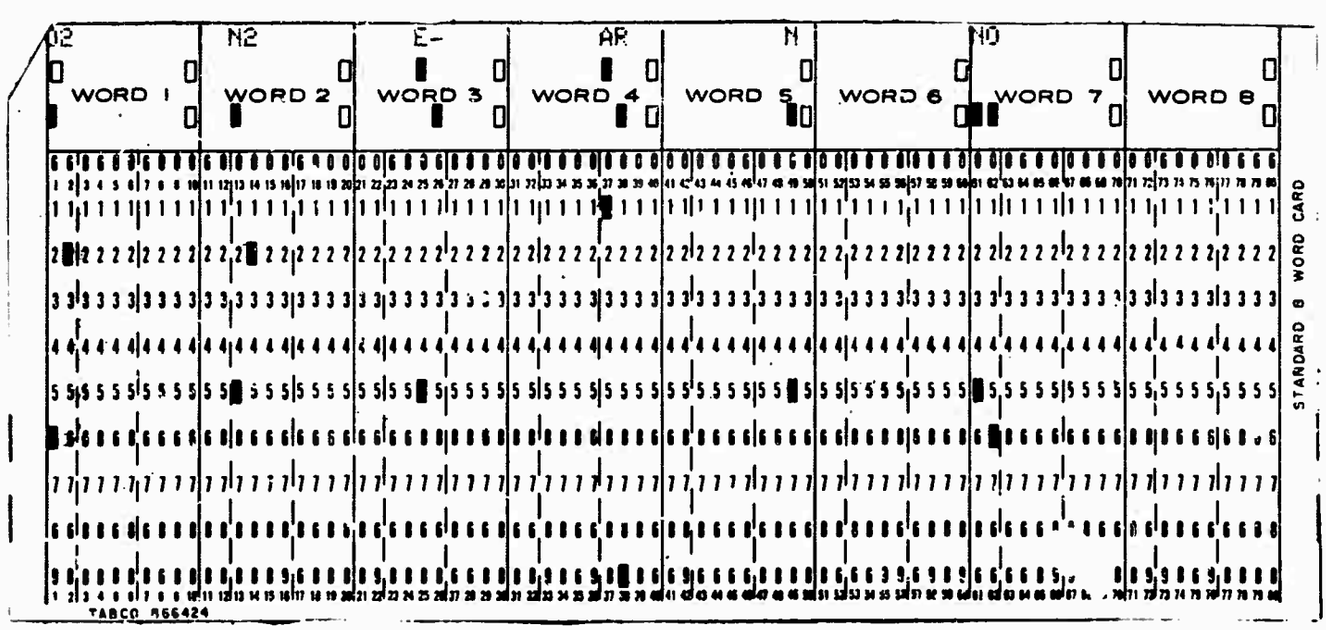

WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
STANDARD 8 WORD CARD
TABCO 056424

2

CARD NO.

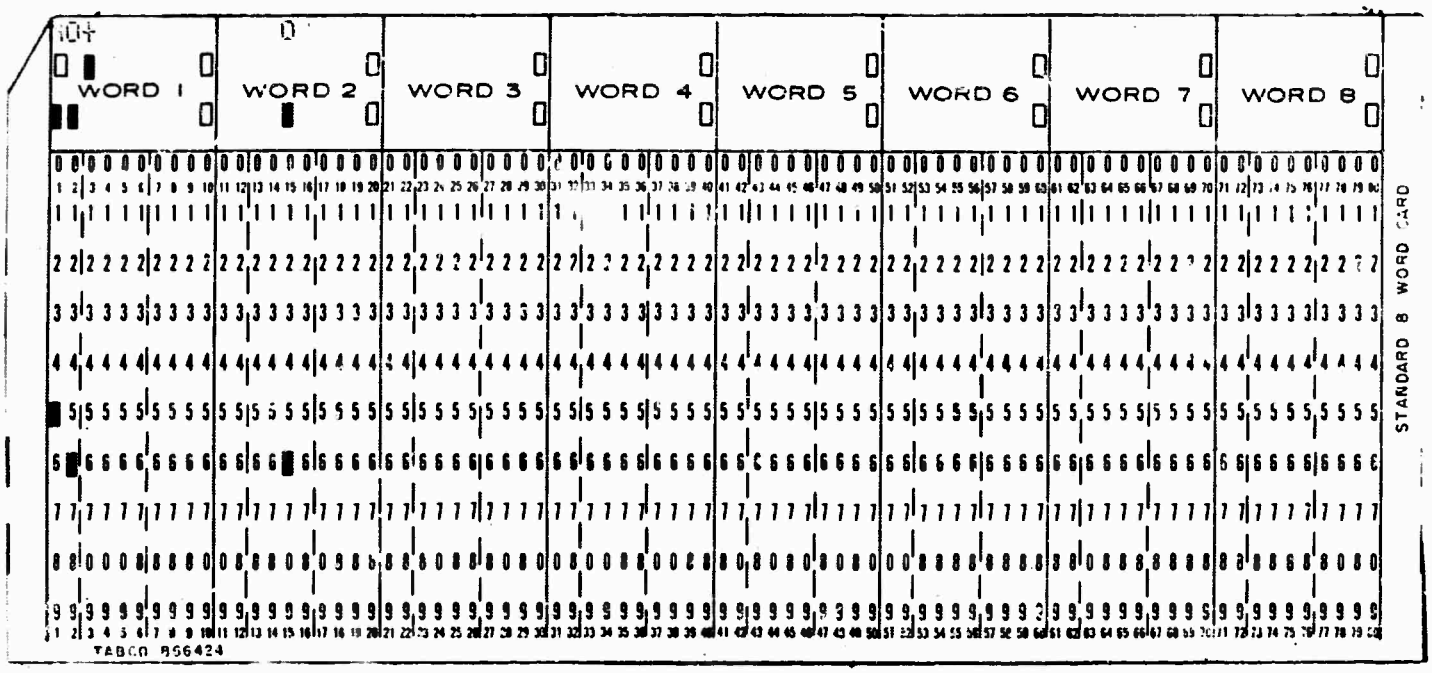

WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
STANDARD 8 WORD CARD
TABCO R66424

3


02
N2
E-
AR
N
NO
WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
STANDARD 8 WORD CARD
TABCO R66424

4

CARD NO. 5


WORD 1
WORD 2
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
STANDARD 8 WORD CARD
TABCO 956424