FORTRAN IV PROGRAM
FOR CALCULATION OF
THERMODYNAMIC DATA

by Bonnie J. McBride and Sanford Gordon

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1967
FORTRAN IV PROGRAM FOR CALCULATION OF THERMODYNAMIC DATA

By Bonnie J. McBride and Sanford Gordon

Lewis Research Center
Cleveland, Ohio
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CALCULATION OF IDEAL GAS THERMODYNAMIC FUNCTIONS</td>
<td>2</td>
</tr>
<tr>
<td>Internal Partition Functions for Monatomic Gases</td>
<td>3</td>
</tr>
<tr>
<td>Cutoff methods</td>
<td>3</td>
</tr>
<tr>
<td>Inclusion of predicted levels</td>
<td>4</td>
</tr>
<tr>
<td>Internal Partition Function for Diatomic and Polyatomic Molecules</td>
<td>6</td>
</tr>
<tr>
<td>EMPRICAL EQUATIONS FOR THERMODYNAMIC FUNCTIONS</td>
<td>6</td>
</tr>
<tr>
<td>ASSIGNED ENTHALPY VALUES</td>
<td>7</td>
</tr>
<tr>
<td>ASSIGNED REFERENCE ELEMENTS</td>
<td>7</td>
</tr>
<tr>
<td>HEATS OF FORMATION AND EQUILIBRIUM CONSTANTS</td>
<td>8</td>
</tr>
<tr>
<td>COMPUTER PROGRAM</td>
<td>8</td>
</tr>
<tr>
<td>Availability to Other Organizations</td>
<td>9</td>
</tr>
<tr>
<td>Options</td>
<td>9</td>
</tr>
<tr>
<td>Partition functions - monatomic gases</td>
<td>9</td>
</tr>
<tr>
<td>Partition functions - diatomic and polyatomic gases</td>
<td>10</td>
</tr>
<tr>
<td>Thermodynamic functions from empirical equations</td>
<td>10</td>
</tr>
<tr>
<td>Least-squares fit</td>
<td>11</td>
</tr>
<tr>
<td>Heat of formation and log K values</td>
<td>11</td>
</tr>
<tr>
<td>Input</td>
<td>12</td>
</tr>
<tr>
<td>Types of data</td>
<td>12</td>
</tr>
<tr>
<td>Identification of cards</td>
<td>12</td>
</tr>
<tr>
<td>Uniform format</td>
<td>12</td>
</tr>
<tr>
<td>Contents of individual cards</td>
<td>12</td>
</tr>
<tr>
<td>General Flow of Program</td>
<td>13</td>
</tr>
<tr>
<td>Output</td>
<td>14</td>
</tr>
<tr>
<td>Punched card output</td>
<td>14</td>
</tr>
<tr>
<td>Listed output</td>
<td>15</td>
</tr>
<tr>
<td>Examples</td>
<td>15</td>
</tr>
<tr>
<td>Main Routine and Subroutines</td>
<td>16</td>
</tr>
<tr>
<td>PAC1(C10 to C60)</td>
<td>16</td>
</tr>
<tr>
<td>INPUT(C70)</td>
<td>16</td>
</tr>
<tr>
<td>PAGEID(C80)</td>
<td>16</td>
</tr>
<tr>
<td>EFTAPE(C90 to C130)</td>
<td>16</td>
</tr>
</tbody>
</table>
IDENT(C140 to C160) ................................................. 16
TEMPER(C170 to C180) .............................................. 16
RECO(C190 to C250) .................................................. 16
ATOM(C260 to C310) .................................................. 17
POLY(C320 to C410) ................................................... 17
LINK1(C420 to C480) .................................................. 17
KD(C480) ................................................................. 17
DERIV(C490) ............................................................. 17
QSUM(C500) .............................................................. 17
DELH(C510 to C530) ................................................... 17
TABLES(C540 to C570) ................................................. 18
LOGK(C580 to C650) ................................................... 18
LEAST(C660 to C760) ................................................... 18
PUNCH(C770 to C790) ................................................... 18

APPENDIXES
A - SYMBOLS ............................................................ 19
B - FORTRAN LISTING (FORTRAN ROUTINES) .................... 22
C - MAP ROUTINES (DESCRIPTION AND LISTING) ............... 65
   SKFIEL(N, M) ............................................................. 65
   BCDUMP(A, B) ........................................................... 68
   BCREAD(A, B) ........................................................... 72
   IALS(N, M) .............................................................. 74
   IARS(N, M) .............................................................. 74
D - DETAILS IN PREPARING INPUT .................................... 75
   Uniform Format ........................................................ 75
   Order of Input Cards ............................................... 76
   General Data Cards .................................................. 76
      CONSTS card .......................................................... 76
      ATOM cards .......................................................... 77
      EFDATA and binary EF data cards ................................ 77
      LISTEF card .......................................................... 77
   Specific Input Cards ................................................ 77
      Formula card .......................................................... 77
      REFNCE card .......................................................... 78
      EFTAPE card .......................................................... 78
      LOGK card ............................................................ 78
      LISTSQS card(s) ...................................................... 79
      INTERM card ........................................................... 79
      DATE card ............................................................ 79
| TEMP card(s) | .......................................................... | 79 |
| METHOD card | .......................................................... | 80 |
| DATA cards  | .......................................................... | 81 |
| DATA cards for READIN method | .......................................................... | 81 |
| DATA cards for COEF method | .......................................................... | 81 |
| DATA cards for FIXEDN, ALLN, or TEMPER methods | .......................................................... | 82 |
| DATA cards for RRHO, PANDK, JANAF, NRRHO1, or NRRHO2 methods | .......................................................... | 82 |
| FINISH card | .......................................................... | 82 |

**E - DETAILS IN OUTPUT**

| Punched Cards | .......................................................... | 83 |
| EFDATA and binary EF data | .......................................................... | 83 |
| Coefficients for empirical equations | .......................................................... | 83 |
| Listed Output | .......................................................... | 84 |
| Input data | .......................................................... | 84 |
| Punched card output | .......................................................... | 84 |
| Tables of thermodynamic properties | .......................................................... | 84 |
| Least-squares polynomial and errors | .......................................................... | 85 |
| Intermediate data with FILL option for monatomic gases | .......................................................... | 86 |
| Intermediate data with INTERM card | .......................................................... | 86 |
| Monatomic gases | .......................................................... | 86 |
| Diatomic and polyatomic gases | .......................................................... | 87 |

**F - EXAMPLES**

| Example 1 (MgF$_2$(g) with RRHO Method) | .......................................................... | 89 |
| Problem | .......................................................... | 89 |
| Punched card input | .......................................................... | 89 |
| Listed output | .......................................................... | 90 |
| Example 2 (F$_2$(g) with PANDK Method and EFTAPE Option) | .......................................................... | 90 |
| Problem | .......................................................... | 90 |
| Punched card input | .......................................................... | 90 |
| Listed output | .......................................................... | 91 |
| Example 3 (F(g) with LOGK Option) | .......................................................... | 94 |
| Problem | .......................................................... | 94 |
| Punched card input | .......................................................... | 94 |
| Listed output | .......................................................... | 94 |
| Example 4 (P(s) with Least-Squares Fit) | .......................................................... | 95 |
| Problem | .......................................................... | 95 |
| Punched card input | .......................................................... | 95 |
| Listed output | .......................................................... | 96 |
FORTRAN IV PROGRAM FOR CALCULATION OF THERMODYNAMIC DATA

by Bonnie J. McBride and Sanford Gordon

Lewis Research Center

SUMMARY

A FORTRAN IV program is described which (1) calculates thermodynamic functions (heat capacity, enthalpy, entropy, and free energy), (2) fits these functions to empirical equations, and (3) calculates, as a function of temperature, heats of formation and equilibrium constants.

The program provides several methods for calculating ideal gas properties. For monatomic gases, three methods are given which differ in the technique used for truncating the partition function. For diatomic and polyatomic molecules, five methods are given which differ in the corrections to the rigid-rotator harmonic-oscillator approximation.

In addition the program provides for calculating thermodynamic functions for solids, liquids, and gases from empirical heat capacity equations.

INTRODUCTION

Numerous compilations of thermodynamic data are available (refs. 1 to 12). However, there is a continuing need for additional calculations due to (1) discovery of new species, (2) revision of existing molecular constant data and structural parameters, (3) need for data at temperatures other than already published, (4) availability of new or revised heats of formation, dissociation, or transition, (5) revision of fundamental constants or atomic weights, and (6) preference for thermodynamic data in functional rather than tabular form. Calculations may also be needed to compare the results of assuming various possible forms of the partition function.

For these reasons, a flexible FORTRAN IV program has been prepared for the IBM 7094 which can perform any combination of the following: (1) calculate thermodynamic functions (heat capacity, enthalpy, entropy, and free energy) for any set of 1 to
200 temperatures, (2) fit the functions to empirical equations, and (3) calculate, as a function of temperature, heats of formation and equilibrium constants from assigned reference elements and/or from these elements in their atomic gaseous state.

The thermodynamic functions for ideal gases may be calculated from molecular constant data using one of several partition function variations provided by the program. For monatomic gases, (1) one of three partition function cutoff techniques may be selected and (2) unobserved but predicted electronic energy levels may be included by the program. For diatomic and polyatomic gases, (1) one of five partition functions may be selected which differ in the correction factors for nonrigid rotation, anharmonicity, and vibration-rotation interactions and (2) excited electronic states may be included.

For the purpose of additional processing, known thermodynamic functions for solids, liquids, or gases may be (1) calculated from heat capacity equations or (2) read in directly from IBM cards.

Because of the variety of options provided and the resulting variety of input data required, an objective was to provide for a relatively simple procedure for reading input data. This was accomplished by means of a uniform input format.

The program and the equations used are described in detail. Examples of input and output are given for several typical species.

**CALCULATION OF IDEAL GAS THERMODYNAMIC FUNCTIONS**

For gaseous species, the thermodynamic functions may be calculated from spectroscopic constants. A general discussion of methods of calculation is given in reference 3. Many of the equations will be repeated here for convenience. The properties are expressed as functions of the internal partition function Q; that is,

\[
\frac{C^0_P}{R} = T^2 \frac{d^2 (\ln Q)}{dT^2} + 2T \frac{d(\ln Q)}{dT} + \frac{5}{2}
\]

(1)

\[
\frac{H^0_T - H^0_0}{RT} = T \frac{d(\ln Q)}{dT} + \frac{5}{2}
\]

(2)

\[
\frac{S^0_T}{R} = T \frac{d(\ln Q)}{dT} + \ln Q + \frac{3}{2} \ln M + \frac{5}{2} \ln T + S_c + \frac{5}{2}
\]

(3)
\[
\frac{F^0_T - H^0_0}{RT} = \frac{S^0_T}{R} - \frac{H^0_T - H^0_0}{RT} = \ln Q + \frac{3}{2} \ln M + \frac{5}{2} \ln T + S_c
\]

where

\[
S_c = \ln \left[ \frac{k}{\left( \frac{2\pi k}{N_0 h^2} \right)^{3/2}} \right]
\]

(Symbols are defined in appendix A).

The internal partition function \( Q \) in equations (1) to (4) is given by

\[
Q = \sum_{m=1}^{L} Q^m
\]

where \( Q^m \) is the internal partition function for the \( m^{th} \) electronic state and \( L \) is the number of electronic states.

**Internal Partition Functions for Monatomic Gases**

For monatomic molecules, internal energy consists of electronic energy only. Equation (6) then becomes

\[
Q = \sum_{m=1}^{L} Q^m = \sum_{m=1}^{L} \left( 2J_m + 1 \right) e^{-\epsilon_m/kT} = \sum_{m=1}^{L} g_m e^{-\epsilon_m/kT}
\]

where \( Q^m_e, J_m, \epsilon_m, \) and \( g_m \) are the electronic energy partition function, total angular momentum quantum number, electronic excitation energy, and statistical weight, respectively, for the \( m^{th} \) electronic state.

**Cutoff methods.** - An infinite number of bound states exists below the ionization limit for a hypothetical isolated atom \((L = \infty \) in equation (7)). Inasmuch as the partition function diverges and approaches infinity as \( L \to \infty \), the summation must be cut off. A recent review of various cutoff methods is given by reference 13. These cutoff methods may be considered to be of the following types:

(1) No dependence on temperature or pressure
(2) Dependence on temperature only

(3) Dependence on temperature and pressure (or density) and possibly degree of ionization

In the first of the three types, the summation may include various numbers of levels. For example, only the ground state is used in the Saha equation (see ref. 14). The summation of equation (7) may be over a fixed and usually arbitrary number of levels (such as for lithium in ref. 15 or for all species in ref. 11) or equation (7) may be summed through all observed levels (as in ref. 2, for example).

The second cutoff type is temperature dependent. The ionization potential is reduced by a quantity referred to as the "ionization potential lowering," which in this case is a function of temperature only. The partition function is then permitted to include only those levels below the "lowered" ionization potential. Reference 16 suggested that the ionization potential be lowered by an amount equal to the temperature function kT. This suggested method was used in reference 3. Other temperature functions are summarized in reference 13.

The first two cutoff types are distinguished by the fact that they permit the partition function and related thermodynamic properties to be calculated as functions of temperature only. For the third type, it is not possible to calculate the partition function by specifying temperature only. One cutoff technique of this type relates the highest permitted principal quantum number n to the number of particles per unit volume (number density) such as suggested by Bethe (see discussion in ref. 13). Another technique uses the ionization potential lowering procedure previously described, but in this case the quantity by which the potential is lowered is a function of electron and ionized particle number densities. Several such quantities are summarized in reference 13.

This last technique involves mixtures of species and therefore precludes, for all practical purposes, the possibility of generating tables for individual species as a function of temperature only. This is due to the fact that the cutoff criterion needed to calculate the partition function depends on mixture composition, while the calculation of mixture composition depends on the partition function. Thus an iterative procedure is required where the partition function at a specified temperature may be changing from one iteration to the next. Consequently, only the first two cutoff types are considered in this report.

Inclusion of predicted levels. - In addition to the divergence problem, there is the problem of whether to include observed energy levels only or also to include levels for predicted terms which, so far, have not been observed. From atomic theory, as presented in texts such as reference 17, predicted terms can be derived. Some of these terms are given in tables 10 and 11 in reference 17 and tables 5 to 20 in references 18 to 20. An examination of the tabulated observed terms in references 18 to 20 shows that many predicted terms are missing, especially for the higher quantum numbers.
It has been shown that various series of levels can be represented by formulas such as the Rydberg or the Rydberg-Ritz formulas (e.g., ref. 21). The constants in these formulas can be determined from known levels and used to extrapolate for the unobserved levels. However, the number of observed levels differ from species to species and, therefore, some judgment must be exercised in obtaining these constants. Thus, while in principle this technique of obtaining predicted, but unobserved, levels can be programmed, in practice it amounts to essentially a special program for each species. Therefore, this technique was not considered further for this program.

An alternate, but considerably simpler, technique for filling in unobserved levels, which gives essentially the same results for the partition function for many species as does the use of the Rydberg-Ritz equations, was included in the program. This alternate technique will now be described.

By examining the statistical weights \( g_i \) corresponding to predicted terms, it was determined that for at least the first 20 chemical elements, the sum of the statistical weights could be expressed by the following simple function of the principal quantum number \( n \) (except for the ground state \( n \) of most species)

\[
\sum g_i = \sum (2J_i + 1) = bn^2 
\]

Equation (8) applies only to terms arising from excitation of the emission electron and does not account for other possible terms. The table at the left lists (1) the derived constants \( b \) to be used in equation (8) to obtain \( \sum g_i \) for any \( n \) above the ground state and (2) \( g_i \) values for the ground state.

The usefulness of equation (8) arises from the fact that the inclusion of an unobserved level generally makes considerably more difference than a small error in the estimated energy for this level. Therefore, an option is provided in the program to determine for each \( n \) the difference in statistical weight sums between the observed levels which have been read in as input and that given by equation (8). The program then assigns to this difference the

<table>
<thead>
<tr>
<th>Atomic number</th>
<th>Chemical symbol</th>
<th>Constant in equation (8), ( b )</th>
<th>Sum of statistical weights for ground state, ( \sum g_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hydrogen</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Helium</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Lithium</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Beryllium</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Boron</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Carbon</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>Nitrogen</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>Oxygen</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Fluorine</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Neon</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Sodium</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>Magnesium</td>
<td>4</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Aluminum</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>Silicon</td>
<td>12</td>
<td>75</td>
</tr>
<tr>
<td>15</td>
<td>Phosphorus</td>
<td>30</td>
<td>170</td>
</tr>
<tr>
<td>16</td>
<td>Sulfur</td>
<td>40</td>
<td>215</td>
</tr>
<tr>
<td>17</td>
<td>Chlorine</td>
<td>30</td>
<td>156</td>
</tr>
<tr>
<td>18</td>
<td>Argon</td>
<td>12</td>
<td>61</td>
</tr>
<tr>
<td>19</td>
<td>Potassium</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>20</td>
<td>Calcium</td>
<td>4</td>
<td>61</td>
</tr>
</tbody>
</table>
highest observed level for the corresponding \( n \) and includes it with the observed levels.

This method of "filling in" predicted, but unobserved, levels by means of equation (8) was used to calculate the thermodynamic functions of the atomic species in reference 3.

### Internal Partition Function for Diatomic and Polyatomic Molecules

For diatomic and polyatomic molecules, \( Q^m \) in equation (6) involves vibrational and rotational as well as electronic energy. In this report the following factored form is used to calculate \( Q^m \):

\[
Q^m = Q_e^m Q_V^m Q_R^m Q_\theta^m Q_W^m Q_c^m
\]

or

\[
\ln Q^m = \ln Q_e^m + \ln Q_V^m + \ln Q_R^m + \ln Q_\theta^m + \ln Q_W^m + \ln Q_c^m
\]  \( (9) \)

The quantities \( Q_e^m \), \( Q_V^m \), and \( Q_R^m \) are the electronic, harmonic-oscillator, and classical-rotation contributions to the partition function, respectively, as given in standard texts (see refs. 22 to 25). The remaining quantities in equation (9) are as follows: rotational stretching \( Q_R^m \) (ref. 25 or 26), low-temperature rigid rotation \( Q_\theta^m \) (refs. 25 and 27), Fermi resonance \( Q_W^m \) (ref. 28), and both anharmonicity and vibration-rotation interaction \( Q_c^m \) (refs. 29 to 31).

The program provides five methods of calculating the partition function which vary in the inclusion of and formulas for the correction terms (\( \ln Q_R^m \), \( \ln Q_\theta^m \), \( \ln Q_W^m \), and \( \ln Q_c^m \)). This provision is made so that the results of the various methods may be compared.

Table I contains detailed formulas for all the \( \ln Q^m \) terms and their derivatives except those for \( \ln Q_c^m \) which are given in table II. The derivatives of \( \ln Q_c^m \) are not given directly as are the derivatives in table I. It was found to be considerably more convenient to express the derivatives of \( \ln Q_c^m \) by means of general formulas than to obtain the derivatives directly. These general formulas are given in a footnote of table II.

### EMPIRICAL EQUATIONS FOR THERMODYNAMIC FUNCTIONS

Empirical equations for thermodynamic functions are often used for convenience.
These equations are usually based on the following form for heat capacity:

\[ C_p^o = \sum_{i=1}^{r} a_i T^q_i \]  \hspace{1cm} (10)

Enthalpy and entropy are related thermodynamically to \( C_p^o \) as follows:

\[ H_T^o = a_{r+1} + \int C_p^o \, dT \]  \hspace{1cm} (11)

\[ S_T^o = a_{r+2} + \int \left( \frac{C_p^o}{T} \right) \, dT \]  \hspace{1cm} (12)

where \( a_{r+1} \) and \( a_{r+2} \) are integration constants.

The program uses equations (10) to (12) in two ways, either in generating the coefficients \( a_i \) from a set of thermodynamic data using the least-squares technique given in reference 32, or conversely, in generating the thermodynamic data from the empirical equations. The least squares method differs from the usual least squares treatment in that it simultaneously fits heat capacity, enthalpy, and entropy.

**ASSIGNED ENTHALPY VALUES**

For some applications (see ref. 33) it is convenient to combine sensible enthalpy and energies of chemical and physical changes into one numerical value. An arbitrary base may be adopted for assigning absolute values to the enthalpy of the various substances, inasmuch as only differences in enthalpy are measurable. For example, the arbitrary base selected in reference 3 was a value of zero at \( 298.15^\circ K \) \( (H_{298.15}^o = 0) \) for a selected set of elements. This selection makes the assigned value, \( H_{298.15}^o \) of any substance equal to its heat of formation at \( 298.15^\circ K \) from this set of selected elements.

**ASSIGNED REFERENCE ELEMENTS**

The designation of an element in a particular phase to be a reference element is needed in order that values of heats of formation and equilibrium constants be unambiguously related to specific reactions. Some reference elements which are commonly
found in the literature are the following (see ref. 3): the inert gases, He, Ne, and Ar; the diatomic gases, H₂, N₂, O₂, F₂, and Cl₂; and the condensed elements, Li(c,l), Be(c,l), B(c,l), C(graphite), Na(c,l), Al(c,l), Si(c,l), P(c IV, c III, l), and S(c II, cI, l) where c is a crystal phase and l is a liquid phase. Assigned reference elements used for the examples in this report were taken from this set.

HEATS OF FORMATION AND EQUILIBRIUM CONSTANTS

In the program described in this report, heats of formation and log K for a species are calculated as a function of temperature for two reactions. These reactions are for the formation of the species from the elements in either their assigned reference state discussed previously or in their atomic gaseous state.

The following are examples of how these properties are calculated for CO(g) at 1000° K:

Relative to reference elements,

\[ \Delta H_{1000}^{\circ} = (H_{1000}^{\circ})_{\text{CO}} - (H_{1000}^{\circ})_{\text{C(graphite)}} - \frac{1}{2} (H_{1000}^{\circ})_{\text{O}_2} \]  

\[ \Delta F_{1000}^{\circ} = (F_{1000}^{\circ})_{\text{CO}} - (F_{1000}^{\circ})_{\text{C(graphite)}} - \frac{1}{2} (F_{1000}^{\circ})_{\text{O}_2} \]  

or relative to gaseous atoms,

\[ \Delta H_{1000}^{\circ} = (H_{1000}^{\circ})_{\text{CO}} - (H_{1000}^{\circ})_{\text{C}} - (H_{1000}^{\circ})_{\text{O}} \]  

\[ \Delta F_{1000}^{\circ} = (F_{1000}^{\circ})_{\text{CO}} - (F_{1000}^{\circ})_{\text{C}} - (F_{1000}^{\circ})_{\text{O}} \]

By definition,

\[ \log_{10} K = \frac{-\Delta F_{T}^{\circ}}{2.3025851 RT} \]  

COMPUTER PROGRAM

The computer program was written for an IBM 7094 with 32 thousand core storage.
and IBM 1403 printers with 132 print positions. FORTRAN tape 3 is used as a binary scratch tape. Input and output tapes are FORTRAN tapes 5 and 6, respectively.

The program consists of a main routine and 17 subroutines written in FORTRAN IV and, in addition, five Lewis Research Center subroutines written in 7094 MAP assembly language. A listing of the FORTRAN program is given in appendix B and a discussion of the routines is given later.

A listing and brief discussion of the five Lewis subroutines (named SKFILE, BCDUMP, BCREAD, IALS, and IARS) are given in appendix C. These MAP routines require version 13 IBSYS operating system.

Availability to Other Organizations

The source program decks will be made available on written request to the authors. The input data used for the examples in this report will be included for check out purposes. In addition, for use in calculating log K, the enthalpy and free energy data for at least the first 18 elements in their atomic gas as well as their assigned reference state will be included. These data are essentially those of reference 2.

The following sections give a general discussion of the program. Included in this discussion are options, input, output, general flow of the program, and subroutines.

Options

The program provides a choice of several methods for calculating the thermodynamic functions \( C_p^O, H_T^0 - H_0^0, H_T^0 - H_298^0, 15, S_T^0, -(F_T^0 - H_0^0), \) and \( -(F_T^0 - H_298^0, 15) \). For ideal gases, these functions may be obtained from one of several assumed forms of the partition function or else from empirical equations. For solids and liquids, the thermodynamic functions may be calculated only from empirical equations. In addition, thermodynamic functions for any phase of a species may be read directly from cards for additional processing.

The program also has two other capabilities which are optional: (1) least-squares fitting of the thermodynamic functions to empirical equations (eqs. (10) to (12)) and (2) calculating heats of formation and log K values for the same temperature range as the functions.

The following is a discussion of these optional features.

Partition functions - monatomic gases. - The partition function for monatomic gases is given by equation (7). The program permits three optional ways of terminating the number of energy levels \( L \) to be included in calculating this partition function.
These three options, indicated by their program code names given in capital letters, are:

(1) **ALLN** - inclusion of all electronic levels in the input data,
(2) **FIXEDN** - inclusion of all levels through a specified principal quantum number \( n \), and
(3) **TEMPER** - inclusion of all energy levels that are less than or equal to the ionization potential lowered by an amount \( kT \) (see section Cutoff methods).

With any of these three cutoff options, an additional option (FILL) is provided to include predicted but unobserved levels automatically (see discussion in the section Inclusion of predicted levels).

**Partition functions** - diatomic and polyatomic gases. - For diatomic and polyatomic gases, the program provides for a selection of five methods of calculating the partition function which varies in the inclusion of and formulas for the correction terms (\( \ln Q_\rho \), \( \ln Q_\theta \), \( \ln Q_W \), and \( \ln Q_C \)) in equation (9). The formulas for the \( \ln Q \) terms included in each of the five methods are given in tables I and II. If certain spectroscopic constants are not available as input, the program automatically excludes those \( \ln Q \) terms involving them. The methods (with their program code names in parentheses) are as follows:

(1) **Rigid-Rotator Harmonic-Oscillator** (RRHO) approximation - This method excludes all the correction terms in equation (9) (i.e., \( \ln Q_\rho \), \( \ln Q_\theta \), \( \ln Q_W \), and \( \ln Q_C \)).

(2) **Modified Pennington and Kobe** (PANDK) method - The formulas given in table II for \( \ln Q_C \) are similar to those given in reference 29. The method in this report is equivalent to the one described in reference 3 except for the formula for \( \ln Q_\theta \) (formula 6 in table I). All correction terms in equation (9) are included with the exception of the Fermi resonance \( \ln Q_W \) as indicated in table I.

(3) **Joint Army Navy Air Force** (JANAF) method - This method is described and used in reference 2. For diatomic molecules, it is the same as the PANDK method except for the definitions of \( a_1 \) and \( X_{11} \) which are used in formulas 9 and 12, respectively, in table II. For polyatomic molecules, the JANAF method is the same as the RRHO method.

(4) **Nonrigid-Rotator Anharmonic-Oscillator 1** (NRRAO1) - In addition to the \( \ln Q_\theta \) and \( \ln Q_\rho \) terms, all the \( \ln Q_C \) terms given in references 30 and 31 were included which do not contain a \( (c_2/T)^2 \) or \( (c_2/T)^3 \) factor.

(5) **Nonrigid-Rotator Anharmonic-Oscillator 2** (NRRAO2) - This method includes the same \( \ln Q_C \) terms as NRRAO1 with the addition of \( \ln Q_C \) terms from references 30 and 31 which contain \( (c_2/T)^2 \) factors.

**Thermodynamic functions from empirical equations.** - The routine for calculating thermodynamic functions from the empirical equations (eqs. (10), (11), and (12)) has the following features:

(1) The value of \( r \) (number of coefficients \( a_i \)) may be any number from 1 to 10.
(2) The temperature exponents \( q_i \) may be any positive or negative numbers or zero.
(3) Any number of sets of \( a_i \) and \( q_i \) may be read in for various temperature intervals for a particular species.

(4) The integration constants, \( a_{r+1} \) and \( a_{r+2} \) may be read in or calculated by the program from the enthalpy and entropy values, respectively, for a specific temperature.

(5) When a phase transition occurs, the integration constants, \( a_{r+1} \) and \( a_{r+2} \) for the second phase may be read in or calculated by the program from either the enthalpy or entropy of transition.

(6) There is an option to punch on binary cards up to five coefficients and two integration constants for each temperature interval. This option has been included in order to provide thermodynamic data in the form required by reference 33.

Least-squares fit. - The least-squares routine fits the thermodynamic functions to equations (10), (11), and (12). The routine has the following features:

(1) The value of \( r \) (number of coefficients \( a_i \)) may be any number from 1 to 10.

(2) The temperature exponents \( q_i \) may be any positive or negative numbers or zero.

(3) An option is provided to permit the data to be divided into any number of specified intervals from 1 to 9. The purpose in providing for several intervals is to increase the accuracy of the fit.

(4) The equations for each temperature interval are constrained at an endpoint to fit either the original data or the values obtained from fitting an adjacent interval. The purpose of these constraints is to give equal values of the functions at the common point and thus avoid discontinuities between consecutive intervals. However, only one temperature may be specified in the input for which the fitted equations reproduce the original values. (If no temperature is specified, the program assigned 1000° K.)

(5) For two or more phases, the data for each phase is fitted separately and the equations constrained to fit the original data at the transition point.

(6) For each temperature interval, up to five of the coefficients \( a_i \) plus the two integration constants will be punched on binary cards. These cards are made in order to provide thermodynamic data in the form required by reference 33.

Heat of formation and \( \log K \) values. - The program provides an option for calculating heats of formation and \( \log K \) values as a function of temperature for two reactions. The reactants for these two formation reactions are either monatomic gases or assigned reference elements (see sections Assigned Reference Elements and Heats of Formation and Equilibrium Constants).

Heats of formation and \( \log K \) values for a particular species can be calculated if the necessary enthalpy and free energy data for the reactants as well as for that species are available. Therefore the monatomic gases and assigned reference elements are processed first. For these reactants, there is an option to reserve the enthalpy and free energy data in two ways: (1) by writing the data on tape and (2) by punching the data on cards. The data on tape are saved only for use with other species being processed. 11
during the same computer run. For later computer runs, the data on the binary cards may be read in as part of the input and, if so, are automatically put on tape.

If there is a temperature in the data for a particular species which is not contained in the data on tape for the required reactants, the reactant data are interpolated using three-point Lagrangian interpolation.

**Input**

**Types of data.** - The input data are grouped into two categories; namely, general and specific. General data are read into storage and retained for use with any number of species to be processed in any particular computer run. Physical constants, atomic weights, and reactant enthalpy and free energy values fall into this category of input data. (See previous section.)

On the other hand, a set of specific data cards is required for each species to be processed. The data in each set are read, processed, and cleared before the next set is read. A set of specific data cards for a diatomic gas would contain the chemical formula; the method of calculation, such as PANDK; molecular data such as $\omega_e$, $\omega_e x_e$, $B_e$, and $\alpha_e$; desired options such as a least-squares fit or a special temperature schedule; and finally, a card to indicate the end of the set of specific data.

**Identification of cards.** - All input cards will be referred to in one of the following three ways:

1. Most cards will be identified by the code word punched in card columns 1 to 6. For example, the input card containing physical constants has the code CONSTS in these columns. Thus, this card will be referred to as the CONSTS card.

2. The first card of a set of specific data cards has the chemical formula punched in card columns 1 to 12. This card will be referred to as a formula card. The word "formula" does not appear on the card.

3. Column binary cards containing enthalpy and free energy data for the reactants will be referred to as binary EF data cards. The word "EF data" does not appear on the card.

**Uniform format.** - All cards of types (1) and (2) are read with a single format which will be referred to as the uniform format. Format details are given in appendix D.

**Contents of individual cards.** - A brief description of the contents of the individual cards is given in table III. (Detailed descriptions are given in appendix D.) The right-hand column indicates which cards are optional. Table III indicates that the card code in card columns 1 to 6 is a mnemonic device which does one or more of the following:

1. Indicates what data are on the card (i.e., CONSTS, ATOM, EFDATA, TEMP, LSTSQS and DATA)
(2) Indicates an option discussed in the section Options (i.e., LOGK, LSTSQS, EFTAPE, and METHOD)
(3) Identifies the data on the binary cards which follow it (i.e., EFDATA)
(4) Calls for some intermediate output (i.e., LISTEF and INTERM)
(5) Identifies the input data sources (i.e., REFNCE) or gives a date (i.e., DATE)
(6) Indicates the end of a set of specific data (i.e., FINISH)

General Flow of Program

The general flow of the program is given in figure 1. For convenience in locating various sections of the FORTRAN program, 79 location numbers, referred to as C10, C20, . . . , C790, were included as comments in the program. Some of these location numbers are also shown in figure 1. Subroutine names are given in parentheses.

From figure 1, the following are evident:

1. Each card (except for the binary EF data cards) is read and listed. The flow is directed according to the code in card columns 1 to 6.
2. The general data storage is cleared only at the beginning of each computer run. Thus, these data are retained as they are read in.
3. The order of the general data is immaterial except for the fact that the EF DATA and binary EF data cards must remain in sets for each reactant.
4. The specific data (including options) are cleared at the beginning of the program and after each FINISH card.
5. There may be any number of sets of specific data - each having any combination of options.
6. The order of the optional cards (EFTAPE, LOGK, LSTSQS, INTERM, DATE, and REFNCE) in the specific data is immaterial.
7. The temperature schedule (TEMP cards), if not the standard 100 (100) 6000° K, must be read before the METHOD card.
8. The DATA cards must follow the METHOD cards.
9. Any card which is not recognized by the code in card columns 1 to 6 is assumed to be a formula card.
10. From the chemical formula, the following items are determined by the program:
   a) the molecular weight
   b) the phase of the species
   c) the number of atoms (i.e., whether species is monatomic, diatomic, or polyatomic)
11. The $H_0^O$ value may be calculated from an assigned value at any temperature or a heat of reaction (see formula card in appendix D). (The $H_0^O$ value is used in calcu-
lating $\Delta H_T^0$ and log K and the integration constants $a_{r+1}$ (eq. (11)).

(12) Thermodynamic functions are calculated immediately after the DATA cards are read.

(13) After the FINISH card is read, $H_0^0$ is calculated, the least-squares fit option is checked, tables of thermodynamic functions are listed, and the $\Delta H_T^0$ and log K option is checked.

(14) General data may be modified or added following any FINISH card. If a second CONSTS card, ATOM card for a particular atom, or a second set of EFDATA and binary EF data cards for a particular reactant is read, the data on these cards will be used for succeeding calculations.

(15) With an EFTAPE option card in a set of specific data, EFDATA and binary EF data cards are punched and the data are put on tape. The data on tape will be available for use with any succeeding calculations in the same computer run.

(16) Any number of sets of METHOD and corresponding DATA cards may be read for a set of specific data. This is useful for species with more than one phase in the temperature range of interest. For example, the thermodynamic functions for the solid may be read in directly while the liquid data may be obtained from empirical equations. The data for both phases will appear in the same listed tables of the thermodynamic properties.

A feature of the program which is not indicated in the flow diagram is that contributions of excited electronic states may be included in the calculation of the thermodynamic functions for diatomic and polyatomic gases. There may be any number of states, each having its own set of molecular constants. This is accomplished by grouping the DATA cards for each state together with a code number in card columns 79 and 80. The values of $Q_m$, $T dQ_m/dT$, and $T^2 d^2Q_m/dT^2$ are calculated after the DATA cards for each state are read. These values are summed as they are calculated.

Output

A brief description of punched card and listed output is given in this section; a detailed description is given in appendix E.

Punched card output. - Cards are punched with certain options as indicated by the following:

(1) With an EFTAPE specific data card, an EFDATA card and binary EF data cards are punched.

(2) With a LSTSQS card, column binary cards are punched which contain the chemical formula of the species, the temperature intervals, and the least-square coefficients (eqs. (10) to (12)).
(3) With DATA cards which contain coefficients (eqs. 10 to 12) as well as a TPUNCH code, the coefficients will be punched in the same format as item (2). (The TPUNCH code is described in appendix D and its uses illustrated in example 5, appendix F.)

Listed output. - The following data are always listed:

(1) The contents of all input cards in the uniform format

(2) Table of T, C_p^0/R, (H^0_T - H^0_0)/RT, (H^0_T - H^0_298.15)/RT, S^0_T/R, -(F^0_T - H^0_0)/RT, -(F^0_T - H^0_298.15)/RT, H^0_T/RT, and -F^0_T/RT

(3) Table of T, C_p^0, H^0_T - H^0_0, H^0_T - H^0_298.15, S^0_T, -(F^0_T - H^0_0), -(F^0_T - H^0_298.15), H^0_T, and -F^0_T

The following data are listed only with the indicated options:

(1) With a LISTEF card, the contents of the binary EF data cards are listed.

(2) With an INTERM card included in the specific data of a particular species, intermediate data are listed as detailed in appendix E.

(3) With a LSTSQS card, the following data are listed for each temperature interval fitted:

(a) The thermodynamic functions (both the original and those obtained from the least-squares fit

(b) The errors between the original and the fitted data

(c) The least-squares equation for heat capacity and the integration constants (eqs. 10 to 12)

(d) The contents of the punched binary cards (see item (2) in the section

Punched card output)

(4) With a LOGK card, two tables are listed:

(a) Table of T, C_p^0/R, (H^0_T - H^0_0)/RT, S^0_T/R, -(F^0_T - H^0_0)/RT, H^0_T/RT, -F^0_T/RT, and \( \Delta H^0_T/RT \) and \( -\Delta F^0_T/RT \) for formation of the species from both assigned reference elements and monatomic gases

(b) Table of T, C_p^0, H^0_T - H^0_0, S^0_T, -F^0_T, and \( \Delta H^0_T \) and \( \log_{10} K \) for the same reactions as the previous table

Examples

Sample problems with punched card input and listed output are given in appendix F.
Main Routine and Subroutines

The FORTRAN listing in appendix B has a number of comments to indicate the operations of various sections of the program as well as location numbers C10, C20, . . . , C790. A short description of each subroutine follows.

PAC1(C10 to C60). - This is the main routine and directs the general flow of the program as given in figure 1 and discussed in the section General Flow of the Program. Subroutines called by PAC1 are indicated in figure 1 in parenthesis in or near the appropriate boxes.

INPUT(C70). - This routine reads and lists all cards that have been punched in the uniform format. The output format for listing numerical values is varied according to the size of the numbers.

PAGEID(C80). - This routine lists the chemical formula at the bottom of a page in the output listing and skips to a new page. The program allows approximately 55 lines to be printed on a page.

EFTAPE(C90 to C130). - This routine (1) reads binary EF data cards, (2) punches new sets of EFDATA and binary EF data cards, and (3) stores these data on tape.

IDENT(C140 to C160). - This routine analyzes the chemical formula on either the formula card or the EFDATA card. It separates and stores each chemical symbol and corresponding number of atoms in the chemical formula. The chemical symbols are matched in the SYMBOL array and corresponding indexes are stored.

When analyzing a chemical formula from a formula card, the molecular weight is calculated.

TEMPER(C170 to C180). - This routine stores the temperature schedule as given on one or more TEMP cards. It is called from PAC1 after a TEMP card has been read.

RECO(C190 to C250). - This routine processes the METHOD and DATA cards for methods READIN and COEF. The routine is called from PAC1 after a METHOD card has been read with either a COEF or READIN code. The RECO routine calls INPUT to read the DATA cards plus the next card.

For READIN, the thermodynamic functions on each card are simply stored. For COEF, the thermodynamic functions are calculated and stored.

The RECO routine is also used to relate the enthalpy of two phases of the same species by means of an enthalpy or entropy of transition. One of these transition values is given on the METHOD card of the second phase (DELTAH or DELTAS code described in appendix D) and used to calculate the enthalpy of the second phase at the transition temperature. The free energy value of the second phase is taken to be equal to the free energy value of the first phase at the transition temperature.

If a transition is present, the routine calls DELH (discussed in the section DELH (C510 to C530)) to check for the options of least-squares fit or punching coefficients for
the first phase.

ATOM(C260 to C310). - This routine calculates thermodynamic functions for monatomic gases.

The routine calls INPUT to read all DATA plus the next card. The J values, which are read with an alphanumeric format, are changed to floating point numbers and stored.

Energy levels are sorted in order of increasing energy values. The number of levels included in the calculations is determined by the cutoff method (ALLN, FIXEDN, or TEMPER) given on the METHOD card. Predicted but unobserved levels will be included with the FILL option.

POLY(C320 to C410). - This routine calculates thermodynamic functions for diatomic and polyatomic gases.

Subroutine INPUT is called to read the DATA cards plus the next card. Subroutine LINK1 is called to calculate the partition function according to the method specified (RRHO, PANDK, JANAF, NRRAO1, or NRRAO2).

If more than one electronic state is present, the various states are identified by a code in card columns 79 to 80. In this case, DATA cards for only one state at a time are read in and stored. The partition function for each state is calculated prior to processing DATA cards for the next state.

LINK1(C420 to C480). - This routine calculates the partition function for diatomic and polyatomic molecules. The formulas given in tables I and II are evaluated according to the method specified.

The routine is called from subroutine POLY. LINK1 in turn calls two subroutines, DERIV to calculate the derivatives of the partition function and QSUM to keep a running total of the various contributions to the partition function.

KD(C480). - This function subprogram calculates Kronecker delta.

DERIV(C490). - This routine calculates the derivatives of the partition function using the method given in the footnote of table II. The routine is called from a number of places in LINK1.

QSUM(C500). - This routine keeps a running total of all, except translational, contributions to the partition function and its derivative for each electronic state. These contributions are listed if an INTERM card has been included in the input.

QSUM is called from a number of places in LINK1.

DELH(C510 to C530). - This routine calculates the $H_0^0$ value, calls LEAST for the least-squares fit option, and calls PUNCH for the option of punching coefficients read with the COEF method. The information given on the formula card ($\Delta H_T^0$ of formation, $D_T^0$, $H_T^0$, $T$) is used in calculating the $H_0^0$ value.

The routine is called from PAC1 after the FINISH card has been read. However, it will also be called from RECO for phase transition points. In this latter case, any
processing (the $H^0$ calculation, the least-squares fit, or the punching of coefficients) will be for the species phase coming ahead of the transition point in the input. For example, for a species with input data for the solid followed by the liquid, DELH will process the solid when it has been called from RECO. The liquid will be processed when DELH is called from PAC1.

**TABLES(C540 to C570).** - This routine lists tables of thermodynamic functions as discussed in appendix E. The output format varies depending on the availability of the following values: (1) the $H^0_{298.15} - H^0_0$ value which is required in obtaining $H^0_T - H^0_{298.15}$ and $-(F^O_T - H^0_{298.15})$, and (2) the $H^0_0$ value which is required in obtaining $H^0_T$ and $-F^O_T$.

**LOGK(C580 to C650).** - This routine is called only if a LOGK option card has been included in the input. It calculates $\Delta H^O_T/RT$, $\Delta H^O_T$, $-\Delta F^O_T/RT$, and $\log K$ for the formation of the species from the assigned reference elements and the monatomic gases. The required enthalpy and free-energy data for these reactants have been stored on tape by the EFTAPE subroutine.

The LOGK routine lists two tables of properties as detailed in appendix E. If any reactant species for either of the formation reactions is not on tape, the appropriate columns in these tables are left blank.

**LEAST(C660 to C760).** - This routine is called from DELH only if one or more LSTSQS cards have been included in the input. It calculates the least-squares coefficients, lists certain information detailed in appendix E, and calls PUNCH to punch the coefficients on cards.

**PUNCH(C770 to C790).** - This routine punches binary cards containing the coefficients obtained either from a least-squares fit or from the DATA cards associated with method COEF. For these two options, PUNCH is called from subroutines LEAST and DELH, respectively.

The contents of each card are listed in the order they are punched. See output details in appendix E.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, February 14, 1967,
129-01-02-01-22
APPENDIX A

SYMBOLS

$A_e, B_e, C_e$: rotational constants corresponding to equilibrium separation of atoms

$A_0, B_0, C_0$: rotational constants for lowest vibrational state

$a_i$: polynomial coefficients used in eqs. (10) to (12)

$a_{r+1}$: integration constant defined by eq. (11)

$a_{r+2}$: integration constant defined by eq. (12)

$b$: constant defined in eq. (8)

$C_p^o$: heat capacity at constant pressure for standard state

$c$: velocity of light or crystal phase of chemical substance

$c_2$: second radiation constant, $hc/k$

$D, D_e$: spectroscopic constants for rotational stretching

$D_0, D_{000}$: rotational stretching constants for lowest vibrational state

$D_T^o$: dissociation energy at temperature $T$ for standard state

$d_i$: degeneracy associated with $\nu_i$

$F_T^o$: $\left( F_T^o - H_0^o \right) + H_0^o$

$F_T^o - H_0^o$: sensible free energy at temperature $T$ relative to $0^o K$ for standard state

$F_T^o - H_{298.15}^o$: sensible free energy at temperature $T$ relative to $298.15^o K$ for standard state

$g_i, g_m$: electronic statistical weight

$g_{II}$: anharmonicity constant for doubly degenerate vibrations in linear molecules

$H_0^o$: chemical energy at $0^o K$ for standard state

$H_T^o$: $\left( H_T^o - H_0^o \right) + H_0^o$

$H_T^o - H_0^o$: sensible enthalpy at temperature $T$ relative to $0^o K$ for standard state

$H_T^o - H_{298.15}^o$: sensible enthalpy at temperature $T$ relative to $298.15^o K$ for standard state

$h$: Planck's constant
\( I_A, I_B, I_C \) principal moments of inertia
\( J_i, J_m \) total angular momentum quantum number
\( K \) equilibrium constant
\( k \) Boltzmann constant
\( L \) total number of electronic energy states
\( \ell \) liquid phase of chemical substance
\( M \) molecular weight
\( N_0 \) Avogadro's constant
\( n \) number of unique frequencies or principal quantum number
\( p \) partial pressure
\( Q \) internal partition function
\( Q^m \) internal partition function for \( m \)th electronic state
\( Q^m_c \) correction factor to the partition function for anharmonicity and vibration-rotation interaction for \( m \)th electronic state
\( Q^m_e \) electronic partition function for \( m \)th electronic state
\( Q^m_R \) classical-rotation partition function for \( m \)th electronic state
\( Q^m_V \) harmonic-oscillator partition function for \( m \)th electronic state
\( Q^m_W \) Fermi resonance correction factor to partition function for \( m \)th electronic state
\( Q^m_\rho \) low temperature rotational correction factor to partition function for \( m \)th electronic state
\( Q^m_\theta \) rotational-stretching correction factor to partition function for \( m \)th electronic state
\( q_i \) temperature exponents in eq. (10)
\( R \) universal gas constant
\( r \) number of coefficients \( a_i \) in eq. (10)
\( S_c \) constant defined by eq. (5)
\( S^0_T \) entropy for standard state
\( T \) temperature, °K
\( T_0 \) electronic excitation energy between lowest vibrational states (\( \nu = 0 \)) of ground and excited state for diatomic and polyatomic gases
\begin{align*}
u_i & \quad c_2 \nu_1 / T \\
v, v_1 & \quad \text{vibrational quantum number} \\
W_0 & \quad \text{Fermi resonance constant} \\
x_{ij}, y_{ijk} & \quad \text{anharmonicity constants for polyatomic molecules} \\
\alpha_e, \alpha_1 & \quad \text{vibration-rotation interaction constants for diatomic and linear polyatomic molecules} \\
\alpha_A, \alpha_B, \alpha_C, \alpha_{1j} & \quad \text{vibration-rotation interaction constants for polyatomic molecules} \\
\beta_i & \quad \text{rotational-stretching – vibration interaction constant} \\
\epsilon_m & \quad \text{energy of } m^{th} \text{ electronic state} \\
\nu_i & \quad \text{observed fundamental frequency} \\
\rho & \quad \text{rotational-stretching spectroscopic constant} \\
\sigma & \quad \text{symmetry number} \\
\omega_e & \quad \text{zero-order vibrational frequency for diatomic molecule} \\
\omega_e x_e, \omega_e y_e, \omega_e z_e & \quad \text{anharmonicity constants for diatomic molecules}
\end{align*}
APPENDIX B

FORTRAN LISTING (FORTRAN ROUTINES)

MAIN PROGRAM - PAC1

TEST11) LIST EF DATA
TEST(3) SPECIES IS AN ION.
TEST(4) SPECIES IS A GAS.
TEST(5) SPECIES IS A LIQUID.
TEST(6) SPECIES IS A SOLID.
TEST(7) SUBROUTINE HFTAPE IS CALLING SUBROUTINE IDENT.
TEST(8) AN ASSIGNED M IS AVAILABLE
TEST(9) CP/R. H-H/RT, AND S/R ARE READY TO BE OUTPUTTED
TEST(10) SPECIES TO BE REACTANT. PUNCH EF DATA AND WRITE ON TAPE.
TEST(12) LOG K CALLED FOR
TEST(13) DATA ARE IN THE FORM, H-H298 AND -(F-H298)
TEST(14) INTERMEDIATE OUTPUT CALLED FOR
TEST(15) LEAST SQUARES CALLED FOR
TEST(16) ERROR IN INPUT. GO TO NEXT SPECIES
TEST(17) PUNCH READ-IN COEFFICIENTS
TEST(18) ENTHALPY IS ABSOLUTE
TEST(19) SPECIES IS
TEST(20) TEMPERATURE SCHEDULE HAS BEEN STORED

COMMON NAME(12), SYMBOL(70), ATWIT(70), R, HCK, NEL, ICARD, WORD(5), FORMULA, MPLACE(110), LPLACE(110), NMLA(110), NOFILE
COMMON PCH/LEVEL, NF1, NF2, ANS(9,15), TCI(10), NTC, NF, LDATE, NNN, NLAST

INTEGER FORMLN, SYMBOL, SYMBOL, ELEMNT
LOGICAL TEST

INITIALIZE ONCE.

3 FORMAT(1H1)
TEST(1) = .FALSE.
R = 0
HCK = 0
NEL = 0
REND 3
END FILE 3

DO 22 = 1, 106
MPLACE(i) = 0
LPLACE(i) = 0

32 NMLA(i) = 0

INITIALIZATION FOR EACH SPECIES. FOLLOWS FINISH CARD .

103 DO 101 =1,10
EXP(i) = 0.0
101 TRANGE(i) = 0.0
TCONST = 0.0
DO 109 = 1, 3, 20
109 TEST(i) = .FALSE.
LDATE = 0
NAME(i) = IBLNK
NAME(2) = IBLNK
IHEAT = IBLNK
TINTVL = 0.0
NT = 0
NIT = 1
NMP = 1
NMP = 1
ASINDT = 0.0
ASINDH = 0.0
SPECH = 0.0
H298HR = 0
LEVEL = 1
NTC = 0
NPR = 0
IFX = 0
ITR = 0
NF = 0
NFI = 1
DO 102 = 1, 202
C CALL INPUT TO READ AND WRITE CONTENTS OF ONE INPUT CARD

C

104 CALL INPUT (LINES)

194 IF(I CARD.EQ.IFINS HIGH) GO TO 111

IF(I CARD.EQ.LIST) GO TO 2

IF(I CARD.EQ.INTERN) GO TO 209

IF(I CARD.EQ.IDATE) GO TO 205

IF(I CARD.EQ.ITEMP) GO TO 105

IF(I CARD.EQ.MEXT) GO TO 107

IF(I CARD.EQ.IHF TAP .OR. (I CARD.EQ.IEFTAP)) GO TO 110

IF(I CARD.EQ.I LGK) GO TO 319

IF(I CARD.EQ.IREF) GO TO 104

IF(I CARD.EQ.NOLEAS) GO TO 106

IF(I CARD.EQ.LSTSOS) GO TO 180

IF(I CARD.EQ.IATOM) GO TO 13

IF(I CARD.EQ.ICONST) GO TO 5

IF(I CARD.EQ.IHF DAT .OR. I CARD.EQ.IEFDAT) GO TO 147

C

C If CC 1-6 contain no recognizable code, assume card contains formula

C

C CALL IDENT TO ANALYZE FORMULA

CALL IDENT

IF (TEST(161)) GO TO 152

DATA I9DEL/I9DELT=/9DIS/9DISOC/.1ASH/6HASIND/.1INT/.1HT/

DATA I9NCM/5I9NCM/.KCAL/4HKCAL/.1EV2/.1HEV/.JOULES/6JOULES/

DATA I9CAL/3HKCAL./1PZ2/.IPZATOMP/.IPZATOM/.IHF298/5HKF298/

C

C STORE HEAT OF REACTION AND ASSIGNED T FROM FORMULA CARD

DO 121 I = 2,4

IF(I WORD(I).EQ.IDELH .OR. I WORD(I).EQ.IDIS) GO TO 122

IF(I WORD(I).EQ.IHT) ASINOT = WORD(I)

IF(I WORD(I).EQ.IHF298) GO TO 125

IF(I WORD(I).EQ.IF) PT = WORD(I)

IF(I WORD(I).NE.IPATOM) GO TO 121

IHEAT = IDIS

ASINOH = -WORD(I)

GO TO 121

125 IHEAT = IASH

ASINOT = 298.15

GO TO 126

122 IHEAT = IWORD(I)

126 ASINOH = WORD(I)

121 CONTINUE

IF(IHEAT.NE.19BLK .AND. ASINOT.EQ.0.) TEST(19) = .TRUE.

IF(IHEAT.IASH .AND. ASINOT.EQ.0.) TEST(8) = .TRUE.

C

C CONVERT HEAT OF REACTION TO PROPER UNITS IF NECESSARY.

CONV = 1.

DO 123 I = 2,4

IF(I WORD(I).EQ.INVCN) CONV = 2.85927

IF(I WORD(I).EQ.KCAL) CONV = 1000.

IF(I WORD(I).EQ.VEV) CONV = 2306.3

IF(I R.GT.0.0 .AND. (I WORD(I).EQ.ICAL .OR. CONV .LE. 1.) ) CONV = CONV/4.184

123 CONTINUE

ASINOH = ASINOH*CONV

GO TO 104

C

C STORE GENERAL DATA

C

2 TEST(1) = .TRUE.

GO TO 104

DATA LTRCN/6HOO0000E/.

13 X = AND (MASK, I WORD(I))

IF(IX.EQ.IWORD(I)) GO TO 20

SYM = IARSZ(I WORD(I))

GO TO 21
20 SYMBl = IARSI30, l WORD(l))
       IF (NATOM.EQ.0) GO TO 33
21       DO 30 IND = 1,NATOM
       IF (SYMBOL.EQ.SYMBOL(IND)) GO TO 35
       CONTINUE
30      NATOM = NATOM + 1
       IF ISYMRL.EQ.SYMBOL(INO1) GO TO 35
       INO = NATOM
       SYMBOL(INO) = SYMBL
       ELEMENT(INO) = l WORD(2)
       AGI(INO) = l TRON(I)
       EGI(INO) = l CONSl
       SCONST = WORO(INO)
       GO TO 104
5      CONTINUE
      GO TO 104
14     CALL EFTAPE
      GO TO 104
C
C STORE OPTIONS. SEE C60 FOR LSTSQS OPTION.
205      DO 206 IJ=1,4
206     IF WORD(IJ).NE.1BLNK) LDAT = l WORD(IJ)
      CONTINUE
      GO TO 104
209    TEST(I4) = .TRUE.
      GO TO 104
110   TEST(10) = .TRUE.
      GO TO 104
105    CALL TEMPER (NT,TINTV) ,T, l WORD,WORD)
       TEST(20) = .TRUE.
      GO TO 104
319   TEST(12) = .TRUE.
      GO TO 104
106    TEST(15) = .FALSE.
      ITR = 0
      NF = 0
      DN 2106 I=1,10
      EXP(I) = 0.
      GO TO 104
2106  TRANGE(I) = 0
      TCONST = 0.
      GO TO 104
C
C C60
C METHOD CARD HAS BEEN READ.
107      DO 2000 I = 1,4
2000     DATA IREAD/6HREADIN/, ICOEF/4HCOEF/
       IF l WORD(I).EQ.1READ.OR. l WORD(I).EQ.1COEF) ICARD = l WORD(I)
       IF l WORD(I).EQ.IPIPI I = WORD(I)
      CONTINUE
      IF ( R.EQ.0. ) GO TO 150
      IF TEST(20) OR ICARD.EQ.1READ) GO TO 130
      GO TO 130
C
C C STORE STANDARD T SCHEDULE IF NO TEMP CARDS HAVE BEEN READ.
      T(I) = 100.0
      T(2) = 200.0
      T(3) = 298.15
      T(4) = 300.0
      DO 131 NT = 5,61
      NT = NT + 100.0
131    IF NT = 100.0
      CALL RECO
      CALL ATOM FOR MONATOMIC GASES
      CALL POLY FOR DIATOMIC OR POLYATOMIC GASES
      IF ( ICARD.NE.1READ .AND. ICARD.NE.1COEF) GO TO 235
      CALL RECO
      GO TO 1161
235    IF (HCK.EQ.0. ) OR WEIGHT.EQ.0. ) GO TO 150
      IF I MATMS.EQ.1) GO TO 148
      IF I MATMS,EQ.2) GO TO 149
150 WRITE(6,151)  
C 151 FORMAT (5HOERROR IN INPUT. GO TO NEXT SPECIES, C40)  
C 152 IF (ICARD.EQ.IFINISH) GO TO 88  
READ (5, 151ICARD  
1 FORMAT(A6)  
GO TO 152  
88 TEST(16) = .FALSE.  
LINES = LINES + 2  
CALL PAGEIO (LINES)  
GO TO 103  
148 CALL ATOM  
GO TO 1160  
149 CALL POLY  
1160 NIT = NT + 1  
1161 IF (TEST(161) GO TO 152  
161 CALL PAGEIO (LINES)  
GO TO 194  
C 153  
C50  
C 111 IF (TEST(91) GO TO 112  
WRITE (6,113)  
113 FORMAT(54HOC/P,R,(H-HO)/RT, AND S/R ARE NOT READY FOR OUTPUT, C50)  
GO TO 103  
112 NLAST = NT  
C 113 CALL DELM TO CALCULATE HO IF NECESSARY. DELM WILL CALL LEAST FOR  
C LEAST SQUARES FIT IF OPTION HAS BEEN REQUESTED.  
IF(NNN.LT.NLAST) CALL DELM  
C 114 CALL TABLES TO PUNCH FIRST TWO TABLES OF FUNCTIONS.  
1367 CALL TABLES  
C 115 FOR EFTAPE OPTION, CALL HFTAPE TO PUNCH EF DATA AND PUT DATA ON TAPE.  
IF (TEST(10)) CALL EFTAPE  
C 116 IF LOGK OPTION, CALL LOGK TO CALCULATE DELTAH AND LOG K AND  
C PRINT TWO TABLES OF PROPERTIES.  
367 IF (TEST(12)) CALL LOGK  
GO TO 103  
C 117  
C60  
C STORE DATA FROM LSTSOS CARD.  
C 180 TEST (15) = .TRUE.  
DO 185 I = 1,4  
IF (IWORD(I) .EQ. IHT) GO TO 181  
IF (IWORD(I) .EQ. ITCONS) GO TO 186  
181 IF (IWORD(I) .EQ. IEXP) GO TO 183  
IF (IWORD(I) .EQ. TBLNK) GO TO 185  
WRITE (6,187) IWORD(I), WORD(I)  
187 FORMAT (1HO, A6, 39H IS AN INCORRECT LABEL FOR THE NUMBER--  
1 E12.4, 29H VALUE IGNORED, C60  
1 IF (IWORD(I) .EQ. NNS) GO TO 186  
186 ICONST = WORD(I)  
GO TO 185  
181 ITR = ITR + 1  
IF(ITR .GT. 10) GO TO 182  
TRANGE(ITR) = WORD(I)  
GO TO 103  
182 IF (IEXP .LT. 184)  
184 IF (IEXP .GT. 10)  
WRITE (6,184)  
184 FORMAT (69HOFIRST T'S ONLY WERE ACCEPTED FOR THE LEAST SQUARES  
1 ROUTINE, C60  
1 IF (IEXP .LE. 184)  
185 CONTINUE  
GO TO 104  
END
SUBROUTINE INPUT(LINES)
    
    COMMON ICARD, IWORD(5), WORD(4)
    
    DIMENSION FMT(12), WRD(15)
    
    DATA (FMT(1), I = 1, 31) / 15H1H0, 6X, A6, (/, (FMT(J), J = 5, 9, 2) / 3H6H6, A6/)
    
    L, /F12/1H, /FB/HF15.8, /F3/HF15.3, /F5/HF15.5, /E8/6HE15.8/)
    
    I = F12/6H2X, 12, /F1/HF15.0, /1B/HF6H9, A6, /)
    
    IF (NAME(1) .EQ. IB .OR. ISTART .NE. 0) GO TO 901
    
    CALL PAGEID(LINES)
    
    ISTART = 1
    
    WRITE (LINES, FMT) ICARD, IWORD(1), WRD(1), I = 1, 4, IWRD
    
    901 READ(5, 1) ICARD, IWORD(1), WORD(1), I = 1, 4, IWORD(5)
    
    1 FORMAT (2A6, F12.0, A6, F12.0, A6, F12.0, A6, F12.0, I12)
    
    DO 904 I = 1, 4
    
    J = 2*I+2
    
    IF (IWORD(1) .EQ. 0) GO TO 902
    
    WRD(I) = IWORD(I)
    
    ABSV = ABS(IWORD(I))
    
    FMT(J) = F8
    
    IF (ABSV .GE. 1000) FMT(J) = F5
    
    IF (ABSV .LE. 1.E-3) FMT(J) = F1
    
    GO TO 904
    
    902 FMT(J) = F8
    
    WRD(I) = B
    
    CONTINUE
    
    904 CONTINUE
    
    FMT(I1) = A2
    
    IWRD = IB
    
    IF (IWORD(5) .EQ. 0) GO TO 906
    
    FMT(I1) = F12
    
    IWRD = IWORD(5)
    
    WRITE (LINES, FMT) ICARD, IWORD(1), WRD(1), I = 1, 4, IWRD
    
    906 WRITE (LINES, FMT) ICARD, IWORD(1), WRD(1), I = 1, 4, IWRD
    
    IF (LINES .GE. 55) CALL PAGEID(LINES)
    
    RETURN
    
END
SUBROUTINE PAGE10 (LINES)
C
C PRINTS CHEMICAL FORMULA AT BOTTOM OF PAGE AND SKIPS TO NEXT SHEET.
C
COMMON NAME(2)
DATA SKIP /1H /, ZERO /1H0/
SKP = ZERO
50 IF (LINES .LT. 55) GO TO 400
   IF (LINES .GT. 57) SKP = SKP
   WRITE (6,100) SKP, NAME(1), NAME(2), NAME(1), NAME(2)
100 FORMAT (A1, 2A6, 95X, 2A6)
200 WRITE (6,300)
300 FORMAT (1H, /////)
   LINES = 4
   RETURN
400 WRITE (6,500)
500 FORMAT (1H )
   LINES = LINES + 1
   GO TO 50
END
SUBROUTINE EFTAPE  EFTP0001
  EFTP0002
  COMMON NAME(2),SYMBO(70),ATMWT(70),R,HCK,NEL,ICARD,IWORD(5)  EFTP0003
  1  WORD(4),TEST(20),WEIGHT,FORMULA(5),MLA(15),BLANK,ELEMENT(70)  EFTP0004
  2  NATOM,NT,GRIP(2021),NMAT(202),ASINDH,T(202),ASINDT,FHRT(202),  EFTP0005
  3  SPECH,TAPE(606),PTMELT,PEX(60),TRAN(6010),TCOSNT,NKIND,  EFTP0006
  4  NP,LINES,ITR,NTMP,AG(701),GG(701),NIT,PI,HZ98HR,HEAT,JP(5)  EFTP0007
  5  EFTP0008
C  EFTP0009
C90  EFTP010
C  EFTP011
  EFTP012
  LOGICAL TEST  EFTP013
  INTEGER SYMBOL, ELEMENT, FORMULA  EFTP014
C  EFTP015
  TEST(10)—PUNCH EF DATA AND PUT DATA ON TAPE FOR REACTANT WITH  EFTP016
  EFTAPE CARD IN SPECIFIC DATA.  EFTP017
  IF (.NOT.TEST(10)) GO TO 147  EFTP018
  REMIND 3  EFTP019
  NDF = NDFE + 1  EFTP020
  CALL/SKFILE(NDF,NDF)  EFTP021
  DATA IN/0/971200, MELTPT/6MELTPT/ITNO/6H TD./  EFTP022
  DATA IN/0/971200, MELTPT/6MELTPT/ITNO/6H TD./  EFTP023
  IWORD(I) = NAME(1)  EFTP024
  WORD(2) = ASINDH  EFTP025
  WORD(3) = PTMELT  EFTP026
  WORD(4) = NT  EFTP027
  WORD(5) = 0  EFTP028
C  EFTP029
C100  EFTP030
  CALL BCREAO(TAPE(A),TAPE(NX))  EFTP031
  FORMAT(9I4,1H1,1I4,1A6,1F12.4,1A6,1F12.4,1A6)  EFTP032
  WRITE (A10) NAME(I),HOT,ASINDH,MELTPT,PTMELT,ITNO,NT  EFTP033
  WRITE (A10) NAME(I),HOT,ASINDH,MELTPT,PTMELT,ITNO,NT  EFTP034
  10 FORMAT(IHO,6X,A6,6X,15X,25X,A6,F15.4,6X,A15)  EFTP035
  IWORD(I) = NAME(I)  EFTP036
  KX = 0  EFTP037
C  EFTP038
C  EFTP039
  CALL BCDUMP(TAPE(I),TAPE(NX))  EFTP040
  CALL BCDUMP(TAPE(I),TAPE(NX))  EFTP041
  C  EFTP042
  READ IN BINARY EF DATA AND PUT ON TAPE 3  EFTP043
  ORDER OF WORDS ON TAPE FOR EACH ELEMENT OR ATOM—  EFTP044
  1. NAME (IWORD(I)) ON EF DATA CARD  EFTP045
C  EFTP046
  2. NAME (IWORD(I)) ON EF DATA CARD  EFTP047
  3. MELTPT (WORD(3)) ON EF DATA CARD  EFTP048
C  EFTP049
  4. TD. (WORD(4)) ON EF DATA CARD  EFTP050
C  EFTP051
  5. TEMPS (NEXT T NO. OF WORDS)  EFTP052
C  EFTP053
  6. HHR (NEXT T NO. OF WORDS)  EFTP054
C  EFTP055
  7. FHR (NEXT T NO. OF WORDS)  EFTP056
C  EFTP057
  147 N = 3,0#WORD(4) + 0,1  EFTP058
  NX = N + N/21  EFTP059
  IF (MOD(N,211.NE.0) NX-NX+1  EFTP060
  DO 190 I=1,NX  EFTP061
  IF (TAPE(I).EQ.AME) GO TO 999  EFTP062
  IX = IX + 1  EFTP063
  TAPE(I) = TAPE(I)  EFTP064
  999 CONTINUE  EFTP065
  EFTP066
  EFTP067
  EFTP068
  EFTP069
  EFTP070
  EFTP071
  CALL BCREAO(TAPE(I),TAPE(NX))  EFTP072
  CALL BCREAO(TAPE(I),TAPE(NX))  EFTP073
  EFTP074
  EFTP075
  EFTP076
  EFTP077
  EFTP078
  EFTP079
  EFTP080
  EFTP081
IF (N.EQ.1) GO TO 1100
WRITE (6,1105) NAME
RETURN
C
C FORMAT FOR ERROR MESSAGE.
1105 FORMAT(1X,'ERROR IN DATA. (1OHO')
C
C WRITE ERROR MESSAGE.
1105 FORMAT(1OHOERROR IN DATA. C110)
C
C WRITE ERROR MESSAGE.
C120
DATA ON TAPE.
C
C WRITE ERROR MESSAGE.
C120
DATA ON TAPE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
C
C WRITE ERROR MESSAGE.
SUBROUTINE IDENT
C FROM FORMULA, DETERMINE--
C 1) PHASE OF SPECIES.
C 2) NUMBER OF ATOMS IN SPECIES,
C 3) MOLECULAR WEIGHT,
C 4) IF ION, NUMBER OF ELECTRONS ADDED OR SUBTRACTED FROM NEUTRAL SPECIES.
C
COMMON NAME(2), SYMBOL(70), ATMT(70), R, HCK, NEL, I, CARD, IWORD(i),
1 WORD(4), TEST(2), WEIGHT, FORMLA(S), MLA(S), NLPLUS, ELEMT(70),
2 NATON, NT, CPR(20), HRT(20), ASINDTH(20), ASINDT, FHRT, THERM,
3 SCONST, NDATMS, NLPLACE(70), LPLACE(70), NNLMA(70), NLMDIF,
4 SPECCH, TPE(202,3), PTE, THERM, XTRANGE, ICONST, NKND,
5 NF, LINES, INTR, NME, AG(70), G10(70), NIT, PI, H29BHR, HHEAT, IFF(5)
1 DATA IBLNK/IH, IPLUS/6H00000/, MINUS/6H00000/, LFTPAR/6H00000/
1 IGAS/6H00000G/, LIQ/6H00000L/

C INTEGER SYMBOL, ELEMNT, FORMLA
LOGICAL TEST
DIMENSION I(12), ND(11), NUM(I)
00 49 I = 1, 11
NO(I) = 0
NUM(I) = 0
C Puts each alphanumeric character of formula in IA array (rt adjusted)
C
NAME(I) = I CARD
NAME(2) = I WORD(I)
J = 1
50 CONTINUE
NAME(I) = I CARD
NAME(2) = I WORD(I)
C WHICH CHARACTERS ARE NUMBERS AND WHAT ARE THEY
C
IND = NUMBER OF NUMBERS
NUM(I) = LOCATION OF NUMBERS IN IA ARRAY
NO(I) = NUMBERS IN THESE LOCATIONS
C
WEIGHT = 0.0
IND = 0
IONNUM = 0
DO 50 N = 1, 12
IF (IA(N).EQ.91) GO TO 53
IF (IA(N).EQ.IBLNK) GO TO 54
53 IF (IND.NE.0.AND.N.GT.NUMINDI+3) GO TO 55
IND = IND + 1
NOINDI = I(N)
NUMINDI = N
50 CONTINUE
C
IF NO NUMBERS (IND=0) PROBABLY NOT A FORMULA CARD. RETURN TO PCL.
C
54 IF (IND.NE.0) GO TO 57
55 WRITE (6, 56)
56 FORMAT (45HERROR IN ABOVE CARD, IGNORE CONTENTS, C150)
RETURN
57 IF (IONNUM.EQ.0) GO TO 61
C IONIC SPECIES, CALCULATE CORRECTION TO MOLECULAR WEIGHT.
C
TEST(3) = .TRUE.
FIONNUM = IONNUM
WEIGHT = FIONNUM * ATMT(NEL)
IF (NEL.NE.0) GO TO 66
WEIGHT = 0
C
WRITE (6,700) 70
70 FORMAT (30HNOELECTRON DATA MISSING, C150)
GO TO 66
C
C
61 NEXT = NUM(INO) + 1

C DETERMINE PHASE OF SPECIES.
   IF (IA(NEXT).EQ.IBLNK .OR. IA(NEXT+1).EQ.IGAS) GO TO 66
   IF (IA(NEXT).EQ.IFTPAR) GO TO 165

64 WRITE (6,65)

65 FORMAT (12H ERROR IN NAMES GO TO NEXT SPECIES.
       100 TEST(6) = .TRUE.
       RETURN

165 IF (IA(NEXT+1).EQ.LIQ) TEST(5) = .TRUE.
   IF (IA(NEXT+1).NE.LIQ) TEST(6) = .TRUE.
   NPLUS = NEXT + 1
   GO TO 67
   TEST(4) = .TRUE.

66  TEST(1) = .TRUE.

C
C 160

67 I = 1
J = 1
K = 0
  DO 100 LMIN = 1,5
    FORMLA(LMN) = 0
  100 MLA(LMN) = 0
    NOATMS = 0
  C
  STORE EACH ATOMIC SYMBOL IN FORMLA(J). NUMBER OF ATOMS IN MLA(J).
  69 IF (NUM(I).EQ.(K+2)) GO TO 70
    IF (NUM(I).NE.(K+3)) GO TO 64
    FORMLA(I) = IALS(I6,IA(K+1)) = IA(K+2)
    GO TO 71
  70 FORMLA(I) = IALS(I)
  71 IF (NUM(I+1).EQ.NUM(I+1)) GO TO 72
    MLA(I) = NOAT(I)
    GO TO 73
  72 MLA(I) = 10*NOAT(I) + NOAT(I+1)

C NOATMS = TOTAL NUMBER OF ATOMS IN MOLECULE.
  73 NOATMS = NOATMS + MLA(I)

  C FIND ATOM FORMULA IN SYMBOL TABLE
  DO 91 L = 1, NATOM
    IF (FORMLA(J).EQ.SYMBOL(L)) GO TO 91
  91 CONTINUE

  92 FORMAT (50H ATOM CARD MISSING OR FORMULA INCORRECT.
   WEIGHT = 0
   GO TO 85

C CALCULATE MOLECULAR WEIGHT.
C STORE POSITION OF ELEMENT DATA IN JF.
  93 JF(J) = L
  IF (ATMWT(I).EQ.0.0) GO TO 90
  75 WEIGHT = WEIGHT + ATMWT(I)*FLOAT(MLA(I))
  85 IF (INOLE. .EQ. I) GO TO 88
    K = NUM(I)
    I = I+1
    J = J+1
    GO TO 69
  88 IF (.NOT.TEST(3)) .OR. NEL.EQ.0) GO TO 900
    J = J + 1
    JF(J) = NEL
    FORMLA(J) = SYMBOLNEL
    MLA(J) = IONNUM

C NKIND = NUMBER OF ELEMENTS IN FORMULA.
  900 NKIND = J
  IF (TEST(3) .AND. NEL.EQ.0) WEIGHT = 0.
  IF (.NOT.TEST(7)) RETURN
  NAME(I) = IBLNK
  RETURN

END
SUBROUTINE TEUPER(NT,TINTVL,T,IWORO,WORO)
C
C STORES T SCHEDULE IN T ARRAY FROM DATA ON TEMP CARDS.
C NT = NUMBER OF TEMPERATURES
C TINTVL = 1 VALUE ON TEMP CARD. PRESERVED IF LAST VALUE ON CARD SO
C IT WILL BE AVAILABLE FOR USE WITH DATA ON NEXT TEMP CARD.
C
DATA IT/HIT/+1/IHI/+IBLANK/0606050605060/
DIMENSION T(202),IWORO(5),WORD(4)
103 DO 120 J=1,4
   IF(IWORD(J).EQ.IBLANK) GO TO 120
   IF(IWORD(J).EQ.1T) GO TO 121
   IF(IWORD(J).EQ.II) GO TO 122
124 WRITE(6,123)
123 FORMAT(35H0ERROR IN LABELS ON TEMP CARD, C170)
GO TO 139
122 IF (NT.GT.01) GO TO 125
GO TO 124
125 TINTVL = WORD(J)
GO TO 120
121 IF (NT.EQ.01) GO TO 126
IF (TINTVL.EQ.0.0) GO TO 127
131 IF (T(N)+TINTVL).GT.(298.15+.0001) GO TO 128
130 IF (T(N)+TINTVL).GT.(298.15+.00011) GO TO 141
GO TO 131
141 TINTVL = 0.0
GO TO 120
129 NT = NT+1
128 NT = NT+1
T(N) = T(N-2)+TINTVL
GO TO 130
126 NT = 1
T (N) = WORD(J)
GO TO 120
127 IF (T(N).GE.(298.15-.00011)) GO TO 132
132 NT = NT+1
IF (NT.GT.202) GO TO 133
133 NT = NT+1
T(N) = 298.15
GO TO 132
120 CONTINUE
C
RETURN
1140 NT = 202
WRITE(6,140)
140 FORMAT(35H0NUMBER OF TEMPERATURES EXCEEDS 202, C180)
RETURN
C
TEMP CARD IS BLANK--USE STANDARD TEMPERATURE RANGE
C
139 IWORD(1) = IT
WORD(1) = 100.0
IWORD(2) = I
WORD(2) = 100.0
IWORD(3) = IT
WORD(3) = 6000.0
GO TO 103
END
SUBROUTINE RECO
C  READAIN AND COEF METHODS.
C
C COMMON NAME(20), SYMBOLOI, ATNUT(1), R, HCK, ELECTR, ICARD, IWORD(5),
C  HMDN(1), TEST(20), WEIGHT(20), NLAI(3), NLPL(8), NMLAI(10), NMLLET(11),
C  SPECH(202), PTMELT, PEX(10), TRANGE(10), TCART(10), TCART(20), TCART(100),
C  TNLH, LIN5, IT, NTRP, AG(10), JO(1), NIT, PI, H298HR
C
C  COMMON /PCH/LEVEL, NF, NP, CI3, TC110, TNE, NPLT, DATE, NNN, NLAST
C  C190
C
C  DIMENSION EX(15)
C  LOGICAL TEST, TSTREO
C  EQUIVALENCE (ANAP/NAP, DATA/IT, XHT/I. BLNK, I., IE /6H00000/)
C  DATA (IT, /XHT, I.), IE /6H00000/
C  COMMON /PCH/LEVEL, NF, NP, CI3, TC110, TNE, NPLT, DATE, NNN, NLAST
C
C INITIALIZE. NIT IS INDEX FOR NEXT T AND CORRESPONDING FUNCTIONS.
C NIT = NIT
C TT = T(NIT-1)
C HRT(NIT) = 1.0
C TJ = T(NIT)
C TJ2 = T(NIT)
C TSTREO = .FALSE.
C H298HO = 0.
C TEST = .TRUE.
C WRITE(16,2154)
C IF(H298HR.EQ.0.1 H298HR = H298HO/R
C IF(HRINT(NIT).EQ.0.1 OR.HRINT(NIT).EQ.HRINT(11) GO TO 9
C NLAST = NIT-1
C C
C IF THERE HAS BEEN A HEAT OF TRANSITION, CALL DELTA TO CHECK FOR
C MTSOS OR PUNCHED COEFFICIENTS FOR PREVIOUS PHASE.
C CALL DELTA
C NNN = NIT
C NLAST = NIT
C C
C IF (LEVEL(1) = LEVEL(9)
C  DO 10 J = 1, 15
C 10 (C(1:J) = 0.0
C  NEK = 0
C  NTS = 0
C  NDT = 0
C  JT = 0
C  NFIRST = 0
C  50 CALL INPUT (LINES)
C  NWOR = 1
C  IF (NFIRST .NE. 0) GO TO 100
C C
C INITIALIZE FOR FIRST CARD.
C NSUB = ICARD
C
C IF CC 1-6 NOT = TO CC 1-6 PREVIOUS CARD, GO TO 210 (C240).
100 IF (ICARD .NE. NE .AND. NSUB) GO TO 210
102 IF (TSCO .AND. NFIRST. NE. 0) GO TO 3000

NFIRST = NFIRST + 1
IOUT = 0
DO 410 I = 1, 4
   IF (WORD(I).EQ. IT) GO TO 15
410 CONTINUE

IOUT = IT
11 WRITE (6,12) IOUT
12 FORMAT (31H00ATA CARO YLS SKIPPED BECAUSE A6.24HVALUE MAS

IF((( YORD(I.EQ.IFHO).OR((YORD(I .EQ. IFH2).OR.(YORD(I = IFH2T) .OR. YORD(I .EQ. IFHRT)) .OR. (YORD(I .EQ. IFHRT) .AND. H298HO .GT. 0.0) TEST13) .OR. H298HO .EQ. 0.0) .AND. (YORD(I .EQ. IHOO) .AND. (YORD(I .EQ. IH298HO) .EQ. H298HO = -WORD(I)) .GO TO 50)

C H-H298 FUNCTIONS.
52 IF (YORD(I).EQ. IH298) HHRT = WORD(I)RT
   IF (YORD(I) .EQ. IHRT) HHRT = WORD(I)RT
   IF (H298HO .EQ. 0.0) TEST13 = .TRUE.
   HHRTNTT = HHRT + H298HO/RT
   GO TO 65
C H-HO FUNCTIONS.
60 IF (YORD(I) .EQ. IHOO) HHATNTT = WORD(I)RT
   IF (YORD(I) .EQ. IHOT) HHATNTT = WORD(I)RT
   IF (YORD(I) .EQ. IHORT) HHATNTT = WORD(I)RT
C CHECK FOR = ASINDT ON FORMULA CARO.
65 IF (ABS(ASINDT).GT. 0.005) GO TO 70
   SPEC = HHATNTT = RT
   TEST19 = .TRUE.
C C CHECK FOR FREE ENERGY FUNCTIONS.
70 FHRTT = -1.0
   SR = -1.
   DO 480 I = 1, 4
      IF(YORD(I).EQ.IFHO) YORD(I) = IFHO
      IF(YORD(I) .EQ. IFHOT) YORD(I) = IFHOT
      IF(YORD(I) .EQ. IFH2T) YORD(I) = IFH2T
      IF(YORD(I) .EQ. IFH2RT) YORD(I) = IFH2RT
      IF(YORD(I) .EQ. IFHRT) YORD(I) = IFHRT
C CHECK FOR ENTROPY FUNCTIONS.
C IF (YORD(I).EQ.IS) SR = WORD(I)RT
C IF (YORD(I).EQ.ISR) SR = WORD(I)RT
   GO TO 480
   107

34
IF (FHRINT(1).EQ.0.00) GO TO 1100
FHRINT = FHRIT
IF (ISOUTNE.IDH) GO TO 1100
FHRINT = FHRINT - HHRTINTT
IF (ABS(HHRT.INTT)) .GT. 0.005) GO TO 9491
SPECH = HHRTINTT * RT
C TEST19--THERE IS AN ENTHALPY FOR THE ASINOT ON THE FORMULA CARD
C STORED IN SPECH-
TEST(19) = .TRUE.
C220 PROCESS COEFFICIENTS
C C220 TO C240--STORE CONTENTS OF DATA CARD. IF FHRINT(1) IS NOT = -1 INTT=0,) CALCULATE INTEGRATION CONSTANTS FROM THE ENTHALPY AND FREE ENERGY WHICH HAVE JUST BEEN READ
3006 IF (FHRTINTT).EQ.0.0) NTT = 0
IF (IWORD(10)).EQ.I) GO TO 200
IF (IWORD(10)).EQ.I) GO TO 11
IF (IWORD(10)).EQ.I) GO TO 160
IF (IWORD(10)).EQ.I) GO TO 165
IF (IWORD(10)).EQ.I) GO TO 140
IF (IWORD(10)).EQ.I) GO TO 145
C ANALYZE CI ICOEFFICIENTS) AND EI (EXPONENTS) LABELS. USE NUMBFR AS INDEX TO STORE VALUES IN C AND EX ARRAYS.
INDEX TO STORE VALUES IN C AND EX ARRAYS.
IWD = IALS(I, IWORD(I))
AN = ANDMASK(IWD)
IF (NN .EQ. 0) GO TO 107
NN = IARS (NN, 1W)
GO TO 108
107 NN = IARS (10, IWD)
108 IF (NN .GT. 15) GO TO 1018
LABEL = IARS(15, IWORD(I))
IF (LABEL .EQ. 1) GO TO 120
IF (LABEL .EQ. 1) GO TO 130
1018 WRITE (6,1019) IWORD(I), IWORD(I)
C1019 FORMAT (IHO,A6, 39H IS AN INCORRECT LABEL FOR THE NUMBER-- .E16.8)
1 3HL. THUS THE VALUE WAS IGNORED.
LINES = LINES + 3
CALL INKH (.GE. 95) CALL PAGEID (LINES)
GO TO 200
C TEST17=--PUNCH COEFFICIENTS
710 TEST17 = .TRUE.
NOTS = NOTS + 1
IF (WORD(I). LE. TCINT(I)) GO TO 200
NTC = NTC + 1
TCINT(I) = WORD(I)
GO TO 200
C TJ1 TO T.J2 = TEMPERATURE RANGE FOR WHICH COEFFICIENTS ARE GOOD.
110 IF ( NOT .EQ. 1) GO TO 114
IF (JTNE.1) TJ1 = WORD(I)
IF (JT. EQ. 1) TJ2 = WORD(I)
JT = 1
GO TO 200
114 IWORD = TD
GO TO 210
120 EXTNN = IWORD(I)
NEX = NEX + 1
GO TO 200
C DIVIDE COEFFICIENTS BY R IF NO REDUCE LABEL ON METHOD CARDITSTRE[FF).
130 IF (.NOT.TSTRE[FF]) WORD(I)/R
I I I I I I
C
C TEST(18)--ABSOLUTE VALUES FOR ENTHALPY.
C(LEVEL,NN) = WORD(ID)
GO TO 200
140 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R
C LEVEL INDEX FOR TEMPERATURE INTERVALS.
145 TEST(18) = .TRUE.
C IF (ASINDT.NE.0) WRITE (6,146)
146 FORMAT (45H0ENTHALPY IS ABSOLUTE--ASINOT SHOULD = 0.
LINES = LINES + 2
155 HCOEF = WORD(ID)
GO TO 200
150 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R
GO TO 155
160 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R
165 SCOEF = WORD(ID)
200 CONTINUE
GO TO 50
C
C240
C210 IF (.NOT.TSTCO) GO TO 601
NF1 = 6
NF2 = 7
220 NT1 = NT
IF (IASINDT,EQ.0) OR TEST(19) GO TO 240
IF (IASINDT,EQ.1) AND ASINDT.LE.TJ2) GO TO 230
IF (ABS(IASINDT-298.15),GT.0.0) OR.H298R,EQ.0) GO TO 240
SPECH = H298HR*R
TEST(19) = .TRUE.
GO TO 240
GO TO 155
230 NT1 = NT + 1
T(NT1) = ASINDT
240 I = T
241 IF (T(J).LT.TJ1 OR T(J).GT.TJ2) GO TO 400
245 CMPI(J) = 0.0
HHRT(J) = 0.0
SR = 0.0
C
C CALCULATE FUNCTIONS FROM EQUATION
DO 300 J = 1, NEX
248 TEX = 1.0
C
C NTT = T IF AN ENTHALPY AND ENTRPY HAS BEEN READ FOR THE PURPOSE OF
C CALCULATING THE INTEGRATION CONSTANTS. IF THESE VALUES HAVE NOT
C BEEN READ, NTT = 0.
IF (NTT.EQ.0) TT = T(J)
IF (EX(J).NE.0.0) TEX = TT **EX(J)
IF(LVLE(J).LE.TJ1) HHRT(J) = HHRT(J) + C(LEVEL,J)*TT *ALOG(TT)
IF(LVLE(J).GT.TJ1) HHRT(J) = HHRT(J) + C(LEVEL,J)/TT **EX(J)
IF (I.IE.NT) NIT = I+1
GO TO 241
TEST(19) = .TRUE.
300 CONTINUE
301 IF (.NOT.EQ.0) GO TO 350
HCOEF = (HHRT(NTI) - HHRTT) *TT
SCOEF = HHRT(NTT) - CILEVEL(J)/EX(J) * TEX
SR = C(LEVEL,J) * EX(J)
IF (NTT.EQ.0) GO TO 350
IF (I.EQ.0) NCAR = NSUB
350 CONTINUE
351 IF (.NOT.EQ.0) GO TO 350
HCOEF = (HHRT(NTI) - HHRTT) *TT
SCOEF = HHRT(NTT) - C(LEVEL,J)/EX(J) * TEX
IF (I.EQ.1) SR = SR + C(LEVEL,J)*TEX
IF (I.EQ.1) SR = SR + C(LEVEL,J)*TEX
IF (I.EQ.0) SR = SR + C(LEVEL,J)*TEX
355 CONTINUE
356 CONTINUE
357 CONTINUE
358 CONTINUE
359 CONTINUE
360 CONTINUE
361 CONTINUE
362 CONTINUE
363 CONTINUE
364 CONTINUE
365 CONTINUE
366 CONTINUE
367 CONTINUE
368 CONTINUE
369 CONTINUE
370 CONTINUE
371 CONTINUE
372 CONTINUE
373 CONTINUE
374 CONTINUE
375 CONTINUE
376 CONTINUE
377 CONTINUE
378 CONTINUE
379 CONTINUE
380 CONTINUE
381 CONTINUE
382 CONTINUE
383 CONTINUE
384 CONTINUE
385 CONTINUE
386 CONTINUE
387 CONTINUE
388 CONTINUE
389 CONTINUE
390 CONTINUE
391 CONTINUE
392 CONTINUE
393 CONTINUE
394 CONTINUE
395 CONTINUE
396 CONTINUE
397 CONTINUE
398 CONTINUE
399 CONTINUE
400 CONTINUE
401 RETURN
END
SUBROUTINE ATOM

COMMON NAME(21), SYMBOL(70), ATMT(70), R, HCK, ELECTR, CAR0, IWORD(51),
1 WORD(41), TEST(20), WEIGHT, FORMA(51), MLA(51), BLANK, ELEMT(70),
2 NATOM, CPE(2021), HRT(2021), ASINOH(120), ASINT(35120),
3 SCONST, NPL(70), NPLACE(70), LPLACE(70), NLNAME(70), NDFILE,
4 SPEC, TAPE, RPEX, PEX(110), TRANGE(110), TCONST, NKIND,
5 NF, LINES, T, NTMP, AGE(70), G(70), NIT, PI, H298HR, HEAT, J(51),
6 ATOM(4000)

DIMENSION AJ(400), ANU(400), GI(400), NN(400), TEMPR(41), TEMPH(41),
7 DATA NTMPR/6TEMPR/, NFIX/6HFIX/, NDFILL/6HDFILL/, NDF/6HDF/,
8 LOGICAL TEST, TSTFIL, GLABEL

LINES = LINES + 2

C INITIALIZ TO NO CUT-OFF AND NO FILL
KUTOFF = NON
TSTFIL = .FALSE.
GLABEL = .FALSE.

C CHECK FOR FILL AND CUTOFF(KUTOFF) ON METHOD CARD.
DO 7 I=1,4
IF (WORD(I) .EQ. IFILL) TSTFIL = .TRUE.
IF (WORD(I) .EQ. LG) GLABEL = .TRUE.
IF (WORD(I) .EQ. NTEMPR) KUTOFF = NTMPR
IF (WORD(I) .EQ. NFIX) GO TO 7
NFK = WORD(I)
KUTOFF = NF
TSTFIL = .FALSE.

7 CONTINUE
K = 0
ALWAT = ALGO*WEIGHT*1.5
NFK = 0

CALL INPUT TO READ AND LIST A DATA CARD.
10 CALL INPUT(LINES)
IF (NFK .NE. 0) GO TO 12
NSUB = ICARD
NFK = 1
12 IF (ICARD .NE. NSUB) GO TO 50
DO 40 I=1,4
IF (WORD(I) .EQ. IFILL) GO TO 13
40 CONTINUE

13 IF (WORD(I) .NE. IBLANK) GO TO 40
DO 20 40 I=1,4
IF (WORD(I) .NE. IP) GO TO 13
20 CONTINUE

K = K+1
NNK = IWORD(5)
IF (NNK .EQ. 0) GO TO 30
NFK = 0

C J VALUES ARE READ WITH ALPHANUMERIC FORMAT. CHANGE TO NUMBER AND
STORE IN AJ ARRAY.
HALF = 0.
ISFT = 0.
DO 14 MLK = 1,6
14 CONTINUE
16 AJK = ISFT
IF (LOOK .EQ. 5) AJK = AJK + HALF
IF (GLABEL = 1) GO TO 22
G(K) = 2.*AJK + 1.
ANUK = WORD(I)
IF (WORD(I) .EQ. 0) GO TO 16
ANUK = ANUK + 1.
ANUK = ANUK + 1.
G(K) = 0.
GO TO 40
30 CONTINUE
40 CONTINUE
GO TO 10
50 KLAST = K

C SORT ENERGY LEVELS IN INCREASING NUMERICAL ORDER.
75 J=1
76 M=J
77 DO 79 M=J+1,KLAST
78 M=J
79 CONTINUE
IF (J .LE. 80.81.80
80 TENV = ANU(M)
ANU(M) = ANU(J)
ANU(J) = TENV
G(M) = G(J)
G(J) = G(M)

END
GIJ = TEMPY
KTEMPY=NN(NI)
NN(M) = NN(J)
NN(J) = KTEMPY
TEMPY = AJ(M)
AJ(M) = AJ(J)
AJ(J) = TEMPY
81 J = J + 1
82 CONTINUE
NNKLAST+1 = 0
AJKLAST+1 = 0
GIKLAST+1 = 0
ANUKLAST+1 = 0
IF (.NOT.TSTFIL) GO TO 1087
WRITE (6,1082)
1082 FORMAT (1HD,4X,1HN,4X,1HJ,7X,1HG,5X,13HENERGY LEVEL ,12X,1HN,6X,1ATOM086)
1HJ,7X,1HG,5X,13HENERGY LEVEL )
LINES = LINES + 2
DO 1089 I = 1, KLAST + 1
NN = I + 1
WRITE (6,1083) (NN(I),AJ(J),GI(J),ANU(I), IN = I,INN)
1083 FORMAT (2116,F8.1,F8.1,F14.3,10X))
LINES = LINES + 1
1085 IF (LINES.GE.55) CALL PAGEID(LINES)
1087 IF (KUTOFF.NE.NFIX) GO TO 100
IF (NFIXED.NE.NFIX) CALL PAGEID(LINES)
-99 FORMULASTATISTOMIC FIXEDN IS LESS THAN FIRST N. FIXEDN IS SET EQUAL
TO 11
IF (J+15.0) (280)
WRITE (6,99) (NN(I))
1100 IF (.NOT.TSTFIL) GO TO 160

C
C290 C290
C ROUTINE FOR ASSIGNING TO LAST LEVEL OF EACH PRINCIPLE QUANTUM NUMBER,ATOM0119
C PON. THAT WEIGHT WHICH GIVES PQN THE TOTAL SUM OF 2J+1, OBTAINED FROMATOM0120
C THE FORMULA ANNN. (IGNORES PQMS LOWER THAN GROUND STATE. AND. WHEN
C NECESSARY, USES SPECIAL NUMBER FOR SUM OF 2J+1 FOR PQN OF GROUND
C STATE.
C
C
C1INDEX = JJ
C
C101 INDEX = JJ
C102 FORMAT (7X,1IB,9X,1HN,3X,15HSTATION, SUM(J+1),3X,14HACT. SUM(J+1),1ATOM0130)
C103,4HDIFF,5X,9HMAX LEVEL,3X,16HJ+1, MAX LEVEL )
LINES = LINES + 1
IF (LINES.GE.55) CALL PAGEID(LINES)
K = 1
NNJ = NN(J)
SUM = 0.0
L = 1
DO 150 J = K,KLAST
IF (NN(J).LT.0) GO TO 150
IF (NN(J) - KUREN) (110,105,110)
105 SUM = SUM + G(J)
M = J
NN(J) = -NN(J)
IF (J.NE.KLAST) GO TO 150
GO TO 115
110 IF (L .NE. 1) GO TO 114
L = 0
K = J
114 IF (J .NE. KLAST) GO TO 150
115 IF (KUREN.EQ.NN 1) GO TO 120
TEMPY = KUREN*KUREN
FORM = AG(INDEX)*TEMPY
GO TO 125
120 FORM = GGI(INDEX)
125 DIFF = FORM - SUM
IF (KUREN .LT. INNI) DIFF = 0.0
NNM = -NN(M)
IF (DIFF.GT.0.0) G(M) = G(M)*DIFF
WRITE (6,1321A1) (NN(I),NN(J, G(M) ,SUM.DIFF, ANU(I), G(M)
LINES = LINES + 1
IF (LINES.GE.55) CALL PAGEID(LINES)
IF (L .NE. 1) GO TO 102
GO TO 160
150 CONTINUE
C
C300 C300
C
C160 IF (ASINDT.NE.0.0) GO TO 162
NTI = NT
GO TO 200
162 NTI = NT+1
Y(NTI) = ASINDT
C
C M = INDEX FOR T. K = INDEX FOR ELECTRONIC LEVELS.
C
C
C200 DO 300 M=NTI,NTI
305 I = 0

38
I.

C CALCULATE THE PARTITION FUNCTION AND DERIVATIVES FOR EACH ELECTRONIC
C LEVEL AND TEMPERATURE.

DO 206 K=1,KLAST

210 X(1NCX(1M)  IF(ANUK)  214,212,212

212 IF (.NOT.TEST(I14)) GO TO 260

WRITE (6,213)  261

213 FORMAT(5DHJNOT ALL LEVELS WERE USED. X IS GREATER THEN 85.  
LINES = LINES + 1  
IF (lines .GE.55) CALL PAGEID (lines)  
GO TO 260

214 IF(TUTOFF.NE.NFIX) GO TO 219

WRITE(6,1215)  272

1215 FORMAT(33MD IONIZATION POTENTIAL IS MISSING)

RETURN

215 THERM= PI-TEM/HCK

216 IF(ANUK).LT.THERM)  240,240,217

217 IF (.NOT.TEST(I14)) GO TO 260

WRITE(6,218) 

218 FORMAT(58HJALL LEVELS HAVE BEEN USED TO THE THERMAL BINDING ENERGY)

F12,3)  AT0110228

LINES = LINES + 1

IF (LINES .GE.55) CALL PAGEID (LINES)  
GO TO 260

219 IF(TUTOFF.NE.NFIX) GO TO 240

220 IF (ABSIN(NK).LT.E-NFIXED) GO TO 240

221 JJ = 0

GO TO 206

240 I = I + 1

G(I+1)=G(I)*EXP(-1.*X)

TODDT(I+1)=DI(I)*X

245 IF(I(I)-0.1E-9) 246.206

246 IF (I=0.1E-9) 246.247

247 IF(XTDODT(I)=0.1E-9) 246.206

248 IF (.NOT.TEST(I14)) GO TO 260

WRITE(6,249)  

249 FORMAT (12HJNOT ALL ASSIGNED LEVELS WERE USED. Q AND DERIVATIVES)

IRE TOO SMALL  

IF (I=0)  

LINES = LINES + 1

IF (LINES .GE.55) CALL PAGEID (LINES)  
GO TO 260

206 CONTINUE

1260 IF (.NOT.TEST(I14)) GO TO 260

IF (.EQ.0) WRITE (6,222) NFIXED

222 FORMAT(28HJALL LEVELS USED THROUGH N = 15) 

IF (.EQ.0) WRITE (6,12621)

1262 FORMAT(48HJALL ASSIGNED LEVELS HAVE BEEN USED)

LINES = LINES + 1

IF (LINES .GE.55) CALL PAGEID (LINES)

C

C310

C CALCULATE TOTAL Q, DERIVATIVES, AND THERMODYNAMIC FUNCTIONS FOR T.

C

260 QSUM=0.0

261 QSUM=QSUM+Q(I)

262 IF (QSUM+0.1E-9) 263,264,266

263 TMP = 0.0

GO TO 265

264 TMP=QSUM/QSUM

265 HRR(M)=TMP+2.5

266 HRR(M)=HTM+2.5*XTODDS/QSUM-TMP+2.5

270 FORMAT (1X,1HT,F12,6X,4HCPR/R,F12,6X,BHF-BO/RT,F11,6,6X,BHF-BH/BO)

271 FORMAT (1X,1HT,F12,6X,4HCQSUM,F12,7,6X,9HTODT/F1S,F13,7,6X,10HTQAT)

300 CONTINUE

301 SPECH=HRR(M1)*1.03

302 TEST(19)=ENTHALPY HAS BEEN CALCULATED FOR T ON FORMULA CARD.

C TEST(9)=FUNCTIONS HAVE BEEN CALCULATED.

IF (TEST(19) .LT. TEST(9)) 301,300,302

RETURN

END
SUBROUTINE POLY
C IF TEST IS TRUE--
C TEST(1) MOLECULE IS NON-LINEAR
C TEST(2) RIGID ROTATOR-HARMONIC OSCILLATOR APPROXIMATION
C TEST(3) SECOND ORDER CORRECTIONS ARE CALLED FOR
C TEST(4) PENNINGTON AND KOBE APPROXIMATION
C TEST(5) JANAF METHOD FOR DIATOMIC MOLECULES
C TEST(6) SPECIES HAS EXCITED ELECTRONIC STATES

C COMMON NAME(2), SYMBOL(T0), ATOM(T0), E.MC, ELECTR, ICARD, WORD(5),
1 WORD(1), TEST, WEIGHT, FORM(5), MLA(5), ANY, ELEMENT(70),
2 NATOM, NT, CPR(202), HRNT(202), ASINDH(202), ASINDH, FHRNT(202),
3 SGCON, SGATS, NPLACE(T0), LPLACE(T0), NML(T0), NDFILE,
4 SPEC, TABLE(202), PFTWET, PEX(10), TRANGE(10), TCONST, NKIND,
5 NF, LINES, TR, NTMP, AG(T0), GG(T0), NRH, PTH, LINE, J5(5)
COMMON /WDOWN/ V(20), DNT(20), NO(20), X(6, 6), Y(6, 6), 'NNU, ALFA(6),
1 ALFA(6), ALFC(6), G(6), BETA(6), A, B, C, RH, D, WP, W,
2 SYM, STW, 100, THETA(5), TEST(6), R(10), 0, 100, 0, L130, 0, 0,
3 DDQ, LABEL, QTOT, OMTOT, OMTOT, OMTOT, DQMTOT, CORT, AIJ(6, 6), AIIII, AI(6), NSUB,

C POLY001
C DIMENSION IE(5), RI(6)
C LOGICAL TEST, TEST
C EQUIVALENCE (IWO, MD)
C DATA IRRHOD4/HRHOD4/, JANAF/5HJANAF/, NRRHOD2/6HRNRA02/
C DATA IPK/5HPANDK//, BCONF/2.7988898/, BLANK/1H/, NRRAO1/6HRNAO1/

C C320
C
C C320 INITIALIZE FOR EACH SET OF METHOD AND DATA CARDS.
1 DO 10 I = NIT, NT
2 CPR(I) = 0.0
3 HRNT(I) = 0.0
10 FHRNT(I) = 0.0
20 HRNT(I) = 0.0
30 FHRNT(I) = 0.0
40 SYM = 1.0
50 DO 1005 I = 2, 6
60 IF (I .EQ. 1) THEN
70 NI = 1
80 DO 900 I = 1, 4
90 IF (IWO .EQ. IRRH0) GO TO 12
100 IF (IWO .EQ. NRRH0) GO TO 13
110 IF (IWO .EQ. IPK) GO TO 14
120 IF (IWO .EQ. JANAF) GO TO 15
130 IF (IWO .EQ. NRRAO) GO TO 21
140 CONTINUE
150 WRITE (6, 19)
160 FORMAT('METHOD CODE WAS NOT RECOGNIZED, USED NRRAO1, C320 )
170 19 FORMAT(S10, 2X, TESTW21) = .TRUE.
180 GO TO 21
190 13 TESTW31 = .TRUE.
200 GO TO 21
210 14 TESTW41 = .TRUE.
220 GO TO 21
230 15 TESTW51 = .TRUE.
240 IF (NATOM .EQ. 2) TESTW21 = .TRUE.
250 21 WRITE (6, 21) WEIGHT
260 FORMAT('1ISOMOLECULAR WT.=F10.5')
270 NFIRST = 0
280 CONTINUE
290 LINES = LINES + 4

C C330
C C CALL INPUT TO READ AND PRINT CONTENTS OF INPUT CARD.
20 CALL INPUT(LINES)
21 IF (NFIRST .NE. 0) GO TO 100
22 IF (NFIRST .GE. 1) NSUB = ICARD

C C330 INITIALIZE FOR EACH ELECTRONIC LEVEL.
1001 STW = 1.0
1200 T0 = 0.0
1300 A = 0.0
1400 B = 0.0
1500 C = 0.0
1600 D = 0.0
1700 WP = 0.0
1800 W = 0.0
1900 THETA(3) = 0.0
2000 AI = 0.0
2100 DO 1002 I = 1, 6
2200 WRITE (6, 1003)

C POLY0001
C POLY0002
C POLY0003
C POLY0004
C POLY0005
C POLY0006
C POLY0007
C POLY0008
C POLY0009
C POLY0010
C POLY0011
C POLY0012
C POLY0013
C POLY0014
C POLY0015
C POLY0016
C POLY0017
C POLY0018
C POLY0019
C POLY0020
C POLY0021
C POLY0022
C POLY0023
C POLY0024
C POLY0025
C POLY0026
C POLY0027
C POLY0028
C POLY0029
C POLY0030
C POLY0031
C POLY0032
C POLY0033
C POLY0034
C POLY0035
C POLY0036
C POLY0037
C POLY0038
C POLY0039
C POLY0040
C POLY0041
C POLY0042
C POLY0043
C POLY0044
C POLY0045
C POLY0046
C POLY0047
C POLY0048
C POLY0049
C POLY0050
C POLY0051
C POLY0052
C POLY0053
C POLY0054
C POLY0055
C POLY0056
C POLY0057
C POLY0058
C POLY0059
C POLY0060
C POLY0061
C POLY0062
C POLY0063
C POLY0064
C POLY0065
C POLY0066
C POLY0067
C POLY0068
C POLY0069
C POLY0070
C POLY0071
C POLY0072
C POLY0073
C POLY0074
C POLY0075
C POLY0076
C POLY0077
C POLY0078
C POLY0079
C POLY0080
C POLY0081
C POLY0082
C POLY0083
ASSUME LINEAR MOLECULE WITH 3N-5 FREqs. IF THERE IS AN A OR IA
IN THE INPUT, CHANGE TO 3N-6—SEE C350.

TESTW11 = .FALSE.
GO TO 1015

IF CARD COLUMNS 1-6 OR 79-80 ARE DIFFERENT FROM PREVIOUS CARD, GO
TO 1051 (C380).

1010 IF (CARD .NE. NuM .OR. Level .NE. 1) WORD(1) GO TO 1051

1015 IF (WD .EQ. BLANK) GO TO 1050
IF (WD .EQ. T1 .OR. WD .EQ. T2) GO TO 100
IF (WD .EQ. STATMT) GO TO 102
IF (WD .EQ. SYMNO) GO TO 104
IF (WD .EQ. B1 .OR. WD .EQ. B2 .OR. WD .EQ. BE) GO TO 106
IF (WORD(ID) .EQ. IB) GO TO 108
ISHT1 = IARS(6,WORD(ID))
IF (ISHT1.EQ.IALFAB .OR. ISHT1.EQ.IALFA
. OR. ISHT1.EQ.IALFAC) GO TO 1030

IF (NOATHS .EQ. 2) GO TO 1045
IF (WD .EQ. RH0) GO TO 110
IF (WD .EQ. DI .OR. WD .EQ. D111) GO TO 112
IF (WD .EQ. D2 .OR. WD .EQ. D22) GO TO 112
IF (WORD(ID) .EQ. IA) GO TO 1020
IF (WORD(ID) .EQ. IC) GO TO 114
IF (WD .EQ. A1 .OR. WD .EQ. A2) GO TO 1023
IF (WD .EQ. CI .OR. WD .EQ. C2) GO TO 116
IF (WD .EQ. WI .OR. WD .EQ. W2) GO TO 118

ISHT75 = IARS (75, WORD(ID))
IF (ISHT75 .EQ. IV) GO TO 1033
IF (ISHT75 .EQ. Nx .OR. ISHT75.EQ.NAIJ) GO TO 1040
IF (ISHT75 .EQ. NY) GO TO 1044
IF (ISHT75 .EQ. IG) GO TO 1025

1019 FORMAT (1H0,46, 39H IS AN INCORRECT LABEL FOR THE NUMBER— .E16.8)

1 36H. THUS THE VALUE WAS IGNORED, C340
)

100100 = WORD(ID)
GO TO 1050

C
102 STHT = WORD(ID)
GO TO 1050
104 SYM = WORD(ID)
GO TO 1050
106 B = WORD(ID)
GO TO 1050
108 B = BCONV/WORD(ID)
GO TO 1050
110 RH = WORD(ID)
GO TO 1050
112 D = WORD(ID)
GO TO 1050
114 C = BCONV/WORD(ID)
GO TO 1050
116 C = WORD(ID)
GO TO 1050
118 MF = WORD(ID)
GO TO 1050
C
C IF IA OR A LABEL. NON-LINEAR MOLECULE.
1020 A = BCONV/WORD(ID)
1021 TESTWIII = .TRUE.
NV = 3*NDATMS - 6
GO TO 1050
GO TO 1021
1030 IBACK1 = IALS(6, ISHTFl)
I = IWORD(ID) - IBACK1
IF (1. GT. 10) I = 1
IF (ISHTFl.EQ.IALPHA .OR. ISHTFl.EQ.IALFAB) ALFB1 = WORD(ID)
IF (ISHTFl.EQ.ICALFAA) ALFA1 = WORD(ID)
IF (ISHTFl.EQ.ICALFAC) ALFC1 = WORD(ID)
GO TO 1050
C
C STORE FREQUENCY AND DEGENERACY ACCORDING TO LABEL.
1033 J = 1
DO 1034 I = 1, 5
IWD = IALS (6, IWD)
IE(J) = IARS(30, IWD)
IF (IE(J) .EQ. 48) GO TO 1034
J = J + 1
1034 CONTINUE
I = 1
KV = IE(I)
I = I + 1
IF (IE(I) .GE. 10) GO TO 1035
KV = 10 + KV + IE(I)
IF (KV .GT. NV) GO TO 1038
1035 VKV = WORD(ID)
1036 I = I + 1
IF (1. GT. J) GO TO 1050
IF (IE(I) .GE. 10) GO TO 1036
ONEKV = IE(I)
NDEKV = IE(I)
GO TO 1050
1038 WRITE (6,1019) IWORD(ID), WORD(ID)
TESTWIII = .TRUE.
RETURN
C
C STORE XIJ ACCORDING TO LABEL.
1040 IWD = IALS(6, IWORD(ID))
IX = IARS(30, IWD)
IWD = IALS(6, IWD)
JX = IARS(30, IWD)
IF (ISHT5.EQ.NX) (IX, JX) = WORD(ID)
IF (ISHT5.EQ.NAJ) (JIJI, JX) = WORD(ID)
GO TO 1050
C
C STORE YIJK ACCORDING TO LABEL.
1044 IWD = IALS(6, IWORD(ID))
IY = IARS(30, IWD)
IWD = IALS(6, IWD)
JY = IARS(30, IWD)
IWD = IALS(6, IWD)
KY = IARS(30, IWD)
Y(IY, JY, KY) = WORD(ID)
GO TO 1050
C SOME INPUT FOR DIATOMIC MOLECULES.
C
1045 IF WD .EQ. NE1 GO TO 300
IF WD .EQ. WE1 .OR. WD .EQ. W1 ) GO TO 301
IF WD .EQ. WE2 .OR. WD .EQ. W2 ) GO TO 302
IF WD .EQ. WE3 .OR. WD .EQ. W3 ) GO TO 303
IF WD .EQ. BETA1 ) GO TO 304
IF WD .EQ. BETA2 ) GO TO 305
IF WD .EQ. BETA3 ) GO TO 306
IF WD .EQ. DE1 ) GO TO 307
IF WD .EQ. DE2 ) GO TO 308
GO TO 1018
300 W = WORD(I)
GO TO 1050
301 WX11 = WORD(I)
GO TO 1050
302 WX12 = WORD(I)
GO TO 1050
303 WX13 = WORD(I)
GO TO 1050
304 WX14 = WORD(I)
GO TO 1050
305 BETA1 = WORD(I)
GO TO 1050
306 BETA2 = WORD(I)
GO TO 1050
307 BETA3 = WORD(I)
GO TO 1050
108 D = WORD(I)
GO TO 1050
1050 CONTINUE
C DATA FOR CARD HAS BEEN STORED. GO TO 28 (C3301) TO READ NEXT CARD.
GO TO 28
C DATA FOR ELECTRONIC LEVEL HAS BEEN STORED--CALCULATE SOME VARIABLES
REQUIRED IN EQUATIONS.
1051 NNU = 0
IF (CARD.EQ.NSUB) TESTW161 = .TRUE.
I = 0
IF (NDATMS .NE. 2) GO TO 1052
C DIATOMIC MOLECULES--
V(1) = W - 2.0*WX11 + 3.25*WX12 + 5.0*WX13 + 7.5625*WX14
X11(1) = WX11+4.5*WX12+14.5*WX13
Y11(1) = WX12 + 8.0*WX13
AI11(1) = ALFB1(1)-ALFB1(2)-.75*ALFB1(3)
AI11(11) = ALFB1(2)-1.5*ALFB1(3)
C CALCULATE AND CHECK NUMBER OF FREQS. (NNU). MAXIMUM 6 FOR NON-RRHO.
1052 NNU = NNU + 1
I = 1 + NINU
IF (VNNU) .EQ. 0.0) GO TO 1094
IF (I .LT. NV) GO TO 1052
IF (I .GT. NV) GO TO 1094
IF (1.EQ.0.01) GO TO 1098
IF (NNU.GT.6) TESTW10 = .TRUE.
IF (NDATMS.EQ.2) GO TO 2054
IF (TESTW12) GO TO 1092
IF (TESTW13) GO TO 1053
GO TO 1056
C DIATOMIC MOLECULES--
2054 D=(BETA1*0.5+BETA21*0.5+BETA11)*0.5
IF (D.EQ.0.01) D=1.4*BB**3/7.1**2
BE= 8
B1=(ALFB1*0.5+ALFB21*0.5+ALFB11)*0.5
IF (.NOT.TESTW14) GO TO 2054
C JANAF CORRECTIONS
AI11(1) = AI11(1)/BE1
X11(1) = X11(1)/BE1**2
RH = 4.*SORT10/BE1**2/HEX**2
GO TO 1059
C IF RRHO, SKIP TO 1090 (C410).
9054 IF (TESTW12) GO TO 1090
C DIATOMICS--NOT JANAF.
IF (TEST(14)) WRITE (6,3056) A,B,C,RH
3056 FORMAT (5HSHAO = F10.6,5X,WHDO = ,E13.6,5X,SHRHO = ,E13.6 )
   1050 CONTINUE
C
C NON-LINEAR MOLECULES---
1053 IF (ICF.EQ.0.0) GO TO 1092
C
C POLYATOMIC MOLECULES. MAKE X AND Y MATRICES SYMMETRIC.
DO 8 I = 1,NNU
   X(I,J) = X(J,I)
8 CONTINUE
C
C IFINDATMS.EQ.2) GO TO 1092
C
C POLYATOMIC MOLECULES. MAKE X AND Y MATRICES SYMMETRIC.
DO 8 I = 1,NNU
   X(I,J) = X(J,I)
8 CONTINUE
C
C I. CI1 CORRECTIONS
    00 860 I=1,NNU
    00 = GI0 + 8
    IF ~.NOT.TESTW(III = V(III)*1.5*DN(III)*3.*G(III))
    IF I.TESTW(III X(I,J) = X(I,J) + DN(I)*1.5*V(I,J) + DN(J)*1.5*V(I,J) + CY)
    860 CONTINUE
C C INTERMEDIATE OUTPUT--XIJS AND LEVEL.
    1092 IF I.NOT.TEST(141) GO TO 1091
    WRITE (6,2860) I,J
    2860 FORMAT (8HOX(I,J)
    DO 2861 I = 1,NNU
    2861 WRITE (6,2862) X(I,J),J=1,NNU)
    2862 FORMAT (1H6F10.4)
    WRITE (6,1093) LEVEL
    1093 FORMAT (BHILEVEL = *12)
    1091 IF (TESTW(5)) TESTW(4) = .TRUE.
C C CALL LINK1 TO CALCULATE PARTITION FUNCTION AND DERIVATIVES FOR LEVEL.
    1090 CALL LINK1
C C INTERMEDIATE OUTPUT--XIJS AND LEVEL.
    1092 IF I.NOT.TEST(141) GO TO 1091
    WRITE (6,2860) I,J
    2860 FORMAT (8HOX(I,J)
    DO 2861 I = 1,NNU
    2861 WRITE (6,2862) X(I,J),J=1,NNU)
    2862 FORMAT (1H6F10.4)
    WRITE (6,1093) LEVEL
    1093 FORMAT (BHILEVEL = *12)
    1091 IF (TESTW(5)) TESTW(4) = .TRUE.
C C CALL LINK1 TO CALCULATE PARTITION FUNCTION AND DERIVATIVES FOR LEVEL.
    1090 CALL LINK1
C C LEVEL AND GO TO 1001 (C330).
C C OTHERWISE CALCULATE FUNCTIONS FROM Q AND DERIVATIVES (VALUES FOR
C C MULTIPLE ELECTRONIC STATES HAVE BEEN SUMMED).
    IF (ICARD .EQ. NSUB) GO TO 1001
    NT1 = NT
    IF (ASINOT.NE.0.) NT1=NT+1
    DO 1000 I = NT,NT1
    IF (.NOT.TESTW(I)) GO TO 999
    Q = FHRT(I)
    FHRT(I) = ALOG(Q)
    DO = HRHT(I)/Q
    HRHT(I) = DO
    CPR(I) = CPR(I)+2.5
    999 FHRT(I) = FHRT(I) + 1.5*ALOG(WEIGHT) + 2.5*ALOG(T(I)) + SCONST
    HRHT(I) = HRHT(I) + 2.5
    1000 CPR(I) = CPR(I) + 2.5
    IF (ASINOT.EQ.0.) GO TO 4001
C C CALCULATE ENTHALPY FOR ASSIGNED T ON FORMULA CARD.
    SPECH = HRHT(NT1)*E*ASINOT
    TEST(19) = .TRUE.
    4001 TEST(9) = .TRUE.
    RETURN
1094 WRITE(6,1095)
1095 FORMAT (5700 WRONG NUMBER OF NU-SIV-S, C410 )
    GO TO 2000
1098 WRITE(6,1099)
1099 FORMAT (5700 THE VALUE OF B IS MISSING, C410 )
    GO TO 2000
1100 WRITE(6,1101)
1101 FORMAT (5700 THE VALUE OF C IS MISSING, C410 )
2000 TEST(16) = .TRUE.
    RETURN
END
SUBROUTINE LINK1
C
C CALCULATE Q
C
C TEST(1) MOLECULE IS NON-LINEAR
C
C TEST(2) RIGID ROTATOR-HARMONIC OSCILLATOR APPROXIMATION
C
C TEST(3) SECOND ORDER CORRECTIONS ARE CALLED FOR
C
C TEST(4) PENNINGTON AND KOB APPROXIMATION
C
C TEST(5) JANAF METHOD FOR DIATOMIC MOLECULES
C
C TEST(6) SPECIES HAS EXCITED ELECTRONIC STATES
C
C COMMON NAME(2),SYMBOL(7),ATNM(7),E,HCK,ELECTR,ICARD,WORD(5),
C 1 WORD(4),TESTO1,WEIGHT,FORMULA(5),MLA(5),ANY,ELEMENT(70),
C 2 NATOM,CP(2),HHR(2),HINT(2),ASINO,T201,HINT(2),ASINT,PHR(2),
C 3 SCONST,NOAHT,MPLACE(70),PLACE(70),NMLA(70),NDFILE,
C 4 SPECCH,TAPE(202),PTMEL,PEX(10),TRANGE(10),TCNST, NKIND,
C 5 NF,LINES,IT,NHMP,AG(70),GG(T0),NIT
C
C C240
C
C COMMON /COMM/ V(201),DNZ(201),X(6,6),Y(6,6),NNU,ALFA(6),
C 1 ALPB(6),ALPC(6),G(6),N(6),N(6),BETA(6),A, B, C, RH, D, WF, W,
C 2 SYN, STW, TDQ, THETA(5), TESTW(6), R(20), S(20), T(3), Q-QLN, QO
C 3 DDL, LABEL, QQT, QNTQT, QQTDT, QQTDT, QRT, A1J(6,6), A1J(6,6),
C 4 NUBNL
C
C LOGICAL TESTY, TEST
C DATA LEL/MHEL/EL/HO/HH/Q, L/LR/R/,LX/3/HHL/I/RLHD/
C 1 3HRH/THLHET/A/HYIJK/,ALPHA/SHALPHA/,LZ/4/HEL/3/
C
C C TEST(14)--INTERN CARD HAS BEEN READ CALLING FOR INTERMEDIATE OUTPUT
C IF (.NOT.TEST(14)) GO TO 6
C DO 1 I = 1, NNU
C 1 WRITE(6,1006) I,VL11,NNU,I1,G(I1)
C 1006 FORMAT(3H0V(11,31)=F9.4,H(11,11)E13.7,H(11,21)=F7.3)
C 6 IF (ASINOT .NE. 0.0) GO TO 7
C NTV = NT
C DO 8 I = 1, NNU
C 8 IF (TEST(I4)) WRITE (6,4) (I1) 25
C 4 FORMAT(4H111=F9.3)
C 1008 CT = HCK/I11
C DO 10 I = 1, NNU
C 10 R(I11) = 0.0
C 10 U = CT * V(I11)
C IF (U.GE.30.0) GO TO 9
C 9 R(I11) = RI. S(I11) = SI. A 2 OR 3 IN THE SECOND SUBSCRIPT
C C INDICATES FIRST OR SECOND DERIVATIVE RESPECTIVELY OF RI AND SI.
C C THESE DERIVATIVES ARE USED TO OBTAIN THE DERIVATIVES OF THE Q
C C CONTRIBUTIONS IN SUBROUTINE DERIV.
C C R(I11) = EXP(-U)
C 9 S(I11) = 1./((1.+R(I11))
C 10 R(I11) = R(I11)
C R(I12) = U
C R(I13) = U
C S(I12) = R(I11)*S(I11)*U
C S(I13) = S(I12)*U - 1.0
C IF (TESTI4)) WRITE (6,1081)
C 1081 FORMAT(7H0 U = ,E13.7, 6H R = ,E13.7, 6H S = ,E13.7,3X,3H)
C 10 CONTINUE
C IF (TEST14)) WRITE (6,1005)
C 1005 FORMAT(3HOCONTRIBUTION,13X,HQ,15X,4HLN Q,11X,BH H=HO/RT,13X,
C 14HCP/R)
C
C C240
C C CALCULATE Q
C C DO = TDQ/DT.
C DDQ = T2DQ/LN Q/DTZ.
C C SUBROUTINE QSUM ACUMULATES CONTRIBUTIONS OF LN Q AND DERIVATIVES.
C C ELECTRONIC PARTITION FUNCTION--FORMULA 1.
C DD = CT*100
C DDQ = ALOGC(2STW) - DD
C DDT = -.0 * DD
C LABEL = LEL
C CALL QSUM (TEST14))
C
C HARMONIC OSCILLATOR PARTITION FUNCTION--FORMULA 2.
C DO 15 I = 1, NNU
C DDQ = DDL+QNI + ALOGC(SI11)
C DD = DDL+QNI + S(I1,3)
C 15 DDQ = DDQ+QNI + S(I1,3)
DDO = DDO - DO
CORT = 0.
LABEL = LNO
CALL QSUM (TEST114)

C RIGID ROTATOR PARTITION FUNCTION--FORMULAS 3 AND 4.
LABEL = LRR
IF (TESTW (1)) GO TO 20
Q = 1.0/(SYM * CT * B)
DO = 1.0
DDO = -1.0
GO TO 30
20 Q = (1.0/CT**3 * 3.1415927 * (A*B*C)) **0.5*1.0 / SYM
DO = 1.5
DDO = -1.5
30 QNL = ALOG(Q)
CALL QSUM (TEST114)

C END RRHO CALCULATIONS. GO TO 900(C440) TO ACCUMULATE Q FOR LEVEL.
IF (TESTW (2)) GO TO 900

C C440

C ROTATIONAL STRETCHING--FORMULA 5.
LABEL = LNRH
ONL = RNHY(II)
DO = QNL
DDO = 0.0
CALL QSUM (TEST114)
LABEL = LTHET
QL = 1.0 + (THETA(3)/TI(T) + THETA(2))/TI(T) + THETA(1))/TI(T)
ONL = ALOG(1)
DO = 1.0/(3.*THETA(3)/TI(T) + THETA(2))/TI(T) + THETA(1))/TI(T)
DDO = (12.*THETA(3)/TI(T) + THETA(2))/TI(T) + THETA(1))/TI(T)

C VIBRATIONAL-ROTATION INTERACTION USING ALPHA CONSTANTS--FORMULAS 0-10.

C FORMULA II.
QL11 = AI1J(1)*DN11
CALL DERIV (1.0,0.0,0.1,0.0,0.0)
IF (TESTW (1)) GO TO 39
QL11 = 1.5*DN11*AI11**2
CALL DERIV (1.0,0.0,0.1,0.0,0.0)
QL11 = DN11/6.*AI11**3
CALL DERIV (1.0,0.0,0.1,1.0,0.0)
QL11 = DN11/6.*AI11**3
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
IF (TESTW (1)) GO TO 39
QL11 = AI1J(1) * DN11
CALL DERIV (1.0,0.0,0.1,1.0,0.0)
QL11 = AJ11) * DN11*AI11
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
DO 37 J = 1.NNU
QL11 = AI11) * DN11
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
QL11 = AI11) * DN11*AI11
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
IFE (GTJ J) GO TO 35
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
35 QL11 = AI11) * AI11) * DN11*DN11
CALL DERIV (1.1,0.0,0.1,1.0,0.0)
37 CALL DERIV (1.1,0.0,0.1,1.0,0.0)
IFE (NOATMS .GT.2) GO TO 39

C FIRST ORDER XI--FORMULA 12.
LABEL = LX1J
DO 50 I=1.NNU
DO 50 J=1.NNU
CON = DN11 *ONJ1
IF (EQOJ) CON = CON * DN11
44 QL11 = CON(+CT) * XI(J)
50 CALL DERIV (1.1,0.0,0.1,1.0,0.0)
CALL OSUM (TEST(14))

C END CALCULATIONS FOR PANDK AND JANAF.
IF (TEST(14)) GO TO 900
C
C FIRST ORDER YIJ---FORMULA 13.
LABEL = YIJK
DO 70 I=1,NNU
DO 70 J=I,NNU
DO 70 K=J,NNU
CON = ND(I)*ND(J)+KD(I,J)+KD(J,I)+KD(I,J)
OL(I)*CON = AL(I)*CON
70 CALL DERIV (I,J,K,0,0,0,0,0)
CALL OSUM (TEST(14))
DATA LAX,HAX,LI/NG/HXJ2/LXY,

C FIRST ORDER ALPHA-XIJ INTERACTION---FORMULA 17.
LABEL = LAX
DO 100 I=1,NNU
AL = AL(I)*CT
100 CALL OERIV (I,EJ,0.0,0.0,0.0,0.0,0.0)
C
C GI¼ CORRECTION--- FORMULA 16.
LABEL = GI¼
OL(I) = GI¼(I)*CT$2.
CALL DERIV (I,J,K,0,0,0,0,0)
CALL OSUM (TEST(14))
C
C WEZE FOR DIATOMIC MOLECULES---FORMULA 15.
LABEL = L2
OL(I) = 24.*WXI*CT$2.
CALL DERIV (I,J,K,0,0,0,0,0)
CALL OSUM (TEST(14))
C
C FERMI RESONANCE---FORMULA 7.
LABEL = LFERMI
CORT = 2.*
CORT = CT*2.*CT$2.
U = CT*2.*CT$2.
R=N*EXP(-U)
SN = 1./R*

C END CALCULATIONS FOR NARA01.
141 IF (.NOT.TEST(1)) GO TO 900
IF (TEST(1)) WRITE (6,142)
142 FORMAT (26ND SECOND ORDER CORRECTIONS )
C
C XIJ - XIJ INTERACTION---FORMULAS 18 and 19.
LABEL = LX2
CORT = 2.*
DO 200 I=1,NNU
DO 200 J=I,NNU
CON = ON(I)*ON(J)
200 CALL DERIV (I,J,0,0,0,0,0,0)
C
C CALL OSUM (TEST(14))
C XIJ - YIJK INTERACTION—FORMULAS 20 AND 21.

LABEL = LXY
DO 300 I=1,NMU
DO 300 J=1,NMU
DO 300 K=1,NMU
CON = 2*ID(JK,II)+KDIIJ+KDIIJ
CALL DERIV (I,J,K,O,O1.I.O,J,K,0)
300 RETURN

C 480 IF (TESTY(61)
C 390 IF (1CARO.NE.NSUB)
C 300 CALL INPUT(LINES)
C 290 CALL OSUM (TEST(I4))
C 280 CALL OSUM (TEST(I4))
C 270 CALL OSUM (TEST(I4))
C 260 CALL OSUM (TEST(I4))
C 250 CALL OSUM (TEST(I4))
C 240 CALL OSUM (TEST(I4))
C 230 CALL OSUM (TEST(I4))
C 220 CALL OSUM (TEST(I4))
C 210 CALL OSUM (TEST(I4))
C 200 CALL OSUM (TEST(I4))
C 190 CALL OSUM (TEST(I4))
C 180 CALL OSUM (TEST(I4))
C 170 CALL OSUM (TEST(I4))
C 160 CALL OSUM (TEST(I4))
C 150 CALL OSUM (TEST(I4))
C 140 CALL OSUM (TEST(I4))
C 130 CALL OSUM (TEST(I4))
C 120 CALL OSUM (TEST(I4))
C 110 CALL OSUM (TEST(I4))
C 100 CALL OSUM (TEST(I4))
C 900 IF (TESTM(I6)) GO TO 902
C 800 RETURN
C 700 END

C CALCULATIONS FOR SPECIES WITH ONE ELECTRONIC STATE

C FMR(II) = QLNTO
C HMR(II) = DQTOT
C CPR(II) = DQTOT + 2*QDOT
C 490 RETURN
C 480 GO TO 1000

C CALCULATIONS FOR SPECIES WITH EXCITED ELECTRONIC STATES

C 902 IF (QLNTOT.LT.88.) GO TO 903
C 903 QDOT = EXP(QLNTOT)
C 904 RETURN
C 905 END

C 906 GO TO 3
FUNCTION KD(I,J)
KD = 0
IF (I.EQ.J) KD = 1
RETURN
END
SUBROUTINE DERIV (I1, I2, I3, I4, I5, J1, J2, J3, J4, J5, J6)
C
C FIND O DERIVATIVES.
C
COMMON /YCDMMN/ V, 20, YN, 20, XI6, YI6, NNU, ALFA(6), WDL0013
1 ALFA6, ALFC(6), G(6), MX(6), BETA(6), A, B, C, RH, D, W, N
DERI0003
2 SYM, STNT, T00, THETA(5), TESTM(5), R(20, 3), S(20, 3), QL(3), QL\nQLN, QD, DERI0004
3 QDQ, LABEL, QTOT, QLMTOT, DQTOT, DDQTOT, CORT, AIJ6, AII, A116, NSUB
DERI0005
C
C DIMENSION I(5), J(6)
I(1) = 11
I(2) = I2
I(3) = 13
I(4) = 14
I(5) = 15
J(1) = J1
J(2) = J2
J(3) = J3
J(4) = J4
J(5) = J5
J(6) = J6
DATA IFERM1/5HFERMI/
IF (LABEL.EQ.IFERM1) GO TO 8
QL(2) = 0.
QL(3) = 0.
8 DO 10 IR = 1, 5
K = I(IR)
IF (K.EQ.0) GO TO 20
QL(1) = QL(1) + R(K, 1)
QL(2) = QL(2) + R(K, 2)
10 QL(3) = QL(3) + R(K, 3)
20 DO 30 IS = 1, 6
K = J(IS)
IF (K.EQ.0) GO TO 40
QL(1) = QL(1) + S(K, 1)
QL(2) = QL(2) + S(K, 2)
30 QL(3) = QL(3) + S(K, 3)
40 QL(3) = QL(3) + QL(1)
QDQ = QDQ + QL(2) - CORT
DO = QDQ + QL(1) + DO
DDQ = QL(1) + QL(3) + QDQ - QDQ + DDQ
RETURN
END
SUBROUTINE QSUM (TEST)

C ACCUMULATE VALUES OF Q AND ITS DERIVATIVES.

C COMMON /WCOMMN/ V(20), DN(20), ND(20), X(6,6), Y(6,6,6), NNx, ALFA(6), WDLX
1 ALFB(6), ALFC(6), GL(6), WX(6), BETA(6), A, B, C, RH, D, W, W
2 SYM, SYMT, TOO, THETA(6), TESTN(6), A(20,3), S(20,3), Q, QLN, DQ, QSUM0004
3 DDO, LABEL, DQTOT, DQNTOT, DQXTOT, DQQTOT, CORT, AIJA(6), AIIA(6) + NSUMQSUM0005

C C500
C C

LOGICALTEST
IF (.NOT.TEST) GO TO 0
Q = 0.
IF (ABS(QLN).LE.0.) Q = EXP(QLN)
QSUM0016
CPROUT = DQQ + 2. * DQ
WRITE (6,6) LABEL, Q, QLN, DQ, CPROUT
QSUM00017
6 FORMAT(4X,4E14.3F18.8)
QSUM0018
8 QLNQT = QLNQT + QLN
QSUM0020
QQTOT = QQTOT + DQ
QSUM0021
DQQTOT = DQQTOT + DDQ
QSUM00022
QLN = 0.0
QSUM0024
DDQ = 0.0
QSUM0025
RETURN
QSUM0026
END
QSUM0028
SUBROUTINE DELH

COMMON NAME(2),SYMBOL(70),ATMT(70),R,HCK,ELCTRIC,ICAR,T,WORD(5),
1 WORD(14),TEST(201),WEIGHT,FORMULA(5),MLA(5),BLANK,LEMHT(70),DELH(303)
2 NATION,CAP(202),HRMT(202),ASIND,T(202),ASIND,FMT(202)
3 SCON,NOATNS,MPLACE(70),PLACE(70),MLA(70),NOFILE
4 SPEECH,TAPE(606),PMELT,PEI(101),TRANGE(101),TCONST,MKING
5 NFLO,FLX,MIT,NEP,AG(70),GI(70),NIT,PLH(296H),HEAT,JF(51)

COMMON/PCHX,NF1,NF2,ANS(9,15),TC(101),TNC,NFP,DATE,NN,NLAST

C

C510

C

C520

C

C530

C

C540

C

C550

C

C560

C

C570

C

C580

C

C590

C

C600

C

C610

C

C620

C

C630

C

C640

C

C650

C

C660

C

C670

C

C680

C

C690

C

C700

C

C710

C

C720

C

C730

C

C740

C

C750

C

C760

C

C770

C

C780

C

C790

C

C800

C

C810

C

C820

C

C830

C

C840

C

C850

C

C860

C

C870

C

C880

C

C890

C

C900

C

C910

C

C920

C

C930

C

C940

C

C950

C

C960

C

C970

C

C980

C

C990

C

END

DELH0001
DELH0002
DELH0003
DELH0004
DELH0005
DELH0006
DELH0007
DELH0008
DELH0009
DELH0010
DELH0011
DELH0012
DELH0013
DELH0014
DELH0015
DELH0016
DELH0017
DELH0018
DELH0019
DELH0020
DELH0021
DELH0022
DELH0023
DELH0024
DELH0025
DELH0026
DELH0027
DELH0028
DELH0029
DELH0030
DELH0031
DELH0032
DELH0033
DELH0034
DELH0035
DELH0036
DELH0037
DELH0038
DELH0039
DELH0040
DELH0041
DELH0042
DELH0043
DELH0044
DELH0045
DELH0046
DELH0047
DELH0048
DELH0049
DELH0050
DELH0051
DELH0052
DELH0053
DELH0054
DELH0055
DELH0056
DELH0057
DELH0058
DELH0059
DELH0060
DELH0061
DELH0062
DELH0063
DELH0064
DELH0065
DELH0066
DELH0067
DELH0068
DELH0069
DELH0070
DELH0071
DELH0072
DELH0073
DELH0074
DELH0075
DELH0076
DELH0077
DELH0078
DELH0079
DELH0080
DELH0081
DELH0082
DELH0083
DELH0084
DELH0085
DELH0086
DELH0087
DELH0088
DELH0089
DELH0090
DELH0091
DELH0092
DELH0093
DELH0094

53
SUBROUTINE TABLES

LIST FIRST 2 TABLES OF THERMODYNAMIC FUNCTIONS.

COMMON NAME(2),SYMBOL(70),ATMWT(70),R,H,C,ELCTR,ICARD,IWORD(5)

1 WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELMTN(70)
2 NATURE,RT,CPR(202),HRT(202),ASINDH(102),ESIND,FPRT(202)
3 NCHMT,NMAPS,NPLACE(70),NPLAC(70),NMLL(70),NDFILE
4 SPECH,TAPE(202),3,PTMELT,PEX(10),TRENGE(10),TCONST,NKIND
5 NPLINES,I,TR,NTMP,AGI(70),GG(70),NIT,PI,H298HR

C LOGICAL TEST

EQUIVALENCE (IT,TT)

DIMENSION FMT(12),F(26),HEAD(22),HEAD2(20)

DATA FMT(1)/4/HH,FMT(3)/6/IFIT/0/.

1 IFIT(11)=10,9,3,IFIT(2)/14,7,IFIT(3)/14,4,IFIT(4)/6,8,IFIT(5)/16,4.
2A6,(/ (FJ,J=11,26) 96H(H-H298(F-H298))/RT H/RT,12X,5H-F/RT.
3H-H298(F-H298) = H1K,2H-F (F-H0,H-H0 -F-H0)/RT.
4/H(1/1) = 126/(HH0K,H0 8,4HCP/R8K).
5 (H-H298)/RT,6HX5/HX8X,25H-(F-H0)/RT -(F-H298)/RT,8X,4H/R/RT.
612X,5H-F/RT // / (HEAD(1),L=1,19)/11AH1HH0K,H0 8,2HCP,11X.
7 19H-H0 = H-H29810K,H0,9X,23H-F-H0 -(F-H298).
8B10K,(1H,14K,2H-F) // /.

WRITE(6,11) NAME(11),NAME(2)

11 FORMAT (1H0,2A6)

DO 2000 I = 4,8

2000 FMT(1) = FF(3)

FMT(9) = FF(7)

DO 2005 K=15,16

HEAD(K+3) = FF(K-1)

HEAD2(K+2) = FF(K+5)

HEAD(K) = FF(10)

HEAD(K-1) = FF(10)

2005 HEAD2(K-1) = FF(10)

HEAD(20) = FF(16)

HEAD(8) = FF(10)

IF (TEST(13)) GO TO 2008

HEAD(6) = FF(22)

HEAD(7) = FF(26)

HEAD(13) = FF(23)

HEAD(14) = FF(26)

HEAD(26) = FF(24)

HEAD2(12) = FF(25)

HEAD(13) = FF(11)

DO 25 I=1,NT

IF (FMT(I) .EQ. 0.) GO TO 25

H298HR = HRT(1) * T(I)

25 CONTINUE

GO TO 24

24 HEAD(8) = FF(11)

HEAD(9) = FF(13)

HEAD(11) = FF(12)

HEAD(16) = FF(13)

HEAD(2) = FF(17)

HEAD2(12) = FF(18)

HEAD2(13) = FF(19)

GO TO 2010

C

2008 HEAD(6) = FF(11)

HEAD(7) = FF(13)

HEAD(13) = FF(12)

HEAD(14) = FF(13)

HEAD2(6) = FF(17)

HEAD2(12) = FF(18)

HEAD2(13) = FF(19)

2009 HH29 = FF(10)

FM29 = FF(10)

FMT(5) = FF(6)

FMT(8) = FMT(5)

2010 IF (TEST(8)) GO TO 2020

H = FF(10)

F = FF(10)

FMT(9) = FF(19)

DO 2015 KK=18,19

HEAD(KK+1) = FF(10)

2015 HEAD(KK-1) = FF(10)

HEAD(181) = FF(10)
C570
C 2020 DO 3000 NTABLE = 1, 2
IF(IFTOS(8)) GO TO 55
WRITE (6, 51)
51 FORMAT (3OHNO ZERO VALUE IS AVAILABLE
GO TO 22
55 IFI.NOT.TEST(13)) WRITE(6, 56) ASINOH
56 FORMAT (8HOH298 = FH2.3)
IFTI.\NOT.TEST(13)) WRITE(6, 57) ASINOH
57 FORMAT (8HOH298 = FH2.3)
22 IF(NTABLE.EQ.1) WRITE(6, HEAD1)
IFTI.\NOT.TEST(13)) WRITE(6, HEAD2)
FMT(10) = FMT(9)
LINES = LINES + 8
DO 399 I = 1, NT
IT = T(I)
FMT(2) = FF(1)
IF(TI.\NOT.TEST(I)) GO TO 2130
T(I) = T(I)
FMT(2) = FF(2)
2130 ART = R*TI(I)
AR = R
IF (NTABLE.EQ.2) GO TO 2135
AR = 1.
ART = L.
2135 CP = CPRI(I)*AR
HH = HHR(I) * ART
S = (FHR(I) + HHR(I)) * AR
FH = FHRT(I) * ART
IF (.NOT.TEST(8)) GO TO 2120
H = (HHR(I) + ASINOH/R/T(I))*ART
F = (FHRT-I) - ASINOH/R/T(I))*ART
2120 IF (H29BHR.EQ. 0.) GO TO 250
H29 = (HHR(I) - H29BHR/T(I))*ART
FH29 = (FHR(I) + H29BHR/T(I))*ART
250 WRITE (6, FMT) T, CP, HH, HH29, S, FH, FH29, H, F
LINES = LINES + 1
IF(LINES.GE.55) CALL PAGEID(LINES)
399 CONTINUE
CALL PAGEID(LINES)
IF (NTABLE.EQ.2) GO TO 4000
DO 2100 I = 4, 8
2100 IF(FMT(I).EQ.FF(3)) FMT(I) = FF(4)
FMT(6) = FF(5)
IF(IFTOS(8)) FMT(9) = FF(8)
3000 CONTINUE
4000 RETURN
END
SUBROUTINE LOGK
COMMON NAME(2),SYMBOL(70),ATMWT(70),R,H,C,ELCTR,ICARD,1WORD(119)
1 WORD(4),TEST(20),WEIGHT,FORMULA(95),MLA(15),BLANK,ELEMENT(70)
2 NATOM,NC,H(202),HRT(202),ASINOH(70),ASINDT(202),ASINDT,HRTZ(202)
3 SCONST,NOATOMS,NPLACE(70),LPLACE(70),NMLA(70),NDFILE.
4 SPECHE,TAPE(202),PTMIEL,EXPI(10),TRANGE(10),TCONST,NKIND.
5 NP,LINES,NTP,HTMP,AG(70),GOT(70),NIT,P,H298HR,H,HEAT,JF(5)
C
C580
C
C DIMENSION LENTIC(4),PT(4),DI2(2,202),DHO(2),NTX(2),INIT(2),FF(10)
1 UORO(41,TEST(20),WEIGHT,FORMLA(95),MLA(15),BLANK,ELEMENT(70)
2 NATOM,NC,H(202),HRT(202),ASINOH(70),ASINDT(202),ASINDT,HRTZ(202)
3 SCONST,NOATOMS,NPLACE(70),LPLACE(70),NMLA(70),NDFILE.
4 SPECHE,TAPE(202),PTMIEL,EXPI(10),TRANGE(10),TCONST,NKIND.
5 NP,LINES,NTP,HTMP,AG(70),GOT(70),NIT,P,H298HR,H,HEAT,JF(5)
C
C
C
C
C630

C610

C INTERPOLATION OF EF DATA

C
N=4
IF (I-M-K-1) N=N-M-K+1
IF (I-K-2) K=K
IF (K-EQ.K-1) K=K
IF (I-EQ.K-1) K=K
N=K+1
DO 2000 L=2,3
TK(I) = 0.0
DO 2000 J=K2,NK
TK(I) = 1.0
DO 1000 JM=1,N
JM=J+JM-1
IF (TLEE(JM)) EQ TEE(I11) GO TO 1000
TK(I) = TK(I)*T(111)-K111*TF(111)-TF(111)*TK(I)
GO TO 211

C CALCULATE DELTA H AND DELTA F

C

2000 T(E) = T(K)*T(K)*T(E)

C

1000 CONTINUE

200 CONTINUE

213 IF (KNT1(111)) OR (I111=LNT1(I111)) NT(I111)=I-1
GO TO 500

C

C LIST HEADING OF FIRST TABLE

C

501 WRITE (6,320) NAME111, NAME112

320 FORMAT (1X,266)

501 WRITE (6,321) NAME111, NAME112

321 FORMAT (1X,266)

220 FORMAT (1H,77X,42HREFERENCE ELEMENTS GASEOUS ATOMS)

205 FORMAT (1H,75X,42HREFERENCE ELEMENTS GASEOUS ATOMS)

205 FORMAT (1H,75X,42HREFERENCE ELEMENTS GASEOUS ATOMS)

205 FORMAT (1H,75X,42HREFERENCE ELEMENTS GASEOUS ATOMS)

C

C620

C

1010 CONTINUE

1010 CONTINUE

1010 CONTINUE

1010 CONTINUE

1010 CONTINUE

C

C630

C

LD = 9
DD 803 LL=1,2
8Z = ZERO
IF (DGOII111) EQ ZERO GO TO 26

57
CZ = BLK
IF(DOLL(1,1).EQ.BLK) GO TO 26
IF((OHO(LL).EQ.BLK).OR.O.LT.INIT(LL)) GO TO 26
DILL(1,1) = DILL(1,1) + DHOI1/RT
DILL(2,1) = DILL(2,1) - DHOII/RT
FMT11(D) = FMT11(7)
GO TO 803
26 DILL(1,1) = BLK
DILL(2,1) = BLK
FMT11(D) = FF(15)
GO TO 217
C
C640
C
C CALCULATE DIMENSIONAL PROPERTIES, DELTAH, AND LOGK
C
18 CP = CPR11 * R
HH = HHR11 * RT
S = SR * R
H = HRT * RT
FH = FHRT(I) * RT
F = FRT * RT
402 FMT2(JX1) = FF(61
00 402 JX = JX1
ID = 8
00 404 LL1,2
IF(O(LL.1).NE.ZERO) GO TO 403
O(LL,1,1) = ZERO4
O(LL,2,1) = ZERO4
GO TO 404
O(LL1,1) = O(LL,1,1)/2 - 3025851
403 O(LL1,2) = O(LL,2,1)/2 - 3025851
404 LO = 10
217 FMT1(1) = FF(1)
IF (IK.EQ.0) GO TO 2999
DO 104 IX = 1,K
IF (MARK(IX).EQ.1) FMT1(1) = FF(2)
104 CONTINUE
2999 FMT2(I) = FMT2(I+1)
IF (NTABLE.EQ.2) RETURN
C
C WRITE HEADING OF ZNO TABLE AND PROPERTIES FOR 0 DEGREES
C
LINES = 7
WRITE (6,2201 NAME(1), NAME(2)
WRITE (6,2201 IF(.NOT.TEST(13)) GO TO 1221
WRITE (6,221) T, CP, H-H298, S
IF INTABLE .EQ.2) GO TO 235
WRITE (6,FMT2) TI, CPR11, HHR11, SR, HRT, FHRT(I), FRT
GO TO 236
235 WRITE(6,FMT211) CP, HH, S, FH, T(I,1,1), T(I,1,2), LL=1,21
236 LINES = LINES+1
IF (AMOD(I,1)+500.0I.EQ.0.0I GO TO 600
WRITE(6,237)
WRITE (6,FMT3) S, S, ASINOH, OHO(I), OHO(2)
RETURN
237 FORMAT (1H0 A CHANGE IN PHASE OF AN ASSIGNED REFERENCE ELEMENT LOGKO247
1 HAS OCCURRED BETWEEN THIS TEMPERATURE AND THE PRECEDING ONE. LOGKO248
GO TO 229
1221 WRITE (6,221)
WRITE (6,221)
WRITE (6,FMT3) SI, S, ASINOH, DHO(I), DHO(2)
RETURN
229 CONTINUE
END
LOGK0182
LOGK0183
LOGK0184
LOGK0185
LOGK0186
LOGK0187
LOGK0188
LOGK0189
LOGK0190
LOGK0191
LOGK0192
LOGK0193
LOGK0194
LOGK0195
LOGK0196
LOGK0197
LOGK0198
LOGK0199
LOGK0200
LOGK0201
LOGK0202
LOGK0203
LOGK0204
LOGK0205
LOGK0206
LOGK0207
LOGK0208
LOGK0209
LOGK0210
LOGK0211
LOGK0212
LOGK0213
LOGK0214
LOGK0215
LOGK0216
LOGK0217
LOGK0218
LOGK0219
LOGK0220
LOGK0221
LOGK0222
LOGK0223
LOGK0224
LOGK0225
LOGK0226
LOGK0227
LOGK0228
LOGK0229
LOGK0230
LOGK0231
LOGK0232
LOGK0233
LOGK0234
LOGK0235
LOGK0236
LOGK0237
LOGK0238
LOGK0239
LOGK0240
LOGK0241
LOGK0242
LOGK0243
LOGK0244
LOGK0245
LOGK0246
LOGK0247
LOGK0248
LOGK0249
LOGK0250
LOGK0251
LOGK0252
LOGK0253
LOGK0254
LOGK0255
LOGK0256
LOGK0257
LOGK0258
LOGK0259
LOGK0260
LOGK0261
LOGK0262
LOGK0263
LOGK0264
LOGK0265
LOGK0266
LOGK0267
LOGK0268
LOGK0269
LOGK0270
SUBROUTINE LEAST
COMMON NAME(2), SYMBO(70), ATMNT(70), R, H, CK, ELECTR, I, CARD, WORD(5),
1 WORD(4), TEST(20), WEIGHT, FORML(45), MLA(45), BLANK, LENMT(70), LEAS0002
2 NATOM, NAT, CPR(202), HINT(202), ASINDH, T(202), ASINDT, HINT(202), LEAS0003
3 SCON, N, MATMS, MPLACE(70), LPLACE(70), NML(70), NDFILE, LEAS0004
4 SPECH, TAPE(202), 3, PTIELT, EXP(10), TRANGE(10), TCONST, NKIND,
5 NF, LINES, NTRANG, NMP(70), GO(70), NET, LEAS0005
COMMON/PCH, LEVEL, NFl, NF2, ANS(I9), TCI(10), TNC, NF1, DTF, NNN, NLAST,
D DIMENSION A(15,16), ANSTYP(I5), F(4), FC(4), E(4), ERR(4),
ITF(4), TOTREL(4), TOTSO(4),
TOSOR(4), AVERR(4), LEAS0006
NAMER(4), AVSOR(4), MAEXR(4), MAXREL(4), MAX(4), TVMAXRL(4),
SRELLR(4), LEAS0007

C
C60
LOGICAL TEST
REAL MAXERR, MAXREL
WRITE (6,2)
1 FORMAT (1H LEAST SQUARES ,/)
LINES = 7
DO 3 I = 1,3
DO 4 J = 1,15
3 ANS(I,J) = 0.0
DO 4 I = 1,5
4 EXP(I) = 1 - 1
6 NF1 = NF1
NF2 = NF2
NF3 = NF3
NF4 = NF4
NF5 = NF5
NF6 = NF6
IDONEA = 0
IDONEB = 0
IDONET = 0
ICONST = 0
IF (TCONST .GE. 0.01) GO TO 7
TCONST = 1000.
IF (PTMELT .GE. 0.01) TCONST = PTMELT
7 IF(NTRANG.EQ.0) GO TO 1006
C
C670
SORT IN INCREASING ORDER TEMPERATURES SPECIFYING INTERVALS
C
8 J = 1
9 M = J
10 DO 12 I=J,NTRANG
11 M = I
12 CONTINUE
13 TEMP = TRANGE(M)
TRANGE(M) = TRANGE(J)
TRANGE(J) = TEMP
GO TO 10
14 J = J+1
1006 TRANGE(1) = 300.0
TRANGE(2) = 100.0
TRANGE(3) = 500.0
NTRANG = 3
1007 DO 24 I = 1, NTRANG
24 TCI(I) = TRANGE(I)
NTRANG = ITRANGE
SAVEC = TCONST
IF(TCONST.GT.T(NNN)) TCONST = T(NNN)
IF (NNN.EQ.1) GO TO 1023
IF (T(NNN-1).GT.T(NNN)) DR.TCONST.LT.T(NNN) TCONST=T(NNN)
C
C680
K = NTRANG - 1
IF ((TRANGE(1).GE.T(NNN))-.00001).AND.((TRANGE(NTC).LE.T(NLAST)+
.00001)) GO TO 1021
LINES = LINES + 2
DO 1028 I = 1, K
1028 CONTINUE
1032 TRANGE(1) = T(NNN)
1035 DO 1038 I = 1, K
IK = K + 2 - I
IF (TRANGE(IK).LE.T(NLAST)+.00001) GO TO 1021
IF (T(NLAST).GT. TRANGE(IK-1)+.00001) GO TO 1042
NTRANG = NTRANG - 1
1038 CONTINUE
LEAS0013
LEAS0014
LEAS0015
LEAS0016
LEAS0017
LEAS0018
LEAS0019
LEAS0020
LEAS0021
LEAS0022
LEAS0023
LEAS0024
LEAS0025
LEAS0026
LEAS0027
LEAS0028
LEAS0029
LEAS0030
LEAS0031
LEAS0032
LEAS0033
LEAS0034
LEAS0035
LEAS0036
LEAS0037
LEAS0038
LEAS0044
LEAS0045
LEAS0046
LEAS0047
LEAS0048
LEAS0049
LEAS0050
LEAS0051
LEAS0052
LEAS0053
LEAS0054
LEAS0055
LEAS0056
LEAS0057
LEAS0058
LEAS0059
LEAS0060
LEAS0061
LEAS0062
LEAS0063
LEAS0040
LEAS0041
LEAS0042
LEAS0043
LEAS0044
LEAS0065
LEAS0066
LEAS0067
LEAS0088
LEAS0089
LEAS0090
LEAS0104
LEAS0110
LEAS0111
LEAS0112
LEAS0114
LEAS0115
LEAS0116
LEAS0117
LEAS0118
LEAS0121
LEAS0122
GO TO 1021
1042 TRANGE(I) = TINLAST
C LOCATE TEMPERATURE CONSTRAINTS
1021 DO 17 I=1,NTRANG
   IF (ABS(TRANGE(I)-TCONST),LT.0.00001) GO TO 1017
   IF (TRANGE(I).GT.TCONST) GO TO 18
17 CONTINUE
18 I = I - 1
1018 TRANGE(I) = TCONST
1017 ICONST = I
C ADJUST TEMPERATURE INTERVALS, IF NECESSARY
C DO 21 I=NNN,NLAST
21 IF (ABS(T(I)-TCONST),LT.0.00001) GO TO 23
23 CPRCON = CPR(I)
   HHRTCN = HHRT(I)
   SRCON = FHRT(I) + HHRT(I)
   GO TO 40
20 DO 41 I=NNN,NLAST
   IF (ABS(T(I)-TLOW),LT.0.00001) GO TO 44
41 CONTINUE
40 DO 41 I=NNN,NLAST
41 IF (ABS(T(I)-TLOW),LT.0.00001) GO TO 44
44 NEND = I
   GO TO 1000
   DO 50 I=NBEGIN,NLAST
50 IF (ABS(T(I)-TLOW),LT.0.00001) GO TO 52
52 NEND = I
   GO TO 1000
C CLEAR MATRIX REGION
C 50 DO 51 I=1,NF5
   DO 52 J=1,NF6
51 A(I,J) = 0.0
52 GO TO 50
C SET UP MATRIX ELEMENTS FOR DIAGONAL AND ABOVE FOR FIRST NF ROWS
C 80 IF (EXP(I)*NE.(-1.0)) GO TO 85
   A(I,NF5) = 1.0/TFIX
   A(I,NF4) = ALOG(TFIX)/TFIX
   A(I,NF5) = -1.0/TFIX
   DO 83 L=NBEGIN,NEND
   C A(I,NF1) = A(I,NF1)+ALOG(T(I))+ALOG(T(I))*T(I)
   A(I,NF2) = A(I,NF2)+ALOG(T(I))
   SR = FHRT(L) + HHRT(L)
   A(I,NF6) = A(I,NF6)+ALOG(T(I))*SR/T(I)
   GO TO 99
83 CONTINUE
C 85 IF (EXP(I)*NE.0.0) GO TO 90
   A(I,NF1) = 1.0
   A(I,NF5) = 1.0
   A(I,NF5) = ALOG(TFIX)
   DO 89 L=NBEGIN,NEND
   A(I,NF1) = A(I,NF1)+1.0/T(I)
   A(I,NF2) = A(I,NF2)+ALOG(T(I))
   SR = FHRT(L) + HHRT(L)
   A(I,NF6) = A(I,NF6)+ALOG(T(I))*SR/T(I)
   GO TO 99
89 CONTINUE
C 90 A(I,NF3) = TFIX*EXP(I)
   A(I,NF6) = A(I,NF3)/EXP(I)+1.0
   A(I,NF5) = A(I,NF3)/EXP(I)
GO TO 1000
DO 92 L=NBEGIN,NEND
   AI(NF1) = AI(NF1)**(EXP(I)-1.0)/(EXP(I)+1.0)
   AI(NF2) = AI(NF2)**(EXP(I))/EXP(I)
   SR = FHRT(L) + HHRT(L)
92   AI(NF6) = AI(NF6) + CPRFX + HHRT(L)/(EXP(I)+1.0) + SR /EXP(I)
1*ITL**EXPI1
C
CT10
C 99 DO 400 J=K,NF
   100 IF (EXP(I)+1.0) 130,105,130
   105 IF (EXP(I)+1.0) 115,110,111
   110 DO 112 L=NBEGIN,NEND
   112 A(I,J) = A(I,J) + (2.0+ALOG(T(L)))*ALOG(T(L)))/(T(L)**T(L))
        GO TO 400
C
   115 IF (EXP(I)) 125,120,125
   120 DO 122 L=NBEGIN,NEND
   122 A(I,J) = A(I,J) + 1.0/T(L)
        GO TO 400
C
   125 EXPIJ = EXPIJ
   126 DO 127 L=NBEGIN,NEND
   127 A(I,J) = A(I,J) + ((EXPIJ+1.0)/EXPIJ+1.0)**(1*T(L)**(EXPIJ-1.0))
        GO TO 400
C
   130 IF (EXP(I) + 1.0) 145,135,145
   135 IF (EXP(I)) 160,120,140
   140 EXPIJ = EXPIJ
        GO TO 126
C
   145 IF (EXP(I)) 165,150,155
   150 IF (EXP(I)) 160,155,160
   155 DO 157 L=NBEGIN,NEND
   157 A(I,J) = A(I,J) + 2.0*ALOG(T(L))**2
        GO TO 400
C
   160 EXPIJ = EXPIJ
   161 DO 163 L=NBEGIN,NEND
   163 A(I,J) = A(I,J) + ((EXPIJ+2.0)/EXPIJ+1.0)*ALOG(T(L))
1* T(L)**EXPIJ
        GO TO 400
C
   165 IF (EXP(I)) 175,170,175
   170 EXPIJ = EXPIJ
        GO TO 161
C
   175 DO 177 L=NBEGIN,NEND
   177 A(I,J) = A(I,J) + (1.0/((EXP(I)+1.0)*EXP(I)+1.0))
1* 1.0/EXP(I)**EXP(I)**T(L)***(EXP(I)+EXP(I))
C 400 CONTINUE
500 K = K+1
C
C SET UP MATRIX FOR DIAGONAL AND ABOVE FOR REMAINING ROWS
C
DO 510 L=NBEGIN,NEND
   A(NF1,NF1) = A(NF1,NF1) + 1.0/(T(L)**T(L))
   A(NF1,NF2) = A(NF1,NF2) + HHRT(L)/T(L)
   A(NF2,NF2) = A(NF2,NF2) + 1.0
510   A(NF2,NF6) = A(NF2,NF6) + HHRT(L) + HHRT(L)
   A(NF1,NF4) = 1.0/FIX
   A(NF2,NF5) = 1.0
   A(NF3,NF6) = CPRFX
   A(NF4,NF6) = HHRTX
   A(NF5,NF6) = SFIX
C
C720
C
C COMPLETE THE MATRIX BY REFLECTING SYMMETRICAL ELEMENTS ABOVE DIAGONAL

C
K = 2
DO 520 I=1,NF4
   DO 518 J=K,NF5
   518 A(I,J) = A(I,J)
520   K = K+1
C
C SOLVE THE MATRIX.
C
N=NFS
DO 551 I=1,N
   551 ANSTPY(I) = 0.0
   DO 560 I=1,N
   560 A(I,J) = A(I,J)/A(I,I)
   IF (I-N) 557,570,557
557 CONTINUE
   K=I+1

CONTINUE
500 K = K+1
C
C SET UP MATRIX FOR DIAGONAL AND ABOVE FOR REMAINING ROWS
C
DO 510 L=NBEGIN,NEND
   A(NF1,NF1) = A(NF1,NF1) + 1.0/(T(L)**T(L))
   A(NF1,NF2) = A(NF1,NF2) + HHRT(L)/T(L)
   A(NF2,NF2) = A(NF2,NF2) + 1.0
510   A(NF2,NF6) = A(NF2,NF6) + HHRT(L) + HHRT(L)
   A(NF1,NF4) = 1.0/FIX
   A(NF2,NF5) = 1.0
   A(NF3,NF6) = CPRFX
   A(NF4,NF6) = HHRTX
   A(NF5,NF6) = SFIX
C
C720
C
C COMPLETE THE MATRIX BY REFLECTING SYMMETRICAL ELEMENTS ABOVE DIAGONAL

C
K = 2
DO 520 I=1,NF4
   DO 518 J=K,NF5
   518 A(I,J) = A(I,J)
520   K = K+1
C
C SOLVE THE MATRIX.
C
N=NFS
DO 551 I=1,N
   551 ANSTPY(I) = 0.0
   DO 560 I=1,N
   560 A(I,J) = A(I,J)/A(I,I)
   IF (I-N) 557,570,557
557 CONTINUE
   K=I+1

CONTINUE
DO 558 II=K,N
DO 558 JJ=J,N
558 A(I,J,J+1) = -A(I,J)*A(I,J+1) + A(I,J+1)
560 CONTINUE
570 ANSTPY(N) = A(I,J+1)
IF (IN-I) 571,580,571
571 J = N-1
DO 572 M=I,1
ANSTPY(J) = ANSTPY(J) + ANSTPY(K)*A(J,K)
572 K = J+1
ANSTPY(J) = A(J,K) - ANSTPY(J)
573 J = J-1
C C
C 580 DO 581 I=1,NF5
581 ANSTPY(I) = ANSTPY(I)
C730C
C CALCULATE FROM THE LEAST SQUARES COEFFICIENTS VALUES OF CP/R,HO/HRT.
C TEMPERATURE. ALSO THE AVERAGE ERROR, AVERAGE RELATIVE ERROR,
C LARGEST ERROR AND LARGEST RELATIVE ERROR.
C TOTERR(N) = TOTERR(N) + ERR(N)*A(N)*A(N)
C IF (TOTAL(N).NE.0.0) GO TO 584
WRITE (6,603) C
584 16,603 FORMAT (5F10.4)
WRITE (6,603) C
2TION)
C
C LINES = LINES + 3
DO 605 I=1,4
TOTERR(I) = 0.
TOTSQR(I) = 0.
MAXERR(I) = 0.
MAXREL(I) = 0.
605 TMAXR(I) = 0.
C C
C DO 635 L=NBEGIN,NEND
F1 = CPR(I)
F2 = HRT(I)
F3 = CPR(I) + HRT(I)
F4 = CPR(I)
FC1 = 0.
FC2 = ANSTPY(NF1)/T(I)
FC3 = ANSTPY(NF2)
C
C DO 618 I=1,NF
TP = T(I)**EXP(1)
IF (EXP(1).NE.0.0) GO TO 610
FC2 = FC2*ANSTPY(I)*ALOG(T(I))/T(I)
GO TO 616
610 FC2 = FC2*ANSTPY(I)*EXP(1)*T(I)
IF (EXP(1).NE.0.0) GO TO 616
FC3 = FC3*ANSTPY(I)*ALOG(T(I))
GO TO 618
616 FC3 = FC3*ANSTPY(I)*EXP(1)
618 FC1 = FC1*ANSTPY(I)*TP
FC4 = FC3-FC2
C
C IF (L.NE.NBEGN.OR.TRANGE(ILOW).GE.TCONST) GO TO 705
IGNEB = IGNEB + 1
GO TO 706
705 IF (L.NE.NEND.OR.TRANGE(ILOW).LT.TCONST) GO TO 707
IGNEA = IGNEA + 1
706 CPRFIX = FC(1)
HRTFX = FC(2)
SRFIX = FC(3)
707 DO 622 I=1,4
ERR(I) = F1-FC(I)
ABSErr = ABSErr(I)
TOTERR(I) = TOTERR(I)+ABSErr
TOTSQ(I) = TOTSQ(I)+ABSErr*ABSErr
IF (EXP(1).NE.0.0) GO TO 619
WRITE (6,619) 584
619 FORMAT (5F10.4) 590
WRITE (6,619) 590
619 REFER(I1) = ERR(11)/F(I1)        LEAS0368
       ABSREL = ABS(RELERR(I))           LEAS0369
       TOTREL(I1) = TOTREL(I1) + ABSREL     LEAS0370
       TOTSQR(I1) = TOTSQR(I1) + ABSREL*ABSREL LEAS0371
       IF (ABSERR,LT,MAXERR(I1)) GO TO 620    LEAS0372
       MAXERR(I1) = ABSERR                LEAS0373
       TMAX(I1) = T(I1)                   LEAS0374
620 IF (ABSREL,LT,MAXREI(I1)) GO TO 622  LEAS0375
       MAXREL(I1) = ABSREL                LEAS0376
       TMAXRI1) = T(I1)                   LEAS0377
622 CONTINUE                           LEAS0378
       C
WRITE (6,625) (T(I),CPRI1),FC(I1),HRHI1),FC2(I),F3(I),F4(I),F5(I)
IFC(I)                                 LEAS0380
625 FORMAT (F12,2,2F13.7,2X,2F13.7,2X,2F14.7,2X,2F14.7)          LEAS0382
WRITE (6,627) HERR(I1),REFER(I1),I=1,I4                     LEAS0383
627 FORMAT (I12X,19HMAX REL ERR SIR =.F10.6.4X.6HTEMP =.F7.0.6X.ZOHAVEL
       LINES = LINES + 2                  LEAS0384
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0386
635 CONTINUE                           LEAS0387
       C
C750 POINTS = NEND-NBEGIN + 1          LEAS0390
       D0 640 I=1,4                          LEAS0391
       AVERRI1) = TOTERR1)/POINTS          LEAS0393
       AVERRL1) = TOTREL(1)/POINTS         LEAS0394
       AVSOR(1) = SQRT(TOTSQR(1)/POINTS)   LEAS0395
       640 AVSOR(I) = SQRT(TOTSQR(I)/POINTS) LEAS0396
       C
WRITE (6,641) MAXREL(I1),TMAXR(I1),AVERRL(I1),AVSOR(I1)       LEAS0399
641 FORMAT (3X,19HMAX-REL ERR CP/R =.F10.6,4X6HTEMP =.F7.0,6X20HAVEREAS0400
       1R REL ERR CP/R =.F10.6,22HREL LST SQ ERR CP/R =.F10.6)    LEAS0401
WRITE (6,642) MAXREL(I2),TMAXR(I2),AVSOR(I2)                   LEAS0402
642 FORMAT (3X,19HMAX REL ERR HH/RT =.F10.6,4X6HTEMP =.F7.0,6X20HAVEREAS0403
       1R REL ERR HH/R =.F10.6,22HREL LST SQ ERR HH/RT =.F10.6)   LEAS0404
       LINES = LINES + 2                  LEAS0405
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0406
WRITE (6,643) MAXREL(3),TMAKR(3),AVERRL(3),AVSOR(3)            LEAS0407
643 FORMAT (3X,19HMAX REL ERR S/R =.F10.6,4X6HTEMP =.F7.0,6X20HAVEREAS0408
       1R REL ERR S/R =.F10.6,22HREL LST SQ ERR S/R =.F10.6)     LEAS0409
       LINES = LINES + 2                  LEAS0410
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0412
WRITE (6,645) MAXERR(I1),TMAXR(1),AVERRL(1),AVSOR(1)           LEAS0415
645 FORMAT (7X,15HMAX ERR CP/R =.F10.6,4X6HTEMP =.F7.0,10X16HAVEREAS0416
       1R CP/R =.F10.6,10X16HTST SQ ERR CP/R =.F10.6)            LEAS0417
WRITE (6,646) MAXERR(I2),TMAXR(2),AVERRL(2),AVSOR(2)           LEAS0418
646 FORMAT (7X,15HMAX ERR HH/RT =.F10.6,4X6HTEMP =.F7.0,10X16HAVEREAS0419
       1R HH/RT =.F10.6,10X16HTST SQ ERR HH/RT =.F10.6)          LEAS0420
       LINES = LINES + 2                  LEAS0421
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0422
WRITE (6,647) MAXERR(3),TMAKR(3),AVERRL(3),AVSOR(3)            LEAS0423
647 FORMAT (7X,15HMAX ERR S/R =.F10.6,4X6HTEMP =.F7.0,10X16HAVEREAS0424
       1R S/R =.F10.6,10X16HTST SQ ERR S/R =.F10.6)              LEAS0425
       LINES = LINES + 2                  LEAS0426
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0427
WRITE (6,648) MAXERR(4),TMAKR(4),AVERRL(4),AVSOR(4)            LEAS0428
648 FORMAT (7X,15HMAX ERR FH/RT =.F10.6,4X6HTEMP =.F7.0,10X16HAVEREAS0429
       1R FH/RT =.F10.6,10X16HTST SQ ERR FH/RT =.F10.6)          LEAS0430
       LINES = LINES + 2                  LEAS0431
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0432
WRITE (6,650) (ANSTPY(I1),EXP(I1),I=1,NF)                      LEAS0433
650 FORMAT (14.11) LINI5 = LINES + 2                              LEAS0434
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0435
       HRTC = ANSTPY[N1] + ASINOH/R           LEAS0436
WRITE (6,660) ANSTPYN1),HRTC,ANSTPY(NF2)                      LEAS0437
660 FORMAT (21H (H-H0) R CONSTANT =.E15.8, 20H, H/R(A6) CONSTANT =.E15.8
       LINES = LINES + 2                  LEAS0439
       IF (LINES .GE.55) CALL PAGEID(LINES)   LEAS0440
       GO TO 25                             LEAS0441
       END                                  LEAS0445
       C
C760 IF (D0 980) I = 1#NTC                LEAS0448
980 TRANGE(I1) = TC(I1)                        LEAS0453
      NTRANG = NT C                                LEAS0456
      TCOND = SAVEC                               LEAS0459
      RETURN                                         LEAS0458
      1000 DO 980 I = 1#NTC                           LEAS0471
SUBROUTINE PUNCH

PUNCH COEFFICIENT CARDS FOR PERFORMANCE PROGRAM

COMMON NAME(21), SYMBS(70), MTHW(10), R, HCK, ELECT, ICARD, IWORD(5)
1 WORD(4), TEST(20), WEIGHT, FORM(5), MAI(5), NPLUS, ELEMT(10)
2 MATHM, CTMP(20), HRTT(20), ASIND, T(21), ASINDT, FHRT(21)
3 SCONST, NDOATM, MPLACE(11), LPLACE(11), NMAI(10), NFILE
4 SPECH1, TAP2(20, 3), PTK2L, PTK2L, TRANGE(10), TCONST, NIND
5 NFLINES, TRNTMP, AG(70), CTMP(1), NIT, Pi

COMMON /PCH/ K, NFI, NF2, ANSI(15), TCI(11), NTC, NFP, DATE, NNN, NLAST
DIMENSION DAT(251, M16)

LOGICAL TEST

EQUIVALENCE (DAT(11).NDATA1), (DAT(21).NDATA2)
DATA MAI(1:10000000), MAI(12:100777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777
APPENDIX C

MAP ROUTINES (DESCRIPTION AND LISTING)

SKFILE(N, M)

This routine causes M end-of-file marks to be skipped over on tape unit N. This routine is called for in the FORTRAN program sections C90, C530, and C590; it is as follows:

```
$IRLDR SKFILE
$TEXT SKFILE

RINARY CARD (NUT PUNCHED)
00000 1 000000 0 00004 10001 SKFILE SAVE (4,2)
00001 0774 00 2 00000 10000
00002 0774 00 4 00000 10000
00003 0620 00 4 00001 10000
00004 0634 00 4 00000 10011
00005 0634 00 4 00001 10001
00006 0634 00 4 00000 10001
00007 0634 00 2 00001 10001
00100 0500 0 0 00000 10000
00101 4734 00 2 00000 10000
00102 7 00000 1 00004 10011
00103 0500 60 4 00000 10000
00104 0734 00 2 00000 10000
00105 7 00000 2 00002 10001
00106 0500 60 4 00003 10000
00107 0621 0 0 00007 10001
00108 0634 0 0 00050 10001
00109 000000000000 00010 CALL ../FVIDI(PAT,UNITAD)
00110 00074 0 0 03000 10011

RINARY CARD (NUT PUNCHED)
00022 1 00002 0 01004 10011
00023 0 00061 0 00013 10100
00024 0 00000 0 00057 10001
00025 0 00000 0 00056 10001
00026 0500 60 0 00056 10001
00027 0734 00 4 00000 10000
00028 1 00000 1 4 00101 10011
00029 0634 0 0 0044 10001
00030 0734 0 0 06000 10011
00031 0634 0 0 00052 10001
00032 0634 0 0 00500 10011
00033 0634 0 0 01001 10011
00034 0634 0 4 05000 10011
00035 0634 0 4 00052 10011
00036 0 00000 0 01002 10101
00037 0 00061 0 00023 10100
00038 00074 0 0 07000 10011
00039 00074 0 0 04000 10011
00040 00074 0 0 01000 10011
00041 00074 0 0 00000 10011
00042 00074 0 0 00045 10001
00043 00074 0 0 00050 10001

RINARY CARD (NUT PUNCHED)
00044 0020 0 0 00045 10001
00045 0020 0 0 00000 10000
00046 0020 0 0 00000 10011
00047 0020 0 0 00000 10011
00048 0020 0 0 00000 10011
00049 0020 0 0 00000 10011
00050 0774 0C 2 00000 10000
00051 2 00001 2 01002 10011
00052 000000 0 00050 10001
00053 0634 00 2 00050 10001
00054 0634 00 2 00045 10001
00055 0634 00 2 00045 10001
00056 00000 0 00050 10001
00057 0 00000 0 00000 10000
00058 0 00000 0 00000 10000
00059 0 00000 0 00000 10000
00060 00000 0 00000 10000
00061 000000600000 10000
00062 624226314325 10000
00000 01111 END
```
BINARY CARD (NOT PUNCHED)

PREFACE

START=0, LENGTH=51, TYPE=7094, CMPLX=5

SKFILE DECK

LOC=0, LENGTH=51

SKFILE REAL

LOC=0, LENGTH=0

**FVIO VIRTUAL SECT. 3.CALL

 реализаций СИМБОЛОВ.

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUM</td>
<td>00055</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>00060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENF</td>
<td>00050</td>
<td>20, 35</td>
<td></td>
</tr>
<tr>
<td>HLD</td>
<td>00045</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>..001</td>
<td>00002</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>..002</td>
<td>00003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>..003</td>
<td>00004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUT</td>
<td>00052</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PAT</td>
<td>00057</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>REED</td>
<td>00060</td>
<td>47</td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES TO VIRTUAL SYMBOLS.

..RSF. 4 35
..FACK 7 40
..FIOC 6 37
..FVIO 8 41
SYSLOC 5 43
REFERENCES TO DEFINED SYMBOLS.

CLASS | SYMBOL | VALUE | REFERENCES
--- | --- | --- | ---
**.RSF.** | 00000 | | |
**.BSF.** | 00001 | 5 | |
LK,DR | 00014 | 1.6 | |
PON | 00010 | 0 | |
ENTRY | .BSF. | | |
**.BSF.** | | | |
**.BSF.** | | | |
**.BSF.** | | | |
**.BSF.** | | | |
**.BSF.** | | | |
**.BSF.** | | | |
**LCTR** | BLCTR | | |
**QUAL** | **UNOS** | | |
**LCTR** | // | | |
**SFL** | 00005 | 5 | |
**SIZE** | 00003 | 5.11 | |

REFERENCES TO VIRTUAL SYMBOLS.

**.FIOS** | 3 | 2 |
BCDUMP(A, B)

This routine causes data to be punched out in absolute binary cards (up to 22 words per card). The arguments A and B are the first and last words to be dumped, respectively. The routine is called for in the FORTRAN program sections C100 and C790; it is as follows:

$SBIR CHAR BCDUMP
$SBG X BCDUMP
ENTRY BCDUMP

**BINARY CARD (NOT PUNCHED)**

<table>
<thead>
<tr>
<th>BCDUMP</th>
<th>SAVE 1.2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>1 00000 0 00005 10001 BCDUMP</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>0774 00 1 00000 10000</td>
<td></td>
</tr>
<tr>
<td>00002</td>
<td>0774 00 4 00000 10000</td>
<td></td>
</tr>
<tr>
<td>00003</td>
<td>0774 00 4 00000 10000</td>
<td></td>
</tr>
<tr>
<td>00004</td>
<td>0700 00 4 00001 10000</td>
<td></td>
</tr>
<tr>
<td>00005</td>
<td>0634 00 4 00000 10011</td>
<td></td>
</tr>
<tr>
<td>00006</td>
<td>0634 00 4 00013 10001</td>
<td></td>
</tr>
<tr>
<td>00007</td>
<td>0634 00 4 00003 10001</td>
<td></td>
</tr>
<tr>
<td>00010</td>
<td>0634 00 1 00002 10001</td>
<td></td>
</tr>
<tr>
<td>00011</td>
<td>0634 00 2 00001 10000</td>
<td></td>
</tr>
<tr>
<td>00012</td>
<td>0500 00 4 00001 10000</td>
<td></td>
</tr>
<tr>
<td>00013</td>
<td>4794 00 2 00000 10000</td>
<td></td>
</tr>
<tr>
<td>00014</td>
<td>7 000002 2 10002 10011</td>
<td></td>
</tr>
<tr>
<td>00015</td>
<td>4570 60 4 00005 10000</td>
<td></td>
</tr>
<tr>
<td>00016</td>
<td>0634 00 0 00056 10001</td>
<td></td>
</tr>
<tr>
<td>00017</td>
<td>0634 00 4 00003 10000</td>
<td></td>
</tr>
<tr>
<td>00020</td>
<td>0040 00 0 01002 10011</td>
<td></td>
</tr>
<tr>
<td>00021</td>
<td>0131 00 0 00000 10000</td>
<td></td>
</tr>
<tr>
<td>00022</td>
<td>4600 00 0 06000 10011</td>
<td></td>
</tr>
</tbody>
</table>

**RINARY CARD (NOT PUNCHED)**

<table>
<thead>
<tr>
<th>LXA .BRDB1</th>
<th>PUT FIRST LOC. IX1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000 1 06000 0 01001</td>
<td></td>
</tr>
<tr>
<td>00000 0 06000 0 10001</td>
<td>SUB .BRDB1</td>
</tr>
<tr>
<td>00000 1 06000 0 01001</td>
<td>PAX 0.2</td>
</tr>
<tr>
<td>00000 1 06000 0 01001</td>
<td>TXI .1.2.1</td>
</tr>
<tr>
<td>00000 1 06000 0 01001</td>
<td>SXA IX1</td>
</tr>
<tr>
<td>00000 1 06000 0 01001</td>
<td>SXA IX2</td>
</tr>
<tr>
<td>00000 1 06000 0 10000</td>
<td>XI AX **1.1</td>
</tr>
<tr>
<td>00000 1 06000 0 10000</td>
<td>XI AX **2</td>
</tr>
<tr>
<td>00000 1 06000 0 10000</td>
<td>SXA CLA.1</td>
</tr>
<tr>
<td>00000 1 06000 0 10000</td>
<td>SXA .BRDB.1</td>
</tr>
<tr>
<td>00000 1 06000 0 10000</td>
<td>SXA XI.1</td>
</tr>
</tbody>
</table>

**RINARY CARD (NOT PUNCHED)**

| AXT 27.4 |
|----------|--------|
| 00046 1 0774 0 4 00033 10000 |
| 00047 1 0000 0 4 00034 10011 CLEAR STZ .BRDB+28.4 CLEAR THE BUFFER. |
| 00050 2 00001 4 10011 | TIX .1.4.1 |
| 00151 1 0774 0 4 00000 10000 | AXT .4 |
| 00052 0 0600 0 4 00000 10000 | CLA CLA **.4 |
| 00053 0 0601 0 4 00023 10011 | STO .BRDB+2.4 |
| 00055 1 77777 4 01001 10011 | TXI .1.4.1 |
| 00056 1 0774 0 1 00000 10000 | AXT .1.1 |
| 00057 0 0500 0 0 00137 10001 | CLA HUNBIT NUMBER |
| 00060 7 00143 1 01002 10011 | TXL **2.1.99 BCDUMP |
| 00062 1 77634 1 41002 10011 | TXI **2.1.100 CARDS |
| 00063 0674 0 0 0137 10001 | STA GP+1 FROM |
| 00064 0 0500 0 0 00141 10001 | CLA BITT 700 |
| 00065 0 0771 0 0 00001 10000 | ARS 1 | TO |
| 00066 7 00011 1 01002 10011 | TXL **2.1.9 999 |
| 00067 1 77666 1 41002 10011 | TXI **2.1.10 |
| 00070 0601 0 0 0135 10001 | STO WORD3 |
REFERENCES TO DEFINED SYMBOLS.

<table>
<thead>
<tr>
<th>CLASS SYMHL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCUDMP</td>
<td>00000</td>
<td>127</td>
</tr>
<tr>
<td>RITI</td>
<td>00141</td>
<td>84</td>
</tr>
<tr>
<td>RITH</td>
<td>00142</td>
<td>71</td>
</tr>
<tr>
<td>CLA</td>
<td>C0052</td>
<td>47</td>
</tr>
<tr>
<td>CLRFAR</td>
<td>00047</td>
<td></td>
</tr>
<tr>
<td>CHUM</td>
<td>00056</td>
<td>16.76.102</td>
</tr>
<tr>
<td>GP</td>
<td>00133</td>
<td>63.103.105</td>
</tr>
<tr>
<td>MNIK1T</td>
<td>00117</td>
<td>57</td>
</tr>
<tr>
<td>I11</td>
<td>00091</td>
<td>27.45.122.124</td>
</tr>
<tr>
<td>IX2</td>
<td>00032</td>
<td>30.35</td>
</tr>
<tr>
<td>...,N001</td>
<td>00003</td>
<td>7.10.11</td>
</tr>
<tr>
<td>...,N002</td>
<td>00004</td>
<td></td>
</tr>
<tr>
<td>...,N003</td>
<td>00005</td>
<td>0</td>
</tr>
<tr>
<td>LASTC</td>
<td>00130</td>
<td>34</td>
</tr>
<tr>
<td>LDRP</td>
<td>00044</td>
<td></td>
</tr>
<tr>
<td>OUT</td>
<td>00147</td>
<td>114</td>
</tr>
<tr>
<td>RETURN</td>
<td>00123</td>
<td>125, 130.131</td>
</tr>
</tbody>
</table>

REFERENCES TO VIRTUAL SYMBOLS.

| ..ACRD        | 4     | 116       |
| ..BCRD        | 6     | 22.23.74.40.43.47.53.104.106.110.111.113 |
| ..BCK        | 8     | 121       |
| ..FTCK       | 7     | 33.126    |
| ..PCN        | 9     | 142       |
| SYSLOC       | 5     | 5.115     |

$SIRLOR .ACRD
$STEXT .ACRD

ENTRY ..BRDB ENTRY ..BCRD
ENTRY ..BCRD ENTRY ..BRDB

00034 SIZE SET 28 RECORD SIZE 11/1/65 JMLR

BINARY CARD (NOT PUNCHED)

00000 0500 00 0 00012 10011 ..ACRD CLA PON READ ENTRY FOR BCRD 11/1/65 JMLR
00001 0620 00 0 01007 10011 TRA *2 GET CORRECT ARG FOR .FIO 11/1/65 JMLR
00002 4550 00 00013 10011 ..BCRD CAL PTH WRITE ENTRY FOR BCRDUMP 11/1/65 JMLR
00003 0614 00 00050 10001 SXA LK.DR.4 SAVE IR4 11/1/65 JMLR
00004 000000000000 00010 CALL ..FIO(SEL) SET UP READ OR WRITE 11/1/65 JMLR
00004 0607 00 0 05000 10011
00005 1 00001 0 01003 10011
00006 0 00050 0 0001 01100
00007 0 00000 0 00007 10001
00007 000007001001 00001 ORG */-1 11/1/65 JMLR
00007 3 00034 0 04000 10011 SEL IORT ..BRDB,SIZE I/O COMMAND 11/1/65 JMLR
00010 0534 00 0 0050 10001 LXA LK.DR.4 RESTORE 4 11/1/65 JMLR
00011 0C2C 00 0 00001 10000 TRA 1.4 11/1/65 JMLR
00012 1 00000 0 00000 10000 PON PON 0x0 11/1/65 JMLR
00013 9 00020 0 00000 10000 PTH PTH 0x0 11/1/65 JMLR
00014 760000000034 00001 ..BRDB BSS SIZE I/O BUFFER 11/1/65 JMLR
00050 000600000000 10000 LK.DR LDIR 11/1/65 JMLR
00051 32223516624 10000
00000 01111 END 11/1/65 JMLR

70
$CDICT .ACRD

BINARY CARD (NOT PUNCHED)
000052000000
000004000005
32272351624
666666666666
33332236624
000000000000
33332251242
000000000014
200001000000

$DCED .ACRD

REFERENCES TO DEFINED SYMBOLS.

CLASS SYMBOL VALUE REFERENCES

$FILE .PCH.'PCH ',P,P,READY,OUTPUT,BLK=28,MULTIREL,BIN,NOLIST

$FDICT .PCH.

BINARY CARD (NOT PUNCHED)
000052000000 PCH FILE 'PCH

ENTRY .PCH.

BINARY CARD (NOT PUNCHED)
00000 00000 00401 PCH PZE PCH

$CDICT .PCH.

REFERENCE TO DEFINED SYMBOLS.

CLASS SYMBOL VALUE REFERENCES

$FILE .PCH 00000

LCTR //
BCREAD(A, B)

This routine causes absolute binary data cards as punched by the BCDUMP routine described previously to be read. Arguments A and B are first and last storage locations of the data being read. The routine is called for in the FORTRAN program section C110 and is as follows:

```
SRLDR .BCREA
STTXT .BCREA
ENTRY BCREAD .ACR0000
           .ACR0001
           11/1/65 JMLR

BINARY CARD (NOT PUNCHED)
000001 000000 0 00004 10001 BCREAD SAVE 1.4 11/1/65 JMLR
000002 0774 00 1 00000 10000
000003 0774 00 4 00000 10000
000004 0634 00 4 05000 10011
000005 0634 00 4 00056 10001
000006 0634 00 4 0002 10001
000007 0634 00 1 00001 10001
000010 0500 00 4 00003 10000 CLA 3.4 GET FIRST ARG. 11/1/65 JMLR
000011 0560 00 4 00004 10000 LDO 4.4 GET SECOND ARG. 11/1/65 JMLR
000012 0642 00 0 01002 10111 TLQ **2 COMPARE 11/1/65 JMLR
000013 0131 00 0 00000 10000 XCA IF 2ND LESS EXCHANGE 11/1/65 JMLR
000014 4600 00 0 00055 10001 STD TEMP STORE SMALLEST ARG 11/1/65 JMLR
000015 0400 00 0 06000 10011 ADD SYSIZE ADD 1 11/1/65 JMLR
000016 0621 00 0 00034 10001 STA STD STORE FOR MOVE 11/1/65 JMLR
000017 0467 00 0 00055 10001 SUB TFMP COMPUTE COUNT 11/1/65 JMLR
000020 0421 00 0 00030 10001 STA IXI STORE FOR MOVE 11/1/65 JMLR
000021 4774 00 4 00051 10001 AXC UNI-3.4 LOCATE UNOS LIKE FIV CALL 11/1/65 JMLR
000022 0634 00 4 05000 10011 SXA SYSLOG-4 AND SAVE IN SYSLOG 11/1/65 JMLR

BINARY CARD (NOT PUNCHED)
000023 0074 00 4 04000 10011 CALL **BCRD SET UP READ 11/1/65 JMLR
000024 1 00000 0 01002 10011
000025 0 00056 0 00016 10100
000026 0774 00 4 07000 10101 READ TSX **FI0C-4 READ RECORD 11/1/65 JMLR
000027 0774 00 4 10000 10101 TSX **FI0C-4 CHECK READ 11/1/65 JMLR
000030 0774 00 1 00000 10000 IXI AXT **1 PICK UP COUNT LEFT 11/1/65 JMLR
000031 7 00006 1 00042 10001 TIX LASTC-1.22 IS ONLY 1 REC LEFT 11/1/65 JMLR
000032 0776 00 4 00000 10000 IXK AXT 0.4 REC CNT 11/1/65 JMLR
000033 0500 00 4 11001 10001 CLA **BK87-4 MOVE WORDS 11/1/65 JMLR
000034 0601 00 1 00000 10000 STD STD **1 TO STORE 11/1/65 JMLR
000035 2 00001 1 01001 10101 TIX **1.1.1 DECR. COUNT 11/1/65 JMLR
000036 1 77777 4 01001 10101 TIX **1.1.1.1 DECR. REC CNT 11/1/65 JMLR
000037 7 77772 4 00033 10001 CKIR4 TIX STO-1.1.22 CR. REC CNT 11/1/65 JMLR
000040 0634 00 1 00030 10001 SXA IX1.1.1 NO SAVE REMAINING COUNT 11/1/65 JMLR
000041 0220 00 0 00026 10001 TRA READ GO READ NEXT RECORD 11/1/65 JMLR
000042 1 00000 1 00047 10001 LASTC TIX DONE-1.0 ANY MORE WORDS 04/26/66 TS
000043 0774 00 4 00047 10001 AXT DONN-4 YES STORE TO EXIT NEXT 11/1/65 JMLR
000044 0634 00 4 00041 10001 SXA LASTC-1.4 TIME 11/1/65 JMLR

BINARY CARD (NOT PUNCHED)
000045 4636 00 1 00037 10001 SCD CKIR4.1 SET REC CNT = NO W3RS LEFT 11/1/65 JMLR
000046 0020 00 0 0032 10001 TRA IX4 GO PROCESS RECORD 11/1/65 JMLR
000047 0774 00 4 0026 10001 DONE AXT READ-4 RESTORE EXIT 11/1/65 JMLR
000048 0634 00 4 00041 10001 SXA LASTC-1.4 11/1/65 JMLR
000051 0774 00 4 77752 10000 AXT -22.4 RESTORE RFC CNT 11/1/65 JMLR
000052 4634 00 4 00037 10001 SXD CKIR4.4 11/1/65 JMLR
000053 0053 RETURN BCREAD 11/1/65 JMLR
000054 0 00000 0 12000 10011 UNC PZE **UNOS ADD OF UNIT 5 11/1/65 JMLR
000055 0 00000 0 00000 10000 TEMP PZE 11/1/65 JMLR
000056 000000000000 10000 *DIR 11/1/65 JMLR
000057 1322391125 11 0000 01111 FND 11/1/65 JMLR
```
**BCREA DECK**  START=0, LENGTH=48, TYPE=7094, COMPLEX=5

```
R E F E R E N C E S

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKFAD</td>
<td>00000</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>CKIR4</td>
<td>00037</td>
<td>45-52</td>
<td></td>
</tr>
<tr>
<td>DONE</td>
<td>00047</td>
<td>42-43</td>
<td></td>
</tr>
<tr>
<td>IX1</td>
<td>00030</td>
<td>20-40</td>
<td></td>
</tr>
<tr>
<td>IX4</td>
<td>00032</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>..0001</td>
<td>00002</td>
<td>6-7</td>
<td></td>
</tr>
<tr>
<td>..0002</td>
<td>00003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>..0003</td>
<td>00004</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>IASTC</td>
<td>00042</td>
<td>31-44,50</td>
<td></td>
</tr>
<tr>
<td>RFAD</td>
<td>00026</td>
<td>41-47</td>
<td></td>
</tr>
</tbody>
</table>
```

**REFERENCES TO VIRTUAL SYMBOLS.**

```
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>VALUE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>..ACKD</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>..ARKD</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>..FIQC</td>
<td>7</td>
<td>26</td>
</tr>
<tr>
<td>..FTCK</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>..UNOS</td>
<td>10</td>
<td>54</td>
</tr>
<tr>
<td>SYSLOC</td>
<td>5</td>
<td>4-72</td>
</tr>
<tr>
<td>SYSINF</td>
<td>6</td>
<td>15</td>
</tr>
</tbody>
</table>
```
IALS(N, M)

This function shifts the fixed point variable M left N places in the accumulator. The function is used in the FORTRAN program sections C140 and C160.

IARS(N, M)

This function shifts the fixed point variable M right N places in the accumulator. The function is used in the FORTRAN program sections C30 (twice), C140, and C770. The two shift functions are as follows:

7094 RFLMOD ASSEMBLY.

ENTRY    IALS ENTRY    IARS

BINARY CARD (NOT Punched)
00000 0500 60 4 00003 10000 IALS CAL* 3.4
00001 0621 00 0 01007 10011 STA **2
00002 4500 60 4 00004 10000 CAL* 4.4
00003 0767 00 0 00000 10000 ALS **
00004 4130 00 0 00000 10000 XCL
00005 0111 00 0 00000 10000 XCA
00006 0220 00 4 00001 10000 TRA 1.4
00007 0500 60 4 00003 10000 IARS CAL* 3.4
00010 0621 00 0 01002 10011 STA **2
00011 4500 60 4 00004 10000 CAL* 4.4
00012 0771 00 0 00000 10000 ARS **
00013 4130 00 0 00000 10000 XCL
00014 0131 00 0 00000 10000 XCA
00015 0220 00 4 00001 10000 TRA 1.4
00000 01111 END

ENTRY    IALS ENTRY    IARS

BINARY CARD (NOT Punched)
00000 0500 60 4 00003 10000 IALS CAL* 3.4
00001 0621 00 0 01007 10011 STA **2
00002 4500 60 4 00004 10000 CAL* 4.4
00003 0767 00 0 00000 10000 ALS **
00004 4130 00 0 00000 10000 XCL
00005 0111 00 0 00000 10000 XCA
00006 0220 00 4 00001 10000 TRA 1.4
00007 0500 60 4 00003 10000 IARS CAL* 3.4
00010 0621 00 0 01002 10011 STA **2
00011 4500 60 4 00004 10000 CAL* 4.4
00012 0771 00 0 00000 10000 ARS **
00013 4130 00 0 00000 10000 XCL
00014 0131 00 0 00000 10000 XCA
00015 0220 00 4 00001 10000 TRA 1.4
00000 01111 END

$END .SHIFT

REFERENCE TO DEFINED SYMBOLS.

CLASS SYMBOL VALUE REFERENCES
IALS 000000
IARS 000007
LCTR BLETR
QUAL UNOS
LCTR //
APPENDIX D

DETAILS IN PREPARING INPUT

Uniform Format

Except for binary EF data cards, all input cards are read in with a uniform format, namely A6, 4(A6, F12.0), I2. The sections of the card will be referred to as follows:

<table>
<thead>
<tr>
<th>Card columns</th>
<th>Label 1</th>
<th>Numerical value 1</th>
<th>Label 2</th>
<th>Numerical value 2</th>
<th>Label 3</th>
<th>Numerical value 3</th>
<th>Label 4</th>
<th>Numerical value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 6</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
</tr>
<tr>
<td>7 to 12</td>
<td></td>
<td></td>
<td>25 to 30</td>
<td></td>
<td>43 to 48</td>
<td></td>
<td>49 to 60</td>
<td></td>
</tr>
<tr>
<td>13 to 24</td>
<td></td>
<td></td>
<td>31 to 42</td>
<td></td>
<td></td>
<td></td>
<td>61 to 66</td>
<td></td>
</tr>
<tr>
<td>25 to 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67 to 78</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 to 42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79 to 80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43 to 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49 to 60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61 to 66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67 to 78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79 to 80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
<td>A6</td>
<td>F12.0</td>
</tr>
<tr>
<td>79 to 80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The labels (label 1, label 2, . . . ) are codes on all types of input cards except one. (The exception, described in the section Data cards for FIXEDN, ALLN, or TEMPER methods, is the card containing spectroscopic data for atoms). These codes serve two purposes. One purpose is to specify an option in the program. For example, the label RRHO specifies a method of calculation. The second purpose is to identify the number which follows it. For example, the label R on the CONSTS card precedes the numerical value of the universal gas constant.

The last two columns (79 and 80) are used only with molecular constant data. For atomic gases, the principal quantum numbers are punched in these columns if needed with the method being specified. For diatomic and polyatomic gases, the electronic level identification is punched in these columns if excited states are included.

Some general rules in keypunching the input cards are given as follows:

1. With one exception, card columns 1 to 6 and labels are alphanumeric and must be left-adjusted. The exception is that the labels on the DATA cards which contain spectroscopic constants for monatomic gases are numbers and do not need to be left-adjusted. (See DATA cards.)

2. All blank labels are ignored by the program.

3. For the specific data, each numerical value must be immediately preceded by its label. However, the order of values is usually immaterial. Exceptions are noted in the details for the individual cards.

4. The numerical values may be the following:
   (a) A right-adjusted integer
   (b) A floating-point number without exponent (e.g., 0.00021), anywhere in the field

75
(c) A right-adjusted floating-point number with exponent indicating decimal place (e.g., 2.1-4 is $2.1 \times 10^{-4}$)

(5) The last two columns (79 and 80) are right-adjusted integers.

Order of Input Cards

Some discussion on the order of the input cards is given in the section General Flow of the Program. Specific instructions for placement of the individual cards are given in the details for making up the cards.

Ordinarily the general data cards should precede the specific data cards. However, general data cards may be inserted after the specific data for one or more species. The information on these cards, however, will be available only for the calculations called for by specific data which follow it. If a second CONST card, ATOM card for a particular atom, or set of EFDATA and binary EF data cards for a particular reactant is inserted, the data on the second card(s) are used for the succeeding calculations.

Otherwise, the general data may be read in any order as long as the EFDATA and binary EF data cards remain in an ordered set for each reactant, namely EFDATA card followed by binary EF data cards as numbered in card columns 79 to 80.

For a single computer run, there may be any number of species processed where each species requires its own set of specific data. The set of specific data cards for each species should be in the following order:

1. Formula card
2. Optional cards (REFNCE, EFTAPE, LOGK, LSTSQS, INTERM, or DATE) in any order
3. TEMP card(s), if any
4. METHOD card
5. DATA card(s)
6. FINISH card

There may be more than one set of these cards for a single species.

General Data Cards

Examples of the individual cards discussed in this section are given in appendix F. CONSTS card. - This card, which contains physical constants, is not optional. An example of the necessary labels and one possible set of numerical values is as follows:

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Value (ref. 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCK</td>
<td>$hc/k$, second radiation constant</td>
<td>1.4388</td>
</tr>
<tr>
<td>R</td>
<td>Universal gas constant</td>
<td>1.98726</td>
</tr>
<tr>
<td>SCONST</td>
<td>Entropy constant $S_c$ (see eqs. (4) and (5))</td>
<td>3.66511</td>
</tr>
</tbody>
</table>
A more recent set of physical constants is given in reference 35.

ATOM cards. - The order of the labels and information on each of these cards must be as follows:

<table>
<thead>
<tr>
<th>Card section</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns 1 to 6</td>
<td>ATOM</td>
</tr>
<tr>
<td>Label 1</td>
<td>Left-adjusted atomic symbol, for example, H, HE, LI</td>
</tr>
<tr>
<td>Numerical value 1</td>
<td>Atomic weight</td>
</tr>
<tr>
<td>Label 2</td>
<td>Left-adjusted formula of assigned reference element. The formula must give the atomic symbol, the number of atoms (even if just one), a left parenthesis, G or S depending on whether the elemental form is gas or solid, respectively, and finally a right parenthesis. Examples: P1(S), O2(G), Li1(S)</td>
</tr>
<tr>
<td>Numerical value 2</td>
<td>Coefficient, b, in equation (8)</td>
</tr>
<tr>
<td>Numerical value 3</td>
<td>Sum of the statistical weights ( \sum g_i ) (eq. (8)) for the ground electronic state</td>
</tr>
</tbody>
</table>

Numerical values 2 and 3 are needed only with the FILL option on the METHOD card for monatomic gases. These values were included for Mg(g) in example 5 in appendix F.

EFDATA and binary EF data cards. - A set of these cards contains enthalpy and free energy data for either a monatomic gas or an assigned reference element. The data will be put on FORTRAN tape number 3 and used for \( \Delta H^0_T \) and log \( K \) calculations. There may be any number of sets, or none, of these cards in the general data. These cards are not keypunched, but rather are part of the punched card output of a previous run. In order to obtain these punched cards, the previous run required an EFTAPE option card in the species input data for either an assigned reference element or a monatomic gas. For example, a set of these cards were punched in example 2, appendix F, for F2(g) and used as input in example 3.

LISTEF card. - The LISTEF card is optional and contains the card columns 1 to 6 code only. The data on any binary EF data cards which follow the LISTEF card will be listed. The binary EF data cards for each reactant must still be immediately preceded by an EFDATA card. (See example 5 in appendix F.)

Specific Input Cards

Examples of the individual cards discussed in this section are given in appendix F.

Formula card. - This card is the first card for each species and is reserved for two pieces of information. First, the species formula, as detailed below, is always
required. Second, either an assigned enthalpy or a heat of reaction value with the corresponding units and temperature is required only when calling for either of the following two options:

1. log $K$ and $\Delta H$ calculations, or
2. Least-squares fit of the thermodynamic functions

The first 12 columns are reserved for the formula of the species. Even when the formula takes less than 7 columns, columns 7 to 12 (label 1) should never be used for any code as is done on other types of cards. The formula should be left-adjusted and contain no blanks. It should be punched in the following order:

1. Each atomic symbol followed by the number of atoms even if the number is 1; these atomic symbols should correspond to the symbols on the ATOM card in the general input
2. For ionic species, the proper number of pluses or minuses should be punched
3. A left parenthesis
4. A G for a gas, an L for a liquid, or an S for solid
5. A right parenthesis

The following are examples for ionized species:

<table>
<thead>
<tr>
<th>Species</th>
<th>Columns 1 to 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F^-$</td>
<td>F1-(G)</td>
</tr>
<tr>
<td>$N^+$</td>
<td>N1+(G)</td>
</tr>
<tr>
<td>$O^{++}$</td>
<td>O1++(G)</td>
</tr>
<tr>
<td>$O^-$</td>
<td>O2-(G)</td>
</tr>
</tbody>
</table>

The remainder of the card is reserved for a heat of reaction, the energy units for the reaction, and the temperature of the reaction. There are five forms in which the heat of reaction may be expressed and five choices of units. These are summarized in table IV.

REFERENCE card. - The only purpose of this card is to identify sources of input data. The labels and numerical values are arbitrary. (See MgO(g) in example 5, appendix F.)

EFTAPE card. - This option card is used either with an atomic gas or an assigned reference element whose data are needed for succeeding $\Delta H^O_T$ and log $K$ calculations. The card has only the letters EFTAPE punched in card columns 1 to 6. Inclusion of the card causes the $H^O_0$ and the $(H^O_T - H^O_0)/RT$ and $-(F^O_T - H^O_0)/RT$ data for this species to be (1) put on the end of FORTRAN tape 3 where they are available for use with succeeding calculations during the same computer run and (2) punched on cards to be included with other general data during future computer runs. (See example 2 and Mg(g) in example 5, appendix F.)

LOGK card. - Inclusion of this option card causes tables of thermodynamic prop-
erties including log K and ΔH to be listed. It simply has the code LOGK punched in card columns 1 to 4. The log K and ΔH calculations will be for reactions involving either the assigned reference elements or the monatomic gases or both depending on what data are available on FORTRAN tape 3. If there is no matching temperature in the appropriate atomic gas or assigned reference element data on FORTRAN tape 3, the data that are there will be interpolated by three-point Lagrangian interpolation. (See example 3 and Mg(s) in example 5, appendix F.)

LSTSQS card(s). - Inclusion of one or more of these cards calls for a least-squares fit of the functions to equations (10) to (12) as discussed in the section on Least-squares fit. The LSTSQS card(s) may contain three possible labels, T, EXP, and TCONST, and their corresponding values. In table V, desired temperature intervals are given by using T labels, and exponents (q_i in eqs. (10) to (12)) are given by using EXP labels. The fit will be constrained in two ways, (1) to fit the data at one temperature which must be an endpoint of an interval (TCONST label) and (2) to give equal values of the thermodynamic functions at common endpoints of the intervals. The numerical values associated with the T and TCONST codes must be equal to some temperature in the temperature schedule for the thermodynamic functions. (See Mg(g) in example 5, appendix F.)

If any of the three possible labels are omitted on the LSTSQS card(s), values will be assigned by the program. If no TCONST is given, it will be assigned to be either 1000°K or, if a phase transition takes place, the temperature of transition (each phase will be fitted separately). If no EXP is given, the q_i values will be assigned to be 0, 1, 2, 3, and 4. If no T's are given, the temperature intervals will be assigned to be 300° to 1000°K and 1000° to 5000°K. (See example 4, appendix F.)

INTERM card. - This card calls for intermediate output to be printed out when thermodynamic functions are being calculated from molecular constants. (See H_2O(g) in example 5, appendix F.)

DATE card. - The purpose of the DATE card is to punch a date or code on the binary least squares coefficient cards. The card should contain only one label which will be punched as the last word on the last least squares coefficient card punched for each species. (See Mg(g) in example 5, appendix F.)

TEMP card(s). - These cards give a temperature schedule for which thermodynamic functions are to be calculated. The program allows for a maximum of 201 temperatures per species.

Each temperature in the desired temperature schedule may be specified individually with a T label. (See table V.) However, if there are several temperatures incremented by a fixed amount, this part of the temperature schedule may be specified by punching in order: the lowest temperature labeled T, the increment labeled I, and the highest temperature labeled T. For example, the temperature schedule, 100, 200,
298.15, 300, 400, 500, 600, 688.2, 700, 750, 800, 850, 900, 962.3, and 1000, could be keypunched as follows:

<table>
<thead>
<tr>
<th>Card columns 1 to 6</th>
<th>Label 1</th>
<th>Numerical value 1</th>
<th>Label 2</th>
<th>Numerical value 2</th>
<th>Label 3</th>
<th>Numerical value 3</th>
<th>Label 4</th>
<th>Numerical value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP T</td>
<td>100.</td>
<td>I</td>
<td>100.</td>
<td>T</td>
<td>600.</td>
<td>T</td>
<td>688.2</td>
<td></td>
</tr>
<tr>
<td>TEMP T</td>
<td>700.</td>
<td>I</td>
<td>50.</td>
<td>T</td>
<td>900.</td>
<td>T</td>
<td>962.3</td>
<td></td>
</tr>
<tr>
<td>TEMP T</td>
<td>1000.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The temperature, 298.15°K, is always inserted in the temperature schedule when there are temperature values below and above 298.15°K. (See examples 1 and 3, and Ar(g), H2O(g), Mg(s), and MgO(g) in example 5, appendix F.)

If there are no TEMP cards in a set of data where the thermodynamic functions are to be calculated, the program (section C40) assumes the standard temperature schedule used in reference 3, namely, every 100° from 100° to 6000° K with 298.15° inserted between 200° and 300° K. (See example 2 and Mg(g) in example 5, appendix F.)

The only option for which TEMP cards must not be used is READIN (see METHOD card). For this option, the temperatures are read in on DATA cards together with the thermodynamic functions to which they correspond. (See example 4 and Mg(s) in example 5, appendix F.)

METHOD card. - This card follows the option cards and must be included for any calculations to take place. It specifies the technique for obtaining the thermodynamic functions (see section Options) and immediately precedes the data required by the method (DATA cards). The card has the code word METHOD in card columns 1 to 6. The possible codes in the label and numerical value columns are summarized in table VI. The functions may be (1) calculated from molecular constants for ideal gases (labels FIXEDN, ALLN, or TEMPER for monatomic molecules and labels RRHO, PANDK, JANAF, NRRHO1, or NRRHO2 for diatomic and polyatomic molecules) (see examples 1 to 3 and H2O(g), Mg(g), and MgO(g) in example 5, appendix F), (2) calculated from coefficients and exponents using equations (10) to (12) (label COEF), (see Ar(g) and Mg(g) in example 5, appendix F), or (3) read in directly (label READIN) (see example 4 and Mg(s) in example 5, appendix F). These calculation techniques are discussed in the section CALCULATION OF IDEAL GAS THERMODYNAMIC FUNCTIONS.

In conjunction with these method codes, the METHOD card may contain some additional codes and information as indicated in table VI.

Occasionally, a single method may not apply to the entire desired temperature range for a species. In this case the following cards must be included for each temperature
interval, in order: (1) TEMP card(s) for the desired temperature interval (if the method is not READIN), (2) a METHOD card for this temperature interval, and (3) the associated DATA cards. The sets should be in order of increasing temperature. (See Mg(s, l) in example 5, appendix F.)

**DATA cards.** - These cards follow the METHOD card and contain the input data required by the method. Except for the spectroscopic data of monatomic gases (see example 2 and Mg(g) in example 5, appendix F), the labels are codes identifying the numerical values that follow them. Table VII is a summary of the labels and numerical values to be used on DATA cards for the various methods given in table VI. A further description of the DATA cards for the various methods follows:

**DATA cards for READIN method:** Each card must contain four labels with the four corresponding numerical values as indicated in table VII. The four labels correspond to temperature, heat capacity, enthalpy, and either entropy or free energy. Temperature always has the label T; however, the other three have several options as given in table VII depending on the data to which they correspond. If enthalpy and free energy are referred to $H_{298.15}^0$ rather than $H_0^0$, the $H_{298.15}^0 - H_0^0$ value must be included on the METHOD card (label H298H0) if $H_T^0 - H_0^0$ values are desired in the final tables. (See example 4 and Mg(s) in example 5, appendix F.)

**DATA cards for COEF method:** The coefficient and exponent values for each set of empirical equations (eqs. (10) to (12)) must be preceded by the values of the temperature limits (T labels in table VII) for which the equation applies (see Ar(g) and Mg(s) in example 5, appendix F). The lower T value must be the first numerical value.

Occasionally, the coefficients $a_i$ ($i = 1, r$) are available while the integration constants for enthalpy and entropy $a_{r+1}$ and $a_{r+2}$ are not. For this case, $a_{r+1}$ and $a_{r+2}$ values may be calculated by the program in one of the following ways:

1. Reading in an enthalpy and an entropy or free energy value with the corresponding temperature on the first card. The labels and values should be the same as for the DATA cards for the READIN method except that $C_p^0$ or $C_p^0/R$ may be omitted.

2. Using the value of enthalpy or entropy of transition (DELTAH or DELTAS on the METHOD card (see table VI)). This method may be used only when the two phases related by the transition value are being processed in the same run. The reason is that the transition value is combined with the enthalpy or entropy value for the last temperature of the preceding phase. (See Mg(s, l) in example 5, appendix F.)

With COEF method there is an option to punch these coefficients on binary cards in the form required for use with the IBM program described in reference 33. (With LSTSQS option, similar binary cards are always made and are not optional.) For each set of coefficients the temperature intervals to be punched are indicated with TPUNCH
labels and corresponding values which give the endpoints of the intervals. These TPUNCH values may or may not be the same as the T values for the set. For reference 33, coefficients for two temperature intervals are required. In the event there is only one set of coefficients available, the same set can be used in two intervals by using three TPUNCH values. (See Ar(g) in example 5, appendix F.)

DATA cards for FIXEDN, ALLN, or TEMPER methods: In contrast to all other types of cards using the universal format, these cards use the label columns as well as the numerical columns for numbers. The labels contain the total angular momentum quantum number \( J_m \) (eq. (7), and the numerical values contain the excitation energy \( \epsilon_m/\hbar c \) (eq. (7)) in centimeters\(^{-1}\). For either the FILL option or the FIXEDN method, the principal quantum numbers must be punched in card columns 79 to 80, right-adjusted. The data on the remaining portion of the card must correspond to that principal quantum number.

DATA cards for RRHO, PANDK, JANAF, NRRHO1, or NRRHO2 methods: The equations for the partition function of the various methods are given in tables I and II. The input data must always contain at least the following quantities for each electronic state:

1. The fundamental vibrational frequencies of the molecule \((\omega_e \text{ or } \nu_1)\)
2. Either the rotational constant(s) \((B_0 \text{ for linear; } A_0, B_0, \text{ and } C_0 \text{ for nonlinear molecules}) \) or the moment(s) of inertia \((I_B \text{ for linear; } I_A, I_B, \text{ and } I_C \text{ for nonlinear molecules}) \)
3. The symmetry number
4. The statistical weight

Other spectroscopic constants such as anharmonicity or rotation-vibration interaction constants are optional. If these optional constants are not included, correction terms involving them are automatically excluded from the partition function. (See example 1 (RRHO), example 3 (PANDK), and \( H_2O(g) \) (NRRAO2) and \( MgO(g) \) (PANDK) in example 5, appendix F.)

When excited electronic states are involved, the data for each state are read and processed separately. Therefore, the data cards must be grouped together with an identifying number in card columns 79 to 80. For example, the data for the three electronic states included for \( MgO(g) \) in example 5, appendix F, are distinguished by 1, 2, or 3 punches in card column 80.

FINISH card. - This card is the last card in the specific input cards for each species. It contains only the code in card columns 1 to 6.
APPENDIX E

DETAILS IN OUTPUT

Punched Cards

Certain options in the specific data cause cards to be punched. A description of these punched cards follows.

**EFDATA and binary EF data.** - A set of EFDATA and binary EF data cards is punched when an EFTAPE card has been included in the specific data for either an assigned reference element or a monatomic gas.

The first card is the EFDATA card and is punched in the uniform format. It contains the formula, the $H_0^O$ value, the melting point, and the number of temperatures for which enthalpy and free energy data are available on succeeding binary cards.

The remaining cards are binary EF data cards and are punched in column binary. Each binary card contains the chemical formula and seven temperatures with corresponding $(H_T^O - H_0^O)/RT$ and $-(F_T^O - H_0^O)/RT$ values (except possibly the last card which may have seven or less).

These cards are punched so that they may be used as general input for subsequent computer runs and be available for $\Delta H_T^O$ and log $K$ calculations. (See the general input in examples 3 and 5 in appendix F.)

**Coefficients for empirical equations.** - Coefficients for empirical equations (eqs. (10) to (12)) will be punched on column binary cards if one of the following is true:

1. A LSTSQS option card is included in the specific data for a particular species. The coefficients are obtained from a least-squares fit of the functions.
2. Predetermined coefficients are read in directly (method COEF), and TPUNCH codes are on the DATA cards.

The format used for punching these coefficient cards was selected to be consistent with that used in reference 33. The following information is punched on these cards:

1. The formula of the species as given on the formula card
2. The ionization potential if there is one
3. The entire temperature range
4. The temperature ranges of the intervals
5. Seven coefficients ($a_1$, . . . , $a_5$ in eq. (10), $a_{r+1}$ in eq. (11), and $a_{r+2}$ in eq. (12)); if there are fewer than 5 coefficients in equation (10), zeros will be inserted but if there are more than 5 coefficients, only the first 5 will be punched

Note the exponents ($q_1$ in eqs. (10) to (12)) are not punched. However, they are listed together with the $a_i$ following the intermediate data associated with the least-squares fit for each temperature interval. (See output listings for example 4 and Mg(g)
in example 5, appendix F.) For reference 33, \( q_1 = 0, q_2 = 1, \ldots, q_5 = 4 \).

There are 24 binary words on a card. Table VIII shows the contents of these cards for up to the nine interval limit. Only as much data are punched as required by the temperature intervals. The temperature intervals are the values following \( T \) labels on the LSTSQS cards or TPUNCH labels on the DATA cards following a METHOD COEF card. When no \( T \) labels are punched on the LSTSQS card(s), two intervals are assumed, \( T = 300^\circ \) to \( T = 1000^\circ \) K and \( T = 1000^\circ \) to \( T = 5000^\circ \) K. The contents of these cards are listed as they are punched. (See output listings for examples 4 and Ar(g) and Mg(g) in example 5, appendix F.)

Listed Output

Input data cards in the uniform format as well as tables of thermodynamic functions resulting from each set of specific data are always listed. Other data will be listed with certain options.

Input data. - All input data cards in the uniform format are listed immediately after they are read. The output format is similar to the uniform input format with spacing between the labels and values. Numerical values which are zero are left blank. (See examples in appendix F.)

The data on the binary EF data cards which are read in as input will be listed only when an LISTEF card precedes the EFDATA card somewhere in the deck. (See the general input and output data in example 5, appendix F; only the \( \text{O}_2 \) data are listed.)

Punched card output. - The contents of all cards punched by the program will be listed except for the binary EF data cards. For this latter case, the punched data will be listed only when a LISTEF card precedes the specific data somewhere in the deck. In example 3, appendix F, the punched binary EF data are not listed while for \( \text{Mg}(g) \) in example 5, they are. This is because of the LISTEF card in the general data of example 5.

Tables of thermodynamic properties. - Two tables of thermodynamic functions are always listed with each set of specific data. These tables are the following:

1. Table of dimensionless properties as follows:
   
   \( \begin{align*}
   T, & \quad C_p^O/R, \quad (H_T^O - H_0^O)/RT, \quad (H_T^O - H_{298.15}^O)/RT \quad \text{(if } T = 298.15 \text{ is in } T \text{ range}), \\
   S_T^O/R, & \quad -(F_T^O - H_0^O)/RT, \quad -(F_T^O - H_{298.15}^O)/RT \quad \text{(if } T = 298.15 \text{ is in } T \text{ range}), \\
   H_T^O/RT, & \quad \text{(if an } H_0^O \text{ value is available), and } -F_T^O/RT \quad \text{(if an } H_0^O \text{ value is available)}
   \end{align*} \)

2. Table of dimensioned properties as follows:

84
T, C_p, H^o_T - H_0^o, H^o_T - H^o_{298.15} (if T = 298.15 is in T range), S^o_T, -(F^o_T - H^o_0),
-(F^o_T - H^o_{298.15}) (if T = 298.15 is in T range), H^o_T (if an H^o_0 value is available) and -F^o_T (if an H^o_0 value is available)

See output for the examples in appendix F.

When a LOG K option card is included in a set of specific data, two additional tables are listed for that particular species. (See example 3 and Mg(s) and MgO(g) in example 5, appendix F.) These two tables are the following:

(1) Table of dimensionless properties as follows:
T, C_p/R, (H^o_T - H_0^o)/RT, S^o/R, -(F^o_T - H^o_0)/RT, H^o_T/RT, F^o_T/RT, and
ΔH^o_T/RT and -ΔF^o_T/RT for reactions from the assigned reference elements,
and ΔH^o_T/RT and -ΔF^o_T/RT for reactions from the monatomic gases

(2) Table of dimensional properties as follows:
T, C_p, H^o_T - H_0^o, S^o_T, -(F^o_T - H^o_0), H^o_T, and ΔH^o_T and log_10 K for formation
from the assigned reference elements, and ΔH^o_T and log_10 K for formation
from monatomic gases

These tables will have an asterisk and a footnote indicating where a melting point has occurred in an assigned reference element. (See MgO(g) in example 5, appendix F.)

Least squares polynomial and errors. - A least-squares fit of the functions, C_p/R, (H^o_T - H^o_0)/RT, and S^o_T/R, results when LSTSQ card is included in a set of specific data. (See example 4 and Mg(g) in example 5, appendix F.)

For each temperature interval, the following information is listed:

(1) For each T within the interval,
(a) C_p/R, (H^o_T - H^o_0)/RT, and -(F^o_T - H^o_0)/RT
(b) Functions in (1a) as calculated from least-square coefficients and equations (10) to (12)
(c) Differences in (1a and b); these values are referred to as errors hereinafter
(d) Values in (1c) divided by original values in (1a); these values are referred to as relative errors hereinafter

(2) For errors in entire interval for each function in (1a):
(a) Maximum relative error (MAX REL ERR) and corresponding temperature - see (1d)
(b) Average relative error (AVER REL ERR) - see (1d)
(c) Root mean square of relative errors (REL LST SQ ERR) - see (1d)
(d) Maximum error (MAX ERR) and corresponding temperature - see (1c)
(e) Average error (AVER ERR) - see (1c)
(f) Root mean square of errors (LST SQ ERR) - see (1c)
(g) $C_p^0/R$ equation (see eq. (10)) for coefficients $a_i/R$

(h) Integration constants in equations (11) and (12) as follows:

$$
\frac{(H - H_0)}{R} \text{CONSTANT} = \frac{(a_{r+1} - H_0)}{R} \\
\frac{H}{R(A6)} \text{CONSTANT} = \frac{a_{r+1}}{R} \\
\frac{S}{R} \text{CONSTANT} = \frac{a_{r+2}}{R}
$$

Finally, the contents of the punched binary coefficient cards are listed. See the section on Punched card output.

**Intermediate data with FILL option for monatomic gases.** - Unobserved but predicted energy levels for monatomic gases will be included in the partition function (eq. (7)) if the FILL code is punched on the METHOD card. See the section on Inclusion of predicted levels for the method of predicting the levels.

In Mg(g) in example 5, appendix F, the following data are listed in columns from left to right:

1. $b$ value from ATOM card (see eq. (8))
2. Principal quantum number $n$
3. $bn^2$ [predicted $\sum (2J + 1)$]
4. $\sum (2J + 1)$ from input data
5. Column (3) minus column (4)
6. Highest energy level for principal quantum number
7. Sum of column (5) and $2J + 1$ for level of column (6)

**Intermediate data with INTERM card.** - Intermediate data are listed for ideal gas calculations if an INTERM card is included in the specific data for a particular species.

**Monatomic gases:** For monatomic gases several items are listed. The input data are listed in order of increasing energy level values. The data include, from left to right, values for the principal quantum number, $J$, $2J + 1$, and the energy level.

For each temperature, three lines of data are listed as follows:

1. A statement indicating where the energy levels were cut off; five possible statements are the following:
   
   (a) **NOT ALL LEVELS WERE USED. X IS GREATER THAN 85.** - This statement indicates that not all atomic energy levels were used because $\epsilon/kT > 85$ in equation (7).
   
   (b) **ALL LEVELS USED THROUGH $N = (\text{FIXEDN value})** - This statement indicates all atomic levels were used through a fixed principal quantum number (method FIXEDN).
   
   (c) **ALL ASSIGNED LEVELS HAVE BEEN USED** - This statement indicates all
atomic levels in input were used (method ALLN)

(d) NOT ALL ASSIGNED LEVELS WERE USED, Q AND DERIVATIVES ARE TOO SMALL - This statement indicates not all atomic levels were used because the following conditions occurred:

\[ Q^m \leq 1 \times 10^{-10} \]

and

\[ (\epsilon_m / kT)^2 Q^m \leq 1 \times 10^{-10} \]

when \( \epsilon_m / kT > 2 \).

(e) ALL LEVELS HAVE BEEN USED TO THE THERMAL BINDING ENERGY (lowered IP value). - This statement gives the lowered ionization potential value (i.e., ionization potential - Tk/hc) where levels have been cut off.

(2) Values of \( T, C_p^0 / R, (H^0_T - H^0_0) / RT, \) and \( -(F^0_T - H^0_0) / RT \)

(3) Values of \( \epsilon / kT, Q, T \, dQ / dT, \, T^2 \, d^2Q / dT^2 + 2T \, dQ / dT \) (eq. (7))

Diatomic and polyatomic gases: Intermediate results are listed when an INTERM card is included with the specific input data cards for a diatomic or polyatomic gas and the method of calculation is RRHO, JANAF, PANDK, NRRAO1, or NRRAO2. These results include values for the formulas and variables defined in tables I and II. Although the molecular constants are always listed as they are punched on the DATA cards with an INTERM card, many of them are listed again.

The following data are listed (see tables I and II for definitions and \( H_2O(g) \) in example 5, appendix F):

(1) \( A_0, B_0, C_0, \rho \)

(2) \( a_i, \alpha_i^A, \alpha_i^B, \alpha_i^C \) where \( i = 1 \) to the number of unique frequencies

(3) \( \theta_1, \theta_2, \theta_3 \)

(4) \( y_{ijk} \)

(5) \( x_{ij} \)

(6) \( \text{LEVEL} = \) (value in card columns 79 to 80 which is used to identify the electronic levels)

(7) \( \nu_i, d_i, g_{ii} \)

(8) \( T \)

(9) \( u_i, r_i, s_i, i \)

(10) As required by the method of calculation, values for the formulas in tables I and II are listed for \( Q, \ln Q, \, T \, d(\ln Q) / dT, \) and \( T^2 \, d^2(\ln Q) / dT^2 + \)
2T d(ln Q)/dT. The latter three values are additive contributions to
-(F^O_T - H^O_0)/RT, (H^O_T - H^O_0)/RT, and C^O_P/R, respectively, when only the ground
electronic state is considered. These values are identified in the listing by
codes which correspond to the formula numbers as follows:

<table>
<thead>
<tr>
<th>Code on listing</th>
<th>Formula numbers in tables I and II</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTR</td>
<td>1</td>
</tr>
<tr>
<td>H.O.</td>
<td>2</td>
</tr>
<tr>
<td>R.R.</td>
<td>3 or 4</td>
</tr>
<tr>
<td>RHO</td>
<td>5</td>
</tr>
<tr>
<td>THETA</td>
<td>6</td>
</tr>
<tr>
<td>FERMI</td>
<td>7</td>
</tr>
<tr>
<td>ALPHA</td>
<td>8 to 11</td>
</tr>
<tr>
<td>XIJ</td>
<td>12 or 14</td>
</tr>
<tr>
<td>YIJK</td>
<td>13</td>
</tr>
<tr>
<td>G+AG</td>
<td>16</td>
</tr>
<tr>
<td>WEZE</td>
<td>15</td>
</tr>
<tr>
<td>AXIJ</td>
<td>17</td>
</tr>
<tr>
<td>(XIJ)2</td>
<td>18 and 19</td>
</tr>
<tr>
<td>XY</td>
<td>20 and 21</td>
</tr>
<tr>
<td>G2, GX</td>
<td>22 and 23</td>
</tr>
<tr>
<td>AX2</td>
<td>24 to 27</td>
</tr>
</tbody>
</table>
APPENDIX F
EXAMPLES

The punched card input and listed output are given for several sample problems. The first four examples are simple problems with minimal input. Each of these examples is for a particular species with only as much general data shown as required. These four problems may be run individually, or they may be run together in a single computer run. For this latter case, the general data may be combined. A listing of input cards is given with the combined general data of these first four species.

The fifth example includes specific input cards for five species and general input cards which accommodate a much larger variety of problems. It has, for example, ATOM cards for the first 20 elements. Such a set of ATOM cards may be considered a permanent part of the operating deck.

All input data are the same as the data used in reference 3. Format details for keypunching input cards are described in appendix D.

Example 1 (MgF$_2$(g) with RRHO Method)

Problem. - Calculate the thermodynamic functions for MgF$_2$(gas) from 100$^0$ to 500$^0$ K at 100$^0$ intervals assuming a rigid-rotator harmonic-oscillator approximation and using the following data:

1. Physical constants: $h\nu/k = 1.4388$ (centimeters)(degrees), $R = 1.98726$ calories per mole per $^0$K, and $S_c = -3.66511$ calories per mole per $^0$K
2. Atomic weights: $F = 19.00$ grams per mole and $Mg = 24.32$ grams per mole
3. Molecular constants: statistical weight = 1, $\nu_1 = 540$ centimeter$^{-1}$, $\nu_2 = 500(2)$ centimeter$^{-1}$, $\nu_3 = 820$ centimeter$^{-1}$, $I_B = 19.77 \times 10^{-39}$ (grams)(centimeters$^2$), and symmetry number = 2

Punched card input. - The punched card input is as follows:

<table>
<thead>
<tr>
<th>Card columns 1 to 6</th>
<th>Label</th>
<th>Numerical value 1</th>
<th>Label</th>
<th>Numerical value 2</th>
<th>Label</th>
<th>Numerical value 3</th>
<th>Label</th>
<th>Numerical value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTS</td>
<td>F</td>
<td>19.0000</td>
<td>HCK</td>
<td>1.4388</td>
<td>R</td>
<td>1.98726</td>
<td>SCONST</td>
<td>-3.66511</td>
</tr>
<tr>
<td>ATOM</td>
<td>F2(G)</td>
<td></td>
<td>MGL2(G)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM</td>
<td>Mg</td>
<td>24.3200</td>
<td>MGL1(S)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>T</td>
<td>100.</td>
<td>I</td>
<td>100.</td>
<td>T</td>
<td>500.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td>RRHO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINISH</td>
<td></td>
<td></td>
<td>SYMNO</td>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Listed output. - The listed output is as follows:

<table>
<thead>
<tr>
<th>ATOM</th>
<th>F 19. F2(G)</th>
<th>1.4388000</th>
<th>1.9872600</th>
<th>-3.6651100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOM</td>
<td>MG 24.3199999</td>
<td>MG1(S)</td>
<td>62.320000</td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>T 100. I 100.</td>
<td>T 500.</td>
<td>MG1F2(G)</td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td>RRHO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOLECULAR WT</td>
<td>62.320000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td></td>
<td>18</td>
<td>19.7700000</td>
<td>SYMNO</td>
</tr>
<tr>
<td>FINISH</td>
<td>MG1F2(G)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MG1F2(G)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO HZERO VALUE IS AVAILABLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>CP/R</th>
<th>(H-HO)/RT</th>
<th>(H-H298)/RT</th>
<th>S/R</th>
<th>-(F-HO)/RT</th>
<th>-(F-H298)/RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>3.6044264</td>
<td>3.5141867</td>
<td>-9.1305719</td>
<td>23.0655146</td>
<td>19.5513279</td>
<td>32.2512326</td>
</tr>
<tr>
<td>200</td>
<td>4.6699711</td>
<td>3.6005604</td>
<td>-2.5499019</td>
<td>25.8549500</td>
<td>22.0549996</td>
<td>28.4046158</td>
</tr>
<tr>
<td>298.15</td>
<td>5.6532787</td>
<td>4.2595689</td>
<td>0.0000001</td>
<td>27.9111222</td>
<td>23.6565433</td>
<td>27.9161119</td>
</tr>
<tr>
<td>300</td>
<td>5.6683680</td>
<td>4.2682114</td>
<td>0.0340988</td>
<td>27.9511299</td>
<td>23.6829185</td>
<td>27.9162199</td>
</tr>
<tr>
<td>400</td>
<td>6.2824038</td>
<td>4.7015682</td>
<td>1.5268021</td>
<td>29.6735663</td>
<td>24.9719081</td>
<td>28.1468842</td>
</tr>
<tr>
<td>500</td>
<td>6.6502306</td>
<td>5.0576122</td>
<td>2.5176313</td>
<td>31.1183319</td>
<td>26.0607197</td>
<td>28.6007006</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>H-HO</th>
<th>H-H298</th>
<th>S</th>
<th>-(F-HO)</th>
<th>-(F-H298)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>7.1629324</td>
<td>698.3603</td>
<td>-1825.4410</td>
<td>45.837174</td>
<td>3885.3572</td>
<td>6409.1584</td>
</tr>
<tr>
<td>200</td>
<td>9.7785994</td>
<td>1104.4966</td>
<td>-1813.3046</td>
<td>51.385076</td>
<td>8765.6049</td>
<td>11289.4061</td>
</tr>
<tr>
<td>298.15</td>
<td>11.2354270</td>
<td>2523.4013</td>
<td>0.000000</td>
<td>55.476573</td>
<td>14016.3389</td>
<td>16540.3401</td>
</tr>
<tr>
<td>300</td>
<td>11.2645209</td>
<td>2544.6137</td>
<td>20.8125</td>
<td>55.546162</td>
<td>14119.2349</td>
<td>16643.0359</td>
</tr>
<tr>
<td>400</td>
<td>12.4476973</td>
<td>3737.3669</td>
<td>2123.5657</td>
<td>58.969091</td>
<td>19850.2695</td>
<td>23737.0708</td>
</tr>
<tr>
<td>500</td>
<td>13.2157372</td>
<td>5025.3952</td>
<td>2501.5940</td>
<td>61.840216</td>
<td>25894.7126</td>
<td>28418.5139</td>
</tr>
</tbody>
</table>

Example 2 (F2(g) with PANDK Method and EFTAPE Option)

**Problem.** - Calculate the thermodynamic functions for the reference element, F2(g), where the standard temperature schedule is assumed. Prepare a set of EFDATA and binary EF data cards for future ΔH^0_T and log K calculations. Use the PANDK method and the following data:

1. Physical constants and atomic weights: Same as for example 1
2. Heat of formation: ΔH^0_f(298, 15°K) = 0
3. Molecular constants: statistical weight = 1, ω_e = 923 centimeter^−1, ω_e x_e = 15.6 centimeter^−1, B_e = 0.8909 centimeter^−1, α_e = 0.0162 centimeter^−1, and symmetry number = 2

**Punched card input.** - The punched card input is as follows:
Listed output. The listed output is as follows:

<table>
<thead>
<tr>
<th>Card columns</th>
<th>Numerical value 1</th>
<th>Numerical value 2</th>
<th>Numerical value 3</th>
<th>Numerical value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM F</td>
<td>19. F2(G)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2(G)</td>
<td></td>
<td>HF298</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFTAPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD PANOK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA WE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA SYMNO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINISH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
LISTED OUTPUT

CONSTS
ATOM F
F2(G) HF298
METHOD PANOK
MOLECULAR WT. = 38.000000
DATA STATWT 1. WE 923. MEXE 15.600000 RE 0.89090000
DATA ALPHA 0.01620000 SYMNO 2.
FINISH F2(G)
```
Example 3 (F(g) with LOGK Option)

Problem. - In addition to calculating thermodynamic functions, calculate the heat of formation and equilibrium constant values for F(g) from F_2(g) for the temperatures, 298.15°, 1000°, 2156°, 3000°, and 5000° K. Use the enthalpy and free energy values for F_2 calculated in example 2 (i.e., the EFDATA and binary EF data cards for F_2). For F(g), use the following data:

1. Physical constants and atomic weight: same as example 1
2. Heat of formation: \( \Delta H_f^{\circ}(298.15) = 18,858.2 \) calories per mole
3. Spectroscopic data: \( J_1 = 3/2, \epsilon_1 = 0 \) and \( J_2 = 1/2, \epsilon_2/\hbar c = 404.1 \) centimeter\(^{-1}\)

Punched card input. - The punched card input is as follows:

```plaintext
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>col-1 to 6</td>
<td>value 1</td>
<td>value 2</td>
<td>value 3</td>
<td>value 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTS</td>
<td>ATOM</td>
<td>HCK</td>
<td>1.4388</td>
<td>R</td>
<td>1.98726</td>
<td>SCONST</td>
<td>-3.66511</td>
</tr>
<tr>
<td>EFDATA</td>
<td>F2(G)</td>
<td>HZERO</td>
<td>-2109.6975</td>
<td>MELTPT</td>
<td>0.</td>
<td>T NO.</td>
<td>61.0000</td>
</tr>
<tr>
<td>19.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Binary EF data for F_2(g)

```plaintext
<table>
<thead>
<tr>
<th>Card</th>
<th>Label</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>col-79 to 80</td>
<td>value 1</td>
<td></td>
</tr>
<tr>
<td>LOGK</td>
<td>T</td>
<td>298.15</td>
</tr>
<tr>
<td>TEMP</td>
<td>T</td>
<td>5000.</td>
</tr>
<tr>
<td>METHOD</td>
<td>ALLN</td>
<td>1.5</td>
</tr>
<tr>
<td>DATA</td>
<td>FINISH</td>
<td></td>
</tr>
<tr>
<td>18858.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>404.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Listed output. - The listed output is as follows:

```plaintext
| CONSTS | HCK | 1.43880000 | R | 1.98726000 | SCONST | -3.6651100 |
| ATOM | F | 19. | F2(G) |
| FFDATA | F2(G) | HZERO | -2109.69751 | MELTPT | T NO. | 61. |
| FI1(G) | HF298 | 18858.200 |
| LOGK | TEMP | T | 298.150000 | T | 1000. | T | 2156. | T | 3000. |
| TEMP | T | 5000. |
| METHOD | ALLN |
| DATA | 1.5 |
| FINISH |
| 0.5 |
| 404.10000 |
| FI1(G) |
| HZERO | 17300.217 |
```

94
Problem. - Use the data for P(solid) given in reference 3 to calculate the least-squares coefficients and punch them on cards as required for use with the program described in reference 33. Use functional form given in equations (10) to (12) with \( q_i = 1, 2, 3, 4, \) and 5. The data are normally fitted in two temperature intervals, 300° to 1000° K and 1000° to 5000° K. However, since P(solid) melts at 317.3° K, there will be only one set of coefficients for this case.

Punched card input. - The punched card input is as follows:

<table>
<thead>
<tr>
<th>Card column</th>
<th>Label</th>
<th>Numerical value 1</th>
<th>Label</th>
<th>Numerical value 2</th>
<th>Label</th>
<th>Numerical value 3</th>
<th>Label</th>
<th>Numerical value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI(S)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHODadin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA T</td>
<td>298.15</td>
<td>CP</td>
<td>2.7356769</td>
<td>2.6294998</td>
<td>0.0000000</td>
<td>19.8000488</td>
<td>16.4505491</td>
<td>19.8000488</td>
</tr>
<tr>
<td>DATA T</td>
<td>1000</td>
<td>CP</td>
<td>2.5571719</td>
<td>2.6270264</td>
<td>1.8430411</td>
<td>22.807670</td>
<td>19.6537406</td>
<td>20.6473758</td>
</tr>
<tr>
<td>DATA T</td>
<td>2156</td>
<td>CP</td>
<td>2.5164423</td>
<td>2.5745148</td>
<td>2.2387868</td>
<td>24.227791</td>
<td>21.651384</td>
<td>22.9149938</td>
</tr>
<tr>
<td>DATA T</td>
<td>3000</td>
<td>CP</td>
<td>2.5076611</td>
<td>2.5565408</td>
<td>2.2952124</td>
<td>25.055244</td>
<td>22.4987037</td>
<td>22.7600319</td>
</tr>
<tr>
<td>DATA T</td>
<td>5000</td>
<td>CP</td>
<td>2.5026820</td>
<td>2.5358189</td>
<td>2.3790197</td>
<td>26.348284</td>
<td>23.7990117</td>
<td>23.9585096</td>
</tr>
<tr>
<td>HZERO =</td>
<td>17300.217</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>CP</td>
<td>H-HO</td>
<td>2.5145423</td>
<td>2.5076611</td>
<td>2.5026820</td>
<td>2.500291</td>
<td>2.45358</td>
<td>2.353487</td>
</tr>
<tr>
<td>T</td>
<td>CP</td>
<td>H-H298</td>
<td>2.5145423</td>
<td>2.5076611</td>
<td>2.5026820</td>
<td>2.500291</td>
<td>2.45358</td>
<td>2.353487</td>
</tr>
<tr>
<td>H/RT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>CP</td>
<td>H-HO</td>
<td>2.5145423</td>
<td>2.5076611</td>
<td>2.5026820</td>
<td>2.500291</td>
<td>2.45358</td>
<td>2.353487</td>
</tr>
<tr>
<td>T</td>
<td>CP</td>
<td>H-H298</td>
<td>2.5145423</td>
<td>2.5076611</td>
<td>2.5026820</td>
<td>2.500291</td>
<td>2.45358</td>
<td>2.353487</td>
</tr>
<tr>
<td>H/RT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 4 (P(s) with Least-Squares Fit)
The listed output is as follows:

<table>
<thead>
<tr>
<th>R</th>
<th>1.9872600</th>
</tr>
</thead>
</table>

Punched Card Input for Examples 1 to 4 Combined

Examples 1 to 4 may be all run in a single machine pass as well as individually. In this case, however, the general data may be combined. Thus the punched card input is as follows:
Example 5 (Ar(g), H_2O(g), Mg(g), Mg(s, l), and MgO(g))

Description of problems. - This example is a combination of several problems. The input includes a more complete set of general input data and specific data for five species, Ar(g), H_2O(g), Mg(g), Mg(s, l), and MgO(g).

The general data include ATOM cards for the first 20 elements. For simplicity, however, only five sets of EFDATA and binary EF data were included for three assigned reference elements and two monatomic gases, namely, Mg(s, l), H_2(g), O_2(g), H(g), and O(g). A LISTEF card is inserted before the O_2 EFDATA, and so the data on the binary EF data cards for O_2 will be listed in the output.

The specific input data cards are for solving the following problems:

1. Ar(g) - Calculate thermodynamic functions from the following empirical equations (method COEF): C^O_D/R = 2.5, (H_0^O - H_T^O)/RT = 2.5, and S_T^O/R = 2.5 ln T + 4.3661076. Punch these coefficients for use with reference 33. Assume H_298.15^O = \Delta H_f(298.15) = 0.

2. H_2O(g) - Calculate the thermodynamic functions for T = 5000^0 K using the NRRAO2 method. List intermediate results. Use H_0^O = 57 103.5 calories per mole.

3. Mg(g) - Perform the following options in the calculations:
   (a) Calculate the thermodynamic functions using the lowered ionization potential cutoff technique (method TEMPER).
(b) Include unobserved but predicted electronic levels (FILL option).
(c) Include option for punching EFDATA and binary EF data cards and putting data on tape (EFTAPE card).
(d) Do a least-squares fit of the functions from \(1000^\circ\) to \(5000^\circ\) K assuming the following \(C_p^\circ\) equation: \(a_1 + a_2T + a_3T^2 + a_4T^3\). Constrain the curve fit to fit the functions exactly at \(1000^\circ\) K.
(e) Include DATE cards so 4/63 will be punched on coefficient cards.

4) \(\text{Mg}(s, l)\) - Perform the following options in the calculations:
(a) Calculate \(\Delta H^\circ_T\) and log \(K\), and tabulate the values with the thermodynamic functions (LOGK card).
(b) Read in data directly for the solid. (Note that DATA cards have several examples of various possible labels as given in table VIII.
(c) Assume \(C_p^\circ = 8\) calories per mole per 0 K for the liquid.
(d) Calculate the integration constants (eqs. (11) and (12)) using a heat of melting value of 2140 calories at the melting point, 923\(^\circ\) K.

5) \(\text{MgO}(g)\) - Calculate thermodynamic functions using PANDK method and including two excited electronic states. (Note that columns 79 to 80 identify to which of the three states the data belong). Calculate and list tables which include \(\Delta H^\circ_T\) and log \(K\) values. Use a dissociation energy of 90 kilocalories per mole at 0 K.

Punched card input. - The punched card input is as follows:

<table>
<thead>
<tr>
<th>Card columns 1 to 6</th>
<th>Label 1</th>
<th>Numerical value 1</th>
<th>Label 2</th>
<th>Numerical value 2</th>
<th>Label 3</th>
<th>Numerical value 3</th>
<th>Label 4</th>
<th>Numerical value 4</th>
<th>Card columns 79 to 80</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTCK</td>
<td>1.4388</td>
<td>R</td>
<td>1.98726</td>
<td>SCONST</td>
<td>-3.66511</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM AL</td>
<td>26.9800</td>
<td>AL1(S)</td>
<td>2</td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM AR</td>
<td>39.9440</td>
<td>AR1(G)</td>
<td>12</td>
<td></td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM B</td>
<td>10.8200</td>
<td>B1(S)</td>
<td>2</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM BE</td>
<td>9.013</td>
<td>BE1(S)</td>
<td>4</td>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM C</td>
<td>12.0110</td>
<td>C1(S)</td>
<td>12</td>
<td></td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM CA</td>
<td>40.0800</td>
<td>CA1(S)</td>
<td>4</td>
<td></td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM CL</td>
<td>35.4570</td>
<td>CL2(G)</td>
<td>30</td>
<td></td>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM CS</td>
<td>132.9100</td>
<td>CS1(S)</td>
<td>2</td>
<td></td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM E</td>
<td>0.000548613</td>
<td>E1(G)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM F</td>
<td>19.0000</td>
<td>F2(G)</td>
<td>30</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM H</td>
<td>1.008</td>
<td>H2(G)</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM HE</td>
<td>4.003</td>
<td>HE1(G)</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM K</td>
<td>39.1000</td>
<td>K1(S)</td>
<td>2</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM Li</td>
<td>6.940</td>
<td>Li1(S)</td>
<td>2</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM Mg</td>
<td>24.3200</td>
<td>Mg1(S)</td>
<td>4</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM N</td>
<td>14.0080</td>
<td>N2(G)</td>
<td>30</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM NA</td>
<td>22.9910</td>
<td>NA1(S)</td>
<td>2</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM Ne</td>
<td>20.1830</td>
<td>Ne1(S)</td>
<td>12</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM O</td>
<td>16.0000</td>
<td>O2(G)</td>
<td>40</td>
<td></td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM P</td>
<td>30.9750</td>
<td>P1(S)</td>
<td>30</td>
<td></td>
<td>170</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM S</td>
<td>32.0660</td>
<td>S1(S)</td>
<td>40</td>
<td></td>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATOM Si</td>
<td>28.0900</td>
<td>Si1(S)</td>
<td>12</td>
<td></td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Binary EF data for \(\text{Mg}(s)\)**

<table>
<thead>
<tr>
<th>EFDATA</th>
<th>MG1(S)</th>
<th>HZERO</th>
<th>MELTPT</th>
<th>TNOG</th>
<th>28.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>-1190.3000</td>
<td>923.0000</td>
<td>20.0000</td>
<td>20.0000</td>
<td></td>
</tr>
<tr>
<td>EFDATA1(8G)</td>
<td>0.</td>
<td>HZERO</td>
<td>-2023.800</td>
<td>MELTPL</td>
<td>0.</td>
</tr>
<tr>
<td>BCDUM000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCDUM008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Binary EF data for H₂(g)

| EFDATA1(1G) | 0. | HZERO | 50616.499 | MELTPL | -0. | T NO. | 61.0000 |
| BCDUM000 |
| BCDUM001 |
| BCDUM002 |
| BCDUM003 |
| BCDUM004 |
| BCDUM005 |
| BCDUM006 |
| BCDUM007 |
| BCDUM008 |

Binary EF data for H(g)

| EFDATA1(1G) | 0. | HZERO | 57949.1504 | MELTPL | -0. | T NO. | 61.0000 |
| BCDUM000 |
| BCDUM001 |
| BCDUM002 |
| BCDUM003 |
| BCDUM004 |
| BCDUM005 |
| BCDUM006 |
| BCDUM007 |
| BCDUM008 |

Binary EF data for O(g)

| LISTEDEF |
| EFDATA02(8G) | 0. | HZERO | -2074.7390 | MELTPL | -0. | T NO. | 61.0000 |
| BCDUM000 |
| BCDUM001 |
| BCDUM002 |
| BCDUM003 |
| BCDUM004 |
| BCDUM005 |
| BCDUM006 |
| BCDUM007 |
| BCDUM008 |

Binary EF data for O₂(g)
Listed output. - The listed output is as follows:

<table>
<thead>
<tr>
<th>ATOM</th>
<th>HCK</th>
<th>1.4398000</th>
<th>R</th>
<th>1.9872600</th>
<th>CONST</th>
<th>3.6651100</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>26.9860000</td>
<td>AL1 (S)</td>
<td>2.</td>
<td>16.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>79.9449998</td>
<td>AR1 (G)</td>
<td>12.</td>
<td>61.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>10.8200001</td>
<td>R1 (S)</td>
<td>2.</td>
<td>6.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>9.0130000</td>
<td>R1 (S)</td>
<td>4.</td>
<td>13.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>12.0110000</td>
<td>C1 (S)</td>
<td>12.</td>
<td>15.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>40.0799999</td>
<td>CA1 (S)</td>
<td>4.</td>
<td>61.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>15.4569998</td>
<td>CL2 (G)</td>
<td>30.</td>
<td>156.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS</td>
<td>132.910000</td>
<td>Li1 (S)</td>
<td>2.</td>
<td>72.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0.548613000</td>
<td>F1 (G)</td>
<td>2.</td>
<td>2.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1.9000000</td>
<td>F2 (G)</td>
<td>30.</td>
<td>6.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>1.0080000</td>
<td>H2 (G)</td>
<td>2.</td>
<td>2.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HE</td>
<td>4.0030000</td>
<td>HE1 (G)</td>
<td>4.</td>
<td>1.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>39.0099999</td>
<td>K1 (S)</td>
<td>2.</td>
<td>32.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>6.9400000</td>
<td>I1 (S)</td>
<td>2.</td>
<td>8.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MG</td>
<td>24.3199999</td>
<td>Mg1 (S)</td>
<td>6.</td>
<td>33.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>14.0080000</td>
<td>N2 (G)</td>
<td>30.</td>
<td>20.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NA</td>
<td>22.9909999</td>
<td>Na1 (S)</td>
<td>2.</td>
<td>18.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NF</td>
<td>20.1830001</td>
<td>Nf1 (G)</td>
<td>12.</td>
<td>1.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>1.9000000</td>
<td>O2 (G)</td>
<td>40.</td>
<td>15.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>30.9749999</td>
<td>P1 (S)</td>
<td>30.</td>
<td>170.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>32.0660000</td>
<td>S1 (S)</td>
<td>40.</td>
<td>215.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Si</td>
<td>28.0099999</td>
<td>Si1 (S)</td>
<td>12.</td>
<td>75.000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FFDATA</th>
<th>HZERO</th>
<th>-1190.30000</th>
<th>MELTPT</th>
<th>923.00000</th>
<th>T NO.</th>
<th>28.000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 (G)</td>
<td>HZERO</td>
<td>-2023.80000</td>
<td>MELTPT</td>
<td>923.00000</td>
<td>T NO.</td>
<td>61.000000</td>
</tr>
<tr>
<td>M2 (G)</td>
<td>HZERO</td>
<td>50616.500</td>
<td>MELTPT</td>
<td>61.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3 (G)</td>
<td>HZERO</td>
<td>57949.150</td>
<td>MELTPT</td>
<td>61.000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LISTFF</th>
<th>B3 (G)</th>
<th>H-H/FRT</th>
<th>HZERO</th>
<th>-2074.73901</th>
<th>MELTPT</th>
<th>T NO.</th>
<th>61.000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.000</td>
<td>1.44901537</td>
<td>17.3611386</td>
<td>200.000</td>
<td>200.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>798.150</td>
<td>3.50165909</td>
<td>22.15892196</td>
<td>300.000</td>
<td>300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>460.000</td>
<td>3.51978737</td>
<td>27.19011392</td>
<td>500.000</td>
<td>500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400.000</td>
<td>3.59112224</td>
<td>23.62983251</td>
<td>700.000</td>
<td>700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800.000</td>
<td>3.68991921</td>
<td>24.67412839</td>
<td>900.000</td>
<td>900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000.000</td>
<td>3.77489391</td>
<td>25.50836012</td>
<td>1100.000</td>
<td>1100.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1200.000</td>
<td>3.85329677</td>
<td>26.20367392</td>
<td>1300.000</td>
<td>1300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400.000</td>
<td>3.92130366</td>
<td>76.80289793</td>
<td>1500.000</td>
<td>1500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1600.000</td>
<td>4.98071118</td>
<td>27.33048098</td>
<td>1700.000</td>
<td>1700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800.000</td>
<td>4.93372759</td>
<td>27.90254972</td>
<td>1900.000</td>
<td>1900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000.000</td>
<td>6.00000000</td>
<td>28.22298412</td>
<td>2100.000</td>
<td>2100.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200.000</td>
<td>6.16791654</td>
<td>36.61113666</td>
<td>2300.000</td>
<td>2300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2400.000</td>
<td>6.22718788</td>
<td>28.98196840</td>
<td>2500.000</td>
<td>2500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2500.000</td>
<td>6.20408076</td>
<td>29.31714988</td>
<td>2700.000</td>
<td>2700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2800.000</td>
<td>6.22605004</td>
<td>29.63103266</td>
<td>2900.000</td>
<td>2900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000.000</td>
<td>6.28065569</td>
<td>29.92424012</td>
<td>3100.000</td>
<td>3100.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3200.000</td>
<td>6.30489400</td>
<td>30.20178723</td>
<td>3300.000</td>
<td>3300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3400.000</td>
<td>6.34773630</td>
<td>30.4436810</td>
<td>3500.000</td>
<td>3500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3600.000</td>
<td>6.33902075</td>
<td>30.71377916</td>
<td>3700.000</td>
<td>3700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3800.000</td>
<td>6.40460883</td>
<td>30.95136023</td>
<td>3900.000</td>
<td>3900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3900.000</td>
<td>6.43852939</td>
<td>31.17972082</td>
<td>4100.000</td>
<td>4100.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000.000</td>
<td>6.45905357</td>
<td>31.39548731</td>
<td>4300.000</td>
<td>4300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4400.000</td>
<td>6.47919421</td>
<td>31.60507572</td>
<td>4500.000</td>
<td>4500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000.000</td>
<td>6.51779236</td>
<td>31.8011649</td>
<td>4700.000</td>
<td>4700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000.000</td>
<td>6.5250646</td>
<td>31.9960747</td>
<td>4900.000</td>
<td>4900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000.000</td>
<td>6.56515411</td>
<td>32.18279886</td>
<td>5100.000</td>
<td>5100.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5200.000</td>
<td>6.58749345</td>
<td>32.36229420</td>
<td>5300.000</td>
<td>5300.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5400.000</td>
<td>6.60911517</td>
<td>32.5361935</td>
<td>5500.000</td>
<td>5500.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5600.000</td>
<td>6.6272367</td>
<td>32.70363739</td>
<td>5700.000</td>
<td>5700.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5900.000</td>
<td>6.66858687</td>
<td>33.02499717</td>
<td>5900.000</td>
<td>5900.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARI G1</td>
<td>ASIONH</td>
<td>T</td>
<td>298.15000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>---</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>T</td>
<td>298.15000</td>
<td>T</td>
<td>1000.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td>CHEF</td>
<td>REDUCE H2980</td>
<td>T</td>
<td>1000.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>T</td>
<td>6000.00000</td>
<td>C</td>
<td>2.5000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>TPUNCH</td>
<td>1000.00000</td>
<td>T</td>
<td>5000.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FINISH

PUNCHED BINARY CARDS--

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>0.30000000E03</th>
<th>0.50000000E04</th>
<th>0.10000000E04</th>
<th>0.50000000E04</th>
<th>0.25000000E01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.30000000E04</td>
<td>0.25000000E01</td>
<td>0.00000000E00</td>
<td>0.00000000E00</td>
<td>18.01600000E00</td>
<td></td>
</tr>
</tbody>
</table>

FINISH

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>ASIONH</th>
<th>T</th>
<th>298.15000</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>T</td>
<td>298.15000</td>
<td>T</td>
</tr>
<tr>
<td>METHOD</td>
<td>CHEF</td>
<td>REDUCE H2980</td>
<td>T</td>
</tr>
<tr>
<td>DATA</td>
<td>T</td>
<td>6000.00000</td>
<td>C</td>
</tr>
<tr>
<td>DATA</td>
<td>TPUNCH</td>
<td>1000.00000</td>
<td>T</td>
</tr>
</tbody>
</table>

FINISH

PUNCHED BINARY CARDS--

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>0.30000000E03</th>
<th>0.50000000E04</th>
<th>0.10000000E04</th>
<th>0.50000000E04</th>
<th>0.25000000E01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.30000000E04</td>
<td>0.25000000E01</td>
<td>0.00000000E00</td>
<td>0.00000000E00</td>
<td>18.01600000E00</td>
<td></td>
</tr>
</tbody>
</table>

FINISH

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>ASIONH</th>
<th>T</th>
<th>298.15000</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>T</td>
<td>298.15000</td>
<td>T</td>
</tr>
<tr>
<td>METHOD</td>
<td>CHEF</td>
<td>REDUCE H2980</td>
<td>T</td>
</tr>
<tr>
<td>DATA</td>
<td>T</td>
<td>6000.00000</td>
<td>C</td>
</tr>
<tr>
<td>DATA</td>
<td>TPUNCH</td>
<td>1000.00000</td>
<td>T</td>
</tr>
</tbody>
</table>

FINISH

PUNCHED BINARY CARDS--

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>0.30000000E03</th>
<th>0.50000000E04</th>
<th>0.10000000E04</th>
<th>0.50000000E04</th>
<th>0.25000000E01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.30000000E04</td>
<td>0.25000000E01</td>
<td>0.00000000E00</td>
<td>0.00000000E00</td>
<td>18.01600000E00</td>
<td></td>
</tr>
</tbody>
</table>

FINISH

PUNCHED BINARY CARDS--

<table>
<thead>
<tr>
<th>ARI G1</th>
<th>0.30000000E03</th>
<th>0.50000000E04</th>
<th>0.10000000E04</th>
<th>0.50000000E04</th>
<th>0.25000000E01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.30000000E04</td>
<td>0.25000000E01</td>
<td>0.00000000E00</td>
<td>0.00000000E00</td>
<td>18.01600000E00</td>
<td></td>
</tr>
</tbody>
</table>
\[ \begin{align*}
A &= 27.8487807, \quad \alpha = 14.8976400, \quad \theta = 9.2828590, \quad \rho = 0.21300000 - 0.04 \\
A &= 0.325878, \quad \alpha = 0.7560000, \quad \theta = 0.2380000, \quad \rho = 0.2018000, \quad I = \text{null} \\
A &= -0.4962816, \quad \alpha = -2.9410000, \quad \theta = -0.1600000, \quad \rho = 0.1392000, \quad I = \text{null} \\
A &= 0.3296888, \quad \alpha = 1.2530000, \quad \theta = 0.0780000, \quad \rho = 0.1445000, \quad I = \text{null} \\
\theta &= 4.4457220, \quad \theta = 3.7102338, \quad \theta = \text{null} \\
Y(1,1,1) &= 0.70, \quad Y(1,1,2) = -0.10, \quad Y(1,1,3) = 0.68, \quad Y(1,2,1) = -0.10, \quad Y(1,2,2) = -0.10, \quad Y(1,2,3) = -0.45 \\
Y(1,3,1) &= -16.400, \quad Y(1,3,2) = -6.70, \quad Y(1,3,3) = -4.645000 \\
LAW &= \text{null} \\
V(1) &= 3656.65000, \quad \gamma_1 = 0. \\
V(2) &= 354.78000, \quad \gamma_2 = 0. \\
V(3) &= 375.79000, \quad \gamma_3 = 0. \\
T &= 5000.000, \\
\alpha &= 1.042776, \quad \theta = 0.7492506, \quad \rho = 1.5166566, \quad I = 1 \\
\beta &= 0.4503919, \quad \theta = 0.6319697, \quad \rho = 2.7171573, \quad I = 2 \\
\gamma &= 1.0807691, \quad \theta = 0.3393559, \quad \rho = 5.1363757, \quad I = 3 \\
\text{CONTRIBUTIONS} &= \text{null} \\
\text{FI-FCR} &= 1.000, \quad \theta = 0.6319697, \quad \rho = 2.7171573, \quad \xi = 1.000, \\
\text{F} &= 294.707, \quad \theta = 7.9943354, \quad \rho = 1.5000000, \quad \xi = 2.8033517 \\
\text{F} &= 1.0124, \quad \theta = 0.0670000, \quad \rho = 0.1060000, \quad \xi = 2.0130000 \\
\text{F} &= 2.000, \quad \theta = 1.0660000, \quad \rho = 0.0789000, \quad \xi = 1.0000000 \\
\text{FST} &= \text{null} \\
\text{FIRST ORDER CORRECTIONS} &= \text{null} \\
\text{F} &= 2.5846, \quad \theta = -0.0444285, \quad \rho = -0.0405801, \quad \xi = -0.0619340 \\
\text{F} &= 1.0700, \quad \theta = 0.0685679, \quad \rho = 0.1264388, \quad \xi = 0.3033360 \\
\text{F} &= 0.9970, \quad \theta = 0.0099900, \quad \rho = 0.0116778, \quad \xi = 0.0387308 \\
\text{SECOND ORDER CORRECTIONS} &= \text{null} \\
\text{F} &= 1.0098, \quad \theta = 0.0075028, \quad \rho = 0.2659192, \quad \xi = 0.0922170 \\
\text{F} &= 1.0043, \quad \theta = 0.0075028, \quad \rho = 0.2659192, \quad \xi = 0.0922170 \\
\text{F} &= 0.9977, \quad \theta = -0.0022621, \quad \rho = -0.0755046, \quad \xi = 0.0320273 \\
\text{HIGH} &= \text{null} \\
\text{H_R} &= -87103.500, \\
\text{T} &= 5000, \quad \theta = 7.3572721, \quad \rho = 2.1244898, \quad \xi = 38.0642924, \quad \eta = 31.9398026, \quad \delta = 37.6867604 \\
\text{H_L} &= -87103.500, \\
\text{T} &= 5000, \quad \theta = 14.6107741, \quad \rho = 6.0954.7871, \quad \xi = 75.693645, \quad \eta = 317363.4570, \quad \delta = 3751.2828, \quad \gamma = 374366.4531 \\
\end{align*} \]
<table>
<thead>
<tr>
<th>MG1(G)</th>
<th>DELTAH</th>
<th>KCAL</th>
<th>T</th>
<th>298.1500</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTATP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTSBS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTSOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**METHOD**

**DATE** 4/63

**DATA**

<table>
<thead>
<tr>
<th>DATA</th>
<th>0</th>
<th>0</th>
<th>21850.368</th>
<th>1</th>
<th>21870.426</th>
<th>2</th>
<th>21911.140</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>1</td>
<td>35051.360</td>
<td>2</td>
<td>46403.140</td>
<td>3</td>
<td>47957.035</td>
<td>2</td>
<td>47957.018</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>41197.370</td>
<td>0</td>
<td>43503.940</td>
<td>0</td>
<td>47841.200</td>
<td>1</td>
<td>47841.440</td>
</tr>
<tr>
<td>DATA</td>
<td>2</td>
<td>47851.140</td>
<td>1</td>
<td>49346.710</td>
<td>2</td>
<td>53134.700</td>
<td>7</td>
<td>54192.440</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>47957.047</td>
<td>0</td>
<td>57812.720</td>
<td>1</td>
<td>57833.280</td>
<td>2</td>
<td>57873.890</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>51872.360</td>
<td>0</td>
<td>52556.370</td>
<td>4</td>
<td>54252.600</td>
<td>1</td>
<td>54699.400</td>
</tr>
<tr>
<td>DATA</td>
<td>2</td>
<td>54676.760</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>54676.660</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>2</td>
<td>55891.830</td>
<td>0</td>
<td>56187.030</td>
<td>1.5</td>
<td>57018.800</td>
<td>2</td>
<td>57020.100</td>
</tr>
<tr>
<td>DATA</td>
<td>7</td>
<td>56308.430</td>
<td>7</td>
<td>56968.310</td>
<td>10</td>
<td>57204.220</td>
<td>7</td>
<td>57204.220</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>57895.500</td>
<td>0</td>
<td>58009.460</td>
<td>4</td>
<td>58678.400</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>58023.270</td>
<td>7</td>
<td>58442.620</td>
<td>10</td>
<td>58575.540</td>
<td>3</td>
<td>58575.940</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>58962.490</td>
<td>2</td>
<td>59690.020</td>
<td>7</td>
<td>59880.300</td>
<td>13.5</td>
<td>59935.980</td>
</tr>
<tr>
<td>DATA</td>
<td>2</td>
<td>59061.090</td>
<td>7</td>
<td>59317.400</td>
<td>10</td>
<td>59400.770</td>
<td>3</td>
<td>59400.770</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>59668.200</td>
<td>2</td>
<td>60127.310</td>
<td>7</td>
<td>60263.400</td>
<td>13.5</td>
<td>60301.300</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>60103.500</td>
<td>2</td>
<td>60435.150</td>
<td>7</td>
<td>60534.500</td>
<td>13.5</td>
<td>60562.640</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>60420.200</td>
<td>2</td>
<td>60658.370</td>
<td>7</td>
<td>60734.900</td>
<td>13.5</td>
<td>60755.780</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>60649.200</td>
<td>2</td>
<td>60826.600</td>
<td>7</td>
<td>60884.800</td>
<td>13.5</td>
<td>60902.530</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>60820.900</td>
<td>2</td>
<td>60955.800</td>
<td>7</td>
<td>61002.200</td>
<td>3</td>
<td>61016.420</td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>60952.700</td>
<td>7</td>
<td>61094.600</td>
<td>3</td>
<td>61106.980</td>
<td>15</td>
<td>61664.140</td>
</tr>
</tbody>
</table>

**FINISH**

<table>
<thead>
<tr>
<th>N</th>
<th>PRELI</th>
<th>SUM(?J+1)</th>
<th>ACT.</th>
<th>SUM(2J+1)</th>
<th>DIFF</th>
<th>MAX LEVEL</th>
<th>2J+1, MAX LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>3</td>
<td>38.0</td>
<td>42.0</td>
<td>-6.0</td>
<td>57873.690</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>4</td>
<td>64.0</td>
<td>64.0</td>
<td>0.0</td>
<td>54676.759</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>5</td>
<td>100.0</td>
<td>64.0</td>
<td>36.0</td>
<td>57204.220</td>
<td>43.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>6</td>
<td>144.0</td>
<td>61.0</td>
<td>83.0</td>
<td>58575.540</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>7</td>
<td>196.0</td>
<td>61.0</td>
<td>135.0</td>
<td>59400.770</td>
<td>142.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>11</td>
<td>267.0</td>
<td>51.0</td>
<td>205.0</td>
<td>59935.980</td>
<td>233.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>12</td>
<td>324.0</td>
<td>51.0</td>
<td>273.0</td>
<td>60301.299</td>
<td>301.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>13</td>
<td>384.0</td>
<td>51.0</td>
<td>349.0</td>
<td>60884.800</td>
<td>377.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>14</td>
<td>444.0</td>
<td>51.0</td>
<td>433.0</td>
<td>60755.780</td>
<td>461.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>15</td>
<td>504.0</td>
<td>51.0</td>
<td>525.0</td>
<td>60902.530</td>
<td>593.0</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>16</td>
<td>564.0</td>
<td>51.0</td>
<td>619.0</td>
<td>61664.140</td>
<td>766.0</td>
<td></td>
</tr>
</tbody>
</table>

104
<table>
<thead>
<tr>
<th>T</th>
<th>FPR INPUT</th>
<th>FPR CALC</th>
<th>FHR INPUT</th>
<th>HHR CALC</th>
<th>SFR INPUT</th>
<th>SFR CALC</th>
<th>-PRH INPUT</th>
<th>-PRH CALC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1100.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1200.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1300.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1400.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1500.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1600.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1700.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1800.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>1900.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2000.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2100.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2200.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2300.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2400.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2500.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2600.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2700.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
<tr>
<td>2800.00</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
<td>2.5000000</td>
</tr>
</tbody>
</table>

**Note:** The table continues with similar entries for various temperatures and values.
<table>
<thead>
<tr>
<th>MGI ID</th>
<th>ASHHOM</th>
<th>T (°C)</th>
<th>MTHID</th>
<th>RADIN</th>
<th>MELPT</th>
<th>T</th>
<th>MTHID</th>
<th>RADIN</th>
<th>MELPT</th>
<th>T</th>
<th>MTHID</th>
<th>RADIN</th>
<th>MELPT</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>107</td>
<td>298.1500</td>
<td>1.885299</td>
<td>0.769401</td>
<td>-5.220259</td>
<td>1.138759</td>
<td>0.3693528</td>
<td>6.3500068</td>
<td>-5.220259</td>
<td>6.3500068</td>
<td>6.3500068</td>
<td>6.3500068</td>
<td>6.3500068</td>
<td>6.3500068</td>
<td>6.3500068</td>
</tr>
<tr>
<td>200</td>
<td>2.726569</td>
<td>1.598186</td>
<td>-1.409765</td>
<td>2.797165</td>
<td>1.1855036</td>
<td>4.1867084</td>
<td>1.1855036</td>
<td>4.1867084</td>
<td>1.1855036</td>
<td>4.1867084</td>
<td>1.1855036</td>
<td>4.1867084</td>
<td>1.1855036</td>
<td>4.1867084</td>
</tr>
<tr>
<td>300</td>
<td>2.995039</td>
<td>2.010502</td>
<td>0.016459</td>
<td>2.9395567</td>
<td>0.7119555</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
<td>4.3733844</td>
</tr>
<tr>
<td>400</td>
<td>3.297128</td>
<td>2.823693</td>
<td>1.259610</td>
<td>3.297128</td>
<td>2.823693</td>
<td>1.259610</td>
<td>3.297128</td>
<td>2.823693</td>
<td>1.259610</td>
<td>3.297128</td>
<td>2.823693</td>
<td>1.259610</td>
<td>3.297128</td>
<td>2.823693</td>
</tr>
<tr>
<td>500</td>
<td>3.476000</td>
<td>2.601000</td>
<td>1.097263</td>
<td>3.476000</td>
<td>2.601000</td>
<td>1.097263</td>
<td>3.476000</td>
<td>2.601000</td>
<td>1.097263</td>
<td>3.476000</td>
<td>2.601000</td>
<td>1.097263</td>
<td>3.476000</td>
<td>2.601000</td>
</tr>
</tbody>
</table>

The above table represents a grid with various entries, likely related to scientific data or another structured dataset. Each row contains numerical data, and the table is sourced from a page labeled with a number '108'.
<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>H-HO/RT</th>
<th>S</th>
<th>-(F-HO)/RT</th>
<th>H/RT</th>
<th>REFERENCE ELEMENTS</th>
<th>GASEOUS ATOMS</th>
<th>DELTA H/RT</th>
<th>DELTA F/RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.6480</td>
<td>0.0640</td>
<td>1.1388</td>
<td>0.3694</td>
<td>-5.2203</td>
<td>6.3900</td>
<td>0</td>
<td>0</td>
<td>-1.7967</td>
</tr>
<tr>
<td>200</td>
<td>2.7724</td>
<td>1.5874</td>
<td>2.7732</td>
<td>1.1858</td>
<td>-1.4075</td>
<td>4.1806</td>
<td>0</td>
<td>0</td>
<td>-0.2606</td>
</tr>
<tr>
<td>250</td>
<td>2.0835</td>
<td>0.0700</td>
<td>5.9149</td>
<td>1.0560</td>
<td>0.0000</td>
<td>3.9149</td>
<td>0</td>
<td>0</td>
<td>-1.2071</td>
</tr>
<tr>
<td>300</td>
<td>2.8971</td>
<td>2.0150</td>
<td>5.9316</td>
<td>1.9186</td>
<td>0.0185</td>
<td>3.9316</td>
<td>0</td>
<td>0</td>
<td>-0.4456</td>
</tr>
<tr>
<td>400</td>
<td>1.1481</td>
<td>2.2786</td>
<td>4.8152</td>
<td>2.9365</td>
<td>0.7812</td>
<td>4.0339</td>
<td>0</td>
<td>0</td>
<td>-35.5170</td>
</tr>
<tr>
<td>500</td>
<td>1.2676</td>
<td>2.4034</td>
<td>5.2527</td>
<td>3.0663</td>
<td>1.2655</td>
<td>4.0463</td>
<td>0</td>
<td>0</td>
<td>-29.5063</td>
</tr>
<tr>
<td>600</td>
<td>3.4447</td>
<td>2.6089</td>
<td>6.1366</td>
<td>3.5286</td>
<td>1.0097</td>
<td>4.2629</td>
<td>0</td>
<td>0</td>
<td>-11.4999</td>
</tr>
<tr>
<td>700</td>
<td>3.6367</td>
<td>2.7313</td>
<td>7.6750</td>
<td>3.8999</td>
<td>1.1875</td>
<td>4.7995</td>
<td>0</td>
<td>0</td>
<td>-21.8169</td>
</tr>
<tr>
<td>800</td>
<td>3.7388</td>
<td>2.8677</td>
<td>7.1601</td>
<td>4.1124</td>
<td>2.0990</td>
<td>5.0611</td>
<td>0</td>
<td>0</td>
<td>-19.2853</td>
</tr>
<tr>
<td>900</td>
<td>3.9210</td>
<td>2.9868</td>
<td>7.0110</td>
<td>4.6544</td>
<td>2.2911</td>
<td>5.3199</td>
<td>0</td>
<td>0</td>
<td>-18.7688</td>
</tr>
<tr>
<td>1000</td>
<td>4.0756</td>
<td>4.4179</td>
<td>8.8768</td>
<td>4.7289</td>
<td>3.9498</td>
<td>5.8779</td>
<td>0</td>
<td>0</td>
<td>-17.6021</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>H-HO/RT</th>
<th>S</th>
<th>-(F-HO)/RT</th>
<th>H/RT</th>
<th>REFERENCE ELEMENTS</th>
<th>GASEOUS ATOMS</th>
<th>DELTA H/RT</th>
<th>DELTA F/RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1100</td>
<td>4.0756</td>
<td>4.1282</td>
<td>9.5831</td>
<td>5.6458</td>
<td>5.8387</td>
<td>5.9994</td>
<td>0</td>
<td>0</td>
<td>-14.5263</td>
</tr>
<tr>
<td>1300</td>
<td>4.0756</td>
<td>4.1174</td>
<td>10.2556</td>
<td>6.1431</td>
<td>5.6517</td>
<td>6.6039</td>
<td>0</td>
<td>0</td>
<td>-12.5050</td>
</tr>
<tr>
<td>1400</td>
<td>4.0756</td>
<td>4.1062</td>
<td>10.5959</td>
<td>6.4477</td>
<td>5.7874</td>
<td>6.8755</td>
<td>0</td>
<td>0</td>
<td>-11.0850</td>
</tr>
<tr>
<td>1600</td>
<td>4.0756</td>
<td>4.1039</td>
<td>10.8316</td>
<td>6.7308</td>
<td>5.3075</td>
<td>5.4589</td>
<td>0</td>
<td>0</td>
<td>-10.2463</td>
</tr>
<tr>
<td>1900</td>
<td>4.0756</td>
<td>4.0881</td>
<td>11.5656</td>
<td>7.4773</td>
<td>3.7556</td>
<td>7.8100</td>
<td>0</td>
<td>0</td>
<td>-8.2826</td>
</tr>
<tr>
<td>2100</td>
<td>4.0756</td>
<td>4.0850</td>
<td>11.7882</td>
<td>7.6982</td>
<td>3.7268</td>
<td>8.0135</td>
<td>0</td>
<td>0</td>
<td>-7.7644</td>
</tr>
<tr>
<td>2200</td>
<td>4.0756</td>
<td>4.0821</td>
<td>11.9997</td>
<td>7.9077</td>
<td>3.7826</td>
<td>8.2072</td>
<td>0</td>
<td>0</td>
<td>-7.3018</td>
</tr>
<tr>
<td>2500</td>
<td>4.0756</td>
<td>4.0769</td>
<td>12.1734</td>
<td>8.2965</td>
<td>3.8074</td>
<td>8.5867</td>
<td>0</td>
<td>0</td>
<td>-6.4994</td>
</tr>
<tr>
<td>2700</td>
<td>4.0756</td>
<td>4.0747</td>
<td>12.5524</td>
<td>8.4777</td>
<td>3.8148</td>
<td>8.7391</td>
<td>0</td>
<td>0</td>
<td>-6.1355</td>
</tr>
<tr>
<td>2900</td>
<td>4.0756</td>
<td>4.0727</td>
<td>12.7217</td>
<td>8.6510</td>
<td>3.8731</td>
<td>8.9096</td>
<td>0</td>
<td>0</td>
<td>-5.8308</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>H-HO/RT</th>
<th>S</th>
<th>-(F-HO)/RT</th>
<th>H/RT</th>
<th>REFERENCE ELEMENTS</th>
<th>GASEOUS ATOMS</th>
<th>DELTA H/RT</th>
<th>DELTA F/RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>4.0756</td>
<td>4.0708</td>
<td>12.8880</td>
<td>8.8173</td>
<td>3.8312</td>
<td>9.0568</td>
<td>0</td>
<td>0</td>
<td>-5.5367</td>
</tr>
<tr>
<td>METHOD</td>
<td>PANKA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MULTIPLACED WEIT = 40, 12700**

**DATA**

<table>
<thead>
<tr>
<th>T</th>
<th>204.16000</th>
<th>T</th>
<th>10000.0</th>
<th>T</th>
<th>10000.0</th>
<th>T</th>
<th>6000.0</th>
</tr>
</thead>
</table>

**FINISH**

<table>
<thead>
<tr>
<th>H</th>
<th>2047, 846</th>
</tr>
</thead>
</table>

**MULTIPLACED WEIT = 2047.846**

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>(H-H0)</th>
<th>s</th>
<th>(H-H298)</th>
<th>S</th>
<th>F</th>
<th>H</th>
<th>-F</th>
</tr>
</thead>
</table>

**MULTIPLACED WEIT = 2047.846**

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>(H-H0)</th>
<th>H</th>
<th>(H-H298)</th>
<th>S</th>
<th>REFERENCE ELEMENTS</th>
<th>GASEOUS ATOMS</th>
</tr>
</thead>
</table>

*Change in Phase if an assigned reference element has occurred between this temperature and the preceding one;***

<table>
<thead>
<tr>
<th>T</th>
<th>CP</th>
<th>(H-H0)</th>
<th>F</th>
<th>H</th>
<th>DELTA H</th>
<th>LOG K</th>
<th>DELTA H</th>
<th>LOG K</th>
</tr>
</thead>
</table>

*Change in Phase if an assigned reference element has occurred between this temperature and the preceding one;***

**110**
REFERENCES


11. Hilsenrath, Joseph; Messina, Carla G.; and Evans, William H.: Tables of Ideal Gas Thermodynamic Functions for 73 Atoms and Their First and Second Ions to 10 000°K. (AFWL-TDR-64-44, DDC No. AD-606163), National Bureau of Standards, 1964


### Table I. - Some Terms in

<table>
<thead>
<tr>
<th>Number</th>
<th>Method</th>
<th>Subscript in equation (9)</th>
<th>In $Q_{m}^{m}$ terms</th>
<th>$T \frac{d(ln Q_{m}^{m})}{dT}$ terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RRHO</td>
<td>e</td>
<td>$\ln g_m - \frac{c_2 T_0}{T}$</td>
<td>$\frac{c_2 T_0}{T}$</td>
</tr>
<tr>
<td>2</td>
<td>PANDK</td>
<td>v</td>
<td>$\sum_{i=1}^{n} d_i \ln (s_i)$</td>
<td>$\sum_{i=1}^{n} d_i u_i r_i s_i$</td>
</tr>
<tr>
<td>3</td>
<td>JANAF</td>
<td>r</td>
<td>For diatomic and linear molecules,</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>JANAF</td>
<td>r</td>
<td>For nonlinear molecules,</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>NRRAO</td>
<td>$\rho$</td>
<td>$\rho T$</td>
<td>$\rho T$</td>
</tr>
<tr>
<td>6</td>
<td>NRRAO</td>
<td>$\theta$</td>
<td>$\ln \left( 1 + \frac{\theta_1}{T} + \frac{\theta_2}{T^2} + \frac{\theta_3}{T^3} \right)$</td>
<td>$\frac{\theta_1}{T} + \frac{2 \theta_2}{T^2} + \frac{3 \theta_3}{T^3} \frac{1}{Q_0}$</td>
</tr>
<tr>
<td>7</td>
<td>NRRAO</td>
<td>$W$</td>
<td>Tristatic linear molecules where Fermi resonance occurs</td>
<td>$\ln Q_w \left( S \frac{r_1 r_2}{1 - r_1} \right)$</td>
</tr>
</tbody>
</table>

* Rigid-Rotator Harmonic-Oscillator approximation.
* Modified Pennington and Kobe method.
* Joint Army Navy Air Force method.
\[ T^2 \frac{d^2 \ln Q^m}{dT^2} \text{ terms} \]

<table>
<thead>
<tr>
<th>Type of molecule</th>
<th>Remarks</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diatomic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear polyatomic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-linear</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ -2c_2 \frac{T_0}{T} \]

\[ \sum_{i=1}^{n} d_i u_i r_i s_i (u_i s_i - 2) \]

\[ c_2 = \frac{hc}{k} \]

\[ g_m = \text{statistical weight} \]

\[ T_0 = \text{electronic excitation energy} \]

\[ d_i = \text{degeneracy} \]

\[ n = \text{number of unique frequencies} \]

\[ u_i = \frac{c_2 v_i}{T} \]

\[ r_i = e^{-r_i} \]

\[ s_i = \frac{1}{1 - r_i} \]

\[ \mu_1 = \omega_e - 2a_{0}v'_{e} + 3.25 \omega_e y_{e} + 5a_{0}z_{e} \]

\[ \alpha = \text{symmetry number} \]

\[ A_0 = A_{e} - \frac{1}{2} \sum_{i=1}^{n} d_i \phi_i^A \]

\[ B_0 = B_{e} - \frac{1}{2} \sum_{i=1}^{n} d_i \phi_i^B \]

\[ C_0 = C_{e} - \frac{1}{2} \sum_{i=1}^{n} d_i \phi_i^C \]

\[ \rho = \frac{2D}{B_{0}^2} \text{ and for JANAF, } \rho = \frac{4(D_{B_{0}})^{1/2}}{v_{1} c_{2}} \]

\[ D = D_{e} + \frac{\beta_1}{2} + \frac{\beta_2}{4} + \frac{\beta_3}{8} \text{ if not given, } D_{e} = \frac{4B_{0}^3}{\omega_e} \]

\[ \theta_1 = \frac{c_2 B_{0}}{2}, \theta_2 = \frac{(c_2 B_{0})^2}{15}, \theta_3 = \frac{4(c_2 B_{0})^3}{315} \]

\[ \theta_1 = \frac{c_2}{12} \left[ 2(A_0 + B_0 + C_0) - \frac{A_0 B_0}{C_0} - \frac{A_0 C_0}{B_0} - \frac{B_0 C_0}{A_0} \right] \]

\[ \theta_2 = \frac{c_2}{480} \left[ 10(A_0^2 + B_0^2 + C_0^2) + 12(A_0 B_0 + A_0 C_0 + B_0 C_0) \right. \]

\[ \left. - 12 \left( \frac{A_0^2 B_0 + A_0^2 C_0 + B_0^2 C_0 + B_0^2 A_0 + C_0^2 A_0}{A_0} \right) \right] \]

\[ \theta_3 = 0 \]

\[ \ln Q_{W} = \frac{2u_{W} r_{W} s_{W}}{u_{W} + r_{W} s_{W}} (1 + r_{W} s_{W}) \]

\[ u_{W} = 2c_{2} r_{W}^{2}/T \]

\[ r_{W} = e^{-u_{W}} \]

\[ s_{W} = \frac{1}{1 - r_{W}} \]

\[ S = (1 + 2r_{W} s_{W})u_{W} + 2(r_{W} r_{W} s_{W} - 1) \]

\( ^{d}\text{Nonrigid-Rotator Anharmonic-Oscillator 1.} \)

\( ^{e}\text{Nonrigid-Rotator Anharmonic-Oscillator 2.} \)
<table>
<thead>
<tr>
<th>Formula number</th>
<th>Method</th>
<th>NRRAO1</th>
<th>NRRAO2</th>
<th>( \ln Q_c^m ) terms (^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>( \sum_{i=1}^{n} d_i a_i r_i s_i \left[1 + \frac{\delta_j}{2} a_i r_i s_i + \frac{\delta_k}{6} a_i^2 r_i^2 s_i^2 (1 + r_i) \right] )</td>
</tr>
<tr>
<td>9</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>( \sum_{i=1}^{n} d_i P_i r_i s_i )</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>( \sum_{i=1}^{n} \left[2 d_i a_i r_i s_i (a_i s_i + a_i r_i s_i + 1) + \sum_{j=1}^{n} d_i d_j a_i r_i r_j s_i s_j \right] ) ( + \sum_{j=1}^{n} d_i d_j a_i a_j (1 + \delta) r_i r_j r_i^2 s_i^2 s_j )</td>
</tr>
<tr>
<td>11</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>( a_{111} r_1^2 \left(1 + 4 r_1 + r_1^2 \right) ) (diatomics only)</td>
</tr>
<tr>
<td>12</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>( -\frac{c_2}{T} \sum_{i=1}^{n} d_i (d_j + \delta_j) X_{ij} r_i r_j s_i s_j )</td>
</tr>
<tr>
<td>13</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>( -\frac{c_2}{T} \sum_{i=1}^{n} d_i (d_j + \delta_j) (d_k + \delta_k) Y_{ijk} r_i r_j r_k s_i s_j s_k )</td>
</tr>
<tr>
<td>14</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>( -\frac{c_2}{T} \sum_{i=1}^{n} d_i (d_j + \delta_j) (X_{ij} + G_j) r_i r_j s_i s_j )</td>
</tr>
</tbody>
</table>

\(^a\) Rigid-Rotator Harmonic-Oscillator approximation.  
\(^b\) Modified Pennington and Kobe method.  
\(^c\) Nonrigid-Rotator Anharmonic-Oscillator 1.  
\(^d\) Nonrigid-Rotator Anharmonic-Oscillator 2.
<table>
<thead>
<tr>
<th>Type of molecule</th>
<th>Linear polyatomic</th>
<th>Non-linear</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diatomic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_i = degeneracy</td>
<td>r_j = e</td>
<td></td>
</tr>
<tr>
<td></td>
<td>v_i = c_2ν_i/T</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>a_i = 1/(1 - r_i)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>a_i = (α_1 - α_2) - 0.75 α_3 / B_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P_i = a_i (A_i + 1) + 1/4 \left[ \frac{A_i^2}{A_0} + \frac{B_i^2}{B_0} + \frac{C_i^2}{C_0} \right]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X_i = x_i + (d_i + 1)γ_ijkl + \sum_{k=1}^{n} \frac{d_k}{2} \gamma_{ilk}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M_i = \omega \nu e + 5\omega \nu e</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>\alpha_{ij} = \sigma_{ij} / B_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>r_i = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>δ_i = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>a_{111} = -α_3 / B_0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Remarks:

- Definitions

<table>
<thead>
<tr>
<th>Remarks</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Expressions for the terms in \( \ln Q_c^m \):

\[
\frac{d}{[\ln Q_c^m/T]} = \sum \ln Q_{c_j} S_j + T^2 \frac{\delta^2[\ln Q_c^m]/\delta T^2}{T^4}
\]

\[
= \sum \ln Q_{c_j} \left[ \sum m_j^2 h_j s_j (s_j + 1) - 2S_j + P_j \right] \quad \text{where} \quad \ln Q_c^m = \sum \ln Q_{c_j} \quad \text{and} \quad \ln Q_{c_j}
\]

any term in formulas 8 to 27 which has the formula \( \ln Q_{c_j} = (c_2 / T)^{P_j} C_j \), \( C_j \) is a constant; \( n_i \) and \( m_i \) are integer exponents; and \( h_j \) is an integer subscript, and where \( S_j = \sum_{i=1}^{n_i} \omega h_i (n_i + m_i + 1) - p_j \).
<table>
<thead>
<tr>
<th>Formula number</th>
<th>Method</th>
<th>PANDR$^a$ or JANAP$^b$</th>
<th>NRRAOI$^c$</th>
<th>NRRAG$^d$</th>
<th>$\ln Q_c^{m}$ terms$^e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>$\frac{24c^2}{T} \omega_0^2 \epsilon_0^4 \epsilon_1^4$ (diatomics only)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>$\frac{c^2}{T} \sum_{i=1}^{n} 2r_i^4 s_i^4 (1 - 2a_i r_i^2 s_i^2)$</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>$\frac{c^2}{T} \sum_{j=1}^{n} d_j (1 + \delta_j) (1 + \delta_j)<em>{1i} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} d_j (1 + \delta_j) (1 + \delta_j)<em>{1i} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{c^2}{T} \sum_{i=1}^{n} (1 + 8r_i^2 + r_i^2)$</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{c^2}{T} \sum_{i=1}^{n} 2r_i^4 s_i^4 [1 + 76r_i^2 + r_i^2 (1 + 55r_i^2)]$</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} 4a_j [X_{ij} d_j (1 + 1)]_{r_i^2 s_i^2}$</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>$\frac{1}{2 \sqrt{T}} \sum_{j=1}^{n} \mathcal{D}<em>{ij} X</em>{ij} r_{ij}^2 s_{ij}^2 s_{j}$</td>
<td></td>
</tr>
</tbody>
</table>

$^a$Rigid-Rotator Harmonic-Oscillator approximation.

$^b$Modified Pennington and Kobe method.

$^c$Nonrigid-Rotator Anharmonic-Oscillator 1.

$^d$Nonrigid-Rotator Anharmonic-Oscillator 2.

118
### TERMS IN \( \ln Q^m_c \)

<table>
<thead>
<tr>
<th>Type of molecule</th>
<th>Remarks</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diatomic Linearmolyatomic Non-linear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ijk} = (2 - \delta</em>{jk})(1 + \delta_{ij})(1 + \delta_{ik})d_i(d_j + \delta_{ij})(d_k + \delta_{jk}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ijk} = 2(1 + \delta</em>{ij})(1 + \delta_{ik} + \delta_{jk})(d_i + \delta_{ij})d_j(d_k + \delta_{jk}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ijk} = 2(1 + \delta</em>{ij})(1 + \delta_{ik} + \delta_{jk})d_i(d_j + \delta_{ij})(d_k + \delta_{jk}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ij} = (1 + \delta</em>{ij})^2 d_i(d_j + \delta_{ij}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ijk} = (1 + \delta</em>{ij})(1 + \delta_{ik})d_i(d_j + \delta_{ij})(d_k + \delta_{jk}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes Yes Yes ( \mathcal{P}<em>{ijk} = (1 + \delta</em>{ij})(1 + \delta_{jk})(2 - \delta_{ik})d_i(d_j + \delta_{ij}) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\*Derivatives: \( T \left[ \frac{d}{dT} (\ln Q^m_c) \right] = \sum_j \ln Q_{c_j} S_j \) and \( T^2 \left[ \frac{d^2}{dT^2} (\ln Q^m_c) \right] \)

\( = \sum_j \ln Q_{c_j} \left[ \sum_l m_{ij}^2 b_i^l b_j^l \left( (h_i^l n_i^l + 1) - 2s_j + s_j^2 - n_j^2 \right) \right] \) where \( \ln Q^m_c = \sum_j \ln Q_{c_j} \) and \( \ln Q_{c_j} \) is any term in formulas 8 to 27 which has the formula \( \ln Q_{c_j} = (c_j/T)^{p_j} C_j \sum_l n_i^l b_i^l \) where \( p_j = 0, 1, \) or 2; \( C_j \) is a constant; \( n_1 \) and \( m_1 \) are integer exponents; and \( b_i \) is an integer subscript, and where \( S_j = \sum_l n_i^l (b_i^l + m_1^l h_i^l b_i^l) - p_j \).
<table>
<thead>
<tr>
<th>Type of card</th>
<th>Contents</th>
<th>Is card optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONSTS</strong></td>
<td>Physical constants, $\frac{hc}{k}$, $R$, and $S_c$ (eqs. (4) and (5))</td>
<td>No</td>
</tr>
<tr>
<td><strong>ATOM</strong></td>
<td>Chemical symbol, atomic weight, and reference form of each element. (If FILL option is used, also include the coefficient $b$ in equation (8) and $\sum g_i$ in equation (8) for the ground state).</td>
<td>No</td>
</tr>
<tr>
<td><strong>LISTEF</strong></td>
<td>Code in card columns 1 to 6 only which calls for listing the contents of the binary EF data cards that are processed after the LISTEF card</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>EFDATA</strong></td>
<td>Chemical formula for reactant (monatomic gas or element in its reference form), the $H^0$ value, the melting point if any, and the number of temperatures for which there are binary EF data following this card</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Binary EF data</strong></td>
<td>Enthalpy and free energy data for the reactants. These data for each reactant consist of a set of column binary cards; the number of cards depends on the amount of data. Each set must be preceded by the EFDATA card which identifies it (see previous card).</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**General data**

**Specific data**

- **Formula**: Chemical formula of species (This card may also contain a heat of formation and its corresponding temperature) | No |
- **TEMP**: Temperature schedule | Yes |
- **REFNCE**: Numbers to identify input data sources | Yes |
- **EFTAPE**: Code in card columns 1 to 6 only which calls for EFDATA and corresponding binary EF data cards to be punched and for the data to be put on tape | Yes |
- **LOGK**: Code in card columns 1 to 4 only which calls for tables of thermodynamic properties including $\Delta H^0_I$ and $\log K$ | Yes |
- **LSTSQS**: Temperature intervals for a least-squares fit, temperature exponents in the polynomial, and a temperature where the data are to be constrained | Yes |
- **INTERM**: Code in card columns 1 to 6 only which calls for intermediate output | Yes |
- **DATE**: Date which will be punched with least-squares coefficients | Yes |
- **METHOD**: A method code which specifies the method for obtaining thermodynamic functions; for example, RRHO or PANDK for diatomic or polyatomic gases, or READIN for reading in the functions directly | No |
- **DATA**: Data required by method given on METHOD card | No |
- **FINISH**: Code in card columns 1 to 6 only which indicates the end of a set of specific data | No |
TABLE IV. - CONTENTS OF FORMULA CARDS

<table>
<thead>
<tr>
<th>Labels 2, 3, or 4</th>
<th>Numerical value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>HF298</td>
<td>An assigned enthalpy $H^o_{298.15}$</td>
<td>Numerically equal to heat of formation at 298, 150 K</td>
</tr>
<tr>
<td>ASINDH</td>
<td>An assigned enthalpy, $H^o_T$</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DISSOC</td>
<td>Dissociation energy $(D^o_T$ or $-\Delta H^o_T$)</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DELTAH</td>
<td>Heat of formation from the assigned reference elements $(\Delta H^o_T)$</td>
<td>--------------------------</td>
</tr>
<tr>
<td>IPATOM$^b$</td>
<td>Heat of ionization from the electron and neutral atom</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>(ions only)</td>
<td>(blank) Units are cm$^{-1}$/mole</td>
</tr>
<tr>
<td>INVCM</td>
<td>(blank)</td>
<td>Units are cal/mole</td>
</tr>
<tr>
<td>CAL</td>
<td>(blank)</td>
<td>Units are kcal/mole</td>
</tr>
<tr>
<td>KCAL</td>
<td>(blank)</td>
<td>Units are eV/mole</td>
</tr>
<tr>
<td>EV</td>
<td>(blank)</td>
<td>Units are J/mole</td>
</tr>
<tr>
<td>JOULES</td>
<td>(blank)</td>
<td>Not required with HF298</td>
</tr>
<tr>
<td>T</td>
<td>Temperature</td>
<td>(a) Use only one.</td>
</tr>
</tbody>
</table>

$^a$The following are examples of IPATOM:

$\text{Cl}^+$: $\text{Cl} + \text{IPATOM} = \text{Cl}^+ + e^-$
$\text{IPATOM} = 104995.46$ cm$^{-1}$ (refs. 18 and 20)

$\text{Cl}^{++}$: $\text{Cl} + \text{IPATOM} = \text{Cl}^{++} + 2e^-$
$\text{IPATOM} = 296995.46$ cm$^{-1}$ (refs. 18 and 20)

$\text{Cl}^-$: $\text{Cl} + e^- + \text{IPATOM} = \text{Cl}^-$
$\text{IPATOM} = -3.613$ eV (ref. 34).
<table>
<thead>
<tr>
<th>Card columns 1 to 6</th>
<th>Labels</th>
<th>Numerical value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFNCE</td>
<td>Any alphabetic characters</td>
<td>Any numbers within the machine capabilities</td>
<td>Code calling for the ( H^0 ) value and ( \frac{H_T^0 - H_0^0}{RT} ) and ( \frac{(f_T^0 - H_0^0)}{RT} ) data to be put on tape and punched for future ( \log K ) and ( \Delta H_T^0 ) calculations</td>
</tr>
<tr>
<td>EFTAPE</td>
<td>(blank)</td>
<td>(blank)</td>
<td>Code calling for ( \Delta H_T^0 ) and ( \log K ) calculations</td>
</tr>
<tr>
<td>LOGK</td>
<td>(blank)</td>
<td>(blank)</td>
<td>Card calls for a least-squares fit</td>
</tr>
<tr>
<td>LSTSQS</td>
<td>T</td>
<td>Temperature ( ^0 K ) at the beginning or end of interval to be fit</td>
<td>( q_i ) values in equation (10)</td>
</tr>
<tr>
<td>EXP</td>
<td></td>
<td>Temperature exponent</td>
<td>Calls for the data at this temperature to be fitted exactly. Numerical value of ( T ) must be the same as some value in the ( T ) interval schedule. If omitted, it is assumed to be the melting point, if there is one; otherwise, ( 1000^0 K ).</td>
</tr>
<tr>
<td>TCONST</td>
<td></td>
<td>Temperature constraint, ( ^0 K )</td>
<td></td>
</tr>
<tr>
<td>INTERM</td>
<td>(blank)</td>
<td>(blank)</td>
<td>Calls for intermediate output data</td>
</tr>
<tr>
<td>DATE</td>
<td>(any six optional characters)</td>
<td>(blank)</td>
<td>Punches the label as the last word on the binary least-squares coefficient cards</td>
</tr>
<tr>
<td>TEMP</td>
<td>T</td>
<td>Temperature, ( ^0 K )</td>
<td>This may be a single value or the beginning or end of an interval</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>Temperature increment, ( ^0 K )</td>
<td>This must be preceded by a lower and followed by a higher ( T ) value. (See section TEMP card(s).)</td>
</tr>
<tr>
<td>Method code (any label)</td>
<td>Type of species</td>
<td>Labels 1, 2, 3, or 4</td>
<td>Numerical value</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
<td>---------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>READIN</td>
<td>All species</td>
<td>H298H0</td>
<td>(blank)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MELTPT</td>
<td>Melting point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REDUCE</td>
<td>(blank)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MELTPT</td>
<td>Melting point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELTAH</td>
<td>Heat of transition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELTAS</td>
<td>Entropy of transition</td>
</tr>
<tr>
<td>FIXEDN</td>
<td>Monatomic gases</td>
<td>FILL</td>
<td>(blank)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Labels 1, 2, 3, or 4</td>
<td>Numerical value</td>
<td>Comments</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>READIN</td>
<td>T</td>
<td>Temperature in °K</td>
<td>One value on each card</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>$C_P^0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CP/R</td>
<td>$C_P^0/R$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-H0</td>
<td>$H_T^0 - H_0^0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-H2</td>
<td>$H_T^0 - H_{298,15}^0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-H0/T</td>
<td>$(H_T^0 - H_0^0)/T$</td>
<td>Either one of these values on each card</td>
</tr>
<tr>
<td></td>
<td>H-H2/T</td>
<td>$(H_T^0 - H_{298,15}^0)/T$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-H0RT</td>
<td>$(H_T^0 - H_0^0)/RT$</td>
<td>Any one of these values on each card</td>
</tr>
<tr>
<td></td>
<td>H-H2RT</td>
<td>$(H_T^0 - H_{298,15}^0)/RT$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>$S_T^0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S/R</td>
<td>$S_T^0/R$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F-H0</td>
<td>$-(F_T^0 - H_0^0)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-F-H2</td>
<td>$-(F_T^0 - H_{298,15}^0)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-FH0/T</td>
<td>$-(F_T^0 - H_0^0)/T$</td>
<td>Any one of these values on each card</td>
</tr>
<tr>
<td></td>
<td>-FH2/T</td>
<td>$-(F_T^0 - H_{298,15}^0)/T$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-FH0RT</td>
<td>$-(F_T^0 - H_0^0)/RT$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-FH2RT</td>
<td>$-(F_T^0 - H_{298,15}^0)/RT$</td>
<td></td>
</tr>
<tr>
<td>COEF</td>
<td>See comments</td>
<td>---------------</td>
<td>First card may be the same as aforementioned READIN card with $C_P^0$ or $C_P^0/R$ value omitted. The data will be used in obtaining the integration constants, $a_{r+1}$ and $a_{r+2}$, in equations (10) to (12).</td>
</tr>
<tr>
<td>( T )</td>
<td>Temperature at beginning or end of temperature range</td>
<td>Two ( T ) labels must precede exponents and coefficients for the temperature range.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>( E_i(i=1,2,\ldots, \text{or } 10) )</td>
<td>( q_i ) in equation (10)</td>
<td>( )</td>
<td></td>
</tr>
<tr>
<td>( C_i(i=1,2,\ldots, \text{or } 10) )</td>
<td>( a_i ) or ( a_i/R ) in equation (10)</td>
<td>( a_i/R ) with REDUCE code in METHOD card</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
CH & \quad a_{r+1} \quad \text{(eq. (11))} \\
CH/R & \quad a_{r+1}/R \quad \text{(eq. (11))} \\
CH-HO & \quad a_{r+1} - H_0^0 \quad \text{(eq. (11))} \\
CHH0/R & \quad (a_{r+1} - H_0^0)/R \quad \text{(eq. (11))} \\
\end{align*}
\]

Use one if \( a_{r+1} \) has not been set by previous enthalpy value.

\[
\begin{align*}
CS & \quad a_{r+2} \quad \text{(eq. (12))} \\
CS/R & \quad a_{r+2}/R \quad \text{(eq. (12))} \\
\end{align*}
\]

Use one if \( a_{r+2} \) has not been set by previous entropy value.

| TPUNCH | Temperature value to be punched on coefficient cards | Calls for cards to be punched (See appendix E) |

<table>
<thead>
<tr>
<th>FIXEDN, ALLN, or TEMPER ( a )</th>
<th>IP</th>
<th>Ionization potential in ( \text{cm}^{-1} )</th>
<th>Required only with TEMPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>J ( _m ) value</td>
<td>( \epsilon_m/hc ) in ( \text{cm}^{-1} ) ( \text{(eq. (7))} )</td>
<td>J ( _m ) value</td>
<td></td>
</tr>
</tbody>
</table>

(1) does not have to be right- or left-adjusted
(2) may be integer, 0, or decimal number (if decimal, it can have only 5 or 0 to right of decimal point)
(3) must be punched if 0

\( a \)For FILL option (METHOD card) or FIXEDN, the principal quantum number for the data on each card must be in card columns 79 to 80, right-adjusted.
<table>
<thead>
<tr>
<th>Method</th>
<th>Labels</th>
<th>Numerical value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRHO, PANDK, JANAF, NRRHO1, or NRRHO2</td>
<td>1, 2, 3, or 4</td>
<td>SYMNO Symmetry number</td>
<td>Taken to be 1 if omitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STATWT Statistical weight</td>
<td>Taken to be 1 if omitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T0 T0 Use with excited electronic state.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0 B0 Be, B_0, or B value must be included for all molecules.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BE Be See comments for label B0. Use only for linear molecules.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WE (\omega_e)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WEXE (\omega_e x_e)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WHEY (\omega_e y_e)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WEZE (\omega_e z_e)</td>
<td>Diatomics only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WX4 Anharmonic constant one order higher than (\omega_e z_e)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALPHAi, (i \leq 3) (\alpha_i) (See comments for definition.)</td>
<td>Diatomics only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALFAAi (i \leq 6) (\alpha_A^i)</td>
<td>Linear polyatomics only. (B_v = B_e - \sum_{i=1}^{n \leq 6} \alpha_A^i (v_1 + \frac{d_1}{2}))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALFAij (i, j \leq 6) (\alpha_{ij})</td>
<td>Linear polyatomics only. (B_\ell = B_e - \sum_{i=1}^{n \leq 6} \alpha_{ij} (v_1 + \frac{d_1}{2}))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALFAAi (i \leq 6) (\alpha_A^i)</td>
<td>Nonlinear molecules only. (A_v = A_e - \sum_{i=1}^{n \leq 6} \alpha_A^i (v_1 + \frac{d_1}{2}))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALFABi (i \leq 6) (\alpha_B^i)</td>
<td>Nonlinear molecules only. (B_v = B_e + \sum_{i=1}^{n \leq 6} \alpha_B^i (v_1 + \frac{d_1}{2}))</td>
</tr>
</tbody>
</table>

Where \(v_i\) and \(d_i\) are the vibrational quantum number and degeneracy respectively for the \(i^{th}\) fundamental frequency.
ALFAi (1 ≤ 6)  \( \alpha_i^C \)  Nonlinear molecules only. \( C_{[v]} = C_e - \sum_{i=1}^{n=6} \alpha_i^C \left( v_i + \frac{d_i}{2} \right) \)

DE  \( D_e \)  Diatomics only

BETAi (1 ≤ 3)  \( \beta_i \)  Diatomics only, where \( D_v = D_e - \sum_{i=1}^{n=3} \beta_i (v + 1/2)^i \)

Vi(d_i) or Vi (i ≤ 20)  \( \nu_i(d_i) \) or \( \nu_i \)  \( d_i \) is degeneracy (an integer) of \( \nu_i \) and may be omitted when \( d_i = 1 \)

Xij (i ≤ 6, j ≤ 6)  \( x_{ij} \)  Polyatomics only

Yijk (i ≤ 6, j ≤ 6, k ≤ 6)  \( y_{ijk} \)  Polyatomics only

W0  \( W_0 \) (Fermi resonance constant)  Linear polyatomics only

Gii (i ≤ 6)  \( g_{ii} \)  Linear polyatomics only

D0 or D000  \( D_0 \) or \( D_{000} \)  Polyatomics only

RHO  \( \rho_i \), \( \Omega_i \)  Polyatomics only

A0  \( A_0 \)  An \( I_A \) or \( A_0 \) must be included for all nonlinear polyatomics.

C0  \( C_0 \)  An \( I_C \) or \( C_0 \) must be included for all nonlinear polyatomics.

IB  \( I_B \times 10^{39} \), (g)(cm²)  See comments for label \( B_0 \).

IA  \( I_A \times 10^{39} \), (g)(cm²)  See comments for label \( A_0 \).

IC  \( I_C \times 10^{39} \), (g)(cm²)  See comments for label \( C_0 \).

\( ^b \) For excited electronic states, the data for each state should be put on separate cards with an identifying number in card columns 79 to 80. Data cards for each state must be grouped together.
<table>
<thead>
<tr>
<th>Binary word number</th>
<th>Card 1</th>
<th>Card 2</th>
<th>Cards 3 to 5 as required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Machine assigned storage</td>
<td>Card 1</td>
<td>Same as card 1</td>
</tr>
<tr>
<td>2</td>
<td>Machine assigned storage</td>
<td>Card 2</td>
<td>Same as card 1</td>
</tr>
<tr>
<td>3</td>
<td>First 6 characters of formula</td>
<td></td>
<td>Lowest T in interval</td>
</tr>
<tr>
<td>4</td>
<td>Second 6 characters of formula</td>
<td></td>
<td>Highest T in interval</td>
</tr>
<tr>
<td>5</td>
<td>Ionization potential, if any</td>
<td></td>
<td>a_1</td>
</tr>
<tr>
<td>6</td>
<td>Lowest T in intervals or melting point if liquid</td>
<td></td>
<td>a_2</td>
</tr>
<tr>
<td>7</td>
<td>Highest T in intervals or melting point if solid</td>
<td></td>
<td>a_3</td>
</tr>
<tr>
<td>8</td>
<td>Second highest T</td>
<td></td>
<td>a_4</td>
</tr>
<tr>
<td>9</td>
<td>Highest T in interval</td>
<td></td>
<td>a_5</td>
</tr>
<tr>
<td>10</td>
<td>a_1</td>
<td></td>
<td>a_{r+1}</td>
</tr>
<tr>
<td>11</td>
<td>a_2</td>
<td></td>
<td>a_{r+2}</td>
</tr>
<tr>
<td>12</td>
<td>a_3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>a_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>a_5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>a_{r+1} (eq. 11)</td>
<td>Fourth highest T</td>
<td>a_1</td>
</tr>
<tr>
<td>16</td>
<td>a_{r+2} (eq. 12)</td>
<td>Fifth highest T</td>
<td>a_2</td>
</tr>
<tr>
<td>17</td>
<td>Third highest T</td>
<td>Fourth highest T</td>
<td>a_3</td>
</tr>
<tr>
<td>18</td>
<td>Second highest T</td>
<td></td>
<td>a_4</td>
</tr>
<tr>
<td>19</td>
<td>a_1</td>
<td></td>
<td>a_5</td>
</tr>
<tr>
<td>20</td>
<td>a_2</td>
<td></td>
<td>a_{r+1}</td>
</tr>
<tr>
<td>21</td>
<td>a_3</td>
<td></td>
<td>a_{r+2}</td>
</tr>
<tr>
<td>22</td>
<td>a_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>a_5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>a_{r+1}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enter

C10 Initialize general data.

C10 Initialize specific data.

C380-C690 (LOGK)
Calculate \( \delta_{H}, \delta_{F} \)
and LOGK for all \( T \).
List tables of properties.

C50
Has a LOGK card been read?

C50
Has an EFTAPE card been read?

C540-C570 (TABLES)
List tables of thermodynamic functions.

C660-C740 (LEAST)
Do least-squares fit. List results.
Punch coefficients.

C90-C130 (EFTAPE)
Punch EFCDATA card. Punch binary EDATA cards. Put data on tape.

C540-C570 (EFTAPE)
Read EFCDATA card. Put data on tape.

C20 (INPUT)
Read and list contents of one data card.

C20
Is this a CONS, ATOM, LISTSF, TEMP, EFTAPE, LOGK, LSTSE, INTERM, or DATE card?

C20
Is this a REFNC card?

C20
Is this a METHOD card?

C50
Has a LSTSE card been read?

C510-C530 (KELH)
Calculate \( \alpha_{K} \).

C10, C140-C160 (IDENT)
Assume card is formula card. Store card contents. From formula, calculate molecular weight and number of atoms. Also determine if gas, solid, or liquid.

C20, C140-C160 (IDENT)
Assume card is formula card. Store card contents. From formula, calculate molecular weight and number of atoms. Also determine if gas, solid, or liquid.

C20
Has \( T \) schedule been read from TEMP cards?

C40
Has \( T \) been read?

C20
Has a DATA card?

Calculate \( \alpha_{K}, \delta_{H} - \delta_{F} \), and \( \alpha_{K} \) according to method.

Store data.

Read and list contents of one card.

Figure 1. - General flow of program.
"The aeronautical and space activities of the United States shall be conducted so as to contribute .... to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546