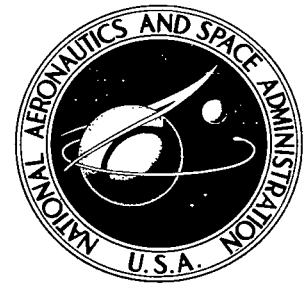
NASA TECHNICAL NOTE

NASA TN D-4097



NASA TN D-4097
c. 1

LOAN COPY: RE
AFWL
KIRTLAND AF

TECH LIBRARY KAFB, NM
0131000

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

TECH LIBRARY KAFB, NM
0131000

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

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

CONTENTS

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,

$$\frac{C_p^o}{R} = T^2 \frac{d^2(\ln Q)}{dT^2} + 2T \frac{d(\ln Q)}{dT} + \frac{5}{2} \tag{1}$$

$$\frac{H_T^o - H_0^o}{RT} = T \frac{d(\ln Q)}{dT} + \frac{5}{2} \tag{2}$$

$$\frac{S_T^o}{R} = T \frac{d(\ln Q)}{dT} + \ln Q + \frac{3}{2} \ln M + \frac{5}{2} \ln T + S_c + \frac{5}{2} \tag{3}$$

$$-\frac{F_T^o - H_0^o}{RT} = \frac{S_T^o}{R} - \frac{H_T^o - H_0^o}{RT} = \ln Q + \frac{3}{2}\ln M + \frac{5}{2}\ln T + S_c \tag{4}$$

where

$$S_c = \ln\left[k\left(\frac{2\pi k}{N_o h^2}\right)^{3/2}\right] \tag{5}$$

(Symbols are defined in appendix A).

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

$$Q = \sum_{m=1}^{L} Q^m \tag{6}$$

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

$$Q = \sum_{m=1}^{L} Q_e^m = \sum_{m=1}^{L} (2J_m + 1)e^{-\epsilon_m/kT} = \sum_{m=1}^{L} g_m e^{-\epsilon_m/kT} \tag{7}$$

where Q_e^m, J_m, ϵ_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 programed, 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)

$$\sum g_i = \sum (2J_i + 1) = bn^2 \qquad (8)$$

Atomic number	Chemical symbol	Constant in equation (8), b	Sum of statistical weights for ground state, $\sum g_i$
1	Hydrogen	2	2
2	Helium	4	1
3	Lithium	2	8
4	Beryllium	4	13
5	Boron	2	6
6	Carbon	12	15
7	Nitrogen	30	20
8	Oxygen	40	15
9	Fluorine	30	6
10	Neon	12	1
11	Sodium	2	18
12	Magnesium	4	33
13	Aluminum	2	16
14	Silicon	12	75
15	Phosphorus	30	170
16	Sulfur	40	215
17	Chlorine	30	156
18	Argon	12	61
19	Potassium	2	32
20	Calcium	4	61

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) $\sum g_i$ 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

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:

$$Q^m = Q_e^m Q_V^m Q_R^m Q_\rho^m Q_\theta^m Q_W^m Q_c^m$$

or

$$\ln Q^m = \ln Q_e^m + \ln Q_V^m + \ln Q_R^m + \ln Q_\rho^m + \ln Q_\theta^m + \ln Q_W^m + \ln Q_c^m \qquad (9)$$

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_ρ^m (ref. 25 or 26), low-temperature rigid rotation Q_θ^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 ($\ln Q_\rho^m$, $\ln Q_\theta^m$, $\ln Q_W^m$, 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:

$$C_p^o = \sum_{i=1}^{r} a_i T^{q_i} \tag{10}$$

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

$$H_T^o = a_{r+1} + \int C_p^o \, dT \tag{11}$$

$$S_T^o = a_{r+2} + \int \left(\frac{C_p^o}{T}\right) dT \tag{12}$$

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° K ($H^o_{298.15} = 0$) for a selected set of elements. This selection makes the assigned value, $H^o_{298.15}$, of any substance equal to its heat of formation at 298.15° 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, H_2, N_2, O_2, F_2, and Cl_2; 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,

$$\Delta H^{\circ}_{1000} = (H^{\circ}_{1000})_{CO(g)} - (H^{\circ}_{1000})_{C(graphite)} - \frac{1}{2}(H^{\circ}_{1000})_{O_2(g)} \tag{13}$$

$$\Delta F^{\circ}_{1000} = (F^{\circ}_{1000})_{CO(g)} - (F^{\circ}_{1000})_{C(graphite)} - \frac{1}{2}(F^{\circ}_{1000})_{O_2(g)} \tag{14}$$

or relative to gaseous atoms,

$$\Delta H^{\circ}_{1000} = (H^{\circ}_{1000})_{CO(g)} - (H^{\circ}_{1000})_{C(g)} - (H^{\circ}_{1000})_{O(g)} \tag{15}$$

$$\Delta F^{\circ}_{1000} = (F^{\circ}_{1000})_{CO(g)} - (F^{\circ}_{1000})_{C(g)} - (F^{\circ}_{1000})_{O(g)} \tag{16}$$

By definition,

$$\log_{10} K = \frac{-\Delta F^{\circ}_{T}}{2.3025851\ RT} \tag{17}$$

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, $H_T^o - H_0^o$, $H_T^o - H_{298.15}^o$, S_T^o, $-(F_T^o - H_0^o)$, and $-(F_T^o - H_{298.15}^o)$. 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

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 ω_e, $\omega_e x_e$, B_e, and α_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 EFDATA 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 ΔH_T^o 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^o is calculated, the least-squares fit option is checked, tables of thermodynamic functions are listed, and the ΔH_T^o 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^o/R, $(H_T^o - H_0^o)/RT$, $(H_T^o - H_{298.15}^o)/RT$, S_T^o/R, $-(F_T^o - H_0^o)/RT$, $-(F_T^o - H_{298.15}^o)/RT$, H_T^o/RT, and $-F_T^o/RT$

(3) Table of T, C_p^o, $H_T^o - H_0^o$, $H_T^o - H_{298.15}^o$, S_T^o, $-(F_T^o - H_0^o)$, $-(F_T^o - H_{298.15}^o)$, H_T^o, and $-F_T^o$

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^o/R, $(H_T^o - H_0^o)/RT$, S_T^o/R, $-(F_T^o - H_0^o)/RT$, H_T^o/RT, $-F_T^o/RT$, and $\Delta H_T^o/RT$ and $-\Delta F_T^o/RT$ for formation of the species from both assigned reference elements and monatomic gases

(b) Table of T, C_p^o, $H_T^o - H_0^o$, S_T^o, $-F_T^o$, and ΔH_T^o 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^o 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 (ΔH_T^o of formation, D_T^o, H_T^o, T) is used in calculating the H_0^o 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^o 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 $H_{298.15}^o - H_0^o$ value which is required in obtaining $H_T^o - H_{298.15}^o$ and $-(F_T^o - H_{298.15}^o)$, and (2) the H_0^o value which is required in obtaining H_T^o and $-F_T^o$.

LOGK(C580 to C650). - This routine is called only if a LOGK option card has been included in the input. It calculates $\Delta H_T^o/RT$, ΔH_T^o, $-\Delta F_T^o/RT$, 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 ν_i
F_T^o	$(F_T^o - H_0^o) + H_0^o$
$F_T^o - H_0^o$	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	$(H_T^o - H_0^o) + H_0^o$
$H_T^o - H_0^o$	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
l	liquid phase of chemical substance
M	molecular weight
N_o	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_c^m	correction factor to the partition function for anharmonicity and vibration-rotation interaction for m^{th} electronic state
Q_e^m	electronic partition function for m^{th} electronic state
Q_R^m	classical-rotation partition function for m^{th} electronic state
Q_V^m	harmonic-oscillator partition function for m^{th} electronic state
Q_W^m	Fermi resonance correction factor to partition function for m^{th} electronic state
Q_ρ^m	low temperature rotational correction factor to partition function for m^{th} electronic state
Q_θ^m	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_T^o	entropy for standard state
T	temperature, ^oK
T_0	electronic excitation energy between lowest vibrational states (v = 0) of ground and excited state for diatomic and polyatomic gases

u_i	$c_2\nu_i/T$
v, v_i	vibrational quantum number
W_0	Fermi resonance constant
x_{ij}, y_{ijk}	anharmonicity constants for polyatomic molecules
α_e, α_i	vibration-rotation interaction constants for diatomic and linear polyatomic molecules
$\alpha_i^A, \alpha_i^B, \alpha_i^C, \alpha_{ij}$	vibration-rotation interaction constants for polyatomic molecules
β_i	rotational-stretching - vibration interaction constant
ϵ_m	energy of m^{th} electronic state
ν_i	observed fundamental frequency
ρ	rotational-stretching spectroscopic constant
σ	symmetry number
ω_e	zero-order vibrational frequency for diatomic mólecule
$\omega_e x_e, \omega_e y_e, \omega_e z_e$	anharmonicity constants for diatomic molecules

APPENDIX B

FORTRAN LISTING (FORTRAN ROUTINES)

```
C     MAIN PROGRAM - PAC1                                                    PAC10001
C     TEST(1)  LIST EF DATA                                                  PAC10002
C     TEST(3)  SPECIES IS AN ION.                                            PAC10003
C     TEST(4)  SPECIES IS A GAS.                                             PAC10004
C     TEST(5)  SPECIES IS A LIQUID.                                          PAC10005
C     TEST(6)  SPECIES IS A SOLID.                                           PAC10006
C     TEST(7)  SUBROUTINE HFTAPE IS CALLING SUBROUTINE IDENT.                PAC10007
C     TEST(8)  AN ASSIGNED H IS AVAILABLE                                    PAC10008
C     TEST(9)  CP/R,H-HO/RT, AND S/R ARE READY TO BE OUTPUTTED               PAC10009
C     TEST(10) SPECIES TO BE REACTANT. PUNCH EF DATA AND WRITE ON TAPE.      PAC10010
C     TEST(12) LOG K CALLED FOR                                              PAC10011
C     TEST(13) DATA ARE IN THE FORM, H-H298 AND -(F-H298)                    PAC10012
C     TEST(14) INTERMEDIATE OUTPUT CALLED FOR                                PAC10013
C     TEST(15) LEAST SQUARES CALLED FOR                                      PAC10014
C     TEST(16) ERROR IN INPUT.  GO TO NEXT SPECIES                           PAC10015
C     TEST(17) PUNCH READ-IN COEFFICIENTS                                    PAC10016
C     TEST(18) ENTHALPY IS ABSOLUTE                                          PAC10017
C     TEST(19) SPECH IS SET                                                  PAC10018
C     TEST(20) TEMPERATURE SCHEDULE HAS BEEN STORED                          PAC10019
C                                                                            PAC10020
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,NEL   ,ICARD,IWORD(5),       PAC10021
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70),      PAC10022
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),     PAC10023
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,            PAC10024
     4       SPECH,TAPE(606),PTMELT,EXP(10),TRANGE(10),TCONST,NKIND,         PAC10025
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR,IHEAT,JF(5)       PAC10026
C                                                                            PAC10027
C10                                                                          PAC10028
C                                                                            PAC10029
      COMMON/PCH/LEVEL,NF1,NF2,ANS(9,15),TC(10),NTC,NFP,LDATE,NNN,NLAST      PAC10030
      INTEGER FORMLA, SYMBL, SYMBOL, ELEMNT                                  PAC10031
      LOGICAL TEST                                                           PAC10032
      EQUIVALENCE (X,IX)                                                     PAC10033
C                                                                            PAC10034
C  INITIALIZE ONCE.                                                          PAC10035
      WRITE(6,3)                                                             PAC10036    1
    3 FORMAT(1H1)                                                            PAC10037
      TEST(1) = .FALSE.                                                      PAC10038
      R = 0                                                                  PAC10039
      HCK = 0                                                                PAC10040
      NEL = 0                                                                PAC10041
      REWIND 3                                                               PAC10042    2
      END FILE 3                                                             PAC10043    3
      NATOM = 0                                                              PAC10044
      NDFILE = 0                                                             PAC10045
      DO  32 I = 1,106                                                       PAC10046
      MPLACE(I) = 0                                                          PAC10047
      LPLACE(I) = 0                                                          PAC10048
   32 NMLA(I) = 0                                                            PAC10049
C                                                                            PAC10050
C  INITIALIZATION FOR EACH SPECIES.  FOLLOWS FINISH CARD .                   PAC10051
  103 DO 101 I=1,10                                                          PAC10052
      EXP(I) = 0.0                                                           PAC10053
  101 TRANGE(I) = 0.0                                                        PAC10054
      TCONST = 0.0                                                           PAC10055
      DO 109  I = 3,20                                                       PAC10056
  109 TEST(I) = .FALSE.                                                      PAC10057
      LDATE = 0                                                              PAC10058
      NAME(1) = IBLNK                                                        PAC10059
      NAME(2) = IBLNK                                                        PAC10060
      IHEAT = IBLNK                                                          PAC10061
      TINTVL = 0.0                                                           PAC10062
      NT = 0                                                                 PAC10063
      NIT = 1                                                                PAC10064
      NTMP = 1                                                               PAC10065
      NNN = 1                                                                PAC10066
      ASINDT = 0.0                                                           PAC10067
      ASINDH = 0.0                                                           PAC10068
      SPECH = 0.0                                                            PAC10069
      H298HR = 0                                                             PAC10070
      LEVEL = 1                                                              PAC10071
      NTC= 0                                                                 PAC10072
      NPR = 0                                                                PAC10073
      IEX = 0                                                                PAC10074
      ITR = 0                                                                PAC10075
      NF = 0                                                                 PAC10076
      NF1 = 1                                                                PAC10077
      DO 102 I=1,202                                                         PAC10078
```

```
  102 T(I)=0.0                                                          PAC10079
      PI = 0.                                                           PAC10080
      PTMELT = 0.0                                                      PAC10081
      DATA ITEMP/4HTEMP/,METHOD/6HMETHOD/,IHFTAP/6HHFTAPE/,LSTSQS/6HLSTSPAC10082
     1QS/,ILGK/4HLOGK/,IREF/6HREFNCE/,IFINSH/6HFINISH/,INTERM/6HINTERM/ PAC10083
      DATA  IATOM/4HATOM/,ICONST/6HCONSTS/,ISCONS/6HSCONST/,IR/1HR/,    PAC10084
      DATA IBLNK /1H  /,IDATE/4HDATE/,NOLEAS/6HNOLEAS/,IHCK/3HHCK/,     PAC10085
     1IHFDAT/6HHFDATA/,IEFDAT/6HEFDATA/,IEFTAP/6HEFTAPE/,LIST/6HLISTEF/ PAC10086
C                                                                       PAC10087
C20                                                                     PAC10088
C  CALL INPUT TO READ AND WRITE CONTENTS OF ONE INPUT CARD              PAC10089
                                                                        PAC10090
C                                                                       PAC10091
  104 CALL INPUT (LINES)                                                PAC10092    52
  194 IF(ICARD.EQ.IFINSH)GO TO 111                                      PAC10093
      IF(ICARD.EQ.LIST) GO TO 2                                         PAC10094
      IF(ICARD.EQ.INTERM) GO TO 209                                     PAC10095
      IF(ICARD.EQ.IDATE)  GO TO 205                                     PAC10096
      IF(ICARD.EQ.ITEMP) GO TO 105                                      PAC10097
      IF(ICARD.EQ.METHOD) GO TO 107                                     PAC10098
      IF(ICARD.EQ.IHFTAP .OR. ICARD.EQ.IEFTAP) GO TO 110                PAC10099
      IF(ICARD.EQ.ILGK)   GO TO 319                                     PAC10100
      IF(ICARD.EQ.IREF)   GO TO 104                                     PAC10101
      IF (ICARD.EQ.NOLEAS) GO TO 106                                    PAC10102
      IF(ICARD.EQ.LSTSQS) GO TO 180                                     PAC10103
      IF (ICARD.EQ.IATOM) GO TO 13                                      PAC10104
      IF (ICARD.EQ.ICONST) GO TO 5                                      PAC10105
      IF (ICARD.EQ.IHFDAT .OR. ICARD.EQ.IEFDAT) GO TO 147               PAC10106
C                                                                       PAC10107
C  IF CC 1-6 CONTAIN NO RECOGNIZABLE CODE, ASSUME CARD CONTAINS FORMULA PAC10108
C  CALL IDENT TO ANALYZE FORMULA                                        PAC10109
      CALL IDENT                                                        PAC10110    95
      IF (TEST(16)) GO TO 152                                           PAC10111
      DATA IDELH/6HDELTAH/,IDIS/6HDISSOC/,IASH/6HASINDH/,IHT/1HT/       PAC10112
      DATA INVCM/5HINVCM/,KCAL/4HKCAL/,IEV/2HEV/,JOULES/6HJOULES/       PAC10113
      DATA ICAL/3HCAL/,IP/2HIP/,IPATOM/6HIPATOM/,IHF298/5HHF298/        PAC10114
C                                                                       PAC10115
C  STORE HEAT OF REACTION AND ASSIGNED T FROM FORMULA CARD              PAC10116
      DO 121 I = 2,4                                                    PAC10117
      IF (IWORD(I).EQ.IDELH .OR. IWORD(I).EQ.IDIS) GO TO 122            PAC10118
      IF (IWORD(I).EQ.IASH)   GO TO 122                                 PAC10119
      IF (IWORD(I).EQ.IHT)  ASINDT = WORD(I)                            PAC10120
      IF (IWORD(I).EQ.IHF298) GO TO 125                                 PAC10121
      IF (IWORD(I).EQ.IP) PI = WORD(I)                                  PAC10122
      IF (IWORD(I).NE.IPATOM) GO TO 121                                 PAC10123
      IHEAT = IDIS                                                      PAC10124
      ASINDH = -WORD(I)                                                 PAC10125
      GO TO 121                                                         PAC10126
  125 IHEAT = IASH                                                      PAC10127
      ASINDT = 298.15                                                   PAC10128
      GO TO 126                                                         PAC10129
  122 IHEAT = IWORD(I)                                                  PAC10130
  126 ASINDH = WORD(I)                                                  PAC10131
  121 CONTINUE                                                          PAC10132
      IF (IHEAT.NE.IBLNK.AND.ASINDT.EQ.0.)  TEST(19)=.TRUE.             PAC10133
      IF (IHEAT.EQ.IASH.AND.ASINDT.EQ.0.) TEST(8) = .TRUE.              PAC10134
C                                                                       PAC10135
C  CONVERT HEAT OF REACTION TO PROPER UNITS IF NECESSARY.               PAC10136
      CONV = 1.                                                         PAC10137
      DO 123 I = 2,4                                                    PAC10138
      IF(IWORD(I).EQ.INVCM ) CONV = 2.85927                             PAC10139
      IF(IWORD(I).EQ.KCAL)   CONV = 1000.                               PAC10140
      IF(IWORD(I).EQ.IEV)    CONV = 23063.                              PAC10141
      IF(R.GT.8.0.AND.(IWORD(I).EQ.ICAL.OR.CONV.NE.1.)) CONV=CONV*4.184 PAC10142
      IF(IWORD(I).EQ.JOULES .AND. R.LT.2.) CONV = 1./4.184              PAC10143
  123 CONTINUE                                                          PAC10144
      ASINDH = ASINDH*CONV                                              PAC10145
      GO TO 104                                                         PAC10146
C                                                                       PAC10147
C30                                                                     PAC10148
C  STORE GENERAL DATA                                                   PAC10149
C                                                                       PAC10150
    2 TEST(1) = .TRUE.                                                  PAC10151
      GO TO 104                                                         PAC10152
      DATA LTRON/6H000000E/,              MASK/07760777777777/          PAC10153
   13 X = AND (MASK,IWORD(1))                                           PAC10154
      IF (IX.EQ.IWORD(1))GO TO 20                                       PAC10155
      SYMBL = IARS(24,IWORD(1))                                         PAC10156    173
      GO TO 21                                                          PAC10157
```

```
   20 SYMBL = IARS(30,IWORD(1))                                          PAC10158    176
      IF (NATOM.EQ.0) GO TO 33                                           PAC10159
   21 DO 30  IND = 1,NATOM                                               PAC10160
      IF (SYMBL.EQ.SYMBOL(IND)) GO TO 35                                 PAC10161
   30 CONTINUE                                                           PAC10162
   33 NATOM = NATOM + 1                                                  PAC10163
      IND = NATOM                                                        PAC10164
      SYMBOL(IND) = SYMBL                                                PAC10165
   35 ATMWT(IND) = WORD(1)                                               PAC10166
      ELEMNT(IND) = IWORD(2)                                             PAC10167
      IF (SYMBL.EQ.LTRON) NEL = IND                                      PAC10168
      AG(IND) = WORD(2)                                                  PAC10169
      GG(IND) = WORD(3)                                                  PAC10170
      GO TO 104                                                          PAC10171
    5 DO 14 I=1,4                                                        PAC10172
      IF(IWORD(I).EQ.IR) R=WORD(I)                                       PAC10173
      IF(IWORD(I).EQ.IHCK) HCK=WORD(I)                                   PAC10174
      IF(IWORD(I).EQ.ISCONS) SCONST = WORD(I)                            PAC10175
   14 CONTINUE                                                           PAC10176
      GO TO 104                                                          PAC10177
  147 CALL EFTAPE                                                        PAC10178    225
      GO TO 104                                                          PAC10179
C                                                                        PAC10180
C  STORE OPTIONS.  SEE C60 FOR LSTSQS OPTION.                            PAC10181
                                                                         PAC10182
  205 DO 206 IJ=1,4                                                      PAC10183
      IF(IWORD(IJ).NE.IBLNK) LDATE = IWORD(IJ)                           PAC10184
  206 CONTINUE                                                           PAC10185
      GO TO 104                                                          PAC10186
  209 TEST(14) = .TRUE.                                                  PAC10187
      GO TO 104                                                          PAC10188
  110 TEST(10) = .TRUE.                                                  PAC10189
      GO TO 104                                                          PAC10190
  105 CALL TEMPER (NT,TINTVL        ,T,IWORD,WORD)                       PAC10191    244
      TEST(20) = .TRUE.                                                  PAC10192
      GO TO 104                                                          PAC10193
  319 TEST(12)  = .TRUE.                                                 PAC10194
      GO TO 104                                                          PAC10195
  106 TEST(15) = .FALSE.                                                 PAC10196
      ITR = 0                                                            PAC10197
      NF = 0                                                             PAC10198
      DO 2106   I=1,10                                                   PAC10199
      EXP(I) = 0.                                                        PAC10200
 2106 TRANGE(I) = 0                                                      PAC10201
      TCONST = 0.                                                        PAC10202
      GO TO 104                                                          PAC10203
C                                                                        PAC10204
C40                                                                      PAC10205
C  METHOD CARD HAS BEEN READ.                                            PAC10206
C                                                                        PAC10207
  107 DO 2000  I = 1,4                                                   PAC10208
      DATA IREAD/6HREADIN/, ICOEF/4HCOEF/                                PAC10209
      IF(IWORD(I).EQ.IREAD.OR. IWORD(I).EQ.ICOEF)  ICARD = IWORD(I)      PAC10210
      IF(IWORD(I).EQ.IP)PI = WORD(I)                                     PAC10211
 2000 CONTINUE                                                           PAC10212
      IF ( R.EQ.0. ) GO TO 150                                           PAC10213
      IF(TEST(20) .OR. ICARD.EQ.IREAD) GO TO 130                         PAC10214
C                                                                        PAC10215
C  STORE STANDARD T SCHEDULE IF NO TEMP CARDS HAVE BEEN READ.            PAC10216
      T(1) = 100.0                                                       PAC10217
      T(2) = 200.0                                                       PAC10218
      T(3) = 298.15                                                      PAC10219
      T(4) = 300.0                                                       PAC10220
      DO 131 NT = 5,61                                                   PAC10221
  131 T(NT) = T(NT-1) + 100.0                                            PAC10222
      NT =61                                                             PAC10223
C                                                                        PAC10224
C       CALL RECO FOR READIN OR COEF METHODS                             PAC10225
C       CALL ATOM FOR MONATOMIC GASES                                    PAC10226
C       CALL  POLY  FOR DIATOMIC OR POLYATOMIC GASES                     PAC10227
  130 IF (ICARD.NE.IRFAD .AND. ICARD.NE.ICOEF) GO TO 235                 PAC10228
 2001 CALL RECO                                                          PAC10229    295
      GO TO 1161                                                         PAC10230
  235 IF((HCK.EQ.0.) .OR. WEIGHT.EQ.0.) GO TO 150                        PAC10231
      IF(NOATMS.EQ.1) GO TO 148                                          PAC10232
      IF(NOATMS.GE.2) GO TO 149                                          PAC10233
```

```
  150 WRITE(6,151)                                                      PAC10234
C                                                                       PAC10235    306
  151 FORMAT (50HOERROR IN INPUT, GO TO NEXT SPECIES, C40              )PAC10236
C                                                                       PAC10237
  152 IF (ICARD.EQ.IFINSH) GO TO 88                                     PAC10238
      READ (5, 1)ICARD                                                  PAC10239    309
    1 FORMAT(A6)                                                        PAC10240
      GO TO 152                                                         PAC10241
   88 TEST(16) = .FALSE.                                                PAC10242
      LINES = LINES + 2                                                 PAC10243
      CALL PAGEID (LINES)                                               PAC10244    314
      GO TO 103                                                         PAC10245
  148 CALL ATOM                                                         PAC10246    317
      GO TO 1160                                                        PAC10247
  149 CALL POLY                                                         PAC10248    320
 1160 NIT = NT + 1                                                      PAC10249
 1161 IF (TEST(16)) GO TO 152                                           PAC10250
  161 CALL PAGEID (LINES)                                               PAC10251    326
      GO TO 194                                                         PAC10252
C                                                                       PAC10253
C50                                                                     PAC10254
C                                                                       PAC10255
  111 IF (TEST(9))  GO TO 112                                           DELH0016
      WRITE (6,163)                                                     DELH0017    330
  163 FORMAT(54HOCP/R,(H-HO)/RT,AND S/R ARE NOT READY FOR OUTPUT, C50 ) DELH0018
      GO TO 103                                                         DELH0020
  112 NLAST = NT                                                        PAC10257
C                                                                       PAC10258
C  CALL DELH TO CALCULATE HO IF NECESSARY.  DELH WILL CALL LEAST FOR    PAC10259
C       LEAST SQUARES FIT IF OPTION HAS BEEN REQUESTED.                 PAC10260
      IF(NNN.LT.NLAST) CALL DELH                                        PAC10261
C                                                                       PAC10262
C  CALL TABLES TO PUNCH FIRST TWO TABLES OF FUNCTIONS.                  PAC10263    335
 1367 CALL TABLES                                                       PAC10264
C                                                                       PAC10265
C  FOR EFTAPE OPTION, CALL HFTAPE TO PUNCH EF DATA AND PUT DATA ON TAPE.PAC10266    338
      IF (TEST(10)) CALL EFTAPE                                         PAC10267
C                                                                       PAC10268
C      IF LOGK OPTION, CALL LOGK TO CALCULATE DELTAH AND LOG K AND      PAC10269
C       PRINT TWO TABLES OF PROPERTIES.                                 PAC10270    341
  367 IF (TEST(12))  CALL LOGK                                          PAC10271    345
      GO TO 103                                                         PAC10272
C                                                                       PAC10273
C60                                                                     PAC10274
C  STORE DATA FROM LSTSQS CARD.                                         PAC10275
C                                                                       PAC10276
      DATA ITCONS/6HTCONST/,IEXP/3HEXP/                                 PAC10277
  180 TEST (15) = .TRUE.                                                PAC10278
      DO 185 I = 1,4                                                    PAC10279
      IF (IWORD(I) .EQ. IHT) GO TO 181                                  PAC10280
      IF (IWORD(I) .EQ.ITCONS)  GO TO 186                               PAC10281
      IF (IWORD(I) .EQ. IEXP) GO TO 183                                 PAC10282
      IF (IWORD(I) .EQ. IBLNK) GO TO 185                                PAC10283
      WRITE (6,187) IWORD(I), WORD(I)                                   PAC10284    367
  187 FORMAT (1HO, A6, 39H IS AN INCORRECT LABEL FOR THE NUMBER--       PAC10285
     1 E12.4, 29H.  VALUE IGNORED, C60                              )   PAC10286
      GO TO 185                                                         PAC10287
  186 TCONST = WORD(I)                                                  PAC10288
      GO TO 185                                                         PAC10289
  181 ITR = ITR + 1                                                     PAC10290
      IF(ITR .GT. 10) GO TO 182                                         PAC10291
      TRANGE(ITR) = WORD(I)                                             PAC10292
      GO TO 185                                                         PAC10293
  182 WRITE (6,184)                                                     PAC10294    382
  184 FORMAT (69HOFIRST 10 T'S ONLY WERE ACCEPTED FOR THE LEAST SQUARES PAC10295
     1ROUTINE, C60          )                                           PAC10296
      GO TO 185                                                         PAC10297
  183 NF = NF + 1                                                       PAC10298
      EXP(NF ) = WORD(I)                                                PAC10299
  185 CONTINUE                                                          PAC10300
      GO TO 104                                                         PAC10301
      END                                                               PAC10302
```

```
      SUBROUTINE INPUT(LINES)                                                  INPT0001
C                                                                              INPT0002
C70                                                                            INPT0003
C  READ AND WRITE INPUT                                                        INPT0004
C                                                                              INPT0005
      COMMON NAME(2),C(143),ICARD,IWORD(5),WORD(4)                             INPT0006
      DIMENSION FMT(12),WRD(5)                                                 INPT0007
      DATA (FMT(I),I=1,3)/15H (1HO,A6,6X,A6,/,(FMT(J),J=5,9,2)/3*6H6X,A6INPT0008
     1,/,FMT(12)/1H)/,F8/6HF15.8,/,F3/6HF15.3,/,F5/6HF15.5,/,E8/6HE15.8,INPT0009
     2/, FI2/6H2X,I2,/,F1/6HF15.0,/, IB/1H /,FB/6H9X,A6,/,                   INPT0010
     3  ISTART/  0     /,B/1H /,A2/6H2X,A2,/,F7/6HF15.7,/                    INPT0011
      IF (NAME(1).EQ.IB.OR.ISTART.NE.0)  GO TO 901                             INPT0012
      CALL PAGEID(LINES)                                                       INPT0013    5
      ISTART = 1                                                               INPT0014
      WRITE (6,FMT) ICARD,(IWORD(I), WRD(I),I=1,4),IWRD                        INPT0015    7
  901 READ(5,1) ICARD,(IWORD(I),WORD(I),I=1,4),IWORD(5)                        INPT0016   15
    1 FORMAT (2A6,F12.0,A6,F12.0,A6,F12.0,A6,F12.0,I2)                         INPT0017
      DO 904 I = 1,4                                                           INPT0018
      J = 2*I+2                                                                INPT0019
      IF (WORD(I).EQ.0.) GO TO 902                                             INPT0020
       WRD(I) = WORD(I)                                                        INPT0021
      ABSV = ABS(WORD(I))                                                      INPT0022
      FMT(J) = F8                                                              INPT0023
      IF (ABSV.GE.1.    ) FMT(J) = F7                                          INPT0024
      IF (ABSV.GE.100.  ) FMT(J) = F5                                          INPT0025
      IF (ABSV.GE.10000.)FMT(J) = F3                                           INPT0026
      IF (ABSV.LE.1.0E-3) FMT(J) = E8                                          INPT0027
      IF (AMOD(ABSV,1.).EQ.0.) FMT(J) = F1                                     INPT0028
      GO TO 904                                                                INPT0029
  902 FMT(J) = FB                                                              INPT0030
      WRD(I) = B                                                               INPT0031
  904 CONTINUE                                                                 INPT0032
      FMT (11) = A2                                                            INPT0033
      IWRD = IB                                                                INPT0034
      IF (IWORD(5) .EQ.0) GO TO 906                                            INPT0035
      FMT(11) =FI2                                                             INPT0036
      IWRD = IWORD(5)                                                          INPT0037
  906 WRITE (6,FMT) ICARD,(IWORD(I), WRD(I),I=1,4),IWRD                        INPT0038   62
      LINES = LINES + 2                                                        INPT0039
      IF (LINES.GE.55) CALL PAGEID(LINES)                                      INPT0040   72
      RETURN                                                                   INPT0041
      END                                                                      INPT0042
```

```
      SUBROUTINE PAGEID (LINES)                                          PAGE0001
C                                                                        PAGE0002
C80                                                                      PAGE0003
C  PRINTS CHEMICAL FORMULA AT BOTTOM OF PAGE AND SKIPS TO NEXT SHEET.
C                                                                        PAGE0004
      COMMON NAME(2)                                                     PAGE0005
      DATA SKIP /1H  /, ZERO /1H0/                                       PAGE0006
      SKP = ZERO                                                         PAGE0007
   50 IF (LINES .LT. 55) GO TO 400                                       PAGE0008
      IF (LINES .GT. 57) SKP = SKIP                                      PAGE0009
      WRITE (6,100) SKP, NAME(1), NAME(2), NAME(1), NAME(2)              PAGE0010    6
  100 FORMAT (A1, 2A6, 95X, 2A6)                                         PAGE0011
  200 WRITE (6,300)                                                      PAGE0012    7
  300 FORMAT (1H1 ////)                                                  PAGE0013
      LINES = 4                                                          PAGE0014
      RETURN                                                             PAGE0015
  400 WRITE (6,500)                                                      PAGE0016    9
  500 FORMAT (1H  )                                                      PAGE0017
      LINES = LINES + 1                                                  PAGE0018
      GO TO 50                                                           PAGE0019
      END                                                                PAGE0020
```

```
      SUBROUTINE EFTAPE                                                 EFTP0001
C                                                                       EFTP0002
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,NEL    ,ICARD,IWORD(5), EFTP0003
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70), EFTP0004
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),EFTP0005
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       EFTP0006
     4       SPECH,TAPE(606  ),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,  EFTP0007
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR,IHEAT,JF(5)  EFTP0008
C                                                                       EFTP0009
C90                                                                     EFTP0010
C                                                                       EFTP0011
      EQUIVALENCE       (NAM,AME)                                       EFTP0012
      LOGICAL TEST                                                      EFTP0013
      INTEGER SYMBOL, ELEMNT, FORMLA                                    EFTP0014
C                                                                       EFTP0015
C  TEST(10)--PUNCH EF DATA AND PUT DATA ON TAPE FOR REACTANT WITH       EFTP0016
C      EFTAPE CARD IN SPECIFIC DATA.                                    EFTP0017
      IF (.NOT.TEST(10)) GO TO 147                                      EFTP0018
      REWIND 3                                                          EFTP0019    4
      NDF = NDFILE + 1                                                  EFTP0020
      CALL SKFILE(3,NDF)                                                EFTP0021    6
      DATA IHO/5HHZERO/, MELTPT/6HMELTPT/,ITNO/6H T NO./                EFTP0022
      DATA IHFDAT/6HEFDATA/,IBLANK/1H /,IE/6H00000E/                    EFTP0023
      IWORD(1)= NAME(1)                                                 EFTP0024
      WORD(2) = ASINDH                                                  EFTP0025
      WORD(3) = PTMELT                                                  EFTP0026
      WORD(4) = NT                                                      EFTP0027
C                                                                       EFTP0028
C100                                                                    EFTP0029
C                                                                       EFTP0030
C  PUNCH AND LIST EFDATA CARD.                                          EFTP0031
      PUNCH    9, IHFDAT,NAME(1),IHO,ASINDH,MELTPT,PTMELT,ITNO,NT       EFTP0032    8
    9 FORMAT (    2A6,12X,2(A6,F12.4),A6,I12)                         EFTP0033
      WRITE (6,10)IHFDAT,NAME(1),IHO,ASINDH,MELTPT,PTMELT,ITNO,NT       EFTP0034    9
   10 FORMAT(1HO,A6,6X,A6,15X,2(6X,A6,F15.4),6X,A6,I15)                 EFTP0035
      NAM = IWORD(1)                                                    EFTP0036
      KX = 0                                                            EFTP0037
C                                                                       EFTP0038
C  ARRANGE DATA FOR PUNCHING BINARY EF DATA CARDS.  EACH BINARY CARD    EFTP0039
C      CONTAINS THE FORMULA (3RD WORD PHYSICALLY) AND 7 SETS OF T,      EFTP0040
C      (H-HO)/RT AND -(F-HO)/RT VALUES.                                 EFTP0041
      DO 191 I = 1,3                                                    EFTP0042
      DO 191 LX = 1, NT                                                 EFTP0043
      KX = KX + 1                                                       EFTP0044
      IF (MOD(KX,22).NE.1) GO TO 190                                    EFTP0045
      TAPE(KX) = AME                                                    EFTP0046
      KX = KX + 1                                                       EFTP0047
  190 IF (I.EQ.1) TAPE(KX) = T(LX)                                      EFTP0048
      IF (I.EQ.2) TAPE(KX) = HHRT(LX)                                   EFTP0049
      IF (I.EQ.3) TAPE(KX) = FHRT(LX)                                   EFTP0050
  191 CONTINUE                                                          EFTP0051
C                                                                       EFTP0052
C  BCDUMP IS MAP ROUTINE FOR PUNCHING BINARY CARDS.                     EFTP0053
      CALL BCDUMP (TAPE(1),TAPE(KX))                                    EFTP0054
C                                                                       EFTP0055
C                                                                       EFTP0056
C110                                                                    EFTP0057
C                                                                       EFTP0058
C READ IN BINARY EF DATA AND PUT ON TAPE 3                              EFTP0059
C     ORDER OF WORDS ON TAPE FOR EACH ELEMENT OR ATOM--                 EFTP0060
C            1.  NAME         (IWORD(1) ON EFDATA CARD)                EFTP0061
C            2.  HZERO        (WORD(2)  ON EFDATA CARD)                EFTP0062
C            3.  MELTPT       (WORD(3)  ON EFDATA CARD)                EFTP0063
C            4.  T NO.        (WORD(4)  ON EFDATA CARD)                EFTP0064
C            5.  TEMPS        (NEXT T NO. OF WORDS)                    EFTP0065
C            6.  HHRT         (NEXT T NO. OF WORDS)                    EFTP0066
C            7.  FHRT         (NEXT T NO. OF WORDS)                    EFTP0067
C                                                                       EFTP0068   40
  147 N = 3.0*WORD(4) + 0.1                                             EFTP0069
      NX = N + N/21                                                     EFTP0070
      IF (MOD(N,21).NE.0) NX=NX+1                                       EFTP0071
C                                                                       EFTP0072
C  BCREAD IS MAP ROUTINE FOR READING BINARY CARDS.                      EFTP0073
      IF (.NOT.TEST(10)) CALL BCREAD (TAPE(1), TAPE(NX))                EFTP0074   48
      NAM = IWORD(1)                                                    EFTP0075
      IX = 0                                                            EFTP0076
      DO 999 IXX=1,NX                                                   EFTP0077
      IF (TAPE(IXX).EQ.AME) GO TO 999                                   EFTP0078
      IX = IX + 1                                                       EFTP0079
      TAPE(IX) = TAPE(IXX)                                              EFTP0080
  999 CONTINUE                                                          EFTP0081
```

```
      IF (N.EQ.IX) GO TO 1100                                          EFTP0082
      WRITE (6,1105) NAM                                               EFTP0083    67
 1105 FORMAT(10HOERROR IN  ,A6,13H EFDATA, C110)                       EFTP0084
      RETURN                                                           EFTP0085
C                                                                      EFTP0086
C120                                                                   EFTP0087
C                                                                      EFTP0088
C  WRITE EF DATA ON TAPE.                                              EFTP0089
 1100 NDFILE = NDFILE + 1                                              EFTP0090
      WRITE (3) IWORD(1), WORD(2), WORD(3), WORD(4)                    EFTP0091    69
      WRITE(3) (TAPE(I),I=1,N)                                         EFTP0092    70
      END FILE 3                                                       EFTP0093
C                                                                      EFTP0094
C  TEST(1)--LISTEF CARD HAS BEEN READ.  THUS LIST EF BINARY DATA.      EFTP0095    77
      IF (.NOT.TEST(1 ))  GO TO 210                                    EFTP0096
      WRITE (6,201)                                                    EFTP0097    80
  201 FORMAT(11X,1HT,13X,8H H-HO/RT,12X,10H-(F-HO)/RT,19X,1HT,13X,8H H-HEFTP0098
     10/RT,12X,10H-(F-HO)/RT)                                          EFTP0099
      LINES = LINES+1                                                  EFTP0100
      N3 = N/3                                                         EFTP0101
      DO 205 I = 1,N3,2                                                EFTP0102
      J = N3 + I                                                       EFTP0103
      K = 2 * N3 + I                                                   EFTP0104
      LINES = LINES+1                                                  EFTP0105
      IF (LINES.GE.55)  CALL PAGEID(LINES)                             EFTP0105    90
  205 WRITE(6,202)TAPE(I),TAPE(J),TAPE(K),TAPE(I+1),TAPE(J+1),TAPE(K+1) EFTP0106    92
  202 FORMAT (F15.3,2F20.8,8X,F15.3,2F20.8)                            EFTP0107
  210 INDEX = 1                                                        EFTP0108
C130                                                                   EFTP0109
      IF (TEST(10)) GO TO 146                                          EFTP0110
      TEST(7) = .TRUE.                                                 EFTP0111
      ICARD = IWORD(1)                                                 EFTP0112
      IWORD(1) = IBLANK                                                EFTP0113
C                                                                      EFTP0114
C  TEST(7)--SUBROUTINE IDENT IS BEING CALLED FROM EFTAPE.              EFTP0115
C  SUBROUTINE IDENT IS CALLED TO DETERMINE NUMBER OF ATOMS IN REACTANT. EFTP0116
C                                                                      EFTP0117
      CALL IDENT                                                       EFTP0118    107
      NAME(1) = IBLANK                                                 EFTP0119
  146 IF (NOATMS.EQ.1.AND.TEST(4)) GO TO 141                           EFTP0120
C                                                                      EFTP0121
C  NATOM = NUMBER OF REACTANT SPECIES ON TAPE AT THIS TIME.            EFTP0122
C  SYMBOL = ATOMIC SYMBOL = FORMLA(1) FROM IDENT.                      EFTP0123
C  ELEMNT = ASSIGNED REFERENCE FORM                                    EFTP0124
C  NMLA = NUMBER OF ATOMS IN ELEMENT.                                  EFTP0125
C  MPLACE = POSITION OF MONATOMIC REACTANT SPECIES ON TAPE.            EFTP0126
C  LPLACE = POSITION OF ASSIGNED REFERENCE REACTANT ON TAPE.           EFTP0127
C                                                                      EFTP0128
      IF (NATOM.EQ.0) GO TO 142                                        EFTP0129
      DO 140  INDEX = 1,NATOM                                          EFTP0130
      IF (ICARD.EQ.ELEMNT(INDEX)) GO TO 151                            EFTP0131
  140 CONTINUE                                                         EFTP0132
      DO 150  INDEX = 1,NATOM                                          EFTP0133
      IF (FORMLA(1).EQ.SYMBOL(INDEX))GO TO 152                         EFTP0134
  150 CONTINUE                                                         EFTP0135
  142 NATOM = NATOM + 1                                                EFTP0136
      INDEX = NATOM                                                    EFTP0137
  151 SYMBOL (INDEX) = FORMLA(1)                                       EFTP0138
  152 NMLA(INDEX) = MLA(1)                                             EFTP0139
      LPLACE(INDEX) = NDFILE                                           EFTP0140
      GO TO 163                                                        EFTP0141
C                                                                      EFTP0142
C  REACTANT SPECIES IS MONATOMIC GAS.                                  EFTP0143
  141 IF (NATOM.EQ.0) GO TO 161                                        EFTP0144
      DO 160  INDEX = 1,NATOM                                          EFTP0145
      IF (FORMLA(1).EQ.SYMBOL(INDEX)) GO TO 162                        EFTP0146
  160 CONTINUE                                                         EFTP0147
  161 NATOM = NATOM + 1                                                EFTP0148
      SYMBOL(NATOM) = FORMLA(1)                                        EFTP0149
      INDEX = NATOM                                                    EFTP0150
  162 MPLACE(INDEX) = NDFILE                                           EFTP0151
C                                                                      EFTP0152
C  NEL = INDEX FOR POSITION OF ELECTRON GAS IN ARRAYS OF DATA FOR      EFTP0153
C       REACTANT SPECIES.                                              EFTP0154
      IF (FORMLA(1).NE.IE)  GO TO 163                                  EFTP0155
      NEL = INDEX                                                      EFTP0156
      NMLA(NEL) = 1                                                    EFTP0157
      LPLACE(NEL) = MPLACE(NEL)                                        EFTP0158
  163 IF (TEST(10))  RETURN                                            EFTP0159
      DO 145  I=3,7                                                    EFTP0160
  145 TEST(I) = .FALSE.                                                EFTP0161
      RETURN                                                           EFTP0162
      END
```

```
      SUBROUTINE IDENT                                                      IDEN0001
C  FROM FORMULA, DETERMINE--                                               IDEN0002
C      1)  PHASE OF SPECIES,                                               IDEN0003
C      2)  NUMBER OF ATOMS IN SPECIES,                                     IDEN0004
C      3)  MOLECULAR WEIGHT,                                               IDEN0005
C      4)  IF ION, NUMBER OF ELECTRONS ADDED OR SUBTRACTED FROM NEUTRAL    IDEN0006
C            SPECIES.                                                      IDEN0007
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,NEL    ,ICARD,IWORD(5),   IDEN0008
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),NPLUS,ELEMNT(70),   IDEN0009
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),  IDEN0010
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,         IDEN0011
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,    IDEN0012
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI  ,H298HR,IHEAT,JF(5)  IDEN0013
      DATA  IBLNK/1H /,IPLUS/6H00000+/,MINUS/6H00000-/,LFTPAR/6H00000(/,   IDEN0014
     1   IGAS/6H00000G/,LIQ/6H00000L/                                       IDEN0015
C                                                                           IDEN0016
C140                                                                        IDEN0017
C                                                                           IDEN0018
      INTEGER SYMBOL, ELEMNT, FORMLA                                        IDEN0019
      LOGICAL TEST                                                          IDEN0020
      DIMENSION IA(12),NO(11),NUM(11)                                       IDEN0021
      DO 49  I = 1,11                                                       IDEN0022
      NO(I) = 0                                                             IDEN0023
   49 NUM(I) = 0                                                            IDEN0024
C                                                                           IDEN0025
C  PUTS EACH ALPHANUMERIC CHARACTER OF FORMULA IN IA ARRAY(RT ADJUSTED)     IDEN0026
C                                                                           IDEN0027
      NAME(1) = ICARD                                                       IDEN0028
      NAME(2) = IWORD(1)                                                    IDEN0029
      J=1                                                                   IDEN0030
      DO 50 I=1,2                                                           IDEN0031
      DO 51 K=1,6                                                           IDEN0032
      IA(J)=IARS(30,NAME(I))                                                IDEN0033    16
      NAME(I) = IALS(6,NAME(I))                                             IDEN0034    19
   51 J=J+1                                                                 IDEN0035
   50 CONTINUE                                                              IDEN0036
      NAME(1) = ICARD                                                       IDEN0037
      NAME(2) = IWORD(1)                                                    IDEN0038
C                                                                           IDEN0039
C150                                                                        IDEN0040
C WHICH CHARACTERS ARE NUMBERS AND WHAT ARE THEY                            IDEN0041
C                                                                           IDEN0042
C     INO=NUMBER OF NUMBERS                                                 IDEN0043
C     NUM(I)= LOCATION OF NUMBERS IN IA ARRAY                               IDEN0044
C     NO(I)= NUMBERS IN THESE LOCATIONS                                     IDEN0045
C                                                                           IDEN0046
      WEIGHT=0.0                                                            IDEN0047
      INO = 0                                                               IDEN0048
      IONNUM = 0                                                            IDEN0049
      DO 60 N = 2,12                                                        IDEN0050
      IF (IA(N).LE. 9) GO TO 53                                             IDEN0051
      IF (IA(N).EQ.IBLNK) GO TO 54                                          IDEN0052
      IF (IA(N).EQ.IPLUS) IONNUM = IONNUM - 1                               IDEN0053
      IF (IA(N).EQ.MINUS) IONNUM = IONNUM + 1                               IDEN0054
      GO TO 60                                                              IDEN0055
   53 IF (INO.NE.0.AND.N.GT.NUM(INO)+3) GO TO 55                            IDEN0056
      INO = INO + 1                                                         IDEN0057
      NO(INO) = IA(N)                                                       IDEN0058
      NUM(INO) = N                                                          IDEN0059
   60 CONTINUE                                                              IDEN0060
C                                                                           IDEN0061
C  IF NO NUMBERS (INO=0) PROBABLY NOT A FORMULA CARD.  RETURN TO PAC1.      IDEN0062
   54 IF (INO.NE.0) GO TO 57                                                IDEN0063
   55 WRITE (6,56)                                                          IDEN0064    62
   56 FORMAT (45HOERROR IN ABOVE CARD, IGNORE CONTENTS, C150               )IDEN0065
      RETURN                                                                IDEN0066
   57 IF (IONNUM .EQ.0) GO TO 61                                            IDEN0067
C                                                                           IDEN0068
C  IONIC SPECIES,  CALCULATE CORRECTION TO MOLECULAR WEIGHT.                IDEN0069
      TEST(3) = .TRUE.                                                      IDEN0070
      FIONNO = IONNUM                                                       IDEN0071
      WEIGHT = FIONNO * ATMWT(NEL)                                          IDEN0072
      IF (NEL .NE.0) GO TO 66                                               IDEN0073
      WEIGHT = 0                                                            IDEN0074
      WRITE (6,700)                                                         IDEN0075    70
  700 FORMAT (30HOELECTRON DATA MISSING, C150         )                     IDEN0076
      GO TO 66                                                              IDEN0077
```

```
   61 NEXT = NUM(INO) + 1                                                  IDEN0078
C                                                                          IDEN0079
C  DETERMINE PHASE OF SPECIES.                                             IDEN0080
      IF(IA(NEXT).EQ.IBLNK .OR. IA(NEXT+1).EQ.IGAS) GO TO 66               IDEN0081
      IF(IA(NEXT).EQ.LFTPAR) GO TO 165                                     IDEN0082
   64 WRITE (6,65)                                                         IDEN0083
C                                                                          IDEN0084     84
   65 FORMAT(42H ERROR IN NAME,GO TO NEXT SPECIES, C150     )              IDEN0085
      TEST(16) = .TRUE.                                                    IDEN0086
      RETURN                                                               IDEN0087
  165 IF(IA(NEXT+1).EQ.LIQ)TEST(5)=.TRUE.                                  IDEN0088
      IF(IA(NEXT+1).NE.LIQ)TEST(6)=.TRUE.                                  IDEN0089
      NPLUS = NEXT + 1                                                     IDEN0090
      GO TO 67                                                             IDEN0091
   66 TEST(4)=.TRUE.                                                       IDEN0092
C                                                                          IDEN0093
C160                                                                       IDEN0094
C                                                                          IDEN0095
   67 I = 1                                                                IDEN0096
      J = 1                                                                IDEN0097
      K = 0                                                                IDEN0098
      DO 100 LMN = 1,5                                                     IDEN0099
      FORMLA(LMN) = 0                                                      IDEN0100
  100 MLA(LMN) = 0                                                         IDEN0101
      NOATMS = 0                                                           IDEN0102
C                                                                          IDEN0103
C  STORE EACH ATOMIC SYMBOL IN FORMLA(J).  NUMBER OF ATOMS IN MLA(J).      IDEN0104
   69 IF(NUM(I).EQ.(K+2))GO TO 70                                          IDEN0105
      IF(NUM(I).NE.(K+3))GO TO 64                                          IDEN0106
      FORMLA(J)=IALS(6,IA(K+1))+IA(K+2)                                    IDEN0107     115
      GO TO 71                                                             IDEN0108
   70 FORMLA(J)=IA(K+1)                                                    IDEN0109
   71 IF((NUM(I)+1).EQ.NUM(I+1))GO TO 72                                   IDEN0110
      MLA(J)=NO(I)                                                         IDEN0111
      GO TO 73                                                             IDEN0112
   72 MLA(J)=10*NO(I)+NO(I+1)                                              IDEN0113
      I=I+1                                                                IDEN0114
C                                                                          IDEN0115
C  NOATMS = TOTAL NUMBER OF ATOMS IN MOLECULE.                             IDEN0116
   73 NOATMS=NOATMS + MLA(J)                                               IDEN0117
      IF(TEST(7)) GO TO 85                                                 IDEN0118
C                                                                          IDEN0119
C FIND ATOM FORMULA IN SYMBOL TABLE                                        IDEN0120
C                                                                          IDEN0121
      DO 74 L=1,NATOM                                                      IDEN0122
      IF(FORMLA(J).EQ.SYMBOL(L))GO TO 91                                   IDEN0123
   74 CONTINUE                                                             IDEN0124
   90 WRITE (6,92)                                                         IDEN0125     151
   92 FORMAT (50HOATOM CARD MISSING OR FORMULA INCORRECT, C160          )IDEN0126
      WEIGHT = 0                                                           IDEN0127
      GO TO 85                                                             IDEN0128
C                                                                          IDEN0129
C  CALCULATE MOLECULAR WEIGHT.                                             IDEN0130
C  STORE POSITION OF ELEMENT DATA IN JF.                                   IDEN0131
   91 JF(J) = L                                                            IDEN0132
      IF (ATMWT(L).EQ.0.0) GO TO 90                                        IDEN0133
   75 WEIGHT=WEIGHT+ATMWT(L)*FLOAT(MLA(J))                                 IDEN0134
   85 IF(INO.LE.I) GO TO 88                                                IDEN0135
      K=NUM(I)                                                             IDEN0136
      I=I+1                                                                IDEN0137
      J=J+1                                                                IDEN0138
      GO TO 69                                                             IDEN0139
   88 IF (.NOT.TEST(3) .OR. NEL.EQ.0) GO TO 900                            IDEN0140
      J = J + 1                                                            IDEN0141
      JF(J) =NEL                                                           IDEN0142
      FORMLA(J) = SYMBOL(NEL)                                              IDEN0143
      MLA(J) = IONNUM                                                      IDEN0144
C                                                                          IDEN0145
C  NKIND = NUMBER OF ELEMENTS IN FORMULA.                                  IDEN0146
  900 NKIND = J                                                            IDEN0147
      IF (TEST(3) .AND. NEL .EQ. 0) WEIGHT = 0.                            IDEN0148
      IF (.NOT.TEST(7)) RETURN                                             IDEN0149
      NAME(1) = IBLNK                                                      IDEN0150
      RETURN                                                               IDEN0151
      END                                                                  IDEN0152
```

```
      SUBROUTINE TEMPER(NT,TINTVL,       T,IWORD,WORD)                    TEMP0001
C
C  STORES T SCHEDULE IN T ARRAY FROM DATA ON TEMP CARDS.
C  NT = NUMBER OF TEMPERATURES
C  TINTVL = I 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                                                                         TEMP0002
C170                                                                      TEMP0003
C                                                                         TEMP0004
      DATA IT/1HT/,I/1HI/,IBLANK/0606060606060/                           TEMP0005
      DIMENSION              T(202),IWORD(5),WORD(4)                      TEMP0006
  103 DO 120  J=1,4                                                       TEMP0011
      IF(IWORD(J).EQ.IBLANK) GO TO 120                                    TEMP0012
      IF(IWORD(J).EQ.IT) GO TO 121                                        TEMP0013
      IF (IWORD(J).EQ.I) GO TO 122                                        TEMP0014
  124 WRITE (6,123)                                                       TEMP0015    17
  123 FORMAT(35HOERROR IN LABELS ON TEMP CARD, C170)                      TEMP0016
      GO TO 139                                                           TEMP0017
  122 IF (NT.GT.0) GO TO 125                                              TEMP0018
      GO TO 124                                                           TEMP0019
  125 TINTVL = WORD(J)                                                    TEMP0020
      GO TO 120                                                           TEMP0021
  121 IF (NT.EQ.0) GO TO 126                                              TEMP0022
      IF (TINTVL.EQ.0.0) GO TO 127                                        TEMP0023
  131 IF (T(NT).GE.(298.15-.0001)) GO TO 128                              TEMP0024
      IF ((T(NT)+TINTVL).GT.(298.15+.0001)) GO TO 129                     TEMP0025
  128 NT = NT+1                                                           TEMP0026
      IF (NT.GT.202) GO TO 1140                                           TEMP0027
      T(NT) = T(NT-1)+TINTVL                                              TEMP0028
  130 IF (T(NT).GE.(WORD(J)-.0001)) GO TO 141                             TEMP0029
      GO TO 131                                                           TEMP0030
  141 TINTVL = 0.0                                                        TEMP0031
      GO TO 120                                                           TEMP0032
  129 NT = NT+1                                                           TEMP0033
      T(NT) = 298.15                                                      TEMP0034
      NT = NT+1                                                           TEMP0035
      T(NT) = T(NT-2)+TINTVL                                              TEMP0036
      GO TO 130                                                           TEMP0037
  126 NT = 1                                                              TEMP0038
      T (NT) = WORD(J)                                                    TEMP0039
      GO TO 120                                                           TEMP0040
  127 IF (T(NT).GE.(298.15-.0001)) GO TO 132                              TEMP0041
      IF (WORD(J).GT.(298.15+.0001))GO TO 133                             TEMP0042
  132 NT = NT+1                                                           TEMP0043
      IF (NT.GT.202) GO TO 1140                                           TEMP0044
      T(NT) = WORD(J)                                                     TEMP0045
      GO TO 120                                                           TEMP0046
  133 NT = NT+1                                                           TEMP0047
      T(NT) = 298.15                                                      TEMP0048
      GO TO 132                                                           TEMP0049
  120 CONTINUE                                                            TEMP0050
C                                                                         TEMP0051
C180                                                                      TEMP0052
      RETURN                                                              TEMP0053
 1140 NT = 202                                                            TEMP0054
      WRITE (6,140)                                                       TEMP0055    91
  140 FORMAT(41HONUMBER OF TEMPERATURES EXCEEDS 202, C180)                TEMP0056
      RETURN                                                              TEMP0057
C                                                                         TEMP0058
C   TEMP CARD IS BLANK--USE STANDARD TEMPERATURE RANGE                    TEMP0059
C                                                                         TEMP0060
  139 IWORD(1) = IT                                                       TEMP0061
      WORD(1) = 100.0                                                     TEMP0062
      IWORD(2) = I                                                        TEMP0063
      WORD(2) = 100.0                                                     TEMP0064
      IWORD(3) = IT                                                       TEMP0065
      WORD(3) = 6000.0                                                    TEMP0066
      GO TO 103                                                           TEMP0067
      END                                                                 TEMP0068
```

```
      SUBROUTINE RECO                                                   RECO0001
C                                                                       RECO0002
C  READIN AND COEF METHODS.                                             RECO0003
C                                                                       RECO0004
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),  RECO0005
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70), RECO0006
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),RECO0007
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       RECO0008
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,  RECO0009
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR              RECO0010
      COMMON /PCH/LEVEL,NF1,NF2,C(9,15),TC(10), NTC,NEX,LDATE,NNN,NLAST RECO0011
C                                                                       RECO0012
C190                                                                    RECO0013
C                                                                       RECO0014
      DIMENSION EX(15)                                                  RECO0015
      LOGICAL TEST, TSTRED,TSTCO                                        RECO0016
      EQUIVALENCE (AN,NN)                                               RECO0017
      DATA  IT    /2HT /,     IBLNK /1H /,      IE    /6H00000E/,       RECO0018
     1      IC    /6H00000C/, IPNCH /6HTPUNCH/, IHH1  /6HCHHO/R/,       RECO0019
     2      IHH2C /6HCHHO/R/, IHR   /4HCH/R/,   ISRC  /4HCS/R/,         RECO0020
     3      MASK  /07760777777777/,             IREDUC/6HREDUCE/        RECO0021
      DATA  IHHD1 /5HCH-HO/,  IHHD2 /5HCH-HO/,ISC/2HCS/, IH/2HCH/       RECO0022
      DATA ICP, IHHO, IHH2, IFHO, IFH2, IS, IHHOT, IHH2T, IFHOT, IFH2T, RECO0023
     1ICPR, IHHORT, IHH2RT, IFHORT, IFH2RT, ISR      /2HCP, 4HH-HO,     RECO0024
     24HH-H2, 5H-F-HO, 5H-F-H2, 1HS, 6HH-HO/T, 6HH-H2/T, 6H-FHO/T,      RECO0025
     36H-FH2/T, 4HCP/R, 6HH-HORT, 6HH-H2RT, 6H-FHORT, 6H-FH2RT, 3HS/R/, RECO0026
     4      IH29HO/6HH298HO/,IH29HO/6HH298HO/,IDELH/6HDELTAH/,IDELS     RECO0027
     5 /6HDELTAS/,ICOEF/4HCOEF/,                 MP/6HMELTPT/,          RECO0028
     6      IHHO/4HH-HO/,IHHOT/6HH-HO/T/,IHHORT/6HH-HORT/,IFHO/5H-F-HO/, RECO0029
     7 IFHOT/6HF-HO/T/,IFHORT/6H-FHORT/                                 RECO0030
      IF (LEVEL.NE.1) LEVEL=LEVEL+1                                     RECO0031
C                                                                       RECO0032
C  INITIALIZE.  NIT IS INDEX FOR NEXT T AND CORRESPONDING FUNCTIONS.    RECO0033
      NTT = NIT                                                         RECO0034
      TT = T(NIT-1)                                                     RECO0035
      HHRT(NTT) = 0.                                                    RECO0036
      FHRT(NTT) = -1.0                                                  RECO0037
      TJ1 = T(NIT)                                                      RECO0038
      TJ2 = T(NT)                                                       RECO0039
      TSTCO = .FALSE.                                                   RECO0040
      H298HO = 0.                                                       RECO0041
      TSTRED = .FALSE.                                                  RECO0042
C                                                                       RECO0043
C  STORE INFORMATION FROM METHOD CARD.                                  RECO0044
      DO 2200 I = 1,4                                                   RECO0045
      IF(IWORD(I).EQ.IH29HO.OR.IWORD(I).EQ.IH29HO) H298HO = WORD(I)     RECO0046
      IF (IWORD(I).EQ.IDELH) GO TO 2155                                 RECO0047
      IF (IWORD(I).EQ.IDELS) GO TO 2150                                 RECO0048
      IF (IWORD(I).EQ.ICOEF) TSTCO = .TRUE.                             RECO0049
C                                                                       RECO0050
C  IF REDUCE LABEL, COEFFICIENTS ARE FOR CP/R.                          RECO0051
      IF (IWORD(I).EQ.IREDUC)TSTRED=.TRUE.                              RECO0052
      IF (IWORD(I).EQ.MP) PTMELT = WORD(I)                              RECO0053
      GO TO 2200                                                        RECO0054
C                                                                       RECO0055
C  DELTAH OR DELTAS FOR HEAT OF TRANSITION.  CALCULATE H-HO/RT AND      RECO0056
C     -(F-HO)/RT FOR NEW PHASE.                                         RECO0057
 2150 WORD(I) = WORD(I)*TT                                              RECO0058
 2155 IF(NIT.GT.1)  GO TO 2156                                          RECO0059
      TEST(16)= .TRUE.                                                  RECO0060
      WRITE(6,2154)                                                     RECO0061    43
 2154 FORMAT(47HOT FOR TRANSITION UNKNOWN. GO TO NEXT SPECIES. )        RECO0062
      RETURN                                                            RECO0063
 2156 HHRT(NTT) = HHRT(NIT-1) + WORD(I)/(R*TT)                          RECO0064
      FHRT(NTT) = FHRT(NIT-1)                                           RECO0065
 2200 CONTINUE                                                          RECO0066
C                                                                       RECO0067
C200                                                                    RECO0068
C                                                                       RECO0069
      IF(H298HR.EQ.0.)  H298HR = H298HO/R                               RECO0070
      IF((HHRT(NTT).EQ.0.).OR.HHRT(NIT).EQ.HHRT(NIT-1))  GO TO 9        RECO0071
      NLAST = NIT-1                                                     RECO0072
C                                                                       RECO0073
C  IF THERE HAS BEEN A HEAT OF TRANSITION, CALL DELH TO CHECK FOR       RECO0074
C     LSTSQS OR PUNCHED COEFFICIENTS FOR PREVIOUS PHASE.                RECO0075
      CALL DELH                                                         RECO0076    62
      NNN = NIT                                                         RECO0077
      NLAST = NT                                                        RECO0078
C                                                                       RECO0079
      LEVEL=1                                                           RECO0080
    9 DO 10 I = LEVEL,9                                                 RECO0081
      DO 10 J = 1, 15                                                   RECO0082
   10 C(I,J) = 0.0                                                      RECO0083
      NEX = 0                                                           RECO0084
      NOTS = 0                                                          RECO0085
      NDT = 0                                                           RECO0086
      JT = 0                                                            RECO0087
      NFIRST = 0                                                        RECO0088
   50 CALL INPUT (LINES)                                                RECO0089    82
      NWORD = 1                                                         RECO0090
      IF (NFIRST .NE. 0) GO TO 100                                      RECO0091
C                                                                       RECO0092
C  INITIALIZE FOR FIRST CARD.                                           RECO0093
      NSUB = ICARD                                                      RECO0094
```

```
      HCOEF = 0.0                                                          RECO0095
      SCOEF = 0.0                                                          RECO0096
C                                                                          RECO0097
C  IF CC 1-6 NOT = TO CC 1-6 PREVIOUS CARD, GO TO 210 (C240).              RECO0098
  100 IF (ICARD .NE. NSUB) GO TO 210                                       RECO0099
  102 IF (TSTCO.AND.NFIRST.NE.0) GO TO 3000                                RECO0100
      NFIRST = NFIRST+1                                                    RECO0101
      IOUT = 0                                                             RECO0102
      DO 410 I = 1,4                                                       RECO0103
      IF (IWORD(I) .EQ. IT ) GO TO 15                                      RECO0104
  410 CONTINUE                                                             RECO0105
      IOUT = IT                                                            RECO0106
   11 WRITE (6,12) IOUT                                                    RECO0107    107
   12 FORMAT (31HODATA CARD WAS SKIPPED BECAUSE A6,24HVALUE WAS MISSING    RECO0108
     1, C200)                                                              RECO0109
   13 LINES = LINES + 2                                                    RECO0110
      GO TO 50                                                             RECO0111
C                                                                          RECO0112
C210                                                                       RECO0113
C                                                                          RECO0114
C  IF TWO CONSECUTIVE T LABELS, ASSUME COEFFICIENTS, GO TO 3000 (C230).    RECO0115
C                                                                          RECO0116
   15 IF (IWORD(I+1).EQ.IT) GO TO 3000                                     RECO0117
      TT = WORD(I)                                                         RECO0118
      RT = R*TT                                                            RECO0119
      IF (TT .EQ. 0.0 .OR. TSTCO) GO TO 30                                 RECO0120
C                                                                          RECO0121
C  CHECK FOR CP.                                                           RECO0122
      DO 20 I = 1,4                                                        RECO0123
      IF (IWORD(I) .EQ. ICP) GO TO 22                                      RECO0124
      IF (IWORD(I) .EQ. ICPR)GO TO 24                                      RECO0125
   20 CONTINUE                                                             RECO0126
      IOUT = ICP                                                           RECO0127
      GO TO 11                                                             RECO0128
   22 CPR(NTT) = WORD(I)/R                                                 RECO0129
      GO TO 30                                                             RECO0130
   24 CPR(NTT) = WORD(I)                                                   RECO0131
C                                                                          RECO0132
C  CHECK FOR ENTHALPY.  SKIP IF CALCULATED FROM DELTAH.                    RECO0133
   30 IF((HHRT(NTT).NE.0.).AND.NFIRST.EQ.1) GO TO 9491                     RECO0134
      DO 40 I = 1,4                                                        RECO0135
      IF(IWORD(I).EQ.IHHO   ) IWORD(I) =IHHO                               RECO0136
      IF(IWORD(I).EQ.IHHOT  ) IWORD(I) =IHHOT                              RECO0137
      IF(IWORD(I).EQ.IHHORT ) IWORD(I) =IHHORT                             RECO0138
      IF (IWORD(I).EQ.IHHO .OR. IWORD(I).EQ.IHHOT .OR. IWORD(I).EQ.        RECO0139
     1    IHHORT) GO TO 60                                                 RECO0140
      IF (IWORD(I).EQ.IHH2 .OR. IWORD(I).EQ.IHH2T .OR. IWORD(I).EQ.        RECO0141
     1    IHH2RT) GO TO 850                                                RECO0142
   40 CONTINUE                                                             RECO0143
      IOUT = IHHO                                                          RECO0144
      GO TO 70                                                             RECO0145
  850 IF (TT .NE. 0.0) GO TO 52                                            RECO0146
      IF (H298HO .EQ. 0.0 .AND. IWORD(I) .EQ. IHH2) H298HO = -WORD(I)      RECO0147
      GO TO 50                                                             RECO0148
C                                                                          RECO0149
C  H-H298 FUNCTIONS.                                                       RECO0150
   52 IF (IWORD(I) .EQ. IHH2)   HHRTT = WORD(I)/RT                         RECO0151
      IF (IWORD(I) .EQ. IHH2T)  HHRTT = WORD(I)/R                          RECO0152
      IF (IWORD(I) .EQ. IHH2RT) HHRTT = WORD(I)                            RECO0153
      IF (H298HO .EQ. 0.0) TEST(13) = .TRUE.                               RECO0154
      HHRT(NTT) = HHRTT + H298HO/RT                                        RECO0155
      GO TO 65                                                             RECO0156
C                                                                          RECO0157
C  H-HO FUNCTIONS.                                                         RECO0158
   60 IF (IWORD(I) .EQ. IHHO)   HHRT(NTT) = WORD(I)/RT                     RECO0159
      IF (IWORD(I) .EQ. IHHOT)  HHRT(NTT) = WORD(I)/R                      RECO0160
      IF (IWORD(I) .EQ. IHHORT) HHRT(NTT) = WORD(I)                        RECO0161
C                                                                          RECO0162
C  CHECK FOR T= ASINDT ON FORMULA CARD.                                    RECO0163
   65 IF (ABS(TT-ASINDT) .GT. 0.005) GO TO 70                              RECO0164
      SPECH = HHRT(NTT) * RT                                               RECO0165
      TEST(19) = .TRUE.                                                    RECO0166
C                                                                          RECO0167
C  CHECK FOR FREE ENERGY FUNCTIONS.                                        RECO0168
   70 FHRTT = -1.0                                                         RECO0169
      SR = -1.                                                             RECO0170
      DO 480 I = 1,4                                                       RECO0171
      IF(IWORD(I).EQ.IFHO   ) IWORD(I) =IFHO                               RECO0172
      IF(IWORD(I).EQ.IFHOT  ) IWORD(I) =IFHOT                              RECO0173
      IF(IWORD(I).EQ.IFHORT ) IWORD(I) =IFHORT                             RECO0174
      IF (IWORD(I).EQ.IFHO  ) FHRTT = WORD(I)/RT                           RECO0175
      IF (IWORD(I).EQ.IFHORT) FHRTT = WORD(I)                              RECO0176
      IF (IWORD(I).EQ.IFHOT ) FHRTT = WORD(I)/R                            RECO0177
      IF (IWORD(I).EQ.IFH2  ) FHRTT = (WORD(I) - H298HO) / RT              RECO0178
      IF (IWORD(I).EQ.IFH2T ) FHRTT = (WORD(I) - H298HO/TT)/ R             RECO0179
      IF (IWORD(I).EQ.IFH2RT) FHRTT =  WORD(I) - H298HO/RT                 RECO0180
      IF(((IWORD(I).EQ.IFH2).OR.(IWORD(I).EQ.IFH2T).OR.(IWORD(I).EQ.       RECO0181
     1    IFH2RT)).AND. H298HO.EQ.0.0) TEST(13) = .TRUE.                   RECO0182
      IF (IOUT.NE. IHHO) GO TO 9048                                        RECO0183
C                                                                          RECO0184
C  CHECK FOR ENTROPY FUNCTIONS.                                            RECO0185
      IF (IWORD(I).EQ.IS)     SR = WORD(I)/R                               RECO0186
      IF (IWORD(I).EQ.ISR)    SR = WORD(I)                                 RECO0187
      GO TO 480                                                            RECO0188
```

```
 9048 IF (IWORD(I).EQ.IS)  FHRTT = WORD(I)/R - HHRT(NTT)                  RECO0189
      IF (IWORD(I).EQ.ISR) FHRTT = WORD(I)   - HHRT(NTT)                  RECO0190
  480 CONTINUE                                                            RECO0191
      IF (FHRTT.EQ.(-1.0))  GO TO 1100                                    RECO0192
      FHRT(NTT) = FHRTT                                                   RECO0193
      IF (IOUT.NE.IHHO)  GO TO 9491                                       RECO0194
      IF (SR.EQ.(-1.))  GO TO 11                                          RECO0195
      HHRT(NTT)=SR-FHRTT                                                  RECO0196
      IF (ABS(TT-ASINDT) .GT. 0.005) GO TO 9491                           RECO0197
      SPECH = HHRT(NTT) * RT                                              RECO0198
C  TEST(19)--THERE IS AN ENTHALPY FOR THE ASINDT ON THE FORMULA CARD      RECO0199
C       STORED IN SPECH.                                                  RECO0200
      TEST(19) = .TRUE.                                                   RECO0201
C                                                                         RECO0202
C220                                                                      RECO0203
C                                                                         RECO0204
C  TEST(9)--THERE ARE THERMODYNAMIC FUNCTIONS FOR AT LEAST ONE T.         RECO0205
 9491 TEST(9) = .TRUE.                                                    RECO0206
      IF (NIT.EQ.NNN) GO TO 94                                            RECO0207
      IF(T(NIT-1).LT.PTMELT.AND.TT    .GT.PTMELT) GO TO 9493              RECO0208
      IF(TT.NE.T(NIT-1).OR.ABS(HHRT(NIT)-HHRT(NIT-1)).LT.(.01))GO TO 94   RECO0209
 9493 NLAST = NIT - 1                                                     RECO0210
      CALL DELH                                                           RECO0211    323
      NNN = NIT                                                           RECO0212
      NLAST = NT                                                          RECO0213
   94 IF (TSTCO) GO TO 50                                                 RECO0214
      NT = NT + 1                                                         RECO0215
      T(NT) = TT                                                          RECO0216
      NTT = NT+1                                                          RECO0217
      NIT = NTT                                                           RECO0218
      GO TO 50                                                            RECO0219
 1100 IOUT = IFHO                                                         RECO0220
      GO TO 11                                                            RECO0221
C                                                                         RECO0222
C230             PROCESS COEFFICIENTS                                     RECO0223
C                                                                         RECO0224
C  C230 TO C240--STORE CONTENTS OF DATA CARD.                             RECO0225
C  IF FHRT(NTT) IS NOT =  -1 (NTT=0,) CALCULATE INTEGRATION CONSTANTS     RECO0226
C     FROM THE ENTHALPY AND FREE ENERGY (OR S) WHICH HAVE JUST BEEN READ  RECO0227
C                                                                         RECO0228
 3000 IF (FHRT(NTT).EQ.(-1.0)) NTT = 0                                    RECO0229
      DO 200 ID = NWORD,4                                                 RECO0230
      IF (IWORD(ID) .EQ. IBLNK) GO TO 200                                 RECO0231
      IF (IWORD(ID) .EQ. IT) GO TO 110                                    RECO0232
      NDT = 1                                                             RECO0233
      IF (IWORD(ID) .EQ. IPNCH) GO TO 710                                 RECO0234
      IF (IWORD(ID) .EQ. IHHD1.OR. IWORD(ID) .EQ. IHHD2)GO TO 150         RECO0235
      IF (IWORD(ID) .EQ. IHH1 .OR. IWORD(ID) .EQ. IHH2C)GO TO 155         RECO0236
      IF (IWORD(ID) .EQ. ISC) GO TO 160                                   RECO0237
      IF (IWORD(ID) .EQ. ISRC)GO TO 165                                   RECO0238
      IF (IWORD(ID) .EQ. IH ) GO TO 140                                   RECO0239
      IF (IWORD(ID) .EQ. IHR) GO TO 145                                   RECO0240
C                                                                         RECO0241
C  ANALYZE CI (COEFFICIENTS) AND EI (EXPONENTS) LABELS.  USE NUMBER AS    RECO0242
C       INDEX TO STORE VALUES IN C AND EX ARRAYS.                         RECO0243    382
      IWD = IALS(6, IWORD(ID))                                            RECO0244
      AN = AND(MASK,IWD)                                                  RECO0245
      IF (NN .EQ. IWD) GO TO 107                                          RECO0246    387
      NN = IARS (24,IWD)                                                  RECO0247
      GO TO 108                                                           RECO0248    390
  107 NN = IARS (30,IWD)                                                  RECO0249
  108 IF (NN .GT. 15) GO TO 1018                                          RECO0250    395
      LABEL = IARS(30, IWORD(ID))                                         RECO0251
      IF (LABEL .EQ. IE) GO TO 120                                        RECO0252
      IF (LABEL .EQ. IC) GO TO 130                                        RECO0253
 1018 WRITE (6,1019) IWORD(ID), WORD(ID)                                  RECO0254    403
C                                                                         RECO0255
 1019 FORMAT (1HO,A6, 39H IS AN INCORRECT LABEL FOR THE NUMBER-- ,E16.8,  RECO0256
     1 31H.  THUS THE VALUE WAS IGNORED.           )                     RECO0257
      LINES = LINES + 3                                                   RECO0258    408
      IF (LINES .GE. 55) CALL PAGEID (LINES)                              RECO0259
      GO TO 200                                                           RECO0260
C                                                                         RECO0261
C  TEST(17)--PUNCH COEFFICIENTS                                           RECO0262
  710 TEST(17) = .TRUE.                                                   RECO0263
      NOTS = NOTS + 1                                                     RECO0264
      IF (WORD(ID) .LE. TC(NTC)) GO TO 200                                RECO0265
      NTC = NTC + 1                                                       RECO0266
      TC(NTC) = WORD(ID)                                                  RECO0267
      GO TO 200                                                           RECO0268
C                                                                         RECO0269
C  TJ1 TO TJ2 = TEMPERATURE RANGE FOR WHICH COEFFICIENTS ARE GOOD.        RECO0270
  110 IF ( NDT    .EQ. 1) GO TO 114                                       RECO0271
      IF (JT.NE.1) TJ1 = WORD(ID)                                         RECO0272
      IF (JT.EQ.1) TJ2 = WORD(ID)                                         RECO0273
      JT = 1                                                              RECO0274
      GO TO 200                                                           RECO0275
  114 NWORD = ID                                                          RECO0276
      GO TO 210                                                           RECO0277
  120 EX(NN) = WORD(ID)                                                   RECO0278
      NEX = NEX + 1                                                       RECO0279
      GO TO 200                                                           RECO0280
C                                                                         RECO0281
C  DIVIDE COEFFICIENTS BY R IF NO REDUCE LABEL ON METHOD CARD(TSTRED=F).  RECO0282
  130 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R                              RECO0283
```

```
C                                                                       REC00283
C  TEST(18)--ABSOLUTE VALUES FOR ENTHALPY.                              REC00284
      C(LEVEL,NN) = WORD(ID)                                            REC00285
      GO TO 200                                                         REC00286
  140 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R                            REC00287
  145 TEST(18) = .TRUE.                                                 REC00288
      IF (ASINDT.NE.0.) WRITE (6,146)                                   REC00289    452
  146 FORMAT (45HOENTHALPY IS ABSOLUTE--ASINDT SHOULD = 0.        )     REC00290
      LINES = LINES + 2                                                 REC00291
  155 HCOEF = WORD(ID)                                                  REC00292
      GO TO 200                                                         REC00293
  150 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R                            REC00294
      GO TO 155                                                         REC00295
  160 IF (.NOT.TSTRED) WORD(ID) = WORD(ID)/R                            REC00296
  165 SCOEF = WORD(ID)                                                  REC00297
  200 CONTINUE                                                          REC00298
      GO TO 50                                                          REC00299
C                                                                       REC00300
C240                                                                    REC00301
C                                                                       REC00302
  210 IF(.NOT.TSTCO)  GO TO 601                                         REC00303
      NF1 = 6                                                           REC00304
      NF2 = 7                                                           REC00305
  220 NT1 = NT                                                          REC00306
      IF ((ASINDT.EQ.0.).OR.TEST(19))  GO TO 240                        REC00307
      IF (ASINDT.GE.TJ1.AND.ASINDT.LE.TJ2)  GO TO 230                   REC00308
      IF (ABS(ASINDT-298.15).GT.(.01).OR.H298HR.EQ.0.)  GO TO 240       REC00309
      SPECH = H298HR*R                                                  REC00310
      TEST(19) = .TRUE.                                                 REC00311
      GO TO 240                                                         REC00312
  230 NT1 = NT + 1                                                      REC00313
      T(NT1) = ASINDT                                                   REC00314
  240 I = NIT                                                           REC00315
  241 IF (T(I).LT.TJ1 .OR. T(I).GT.TJ2) GO TO 400                       REC00316
  245 CPR(I) = 0.0                                                      REC00317
      HHRTT = 0.0                                                       REC00318
      SR = 0.0                                                          REC00319
C                                                                       REC00320
C  CALCULATE FUNCTIONS FROM EQUATION                                    REC00321
      DO 300 J = 1, NEX                                                 REC00322
  248 TEX = 1.0                                                         REC00323
C                                                                       REC00324
C  NTT = T IF AN ENTHALPY AND ENTROPY HAS BEEN READ FOR THE PURPOSE OF  REC00325
C       CALCULATING THE INTEGRATION CONSTANTS.  IF THESE VALUES HAVE NOT REC00326
C       BEEN READ, NTT = 0.                                             REC00327
      IF (NTT.EQ. 0) TT = T(I)                                          REC00328
      IF (EX(J) .NE. 0.0) TEX = TT  **EX(J)                             REC00329    511
      IF(EX(J).EQ.(-1.))HHRTT   = HHRTT   + C(LEVEL,J)*TT  *ALOG(TT)    REC00330    515
      IF(EX(J).NE.(-1.))HHRTT   = HHRTT   + C(LEVEL,J)/(EX(J)+1.0)*TEX  REC00331
      IF (I .GT. NT) GO TO 300                                          REC00332
      CPR(I) = CPR(I) + C(LEVEL,J) * TEX                                REC00333
      IF (TEX .EQ. 1.0) SR = C(LEVEL,J) * ALOG(TT  ) + SR               REC00334    530
      IF (TEX .NE. 1.0) SR = SR + C(LEVEL,J)/EX(J) * TEX                REC00335
  300 CONTINUE                                                          REC00336
      IF(NTT.EQ.0) GO TO 350                                            REC00337
      HCOEF = (HHRT(NTT) - HHRTT) *TT                                   REC00338
      SCOEF = FHRT(NTT) - SR + HHRT(NTT)                                REC00339
      NTT = 0                                                           REC00340
      GO TO 245                                                         REC00341
  350 IF (I .GT. NT) GO TO 450                                          REC00342
      HHRT(I) = HHRTT   + HCOEF/T(I)                                    REC00343
      FHRT(I) = SR + SCOEF - HHRT(I)                                    REC00344
      IF (I.LE.NT) NIT = I+1                                            REC00345
  400 IF (I .EQ. NT1) GO TO 490                                         REC00346
      I = I + 1                                                         REC00347
      GO TO 241                                                         REC00348
  450 SPECH = (HHRTT * ASINDT + HCOEF) * R                              REC00350
      TEST(19) = .TRUE.                                                 REC00351
C                                                                       REC00352
C250                                                                    REC00353
C                                                                       REC00354
C  LEVEL = INDEX FOR TEMPERATURE INTERVALS.                             REC00355
  490 C(LEVEL,NF1) = HCOEF                                              REC00356
      C(LEVEL,NF2) = SCOEF                                              REC00357
      NOTS = NOTS - 2                                                   REC00358
      IF ((NOTS.LE.0).OR.LEVEL.EQ.(NTC-1))  GO TO 500                   REC00359
      DO 216 K = 1, NOTS                                                REC00360
      LEVEL = LEVEL + 1                                                 REC00361
      IF (TJ2.LE.TC(LEVEL)) GO TO 216                                   REC00362
      DO 214 NN = 1, 7                                                  REC00363
      C(LEVEL,NN) = C(LEVEL-1,NN)                                       REC00364
      IF (TJ1.GE.TC(LEVEL)) C(LEVEL-1,NN) = 0                           REC00365
  214 CONTINUE                                                          REC00366
  216 CONTINUE                                                          REC00367
  500 IF (ICARD .NE. NSUB) GO TO 600                                    REC00368
      NDT = 0                                                           REC00369
      JT = 0                                                            REC00370
      LEVEL = LEVEL +1                                                  REC00371
      NEX = 0                                                           REC00372
      NOTS = 0                                                          REC00373
      GO TO 3000                                                        REC00374
  600 TEST(9) = .TRUE.                                                  REC00375
  601 RETURN                                                            REC00376
      END                                                               REC00377
```

```
      SUBROUTINE ATOM                                                    ATOM0001
C                                                                        ATOM0002
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),   ATOM0003
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70),  ATOM0004
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),ATOM0005
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,        ATOM0006
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,   ATOM0007
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI ,H298HR,IHEAT,JF(5) ATOM0008
C                                                                        ATOM0009
C260                                                                     ATOM0010
C                                                                        ATOM0011
      DIMENSION AJ(400),ANU(400),G(400),NN(400),TEMPJ(4),TEMPNU(4)       ATOM0012
      DIMENSION Q(203), TDQDT(203), XTDQDT(203)                          ATOM0013
      DATA    NTEMPR/6HTEMPER/,NFIX  /6HFIXEDN/, (FILL/4HFILL/,NON/3HNON/ATOM0014
      DATA IBLANK/1H /,IP/2HIP/,LG/6HGLABEL/                             ATOM0040
      LOGICAL TEST,TSTFIL,GLABEL                                         ATOM0015
      LINES = LINES + 2                                                  ATOM0018
C
C  INITIALIZE TO NO CUT-OFF AND NO FILL
      KUTOFF = NON                                                       ATOM0019
      TSTFIL = .FALSE.                                                   ATOM0020
      GLABEL = .FALSE.
C
C  CHECK FOR FILL AND CUTOFF(KUTOFF) ON METHOD CARD.
      DO 7  I=1,4                                                        ATOM0021
      IF(IWORD(I).EQ. IFILL)TSTFIL = .TRUE.                              ATOM0022
      IF (IWORD(I).EQ.LG)  GLABEL=.TRUE.
      IF(IWORD(I).EQ.NTEMPR) KUTOFF = NTEMPR                             ATOM0023
      IF(IWORD(I).NE.NFIX)  GO TO 7                                      ATOM0024
      NFIXED = WORD(I)                                                   ATOM0025
      KUTOFF = NFIX                                                      ATOM0026
    7 CONTINUE                                                           ATOM0027
      K=0                                                                ATOM0028
      ALNWT=ALOG(WEIGHT)*1.5                                             ATOM0029   26
      NFIRST = 0                                                         ATOM0030
C                                                                        ATOM0031
C270                                                                     ATOM0032
C                                                                        ATOM0033
C
C  CALL INPUT TO READ AND LIST A DATA CARD.
   10 CALL INPUT(LINES)                                                  28
      IF (NFIRST.NE.0) GO TO 12                                          ATOM0036
      NSUB = ICARD                                                       ATOM0037
      NFIRST = 1                                                         ATOM0038
   12 IF (ICARD.NE.NSUB) GO TO 50                                        ATOM0039
      DO 40  I=1,4                                                       ATOM0041
      IF(IWORD(I).NE.IP) GO TO 13                                        ATOM0042
      PI = WORD(I)                                                       ATOM0043
      GO TO 40                                                           ATOM0044
   13 IF (IWORD(I).EQ.IBLANK) GO TO 40                                   ATOM0045
      K = K+1                                                            ATOM0046
      NN(K) = IWORD(5)                                                   ATOM0047
      IF (NN(K).EQ.0 .AND. KUTOFF.NE.NON) GO TO 30                       ATOM0048
C
C  J VALUES ARE READ WITH ALPHANUMERIC FORMAT.  CHANGE TO NUMBER AND
C     STORE IN AJ ARRAY.
      HALF = 0.                                                          ATOM0049
      ISFT = 0                                                           ATOM0050
      DO 14 MLK = 1,6                                                    ATOM0051
      LOOK = IARS (30,IWORD(I))                                          ATOM0052   59
      IF (HALF.EQ.(.5)) GO TO 16                                         ATOM0053
      IWORD(I) = IALS(6,IWORD(I))                                        ATOM0054   65
      IF (LOOK.EQ. 27) HALF = .5                                         ATOM0055
      IF (LOOK.LT.10) ISFT = ISFT*10+LOOK                                ATOM0056
   14 CONTINUE                                                           ATOM0057
   16 AJ(K) = ISFT                                                       ATOM0058
      IF (LOOK .EQ. 5) AJ(K) = AJ(K) + HALF                              ATOM0059
      IF (GLABEL)AJ(K) = (AJ(K)-1.)/2.
      G(K) = 2.*AJ(K) + 1.                                               ATOM0060
      ANU(K) = WORD(I)                                                   ATOM0061
      IF ((WORD(I).EQ.0.).AND.(K.GT.1))ANU(K) = ANU(K-1)                 ATOM0062
      GO TO 40                                                           ATOM0063
   30 WRITE(6,31)IWORD(I),NN(K)                                          ATOM0064   94
   31 FORMAT(19HOERROR IN DATA--J = ,A6,6X, 7HLEVEL =,I3  )              ATOM0065
      LINES = LINES+2                                                    ATOM0066
   40 CONTINUE                                                           ATOM0067
      GO TO 10                                                           ATOM0068
   50 KLAST  = K                                                         ATOM0069
C                                                                        ATOM0070
C280                                                                     ATOM0071
C  SORT ENERGY LEVELS IN INCREASING NUMERICAL ORDER.
   75 J=1                                                                ATOM0074
   76 M=J                                                                ATOM0075
   77 DO 79 I=J;KLAST                                                    ATOM0076
      IF(ANU(M)-ANU(I))79,79,78                                          ATOM0077
   78 M=I                                                                ATOM0078
   79 CONTINUE                                                           ATOM0079
      IF(M-J) 80,81,80                                                   ATOM0080
   80 TEMPY=ANU(M)                                                       ATOM0081
      ANU(M)=ANU(J)                                                      ATOM0082
      ANU(J)=TEMPY                                                       ATOM0083
      TEMPY=G(M)                                                         ATOM0084
      G(M)=G(J)                                                          ATOM0085
```

```
      G(J)=TEMPY                                                        ATOM0086
      KTEMPY=NN(M)                                                      ATOM0087
      NN(M)=NN(J)                                                       ATOM0088
      NN(J)=KTEMPY                                                      ATOM0089
      TEMPY = AJ(M)                                                     ATOM0090
      AJ(M) = AJ(J)                                                     ATOM0091
      AJ(J) = TEMPY                                                     ATOM0092
   81 J=J+1                                                             ATOM0093
      IF (KLAST-J) 82,82,76                                             ATOM0094
   82 CONTINUE                                                          ATOM0095
      NN(KLAST+1) = 0                                                   ATOM0096
      AJ(KLAST+1) = 0.0                                                 ATOM0097
      G(KLAST+1) = 0.0                                                  ATOM0098
      ANU(KLAST+1) = 0.0                                                ATOM0099
      IF (.NOT.TEST(14))  GO TO 1087
      WRITE ( 6,1082)                                                   ATOM0100    144
 1082 FORMAT (1H0,4X,1HN,6X,1HJ,7X,1HG,5X,13HENERGY LEVEL  ,12X,1HN,6X,1ATOM0101
     1HJ,7X,1HG,5X,13HENERGY LEVEL   )                                  ATOM0102
      LINES = LINES + 2                                                 ATOM0103
      DO 1085 I = 1, KLAST , 2                                          ATOM0104
      INN = I + 1                                                       ATOM0105
      WRITE (6,1083)(NN(IN),AJ(IN),G(IN),ANU(IN), IN = I,INN)           ATOM0106    149
 1083 FORMAT (2(I6, F8.1,F8.1,F14.3, 10X))                              ATOM0107
      LINES = LINES + 1                                                 ATOM0108
 1085 IF (LINES.GE.55) CALL PAGEID(LINES)                               ATOM0109    159
 1087 IF(KUTOFF.NE.NFIX) GO TO 100                                      ATOM0110
      IF(NFIXED.GE.NN(1)) GO TO 100                                     ATOM0111
      WRITE(6,99)NN(1)                                                  ATOM0112    168
   99 FORMAT(57HOSINCE FIXEDN IS LESS THAN FIRST N, FIXEDN IS SET EQUAL ATOM0113
     1 ,I3,5H C280)                                                     ATOM0114
      NFIXED = NN(1)                                                    ATOM0115
      LINES = LINES+2                                                   ATOM0116
  100 IF (.NOT.TSTFIL) GO TO 160                                        ATOM0125
C                                                                       ATOM0117
C290                                                                    ATOM0118
C  ROUTINE FOR ASSIGNING TO LAST LEVEL OF EACH PRINCIPLE QUANTUM NUMBER,ATOM0119
C  PQN, THAT WEIGHT WHICH GIVES PQN THE TOTAL SUM OF 2J+1, OBTAINED FROMATOM0120
C  THE FORMULA A*N*N. (IGNORES PQN*S LOWER THAN GROUND STATE, AND, WHEN ATOM0121
C  NECESSARY, USES SPECIAL NUMBER FOR SUM OF 2J+1 FOR PQN OF GROUND     ATOM0122
C  STATE.                                                               ATOM0123
C                                                                       ATOM0124
      INDX = JF(1)
 1102 WRITE (6,101)                                                     ATOM0129    173
  101 FORMAT (7X,1HB,9X,1HN,3X,15HPRED. SUM(2J+1),3X,14HACT. SUM(2J+1), ATOM0130
     13X,4HDIFF,5X,9HMAX LEVEL,3X,16H2J+1, MAX LEVEL  )                 ATOM0131
      LINES = LINES + 1                                                 ATOM0132
      IF (LINES .GE.55) CALL PAGEID (LINES)                             ATOM0133    176
      K = 1                                                             ATOM0134
      NN1 = NN(1)                                                       ATOM0135
  102 KUREN = NN(K)                                                     ATOM0136
      SUM = 0.0                                                         ATOM0137
      L = 1                                                             ATOM0138
      DO 150  J=K,KLAST                                                 ATOM0139
      IF (NN(J).LT.0) GO TO 150                                         ATOM0140
      IF (NN(J) - KUREN) 110,105,110                                    ATOM0141
  105 SUM = SUM+G(J)                                                    ATOM0142
      M = J                                                             ATOM0143
      NN(J) = -NN(J)                                                    ATOM0144
      IF (J.NE.KLAST) GO TO 150                                         ATOM0145
      GO TO 115                                                         ATOM0146
  110 IF (L .NE. 1) GO TO 114                                           ATOM0147
      L = 0                                                             ATOM0148
      K = J                                                             ATOM0149
  114 IF (J .NE. KLAST) GO TO 150                                       ATOM0150
  115 IF (KUREN.EQ.NN 1 ) GO TO 120                                     ATOM0151
      TEMPY = KUREN*KUREN                                               ATOM0152
      FORM   = AG(INDX)*TEMPY                                           ATOM0153
      GO TO 125                                                         ATOM0154
  120 FORM = GG(INDX)                                                   ATOM0155
  125 DIFF = FORM  -SUM                                                 ATOM0156
      IF(KUREN.LT.NN1) DIFF = 0.0                                       ATOM0157
      NNM = -NN(M)                                                      ATOM0158
      IF (DIFF.GT.0.0) G(M) = G(M)+DIFF                                 ATOM0159
      WRITE (6,132)AG(INDX) ,NNM  ,FORM  ,SUM,DIFF,ANU(M) ,G(M)         ATOM0160    223
  132 FORMAT (F9.1,I9  ,F13.1,F17.1,F12.1,F14.4,F9.1)                   ATOM0161
      LINES = LINES + 1                                                 ATOM0162
      IF (LINES .GE.55) CALL PAGEID (LINES)                             ATOM0163    229
      IF (L .NE. 1) GO TO 102                                           ATOM0164
      GO TO 160                                                         ATOM0165
  150 CONTINUE                                                          ATOM0166
C                                                                       ATOM0167
C300                                                                    ATOM0168
C                                                                       ATOM0171
  160 IF (ASINDT.NE.0.0) GO TO 162                                      ATOM0172
      NT1 = NT                                                          ATOM0173
      GO TO 200                                                         ATOM0174
  162 NT1 = NT+1                                                        ATOM0175
      T(NT1) = ASINDT                                                   ATOM0176
C
C  M = INDEX FOR T.   K = INDEX FOR ELECTRONIC LEVELS.
  200 DO 300 M=NIT,NT1                                                  ATOM0177
  205 I = 0                                                             ATOM0178
```

```
      JJ = 1                                                              ATOM0179
C
C  CALCULATE THE PARTITION FUNCTION AND DERIVATIVES FOR EACH ELECTRONIC
C     LEVEL AND TEMPERATURE.
      DO 206  K=1,KLAST                                                   ATOM0180
  210 X=(HCK/T(M))*ANU(K)                                                 ATOM0181
  211 IF(X-85.) 214,212,212                                               ATOM0182
  212 IF (.NOT.TEST(14))  GO TO 260
      WRITE (6,213)                                                       ATOM0183    261
  213 FORMAT(50HJNOT ALL LEVELS WERE USED.  X IS GREATER THEN 85.        )ATOM0184
      LINES = LINES + 1                                                   ATOM0185
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0186    264
      GO TO 260                                                           ATOM0187
  214 IF(KUTOFF.NE.NTEMPR) GO TO 219                                      ATOM0188
      IF(PI.NE.0.   ) GO TO 215                                           ATOM0189
      WRITE(6,1215)                                                       ATOM0190    272
 1215 FORMAT(33HO IONIZATION POTENTIAL IS MISSING)                        ATOM0191
      RETURN                                                              ATOM0192
  215 THERM= PI-T(M)/HCK                                                  ATOM0193
  216 IF(ANU(K) - THERM) 240,240,217                                      ATOM0194
  217 IF (.NOT.TEST(14))  GO TO 260
      WRITE(6,218)THERM                                                   ATOM0195    280
  218 FORMAT(58HJALL LEVELS HAVE BEEN USED TO THE THERMAL BINDING ENERGYATOM0196
     1  F12.3)                                                           ATOM0197
      LINES = LINES + 1                                                   ATOM0198
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0199    283
      GO TO 260                                                           ATOM0200
  219 IF(KUTOFF.NE.NFIX) GO TO 240                                        ATOM0201
  220 IF(IABS(NN(K)).LE.NFIXED) GO TO 240                                 ATOM0202
  221 JJ = 0                                                              ATOM0203
      GO TO 206                                                           ATOM0204
  240 I = I + 1                                                           ATOM0205
      Q(I)=G(K)*EXP(-1.*X)                                                ATOM0206    297
      TDQDT(I)=Q(I)*X                                                     ATOM0207
      XTDQDT(I)=TDQDT(I)*X                                                ATOM0208
  245 IF(Q(I)-0.1E-9)246,246,206                                          ATOM0209
  246 IF(X-2.0) 206,206,247                                               ATOM0210
  247 IF(XTDQDT(I)-0.1E-9)248,248,206                                     ATOM0211
  248 IF (.NOT.TEST(14))  GO TO 260
      WRITE(6,249)                                                        ATOM0212    314
  249 FORMAT (72HJNOT ALL ASSIGNED LEVELS WERE USED, Q AND DERIVATIVES AATOM0213
     1RE TOO SMALL                )                                     ATOM0214
      LINES = LINES + 1                                                   ATOM0215
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0216    317
      GO TO 260                                                           ATOM0217
  206 CONTINUE                                                            ATOM0218
 1260 IF (.NOT.TEST(14))  GO TO 260
      IF (JJ.EQ.0) WRITE (6,222) NFIXED                                   ATOM0219    325
  222 FORMAT(28HJALL LEVELS USED THROUGH N = I5)                          ATOM0220
      IF (JJ.NE.0) WRITE (6,1262)                                         ATOM0221    327
 1262 FORMAT (50HJALL ASSIGNED LEVELS HAVE BEEN USED                    )ATOM0222
      LINES = LINES + 1                                                   ATOM0223
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0224
C                                                                         ATOM0225
C310                                                                      ATOM0226
C  CALCULATE TOTAL Q, DERIVATIVES, AND THERMODYNAMIC FUNCTIONS FOR T.
C                                                                         ATOM0227    331
  260 QSUM=0.0                                                            ATOM0228
      TDQDTS=0.0                                                          ATOM0229
      XTDQDS=0.0                                                          ATOM0230
      J=I                                                                 ATOM0231
      DO 261 II=1,J                                                       ATOM0232
      QSUM=QSUM+Q(I)                                                      ATOM0233
      TDQDTS= TDQDTS+TDQDT(I)                                             ATOM0234
      XTDQDS= XTDQDS+XTDQDT(I)                                            ATOM0235
  261 I=I-1                                                               ATOM0236
  262 IF (TDQDTS - 0.1 E-9) 263,264,264                                   ATOM0237
  263 TMP = 0.0                                                           ATOM0238
      GO TO 265                                                           ATOM0239
  264 TMP=TDQDTS/QSUM                                                     ATOM0240
  265 HHRT(M)=TMP+2.5                                                     ATOM0241
      IF(M  .GT.NT) GO TO 301                                             ATOM0242
      CPR(M)=2.5+XTDQDS/QSUM -TMP*TMP                                     ATOM0243
      FHRT(M)= ALOG(QSUM)+2.5*ALOG(T(M))+ALNWT +SCONST                    ATOM0244    354   356
      IF (.NOT.TEST(14))  GO TO 300
      WRITE (6,270) T(M),CPR(M),HHRT(M),FHRT(M)                           ATOM0245    360
  270 FORMAT (1X,1HT,F11.2,6X,4HCP/R,F12.6,6X,8HH-HO/RT,F11.6,6X,8HF-HO/ATOM0246
     1RT,F12.6)                                                           ATOM0247
      LINES = LINES + 1                                                   ATOM0248
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0249    367
      WRITE (6,271) X,QSUM,TDQDTS,XTDQDS                                  ATOM0250    368
  271 FORMAT (1X,1HX,F12.7,6X,4HQSUM,F12.7,6X,9HT(DQ/DT)S,F13.7,6X,10HXTATOM0251
     1(DQ/DT)S,F13.7)                                                     ATOM0252
      LINES = LINES + 1                                                   ATOM0253
      IF (LINES .GE.55) CALL PAGEID (LINES)                               ATOM0254    371
  300 CONTINUE                                                            ATOM0255
      GO TO 302                                                           ATOM0256
  301 SPECH = HHRT(NT1)*R*ASINDT                                          ATOM0257
C
C  TEST(19)-- ENTHALPY HAS BEEN CALCULATED FOR T  ON FORMULA CARD.
C  TEST(9)--FUNCTIONS HAVE BEEN CALCULATED.
      TEST(19) = .TRUE.
  302 TEST(9) = .TRUE.                                                    ATOM0258
      RETURN                                                              ATOM0259
      END                                                                 ATOM0260
```

```
      SUBROUTINE POLY                                                    POLY0001
C               IF TEST IS TRUE--                                        POLY0002
C     TESTW(1)  MOLECULE IS NON-LINEAR                                   POLY0003
C     TESTW(2)  RIGID ROTATOR-HARMONIC OSCILLATOR APPROXIMATION          POLY0004
C     TESTW(3)  SECOND ORDER CORRECTIONS ARE CALLED FOR                  POLY0005
C     TESTW(4)  PENNINGTON AND KOBE APPROXIMATION                        POLY0006
C     TESTW(5)  JANAF METHOD FOR DIATOMIC MOLECULES                      POLY0007
C     TESTW(6) SPECIES HAS EXCITED ELECTRONIC STATES
      COMMON NAME(2),SYMBOL(70),ATMWT(70),E,HCK,ELECTR,ICARD,IWORD(5),   POLY0008
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),ANY,ELEMNT(70),    POLY0009
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),POLY0010
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,        POLY0011
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,   POLY0012
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR,IHEAT,J5(5)   POLY0013
      COMMON /WCOMMN/ V(20),DN(20),ND(20),X(6,6),Y(6,6,6),NNU,ALFA(6),   POLY0014
     1 ALFB(6), ALFC(6), G(6), WX(6), BETA(6), A, B, C, RH, D, WF, W,    POLY0015
     2 SYM, STWT, T00, THETA(5),TESTW(6),R(20,3),S(20,3),QL(3),Q,QLN,DQ,POLY0016
     3 DDQ,LABEL,QTOT,QLNTOT,DQTOT,DDQTOT,CORT,AIJ(6,6),AIII,AI(6) ,NSUBPOLY0017
C                                                                        POLY0018
C320                                                                     POLY0019
C                                                                        POLY0020
      DIMENSION IE(5), RI(6)                                             POLY0021
      LOGICAL TESTW, TEST                                                POLY0022
      EQUIVALENCE (IWD,WD)                                               POLY0023
      DATA IRRHO/4HRRHD/,JANAF/5HJANAF/   ,NRRHO2/6HNRRAO2/              POLY0024
      DATA IPK/5HPANDK/,   BCONV/2.7988898/, BLANK/1H /,NRRAO1/6HNRRAO1/POLY0025
C                                                                        POLY0026
C  INITIALIZE FOR EACH SET OF METHOD AND DATA CARDS.                     POLY0027
      DO 10 I = NIT,NT                                                   POLY0028
      CPR(I) = 0.0                                                       POLY0029
      HHRT(I) = 0.0                                                      POLY0030
   10 FHRT(I) = 0.0                                                      POLY0031
      HHRT(NT + 1) = 0.0                                                 POLY0032
      FHRT(NT + 1) = 0.0                                                 POLY0033
      SYM = 1.0                                                          POLY0034
      DO 1005 I = 2,6                                                    POLY0035
 1005 TESTW(I) = .FALSE.                                                 POLY0036
C                                                                        POLY0037
C  CHECK METHOD                                                          POLY0038
      DO 800 I = 1,4                                                     POLY0039
      IF  (IWORD(I) .EQ. IRRHO)   GO TO 12                               POLY0040
      IF  (IWORD(I) .EQ. NRRHO2)   GO TO 13                              POLY0041
      IF  (IWORD(I) .EQ. IPK)    GO TO 14                                POLY0042
      IF  (IWORD(I) .EQ. JANAF)   GO TO 15                               POLY0043
      IF (IWORD(I).EQ.NRRAO1)  GO TO 21                                  POLY0044
  800 CONTINUE                                                           POLY0045
      WRITE (6,19)                                                       POLY0046   45
   19 FORMAT(50HOMETHOD CODE WAS NOT RECOGNIZED, USED NRRAO1, C320    )  POLY0047
   12 TESTW(2) = .TRUE.                                                  POLY0048
      GO TO 21                                                           POLY0049
   13 TESTW(3) = .TRUE.                                                  POLY0050
      GO TO 21                                                           POLY0051
   14 TESTW(4) = .TRUE.                                                  POLY0052
      GO TO 21                                                           POLY0053
   15 TESTW(5) = .TRUE.                                                  POLY0054
      IF (NOATMS.GT.2)  TESTW(2) = .TRUE.                                POLY0055
   21 WRITE (6,22)WEIGHT                                                 POLY0056   55
   22 FORMAT (15HOMOLECULAR WT.=F10.5)                                   POLY0057
      NFIRST = 0                                                         POLY0058
      LINES = LINES + 4                                                  POLY0059
C                                                                        POLY0060
C330                                                                     POLY0061
C  CALL INPUT TO READ AND PRINT CONTENTS OF INPUT CARD.                  POLY0062
C                                                                        POLY0063
   28 CALL INPUT(LINES)                                                  POLY0064   58
      IF (NFIRST .NE. 0) GO TO 1010                                      POLY0065
C                                                                        POLY0066
C  INITIALIZE FOR FIRST CARD ONLY.                                       POLY0067
      NFIRST = 1                                                         POLY0068
      NSUB = ICARD                                                       POLY0069
C                                                                        POLY0070
C  INITIALIZE FOR EACH ELECTRONIC LEVEL.                                 POLY0071
 1001 STWT = 1.0                                                         POLY0072
      T00=0.0                                                            POLY0073
      A=0.0                                                              POLY0074
      B=0.0                                                              POLY0075
      C=0.0                                                              POLY0076
      RH=0.0                                                             POLY0077
      D=0.0                                                              POLY0078
      WF=0.0                                                             POLY0079
      W=0.0                                                              POLY0080
      THETA(3) = 0.0                                                     POLY0081
      AIII= 0.0                                                          POLY0082
      DO 1002 I=1,6                                                      POLY0083
```

```
      ALFA(I)=0.0                                                       POLY0084
      ALFB(I)=0.0                                                       POLY0085
      ALFC(I)=0.0                                                       POLY0086
      RI(I) = 0.0                                                       POLY0087
      G(I)=0.0                                                          POLY0088
      WX(I)=0.0                                                         POLY0089
      BETA(I)=0.0                                                       POLY0090
      DO 1002 J=1,6                                                     POLY0091
      X(I,J)=0.0                                                        POLY0092
      AIJ(I,J) = 0.0                                                    POLY0093
 1002 CONTINUE                                                          POLY0094
      DO 1003 I=1,20                                                    POLY0095
      V(I)=0.0                                                          POLY0096
      ND(I) = 1                                                         POLY0097
 1003 DN(I) = 1.0                                                       POLY0098
      DO 1004 I=1,4                                                     POLY0099
      DO 1004 J=1,4                                                     POLY0100
      DO 1004 K=1,4                                                     POLY0101
 1004  Y(I,J,K)=0.0                                                    POLY0102
      LEVEL = IWORD(5)                                                  POLY0103
C                                                                       POLY0104
C  ASSUME LINEAR MOLECULE WITH 3N-5 FREQS.  IF THERE IS AN A OR IA      POLY0105
C      IN THE INPUT CHANGE TO 3N-6--SEE C350.                           POLY0106
      NV = 3*NOATMS - 5                                                 POLY0107
      TESTW(1) = .FALSE.                                                POLY0108
      GO TO 1015                                                        POLY0109
C                                                                       POLY0110
C  IF CARD COLUMNS 1-6 OR 79-80 ARE DIFFERENT FROM PREVIOUS CARD, GO    POLY0111
C      TO 1051 (C380).                                                  POLY0112
 1010 IF(ICARD.NE.NSUB .OR. LEVEL .NE. IWORD(5)) GO TO 1051             POLY0113
C                                                                       POLY0114
C340                                                                    POLY0115
C                                                                       POLY0116
C  SOME LABELS FOR DIATOMICS CHECKED AND VALUES STORED IN SECTION C370. POLY0141
      DATA T1/2HTO/, SYMNO/5HSYMNO/, STATWT/6HSTATWT/, IV/6H00000V/,    POLY0117
     1 NX/6H00000X/, NY/6H00000Y/, A1/2HAO/, B1/2HBO/, C1/2HCO/,        POLY0118
     2 IA/2HIA/, IB/2HIB/, RHO/3HRHO/,IG/6H00000G/, IALPHA/6HALPHA/,    POLY0119
     3 WE/2HWE/, WEXE/4HWEXE/, WEYE/4HWEYE/, WEZE/4HWEZE/, DE/2HDE/,    POLY0120
     4 IALFAB/6HALFAB/, W1/2HWO/, T2/2HTO/, A2/2HAO/, B2/2HBO/,         POLY0121
     5 C2/2HCO/, D1/2HDO/, D111/4HDOOO/, D2/2HDO/, D222/4HDOOO/,        POLY0122
     6 WX1/3HWX1/, WX2/3HWX2/, WX3/3HWX3/, WX4/3HWX4/, BETA1/5HBETA1/,  POLY0123
     7 BETA2/5HBETA2/, BETA3/5HBETA3/, W2/2HWO/, BE/2HBE/,IC/2HIC/,     POLY0124
     8 IALFAA/6HALFAA/, IALFAC/6HALFAC/,NAIJ/6H00000A/                  POLY0125
C                                                                       POLY0126
C  IN DO LOOP THRU 1050 (C370) CHECK EACH LABEL ON DATA CARD AND STORE DPOLY0127
C      ATA.                                                             POLY0128
 1015 DO 1050 ID = 1,4                                                  POLY0129
      IWD = IWORD(ID)                                                   POLY0130
      IF (WD .EQ. BLANK)  GO TO 1050                                    POLY0131
      IF (WD .EQ. T1 .OR. WD .EQ. T2) GO TO 100                         POLY0132
      IF (WD .EQ. STATWT) GO TO 102                                     POLY0133
      IF (WD .EQ. SYMNO)  GO TO 104                                     POLY0134
      IF (WD .EQ. B1 .OR. WD .EQ. B2 .OR. WD .EQ. BE) GO TO 106         POLY0135
      IF (IWORD(ID) .EQ. IB) GO TO 108                                  POLY0136
      ISHFT1 = IARS(6,IWORD(ID))                                        POLY0137    137
      IF (ISHFT1.EQ.IALPHA .OR. ISHFT1.EQ.IALFAB .OR. ISHFT1.EQ.IALFAA  POLY0138
     1 .OR. ISHFT1.EQ.IALFAC) GO TO 1030                                POLY0139
C                                                                       POLY0140
      IF (NOATMS .EQ. 2) GO TO 1045                                     POLY0142
      IF (WD .EQ. RHO) GO TO 110                                        POLY0143
      IF (WD .EQ. D1 .OR. WD .EQ. D111) GO TO 112                       POLY0144
      IF (WD .EQ. D2 .OR. WD .EQ. D222) GO TO 112                       POLY0145
      IF (IWORD(ID) .EQ. IA) GO TO 1020                                 POLY0146
      IF (IWORD(ID) .EQ. IC) GO TO 114                                  POLY0147
      IF (WD .EQ. A1 .OR. WD .EQ. A2) GO TO 1023                        POLY0148
      IF (WD .EQ. C1 .OR. WD .EQ. C2) GO TO 116                         POLY0149
      IF (WD .EQ. W1 .OR. WD .EQ. W2) GO TO 118                         POLY0150
      ISHFT5 = IARS (30, IWORD(ID))                                     POLY0151    172
      IF (ISHFT5 .EQ. IV) GO TO 1033                                    POLY0152
      IF (ISHFT5 .EQ. NX .OR. ISHFT5.EQ.NAIJ) GO TO 1040                POLY0153
      IF (ISHFT5 .EQ. NY) GO TO 1044                                    POLY0154
      IF (ISHFT5 .EQ. IG) GO TO 1025                                    POLY0155
 1018 WRITE (6,1019) IWORD(ID), WORD(ID)                                POLY0156    186
 1019 FORMAT (1HO,A6, 39H IS AN INCORRECT LABEL FOR THE NUMBER-- ,E16.8,POLY0157
     1 36H.  THUS THE VALUE WAS IGNORED, C340     )                     POLY0158
      LINES = LINES + 3                                                 POLY0159
      IF (LINES .GE. 55) CALL PAGEID (LINES)                            POLY0160    191
      GO TO 1050                                                        POLY0161
C                                                                       POLY0162
C350                                                                    POLY0163
C                                                                       POLY0164
  100 TOO = WORD(ID)                                                    POLY0165
      GO TO 1050                                                        POLY0166
```

```
  102 STWT = WORD(ID)                                                    POLY0167
      GO TO 1050                                                         POLY0168
  104 SYM = WORD(ID)                                                     POLY0169
      GO TO 1050                                                         POLY0170
  106 B = WORD(ID)                                                       POLY0171
      GO TO 1050                                                         POLY0172
  108 B = BCONV/WORD(ID)                                                 POLY0173
      GO TO 1050                                                         POLY0174
  110 RH = WORD(ID)                                                      POLY0175
      GO TO 1050                                                         POLY0176
  112 D = WORD(ID)                                                       POLY0177
      GO TO 1050                                                         POLY0178
  114 C = BCONV/WORD(ID)                                                 POLY0179
      GO TO 1050                                                         POLY0180
  116 C = WORD(ID)                                                       POLY0181
      GO TO 1050                                                         POLY0182
  118 WF = WORD(ID)                                                      POLY0183
      GO TO 1050                                                         POLY0184
C                                                                        POLY0185
C  IF IA OR A LABEL, NON-LINEAR MOLECULE.                                POLY0186
 1020 A = BCONV/WORD(ID)                                                 POLY0187
 1021 TESTW(1) = .TRUE.                                                  POLY0188
      NV = 3*NOATMS - 6                                                  POLY0189
      GO TO 1050                                                         POLY0190
 1023 A = WORD(ID)                                                       POLY0191
      GO TO 1021                                                         POLY0192
 1025 IWD = IALS(12, IWORD(ID))                                          POLY0193    233
      IWD = IARS(30, IWD)                                                POLY0194    235
      G(IWD) = WORD(ID)                                                  POLY0195
      GO TO 1050                                                         POLY0196
 1030 IBACK1 = IALS(6, ISHFT1)                                           POLY0197    241
      I = IWORD(ID) - IBACK1                                             POLY0198
      IF (I .GT. 10) I = 1                                               POLY0199
      IF (ISHFT1.EQ.IALPHA .OR. ISHFT1 .EQ.IALFAB) ALFB(I) = WORD(ID)    POLY0200
      IF (ISHFT1.EQ.IALFAA) ALFA(I) = WORD(ID)                           POLY0201
      IF (ISHFT1.EQ.IALFAC) ALFC(I) = WORD(ID)                           POLY0202
      GO TO 1050                                                         POLY0203
C                                                                        POLY0204
C360                                                                     POLY0205
C  STORE FREQUENCY AND DEGENERACY ACCORDING TO LABEL.                    POLY0206
C                                                                        POLY0207
 1033 J = 1                                                              POLY0208
      DO 1034 I = 1,5                                                    POLY0209
      IWD = IALS (6,IWD)                                                 POLY0210    264
      IE(J) = IARS(30,IWD)                                               POLY0211    266
      IF (IE(J) .EQ. 48) GO TO 1034                                      POLY0212
      J = J + 1                                                          POLY0213
 1034 CONTINUE                                                           POLY0214
      I = 1                                                              POLY0215
      KV = IE(1)                                                         POLY0216
      I = I + 1                                                          POLY0217
      IF (IE(I) .GE. 10) GO TO 1035                                      POLY0218
      KV = 10 * KV + IE(I)                                               POLY0219
      IF(KV .GT. NV) GO TO 1038                                          POLY0220
 1035 V(KV) = WORD(ID)                                                   POLY0221
 1036 I = I + 1                                                          POLY0222
      IF (I .GT. J) GO TO 1050                                           POLY0223
      IF (IE(I) .GE. 10) GO TO 1036                                      POLY0224
      DN(KV) = IE(I)                                                     POLY0225
      ND(KV) = IE(I)                                                     POLY0226
      GO TO 1050                                                         POLY0227
 1038 WRITE (6,1019) IWORD(ID), WORD(ID)                                 POLY0228    303
      TEST(16) = .TRUE.                                                  POLY0229
      RETURN                                                             POLY0230
C                                                                        POLY0231
C  STORE XIJ ACCORDING TO LABEL                                          POLY0232
 1040 IWD = IALS (6,IWORD(ID))                                           POLY0233    308
      IX = IARS (30,IWD)                                                 POLY0234    310
      IWD = IALS (6,IWD)                                                 POLY0235    312
      JX = IARS(30,IWD)                                                  POLY0236    314
      IF(ISHFT5.EQ.NX) X(IX,JX) = WORD(ID)                               POLY0237
      IF(ISHFT5.EQ.NAIJ) AIJ(IX,JX) = WORD(ID)                           POLY0238
      GO TO 1050                                                         POLY0239
C                                                                        POLY0240
C  STORE YIJK ACCORDING TO LABEL.                                        POLY0241
 1044 IWD = IALS (6,IWORD(ID))                                           POLY0242    327
      IY = IARS (30,IWD)                                                 POLY0243    329
      IWD = IALS (6,IWD)                                                 POLY0244    331
      JY = IARS (30,IWD)                                                 POLY0245    333
      IWD = IALS (6,IWD)                                                 POLY0246    335
      KY = IARS (30, IWD)                                                POLY0247    337
      Y(IY,JY,KY) = WORD(ID)                                             POLY0248
      GO TO 1050                                                         POLY0249
```

```
C                                                                       POLY0250
C370                                                                    POLY0251
C  SOME INPUT FOR DIATOMIC MOLECULES.                                   POLY0252
C                                                                       POLY0253
 1045 IF(WD .EQ. WE) GO TO 300                                          POLY0254
      IF(WD .EQ. WEXE .OR. WD .EQ. WX1) GO TO 301                       POLY0255
      IF(WD .EQ. WEYE .OR. WD .EQ. WX2) GO TO 302                       POLY0256
      IF(WD .EQ. WEZE .OR. WD .EQ. WX3) GO TO 303                       POLY0257
      IF(WD .EQ. WX4) GO TO 304                                         POLY0258
      IF( WD .EQ. BETA1) GO TO 305                                      POLY0259
      IF( WD .EQ. BETA2) GO TO 306                                      POLY0260
      IF( WD .EQ. BETA3) GO TO 307                                      POLY0261
      IF( WD .EQ. DE) GO TO 308                                         POLY0262
      GO TO 1018                                                        POLY0263
  300 W = WORD(ID)                                                      POLY0264
      GO TO 1050                                                        POLY0265
  301 WX(1) = WORD(ID)                                                  POLY0266
      GO TO 1050                                                        POLY0267
  302 WX(2) = WORD(ID)                                                  POLY0268
      GO TO 1050                                                        POLY0269
  303 WX(3) = WORD(ID)                                                  POLY0270
      GO TO 1050                                                        POLY0271
  304 WX(4) = WORD(ID)                                                  POLY0272
      GO TO 1050                                                        POLY0273
  305 BETA(1) = WORD(ID)                                                POLY0274
      GO TO 1050                                                        POLY0275
  306 BETA(2) = WORD(ID)                                                POLY0276
      GO TO 1050                                                        POLY0277
  307 BETA(3) = WORD(ID)                                                POLY0278
      GO TO 1050                                                        POLY0279
  308 D = WORD(ID)                                                      POLY0280
      GO TO 1050                                                        POLY0281
 1050 CONTINUE                                                          POLY0282
C                                                                       POLY0283
C  DATA FOR  CARD HAS BEEN STORED.  GO TO 28 (C330) TO READ NEXT CARD.  POLY0284
      GO TO 28                                                          POLY0285
C                                                                       POLY0286
C380                                                                    POLY0287
C                                                                       POLY0288
C  DATA FOR  ELECTRONIC LEVEL HAS BEEN STORED--CALCULATE SOME VARIABLES POLY0289
C     REQUIRED IN EQUATIONS.                                            POLY0290
 1051 NNU = 0                                                           POLY0291
      IF (ICARD.EQ.NSUB)   TESTW(6) = .TRUE.
      I = 0                                                             POLY0292
      IF (NOATMS .NE. 2) GO TO 1052                                     POLY0293
C                                                                       POLY0294
C  DIATOMIC MOLECULES--                                                 POLY0295
      V(1) = W - 2.0*WX(1) + 3.25*WX(2) + 5.0*WX(3) + 7.5625*WX(4)      POLY0296
                 X(1,1) = -WX(1)+4.5*WX(2)+14.5*WX(3)                   POLY0297
                      Y(1,1,1) = WX(2) + 8.*WX(3)                       POLY0298
      AI(1) = ALFB(1)-ALFB(2)-.75*ALFB(3)                               POLY0299
      AIJ(1,1) = -ALFB(2)-1.5*ALFB(3)                                   POLY0300
C                                                                       POLY0301
C  CALCULATE AND CHECK NUMBER OF FREQS. (NNU).  MAXIMUM 6 FOR NON-RRHO. POLY0302
 1052 NNU = NNU + 1                                                     POLY0303
      I = I + ND(NNU)                                                   POLY0304
      IF (V(NNU) .EQ. 0.0) GO TO 1094                                   POLY0305
      IF (I .LT. NV) GO TO 1052                                         POLY0306
      IF (I.GT.NV)  GO TO 1094                                          POLY0307
      IF(B.EQ.0.0) GO TO 1098                                           POLY0308
      IF (NNU.GT.6) TESTW(2) = .TRUE.                                   POLY0309
      IF (NOATMS.EQ.2) GO TO 2054                                       POLY0310
      IF (TESTW(2)) GO TO 1092                                          POLY0311
      IF(TESTW(1)) GO TO 1053                                           POLY0312
      GO TO 1056                                                        POLY0313
C                                                                       POLY0314
C  DIATOMIC MOLECULES.                                                  POLY0315
 2054 D=((BETA(3)*0.5+BETA(2))*0.5+BETA(1))*0.5+D                       POLY0316
      IF(D.EQ.0.0) D=(4.0*B**3)/W**2                                    POLY0317
      BEJ= B                                                            POLY0318
      B=((ALFB(3)*0.5+ALFB(2))*0.5-ALFB(1))*0.5+B                       POLY0319
      IF (.NOT.TESTW(5)) GO TO 9054                                     POLY0320
C                                                                       POLY0321
C  JANAF CORRECTIONS                                                    POLY0322
      AI(1) = AI(1)/BEJ                                                 POLY0323
      X(1,1) = X(1,1)*V(1)/W                                            POLY0324
      RH = 4.*SQRT(D/BEJ)/(HCK*V(1))                                    POLY0325    444
      GO TO 1059                                                        POLY0326
C                                                                       POLY0327
C  IF RRHO, SKIP TO 1090 (C410).                                        POLY0328
 9054 IF (TESTW(2)) GO TO 1090                                          POLY0329
C                                                                       POLY0330
C  DIATOMICS--NOT JANAF.                                                POLY0331
```

```
      IF(TESTW(4))  AI(1 ) =ALFB(1)-2.*ALFB(2)-3.25*ALFB(3)              POLY0332
      AI(1) = AI(1)/B                                                    POLY0333
      AIJ(1,1) = AIJ(1,1)/B                                              POLY0334
      AIII = -ALFB(3)/B                                                  POLY0335
      IF (TESTW(4)) AI(1) = (AI(1) + 1.)*AI(1)                           POLY0336
      GO TO 1059                                                         POLY0337
C                                                                        POLY0338
C  LINEAR POLYATOMIC MOLECULES--                                         POLY0339
 1056 DO 1058 I = 1,NNU                                                  POLY0340
      AI(I) = ALFB(I)/B                                                  POLY0341
      DO 1057 J = 1,NNU                                                  POLY0342
      CON = DN(J)/2.                                                     POLY0343
      IF (I.EQ.J) CON = DN(I)                                            POLY0344
      AIJ(I,J) = -AIJ(I,J)/B                                             POLY0345
      IF (J.LT.I) AIJ(I,J) = AIJ(J,I)                                    POLY0346
 1057 AI(I) =  AI(I) + CON*AIJ(I,J)                                      POLY0347
      IF (TESTW(4)) AI(I) = (AI(I) + 1.)*AI(I)                           POLY0348
 1058 CONTINUE                                                           POLY0349
C                                                                        POLY0350
C390                                                                     POLY0351
C  LINEAR AND DIATOMIC MOLECULES--CALULATE RHO AND THETAS.               POLY0352
C                                                                        POLY0353
 1059 IF (RH .EQ. 0.0) RH = (2.0*D)/(B**2 * HCK)                         POLY0354
      THETA(1)=(HCK*B)/3.0                                               POLY0355
      THETA(2)=THETA(1)**2*0.6                                           POLY0356
      THETA(3)=(THETA(1)*THETA(2)*4.0)/7.0                               POLY0357
      IF (.NOT.TEST(14)) GO TO 1075                                      POLY0358
      WRITE (6,2053) B, D, RH                                            POLY0359    490
 2053 FORMAT (5HOBO = F10.6,5X,4HDO = ,E13.6,5X,5HRHO = ,E13.6 )         POLY0360
      DO 2055 I = 1, NNU                                                 POLY0361
      WRITE (6,2057) I,AI(I)                                             POLY0362    493
 2057 FORMAT (4HOAI(I1,3H) = , F10.7)                                    POLY0363
 2055 WRITE (6,2056) (I,J,AIJ(I,J),J=1,NNU)                              POLY0364    495
 2056 FORMAT(1HO,6(2HA( I1,1H,I1,3H) = F10.7,5X))                        POLY0365
      GO TO 1075                                                         POLY0366
C                                                                        POLY0367
C  NON-LINEAR MOLECULES--                                                POLY0368
 1053 IF(C.EQ.0.0) GO TO 1100                                            POLY0369
      IF (TEST(14)) WRITE (6,3056) A,B,C,RH                              POLY0370    506
 3056 FORMAT ( 5HOAO = F10.6,5X,4HBO = F10.6,5X,4HCO = F10.6 ,5X,5HRHO =POLY0371
     1E15.8)                                                             POLY0372
      DO 1054 I = 1,NNU                                                  POLY0373
      AI(I) = (ALFA(I)/A + ALFB(I)/B + ALFC(I)/C)/2.                     POLY0374
      IF (TESTW(4)) AI(I) = (.5*AI(I)+1.)*AI(I)+((ALFA(I)/A)**2+(ALFB(I POLY0375
     2)/B)**2+(ALFC(I)/C)**2)/4.                                         POLY0376
      IF (TEST(14)) WRITE (6,3055) AI(I),ALFA(I),ALFB(I),ALFC(I),I       POLY0377    523
 3055 FORMAT(5HOAI = F10.7,4X,9HALPHA A = F10.7,4X,9HALPHA B = F10.7,4X,POLY0378
     1  9HALPHA C = F10.7,4X,3HI = , I1)                                 POLY0379
 1054 CONTINUE                                                           POLY0380
      ASQ=A**2                                                           POLY0381
      BSQ=B**2                                                           POLY0382
      CSQ=C**2                                                           POLY0383
      THETA(1) = (2.0*(A+B+C) - A*B/C - A*C/B - B*C/A) * (HCK/12.0)      POLY0384
      THETA(2) = (10.0*(ASQ+BSQ+CSQ) + 12.0 * (A*B + B*C + A*C)          POLY0385
     1 - 12.0*(ASQ*B/C + A*BSQ/C + BSQ*C/A + B*CSQ/A + ASQ*C/B + A*CSQ/BPOLY0386
     2 ) + 7.0*(ASQ*BSQ/CSQ + ASQ*CSQ/BSQ + BSQ*CSQ/ASQ)) *  HCK**2/480.POLY0387
      THETA(3) = 0.0                                                     POLY0388
 1075 IF (TEST(14)) WRITE (6,3075) (I,THETA(I),I=1,3)                    POLY0389    533
 3075 FORMAT (1HO 3(6HTHETA( I1,3H) = , F9.8,4X)/)                       POLY0390
C                                                                        POLY0391
C400                                                                     POLY0392
C                                                                        POLY0393
      IF(NOATMS.EQ.2) GO TO 1092                                         POLY0394
C                                                                        POLY0395
C  POLYATOMIC MOLECULES.  MAKE X AND Y MATRICES SYMMETRIC.               POLY0396
      DO 8 I = 1,NNU                                                     POLY0397
      DO 8 J = I,NNU                                                     POLY0398
      X(J,I)  = X(I,J)                                                   POLY0399
    8 CONTINUE                                                           POLY0400
      DO   2 I=1,NNU                                                     POLY0401
      DO   2 J=I,NNU                                                     POLY0402
      DO   2 L=J,NNU                                                     POLY0403
      IF (I.NE.J) GO TO 5                                                POLY0404
      IF(J-L) 4,2,4                                                      POLY0405
    5 IF (J-L) 6,4,6                                                     POLY0406
    4 Y(J,L,I)=Y(I,J,L)                                                  POLY0407
      Y(L,I,J)=Y(I,J,L)                                                  POLY0408
      GO TO 2                                                            POLY0409
    6 Y(I,L,J)=Y(I,J,L)                                                  POLY0410
      Y(J,I,L)=Y(I,J,L)                                                  POLY0411
      Y(J,L,I)=Y(I,J,L)                                                  POLY0412
      Y(L,I,J)=Y(I,J,L)                                                  POLY0413
      Y(L,J,I)=Y(I,J,L)                                                  POLY0414
```

```
    2 CONTINUE                                                          POLY0415
      IF (TEST(14)) WRITE (6,2004)(((I,J,L,Y(I,J,L), L=J,NNU), J=I,NNU),POLY0416
     1 I=1,NNU)                                                         POLY0417    590
 2004 FORMAT(5(3H Y(I1,1H,,I1,1H,,I1,3H) =,F7.3,3X))                    POLY0418
C                                                                       POLY0419
C  APPLY X CORRECTIONS FOR NRRA01 AND 2.                                POLY0420
      DO 990 I = 1, NNU                                                 POLY0421
      DO 990 J = 1, NNU                                                 POLY0422
      CY = 0.0                                                          POLY0423
      DO 910 K = 1, NNU                                                 POLY0424
      IF((K.NE.I).AND.(K.NE.J))CY=CY+DN(K)*Y(I,J,K)/2.                  POLY0425
  910 CONTINUE                                                          POLY0426
      IF(I.EQ.J)X(I,I)=X(I,I)+Y(I,I,I)*(1.5*DN(I)+3.)+CY                POLY0427
      IF(I.NE.J)X(I,J)=X(I,J)+(DN(I)+1.)*Y(I,I,J)+(DN(J)+1.)*Y(I,J,J)+CYPOLY0428
  990 CONTINUE                                                          POLY0429
C                                                                       POLY0430
C  GII CORRECTIONS                                                      POLY0431
      DO 860 I=1,NNU                                                    POLY0432
      IF(G(I).EQ.0.) GO TO 860                                          POLY0433
      G(I) = G(I) + B                                                   POLY0434
      IF (TESTW(4)) X(I,I) = X(I,I)+G(I)/3.                             POLY0435
      IF (.NOT.TESTW(4))V(I) = V(I) - G(I)                              POLY0436
  860 CONTINUE                                                          POLY0437
C                                                                       POLY0438
C410                                                                    POLY0439
C  INTERMEDIATE OUTPUT--XIJS AND LEVEL.                                 POLY0440
C                                                                       POLY0441
 1092 IF (.NOT.TEST(14)) GO TO 1091                                     POLY0442
      WRITE (6,2860)                                                    POLY0443    659
 2860 FORMAT (8HOX(I,J)  )                                              POLY0444
      DO 2861 I =1,NNU                                                  POLY0445
 2861 WRITE (6,2862) (X(I,J),J=1,NNU)                                   POLY0446    663
 2862 FORMAT (1H ,6F10.4)                                               POLY0447
                    WRITE (6,1093) LEVEL                                POLY0448    669
 1093 FORMAT (8H1LEVEL = ,I2)                                           POLY0449
 1091 IF (TESTW(5)) TESTW(4) = .TRUE.                                   POLY0450
C                                                                       POLY0451
C  CALL LINK1 TO CALCULATE PARTITION FUNCTION AND DERIVATIVES FOR LEVEL.POLY0452
 1090 CALL LINK1                                                        POLY0453
C                                                                       POLY0454
C  IF CC 1-6 = CC1-6 OF PREVIOUS CARD, ASSUME THERE IS ANOTHER ELECTRONIPOLY0455
C     LEVEL AND GO TO 1001 (C330).                                      POLY0456
C  OTHERWISE CALCULATE FUNCTIONS FROM Q AND DERIVATIVES (VALUES FOR     POLY0457
C     MULTIPLE ELECTRONIC STATES HAVE BEEN SUMMED).                     POLY0458    674
      IF (ICARD .EQ. NSUB) GO TO 1001                                   POLY0459
      NT1 = NT
      IF (ASINDT.NE.0.)  NT1=NT+1
      DO 1000  I = NIT,NT1                                              POLY0460
      IF (.NOT.TESTW(6))  GO TO 999
      Q = FHRT(I)                                                       POLY0461
      FHRT(I) = ALOG(Q)                                                           689
      DQ = HHRT(I)/Q                                                    POLY0463
      HHRT(I) = DQ
      CPR(I) = CPR(I)/Q +(2.-DQ)*DQ
  999 FHRT(I) = FHRT(I) + 1.5*ALOG(WEIGHT) + 2.5*ALOG(T(I)) + SCONST    POLY0462    696   698
      HHRT(I) = HHRT(I) + 2.5
 1000 CPR(I) = CPR(I) + 2.5
      IF (ASINDT.EQ.0.) GO TO 4001                                      POLY0466
C                                                                       POLY0467
C  CALCULATE ENTHALPY FOR ASSIGNED T ON FORMULA CARD.                   POLY0468
      SPECH = HHRT(NT1)*E*ASINDT                                        POLY0469
      TEST(19) = .TRUE.                                                 POLY0470
 4001 TEST(9) = .TRUE.                                                  POLY0471
      RETURN                                                            POLY0472
 1094 WRITE(6,1095)                                                     POLY0473    713
 1095 FORMAT (37HO WRONG NUMBER OF NU-S(V-S), C410     )                POLY0474
      GO TO 2000                                                        POLY0475
 1098 WRITE(6,1099)                                                     POLY0476    715
 1099 FORMAT(35HO THE VALUE OF B IS MISSING, C410     )                 POLY0477
      GO TO 2000                                                        POLY0478
 1100 WRITE(6,1101)                                                     POLY0479    717
 1101 FORMAT(35HO THE VALUE OF C IS MISSING, C410      )                POLY0480
 2000 TEST(16) = .TRUE.                                                 POLY0481
      RETURN                                                            POLY0482
      END                                                               POLY0483
```

```
      SUBROUTINE LINK1                                                  LINK0001
C                                                                       LINK0002
C  CALCULATE Q                                                          LINK0003
C     TESTW(1)  MOLECULE IS NON-LINEAR                                  LINK0004
C     TESTW(2)  RIGID ROTATOR-HARMONIC OSCILLATOR APPROXIMATION         LINK0005
C     TESTW(3)  SECOND ORDER CORRECTIONS ARE CALLED FOR                 LINK0006
C     TESTW(4)  PENNINGTON AND KOBE APPROXIMATION                       LINK0007
C     TESTW(5)  JANAF METHOD FOR DIATOMIC MOLECULES                     POLY0007
C     TESTW(6) SPECIES HAS EXCITED ELECTRONIC STATES
      COMMON NAME(2),SYMBOL(70),ATMWT(70),E,HCK,ELECTR,ICARD,IWORD(5),  LINK0008
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),ANY   ,ELEMNT(70), LINK0009
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),LINK0010
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       LINK0011
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,  LINK0012
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT                        LINK0013
C                                                                       LINK0014
C420                                                                    LINK0015
C                                                                       LINK0016
      COMMON /WCOMMN/ V(20),DN(20),ND(20),X(6,6),Y(6,6,6),NNU,ALFA(6),  LINK0017
     1 ALFB(6), ALFC(6), G(6), WX(6), BETA(6), A, B, C, RH, D, WF, W,   LINK0018
     2 SYM, STWT, TOO, THETA(5),TESTW(6),R(20,3),S(20,3),QL(3),Q,QLN,DQ,LINK0019
     3 DDQ,LABEL,QTOT,QLNTOT,DQTOT,DDQTOT,CORT,AIJ(6,6),AIII,AI(6) ,NSUBLINK0020
      LOGICAL TESTW, TEST                                               LINK0021
      DATA LEL/6HELECTR/, LHO/4HH.O./, LRR/4HR.R./,LXIJ/3HXIJ/,LRHO/    LINK0022
     1 3HRHO/,LTHETA/5HTHETA/,LYIJK/4HYIJK/,LALPHA/5HALPHA/,LZ/4HWEZE/  LINK0023
C                                                                       LINK0024
C  TEST(14)--INTERM CARD HAS BEEN READ CALLING FOR INTERMEDIATE OUTPUT  LINK0025
      IF(.NOT.TEST(14)) GO TO 6                                         LINK0026
      DO 5 I = 1, NNU                                                   LINK0027
      NND=DN(I)                                                         LINK0028
    5 WRITE(6,1006) I,V(I),NND,I,I,G(I)                                 LINK0029    9
 1006 FORMAT(3HOV(,I1,3H) =,F9.4,1H(,I1,1H) 6X,1HG,2I1,2H =,F7.3)       LINK0030
    6 IF (ASINDT .NE. 0.0) GO TO 7                                      LINK0031
      NT1 = NT                                                          LINK0032
      GO TO 8                                                           LINK0033
    7 NT1 = NT + 1                                                      LINK0034
      T(NT1) = ASINDT                                                   LINK0035
C                                                                       LINK0036
C  DO LOOP THRU 1000(C480) CALCULATES Q AND DERIVATIVES FOR ELECTRONIC  LINK0037
C      LEVEL.  IT = T INDEX.                                            LINK0038
    8 DO 1000 IT = NIT,NT1                                              LINK0039
      QTOT = 1.0                                                        LINK0040
      QLNTOT = 0.0                                                      LINK0041
      DQTOT = 0.0                                                       LINK0042
      DDQTOT = 0.0                                                      LINK0043
      Q = 1.0                                                           LINK0044
      IF (TEST(14)) WRITE (6,4) T(IT)                                   LINK0045   25
    4 FORMAT(4HLT =F9.3)                                                LINK0046
 1008 CT = HCK/T(IT)                                                    LINK0047
      DO 10  I=1,NNU                                                    LINK0048
      R(I,1) = 0.0                                                      LINK0049
      U = CT * V(I)                                                     LINK0050
      IF (U.GE.30.) GO TO 9                                             LINK0051
C                                                                       LINK0052
C  R(I,1) = RI.   S(I,1) = SI.  A 2 OR  3 IN THE SECOND SUBSCRIPT       LINK0053
C      INDICATES FIRST OR  SECOND DERIVATIVE RESPECTIVELY OF RI AND SI. LINK0054
C  THESE DERIVATIVES ARE USED TO OBTAIN THE DERIVATIVES OF THE Q        LINK0055
C     CONTRIBUTIONS IN SUBROUTINE DERIV.                                LINK0056
C                                                                       LINK0057
      R(I,1) = EXP(-U)                                                  LINK0058   37
    9 S(I,1) = 1./(1.-R(I,1))                                           LINK0059
      R(I,2) = U                                                        LINK0060
      R (I,3) = -U                                                      LINK0061
      S(I,2) = R(I,1)*S(I,1)*U                                          LINK0062
      S(I,3) = S(I,2)*(S(I,2) + U - 1.)                                 LINK0063
      IF(TEST(14)) WRITE(6,1018)        U,R(I,1),S(I,1),I               LINK0064   50
 1018 FORMAT(7HO   U = ,E13.7, 6H   R = ,E13.7, 6H   S = ,E13.7,3X,3HI =LINK0065
     1I2)                                                               LINK0066
   10 CONTINUE                                                          LINK0067
      IF(TEST(14)) WRITE (6,1005)                                       LINK0068   56
 1005 FORMAT(13HOCONTRIBUTION,13X,1HQ,15X,4HLN Q,11X,8H H-HO/RT,13X,    LINK0069
     14HCP/R)                                                           LINK0070
C                                                                       LINK0071
C430                                                                    LINK0072
C  QLN = LN Q.       DQ = TDLNQ/DT.       DDQ = T2D2(LN Q)/DT2.        LINK0073
C  SUBROUTINE QSUM ACCUMULATES CONTRIBUTIONS OF LN Q AND DERIVATIVES.   LINK0074
C                                                                       LINK0075
C  ELECTRONIC PARTITION FUNCTION--FORMULA 1.                            LINK0076
      DQ = CT*TOO                                                       LINK0077
      QLN = ALOG(STWT) - DQ                                             LINK0078   58
      DDQ    = -2.0 * DQ                                                LINK0079
      LABEL = LEL                                                       LINK0080
      CALL QSUM (TEST(14))                                              LINK0081
C                                                                       LINK0082
C  HARMONIC OSCILLATOR PARTITION FUNCTION--FORMULA 2.                   LINK0083   60
      DO 15 I = 1, NNU                                                  LINK0084
      QLN = QLN + DN(I) * ALOG(S(I,1))                                  LINK0085   65
      DQ = DQ+DN(I) * S(I,2)                                            LINK0086
   15 DDQ = DDQ+DN(I) * S(I,3)                                          LINK0087
```

```
      DDQ = DDQ - DQ                                                    LINK0088
      CORT = 0.                                                         LINK0089
      LABEL = LHO                                                       LINK0090
      CALL QSUM (TEST(14))                                              LINK0091
C                                                                       LINK0092
C  RIGID ROTATOR PARTITION FUNCTION--FORMULAS 3 AND 4.                  LINK0093    74
      LABEL = LRR                                                       LINK0094
      IF (TESTW(1)) GO TO 20                                            LINK0095
      Q = 1.0/(SYM * CT * B)                                            LINK0096
      DQ = 1.0                                                          LINK0097
      DDQ = -1.0                                                        LINK0098
      GO TO 30                                                          LINK0099
   20 Q = (1.0/CT**3 * 3.1415927/ (A*B*C)) **0.5*1.0/ SYM               LINK0100    82
      DQ = 1.5                                                          LINK0101
      DDQ = -1.5                                                        LINK0102
   30 QLN = ALOG(Q)                                                     LINK0103    84
      CALL QSUM (TEST(14))                                              LINK0104
C                                                                       LINK0105
C  END RRHO CALCULATIONS.  GO TO 900(C480) TO ACCUMULATE Q FOR LEVEL.   LINK0106    85
      IF (TESTW(2)) GO TO 900                                           LINK0107
C                                                                       LINK0108
C440                                                                    LINK0109
C                                                                       LINK0110
C  ROTATIONAL STRETCHING--FORMULA 5.                                    LINK0111
      LABEL = LRHO                                                      LINK0112
      QLN = RH*T(IT)                                                    LINK0113
      DQ = QLN                                                          LINK0114
      DDQ = 0.0                                                         LINK0115    92
      CALL QSUM (TEST(14))                                              LINK0116
      LABEL = LTHETA                                                    LINK0117
      Q=1.+ ((THETA(3)/T(IT) + THETA(2))/T(IT) + THETA(1)) / T(IT)      LINK0117    98
      QLN = ALOG(Q)                                                     LINK0118
      DQ = -((3.*THETA(3)/T(IT) +2.*THETA(2))/T(IT) + THETA(1)) /T(IT)/QLINK0119
      DDQ = ((2.*THETA(3)/T(IT) + THETA(2)) * 3./T(IT) + THETA(1)) * 2./LINK0120
     1   T(IT)      /Q - DQ**2                                          LINK0121
      CALL QSUM (TEST(14))                                              LINK0122
C                                                                       LINK0123    105
C  VIBRATIONAL-ROTATION INTERACTION USING ALPHA CONSTANTS--FORMULAS 8-10LINK0124
      LABEL = LALPHA                                                    LINK0125
      DO 39  I=1,NNU                                                    LINK0126
      QL(I ) =AI(I)* DN(I)                                              LINK0127    112
      CALL DERIV (I,0,0,0,0,I,0,0,0,0,0)                                LINK0128
      IF(TESTW(4)) GO TO 39                                             LINK0129
      QL(1) = .5*DN(I)*AI(I)**2                                         LINK0130    120
      CALL DERIV (I,0,0,0,0,I,I,0,0,0,0)                                LINK0131
      QL(1) = DN(I)/6.*AI(I)**3                                         LINK0132    126
      CALL DERIV (I,0,0,0,0,I,I,I,0,0,0)                                LINK0133
      QL(1) = DN(I)/6.*AI(I)**3                                         LINK0134    132
      CALL DERIV (I,I,0,0,0,I,I,I,0,0,0)                                LINK0135
      IF(TESTW(1)) GO TO 39                                             LINK0136
      QL(1) = AIJ(I,I) *DN(I)                                           LINK0137    139
      CALL DERIV (I,0,0,0,0,I,I,0,0,0,0)                                LINK0138
      QL(1) =.AIJ(I,I)*DN(I)*AI(I)                                      LINK0139    144
      CALL DERIV (I,0,0,0,0,I,I,I,0,0,0)                                LINK0140
      QL(1) = AIJ(I,I)*DN(I)*AI(I)                                      LINK0141    149
      CALL DERIV (I,I,0,0,0,I,I,I,0,0,0)                                LINK0142
      DO 37 J = 1,NNU                                                   LINK0143
      QL(1) = AIJ(I,J)*DN(I)*DN(J)                                      LINK0144
      IF (I.GT.J) GO TO 35                                              LINK0145    159
      CALL DERIV (I,J,0,0,0,I,J,0,0,0,0)                                LINK0146
   35 QL(1) = AI(I)*AIJ(I,J) * DN(I)* DN(J)                             LINK0147
      IF (I.EQ.J) QL(1) = QL(1) * 2.                                    LINK0148    168
   37 CALL DERIV (I,J,0,0,0,I,I,J,0,0,0)                                LINK0149
      IF (NOATMS .GT.2) GO TO 39                                        LINK0150
C                                                                       LINK0151
C  FORMULA 11.                                                          LINK0152
      QL(1) = AIII                                                      LINK0153    174
      CALL DERIV (I,0,0,0,0,I,I,I,0,0,0)                                LINK0154
      QL(1) = 4. * AIII                                                 LINK0155    176
      CALL DERIV (I,I,0,0,0,I,I,I,0,0,0)                                LINK0156
      QL(1) = AIII                                                      LINK0157    178
      CALL DERIV ( I,I,I,0,0,I,I,I,0,0,0)                               LINK0158
   39 CONTINUE                                                          LINK0159
C                                                                       LINK0160
C450                                                                    LINK0161
C                                                                       LINK0162    181
      IF (TEST(14)) WRITE (6,40)                                        LINK0163
   40 FORMAT(25HOFIRST ORDER CORRECTIONS     )                          LINK0164    183
      CALL QSUM (TEST(14))                                              LINK0165
      CORT = 1.0                                                        LINK0166
C                                                                       LINK0167
C  FIRST ORDER XIJ--FORMULA 12.                                         LINK0168
      LABEL = LXIJ                                                      LINK0169
      DO 50  I=1,NNU                                                    LINK0170
      DO 50  J=I,NNU                                                    LINK0171
      CON = DN(I)*DN(J)                                                 LINK0172
      IF (I.EQ.J) CON=CON+DN(I)                                         LINK0173
   44 QL(1) = CON*(-CT)*X(I,J)                                          LINK0174    198
   50 CALL DERIV (I,J,0,0,0,I,J,0,0,0,0)
```

```
      CALL QSUM (TEST(14))                                                  LINK0175
C                                                                           LINK0176
C  END CALCULATIONS FOR PANDK AND JANAF.                                    LINK0177    202
      IF (TESTW(4)) GO TO 900                                               LINK0178
C                                                                           LINK0179
C  FIRST ORDER YIJK--FORMULA13.                                             LINK0180
      LABEL = LYIJK                                                         LINK0181
      DO 70  I=1,NNU                                                        LINK0182
      DO 70  J=I,NNU                                                        LINK0183
      DO 70  K=J,NNU                                                        LINK0184
      CON =  ND(I)*(ND(J)+ KD(I,J))*(ND(K)+ KD(I,K)+ KD(J,K))               LINK0185    214   216   217
      QL(1)=CON * (-CT) *Y(I,J,K)                                           LINK0186
   70 CALL DERIV (I,J,K,0,0,I,J,K,0,0,0)                                    LINK0187    222
      CALL QSUM (TEST(14))                                                  LINK0188    227
      DATA LAX/4HAXIJ/,LG/4HG+AG/,            LX2 /6H(XIJ)2/ ,LXY           LINK0189
     1/2HXY/, LG2/5HG2,GX/,LAX2 /3HAX2/,LFERMI/5HFERMI/                     LINK0190
C                                                                           LINK0191
C  FIRST ORDER ALPHA-XIJ INTERACTION--FORMULA 17.                           LINK0192
      LABEL = LAX                                                           LINK0193
      DO 100 I=1,NNU                                                        LINK0194
      AL =  AI(I)*(-CT)                                                     LINK0195
      DO 100  J=1,NNU                                                       LINK0196
      IF (I.EQ.J)QL(1)=AL*X(I,I)*2.*DN(I)*(DN(I)+1.)                        LINK0197
      IF (I.NE.J)QL(1)= AL *X(I,J)*DN(I)*DN(J)                              LINK0198
  100 CALL DERIV (I,J,0,0,0,I,I,J,0,0,0)                                    LINK0199    246
      CALL QSUM (TEST(14))                                                  LINK0200
C                                                                           LINK0201
C460                                                                        LINK0202
C                                                                           LINK0203    250
      DO 120 I=1,NNU                                                        LINK0204
      IF(G(I).EQ.0.) GO TO 120                                              LINK0205
C                                                                           LINK0206
C  GII CORRECTION-- FORMULA 16.                                             LINK0207
      LABEL = LG                                                            LINK0208
      QL(1 )=G(I)*(-CT)*2.                                                  LINK0209
      CALL DERIV (I,0,0,0,0,I,I,0,0,0,0)                                    LINK0210    260
      QL(1) = 4.*G(I)*AI(I)*CT                                              LINK0211
      CALL DERIV (I,I,0,0,0,I,I,I,0,0,0)                                    LINK0212    264
  120 CONTINUE                                                              LINK0213
      IF(LABEL.EQ.LG) CALL QSUM(TEST(14))                                   LINK0214    268
      IF (NDATMS .GT.2) GO TO 130                                           LINK0215
C                                                                           LINK0216
C  WEZE FOR DIATOMIC MOLECULES--FORMULA 15.                                 LINK0217
      LABEL = LZ                                                            LINK0218
      QL(1) = 24.*WX(3)*(-CT)                                               LINK0219
      CALL DERIV (1,1,1,1,0,1,1,1,1,0,0)                                    LINK0220    274
      CALL QSUM (TEST(14))                                                  LINK0221    276
  130 CTT=CT **2/2.                                                         LINK0222
      IF(WF.EQ.0.0) GO TO 141                                               LINK0223
C                                                                           LINK0224
C  FERMI RESONANCE--FORMULA 7.                                              LINK0225
      LABEL = LFERMI                                                        LINK0226
      CORT = 2.                                                             LINK0227
      U = CT*2.*V(2)                                                        LINK0228
      RW= EXP(-U)                                                           LINK0229    282
      SW= 1./(1.-RW)                                                        LINK0230
      CON    = WF**2*CTT *RW * SW**2*S(2,1)**2                              LINK0231
      QL(1) = CON                                                           LINK0232
      QL(2) = U + 2.*RW*SW*U + 2.*S(2,2)                                    LINK0233
      QL(3)= -U + 2.*RW*SW*U*(U + RW*SW*U-1.)+ 2.*S(2,3)                    LINK0234
      CALL DERIV(0,0,0,0,0,0,0,0,0,0,0)                                     LINK0235    283
      QL(1) = -CON                                                          LINK0236
      CALL DERIV (1,0,0,0,0,0,0,0,0,0,0)                                    LINK0237    285
      CALL QSUM (TEST(14))                                                  LINK0238
C                                                                           LINK0239
C  END CALCULATIONS FOR NRRAO1.                                             LINK0240    287
  141 IF(.NOT.TESTW(3)) GO TO 900                                           LINK0241
      IF (TEST(14)) WRITE (6,142)                                           LINK0242    291
  142 FORMAT (26HO SECOND ORDER CORRECTIONS    )                            LINK0243
C                                                                           LINK0244
C470                                                                        LINK0245
C                                                                           LINK0246
C  XIJ - XIJ INTERACTION--FORMULAS 18 AND 19.                               LINK0247
      LABEL = LX2                                                           LINK0248
      CORT = 2.0                                                            LINK0249
      DO 200 I=1,NNU                                                        LINK0250
      DO 180 J=I,NNU                                                        LINK0251
      CON = DN(I)*DN(J)                                                     LINK0252
      IF(I.EQ.J) CON = 2.*DN(I)*(DN(I)+1.)                                  LINK0253
      QL(1)=CON*X(I,J)**2*CTT                                               LINK0254
  180 CALL DERIV (I,J,0,0,0,I,J,J,0,0)                                      LINK0255    307
      DO 200 J=1,NNU                                                        LINK0256
      DO 200 K=J,NNU                                                        LINK0257
      CON = (2-KD(J,K))*(1+KD(I,J))*(1+KD(I,K))*ND(I)*(ND(J)+KD(I,J))*      LINK0258
     1(ND(K)+KD(I,K))                                                       LINK0259    314   315   316   317   319
      QL(1) = CON * X(I,J)*X(I,K)*CTT                                       LINK0260
  200 CALL DERIV (I,J,K,0,0,I,J,K,0,0)                                      LINK0261    325
      CALL QSUM (TEST(14))                                                  LINK0262
C                                                                           LINK0263
```

```
C  XIJ - YIJK INTERACTION--FORMULAS 20 AND 21.                          LINK0264    330
      LABEL = LXY                                                       LINK0265
      DO 300  I=1,NNU                                                   LINK0266
      DO 300  J=I,NNU                                                   LINK0267
      DO 300  K=1,NNU                                                   LINK0268
      CON = 2*(1+KD(I,J))*(1+KD(I,K) + KD(J,K))*(ND(I)+ KD(I,J))*ND(J)* LINK0269
     1(ND(K) + KD(J,K) + KD(I,K))                                       LINK0270    339   340   341   342   344   345
      QL(1) = CON*CTT*X(I,J)*Y(I,J,K)                                   LINK0271
  300 CALL DERIV (I,J,K,0,0,I,I,J,J,K,0)                                LINK0272    351
      DO 400  I=1,NNU                                                   LINK0273
      DO 400  J=1,NNU                                                   LINK0274
      DO 400  K=I,NNU                                                   LINK0275
      DO 400  L=K,NNU                                                   LINK0276
      CON = (1+KD(I,J))*(1+KD(I,K)+KD(I,L))*ND(I)*(ND(J)+KD(I,J))*(ND(K)LINK0277
     1+ KD(I,K))*(ND(L)+ KD(I,L)+ KD(K,L))*2                            LINK0278    365   366   367   368   370   372   373
      QL(1) = CON*CTT*X(I,J)*Y(I,K,L)                                   LINK0279
  400 CALL DERIV (I,J,K,L,0,I,I,J,K,L,0)                                LINK0280    379
      CALL QSUM (TEST(14))                                              LINK0281
C                                                                       LINK0282
C480                                                                    LINK0283
C                                                                       LINK0284    385
      DO 500  I=1,NNU                                                   LINK0285
      IF (G(I).EQ.0.) GO TO 500                                         LINK0286
C                                                                       LINK0287
C  GII - GII AND GII - XIJ INTERACTIONS--FORMULAS 22 AND 23.            LINK0288
      LABEL = LG2                                                       LINK0289
      QL(1) = 2.*G(I)**2*CTT                                            LINK0290
      CALL DERIV (I,0,0,0,0,I,I,I,I,0,0)                                LINK0291    396
      DO 490  J=1,NNU                                                   LINK0292
      CON = 4.*G(I)*X(I,J)*CTT                                          LINK0293
      IF (I.EQ.J) CON = 16.*G(I)*(G(I) +2.*X(I,I))*CTT                  LINK0294
      QL(1) = CON                                                       LINK0295
      CALL DERIV (I,J,0,0,0,I,I,I,J,0,0)                                LINK0296    407
      QL(1) = CON                                                       LINK0297
      IF (I.EQ.J)QL(1) =       CTT*2.*(G(I)+12.*X(I,I))*G(I)            LINK0298
  490 CALL DERIV (I,I,J,0,0,I,I,I,J,0,0)                                LINK0299    415
  500 CONTINUE                                                          LINK0300
      IF(LABEL.EQ.LG2) CALL QSUM(TEST(14))                              LINK0301
C                                                                       LINK0302
C  ALPHA - XIJ - XIJ INTERACTION--FORMULAS 24 THRU 27.                  LINK0303    421
      LABEL = LAX2                                                      LINK0304
      DO 600  I=1,NNU                                                   LINK0305
      AL = AI(I)*CTT                                                    LINK0306
      QL(1) = 4.*AL*(X(I,I)*DN(I)*(DN(I)+1.))**2                        LINK0307
      CALL DERIV (I,I,I,I,I,I,I,I,I,0)                                  LINK0308    430
      DO 600  J=1,NNU                                                   LINK0309
      IF (I.EQ.J) CON = 4.*DN(I)*(DN(I)+1.)                             LINK0310
      IF (I.NE.J) CON = DN(I)*DN(J)                                     LINK0311
      QL(1) = CON*X(I,J)**2*AL                                          LINK0312
      CALL DERIV (I,J,0,0,0,I,I,I,J,J,0)                                LINK0313    444
      DO 600  K=1,NNU                                                   LINK0314
      CON = (1+ KD(I,J))*(1+ KD(I,K))*ND(I)*(ND(J)+ KD(I,J))*(ND(K)+ KD(LINK0315
     1I,K))                                                             LINK0316    448   449   450   452
      QL(1) = CON*AL*X(I,J)*X(I,K)                                      LINK0317
      CALL DERIV (I,J,K,0,0,I,I,I,J,K,0)                                LINK0318    457
      QL(1) = CON*AL*X(I,J)*X(I,K)                                      LINK0319
      CALL DERIV (I,J,K,I,0,I,I,I,J,K,0)                                LINK0320    461
      CON = (1+ KD(I,J))*(1+ KD(J,K))*(2- KD(I,K))*ND(I)*(ND(J)+ KD(I,J)LINK0321
     1)*((1+ KD(I,K))*ND(K)+ KD(I,K)+ KD(J,K)+ KD(I,J)*KD(J,K))         LINK0322    463   464   465   466   468   469   470   471
                                                                                 472
      QL(1) = CON * AL* X(I,J) * X(J,K)                                 LINK0323
      CALL DERIV (I,J,K,0,0,I,I,J,J,K,0)                                LINK0324    477
  600 CONTINUE                                                          LINK0325
      CALL QSUM(TEST(14))                                               LINK0326    482
  900 IF (TESTW(6))   GO TO 902
C
C   CALCULATIONS FOR SPECIES WITH ONE ELECTRONIC STATE
C
      FHRT(IT) = QLNTOT
      HHRT(IT) = DQTOT
      CPR(IT) = DDQTOT + 2.*DQTOT
      GO TO 1000
C
C   CALCULATIONS FOR SPECIES WITH EXCITED ELECTRONIC STATES
C
  902 IF (QLNTOT.LE.88.)  GO TO 903
      WRITE (6,2)                                                               492
    2 FORMAT(44HOO TOO LARGE TO INCLUDE EXCITED STATES. C480)
    3 IF (ICARD.NE.NSUB)  RETURN
      CALL INPUT(LINES)                                                         496
      GO TO 3
  903 QTOT = EXP(QLNTOT)                                                        499
      FHRT(IT) = QTOT   + FHRT(IT)                                      LINK0327
      HHRT(IT) = DQTOT*QTOT + HHRT(IT)                                  LINK0328
      CPR(IT) = (DDQTOT + .DQTOT**2)*QTOT + CPR(IT)                     LINK0329
 1000 CONTINUE                                                          LINK0330
      RETURN                                                            LINK0331
      END                                                               LINK0332
```

```
      FUNCTION KD(I,J)                                          KDEL0001
      KD = 0                                                    KDEL0002
      IF (I.EQ.J) KD = 1                                        KDEL0003
      RETURN                                                    KDEL0004
      END                                                       KDEL0005
```

```
      SUBROUTINE DERIV (I1,I2,I3,I4,I5,J1,J2,J3,J4,J5,J6)                DERI0001
C
C  FIND Q DERIVATIVES.
C
      COMMON /WCOMMN/ V(20),DN(20),ND(20),X(6,6),Y(6,6,6),NNU,ALFA(6),   WOOL0013
     1 ALFB(6), ALFC(6), G(6), WX(6), BETA(6), A, B, C, RH, D, WF, W,    DERI0003
     2 SYM, STWT, TOO, THETA(5),TESTW(6),R(20,3),S(20,3),QL(3),Q,QLN,DQ,DERI0004
     3 DDQ,LABEL,QTOT,QLNTOT,DQTOT,DDQTOT,CORT,AIJ(6,6),AIII,AI(6),NSUB DERI0005
C                                                                        DERI0006
C490                                                                     DERI0007
C                                                                        DERI0008
      DIMENSION I(5), J(6)                                               DERI0009
      I(1) = I1                                                          DERI0010
      I(2) = I2                                                          DERI0011
      I(3) = I3                                                          DERI0012
      I(4) = I4                                                          DERI0013
      I(5) = I5                                                          DERI0014
      J(1) = J1                                                          DERI0015
      J(2) = J2                                                          DERI0016
      J(3) = J3                                                          DERI0017
      J(4) = J4                                                          DERI0018
      J(5) = J5                                                          DERI0019
      J(6) = J6                                                          DERI0020
      DATA IFERMI/5HFERMI/                                               DERI0021
      IF (LABEL.EQ.IFERMI)  GO TO 8                                      DERI0022
      QL(2)=0.                                                           DERI0023
      QL(3)=0.                                                           DERI0024
    8 DO 10 IR=1,5                                                       DERI0025
      K = I(IR)                                                          DERI0026
      IF(K.EQ.0) GO TO 20                                                DERI0027
      QL(1) = QL(1)*R(K,1)                                               DERI0028
      QL(2) = QL(2)+R(K,2)                                               DERI0029
   10 QL(3) = QL(3)+R(K,3)                                               DERI0030
   20 DO 30 IS =1,6                                                      DERI0031
      K = J(IS)                                                          DERI0032
      IF (K.EQ.0) GO TO 40                                               DERI0033
      QL(1) = QL(1)*S(K,1)                                               DERI0034
      QL(2) = QL(2)+S(K,2)                                               DERI0035
   30 QL(3) = QL(3)+S(K,3)                                               DERI0036
   40 QLN = QLN + QL(1)                                                  DERI0038
      QCORT = QL(2) - CORT                                               DERI0039
      DQ = QCORT*QL(1) + DQ                                              DERI0040
      DDQ = QL(1)*(QL(3)+QCORT**2 - QCORT) + DDQ                         DERI0041
      RETURN                                                             DERI0042
      END                                                                DERI0043
```

```
      SUBROUTINE QSUM (TEST)                                          QSUM0001
C
C  ACCUMULATE VALUES OF Q AND ITS DERIVATIVES.
C
      COMMON /WCOMMN/ V(20),DN(20),ND(20),X(6,6),Y(6,6,6),NNU,ALFA(6),  WOOL0013
     1 ALFB(6), ALFC(6), G(6), WX(6), BETA(6), A, B, C, RH, D, WF, W,   QSUM0003
     2 SYM, STWT, TOO, THETA(5),TESTW(6),R(20,3),S(20,3),QL(3),Q,QLN,DQ,QSUM0004
     3 DDQ,LABEL,QTOT,QLNTOT,DQTOT,DDQTOT,CORT,AIJ(6,6),AIII,AI(6) ,NSUBQSUM0005
C                                                                     QSUM0006
C500                                                                  QSUM0007
C                                                                     QSUM0008
      LOGICAL TEST                                                    QSUM0009
      IF (.NOT.TEST) GO TO 8
      Q = 0.
      IF (ABS(QLN).LE.88.)  Q=EXP(QLN)                                        6
      CPROUT = DDQ + 2. * DQ                                          QSUM0016
      WRITE (6,6)  LABEL,Q,QLN,DQ,CPROUT                              QSUM0017    8
    6 FORMAT(4X,A6,E21.4,3F18.8)                                      QSUM0018
    8 QLNTOT = QLNTOT + QLN                                           QSUM0020
      DQTOT = DQTOT + DQ                                              QSUM0021
      DDQTOT = DDQTOT + DDQ                                           QSUM0022
      QLN = 0.0                                                       QSUM0024
      DQ = 0.0                                                        QSUM0025
      DDQ = 0.0                                                       QSUM0026
      RETURN                                                          QSUM0027
      END                                                             QSUM0028
```

```
      SUBROUTINE DELH                                                    DELH0001
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),   DELH0002
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70),  DELH0003
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),DELH0004
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,        DELH0005
     4       SPECH,TAPE(606  ),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,   DELH0006
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR,IHEAT,JF(5)   DELH0007
      COMMON/PCH/K,NF1,NF2,ANS(9,15),TC(10),NTC,NFP,LDATE,NNN,NLAST      DELH0008
C                                                                        DELH0009
C510                                                                     DELH0010
C                                                                        DELH0011
      EQUIVALENCE  (AME,NAM)                                             DELH0012
      DATA IDELH/6HDELTAH/,IDIS/6HDISSOC/,IASH/6HASINDH/,IB/1H /         DELH0013
      LOGICAL TEST                                                       DELH0014
      INTEGER ELEMNT                                                     DELH0015
      IF (TEST(18)) GO TO 67                                             DELH0021
   12 IF (IHEAT.EQ.IB)  GO TO 164                                        DELH0022
      IF(TEST(13).AND.ASINDT.EQ.298.15)  TEST(8)=.TRUE.                  DELH0023
      IF (TEST(8))  GO TO 66                                             DELH0024
      IF (TEST(19))  GO TO 120                                           DELH0025
      WRITE (6,1064)                                                     DELH0026    15
 1064 FORMAT (42HOINSUFFICIENT DATA FOR AN HO VALUE, C510   )            DELH0027
      RETURN                                                             DELH0028
   66 IF (.NOT.TEST(17)) GO TO 1066                                      DELH0029
      CALL PUNCH                                                         DELH0030    19
      TEST(17) = .FALSE.                                                 DELH0031
 1066 IF(TEST(15)) CALL LEAST                                            DELH0032    23
      RETURN                                                             DELH0033
   67 IF (TEST(8)) GO TO 69                                              DELH0034
      IF (IHEAT.EQ.IASH .AND. ASINDT.EQ.298.15) GO TO 68                 DELH0035
      ASINDH = 0.                                                        DELH0036
      GO TO 66                                                           DELH0037
   68 TEST(13) = .TRUE.                                                  DELH0038
      TEST(8) = .TRUE.                                                   DELH0039
   69 DO 70 I = 1, NT                                                    DELH0040
      HHRT(I) = HHRT(I) - ASINDH/(R*T(I))                                DELH0041
   70 FHRT(I) = FHRT(I)+ ASINDH/(R*T(I))                                 DELH0042
      GO TO 66                                                           DELH0043
C                                                                        DELH0044
C520                                                                     DELH0045
C                                                                        DELH0046
  120 IF (IHEAT .EQ. IASH)  GO TO 167                                    DELH0047
      IF (IHEAT .EQ. IDELH) GO TO 166                                    DELH0048
      IF (IHEAT .EQ. IDIS)  GO TO 166                                    DELH0049
  164 WRITE(6,165)                                                       DELH0050    56
  165 FORMAT(92HOEITHER ASINDH,DELTAH,HF298,IPATOM,OR DISSOC WAS NOT FOUDELH0051
     1ND ON THE FORMULA CARD, C520           )                           DELH0052
  100 LINES = LINES + 2                                                  DELH0053
  184 TEST(8) = .FALSE.                                                  DELH0054
      RETURN                                                             DELH0055
  167 ASINDH = ASINDH - SPECH                                            DELH0056
  162 TEST(8) = .TRUE.                                                   DELH0057
      GO TO 66                                                           DELH0058
C                                                                        DELH0059
C530                                                                     DELH0060
C                                                                        DELH0061
  166 IF (IHEAT .EQ. IDELH) ASINDH =ASINDH- SPECH                        DELH0062
      IF (IHEAT .EQ. IDIS)  ASINDH =-ASINDH- SPECH                       DELH0063
      DO 180 I = 1,NKIND                                                 DELH0064
      J = JF(I)                                                          DELH0065
      REWIND 3                                                           DELH0066    71
      IF(IHEAT .EQ. IDELH) N = LPLACE(J)                                 DELH0067
      IF(IHEAT .EQ. IDIS)  N = MPLACE(J)                                 DELH0068
      IF (N.EQ.0) GO TO 200                                              DELH0069
      CALL SKFILE (3,N)                                                  DELH0070    81
      READ(3) NAM,HZERO,PT,TNO                                           DELH0071    82
      IF(ASINDT.EQ.0.0) GO TO 185                                        DELH0072
      NOT = TNO + 0.0000001                                              DELH0073
      KK = 3*NOT                                                         DELH0074
      READ (3) (TAPE(K), K = 1, KK)                                      DELH0075    88
      DO 186 K = 1, NOT                                                  DELH0076
      IF (TAPE(K  ).GE.ASINDT-0.0000001) GO TO 187                       DELH0077
  186 CONTINUE                                                           DELH0078
      WRITE (6,188)                                                      DELH0079    100
  188 FORMAT(50HOHHRT FOR ASINDT WAS NOT FOUND ON EF TAPE, C530         )DELH0080
      GO TO 100                                                          DELH0081
  200 WRITE (6,201) FORMLA(I)                                            DELH0082    102
  201 FORMAT (1HO,A6, 40HDATA WERE NOT FOUND ON EF TAPE, C530        )   DELH0083
      GO TO 100                                                          DELH0084
  187 KK = NOT + K                                                       DELH0085
      HA = TAPE(K)*R*TAPE(KK) + HZERO                                    DELH0086
  189 IF(IHEAT .EQ. IDELH)ASINDH=ASINDH+FLOAT(MLA(I))*HA/FLOAT(NMLA(J)) DELH0087
      IF (IHEAT .EQ. IDIS) ASINDH = ASINDH + FLOAT(MLA(I))*HA            DELH0088
      GO TO 180                                                          DELH0089
  185 HA = HZERO                                                         DELH0090
      GO TO 189                                                          DELH0091
  180 CONTINUE                                                           DELH0092
      GO TO 162                                                          DELH0093
      END                                                                DELH0094
```

```
      SUBROUTINE TABLES                                                 TABL0001
C
C  LIST FIRST 2 TABLES  OF THERMODYNAMIC FUNCTIONS.                     TABL0002
C                                                                       TABL0002
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),  TABL0003
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70), TABL0004
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),TABL0005
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       TABL0006
     4       SPECH,TAPE(202,3),PTMELT,PEX(10),TRANGE(10),TCONST,NKIND,  TABL0007
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR              TABL0008
C                                                                       TABL0009
C540                                                                    TABL0010
C                                                                       TABL0011
      LOGICAL TEST                                                      TABL0012
      EQUIVALENCE (IT,TI)                                               TABL0013
      DIMENSION FMT(12),FF(26),HEAD1(22),HEAD2(20)                      TABL0014
      DATA FMT(1)/4H( 1H /,FMT(3)/6HF12.7,/, FMT(11)/1H)/,              TABL0015
     1(FF(I),I=1,10)/55HI9,3X,F12.2,F14.7,F14.4,F14.6,A6,8X,F16.7,F16.4,TABL0016
     2A6,     /,(FF(J),J=11,26)/  96H(H-H29(F-H298)/RT H/RT, 12X,5H-F/RT TABL0017
     3H-H298-(F-H298)   H,14X,2H-F   (H-HO)(F-HO)H-HO  -(F-HO/RT    /,  TABL0018
     4(HEAD1(K),K=1,21)/  126H(1HO8X,1HT 8X,4HCP/R8X,    24H(H-HO)/RT   TABL0019
     5 (H-H298)/RT,6X3HS/R8X,   25H-(F-HO)/RT   -(F-H298)/RT,8X,4HH/RT, TABL0020
     612X,5H-F/RT //)   /,(HEAD2(L),L=1,19)/114H(1HO8X,1HT,10X,2HCP,11X,TABL0021
     7  19H H-O          H-H2981OX,1HS,9X,    23H-(F-HO)       -(F-H298),TABL0022
     810X,1HH,14X,2H-F   //)    /                                       TABL0023
      WRITE(6,11) NAME(1),NAME(2)                                       TABL0024    1
   11 FORMAT (1HO,2A6)                                                  TABL0025
C                                                                       TABL0032
C550                                                                    TABL0033
C                                                                       TABL0034
      DO 2000 I = 4,8                                                   TABL0036
 2000 FMT(I) = FF(3)                                                    TABL0037
      FMT(9) = FF(7)                                                    TABL0038
      DO 2005 K=15,16                                                   TABL0039
      HEAD1(K+3) = FF(K-1)                                              TABL0040
      HEAD2(K+2) = FF(K+5)                                              TABL0041
      HEAD1(K) = FF(10)                                                 TABL0042
      HEAD1(K-7) = FF(10)                                               TABL0043
 2005 HEAD2(K-1) = FF(10)                                               TABL0044
      HEAD1(20) = FF(16)                                                TABL0045
      HEAD2(8) = FF(10)                                                 TABL0046
      IF(TEST(13)) GO TO 2008
      HEAD1 (6)=FF(22)
      HEAD1 (7)=FF(26)
      HEAD1 (13)=FF(23)
      HEAD1 (14)=FF(26)
      HEAD2(6) =FF(24)
      HEAD2(12)=FF(25)
      HEAD2(13) =FMT(11)
      DO 25 I=1,NT                                                      TABL0047
      IF(ABS(T(I)-298.15).GT.0.01) GO TO 25                             TABL0048
      H298HR = HHRT(I) * T(I)                                           TABL0057
      GO TO 24
   25 CONTINUE                                                          TABL0059
      IF (H298HR.EQ.0.)  GO TO 2009
   24 HEAD1(8) = FF(11)                                                 TABL0050
      HEAD1(9) = FF(13)                                                 TABL0051
      HEAD1(15)= FF(12)                                                 TABL0052
      HEAD1(16)= FF(13)                                                 TABL0053
      HEAD2(8) = FF(17)                                                 TABL0054
      HEAD2(14)= FF(18)                                                 TABL0055
      HEAD2(15)= FF(19)                                                 TABL0056
      GO TO 2010
C                                                                       TABL0060
C560                                                                    TABL0061
C                                                                       TABL0062
 2008 HEAD1(6) =FF(11)
      HEAD1(7) =FF(13)
      HEAD1(13) =FF(12)
      HEAD1(14)=FF(13)
      HEAD2(6) =FF(17)
      HEAD2(12) =FF(18)
      HEAD2(13) =FF(19)
 2009 HH29 = FF(10)                                                     TABL0063
      FH29 = FF(10)                                                     TABL0064
      FMT(5) = FF(6)                                                    TABL0065
      FMT(8) = FMT(5)                                                   TABL0066
 2010 IF (TEST(8)) GO TO 2020                                           TABL0067
      H = FF(10)                                                        TABL0068
      F = FF(10)                                                        TABL0069
      FMT(9) = FF(9)                                                    TABL0070
      DO 2015 KK=18,19                                                  TABL0071
      HEAD1(KK+1) = FF(10)                                              TABL0072
 2015 HEAD2(KK-1) = FF(10)                                              TABL0073
      HEAD1(18) = FF(10)                                                TABL0074
```

```
C                                                                       TABL0078
C570                                                                    TABL0079
C                                                                       TABL0080
 2020 DO 3000 NTABLE = 1,2                                              TABL0081
      IF(TEST(8)) GO TO 55                                              TABL0026
      WRITE (6,51)                                                      TABL0027    62
   51 FORMAT (30HONO HZERO VALUE IS AVAILABLE                    )      TABL0028
      GO TO 22                                                          TABL0029
   55 IF(.NOT.TEST(13))WRITE(6,56) ASINDH                               TABL0030    65
   56 FORMAT (8HOHZERO = F12.3)                                         TABL0031
      IF(TEST(13)) WRITE(6,57) ASINDH                                   67
   57 FORMAT (8HOH298 = F12.3)
   22 IF(NTABLE.EQ.1)  WRITE(6,HEAD1)                                   70
      IF(NTABLE.EQ.2)  WRITE(6,HEAD2)                                   72
      FMT(10) = FMT(9)                                                  TABL0114
      LINES = LINES + 8                                                 TABL0116
      DO 399 I = 1, NT                                                  TABL0082
      IT = T(I)                                                         TABL0083
      FMT(2) = FF(1)                                                    TABL0084
      IF(AMOD(T(I),1.0).EQ.0.) GO TO 2130                               TABL0085
      TI = T(I)                                                         TABL0086
      FMT(2) = FF(2)                                                    TABL0087
 2130 ART = R*T(I)                                                      TABL0088
      AR = R                                                            TABL0089
      IF (NTABLE.EQ.2) GO TO 2135                                       TABL0090
      AR = 1.                                                           TABL0091
      ART = 1.                                                          TABL0092
 2135 CP = CPR(I)*AR                                                    TABL0093
      HH = HHRT(I) * ART                                                TABL0094
      S = (FHRT(I) + HHRT(I)) * AR                                      TABL0095
      FH = FHRT(I) * ART                                                TABL0096
      IF (.NOT.TEST(8)) GO TO 2120                                      TABL0097
      H = (HHRT(I)+ASINDH/R/T(I))*ART                                   TABL0098
      F = (FHRT(I)-ASINDH/R/T(I))*ART                                   TABL0099
 2120 IF (H298HR .EQ. 0.) GO TO 250                                     TABL0100
      HH29 = (HHRT(I)-H298HR/T(I))*ART                                  TABL0101
      FH29 = (FHRT(I)+H298HR/T(I))*ART                                  TABL0102
  250 WRITE (6,FMT) TI,CP,HH,HH29,S,FH,FH29,H,F                         TABL0103    109
      LINES = LINES + 1                                                 TABL0104
      IF(LINES.GE.55) CALL PAGEID(LINES)                                TABL0105    112
  399 CONTINUE                                                          TABL0106
      CALL PAGEID(LINES)                                                TABL0107    116
      IF (NTABLE.EQ.2) GO TO 4000                                       TABL0108
      DO 2100 I = 4,8                                                   TABL0109
 2100 IF(FMT(I).EQ.FF(3)) FMT(I) = FF(4)                                TABL0110
      FMT(6) = FF(5)                                                    TABL0111
      IF(TEST(8)) FMT(9) = FF(8)                                        TABL0113
 3000 CONTINUE                                                          TABL0117
 4000 RETURN                                                            TABL0118
      END                                                               TABL0119
```

```
      SUBROUTINE LOGK                                                    LOGK0001
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),   LOGK0002
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70),  LOGK0003
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),LOGK0004
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,        LOGK0005
     4       SPECH,TAPE(202,3),PTMELT,EXP(10),TRANGE(10),TCONST,NKIND,   LOGK0006
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI,H298HR,IHEAT,JF(5)   LOGK0007
C                                                                        LOGK0008
C580                                                                     LOGK0009
C                                                                        LOGK0010
      DIMENSION LMENT(4),PT(4), D(2,2,202),DHO(2),NTX(2),INIT(2),FF(10),LOGK0011
     1                  MARK(6),TK(3),FMT1(14),FMT2(14),FMT3(16)       LOGK0012
      EQUIVALENCE (IT,TI)                                                LOGK0013
      LOGICAL TEST                                                       LOGK0014
      INTEGER ELEMNT, FORMLA, SYMBOL                                     LOGK0015
      DATA BLK/1H /, ZERO/1HO/, ZERO4/4H   0/, DS1/4H----/,DS2/3H---/    LOGK0016
      DATA(FF(I),I=1,7 )/42H(2H  ,(2H *,F9.2, I6,3X,6X,A6,7X,A6,F13.4,/),LOGK0017
     1 (FMT1(J),J=3,8)  /5HF8.4,, 3*6HF10.4,, 2*6HF12.4,/,               LOGK0018
     2 (FMT2(K),K=3,7)  /30HF8.4, F12.1,F12.4,F12.1,F13.1,/,             LOGK0019
     3 (FMT3(L),L=1,16) /91H(6X,1HO,6X,6H------,9X,1HO,7X,A4,A3,9X,1HO,  LOGK0020
     4    F15.1,F13.1,6X,7H-------F13.1,6X,7H-------)/,                  LOGK0021
     5 (FMT1(M),M=12,13)/7H       )/,(FMT2(N),N=12,14)/13H             )/,LOGK0022
     6    FF(9)/6H5X,A6)/,FF(10)'4HF11.1)/                               LOGK0023
      IK = 0                                                             LOGK0024
      FMT3(10) = FMT2(7)                                                 LOGK0025
      FMT3(13) = FMT2(7)                                                 LOGK0026
      DO 195  I=1,NT                                                     LOGK0027
      DO 195  LL=1,2                                                     LOGK0028
      D(LL,1,I) = HHRT(I)                                                LOGK0029
      D(LL,2,I) = FHRT(I)                                                LOGK0030
      INIT(LL) = 1                                                       LOGK0031
      NTX(LL) = 0                                                        LOGK0032
  195 DHO(LL) = ASINDH                                                   LOGK0033
C                                                                        LOGK0034
C590                                                                     LOGK0035
C                                                                        LOGK0036
      DO 200 II = 1, NKIND                                               LOGK0037
      J = JF(II)                                                         LOGK0038
C                                                                        LOGK0039
C  LL=1  FORMATION FROM THE ELEMENTS                                     LOGK0040
C  LL=2  FORMATION FROM THE MONATOMIC GASES                              LOGK0041
C                                                                        LOGK0042
  202 LL = 1                                                             LOGK0043
      IF (NPLACE(J) .EQ. 0)   DHO(2) = BLK                               LOGK0044
      IF (LPLACE(J) .EQ. 0)   DHO(1) = BLK                               LOGK0045
      IF (NAME(1) .EQ. ELEMNT(J)) DHO(1) = ZERO4                         LOGK0046
      IF (.NOT.TEST(3).AND.TEST(4).AND.NOATMS.EQ.1) DHO(2) = ZERO4       LOGK0047
      IF(DHO(1).NE.ZERO4.AND.DHO(1).NE.BLK)  GO TO 505                   LOGK0048
      FMT3(10) = FF(6)                                                   LOGK0049
  204 LL = 2                                                             LOGK0050
      IF(DHO(2).NE.ZERO4.AND.DHO(2).NE.BLK)  GO TO 505                   LOGK0051
      FMT3(13) = FF(6)                                                   LOGK0052
      GO TO 501                                                          LOGK0053
  505 IF (LL.EQ.1) NN = LPLACE(J)                                        LOGK0054
      IF (LL.EQ.2) NN = MPLACE(J)                                        LOGK0055
      IF(NN.EQ.0) GO TO 501                                              LOGK0056
      REWIND 3                                                           LOGK0057
C                                                                        LOGK0058
C  READ EF DATA FOR REACTANT FROM TAPE                                   LOGK0059
C                                                                        LOGK0060    54
      CALL SKFILE (3,NN)                                                 LOGK0061    55
      READ (3) NAM, HZERO, AMP, TNO                                      LOGK0062    56
      M = TNO                                                            LOGK0063
      READ (3) ((TAPE(K,L), K=1,M), L=1,3)                               LOGK0064    58
      SUB = MLA(II)                                                      LOGK0065
      COEF = NMLA(J)                                                     LOGK0066
      IF(LL.EQ.2) COEF=1                                                 LOGK0067
      DHO(LL)= DHO(LL)-HZERO*SUB/COEF                                    LOGK0068
C                                                                        LOGK0069
C  FIND INDEX(NMP) FOR M.P.(AMP) OF REACTANT                             LOGK0070
C                                                                        LOGK0071
      IF  (AMP.EQ.0.)  GO TO 1241                                        LOGK0072
      DO 1505  K=1,M                                                     LOGK0073
      IF (TAPE(K,1).LT.AMP)  GO TO 1505                                  LOGK0074
      NMP = K + 1                                                        LOGK0075
      GO TO 1241                                                         LOGK0076
 1505 CONTINUE                                                           LOGK0077
C                                                                        LOGK0078
C600                                                                     LOGK0079
C                                                                        LOGK0080
 1241 DO 206  I=1,NT                                                     LOGK0081
C                                                                        LOGK0082
C   FIND T IN EFDATA                                                     LOGK0083
C                                                                        LOGK0084
      IF (I.EQ.1) GO TO 241                                              LOGK0085
      IF(T(I).GT.AMP.AND.AMP.GE.T(I-1)) GO TO 308                        LOGK0086
  241 IF (TAPE(1,1) .LE.T(I)) GO TO 208                                  LOGK0087
      INIT(LL) = INIT(LL) + 1                                            LOGK0088
      GO TO 206                                                          LOGK0089
C                                                                        LOGK0090
C  MP OF REACTANT, PUT * IN MARK FOR FOOTNOTE                            LOGK0091
C                                                                        LOGK0092
  308 IF (IK.LT.4) IK = IK+1                                             LOGK0093
      MARK(IK) = I                                                       LOGK0094
```

```
      PT(IK) = AMP                                                          LOGK0095
      LMENT(IK) = ELEMNT(J)                                                 LOGK0096
      GO TO 241                                                             LOGK0097
  208 K1 = 1                                                                LOGK0098
      IF ((AMP.EQ.0.).OR.INT(TNO).LE.NMP)  GO TO 1208                       LOGK0099
      M = NMP                                                               LOGK0100
      IF (T(I).LT.AMP)  GO TO 1208                                          LOGK0101
      K1 = NMP                                                              LOGK0102
      M = TNO                                                               LOGK0103
 1208 DO 209  K=K1,M                                                        LOGK0104
      TK(2) = TAPE(K,2)                                                     LOGK0105
      TK(3) = TAPE(K,3)                                                     LOGK0106
      IF (ABS(TAPE(K,1)-T(I)) .LT.0.01) GO TO 211                           LOGK0107
      IF(TAPE(K,1).LT.T(I)) GO TO 209                                       LOGK0108
      IF ((M-K1).EQ.0)  GO TO 209                                           LOGK0109
C                                                                           LOGK0110
C610                                                                        LOGK0111
C                                                                           LOGK0112
C  INTERPOLATION OF EF DATA                                                 LOGK0113
C                                                                           LOGK0114
      N=4                                                                   LOGK0115
      IF  ((M-K1).LT.3)  N=M-K1+1                                           LOGK0116
      K2=K-2                                                                LOGK0117
      IF (K.EQ.K1)  K2=K                                                    LOGK0118
      IF (K.EQ.(K1+1))  K2=K-1                                              LOGK0119
      NK = K2+N-1                                                           LOGK0120
      DO 2000  L=2,3                                                        LOGK0121
      TK(L) = 0.0                                                           LOGK0122
      DO 2000 JJ=K2,NK                                                      LOGK0123
      TK(1) = 1.0                                                           LOGK0124
      DO 1000 JM=1,N                                                        LOGK0125
        IM=K2+JM-1                                                          LOGK0126
      IF (TAPE(JJ,1).EQ.TAPE (IM,1)) GO TO 1000                             LOGK0127
      TK(1) = TK(1)*(T(I)-TAPE(IM,1))/(TAPE(JJ,1)-TAPE(IM,1))               LOGK0128
 1000 CONTINUE                                                              LOGK0129
 2000 TK(L) = TK(L)+TK(1)*TAPE(JJ,L)                                        LOGK0130
      GO TO 211                                                             LOGK0131
  209 CONTINUE                                                              LOGK0132
  213 IF((NTX(LL).EQ.0).OR.(((I-1).LT.NTX(LL)))  NTX(LL)=I-1                LOGK0133
      GO TO 500                                                             LOGK0134
C                                                                           LOGK0135
C  CALCULATE DELTA H AND DELTA F                                            LOGK0136
C                                                                           LOGK0137
  211 D(LL,1,I) = D(LL,1,I)-TK(2)*SUB/COEF                                  LOGK0138
      D(LL,2,I) = D(LL,2,I) - TK(3)*SUB/COEF                                LOGK0139
  206 CONTINUE                                                              LOGK0140
  500 IF (LL.NE.2)  GO TO 204                                               LOGK0141
  200 CONTINUE                                                              LOGK0142
C                                                                           LOGK0143
C620                                                                        LOGK0144
C                                                                           LOGK0145
C  LIST HEADING OF FIRST TABLE                                              LOGK0146
C                                                                           LOGK0147
  501 WRITE (6,320) NAME(1), NAME(2)                                        LOGK0148   196
  320 FORMAT (1H 2A6)                                                       LOGK0149
      IF (.NOT.TEST(8)) GO TO 321                                           LOGK0150
      IF (.NOT.TEST(13)) WRITE (6,322) ASINDH                               LOGK0151   200
  322 FORMAT (8HOHZERO = F12.3)                                             LOGK0152
      IF (TEST(13)) WRITE(6,323) ASINDH                                              202
  323 FORMAT (8HOH298 = F12.3 )                                             LOGK0153   204
  321 WRITE (6,220)                                                         LOGK0154
  220 FORMAT(1H ,77X,    42HREFERENCE ELEMENTS          GASEOUS ATOMS    )LOGK0155   205
      IF(.NOT.TEST(13))  WRITE(6,205)                                       LOGK0156
  205 FORMAT(123H      T        CP/R    (H-H0)/RT    S/R   -(F-H0)/RT     LOGK0157
     1H/RT        -F/RT     DELTA H/RT  -DELTA F/RT  DELTA H/RT -DELTA F/LOGK0158
     2RT       )                                                                207
      IF (TEST(13))  WRITE(6,1205)                                          LOGK0156
 1205 FORMAT(123H      T        CP/R  (H-H298)/RT   S/R  -(F-H298)/RT     LOGK
     1H/RT        -F/RT     DELTA H/RT  -DELTA F/RT  DELTA H/RT -DELTA F/LOGK
     2RT       )                                                           LOGK0159
      LINES = 10                                                            LOGK0160
  101 DO 229 NTABLE = 1,2                                                   LOGK0161
      DO 600  I=1,NT                                                        LOGK0162
      IT = T(I)                                                             LOGK0163
      FMT1(2) = FF(4)                                                       LOGK0164
      FMT2(2) = FF(4)                                                       LOGK0165
      IF (AMOD(T(I),1.0).EQ.0.) GO TO 103                                   LOGK0166
      T1 = T(I)                                                             LOGK0167
      FMT1(2) = FF(3)                                                       LOGK0168
      FMT2(2) = FF(3)                                                       LOGK0169
  103 RT = R*T(I)                                                           LOGK0170
      HOORT = ASINDH / RT                                                   LOGK0171
      SR = FHRT(I) + HHRT(I)                                                LOGK0172
      HRT = HHRT(I) + HOORT                                                 LOGK0173
      FRT = FHRT(I) - HOORT                                                 LOGK0174
      IF (NTABLE .EQ.2) GO TO 18                                            LOGK0175
C                                                                           LOGK0176
C630                                                                        LOGK0177
C                                                                           LOGK0178
      LD = 9                                                                LOGK0179
      DO 803  LL=1,2                                                        LOGK0180
      BZ = ZERO                                                             LOGK0181
      IF(DHO(LL).EQ.ZERO4)  GO TO 26
```

```
      BZ = BLK                                                          LOGK0182
      IF(DHO(LL).EQ.BLK)  GO TO 26                                      LOGK0183
      IF((I.GT.NTX(LL).AND.NTX(LL).NE.0).OR.I.LT.INIT(LL))  GO TO 26    LOGK0184
      D(LL,1,I) = D(LL,1,I) + DHO(LL)/RT                                LOGK0185
      D(LL,2,I) = D(LL,2,I) - DHO(LL)/RT                                LOGK0186
      FMT1(LD) = FMT1(7)                                                LOGK0187
      GO TO 803                                                         LOGK0188
   26 D(LL,1,I) = BZ                                                    LOGK0189
      D(LL,2,I) = BZ                                                    LOGK0190
      FMT1(LD) = FF(5)                                                  LOGK0191
  803 LD = 11                                                           LOGK0192
      FMT1(10) = FMT1(9)                                                LOGK0193
      FMT1(12) = FMT1(11)                                               LOGK0194
      GO TO 217                                                         LOGK0195
C                                                                       LOGK0196
C640                                                                    LOGK0197
C                                                                       LOGK0198
C                                                                       LOGK0199
C  CALCULATE DIMENSIONAL PROPERTIES, DELTAH, AND LOGK                   LOGK0200
C                                                                       LOGK0201
   18 CP = CPR(I) * R                                                   LOGK0202
      HH = HHRT(I) * RT                                                 LOGK0203
      S = SR * R                                                        LOGK0204
      H = HRT * RT                                                      LOGK0205
      FH = FHRT(I) * RT                                                 LOGK0206
      F = FRT * RT                                                      LOGK0207
      DO 402 JX = 8,11                                                  LOGK0208
  402 FMT2(JX) = FF(6)                                                  LOGK0209
      LD = 8                                                            LOGK0210
      DO 404  LL=1,2                                                    LOGK0211
      IF(D(LL,1,I).EQ.BLK)  GO TO 404                                   LOGK0212
      IF(D(LL,1,I).NE.ZERO)  GO TO 403                                  LOGK0213
      D(LL,1,I) = ZERO4                                                 LOGK0214
      D(LL,2,I) = ZERO4                                                 LOGK0215
      GO TO 404                                                         LOGK0216
  403 D(LL,2,I) = D(LL,2,I)/2.3025851                                   LOGK0217
      D(LL,1,I) = D(LL,1,I)*RT                                          LOGK0218
      FMT2(LD) = FMT2(7)                                                LOGK0219
      FMT2(LD+1) = FF(7)                                                LOGK0220
  404 LD = 10                                                           LOGK0221
  217 FMT1(1) = FF(1)                                                   LOGK0222
      IF (IK.EQ.0) GO TO 2999                                           LOGK0223
      DO 104 IX = 1,IK                                                  LOGK0224
      IF (MARK(IX).EQ.1) FMT1(1) = FF(2)                                LOGK0225
  104 CONTINUE                                                          LOGK0226
 2999 FMT2(1) = FMT1(1)                                                 LOGK0227
      IF (NTABLE .EQ.2) GO TO 235                                       LOGK0228
                        WRITE(6,FMT1) TI,CPR(I),HHRT(I),SR,FHRT(I),     LOGK0229
     1 HRT, FRT, (D(LL,1,I),D(LL,2,I),LL=1,2)                           LOGK0230    312
      GO TO 236                                                         LOGK0231
  235 WRITE(6,FMT2)TI,CP,HH,S,FH,H,((D(LL,KK,I),KK=1,2),LL=1,2)         LOGK0232    322
  236 LINES = LINES+1                                                   LOGK0233
      IF (AMOD(T(I),500.0).NE.0.0) GO TO 600                            LOGK0234
      WRITE (6,237)                                                     LOGK0235    335
  237 FORMAT (1H  )                                                     LOGK0236
      LINES = LINES+1                                                   LOGK0237
  600 IF(LINES.GE.55)  CALL PAGE ID(LINES)                              LOGK0238
C                                                                       LOGK0239
C650                                                                    LOGK0240
C                                                                       LOGK0241    338
  102 IF (IK.EQ.0) GO TO 601                                            LOGK0242
C                                                                       LOGK0243
C     WRITE FOOTNOTE                                                    LOGK0244
C                                                                       LOGK0245
      WRITE (6,265)                                                     LOGK0246    343
  265 FORMAT(     114H0*A CHANGE IN PHASE OF AN ASSIGNED REFERENCE ELEMENLOGK0247
     1T HAS OCCURRED BETWEEN THIS TEMPERATURE AND THE PRECEDING ONE,    )LOGK0248
      WRITE (6,267) (LMENT(I), PT(I), I=1,IK)                           LOGK0249    344
  267 FORMAT (1H A6,3H--  F8.3, 4H       )                             LOGK0250
      LINES = LINES + 4                                                 LOGK0251
  601 CALL PAGEID (LINES)                                               LOGK0252    351
      IF (NTABLE .EQ.2) RETURN                                          LOGK0253
C                                                                       LOGK0254
C  WRITE HEADING OF 2ND TABLE AND PROPERTIES FOR 0 DEGREES              LOGK0255
C                                                                       LOGK0256
      LINES = 7
      WRITE (6,320) NAME(1), NAME(2)                                    LOGK0257    355
      WRITE (6,220)                                                     LOGK0258    356
      IF(.NOT.TEST(13))  GO TO 1221
      WRITE(6,3221)                                                             359
 3221 FORMAT(120H      T         CP        H-H298          S        -(F-H298LOGK0260
     1)         H          DELTA H       LOG K        DELTA H       LOG K )LOGK0261
      GO TO 229
 1221 WRITE (6,221)                                                     LOGK0259    361
  221 FORMAT(120H      T         CP         H-H0           S         -(F-H0)LOGK0260
     1         H          DELTA H       LOG K        DELTA H       LOG K )LOGK0261
      S1 = DS1                                                          LOGK0262
      S2 = DS2                                                          LOGK0263
      IF (TEST(4))  GO TO 1230                                          LOGK0264
      S1 = ZERO4                                                        LOGK0265
      S2 = BLK                                                          LOGK0266
 1230 WRITE (6,FMT3) S1,S2,ASINDH, DHO(1), DHO(2)                       LOGK0268    365
  229 CONTINUE
      RETURN                                                            LOGK0269
      END                                                               LOGK0270
```

```
      SUBROUTINE LEAST                                                  LEAS0001
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),  LEAS0002
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),BLANK,ELEMNT(70), LEAS0003
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),LEAS0004
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       LEAS0005
     4       SPECH,TAPE(202,3),PTMELT,EXP(10),TRANGE(10),TCONST,NKIND,  LEAS0006
     5       NF,LINES,NTRANG,NTMP,AG(70),GG(70),NIT                     LEAS0007
      COMMON/PCH/LEVEL,NF1,NF2,ANS(9,15),TC(10),NTC,NFP,LDATE,NNN,NLAST
      DIMENSION A(15,16),            ANSTPY(15),F(4),FC(4),ERR(4),      LEAS0009
     1TOTERR(4),TOTREL(4),TOTSQ(4),          TOTSQR(4),AVERR(4),        LEAS0010
     2AVREL(4),AVSQ(4),AVSQR(4),MAXERR(4),MAXREL(4),TMAX(4),TMAXRL(4),  LEAS0011
     3RELERR(4)                                                         LEAS0012
C                                                                       LEAS0013
C660                                                                    LEAS0014
C                                                                       LEAS0015
      LOGICAL TEST                                                      LEAS0016
      REAL MAXERR,MAXREL                                                LEAS0017
      WRITE (6,2)                                                       LEAS0018    1
    2 FORMAT (1H  // 14H LEAST SQUARES ,//)                             LEAS0019
      LINES = 7                                                         LEAS0020
      DO 3 I = 1,3                                                      LEAS0021
      DO 3 J = 1,15                                                     LEAS0022
    3 ANS(I,J) = 0.0                                                    LEAS0023
      IF (NF .NE. 0 ) GO TO 6                                           LEAS0024
      NF = 5                                                            LEAS0025
      DO 4 I = 1,5                                                      LEAS0026
    4 EXP(I) = I - 1                                                    LEAS0027
    6 NF1 = NF+1                                                        LEAS0028
      NF2 = NF+2                                                        LEAS0029
      NF3 = NF+3                                                        LEAS0030
      NF4 = NF+4                                                        LEAS0031
      NF5 = NF+5                                                        LEAS0032
      NF6 = NF+6                                                        LEAS0033
      IDONEA = 0                                                        LEAS0034
      IDONEB = 0                                                        LEAS0035
      IDONET = 0                                                        LEAS0036
      IDONES = 0                                                        LEAS0037
      ICONST = 0                                                        LEAS0038
      IF (TCONST .NE. 0.0) GO TO 7                                      LEAS0044
      TCONST = 1000.                                                    LEAS0045
      IF (PTMELT .NE. 0.0) TCONST = PTMELT                              LEAS0046
    7 IF(NTRANG.EQ.0)  GO TO 1006                                       LEAS0047
C                                                                       LEAS0048
C670                                                                    LEAS0049
C  SORT IN INCREASING ORDER TEMPERATURES SPECIFYING INTERVALS           LEAS0050
C                                                                       LEAS0051
    8 J = 1                                                             LEAS0052
    9 M = J                                                             LEAS0053
   10 DO 12  I=J,NTRANG                                                 LEAS0054
      IF (TRANGE (M)-TRANGE(I)) 12,12,11                                LEAS0055
   11 M = I                                                             LEAS0056
   12 CONTINUE                                                          LEAS0057
      IF (M-J) 13,14,13                                                 LEAS0058
   13 TEMPY = TRANGE(M)                                                 LEAS0059
      TRANGE(M) = TRANGE(J)                                             LEAS0060
      TRANGE(J) = TEMPY                                                 LEAS0061
      GO TO 10                                                          LEAS0062
   14 J = J+1                                                           LEAS0063
      IF (NTRANG-J) 1007,1007,9                                         LEAS0064
 1006 TRANGE(1) = 300.0                                                 LEAS0065
      TRANGE(2) = 1000.                                                 LEAS0066
      TRANGE(3) = 5000.0                                                LEAS0067
      NTRANG = 3                                                        LEAS0068
 1007 DO 24 I = 1, NTRANG                                               LEAS0069
   24 TC(I) = TRANGE(I)                                                 LEAS0070
      NTC = NTRANG                                                      LEAS0071
      SAVEC = TCONST                                                    LEAS0072
      IF(TCONST.GT.T(NLAST))TCONST =T(NLAST)
      IF (NNN.EQ.1)  GO TO 1023
      IF (T(NNN-1).EQ.T(NNN).OR.TCONST.LT.T(NNN))  TCONST=T(NNN)
C                                                                       LEAS0093
C680                                                                    LEAS0094
C                                                                       LEAS0095
 1023 K = NTRANG - 1                                                    LEAS0109
      IF ((TRANGE(1).GE.T(NNN)-.00001).AND.(TRANGE(NTC).LE.T(NLAST)+
     1    .00001)) GO TO 1021
      LINES = LINES + 2                                                 LEAS0115
      DO 1028 I = 1,K                                                   LEAS0116
      IF (TRANGE(I).GE.T(NNN)-.00001)GO TO 1035                         LEAS0117
      IF (T(NNN).LT.TRANGE(I + 1) -.00001) GO TO 1032
      IDONES = IDONES + 1                                               LEAS0119
 1028 CONTINUE                                                          LEAS0120
 1032 TRANGE(I) = T(NNN)                                                LEAS0121
 1035 DO 1038 I = 1,K                                                   LEAS0122
      IK = K + 2 - I                                                    LEAS0123
      IF (TRANGE(IK).LE.T(NLAST) +.00001) GO TO 1021
      IF (T(NLAST) .GT. TRANGE(IK-1) +.00001) GO TO 1042
      NTRANG = NTRANG - 1                                               LEAS0126
 1038 CONTINUE                                                          LEAS0127
```

```
      GO TO 1021                                                        LEAS0123
 1042 TRANGE(IK) = T(NLAST)                                             LEAS0124
C                                                                       LEAS0092
C  LOCATE TEMPERATURE CONSTRAINTS
 1021 DO 17  I=1,NTRANG
      IF(ABS(TRANGE(I)-TCONST).LT.0.00001) GO TO 1017                   LEAS0094
      IF (TRANGE(I).GT.TCONST)  GO TO 18                                LEAS
   17 CONTINUE                                                          LEAS0095
      GO TO 1018
   18 I = I - 1
 1018 TRANGE(I) = TCONST
 1017 ICONST = I                                                        LEAS0100
C
C  ADJUST TEMPERATURE INTERVALS, IF NECESSARY
C
      DO 21 I=NNN,NLAST                                                 LEAS0127
   21 IF (ABS(T(I)-TCONST).LT.0.00001) GO TO 23                         LEAS0128
   23 CPRCON = CPR(I)                                                   LEAS0133
      HHRTCN = HHRT(I)                                                  LEAS0134
      SRCON  = FHRT(I) + HHRT(I)                                        LEAS0135
C                                                                       LEAS0136
C690                                                                    LEAS0137
C                                                                       LEAS0138
C  IF ALL INTERVALS ARE COMPLETE, RETURN TO MAIN. OTHERWISE LOCATE      LEAS0139
C     CONSTRAINT TEMPERATURE AND CURRENT INTERVAL END POINTS.           LEAS0140
C                                                                       LEAS0141
   25 ILOW = ICONST-IDONEB-1                                            LEAS0142
      IF((ILOW-IDONES).EQ.0) GO TO 27                                   LEAS0143
      TFIX = TRANGE (ILOW+1)                                            LEAS0144
      GO TO 28                                                          LEAS0145
   27 ILOW = ICONST+IDONEA                                              LEAS0146
      IF (ILOW.EQ.NTRANG) GO TO 900                                     LEAS0147
      TFIX = TRANGE(ILOW)                                               LEAS0148
   28 IF (ABS(TFIX-TCONST).GT.0.00001) GO TO 40                         LEAS0149
      CPRFIX = CPRCON                                                   LEAS0150
      HHRTFX = HHRTCN                                                   LEAS0151
      SRFIX = SRCON                                                     LEAS0152
   40 DO 41  I=NNN,NLAST                                                LEAS0153
      IF (ABS(T(I)-TRANGE(ILOW)).LT.0.00001) GO TO 44                   LEAS0154
   41 CONTINUE                                                          LEAS0155
      WRITE (6,42)                                                      LEAS0156    203
   42 FORMAT (95H LEAST SQUARES NOT COMPLETED. INTERVAL TEMPERATURES NOTLEAS0157
     1 FOUND IN TEMPERATURE SCHEDULE, C690     )                      LEAS0158
      GO TO 1000                                                        LEAS0159
   44 NBEGIN = I                                                        LEAS0160
      DO 46  I=NBEGIN,NLAST                                             LEAS0161
      IF (ABS(T(I)-TRANGE(ILOW+1)).LT.0.00001) GO TO 48                 LEAS0162
   46 CONTINUE                                                          LEAS0163
      WRITE (6,42)                                                      LEAS0164    217
   48 NEND = I                                                          LEAS0165
C                                                                       LEAS0166
C700                                                                    LEAS0167
C                                                                       LEAS0168
C CLEAR MATRIX REGION                                                   LEAS0169
C                                                                       LEAS0170
   50 DO 51  I=1,NF5                                                    LEAS0171
      DO 51  J=1,NF6                                                    LEAS0172
   51 A(I,J) = 0.0                                                      LEAS0173
C                                                                       LEAS0174
C SET UP MATRIX ELEMENTS FOR DIAGONAL AND ABOVE FOR FIRST NF ROWS       LEAS0175
C                                                                       LEAS0176
      K = 1                                                             LEAS0177
      DO 500  I=1,NF                                                    LEAS0178
   80 IF (EXP(I).NE.(-1.0)) GO TO 85                                    LEAS0179
      A(I,NF3) = 1.0/TFIX                                               LEAS0180
      A(I,NF4) = ALOG(TFIX)/TFIX                                        LEAS0181    239
      A(I,NF5) = -1.0/TFIX                                              LEAS0182
      DO 83  L=NBEGIN,NEND                                              LEAS0183
      A(I,NF1) = A(I,NF1)+ALOG(T(L))/(T(L)*T(L))                        LEAS0184    245
      A(I,NF2) = A(I,NF2)-1.0/T(L)                                      LEAS0185
      SR = FHRT(L) + HHRT(L)                                            LEAS0186
   83 A(I,NF6) = A(I,NF6) + (CPR(L)+HHRT(L)*ALOG(T(L))-SR    )/T(L)      LEAS0187    257
      GO TO 99                                                          LEAS0188
C                                                                       LEAS0189
   85 IF (EXP(I).NE.0.0) GO TO 90                                       LEAS0190
      A(I,NF3 ) = 1.0                                                   LEAS0191
      A(I,NF4 ) = 1.0                                                   LEAS0192
      A(I,NF5) = ALOG(TFIX)                                             LEAS0193    271
      DO 89  L=NBEGIN,NEND                                              LEAS0194
      A(I,NF1) = A(I,NF1) + 1.0/T(L)                                    LEAS0195
      A(I,NF2) = A(I,NF2) + ALOG(T(L))                                  LEAS0196    279
      SR = FHRT(L) + HHRT(L)                                            LEAS0197
   89 A(I,NF6) = A(I,NF6) + CPR(L)+HHRT(L)+SR    *ALOG(T(L))           LEAS0198    286
      GO TO 99                                                          LEAS0199
C                                                                       LEAS0200
   90 A(I,NF3 ) = TFIX**EXP(I)                                          LEAS0201    295
      A(I,NF4) = A(I,NF3)/(EXP(I)+1.0)                                  LEAS0202
      A(I,NF5) = A(I,NF3)/EXP(I)                                        LEAS0203
```

```
      DO 92  L=NBEGIN,NEND                                                  LEAS0204
      A(I,NF1) = A(I,NF1)+T(L)**(EXP(I)-1.0)/(EXP(I)+1.0)                   LEAS0205    308
      A(I,NF2) = A(I,NF2)+(T(L)**EXP(I))/EXP(I)                             LEAS0206    312
      SR = FHRT(L) + HHRT(L)                                                LEAS0207
   92 A(I,NF6) = A(I,NF6)+(CPR(L)+HHRT(L)/(EXP(I)+1.0)+SR   /EXP(I))        LEAS0208
     1*(T(L)**EXP(I))                                                       LEAS0209
C                                                                           LEAS0210
C710                                                                        LEAS0211
C                                                                           LEAS0212    325
   99 DO 400  J=K,NF                                                        LEAS0213
  100 IF (EXP(I)+1.0) 130,105,130                                           LEAS0214
  105 IF (EXP(J)+1.0) 115,110,115                                           LEAS0215
  110 DO 112  L=NBEGIN,NEND                                                 LEAS0216
  112 A(I,J) = A(I,J)+(2.0+ALOG(T(L))*ALOG(T(L)))/(T(L)*T(L))               LEAS0217    344    346
      GO TO 400                                                             LEAS0218
C                                                                           LEAS0219
  115 IF (EXP(J)) 125,120,125                                               LEAS0220
  120 DO 122  L=NBEGIN,NEND                                                 LEAS0221
  122 A(I,J) = A(I,J) + 1.0/T(L)                                            LEAS0222
      GO TO 400                                                             LEAS0223
C                                                                           LEAS0224
  125 EXPIJ = EXP(J)                                                        LEAS0225
  126 DO 127  L=NBEGIN,NEND                                                 LEAS0226
  127 A(I,J) = A(I,J) + ((EXPIJ-1.0)/EXPIJ+ALOG(T(L))/(EXPIJ+1.0))/         LEAS0227
     1T(L)**(EXPIJ-1.0)                                                     LEAS0228    372    374
      GO TO 400                                                             LEAS0229
C                                                                           LEAS0230
  130 IF (EXP(J) + 1.0) 145,135,145                                         LEAS0231
  135 IF (EXP(I)) 140,120,140                                               LEAS0232
  140 EXPIJ = EXP(I)                                                        LEAS0233
      GO TO 126                                                             LEAS0234
C                                                                           LEAS0235
  145 IF (EXP(J)) 165,150,165                                               LEAS0236
  150 IF (EXP(I)) 160,155,160                                               LEAS0237
  155 DO 157  L=NBEGIN,NEND                                                 LEAS0238
  157 A(I,J) = A(I,J) + 2.0+ALOG(T(L))**2                                   LEAS0239    399
      GO TO 400                                                             LEAS0240
C                                                                           LEAS0241
  160 EXPIJ = EXP(I)                                                        LEAS0242
  161 DO 163  L=NBEGIN,NEND                                                 LEAS0243
  163 A(I,J) = A(I,J) + ((EXPIJ+2.0)/(EXPIJ+1.0)+ALOG(T(L))/EXPIJ)          LEAS0244
     1  *T(L)**EXPIJ                                                       LEAS0245    411    413
      GO TO 400                                                             LEAS0246
C                                                                           LEAS0247
  165 IF (EXP(I)) 175,170,175                                               LEAS0248
  170 EXPIJ = EXP(J)                                                        LEAS0249
      GO TO 161                                                             LEAS0250
  175 DO 177 L = NBEGIN,NEND                                                LEAS0251
  177 A(I,J) = A(I,J)+(1.0+1.0/((EXP(I)+1.0)*(EXP(J)+1.0))                  LEAS0252
     1+ 1.0/(EXP(I)*EXP(J)))*T(L)**(EXP(I)+EXP(J))                          LEAS0253
C                                                                           LEAS0254    435
  400 CONTINUE                                                              LEAS0255
  500 K = K+1                                                               LEAS0256
C                                                                           LEAS0257
C SET UP MATRIX FOR DIAGONAL AND ABOVE FOR REMAINING ROWS                   LEAS0258
C                                                                           LEAS0259
      DO 510  L=NBEGIN,NEND                                                 LEAS0260
      A(NF1,NF1) = A(NF1,NF1) + 1.0/(T(L)*T(L))                             LEAS0261
      A(NF1,NF6) = A(NF1,NF6) + HHRT(L)/T(L)                                LEAS0262
      A(NF2,NF2) = A(NF2,NF2) + 1.0                                         LEAS0263
  510 A(NF2,NF6) = A(NF2,NF6) + FHRT(L) + HHRT(L)                           LEAS0264
      A(NF1,NF4) = 1.0/TFIX                                                 LEAS0265
      A(NF2,NF5) = 1.0                                                      LEAS0266
      A(NF3,NF6) = CPRFIX                                                   LEAS0267
      A(NF4,NF6) = HHRTFX                                                   LEAS0268
      A(NF5,NF6) = SRFIX                                                    LEAS0269
C                                                                           LEAS0270
C720                                                                        LEAS0271
C                                                                           LEAS0272
C COMPLETE THE MATRIX BY REFLECTING SYMMETRICAL ELEMENTS ABOVE DIAGONAL     LEAS0273
C                                                                           LEAS0274
      K = 2                                                                 LEAS0275
      DO 520  I=1,NF4                                                       LEAS0276
      DO 518  J=K,NF5                                                       LEAS0277
  518 A(J,I) = A(I,J)                                                       LEAS0278
  520 K = K+1                                                               LEAS0279
C                                                                           LEAS0280
C SOLVE THE MATRIX.                                                         LEAS0281
C                                                                           LEAS0282
      N=NF5                                                                 LEAS0283
      DO 551 I=1,N                                                          LEAS0284
  551 ANSTPY(I) = 0.0                                                       LEAS0285
      DO 560 I=1,N                                                          LEAS0286
      DO 557 J=I,N                                                          LEAS0287
      A(I,J+1) = A(I,J+1)/A(I,I)                                            LEAS0288
      IF (I-N) 557,570,557                                                  LEAS0289
  557 CONTINUE                                                              LEAS0290
      K=I+1                                                                 LEAS0291
```

```
      DO 558 II=K,N                                                    LEAS0292
      DO 558 JJ=I,N                                                    LEAS0293
  558 A(II,JJ+1) = -A(II,I)*A(I,JJ+1) + A(II,JJ+1)                     LEAS0294
  560 CONTINUE                                                         LEAS0295
  570 ANSTPY(N) = A(I,J+1)                                             LEAS0296
      IF (N-1) 571,580,571                                             LEAS0297
  571 J = N-1                                                          LEAS0298
      II = J                                                           LEAS0299
      DO 573 I=1,II                                                    LEAS0300
      K = J+1                                                          LEAS0301
      DO 572 MM=1,I                                                    LEAS0302
      ANSTPY(J) = ANSTPY(J) + ANSTPY(K)*A(J,K)                         LEAS0303
  572 K = K+1                                                          LEAS0304
      ANSTPY(J) = A(J,K) - ANSTPY(J)                                   LEAS0305
  573 J=J-1                                                            LEAS0306
C                                                                      LEAS0307
C                                                                      LEAS0308
  580 DO 581 I=1,NF5                                                   LEAS0309
  581 ANS(ILOW,I) = ANSTPY(I)                                          LEAS0310
C                                                                      LEAS0311
C730                                                                   LEAS0312
C                                                                      LEAS0313
C CALCULATE FROM THE LEAST SQUARES COEFFICIENTS VALUES OF CP/R,H-HO/RT, LEAS0314
C   S/R,F-HO/RT, AND THE ERRORS AND RELATIVE ERRORS IN THESE AT EACH   LEAS0315
C   TEMPERATURE. ALSO THE AVERAGE ERROR, AVERAGE RELATIVE ERROR,       LEAS0316
C   LARGEST ERROR AND LARGEST RELATIVE ERROR.                          LEAS0317
C                                                                      LEAS0318
      WRITE (6,602)                                                    LEAS0319    553
  602 FORMAT ( 1HO,7X,1HT,6X,10HCP/R INPUT,4X,9HCP/R CALC,5X,11HHH/RT INLEAS0320
     1PUT,3X,10HHH/RT CALC,6X,9HS/R INPUT,5X,8HS/R CALC,6X,12H-FH/RT INPLEAS0321
     2UT,4X,11H-FH/RT CALC  )                                          LEAS0322
      WRITE (6,603)                                                    LEAS0323    554
  603 FORMAT (1H ,14X,10HINPUT-CALC,5X,8HFRACTION,6X,10HINPUT-CALC,4X,8HLEAS0324
     1FRACTION,7X,10HINPUT-CALC,5X,8HFRACTION,7X,10HINPUT-CALC,4X,8HFRACLEAS0325
     2TION)                                                            LEAS0326
C                                                                      LEAS0327
      LINES = LINES + 3                                                LEAS0328
      DO 605  I=1,4                                                    LEAS0329
      TOTERR(I) = 0.                                                   LEAS0330
      TOTREL(I) = 0.                                                   LEAS0331
      TOTSQR(I) = 0.                                                   LEAS0332
      TOTSQ (I) = 0.                                                   LEAS0333
      MAXERR(I) = 0.                                                   LEAS0334
      MAXREL(I) = 0.                                                   LEAS0335
      TMAX(I) = 0.                                                     LEAS0336
  605 TMAXRL(I) = 0.                                                   LEAS0337
C                                                                      LEAS0338
C740                                                                   LEAS0339
C                                                                      LEAS0340
      DO 635  L=NBEGIN,NEND                                            LEAS0341
      F(1) = CPR(L)                                                    LEAS0342
      F(2) = HHRT(L)                                                   LEAS0343
      F(3) = FHRT(L) + HHRT(L)                                         LEAS0344
      F(4) = FHRT(L)                                                   LEAS0345
      FC(1) = 0.                                                       LEAS0346
      FC(2) = ANSTPY(NF1)/T(L)                                         LEAS0347
      FC(3) =ANSTPY(NF2)                                               LEAS0348
C                                                                      LEAS0349
      DO 618  I=1,NF                                                   LEAS0350
      TP = T(L)**EXP(I)                                                LEAS0351    584
      IF (EXP(I).NE.(-1.0)) GO TO 610                                  LEAS0352
      FC(2) = FC(2)+ANSTPY(I)*ALOG(T(L))/T(L)                          LEAS0353    590
      GO TO 616                                                        LEAS0354
  610 FC(2) = FC(2)+ANSTPY(I)*TP/(EXP(I)+1.0)                          LEAS0355
      IF (EXP(I).NE.0.0) GO TO 616                                     LEAS0356
      FC(3) = FC(3)+ANSTPY(I)*ALOG(T(L))                               LEAS0357    602
      GO TO 618                                                        LEAS0358
  616 FC(3) = FC(3)+ANSTPY(I)*TP/EXP(I)                                LEAS0359
  618 FC(1) = FC(1) + ANSTPY(I)*TP                                     LEAS0360
      FC(4) = FC(3)-FC(2)                                              LEAS0361
C                                                                      LEAS0362
      IF (L.NE.NBEGIN.OR.TRANGE(ILOW).GE.TCONST)  GO TO 705
      IDONEB = IDONEB + 1
      GO TO 706
  705 IF (L.NE.NEND.OR.TRANGE(ILOW).LT.TCONST)  GO TO 707
      IDONEA = IDONEA + 1
  706 CPRFIX = FC(1)
      HHRTFX = FC(2)
      SRFIX = FC(3)
  707 DO 622  I=1,4                                                    LEAS0363
      ERR(I) = F(I)-FC(I)                                              LEAS0364
      ABSERR = ABS(ERR(I))                                             LEAS0365
      TOTERR(I) = TOTERR(I)+ABSERR                                     LEAS0366
      TOTSQ (I) = TOTSQ (I)+ABSERR*ABSERR                              LEAS0367
      IF(F(I).NE.0.)  GO TO 619
      WRITE(6,1619)                                                                636
 1619 FORMAT (52HO ERROR IN DATA.  LEAST SQUARES NOT COMPLETED, C740  )
      GO TO 1000
```

```
  619 RELERR(I) = ERR(I)/F(I)                                          LEAS0368
      ABSREL = ABS(RELERR(I))                                          LEAS0369
      TOTREL(I) = TOTREL(I)+ABSREL                                     LEAS0370
      TOTSQR(I) = TOTSQR(I) + ABSREL*ABSREL                            LEAS0371
      IF (ABSERR.LT.MAXERR(I)) GO TO 620                               LEAS0372
      MAXERR(I) = ABSERR                                               LEAS0373
      TMAX(I) = T(L)                                                   LEAS0374
  620 IF (ABSREL .LT. MAXREL(I)) GO TO 622                             LEAS0375
      MAXREL(I) = ABSREL                                               LEAS0376
      TMAXRL(I) = T(L)                                                 LEAS0377
  622 CONTINUE                                                         LEAS0378
C                                                                      LEAS0379
      WRITE (6,625) T(L),CPR(L),FC(1),HHRT(L),FC(2),F(3), FC(3),F(4),  LEAS0380
     1FC(4)                                                            LEAS0381   662
  625 FORMAT (F12.2,2F13.7,2X,2F13.7,2X,2F14.7,2X,2F14.7)              LEAS0382
      WRITE (6,627)(ERR(I),RELERR(I),I=1,4)                            LEAS0383   666
  627 FORMAT ( 12X ,2F13.7,2X,2F13.7,2X,2F14.7,2X,2F14.7)              LEAS0384
      LINES = LINES + 2                                                LEAS0385
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0386   674
  635 CONTINUE                                                         LEAS0387
C                                                                      LEAS0388
C750                                                                   LEAS0389
C                                                                      LEAS0390
      POINTS = NEND-NBEGIN + 1                                         LEAS0391
      DO 640  I=1,4                                                    LEAS0392
      AVERR(I) = TOTERR(I)/POINTS                                      LEAS0393
      AVREL(I) = TOTREL(I)/POINTS                                      LEAS0394
      AVSQ(I) = SQRT(TOTSQ(I)/POINTS)                                  LEAS0395   687
  640 AVSQR(I) = SQRT(TOTSQR(I)/POINTS)                                LEAS0396
C                                                                      LEAS0397
C                                                                      LEAS0398   691
      WRITE (6,641) MAXREL(1),TMAXRL(1),AVREL(1),AVSQR(1)              LEAS0399   694
  641 FORMAT (3X,19HMAX-REL ERR CP/R  =,F10.6,4X,6HTEMP =,F7.0,6X,20HAVELEAS0400
     1R REL ERR CP/R  =,F10.6,6X,22HREL LST SQ ERR CP/R  =,F10.6)      LEAS0401
      WRITE (6,642) MAXREL(2),TMAXRL(2),AVREL(2),AVSQR(2)              LEAS0402   695
  642 FORMAT (3X,19HMAX REL ERR HH/RT =,F10.6,4X,6HTEMP =,F7.0,6X,20HAVELEAS0403
     1R REL ERR HH/RT =,F10.6,6X,22HREL LST SQ ERR HH/RT =,F10.6)      LEAS0404
      LINES = LINES + 2                                                LEAS0405
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0406   698
      WRITE (6,643) MAXREL(3),TMAXRL(3),AVREL(3),AVSQR(3)              LEAS0407   699
  643 FORMAT (3X,19HMAX REL ERR S/R   =,F10.6,4X,6HTEMP =,F7.0,6X,20HAVELEAS0408
     1R REL ERR S/R   =,F10.6,6X,22HREL LST SQ ERR S/R   =,F10.6)      LEAS0409
      WRITE (6,644) MAXREL(4),TMAXRL(4),AVREL(4),AVSQR(4)              LEAS0410   700
  644 FORMAT (3X,19HMAX REL ERR FH/RT =,F10.6,4X,6HTEMP =,F7.0,6X,20HAVELEAS0411
     1R REL ERR FH/RT =,F10.6,6X,22HREL LST SQ ERR FH/RT =,F10.6 )     LEAS0412
      LINES = LINES + 2                                                LEAS0413
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0414   703
      WRITE (6,645) MAXERR(1),TMAX(1),AVERR(1),AVSQ(1)                 LEAS0415   704
  645 FORMAT (7X,15HMAX ERR CP/R  =,F10.6,4X,6HTEMP.=,F7.0,10X,16HAVER ELEAS0416
     1RR CP/R  =,F10.6,10X,18HLST SQ ERR CP/R  =,F10.6)                LEAS0417
      WRITE (6,646) MAXERR(2),TMAX(2),AVERR(2),AVSQ(2)                 LEAS0418   705
  646 FORMAT (7X,15HMAX ERR HH/RT =,F10.6,4X,6HTEMP =,F7.0,10X,16HAVER ELEAS0419
     1RR HH/RT =,F10.6,10X,18HLST SQ ERR HH/RT =,F10.6)                LEAS0420
      LINES = LINES + 2                                                LEAS0421
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0422   708
      WRITE (6,647) MAXERR(3),TMAX(3),AVERR(3),AVSQ(3)                 LEAS0423   709
  647 FORMAT (7X,15HMAX ERR S/R   =,F10.6,4X,6HTEMP =,F7.0,10X,16HAVER ELEAS0424
     1RR S/R   =,F10.6,10X,18HLST SQ ERR S/R   =,F10.6)                LEAS0425
      WRITE (6,648) MAXERR(4),TMAX(4),AVERR(4),AVSQ(4)                 LEAS0426   710
  648 FORMAT (7X,15HMAX ERR FH/RT =,F10.6,4X,6HTEMP =,F7.0,10X,16HAVER ELEAS0427
     1RR FH/RT =,F10.6,10X,18HLST SQ ERR FH/RT =,F10.6  )              LEAS0428
      LINES = LINES + 2                                                LEAS0429
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0430   713
      WRITE (6,650) (ANSTPY(I),EXP(I),I=1,NF)                          LEAS0431   714
  650 FORMAT ( 8H CP/R = ,5(1PE16.7,3HT**,0PF4.1)/8X,5(1PE16.7,3HT**,0PFLEAS0432
     14.1))                                                            LEAS0433
      LINES = LINES + 2                                                LEAS0434
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0435   724
      HRTC = ANSTPY(NF1) + ASINDH/R                                    LEAS0436
      WRITE (6,660) ANSTPY(NF1), HRTC, ANSTPY(NF2)                     LEAS0437   727
  660 FORMAT (21H (H-HO)/R CONSTANT = ,E15.8, 20H, H/R(A6) CONSTANT =  LEAS0438
     1 E15.8, 16H, S/R CONSTANT = E15.8                )               LEAS0439
      LINES = LINES + 2                                                LEAS0440
      IF (LINES .GE.55) CALL PAGEID (LINES)                            LEAS0441   732
      GO TO 25                                                         LEAS0455
C                                                                      LEAS0442
C760                                                                   LEAS0443
C                                                                      LEAS0444
  900 SAVEL = LEVEL                                                    LEAS0456
      LEVEL = NTC-1
      NFP = NF                                                         LEAS0457
      CALL PUNCH                                                       LEAS0458   738
      LEVEL = SAVEL
 1000 DO 980 I = 1,NTC
  980 TRANGE(I) = TC(I)                                                LEAS0463
      NTRANG = NTC                                                     LEAS0464
      TCONST = SAVEC                                                   LEAS0465
      RETURN
      END                                                              LEAS0471
```

```
      SUBROUTINE PUNCH                                                  PNCH0001
C                                                                       PNCH0002
C  PUNCH COEFFICIENT CARDS FOR PERFORMANCE PROGRAM                      PNCH0003
      COMMON NAME(2),SYMBOL(70),ATMWT(70),R,HCK,ELECTR,ICARD,IWORD(5),  PNCH0004
     1       WORD(4),TEST(20),WEIGHT,FORMLA(5),MLA(5),NPLUS,ELEMNT(70), PNCH0005
     2       NATOM,NT,CPR(202),HHRT(202),ASINDH,T(202),ASINDT,FHRT(202),PNCH0006
     3       SCONST,NOATMS,MPLACE(70),LPLACE(70),NMLA(70),NDFILE,       PNCH0007
     4       SPECH,TAPE(202,3),PTMELT,EXP(10),TRANGE(10),TCONST,NKIND,  PNCH0008
     5       NF,LINES,ITR,NTMP,AG(70),GG(70),NIT,PI                     PNCH0009
C                                                                       PNCH0010
C770                                                                    PNCH0011
C                                                                       PNCH0012
      COMMON /PCH/ K,NF1, NF2, ANS(9,15),TC(10), NTC,NFP,DATE,NNN,NLAST PNCH0013
      DIMENSION DAT (23),MA(6)                                          PNCH0014
      LOGICAL TEST                                                      PNCH0015
      EQUIVALENCE (DAT(1),NDATA1), (DAT(2),NDATA2)                      PNCH0016
      DATA MA(1)/00077777777777/, LIQUID/6HL00000/,MA(2)/07700777777777/, PNCH0017
     1MA(3)/07777007777777/,MA(4)/07777770077777/,MA(5)/07777777700777/, PNCH0018
     2MA(6)/07777777777700/,BLANK/1H /                                  PNCH0019
      DO 44 IA = 1,K                                                    PNCH0020
   44 IF(ANS(IA,1).NE.0.0)ANS(IA,NF1) = ANS(IA,NF1) + ASINDH/R          PNCH0021
      NDATA1 = NAME(1)                                                  PNCH0022
      NDATA2 = NAME(2)                                                  PNCH0023
      DAT(3) = PI                                                       PNCH0024
      DAT(4) = TC(1)                                                    PNCH0025
      IF (.NOT.TEST(6) .OR.PTMELT.EQ.0.) GO TO 40                       PNCH0026
      IF (T(NLAST).LE.PTMELT) GO TO 40                                  PNCH0027
C     PUNCHING CARDS FOR LIQUID - - INSERT L IN NAME                    PNCH0028
      KN = 1                                                            PNCH0029
      NPLU = NPLUS                                                      PNCH0030
      IF (NPLUS .GT. 6) GO TO 30                                        PNCH0031
   20 LN = 6*(NPLU -1)                                                  PNCH0032
      IL = IARS(LN,LIQUID)                                              PNCH0033     29
      DAT(KN) = AND(MA(NPLU ),NAME(KN))                                 PNCH0034
      DAT(KN) = OR(DAT(KN),IL)                                          PNCH0035
      GO TO 40                                                          PNCH0036
   30 KN = 2                                                            PNCH0037
      NPLU  = NPLUS - 6                                                 PNCH0038
      GO TO 20                                                          PNCH0039
C                                                                       PNCH0040
C780                                                                    PNCH0041
C                                                                       PNCH0042
   40 DO 930  I=6,23                                                    PNCH0043
  930 DAT(I) = 0.                                                       PNCH0044
      IF (T(NLAST).LE.TC(1).OR.T(NNN).GT.TC(1))  DAT(4)=T(NNN)          PNCH0045
      DAT(5) = TC(NTC)                                                  PNCH0046
      IF (TC(NTC).GT.T(NLAST)) DAT(5) = T(NLAST)                        PNCH0047
  936 I1 = 7                                                            PNCH0048
      I2 = 16                                                           PNCH0049
      WRITE (6,954)                                                     PNCH0050     61
  954 FORMAT (24HOPUNCHED BINARY CARDS--  )                             PNCH0051
      LINES = LINES + 3                                                 PNCH0052
  940 DO 952 I = I1,I2,9                                                PNCH0053
      DAT(I-1) = TC(K)                                                  PNCH0054
      DAT(I) = TC(K+1)                                                  PNCH0055
      DO 950 J = 1,NFP                                                  PNCH0056
      IIJ = I + J                                                       PNCH0057
  950 DAT(IIJ) = ANS(K ,J)                                              PNCH0058
      DAT(I+6) = ANS (K,NF1)                                            PNCH0059
      DAT(I+7) = ANS (K,NF2)                                            PNCH0060
      K = K- 1                                                          PNCH0061
      IF (K.LE.0 .AND. I.EQ.I1) GO TO 970                               PNCH0062
  952 CONTINUE                                                          PNCH0063
      I = I2                                                            PNCH0064
  970 I8 = I+8                                                          PNCH0065
  951 IF(I8.GT.22)  GO TO 961                                           PNCH0066
      DO 953  I=I8,22                                                   PNCH0067
  953 DAT(I) =0.                                                        PNCH0068
      DAT(22) = DATE                                                    PNCH0069
C                                                                       PNCH0070
C790                                                                    PNCH0071
C                                                                       PNCH0072
  961 IF (DAT(3).EQ.0.) DAT(3)=BLANK                                    PNCH0073
      CALL BCDUMP (DAT(1), DAT(22))                                     PNCH0074     101
      IF (DAT(3).EQ.BLANK) DAT(3)=0.                                    PNCH0075
      IF (I8.GT.22)  GO TO 1970                                         PNCH0076
      WRITE (6,956) (DAT(II), II=1,22)                                  PNCH0077     107
  956 FORMAT (1H0,2A6,F17.3,4X,5E17.8/7E17.8/6E17.8,11X,A6 )            PNCH0078
      GO TO 957                                                         PNCH0079
 1970 WRITE (6,955) (DAT(II), II=1,22)                                  PNCH0080     115
  955 FORMAT (1H0,2A6,F17.3,4X,5E17.8/7E17.8/7E17.8 )                   PNCH0081
  957 LINES = LINES + 4                                                 PNCH0082
      IF (LINES.GE.55)  CALL PAGEID(LINES)                              PNCH0083     125
      IF (I8.EQ.5) GO TO 1000                                           PNCH0084
      IF (I8.EQ.24) I8 = 5                                              PNCH0085
      DAT(4) = DAT(23)                                                  PNCH0086
      IF (K.EQ.0) GO TO 1956                                            PNCH0087
      IF (I1.EQ.5)  GO TO 940                                           PNCH0088
      I1 = I1-1                                                         PNCH0089
      I2 = I1+9                                                         PNCH0090
      GO TO 940                                                         PNCH0091
 1956 IF(I8.EQ.5) GO TO 951                                             PNCH0092
 1000 CALL PAGEID(LINES)                                                PNCH0093     144
      RETURN                                                            PNCH0094
      END                                                               PNCH0095
```

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:

```
                $IBLDR SKFILE

                $TEXT  SKFILE

BINARY CARD (NOT PUNCHED)
      00000  1 00000 0 00004  10001  SKFILE SAVE    (4,2)
      00001  0774 00 2 00000  10000
      00002  0774 00 4 00000  10000
      00003  0020 00 4 00001  10000
      00004  0634 00 4 05000  10011
      00005  0634 00 4 00061  10001
      00006  0634 00 4 00002  10001
      00007  0634 00 2 00001  10001
      00010  0500 00 4 00001  10000         CLA     1,4
      00011  4734 00 2 00000  10000         PDX     0,2
      00012  7 00001 2 01004  10011         TXL     *+4,2,1
      00013  0500 60 4 00004  10000         CLA*    4,4
      00014  0734 00 2 00000  10000         PAX     0,2
      00015  7 00000 2 00052  10001         TXL     OUT,2,0
      00016  0500 60 4 00003  10000         CLA*    3,4
      00017  0621 00 0 00057  10001         STA     PAT
      00020  0634 00 2 00050  10001         SXA     EOF,2
      00021  000000000000    00010         CALL    ..FVIO(PAT,UNITAD)
      00021  0074 00 4 03000  10011

BINARY CARD (NOT 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         CLA*    UNITAD
      00027  0734 00 4 00000  10000         PAX     0,4
      00030  1 00001 4 01001  10011         TXI     *+1,4,1
      00031  0634 00 4 00045  10001         SXA     HOLD,4
      00032  0074 00 4 06000  10011         TSX     ..FTCK,4
      00033  4774 00 4 00052  10001         AXC     DUM-3,4
      00034  0634 00 4 05000  10011         SXA     SYSLOC,4
      00035  000000000000    00010         CALL    .BSF.
      00035  0074 00 4 04000  10011
      00036  1 00000 0 01002  10011
      00037  0 00061 0 00023  10100
      00040  0074 00 4 07000  10011  REED   TSX     ..FIOC,4
      00041  0074 00 4 10000  10011         TSX     ..FBCK,4
      00042  0020 00 0 00045  10001         TRA     HOLD
      00043  0020 00 0 00050  10001         TRA     EOF

BINARY CARD (NOT PUNCHED)
      00044  0020 00 0 00045  10001         TRA     HOLD
      00045  0520 00 0 00000  10000  HOLD   ZET     **
      00046  0020 00 0 41001  10011         TRA     *-1
      00047  0020 00 0 00040  10001         TRA     REED
      00050  0774 00 2 00000  10000  EOF    AXT     **,2
      00051  2 00001 2 01002  10011         TIX     *+2,2,1
                        00052               OUT    RETURN  SKFILE
      00053  0634 00 2 00050  10001         SXA     EOF,2
      00054  0020 00 0 00045  10001         TRA     HOLD
      00055  0 00000 0 00056  10001  DUM    PZE     UNITAD
      00056  0 00000 0 00000  10000  UNITAD PZE
      00057  0 00000 0 00000  10000  PAT    PZE
      00060  0 00000 0 00000  10000  EOB    PZE
      00061  000000000000    10000         *LDIR
      00062  624226314325    10000
                        00000  01111         END
```

```
                $CDICT SKFILE

BINARY CARD (NOT PUNCHED)
             000063000000              PREFACE    START=0,LENGTH=51,TYPE=7094,CMPLX=5
             000004000005
             624226314325       SKFILE DECK       LOC=0,LENGTH=51
             000063000000
             624226314325       SKFILE REAL       LOC=0,LENGTH=0
             000000000000
             333326653146       ..FVIO VIRTUAL    SECT. 3,CALL
             200000100000
             332262263360       .BSF.  VIRTUAL    SECT. 4,CALL
             200000100000
             627062434623       SYSLOC VIRTUAL    SECT. 5
             200000000000
             333326632342       ..FTCK VIRTUAL    SECT. 6
             200000000000
             333326314623       ..FIOC VIRTUAL    SECT. 7
             200000000000
             333326222342       ..FBCK VIRTUAL    SECT. 8
             200000000000

                $DKEND SKFILE                                              001938
```

```
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          DUM       00055    33
          EOB       00060
          EOF       00050    20,43,53
          HOLD      00045    31,42,44,54
          ..0001    00002    6,7
          ..0002    00003
          ..0003    00004    0
          OUT       00052    15
          PAT       00057    17,24
          REED      00040    47
  LCTR     BLCTR
  QUAL     UNQS
  LCTR    //
          SKFILE    00000    52
          UNITAD    00056    25,26,55

REFERENCES TO VIRTUAL SYMBOLS.

          .BSF.        4     35
          ..FBCK       8     41
          ..FIOC       7     40
          ..FTCK       6     32
          ..FVIO       3     21
          SYSLOC       5     4,34
```

```
                $IBLDR .BSF.

                $TEXT  .BSF.

                                                ENTRY   .BSF.
                       00003            SIZE    SET     3

BINARY CARD (NOT PUNCHED)
       00000  0500 00 0 00010  10001  .BSF.  CLA     PON
       00001  0634 00 4 00014  10001         SXA     LK.DR,4
       00002  000000000000     00010         CALL    ..FIOS(SEL)
       00002  0C74 00 4 03000  10011
       00003  1 00001 0 01003  10011
       00004  0 00014 0 00005  10100
       00005  0 00000 0 00005  10001
       00005  400005041001     00001         ORG     *-1
       00005  3 00003 0 00011  10001  SEL    IORT    ..BSF..,SIZE
       00006  0534 00 4 00014  10001         LXA     LK.DR,4
       00007  0C20 00 4 00001  10000         TRA     1,4
       00010  1 00000 0 00000  10000   PON   PON     0,,0
       00011  2C0000000003     00001  ..BSF. BSS     SIZE
       00014  0C0000GC0000     10000  LK.DR  LDIR
       00015  332262263360     10000
                       00000   01111         END
                $CDICT .BSF.


BINARY CARD (NOT PUNCHED)
              000016GC0000                PREFACE     START=0,LENGTH=14,TYPE=7094,CMPLX=5
              000004000005
              332262263360        .BSF.   DECK        LOC=0,LENGTH=14
              000016000000
              332262263360        .BSF.   REAL        LOC=0,LENGTH=0
              0C0000000000
              333326314662        ..FIOS  VIRTUAL     SECT. 3,CALL
              200000100000

                $DKEND .BSF.                                                    001938
```

```
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          .BSF.     00000
          ..BSF.    00011    5
          LK.DR     00014    1,6
          PON       00010    0
  LCTR     BLCTR
  QUAL     UNQS
  LCTR    //
          SEL       00005    5
  SET     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:

```
                $IBLDR BCDUMP

                $TEXT  BCDUMP
                                        ENTRY   BCDUMP

BINARY CARD (NOT PUNCHED)
      00000  1 00000 0 00005  10001  BCDUMP SAVE    1,2,4
      00001  0774 00 2 00000  10000
      00002  0774 00 1 00000  10000
      00003  0774 00 4 00000  10000
      00004  0020 00 4 00001  10000
      00005  0634 00 4 05000  10011
      00006  0634 00 4 00143  10001
      00007  0634 00 4 00003  10001
      00010  0634 00 1 00002  10001
      00011  0634 00 2 00001  10001
      00012  0500 00 4 00001  10000          CLA     1,4             IS THERE A
      00013  4734 00 2 00000  10000          PDX     0,2               THIRD
      00014  7 00002 2 01002  10011          TXL     *+2,2,2             ARGUMENT
      00015  4520 60 4 00005  10000          NZT*    5,4             YES. IS IT = 0
      00016  0634 00 0 00056  10001          SXA     CNUM,0               YES
      00017  0443 00 4 00003  10000          DLD     3,4                                          010466 TS
      00020  0040 00 0 01002  10011          TLQ     *+2
      00021  0131 00 0 00000  10000          XCA
      00022  4600 00 0 06000  10011          STQ     ..BRDB          ..BRDB HAS THE FIRST ADDRESS 010466 TS

BINARY CARD (NOT PUNCHED)
      00023  0534 00 1 06000  10011          LXA     ..BRDB,1        PUT FIRST LOC. IN IX1        010466 TS
      00024  0402 00 0 06000  10011          SUB     ..BRDB
      00025  0734 00 2 00000  10000          PAX     0,2             THE NO. OF WORDS OUTPUTED IN INDEX 2
      00026  1 00001 2 01001  10011          TXI     *+1,2,1         TRUE WORD COUNT
      00027  0634 00 1 00031  10001          SXA     IX1,1
      00030  0634 00 2 00032  10001          SXA     IX2,2
      00031  0774 00 1 00000  10000  IX1     AXT     **,1
      00032  0774 00 2 00000  10000  IX2     AXT     **,2
      00033  0074 00 4 07000  10011          TSX     ..FTCK,4                                     010666 TS
      00034  6 00026 2 00130  10001  TEST    TNX     LASTC,2,22                                 011066 TS
      00035  0634 00 2 00032  10001          SXA     IX2,2
      00036  0774 00 2 00026  10000          AXT     22,2
      00037  1 00500 2 01001  10011  TEST4   TXI     *+1,2,320
      00040  4634 00 2 06000  10011          SXD     ..BRDB,2                                     010566 TS
      00041  2 00500 2 01001  10011          TIX     *+1,2,320
      00042  0634 00 1 00052  10001          SXA     CLA,1
      00043  0634 00 1 06000  10011          SXA     ..BRDB,1                                     010466 TS
      00044  1 00026 1 01001  10011  LOOP    TXI     *+1,1,22
      00045  0634 00 1 00031  10001          SXA     IX1,1

BINARY CARD (NOT PUNCHED)
      00046  0774 00 4 00033  10000          AXT     27,4                                         010466 TS
      00047  0600 00 4 06034  10011  CLEAR   STZ     ..BRDB+28,4     CLEAR THE BUFFER.            010466 TS
      00050  2 00001 4 41001  10011          TIX     *-1,4,1
      00051  0774 00 4 00000  10000          AXT     0,4
      00052  0500 00 4 00000  10000  CLA     CLA     **,4            FILL THE BUFFER WITH
      00053  0601 00 4 06002  10011          STO     ..BRDB+2,4                                   010466 TS
      00054  1 77777 4 01001  10011          TXI     *+1,4,-1
      00055  2 00001 2 41003  10011          TIX     *-3,2,1
      00056  0774 00 1 00000  10000  CNUM    AXT     **,1            CONSECUTIVELY
      00057  0500 00 0 00137  10001          CLA     HUNBIT          NUMBER
      00060  0771 00 0 00001  10000          ARS     1               THE
      00061  7 00143 1 01002  10011          TXL     *+2,1,99        BCDUMP
      00062  1 77634 1 41002  10011          TXI     *-2,1,-100      CARDS
      00063  0621 00 0 00134  10001          STA     GP+1            FROM
      00064  0500 00 0 00141  10001          CLA     BITT            ZERO
      00065  0771 00 0 00001  10000          ARS     1               TO
      00066  7 00011 1 01002  10011          TXL     *+2,1,9         999
      00067  1 77766 1 41002  10011          TXI     *-2,1,-10
      00070  0601 00 0 00135  10001          STO     WORD3
```

```
BINARY CARD (NOT PUNCHED)
      00071  0500 00 0 00140  10001          CLA     BITU
      00072  0771 00 0 00001  10000          ARS     1
      00073  7 00000 1 01002  10011          TXL     *+2,1,0
      00074  1 77777 1 41002  10011          TXI     *-2,1,-1
      00075  4602 00 0 00135  10001          ORS     WORD3
      00076  0534 00 1 00056  10001          LXA     CNUM,1
      00077  1 00001 1 01001  10011          TXI     *+1,1,1
      00100  7 01747 1 01002  10011          TXL     *+2,1,999
      00101  0774 00 1 00000  10000          AXT     0,1
      00102  0634 00 1 00056  10001          SXA     CNUM,1
      00103  0443 00 0 00133  10001          DLD     GP                                    010466 TS
      00104  4603 00 0 06030  10011          DST     ..BRDB+24                             010466 TS
      00105  0443 00 0 00135  10001          DLD     GP+2                                  010466 TS
      00106  4603 00 0 06032  10011          DST     ..BRDB+26                             010466 TS
      00107  0774 00 1 00026  10000          AXT     22,1            COMPUTE
      00110  4500 00 0 06000  10011          CAL     ..BRDB          THE                   010466 TS
      00111  0361 00 0 06030  10011          ACL     ..BRDB+24       CHECK SUM             010466 TS
      00112  2 00001 1 41001  10011          TIX     *-1,1,1                               010466 TS
      00113  0602 00 0 06001  10011          SLW     ..BRDB+1                              010466 TS

BINARY CARD (NOT PUNCHED)
      00114  4774 00 1 00137  10001          AXC     OUT-3,1                               010566 TS
      00115  0634 00 1 05000  10011          SXA     SYSLOC,1                              010566 TS
      00116  000000000000     00010          CALL    ..BCWD                                010466 TS
      00116  0074 00 4 04000  10011
      00117  1 00000 0 01002  10011
      00120  0 00143 0 00110  10100
      00121  0074 00 4 10000  10011          TSX     ..FIOC,4                              010666 TS
      00122  0020 00 0 00031  10001          TRA     IX1
      00123  0020 00 0 01001  10011  RETURN  TRA     *+1
      00124  0774 00 1 00031  10001          AXT     IX1,1
      00125  0634 00 1 00122  10001          SXA     RETURN-1,1
      00126  0074 00 4 07000  10011          TSX     ..FTCK,4                    5/4/66 TS861
                    00127                   RETURN  BCDUMP
      00130  0500 00 0 00123  10001  LASTC   CLA     RETURN
      00131  0601 00 0 00122  10001          STO     RETURN-1
      00132  0020 00 0 00037  10001          TRA     TEST4
      00133  420041004040     10000  GP      OCT     420041004040
      00134  104020400000     10000          OCT     104020400000
      00135  0 00000 0 00000  10000  WORD3   PZE

BINARY CARD (NOT PUNCHED)
      00136  0 00000 0 00000  10000          PZE
      00137  000000002000     10000  HUNBIT  OCT     2000
      00140  000020000000     10000  BITU    OCT     20000000
      00141  200000000000     10000  BITT    OCT     200000000000
      00142  0 00000 0 11000  10011  OUT     PZE     .PCH.
      00143  000000000000     10000          *LDIR
      00144  222324644447     10000
                    00000  01111          END
```

```
          $CDICT BCDUMP

BINARY CARD (NOT PUNCHED)
          000145000000              PREFACE     START=0,LENGTH=101,TYPE=7094,CMPLX=5
          000004000005
          222324644447      BCDUMP  DECK        LOC=0,LENGTH=101
          000145000000
          222324644447      BCDUMP  REAL        LOC=0,LENGTH=0
          000000000000
          222324644447      BCDUMP  REAL        LOC=0,LENGTH=0
          000000000000
          333322236624      ..BCWD  VIRTUAL     SECT. 4,CALL
          200000100000
          627062434623      SYSLOC  VIRTUAL     SECT. 5
          200000000000
          333322512422      ..BRDB  VIRTUAL     SECT. 6
          200000000000
          333326632342      ..FTCK  VIRTUAL     SECT. 7
          200000000000
          333326314623      ..FIOC  VIRTUAL     SECT. 8
          200000000000
          334723303360      .PCH.   VIRTUAL     SECT. 9
          200000000000

          $DKEND BCDUMP                                       001938
```

```
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          BCDUMP    00000    127
          BITT      00141    64
          BITU      00140    71
          CLA       00052    42
          CLEAR     00047
          CNUM      00056    16,76,102
          GP        00133    63,103,105
          HUNBIT    00137    57
          IX1       00031    27,45,122,124
          IX2       00032    30,35
          ..0001    00003    7,10,11
          ..0002    00004
          ..0003    00005    0
          LASTC     00130    34
          LOOP      00044
          OUT       00142    114
          RETURN    00123    125,130,131
  LCTR     BLCTR
  QUAL     UNQS
  LCTR    //
          TEST4     00037    132
          TEST      00034
          WORD3     00135    70,75

REFERENCES TO VIRTUAL SYMBOLS.

          ..BCWD       4    116
          ..BRDB       6    22,23,24,40,43,47,53,104,106,110,111,113
          ..FIOC       8    121
          ..FTCK       7    33,126
          .PCH.        9    142
          SYSLOC       5    5,115
```

```
                    $IBLOR .BCRWD                                                    .BCR0000

                    $TEXT  .BCRWD                                                    .BCR0001

                                                ENTRY   ..BRDB                            11/1/65   JMLR
                                                ENTRY   ..BCRD                            11/1/65   JMLR
                                                ENTRY   ..BCWD                            11/1/65   JMLR
                           00034          SIZE  SET     28          RECORD SIZE          11/1/65   JMLR

BINARY CARD (NOT PUNCHED)
00000  0500 00 0 00012  10001  ..BCRD  CLA     PON         READ ENTRY FOR BCREAD          11/1/65   JMLR
00001  0020 00 0 01002  10011          TRA     *+2         GET CORRECT ARG FOR .FIOS      11/1/65   JMLR
00002  4500 00 0 00013  10001  ..BCWD  CAL     PTH         WRITE ENTRY FOR BCDUMP         11/1/65   JMLR
00003  0634 00 4 00050  10001          SXA     LK.DR,4     SAVE IR4                       11/1/65   JMLR
00004  0C00000C0000     00010          CALL    ..FIOS(SEL) SET UP READ OR WRITE           11/1/65   JMLR
00004  0674 00 4 05000  10011
00005  1 00001 0 01003  10011
00006  0 00050 0 00011  10100
00007  C 00C00 0 00007  10001
00007  400007041001      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 4 00050  10001          LXA     LK.DR,4     RESTORE 4                      11/1/65   JMLR
00011  0020 00 4 00001  10000          TRA     1,4                                        11/1/65   JMLR
00012  1 00000 0 00000  10000  PON     PON     0,,0                                       11/1/65   JMLR
00013  3 00000 0 00000  10000  PTH     PTH     0,,0                                       11/1/65   JMLR
00014  2C0000000034      00001  ..BRDB  BSS     SIZE        I/O BUFFER                     11/1/65   JMLR
00050  0C0C000C0000      10000  LK.DR   LDIR                                               11/1/65   JMLR
00051  332223516624      10000
                    00000  01111          END                                            11/1/65   JMLR
```

```
                $CDICT .BCRWD                                                   .BCR0002

BINARY CARD (NOT PUNCHED)
            0000520C0000              PREFACE     START=0,LENGTH=42,TYPE=7094,CMPLX=5
            000004000005
            322223516624      .BCRWD  DECK        LOC=0,LENGTH=42
            0000520C0000
            333322235124      ..BCRD  REAL        LOC=0,LENGTH=0
            0C0000C0000
            333322236624      ..BCWD  REAL        LOC=2,LENGTH=0
            0C0000000002
            333322512422      ..BRDB  REAL        LOC=14,LENGTH=0
            0C0C00000014
            333326314662      ..FIOS  VIRTUAL     SECT. 5,CALL
            2C0000100000

                $DKEND .BCRWD                                   001938      .BCR0003
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          ..BCRD    00000
          ..BCWD    00002
          ..BRDB    00014    7
          LK.DR     00050    3,10
          PDN       00012    0
          PTH       00013    2
  LCTR     BLCTR
  QUAL     UNQS
  LCTR    //
          SFL       00007    7
  SET     SIZE      00034    7,14

REFERENCES TO VIRTUAL SYMBOLS.

          ..FIOS        5    4

                $IBLDR .PCH.                                                    .PCH0000

                $FILE  .PCH.    'PCH    ',PP,READY,OUTPUT,BLK=28,MULTIREEL,BIN,NOLIST   .PCH0001

                $FDICT .PCH.                                                    .PCH0002


BINARY CARD (NOT PUNCHED)
            205C02000034  PCH     FILE    'PCH                          BIN,OUTPUT,VOHCVN,BLK=28
            0C0C800C0000
            4723306C6060
            6C6060C6060
            606C606C6060
                $TEXT  .PCH.                                                    .PCH0003

                                          ENTRY    .PCH.
BINARY CARD (NOT PUNCHED)
      00000  0 00000 0 04001  10010  .PCH.  PZE      PCH
                                    PCH    FILE     ,PP,READY,OUTPUT,BLK=28,MULTIREEL,BIN,NOLIST
                       00000  01111         END

                $CDICT .PCH.                                                    .PCH0004


BINARY CARD (NOT PUNCHED)
            000001C0000               PREFACE     START=0,LENGTH=1,TYPE=7094,CMPLX=5
            000004000005
            334723303360      .PCH.   DECK        LOC=0,LENGTH=1
            CC0001000000
            334723303360      .PCH.   REAL        LOC=0,LENGTH=0
            0C0000000000

                $DKEND .PCH.                                    001938      .PCH0005
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          .PCH.     00000
  FILE    PCH           1    0
  LCTR     BLCTR
  QUAL     UNQS
  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:

```
                $IBLDR .BCREA                                                   .BCR0000
                $TEXT  .BCREA                                                   .BCR0001
                                          ENTRY   BCREAD                        11/1/65   JMLR

BINARY CARD (NOT PUNCHED)
      00000  1 00000 0 00004  10001  BCREAD SAVE    1,4                        11/1/65   JMLR
      00001  0774 00 1 00000  10000
      00002  0774 00 4 0000C  10000
      00003  0020 00 4 00001  10000
      00004  0634 00 4 05000  10011
      00005  0634 00 4 00056  10001
      00006  0634 00 4 00002  10001
      00007  0634 00 1 00001  10001
      00010  0500 00 4 00003  10000          CLA     3,4       GET FIRST ARG.   11/1/65   JMLR
      00011  0560 00 4 00004  10000          LDQ     4,4       GET SECOND ARG.  11/1/65   JMLR
      00012  0C40 00 0 01002  10011          TLQ     *+2       COMPARE          11/1/65   JMLR
      00013  0131 00 0 00000  10000          XCA               IF 2ND LESS  EXCHANGE 11/1/65 JMLR
      00014  4600 00 0 00055  10001          STQ     TEMP      STORE SMALLEST ARG 11/1/65 JMLR
      00015  0400 00 0 06000  10011          ADD     SYSONE    ADD 1            11/1/65   JMLR
      00016  0621 00 0 00034  10001          STA     STO       STORE FOR MOVE   11/1/65   JMLR
      00017  04C2 00 0 00055  10001          SUB     TEMP      COMPUTE COUNT    11/1/65   JMLR
      00020  0621 00 0 00030  10001          STA     IX1       STORE FOR MOVE   11/1/65   JMLR
      00021  4774 00 4 00051  10001          AXC     UNC-3,4   LOCATE UNO5 LIKE FIV CALL 11/1/65 JMLR
      00022  0634 00 4 05000  10011          SXA     SYSLOC,4  AND SAVE IN SYSLOC 11/1/65 JMLR

BINARY CARD (NOT PUNCHED)
      00023  0C0000000000     00010          CALL    ..BCRD    SET UP READ      11/1/65   JMLR
      00023  0074 00 4 04000  10011
      00024  1 00000 0 01002  10011
      00025  0 00056 0 00016  10100
      00026  0C74 00 4 07000  10011  READ    TSX     ..FIOC,4  READ RECORD      11/1/65   JMLR
      00027  0C74 00 4 10000  10011          TSX     ..FTCK,4  CHECK READ       11/1/65   JMLR
      00030  0774 00 1 0000C  10000  IX1     AXT     **,1      PICK UP COUNT LEFT 11/1/65   JMLR
      00031  7 00026 1 00042  10001          TXL     LASTC,1,22  IS ONLY 1 REC LEFT 11/1/65 JMLR
      00032  0774 00 4 0000C  10000  IX4     AXT     0,4       REC CNT          11/1/65   JMLR
      00033  0500 00 4 11002  10011          CLA     ..BRDB+2,4  MOVE WORDS       11/1/65   JMLR
      00034  0601 00 1 00000  10000  STO     STO     **,1      TO STORE         11/1/65   JMLR
      00035  2 00001 1 01001  10011          TIX     *+1,1,1   DECR. COUNT      11/1/65   JMLR
      00036  1 77777 4 01001  10011          TXI     *+1,4,-1  DECR. REC COUNT  11/1/65   JMLR
      00037  3 77752 4 00033  10001  CKIR4   TXH     STO-1,4,-22  CR. REC COUNT 11/1/65   JMLR
      0C040  0634 00 1 00030  10001          SXA     IX1,1     NO SAVE REMAINING COUNT 11/1/65 JMLR
      00041  0020 00 0 00026  10001          TRA     READ      GO READ NEXT RECORD 11/1/65  JMLR
      00042  7 00000 1 00047  10001  LASTC   TXL     DONE,1,0  ANY MORE WORDS    04/26/66  TS
      00043  0774 00 4 00047  10001          AXT     DONE,4    YES STORE TO EXIT NEXT 11/1/65 JMLR
      00044  0634 00 4 00041  10001          SXA     LASTC-1,4 TIME             11/1/65   JMLR

BINARY CARD (NOT PUNCHED)
      00045  4636 00 1 00037  10001          SCD     CKIR4,1   SET REC CNT = NO WORDS LEFT 11/1/65 JMLR
      00046  0020 00 0 00032  10001          TRA     IX4       GO PROCESS RECORD  11/1/65   JMLR
      00047  0774 00 4 00026  10001  DONE    AXT     READ,4    RESTORE EXIT       11/1/65   JMLR
      00050  0634 00 4 00041  10001          SXA     LASTC-1,4                    11/1/65   JMLR
      00051  0774 00 4 77752  10000          AXT     -22,4     RESTORE REC CNT    11/1/65   JMLR
      00052  4634 00 4 00037  10001          SXD     CKIR4,4                      11/1/65   JMLR
                       00053                 RETURN  BCREAD                       11/1/65   JMLR
      00054  0 00000 0 12000  10011  UNC     PZE     .UNO5.    ADD OF UNIT 5      11/1/65   JMLR
      00055  0 00000 0 00000  10000  TEMP    PZE                                  11/1/65   JMLR
      00056  000000000000     10000          *LDIR
      00057  332223512521     10000
                       00000  01111          END
```

```
BINARY CARD (NOT PUNCHED)
              000060000000                PREFACE     START=0,LENGTH=48,TYPE=7094,CMPLX=5
              000004000005
              332223512521        .BCREA DECK         LOC=0,LENGTH=48
              000060000000
              222351252124        BCREAD REAL         LOC=0,LENGTH=0
              000000000000
              222351252124        BCREAD REAL         LOC=0,LENGTH=0
              000000000000
              333322235124        ..BCRD VIRTUAL      SECT. 4,CALL
              200000100000
              627062434623        SYSLOC VIRTUAL      SECT. 5
              200000000000
              627062464525        SYSONE VIRTUAL      SECT. 6
              200000000000
              333326314623        ..FIOC VIRTUAL      SECT. 7
              200000000000
              333326632342        ..FTCK VIRTUAL      SECT. 8
              200000000000
              333322512422        ..BRDB VIRTUAL      SECT. 9
              200000000000
              336445000533        .UNO5. VIRTUAL      SECT. 10
              200000000000

                   $DKEND .BCREA
```

```
REFERENCES TO DEFINED SYMBOLS.

  CLASS   SYMBOL    VALUE    REFERENCES

          BCREAD    00000    53
          CKIR4     00037    45,52
          DONE      00047    42,43
          IX1       00030    20,40
          IX4       00032    46
          ..0001    00002    6,7
          ..0002    00003
          ..0003    00004    0
          LASTC     00042    31,44,50
          READ      00026    41,47
  LCTR     BLCTR
  QUAL     UNOS
  LCTR    //
          STO       00034    16,37
          TEMP      00055    14,17
          UNC       00054    21

REFERENCES TO VIRTUAL SYMBOLS.

          ..BCRD       4     23
          ..BRDB       9     33
          ..FIOC       7     26
          ..FTCK       8     27
          .UNO5.      10     54
          SYSLOC       5     4,22
          SYSONE       6     15
```

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 RELMOD ASSEMBLY.

               $IBLDR .SHIFT                                                    .SHI0000
               $TEXT  .SHIFT                                                    .SHI0001
                                                  ENTRY   IALS
                                                  ENTRY   IARS

BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000  IALS   CLA*    3,4
      00001  0621 00 0 01002  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  0131 00 0 00000  10000         XCA
      00006  0020 00 4 00001  10000         TRA     1,4
      00007  0500 60 4 00003  10000  IARS   CLA*    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  0020 00 4 00001  10000         TRA     1,4
                    00000  01111         END



               $CDICT .SHIFT                                                    .SHI0002

BINARY CARD (NOT PUNCHED)
             0000160C0000              PREFACE    START=0,LENGTH=14,TYPE=7094,CMPLX=5
             000004000005
             336230312663       .SHIFT DECK       LOC=0,LENGTH=14
             000016000000
             312143626060       IALS   REAL       LOC=0,LENGTH=0
             000000000000
             312151626060       IARS   REAL       LOC=7,LENGTH=0
             000000000007

               $DKEND .SHIFT                                    001938          .SHI0003


REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         IALS     00000
         IARS     00007
  LCTR    BLCTR
  QUAL    UNQS
  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:

		Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4	
Card columns	1 to 6	7 to 12	13 to 24	25 to 30	31 to 42	43 to 48	49 to 60	61 to 66	67 to 78	79 to 80
Format	A6	A6	F12.0	A6	F12.0	A6	F12.0	A6	F12.0	I2

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

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

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

(6) FINISH card

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:

Label	Description	Value (ref. 3)
HCK	hc/k, second radiation constant	1.4388
R	Universal gas constant	1.98726
SCONST	Entropy constant S_c (see eqs. (4) and (5))	3.66511

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:

Card section	Contents
Columns 1 to 6	ATOM
Label 1	Left-adjusted atomic symbol, for example, H, HE, LI
Numerical value 1	Atomic weight
Label 2	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)
Numerical value 2	Coefficient, b, in equation (8)
Numerical value 3	Sum of the statistical weights $\sum g_i$ (eq. (8)) for the ground electronic state

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 ΔH_T^o 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 F_2(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 ΔH 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:

Species	Columns 1 to 12
F^-	F1-(G)
N^+	N1+(G)
O^{++}	O1++(G)
O_2^-	O2-(G)

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.

REFNCE 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 ΔH_T^o 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_0^o and the $(H_T^o - H_0^o)/RT$ and $-(F_T^o - H_0^o)/RT$ 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:

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4
TEMP	T	100.	I	100.	T	600.	T	688.2
TEMP	T	700.	I	50.	T	900.	T	962.3
TEMP	T	1000.						

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), H_2O(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 H_2O(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(ℓ) 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 $H^{o}_{298.15}$ rather than H^{o}_{0}, the $H^{o}_{298.15} - H^{o}_{0}$ value must be included on the METHOD card (label H298H0) if $H^{o}_{T} - H^{o}_{0}$ 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^{o}_{p} or C^{o}_{p}/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 ϵ_m/hc (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 (ω_e or ν_i)
(2) Either the rotational constant(s) (B_0 for linear; A_0, B_0, and C_0 for nonlinear molecules) or the moment(s) of inertia (I_B for linear; I_A, I_B, and I_C 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 ΔH_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_i 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, $q_1 = 0$, $q_2 = 1$, . . . , $q_5 = 4$.

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 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 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:

T, C_p°/R, $(H_T^{\circ} - H_0^{\circ})RT$, $(H_T^{\circ} - H_{298.15}^{\circ})/RT$ (if $T = 298.15$ is in T range), S_T°/R, $-(F_T^{\circ} - H_0^{\circ})/RT$, $-(F_T^{\circ} - H_{298.15}^{\circ})/RT$ (if $T = 298.15$ is in T range), H_T°/RT (if an H_0° value is available), and $-F_T^{\circ}/RT$ (if an H_0° value is available)

(2) Table of dimensioned properties as follows:

T, C_p^o, $H_T^o - H_0^o$, $H_T^o - H_{298.15}^o$ (if T = 298.15 is in T range), S_T^o, $-(F_T^o - H_0^o)$, $-(F_T^o - H_{298.15}^o)$ (if T = 298.15 is in T range), H_T^o (if an H_0^o value is available) and $-F_T^o$ (if an H_0^o 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^o/R, $(H_T^o - H_0^o)/RT$, S^o/R, $-(F_T^o - H_0^o)/RT$, H_T^o/RT, F_T^o/RT, and $\Delta H_T^o/RT$ and $-\Delta F_T^o/RT$ for reactions from the assigned reference elements, and $\Delta H_T^o/RT$ and $-\Delta F_T^o/RT$ for reactions from the monatomic gases

(2) Table of dimensional properties as follows:
T, C_p^o, $H_T^o - H_0^o$, S_T^o, $-(F_T^o - H_0^o)$, H_T^o, and ΔH_T^o and $\log_{10}K$ for formation from the assigned reference elements, and ΔH_T^o 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^o/R, $(H_T^o - H_0^o)/RT$, and S_T^o/R, results when LSTSQS 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^o/R, $(H_T^o - H_0^o)/RT$, and $-(F_T^o - H_0^o)/RT$
   (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^o/R equation (see eq. (10)) for coefficients a_i/R

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

$$(H - H0)/R \text{ CONSTANT} = (a_{r+1} - H_0^o)/R$$

$$H/R(A6) \text{ CONSTANT} = a_{r+1}/R$$

$$S/R \text{ CONSTANT} = a_{r+2}/R$$

Finally, the contents of the punched binary coefficient cards are listed. See the section 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 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 = (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:

$$Q^m \leq 1\times10^{-10}$$

and

$$(\epsilon_m/kT)^2 Q^m \leq 1\times10^{-10}$$

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^o/R, $(H_T^o - H_0^o)/RT$, and $-(F_T^o - H_0^o)/RT$

(3) Values of ϵ/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, ρ

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

(3) θ_1, θ_2, θ_3

(4) y_{ijk}

(5) x_{ij}

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

(7) ν_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 $-(F_T^o - H_0^o)/RT$, $(H_T^o - H_0^o)/RT$, and C_p^o/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:

Code on listing	Formula numbers in tables I and II
ELECTR	1
H.O.	2
R.R.	3 or 4
RHO	5
THETA	6
FERMI	7
ALPHA	8 to 11
XLJ	12 or 14
YLJK	13
G+AG	16
WEZE	15
AXIJ	17
(XLJ)2	18 and 19
XY	20 and 21
G2, GX	22 and 23
AX2	24 to 27

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° to 500° K at 100° intervals assuming a rigid-rotator harmonic-oscillator approximation and using the following data:

(1) Physical constants: hc/k = 1.4388 (centimeters)(degrees), R = 1.98726 calories per mole per $^{\circ}$K, and S_c = -3.66511 calories per mole per $^{\circ}$K

(2) Atomic weights: F = 19.00 grams per mole and Mg = 24.32 grams per mole

(3) Molecular constants: statistical weight = 1, ν_1 = 540 centimeter^{-1}, ν_2 = 500(2) centimeter^{-1}, ν_3 = 820 centimeter^{-1}, I_B = 19.77×10^{-39} (grams)(centimeters^2), and symmetry number = 2

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

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4
CONSTS			HCK	1.4388	R	1.98726	SCONST	-3.66511
ATOM	F	19.0000	F2(G)					
ATOM	MG	24.3200	MG1(S)					
MG1F2(G)								
TEMP	T	100.	I	100.	T	500.		
METHOD	RRHO							
DATA	STATWT	1.	V1	540.	V2(2)	500.	V3	820.
DATA	IB	19.77	SYMNO	2.				
FINISH								

Listed output. - The listed output is as follows:

```
CONSTS                                  HCK        1.4388000      R          1.9872600      SCONST     -3.6651100
ATOM        F              19.          F2(G)
ATOM        MG         24.3199999       MG1(S)
MG1F2(G)                                                                                                MG1F2(G)
TEMP        T             100.          I            100.        T           500.
METHOD      RRHO
MOLECULAR WT.= 62.32000
DATA        STATWT          1.          V1           540.        V2(2)       500.        V3          820.
DATA        IB         19.7700000       SYMNO          2.
FINISH
MG1F2(G)                                                                                                MG1F2(G)
EITHER ASINDH,DELTAH,HF298,IPATOM,OR DISSOC WAS NOT FOUND ON THE FORMULA CARD, C520
MG1F2(G)
NO HZERO VALUE IS AVAILABLE
```

T	CP/R	(H-H0)/RT	(H-H298)/RT	S/R	-(F-H0)/RT	-(F-H298)/RT
100	3.6044264	3.5141867	-9.1857179	23.0655146	19.5513279	32.2512326
200	4.6689711	3.8004504	-2.5495019	25.8549500	22.0544996	28.4044518
298.15	5.6537278	4.2595689	0.0000001	27.9161122	23.6565433	27.9161119
300	5.6683680	4.2682114	0.0349098	27.9511299	23.6829185	27.9162199
400	6.2824038	4.7016582	1.5266821	29.6735663	24.9719081	28.1468842
500	6.6502306	5.0576122	2.5176313	31.1183319	26.0607197	28.6007006

```
MG1F2(G)                                                                                                MG1F2(G)
```

T	CP	H-H0	H-H298	S	-(F-H0)	-(F-H298)
100	7.1629324	698.3603	-1825.4410	45.837174	3885.3572	6409.1584
200	9.2784594	1510.4966	-1013.3046	51.380507	8765.6049	11289.4061
298.15	11.2354270	2523.8013	0.0000	55.476573	14016.5389	16540.3401
300	11.2645209	2544.6137	20.8125	55.546162	14119.2349	16643.0359
400	12.4847697	3737.3669	1213.5657	58.969091	19850.2695	22374.0708
500	13.2157372	5025.3952	2501.5940	61.840216	25894.7126	28418.5139

```
MG1F2(G)                                                                                                MG1F2(G)
```

Example 2 (F_2(g) with PANDK Method and EFTAPE Option)

Problem. - Calculate the thermodynamic functions for the reference element, F_2(g), where the standard temperature schedule is assumed. Prepare a set of EFDATA and binary EF data cards for future ΔH^o_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: $\Delta H^o_{f(298.15^o\,K)}$ (assigned enthalpy at 298.15° K) = 0

(3) Molecular constants: statistical weight = 1, ω_e = 923 centimeter^{-1}, $\omega_e x_e$ = 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:

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4
CONSTS			HCK	1.4388	R	1.98726	SCONST	-3.66511
ATOM	F	19.	F2(G)					
F2(G)			HF298	0.				
EFTAPE								
METHOD	PANDK							
DATA	STATWT	1.	WE	923.	WEXE	15.6	BE	.8909
DATA	ALPHAE	.0162	SYMNO	2.				
FINISH								

Listed output. - The listed output is as follows:

```
CONSTS                                HCK       1.4388000    R         1.9872600    SCONST    -3.6651100
ATOM          F              19.      F2(G)
F2(G)                                 HF298
EFTAPE
METHOD        PANDK
MOLECULAR WT.= 38.00000
DATA          STATWT          1.      WE          923.       WEXE      15.6000000   BE        0.89090000
DATA          ALPHAE   0.01620000     SYMNO         2.
FINISH
F2(G)                                                                                          F2(G)
```

HZERO = -2109.698

T	CP/R	(H-H0)/RT	(H-H298)/RT	S/R	-(F-H0)/RT	-(F-H298)/RT	H/RT	-F/RT
100	3.5016367	3.4963896	-7.1197225	20.4782846	16.9818952	27.5980072	-7.1197224	27.5980072
200	3.5712715	3.5097823	-1.7982737	22.9178321	19.4080498	24.7161057	-1.7982737	24.7161057
298.15	3.7674544	3.5606615	0.0000000	24.3783007	20.8176394	24.3783007	0.0000001	24.3783007
300	3.7714767	3.5619491	0.0232451	24.4016178	20.8396688	24.3783727	0.0232451	24.3783727
400	3.9723299	3.6402861	0.9862581	25.5150812	21.8747952	24.5288231	0.9862581	24.5288231
500	4.1255408	3.7228256	1.5996032	26.4188421	22.6960166	24.8192389	1.5996032	24.8192389
600	4.2362054	3.7996674	2.0303153	27.1813467	23.3816793	25.1510313	2.0303153	25.1510313
700	4.3168867	3.8680834	2.3514960	27.8407331	23.9726498	25.4892371	2.3514960	25.4892371
800	4.3774085	3.9281369	2.6011229	28.4213059	24.4931691	25.8201830	2.6011229	25.8201830
900	4.4243385	3.9807662	2.8011982	28.9397089	24.9589429	26.1385107	2.8011982	26.1385107
1000	4.4619371	4.0270678	2.9654565	29.4078734	25.3808057	26.4424169	2.9654566	26.4424169
1100	4.4929844	4.0680542	3.1029531	29.8346412	25.7665870	26.7316880	3.1029531	26.7316880
1200	4.5193276	4.1045904	3.2199144	30.2267406	26.1221502	27.0068262	3.2199144	27.0068262
1300	4.5422208	4.1373928	3.3207687	30.5894034	26.4520106	27.2686346	3.3207687	27.2686346
1400	4.5625337	4.1670478	3.4087541	30.9267755	26.7597277	27.5180213	3.4087541	27.5180213
1500	4.5808824	4.1940346	3.4862938	31.2421932	27.0481586	27.7558994	3.4862938	27.7558994
1600	4.5977121	4.2187455	3.5552385	31.5383811	27.3196356	27.9831426	3.5552385	27.9831426
1700	4.6133497	4.2415028	3.6170256	31.8175902	27.5760875	28.2005646	3.6170256	28.2005646
1800	4.6280400	4.2625729	3.6727889	32.0817013	27.8191288	28.4089127	3.6727889	28.4089127
1900	4.6419680	4.2821776	3.7234348	32.3323030	28.0501258	28.6088684	3.7234348	28.6088684
2000	4.6552764	4.3005022	3.7696966	32.5707455	28.2702436	28.8010492	3.7696966	28.8010492
2100	4.6680759	4.3177027	3.8121735	32.7981887	28.4804864	28.9860153	3.8121735	28.9860153
2200	4.6804542	4.3339116	3.8513610	33.0156350	28.6817236	29.1642740	3.8513610	29.1642740
2300	4.6924804	4.3492413	3.8876711	33.2239561	28.8747149	29.3362849	3.8876711	29.3362849
2400	4.7042105	4.3637882	3.9214502	33.4239154	29.0601273	29.5024652	3.9214502	29.5024652
2500	4.7156897	4.3776355	3.9529910	33.6161838	29.2385483	29.6631927	3.9529910	29.6631927
2600	4.7269547	4.3908548	3.9825428	33.8013563	29.4105020	29.8188138	3.9825428	29.8188138
2700	4.7380363	4.4035087	4.0103194	33.9799619	29.5764532	29.9696424	4.0103194	29.9696424
2800	4.7489594	4.4156516	4.0365047	34.1524711	29.7368197	30.1159666	4.0365047	30.1159666
2900	4.7597451	4.4273313	4.0612585	34.3193064	29.8919754	30.2580481	4.0612585	30.2580481
3000	4.7704111	4.4385898	4.0847194	34.4808493	30.0422597	30.3961301	4.0847194	30.3961301
3100	4.7809725	4.4494644	4.1070091	34.6374431	30.1879787	30.5304339	4.1070091	30.5304339
3200	4.7914417	4.4599878	4.1282343	34.7893977	30.3294101	30.6611636	4.1282343	30.6611636
3300	4.8018298	4.4701895	4.1484891	34.9369979	30.4668088	30.7885091	4.1484891	30.7885091
3400	4.8121458	4.4800954	4.1678568	35.0805006	30.6004052	30.9126437	4.1678568	30.9126437
3500	4.8223977	4.4897292	4.1864116	35.2201405	30.7304115	31.0337288	4.1864117	31.0337288
3600	4.8325928	4.4991117	4.2042197	35.3561349	30.8570232	31.1519151	4.2042197	31.1519151
3700	4.8427367	4.5082619	4.2213399	35.4886808	30.9804192	31.2673409	4.2213399	31.2673409
3800	4.8528348	4.5171968	4.2378255	35.6179628	31.1007662	31.3801374	4.2378255	31.3801374
3900	4.8628918	4.5259320	4.2537239	35.7441478	31.2182159	31.4904239	4.2537239	31.4904239
4000	4.8729115	4.5344812	4.2690784	35.8673916	31.3329105	31.5983133	4.2690784	31.5983133
4100	4.8828976	4.5428575	4.2839279	35.9878392	31.4449821	31.7039115	4.2839279	31.7039115
4200	4.8928531	4.5510722	4.2983077	36.1056252	31.5545530	31.8073175	4.2983077	31.8073175
4300	4.9027809	4.5591362	4.3122498	36.2208724	31.6617365	31.9086227	4.3122498	31.9086227
4400	4.9126835	4.5670589	4.3257836	36.3336983	31.7666397	32.0079145	4.3257836	32.0079145
4500	4.9225630	4.5748492	4.3389356	36.4442115	31.8693624	32.1052756	4.3389356	32.1052756
4600	4.9324214	4.5825154	4.3517303	36.5525117	31.9699965	32.2007813	4.3517303	32.2007813
4700	4.9422605	4.5900649	4.3641901	36.6586943	32.0686297	32.2945042	4.3641901	32.2945042
4800	4.9520819	4.5975047	4.3763356	36.7628493	32.1653447	32.3865137	4.3763356	32.3865137
4900	4.9618869	4.6048410	4.3881856	36.8650584	32.2602177	32.4768729	4.3881856	32.4768729
5000	4.9716768	4.6120798	4.3997575	36.9654007	32.3533211	32.5656433	4.3997575	32.5656433
5100	4.9814530	4.6192266	4.4110675	37.0639491	32.4447227	32.6528816	4.4110675	32.6528816
5200	4.9912161	4.6262864	4.4221303	37.1607738	32.5344877	32.7386436	4.4221303	32.7386436
5300	5.0009676	4.6332639	4.4329599	37.2559404	32.6226768	32.8229804	4.4329599	32.8229804
5400	5.0107090	4.6401635	4.4435688	37.3495102	32.7093468	32.9059410	4.4435688	32.9059410
5500	5.0204380	4.6469890	4.4539688	37.4415412	32.7945523	32.9875722	4.4539688	32.9875722
5600	5.0301589	4.6537446	4.4641712	37.5320892	32.8783450	33.0679183	4.4641712	33.0679183
5700	5.0398707	4.6604336	4.4741860	37.6212068	32.9607735	33.1470208	4.4741860	33.1470208
5800	5.0495740	4.6670592	4.4840228	37.7089429	33.0418839	33.2249203	4.4840228	33.2249203
5900	5.0592700	4.6736248	4.4936906	37.7953453	33.1217208	33.3016548	4.4936906	33.3016548
6000	5.0689586	4.6801329	4.5031976	37.8804584	33.2003255	33.3772607	4.5031976	33.3772607

F2(G) F2(G)

HZERO = -2109.698

T	CP	H-HO	H-H298	S	-(F-HO)	-(F-H298)	H	-F
100	6.9586626	694.8235	-1414.8740	40.695675	3374.7441	5484.4415	-1414.8739	5484.4415
200	7.0970451	1394.9700	-714.7275	45.543691	7713.7682	9823.4656	-714.7275	9823.4655
298.15	7.4369113	2109.6975	0.0000	48.446022	12334.4840	14444.1814	0.0001	14444.1814
300	7.4949048	2123.5557	13.8582	48.492359	12424.1519	14533.8492	13.8582	14533.8492
400	7.8940524	2893.6779	783.9805	50.705100	17388.3621	19498.0596	783.9805	19498.0596
500	8.1985221	3699.1112	1589.4137	52.501108	22551.4426	24661.1401	1589.4137	24661.1401
600	8.4184415	4530.5561	2420.8586	54.016403	27879.2852	29988.9827	2420.8586	29988.9827
700	8.5787762	5380.8211	3271.1236	55.326775	33347.9209	35457.6182	3271.1237	35457.6182
800	8.6990488	6244.9834	4135.2859	56.480524	38939.4360	41049.1333	4135.2859	41049.1333
900	8.7923110	7119.7356	5010.0381	57.510725	44639.9175	46749.6147	5010.0381	46749.6147
1000	8.8670241	8002.8307	5893.1331	58.441090	50438.2598	52547.9570	5893.1332	52547.9570
1100	8.9287281	8892.7095	6783.0119	59.289189	56325.3979	58435.0952	6783.0120	58435.0952
1200	8.9810790	9788.2659	7678.5684	60.068392	62293.8042	64403.5015	7678.5684	64403.5015
1300	9.0265737	10688.6975	8579.0001	60.789097	68337.1289	70446.8262	8579.0001	70446.8262
1400	9.0669407	11593.4102	9483.7126	61.459544	74449.9502	76559.6475	9483.7128	76559.6475
1500	9.1034043	12501.9558	10392.2583	62.086360	80627.5850	82737.2822	10392.2583	82737.2822
1600	9.1368493	13413.9907	11304.2932	62.674963	86865.9502	88975.6475	11304.2932	88975.6475
1700	9.1679254	14329.2469	12219.5494	63.229824	93161.4531	95271.1504	12219.5496	95271.1504
1800	9.1971188	15247.5132	13137.8157	63.754682	99510.9150	101620.6113	13137.8157	101620.6113
1900	9.2247971	16168.6702	14058.9226	64.252692	105911.4961	108021.1924	14058.9226	108021.1924
2000	9.2512444	17092.4316	14982.7343	64.726539	112360.6484	114470.3457	14982.7343	114470.3457
2100	9.2766805	18018.8352	15909.1378	65.178528	118856.0742	120965.7715	15909.1378	120965.7715
2200	9.3012793	18947.7400	16838.0425	65.610650	125395.6914	127505.3887	16838.0425	127505.3887
2300	9.3251786	19879.0679	17769.3704	66.024638	131977.5977	134087.2949	17769.3704	134087.2949
2400	9.3484893	20812.7559	18703.0583	66.422009	138600.0664	140709.7637	18703.0583	140709.7637
2500	9.3713015	21748.7495	19639.0518	66.804097	145261.4922	147371.1895	19639.0518	147371.1895
2600	9.3936880	22687.0020	20577.3044	67.172083	151960.4160	154070.1113	20577.3044	154070.1113
2700	9.4157100	23627.4751	21517.7773	67.527019	158695.4746	160805.1719	21517.7773	160805.1719
2800	9.4374169	24570.1335	22460.4360	67.869839	165465.4160	167575.1133	22460.4360	167575.1133
2900	9.4588509	25514.9492	23405.2517	68.201385	172269.0664	174378.7617	23405.2517	174378.7617
3000	9.4800470	26461.8958	24352.1980	68.522412	179105.3418	181215.0391	24352.1980	181215.0391
3100	9.5010355	27410.9519	25301.2542	68.833605	185973.2227	188082.9199	25301.2542	188082.9199
3200	9.5218403	28362.0972	26252.3994	69.135578	192871.7539	194981.4512	26252.3994	194981.4512
3300	9.5424842	29315.3142	27205.6167	69.428898	199800.0508	201909.7480	27205.6167	201909.7480
3400	9.5629847	30270.5889	28160.8911	69.714075	206757.2656	208866.9629	28160.8911	208866.9629
3500	9.5833580	31227.9070	29118.2090	69.991576	213742.6094	215852.3066	29118.2095	215852.3066
3600	9.6036183	32187.2568	30077.5593	70.261832	220755.3398	222865.0352	30077.5593	222865.0352
3700	9.6237769	33148.6274	31038.9297	70.525235	227794.7441	229904.4395	31038.9297	229904.4395
3800	9.6438445	34112.0088	32002.3115	70.782152	234860.1699	236969.8672	32002.3115	236969.8672
3900	9.6638303	35077.3936	32967.6958	71.032914	241950.9746	244060.6719	32967.6958	244060.6719
4000	9.6837419	36044.7725	33935.0742	71.277832	249066.5566	251176.2559	33935.0742	251176.2559
4100	9.7035869	37014.1396	34904.4419	71.517193	256206.3535	258316.0508	34904.4419	258316.0508
4200	9.7233711	37985.4873	35875.7900	71.751265	263369.8203	265479.5156	35875.7900	265479.5156
4300	9.7431003	38958.8115	36849.1138	71.980290	270556.4336	272666.1328	36849.1138	272666.1328
4400	9.7627795	39934.1064	37824.4087	72.204505	277765.7148	279875.4102	37824.4087	279875.4102
4500	9.7824125	40911.3662	38801.6685	72.424123	284997.1875	287106.8789	38801.6685	287106.8789
4600	9.8020037	41890.5874	39780.8896	72.639344	292250.3906	294360.0859	39780.8896	294360.0859
4700	9.8215566	42871.7656	40762.0674	72.850356	299524.9102	301634.6055	40762.0674	301634.6055
4800	9.8410742	43854.8975	41745.1997	73.057340	306820.3281	308930.0273	41745.1997	308930.0273
4900	9.8605593	44839.9795	42730.2817	73.260455	314136.2539	316245.9492	42730.2817	316245.9492
5000	9.8800143	45827.0078	43717.3101	73.459862	321472.3008	323581.9961	43717.3101	323581.9961
5100	9.8994422	46815.9814	44706.2837	73.655703	328828.1016	330937.7969	44706.2837	330937.7969
5200	9.9188441	47806.8955	45697.1978	73.848119	336203.3242	338313.0195	45697.1978	338313.0195
5300	9.9382229	48799.7490	46690.0513	74.037240	343597.6211	345707.3125	46690.0513	345707.3125
5400	9.9575796	49794.5405	47684.8428	74.223187	351010.6719	353120.3633	47684.8428	353120.3633
5500	9.9769156	50791.2646	48681.5669	74.406076	358442.1563	360551.8516	48681.5669	360551.8516
5600	9.9962335	51789.9224	49680.2246	74.586020	365891.7852	368001.4805	49680.2246	368001.4805
5700	10.0155333	52790.5112	50680.8135	74.763119	373359.2695	375468.9648	50680.8135	375468.9648
5800	10.0348164	53793.0283	51683.3306	74.937473	380844.3164	382954.0156	51683.3306	382954.0156
5900	10.0540849	54797.4741	52687.7764	75.109178	388346.6758	390456.3711	52687.7764	390456.3711
6000	10.0733386	55803.8447	53694.1465	75.278319	395866.0703	397975.7656	53694.1465	397975.7656

F2(G)

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: $\Delta H^{o}_{f(298.15)}$ = 18 858.2 calories per mole

(3) Spectroscopic data: $J_1 = 3/2$, $\epsilon_1 = 0$ and $J_2 = 1/2$, ϵ_2/hc = 404.1 centimeter^{-1}

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

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4	Card columns 79 to 80
CONSTS			HCK	1.4388	R	1.98726	SCONST	-3.66511	
ATOM	F	19.	F2(G)						
EFDATA	F2(G)	0.	HZERO	-2109.6975	MELTPT	0.	T NO.	61.0000	
Binary EF data for F_2(g)									BCDUM000 BCDUM001 BCDUM002 BCDUM003 BCDUM004 BCDUM005 BCDUM006 BCDUM007 BCDUM008
F1(G)			HF298	18858.2					
LOGK									
TEMP	T	298.15	T	1000.	T	2156.	T	3000.	
TEMP	T	5000.							
METHOD	ALLN								
DATA	1.5	0.	.5	404.1					
FINISH									

Listed output. - The listed output is as follows:

```
CONSTS                          HCK        1.4388000    R        1.9872600    SCONST    -3.6651100
ATOM        F          19.      F2(G)
EFDATA      F2(G)               HZERO     -2109.69751   MELTPT                T NO.          61.
F1(G)                           HF298      18858.200
LOGK
TEMP        T      298.15000    T            1000.      T           2156.     T             3000.
TEMP        T           5000.
METHOD      ALLN
DATA         1.5                .5        404.10000
FINISH
F1(G)                                                                                    F1(G)
HZERO =    17300.217
```

T	CP/R	(H-H0)/RT	(H-H298)/RT	S/R	-(F-H0)/RT	-(F-H298)/RT	H/RT	-F/RT
298.15	2.7357659	2.6294998	0.0000000	19.0800488	16.4505491	19.0800488	31.8281004	-12.7480516
1000	2.5577199	2.6270264	1.8430411	22.2807670	19.6537406	20.4377258	11.3325893	10.9481778
2156	2.5145423	2.5745148	2.2108852	24.2257791	21.6512644	22.0148938	6.6123454	17.6134338
3000	2.5077611	2.5565408	2.2952124	25.0552444	22.4987037	22.7600319	5.4583951	19.5968494
5000	2.5028820	2.5358168	2.3790197	26.3348284	23.7990117	23.9558086	4.2769293	22.0578990

HZERO = 17300.217

T	CP	H-H0	H-H298	S	-(F-H0)	-(F-H298)	H	-F
298.15	5.4366781	1557.9827	0.0000	37.917017	9746.9762	11304.9589	18858.1995	-7553.2406
1000	5.0828543	5220.5845	3662.6018	44.277677	39057.0923	40615.0747	22520.8013	21755.8757
2156	4.9970493	11030.5924	9472.6096	48.142921	92765.5459	94323.5283	28330.8088	75465.3301
3000	4.9835732	15241.5339	13683.5511	49.791285	134132.3203	135690.3008	32541.7505	116832.1035
5000	4.9738773	25196.6357	23638.6528	52.334151	236474.1172	238032.0996	42496.8521	219173.8984

F1(G) F1(G)

HZERO = 17300.217

T	CP/R	(H-H0)/RT	S/R	-(F-H0)/RT	H/RT	-F/RT	REFERENCE ELEMENTS DELTA H/RT	REFERENCE ELEMENTS -DELTA F/RT	GASEOUS ATOMS DELTA H/RT	GASEOUS ATOMS -DELTA F/RT
298.15	2.7358	2.6295	19.0800	16.4505	31.8281	-12.7481	31.8281	-24.9372	0	0
1000	2.5577	2.6270	22.2808	19.6537	11.3326	10.9482	9.8499	-2.2730	0	0
2156	2.5145	2.5745	24.2258	21.6513	6.6123	17.6134	4.6951	3.0701	0	0
3000	2.5078	2.5565	25.0552	22.4987	5.4584	19.5968	3.4160	4.3988	0	0
5000	2.5029	2.5358	26.3348	23.7990	4.2769	22.0579	2.0771	5.7751	0	0

T	CP	H-H0	S	-(F-H0)	H	REFERENCE ELEMENTS DELTA H	REFERENCE ELEMENTS LOG K	GASEOUS ATOMS DELTA H	GASEOUS ATOMS LOG K
0	------	0	-------	0	17300.2	18355.1	-------	0	-------
298.15	5.4367	1558.0	37.9170	9747.0	18858.2	18858.2	-10.8301	0	0
1000	5.0829	5220.6	44.2777	39057.1	22520.8	19574.2	-0.9872	0	0
2156	4.9970	11030.6	48.1429	92765.5	28330.8	20116.3	1.3333	0	0
3000	4.9836	15241.5	49.7913	134132.3	32541.8	20365.7	1.9104	0	0
5000	4.9739	25196.6	52.3342	236474.1	42496.9	20638.2	2.5081	0	0

F1(G) F1(G)

Example 4 (P(s) with Least-Squares Fit)

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:

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4
CONSTS			R	1.98726				
P1(S)			HF298	0.				
LSTSQS								
METHOD	READIN		MELTPT	317.3				
DATA	T	298.15	CP	5.694	H-H0	1282.3	S	9.981
DATA	T	300.	CP	5.705	H-H0	1292.8	S	10.016
DATA	T	317.3	CP	5.798	H-H0	1392.3	S	10.338
FINISH								

Listed output. - The listed output is as follows:

```
CONSTS                                  R           1.9872600

P1(S)                                   HF298

ATOM CARD MISSING OR FORMULA INCORRECT,. C160

LSTSQS

METHOD      READIN                      MFLTPT      317.30000

DATA        T           298.15000       CP          5.6940000       H-H0        1282.30000      S           9.9809999

DATA        T                300.       CP          5.7050000       H-H0        1292.80000      S           10.0160000

DATA        T           317.30000       CP          5.7980000       H-H0        1392.30000      S           10.3380001

FINISH

LEAST SQUARES

        T       CP/R INPUT      CP/R CALC       HH/RT INPUT     HH/RT CALC      S/R INPUT       S/R CALC        -FH/RT INPUT    -FH/RT CALC
                INPUT-CALC       FRACTION        INPUT-CALC      FRACTION        INPUT-CALC      FRACTION         INPUT-CALC      FRACTION
      300.00     2.8707869       2.8707715       2.1684799       2.1684747       5.0401055       5.0398830       2.8716256       2.8714083
                 0.0000154       0.0000053       0.0000052       0.0000024       0.0002225       0.0000441       0.0002173       0.0000757
      317.30     2.9175850       2.9175850       2.2080457       2.2080456       5.2021376       5.2021372       2.9940919       2.9940915
                 0.              0.              0.0000001       0.0000000       0.0000005       0.0000001       0.0000004       0.0000001
   MAX-REL ERR CP/R   =  0.000005     TEMP =   300.       AVER REL ERR CP/R   =  0.000003      REL LST SQ ERR CP/R   =  0.000004
   MAX REL ERR HH/RT  =  0.000002     TEMP =   300.       AVER REL ERR HH/RT  =  0.000001      REL LST SQ ERR HH/RT  =  0.000002
   MAX REL ERR S/R    =  0.000044     TEMP =   300.       AVER REL ERR S/R    =  0.000022      REL LST SQ ERR S/R    =  0.000031
   MAX REL ERR FH/RT  =  0.000076     TEMP =   300.       AVER REL ERR FH/RT  =  0.000038      RFL LST SQ ERR FH/RT  =  0.000054
       MAX ERR CP/R   =  0.000015     TEMP.=   300.           AVER ERR CP/R   =  0.000008          LST SQ ERR CP/R   =  0.000011
       MAX ERR HH/RT  =  0.000005     TEMP =   300.           AVER ERR HH/RT  =  0.000003          LST SQ ERR HH/RT  =  0.000004
       MAX ERR S/R    =  0.000223     TEMP =   300.           AVER ERR S/R    =  0.000111          LST SQ ERR S/R    =  0.000157
       MAX ERR FH/RT  =  0.000217     TEMP =   300.           AVER ERR FH/RT  =  0.000109          LST SQ ERR FH/RT  =  0.000154
CP/R =    2.7877384E 00T** 0.   -2.1253577E-03T** 1.0  -3.5459652E-07T** 2.0   5.5271664E-08T** 3.0  -9.1330942E-11T** 4.0

(H-H0)/R CONSTANT = -0.15448489E 03, H/R(A6) CONSTANT =-0.79974519E 03, S/R CONSTANT =-0.10519705E 02

PUNCHED BINARY CARDS--

P1(S)                0.          0.30000000E 03  0.31730000E 03  0.10000000E 04  0.50000000E 04  0.
  0.            0.                0.              0.              0.              0.              0.30000000E 03
  0.10000000E 04  0.27877383E 01  -0.21253577E-02  -0.35459652E-06  0.55271665E-07  -0.91330943E-10  -0.79974519E 03

P1(S)                0.          -0.10519705E 02  0.              0.              0.              0.
  0.            0.                0.              0.              0.              0.              0.
  0.            0.                0.              0.              0.              0.                      000000


P1(S)

HZERO =   -1282.300

        T       CP/R        (H-H0)/RT       (H-H298)/RT     S/R         -(F-H0)/RT      -(F-H298)/RT    H/RT            -F/RT

      298.15    2.8652516   2.1642137       0.0000000       5.0224932   2.8582796       5.0224932       0.0000000       5.0224932
      300       2.8707869   2.1684799       0.0176122       5.0401055   2.8716256       5.0224932       0.0176122       5.0224932
      317.30    2.9175850   2.2080457       0.1744488       5.2021376   2.9940919       5.0276888       0.1744488       5.0276888


HZERO =   -1282.300

        T       CP          H-H0            H-H298          S           -(F-H0)         -(F-H298)       H               -F

      298.15    5.6939999   1282.3000       0.0000          9.981000    1693.5351       2975.8351       0.0000          2975.8351
      300       5.7049999   1292.8000       10.5000         10.016000   1712.0000       2994.2999       10.5000         2994.2999
      317.30    5.7979999   1392.3000       110.0000        10.338000   1887.9474       3170.2473       110.0000        3170.2473

P1(S)                                                                                                   P1(S)
```

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:

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4
CONSTS			HCK	1.4388	R	1.98726	SCONST	-3.66511
ATOM	F	19.0000	F2(G)					
ATOM	MG	24.3200	MG1(S)					
MG1F2(G)								
TEMP	T	100.	I	100.	T	500.		
METHOD	RRHO							
DATA	STATWT	1.	V1	540.	V2(2)	500.	V3	820.
DATA	IB	19.77	SYMNO	2.				
FINISH								
F2(G)			HF298	0.				
EFTAPE								
METHOD	PANDK							
DATA	STATWT	1.	WE	923.	WEXE	15.6	BE	.8909
DATA	ALPHAE	.0162	SYMNO	2.				
FINISH								
F1(G)			HF298	18858.2				
LOGK								
TEMP	T	298.15	T	1000.	T	2156.	T	3000.
TEMP	T	5000.						
METHOD	ALLN							
DATA	1.5	0.	.5	404.1				
FINISH								
P1(S)			HF298	0.				
LSTSQS								
METHOD	READIN		MELTPT	317.3				
DATA	T	298.15	CP	5.694	H-HO	1282.3	S	9.981
DATA	T	300.	CP	5.705	H-HO	1292.8	S	10.016
DATA	T	317.3	CP	5.798	H-HO	1392.3	S	10.338
FINISH								

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): $C_p^o/R = 2.5$, $(H_T^o - H_0^o)/RT = 2.5$, and $S_T^o/R = 2.5 \ln T + 4.3661076$. Punch these coefficients for use with reference 33. Assume $H^o_{298.15} = \Delta H^o_{f(298.15)} = 0$.

(2) H_2O(g) - Calculate the thermodynamic functions for $T = 5000^o$ 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^{o} to 5000^{o} K assuming the following C_p^o equation: $a_1 + a_2T + a_3T^2 + a_4T^3$. Constrain the curve fit to fit the functions exactly at 1000^{o} K.

(e) Include DATE cards so 4/63 will be punched on coefficient cards.

(4) Mg(s, ℓ) - Perform the following options in the calculations:

(a) Calculate ΔH_T^o 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^o = 8 calories per mole per ^{o}K for the liquid.

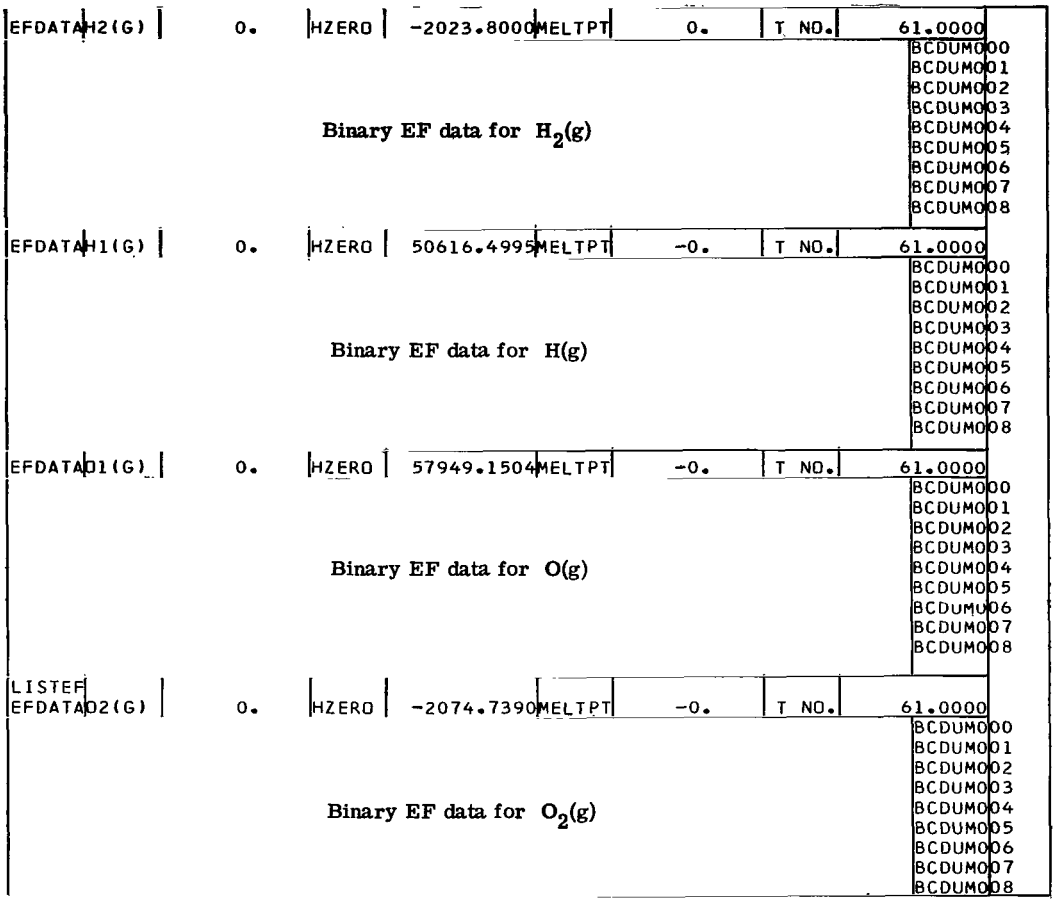
(d) Calculate the integration constants (eqs. (11) and (12)) using a heat of melting value of 2140 calories at the melting point, 923^{o} K.

(5) 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 ΔH_T^o and log K values. Use a disssociation energy of 90 kilocalories per mole at 0^{o} K.

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

Card columns 1 to 6	Label 1	Numerical value 1	Label 2	Numerical value 2	Label 3	Numerical value 3	Label 4	Numerical value 4	Card columns 79 to 80
CONSTS	HCK	1.4388	R	1.98726	SCONST	-3.66511			
ATOM	AL	26.9800	AL1(S)	2		16			
ATOM	AR	39.9440	AR1(G)	12		61			
ATOM	B	10.8200	B1(S)	2		6			
ATOM	BE	9.013	BE1(S)	4		13			
ATOM	C	12.0110	C1(S)	12		15			
ATOM	CA	40.0800	CA1(S)	4		61			
ATOM	CL	35.4570	CL2(G)	30		156			
ATOM	CS	132.9100	CS1(S)	2		72			
ATOM	E	.000548613	E1(G)	2		2			
ATOM	F	19.0000	F2(G)	30		6			
ATOM	H	1.008	H2(G)	2		2			
ATOM	HE	4.003	HE1(G)	4		1			
ATOM	K	39.1000	K1(S)	2		32			
ATOM	LI	6.940	LI1(S)	2		8			
ATOM	MG	24.3200	MG1(S)	4		33			
ATOM	N	14.0080	N2(G)	30		20			
ATOM	NA	22.9910	NA1(S)	2		18			
ATOM	NE	20.1830	NE1(G)	12		1			
ATOM	O	16.0000	O2(G)	40		15			
ATOM	P	30.9750	P1(S)	30		170			
ATOM	S	32.0660	S1(S)	40		215			
ATOM	SI	28.0900	SI1(S)	12		75			
EFDATA	MG1(S)	0.	HZERO	-1190.3000	MELTPT	923.0000	T NO.	28.0000	BCDUM000
									BCDUM001
									BCDUM002
									BCDUM003

Binary EF data for Mg(s)


EFDATA H2(G) 0. HZERO -2023.8000 MELTPT 0. T NO. 61.0000
BCDUM000
BCDUM001
BCDUM002
BCDUM003
BCDUM004
BCDUM005
BCDUM006
BCDUM007
BCDUM008
Binary EF data for $H_2(g)$
EFDATA H1(G) 0. HZERO 50616.4995 MELTPT -0. T NO. 61.0000
BCDUM000
BCDUM001
BCDUM002
BCDUM003
BCDUM004
BCDUM005
BCDUM006
BCDUM007
BCDUM008
Binary EF data for H(g)
EFDATA O1(G) 0. HZERO 57949.1504 MELTPT -0. T NO. 61.0000
BCDUM000
BCDUM001
BCDUM002
BCDUM003
BCDUM004
BCDUM005
BCDUM006
BCDUM007
BCDUM008
Binary EF data for O(g)
LISTEF
EFDATA O2(G) 0. HZERO -2074.7390 MELTPT -0. T NO. 61.0000
BCDUM000
BCDUM001
BCDUM002
BCDUM003
BCDUM004
BCDUM005
BCDUM006
BCDUM007
BCDUM008
Binary EF data for $O_2(g)$

```
AR1(G)                  ASINDH                  T       298.15
TEMP    T       298.15  T       1000.           I       1000.   T       6000.
METHODCOEF              T       6000.           REDUCE          H298HO  1481.254
DATA    T       100.    E1      0               CH-HO   0.      CS      4.3661076
DATA    C1      2.5
AR1(G)TPUNCH            300TPUNCH               1000TPUNCH              5000
FINISH

H201(G)                 ASINDH  -57103.5        T       0.
TEMP    T       5000.
INTERM
METHOD                  NRRAO2
DATA    V1      3656.65 V2      1594.78         V3      3755.79         X11     -45.18
DATA    X12     -15.14  X33     -44.62          X13     -165.48         X22     -17.04
DATA    Y233    -.81    Y333    -.45            X23     -19.99
DATA    Y111    .47     Y112    -.10            Y113    .68             Y122    -0.1
DATA    Y123    -1.72   Y133    1.17            Y222    -.60            Y223    1.55
DATA    AO      27.848  BO      14.5064         CO      9.28285         SYMNO   2.
DATA    ALFAA1  .750    ALFAA2  -2.941          ALFAA3  1.253           ALFAB1  .238
DATA    ALFAB2  -.160   ALFAB3  .078            ALFAC1  .2018           ALFAC2  .1392
DATA    ALFAC3  .1445   RHO     .0000213        STATWT  1.
FINISH

MG1(G)                  DELTAH  35.6            KCAL                    T       298.15
EFTAPE
LSTSQST         1000.   T       5000.           TCONST  1000.          EXP     0.
LSTSQSEXP       1.      EXP     2.              EXP     3.
DATE    4/63
METHODTEMPER            FILL
DATA            0       0.      0       21850.368       1       21870.426       2       21911.140       3
DATA    1       35051.36        2       46403.14        3       47957.035       2       47957.018       3
DATA    1       41197.37        0       43503.34        0       47841.20        1       47844.44        4
DATA    2       47851.14        1       49346.71        2       53134.70        7       54192.43        4
DATA    1       47957.047       0       57812.72        1       57833.28        2       57873.89        3
DATA    1       51872.36        0       52556.37        4       54252.6         1       54699.4         5
DATA    2       54676.76        3                       4                       3       54676.66        4
DATA    2       55891.83        0       56187.03        1.5     57018.8         2       57020.1         6
DATA    2       56308.43        7       56968.31        10      57204.22        3       57204.22        5
DATA    1       57853.5         0       58009.46        4       58478.4                                 7
DATA    1       58023.27        7       58442.62        10      58575.54        3       58575.54        6
DATA    1       58962.49        2       59690.02        7       59880.3         13.5    59935.38        8
DATA    2       59041.09        7       59317.4         10      59400.77        3                       7
DATA    1       59648.2         2       60127.31        7       60263.0         13.5    60301.30        9
DATA    1       60103.5         2       60435.15        7       60534.5         13.5    60562.64        10
DATA    1       60420.2         2       60658.37        7       60734.0         13.5    60755.78        11
DATA    1       60649.2         2       60826.6         7       60884.8         13.5    60902.53        12
DATA    1       60820.9         2       60955.8         7       61002.2         3       61016.42        13
DATA    1       60952.0         7       61094.6         3       61106.98        IP      61669.14        14
FINISH

MG1(S)                  ASINDH  0.              T       298.15
LOGK
METHODREADIN            MELTPT  923.
DATA    T       100.    CP      3.753           H-HO/T  1.529           S       2.263
DATA    T       200.    CP      5.418           H-HO    630.9           S       5.511
DATA    T       298.15  CP      5.929           H-HO    1190.3          S       7.780
DATA    T       300.    CP      5.937           H-HO    1201.3          S       7.817
DATA    T       400.    CP      6.241           H-HO    1811.3          S       9.569
DATA    T       500.    CP      6.493           H-HO    2447.7          S       10.989
DATA    T       600.    CP/R    3.4047          H-HORT  2.608           -FHORT  3.5286
DATA    T       700.    CP      7.084           H-HO    3802.0          S/R     6.6730
DATA    T       800.    CP      7.426           H-HO    4527.3          -FHO/T  8.569875
DATA    T       900.    CP      7.792           H-HO    5288.0          S       15.125
DATA    T       923.    CP      7.880           H-HO    5468.2          S       15.322

METHODCOEF              DELTAH2140.
DATA    T       923.    T       6000.           C1      8.0             E1      0.
FINISH

MG101(G)                DISSOC          90KCAL
TEMP    T       298.15  T       1000.           I       1000.           T       6000.
REFNCESPECTR            23DISSOC        319
LOGK
METHODPANDK
DATA    WE      782.99  WEXE    5.15            BE      .5713           ALPHA1  .0050           1
DATA    DE      1.21-06 BETA1   .02-06                                                          1
DATA    STATWT  2.      TO      3503.28         WE      662.69          WEXE    3.89            2
DATA    BE      .5029   ALPHA1  .0046           DE      1.172-06        BETA1   -.05-06         2
DATA    STATWT  1.0     TO      20003.57        WE      821.91          WEXE    4.74            3
DATA    BE      .5791   ALPHA1  .0045           DE      1.13-06         BETA1   .025-06         3
FINISH
```

Listed output. - The listed output is as follows:

```
CONSTS                                HCK      1.4388000   R        1.9872600   SCONST   -3.6651100
ATOM    AL        26.9800000   AL1(S)        2.                    16.
ATOM    AR        39.9439998   AR1(G)       12.                    61.
ATOM    B         10.8200001   B1(S)         2.                     6.
ATOM    BE        9.0130000    BE1(S)        4.                    13.
ATOM    C         12.0110000   C1(S)        12.                    15.
ATOM    CA        40.0799999   CA1(S)        4.                    61.
ATOM    CL        35.4569998   CL2(G)       30.                   156.
ATOM    CS        132.91000    CS1(S)        2.                    72.
ATOM    F         0.54861300E-03 F1(G)       2.                     2.
ATOM    F         19.          F2(G)        30.                     6.
ATOM    H         1.0080000    H2(G)         2.                     2.
ATOM    HE        4.0030000    HE1(G)        4.                     1.
ATOM    K         39.0999999   K1(S)         2.                    32.
ATOM    LI        6.9400000    LI1(S)        2.                     8.
ATOM    MG        24.3199999   MG1(S)        4.                    33.
ATOM    N         14.0080000   N2(G)        30.                    20.
ATOM    NA        22.9909999   NA1(S)        2.                    18.
ATOM    NE        20.1830001   NE1(G)       12.                     1.
ATOM    O         16.          O2(G)        40.                    15.
ATOM    P         30.9749999   P1(S)        30.                   170.
ATOM    S         32.0660000   S1(S)        40.                   215.
ATOM    SI        28.0899999   SI1(S)       12.                    75.
EFDATA  MG1(S)                 HZERO    -1190.30000   MELTPT    923.    T NO.      28.
EFDATA  H2(G)                  HZERO    -2023.80000   MELTPT              T NO.      61.
EFDATA  H1(G)                  HZERO     50616.500    MELTPT              T NO.      61.
EFDATA  O1(G)                  HZERO     57949.150    MELTPT              T NO.      61.
LISTEF
EFDATA  O2(G)                  HZERO    -2074.73901   MELTPT              T NO.      61.
```

T	H-H0/RT	-(F-H0)/RT	T	H-H0/RT	-(F-H0)/RT
100.000	3.48991537	17.34113860	200.000	3.49561310	19.76229215
298.150	3.50165999	21.15892196	300.000	3.50185835	21.18058252
400.000	3.51979873	22.19011092	500.000	3.55167121	22.97872639
600.000	3.59312224	23.62983251	700.000	3.63907662	24.18713498
800.000	3.68591803	24.67612839	900.000	3.73153207	25.11291909
1000.000	3.77485391	25.50834012	1100.000	3.81546095	25.87004900
1200.000	3.85329697	26.20367932	1300.000	3.88850057	26.51351452
1400.000	3.92130366	26.80289793	1500.000	3.95197189	27.07449675
1600.000	3.98077118	27.33048058	1700.000	4.00794911	27.57263613
1800.000	4.03372759	27.80245972	1900.000	4.05829930	28.02121544
2000.000	4.08182836	28.22998142	2100.000	4.10445231	28.42968559
2200.000	4.12628472	28.62113166	2300.000	4.14741892	28.80502057
2400.000	4.16793066	28.98196840	2500.000	4.18788111	29.15251803
2600.000	4.20731878	29.31714988	2700.000	4.22628319	29.47629261
2800.000	4.24480504	29.63032866	2900.000	4.26290929	29.77960134
3000.000	4.28061569	29.92442012	3100.000	4.29794002	30.06506443
3200.000	4.31489480	30.20178723	3300.000	4.33149058	30.33481860
3400.000	4.34773630	30.46436810	3500.000	4.36363965	30.59062862
3600.000	4.37920725	30.71377516	3700.000	4.39444572	30.83396959
3800.000	4.40936083	30.95136023	3900.000	4.42395830	31.06608510
4000.000	4.43824393	31.17827082	4100.000	4.45222443	31.28803492
4200.000	4.46590537	31.39548731	4300.000	4.47929311	31.50073051
4400.000	4.49239421	31.60385752	4500.000	4.50521505	31.70495868
4600.000	4.51776236	31.80411649	4700.000	4.53004336	31.90140796
4800.000	4.54206461	31.99690747	4900.000	4.55383325	32.09068251
5000.000	4.56535631	32.18279886	5100.000	4.57664078	32.27331686
5200.000	4.58769345	32.36229420	5300.000	4.59852165	32.44978428
5400.000	4.60913157	32.53583956	5500.000	4.61953032	32.62050772
5600.000	4.62972367	32.70383739	5700.000	4.63971829	32.78586960
5800.000	4.64952016	32.86664820	5900.000	4.65913516	32.94621086
6000.000	4.66856867	33.02459717			

```
AR1(G)                          ASINDH                  T          298.15000

TEMP        T        298.15000  T          1000.        I          1000.      T          6000.

METHOD      COEF                                        REDUCE                H298HO     1481.25400

DATA        T            100.   T          6000.

DATA        C1       2.5000000  E1                      CH-HO                 CS         4.3661076

DATA        TPUNCH       300.   TPUNCH     1000.        TPUNCH     5000.

FINISH

PUNCHED BINARY CARDS--

AR1(G)              0.          0.30000000E 03  0.50000000E 04  0.10000000E 04  0.50000000E 04  0.25000000E 01
  0.          0.                0.              0.              -0.74537499E 03 0.43661076E 01  0.30000000E 03
  0.10000000E 04  0.25000000E 01  0.              0.              0.              0.              -0.74537499E 03

AR1(G)              0.          0.43661076E 01  0.              0.              0.              0.
  0.          0.                0.              0.              0.              0.              0.
  0.          0.                0.              0.              0.              0.                      000000

AR1(G)

HZERO =    -1481.254

        T        CP/R      (H-HO)/RT   (H-H298)/RT       S/R      -(F-HO)/RT  -(F-H298)/RT       H/RT        -F/RT
   298.15   2.5000000   2.5000000   0.0000000  18.6100993  16.1100993  18.6100991   0.0000001  18.6100991
  1000      2.5000000   2.5000000   1.7546250  21.6354954  19.1354954  19.8808703   1.7546250  19.8808703
  2000      2.5000000   2.5000000   2.1273125  23.3683634  20.8683634  21.2410507   2.1273125  21.2410507
  3000      2.5000000   2.5000000   2.2515416  24.3820262  21.8820262  22.1304843   2.2515416  22.1304843
  4000      2.5000000   2.5000000   2.3136562  25.1012313  22.6012313  22.7875750   2.3136562  22.7875750
  5000      2.5000000   2.5000000   2.3509250  25.6590903  23.1590903  23.3081651   2.3509250  23.3081651
  6000      2.5000000   2.5000000   2.3757708  26.1148942  23.6148942  23.7391231   2.3757708  23.7391231

HZERO =    -1481.254

        T          CP          H-HO        H-H298         S          -(F-HO)     -(F-H298)        H           -F
   298.15   4.9681500   1481.2539      0.0000  36.983106    9545.2592  11026.5129      0.0000  11026.5129
  1000      4.9681500   4968.1500   3486.8950  42.995354   38027.2041  39508.4580   3486.8961  39508.4580
  2000      4.9681500   9936.2999   8455.0459  46.439013   82941.7266  84422.9805   8455.0459  84422.9805
  3000      4.9681500  14904.4498  13423.1958  48.453425  130455.8252 131937.0781  13423.1958 131937.0781
  4000      4.9681500  19872.5999  18391.3459  49.882673  179658.0898 181139.3438  18391.3459 181139.3438
  5000      4.9681500  24840.7498  23359.4958  50.991283  230115.6660 231596.9180  23359.4958 231596.9180
  6000      4.9681500  29808.8997  28327.6455  51.897084  281573.6055 283054.8555  28327.6455 283054.8555

H201(G      )           ASINDH     -57103.500  T
TEMP)       T        5000.

INTERM

METHOD                          NRRA02

MOLECULAR WT.= 18.01600

DATA        V1       3656.64999 V2         1594.78000  V3         3755.79001  X11       -45.1799998
DATA        X12       -15.1400000 X33       -44.6199999 X13        -165.48000  X22       -17.0400000
DATA        Y233       -0.81000000 Y333      -0.45000000 X23        -19.9900000
DATA        Y111        0.47000000 Y112      -0.09999999 Y113        0.68000000 Y122       -0.09999999
DATA        Y123       -1.7200000 Y133        1.1700000 Y222       -0.60000000 Y223        1.5500000
DATA        A0         27.8480000 B0         14.5064000  C0         9.2828500  SYMNO          2.
DATA        ALFAA1      0.75000000 ALFAA2    -2.9410000  ALFAA3      1.2530000  ALFABI     0.23800000
DATA        ALFAB2     -0.16000000 ALFAB3     0.07799999 ALFAC1     0.20180000 ALFAC2     0.13919999
DATA        ALFAC3      0.14449999 RHO     0.21300000E-04 STATWT          1.

FINISH
```

```
A0 = 27.848000      B0 = 14.506400      C0 =  9.282850      RHO = 0.21300000E-04

AI = 0.0325387     ALPHA A = 0.7500000     ALPHA B = 0.2380000     ALPHA C = 0.2018000     I =1

AI =-0.0508216     ALPHA A =-2.9410000     ALPHA B =-0.1600000     ALPHA C = 0.1392000     I =2

AI = 0.0329688     ALPHA A = 1.2530000     ALPHA B = 0.0780000     ALPHA C = 0.1445000     I =3

THETA(1) =.44832230     THETA(2) =.37102938     THETA(3) =.
Y(1,1,1) =  0.470     Y(1,1,2) = -0.100     Y(1,1,3) =  0.680     Y(1,2,2) = -0.100     Y(1,2,3) = -1.720
Y(1,3,3) =  1.170     Y(2,2,2) = -0.600     Y(2,2,3) =  1.550     Y(2,3,3) = -0.810     Y(3,3,3) = -0.450

X(I,J)
  -42.7750  -16.4000 -162.6400
  -16.4000  -19.0150  -19.3700
 -162.6400  -19.3700  -46.4650

LEVEL =-0

V(1) =3656.6500(1)      G11 =  0.

V(2) =1594.7800(1)      G22 =  0.

V(3) =3755.7900(1)      G33 =  0.


T = 5000.000

   U =1.0522376   R =0.3491556   S =1.5364656   I = 1

   U =0.4589139   R =0.6319697   S =2.7171673   I = 2

   U =1.0807661   R =0.3393355   S =1.5136275   I = 3

CONTRIBUTION            Q            LN Q          H-H0/RT          CP/R
   ELECTR             1.0000          0.          0.          0.
   H.O.               6.3191       1.84358367     1.90763232     2.80335173
   R.R.            2964.7072       7.99453354     1.50000000     1.50000000
   RHO                1.1124       0.10650000     0.10650000     0.21300000
   THETA              1.0009       0.00099003    -0.00089042     0.00000075

FIRST ORDER CORRECTIONS
   ALPHA              0.9546      -0.04642865    -0.04005801    -0.06193407
   XIJ                1.0710       0.06856769     0.12646388     0.30333602
   YIJK               1.0043       0.00428596     0.01142752     0.03873087
   AXIJ               0.9930      -0.00702780    -0.01502021    -0.04561825

 SECOND ORDER CORRECTIONS
   (XIJ)2             1.0098       0.00975028     0.02659192     0.09221710
   XY                 1.0027       0.00264834     0.00939342     0.04116510
   AX2                0.9977      -0.00226218    -0.00755046    -0.03202735


H201(G)

HZERO =  -57103.500

        T          CP/R        (H-H0)/RT                S/R        -(F-H0)/RT                H/RT          -F/RT

     5000       7.3522221       6.1244898       38.0642924       31.9398026          0.3775317      37.6867604

HZERO =  -57103.500

        T           CP          H-H0                S          -(F-H0)                H          -F

     5000      14.6107768     60854.7671       75.643645     317363.4570          3751.2682     374466.9531
```

```
MG1(G)                          DELTAH    35.5999999    KCAL                T       298.15000
EFTAPE
LSTSQS   T          1000.       T          5000.      TCONST     1000.     EXP
LSTSQS   EXP          1.       EXP           2.      EXP           3.
DATE       4/63
METHOD   TEMPER                 FILL
DATA          0                      0   21850.368        1   21870.426    2   21911.140  3
DATA     1          35051.360        2   46403.140        3   47957.035    2   47957.018  3
DATA       1        41197.370        0   43503.340        0   47841.200    1   47844.440  4
DATA       2        47851.140        1   49346.710        2   53134.700    7   54192.430  4
DATA       1        47957.047        0   57812.720        1   57833.280    2   57873.890  3
DATA       1        51872.360        0   52556.370        4   54252.600    1   54699.400  5
DATA       2        54676.760        3                    4                3   54676.660  4
DATA       2        55891.830        0   56187.030      1.5   57018.800    2   57020.100  6
DATA       2        56308.430        7   56968.310       10   57204.220    3   57204.220  5
DATA       1        57853.500        0   58009.460        4   58478.400                   7
DATA       1        58023.270        7   58442.620       10   58575.540    3   58575.540  6
DATA       1        58962.490        2   59690.020        7   59880.300   13.5 59935.380  8
DATA       2        59041.090        7   59317.400       10   59400.770    3              7
DATA       1        59648.200        2   60127.310        7     60263.     13.5 60301.300  9
DATA       1        60103.500        2   60435.150        7   60534.500    13.5 60562.640 10
DATA       1        60420.200        2   60658.370        7     60734.     13.5 60755.780 11
DATA       1        60649.200        2   60826.600        7   60884.800    13.5 60902.530 12
DATA       1        60820.900        2   60955.800        7   61002.200    3    61016.420 13
DATA       1          60952.         7   61094.600        3   61106.980    IP   61669.140 14
MG1(G)                                                                      MG1(G)
FINISH
```

B	N	PRED. SUM(2J+1)	ACT. SUM(2J+1)	DIFF	MAX LEVEL	2J+1, MAX LEVEL
4.0	3	36.0	42.0	-6.0	57873.8901	5.0
4.0	4	64.0	64.0	0.	54676.7598	9.0
4.0	5	100.0	64.0	36.0	57204.2202	43.0
4.0	6	144.0	61.0	83.0	58575.5400	90.0
4.0	7	196.0	61.0	135.0	59400.7700	142.0
4.0	8	256.0	51.0	205.0	59935.3799	233.0
4.0	9	324.0	51.0	273.0	60301.2998	301.0
4.0	10	400.0	51.0	349.0	60562.6401	377.0
4.0	11	484.0	51.0	433.0	60755.7798	461.0
4.0	12	576.0	51.0	525.0	60902.5298	553.0
4.0	13	676.0	30.0	646.0	61016.4199	653.0
4.0	14	784.0	25.0	759.0	61106.9800	766.0

LEAST SQUARES

T	CP/R INPUT INPUT-CALC	CP/R CALC FRACTION	HH/RT INPUT INPUT-CALC	HH/RT CALC FRACTION	S/R INPUT INPUT-CALC	S/R CALC FRACTION	-FH/RT INPUT INPUT-CALC	-FH/RT CALC FRACTION
1000.00	2.5000000	2.4999999	2.5000000	2.4999999	20.8912263	20.6912261	18.3912263	18.3912261
	0.0000001	0.0000000	0.0000031	0.0000030	0.0000002	0.0000000	0.0000002	0.0000000
1100.00	2.5000000	2.5016205	2.5000000	2.5000796	21.1295018	21.1295836	18.6295018	18.6295040
	-0.0016205	-0.0006482	-0.0000796	-0.0000318	-0.0000818	-0.0000039	-0.0000021	-0.0000001
1200.00	2.5000000	2.5025073	2.5000000	2.5002497	21.3470304	21.3472960	18.8470304	18.8470461
	-0.0025073	-0.0010029	-0.0002497	-0.0000999	-0.0002656	-0.0000124	-0.0000157	-0.0000008
1300.00	2.5000001	2.5027796	2.5000000	2.5004374	21.5471370	21.5476179	19.0471370	19.0471804
	-0.0027795	-0.0011118	-0.0004374	-0.0001749	-0.0004809	-0.0000223	-0.0000434	-0.0000023
1400.00	2.5000007	2.5025565	2.5000000	2.5005993	21.7324071	21.7330883	19.2324071	19.2324889
	-0.0025558	-0.0010223	-0.0005993	-0.0002397	-0.0006812	-0.0000313	-0.0000818	-0.0000042
1500.00	2.5000030	2.5019571	2.5000001	2.5007115	21.9048891	21.9057286	19.4048891	19.4050169
	-0.0019542	-0.0007817	-0.0007114	-0.0002846	-0.0008395	-0.0000383	-0.0001278	-0.0000066
1600.00	2.5000098	2.5011006	2.5000005	2.5007637	22.0662360	22.0671749	19.5662355	19.5664113
	-0.0010908	-0.0004363	-0.0007632	-0.0003053	-0.0009389	-0.0000425	-0.0001757	-0.0000090
1700.00	2.5000277	2.5001061	2.5000015	2.5007546	22.2177985	22.2187736	19.7177970	19.7180190
	-0.0000784	-0.0000313	-0.0007531	-0.0003012	-0.0009751	-0.0000439	-0.0002220	-0.0000113
1800.00	2.5000693	2.4990927	2.5000039	2.5006903	22.3606968	22.3616467	19.8606930	19.8609562
	0.0009766	0.0003906	-0.0006863	-0.0002745	-0.0009499	-0.0000425	-0.0002632	-0.0000132
1900.00	2.5001563	2.4981796	2.5000094	2.5005814	22.4958711	22.4967408	19.9958618	19.9961593
	0.0019766	0.0007906	-0.0005720	-0.0002288	-0.0008698	-0.0000387	-0.0002975	-0.0000149
2000.00	2.5003231	2.4974859	2.5000205	2.5004428	22.6241159	22.6248617	20.1240954	20.1244187
	0.0028372	0.0011347	-0.0004223	-0.0001689	-0.0007458	-0.0000330	-0.0003233	-0.0000161
2100.00	2.5006204	2.4971308	2.5000414	2.5002920	22.7461135	22.7467041	20.2460723	20.2464120
	0.0034897	0.0013955	-0.0002506	-0.0001002	-0.0005906	-0.0000260	-0.0003397	-0.0000168
2200.00	2.5011179	2.4972333	2.5000781	2.5001487	22.8624530	22.8628712	20.3623750	20.3627224
	0.0038846	0.0015531	-0.0000706	-0.0000282	-0.0004182	-0.0000183	-0.0003474	-0.0000171
2300.00	2.5019061	2.4979127	2.5001392	2.5000344	22.9736483	22.9738898	20.4735093	20.4738553
	0.0039934	0.0015961	0.0001048	0.0000419	-0.0002415	-0.0000105	-0.0003459	-0.0000169
2400.00	2.5030976	2.4992880	2.5002360	2.4999720	23.0801520	23.0802267	20.5799162	20.5802546
	0.0038096	0.0015220	0.0002640	0.0001056	-0.0000746	-0.0000032	-0.0003383	-0.0000164
2500.00	2.5048760	2.5014784	2.5003830	2.4999855	23.1823664	23.1822941	20.6819835	20.6823084
	0.0033476	0.0013365	0.0003974	0.0001590	0.0000722	0.0000031	-0.0003250	-0.0000157
2600.00	2.5072446	2.5046031	2.5005979	2.5000999	23.2806518	23.2804615	20.7800541	20.7803617
	0.0026415	0.0010536	0.0004981	0.0001992	0.0001903	0.0000082	-0.0003076	-0.0000148
2700.00	2.5105232	2.5087810	2.5009019	2.5003406	23.3753347	23.3750608	20.8744330	20.8747201
	0.0017422	0.0006940	0.0005613	0.0002244	0.0002739	0.0000117	-0.0002871	-0.0000137
2800.00	2.5148447	2.5141315	2.5013193	2.5007339	23.4667113	23.4663923	20.9653921	20.9656582
	0.0007131	0.0002836	0.0005854	0.0002340	0.0003190	0.0000136	-0.0002661	-0.0000127
2900.00	2.5204001	2.5207737	2.5018776	2.5013065	23.5550535	23.5547283	21.0531759	21.0534217
	-0.0003736	-0.0001482	0.0005711	0.0002283	0.0003252	0.0000138	-0.0002458	-0.0000117
3000.00	2.5273846	2.5288266	2.5026072	2.5020855	23.6406124	23.6403184	21.1380053	21.1382327
	-0.0014420	-0.0005706	0.0005217	0.0002084	0.0002940	0.0000124	-0.0002275	-0.0000108
3100.00	2.5359925	2.5384095	2.5035407	2.5030984	23.7236207	23.7233899	21.2200801	21.2202914
	-0.0024170	-0.0009531	0.0004423	0.0001767	0.0002308	0.0000097	-0.0002112	-0.0000100
3200.00	2.5464129	2.5496414	2.5047127	2.5043730	23.8042943	23.8041537	21.2995818	21.2997806
	-0.0032285	-0.0012679	0.0003397	0.0001356	0.0001407	0.0000059	-0.0001988	-0.0000093
3300.00	2.5588248	2.5626415	2.5061591	2.5059371	23.8828366	23.8828049	21.3766775	21.3768678
	-0.0038167	-0.0014916	0.0002220	0.0000886	0.0000317	0.0000013	-0.0001903	-0.0000089
3400.00	2.5733955	2.5775290	2.5079169	2.5078191	23.9594359	23.9595237	21.4515193	21.4517045
	-0.0041336	-0.0016063	0.0000978	0.0000390	-0.0000877	-0.0000037	-0.0001853	-0.0000086
3500.00	2.5902756	2.5944228	2.5100232	2.5100471	24.0342700	24.0344784	21.5242469	21.5244312
	-0.0041473	-0.0016011	-0.0000240	-0.0000095	-0.0002084	-0.0000087	-0.0001843	-0.0000086
3600.00	2.6095974	2.6134423	2.5125150	2.5126500	24.1075051	24.1078269	21.5949903	21.5951767
	-0.0038449	-0.0014734	-0.0001351	-0.0000538	-0.0003219	-0.0000133	-0.0001864	-0.0000086
3700.00	2.6314733	2.6347066	2.5154286	2.5156563	24.1792979	24.1797173	21.6638694	21.6640608
	-0.0032332	-0.0012287	-0.0002277	-0.0000905	-0.0004194	-0.0000173	-0.0001915	-0.0000088
3800.00	2.6559945	2.6583347	2.5187992	2.5190948	24.2497945	24.2502890	21.7309954	21.7311940
	-0.0023402	-0.0008811	-0.0002957	-0.0001174	-0.0004945	-0.0000204	-0.0001986	-0.0000091
3900.00	2.6832312	2.6844458	2.5226603	2.5229944	24.3191314	24.3196726	21.7964711	21.7966781
	-0.0012146	-0.0004527	-0.0003341	-0.0001325	-0.0005412	-0.0000222	-0.0002069	-0.0000095
4000.00	2.7132329	2.7131589	2.5270438	2.5273840	24.3874369	24.3879931	21.8603933	21.8606091
	0.0000740	0.0000273	-0.0003402	-0.0001346	-0.0005562	-0.0000228	-0.0002158	-0.0000099
4100.00	2.7460277	2.7445937	2.5319792	2.5322928	24.4548316	24.4553690	21.9228525	21.9230762
	0.0014340	0.0005222	-0.0003136	-0.0001239	-0.0005374	-0.0000220	-0.0002236	-0.0000102
4200.00	2.7816251	2.7788685	2.5374938	2.5377499	24.5214257	24.5219123	21.9839320	21.9841623
	0.0027567	0.0009910	-0.0002561	-0.0001009	-0.0004866	-0.0000198	-0.0002303	-0.0000105
4300.00	2.8200166	2.8161030	2.5436123	2.5437844	24.5873232	24.5877309	22.0437109	22.0439465
	0.0039136	0.0013878	-0.0001720	-0.0000676	-0.0004077	-0.0000166	-0.0002356	-0.0000107
4400.00	2.8611778	2.8564162	2.5503568	2.5504256	24.6526198	24.6529274	22.1022632	22.1025016
	0.0047617	0.0016642	-0.0000688	-0.0000270	-0.0003076	-0.0000125	-0.0002384	-0.0000108
4500.00	2.9047892	2.8999273	2.5577315	2.5577028	24.7173891	24.7176003	22.1596577	22.1598973
	0.0048620	0.0016738	0.0000286	0.0000112	-0.0002112	-0.0000085	-0.0002396	-0.0000108
4600.00	2.9512089	2.9467552	2.5657746	2.5656453	24.7817349	24.7818439	22.2159605	22.2161984
	0.0044537	0.0015091	0.0001292	0.0000504	-0.0001090	-0.0000044	-0.0002379	-0.0000107
4700.00	3.0001490	2.9970195	2.5744878	2.5742826	24.8457201	24.8457499	22.2712324	22.2714672
	0.0031295	0.0010431	0.0002051	0.0000797	-0.0000298	-0.0000012	-0.0002348	-0.0000105
4800.00	3.0516366	3.0508386	2.5838878	2.5836439	24.9094193	24.9094052	22.3255317	22.3257611
	0.0007980	0.0002615	0.0002438	0.0000944	0.0000141	0.0000006	-0.0002294	-0.0000103
4900.00	3.1055283	3.1083326	2.5939796	2.5937589	24.9728916	24.9728959	22.3789120	22.3791368
	-0.0028043	-0.0009030	0.0002207	0.0000851	-0.0000043	-0.0000002	-0.0002248	-0.0000100
5000.00	3.1617320	3.1696197	2.6047688	2.6046568	25.0361936	25.0363030	22.4314249	22.4316461
	-0.0078878	-0.0024948	0.0001120	0.0000430	-0.0001094	-0.0000044	-0.0002213	-0.0000099

MAX-RFL FRR CP/R	=	0.002495	TEMP =	5000.	AVER RFL ERR CP/R	=	0.000998	REL LST SQ ERR CP/R	=	0.001141
MAX RFL FRR HH/RT	=	0.000305	TEMP =	1600.	AVER RFL ERR HH/RT	=	0.000129	REL LST SQ ERR HH/RT	=	0.000155
MAX RFL ERR S/R	=	0.000044	TEMP =	1700.	AVER REL ERR S/R	=	0.000016	REL LST SQ ERR S/R	=	0.000020
MAX REL FRR FH/RT	=	0.000017	TEMP =	2200.	AVER REL ERR FH/RT	=	0.000010	REL LST SQ ERR FH/RT	=	0.000011
MAX FRR CP/R	=	0.007888	TEMP =	5000.	AVER ERR CP/R	=	0.002660	LST SQ ERR CP/R	=	0.003091
MAX FRR HH/RT	=	0.000763	TEMP =	1600.	AVER ERR HH/RT	=	0.000325	LST SQ ERR HH/RT	=	0.000389
MAX ERR S/R	=	0.000975	TEMP =	1700.	AVER ERR S/R	=	0.000363	LST SQ ERR S/R	=	0.000459
MAX FRR FH/RT	=	0.000347	TEMP =	2200.	AVER ERR FH/RT	=	0.000216	LST SQ ERR FH/RT	=	0.000233

CP/R = 2.4172320E 00T** 0. 1.6512134E-04T** 1.0 -1.0220946E-07T** 2.0 1.9856143E-11T** 3.0

(H-H0)/R CONSTANT = 0.29313141E 02. H/R(A6) CONSTANT = 0.17198050E 05. S/R CONSTANT = 0.40729443E 01

PUNCHED BINARY CARDS--

MG1(G) 61669.140 0.10000000E 04 0.50000000E 04 0.10000000E 04 0.50000000E 04 0.24172320E 01
0.16512134E-03 -0.10220946E-06 0.19856143E-10 0. 0.17198050E 05 0.40729443E 01 0.
0. 0. 0. 0. 0. 0. 4/63
MG1(G) MG1(G)

HZERO = 34118.745

T	CP/R	(H-H0)/RT	(H-H298)/RT	S/R	-(F-H0)/RT	-(F-H298)/RT	H/RT	-F/RT
100	2.5000000	2.5000000	-4.9537500	15.1347638	12.6347638	20.0885136	174.1873722	-159.0526066
200	2.5000000	2.5000000	-1.2268750	16.8676317	14.3676318	18.0945067	88.3436861	-71.4760542
298.15	2.5000000	2.5000000	0.0000000	17.8658302	15.3658303	17.8658302	60.0842266	-42.2183962
300	2.5000000	2.5000000	0.0154167	17.8812945	15.3812946	17.8658779	59.7291241	-41.8478293
400	2.5000000	2.5000000	0.6365625	18.6004996	16.1004996	17.9639370	45.4218431	-26.8213434
500	2.5000000	2.5000000	1.0092500	19.1583586	16.6583586	18.1491084	36.8374743	-17.6791158
600	2.5000000	2.5000000	1.2577083	19.6141624	17.1141624	18.3564539	31.1145620	-11.5003996
700	2.5000000	2.5000000	1.4351786	19.9995391	17.4995391	18.5643604	27.0267675	-7.0272284
800	2.5000000	2.5000000	1.5682812	20.3333676	17.8333676	18.7650862	23.9609215	-3.6275539
900	2.5000000	2.5000000	1.6718056	20.6278250	18.1278250	18.9560194	21.5763748	-0.9485497
1000	2.5000000	2.5000000	1.7546250	20.8912263	18.3912263	19.1366012	19.6687372	1.2224891
1100	2.5000000	2.5000000	1.8223864	21.1295018	18.6295018	19.3071153	18.1079428	3.0215589
1200	2.5000000	2.5000000	1.8788542	21.3470304	18.8470304	19.4681761	16.8072810	4.5397494
1300	2.5000001	2.5000000	1.9266346	21.5471370	19.0471370	19.6205022	15.7067209	5.8404161
1400	2.5000007	2.5000000	1.9675893	21.7324071	19.2324071	19.7648177	14.7633837	6.9690233
1500	2.5000030	2.5000001	2.0030834	21.9048891	19.4048891	19.9018056	13.9458250	7.9590642
1600	2.5000098	2.5000005	2.0341411	22.0662360	19.5662355	20.0320947	13.2304612	8.8357748
1700	2.5000277	2.5000015	2.0615456	22.2177985	19.7177970	20.1562529	12.5992587	9.6185398
1800	2.5000693	2.5000039	2.0859067	22.3606968	19.8606930	20.2747900	12.0381913	10.3225056
1900	2.5001563	2.5000094	2.1077068	22.4958711	19.9958618	20.3881643	11.5361869	10.9596843
2000	2.5003231	2.5000205	2.1273330	22.6241159	20.1240954	20.4967828	11.0843891	11.5397269
2100	2.5006204	2.5000414	2.1451009	22.7461135	20.2460723	20.6010127	10.6756305	12.0704832
2200	2.5011179	2.5000781	2.1612712	22.8624530	20.3623750	20.7011817	10.3040495	12.5584035
2300	2.5019061	2.5001392	2.1760631	22.9736483	20.4735093	20.7975852	9.9648075	13.0088409
2400	2.5030976	2.5002360	2.1896631	23.0801520	20.5799162	20.8904891	9.6538764	13.4262757
2500	2.5048260	2.5003830	2.2022330	23.1823664	20.6819835	20.9801333	9.3678778	13.8144885
2600	2.5072446	2.5005979	2.2139152	23.2806518	20.7800541	21.0667367	9.1039584	14.1766936
2700	2.5105232	2.5009019	2.2248371	23.3753347	20.8744330	21.1504977	8.8596934	14.5156415
2800	2.5148447	2.5013193	2.2351139	23.4667113	20.9653921	21.2315974	8.6330111	14.8337002
2900	2.5204001	2.5018776	2.2448518	23.5550535	21.0531759	21.3102016	8.4221318	15.1329217
3000	2.5273846	2.5026072	2.2541488	23.6406124	21.1380053	21.3864634	8.2255195	15.4150928
3100	2.5359925	2.5035407	2.2630971	23.7236207	21.2200801	21.4605236	8.0418429	15.6817778
3200	2.5464129	2.5047127	2.2717830	23.8042943	21.2995818	21.5325112	7.8699430	15.9343513
3300	2.5588248	2.5061591	2.2802879	23.8828366	21.3766775	21.6025486	7.7088068	16.1740298
3400	2.5733955	2.5079169	2.2886890	23.9594359	21.4515193	21.6707470	7.5575455	16.4018905
3500	2.5902756	2.5100232	2.2970589	24.0342700	21.5242469	21.7372110	7.4153767	16.6188934
3600	2.6095974	2.5125150	2.3054664	24.1075051	21.5949903	21.8020387	7.2816086	16.8258965
3700	2.6314733	2.5154286	2.3139759	24.1792979	21.6638694	21.8653219	7.1556278	17.0236700
3800	2.6559945	2.5187992	2.3226478	24.2497945	21.7309954	21.9271467	7.0368879	17.2129066
3900	2.6832312	2.5226603	2.3315385	24.3191314	21.7964711	21.9875927	6.9249005	17.3942308
4000	2.7132329	2.5270438	2.3407030	24.3874369	21.8603933	22.0467370	6.8192281	17.5682089
4100	2.7460277	2.5319792	2.3501804	24.4548316	21.9228525	22.1046512	6.7194761	17.7353556
4200	2.7816251	2.5374938	2.3600236	24.5214257	21.9839320	22.1614022	6.6252884	17.8961375
4300	2.8200166	2.5436123	2.3702693	24.5873232	22.0437109	22.2170539	6.5363419	18.0509813
4400	2.8611778	2.5503568	2.3809533	24.6526198	22.1022632	22.2716665	6.4523425	18.2002773
4500	2.9047892	2.5577315	2.3920926	24.7173891	22.1596577	22.3252964	6.3730064	18.3443828
4600	2.9512089	2.5657746	2.4037365	24.7817349	22.2159605	22.3779984	6.2981088	18.4836261
4700	3.0001490	2.5744878	2.4158973	24.8457201	22.2712324	22.4298227	6.2274106	18.6183095
4800	3.0516366	2.5838878	2.4286013	24.9094193	22.3255317	22.4808180	6.1607080	18.7487113
4900	3.1055283	2.5939796	2.4418622	24.9728916	22.3789120	22.5310292	6.0978035	18.8750880
5000	3.1617320	2.6047688	2.4556938	25.0361936	22.4314249	22.5804996	6.0385163	18.9976773
5100	3.2201526	2.6162589	2.4701069	25.0993772	22.4831183	22.6292701	5.9826779	19.1166992
5200	3.2806936	2.6284510	2.4851097	25.1624889	22.5340381	22.6773794	5.9301313	19.2323577
5300	3.3431221	2.6413358	2.5006990	25.2255619	22.5842261	22.7248628	5.8807202	19.3448417
5400	3.4075789	2.6549258	2.5168934	25.2886493	22.6337237	22.7717559	5.8343216	19.4543278
5500	3.4734036	2.6691793	2.5336566	25.3517461	22.6825669	22.8180895	5.7907679	19.5609782
5600	3.5410205	2.6841243	2.5510216	25.4149179	22.7307937	22.8638964	5.7499702	19.6649477
5700	3.6104914	2.6997646	2.5689971	25.4782031	22.7784386	22.9092059	5.7118238	19.7663794
5800	3.6814993	2.7160768	2.5875638	25.5416093	22.8255327	22.9540455	5.6762039	19.8654056
5900	3.7539653	2.7330520	2.6067172	25.6051579	22.8721061	22.9984407	5.6430075	19.9621506
6000	3.8278156	2.7506808	2.6264516	25.6688683	22.9181876	23.0424166	5.6121370	20.0567312

HZERO = 34118.745

T	CP	H-H0	H-H298	S	-(F-H0)	-(F-H298)	H	-F
100	4.9681500	496.8150	-984.4389	30.076711	2510.8560	3992.1099	34615.5596	-31607.8882
200	4.9681500	993.6300	-487.6239	33.520370	5710.4440	7191.6979	35112.3745	-28408.3005
298.15	4.9681500	1481.2539	0.0000	35.504049	9104.2786	10585.5323	35599.9985	-25014.4661
300	4.9681500	1490.4450	9.1911	35.534781	9169.9893	10651.2432	35609.1890	-24948.7549
400	4.9681500	1987.2600	506.0061	36.964029	12798.3514	14279.6053	36106.0044	-21320.3931
500	4.9681500	2484.0750	1002.8211	38.072639	16552.2446	18033.4985	36602.8193	-17566.4998
600	4.9681500	2980.8900	1499.6361	38.978440	20406.1738	21887.4275	37099.6343	-13712.5702
700	4.9681500	3477.7050	1996.4511	39.744284	24343.2935	25824.5471	37596.4492	-9775.4507
800	4.9681500	3974.5200	2493.2661	40.407688	28351.6304	29832.8840	38093.2646	-5767.1143
900	4.9681500	4471.3350	2990.0811	40.992851	32422.2312	33903.4849	38590.0796	-1696.5135
1000	4.9681500	4968.1500	3486.8960	41.516298	36548.1479	38029.4019	39086.8940	2429.4037
1100	4.9681500	5464.9650	3983.7111	41.989813	40723.8296	42205.0835	39583.7090	6605.0854
1200	4.9681500	5961.7799	4480.5260	42.422100	44944.7388	46425.9927	40080.5239	10825.9946
1300	4.9681503	6458.5950	4977.3410	42.819763	49207.0972	50688.3506	40577.3394	15088.3527
1400	4.9681514	6955.4100	5474.1561	43.187943	53507.7100	54988.9634	41074.1543	19388.9656
1500	4.9681559	7452.2253	5970.9713	43.530710	57843.8394	59325.0928	41570.9697	23725.0947
1600	4.9681695	7949.0415	6467.7875	43.851348	62213.1152	63694.3687	42067.7861	28094.3706
1700	4.9682050	8445.8599	6964.6061	44.152542	66613.4609	68094.7148	42564.6040	32494.7168
1800	4.9682876	8942.6840	7461.4301	44.436518	71043.0488	72524.3018	43061.4287	36924.3042
1900	4.9684605	9439.5204	7958.2665	44.705144	75500.2539	76981.5078	43558.2646	41381.5098
2000	4.9687921	9936.3813	8455.1274	44.960001	79983.6191	81464.8721	44055.1260	45864.8750
2100	4.9693829	10433.2875	8952.0336	45.202441	84491.8389	85973.0928	44552.0317	50373.0952
2200	4.9703715	10930.2711	9449.0171	45.433638	89023.7324	90504.9854	45049.0156	54904.9878
2300	4.9719379	11427.3809	9946.1271	45.654612	93578.2266	95059.4795	45546.1250	59459.4819
2400	4.9743056	11924.6854	10443.4315	45.866262	98154.3447	99635.5986	46043.4292	64035.6006
2500	4.9777405	12422.2776	10941.0237	46.069389	102751.1953	104232.4482	46541.0215	68632.4502
2600	4.9825449	12920.2793	11439.0254	46.264708	107367.9619	108849.2148	47039.0234	73249.2168
2700	4.9890624	13418.8440	11937.5901	46.452868	112003.8994	113485.1523	47537.5884	77885.1543
2800	4.9976501	13918.1608	12436.9067	46.634457	116658.3164	118139.5703	48036.9048	82539.5723
2900	5.0086902	14418.4557	12937.2018	46.810015	121330.5879	122811.8418	48537.1997	87211.8438
3000	5.0225703	14919.9934	13438.7394	46.980043	126020.1357	127501.3887	49038.7373	91901.3916
3100	5.0396764	15423.0774	13941.8234	47.145302	130726.4297	132207.6836	49541.8213	96607.6855
3200	5.0603844	15928.0490	14446.7948	47.305322	135448.9805	136930.2324	50046.7935	101330.2363
3300	5.0850501	16435.2861	14954.0320	47.461406	140187.3516	141668.6035	50554.0303	106068.6074
3400	5.1140058	16945.2017	15463.9479	47.613628	144941.1348	146422.3887	51063.9463	110822.3896
3500	5.1475509	17458.2402	15976.9862	47.762343	149709.9609	151191.2129	51576.9844	115591.2158
3600	5.1859485	17974.8738	16493.6196	47.907880	154493.4961	155974.7480	52093.6182	120374.7510
3700	5.2294217	18495.5991	17014.3452	48.050551	159291.4395	160772.6914	52614.3433	125172.6943
3800	5.2781516	19020.9333	17539.6792	48.190646	164103.5215	165584.7754	53139.6777	129984.7773
3900	5.3322781	19551.4092	18070.1550	48.328437	168929.4941	170410.7461	53670.1528	134810.7500
4000	5.3918991	20087.5720	18606.3181	48.464178	173769.1387	175250.3926	54206.3164	139650.3945
4100	5.4570711	20629.9741	19148.7202	48.598108	178622.2715	180103.5254	54748.7188	144503.5254
4200	5.5278123	21179.1716	19697.9177	48.730448	183488.7109	184969.9648	55297.9155	149369.9668
4300	5.6041061	21735.7214	20254.4675	48.861403	188368.3125	189849.5664	55854.4658	154249.5664
4400	5.6859042	22300.1765	20818.9226	48.991165	193260.9492	194742.2031	56418.9209	159142.2051
4500	5.7725714	22872.9482	21391.6943	49.119878	198166.5039	199647.7559	56991.6924	164047.7598
4600	5.8648193	23454.7607	21973.5068	49.247750	203084.8887	204566.1406	57573.5054	168966.1426
4700	5.9620759	24046.0295	22564.7756	49.374906	208016.0254	209497.2773	58164.7739	173897.2793
4800	6.0643953	24647.3125	23166.0586	49.501493	212959.8496	214441.1035	58766.0566	178841.1055
4900	6.1714921	25259.0681	23777.8140	49.627628	217916.3105	219397.5625	59377.8125	183797.5645
5000	6.2831835	25881.7644	24400.5105	49.753426	222885.3633	224366.6152	60000.5088	188766.6191
5100	6.3992803	26515.8513	25034.5974	49.878988	227866.9844	229348.2363	60634.5957	193748.2402
5200	6.5195912	27161.7610	25680.5071	50.004407	232861.1582	234342.4121	61280.5054	198742.4141
5300	6.6436529	27819.8110	26338.5569	50.129750	237867.8613	239349.1152	61938.5552	203749.1172
5400	6.7717451	28490.5500	27009.2961	50.255121	242887.1055	244368.3574	62609.2944	208768.3594
5500	6.9025561	29173.9429	27692.6887	50.380511	247918.8652	249400.1191	63292.6870	213800.1191
5600	7.0369283	29870.6951	28389.4412	50.506050	252963.1797	254444.4336	63989.4395	218844.4355
5700	7.1749851	30581.2649	29100.0107	50.631814	258020.0723	259501.3242	64700.0093	223901.3281
5800	7.3160963	31305.7939	29824.5400	50.757818	263089.5508	264570.8047	65424.5386	228970.8066
5900	7.4601051	32044.5806	30563.3267	50.884106	268171.6445	269652.8984	66163.3252	234052.9004
6000	7.6068648	32797.9067	31316.6528	51.010715	273266.3828	274747.6328	66916.6514	239147.6367

FFDATA MG1(G) HZERO 34118.7451 MELTPT 0. T NO. 61

T	H-H0/RT	-(F-H0)/RT	T	H-H0/RT	-(F-H0)/RT
100.000	2.50000000	12.63476384	200.000	2.50000000	14.36763179
298.150	2.50000000	15.36583030	300.000	2.50000000	15.38129461
400.000	2.50000000	16.10049963	500.000	2.50000000	16.65835857
600.000	2.50000000	17.11416245	700.000	2.50000000	17.49953914
800.000	2.50000000	17.83336759	900.000	2.50000000	18.12782502
1000.000	2.50000000	18.39122629	1100.000	2.50000000	18.62950182
1200.000	2.50000000	18.84703040	1300.000	2.50000000	19.04713702
1400.000	2.50000003	19.23240709	1500.000	2.50000012	19.40488911
1600.000	2.50000048	19.56623554	1700.000	2.50000149	19.71779704
1800.000	2.50000393	19.86069298	1900.000	2.50000942	19.99586177
2000.000	2.50002050	20.12409544	2100.000	2.50004137	20.24607229
2200.000	2.50007805	20.36237502	2300.000	2.50013918	20.47350931
2400.000	2.50023600	20.57991624	2500.000	2.50038299	20.68198347
2600.000	2.50059792	20.78005409	2700.000	2.50090188	20.87443304
2800.000	2.50131929	20.96539211	2900.000	2.50187764	21.05317593
3000.000	2.50260720	21.13800526	3100.000	2.50354069	21.22008014
3200.000	2.50471267	21.29958177	3300.000	2.50615913	21.37667751
3400.000	2.50791690	21.45151925	3500.000	2.51002318	21.52424693
3600.000	2.51251498	21.59499025	3700.000	2.51542860	21.66386938
3800.000	2.51879916	21.73099542	3900.000	2.52266029	21.79647112
4000.000	2.52704379	21.86039329	4100.000	2.53197923	21.92285252
4200.000	2.53749382	21.98393202	4300.000	2.54361233	22.04371095
4400.000	2.55035678	22.10226321	4500.000	2.55773148	22.15965772
4600.000	2.56577456	22.21596050	4700.000	2.57448778	22.27123237
4800.000	2.58388779	22.32553172	4900.000	2.59397960	22.37891197
5000.000	2.60476884	22.43142486	5100.000	2.61625889	22.48311830
5200.000	2.62845105	22.53403807	5300.000	2.64133582	22.58422613
5400.000	2.65492579	22.63372374	5500.000	2.66917932	22.68256688
5600.000	2.68412426	22.73079371	5700.000	2.69976461	22.77843857
5800.000	2.71607679	22.82553267	5900.000	2.73305199	22.87210608
6000.000	2.75068077	22.91818762			

MG1(S)			ASINOH		T	298.15000		
LOGK								
METHOD	READIN		MELTPT	923.				
DATA	T	100.	CP	3.7530000	H-HO/T	1.5290000	S	2.2630000
DATA	T	200.	CP	5.4180000	H-HO	630.90000	S	5.5110000
DATA	T	298.15000	CP	5.9290000	H-HO	1190.30000	S	7.7800000
DATA	T	300.	CP	5.9370000	H-HO	1201.30000	S	7.8170000
DATA	T	400.	CP	6.2410000	H-HO	1811.30000	S	9.5690000
DATA	T	500.	CP	6.4930000	H-HO	2447.70001	S	10.9890000
DATA	T	600.	CP/R	3.4047000	H-HORT	2.6080000	-FHORT	3.5286000
DATA	T	700.	CP	7.0840000	H-HO	3802.	S/R	6.6730000
DATA	T	800.	CP	7.4260000	H-HO	4527.29999	-FHO/T	8.5698750
DATA	T	900.	CP	7.7920000	H-HO	5288.	S	15.1250000
DATA	T	923.	CP	7.8800000	H-HO	5468.20001	S	15.3220000
TEMP	T	923.	T	1000.	I	100.	T	2500.
METHOD	COEF		DELTAH	2140.				
DATA	T	923.	T	6000.	C1	8.	E1	
FINISH								

MG1(S)

HZERO = -1190.300

T	CP/R	(H-HO)/RT	(H-H298)/RT	S/R	-(F-HO)/RT	-(F-H298)/RT	H/RT	-F/RT
100	1.8885299	0.7694011	-5.2202529	1.1387539	0.3693528	6.3590068	-5.2202529	6.3590068
200	2.7263669	1.5873615	-1.4074655	2.7731650	1.1858036	4.1806306	-1.4074655	4.1806306
298.15	2.9835049	2.0089398	0.0000000	3.9149381	1.9059983	3.9149381	0.0000000	3.9149381
300	2.9875306	2.0150023	0.0184509	3.9335567	1.9185545	3.9151058	0.0184509	3.9151058
400	3.1405050	2.2786390	0.7812264	4.8151726	2.5365327	4.0339462	0.7812264	4.0339462
500	3.2673128	2.4633918	1.2654610	5.5297243	3.0663325	4.2642633	1.2654610	4.2642633
600	3.4047000	2.6080000	1.6097243	6.1366000	3.5286000	4.5268757	1.6097243	4.5268757
700	3.5647072	2.7331243	1.8774595	6.6730000	3.9398757	4.7955405	1.8774595	4.7955405
800	3.7368034	2.8477024	2.0989956	7.1601098	4.3124075	5.0611142	2.0989956	5.0611142
900	3.9209766	2.9566114	2.2910943	7.6109819	4.6543705	5.3198876	2.2910943	5.3198876
923	3.9652587	2.9811786	2.3322454	7.7101134	4.7289348	5.3778680	2.3322454	5.3778680
923	4.0256433	4.1478737	3.4989404	8.8768086	4.7289349	5.3778682	3.4989404	5.3778682
1000	4.0256433	4.1384620	3.5394966	9.1993675	5.0609055	5.6598709	3.5394966	5.6598709
1100	4.0256433	4.1282057	3.5836917	9.5830524	5.4548467	5.9993607	3.5836917	5.9993607
1200	4.0256433	4.1196589	3.6205210	9.9333293	5.8136705	6.3128083	3.6205210	6.3128083
1300	4.0256433	4.1124269	3.6516843	10.2555525	6.1431256	6.6038682	3.6516843	6.6038682
1400	4.0256433	4.1062281	3.6783956	10.5538850	6.4476569	6.8754894	3.6783956	6.8754894
1500	4.0256433	4.1008558	3.7015455	10.8316255	6.7307697	7.1300799	3.7015455	7.1300799
1600	4.0256433	4.0961550	3.7218016	11.0914347	6.9952797	7.3696331	3.7218016	7.3696331
1700	4.0256433	4.0920072	3.7396746	11.3354876	7.2434804	7.5958129	3.7396746	7.5958129
1800	4.0256433	4.0883204	3.7555618	11.5655870	7.4772667	7.8100252	3.7555618	7.8100252
1900	4.0256433	4.0850216	3.7697766	11.7832425	7.6982209	8.0134658	3.7697766	8.0134658
2000	4.0256433	4.0820526	3.7825699	11.9897308	7.9076782	8.2071608	3.7825699	8.2071608
2100	4.0256433	4.0793665	3.7941449	12.1861426	8.1067761	8.3919977	3.7941449	8.3919977
2200	4.0256433	4.0769245	3.8046675	12.3734156	8.2964911	8.5687481	3.8046675	8.5687481
2300	4.0256433	4.0746949	3.8142752	12.5523626	8.4776677	8.7380874	3.8142752	8.7380874
2400	4.0256433	4.0726511	3.8230821	12.7236925	8.6510415	8.9006103	3.8230821	8.9006103
2500	4.0256433	4.0707708	3.8311846	12.8880273	8.8172566	9.0568427	3.8311846	9.0568427

HZERO = -1190.300

T	CP	H-HO	H-H298	S	-(F-HO)	-(F-H298)	H	-F
100	3.7530000	152.9000	-1037.4000	2.263000	73.4000	1263.7000	-1037.4000	1263.7000
200	5.4179999	630.9000	-559.4000	5.511000	471.3000	1661.6000	-559.4000	1661.6000
298.15	5.9290000	1190.3000	0.0000	7.780000	1129.3070	2319.6070	0.0000	2319.6070
300	5.9369999	1201.3000	11.0000	7.817000	1143.8000	2334.0999	11.0000	2334.0999
400	6.2409999	1811.3000	621.0000	9.569000	2016.3000	3206.5999	621.0000	3206.5999
500	6.4929999	2447.7000	1257.4000	10.989000	3046.7999	4237.0999	1257.4000	4237.0999
600	6.7660241	3109.6644	1919.3645	12.195020	4207.3473	5397.6473	1919.3645	5397.6473
700	7.0839999	3802.0000	2611.7000	13.260986	5480.6900	6670.9900	2611.7000	6670.9900
800	7.4259999	4527.2999	3337.0000	14.229000	6855.8999	8046.1998	3337.0000	8046.1998
900	7.7919999	5287.9999	4097.7000	15.125000	8324.4999	9514.7998	4097.7000	9514.7998
923	7.8799999	5468.2000	4277.9000	15.322000	8674.0059	9864.3058	4277.9000	9864.3058
923	7.9999999	7608.1998	6417.8998	17.640527	8674.0061	9864.3060	6417.8998	9864.3060
1000	7.9999999	8224.2000	7033.9000	18.281535	10057.3350	11247.6349	7033.9000	11247.6349
1100	7.9999999	9024.1998	7833.8998	19.044017	11924.2184	13114.5183	7833.8998	13114.5183
1200	7.9999999	9824.1998	8633.8998	19.740108	13863.9294	15054.2294	8633.8998	15054.2294
1300	7.9999999	10624.2000	9433.8999	20.380449	15870.3839	17060.6838	9433.8999	17060.6838
1400	7.9999999	11424.1997	10233.8997	20.973313	17938.4387	19128.7385	10233.8997	19128.7385
1500	7.9999999	12224.1998	11033.8998	21.525256	20063.6838	21253.9836	11033.8998	21253.9836
1600	7.9999999	13024.1998	11833.8999	22.041564	22242.3032	23432.6030	11833.8999	23432.6030
1700	7.9999999	13824.1997	12633.8998	22.526561	24470.9539	25661.2534	12633.8998	25661.2534
1800	7.9999999	14624.2000	13433.8999	22.983828	26746.6912	27936.9912	13433.8999	27936.9912
1900	7.9999999	15424.1997	14233.8997	23.416366	29066.8960	30257.1956	14233.8997	30257.1956
2000	7.9999999	16224.1998	15033.8998	23.826712	31429.2249	32619.5247	15033.8998	32619.5247
2100	7.9999999	17024.1997	15833.8998	24.217034	33831.5703	35021.8701	15833.8998	35021.8701
2200	7.9999999	17824.1997	16633.8997	24.589194	36272.0269	37462.3267	16633.8997	37462.3267
2300	7.9999999	18624.1997	17433.8997	24.944808	38748.8579	39939.1577	17433.8997	39939.1577
2400	7.9999999	19424.1995	18233.8994	25.285285	41260.4844	42450.7837	18233.8994	42450.7837
2500	7.9999999	20224.1997	19033.8997	25.611861	43805.4526	44995.7524	19033.8997	44995.7524

MG1(S)

H/FRO = -1190.300

T	CP/R	(H-HO)/RT	S/R	-(F-HO)/RT	H/RT	-F/RT	REFERENCE ELEMENTS DELTA H/RT	REFERENCE ELEMENTS -DELTA F/RT	GASEOUS ATOMS DELTA H/RT	GASEOUS ATOMS -DELTA F/RT
100	1.9885	0.7694	1.1388	0.3694	-5.2203	6.3590	0	0	-179.4076	165.4116
200	2.7264	1.5874	2.7732	1.1858	-1.4075	4.1806	0	0	-89.7512	75.6567
298.15	2.9835	2.0089	3.9149	1.9060	0.0000	3.9149	0	0	-60.0842	46.1333
300	2.9875	2.0150	3.9336	1.9186	0.0185	3.9151	0	0	-59.7107	45.7629
400	3.1405	2.2786	4.8152	2.5365	0.7812	4.0339	0	0	-44.6406	30.8553
500	3.2673	2.4634	5.5297	3.0663	1.2655	4.2643	0	0	-35.5720	21.9434
600	3.4047	2.6080	6.1366	3.5286	1.6097	4.5269	0	0	-29.5048	16.0273
700	3.5647	2.7331	6.6730	3.9399	1.8775	4.7955	0	0	-25.1493	11.8228
800	3.7368	2.8477	7.1601	4.3124	2.0990	5.0611	0	0	-21.8619	8.6887
900	3.9210	2.9566	7.6110	4.6544	2.2911	5.3199	0	0	-19.2853	6.2684
923	3.9653	2.9812	7.7101	4.7289	2.3322	5.3779	0	0	-18.7688	5.7879
923	4.0256	4.1479	8.8768	4.7289	3.4989	5.3779	0	0	-17.6021	5.7879
1000	4.0256	4.1385	9.1994	5.0609	3.5395	5.6599	0	0	-16.1292	4.4374
1100	4.0256	4.1282	9.5831	5.4548	3.5837	5.9994	0	0	-14.5243	2.9778
1200	4.0256	4.1197	9.9333	5.8137	3.6205	6.3128	0	0	-13.1868	1.7731
1300	4.0256	4.1124	10.2556	6.1431	3.6517	6.6039	0	0	-12.0550	0.7635
1400	4.0256	4.1062	10.5539	6.4477	3.6784	6.8755	0	0	-11.0850	-0.0935
1500	4.0256	4.1009	10.8316	6.7308	3.7015	7.1301	0	0	-10.2443	-0.8290
1600	4.0256	4.0962	11.0914	6.9953	3.7218	7.3696	0	0	-9.5087	-1.4661
1700	4.0256	4.0920	11.3355	7.2435	3.7397	7.5958	0	0	-8.8596	-2.0227
1800	4.0256	4.0883	11.5656	7.4773	3.7556	7.8100	0	0	-8.2826	-2.5125
1900	4.0256	4.0850	11.7832	7.6982	3.7698	8.0135	0	0	-7.7664	-2.9462
2000	4.0256	4.0821	11.9897	7.9077	3.7826	8.2072	0	0	-7.3018	-3.3326
2100	4.0256	4.0794	12.1861	8.1068	3.7941	8.3920	0	0	-6.8815	-3.6785
2200	4.0256	4.0769	12.3734	8.2965	3.8047	8.5687	0	0	-6.4994	-3.9897
2300	4.0256	4.0747	12.5524	8.4777	3.8143	8.7381	0	0	-6.1505	-4.2708
2400	4.0256	4.0727	12.7237	8.6510	3.8231	8.9006	0	0	-5.8308	-4.5257
2500	4.0256	4.0708	12.8880	8.8173	3.8312	9.0568	0	0	-5.5367	-4.7574

MG1(S)

T	CP	H-HO	S	-(F-HO)	H	REFERENCE ELEMENTS DELTA H	REFERENCE ELEMENTS LOG K	GASEOUS ATOMS DELTA H	GASEOUS ATOMS LOG K
0	------	0	0	0	-1190.3	0	-------	-35309.0	-------
100	3.7530	152.9	2.2630	73.4	-1037.4	0	0	-35653.0	71.8374
200	5.4180	630.9	5.5110	471.3	-559.4	0	0	-35671.8	32.8573
298.15	5.9290	1190.3	7.7800	1129.3	0.0	0	0	-35600.0	20.0355
300	5.9370	1201.3	7.8170	1143.8	11.0	0	0	-35598.2	19.8746
400	6.2410	1811.3	9.5690	2016.3	621.0	0	0	-35485.0	13.4003
500	6.4930	2447.7	10.9890	3046.8	1257.4	0	0	-35345.4	9.5299
600	6.7660	3109.7	12.1950	4207.3	1919.4	0	0	-35180.3	6.9606
700	7.0840	3802.0	13.2610	5480.7	2611.7	0	0	-34984.7	5.1346
800	7.4760	4527.3	14.2290	6855.9	3337.0	0	0	-34756.3	3.7734
900	7.7920	5288.0	15.1250	8324.5	4097.7	0	0	-34492.4	2.7223
923	7.8800	5468.2	15.3220	8674.0	4277.9	0	0	-34426.4	2.5137
923	8.0000	7608.2	17.6405	8674.0	6417.9	0	0	-32286.4	2.5137
1000	8.0000	8224.2	18.2815	10057.3	7033.9	0	0	-32053.0	1.9271
1100	8.0000	9024.2	19.0440	11924.2	7833.9	0	0	-31749.8	1.2932
1200	8.0000	9824.2	19.7401	13863.9	8633.9	0	0	-31446.6	0.7700
1300	8.0000	10624.2	20.3804	15870.4	9433.9	0	0	-31143.4	0.3316
1400	8.0000	11424.2	20.9733	17938.4	10233.9	0	0	-30840.3	-0.0406
1500	8.0000	12224.2	21.5253	20063.7	11033.9	0	0	-30537.1	-0.3600
1600	8.0000	13024.2	22.0416	22242.3	11833.9	0	0	-30233.9	-0.6367
1700	8.0000	13824.2	22.5256	24471.0	12633.9	0	0	-29930.7	-0.8785
1800	8.0000	14624.2	22.9838	26746.7	13433.9	0	0	-29627.5	-1.0912
1900	8.0000	15424.2	23.4164	29066.9	14233.9	0	0	-29324.4	-1.2795
2000	8.0000	16224.2	23.8267	31429.2	15033.9	0	0	-29021.2	-1.4473
2100	8.0000	17024.2	24.2170	33831.6	15833.9	0	0	-28718.1	-1.5975
2200	8.0000	17824.2	24.5892	36272.0	16633.9	0	0	-28415.1	-1.7327
2300	8.0000	18624.2	24.9448	38748.9	17433.9	0	0	-28112.2	-1.8548
2400	8.0000	19424.2	25.2853	41260.5	18233.9	0	0	-27809.5	-1.9655
2500	8.0000	20224.2	25.6119	43805.5	19033.9	0	0	-27507.1	-2.0662

MG1O1(G)			DISSOC	90.	KCAL				
TEMP	T	298.15000	T	1000.	I	1000.	T	6000.	
REFNCE	SPECTR	23.	DISSOC	319.					
LOGK									
METHOD	PANOK								
MOLECULAR WT.= 40.32000									
DATA	WE	782.99000	WEXE	5.1500300	BE	0.57130000	ALPHA1	0.00500000	1
DATA	DE	0.12100000E-05	BETA1	0.20000000E-07					1
DATA	STATWT	2.	TO	3503.28000	WE	662.69000	WEXE	3.8900000	2
DATA	BE	0.50290000	ALPHA1	0.00460000	DE	0.11720000E-05	BETA1	-0.50000000E-07	2
DATA	STATWT	1.	TO	20303.570	WE	821.91000	WEXE	4.7400000	3
DATA	BE	0.57910000	ALPHA1	0.00449999	DE	0.11300000E-05	BETA1	0.25000000E-07	3
FINISH									

HZERO = 2067.896

T	CP/R	(H-H0)/RT	(H-H298)/RT	S/R	-(F-H0)/RT	-(F-H298)/RT	H/RT	-F/RT
298.15	3.8583739	3.5934587	0.0000000	25.6426661	22.0492074	25.6426661	7.0835684	18.5590975
1000	4.8608449	4.1469393	3.0755496	30.8331439	26.6862047	27.7575943	5.1875155	25.6456285
2000	5.5183770	4.7383821	4.2026872	34.5002456	29.7618637	30.2975585	5.2586702	29.2415755
3000	5.3024821	4.9661335	4.6090036	36.7016397	31.7355065	32.0926361	5.3129922	31.3886478
4000	5.1362751	5.0267534	4.7589059	38.2012162	33.1744628	33.4423099	5.2868974	32.9143186
5000	5.0775057	5.0414273	4.8271493	39.3395185	34.2980914	34.5123692	5.2495425	34.0899758
6000	5.0772868	5.0468796	4.8683147	40.2646432	35.2177639	35.3963284	5.2203090	35.0443344

HZERO = 2067.896

T	CP	H-H0	H-H298	S	-(F-H0)	-(F-H298)	H	-F
298.15	7.6675920	2129.1299	0.0000	50.958644	13064.1901	15193.3199	4197.0254	10996.2944
1000	9.6597626	8241.0466	6111.9167	61.273473	53032.4268	55161.5566	10308.9420	50964.5313
2000	10.9664499	18832.7942	16703.6641	68.560958	118289.1221	120418.2520	20900.6895	116221.2256
3000	10.5374106	29606.9954	27477.8652	72.935699	189200.1055	191329.2344	31674.8904	187132.2109
4000	10.2071140	39957.8638	37828.7334	75.915749	263705.1289	265834.2578	42025.7588	261637.2324
5000	10.0903239	50093.1328	47964.0029	78.177851	340796.1211	342925.2500	52161.0283	338728.2227
6000	10.0898889	60176.7720	58047.6416	80.016315	419921.1172	422050.2422	62244.6665	417853.2188

MG1O1(G)

HZERO = 2067.896

	T	CP/R	(H-H0)/RT	S/R	-(F-H0)/RT	H/RT	-F/RT	REFERENCE ELEMENTS DELTA H/RT	REFERENCE ELEMENTS -DELTA F/RT	GASEOUS ATOMS DELTA H/RT	GASEOUS ATOMS -DELTA F/RT
	298.15	3.8584	3.5935	25.6427	22.0492	7.0836	18.5591	7.0836	2.3139	-153.5179	141.9372
*	1000	4.8608	4.1469	30.8331	26.6862	5.1875	25.6456	0.2826	6.7096	-46.2377	33.7272
	2000	5.5184	4.7384	34.5002	29.7619	5.2587	29.2416	-0.3038	6.6584	-22.9580	10.6428
	3000	5.3025	4.9661	36.7016	31.7355	5.3130	31.3886			-15.1704	3.0230
	4000	5.1363	5.0268	38.2012	33.1745	5.2869	32.9143			-11.3602	-0.7643
	5000	5.0775	5.0414	39.3395	34.2981	5.2495	34.0900			-9.1696	-3.0434
	6000	5.0773	5.0469	40.2646	35.2178	5.2203	35.0443			-7.8175	-4.5862

*A CHANGE IN PHASE OF AN ASSIGNED REFERENCE ELEMENT HAS OCCURRED BETWEEN THIS TEMPERATURE AND THE PRECEDING ONE,
MG1(S)-- 923.000

MG1O1(G)

	T	CP	H-H0	S	-(F-H0)	H	REFERENCE ELEMENTS DELTA H	REFERENCE ELEMENTS LOG K	GASEOUS ATOMS DELTA H	GASEOUS ATOMS LOG K
	0	------	0	-------	0	2067.9	4295.6	-------	-90000.0	-------
	298.15	7.6676	2129.1	50.9586	13064.2	4197.0	4197.0	1.0049	-90959.6	61.6425
*	1000	9.6598	8241.0	61.2735	53037.4	10308.9	561.6	2.9139	-91886.4	14.6475
	2000	10.9664	18832.8	68.5610	118289.1	20900.7	-1207.5	2.8917	-91247.0	4.6221
	3000	10.5374	29607.0	72.9357	189200.1	31674.9			-90442.5	1.3129
	4000	10.2071	39957.9	75.9157	263705.1	42025.8			-90302.7	-0.3319
	5000	10.0903	50093.1	78.1779	340796.1	52161.0			-91111.6	-1.3217
	6000	10.0899	60176.8	80.0163	419921.1	62244.7			-93212.5	-1.9918

*A CHANGE IN PHASE OF AN ASSIGNED REFERENCE ELEMENT HAS OCCURRED BETWEEN THIS TEMPERATURE AND THE PRECEDING ONE,
MG1(S)-- 923.000

REFERENCES

1. Rossini, Frederick, et al.: Selected Values of Chemical Thermodynamic Properties. Cir. 500, NBS, Feb. 1, 1952.

2. Anon.: JANAF Thermochemical Tables. The Dow Chemical Co., Midland, Mich., Dec. 31, 1960 to Mar. 31, 1966.

3. McBride, Bonnie, J.; Heimel, Sheldon; Ehlers, Janet G.; and Gordon, Sanford: Thermodynamic Properties to 6000° K for 210 Substances Involving the First 18 Elements. NASA SP-3001, 1963.

4. Kelley, K. K.: Contributions to the Data on Theoretical Metallurgy. XIII. High-Temperature Heat Content, Heat Capacity, and Entropy Data for the Elements and Inorganic Compounds. Bull. 584, Bureau of Mines, 1960.

5. Kolsky, Harwood G.; Gilmer, Ruth M.; and Gilles, Paul W.: The Thermodynamic Properties of 54 Elements Considered as Ideal Monatomic Gases. Rep. No. LA-2110, Los Alamos Sci. Lab., Univ. Calif., Mar. 15, 1957.

6. Henderson, C. B.: and Scheffee, R. S.: Survey of Thermochemical Data. Atlantic Res. Corp., Jan. 1960.

7. Hildenbrand, D. L.: Study of Metal Additives for Solid Propellants - Tables of Thermodynamic Data for Performance Calculations. Vol. 2. Pub. C-623, Aeronutronic Div., Philco Corp., Sept. 30, 1959.

8. Hultgren, Ralph; Orr, Raymond L.; Anderson, Philip D.; and Kelley, Kenneth K.: Selected Values of Thermodynamic Properties of Metals and Alloys. John Wiley and Sons, Inc., 1963.

9. Gordon, John S.: Thermodynamic Data for Combustion Products. Rep. No. RMD 210-E3, Thiokol Chem. Corp., Jan. 1960.

10. Hilsenrath, Joseph, et al.: Tables of Thermal Properties of Gases. Cir. 564, NBS, Nov. 1955.

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

12. Kubaschewski, O.; and Evans, E. Ll.: Metallurgical Thermochemistry. Third ed., Pergamon Press, 1958.

13. McChesney M.: Equilibrium Shock-Wave Calculations in Inert Gas, Multiply Ionized Debye-Huckel Plasmas. Can. J. Phys., vol. 42, no. 12, Dec. 1964, pp. 2473-2494.

14. Zemansky, Mark W.: Heat and Thermodynamics. Fourth ed. McGraw-Hill Book Co., Inc., 1957, p. 438.

15. Huff, Vearl N.: Gordon, Sanford; and Morrell, Virginia E.: General Method and Thermodynamic Tables for Computation of Equilibrium Composition and Temperature of Chemical Reactions. NACA TR-1037, 1951. (Supersedes NACA TN's 2113 and 2161.)

16. Myers, H.; Buss, J. H.; and Benson, S. W.: Thermodynamic Effects of Coulombic Interactions in Ionized Gases. Planetary Spce Sci., vol. 3, 1961, pp. 257-270.

17. Herzberg, Gerhard (J. W. T. Spinks, trans.): Atomic Spectra and Atomic Structure. Second ed. Dover Publications, 1944.

18. Moore, Charlotte E.: Atomic Energy Levels. Vol. 1. Cir. 467, NBS, June 15, 1949.

19. Moore, Charlotte E.: Atomic Energy Levels. Vol. 2. Cir. 467, NBS, Aug. 15, 1952.

20. Moore, Charlotte E.: Atomic Energy Levels. Vol. 3. Cir. 467, NBS, May 1, 1958.

21. Kuhn, Heinrich: Atomic Spectra. Academic Press, 1962.

22. Herzberg, Gerhard: Molecular Spectra and Molecular Structure. Vol. 1, Second ed. D. Van Nostrand Co., Inc., 1950.

23. Herzberg, Gerhard: Molecular Spectra and Molecular Structure. Vol. 2. D. Van Nostrand Co., Inc., 1945.

24. Mayer, Joseph Edward; and Mayer, Maria Goeppert: Statistical Mechanics. John Wiley and Sons, Inc., 1940.

25. Kassel, Louis S.: The Calculation of Thermodynamic Functions From Spectroscopic Data. Chem. Rev., vol. 18, no. 2, Apr. 1936, pp. 277-313.

26. Wilson, E. Bright, Jr.: The Effect of Rotational Distortion on the Thermodynamic Properties of Water and Other Polyatomic Molecules. J. Chem. Phys., vol. 4, no. 8, Aug. 1936, pp. 526-528.

27. Stripp, Kenneth F.; and Kirkwood, John G.: Asymptotic Expansion of the Partition Function of the Asymmetric Top. J. Chem. Phys., vol. 19, no. 9, Sept. 1951, pp. 1131-1133.

28. Woolley, Harold W.: Effect of Darling-Dennison and Fermi Resonance on Thermodynamic Functions. J. Res. Nat. Bur. Standards, vol. 54, no. 5, May 1955, pp. 299-308.

29. Pennington, R. E.; and Kobe, K. A.: Contributions of Vibrational Anharmonicity and Rotation-Vibration Interaction to Thermodynamic Functions. J. Chem. Phys., vol. 22, no. 8, Aug. 1954, pp. 1442-1447.

30. Woolley, Harold W.: Calculation of Thermodynamic Functions for Polyatomic Molecules. J. Res. Nat. Bur. Standards, vol. 56, no. 2, Feb. 1956, pp. 105-110.

31. Woolley, Harold W.: The Calculation of Thermodynamic Functions for Asymmetric Rotator Molecules and Other Polyatomic Molecules. Ph. D. Thesis, Univ. of Michigan, 1955.

32. Zeleznik, Frank J.; and Gordon Sanford: Simultaneous Least-Squares Approximation of a Function and Its First Integrals with Application to Thermodynamic Data. NASA TN D-767, 1961.

33. Gordon, Sanford: and Zeleznik, Frank J.: A General IBM 704 or 7090 Computer Program for Computation of Chemical Equilibrium Compositions, Rocket Performance, and Chapman-Jouguet Detonations. Supplement I - Assigned Area-Ratio Performance. NASA TN D-1737, 1963.

34. Berry, R. Stephen; and Reimann, Curt W.: Absorption Spectrum of Gaseous F^- and Electron Affinities of the Halogen Atoms. J. Chem. Phys., vol. 38, no. 7, Apr. 1, 1963, pp. 1540-1543.

35. Anon.: New Values for the Physical Constants. NBS Tech. News Bul., vol. 47, no. 10, Oct. 1963, pp. 175-177.

TABLE I. - SOME TERMS IN

Formula number	Method				Subscript in equation (9)	$\ln Q^m$ terms	$T\frac{d(\ln Q^m)}{dT}$ terms
	RRHO[a]	PANDK[b] and JANAF[c]	NRRAO1[d]	NRRAO2[e]			
1	Yes	Yes	Yes	Yes	e	$\ln g_m - \frac{c_2 T_0}{T}$	$\frac{c_2 T_0}{T}$
2	Yes	Yes	Yes	Yes	V	$\sum_{i=1}^{n} d_i \ln (s_i)$	$\sum_{i=1}^{n} d_i u_i r_i s_i$
3	Yes	Yes	Yes	Yes	R	For diatomic and linear molecules, $-\ln \frac{c_2 B_0 \sigma}{T}$	1
4	Yes	Yes	Yes	Yes	R	For nonlinear molecules, $\frac{1}{2} \ln \left[\frac{\pi}{\sigma^2 A_0 B_0 C_0} \left(\frac{T}{c_2} \right)^3 \right]$	$\frac{3}{2}$
5	No	Yes	Yes	Yes	ρ	ρT	ρT
6	No	Yes	Yes	Yes	θ	$\ln \left(1 + \frac{\theta_1}{T} + \frac{\theta_2}{T^2} + \frac{\theta_3}{T^3} \right)$	$-\left(\frac{\theta_1}{T} + \frac{2\theta_2}{T_2} + \frac{3\theta_3}{T_3} \right) \frac{1}{Q_\theta}$
7	No	No	Yes	Yes	W	Triatomic linear molecules where Fermi resonance occurs $\left(\frac{c_2}{T} \right)^2 \frac{W_0^2}{2} r_W s_W^2 s_2^2 (1 - r_1)$	$\ln Q_W \left(S - \frac{r_1 u_1}{1 - r_1} \right)$

[a]Rigid-Rotator Harmonic-Oscillator approximation.
[b]Modified Pennington and Kobe method.
[c]Joint Army Navy Air Force method.

ln Q AND THEIR DERIVATIVES

$T^2 \frac{d^2(\ln Q^m)}{dT^2}$ terms	Type of molecule: Diatomic	Linear polyatomic	Nonlinear	Remarks / Definitions
$\frac{-2c_2 T_0}{T}$				$c_2 = hc/k$ g_m = statistical weight T_0 = electronic excitation energy
$\sum_{i=1}^{n} d_i u_i r_i s_i (u_i s_i - 2)$				d_i = degeneracy n = number of unique frequencies $u_i = c_2 \nu_i / T$ $r_i = e^{-u_i}$ $s_i = 1/(1 - r_i)$
	Yes	No	No	$\nu_1 = \omega_e - 2\omega_e x_e + 3.25\,\omega_e y_e + 5\omega_e z_e$
-1	Yes	Yes	Yes	σ = symmetry number
	Yes	Yes	No	$B_0 = B_e - \frac{\alpha_1}{2} + \frac{\alpha_2}{4} + \frac{\alpha_3}{8}$
-3/2	No	No	Yes	$A_0 = A_e - \frac{1}{2}\sum_{i=1}^{n} d_i \alpha_i^A$
	No	No	Yes	$B_0 = B_e - \frac{1}{2}\sum_{i=1}^{n} d_i \alpha_i^B$
	No	No	Yes	$C_0 = C_e - \frac{1}{2}\sum_{i=1}^{n} d_i \alpha_i^C$
0	Yes	Yes	No	For PANDK, $\rho = \frac{2D}{c_2 B_0^2}$ and for JANAF, $\rho = 4\left(\frac{D}{B_e}\right)^{1/2} \Big/ \nu_1 c_2$
	No	No	Yes	ρ is given
	Yes	No	No	$D = D_e + \frac{\beta_1}{2} + \frac{\beta_2}{4} + \frac{\beta_3}{8}$; if not given, $D_e = \frac{4B_e^3}{\omega_e^2}$
	No	Yes	No	$D = D_{000}$
$\left(\frac{2\theta_1}{T} + \frac{6\theta_2}{T_2} + \frac{12\theta_3}{T_3}\right)\frac{1}{Q_\theta} - \left[T\frac{d(\ln Q_\theta)}{dT}\right]^2$	Yes	Yes	No	$\theta_1 = \frac{c_2 B_0}{3}$, $\theta_2 = \frac{(c_2 B_0)^2}{15}$, $\theta_3 = \frac{4(c_2 B_0)^3}{315}$
	No	No	Yes	$\theta_1 = \frac{c_2}{12}\left[2(A_0 + B_0 + C_0) - \frac{A_0 B_0}{C_0} - \frac{A_0 C_0}{B_0} - \frac{B_0 C_0}{A_0}\right]$
	No	No	Yes	$\theta_2 = \frac{c_2^2}{480}\left[10\left(A_0^2 + B_0^2 + C_0^2\right) + 12(A_0 B_0 + A_0 C_0 + B_0 C_0) - 12\left(\frac{A_0^2 B_0 + A_0 B_0^2}{C_0} + \frac{B_0^2 C_0 + B_0 C_0^2}{A_0} + \frac{A_0^2 C_0 + A_0 C_0^2}{B_0}\right) + 7\left(\frac{A_0^2 B_0^2}{C_0^2} + \frac{A_0^2 C_0^2}{B_0^2} + \frac{B_0^2 C_0^2}{A_0^2}\right)\right]$
	No	No	Yes	$\theta_3 = 0$
$\ln Q_W \left[2u_W^2 r_W s_W (1 + r_W s_W)\right.$	No	Yes	No	W_0 = Fermi resonance constant
$+ 2u_2^2 r_2 s_2 (1 + r_2 s_2) + S^2 - 2S$	↓	↓	↓	$u_W = 2c_2 \nu_2 / T$
$\left. - 2 - \frac{r_1 u_1 (u_1 + 2S - 2)}{1 - r_1}\right]$				$r_W = e^{-u_W}$ $s_W = 1/(1 - r_W)$ $S = (1 + 2r_W s_W)u_W + 2(r_2 u_2 s_2 - 1)$

[d]Nonrigid-Rotator Anharmonic-Oscillator 1.
[e]Nonrigid-Rotator Anharmonic-Oscillator 2.

TABLE II. -

Formula number	Method			$\ln Q_c^m$ terms[e]
	PANDK[a] or JANAF[b]	NRRAO1[c]	NRRAO2[d]	
8	No	Yes	Yes	$\sum_{i=1}^{n} d_i a_i r_i s_i \left[1 + \frac{1}{2} a_i s_i + \frac{1}{6} a_i^2 s_i^2 (1 + r_i)\right]$
9	Yes	No	No	$\sum_{i=1}^{n} d_i P_i r_i s_i$
10	No	Yes	Yes	$\sum_{i=1}^{n} \left[d_i a_{ii} r_i s_i^2 (a_i s_i + a_i r_i s_i + 1) + \sum_{j=i}^{n} d_i d_j a_{ij} r_i r_j s_i s_j + \sum_{j=1}^{n} d_i d_j a_i a_{ij} (1 + \delta_{ij}) r_i r_j s_i^2 s_j \right]$
11	No	Yes	Yes	$a_{111} r_1 s_1^3 (1 + 4r_1 + r_1^2)$ (diatomics only)
12	No	Yes	Yes	$-\frac{c_2}{T} \sum_{\substack{i=1 \\ j=i}}^{n} d_i (d_j + \delta_{ij}) X_{ij} r_i r_j s_i s_j$
13	No	Yes	Yes	$-\frac{c_2}{T} \sum_{\substack{i=1 \\ j=i \\ k=j}}^{n} d_i (d_j + \delta_{ij})(d_k + \delta_{ik} + \delta_{jk}) Y_{ijk} r_i r_j r_k s_i s_j s_k$
14	Yes	No	No	$-\frac{c_2}{T} \sum_{\substack{i=1 \\ j=i}}^{n} d_i (d_j + \delta_{ij})(X_{ij} + G_i) r_i r_j s_i s_j$

[a] Rigid-Rotator Harmonic-Oscillator approximation.
[b] Modified Pennington and Kobe method.
[c] Nonrigid-Rotor Anharmonic-Oscillator 1.
[d] Nonrigid-Rotator Anharmonic-Oscillator 2.

TERMS IN $\ln Q_c^m$

Type of molecule			Remarks / Definitions
Diatomic	Linear polyatomic	Nonlinear	
			d_i = degeneracy
			$r_i = e^{-u_i}$
			$u_i = c_2\nu_i/T$
			$s_i = 1/(1 - r_i)$
			n = number of unique frequencies
Yes	No	No	$\nu_1 = \omega_e - 2\omega_e x_e + 3.25\ \omega_e y_e + 5\omega_e z_e$
Yes	No	No	$a_1 = (\alpha_1 - \alpha_2 - 0.75\ \alpha_3)/B_0$
No	Yes	No	$a_i = \frac{\alpha_i^B}{B_0} - \sum_{j=1}^{n} \frac{(1 + \delta_{ij})\alpha_{ij}}{2B_0}$
No	No	Yes	$a_i = \frac{1}{2}\left(\frac{\alpha_i^A}{A_0} + \frac{\alpha_i^B}{B_0} + \frac{\alpha_i^C}{C_0}\right)$
Yes	Yes	No	$P_i = a_i(a_i + 1)$
Yes	No	No	For PANDK, $a_1 = (\alpha_1 - 2\alpha_2 - 3.25\ \alpha_3)/B_0$ and for JANAF, $a_1 = (\alpha_1 - \alpha_2 - 0.75\ \alpha_3)/B_e$
No	No	Yes	$P_i = a_i\left(\frac{a_i}{2} + 1\right) + \frac{1}{4}\left[\left(\frac{\alpha_i^A}{A_0}\right)^2 + \left(\frac{\alpha_i^B}{B_0}\right)^2 + \left(\frac{\alpha_i^C}{C_0}\right)^2\right]$
Yes	No	No	$a_{11} = (-\alpha_2 - \frac{3}{2}\alpha_3)/B_0$
No	Yes	No	$a_{ij} = \alpha_{ij}/B_0$
No	No	Yes	$a_{ij} = 0$
Yes	Yes	Yes	$\delta_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$
Yes	No	No	$a_{111} = -\alpha_3/B_0$
Yes	No	No	For PANDK, $X_{11} = -\omega_e x_e + 4.5\ \omega_e y_e + 14.5\ \omega_e z_e$ and for JANAF, $X_{11} = (-\omega_e x_e + 4.5\ \omega_e y_e + 14.5\ \omega_e z_e)\nu_1/\omega_e$
No	Yes	Yes	$X_{ii} = x_{ii} + (1.5\ d_i + 3)y_{iii} + \sum_{\substack{k=1 \\ k\neq i}}^{n} \frac{d_k}{2} y_{iik}$
No	Yes	Yes	$X_{ij} = x_{ij} + (d_i + 1)y_{iij} + (d_j + 1)y_{ijj} + \sum_{\substack{k=1 \\ k\neq i}}^{n} \frac{d_k}{2} y_{ijk}$
Yes	No	No	$Y_{111} = \omega_e y_e + 8\omega_e z_e$
No	Yes	Yes	$Y_{ijk} = y_{ijk}$
Yes	No	No	$G_i = 0$
No	Yes	Yes	$G_i = \begin{cases} 0 & \text{if } i \neq j \\ (g_{ii} + B_0)/3 & \text{if } i = j \end{cases}$

[e]Derivatives: $T\left[d\left(\ln Q_c^m\right)/dT\right] = \sum_j \ln Q_{c_j} S_j$ and $T^2\left[d^2\left(\ln Q_c^m\right)/dT^2\right]$ $= \sum_j \ln Q_{c_j}\left[\sum_i m_i u_{h_i}^2 r_{h_i} s_{h_i}\left(r_{h_i} s_{h_i} + 1\right) - 2S_j + S_j^2 - p_j\right]$ where $\ln Q_c^m = \sum_j \ln Q_{c_j}$ and $\ln Q_{c_j}$ is any term in formulas 8 to 27 which has the formula $\ln Q_{c_j} = (c_2/T)^{p_j} C_j \prod_i r_{h_i}^{n_i} s_{h_i}^{m_i}$ where $p_j = 0, 1,$ or 2; C_j is a constant; n_i and m_i are integer exponents; and h_i is an integer subscript, and where $S_j = \sum_i u_{h_i}\left(n_i + m_i r_{h_i} s_{h_i}\right) - p_j$.

TABLE II. - Concluded.

Formula number	Method			$\ln Q_c^m$ terms[e]
	PANDK[a] or JANAF[b]	NRRAO1[c]	NRRAO2[d]	
15	No	Yes	Yes	$-\frac{24c_2}{T}\omega_e z_e r_1^4 s_1^4$ (diatomics only)
16	No	Yes	Yes	$-\frac{c_2}{T}\sum_{i=1}^{n} 2g_{ii} r_i s_i^2 (1 - 2a_i r_i s_i)$
17	No	Yes	Yes	$-\frac{c_2}{T}\sum_{\substack{i=1 \\ j=1}}^{n} d_i(d_j + \delta_{ij})(1 + \delta_{ij}) a_i X_{ij} r_i r_j s_i^2 s_j$
18	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=i}}^{n} d_i(d_j + \delta_{ij})(1 + \delta_{ij}) X_{ij}^2 r_i r_j s_i^2 s_j^2$
19	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1 \\ k=j}}^{n} \mathscr{D}_{ijk} X_{ij} X_{ik} r_i r_j r_k s_i^2 s_j s_k$
20	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=i \\ k=1}}^{n} \mathscr{D}_{ijk} X_{ij} Y_{ijk} r_i r_j r_k s_i^2 s_j^2 s_k$
21	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1 \\ k=1 \\ \ell=k}}^{n} \mathscr{D}_{ijk\ell} X_{ij} Y_{ik\ell} r_i r_j r_k r_\ell s_i^2 s_j s_k s_\ell$
22	No	No	Yes	$\left(\frac{c_2}{T}\right)^2 \sum_{i=1}^{n} g_{ii}^2 r_i s_i^4 (1 + 8r_i + r_i^2)$
23	No	No	Yes	$\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1}}^{n} 2g_{ii} X_{ij} r_i r_j s_i^3 s_j \left[1 + 7\delta_{ij} + r_i(1 + 5\delta_{ij})\right]$
24	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{i=1}^{n} 4a_i \left[X_{ii} d_i(d_i + 1)\right]^2 r_i^5 s_i^5$
25	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1}}^{n} \mathscr{D}_{ij} a_i X_{ij}^2 r_i r_j s_i^3 s_j^2$
26	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1 \\ k=1}}^{n} \mathscr{D}_{ijk} a_i X_{ij} X_{ik} r_i r_j r_k s_i^3 s_j s_k (1 + r_i)$
27	No	No	Yes	$\frac{1}{2}\left(\frac{c_2}{T}\right)^2 \sum_{\substack{i=1 \\ j=1 \\ k=1}}^{n} \mathscr{D}_{ijk} a_i X_{ij} X_{jk} r_i r_j r_k s_i^2 s_j^2 s_k$

[a] Rigid-Rotator Harmonic-Oscillator approximation.
[b] Modified Pennington and Kobe method.
[c] Nonrigid-Rotator Anharmonic-Oscillator 1.
[d] Nonrigid-Rotator Anharmonic-Oscillator 2.

TERMS IN $\ln Q_c^m$

Type of molecule			Remarks / Definitions
Diatomic	Linear polyatomic	Nonlinear	
Yes	Yes	Yes	$\mathscr{D}_{ijk} = (2 - \delta_{jk})(1 + \delta_{ij})(1 + \delta_{ik})d_i(d_j + \delta_{ij})(d_k + \delta_{ik})$
Yes	Yes	Yes	$\mathscr{D}_{ijk} = 2(1 + \delta_{ij})(1 + \delta_{ik} + \delta_{jk})(d_i + \delta_{ij})d_j(d_k + \delta_{ik} + \delta_{jk})$
Yes	Yes	Yes	$\mathscr{D}_{ijk\ell} = 2(1 + \delta_{ij})(1 + \delta_{ik} + \delta_{i\ell})d_i(d_j + \delta_{ij})(d_k + \delta_{ik})(d_\ell + \delta_{i\ell} + \delta_{k\ell})$
Yes	Yes	Yes	$\mathscr{D}_{ij} = (1 + \delta_{ij})^2 d_i(d_j + \delta_{ij})$
Yes	Yes	Yes	$\mathscr{D}_{ijk} = (1 + \delta_{ij})(1 + \delta_{ik})d_i(d_j + \delta_{ij})(d_k + \delta_{ik})$
Yes	Yes	Yes	$\mathscr{D}_{ijk} = (1 + \delta_{ij})(1 + \delta_{jk})(2 - \delta_{ik})d_i(d_j + \delta_{ij})\left[(1 + \delta_{ik})d_k + \delta_{ik} + \delta_{jk} + \delta_{ij}\delta_{jk}\right]$

[e]Derivatives: $T\left[d\left(\ln Q_c^m\right)/dT\right] = \sum_j \ln Q_{c_j} S_j$ and $T^2\left[d^2\left(\ln Q_c^m\right)/dT^2\right]$ $= \sum_j \ln Q_{c_j}\left[\sum_i m_i u_{h_i}^2 r_{h_i} s_{h_i}\left(r_{h_i} s_{h_i} + 1\right) - 2S_j + S_j^2 - p_j\right]$ where $\ln Q_c^m = \sum_j \ln Q_{c_j}$ and $\ln Q_{c_j}$ is any term in formulas 8 to 27 which has the formula $\ln Q_{c_j} = (c_2/T)^{p_j} C_j \prod_i r_{h_i}^{n_i} s_{h_i}^{mi}$ where $p_j = 0$, 1, or 2; C_j is a constant; n_i and m_i are integer exponents; and h_i is an integer subscript, and where $S_j = \sum_i u_{h_i}\left(n_i + m_i r_{h_i} s_{h_i}\right) - p_j$.

TABLE III. - BRIEF DESCRIPTION OF CONTENTS OF INPUT CARDS

Type of card	Contents	Is card optional?
	General data	
CONSTS	Physical constants, hc/k, R, and S_c (eqs. (4) and (5))	No
ATOM	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).	No
LISTEF	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	Yes
EFDATA	Chemical formula for reactant (monatomic gas or element in its reference form), the H_0^o value, the melting point if any, and the number of temperatures for which there are binary EF data following this card	Yes
Binary EF data	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).	Yes
	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 ΔH_T^o 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

Labels 2, 3, or 4		Numerical value	Comments
HF298	(a)	An assigned enthalpy $H^{o}_{298.15}$	Numerically equal to heat of formation at 298.15° K
ASINDH		An assigned enthalpy, H^{o}_{T}	----------
DISSOC		Dissociation energy $\left(D^{o}_{T} \text{ or } -\Delta H^{o}_{T}\right)$	----------
DELTAH		Heat of formation from the assigned reference elements $\left(\Delta H^{o}_{T}\right)$	----------
IPATOM[b] (ions only)		Heat of ionization from the electron and neutral atom	----------
INVCM	(a)	(blank)	Units are cm^{-1}/mole
CAL		(blank)	Units are cal/mole
KCAL		(blank)	Units are kcal/mole
EV		(blank)	Units are eV/mole
JOULES		(blank)	Units are J/mole
T		Temperature	Not required with HF298

[a]Use only one.

[b]The following are examples of IPATOM:

Cl^{+}: $Cl + IPATOM = Cl^{+} + e^{-}$

IPATOM = 104995.46 cm^{-1} (refs. 18 and 20)

Cl^{++}: $Cl + IPATOM = Cl^{++} + 2e^{-}$

IPATOM = 296995.46 cm^{-1} (refs. 18 and 20)

Cl^{-}: $Cl + e^{-} + IPATOM = Cl^{-}$

IPATOM = -3.613 eV (ref. 34).

TABLE V. - CONTENTS OF OPTIONAL SPECIFIC DATA CARDS

Card columns 1 to 6	Labels 1, 2, 3, or 4	Numerical value	Comments
REFNCE	Any alpha-numeric characters	Any numbers within the machine capabilities	----------
EFTAPE	(blank)	(blank)	Code calling for the H_0^o value and the $\frac{H_T^o - H_0^o}{RT}$ and $\frac{-(F_T^o - H_0^o)}{RT}$ data to be put on tape and punched for future log K and ΔH_T^o calculations
LOGK	(blank)	(blank)	Code calling for ΔH_T^o and log K calculations
LSTSQS	T	Temperature (^oK) at the beginning or end of interval to be fit	Card calls for a least-squares fit
	EXP	Temperature exponent	q_i values in equation (10)
	TCONST	Temperature constraint, ^oK	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^o K.
INTERM	(blank)	(blank)	Calls for intermediate output data
DATE	(any six optional characters)	(blank)	Punches the label as the last word on the binary least-squares coefficient cards
TEMP	T	Temperature, ^oK	This may be a single value or the beginning or end of an interval
	I	Temperature increment, ^oK	This must be preceded by a lower and followed by a higher T value. (See section TEMP card(s).)

TABLE VI. - CONTENTS OF METHOD CARDS

Method code (any label)	Type of species	Labels 1, 2, 3, or 4	Numerical value	Comments
READIN	All species		(blank)	Read in functions directly.
		H298H0	$H^{o}_{298.15} - H^{o}_{0}$	Used in obtaining $H^{o}_{T} - H^{o}_{0}$ values when $H^{o}_{T} - H^{o}_{298.15}$ values are given on DATA cards
		MELTPT	Melting point	Should be included when a set of specific data has both solid and liquid phases
COEF	All species		(blank)	Calculate functions from empirical equations.
		REDUCE	(blank)	Coefficients on DATA cards are those of equations (10) to (12) divided by R.
		MELTPT	Melting point	See MELTPT under READIN.
		DELTAH	Heat of transition	Used between two phases of the same species; code is on METHOD card of second phase
		DELTAS	Entropy of transition	May be used in lieu of a heat of transition (see label DELTAH)
FIXEDN	Monatomic gases		Highest principal quantum number to be included in calculations	All energy levels whose principal quantum number is less than or equal to this number will be included
		FILL	(blank)	Missing energy levels will be estimated and included as discussed in the section Inclusion of predicted levels
ALLN	Monatomic gases		(blank)	Include all levels given in input.
		FILL	(blank)	See FILL option under FIXEDN.
TEMPER	Monatomic gases		(blank)	Cut off all levels above "reduced" ionization potential (See section Internal Partition Function for Monatomic Gases.)
		FILL	(blank)	See FILL option under FIXEDN.
RRHO	Diatomic and polyatomic gases		(blank)	Rigid-rotator harmonic-oscillator approximation (See table I.)
PANDK	Diatomic and polyatomic gases	-	(blank)	Calculation method of reference 3 (See tables I and II.)
JANAF	Diatomic and polyatomic gases		(blank)	Calculation method of reference 2 (See tables I and II.)
NRRAO1	Diatomic and polyatomic gases		(blank)	Calculation method of references 30 and 31 (See tables I and II.)
NRRAO2	Diatomic and polyatomic gases		(blank)	Same as NRRHO1 with some higher order corrections (See tables I and II.)

TABLE VII. - CONTENTS OF DATA CARDS

Method	Labels 1, 2, 3, or 4	Numerical value	Comments
READIN	T	Temperature in ^{o}K	One value on each card
	CP	C_p^o	Either one of these values on each card
	CP/R	C_p^o/R	
	H-H0	$H_T^o - H_0^o$	Any one of these values on each card
	H-H2	$H_T^o - H_{298.15}^o$	
	H-H0/T	$(H_T^o - H_0^o)/T$	
	H-H2/T	$(H_T^o - H_{298.15}^o)/T$	
	H-H0RT	$(H_T^o - H_0^o)/RT$	
	H-H2RT	$(H_T^o - H_{298.15}^o)/RT$	
	S	S_T^o	Any one of these values on each card
	S/R	S_T^o/R	
	-F-H0	$-(F_T^o - H_0^o)$	
	-F-H2	$-(F_T^o - H_{298.15}^o)$	
	-FH0/T	$-(F_T^o - H_0^o)/T$	
	-FH2/T	$-(F_T^o - H_{298.15}^o)/T$	
	-FH0RT	$-(F_T^o - H_0^o)/RT$	
	-FH2RT	$-(F_T^o - H_{298.15}^o)/RT$	
COEF	See comments	----------	First card may be the same as aforementioned READIN card with C_p^o or C_p^o/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).

	T	Temperature at beginning or end of temperature range	Two T labels must precede exponents and coefficients for the temperature range.
	Ei(i = 1, 2, . . . , or 10)	q_i in equation (10)	----------
	Ci(i = 1, 2, . . . , or 10)	a_i or a_i/R in equation (10)	a_i/R with REDUCE code in METHOD card
	CH	a_{r+1} (eq. (11))	Use one if a_{r+1} has not been set by previous enthalpy value.
	CH/R	a_{r+1}/R (eq. (11))	
	CH-H0	$a_{r+1} - H_0^o$ (eq. (11))	
	CHH0/R	$(a_{r+1} - H_0^o)/R$ (eq. (11))	
	CS	a_{r+2} (eq. (12))	Use one if a_{r+2} has not been set by previous entropy value.
	CS/R	a_{r+2}/R (eq. (12))	
	TPUNCH	Temperature value to be punched on coefficient cards	Calls for cards to be punched (See appendix E)
FIXEDN, ALLN, or TEMPER[a]	IP	Ionization potential in cm^{-1}	Required only with TEMPER
	J_m value	ϵ_m/hc in cm^{-1} (eq. (7))	J_m value (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 VII. - Concluded. CONTENTS OF DATA CARDS

Method	Labels 1, 2, 3, or 4	Numerical value	Comments
RRHO, PANDK, JANAF, NRRHO1, or NRRHO2[b]	SYMNO	Symmetry number	Taken to be 1 if omitted
	STATWT	Statistical weight	Taken to be 1 if omitted
	T0	T_0	Use with excited electronic state.
	B0	B_0	B_e, B_0, or I_B value must be included for all molecules.
	BE	B_e	See comments for label B_0. Use only for linear molecules.
	WE	ω_e	Diatomics only
	WEXE	$\omega_e x_e$	
	WEYE	$\omega_e y_e$	
	WEZE	$\omega_e z_e$	
	WX4	Anharmonic constant one order higher than $\omega_e z_e$	
	ALPHAE	α_e	
	ALPHAi, ($i \leq 3$)	α_i (See comments for definition.)	Diatomics only. $B_v = B_e - \alpha_1\left(v + \frac{1}{2}\right) + \alpha_2\left(v + \frac{1}{2}\right)^2 + \alpha_3\left(v + \frac{1}{2}\right)^3$
	ALFABi ($i \leq 6$) ALFAij ($i, j \leq 6$)	α_i α_{ij}	Linear polyatomics only. $B_{[v]} = B_e - \sum_{i=1}^{n\leq 6}\left[\alpha_i\left(v_i + \frac{d_i}{2}\right) + \sum_{j\geq i}^{n\leq 6}\alpha_{ij}\left(v_i + \frac{d_i}{2}\right)\left(v_j + \frac{d_i}{2}\right)\right]$
	ALFAAi ($i \leq 6$)	α_i^A	Nonlinear molecules only. $A_{[v]} = A_e - \sum_{i=1}^{n\leq 6}\alpha_i^A\left(v_i + \frac{d_i}{2}\right)$ where v_i and d_i are the vibrational quantum number and degeneracy respectively for the i^{th} fundamental frequency
	ALFABi ($i \leq 6$)	α_i^B	Nonlinear molescules only. $B_{[v]} = B_e - \sum_{i=1}^{n\leq 6}\alpha_i^B\left(v_i + \frac{d_i}{2}\right)$

ALFACi (i ≤6)	α_i^C	Nonlinear molecules only. $C_{[v]} = C_e - \sum_{i=1}^{n\leq 6} \alpha_i^C \left(v_i + \frac{d_i}{2}\right)$
DE	D_e	Diatomics only
BETAi (i ≤ 3)	β_i	Diatomics only, where $D_v = D_e - \sum_{i=1}^{n\leq 3} \beta_i (v + 1/2)^i$
Vi(d_i) or Vi (i ≤ 20)	$\nu_i(d_i)$ or ν_i	d_i is degeneracy (an integer) of ν_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	ρ, $^oK^{-1}$	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)(cm^2)	See comments for label B_0.
IA	$I_A \times 10^{39}$, (g)(cm^2)	See comments for label A_0.
IC	$I_C \times 10^{39}$, (g)(cm^2)	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 VIII. - PUNCHED COEFFICIENT CARDS

Binary word number	Card 1		Card 2 (2 or more intervals)		Cards 3 to 5 as required (5 to 9 intervals)	
1	Machine assigned storage		Same as card 1		Same as card 1	
2	Machine assigned storage					
3	First 6 characters of formula					
4	Second 6 characters of formula					
5	Ionization potential, if any					
6	Lowest T in intervals or melting point if liquid		a_{r+2} (Second interval)		Lowest T in interval	
7	Highest T in intervals or melting point if solid		Fourth highest T		Highest T in interval	
8	Second highest T		Third highest T		a_1	
9	Highest T in interval		a_1		a_2	Next interval, if any
10	a_1	First interval	a_2	Third interval, if any	a_3	
11	a_2		a_3		a_4	
12	a_3		a_4		a_5	
13	a_4		a_5		a_{r+1}	
14	a_5		a_{r+1}		a_{r+2}	
15	a_{r+1} (eq. 11)		a_{r+2}		Lowest T in interval	
16	a_{r+2} (eq. 12)		Fifth highest T		Highest T in interval	
17	Third highest T		Fourth highest T		a_1	
18	Second highest T		a_1		a_2	Next interval
19	a_1		a_2		a_3	
20	a_2	Second interval, if any	a_3	Fourth interval, if any	a_4	
21	a_3		a_4		a_5	
22	a_4		a_5		a_{r+1}	
23	a_5		a_{r+1}		a_{r+2}	
24	a_{r+1}		a_{r+2}			

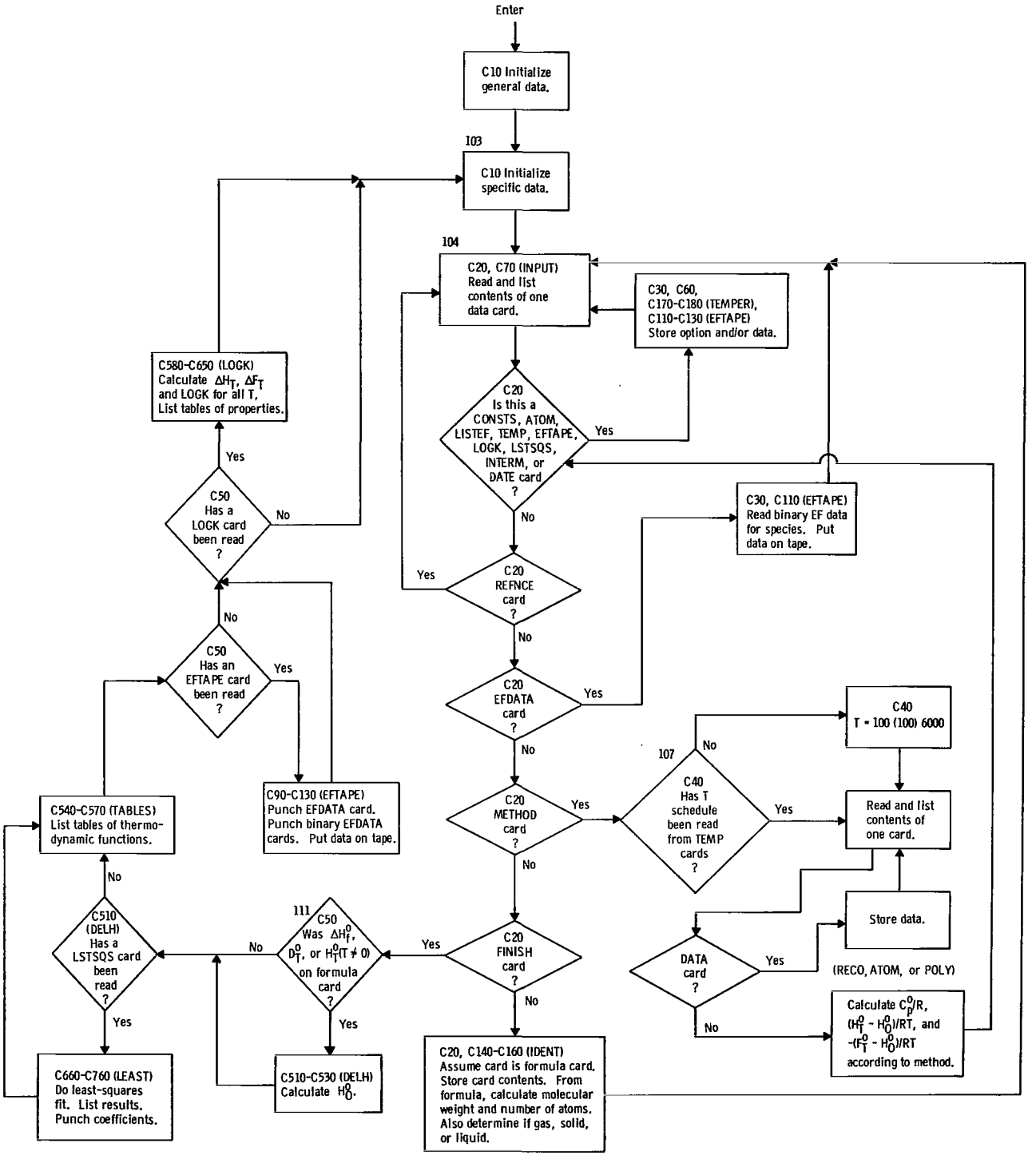


Figure 1. - General flow of program.

"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