MASTER

COO-1149-143

RECEIVED BY DTIC AUG 20 1969

THERMAL DATA FITTING WITH ORTHOGONAL
FUNCTIONS AND COMBINED TABLE GENERATION.

THE FITAB PROGRAM

BRUCE H. JUSTICE

AEC Cryogenics Project
Department of Chemistry
University of Michigan
Ann Arbor, Michigan 48104

Professor Edgar F. Westrum, Jr.
Project Director
Contract AT(11-1)-1149

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Project Report

Title: Thermal Data Fitting with Orthogonal Functions and Combined Table Generation. The FITAB Program.

Authors: Bruce H. Justice

Work performed under: AEC Cryogenics Project
Office of Research Administration
Number 05069

Submitted to
AEC by: Professor Edgar F. Westrum, Jr.
Director, AEC Cryogenics Project
Contract AT(11-1)-1149
Department of Chemistry
University of Michigan
Ann Arbor, Michigan 48104

Designation: COO-1149-143

Date: February 1969

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TABLE OF CONTENTS

	Page
Abstract	1
I Introduction	2
II General Considerations	3
Orthogonal Functions	3
Combining Data Sets	6
Thermodynamic Functions	9
III Programming Considerations	13
Curve Fitting Program With Specialized Output . .	13
Program Input and Strategy	14
Program Output	18
IV Acknowledgment	22
V Bibliography	23

APPENDICES

Appendix I: Program FITAB	25
Appendix II: Typical Output	41

ABSTRACT

A FORTRAN IV program to represent experimental thermal data with orthogonal functions and to tabulate the thermodynamic functions is described. The use of a more general algorithm permits the generation of power series which can contain positive or negative powers. Further, it is possible to start generation at any power with any term-to-term increment in the exponent. This allows the use of an equation with a leading cubic term for very low temperature heat capacities. Enthalpy data can be approximated with a leading reciprocal temperature term. Appropriate coding allows analytical generation of the thermodynamic functions.

The methods of combining thermal data are reviewed and discussed. The program combines any number of heat capacity and/or enthalpy segments (from formats logically selected at the time of execution) in order to generate a complete tabulation presented in rounded, directly publishable form.

I. INTRODUCTION

Accrual of considerable experience in the reduction of cryogenic calorimetric data via digital computer plus greater flexibility in computer hardware and software features permits significant improvements in programs reported earlier.^{1,2} The use of calculational programs (and on-line printer) to develop thermodynamic function tables which are publishable directly (or by photograph) would be economical and efficient. The programming is here developed so that the reliability of the final table depends only on the original (potentiometric) data, to avoid transcription errors, the need for proofreading, etc. Effective application of the computer has thus tended toward direct reduction of calorimetric data to its final form with decreasing intervention of intermediate manual manipulations.

Interest in tabulating both types of thermal data (heat capacity and enthalpy increment) require programs which smooth (and/or combine) data and generate the derived thermodynamic functions. Moreover, it is often desirable to have programs to interrelate different types of thermal data in order to yield a single smooth table from multiply disjoint data sets.

Although the present discussion considers only analytical functions, the variety of polynomials now employed in fitting thermal data occasions still further flexibility. Empirical and quasi-theoretical considerations derived for the temperature range of interest, the nature and phase of the material under study, and the type of data determine the nature of the best polynomial approximation. Utilization of orthogonal functions to generate different analytical forms is another feature of the improvements

reported herein.

II GENERAL CONSIDERATIONS

ORTHOGONAL FUNCTIONS

The orthogonal set suggested by Forsythe³ and the algorithm of Rudin⁴ have been modified to permit an arbitrary starting point in the sequence of powers and an arbitrary increment in this sequence. The basic orthogonal set first used^{1,2} is given by the equations:

$$\begin{aligned} P_0(X) &= 1 \\ P_1(X) &= (X - \alpha_1) P_0(X) \\ P_n(X) &= (X - \alpha_n) P_{n-1}(X) - \beta_{n-1} P_{n-2}(X). \end{aligned}$$

The α_n and β_n are given by

$$\alpha_n = \sum_{i=1}^m X_i [P_{n-1}(X_i)]^2 / \sum_{i=1}^m [P_{n-1}(X_i)]^2,$$

and

$$\beta_n = \sum_{i=1}^m [X_i P_n(X_i) P_{n-1}(X_i)] / \sum_{i=1}^m [P_{n-1}(X_i)]^2,$$

where the summations in the general case are over m arbitrarily spaced data points. To generalize this set to an arbitrary initial power and an arbitrary increment in the power sequence one can generate power series from the equations:

$$\begin{aligned}
 P_j(X) &= X^j \\
 P_{j+k}(X) &= (X^k - \alpha_{j+k}) P_j(X) \\
 P_{j+n}(X) &= (X^k - \alpha_{j+n}) P_{j+n-k}(X) - \beta_{j+n-k} P_{j+n-2k}(X)
 \end{aligned} \tag{2}$$

These polynomials $[P_j(X)]$ remain orthogonal if

$$\alpha_{j+n} = \sum_{i=1}^m X_i^k [P_{j+n-k}(X_i)]^2 / \sum_{i=1}^m [P_{j+n-k}(X_i)]^2$$

and if

$$\beta_{j+n} = \sum_{i=1}^m [X_i^k P_{j+n}(X_i) P_{j+n-k}(X_i)] / \sum_{i=1}^m [P_{j+n-k}(X_i)]^2.$$

The index i is a count of the m arbitrary data points, while j is the initial power of generation and k is the increment in this index. Thus, the polynomials may be started at any integral power (zero, positive, negative) of the argument X and may be generated such that they contain all powers, every other power (odd and even functions), etc.

The application of these arguments to thermal data is in some respects empirical, but quasi-theoretical foundations do exist. It has been suggested by Barron and Morrison⁵ and Morrison and Newsham⁶ that the heat capacity of non-conductors can be approximated by

$$C_v = aT^3 + bT^5 + cT^7 + \dots, \tag{3}$$

when $T < \theta_D/25$. These authors are careful to state that

graphical procedures (plotting) are preferred over analytical approaches. Moreover, experience in this laboratory indicates that heat capacities for the range 4° to 50°K usually can be approximated well with a series similar to Equation (3). These least squares series yield entropy and enthalpy extrapolations entirely consistent with the values derived from a C_p/T vs. T^2 plot, although no claims can be made about the physical significance of the coefficients. Of course, the scaling of the independent variable (discussed later) should be such that the heat capacity at 0°K is zero. It is to be noted that representation of the heat capacity of a conductor at low temperatures would employ a linear term in addition to those given in Equation (3).

Maier and Kelley⁷ have found that the enthalpy of most substances relative to that at 273.15° or 298.15°K* is represented well by the equation,

$$H_T^\circ - H_{TB}^\circ = a/T + b + cT + dT^2 \quad (4)$$

The algorithm suggested by Equations (1) and (2) can be used to derive the "Kelley Equation" by means of least squares orthogonal functions. The mathematical justification for reciprocal powers in the algorithm is in doubt.⁸ Further caution is to be taken when the scaling of the independent variable allows this variable to approach or pass through zero in the usable range of the approximation. The singularity at zero has unknown effects on the approximated curve. The functions developed from the subject program simulating

*Hereafter the base temperature will be symbolized by the subscript "TB".

Equation (4) yield curves equivalent to those reported by Kelley⁹ for a variety of inorganic substances. An advantage in using the orthogonal functions is that higher order terms may be employed for the onset of premelting and other thermal anomalies.

A "normal" power series for the data in the range 30° to 500°K still offers about the best approximation. Experience has shown the optimal approximation obtains in the neighborhood where the coefficients number about one fourth (or slightly more) than the point count for data in this range. However, lower temperature heat capacity data may require more; high temperature enthalpies tend to require relatively few coefficients because of their classical behavior.

COMBINING DATA SETS

The generation of a smooth heat capacity tabulation from overlapping heat capacity polynomials or from disjoint heat capacity and enthalpy polynomials remains an art. The approach used in this laboratory is elementary. The usual procedure followed in reducing low-temperature heat capacities requires the generation of two polynomials for data (with no transitions) over the range 5° to 500°K. The polynomials are joined in the region where the curvature changes sense (i.e., from positive--concave upward--to negative--concave downward). For substances without transitions or quantum effects, the first inflection point in the heat capacity occurs typically between 20° and 50°K. It is found that inclusion of points in a 10° to 20° range as data for the two curves to be overlapped results in tangential curves (with approximately equal curvature) at the midpoint of the common segment. However, more than

normal scatter in the data does affect the internal consistency of the two "tangential" segments adversely. The program averages the heat capacities of the two curves at the overlapping point and continues the tabulation of the thermodynamic functions with the upper curve. The problem of joining heat capacity and enthalpy data usually involves low-temperature heat capacities up to 300° to 350°K, while the enthalpy data commences around 400°K with 298.15°K as the base temperature. The procedure used for fitting yields the best results when the enthalpy data are approximated with the inclusion of at least two auxiliary points. A value of zero for the relative enthalpy at the base temperature (this is automatically weighted by a factor of 10 greater than any other point) and a value for the relative enthalpy close enough to the base temperature such that

$$\frac{H_T - H_{TB}}{T - TB} \doteq C_p \text{ at } (T - TB)/2$$

tend to force the approximation of the enthalpy through zero at TB with a heat capacity concordant with the low-temperature data. Depending upon the relative merits of the low-temperature heat capacities other derived values of the enthalpy can be added to the high temperature data. In the typical case, the heat capacity (smoothed) from the low temperature data and that from differentiation of the enthalpy approximation are averaged at the designated overlap temperature for tabulation and the generated enthalpy equation is used to derive the thermodynamic functions thereafter.

The methods of combining thermal data described above enable tabulations of the heat capacity which are inter-

nally consistent, except in the overlap region. Greater internal consistency is achievable as discussed below, but at the loss of the flexibility gained by using orthogonal polynomials. Shomate¹⁰ has proposed a graphical technique, which by successive approximation represents the smoothed heat capacities and enthalpy increments about the base temperature. The method is laborious and not necessarily convergent. Peck¹¹ proposes constraining Forsythe's³ polynomials to a set of exact points. This method fits a set of "exact" points to an interpolation polynomial which also is a factor of the higher order polynomial representing the data set. The two sets (the "exact" set and the data) must be disjoint. The combination of sets of like data is straight forward. Combination of heat capacity and enthalpy data require differentiation or integration of the exact polynomial yielding a new set of points similar in kind to the data set. The derived (now similar) "exact" set and the data set can now be subjected to the process described. The difficulty with this procedure lies in the fact that inconsistent data sets cannot be weighed according to the confidence levels of the data. Presumably the combination will have most of the difficulties inherent in the present program.

Perhaps the best available solution to the problem of getting smooth curves from overlapping segments are the so-called "Spline Functions". Although the mathematical literature has numerous references⁸ to this type of approximation, the efforts of Hust and McCarty¹² bear on thermodynamic applications. This technique requires prescribed relations and constraining equations between

the approximating segments. For thermal data smooth curvature, d^2C_p/dT^2 , virtually guarantees internally consistent heat capacity tables and functions. The solution is best accomplished according to these authors, by simultaneously minimizing the squared error (least squares normal equations) and satisfying the constraints with the use of Lagrangian multipliers. The difficulty with this technique is the prescription of the best analytical relationship a priori. Orthogonal polynomials allows rapid consecutive solutions to the normal equations without the considerable problems of solving the least squares matrix for each successive power. Hence, they allow a selection of the best series to be made relatively efficiently.

The present program generates several different types of power series with a very simple subroutine. Generation of the best combination curve for the several segments involved in tabulating a full heat capacity curve by orthogonal polynomials is a first step. With these initial solutions for the coefficients of the several segments the constraints may be applied. These probably will not alter the coefficients of the lower power terms significantly. Thus this approach would allow the easy functional selection of orthogonal polynomials and could generate coefficients close enough to the constrained parameters to facilitate solution of the constrained least squares matrix. Further investigations in this laboratory are planned along this line.

THERMODYNAMIC FUNCTIONS

The algorithm for generating the thermodynamic functions from heat capacity polynomials is substantially the same as that reported earlier.^{1,2} The first and second

temperature derivatives of the heat capacity are included with the thermodynamic function output (calculation of derivatives and program output discussed below). The incidence of reciprocal powers (of temperature) in a heat capacity approximation requires a somewhat more complex analytical integration procedure (iterated integration by parts) if the temperature has been linearly translated. The procedures for performing this integration have not been worked out in that the only advantage in translating the temperature is considered to be that gained from a symmetrical placement of the independent variable about zero. However, with reciprocal powers present, the discontinuity at zero is forbidding.

The generation of the thermodynamic functions from a polynomial representing the enthalpy rather than heat capacity requires a somewhat different analytical treatment of the coefficients. The enthalpy is given by the expression,

$$H_T - H_{TB} = \sum_{i=m}^n C_i X^i, \quad (5)$$

in which there are $n - m + 1$ coefficients (the lower limit of the sum, m , is the lowest power or largest negative power) for a polynomial expression in the transformed temperature, $X (= \frac{T - L}{S})$. The same difficulties arise here concerning translation of the independent variable as in heat capacity approximations. The program is coded assuming that where negative powers are generated no linear translation will be applied to temperatures. Upon differentiation of Equation (5) one obtains:

$$\begin{aligned} C_p &= \left(\frac{dH(X)}{dX} \right) \left(\frac{dX}{dT} \right) \\ &= s^{-1} \sum_{i=m}^n i C_i X^{i-1} \end{aligned} \quad (6)$$

for the heat capacity. The entropy increment is given by

$$S_T - S_{TB} = (H_T - H_{TB})/T + \int_{TB}^T \left(\frac{H_T - H_{TB}}{T^2} \right) dT. \quad (7)$$

If only positive powers are used in the enthalpy expression, the independent variable, T , may be scaled and translated. This leads to the expression:

$$\begin{aligned} I &= \int_{TB}^T \frac{H_T - H_{TB}}{T^2} dT \\ &= s^{-1} \int_{XB}^X \frac{H(X)}{X^2 + 2LX/s + (L/s)^2} dX \end{aligned} \quad (8)$$

for the integral in Equation (7) in terms of the coefficients. Equation (8) suggests synthetic division by a quadratic expression¹³ which can be performed recursively. The polynomial resulting from the division is

$$\frac{H(X)}{X^2 + 2LX/s + (L/s)^2} = \sum_{i=2}^n D_{n-i} X^i + \frac{(D_{n-1}X + D_n)}{X^2 + 2LX/s + (L/s)^2}$$

where the D_i 's are generated by the three term recursion,

$$D_i = -C_i + (2L/s)D_{i-1} - (L/s)^2 D_{i-2},$$

with the condition that

$$D_{-1} = 0$$

and

$$D_0 = C_n.$$

The value for D_n then turns out to be

$$D_n = C_0 - (L/s)^2 D_{n-2}.$$

The integral given by Equation (8) then can be evaluated in terms of the D_i 's to yield

$$I = \frac{1}{s} \left[\sum_{i=2}^n D_{n-i} X^{i-1} / (i-1) + \frac{(L/s) D_{n-1} - D_n}{X + L/s} + D_{n-1} \ln(X + L/s) \right] \Bigg|_{XB}^X.$$

The entropy increment is then calculated from Equation (7). The above derivation is general and applicable whether the data has been translated or not.

For approximations in which negative powers have been generated, the entropy is calculated on the assumption that the transformed temperature is only scaled (not translated!).

Selecting m as the most negative power of the approximation, the integral on the right side of Equation (7) becomes

$$I = \frac{1}{s} \sum_{i=-1}^m C_i X^{i-1} / (i-1) \Big|_{XB}^X \quad (9)$$

The entropy increment contributed by the negative powers is calculated from Equation (7), using Equation (9) and summed with that derived from the positive powers. The Gibbs energy function is calculated from

$$\frac{G_T^\circ - H_{TB}^\circ}{T} = \frac{H_T^\circ - H_{TB}^\circ}{T} - S_T^\circ,$$

where S_{TB}° is given as input data or derived from low temperature heat capacities. The first and second derivations of the heat capacity are generated from equations similar to Equation (6) from the enthalpy representation, and similarly from a heat capacity representation. These are given as part of the thermodynamic function output (described below, Section III).

III PROGRAMMING CONSIDERATIONS

CURVE FITTING PROGRAM WITH SPECIALIZED OUTPUT

Computer listing (IBM 360/67) of the present Fortran IV program is given in Appendix I. The statements have extra spaces for better readability where the language permits, although subscript expressions have been concatenated in general.

The main program calls on four subroutines plus several Fortran IV subroutines. The subroutine, WRIREG, prints the thermodynamic functions and heat capacity derivatives with appropriate headings and spacing. This table appears in the output for each curve segment and contains four-decimal-digit numbers. ENTHIN is a subroutine which generates the thermodynamic functions from an enthalpy approximation. This is based on the arguments discussed in the previous section. The functions derived from heat capacity approximations are generated in the main program. The subroutine, GEFMT, generates a compiled tabulation with rounding, parenthication, etc. (discussed below) in a direct access file. The subroutine compiles a single table (any number of condensed phases) for an indeterminate number of curve segments. Finally, the subroutine, ORTHO, generates the polynomials and coefficients desired from the input thermal data. This segment also prints the curve diagnostics, coefficients, and a deviation table.

PROGRAM INPUT AND STRATEGY

The input deck for the program consists of an alphabetic card, two (2) control cards, and cards containing single "T, Cp (or H)" data points. The alphabetic card is reproduced as read by the line printer and may be used to designate the output to follow.

The first control card contain five (5) 4-digit integers (format designation: 5I4) followed by five 10-character decimal numbers (format designation: 5F10.5). The mnemonic names for the variables to be punched on this card are U, NPTS, NS, NI, N, TLD, DT, TH, SL, and HL in that order. U is a control constant possessing non-zero units, tens, and hundreds digits. Table I gives the digit

assignments for the three decades versus the function selected by a particular choice.

TABLE I

Values of Input U and Function Performed

Value	Hundreds Digit Weight Function	Tens Digit Set Function	Units Digit Data Type
1	$W_i = 1$	Initiate 1st segment	T, C _p data
2	$W_i = [(X_i - A) (B - X_i)]^{-\frac{1}{2}}$	Combine with previous seg- ment	T, H data
3	$W_i = [(X_i - A) (B - X_i)]^{-\frac{1}{2}} (Y_i)^{-1}$	Null	X, Y data (no functions ge- nerated)
4	$W_i = (Y_i)^{-1}$	Null	Null

Proper precautions are taken internally in the program so that division by zero is precluded during the generation of the data weights. Hence, zero values for the dependent variable are allowed.

The remaining integer data on the first control card are characterized in the following manner. NPTS is the number of data points (one set per card, space reserved for

500 sets) to be read. NS, NI, and N control the nature of the series to be generated and are integers. NS is the lowest power (or largest negative power) of the polynomial, and N is the maximum power. NI is the increment in the degree of the terms of the polynomial and must be nonzero. The algebraic difference between N and NS must be integrally divisible by NI. The decimal numbers TLD, DT, and TH control rounded temperatures of the output tables. TLD and TH are the lowest and highest temperatures of the tabulation, while DT is the tabular increment. The output table will have TLD and TH irrespective of the value of DT, but the other tabular entries will be integral multiples of DT in the range of TLD to TH. Also 273.15° and 298.15°K are entered at the appropriate point in this range. However, a choice of zero for DT activates the program to use the following tabular increments:

$0 < T < 50$	DT =	5,
$50 \leq T < 300$	DT =	10,
$300 \leq T < 500$	DT =	25,
$500 \leq T < 1000$	DT =	50,
and $T \geq 1000$	DT =	100.

The choice of a non-zero value for DT should be an integer greater than one, although it is a decimal number on the control card. The values for SL and HL are the entropy and enthalpy increments added to the tabular values (accrued from previous curves for overlapping segments) in the range TLD to TH. The possible manipulations with TLD, TH, SL, and HL are discussed fully in the discussion of the program output.

The second control card contains four variables whose mnemonic names are SF, LT, A, and B. These are 10-character decimal numbers (format designation: 4G10.5). SF is the scale factor and must be finite. The linear translation, LT, can be zero or finite, and this choice depends on factors discussed above. A and B are the lower and upper limits of the range for the transformed independent variable. These two data numbers need only be included when the hundreds digit of U is 2 or 3 (Table I). The Tschebychev weight function is used in calculating W_i in these two cases. Generally, A should be about 10% less than the minimum value of X_i [$=(T_i - LT)/SF$]; similarly, B should be about 10% greater than the maximum X_i . The weights for the minimum and maximum values of X_i approach infinity as A and B approach these values of X_i .

The data cards have format specifications for four pieces of information. In columns 3 through 5, alphanumeric data may be placed (format designation: A3) columns 6 through 17 are reserved for the temperature (or independent variable) as a decimal number (format designation: F12.5). The heat capacity or enthalpy (or dependent variable) is punched in columns 18 through 32 as a decimal or floating number (format designation: E15.7). Column 35 is reserved for a zero punch should it be desirable to know the deviation of point from the curve without using the point in the fitting calculations. The appearance of zero in column 35 of a data card causes the weight function for the point to be set to zero. One further function of the weightless datum point should be noted. Such a point (0 in column 35) with zero for the heat capacity and a nonzero value for the temperature should be put in with heat capacity data

extrapolated to 0°K (the first data set) for which the base temperature is other than 0°K. This artifice causes correct parenthication of the extrapolated thermodynamic functions in the compiled (direct access) tabulation.

PROGRAM OUTPUT

The program generates three different sections of output at the line printer and the compiled table from a direct access file. The first and second segments are printed from ORTHO during the approximation calculations. After printing the input alpha numeric card, the error diagnostics for each power are printed with the headings: POWER, (DEV/Y)**2, DEV**2, W*DEV**2, TEST. The respective values under the column heads are the power of the polynomial (each represents the powers of previous lines), the sum of the squares of the fractional deviations,

$$\sum_{i=1}^{NPTS} \left(\frac{Y_i - F(X_i)}{Y_i} \right)^2$$

the sum of the squares of the deviations, the sum of the squares of the weighted deviations (this should reduce at the inclusion of a new term), and the inner product of the current polynomial with the lowest order polynomial divided by the square of the former. This ratio indicates the significance of the coefficients and calculations. The inner product should be zero to within the significance of the machine word length, as it is the largest off-diagonal element of the least squares matrix after solution by ORTHO. The subroutine is written using double precision. There

are 16 digits per decimal number. A value of 10^{-10} in the TEST column would indicate the calculations have about 10 significant digits.

The next section prints a heading for the coefficients. It gives the first term of the power series, the increment in the degree from term to term, and the maximum degree of the polynomial. Immediately following these data is a deviation table, so headed. Appropriate column headings for relevant data concerning the individual points are given. Included in this table (in column order) are the transformed independent variable, the independent variable as read, the dependent variable (C_p , H , Y), the deviation of the point from the smooth curve ($Y_i - f(X_i)$), the percent deviation, and the alphanumeric identification of the data point (read in columns 3 through 5 of the data card).

The thermodynamic functions are generated in the last segment from the line printer. The columns are headed approximately, but the meaning of the data depends on the input parameters. For instance, one column head is $-G/T$ but TB for the data determines which Gibbs energy is tabulated. Temperatures are tabulated to two decimal digits, while the heat capacity and thermodynamic functions are represented with values given to four decimal digits. The first and second derivatives of the heat capacity are given at the rounded temperatures of the table as five digit floating point numbers.

The program generates the output described above as long as data segments are approximated. However, simultaneously the main program and the subroutine GEFMT are recording the thermodynamic functions in a direct-access file. This file is printed upon command. The file is

started anew unless data segments are being combined. If data are being combined, certain logical features of the input data are examined to determine the character of the "page image" in the direct-access file.

This table has the temperature, heat capacity, entropy, enthalpy and Gibbs energy function. It has double spacing between five single spaced tabular lines. The temperatures are printed as integers except 273.15°K, 298.15°K and first order transition temperatures, which have two decimal digits. The digits tabulated are indicated in Table II.

TABLE II

<u>Tabular Sensitivity</u>	<u>Absolute Magnitude of Function</u>	
	<u>Cp, S, G/T</u>	<u>H</u>
10^{-3}	< 15	H < 50
10^{-2}	> 15	50 < H < 500
10^{-1}		500 < H < 5000
10^0		5000 < H < 50,000
10^1		H > 50,000

In some instances, this may involve digits which are not significant, but which have a relative significance useful for internal consistency, in interpolation, etc.

The direct access file parenthetically rounds values outside of the temperature range of the data. This has the effect of warning the reader that an extrapolation has been made. This practice is prevalent in published thermodynamic tabulations. The data for both phases at a

first order transition point can be parenthicated by putting a data point just below and just above (by about one part in 10^7) the transition temperature for the Phase I and Phase II curve segments without losing accuracy due to long extrapolation of a high degree polynomial. Third degree polynomials or less can usually be extrapolated some distance to a transition point. This makes enthalpy data relatively easy to tabulate. Heat capacity data may require two short curve segments about a first order transition.

When TLD is zero (0°K) the program writes a line in the direct access file if the magnitude of the enthalpy is greater than 10^{-4} cal/mole. This allows tabulations of heat capacity data with the nonzero base temperature. The Gibbs energy function will have the word "INFINITE" for the entry at 0°K . The program actually uses 10^{-15}°K as an approximation of 0°K .

The incidence of a "2" in the tens digit (cf. Table I) of U signals a data set to be combined with the previous curve. If TH for the previous curve is equal to TLD for the present data, and if HL and SL are zero, then the program assumes the curve segments are continuous. The result of this assumption is that the heat capacity given in the direct-access file at the present value of TLD (or TH for the previous curve) is the average of the values for the two curves being combined. The values averaged are printed in the thermodynamic tables given for each segment. If TLD and TH are equal and the ratio of HL to SL times TLD is one (to within 10^{-5}), then the program assumes an isothermal (1st order) transition lies between segments. The program rewrites the last line of the previous segment as if it were the melting point by spacing it and placing

the temperature in the line as a two-decimal digit number. The first line of the segment to be processed will be spaced four lines downward and give the data for Phase II at the transition temperature (also printed as a two-decimal digit number). In the case where TH (from the previous segment) and TLD are not equal, but where HL and SL are finite, the computer assumes that there is a transition region (2nd or higher order transition) being processed. The treatment is the same as that for a first order transition except that the temperatures are printed as rounded (integer) values. A sample of the output including the publishable table is given in Appendix II.

ACKNOWLEDGMENT

The support of the Division of Research of the U. S. Atomic Energy Commission is gratefully noted. This endeavor would not have been possible without the encouragement of Professor Edgar F. Westrum, Jr. and J. T. S. Andrews and the use of the University of Michigan computer complex. Discussions with Drs. H. Prophet and A. N. Syverud of the Dow Chemical Company were a significant contribution to the programming endeavor.

BIBLIOGRAPHY

1. B. H. Justice, "Calculation of Heat Capacities and Derived Thermodynamic Functions from Thermal Data with an IBM 704 Digital Computer", U. S. AEC Report Number TID-6206, Oak Ridge, Tenn. (1960).
2. B. H. Justice, "Thermodynamic Properties and Electronic Energy Levels of Eight Rare Earth Sesquioxides," Dissertation Abstr., 23, 3146 (1963) cf. U. S. AEC Report Number TID-12722, Oak Ridge, Tenn. (1961).
3. G. E. Forsythe, J. Soc. Ind. Appl. Math., 5, 74-88 (1957).
4. B. D. Rudin, "On Routines For Least Squares Curve Fitting with Orthogonal Polynomials," Missile Systems Division, Lockheed Aircraft Corp., Palo Alto, Calif., (1957).
(Presented at IBM 650 Computation Seminar, Endicott, N. Y.).
5. T. H. K. Barron and J. A. Morrison, Can. J. Phys., 35, 799 (1957).
6. J. A. Morrison and D. M. T. Newsham, Proc. Phys. Soc., London, Solid State Phys. [2], 1(2), 370 (1968).
7. C. G. Maier and K. K. Kelley, J. Am. Chem. Soc., 54, 3243 (1932).
8. See for instance, J. R. Rice, "The Approximation of Functions," Addison-Wesley Publ. Co., Reading, Mass. (1964).

9. K. K. Kelley, "Contributions to the Data on Theoretical Metallurgy. XIII. High Temperature Heat-Content Heat Capacity, and Entropy Data for the Elements and Inorganic Compounds," U. S. Bur. Mines Bull. 584, U. S. Government Printing Office, Washington (1960).
10. C. H. Shomate, J. Am. Chem. Soc., 66, 928 (1944).
11. J. E. L. Peck, Soc. Ind. Appl. Math Rev., 4, 135 (1962).
12. J. G. Hust and R. D. McCarty, Cryogenics, 7, 200 (1967).
13. F. B. Hildebrand, "Introduction to Numerical Analysis," McGraw Hill Book Co., Inc., New York, 466-469 (1956).

APPENDICES

APPENDIX I

Program FITAB

Reproduction of Fortran IV
Listing of Main and Subroutine Programs

FORTRAN IV G COMPILER

MAIN

```

C      CURVE FITTING OF CP AND H WITH INTEGRATION AND SPECIALIZED OUTPUT
0001      INTEGER RUN,U,WGT,NWT(500),HOLHE(18),NOWGT
0002      REAL LTDSF
0003      REAL*8 CH(500),X(500),W(500),LT,SF,C(50),A,B
0004      LOGICAL DTZERO,T273,T298
0005      COMMON/D/T(500),RUN(500)
0006      COMMON/W/LCR
0007      COMMON/HIN/SF,TDTE,TDTE1,CU(75),LTDSF,NS,N,IZ
0008      COMMON/QFC/TMAX,TMIN,JK,J,LC,TLD,THT,TH,HL,SL,TBASE
0009      DATA NOWGT/'0 '/'
0010      1001 READ (5,1002,END=63)(HOLHE(I),I=1,18)
0011      1002 FORMAT (18A4)
0012      WRITE (6,1003)(HOLHE(I),I=1,18)
0013      1003 FORMAT ('1',24X,18A4)
0014      WRITE (6,1004)
0015      1004 FORMAT (' ')
0016      READ (5,5)U,NPTS,NS,NI,N,TLD,DT,TH,SL,HL
0017      5 FORMAT (5I4,5F10.5)
0018      TRASE = 0.0
0019      DTTEST = 1.0
0020      LCR = 0
0021      WGT = U/100
0022      U = MOD(U,100)
0023      DTZERO = .FALSE.
0024      T273 = .FALSE.
0025      T298 = .FALSE.
0026      IF (U.GT.20) GO TO 6
0027      GO TO 8
0028      6 JK = 1
0029      U = MOD(U,10)
0030      IF (HL.EQ.0.0) GO TO 22
0031      IF (ABS((SL*TLD/HL)-1).LT.1.0E-05) IENT = 1
0032      IF ((HL.GT.0.).AND.(TLD.GT.THT)) IENT = 2
0033      IF (HL.GT.0.) CALL GEFMT(THT,CHP,SP,HP,FFNP,IENT)
0034      GO TO 22
0035      8 JK = 0
0036      U = MOD(U,10)
0037      THT = 0.0
0038      LC = 0
0039      22 READ (5,10)SF,LT,A,B
0040      10 FORMAT (4G10.5)
0041      READ (5,11)(RUN(I),T(I),CH(I),NWT(I),I=1,NPTS)
0042      11 FORMAT (2X,A3,F12.5,E15.7,2X,A1)
0043      TMIN = T(1)
0044      TMAX = T(1)
0045      I298 = 0
0046      DO 12 I=1,NPTS
0047      IF (CH(I).EQ.0.0) I298 = I
0048      IF (T(I).GT.TMAX) TMAX = T(I)
0049      12 IF (T(I).LT.TMIN) TMIN = T(I)
0050      DO 13 I=1,NPTS
0051      X(I) = (T(I) - LT)/SF
0052      GO TO (1,2,3,4),WGT
0053      1 W(I) = 1.0
0054      GO TO 13

```

FORTRAN IV G COMPILER

MAIN

```

0055      2 IF (CH(I).EQ.0.0) GO TO 13
0056      W(I) = 1.0/DSQRT((X(I)-A)*(B-X(I)))
0057      GO TO 13
0058      3 IF (CH(I).EQ.0.0) GO TO 13
0059      W(I) = (1.0/DSQRT((X(I)-A)*(B-X(I))))*(1.0/DABS(CH(I)))
0060      GO TO 13
0061      4 IF (CH(I).EQ.0.0) GO TO 13
0062      W(I) = 1.0/DABS(CH(I))
0063      13 CONTINUE
0064      WMAX = W(I)
0065      DO 14 I = 1,NPTS
0066      IF (NWT(I).EQ.NOWGT) W(I) = 0.0
0067      14 IF (W(I).GT.WMAX) WMAX = W(I)
0068      IF (I298.EQ.0) GO TO 15
0069      TBASE = T(I298)
0070      CH(I298) = 1.0E-06
0071      IF (NWT(I298).NE.NOWGT) W(I298) = 10.0*WMAX
0072      15 CALL ORTHO(X,CH,W,NPTS,NS,NI,N,C)
0073      DO 23 I=1,75
0074      23 CU(I) = 0.0
0075      II = (N-NS+NI)/NI
0076      III=0
0077      IF (NS.GT.0) III=NS
0078      DO 24 I=1,II
0079      24 CU((I-1)*NI+1+III) = C(I)
0080      IZ=I
0081      IF (NS.LT.0) IZ=1-NS
0082      IF (U.EQ.1) GO TO 16
0083      IF (U.EQ.2) GO TO 43
0084      IF (U.EQ.3) GO TO 1001
0085      C      CALCULATE THERMODYNAMIC FUNCTIONS FROM CP AS F(X) POLYNOMIAL
0086      16 J = 1
0087      LTDSF = LT/SF
0088      IF (DT.EQ.0.0) DTZERO = .TRUE.
0089      IF (TLD.LE.273.15) T273 = .TRUE.
0090      IF (TLD.LE.298.15) T298 = .TRUE.
0091      IF (TH.EQ.273.15) T273 = .FALSE.
0092      IF (TH.EQ.298.15) T298 = .FALSE.
0093      TL = TLD
0094      40 IF (DTZERO.AND.(TL.LT.50.0)) DT = 5.0
0095      IF (DTZERO.AND.(TL.GE.50.0)) DT = 10.0
0096      IF (DTZERO.AND.(TL.GE.300.0)) DT = 25.0
0097      IF (DTZERO.AND.(TL.GE.500.0)) DT = 50.0
0098      IF (DTZERO.AND.(TL.GE.1000.0)) DT = 100.0
0099      IF (DTTEST.GT.0.0) GO TO 17
0100      IT = (TLD + 1.0E-05)/DT
0101      TL = (IT + 1)*DT
0102      17 IF (T273.AND.(TL.GT.273.15).AND.(TL.LT.298.15)) GO TO 41
0103      IF (T273.AND.(TL.GT.298.15)) GO TO 18
0104      GO TO 19
0105      18 THT = TL
0106      TL = 273.15
0107      T273 = .FALSE.
0108      GO TO 20
0109      19 IF (T298.AND.(TL.GT.298.15)) GO TO 42

```


FORTRAN IV G COMPILER

MAIN

```

0109      20 XL = (TL - LT)/SF
0110      IF (TL.NE.0.0) GO TO 21
0111      TL = 1.0E-15
0112      XL = (TL - LT)/SF
0113      21 CHP = 0.0
0114      SP = 0.0
0115      HP = 0.0
0116      PREK = 0.0
0117      DO 25 I=1,N
0118      CHP = (CHP + CU(N+IZ+1-I))*XL
0119      HP = (HP+(CU(N+IZ+1-I))/(N+2-I))*XL
0120      SP = (SP + (CU(N+IZ+1-I)-LTDSF*PREK)/(N+1-I))*XL
0121      25 PREK = CU(N+IZ+1-I) - LTDSF*PREK
0122      HP = (HP + CU(IZ))*XL*SF
0123      CHP = CHP + CU(IZ)
0124      SP = SP + (CU(IZ) - PREK*LTDSF)*ALOG(XL+LTDSF)
0125      IF (NS.GE.0) GO TO 29
0126      TDFTE = 0.0
0127      TDFTE1 = 0.0
0128      II = IZ - 1
0129      DO 26 I = 1,II
0130      TDFTE = (TDFTE + CU(I))/XL
0131      26 TDFTE1 = (TDFTE1 + CU(I)/(NS+I-1))/XL
0132      CHP = CHP + TDFTE
0133      SP = SP + TDFTE1
0134      TDFTE = 0.0
0135      II = IZ - 2
0136      IF (II.EQ.0) GO TO 66
0137      DO 28 I=1,II
0138      28 TDFTE = (TDFTE + CU(I)/(NS+I))/XL
0139      66 HP = HP + (TDFTE + CU(IZ-1)*ALOG(XL))*SF
0140      29 IF (N.LT.1) GO TO 34
0141      CP1 = 0.0
0142      II = N - 1
0143      IF (N.EQ.1) GO TO 31
0144      DO 30 I=1,II
0145      30 CP1 = (CP1 + CU(N+IZ+1-I)*(N-I+1))*XL
0146      31 CP1 = (CP1 + CU(IZ+1))/SF
0147      IF (N.LT.2) GO TO 34
0148      CP2 = 0.0
0149      II = N-2
0150      IF (N.EQ.2) GO TO 33
0151      DO 32 I=1,II
0152      32 CP2 = (CP2 + CU(N+IZ-I+1)*(N-I+1)*(N-I))*XL
0153      33 CP2 = (CP2 + 2.0*CU(IZ+2))/SF**2
0154      34 IF (NS.GE.0) GO TO 7
0155      TDFTE = 0.0
0156      TDFTE1 = 0.0
0157      II = IZ - 1
0158      DO 35 I=1,II
0159      TDFTE = (TDFTE+ CU(I)*(NS+I-1))/XL
0160      35 TDFTE1 = (TDFTE1 + CU(I)*(NS+I-1)*(NS+I-2))/XL
0161      CP1 = CP1 + TDFTE/(XL*SF)
0162      CP2 = CP2 + TDFTE/(XL*SF)**2
0163      7 IF (J.EQ.0) GO TO 27

```

FORTRAN IV G COMPILER

MAIN

```

0164      IF (JK.EQ.0) GO TO 36
0165      IF (THT.EQ.TL.AND.HL.EQ.0.0) CHPT = (CHP + CHPT)/2.0
0166      SSL = SL - SP + SPT
0167      HHL = HL - HP + HPT
0168      J = 0
0169      GO TO 27
0170 36     SSL = SL - SP
0171      HHL = HL - HP
0172      J = 0
0173 27     SP = SP + SSL
0174      HP = HP + HHL
0175      IF (TL.EQ.1.0E-15) HFNP = 0.0
0176      IF (TL.EQ.1.0E-15) FFNP = 0.0
0177      IF (TL.EQ.1.0E-15) GO TO 64
0178      HFNP = HP/TL
0179      FFNP = SP - HFNP
0180 64     IENT = 3
0181      CALL WRIREG(TL,CHP,SP,HP,HFNP,FFNP,CP1,CP2)
0182      IF ((THT.EQ.TLD).AND.(JK.EQ.1).AND.(HL.EQ.0.0).AND.(TL.EQ.TLD))
      ICHP = CHPT
0183      CALL GEFMT(TL,CHP,SP,HP,FFNP,IENT)
0184      IF (TL.NE.TH) GO TO 37
0185      THT = TH
0186      CHPT = CHP
0187      HPT = HP
0188      SPT = SP
0189 37     IF ((TL.NE.273.15).OR.(.NOT.T273)) GO TO 38
0190      THT = 273.15 + DT
0191      T273 = .FALSE.
0192 38     IF ((TL.NE.298.15).OR.(.NOT.T298)) GO TO 39
0193      THT = 298.15 + DT
0194      T298 = .FALSE.
0195 39     IF (TL.NE.273.15) GO TO 67
0196      TL = THT - DT
0197      THT = 0.0
0198 67     IF (TL.NE.298.15) GO TO 68
0199      TL = THT - DT
0200      THT = 0.0
0201 68     IF (((TL + DT).GT.TH).AND.((TL + DT - TH).LT.DT)) TL = TH - DT
0202      TL = TL + DT
0203      DTTEST = TL - DT - TLD
0204      IF (TL.LE.TH) GO TO 40
0205      TL = TL - DT
0206      GO TO 1001
0207 41     THT = TL
0208      TL = 273.15
0209      T273 = .FALSE.
0210      GO TO 20
0211 42     IF (TL.NE.(273.15 + DT)) THT = TL
0212      TL = 298.15
0213      T298 = .FALSE.
0214      GO TO 20
C      CALCULATE THERMODYNAMIC FUNCTIONS FROM H-HTB AS F(X) POLYNOMIAL
0215 43     J = 1
0216      LTDSF = LT/SF

```

FORTRAN IV C COMPILER

MAIN

```

0217      IF (DT.EQ.0.0) DTZERO = .TRUE.
0218      IF (TLD.LE.273.15) T273 = .TRUE.
0219      IF (TLD.LE.298.15) T298 = .TRUE.
0220      IF (TH.EQ.273.15) T273 = .FALSE.
0221      IF (TH.EQ.298.15) T298 = .FALSE.
0222      TL = TLD
0223      44 IF (DTZERO.AND.(TL.LT.50.0)) DT = 5.0
0224      IF (DTZERO.AND.(TL.GE.50.0)) DT = 10.0
0225      IF (DTZERO.AND.(TL.GE.300.0)) DT = 25.0
0226      IF (DTZERO.AND.(TL.GE.500.0)) DT = 50.0
0227      IF (DTZERO.AND.(TL.GE.1000.0)) DT = 100.0
0228      IF (DTTEST.GT.0.0) GO TO 45
0229      IT = (TLD + 1.0E-05)/DT
0230      TL = (IT + 1)*DT
0231      45 IF (T273.AND.(TL.GT.273.15).AND.(TL.LT.298.15)) GO TO 61
0232      IF (T273.AND.(TL.GT.298.15)) GO TO 46
0233      GO TO 47
0234      46 THT = TL
0235      TL = 273.15
0236      T273 = .FALSE.
0237      GO TO 48
0238      47 IF (T298.AND.(TL.GT.298.15)) GO TO 62
0239      48 XL = (TL - LT)/SF
0240      IF (TL.NE.0.0) GO TO 57
0241      TL = 1.0E-15
0242      XL = (TL - LT)/SF
0243      57 CALL ENTHIN(XL,CHP,SP,HP,TL)
0244      CP1 = 0.0
0245      CP2 = 0.0
0246      IF (N.LE.1) GO TO 51
0247      DO 49 I=2,N
0248      49 CP1 = (CP1 + CU(N+IZ+2-I)*(N+2-I)*(N+1-I))*XL
0249      CP1 = CP1/(XL*SF**2)
0250      IF (N.LE.2) GO TO 51
0251      DO 50 I=3,N
0252      50 CP2 = (CP2 + CU(N+IZ+3-I)*(N+3-I)*(N+2-I)*(N+1-I))*XL
0253      CP2 = CP2/(XL*SF**3)
0254      51 IF (NS.GE.0) GO TO 53
0255      TDFTE = 0.0
0256      TDFTE1 = 0.0
0257      II = IZ-1
0258      DO 52 I=1,II
0259      TDFTE = (TDFTE + CU(I)*(NS+1-I)*(NS-I))/XL
0260      52 TDFTE1 = (TDFTE1 + CU(I)*(NS+1-I)*(NS-I)*(NS-I-1))/XL
0261      CP1 = CP1 + TDFTE/(SF*XL)**2
0262      CP2 = CP2 + TDFTE1/(SF*XL)**3
0263      53 IF (J.EQ.0) GO TO 56
0264      STB = 0.0
0265      HTB = 0.0
0266      IF ((I298.EQ.0).OR.(JK.EQ.1)) GO TO 54
0267      XTB = (TBASE - LT)/SF
0268      CALL ENTHIN(XTB,CHTB,STB,HTB,TBASE)
0269      54 IF ((JK.EQ.1).AND.(THT.EQ.TL).AND.(HL.EQ.0.0)) CHPT = (CHP + CHPT)
0270      1/2.0
      IF (JK.EQ.0) GO TO 55

```

FORTRAN IV G COMPILER

MAIN

```

0271      SSL = SL - SP + SPT
0272      HHL = HL - HP + HPT
0273      J = 0
0274      55 IF (JK.EQ.1) GO TO 56
0275      IF (I298.EQ.0) SSL = SL - SP
0276      IF (I298.NE.0) SSL = SL STD
0277      HHL = HL - HTB
0278      J = 0
0279      56 HP = HP + HHL
0280      SP = SP + SSL
0281      IF (TL.EQ.1.0E-15) HFNP = 0.0
0282      IF (TL.EQ.1.0E-15) FFNP = 0.0
0283      IF (TL.EQ.1.0E-15) GO TO 65
0284      HFNP = HP/TL
0285      FFNP = SP - HFNP
0286      65 IENT = 3
0287      CALL WRIREG(TL,CHP,SP,HP,HFNP,FFNP,CP1,CP2)
0288      IF ((THT.EQ.TLD).AND.(JK.EQ.1).AND.(HL.EQ.0.0).AND.(TL.EQ.TLD))
1CHP = CHP
0289      CALL GEFMT(TL,CHP,SP,HP,FFNP,IENT)
0290      IF (TL.NE.TH) GO TO 58
0291      THT = TH
0292      CHPT = CHP
0293      HPT = HP
0294      SPT = SP
0295      58 IF ((TL.NE.273.15).OR.(.NOT.T273)) GO TO 59
0296      THT = 273.15 + DT
0297      T273 = .FALSE.
0298      59 IF ((TL.NE.298.15).OR.(.NOT.T298)) GO TO 60
0299      THT = 298.15 + DT
0300      T298 = .FALSE.
0301      60 IF (TL.NE.273.15) GO TO 69
0302      TL = THT - DT
0303      THT = 0.0
0304      69 IF (TL.NE.298.15) GO TO 70
0305      TL = THT - DT
0306      THT = 0.0
0307      70 IF (((TL + DT).GT.TH).AND.((TL + DT - TH).LT.DT)) TL = TH - DT
0308      TL = TL + DT
0309      DTTEST = TL - DT - TLD
0310      IF (TL.LE.TH) GO TO 44
0311      TL = TL - DT
0312      GO TO 1001
0313      61 THT = TL
0314      TL = 273.15
0315      T273 = .FALSE.
0316      GO TO 48
0317      62 IF (TL.NE.(273.15 + DT)) THT = TL
0318      TL = 298.15
0319      T298 = .FALSE.
0320      GO TO 48
0321      63 CALL SYSTEM
0322      END

```

FORTRAN IV G COMPILER

WRIREG

```

0001      SUBROUTINE WRIREG(T,CP,S,H,HDT,FDT,DCP,DTCP)
0002      COMMON/W/LCR
0003      IF (LCR.NE.0) GO TO 3
0004      LCR = 6
0005      WRITE (6,1)
0006      1 FORMAT ('1')
0007      WRITE (6,2)
0008      2 FORMAT (///7X,'T',10X,'CP',12X,'S',14X,'H',12X,'H/T',10X,'-G/T',10
0009      1X,'DCP/DT',7X,'D2CP/DT2'/)
0010      3 IF (MOD(LCR,6).NE.0) GO TO 5
0011      IF (LCR.EQ.54) WRITE (6,1)
0012      IF (LCR.EQ.54) LCR = 0
0013      LCR = LCR + 1
0014      WRITE (6,4)
0015      4 FORMAT (' ')
0016      5 WRITE (6,6)T,CP,S,H,HDT,FDT,DCP,DTCP
0017      6 FORMAT (F10.2,F13.4,F14.4,F15.4,F13.4,F14.4,1PE16.4,1PE14.4)
0018      LCR = LCR + 1
0019      RETURN
0019      END

```

FORTAN IV G COMPILER

ENTHIN

```

0001      SUBROUTINE ENTHIN(X,CP,S,H,T)
0002      REAL LTDSF
0003      REAL*8 SF
0004      COMMON/HINZ/SF,TDFTE,TDFTE1,CU(75),LTDSF,NS,N,I2
0005      DIMENSION D(102)
0006      DO 1 I=1,102
0007      1 D(I) = 0.0
0008      CP = 0.0
0009      S = 0.0
0010      H = 0.0
0011      D(1) = CU(N+12)
0012      IF (N.LT.2) GO TO 5
0013      DO 2 I=2,N
0014      CP = (CP + CU(N+12-I+2)*N-I+2)*X
0015      H = (H + CU(N+12-I+2))*X
0016      S = (S + D(I+1)/(N-I+1))*X
0017      2 D(I+2) = CU(N+12-I+1) - 2.0*LTDSF*D(I+1) - D(I)*LTDSF**2
0018      3 CP = (CP + CU(I2+1))/SF
0019      H = (H + CU(I2+1))*X + CU(I2)
0020      D(N+3) = CU(I2) - (LTDSF**2)*D(N+1)
0021      S = H/T + (S + (D(N+2)*LTDSF - D(N+3))/(X + LTDSF) + D(N+2)*ALOG(X
      + LTDSF))/SF
0022      IF (NS.GE.0) GO TO 5
0023      TDFTE = 0.0
0024      TDFTE1 = 0.0
0025      TDFTE2 = 0.0
0026      I1 = I2 - 1
0027      DO 4 I=1,I1
0028      TDFTE = (TDFTE + CU(I)*(I+NS-1))/X
0029      TDFTE1 = (TDFTE1 + CU(I))/X
0030      4 TDFTE2 = (TDFTE2 + CU(I)/(I+NS-1))/X
0031      CP = CP + TDFTE/(SF*X)
0032      H = H + TDFTE1
0033      S = S + TDFTE1/T + TDFTE2/(X*SF)
0034      5 RETURN
0035      END

```

[illegible]

FORTAN IV G COMPILER

GEFMT

```

0048      WRITE (8'1000*LC,OFMT)T,CP,S,H,FDT
0049      LC = LC + 4
0050      LPC = 5
0051      RETURN
0052      11 LC = LC - 2
0053      12 WRITE (8'1000*LC,1)
0054      LC = LC + 1
0055      IF (MOD(LC,57).NE.1) GO TO 12
0056      WRITE (8'1000*LC,3)
0057      LC = LC + 4
0058      WRITE (8'1000*LC,OFMT)T,CP,S,H,FDT
0059      LC = LC + 4
0060      LPC = 5
0061      IPCT = IPCT + 1
0062      RETURN
0063      13 IF ((THT.NE.273.15).OR.(THT.NE.298.15)) ITK = THT
0064      IF ((THT.NE.273.15).OR.(THT.NE.298.15)) THTEST = .TRUE.
0065      IF ((THT.EQ.273.15).OR.(THT.EQ.298.15)) T = THT
0066      IF (ABS(H).GE.5000.0) HTEST = .TRUE.
0067      IF (HTEST.AND.(ABS(H).LT.50000.0)) IHK = H + (0.5*ABS(H)/H)
0068      IF (HTEST.AND.(ABS(H).GT.50000.0))
1IHK = ((IFIX(H) + 5*ABS(IFIX(H))/IFIX(H))/10)*10
GO TO 8
0069
0070      14 IF ((JK.EQ.0).AND.(T.EQ.1.0E-15).AND.(ABS(H).LT.5.0E-04)) RETURN
0071      IF ((T.NE.THT).OR.(HL.NE.0).OR.(JK.NE.1)) GO TO 15
0072      ITK = T
0073      THTEST = .TRUE.
0074      IF (ABS(H).GE.5000.0) HTEST = .TRUE.
0075      IF (HTEST.AND.(ABS(H).LT.50000.0)) IHK = H + (0.5*ABS(H)/H)
0076      IF (HTEST.AND.(ABS(H).GT.50000.0))
1IHK = ((IFIX(H) + 5*ABS(IFIX(H))/IFIX(H))/10)*10
LC = LC - 2
0077      WRITE (8'1000*LC,OFMT)T,CP,S,H,FDT
0078      LC = LC + 1
0079      WRITE (8'1000*LC,1)
0080      RETURN
0081
0082      15 IF ((T.GE.TMIN).AND.(T.LE.TMAX)) OFMT(19) = ZH(11)
0083      IF ((T.GE.TMIN).AND.(T.LE.TMAX)) OFMT(28) = ZH(11)
0084      IF (JK.EQ.0) GO TO 16
0085      IF ((T.LT.TMIN).OR.(T.GT.TMAX)) GO TO 17
0086      GO TO 19
0087      16 IF ((T.LT.TMIN).OR.(T.GT.TMAX)) GO TO 17
0088      IF ((T.EQ.TLD).AND.(T.GE.TMIN).AND.(ABS(HL).GT.0.0)) GO TO 18
0089      IF ((TLD.EQ.0.0).AND.(T.EQ.5.0).AND.(T.GE.TMIN).AND.(TRASE.EQ.0.0))
1) GO TO 18
GO TO 19
0090
0091      17 OFMT(19) = ZH(16)
0092      OFMT(28) = ZH(17)
0093      18 OFMT(35) = ZH(16)
0094      OFMT(44) = ZH(17)
0095      OFMT(52) = ZH(16)
0096      OFMT(61) = ZH(17)
0097      OFMT(68) = ZH(16)
0098      OFMT(83) = ZH(17)
0099      GO TO 20

```


FORTRAN IV G COMPILER

GEFMT

0100	19	OFMT(19) = ZH(11)
0101		OFMT(28) = ZH(11)
0102		OFMT(35) = ZH(11)
0103		OFMT(44) = ZH(11)
0104		OFMT(52) = ZH(11)
0105		OFMT(61) = ZH(11)
0106		OFMT(68) = ZH(11)
0107		OFMT(83) = ZH(11)
0108	20	IF ((T.EQ.273.15).OR.(T.EQ.298.15)) GO TO 21
0109		IF (HL.LE.0.0) GO TO 22
0110		IF ((T.EQ.THT).AND.(ABS((T*SL/HL)-1).LT.1.0E-05)) GO TO 21
0111		GO TO 22
0112	21	OFMT(9) = ZH(12)
0113		OFMT(10) = ZH(8)
0114		OFMT(11) = ZH(22)
0115		OFMT(12) = ZH(3)
0116		OFMT(13) = ZH(11)
0117		GO TO 23
0118	22	ITK = T
0119		THTEST = .TRUE.
0120		OFMT(9) = ZH(13)
0121		OFMT(10) = ZH(5)
0122		OFMT(11) = ZH(23)
0123		OFMT(12) = ZH(4)
0124		OFMT(13) = ZH(14)
0125	23	IF (CP.LT.10.0) IBL1 = 0
0126		IF (CP.GE.10.0) IBL1 = 1
0127		IF (CP.GE.100.) IBL1 = 2
0128		IF (CP.LT.15.0) IBL2 = 0
0129		IF (CP.GE.15.0) IBL2 = 1
0130		IF (CP.GE.150.) IBL2 = 2
0131		OFMT(15) = ZH(7-IBL1)
0132		OFMT(23) = ZH(6+IBL1-IBL2)
0133		OFMT(25) = ZH(4-IBL2)
0134		IF (ABS(S).LT.10.0) IBL1 = 0
0135		IF (ABS(S).GE.10.0) IBL1 = 1
0136		IF (ABS(S).GE.100.0) IBL1 = 2
0137		IF (S.LT.0.0) IBL1 = IBL1 + 1
0138		OFMT(31) = ZH(7+IBL2-IBL1)
0139		IF (ABS(S).GE.15.0) IBL2 = 1
0140		IF (ABS(S).LT.15.0) IBL2 = 0
0141		OFMT(39) = ZH(6+IBL1-IBL2)
0142		OFMT(41) = ZH(4-IBL2)
0143		IF (ABS(H).GE.0.0) IBL1 = 0
0144		IF (ABS(H).GE.10.0) IBL1 = 1
0145		IF (ABS(H).GE.100.0) IBL1 = 2
0146		IF (ABS(H).GE.1000.0) IBL1 = 3
0147		IF (ABS(H).GE.10000.0) IBL1 = 4
0148		IF (ABS(H).GE.100000.0) IBL1 = 5
0149		IF (H.LT.0) IBL1 = IBL1 + 1
0150		IBL2 = 10 + IBL2 - IBL1
0151		OFMT(47) = ZH(1 + IBL2/10)
0152		OFMT(48) = ZH(MOD(IBL2,10)+1)
0153		IF (IBL2.LT.10) OFMT(47) = ZH(11)
0154		IF (ABS(H).LT.5000.0) GO TO 24

FORTTRAN IV G COMPILER

GEFMT

```

0155      IBL2 = 4
0156      IF (ABS(H).GE.5000.0) HTEST = .TRUE.
0157      IF (HTEST.AND.(ABS(H).LT.50000.0)) IHK = H + (0.5*ABS(H)/H)
0158      IF (HTEST.AND.(ABS(H).GT.50000.0))
1IHK = ((IFIX(H) + 5*IARS(IFIX(H))/IFIX(H))/10)*10
0159      OFMT(55) = ZH(13)
0160      OFMT(57) = ZH(11)
0161      OFMT(58) = ZH(11)
0162      IF (IHK.GE.0) IBL1 = 0
0163      IF (IHK.LT.0) IBL1 = 1
0164      IF (IABS(IHK).LT.10000) OFMT(56) = ZH(5+IBL1)
0165      IF (IABS(IHK).GE.10000) OFMT(56) = ZH(6+IBL1)
0166      IF (IABS(IHK).GE.100000) OFMT(56) = ZH(7+IBL1)
0167      GO TO 25
0168      24 OFMT(55) = ZH(12)
0169      OFMT(57) = ZH(22)
0170      IF (ABS(H).LT.50.0) IBL2 = 0
0171      IF (ABS(H).GE.50.0) IBL2 = 1
0172      IF (ABS(H).GE.500.0) IBL2 = 2
0173      OFMT(56) = ZH(6+IBL1-IBL2)
0174      OFMT(58) = ZH(4-IBL2)
0175      25 IF (ITK.EQ.0) GO TO 27
0176      IF (ABS(FDT).LT.10.0) IBL1 = 0
0177      IF (ABS(FDT).GE.10.0) IBL1 = 1
0178      IF (ABS(FDT).GE.100.0) IBL1 = 2
0179      IF (ABS(FDT).GE.1000.0) IBL1 = 3
0180      IF (FDT.LT.0.0) IBL1 = IBL1 + 1
0181      OFMT(64) = ZH(6+IBL2-IBL1)
0182      OFMT(71) = ZH(12)
0183      OFMT(73) = ZH(22)
0184      DO 26 I=75,80
0185      26 OFMT(I) = ZH(11)
0186      IF (ABS(FDT).LT.15.0) IBL2 = 0
0187      IF (ABS(FDT).GE.15.0) IBL2 = 1
0188      OFMT(72) = ZH(6+IBL1-IBL2)
0189      OFMT(74) = ZH(4-IBL2)
0190      GO TO 28
0191      27 OFMT(68) = ZH(11)
0192      OFMT(83) = ZH(11)
0193      OFMT(71) = ZH(21)
0194      OFMT(72) = ZH(13)
0195      OFMT(73) = ZH(18)
0196      OFMT(74) = ZH(12)
0197      OFMT(75) = ZH(13)
0198      OFMT(76) = ZH(18)
0199      OFMT(77) = ZH(13)
0200      OFMT(78) = ZH(19)
0201      OFMT(79) = ZH(20)
0202      OFMT(80) = ZH(21)
0203      OFMT(64) = ZH(3+IBL2)
0204      CP = 0.0001
0205      S = 0.0001
0206      28 IF (LC.GT.57*IPCT) GO TO 30
0207      IF ((LC+1).GT.57*IPCT).AND.(MOD(LPC,6).EQ.0)) GO TO 29
0208      IF (MOD(LPC,6).EQ.0) GO TO 31

```

FORTRAN IV G COMPILER

GEFMT

```
0209          GO TO 32
0210      29 LC = LC + 1
0211      30 WRITE (8*1000*LC,3)
0212          LC = LC + 4
0213          LPC = 1
0214          IPCT = IPCT + 1
0215          GO TO 32
0216      31 LC = LC + 1
0217          LPC = 1
0218      32 WRITE (8*1000*LC,GFMT)T,CP,S,H,FDT
0219          LC = LC + 1
0220          LPC = LPC + 1
0221          IF (THTEST) T = TDU
0222          IF (T.NE.TH) GO TO 33
0223          WRITE (8*1000*LC,3)
0224          LC = LC + 1
0225      33 RETURN
0226          END
```

FORTRAN IV G COMPILER

ORTHO

```

0001      SUBROUTINE ORTHO(X,CH,W,M,NS,NI,N,C)
0002      REAL*8 X(500),W(500),C(50),DEV(500),CHT(500),Y(500),PN(500),PM(500)
0003      1,PL(500),BI(50),BJ(50),BK(50),DM,SUM(5),ALFA,BETA,AP,PCT,SIG,WTD
0004      COMMON/O/T(500),RUN(500)
0005      WRITE(6,9)
0006      9 FORMAT (T2,'POWER',I21,'(DEV/Y)**2',T48,'DEV**2',T72,'W*DEV**2',T8
0007      19,'TEST'//)
0008      DO 1 I=1,M
0009      PN(I) = X(I)**NS
0010      IF (NS.EQ.0) PN(I) = 1.0
0011      PM(I) = 0.0
0012      PL(I) = 0.0
0013      1 Y(I) = CH(I)
0014      DO 2 I=1,50
0015      BI(I) = 0.0
0016      BJ(I) = 0.0
0017      BK(I) = 0.0
0018      2 C(I) = 0.0
0019      DM = 1.0
0020      NR = (N-NS+NI)/NI
0021      NIT = 1
0022      BK(1) = 1.0
0023      15 DO 3 I=1,5
0024      3 SUM(I) = 0.0
0025      DO 5 I=1,M
0026      IF (NIT.EQ.1) GO TO 4
0027      PN(I) = (X(I)**NI - ALFA)*PM(I) - BETA*PL(I)
0028      4 SUM(1) = SUM(1) + W(I)*(PN(I)**2)
0029      SUM(2) = SUM(2) + W(I)*(X(I)**NI)*(PN(I)**2)
0030      SUM(3) = SUM(3) + W(I)*(X(I)**NI)*PN(I)*PM(I)
0031      SUM(4) = SUM(4) + W(I)*PN(I)*CH(I)
0032      IF (NS.EQ.0) SUM(5) = SUM(5) + W(I)*PN(I)
0033      IF (NS.NE.0) SUM(5) = SUM(5) + W(I)*PN(I)*X(I)**NS
0034      PL(I) = PM(I)
0035      5 PM(I) = PN(I)
0036      ALFA = SUM(2)/SUM(1)
0037      AP = SUM(4)/SUM(1)
0038      BETA = SUM(3)/DM
0039      DM = SUM(1)
0040      ORTE = SUM(5)/SUM(1)
0041      SIG = 0.0
0042      WTD = 0.0
0043      PCT = 0.0
0044      DO 7 I=1,M
0045      Y(I) = Y(I) - AP*PM(I)
0046      IF (CH(I).EQ.1.0E-06) DEV(I) = 0.0
0047      IF (CH(I).EQ.1.0E-06) GO TO 6
0048      PCT = PCT + (Y(I)/CH(I))**2
0049      DEV(I) = 100.0*(Y(I)/CH(I))
0050      6 SIG = SIG + Y(I)**2
0051      7 WTD = WTD + W(I)*Y(I)**2
0052      DO 8 I=1,NIT
0053      8 C(I) = C(I) + AP*BK(I)
0054      SIG = SIG/(M-NIT)
0055      PCT = PCT/(M-NIT)

```

FORTRAN IV G COMPILER

ORTHO

```

0054      IPOW = NS + (NIT - 1)*NI
0055      WRITE (6,10)IPOW,PCT,SIG,WTD,ORTE
0056      10 FORMAT (I5,1P3E25.4,1PE14.2)
0057      IF (NIT.GE.NR) GO TO 16
0058      II = NIT
0059      NIT = NIT + 1
0060      BK(NIT) = 1.0
0061      DO 12 I=1,II
0062      IF ((NIT-I-1).EQ.0) GO TO 11
0063      BI(NIT-I-1) = BJ(NIT-I-1)
0064      11 BJ(NIT-I) = BK(NIT-I)
0065      12 CONTINUE
0066      DO 14 I=1,II
0067      BK(I) = 0.0
0068      IF (I.EQ.1) GO TO 13
0069      BK(I) = BK(I) + BJ(I-1)
0070      13 BK(I) = BK(I) - ALFA*BJ(I)
0071      IF ((NIT.LT.3).OR.((NIT-2).LT.1)) GO TO 14
0072      BK(I) = BK(I) - BETA*BI(I)
0073      14 CONTINUE
0074      GO TO 15
0075      16 WRITE (6,18)NS,NI,N
0076      18 FORMAT ('1',T9,'COEFFICIENTS FOR F(X), MIN POWER =',T43,I3,T47,'DE
      ILTA POWER =',T60,I2,T63,'MAX POWER =',T74,I3//)
0077      WRITE (6,19)(C(I),I=1,NIT)
0078      19 FORMAT (1P4E21.9)
0079      WRITE (6,20)
0080      20 FORMAT (//T52,'DEVIATION TABLE'///)
0081      WRITE (6,21)
0082      21 FORMAT (/T14,'X',T36,'T',T54,'Y',T74,'Y-F(X)',T95,'O/O DEV',T109,'
      INO.'//)
0083      WRITE (6,22)(X(I),T(I),CH(I),Y(I),DEV(I),RUN(I),I=1,M)
0084      22 FORMAT (5(F20.6,F20.4,2F20.6,F20.3,8X,A3,9X)//)
0085      RETURN
0086      END

```

APPENDIX II

Typical Output

Photographically Reproduced Copy of Typical Output
of Finished Tabulation from Program FITAB

HEAT CAPACITY DATA FROM 5 TO 30K									
TRIPLYCENE									
POWER	(DEVI/Y)**2	DEVI**2	MEDEV**2	TEST					
3	2.6853D-01	1.3895D 00	4.9544D 01	1.00E 00					
5	2.6029D-02	1.0756D-01	3.6245D 00	2.62E-15					
7	1.8626D-02	9.9288D-03	1.8595D-01	-4.03E-14					
9	3.6086D-02	2.5601D-03	7.7246D-02	-4.40E-13					
11	2.9948D-02	2.4637D-03	6.7683D-02	-2.37E-12					
13	1.5546D-02	1.7096D-03	4.1731D-02	-2.21E-11					
15	7.2526E-03	1.4507D-03	3.1289D-02	-7.19E-10					
COEFFICIENTS FOR F(X), MIN POWER = 3 DELTA POWER = 2 MAX POWER = 15									
3.041395465D 01 1.777021295D 02 -1.741719670D 03 5.992312670D 03									
-1.038520514D 04 9.019641716D 03 -3.113198901D 03									
DEVIATION TABLE									
X	T	Y	Y-F(X)	O/D DEV	NO.				
0.163225	5.7130	0.124270	-0.023684	-19.058	34				
0.187446	6.5606	0.199223	-0.029258	-14.887	35				
0.214169	7.4959	0.321683	-0.026409	-8.210	36				
0.240576	8.4201	0.484213	-0.015863	-3.276	37				
0.268910	9.4119	0.740131	0.037060	5.007	38				
0.300491	10.5172	1.025592	0.047928	4.673	39				
0.336433	11.7751	1.358600	0.012111	0.891	40				
0.376163	13.1657	1.804993	-0.003978	-0.220	41				
0.414582	14.5104	2.212277	-0.074868	-3.380	42				
0.452245	15.8266	2.822265	0.048161	1.706	43				
0.496463	17.3762	3.330051	-0.011523	-0.346	44				
0.546383	19.1234	3.978941	0.004593	0.115	45				
0.600514	21.0180	4.674088	0.008951	0.191	46				
0.662697	23.1944	5.454899	-0.010560	-0.193	47				
0.736779	25.7873	6.358299	0.003703	0.058	48				
0.822285	28.7800	7.343970	-0.000540	-0.007	49				
0.904616	31.6616	8.252118	0.000034	0.000	50				
T	CP	S	H	H/T	-G/T	DCP/DT	DCP/DT2		
0.00	0.0000	0.0	0.0	0.0	0.0	2.1201E-33	4.2562E-18	6.1070E-02	2.6591E-02
5.00	0.0973	0.0314	0.1184	0.0237	0.0077	6.1070E-02	2.6591E-02	2.5004E-01	4.0586E-02
10.00	0.8430	0.2730	2.0665	0.2067	0.0663	2.5004E-01	4.0586E-02	3.6739E-01	3.1963E-03
15.00	2.4695	0.9021	10.1000	0.6733	0.2288	3.6739E-01	3.1963E-03	3.6410E-01	3.8347E-03
20.00	4.2923	1.8631	27.0210	1.3510	0.5121	3.6410E-01	3.8347E-03	3.3111E-01	2.3583E-02
25.00	6.1010	3.0178	53.0639	2.1226	0.8953	3.3111E-01	2.3583E-02		

TRIPTYCENE		HEAT CAPACITY DATA FROM 20 TO 522K		
POWER	(DEV/Y)**2	DEV**2	W*DEV**2	TEST
0	9.11850 00	1.38120 03	2.17380 05	1.00E 00
1	2.51450-03	1.59900 00	1.68240 02	-1.40E-16
2	2.79480-03	1.34740 00	1.48190 02	5.97E-15
3	6.99930-03	6.28280-01	8.35700 01	-1.59E-14
4	2.52100 03	3.35540-01	4.16790 01	-6.16E-14
5	3.66520-04	5.67880-02	6.84860 00	8.27E-14
6	5.39790-05	2.18230-02	2.36750 00	8.91E-14
7	9.17560-06	1.69460-02	1.66160 00	-4.43E-14
8	3.79410-06	1.60650-02	1.56210 00	-1.89E-12
9	4.50320-06	1.52700-02	1.45700 00	-1.69E-13
10	3.16010-06	1.52650-02	1.43260 00	3.37E-14
11	3.05070-06	1.54600-02	1.43140 00	2.67E-12
12	2.99400-06	1.55320-02	1.42620 00	-1.64E-11

COEFFICIENTS FOR F(X), MIN POWER = 0 DELTA POWER = 1 MAX POWER = 12

6.1390356810 01	7.0035465540 01	4.9410560540 00	-2.49664446350 01
5.3858403730-01	3.5241930240 01	-1.6666312160 01	-4.9259594620 01
6.4579963680 01	4.1770975990 01	-1.0349005060 02	-7.6952196280 00
4.9008200870 01			

DEVIATION TABLE

X	T	Y	Y-F(X)	O/O DEV	NO.
-0.926824	19.1234	3.978941	0.002117	0.053	45
-0.919935	21.0180	4.674088	-0.005617	-0.120	46
-0.912020	23.1944	5.458489	-0.002614	-0.048	47
-0.902592	25.7873	6.358299	0.001673	0.026	48
-0.891709	28.7800	7.343970	-0.000256	-0.003	49
-0.881231	31.6616	8.252118	0.001325	0.016	50
-0.870146	34.7098	9.177420	0.012262	0.134	51
-0.869359	34.9264	9.238914	0.010472	0.113	53
-0.856210	38.5422	10.259418	0.005186	0.051	52
-0.854743	38.9456	10.365936	0.000759	0.007	54
-0.840105	42.9711	11.415505	-0.021046	-0.184	55
-0.824212	47.3418	12.527288	-0.004899	-0.039	56
-0.807154	52.0325	13.635842	-0.004093	-0.030	57
-0.788752	57.0931	14.745740	-0.022332	-0.151	58
-0.768663	62.6176	15.947507	0.012715	0.080	59
-0.746290	68.7704	17.177939	0.005788	0.034	60
-0.721238	75.6595	18.471953	-0.029354	-0.159	61
-0.693360	83.3259	19.983261	0.048061	0.241	62
-0.662945	91.6901	21.517645	0.046327	0.215	63
-0.650492	95.1146	22.121756	0.024770	0.112	17
-0.629945	100.7651	23.105420	-0.024420	-0.106	64
-0.622267	102.8767	23.490078	-0.026659	-0.113	18
-0.591385	111.3691	25.038120	-0.044617	-0.178	19
-0.559124	120.2410	26.725196	-0.017815	-0.067	20
-0.526295	129.2688	28.465140	0.000186	0.001	21
-0.492678	138.5136	30.281264	0.014552	0.048	22
-0.458626	147.8779	32.136727	0.002849	0.009	23
-0.424423	157.2836	34.049921	-0.003490	-0.010	24
-0.391258	166.4042	35.976725	0.019007	0.053	25
-0.358343	175.4556	37.917341	0.027486	0.072	26
-0.325190	184.5728	39.899278	0.020476	0.051	27
-0.292035	193.6904	41.885160	-0.025574	-0.061	28
-0.258709	202.8551	43.985165	-0.010747	-0.024	29
-0.225286	212.0463	46.161944	0.032827	0.071	30
-0.216512	214.4592	46.542183	-0.153709	-0.330	1
-0.198712	219.3542	47.963450	0.109342	0.228	2
-0.191749	221.2690	48.342412	0.032266	0.067	31
-0.171105	226.9460	49.552948	-0.118820	-0.240	3

-0.157874	230.5847	50.596504	0.044749	0.088	32
-0.138300	235.9674	51.884525	0.021194	0.041	4
-0.124126	239.8654	52.913078	0.092995	0.176	33
-0.101297	246.1432	54.360282	-0.011994	-0.022	5
-0.065672	255.9403	56.789114	-0.030244	-0.053	6
-0.031295	265.3938	59.176545	-0.027636	-0.047	7
0.001936	274.5325	61.456295	-0.069687	-0.113	8
0.034126	283.3848	63.703256	-0.081925	-0.129	9
0.067616	292.5945	66.056683	-0.084144	-0.127	10
0.078702	295.6431	66.949859	0.029010	0.043	1
0.089558	298.6284	67.746170	0.061661	0.091	2
0.102363	302.1499	68.457205	-0.127638	-0.186	11
0.116248	305.9683	69.830632	0.270432	0.387	3
0.127263	308.9973	70.527880	0.194812	0.276	***
0.136091	311.4250	70.840421	-0.111412	-0.157	12
0.154499	316.4873	72.488120	0.248363	0.343	5
0.168875	320.4407	73.069556	-0.173469	-0.237	13
0.186501	325.2878	74.721998	0.252418	0.338	6
0.204093	330.1255	75.464081	-0.225260	-0.298	14
0.219516	334.3669	76.893651	0.138864	0.181	7
0.241644	340.4521	77.967085	-0.309157	-0.397	15
0.253631	343.7485	79.270374	0.173714	0.219	8
0.266708	347.3447	79.459326	-0.525182	-0.666	16
0.288841	353.4314	81.625255	0.135199	0.166	9
0.323226	362.8872	83.921943	0.120067	0.143	10
0.356548	372.1606	85.975857	-0.067303	-0.078	11
0.381031	378.7834	87.653673	0.026109	0.030	12
0.415255	388.2061	89.961970	0.103891	0.115	*13
0.450732	397.9514	92.227483	0.091800	0.100	14
0.485398	407.4844	94.365964	0.030851	0.033	15
0.519299	416.8071	96.485589	0.026379	0.027	16
0.549560	425.1289	98.393106	0.059986	0.061	17
0.576404	432.5112	99.890461	-0.087549	-0.088	18
0.602952	439.8118	101.500090	-0.088113	-0.087	19
0.614498	442.9871	102.133060	-0.150207	-0.147	20
0.640409	450.1125	103.862100	0.031273	0.030	*21
0.666104	457.1787	105.339560	-0.008223	-0.008	22
0.691428	464.1428	106.837140	0.013162	0.012	23
0.716391	471.0076	108.272670	0.013651	0.013	24
0.740025	477.5068	109.617750	0.020084	0.018	25
0.762425	483.6670	111.091670	0.244042	0.220	26
0.786389	490.2571	112.070140	-0.094134	-0.084	27
0.812091	497.3250	113.619490	0.064679	0.057	28
0.837474	504.3054	114.938360	0.026061	0.023	29
0.848468	507.3286	115.162070	-0.336141	-0.292	30
0.873491	514.2100	116.877880	0.039722	0.034	31
0.889649	518.6536	117.716390	-0.001879	-0.002	54
0.894801	520.0703	117.984700	-0.018583	-0.016	*61
0.897900	520.9226	118.259720	0.083650	0.071	32
0.900887	521.7439	118.407440	0.063816	0.054	46
0.902037	522.0603	118.511330	0.102859	0.087	55
0.920653	527.1797	119.380000	-0.105819	-0.089	EXC

I	CP	S	H	H/I	-6/I	DCP/DI	DCP/DI2
25.00	6.0887	3.0178	53.0625	2.1225	0.8953	3.4250E-01	-5.5889E-03
30.00	7.7332	4.2753	87.6758	2.9225	1.3528	3.1570E-01	-5.1180E-03
35.00	9.2499	5.5829	130.1797	3.7194	1.8635	2.9142E-01	-4.5863E-03
40.00	10.6520	6.9108	179.9805	4.4995	2.4112	2.6988E-01	-4.0284E-03
45.00	11.9534	8.2414	236.5352	5.2563	2.9851	2.5113E-01	-3.4708E-03
50.00	13.1679	9.5644	295.3672	5.9873	3.5771	2.3514E-01	-2.9327E-03
60.00	15.3895	12.1653	442.3594	7.3127	4.7926	2.1079E-01	-1.9636E-03
70.00	17.4132	14.6915	606.5000	8.6643	6.0273	1.9525E-01	-1.1773E-03
80.00	19.3175	17.1420	790.2266	9.8778	7.2642	1.8661E-01	-5.8193E-04
90.00	21.1622	19.5242	992.6523	11.0295	8.4947	1.8304E-01	-1.5878E-04
100.00	22.9898	21.8484	1213.4141	12.1341	9.7143	1.8296E-01	1.2373E-04
110.00	24.8290	24.1257	1452.4922	13.2045	10.9212	1.8516E-01	3.0036E-04
120.00	26.6975	26.3659	1710.0977	14.2508	12.1151	1.8873E-01	4.0331E-04
130.00	28.6060	28.5779	1986.5742	15.2813	13.2966	1.9307E-01	4.5918E-04
140.00	30.5602	30.7691	2282.3672	16.3026	14.4665	1.9782E-01	4.8782E-04
150.00	32.5631	32.9455	2597.9453	17.3196	15.6259	2.0278E-01	5.0248E-04
160.00	34.6161	35.1124	2933.7969	18.3362	16.7761	2.0785E-01	5.1068E-04
170.00	36.7202	37.2738	3250.4375	19.3555	17.9183	2.1298E-01	5.1554E-04
180.00	38.8759	39.4335	3668.3750	20.3799	19.0536	2.1815E-01	5.1713E-04
190.00	41.0832	41.5943	4068.1250	21.4112	20.1831	2.2331E-01	5.1377E-04
200.00	43.3418	43.7581	4490.2031	22.4510	21.3078	2.2840E-01	5.0317E-04
210.00	45.6506	45.9291	4935.1250	23.5006	22.4285	2.3334E-01	4.8322E-04
220.00	48.0077	48.1070	5403.3750	24.5608	23.5462	2.3803E-01	4.5255E-04
230.00	50.4100	50.2938	5895.4297	25.6323	24.6615	2.4233E-01	4.1079E-04
240.00	52.8532	52.4908	6411.7109	26.7155	25.7753	2.4621E-01	3.5866E-04
250.00	55.3322	54.6985	6952.6133	27.8104	26.8880	2.4950E-01	2.9777E-04
260.00	57.8410	56.9174	7518.4570	28.9171	28.0003	2.5214E-01	2.3047E-04
270.00	60.3728	59.1478	8105.5078	30.0392	29.1126	2.5409E-01	1.5947E-04
273.15	61.1739	59.8527	8300.5375	30.3857	29.4630	2.5456E-01	1.3680E-04
280.00	62.9205	61.3894	8725.9648	31.1642	30.2253	2.5533E-01	8.7591E-05
290.00	65.4770	63.6420	9367.9453	32.3033	31.3387	2.5585E-01	1.7458E-05
298.15	67.5621	65.4854	9910.0742	32.2386	32.2469	2.5577E-01	-3.8820E-05
300.00	68.0352	65.9048	10035.5078	33.4517	32.4531	2.5569E-01	-4.8662E-05
325.00	76.3968	71.6023	11816.0664	36.3571	35.2452	2.5266E-01	-1.8604E-04
350.00	80.6448	77.3452	13754.3945	39.2983	38.0470	2.4680E-01	-2.7404E-04
375.00	86.7241	83.1175	15466.8906	42.2584	40.8592	2.3938E-01	-3.1178E-04
400.00	92.6107	88.9035	18088.9844	45.2225	43.6811	2.3156E-01	-3.9955E-04
425.00	98.3042	94.6898	20475.8125	48.1784	46.5114	2.2397E-01	-3.0101E-04
450.00	103.8065	100.4654	23002.6172	51.1169	49.3485	2.1600E-01	-3.5231E-04
475.00	109.0837	106.2205	25664.2812	54.0301	52.1904	2.0563E-01	-4.7875E-04
500.00	115.0764	111.9439	28454.3750	56.9087	55.0352	1.9459E-01	-2.5559E-04
527.18	119.4859	118.1220	31627.2734	59.9933	58.1287	2.1656E-01	2.7588E-03

TRIPTYCENE LIQUID HEAT CAPACITY DATA				
POWER	(DEV/Y)**2	DEV**2	W*DEV**2	TEST
0	3.70510-05	6.47560-01	6.69310 00	1.00E 00
1	3.72330-06	6.43610-02	5.47680-01	0.0
2	1.36920-06	2.36650-02	1.45760-01	2.12E-15

COEFFICIENTS FOR F(X), MIN POWER = 0 DELTA POWER = 1 MAX POWER = 2				
1.3152515C3D 02	1.654252790D 00	9.637905F47D-01		

DEVIATION TABLE					
X	T	Y	Y-F(X)	O/O DEV	NO.
-0.721305	527.1804	130.860000	0.026618	0.020	EX1
-0.366439	532.5034	131.010960	-0.037448	-0.029	42
-0.279395	533.8091	130.943780	-0.194445	-0.148	53
-0.176302	535.3555	131.292590	0.029098	0.022	65
-0.125570	536.1165	131.595000	0.262342	0.199	43
0.039465	538.5920	131.461890	-0.130096	-0.099	66
0.267497	542.0125	132.138130	0.101458	0.077	67
0.454846	545.4226	132.508800	-0.071002	-0.054	68
0.720980	548.8147	133.230630	0.011738	0.009	69

T	CP	S	H	H/T	-G/T	CCP/DT	D2CP/DT2
527.18	130.8334	131.8516	38863.4883	73.7196	58.1320	1.7590E-02	8.5670E-03
530.00	130.9170	132.5469	39232.5469	74.0237	58.5232	4.1750E-02	8.5670E-03
540.00	131.7629	135.0039	40545.2344	75.0838	59.9202	1.2742E-01	8.5670E-03
550.00	133.4654	137.4375	41876.6602	76.1285	61.3090	2.1309E-01	8.5670E-03

EXECUTION TERMINATED

Thermodynamic Functions of Triptycene

T	C_p	S°	$H^\circ - H_0^\circ$	$-(G^\circ - H_0^\circ)/T$
Crystal I				
5	(0.097)	(0.031)	(0.118)	(0.008)
10	0.843	0.273	2.067	0.066
15	2.469	0.902	10.100	0.229
20	4.292	1.863	27.021	0.512
25	6.095	3.018	53.06	0.895
30	7.733	4.275	87.68	1.353
35	9.250	5.583	130.18	1.863
40	10.652	6.911	179.98	2.411
45	11.953	8.241	236.54	2.985
50	13.168	9.564	299.37	3.577
60	15.39	12.165	442.36	4.793
70	17.41	14.692	606.5	6.027
80	19.32	17.14	790.2	7.264
90	21.16	19.52	992.7	8.495
100	22.99	21.85	1213.4	9.714
110	24.83	24.13	1452.5	10.921
120	26.70	26.37	1710.1	12.115
130	28.61	28.58	1986.6	13.297
140	30.56	30.77	2282.4	14.466
150	32.56	32.95	2597.9	15.63
160	34.62	35.11	2933.8	16.78
170	36.72	37.27	3290.4	17.92
180	38.88	39.43	3668.4	19.05
190	41.08	41.59	4068.1	20.18
200	43.34	43.76	4490.2	21.31
210	45.65	45.93	4935.1	22.43
220	48.01	48.11	5403	23.55
230	50.41	50.29	5895	24.66
240	52.85	52.49	6412	25.78
250	55.33	54.70	6953	26.89
260	57.84	56.92	7518	28.00
270	60.37	59.15	8110	29.11
273.15	61.17	59.85	8301	29.46
280	62.92	61.39	8726	30.23
290	65.48	63.64	9368	31.34

Table (continued)

T	C _p	S°	H°-H° ₀	-(G°-H° ₀)
298.15	67.56	65.49	9910	32.25
300	68.04	65.90	10036	32.45
325	74.40	71.60	11816	35.25
350	80.64	77.35	13754	38.05
375	86.72	83.12	15847	40.86
400	92.61	88.90	18089	43.68
425	98.30	94.69	20476	46.51
450	103.81	100.47	23003	49.35
475	109.08	106.22	25664	52.19
500	114.08	111.94	28454	55.04
527.18	(119.49)	(118.12)	(31627)	(58.13)
Liquid				
527.18	(130.83)	(131.85)	(38863)	(58.13)
530	130.92	132.55	39233	58.52
540	131.76	135.00	40545	59.92
550	(133.47)	(137.44)	(41871)	(61.31)

* Units: cal/(mole °K)