Title
USE OF COMPUTERS IN THE EVALUATION OF THERMODYNAMIC DATA

Permalink
https://escholarship.org/uc/item/3db4n3d5

Author
Hawkins, Donald T.

Publication Date
1966-07-01
This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545.
DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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 its 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, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
USE OF COMPUTERS IN THE EVALUATION OF THERMODYNAMIC DATA

Donald T. Hawkins

July 1966
USE OF COMPUTERS IN THE EVALUATION OF THERMODYNAMIC DATA

Donald T. Hawkins

Inorganic Materials Research Division
Lawrence Radiation Laboratory
and
Department of Mineral Technology
College of Engineering
University of California
Berkeley, California

July 1966

In 1955, a project for the evaluation of the thermodynamic properties of metals and binary metallic alloys was begun. The first edition of the results appeared in 1963, and contained data on 67 elements and 180 alloy systems. The project is currently continuing; the scope of the evaluation has been expanded to include several non-metallic elements (e.g. C, Si, and As) and alloy systems (e.g. carbides). It is proposed to expand the evaluation to include systems such as the oxides, nitrides, and borides.

At the outset of the project, all data were calculated by hand, but with the rapidly increasing volume of new data, this became a monumental task. Continual refinements of experimental techniques are leading to more precise data, which will necessitate further revision of the previous values. In order to cope with the production of data and to reduce the probabilities of error, an investigation into the possibility of utilizing high-speed digital computers in the data...
evaluation process was undertaken. Several programs for making some of the more routine calculations have been developed. The purpose of this report is to present these programs as an illustration of the application of computers to this problem.

Most of the programs were originally written in the FORTRAN IV language for use on an IBM 7044 computer. With the acquisition by the Lawrence Radiation Laboratory of a CDC 6600 machine, which uses Chippewa FORTRAN, the older programs were converted to the Chippewa language. Newer programs have been written in Chippewa FORTRAN.

The methods used in the evaluation have been described in detail previously, and will be mentioned only briefly here. A short description and listing of each program are given in the Appendix of this report.

**ELEMENTS**

Low-Temperature Data. The only application of computers to the evaluation of low-temperature data is the integration of the selected \( C_p \) versus \( T \) curve by Simpson's Rule to obtain \( S^o_{298} \). One short program has been written to perform this integration, but in practice, no saving is gained by the time the cards are punched, verified, and read into the computer. Therefore, this program has not been used for some time, and no summary is included in this report.*

Gas Properties. Several compilations of tables of ideal gas thermodynamic properties of metals have been published. As each appeared,
a program was written to convert the values to a consistent system of units. The tables currently in use are those by Hilsenrath, Messina, and Evans. Since these tables can be used for any system of units by multiplying by the appropriate value of \( R \), no unit conversion is necessary. A program (HILSE), which has been incorporated as a subroutine into the master program for evaluating the data on an element (ELEMNT), calculates the table of selected gas phase values using the units recommended by the National Bureau of Standards. A small, additive factor given by Hilsenrath, et al., to correct atomic weights to the values tabulated by Cameron and Wichers is also used in the calculation.

**High-Temperature Data.** The first step is to obtain, plot, and select values for the function

\[
Y = \frac{H^\circ_T - H^\circ_{298.15}}{T - 298.15}
\]

Heat content measurements are often quite voluminous and are reported in a variety of ways. Considerable effort is often expended in the rather mundane tasks of unit conversion and calculation of \( Y \). The first program listed in the Appendix (HEATCO) calculates values for \( Y \) from reported values of \( H^\circ_T - H^\circ_{298.15} \) expressed in units of joules/gm, joules/gm-atom, cal/gm, or cal/gm-atom, and values of \( T \) expressed in °C or °K, and prints tables of \( T \) in °K and \( Y \) in cal/gm-atom °degree. Values of \( H^\circ_T - H^\circ_{298.15} \) in cal/gm-atom are also
Once the selected $Y$ curve has been drawn, calculation of the tables of selected values begins. Several shorter programs have been incorporated into one master program (ELEMNT), which completely calculates and prints the table in the standard format used in References 1 and 2. Values of $Y$ are read in at $50^\circ K$ intervals, and $C_p$ values are calculated from the relation

$$C_p = Y + (T - 298.15) \frac{dY}{dT}.$$  

The slope of the $Y$-function curve, $\frac{dY}{dT}$, is approximated by successively taking $100^\circ$ intervals along the chosen curve, finding the slope of the line between the two end points, and assuming that the slope at the midpoint of the interval has the same value. Since the $Y$-function is generally a smooth curve, this assumption should not introduce a very large error. An excellent check is available because the $C_p$ values thus obtained can be integrated to yield values of $H_T^0 - H_{298.15}^0$, which are then compared with values obtained from the definition of the $Y$-function:

$$H_T^0 - H_{298.15}^0 = Y(T - 298.15).$$

In several trial cases, the discrepancy in the enthalpy values was less than 1 calorie, which is well within the experimental scatter. Should the occasion arise where an error of more than one calorie were introduced due to sharp inflections in the $Y$ curve, $Y$ values taken at closer intervals would reduce the error. Provision has been made in
the program to use values at 25°K intervals, if needed. Once the Cp values are obtained, values of $S^\circ_T - S^\circ_{298.15}$ can be calculated by integration of Cp/T values. Combining the $S^\circ_T - S^\circ_{298.15}$ and $H^\circ_T - H^\circ_{298.15}$ values with the value of $S^\circ_{298}$ chosen from the low-temperature data yields values of the Gibbs energy function, $(G^\circ_T - H^\circ_{298.15})/T$.

For the liquid metals, a constant Cp is often assumed. Its value and the value of $\Delta S_m$ are read in as input data. Calculation of the remainder of the table is then straightforward. In cases where the Cp has been measured and is not constant, the data are at present calculated by hand, but these cases are well in the minority. When reliable data on many liquid metals become available, a program to calculate values in the liquid region when Cp is not constant can easily be written.

In the case of a polymorphic metal, provision has been made to include the values of the β-phase if Cp(β) is constant. Since the majority of polymorphic metals have only one additional phase, no provision was made for more than 1 additional phase.

Calculation of the gas properties is done by a subroutine (HILSE) as described in the section entitled Gas Properties.

Since values of the change in Gibbs energy function $\Delta \left[ (G^\circ_T - H^\circ_{298.15})/T \right]$ for the reaction

$$M(c) = M(g, 1\text{ atm})$$
are useful in evaluating vapor pressure data, values of this function at each tabulation temperature are calculated from the gas and condensed phase tables and printed. These values are also punched on cards for use in other programs.

**Vapor Pressure Data.** The vapor pressure data are evaluated by the Third Law Method. If the data are numerous, as they often are, doing the Third Law calculation by computer to determine the value of \( \Delta H^o_{v,298} \) from the measured vapor pressures saves much time. In this and all similar programs the value of the change in Gibbs energy function, \( \Delta \left( \frac{(G^o_T - H^o_{298})}{T} \right) \), at temperatures other than the even 100° temperatures is found by a subroutine (LAGINT) which uses the method of second order Lagrangian interpolation. The error introduced by this procedure is of the order of 0.001, which is an order of magnitude less than if the calculation were done graphically. The program to do the Third Law calculation (S3 LAWS) calculates \( \Delta H^o_{v,298} \) from the relation

\[
\Delta H^o_{v,298} = -RTlnP - T \Delta \left( \frac{(G^o_T - H^o_{298})}{T} \right)
\]

The cards obtained as output from ELEMNT, giving values of T and the Gibbs energy function at even 100° intervals, are used as input, as well as cards containing measured temperatures and pressures. The output consists of a table of values for T, P, log P, and \( \Delta H^o_{v,298} \). In addition, an average value of \( \Delta H^o_{v,298} \) is calculated, as well as the deviation of
each measurement from the average, and the average and standard deviations. An option is available for obtaining punched card output as well as printed output. If the pressure data are in millimeters of Hg, instead of atmospheres, provision has been made for unit conversion before the calculation of $\Delta H^\circ_{v,298}$ takes place.

If no thermal data are available, making impossible the calculation of the Gibbs energy function, a program (VAPOR) is used to calculate values of $R\ln P$ and $\frac{1}{T}$ for use in the Second Law method of determining $\Delta H^\circ_v$ over the measured temperature range. In this program also, conversion from millimeters Hg to atmospheres is available.

Once the selected value of $\Delta H^\circ_{v,298}$ is known, program VPCALC is used to obtain Table 3 (Vapor Pressure Data). The cards from ELEMNT containing Gibbs energy function values are used as input as in the case of S 3 LAWS. Values of $\Delta G^\circ_{v,T}$ and $P$ are calculated from the relations

$$\Delta G^\circ_{v,T} = T \left[ \frac{G^\circ_T - H^\circ_{298}}{T} \right] + \Delta H^\circ_{v,298}$$

$$\Delta G^\circ_{v,T} = -RT \ln P.$$  

In calculating $P$, special provision had to be made for the extremely small values sometimes obtained (e.g. for W at 298.15°K, $P = 3.8 \times 10^{-142}$ atm.). If these values were calculated directly, an error would occur due to the limitation on the magnitude of numbers which can be stored in the computer. In order to avoid this, the value
of the exponent and the mantissa are calculated separately. By setting

\[ P = 1, \]

the normal boiling point can be found:

\[
T_{bp} = \frac{-\Delta H^\circ_{v, 298}}{\Delta \left[ \frac{(G^\circ_T - H^\circ_{298})}{T} \right]}
\]

However, since the Gibbs energy function is dependent on temperature, \( T_{bp} \) cannot be solved for explicitly. An iteration procedure using Subroutine LAGINT to interpolate the Gibbs energy function is employed to calculate \( T_{bp} \). The limit of accuracy set on \( T_{bp} \) is 1 degree. In a similar way, values of \( T \) at even pressures \( (10^{-10}, 10^{-9}, \ldots, 10^{-1}) \) are also calculated:

\[
T_p = \frac{-\Delta H^\circ_{v, 298}}{-R \ln P - \Delta \left[ \frac{(G^\circ_T - H^\circ_{298})}{T} \right]}
\]

thus completing the evaluation.

In the previous programs it has been assumed that the vapor is an ideal monatomic gas. For several metals the vapor also contains higher atomic species. In these cases, several methods are available, but each case is somewhat specialized so that general programs for the complete evaluation cannot be written. Some programs for special cases where the calculations are lengthy have been written, and an example of one of these will serve as an illustration of what can be done.

The program (B3LAW) was written for selenium, in which both diatomic and hexatomic species exist. The program calculates \( \Delta H^\circ_{v, 298} \).
for both species and averages the values for each species. The deviation of each measurement from the average as well as the average deviation are calculated for each species in an analogous manner to that of Program S3LAWS.

**ALLOYS**

The data for alloys are far more varied than the data for elements. Each system usually has to be treated as an individual case, and very few standard procedures are available. In addition, the data on alloys are much more sparse than the data on elements. This means that the application of computers to the evaluation of the thermodynamic data on alloys is very difficult. Thus far, only two programs have been proven useful.

The first is a rather short program (ACTIV) which calculates activity values from vapor pressure data. The punched cards from Program ELEMNT are used as input to calculate $P^o$. Subroutine LAGINT is used to interpolate the Gibbs energy values.

$$P^o = \text{antilog} \left[ \frac{1}{R} \left( \Delta \left\{ \frac{G^o_T - H^o_{298}}{T} \right\} - \frac{\Delta H^o_{V, 298}}{T} \right) \right]$$

Vapor Pressures over the alloy are then read in and activities calculated:

$$a_i = \frac{P_i}{P^o_i}$$

From these values and the mole fraction of component $i$, $x_i$, other
thermodynamic quantities follow:

\[ \gamma_i = \frac{a_i}{x_i} \]

\[ \Delta G_i = RT \ln a_i \]

\[ \Delta G_i^{xs} = RT \ln \gamma_i \]

Output consists of tables of the above values plus an option to have them on punched cards.

The only other program that has been developed is useful when data exist over the entire composition range. This program (ALLOY) takes equilibrium data in the form of the \( \alpha \)-function \( \alpha_1 = \frac{\Delta G_1^{xs}}{(1-x_1)^2} \) and performs the Gibbs-Duhem integration to obtain \( \Delta G_2^{xs} \). The integration is done by Simpson's Rule. Values for \( \Delta G_1^{xs} \), \( \Delta G_1^{xs} \), \( \gamma_i \), \( a_i \), and the same quantities for component 2 are calculated and printed in the standard format used in Reference 1. The integral quantities \( \Delta G \) and \( \Delta G^{xs} \) are also calculated.

If heat or entropy data are available, subroutines are used to calculate the remaining quantities. Heat data are correlated using the \( Q \)-function \( Q = \Delta H/\chi_1 \chi_2 \). The partial molar heats for one component can be obtained graphically, and from these and the Gibbs energy values, the remaining partial molar and integral heats and entropies are calculated and printed in the standard format.

Entropy data are correlated with the \( \beta \)-function \( \beta_1 = \frac{\Delta S_1^{xs}}{(1-x_1)^2} \).
From this function, the Gibbs-Duhem integration will yield $\Delta S^x_s$, and all other quantities then follow as described above.

**CONCLUSION**

Programs for evaluating the thermodynamic data of metals and alloys have been described. Listings of these programs follow in the Appendix.
ACKNOWLEDGEMENTS

The assistance of Carl Quong and Bill Dempster of the Math and Computing Group, Lawrence Radiation Laboratory, in debugging the programs described in this paper is gratefully acknowledged. Subroutines LAGINT, LAGR, and BAINS were furnished by the Computer Library. Mrs. Marian Smith wrote several of the preliminary programs used in the evaluation project.

This work was performed under the auspices of the United States Atomic Energy Commission.
REFERENCES


This section gives listings of all programs discussed in the preceding section, as follows:

<table>
<thead>
<tr>
<th>Name of Program</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEATCO</td>
<td>A-2</td>
</tr>
<tr>
<td>ELEMNT</td>
<td>A-4</td>
</tr>
<tr>
<td>ENDPG (used by ELEMNT)</td>
<td>A-10</td>
</tr>
<tr>
<td>AREA (used by ELEMNT)</td>
<td>A-11</td>
</tr>
<tr>
<td>HILSE (used by ELEMNT)</td>
<td>A-12</td>
</tr>
<tr>
<td>HILSE (as a separate program)</td>
<td>A-14</td>
</tr>
<tr>
<td>S3LAWS</td>
<td>A-15</td>
</tr>
<tr>
<td>LAGINT</td>
<td>A-18</td>
</tr>
<tr>
<td>LAGRNG (used by LAGINT)</td>
<td>A-19</td>
</tr>
<tr>
<td>BAINS (used by LAGINT)</td>
<td>A-20</td>
</tr>
<tr>
<td>VAPOR</td>
<td>A-21</td>
</tr>
<tr>
<td>VPCALC</td>
<td>A-22</td>
</tr>
<tr>
<td>B3LAW</td>
<td>A-26</td>
</tr>
<tr>
<td>ACTIV</td>
<td>A-28</td>
</tr>
<tr>
<td>ALLOY</td>
<td>A-31</td>
</tr>
<tr>
<td>AREA (used by ALLOY)</td>
<td>A-35</td>
</tr>
<tr>
<td>QPLOT (used by ALLOY)</td>
<td>A-36</td>
</tr>
<tr>
<td>BETA (used by ALLOY)</td>
<td>A-37</td>
</tr>
</tbody>
</table>
PROGRAM HEATCO (INPUT, OUTPUT, TAPE 2=INPUT, TAPE 3=OUTPUT)

HIGH TEMPERATURE HEAT CONTENT CALCULATION

DIMENSION IDENT (12), T (1000), H (1000), DIFF (1000), Y (1000), TT (1000), HH (1000), P (1000)

READ INPUT DATA

10 READ (2, 1) IDENT
   1 FORMAT (12A6)
   IF (IDENT .EQ. 6H6) CALL EXIT

20 READ (2, 2) J, K, L, AW
   2 FORMAT (314, F10.5)
   I = 3

30 I = I + 1
   READ (2, 4) T (I), H (I)
   4 FORMAT (F8.2, F12.3)

IF (T (I) .EQ. 0.0) GO TO 5

GO TO 3

50 NUMX = I - 1

WRITE INPUT DATA

60 WRITE (3, 14) IDENT
   14 FORMAT (1H1, 12A6)
   WRITE (3, 2) J, K, L, AW
   WRITE (3, 13)

70 FORMAT (54HO INPUT DATA)
   WRITE (3, 15)

80 FORMAT (54HO T       H)
   DO 16 I = 1, NUMX
   WRITE (3, 17) T (I), H (I)
   16 CONTINUE

IF (J .EQ. 1) GO TO 6

GO TO 3

90 CONVERT DEGREES C TO DEGREES K

100 DO 7 I = 1, NUMX
    TT (I) = T (I) + 273.15
   7 T (I) = TT (I)

110 IF (K .EQ. 1) GO TO 9

GO TO 31

120 CONVERT JOULES TO CALORIES

130 DO 10 I = 1, NUMX
    H (I) = (H (I)) / (4.1840)
   10 IF (L .EQ. 1) GO TO 30

GO TO 53

140 CONVERT GRAMS TO GRAM-ATOMS

150 DO 32 I = 1, NUMX
    H (I) = H (I) * AW
   32 H (I) = H (I) * AW

160 DO 12 I = 1, NUMX
    DIFF (I) = T (I) - 298.15

CALCULATE Y

170 Y (I) = (H (I)) / DIFF (I)

WRITE RESULTS

180 WRITE (3, 14) IDENT
   WRITE (3, 18)

190 FORMAT (54HO CALCULATION RESULTS)
   WRITE (3, 19)

200 FORMAT (54HO T(KELVIN)       H(T)-H(298)      Y)
   DO 20 I = 1, NUMX
      WRITE (3, 21) T (I), H (I), Y (I)
   20 CONTINUE
A-3

21 FORMAT (5X,F8.2,6X,F8.2,8X,F10.4)
20 CONTINUE
   GO TO 80
END
PROGRAM ELEMNT(INPJT,OUTPUT,TAPE 2=INPUT,TAPE 3=OUTPUT,TAPE 14)
DIMENSION NAME(2),IDENT(5),Y(500),T(500),H(500),SLOPE(500),CP(500)
1,DS(500),DELTAF(500),DELTAH(500),DELTAAS(500),GGEF(500),DATE(2),EVAL(3),TT(100),
2,HLL(100),SLQ(100),FLQ(100),GGEF(500),DATE(2),EVAL(3),TT(100),
3,DELTAF(500),DEVI(100),F3(100),FLF(500),DG(500)
COMMON TINC,N,NUM,T.NAME,NPAGE,LINES,L,MM,IK,TM,TF
READ INPJ DATA
1 READ (2,101) NAME(1),NAME(2),DATE(1),DATE(2),(EVAL(J), J=1,3),
IDENT
100 FORMAT(12A6)
   IF (NAME(1) .EQ. 6H ) CALL EXIT
READ(2,101) SST,IM,DSM,CPM,CPL,TINC,TF,TTR
101 FORMAT(8F10.4)
READ(2,1001) SLOPE(1),SLOPE(2),SLP
1001 FORMAT(3F8.6)
   IF THERE IS NO TRANSITION, TTR = 0
   IF(TTR .EQ. 0.0) GO TO 2
   READ(2,101) CPTR,STR,GPB
2 I=0
3 I=I+8
   N=I-7
READ(2,101) Y(I-7),Y(I-6),Y(I-5),Y(I-4),Y(I-3),Y(I-2),Y(I-1),Y(I)
DO 4 II=N,1
   IF(Y(I1) .LE. 0.0) GO TO 5
4 CONTINUE
   GO TO 3
5 N=II-1
   CALL EXIT
   WRITE(3,102)
102 FORMAT(1H1)
   WRITE(3,104) NAME,NPAGE
104 FORMAT(47H EVALUATION OF THE THERMODYNAMIC PROPERTIES OF ,2A6,5X,
16H PAGE ,12)
   WRITE(3,111) EVAL,DATE,IDENT
111 FORMAT(4H BY ,3A6,10X,2A6,10X//5A6//)
   WRITE(3,105)
105 FORMAT(35X,11H INPUT DATA ...)
   WRITE(3,103)
   WRITE INPUT DATA
   WRITE(3,106) SST,TH,CPM,DSM,CP,CPM,SLOPE(1),SLP,TF
105 FORMAT(7H SST = ,F6.3,3X,6H TH = ,F6.1,3X,12H CP AT MP = ,F5.3,3X,
17H DSM = ,F5.2,3X,7H CPL = ,F5.3 / 23H DYT/DT AT 298 DEG.K = F8.6,
23X,15H MP AT MP = ,F8.6,3X,11H FINAL T = ,F6.1)
   WRITE INPUT DATA FOR BETA PHASE IF THERE IS A TRANSITION
   IF (TT3 .LE. 0.0) WRITE(3,107) TTR,CPB,STR,CPB
107 FORMAT (7H MTR = ,F6.1,3X,12H CP(BETA) = ,F5.3,3X,7H STR = ,F5.2,
13X,13H MP AT TTR = ,F5.3)
   WRITE(3,109)
THE FOLLOWING PROGRAM SEGMENT, WHICH APPEARS SEVERAL TIMES IN THIS PROGRAM, SPACES TO THE END OF THE PAGE AND PRINTS THE LAST LINE ON EACH PAGE. ONLY 55 LINES ARE PRINTED ON EACH PAGE.

WRITE(3,103)
LINES=16
14 DO 10 I=1,N
WRITE(3,103) T(I),Y(I)
10 FORMAT(12X,2H T,11X,2H Y)
WRITE(3,109) T(I),Y(I)
109 FORMAT(10X,F7.2,6X,F7.4)

CALL ENDPG
END

THE FOLLOWING PROGRAM SEGMENT, WHICH APPEARS SEVERAL TIMES IN THIS PROGRAM, SPACES TO THE END OF THE PAGE AND PRINTS THE LAST LINE ON EACH PAGE. ONLY 55 LINES ARE PRINTED ON EACH PAGE.

LINES=LINES+1
IF(LINES.EQ.55) CALL ENDPG
IF(I .NE. N) GO TO 10
WRITE(3,103)
LINES=LINES+1
IF(LINES.LT.55) GO TO 151
CALL ENDPG

CONTINUE

CALCULATE HEAT VALUES DIRECTLY FROM THE Y- PLOT

DO 7 I=1,N
H(I)=Y(I)*T(I)-298.15
7 N=N-1

CALCULATE CP VALUES BY TAKING SLOPES ON THE Y- PLOT THEN INTEGRATE THESE CP VALUES USING SUBROUTINE AREA TO OBTAIN THE HEAT.

DO 8 I=1,N
CP(I) = Y(I)**2*(T(I)-298.15)*SLOPE(I)
8 DS(I) = CP(I)/T(I)
CALL AREA(CP,DELTAS)
CALL AREA(DS,DELTAS)
DELTAS(1) = 0
DELTAS(2) = DS(2)*1.85
DELTAS(1)=DELTAS(1)+DELTAS(2)
DELTAS(2) = H(2)/298.10

NUM = NUMBER OF TEMPERATURES UP TO THE MELTING POINT
GO TO 99 I=3,NUM
DELTAS(1) = DELTAS(1) + DELTAS(2)
99 DELTAS(1) = DELTAS(1) + DELTAS(2)
GEE(I) = SST

REINDEX THE TEMPERATURE, CP, AND HEAT VALUES SO THAT ONLY THE EVEN DEGREE INTERVALS ARE RETAINED IN THE CALCULATION

IF(ITINC .EQ. 25.) GO TO 500
DO 501 I=4,N,2
II=(II+1)/4
CP(I)=CP(II)
T(I)=T(II)
501 CONTINUE
GO TO 503
503 DO 502 I=6,N,4
II=(II+5)/4
CP(I)=CP(II)
T(I)=T(II)
502 H(I)=H(II)
DU 12 I=2,NJM
      FEF(I) = (H(I)/T(I))-DELTAS(I)-SST
      GEF(I) = -FEF(I)
      IF(T(I)-TF) 12,166,166
12 CONTINUE
      WRITE TABLE HEADINGS FOR CONDENSED PHASE
156 WRITE(3,125)
125 FORMAT(32H CONDENSED PHASE SELECTED VALUES																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
IF(T(I)-TM) 23,23,24

23 CONTINUE

24 IK=IK+1

NM=NUM+4

DO 25 I=NM,IK

HH(I)=CPB*(T(I)-TK)+HK

SS(I)=CPB*ALOG(T(I)/TK)+SK

FEF(I)=(HH(I)+T(I))-SS(I)-SST

GGEF(I)=-FEF(I)

WRITE(3,113) T(I),CPB,HH(I),SS(I),GGEF(I)

LINES=LINES+1

IF(LINES.LT.55) GO TO 25

CALL ENDPG

25 CONTINUE

C CALCULATE AND WRITE VALUES AT THE MELTING POINT FOR BOTH THE SOLID

C AND LIQUID

HH=CPB*(TM-T(IK))+HHH(IK)

SSA=CPB*ALOG(TM/T(IK))+SS(IK)

GO TO 26

26 HH=(ICPM+CP(NUM))/2.*T(I-NUM))H(NUM)

SSA=((ICPM+CP(NUM))/2.*ALOG(TIM/T(NUM))+DELTAS(NUM)

27 HH=HH/TM-SSTA-SST

GG=-FEF

WRITE(3,113) TM,CPM,HH,SSA,GG

LINES=LINES+1

IF(LINES.LT.55) GO TO 27

CALL ENDPG

27 HH=TM*DSM+HH

SM=DSM+SSA

FEFEM=(HM/TM)-SM-SST

GGEFM=-FEFEM

WRITE(3,113) TM,CPM,HM,SM,GGEFM

LINES=LINES+1

IF(LINES.LT.55) GO TO 28

CALL ENDPG

28 IF (TTR.EQ.0.0) GO TO 29

TMM=T(IK)+100.

GO TO 30

29 TMN=T(NUM)+100.

30 HMM=CPM*(TMM-TM)+HM

SM=CPM*ALOG(TMM/TM)+SM

FMM=(HMM/TMM)-SMM-SST

GMM=-FMM

WRITE(3,113) TMM,CPM,HMM,SM,GMM

LINES=LINES+1

IF(LINES.LT.55) GO TO 31

CALL ENDPG

31 TT(I)=TMM+100.

DO 32 I=2,70

TT(I)=TT(I-1)+100.

IF(TT(I)-TF) 32,33,33

32 CONTINUE

33 L=I

C CALCULATE AND WRITE VALUES IN LIQUID FROM THE MELTING POINT UP TO THE

C FINAL TEMPERATURE

DO 34 I=1,L

ALOG(I)=CPM*(T(I)-TM)+HM
\[ S_L(t) = C_P \cdot \frac{\text{ALOG}(T(t)/T_M)}{T_M} + S_M \]

\[ E_L(t) = \left( \frac{H_L(t) + S_L(t) - S_L(t) - S_M}{T(t)} \right) \]

\[ G_L(t) = E_L(t) + \sum_{\text{r}} \]

\[ \text{WRITE}(3, 113) \quad T(t), C_P, H_L(t), S_L(t), \]

\[ \text{IF}(\text{LINES} \cdot \text{LT} \cdot 55) \quad \text{GO TO} \quad 34 \]

\[ \text{CALL ENDPG} \]

34 CONTINUE

35 CONTINUE

36 \text{CALL ENDPG} \]

C HILSE CALCULATES THE GAS PHASE TABLE

C THE FOLLOWING SECTION REINDEXES THE VALUES SO THAT THE INDICES RUN FROM

C 1 TO A FINAL VALUE.

37 \text{IF}(TTR \cdot \text{GT} \cdot 0.0) \quad \text{GO TO} \quad 37

38 \text{GO TO} \quad 38

39 \text{CONTINUE} \]

40 \text{CONTINUE}

41 \text{WRITE}(3, 126)

126 \text{FORMAT}(3X, 2H T, 4X, 11H \text{DELTA}(GEF))

42 \text{CONTINUE}

43 \text{WRITE}(3, 103)

44 \text{LINES} = \text{LINES} + 2

C PUNCH VALUES OF \text{DELTA}(GEF) FOR USE IN OTHER PROGRAMS

45 \text{IF}(\text{LINES} \cdot \text{LT} \cdot 55) \quad \text{GO TO} \quad 47

46 \text{CALL ENDPG}

47 \text{CONTINUE}
LINES=_LINES+1.
IF (LINES .LE. 55) GO TO 54
CALL ENDPG
C
CALCULATE AND WRITE A COMPARISON OF HEAT VALUES OBTAINED DIRECTLY FROM
C
THE Y- PLOT WITH THOSE OBTAINED BY INTEGRATION OF THE CP CURVE
WRITE(3,117)
117 FORMAT(54H COMPARISON OF HEATS OBTAINED DIRECTLY FROM THE Y- PLOT
1/53H WITH THOSE OBTAINED BY INTEGRATION OF THE CP VALUES. //)
WRITE(3,118)
118 FORMAT(6X,2H T,5X,12H HEAT(YPLDT1),2X,12H YPLOT SLOPE,3X,17H HEAT
1(INTEGRATED),3X,16H H(YPLOT)-H(INT) //)
LINES=9
DU 43 I=1,NUM
DEV(I)=H(I)-DELTAH(I)
WRITE(3,119) T(I),H(I),SLOPE(I),DELTAH(I),DEV(I)
119 FORMAT(6X,F7.2,6X,F6.0,6X,F9.6,11X,F6.0,12X,F8.1)
LINES=_LINES+1.
IF(LINES .LT. 55) GO TO 43
CALL ENDPG
43 CONTINUE
44 WRITE(3,103)
LINES=_LINES+1.
IF(LINES .LE. 55) GO TO 44
CALL ENDPG
ENDFILE,14
GO TO 1
END
SUBROUTINE ENDPG

WRITES THE LAST LINE ON EACH PAGE AND THE PAGE HEADINGS FOR THE
FOLLOWING PAGE. THIS SUBROUTINE ALSO PEEPS TRACK OF THE PAGE NUMBERS.

DIMENSION NAME(2), T(500)
COMMON TINC, N, NUM, T, NAME, NPAGE, LINES, L, MM
WRITE(3, 110).

110 FORMAT(1H 750X, 50H OFFICIAL NSRDS EVALUATION, CERTIFIED NO MISTE
IAKS)
WRITE(3, 1002)
1002 FORMAT(1H11)
NPAGE = NPAGE + 1
WRITE(3, 1004) NAME, NPAGE
1004 FORMAT(47H EVALUATION OF THE THERMODYNAMIC PROPERTIES OF , 2A6, 5X,
16H PAGE , 12)
WRITE(3, 1003)
1003 FORMAT(1H10)
LINES = 2
RETURN
END
A-11

SUBROUTINE AREA(X, SUM)

INTEGRATES THE CP CURVE BY MEANS OF SIMPSON'S RULE

DIMENSION A(500), X(500), SUM(500), T(500), NAME(2)

COMMON TINC, N, NUM, T, NAME, NPAGE, LINES, L, MM

SUM=0.

IF(TINC .EQ. 25.) GO TO 4

DO 1 I=4, N+2
   A(I) = (100./6.) * (X(I-2) + 4. * X(I-1) + X(I))
   1 SUM(I) = SUM(I-2) + A(I)

DO 3 I=4, N+2
   II=(I/2)+1
   3 SUM(II) = SUM(I)
   NUM=II
   RETURN

DO 5 I=6, N+4
   A(I) = (100./12) * (X(I-4) + 4. * X(I-3) + 2. * X(I-2) + 4. * X(I-1) + X(I))
   5 SUM(I) = SUM(I-4) + A(I)

DO 6 I=6, N+4
   II=(I+5)/4
   6 SUM(II) = SUM(I)
   NUM=II
   RETURN

END
SUBROUTINE HILSE(F3)
C. CALCULATES THE GAS PHASE TABLE
DIMENSION S(100), C(100), T(100), NAME(2), NAME(2)
COMMON TINC, N, NUM, T, NAME, NPAG, LINES, L, MM
DATA R/1, 98717/ _ READ(2, 200) CORR
200 FORMAT(F10.5) 201 FORMAT(1H1) WRITE(3, 203) 203 FORMAT(1H1)
WRITE(3, 204) 204 FORMAT(66H INPUT DATA TO SUBROUTINE HILSE. COLUMNS AS LISTED IN HILSENRAH. //)
WRITE(3, 211) CORR
211 FORMAT(37H CORRECTION FACTOR FOR NEW AT. WT. = F10.5) //) WRITE(3, 205) 205 FORMAT(8X, 2H1, 6X, 2H 4, 8X, 2H 5, 9X, 2H 6 //) LINES = 12
I = 1 7 I = I + 1 READ(2, 206) T(I), S(I), C(I), HA(I)
206 FORMAT(1X, F7.2, 5X, F7.4, 5X, F7.4, 4X, F8.2) IF(T(I), EQ. 0.) GO TO 8
GO TO 7 3 MM = I - 1 DO 1 I = 1, MM
WRITE(3, 207) T(I), S(I), C(I), HA(I) 207 FORMAT(6X, F7.2, 3X, F7.4, 3X, F7.4, 4X, F8.2)
LINES = LINES + 1 IF(LINES .LT. 55) GO TO 1 CALL ENDPG
1 CONTINUE 2 WRITE(3, 203) LINES = LINES + 1 IF(LINES .LT. 55) GO TO 2 CALL ENDPG
WRITE(3, 208) 208 FORMAT(26H GAS PHASE SELECTED VALUES //)
WRITE(3, 209) 209 FORMAT(5X, 8H T DEG K, 8X, 3H CP, 6X, 11H H(T)-H(ST), 2X, 11H S(T)-S(ST), 16X, 4H DEF //)
LINES = 7 S298 = K * (S(I) + CORR) S2(I) = 0.0 3 S2(I) = (R * S(I)) - S298 + R * CORR DO 4 I = 1, MM
C(I) = R * C(I) H(I) = R * (HA(I) - HA(I)) F2(I) = (H(I)/T(I)) - S298 - S2(I) DO 5 I = 1, MM
WRITE(3, 210) T(I), C(I), H(I), S2(I), F3(I) 210 FORMAT(6X, F7.2, 7X, F6.3, 7X, F6.0, 7X, F6.3, 7X, F6.3)
A-13

IF(LINES .LT. 55) GO TO 5
CALL ENDPG

5 CONTINUE

5 WRITE((3,203))
LINES = LINES + 1
IF(LINES .LT. 55) GO TO 6
CALL ENDPG
RETURN
END
PROGRAM HILSE (INPUT, OUTPUT, TAPE 2=INPUT, TAPE 3=OUTPUT)
CORRECTION PROGRAM FOR HILSENRATH ET AL TABLES OF IDEAL GAS THERMODY-

DIMENSION T(100), S(100), C(100), H(100), S1(100), S2(10)

DATA R/1.98717/
C READ INPUT DATA
C READ (2,100) IDENT
C FORMAT (12A6)
IF (IDENT .EQ. 6H) CALL EXIT
READ (2,102) CORR
C FORMAT (F10.5)
I=1
I=I+1
READ (2,101) T(I), S(I), C(I), H(I)
C FORMAT (F11.3,F10.4,F12.4,F12.2)
IF (T(I) .EQ. 0.0) GO TO 2
GO TO 1
N=N-1
C WRITE TABLE HEADINGS AND INPUT DATA
WRITE (3,109) IDENT
C FORMAT (11H1,12A6)
WRITE (3,130)
C FORMAT (50HO INPUT DATA, COLUMNS AS LISTED IN HILSENRATH.)
WRITE (3,106)
C FORMAT (50HQ 1 4 5 6 )
WRITE (3,107)
C DO 3 I=1,N
WRITE (3,101) T(I), S(I), C(I), H(I)
3 CONTINUE
C CALCULATE CORRECTED VALUES
S298 = R*(S(I)+CORR)
S2(1) = 2.9
DO 10 I=2,N
S2(I) = (R*S(I))-S298 + R*CORR
10 CONTINUE
DO 4 I=1,N
C(I) = 3*C(I)
H1(I) = (H(I)-H(1))*R
F2(I) = (H1(I)/T(I))-S298-S2(I)
F3(I) = -F2(I)
4 CONTINUE
C WRITE OUTPUT
WRITE (3,109) IDENT
WRITE (3,104)
C FORMAT (50OH SELECTED VALUES...)
WRITE (3,108)
C FORMAT (100HO T GP HT-HST ST-SST -GFEF)
1 WRITE (3,107)
DO 5 I=1,N
WRITE (3,105) T(I), C(I), H(I), S2(I), F3(I)
5 CONTINUE
GO TO 5
END
PROGRAM S3LAWS (INPUT, OUTPUT, TAPE2=INPUT, TAPE 3=OUTPUT, TAPE 14)

C THIRD LAW CALCULATIONS FOR MONATOMIC SPECIES

DIMENSION IDENT(12), TT(200), FEFD(200), P(200), X(200), XX(200), T(200)

1, TQ(200), FEFQ(200), TR(200), FEFR(200), FF(200), PP(200), PPP(200), PD(2
200), B(200), DHST(200), DHD(200), Q(200), RES(200)

DATA R/1.98717/

C READ INPUT DATA
1 READ (2, 100) IDENT

100 FORMAT (12A6)

IF(IDENT .EQ. '6H )STOP

110 FORMAT(314)

IF(M .EQ. 1) GO TO 155

C READ T AND DELTA(GEF) AT EVEN TEMPERATURES

I=0

2 I=I+1

READ(2,114) TT(I), FEFD(I)

114 FORMAT(1X,F7.2,4X,F6.3)

IF(I .EQ. 1) GO TO 14

C FIND MELTING TEMPERATURE

IF(TT(I) .EQ. TT(I-1)) GO TO 16

14 IF(TT(I) .EQ. 0.0) GO TO 15

GO TO 2

16 TM=TT(I)

C IMELT=NUMBER OF VALUES OF T AND DELTA(GEF) UP TO THE MELTING POINT

IMELT=I-1

GO TO 14

C NN= TOTAL NUMBER OF VALUES OF T AND DELTA(GEF)

15 NN=I-1

C READ MEASURED PRESSURES. X AND XX ARE DUMMIES NOT USED IN THIS PROGRAM.

155 I=0

17 I=I+1

READ(2,101) T(I), P(I), X(I), XX(I)

101 FORMAT(F8.2,E12.4,F10.2,F8.2)

IF(T(I) .EQ. 0.0) GO TO 17

GO TO 18

C NUMX=NUMBER OF DATA POINTS

4 NUMX=I-1

IF(KK .EQ. 0) GO TO 18

C CONVERT MILLIMETERS TO ATMOSPHERES.

DO 19 I=1, NUMX

P(I)=P(I)/760.

19 CONTINUE

C BREAK THE DELTA(GEF) CURVE INTO TWO PARTS AT THE MELTING POINT.

18 DO 20 I=1, IMELT

TQ(I)=TT(I)

20 CONTINUE

IF(IA=1)

DO 21 I=IMELT+1, NN

TR(IA)=TT(I)

FEFR(IA)=FEFD(I)

IA=IA+1

21 CONTINUE

NA=NN-IMELT
FIND WHICH PART OF THE DELTA(gef) CURVE T IS IN AND INTERPOLATE

DO 22 I=1,NUMX
  TA=T(I)
  IF (TA .GT. TM) GO TO 23
  IF (TA .LE. TM) GO TO 200
200 CALL LAGINT(IMELT,TC,FEFQ,2,KX,TA,F)
  FF(I)=F
  GO TO 22
23 CALL LAGINT(NA,TR,FEFR,2,KX,TA,F)
  FF(I)=F
22 CONTINUE
C WRITE INPUT DATA
WRITE(3,102)
102 FORMAT(1H1)
WRITE(3,104)
104 FORMAT(1HO)
WRITE (3,103)
103 FORMAT (50HO INPUT DATA)
C NEXT STATEMENT AVOIDS PRINTING T AND DELTA(gef) AT EVEN TEMPERATURES
C MORE THAN ONCE FOR A GIVEN ELEMENT.
IF(M .EQ. 1) GO TO 25
WRITE (3,111)
111 FORMAT (8H T, DEG K, 1X, 11H DELTA(gef) )
WRITE(3,104)
DO 24 I=1,NN
  WRITE(3,114)TT(I),FEFD(I)
24 CONTINUE
WRITE(3,102)
25 WRITE(3,100) IDENT
WRITE(3,181)
181 FORMAT (50HO T, P, DELTA GEF)
DO 89 I=1,NUMX
  WRITE(3,105)T(I),P(I),FF(I)
89 CONTINUE
105 FORMAT (1X,F8.2,2X,E12.4,2X,F8.3)
WRITE(3,102)
C THIRD LAW CALCULATION OF DELTA H
SUM=0.0
SUM1=0.0
DO 6 I=1,NUMX
  PP(I)=ALOG10(P(I))
  PPP(I)=ALOG(P(I))
  PD(I)=-R*T(I)*PPP(I)
  B(I)=T(I)*FF(I)
  DHST(I)=PD(I)+B(I)
  SUM=SUM+DHST(I)
6 CONTINUE
ANUMX=NUMX
DHA=SUM/ANUMX
NUMX=ANUMX
DO 7 I=1,NUMX
  DHD(I)=DHST(I)-DHA
  SUM1=SUM1+ABS(DHD(I))
7 CONTINUE
ANUMX=NUMX
DEV=SUM1/ANUMX
NUMX=ANUMX
C PRINT RESULTS
WRITE (3,106)
106 FORMAT (5OHOCALCULATION RESULTS)
WRITE (3,107)
107 FORMAT (10OH T P LN P LOG P -RT
1 LN P -T(DELTAFEF) DELTA HST DEV
WRITE (3,104)
L=0
DO 8 I=1,NUMX
WRITE (3,108) T(I),P(I),PPP(I),PP(I),PD(I),B(I),DHST(I),DHD(I)
10.2,2X,F8.2)
C PUNCH OPTION
IF (K-1)'8,9,9
C TO AVOID PUNCHING IDENT BEFORE EACH VALUE
9 IF (L .EQ. 0) GO TO 12
GO TO 11
C PUNCH RESULTS
12 WRITE (14,100) IDENT
11 WRITE(14,101) T(I),P(I),PP(I),DHST(I),DHD(I)
L=1
8 CONTINUE
WRITE(3,109) DHAV.
109 FORMAT (2OHAVEAGE DELTA H298= F10.2)
WRITE(3,112) DEV.
112 FORMAT (19HOAVEAGE DEVIATION= F6.2)
CALCULATION OF STANDARD DEVIATION
SUM2 = 0.0
DO 13 I=1,NUMX
RES(I) = DHD(I)**2
SUM2=SUM2+RES(I)
13 CONTINUE
ANUMX=NUMX
SD = SQRT(SUM2/(ANUMX-1.0))
NUMX=ANUMX
WRITE(3,104)
WRITE(3,113)NUMX,SD
113 FORMAT(34H STANDARD DEVIATION FOR THE ABOVE ,I4, 16H MEASUREME
1NTS = ,F6.2)
ENDFILE 14
GO TO 1
END
SUBROUTINE LAGINI( N, A, F, M, K, X, Y )
DIMENSION A(1), F(1)
C DETERMINE INDEX OF TABLE VALUE NEAREST THE INTERPOLATE.
CALL BAINS( A, N, X, I )
IF( I.EQ.0 ) GO TO 10
IF( X.EQ.A(I) ) GO TO 9
C DETERMINE INDEX OF THE FIRST POINT TO BE USED IN THE INTERPOLATION.
J = MIN( MAX( I - M/2, 1 ), N - M )
IF( M.EQ.1 .AND. I.LE.N-1 .AND. X.LT.A(I) ) J = J - 1
CALL LAGRNG( A(J), F(J), M, X, Y )
10 K = 1
RETURN
9 Y = F(I)
GO TO 10
END
A-19

SUBROUTINE LAGRNG( A, F, M, X, Y )
DIMENSION A(11), F(11), DA(11), D(11,11)
C**94
DATA D(11),D(13),D(25),D(37),D(49),D(61),D(73),D(85),D(97),D(109),
10(121)/11*1.0/
C**94
15 L = M + 1
C FORM TABLES.
FN = 1.
DO 1 KK = 1, L
DA(KK) = A(KK) - X
IF( DA(KK).EQ.0.0 ) GO TO 7
FN = FN*DA(KK)
1 CONTINUE
KK = 1
LL = 2
2 DO 3 JJ = LL, L
D(KK,JJ) = DA(KK) - DA(JJ)
3 D(JJ,KK) = -D(KK,JJ)
KK = KK + 1
LL = KK + 1
IF( LL.LE.L ) GO TO 2
C COMPUTE INTERPOLATED VALUE.
Y = 0.
DO 6 KK = 1, L
FD = 1.
DO 5 JJ = 1, L
5 FD = FD*D(JJ,KK)
6 Y = Y + ( FN*F(KK) )/( DA(KK)*FD )
RETURN
7 Y = F(KK)
RETURN
END
SUBROUTINE BAINS( SLIST, MM, Z, K )
DIMENSION SLIST(1)
C SLIST=TABLE, WHICH MUST BE MONOTONICALLY INCREASING
C M= NUMBER OF ENTRIES IN SLIST
C Z=VALUE TO BE FOUND IN TABLE
C K=SUBSCRIPT OF VALUE IN TABLE NEAREST TO Z
C PROGRAM RETURNS K = 0 IF Z IF OFF TABLE.
M=MM
L1=1
L2=M
K=1
IF(Z-SLIST(1)) 1,15,3
3 K=M
IF(SLIST(M)-Z) 1,15,9
9 K=M/2
IF(Z-SLIST(K)) 20, 15, 9
20 L2=K
GO TO 23
29 L1=K
23 IF(L2-L1-1) 1,14,25
25 M=L1+L2
GO TO 9
14 IF(Z*Z-SLIST(L1)-SLIST(L2)) 30,15,31
30 K=L1
GO TO 15
31 K=L2
GO TO 15
1 K = 0
15 RETURN
END
PROGRAM VAPOR (INPUT,OUTPUT,TAPE 2=INPUT,TAPE 3=OUTPUT)
DIMENSION T(500),P(500),PL(500),PRD(500),DIV(500),IDENT(12),XAXIS
1(50),YAXIS(50)
DATA K/1.98717/
READ INP JD DATA
13 READ (2,1) IDENT
1 FORMAT (12A6)
IF (IDENT .EQ. '6H') CALL EXIT
READ (2,20) K
20 FORMAT (14)
WRITE (3,8) IDENT
3 FORMAT (1H1,12A6)
WRITE (3,3)
3 FORMAT (50H0 TEMP PRESSURE LOG P RLP 1/T )
WRITE (3,4)
4 FORMAT (50H (KELV IN) (ATM) ) I=
9 I=I+1
READ (2,5) T(I),P(I)
5 FORMAT (F8.2,E12.3)
IF (T(I) .EQ. 0.0) GO TO 10
GO TO 9
10 NUMX=I-1
IF (K .EQ. 0) GO TO 21
CONVERSION LOOP MILLIMETERS--ATMOSPHERES
DO 22 I=1,NUMX
22 P(I)=P(I)/760.0
21 DO 7 I=1,NUMX
PL(I)=ALOG10(P(I))
PRD(I)=R*ALOG(P(I))
DIV(I)=1./T(I)
WRITE OUTPUT
WRITE (3,6) T(I),P(I),PL(I),PRD(I),DIV(I)
5 FORMAT (2X,F6.1,2X,E9.3,1X,F10.5,2X,F7.3,2X,E10.3)
7 CONTINUE
GO TO 13
END
PROGRAM VPCALC(INPUT,OUTPUT,TAPE 2=INPUT,TAPE 3=OUTPUT)

CALCULATION OF VAPOR PRESSURES FROM FREE ENERGY FUNCTIONS

DIMENSION NAME(12),T(100),FEFD(100),X(100),DGT(100),Z(100),P(100),
1MZ(100),BZ(100),SZ(100),TQ(100),FQ(100),A(20),TQ(100),FFQ(100),
2TQ(100),FQ(100),TP(20),GAPP(50),TAPP(50),GB(300),TB(300)

DATA R/1.98717/

46 READ (2,100) NAME
100 FORMAT(12A6)

IF (NAME,EQ.,6HSTOP

WRITE(3,101) NAME
101 FORMAT (1H1,12A6)

WRITE(3,102)

WRITE(3,103)

103 FORMAT(50HO

READ(2,104) DHST

104 FORMAT(F10.0)

READ(2,116) TM,TTR

116 FORMAT(2(2X,F7.2))

WRITE (3,105) DHST

105 FORMAT(22MODELTA H(VAP),(298) = F10.0)

I=0

1 I=I+1

READ(2,106) T(I),FEFD(I)

106 FORMAT(1X,F7.2,4X,F6.3)

IF (T(I) .EQ. 0.0) GO TO 2

GO TO 1

C N=TOTAL NUMBER OF POINTS

2 N=I-1

WRITE (3,107)

107 FORMAT(8H T DEG K 1X,11H DELTA(GEF))

WRITE(3,102)

DO 3 I=1,N

WRITE(3,1060) T(I),FEFD(I)

1060 FORMAT(F12.2,2F12.3)

3 CONTINUE

WRITE (3,101) NAME

WRITE (3,110)

110 FORMAT(99HO T DGT P

1EXP )

C CALCULATE DELTA G

DO 16 I=1,N

X(I)=-FEFD(I)

DGT(I)=X(I)*T(I)+DHST

16 C Z(I) = ANTILOG P

Z(I)=-DGT(I)/(R*T(I)*ALOG(10.0))

IF(Z(I) LT. 0.0) GO TO 4

P(I)=10.0**Z(I)

C MZ = EXPONENT OF P

MZ(I)=0

GO TO 15

C CALCULATION OF P AND MZ IF Z IS NEGATIVE

4 BZ(I)=AINT(Z(I))

MZ(I)=INT(BZ(I))-1

SZ(I)=Z(I)-BZ(I)+1.C

P(I)=10.0**SZ(I)
PRINT RESULTS

WRITE(3,111) T(I),DG(T(I)),P(I),MZ(I)

111 FORMAT(6X,F7.2,9X,F8.0,9X,F7.4,9X,F5)

CONTINUE

WRITE(3,102)

FINO

THE TWO TEMPERATURES BETWEEN WHICH DELTA G CHANGES SIGN

IF (DG(T(I)) .LE. 0.0) GO TO 18

CONTINUE

ILOW=I-1

I(HIGH)=I

TLOW AND THIGH ARE THE DESIRED TEMPERATURES. THEY ARE 100 DEGREES APART.

TLOW=T(ILOW)

THIGH=T(I(HIGH))

GENERATE AN ARRAY OF DELTA(GEF) VALUES IN THE LIQUID

DO 19 I=1,N

IF(T(I)) .EQ. TM) GO TO 20

CONTINUE

IMELT=I

IMELT=IMELT+1

NLQ=N-IMELT

IA=1

DO 21 I=IMELT1,N

TG(IA)=T(I)

FQ(IA)=FQ(I)

IA=IA+1

CONTINUE

FIND T(BP) TO THE NEAREST 10 DEGREES

IF=1

CALL LAGINT(NLQ,TQ,FQ,2,KX,TLOW,F)

GAPP(IJ)=DMST-TLOW*F

TAPP(IJ)=TLOW

TLOW=TLOW+10.0

IF(TLOW .GT. THIGH) GO TO 23

I(J)=IJ+1

GO TO 22

FIND THE MINIMUM VALUE OF GAPP

GMIN=1.E100

DO 24 I=1,N

IF(ABS(GAPP(I)) .GE. GMIN) GO TO 24

GMIN=GAPP(I)

IMIN=I

CONTINUE

TMN=TAPP(IMIN) - 10.

TMAX=TMN+20.

J=1

CALL LAGINT(NLQ,TQ,FQ,2,KX,TMIN,F)

GB(J)=DMST-TMIN*F

TB(J)=TMIN

IF(TB(J) .GT. TMAX) GO TO 27

J=J+1

TMN=TMN+0.1

GO TO 26

JFINAL=J

GBP=1.E100
DC 28  I=1, JFINAL
IF(ABS(GB(I)) .GE. GBP) GO TO 28
GBP=GB(I)
IBP=1
28 CONTINUE
WRITE(3,112) NAME(1),NAME(2),TB(IBP),GBP
112 FORMAT(22H THE BOILING POINT OF 2A6, TH IS AT F7.1, 16H DEGREES K
IELVIN. /60H THIS VALUE OF T(BP) RESULTS IN A VALUE OF DELTA G EQUA
2L TO F4.1,1H.)
WRITE(3,113) NAME(1),NAME(2)
113 FORMAT(52H1 CALCULATION OF TEMPERATURES AT EVEN PRESSURES FOR 2A6
1)
WRITE(3,114)
114 FORMAT(50HO, -LOG P , T , DEG K)
C CALCULATION OF T AT EVEN PRESSURES
C CALCULATE (-R LN P) AT EVEN PRESSURES FROM P = 10**-1 TO 10**-10
Q=0.0
DC 49  J=1,10
Q=Q+1.0
A(J)=R*ALOG(10.0)*Q
49 CONTINUE
50 IF(ITR .EQ. 0.0) GO TO 51
C FOR A POLYMORPHIC METAL SPLIT THE DELTA(GEF) CURVE INTO 3 PARTS.
DC 29  I=1,N
IF(T(I) .EQ. TM) IM=I-1
IF(T(I) .EQ. TTR) ITR=I-1
29 CONTINUE
ITR=ITR+1
IM=IM+1
DC 30  I=1, ITR
TC(I)=T(I)
FC(I)=FEFD(I)
30 CONTINUE
J=0
DC 31  I=ITR1,IM
J=J+1
TQ(J)=T(I)
FFQ(J)=FEFD(I)
31 CONTINUE
JA=J
J=0
DC 32  I=IM1,N
J=J+1
TQ(J)=T(I)
FFQ(J)=FEFD(I)
32 CONTINUE
JJA=J
GC TO 36
C FOR A NON-POLYMORPHIC METAL SPLIT THE DELTA(GEF) CURVE INTO 2 PARTS.
51 DC 33  I=1,N
IF(T(I) .EQ. TM) IM=I-1
33 CONTINUE
IM=IM+1
DC 34  I=1,IM
TC(I)=T(I)
FC(I)=FEFD(I)
34 CONTINUE
J=0
DC 35 I=1M1,N
J=J+1
TTQ(J)=T(I)
FFQ(J)=FFD(I)
35 CONTINUE
JA=J
C FIND THE TWO TEMPERATURES BETWEEN WHICH THE DESIRED VALUE OF P LIES
36 J=1
Q=0.0
43 Q=Q+1.0
37 DC 52 I=1,N
IF(Q .GE. (-MZ(I))) GO TO 38
52 CONTINUE
38 TL=T(I-1)
TG=T(I)
C INTERPOLATE
IF(TTR .NE. 0.0) GO TO 33
41 IF(TL .GE. TM) CALL LAGINT(JA,TTQ,FFQ,2,KX,TL,G)
IF(TL .LT. TM) CALL LAGINT(IM,TQ,FQ,2,XX,TL,G)
C CALCULATE T
TEMP=DHST/(A(J)+G)
C TEST TO SEE IF THE CALCULATED VALUE OF T IS WITHIN 1 DEGREE OF THE
C INTERPOLATED VALUE
IF(ABS(TEMP-TL) .LE. 1.0) GO TO 40
C INCREMENT THE INTERPOLATED VALUE AND REPEAT THE PROCESS
TL=TL+0.1
GO TO 41
40 TP(J)=TEMP
J=J+1
IF(J .EQ. 11) GO TO 42
GO TO 47
43 IF(TL .LE. TTR) CALL LAGINT(IJ,TTQ,FFQ,2,XX,TL,G)
IF(TL .GT. TTR AND. TL .LE. TM) CALL LAGINT(JA,TTQ,FFQ,2,KX,TL,G)
IF(TL .GT. TM) CALL LAGINT(JA,TTQ,FFQ,2,XX,TL,G)
TEMP=DHST/(A(J)+G)
IF(ABS(TEMP-TL) .LE. 1.0) GO TO 44
44 TL=TL+0.1
GO TO 43
44 TP(J)=TEMP
J=J+1
IF(J .EQ. 11) GO TO 42
GO TO 47
C WRITE RESULTS
42 DC 45 J=1,10
WRITE(3,115) J,TP(J)
115 FORMAT(4X,13,8X,F6.1)
45 CONTINUE
GO TO 46
END
PROGRAM B3LAW (INPUT,OUTPUT,TAPE 2=INPUT,TAPE 3=OUTPUT)

THIRD LAW CALCULATION OF DELTA H FOR TWO VAPOR SPECIES I.E. SELENIUM

DIMENSION IDENT (12), T(50), P2(50), P6(50), FEF2(50), FEF6(50), PSQ2(50)
 1), PR6(50), PSQ2L(50), PR6L(50), PP(50), PPP(50), A(50), B(50), C(50), D(50)
 150), D+2(50), DH6(50), BB(50), DH2D(50), DH6D(50)

DATA K/1.98717/

READ INPUT DATA
 1 READ (2,2) IDENT
 2 FORMAT (12A6)
     I=
 3 I=1+1
 4 FORMAT (FB.2,2E12.6,2FB.2)
 5 GO TO 3

I=J
 6 READ (2,4) T(I), P2(I), P6(I), FEF2(I), FEF6(I)
 7 IF (T(I) .EQ. 0.0) GO TO 5
 8 GO TO 3

DATA K/1.98717/

READ INPUT DATA
 1 READ (2,2) IDENT
 2 FORMAT (12A6)
     I=
 3 I=1+1
 4 READ (2,4) T(I), P2(I), P6(I), FEF2(I), FEF6(I)
 5 IF (T(I) .EQ. 0.0) GO TO 5

WRITE TABLE HEADINGS
 6 WRITE (3,6) IDENT
 7 WRITE (3,7)
 8 WRITE (50H0DATA... T,P2,P6,FEF2,FEF6)
 9 WRITE (3,12)

WRITE INPUT DATA

DO 8 I=1,NUMX

WRITE (3,9) T(I), P2(I), P6(I), FEF2(I), FEF6(I)

WRITE (3,12)

CONTINUE

3RD LAW CALCULATION

SUM=0.0
 1 SUM=0.0
 2 DO 10 I=1,NUMX
 3 PSQ2(I)=SQRT(P2(I))
 4 BB(I)=(I.6)10*ALOG(P6(I))
 5 PR6(I)=EXP(BB(I))
 6 FEF2(I)=+FEF2(I)
 7 FEF6(I)=+FEF6(I)
 8 PSQ2L(I)=ALOG(PSQ2(I))
 9 PR6L(I)=ALOG(PR6(I))
 10 PPP(I)=ALOG10(PSQ2(I))

A(I)=-2*T(I)*PSQ2L(I)
B(I)=+T(I)*PR6L(I)
C(I)=-T(I)*FEF2(I)
D(I)=-T(I)*FEF6(I)

CALCULATE DELTA H VALUES

DH2(I)=A(I)+C(I)
 1 SUM=SUM+DH2(I)
 2 D(I)=B(I)+D(I)
 3 SUM=SUM1+DH6(I)

CONTINUE

TO AVERAGE THE DELTA H298 VALUES FOR EACH TEMPERATURE.

SUM2=0.0
 5 SUM3=0.0
 6 DO 17 I=1,NUMX
 7 ANJMX=NUMX

A-26
DH2AV = SUM/ANUMX  
DH5AV = SUM1/ANUMX  
DH2D(I) = DH2(I) - DH2AV  
SUM2 = SJM2 + ABS(DH2D(I))  
DH5D(I) = DH5(I) - DH5AV  
SUM3 = SJM3 + ABS(DH5D(I))  
17. CONTINUE  
ANJMX = NUMX  
DEV2 = SJM2/ANUMX  
DEV6 = SJM3/ANUMX  
WRITE RESULTS  
WRITE (3,6) IDENT  
WRITE (3,11)  
11 FORMAT (114HODIATOMIC RESULTS, T, P2, SQRT P2, LN(SQRT P2), LOG(SQRT P2 1), -RT LN(SQRT P2), DELTA(FEF), -T*DELTA(FEF), DELTA H298, DIFF )  
WRITE (3,12)  
DO 13 I = 1, NUMX  
WRITE (3,14) T(I), P2(I), PSQ2(I), PSQ2L(I), PP(I), A(I), FEF2(I), C(I), D 1H2(I), DH2D(I)  
14 FORMAT (F8.2, 2X, 2E12.4, 2F10.5, 2F8.2, 2X, F8.2, 2X, F8.2, 2X, F8.2)  
13 CONTINUE  
WRITE (3,18) DH2AV, DEV2  
18 FORMAT (24H0 AVERAGE DELTA H298 IS F8.2, 26H THE AVERAGE DEVIATION 1 IS, F8.2)  
WRITE (3,6) IDENT  
WRITE (3,15)  
15 FORMAT (114H0 HEXATOMIC RESULTS, T, P6, ROOT P6, LN ROOT P6, LOG ROOT 1P6, -RT LN (RT P6), DELTA(FEF), -T*DELTA(FEF), DELTA H298, DIFF )  
WRITE (3,12)  
DO 16 I = 1, NUMX  
WRITE (3,14) T(I), P6(I), PRT6(I), PRT6L(I), PPP(I), B(I), FEF6(I), D(I), 1DH5(I), DH6D(I)  
16 CONTINUE  
GO TO 1  
CALL EXIT  
END
PROGRAM ACTIV (INPUT, OUTPUT, TAPE 2 = INPUT, TAPE 3 = OUTPUT, TAPE 14)
CALCULATION OF ACTIVITIES FROM VAPOR PRESSURE MEASUREMENTS

DIMENSION IDENT(121, TT(100), FEFD(100), T(100), P(100), TQ(100), EFFQ(100), TRI(100), FEFR(100), FF(100), A(100), PP(100), ACT(100), G(100), DF(10)), DFX(100)

READ INPUT DATA
READ (2, 100) IDENT

10 FORMAT (12A6)
IF (IDENT .EQ. 6H) STOP
READ (2, 101) DHST
READ (2, 102) X
READ (2, 110) K, KK, M

IF (M .EQ. 1) GO TO 151

1 = 0
I = I + 1
READ (2, 114) TT(I), FEFD(I)

14 FORMAT (1X, F7.2, 4X, F6.3)
IF (I .EQ. 1) GO TO 10
IF (TT(I) .EQ. TT(I - 1)) GO TO 2
IF (TT(I) .EQ. 0.0) GO TO 11

GO TO 1

2 TMI = TT(I)
IMELT = I - 1
GO TO 10

11 NN = I - 1

151 I = 0

12 I = I + 1
READ (2, 103) T(I), P(I)

13 FORMAT (F8.2, E12.4, F7.3)
IF (T(I) .EQ. 0.0) GO TO 13

GO TO 12

13 N = I - 1
IF (KK .EQ. 0) GO TO 14

C CONVERSION FROM MILLIMETERS TO ATMOSPHERES
DD 15 I = 1, N
P(I) = P(I) / 760.
CONTINUE

14 DD 16 I = 1, IMELT
TQ(I) = TT(I)
FEFD(I) = FEFD(I)

CONTINUE

C DIVIDE THE DELTA(GEF) CURVE INTO TWO PARTS AT THE MELTING POINT
IMELT = IMELT + 1
IA = 1
HA = NN - IMELT
DO 17 I = IMELT + 1, NN
TR(IA) = TT(I)
FEFR(IA) = FEFD(I)
IA = IA + 1

CONTINUE

C INTERPOLATE IN THE PROPER PART OF THE CURVE TO OBTAIN THE VALUE OF
C DELTA(GEF) AT THE MEASURED TEMPERATURE
DO 20 I=1,N
I=1

IF (TA .GT. TM) GO TO 200
IF (TA .LE. TM) GO TO 300

300 CALL LAGINT(IMELT,TQ,FEFQ,2,KX,TX,F)
FF(I)=F
GO TO 20

200 CALL LAGINT(NA,TR,FEFR,2,KX,TX,F)
FF(I)=F

20 CONTINUE

21 WRITE(3,108)
108 FORMAT (1HL)

C WRITE INPUT DATA
WRITE(3,100) IDENT
WRITE(3,104)

104 FORMAT (1HO)
WRITE(3,105)

105 FORMAT (50HO INPUT DATA )
IF (M .EQ. 1) GO TO 39
WRITE(3,115)

115 FORMAT (8H T DEG K,IX,11H DELTA(GEF) )
WRITE(3,104)

DO 22 I=1,NN
WRITE(3,114) TT(I),FED(I)

22 CONTINUE

39 WRITE(3,104)
WRITE (3,113) DHST, X

113 FORMAT (15HO DELTA H298, = ,F10.1, 10H X = ,F10.7)
WRITE (3,106)

106 FORMAT (50HO T DEG K P (ALLOY) DELTA GEF )
WRITE (3,104)

DO 3 I=1,N
WRITE(3,107) T(I),P(I),FF(I)

107 FORMAT (4X,F7.2,5X,E12.4,5X,F7.3)
3 CONTINUE

C WRITE TABLE HEADINGS
WRITE (3,112)

112 FORMAT (90HO T DEG K P (ALLOY) P (METAL) ACTIVITY G)
WRITE (3,104)

L=0
CALCULATE ACTIVITIES, ACTIVITY COEFFICIENTS, AND GIBBS ENERGIES
DO 4 I=1,N
A(I)=(FF(I)-DHST/T(I))/(R*ALOG(10.0))

PPI(I)=10.0**A(I)
ACT(I)=P(I)/PP(I)

G(I)=ACT(I)/X
DF(I)=R*(T(I)*ALOG(ACT(I))
DFX(I)=R*T(I)*ALOG(G(I))

C WRITE RESULTS
WRITE (3,109) T(I),P(I),PP(I),ACT(I),G(I),DF(I),DFX(I)


C PUNCH OPTION
IF (K-1) 4,5,5

C NEXT STATEMENT AVOIDS WRITING THE TITLE BEFORE EACH VALUE OF T, P, ETC.
5 IF (L.EQ. 0) GO TO 6
    GO TO 7
6 WRITE (14,100) IDENT
7 WRITE(14,111) T(I),P(I),PP(I),ACT(I),G(I),DF(I),DFX(I)
111 FORMAT(1X,F7.2,1X,E12.4,1X, E12.4,1X,F5.3,1X,F7.3,1X,F9.1,1X,F9.1)
11 L=1
4 CONTINUE
    ENDFILE 14
    GO TO 8
END
PROGRAM ALLOY (INPUT, OUTPUT, TAPE 2=INPUT, TAPE 3=OUTPUT)
DIMENSION IDENT(12), NAME(2), ALPHI(30), DGX1(30), AA(30), GAM1(30),
1AGT(30),DGX2(30), GAM2(30), CT2(30), XAREA(30), X1(30), X2(30)
2), DH(30), DL(30), DH2(30), S(30), SX(30), DS1(30), DS2(30), R2X1(30),
3DSX2(30), GX(30), G(30), DGI(30), DG2(30), B(30), VAREA(30), ALPHI(30),
4G(30)
COMMON XAREA, X1, X2, DH, DL, DH2, S, SX, DS1, DS2, DSX1, DSX2, GX, G, DGI,
1DGX1, DS2, DGX2, B
DATA R/1.98717/
READ INPUT DATA
13 READ(2,100) IDENT
100 FORMAT(12A6)
READ(2,131) NAME(1), NAME(2), NSTAT
131 FORMAT(2A3,46)
READ (2,101) T,L
101 FORMAT(100,110)
DO 1 I=1,15,7
READ (2,102) ALPHI(I), ALPHI(I+1), ALPHI(I+2), ALPHI(I+3), ALPHI(I+4)
1, ALPHI(I+5), ALPHI(I+6)
1 CONTINUE
102 FORMAT(7F10.1)
CALCULATE X VALUES IN INTERVALS OF 0.05 FROM X = 0 TO X = 1.
X1(1) = 0.0
DO 2 I = 2,21
X1(I) = X1(I-1) + 0.05
2 CONTINUE
DO 3 I = 1,15,7
READ (2,102) B(I), B(I+1), B(I+2), B(I+3), B(I+4), B(I+5), B(I+6)
3 CONTINUE
WRITE TABLE HEADINGS
WRITE(3,103)
103 FORMAT(1H1)
WRITE(3,104) IDENT
104 FORMAT(12A5)
WRITE(3,105)
105 FORMAT(1H0)
WRITE(3,106) NAME(1), NAME(2), NSTAT, T
106 FORMAT(15H INPUT DATA FOR , A3,2H- , A3,15H ALLOYS IN THE ,
1A6,9H STATE AT , F6.0,15H DEGREES KELVIN )
WRITE(3,105)
WRITE(3,107)
107 FORMAT(50H, X1, ALPHI(I), B(I), , X2(I))
WRITE(3,105)
WRITE INPUT DATA
DO 4 I=1,21
X2(I) = 1.0-X1(I)
4 WRITE (3,108) X1(I), ALPHI(I), B(I), , X2(I)
108 FORMAT(1X,F9.4,2X,F8.0,6X,F12.4,6X,F4.2)
WRITE(3,105)
WRITE (3,116) NAME(1), NAME(2)
116 FORMAT (89H NOTE: 'FIRST' ELEMENT IS THE ONE GIVEN FIRST IN THE T
1ABLE HEADINGS, I.E., ELEMENT 1 IS , A3,17H AND ELEMENT 2 IS
2, A3)
WRITE(3,105)
C L IS A CONTROL NUMBER TELLING WHICH TYPE OF DATA IS AVAILABLE
IF (L-1) 41,42,43
A-32

41 WRITE(3,128) L
123 FORMAT(5H L = ,11,19H: Q PLOT DATA USED )
   GO TO 44
42 WRITE(3,129) L
129 FORMAT(5H L = ,11,22H: BETA PLOT DATA USED )
   GO TO 44
43 WRITE(3,130) L
130 FORMAT(5H L = ,11,24H: FREE ENERGY DATA ONLY )
44 WRITE(3,105)
   WRITE(3,132)
132 FORMAT(43H B IN ABOVE TABLE IS EITHER Q OR BETA.
   AREA INTEGRATES UNDER THE SELECTED CURVE
   CALL AREA(ALPH1,21)
C CALCULATE TABLES
   DO 5 I=1,21,2
   DGX1(I) = ALPH1(I)*X2(I)**2
   IF(X(I) - 0.05) 52,51,51
   51 DG1(I) = DGX1(I) + R*T*ALOG(X1(I))
   52 AA(I) = DGX1(I)/(R*T)
   GAM1(I) = EXP(AA(I))
   ACT1(I) = X1(I)*GAM1(I)
   DGX2(I) = -(ALPH1(I)*X1(I)*X2(I)) + XARE(A(I))
   IF(X2(I) - 0.05) 54,53,53
   53 DG2(I) = DGX2(I) + R*T*ALOG(X2(I))
   54 AA(I) = DGX2(I)/(R*T)
   GAM2(I) = EXP(AA(I))
   ACT2(I) = X2(I)*GAM2(I)
   GX(I) = X1(I)*DGX1(I) + X2(I)*DGX2(I)
5 CONTINUE
   DO 6 I=3,19,2
   6 G(I) = X1(I)*DG1(I) + X2(I)*DG2(I)
   IF (L-1) 7,8,9
C Q PLOT IS USED FOR Q PLOT DATA (HEATS)
7 CALL QPLOT(B,T,21)
   GO TO 9
C BETA IS USED FOR BETA-PLOT DATA (ENTROPIES)
8 CALL BETA (B,T)
9 IF (L-2) 15,10,15
C WRITE TABLE HEADINGS FOR INTEGRAL QUANTITIES (NO HEAT DATA)
10 WRITE(3,103)
   WRITE(3,104) IDENT
   WRITE(3,105)
   WRITE(3,106)
   WRITE(3,109)
   WRITE(3,110) NSTAT,T
109 FORMAT(35H: TABLE 1 )
   WRITE(3,110) NSTAT,T
110 FORMAT(25H: INTEGRAL QUANTITIES FOR ,A6,10H ALLOYS AT, F6.0,15H DE
   GREEs KELVIN
   WRITE(3,105)
   WRITE(3,111)
111 FORMAT(34H: X2 DELTA G DELTA GXS )
   WRITE(3,105)
   DO 11 I=3,19,2
   J=22-I
   11 WRITE(3,112) X2(J),S(J),GX(J)
112 FORMAT(1X,F3.1,6X,F8.0,6X,F8.0)
## Partial Molar Quantities

**Table 2**

<table>
<thead>
<tr>
<th>ACTIV</th>
<th>GAMMA</th>
<th>DELTA G</th>
<th>DELTA GBAR</th>
<th>DELTA GBARXS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Format**: (1X,F3.1,4X,F8.0,5X,F8.0,5X,F7.4,7X,F8.0,7X,F7.4)

**Part 1**

- DO 12 J=1,9,2
- WRITE (3,112) XA,XX,GAM1(I),DGX2(J)
- WRITE (3,113) XA,XX,GAM1(I),DGX2(J)
- WRITE (3,114) XA,XX,GAM1(I),DGX2(J)
- WRITE (3,115) XA,XX,GAM1(I),DGX2(J)

**Part 2**

- DO 16 I=3,19,2
- WRITE (3,103) IDENT
- WRITE (3,104) IDENT
- WRITE (3,105) IDENT
- WRITE (3,109) IDENT
- WRITE (3,110) IDENT
- WRITE (3,111) IDENT

**Part 3**

- WRITE (3,121) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,122) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,123) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,124) X2(1),G(J),DH(J),S(J),DX(J)

**Part 4**

- WRITE (3,121) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,122) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,123) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,124) X2(1),G(J),DH(J),S(J),DX(J)

**Part 5**

- WRITE (3,121) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,122) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,123) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,124) X2(1),G(J),DH(J),S(J),DX(J)

**Part 6**

- WRITE (3,121) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,122) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,123) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,124) X2(1),G(J),DH(J),S(J),DX(J)

**Part 7**

- WRITE (3,121) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,122) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,123) X2(1),G(J),DH(J),S(J),DX(J)
- WRITE (3,124) X2(1),G(J),DH(J),S(J),DX(J)
WRITE(3,103)
WRITE(3,117)
WRITE(3,105)
WRITE(3,113) NSTAT,T
WRITE(3,105)
WRITE(3,123) NAME(1)
123 FORMAT(27X,A3,11H COMPONENT)
WRITE(3,105)
WRITE(3,124)
124 FORMAT(94H XL ACTIV GAMMA DELTA GBAR DELTA GBARXS DE )
WRITE(3,105)
WRITE(PARTIAL MOLAR TABLE FOR COMPONENT 1 IF HEAT OR BETA DATA ARE
C AVAILABLE
WRITE(3,126) XA,XA,XX,XX,XX,XX,XX
125 FORMAT(1X,F3.1,3X,F7.5,3X,F7.5,3X,8H INFIN.,5X,F8.0,7X,F8.0,7X,7H
1 INFIN.,7X,F7.4)
130 DO 19 I=3,19,2
J=22-I
19 WRITE(3,126) XL,J),ACT1(J),GAM1(J),DG1(J),DGX1(J),DH1(J),DS1(J),
1DSX1(J)
125 FORMAT(1X,F3.1,3X,F7.5,3X,F7.5,3X,F8.0,5X,F8.0,7X,F8.0,7X,F7.4,7X,
1F7.4)
1=1
17 WRITE(3,125) XX,XX,GAM1(I),DGX1(I),DH1(I),DSX1(I)
WRITE(3,105)
WRITE(PARTIAL MOLAR TABLE AND HEADINGS FOR COMPONENT 2 IF HEAT OR
C BETA DATA ARE AVAILABLE
WRITE(3,123) NAME(2)
WRITE(3,105)
WRITE(3,133)
133 FORMAT(94H X2 ACTIV GAMMA DELTA GBAR DELTA GBARXS DE )
WRITE(3,105)
1=21
WRITE(3,127) XX,XX,GAM2(I),DGX2(I),DH2(I),DSX2(I)
DQ 20 I=3,19,2
J=22-I
20 WRITE(3,126) X2(J),ACT2(J),GAM2(J),DG2(J),DGX2(J),DH2(J),DS2(J),
1DSX2(J)
XX=0.0
XA=1.00
WRITE(3,126) XA,XA,XX,XX,XX,XX,XX
127 FORMAT(1X,F3.1,3X,F7.5,3X,F7.5,3X,8H INFIN.,5X,F8.0,7X,F8.0,7X,7H
1 INFIN.,7X,F7.4)
GO TO 13
CALL EXIT
END
SUBROUTINE AREA(ALPH,N)

INTEGRATES THE SELECTED CURVE TO OBTAIN THE AREA UNDER IT BY MEANS
OF SIMPSON'S RULE

DIMENSION IDENT(12),NAME(2),ALPH(30),DGX1(30),AA(30),GAM1(30),
1ACT(30),DGX2(30),
GAM2(30),ACT2(30),XAREA(30),X1(30),X2(30)
2),DH(30),DH1(30),DH2(30),S(30),SX(30),DS1(30),DS2(30),DSX1(30),
3DSX2(30),GX(30),G(30),DG1(30),DG2(30),B(30),YAREA(30),ALPH(30),
4\(30)

COMMON XAREA,X1,X2,DH,DH1,DH2,S,SX,DS1,DS2,DSX1,DSX2,GX,G,DSX1

DGX1,DG2,DGXX2,B

SUM=0.0

XAREA(1) = 0.0

DO 1 I=3,N,2

A=I-1

YAREA(I) = ((X1(I)-X1(I))/3.0)*(ALPH(I)+4.0*ALPH(I-1)+ALPH(I-2))

SUM = SUM + YAREA(I)

XAREA(I) = SUM

1 CONTINUE

RETURN

END
SUBROUTINE QPLOT(B,T,N)
C  CALCULATES TABLES IF HEAT DATA ARE AVAILABLE
DIMENSION IDENT(12),NAME(2),ALPH(30),DGX1(30),AA(30),GAM1(30),
1 ACT(30),DGX2(30),_GAM2(30),ACT2(30),XAREA(30),X1(30),X2(30)
2, DH(30), DH1(30), DH2(30), S(30), SX(30), DS1(30), DS2(30), DSX1(30),
3 DSX2(30), GX(30), G1(30), DG1(30), DG2(30), B(30), YAREA(30), ALPH(30),
4 G1(30)
COMMON XAREA,X1,X2,DH,DH1,DH2,S,SX,DS1,DS2,DSX1,DSX2,GX,G,DG1,
1 DGX1,DGX2,DGX2
DO 3 I=1,19,7
3 READ (2,102) DH1(I), DH1(I+1), DH1(I+2), DH1(I+3), DH1(I+4), DH1(I+5),
1 DH1(I+5)
DO 11 I=1,N,2
11 DH(I) = B(I)*X1(I)*X2(I)
IF (X2(I) .LT. 0.05) 5,6,6
5 DH2(I) = (DH(I)-X1(I)*DH1(I))/X2(I)
6 IF (X1(I)) .LT. 0.95) 7,8,8
7 DH1(I) = B(I)
8 IF (X2(I) .LT. 0.95) 9,10,10
9 DH2(I) = B(I)
10 IF (X1(I) .LT. 0.95) 12,12,11
11 DH1(I) = 0.0
12 IF (X2(I) .LT. 0.95) 14,14,13
13 DH2(I) = 0.0
14 SX(I) = (DH(I) - GX(I))/T
15 DS1(I) = (DH1(I) - DG1(I))/T
16 DSX1(I) = (DH1(I) - DGX1(I))/T
17 DS2(I) = (DH2(I) - DG2(I))/T
18 DSX2(I) = (DH2(I) - DGX2(I))/T
1 CONTINUE
102 FORMAT(7F10.1)
RETURN
END
SUBROUTINE BETA (B,T)
CALCULATES TABLES IF BETA DATA ARE AVAILABLE.
DIMENSION IDENT(12),NAME(2),ALPHI(30),DGXI(30),AA(30),GAM1(30),
ACTI(32),DGX2(30),GAM2(30),ACT2(30),XAREA(30),X1(30),X2(30)
2),DH(3),DH1(30),DH2(30),S(30),SX(30),DS1(30),DS2(30),DSXI(30),
DSX2(30),G(30),DG1(30),DG2(30),B(30),YAREA(30),ALPH(30),
4U(30)
COMMON XAREA,X1,X2,DH,DH1,DH2,S,SX,DS1,DS2,DSXI,DSX2,SX,G,DG1,
1DGX1,DG2,DGX2
DATA /1.98717/
CALL AREA(B,21)
DO 1 I=1,21,2
1 DSXI(I) = B(I)*X2(I)**2
IF(X1(I) - 0.05) 2,3,3
3 DS1(I) = DSXI(I) - R*ALOG(X1(I))
2 DSX2(I) = -B(I)*X1(I)*X2(I)*XAREA(I)
IF(X2(I) - 0.05) 4,5,5
5 DS2(I) = DSX2(I) - R*ALOG(X2(I))
4 S(I) = X1(I)*DS1(I) + X2(I)*DS2(I)
SX(I) = X1(I)*DSX1(I) + X2(I)*DSX2(I)
DH1(I) = DGXI(I) + T*DSXI(I)
DH2(I) = DGX2(I) + T*DSX2(I)
1 DH(I) = X1(I)*DH1(I) + X2(I)*DH2(I)
RETURN
END
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.