ACCELERATOR DEVELOPMENT DEPARTMENT

BROOKHAVEN NATIONAL LABORATORY
Associated Universities, Inc.
Upton, New York 11973

SSC TECHNICAL NOTE NO. 48

SSC-N-173

Implementation of the NBS631 Helium Property Program for the IBM Personal Computer

K.C. Wu

May 2, 1986

Implementation of the NBS631 Helium Property Program*[1] for the IBM Personal Computer

Abstract

With the advancement of today's computer technology, PCs (Personal Computers) that have amazing computing power are available at affordable prices. These PCs have been used in every discipline and are available in most offices. The ability to implement computer programs onto PC hardware is well appreciated. In this technical note the implementation of the NBS631 helium property for an IBM PC is given. It should be noted that it is not the intent of this work to provide a program that serves as an "industrial standard" for the famous NBS631 helium properties which colud and should only be provided by the originator at NBS. Rather the present work allows one to implement the NBS631 program on PC with accuracy and enhanced speed. Thus simulation of a cryogenic system can be easily performed on PC.

1. Background

The names of the subroutines/functions of the NBS631 program together with their descriptions are given in Table 1. It consists of 29 routines with a total of approximately 1600 lines of Fortran source code. It is not a large program, but this program runs either very slowly or at a reasonable speed on a PC depending on how the algebra involved is performed, i.e., the efficiency of the algorithm to calculate the same set of equations.

Among the 29 routines in Table 1, one can roughly divide them into two categories. The first category expresses the thermodynamic property explicitly as a function of temperature and/or density. The second category calculates the thermodynamic property through interpolation of the first

*The copy of NBS631 helium property program available at BNL derives from the work originally done at NBS. It runs very quickly on large mainframe computers and agrees well, although not exactly, with the NBS631 publication.

category. In this technical note, improvement over routines belonging to the first category are limited to the replacement of the time consuming algebra to faster executing equivalent expressions. Minor addi tion/ alteration has been added to routines belonging to the second category for better convergence in the iteration. In either case, changes have been kept at minimum.

2. Modification of Fortran Languages to PC Fortran

To run a computer program written for a large mainframe computer on a PC is pretty much the same as implementing a program from mainframe to a mini-computer. It is beneficial to compile individual routines separately and then link them to form an executable file. The program must also be modified to take care of the variations among different Fortran Languages.

In this study, Microsoft PC Fortran[2] is used. In the original NBS631 program, the "ENTRY" statement is not supported by the Microsoft Fortran. Thus the original subroutine DTPRESS with ENTRIES DPDT, DPDD, ENTROP, ENTHAL and CV becomes six individual subroutines and DSATV and DSATL become two routines.

The block data subroutine must be modified so that the DATA statement in that routine is in a form accepted by Microsoft Fortran. Note that a dummy statement:

```
COMMON/COEFF/TESTAR(81),
```

must also be provided in the main program to activate the Block data subroutine.

The two branch arithemetic IF statements are another example of code not accepted by Microsoft Fortran and must be changed to appropriate equivalents.

Trivial discrepancy in the I/O formats have also been modified.

Table I. Descriptions of the NBS631 Routines

Routine	Math. Notation	Description
VPA(T)	Pv(T)	Vapor pressure as a function of temperature.
DPDTVP(T)	dPv/dT	Derivative of vapor pressure as a function of temperature.

VPTEMP(P)	Tv(P)	Vapor temperature as a function of pressure.
DSATV(T)	$\rho_v(T)$	Vapor density as a function of temperature.
DSATL(T)	$\rho_l(T)$	Liquid density as a function of temperature.
PMELT(T)	Pmelt(T)	Melting pressure as a function of temperature.
PLAMDA(T)	$P_\lambda(T)$	Lamda pressure as a function of temperature.
VIRB(T)	$B_2(T)$	Second virial coefficient as a function of temperature.
DBDT(T)	dB_2/dT	First derivative of second virial coefficient.
D2DBDT(T)	d^2B_2/dT^2	Second derivative of second virial coefficient.
DTPRESS(D,T)	$P(\rho,T)$	Pressure as a function of density and temperature.
DPDT(D,T)	dP/dT	Pressure derivative with respect to temperature.
DPDD(D,T)	$dP/d\rho$	Pressure derivative with respect to density.
ENTROP(D,T)	$s(\rho,T)$	Entropy as a function of density and temperature.
ENTHAL(D,T)	$h(\rho,T)$	Enthalpy as a function of density and temperature.
CV(D,T)	$Cv(\rho,T)$	Specific heat at constant volume as a function of density and temperature.
CP(D,T)	$Cp(\rho,T)$	Specific heat at constant pressure as a function of density and temperature.
VISCDT(D,T)	$\mu(\rho,T)$	Viscosity as a function of density and temperature.
THCON(D,T)	$k(\rho,T)$	Thermal conductivity as a function of density and temperature.

FINDD(P,T)	ρ(P,T)	Density as a function of pressure and temperature.
FINDH(P,T)	h(P,T)	Enthalpy as a function of pressure and temperature.
VPH(P,H)	v(P,h)	Specific volume as a function of pressure and enthalpy.
TPH(P,H)	T(P,h)	Temperature as a function of pressure and enthalpy.
HPS(P,S)	h(P,s)	Enthalpy as a function of pressure and entropy.
PHTD(P,H,T,D)	T(P,h) ρ(P,h)	Evaluate temperature and density at given pressure and enthalpy.
PSTD(P,S,T,D)	T(P,s) ρ(P,s)	Evaluate temperature and density at given pressure and entropy.
FLIQPH(P,H)	x(P,h)	Liquid fraction as a function of pressure and enthalpy.
FLIQPS(P,S)	x(P,s)	Liquid fraction as a function of pressure and entropy.
BLKDAT		Block data of 81 coefficients.

3. Reduction of the Time Consuming Computation

The computing speed of the arithmetic operation in a PC is much slower than a large computer. The use of a floating point processor can increase the computing speed. Nevertheless, the computation of raising a real number to some power is slow. Reduction of the arithmetic operations associated with this exponentiation operation can lead to significant savings in the computing time. This is true for a program to be run on a PC or a large computer. However since the overall running time on a large computer is short, there is not as great an improvement when using an existing program.

Examinations of the NBS631 program, it is found that there are approximately 300 exponentiation operations which can be replaced by the faster arithmetic operation such as substitution, multiplication or division. Two examples are given below.

3.1. Fortran source program

```
      N=9
      DO 3 I=1,4
      FI=I
      A(N)=D4*T**(1.5-FI)
    3 N=N+1
```

has been modified to

```
      T12=SQRT(T)
      A(9)=D4*T12
      A(10)=D4/T12
      A(11)=A(10)/T
      A(12)=A(11)/T
```

3.2 Fortran source program

```
      VIRB=0
      DO 5 I=1,9
      FI=I
    5 VIRB=VIRB+T**(1.5-FI/2.)*A(I)
```

has been modified to

```
      T12=SQRT(T)
      VIRB=A(1)*T+A(2)*T12+A(3)
           +(A(4)+A(6)/T+A(8)/T/T)/T12
           +(A(5)+A(7)/T+A(9)/T/T)/T
```

In this process, statement functions

```
UFT(XX)=(1.-XX)*T**XX
TFT(XX)=(XX+1.)*T**XX
SFT(XX)=XX*(XX+1.)*T**XX
```

have also been eliminated.

Depending on the application, up to 40% savings in the computing time has been observed.

4. Interpolation

The NBS631 program uses Newton-Raphson's method in routines FINDD, PHTD and PSTD. The results converge quite fast over almost the whole range of pressures and temperatures. However once in a while, the iteration will fail near the critical point (2.245 atm and 5.201 k). A slow but stable Bisection iterative method has been added into the above routines. Should Newton-Raphson's method fail to converge, the Bisection method will be used.

5. End and odds

The numerical precision in a PC, typically six decimal digits, is much less than the CDC 7600 at BNL so a different convergence criteria should be

used. If a 10^{-4} convergent parameter is chosen instead of 10^{-7} in the original program, there is negligible difference in the results.

Round off error is of concern when replacing the time consuming arithmetic operation by the faster computation, but only negligible differences have been found.

A test program computing density, pressure, enthalpy, entropy, specific heat at constant volume and pressure derivatives at five pressures and twenty seven temperatures are given in Appendix A. As one can see, the results agree with the NBS631 publication quite well with the largest discrepency less than 0.5%. The source program of the current modification is given in Appendix B.

6. Remarks

The NBS631 program has been successfully implemented on the IBM-PC. A few remarks on the hardware and software requirements are given below.

It is recommended that the IBM-PC-XT, AT, equivalent or better should be used. 640 K RAM memory and hard disk are needed. The 8087 floating point processor or equivalent is also beneficial to the current program.

The Microsoft Fortran is chosen for the current work. Other PC Fortran should also be applicable. Compilation of a program in the Microsoft Fortran compiler, is a of two- or three-step process. The two-step process compile a Fortran file with .FOR extension to a object file with .OBJ extension. The command to compile FINDD routine is given below.

6.1 Two-pass compilation of FINDD

```
C>for1

     Microsoft FORTRAN77 V3.20 02/84
     (C) Copyright Microsoft Corp. 1982, 1983, 1984

     Source filename [.FOR]: findd3
     Object filename [FINDD3.OBJ]:
     Source listing  [NUL.LST]: findd3
     Object listing  [NUL.COD]:
          Pass One   No Errors Detected
                    133 Source Lines

C>pas2

     Code Area Size = #0C87    (3207)
     Cons Area Size = #00F4    ( 244)
     Data Area Size = #0318    ( 792)
          Pass Two  No Errors Detected.
```

After all the object modules are created through the two steps given above, one needs to LINK these modules together with appropriate mathematic library to form an executable program with extension .EXE. The linking of TEST program with NBS routines and the 8087 mathematic library is shown below. After the completion of the linking process, the command TEST will start execution of the TEST program.

6.2 Linking of TEST program with the NBS631 routines

```
C>link test+@nbslink3,,,8087

Microsoft 8086 Object Linker
Version 2.41 (C) Copyright Microsoft Corp 1983

Object Modules [.OBJ]: CP+VISCDT+FINDH+HRTR+THCON+PHTD3+TPH+
Object Modules [.OBJ]: HPS+DPDD2+VPTEMP+DPDTVP2+FLIQPH+FLIQPS+
Object Modules [.OBJ]: ENTROP2+DTPRES2+DPDT2+VPH+DSATL2+BLKDAT+CV2+
Object Modules [.OBJ]: VPA2+DSATV2+D2DBDT2+DBDT2+COMPWK+VIRB2+EXPTTH+
Object Modules [.OBJ]: HXEFT+FINDD3+ENTHAL2+PSTD3+PMELT+PLAMDA
Run File [TEST.EXE]:
```

References

1. R.D. McCarty, NBS Technical Note 631, U.S. Dept. of Commerce, November 1972.

2. Microsoft Fortran Compiler for the MS-DOS Operating System, Microsoft Corp.

APPENDIX A

TEST RESULTS FOR HELIUM PROPERTIES

p	t	d	p2	h	s	cv	dpdd	dptt
atm	k	g/cc	atm	j/g	j/g-k	j/g-k	cc-atm/g	atm/k
.5	3.0	.142E+00	.499E+00	.537E+01	.237E+01	.192E+01	.366E+03	.341E+01
.5	4.0	.717E-02	.500E+00	.326E+02	.100E+02	.321E+01	.577E+02	.153E+00
.5	5.0	.533E-02	.500E+00	.386E+02	.114E+02	.314E+01	.854E+02	.112E+00
.5	6.0	.430E-02	.500E+00	.442E+02	.124E+02	.312E+01	.110E+03	.897E-01
.5	7.0	.362E-02	.500E+00	.497E+02	.132E+02	.311E+01	.133E+03	.753E-01
.5	8.0	.313E-02	.500E+00	.551E+02	.139E+02	.311E+01	.156E+03	.651E-01
.5	9.0	.276E-02	.500E+00	.605E+02	.146E+02	.311E+01	.178E+03	.573E-01
.5	10.0	.247E-02	.500E+00	.658E+02	.151E+02	.311E+01	.199E+03	.513E-01
.5	11.0	.224E-02	.500E+00	.711E+02	.156E+02	.311E+01	.221E+03	.464E-01
.5	12.0	.205E-02	.500E+00	.763E+02	.161E+02	.312E+01	.242E+03	.424E-01
.5	13.0	.189E-02	.500E+00	.816E+02	.165E+02	.312E+01	.263E+03	.391E-01
.5	14.0	.175E-02	.500E+00	.868E+02	.169E+02	.312E+01	.284E+03	.362E-01
.5	15.0	.163E-02	.500E+00	.921E+02	.173E+02	.312E+01	.305E+03	.337E-01
.5	16.0	.153E-02	.500E+00	.973E+02	.176E+02	.312E+01	.326E+03	.316E-01
.5	17.0	.144E-02	.500E+00	.103E+03	.179E+02	.312E+01	.347E+03	.297E-01
.5	18.0	.136E-02	.500E+00	.108E+03	.182E+02	.312E+01	.368E+03	.280E-01
.5	19.0	.128E-02	.500E+00	.113E+03	.185E+02	.312E+01	.389E+03	.265E-01
.5	20.0	.122E-02	.500E+00	.118E+03	.188E+02	.312E+01	.409E+03	.252E-01
.5	21.0	.116E-02	.500E+00	.123E+03	.190E+02	.312E+01	.430E+03	.239E-01
.5	22.0	.111E-02	.500E+00	.129E+03	.193E+02	.312E+01	.451E+03	.228E-01
.5	23.0	.106E-02	.500E+00	.134E+03	.195E+02	.312E+01	.472E+03	.218E-01
.5	24.0	.102E-02	.500E+00	.139E+03	.197E+02	.312E+01	.492E+03	.209E-01
.5	25.0	.975E-03	.500E+00	.144E+03	.199E+02	.312E+01	.513E+03	.201E-01
.5	26.0	.938E-03	.500E+00	.150E+03	.201E+02	.312E+01	.534E+03	.193E-01
.5	27.0	.903E-03	.500E+00	.155E+03	.203E+02	.312E+01	.554E+03	.186E-01
.5	28.0	.870E-03	.500E+00	.160E+03	.205E+02	.312E+01	.575E+03	.179E-01
.5	29.0	.840E-03	.500E+00	.165E+03	.207E+02	.312E+01	.596E+03	.173E-01
.5	30.0	.812E-03	.500E+00	.170E+03	.209E+02	.312E+01	.616E+03	.167E-01

p	t	d	p2	h	s	cv	dpdd	dptt
atm	k	g/cc	atm	j/g	j/g-k	j/g-k	cc-atm/g	atm/k
1.0	3.0	.143E+00	.100E+01	.565E+01	.234E+01	.191E+01	.395E+03	.345E+01
1.0	4.0	.130E+00	.100E+01	.871E+01	.321E+01	.227E+01	.189E+03	.377E+01
1.0	5.0	.119E-01	.100E+01	.362E+02	.961E+01	.318E+01	.660E+02	.259E+00
1.0	6.0	.915E-02	.100E+01	.425E+02	.108E+02	.313E+01	.962E+02	.195E+00
1.0	7.0	.753E-02	.100E+01	.483E+02	.117E+02	.311E+01	.123E+03	.159E+00
1.0	8.0	.644E-02	.100E+01	.540E+02	.124E+02	.311E+01	.147E+03	.136E+00
1.0	9.0	.564E-02	.100E+01	.595E+02	.131E+02	.311E+01	.171E+03	.118E+00
1.0	10.0	.502E-02	.100E+01	.649E+02	.136E+02	.311E+01	.193E+03	.105E+00
1.0	11.0	.453E-02	.100E+01	.703E+02	.142E+02	.311E+01	.216E+03	.949E-01
1.0	12.0	.413E-02	.100E+01	.757E+02	.146E+02	.311E+01	.238E+03	.864E-01
1.0	13.0	.380E-02	.100E+01	.810E+02	.150E+02	.312E+01	.260E+03	.793E-01
1.0	14.0	.352E-02	.100E+01	.863E+02	.154E+02	.312E+01	.281E+03	.733E-01
1.0	15.0	.328E-02	.100E+01	.916E+02	.158E+02	.312E+01	.303E+03	.682E-01
1.0	16.0	.307E-02	.100E+01	.969E+02	.161E+02	.312E+01	.324E+03	.638E-01
1.0	17.0	.288E-02	.100E+01	.102E+03	.165E+02	.312E+01	.346E+03	.599E-01
1.0	18.0	.272E-02	.100E+01	.107E+03	.168E+02	.312E+01	.367E+03	.564E-01
1.0	19.0	.257E-02	.100E+01	.113E+03	.171E+02	.312E+01	.388E+03	.534E-01
1.0	20.0	.244E-02	.100E+01	.118E+03	.173E+02	.312E+01	.409E+03	.506E-01
1.0	21.0	.232E-02	.100E+01	.123E+03	.176E+02	.312E+01	.430E+03	.482E-01
1.0	22.0	.222E-02	.100E+01	.128E+03	.178E+02	.312E+01	.451E+03	.459E-01
1.0	23.0	.212E-02	.100E+01	.134E+03	.181E+02	.312E+01	.472E+03	.439E-01
1.0	24.0	.203E-02	.100E+01	.139E+03	.183E+02	.312E+01	.493E+03	.420E-01
1.0	25.0	.195E-02	.100E+01	.144E+03	.185E+02	.312E+01	.513E+03	.403E-01
1.0	26.0	.187E-02	.100E+01	.149E+03	.187E+02	.312E+01	.534E+03	.387E-01
1.0	27.0	.180E-02	.100E+01	.155E+03	.189E+02	.312E+01	.555E+03	.373E-01
1.0	28.0	.174E-02	.100E+01	.160E+03	.191E+02	.312E+01	.576E+03	.359E-01
1.0	29.0	.168E-02	.100E+01	.165E+03	.193E+02	.312E+01	.597E+03	.347E-01
1.0	30.0	.162E-02	.100E+01	.170E+03	.194E+02	.312E+01	.617E+03	.335E-01

p	t	d	p2	h	s	cv	dpdd	dptt
atm	k	g/cc	atm	j/g	j/g-k	j/g-k	cc-atm/g	atm/k
1.5	3.0	.145E+00	.150E+01	.592E+01	.231E+01	.190E+01	.424E+03	.347E+01
1.5	4.0	.132E+00	.150E+01	.886E+01	.315E+01	.226E+01	.222E+03	.390E+01
1.5	5.0	.213E-01	.150E+01	.329E+02	.832E+01	.327E+01	.418E+02	.481E+00
1.5	6.0	.148E-01	.150E+01	.406E+02	.972E+01	.315E+01	.815E+02	.323E+00
1.5	7.0	.118E-01	.150E+01	.469E+02	.107E+02	.312E+01	.112E+03	.254E+00
1.5	8.0	.994E-02	.150E+01	.528E+02	.115E+02	.311E+01	.139E+03	.212E+00
1.5	9.0	.863E-02	.150E+01	.585E+02	.122E+02	.311E+01	.164E+03	.183E+00
1.5	10.0	.765E-02	.150E+01	.641E+02	.127E+02	.311E+01	.188E+03	.162E+00
1.5	11.0	.688E-02	.150E+01	.696E+02	.133E+02	.311E+01	.211E+03	.145E+00
1.5	12.0	.625E-02	.150E+01	.750E+02	.137E+02	.311E+01	.234E+03	.132E+00
1.5	13.0	.574E-02	.150E+01	.804E+02	.142E+02	.312E+01	.256E+03	.121E+00
1.5	14.0	.531E-02	.150E+01	.858E+02	.146E+02	.312E+01	.279E+03	.111E+00
1.5	15.0	.493E-02	.150E+01	.911E+02	.149E+02	.312E+01	.301E+03	.104E+00
1.5	16.0	.461E-02	.150E+01	.965E+02	.153E+02	.312E+01	.322E+03	.966E-01
1.5	17.0	.433E-02	.150E+01	.102E+03	.156E+02	.312E+01	.344E+03	.906E-01
1.5	18.0	.408E-02	.150E+01	.107E+03	.159E+02	.312E+01	.366E+03	.853E-01
1.5	19.0	.386E-02	.150E+01	.112E+03	.162E+02	.312E+01	.387E+03	.806E-01
1.5	20.0	.367E-02	.150E+01	.118E+03	.165E+02	.312E+01	.408E+03	.764E-01
1.5	21.0	.349E-02	.150E+01	.123E+03	.167E+02	.312E+01	.430E+03	.727E-01
1.5	22.0	.333E-02	.150E+01	.128E+03	.170E+02	.312E+01	.451E+03	.692E-01
1.5	23.0	.318E-02	.150E+01	.133E+03	.172E+02	.312E+01	.472E+03	.661E-01
1.5	24.0	.305E-02	.150E+01	.139E+03	.174E+02	.312E+01	.493E+03	.633E-01
1.5	25.0	.292E-02	.150E+01	.144E+03	.176E+02	.312E+01	.514E+03	.607E-01
1.5	26.0	.281E-02	.150E+01	.149E+03	.178E+02	.312E+01	.535E+03	.583E-01
1.5	27.0	.270E-02	.150E+01	.154E+03	.180E+02	.312E+01	.556E+03	.561E-01
1.5	28.0	.261E-02	.150E+01	.160E+03	.182E+02	.312E+01	.577E+03	.541E-01
1.5	29.0	.252E-02	.150E+01	.165E+03	.184E+02	.312E+01	.598E+03	.522E-01
1.5	30.0	.243E-02	.150E+01	.170E+03	.186E+02	.312E+01	.618E+03	.504E-01

p	t	d	p2	h	s	cv	dpdd	dptt
atm	k	g/cc	atm	j/g	j/g-k	j/g-k	cc-atm/g	atm/k
2.0	3.0	.146E+00	.200E+01	.619E+01	.229E+01	.189E+01	.451E+03	.349E+01
2.0	4.0	.135E+00	.200E+01	.905E+01	.310E+01	.224E+01	.253E+03	.401E+01
2.0	5.0	.103E+00	.200E+01	.152E+02	.444E+01	.278E+01	.294E+02	.285E+01
2.0	6.0	.216E-01	.200E+01	.383E+02	.887E+01	.317E+01	.654E+02	.484E+00
2.0	7.0	.165E-01	.200E+01	.453E+02	.995E+01	.312E+01	.100E+03	.362E+00
2.0	8.0	.137E-01	.200E+01	.516E+02	.108E+02	.311E+01	.130E+03	.296E+00
2.0	9.0	.117E-01	.200E+01	.575E+02	.115E+02	.311E+01	.157E+03	.253E+00
2.0	10.0	.103E-01	.200E+01	.632E+02	.121E+02	.311E+01	.182E+03	.222E+00
2.0	11.0	.927E-02	.200E+01	.688E+02	.126E+02	.311E+01	.206E+03	.198E+00
2.0	12.0	.841E-02	.200E+01	.743E+02	.131E+02	.311E+01	.230E+03	.179E+00
2.0	13.0	.770E-02	.200E+01	.798E+02	.135E+02	.312E+01	.253E+03	.164E+00
2.0	14.0	.711E-02	.200E+01	.852E+02	.139E+02	.312E+01	.276E+03	.151E+00
2.0	15.0	.660E-02	.200E+01	.906E+02	.143E+02	.312E+01	.298E+03	.140E+00
2.0	16.0	.617E-02	.200E+01	.960E+02	.147E+02	.312E+01	.321E+03	.130E+00
2.0	17.0	.579E-02	.200E+01	.101E+03	.150E+02	.312E+01	.343E+03	.122E+00
2.0	18.0	.545E-02	.200E+01	.107E+03	.153E+02	.312E+01	.364E+03	.115E+00
2.0	19.0	.516E-02	.200E+01	.112E+03	.156E+02	.312E+01	.386E+03	.108E+00
2.0	20.0	.489E-02	.200E+01	.117E+03	.159E+02	.312E+01	.408E+03	.103E+00
2.0	21.0	.465E-02	.200E+01	.123E+03	.161E+02	.312E+01	.429E+03	.974E-01
2.0	22.0	.444E-02	.200E+01	.128E+03	.164E+02	.312E+01	.451E+03	.928E-01
2.0	23.0	.424E-02	.200E+01	.133E+03	.166E+02	.312E+01	.472E+03	.886E-01
2.0	24.0	.406E-02	.200E+01	.139E+03	.168E+02	.312E+01	.493E+03	.848E-01
2.0	25.0	.390E-02	.200E+01	.144E+03	.170E+02	.312E+01	.514E+03	.813E-01
2.0	26.0	.374E-02	.200E+01	.149E+03	.172E+02	.312E+01	.536E+03	.780E-01
2.0	27.0	.360E-02	.200E+01	.154E+03	.174E+02	.312E+01	.557E+03	.751E-01
2.0	28.0	.347E-02	.200E+01	.160E+03	.176E+02	.312E+01	.578E+03	.723E-01
2.0	29.0	.335E-02	.200E+01	.165E+03	.178E+02	.312E+01	.599E+03	.697E-01
2.0	30.0	.324E-02	.200E+01	.170E+03	.180E+02	.312E+01	.620E+03	.674E-01

p	t	d	p2	h	s	cv	dpdd	dptt
atm	k	g/cc	atm	j/g	j/g-k	j/g-k	cc-atm/g	atm/k
2.5	3.0	.147E+00	.250E+01	.647E+01	.226E+01	.189E+01	.477E+03	.351E+01
2.5	4.0	.136E+00	.250E+01	.925E+01	.306E+01	.223E+01	.283E+03	.410E+01
2.5	5.0	.112E+00	.250E+01	.144E+02	.420E+01	.269E+01	.716E+02	.326E+01
2.5	6.0	.305E-01	.250E+01	.355E+02	.807E+01	.320E+01	.480E+02	.704E+00
2.5	7.0	.218E-01	.250E+01	.437E+02	.933E+01	.313E+01	.891E+02	.487E+00
2.5	8.0	.176E-01	.250E+01	.503E+02	.102E+02	.311E+01	.121E+03	.388E+00
2.5	9.0	.150E-01	.250E+01	.565E+02	.109E+02	.311E+01	.150E+03	.327E+00
2.5	10.0	.131E-01	.250E+01	.624E+02	.116E+02	.311E+01	.177E+03	.284E+00
2.5	11.0	.117E-01	.250E+01	.681E+02	.121E+02	.311E+01	.202E+03	.253E+00
2.5	12.0	.106E-01	.250E+01	.737E+02	.126E+02	.311E+01	.226E+03	.228E+00
2.5	13.0	.969E-02	.250E+01	.792E+02	.130E+02	.312E+01	.250E+03	.208E+00
2.5	14.0	.893E-02	.250E+01	.847E+02	.135E+02	.312E+01	.273E+03	.191E+00
2.5	15.0	.829E-02	.250E+01	.902E+02	.138E+02	.312E+01	.296E+03	.177E+00
2.5	16.0	.773E-02	.250E+01	.956E+02	.142E+02	.312E+01	.319E+03	.164E+00
2.5	17.0	.725E-02	.250E+01	.101E+03	.145E+02	.312E+01	.341E+03	.154E+00
2.5	18.0	.683E-02	.250E+01	.106E+03	.148E+02	.312E+01	.363E+03	.144E+00
2.5	19.0	.645E-02	.250E+01	.112E+03	.151E+02	.312E+01	.385E+03	.136E+00
2.5	20.0	.612E-02	.250E+01	.117E+03	.154E+02	.312E+01	.407E+03	.129E+00
2.5	21.0	.582E-02	.250E+01	.122E+03	.156E+02	.312E+01	.429E+03	.122E+00
2.5	22.0	.555E-02	.250E+01	.128E+03	.159E+02	.312E+01	.451E+03	.117E+00
2.5	23.0	.530E-02	.250E+01	.133E+03	.161E+02	.312E+01	.472E+03	.111E+00
2.5	24.0	.507E-02	.250E+01	.138E+03	.163E+02	.312E+01	.494E+03	.106E+00
2.5	25.0	.487E-02	.250E+01	.144E+03	.166E+02	.312E+01	.515E+03	.102E+00
2.5	26.0	.468E-02	.250E+01	.149E+03	.168E+02	.312E+01	.536E+03	.979E-01
2.5	27.0	.450E-02	.250E+01	.154E+03	.170E+02	.312E+01	.557E+03	.941E-01
2.5	28.0	.434E-02	.250E+01	.159E+03	.172E+02	.312E+01	.579E+03	.907E-01
2.5	29.0	.419E-02	.250E+01	.165E+03	.173E+02	.312E+01	.600E+03	.874E-01
2.5	30.0	.405E-02	.250E+01	.170E+03	.175E+02	.312E+01	.621E+03	.844E-01

APPENDIX B

SOURCE CODE FOR HELIUM PROGRAM

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        common /coeff/ testar(81)
      2        do 20 j=1,5
1     3        p=0.5*j
1     4        write(*,103)
1     5 103    format(1hl,//'    p    t        d       p2        h       s',
1     6       c'       cv     dpdd     dptt',/
1     7       c'  atm    k     g/cc      atm      j/g    j/g-k    j/g-k',
1     8       c' cc-atm/g    atm/k')
1     9        do 20 i=3,30
2    10        t=i
2    11        d=findd(p,t,-1.)
2    12        p2=dtpres(d,t)
2    13        h=enthal(d,t)
2    14        s=entrop(d,t)
2    15        spcv=cv(d,t)
2    16        derd=dpdd(d,t)
2    17        dert=dpdt(d,t)
2    18        write(*,104)p,t,d,p2,h,s,spcv,derd,dert
2    19 104    format(2f5.1,7e9.3)
2    20 20     continue
     21        end
```

Name	Type	Offset	P	Class
D	REAL	232		
DERD	REAL	272		
DERT	REAL	280		
H	REAL	248		
I	INTEGER*4	224		
J	INTEGER*4	2		
P	REAL	6		
P2	REAL	240		
S	REAL	256		
SPCV	REAL	264		
T	REAL	228		
TESTAR	REAL	0		/COEFF /

Name	Type	Size	Class
COEFF		324	COMMON
CV	REAL		FUNCTION
DPDD	REAL		FUNCTION
DPDT	REAL		FUNCTION
DTPRES	REAL		FUNCTION
ENTHAL	REAL		FUNCTION
ENTROP	REAL		FUNCTION
FINDD	REAL		FUNCTION
MAIN			PROGRAM

```
 Pass One     No Errors Detected
              21 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function vpa(t)
      2        dimension c(10),d(14)
      3        DATA C/-3.9394635287,141.27497598,-1640.7741565,11974.557102,
      4       1-55283.309818,166219.56504,-325212.82840,398843.22750,
      5       2-277718.06992,83395.204183/
      6        DATA D/-49.510540356,651.9236417,-3707.5430856,12880.673491,
      7       1 -30048.545554,49532.267436,-59337.558548,52311.296025,
      8       2-33950.233134,16028.674003,-5354.1038967,1199.0301906,
      9       3 -161.46362959,9.8811553386/
     10 C        THIS GIVES THE VAPOR PRESSURE IN MICRONS OF HG
     11        if(t.le.2.1720)go to 10
     12        p=c(1)*t+c(2)+(c(3)+(c(4)+(c(5)+(c(6)+(c(7)+
     13       c(c(8)+(c(9)+c(10)/t)/t)/t)/t)/t)/t)/t)/t
     14        vpa=exp (p)/760000.
     15        RETURN
     16 10     p=d(1)*t+d(2)+(d(3)+(d(4)+(d(5)+(d(6)+(d(7)+(d(8)+(d(9)+
     17       c(d(10)+(d(11)+(d(12)+(d(13)+d(14)/t)/t)/t)/t)/t)/t)/t)/t)
     18       c/t)/t)/t
     19        vpa=exp (p)/760000.
     20        RETURN
     21 C        THIS GIVES THE VAPOR PRESSURE IN MICRONS OF HG
     22        END

Name     Type        Offset P Class

C       REAL              2
D       REAL             42
EXP                           INTRINSIC
P       REAL             98
T       REAL              0 *


Name     Type          Size   Class

VPA     REAL                   FUNCTION


 Pass One     No Errors Detected
              22 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION DPDTVP(tt)
      2        dimension c(10),d(14)
      3        DATA C/-3.9394635287,141.27497598,-1640.7741565,11974.557102,
      4       1-55283.309818,166219.56504,-325212.82840,398843.22750,
      5       2-277718.06992,83395.204183/
      6        DATA D/-49.510540356,651.9236417,-3707.5430856,12880.673491,
      7       1 -30048.545554,49532.267436,-59337.558548,52311.296025,
      8       2-33950.233134,16028.674003,-5354.1038967,1199.0301906,
      9       3 -161.46362959,9.8811553386/
     10        t=0.998*tt-.001
     11        if(t.le.2.1720)go to 10
     12        p=c(1)-(c(3)+(2.*c(4)+(3.*c(5)+(4.*c(6)+(5.*c(7)+(6.*c(8)+
     13       c(7.*c(9)+8.*c(10)/t)/t)/t)/t)/t)/t)/t)/t/t
     14        dpdtvp=p*vpa(t)
     15        RETURN
     16 10     p=d(1)-(d(3)+(2.*d(4)+(3.*d(5)+(4.*d(6)+(5.*d(7)+(6.*d(8)+
     17       c(7.*d(9)+(8.*d(10)+(9.*d(11)+(10.*d(12)+(11.*d(13)+12.*d(14)/t)
     18       c/t)/t)/t)/t)/t)/t)/t)/t)/t)/t/t
     19        dpdtvp=p*vpa(t)
     20        RETURN
     21        END

Name     Type         Offset P Class

C       REAL              2
D       REAL             42
P       REAL            102
T       REAL             98
TT      REAL              0 *


Name     Type           Size   Class

DPDTVP REAL                    FUNCTION
VPA    REAL                    FUNCTION


 Pass One     No Errors Detected
              21 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION VPTEMP(PATM)
      2        p=patm
      3        PP=P
      4        if(p.lt.0.1)go to 12
      5        if(p.lt.0.7894)go to 20
      6        T=5.0
      7        pcal=vpa(t)
      8        GO TO 13
      9     12 pcal=.0497368
     10        if(abs(p-pcal).le.0.00001*pp)go to 11
     11        t=2.1720
     12        GO TO 13
     13     20 T=3.5
     14        pcal=vpa(t)
     15     13 DO 10 I=1,50
1    16        DP=DPDTVP(T)
1    17        DEL=(PCAL-P)/DP
1    18        T=T-DEL
1    19        pcal=vpa(t)
1    20        if(abs(p-pcal).le.0.00001*p)go to 11
1    21        if(abs(del).le.0.00001*t)go to 11
1    22     10 CONTINUE
     23        write(*,100)t
     24     11 VPTEMP=T
     25        RETURN
     26    100 format(' temperature iteration failed at t=',e10.3)
     27        END

Name     Type         Offset P Class

ABS                           INTRINSIC
DEL      REAL            38
DP       REAL            30
I        INTEGER*4       26
P        REAL             2
PATM     REAL             0 *
PCAL     REAL            14
PP       REAL             6
T        REAL            10


Name     Type           Size   Class

DPDTVP REAL                    FUNCTION
VPA    REAL                    FUNCTION
VPTEMP REAL                    FUNCTION


 Pass One     No Errors Detected
              27 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION DSATV(TT)
      2 C                   RETURNS DENSITY IN GM/CC
      3        dimension gv(6)
      4        DATA GV/-.069267495322,-.1292532553,.29347470712,-.40806658212,
      5       1  .35809505624,-.11315580397/
      6        DATA DC/.06964/
      7        DATA TC/5.2014/
      8        T=TT
      9        R=(1.-T/TC)
     10        r3=r**(1./3.)
     11        dsatv=dc+r3*gv(1)+r3*r3*gv(2)+
     12       cr*(gv(3)+r3*gv(4)+r3*r3*gv(5)+r*gv(6))
     13        RETURN
     14        end
```

Name	Type	Offset	P	Class
DC	REAL	26		
GV	REAL	2		
R	REAL	38		
R3	REAL	42		
T	REAL	34		
TC	REAL	30		
TT	REAL	0	*	

Name	Type	Size	Class
DSATV	REAL		FUNCTION

Pass One No Errors Detected
14 Source Lines

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function dsatl(tt)
      2 C                    RETURNS DENSITY IN GM/CC
      3        dimension gl(6)
      4        data gl/.12874326484,-.43128217346,1.7851911824,-3.3509624489,
      5       1  3.0344215824,-1.0981289602/
      6        DATA DC/.06964/
      7        DATA TC/5.2014/
      8        T=TT
      9        R=(1.-T/TC)
     10        r3=r**(1./3.)
     11        dsatl=dc+r3*gl(1)+r3*r3*gl(2)+
     12       cr*(gl(3)+r3*gl(4)+r3*r3*gl(5)+r*gl(6))
     13        RETURN
     14        END

Name     Type         Offset P Class

DC      REAL             26
GL      REAL              2
R       REAL             38
R3      REAL             42
T       REAL             34
TC      REAL             30
TT      REAL              0 *


Name     Type           Size   Class

DSATL   REAL                   FUNCTION


 Pass One     No Errors Detected
              14 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION PMELT(T)
      2 C      MELTING PRESSURE OF SOLID HELIUM IN ATMOS
      3        if(t.lt.5.2)go to 3
      4        pmelt=-17.22+16.746*t**1.555414
      5        RETURN
      6      3 if(t.lt.1.8)go to 5
      7        pkgcm=33.28-44.156*t+31.799*t*t-4.8159*t**3
      8       1+0.30313*T**4
      9        PMELT = PKGCM/1.03393
     10        RETURN
     11      5 if(t.lt.0.)go to 7
     12        pmelt=25.0+0.048*t**8
     13        RETURN
     14      7 write(*,17)
     15        pmelt=25.0
     16        return
     17     17 format('  pmelt is undefined for negative t, arbitrarily set pmelt
     18       1 = 25.0')
     19        END

Name     Type         Offset P Class

PKGCM   REAL               2
T       REAL               0 *


Name     Type           Size   Class

PMELT   REAL                   FUNCTION


 Pass One     No Errors Detected
              19 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION PLAMDA(T)
      2 C     THE PRESSURE AT THE TRANSITION BETWEEEN HE ONE AND HE TWO
      3        if((t.gt.2.172).or.(t.lt.1.762))go to 5
      4        x=t-2.172
      5        x2=x*x
      6        x3=x2*x
      7        x4=x3*x
      8        PLAMDA = 0.42800749 - 95.0719*X - 86.417*X2 - 103.341*X3
      9       1-77.52175*X4 - 0.37827065*EXP (42.2507*X)
     10        RETURN
     11      5 write(*,15)t
     12        plamda=1.0
     13        return
     14     15 format(' the lamda pressure is undefined at t=',f8.3,
     15       1' plamda arbitrarily set = 1.0')
     16        END

Name     Type         Offset P Class

EXP                           INTRINSIC
T        REAL              0 *
X        REAL              2
X2       REAL              6
X3       REAL             10
X4       REAL             14


Name     Type           Size   Class

PLAMDA REAL                    FUNCTION


 Pass One     No Errors Detected
              16 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION VIRB(T)
      2        dimension a(9)
      3        DATA A/-5.0815710041E-7,-1.1168680862E-4,1.1652480354E-2,
      4       1 7.4474587998E-2,-5.3143174768E-1,-9.5759219306E-1,
      5       2 3.9374414843,-5.1370239224,2.0804456338/
      6 C      COEFFICIENTS FROM PROGRAM 5/28/70-1630
      7 C      THIS SUB PROGRAM CALCULATES THE SECOND VIRIAL COEFFICIENT FOR
      8 C      HELIUM.  THE RANGE IS FROM 2 TO 1500 DEG K.  INPUT IS TEMPERATURE
      9 C      IN DEGREES KELVIN AND IUSE, IF ISUE IS 0 OR NEGATIVE THE ROUTINE
     10 C      CALCULATES B FOR THE EQUATION PV=RT(1+BD+..., FOR OTHER VALUES OF
     11 C      ISUE, THE ROUTINE CALCULATES THE VARIANCE OF B AT THE INPUT TEMP
     12 C      UNITS ARE ATM, DEG KELVIN, AND MOLES/LITER,4/3/69-1253,R.D.MCCARTY
     13 C      REVISED 2/12/70-925
     14        t12=sqrt(t)
     15        virb=a(1)*t+a(2)*t12+a(3)+
     16       c(a(4)+a(6)/t+a(8)/t/t)/t12+
     17       c(a(5)+a(7)/t+a(9)/t/t)/t
     18        RETURN
     19        END

Name     Type         Offset P Class

A       REAL               2
SQRT                           INTRINSIC
T       REAL               0 *
T12     REAL              38


Name     Type            Size   Class

VIRB    REAL                   FUNCTION


 Pass One     No Errors Detected
              19 Source Lines
```

```
) Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
     1         FUNCTION DBDT (T)
     2         dimension a(9)
     3 C       THIS SUB PROGRAM CALCULATES THE SECOND VIRIAL COEFFICIENT FOR
     4 C       HELIUM.  THE RANGE IS FROM 2 TO 1500 DEG K.  INPUT IS TEMPERATURE
     5 C       IN DEGREES KELVIN AND IUSE, IF ISUE IS 0 OR NEGATIVE THE ROUTINE
     6 C       CALCULATES B FOR THE EQUATION PV=RT(1+BD+..., FOR OTHER VALUES OF
     7 C       IUSE, THE ROUTINE CALCULATES THE VARIANC OF B AT THE INPUT TEMP
     8         DATA A/-5.0815710041E-7,-1.1168680862E-4,1.1652480354E-2,
     9        1 7.4474587998E-2,-5.3143174768E-1,-9.5759219306E-1,
    10        2 3.9374414843,-5.1370239224,2.0804456338/
    11 C       UNITS ARE ATM, DEG KELVIN, AND MOLES/LITER,5/28/70-1630,R.D.MCCART
    12         t12=sqrt(t)
    13         t2=t*t
    14         dbdt=a(1)+(0.5*a(2)-0.5*a(4)/t-1.5*a(6)/t2-2.5*a(8)/t/t2)/t12-
    15        c(a(5)+2.*a(7)/t+3.*a(9)/t2)/t2
    16         RETURN
    17         END

  e    Type        Offset P Class

     REAL              2
                           INTRINSIC
     REAL              0 *
     REAL             38
     REAL             42


      Type          Size   Class

     REAL                  FUNCTION


    One     No Errors Detected
            17 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION D2DBDT2(T)
      2        DIMENSION A(9)
      3        DATA A/-5.0815710041E-7,-1.1168680862E-4,1.1652480354E-2,
      4       1 7.4474587998E-2,-5.3143174768E-1,-9.5759219306E-1,
      5       2 3.9374414843,-5.1370239224,2.0804456338/
      6 C      THIS SUB PROGRAM CALCULATES THE SECOND VIRIAL COEFFICIENT FOR
      7 C      HELIUM.  THE RANGE IS FROM 2 TO 1500 DEG K.  INPUT IS TEMPERATURE
      8 C      IN DEGREES KELVIN AND IUSE, IF ISUE IS 0 OR NEGATIVE THE ROUTINE
      9 C      CALCULATES B FOR THE EQUATION PV=RT(1+BD+..., FOR OTHER VALUES OF
     10 C      IUSE, THE ROUTINE CALCULATES THE VARIANC OF B AT THE INPUT TEMP
     11 C      UNITS ARE ATM, DEG KELVIN, AND MOLES/LITER,5/28/70-1630,R.D.MCCART
     12        t12=sqrt(t)
     13        t2=t*t
     14        d2dbdt2=(-0.25*a(2)+0.75*a(4)/t+3.75*a(6)/t2+8.75*a(8)/t/t2)/t/t12
     15       c+(2.*a(5)+6.*a(7)/t+12.*a(9)/t2)/t/t2
     16        RETURN
     17        END

Name     Type        Offset P Class

A        REAL             2
SQRT                          INTRINSIC
T        REAL             0 *
T12      REAL            38
T2       REAL            42


Name     Type          Size   Class

D2DBDT REAL                   FUNCTION


 Pass One     No Errors Detected
              17 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function dtpres(dgc,tt)
      2 C         THIS FUNCTION CALCULATES THE EQUATION OF STATE FOR HELIUM
      3 C         INPUT UNITS ARE DENSITY IN GM/CC (DGC) AND DEG. K (TT)
      4 C         ENTRY AT DTPRESS RETURNS PRESSURE AS A FUNCTION OF DENSITY AND
      5 C         TEMPERATURE
      6        COMMON/COEFF/B(27,3)
      7        DIMENSION A(26)
      8        k1=0
      9 C         CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
     10        DD=DGC/4.0026E-3
     11        KH=1
     12        K=1
     13        KK=1
     14        K4=1
     15        IF(TT.GE.15.)GO TO 20
     16        IF(TT.GT.10.)GO TO 30
     17        IF(DD.GT.17.3987) GO TO 40
     18        i=1
     19        T=TT
     20        D=DD
     21      8 d2=d*d
     22        D3=D2*D
     23        D4=D3*D
     24        D5=D4*D
     25        GAMMA=B(27,I)
     26        EX=EXP (D2*GAMMA)
     27        EXD3=EX*D3
     28        EXD5=EX*D5
     29        M=I
     30        a(1)=d5*d
     31        a(2)=a(1)/t
     32        t12=sqrt(t)
     33        t14=t**0.250
     34        a(3)=d5*t12
     35        a(4)=d5*t14
     36        a(5)=d5
     37        a(6)=d5/t14
     38        a(7)=d5/t12
     39        a(8)=a(7)/t14
     40        a(9)=d4*t12
     41        a(10)=d4/t12
     42        a(11)=a(10)/t
     43        a(12)=a(11)/t
     44        a(13)=d3*t
     45        a(14)=d3*t12
     46        a(15)=d3
     47        a(16)=d3/t12
     48        a(17)=d3/t
     49        a(18)=a(16)/t
     50        a(19)=a(17)/t
     51        a(20)=a(18)/t
     52        a(21)=exd3
     53        a(22)=a(21)/t
     54        a(23)=a(22)/t
     55        a(24)=exd5
     56        a(25)=a(24)/t
     57        a(26)=a(25)/t
     58        n=26
     59        I=M
```

```
      7 P=0
        DO 15 J=1,N
     15 P=P+B(J,I)*A(J)
        P=P+.0820558*D*T*(1+VIRB(T)*D)
        GO TO(50,50,30,40),K
     20 K=2
        I=3
        T=TT
        D=DD
        GO TO 8
     30 K=3
        GO TO(33,34,35),KK
     33 D=DD
        T=TT
        KK=2
        IF(DD.GT.17.3987) GO TO 40
        I=1
        GO TO 8
     34 PTI=P
        I=3
        KK=3
        T=TT
        D=DD
        GO TO 8
     35 PTIII=P
        f=(15.-t)/5.
              P=F*PTI+(1.-F)*PTIII
        GO TO 50
     40 IF(K.EQ.3)K1=3
        GO TO(41,42,43,44),K4
     41 K=4
        I=2
        K4=2
        D=DD
        IF(K1.EQ.0)T=TT
        GO TO 8
     42 PIID=P
        D=17.3987
        IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
        K4=3
        GO TO 8
     43 PIIDC=P
        I=1
        K4=4
        GO TO 8
     44 PIDC=P
        P=PIDC+(PIID-PIIDC)
        K4=1
        IF(K1.EQ.3)GO TO 30
     50 dtpres=p
C          RETURNS DTPRESS - ATM
        RETURN
        END
```

Name	Type	Offset	P	Class
A	REAL	2		
B	REAL	0		/COEFF /
D	REAL	138		

```
D Line# 1      7
D2      REAL            142
D3      REAL            146
D4      REAL            150
D5      REAL            154
DD      REAL            110
DGC     REAL              0 *
EX      REAL            162
EXD3    REAL            166
EXD5    REAL            170
EXP                             INTRINSIC
F       REAL            214
GAMMA   REAL            158
I       INTEGER*4       130
J       INTEGER*4       194
K       INTEGER*4       118
K1      INTEGER*4       106
K4      INTEGER*4       126
KH      INTEGER*4       114
KK      INTEGER*4       122
M       INTEGER*4       174
N       INTEGER*4       186
P       REAL            190
PIDC    REAL            230
PIID    REAL            218
PIIDC   REAL            226
PTI     REAL            206
PTIII   REAL            210
SQRT                            INTRINSIC
T       REAL            134
T12     REAL            178
T14     REAL            182
TT      REAL              4 *


Name     Type           Size   Class

COEFF                    324   COMMON
DSATL   REAL                   FUNCTION
DTPRES  REAL                   FUNCTION
VIRB    REAL                   FUNCTION


 Pass One     No Errors Detected
             112 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function dpdt(dgc,tt)
      2 C         INPUT UNITS ARE DENSITY IN GM/CC (DGC) AND DEG. K (TT)
      3 C         OUTPUT UNITS ARE AS INDICATED IN RETURN COMMENT
      4        COMMON/COEFF/B(27,3)
      5        DIMENSION A(26)
      6 C         ENTRY AT DPDT RETURNS ISOCHORE DERIVATIVE - THE PARTIAL
      7 C         DERIVATIVE OF PRESSURE WITH RESPECT TO TEMPERATURE WITH
      8 C         DENSITY HELD CONSTANT
      9        k1=0
     10 C         CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
     11        DD=DGC/4.0026E-3
     12        KH=1
     13        K=1
     14        KK=1
     15        K4=1
     16        IF(TT.GE.15.)GO TO 20
     17        IF(TT.GT.10.)GO TO 30
     18        IF(DD.GT.17.3987) GO TO 40
     19        i=1
     20        T=TT
     21        D=DD
     22      8 d2=d*d
     23        D3=D**3
     24        D4=D3*D
     25        D5=D4*D
     26        D6=D5*D
     27        T2=T*T
     28        T3=T2*T
     29        t4=t2*t2
     30        m=i
     31        gamma=b(27,m)
     32        ex=exp (d2*gamma)
     33        r=.0820558
     34        a(1)=0.0
     35        a(2)=-d6/t2
     36        t12=sqrt(t)
     37        t14=t**0.250
     38        a(3)=d5/2./t12
     39        a(4)=a(3)/2./t14
     40        a(5)=0.0
     41        a(6)=-d5/4./t/t14
     42        a(7)=-d5/2./t/t12
     43        a(8)=a(6)*3./t12
     44        a(9)=d4/2./t12
     45        a(10)=-a(9)/t
     46        a(11)=a(10)*3./t
     47        a(12)=a(11)*5./3./t
     48        a(13)=d3
     49        a(14)=d3/2./t12
     50        a(15)=0.0
     51        a(16)=-a(14)/t
     52        a(17)=-d3/t/t
     53        a(18)=a(16)*3./t
     54        a(19)=a(17)*2./t
     55        a(20)=a(18)*5./3./t
     56        a(21)=0.0
     57        a(22)=-ex*d3/t/t
     58        a(23)=a(22)*2./t
     59        a(24)=0.0
```

```
      a(25)=-ex*d5/t/t
      a(26)=a(25)*2./t
      n=26
      P=0
      DO 115 J=1,N
1   115 P=P+A(J)*B(J,M)
      P=P+R*D*(1.+VIRB(T)*D)+R*D*T*DBDT(T)*D
      I=M
      GO TO(50,50,30,40),K
   20 K=2
      I=3
      T=TT
      D=DD
      GO TO 8
   30 K=3
      GO TO(33,34,35),KK
   33 D=DD
      T=TT
      KK=2
      IF(DD.GT.17.3987) GO TO 40
      I=1
      GO TO 8
   34 PTI=P
      I=3
      KK=3
      T=TT
      D=DD
      GO TO 8
   35 PTIII=P
      f=(15.-t)/5.
            P=F*PTI+(1.-F)*PTIII
      GO TO 50
   40 IF(K.EQ.3)K1=3
      GO TO(41,42,43,44),K4
   41 K=4
      I=2
      K4=2
      D=DD
      IF(K1.EQ.0)T=TT
      GO TO 8
   42 PIID=P
      D=17.3987
      IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
      K4=3
      GO TO 8
   43 PIIDC=P
      I=1
      K4=4
      GO TO 8
   44 PIDC=P
      P=PIDC+(PIID-PIIDC)
      K4=1
      IF(K1.EQ.3)GO TO 30
   50 dpdt=p
C        RETURNS DPDT-ATM/K
      RETURN
      END
```

Name	Type	Offset	P	Class
A	REAL	2		
B	REAL	0		/COEFF /
D	REAL	138		
D2	REAL	142		
D3	REAL	146		
D4	REAL	150		
D5	REAL	154		
D6	REAL	158		
DD	REAL	110		
DGC	REAL	0	*	
EX	REAL	182		
EXP				INTRINSIC
F	REAL	230		
GAMMA	REAL	178		
I	INTEGER*4	130		
J	INTEGER*4	206		
K	INTEGER*4	118		
K1	INTEGER*4	106		
K4	INTEGER*4	126		
KH	INTEGER*4	114		
KK	INTEGER*4	122		
M	INTEGER*4	174		
N	INTEGER*4	198		
P	REAL	202		
PIDC	REAL	246		
PIID	REAL	234		
PIIDC	REAL	242		
PTI	REAL	222		
PTIII	REAL	226		
R	REAL	186		
SQRT				INTRINSIC
T	REAL	134		
T12	REAL	190		
T14	REAL	194		
T2	REAL	162		
T3	REAL	166		
T4	REAL	170		
TT	REAL	4	*	

Name	Type	Size	Class
COEFF		324	COMMON
DBDT	REAL		FUNCTION
DPDT	REAL		FUNCTION
DSATL	REAL		FUNCTION
VIRB	REAL		FUNCTION

Pass One No Errors Detected
116 Source Lines

```
      function dpdd(dgc,tt)
C        INPUT UNITS ARE DENSITY IN GM/CC (DGC) AND DEG. K (TT)
C        OUTPUT UNITS ARE AS INDICATED IN RETURN COMMENT
      COMMON/COEFF/B(27,3)
      DIMENSION A(26)
C        ENTRY AT DPDD RETURNS THE ISOTHERM DERIVATIVE - THE PARTIAL
C        DERIVATIVE OF PRESSURE WITH RESPECT TO DENSITY WITH THE
C        TEMPERATURE HELD CONSTANT
      k1=0
C        CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
      DD=DGC/4.0026E-3
      KH=1
      K=1
      KK=1
      K4=1
      IF(TT.GE.15.)GO TO 20
      IF(TT.GT.10.)GO TO 30
      IF(DD.GT.17.3987) GO TO 40
      i=1
      T=TT
      D=DD
    8 d2=d*d
      D3=D2*D
      D4=D3*D
      D5=D4*D
      M=I
      GAMMA=B(27,M)
      EX=EXP (D2*GAMMA)
      DEX=GAMMA*2.*D*EX
      R=.0820558
      a(1)=6.*d5
      a(2)=a(1)/t
      t12=sqrt(t)
      t14=t**0.250
      a(3)=5.*d4*t12
      a(4)=a(3)/t14
      a(5)=5.*d4
      a(6)=a(5)/t14
      a(7)=a(5)/t12
      a(8)=a(7)/t14
      a(9)=d3*4.*t12
      a(10)=d3*4./t12
      a(11)=a(10)/t
      a(12)=a(11)/t
      a(13)=d2*3.*t
      a(14)=a(13)/t12
      a(15)=d2*3.
      a(16)=a(15)/t12
      a(17)=a(15)/t
      a(18)=a(16)/t
      a(19)=a(17)/t
      a(20)=a(18)/t
      a(21)=dex*d3+3.*d2*ex
      a(22)=a(21)/t
      a(23)=a(22)/t
      a(24)=dex*d5+5.*d4*ex
      a(25)=a(24)/t
      a(26)=a(25)/t
      n=26
```

```
     60          P=0
     61          DO 215 J=1,N
1    62      215 P=P+A(J)*B(J,M)
     63          I=M
     64          P=P+R*T*(1.+2.*D*VIRB(T))
     65          GO TO(50,50,30,40),K
     66       20 K=2
     67          I=3
     68          T=TT
     69          D=DD
     70          GO TO 8
     71       30 K=3
     72          GO TO(33,34,35),KK
     73       33 D=DD
     74          T=TT
     75          KK=2
     76          IF(DD.GT.17.3987) GO TO 40
     77          I=1
     78          GO TO 8
     79       34 PTI=P
     80          I=3
     81          KK=3
     82          T=TT
     83          D=DD
     84          GO TO 8
     85       35 PTIII=P
     86          f=(15.-t)/5.
     87                P=F*PTI+(1.-F)*PTIII
     88          GO TO 50
     89       40 IF(K.EQ.3)K1=3
     90          GO TO(41,42,43,44),K4
     91       41 K=4
     92          I=2
     93          K4=2
     94          D=DD
     95          IF(K1.EQ.0)T=TT
     96          GO TO 8
     97       42 PIID=P
     98          D=17.3987
     99          IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
    100          K4=3
    101          GO TO 8
    102       43 PIIDC=P
    103          I=1
    104          K4=4
    105          GO TO 8
    106       44 PIDC=P
    107          P=PIDC+(PIID-PIIDC)
    108          K4=1
    109          IF(K1.EQ.3)GO TO 30
    110       50 dpdd=p/4.0026e-3
    111 C           RETURNS DPDD - CC-ATM/GM
    112          RETURN
    113          END
```

Name	Type	Offset	P	Class
A	REAL	2		
B	REAL	0		/COEFF /

D	REAL	138	
D2	REAL	142	
D3	REAL	146	
D4	REAL	150	
D5	REAL	154	
DD	REAL	110	
DEX	REAL	170	
DGC	REAL	0 *	
EX	REAL	166	
EXP			INTRINSIC
F	REAL	214	
GAMMA	REAL	162	
I	INTEGER*4	130	
J	INTEGER*4	194	
K	INTEGER*4	118	
K1	INTEGER*4	106	
K4	INTEGER*4	126	
KH	INTEGER*4	114	
KK	INTEGER*4	122	
M	INTEGER*4	158	
N	INTEGER*4	186	
P	REAL	190	
PIDC	REAL	230	
PIID	REAL	218	
PIIDC	REAL	226	
PTI	REAL	206	
PTIII	REAL	210	
R	REAL	174	
SQRT			INTRINSIC
T	REAL	134	
T12	REAL	178	
T14	REAL	182	
TT	REAL	4 *	

Name	Type	Size	Class
COEFF		324	COMMON
DPDD	REAL		FUNCTION
DSATL	REAL		FUNCTION
VIRB	REAL		FUNCTION

Pass One No Errors Detected
113 Source Lines

```
D Line# 1     7                                Microsoft FORTRAN77 V3.20 02/84
      1       function entrop(dgc,tt)
      2 C        INPUT UNITS ARE DENSITY IN GM/CC (DGC) AND DEG. K (TT)
      3 C        OUTPUT UNITS ARE AS INDICATED IN RETURN COMMENT
      4       COMMON/COEFF/B(27,3)
      5       DIMENSION A(26)
      6 C        ENTRY AT ENTROP RETURNS THE ENTROPY OF HELIUM AS A FUNCTION
      7 C        OF DENSITY AND TEMPERATURE
      8       k1=0
      9 C        CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
     10       DD=DGC/4.0026E-3
     11       KH=1
     12       K=1
     13       KK=1
     14       K4=1
     15       IF(TT.GE.15.)GO TO 20
     16       IF(TT.GT.10.)GO TO 30
     17       IF(DD.GT.17.3987) GO TO 40
     18       i=1
     19       T=TT
     20       D=DD
     21     8 d2=d*d
     22       D3=D2*D
     23       D4=D3*D
     24       d5=d4*d
     25       R=.0820558
     26       M=I
     27       GAMMA=B(27,M)
     28       EX=EXP (D2*GAMMA)
     29       a(1)=0.0
     30       a(2)=-d5/5./t/t
     31       t12=sqrt(t)
     32       t14=t**0.250
     33       a(3)=d4/4./2./t12
     34       a(4)=a(3)/2./t14
     35       a(5)=0.0
     36       a(6)=-a(4)/t12
     37       a(7)=-a(3)/t
     38       a(8)=a(7)*3./2./t14
     39       a(9)=d3/3./2./t12
     40       a(10)=-a(9)/t
     41       a(11)=a(10)*3./t
     42       a(12)=a(11)*5./3./t
     43       a(13)=d2/2.
     44       a(14)=a(13)/2./t12
     45       a(15)=0.0
     46       a(16)=-a(14)/t
     47       a(17)=-d2/2./t/t
     48       a(18)=a(16)*3./t
     49       a(19)=a(17)*2./t
     50       a(20)=a(18)*5./3./t
     51       a(21)=0.0
     52       a(22)=-ex/(2.*gamma)/t/t
     53       a(23)=a(22)*2./t
     54       a(24)=0.0
     55       a(25)=-ex/(2.*gamma)*(d2-1./gamma)/t/t
     56       a(26)=a(25)*2./t
     57       n=26
     58       SINT=D*R*(VIRB(T)+T*DBDT(T))
     59       DO 306 I=1,N
```

```
1    306 SINT=SINT+B(I,M)*A(I)
         EX=1.
         D2=0
         a(21)=0.0
         a(22)=-ex/(2.*gamma)/t/t
         a(23)=a(22)*2./t
         a(24)=0.0
         a(25)=-ex/(2.*gamma)*(d2-1./gamma)/t/t
         a(26)=a(25)*2./t
         n=26
         DO 312 I=21,N
1    312 SINT=SINT-B(I,M)*A(I)
         P=(9.2609+5.193043*ALOG(T/4.2144)-25.31469*(SINT+R*ALOG(R*T*D)))
         I=M
         GO TO(50,50,30,40),K
      20 K=2
         I=3
         T=TT
         D=DD
         GO TO 8
      30 K=3
         GO TO(33,34,35),KK
      33 D=DD
         T=TT
         KK=2
         IF(DD.GT.17.3987) GO TO 40
         I=1
         GO TO 8
      34 PTI=P
         I=3
         KK=3
         T=TT
         D=DD
         GO TO 8
      35 PTIII=P
         f=(15.-t)/5.
               P=F*PTI+(1.-F)*PTIII
         GO TO 50
      40 IF(K.EQ.3)K1=3
         GO TO(41,42,43,44),K4
      41 K=4
         I=2
         K4=2
         D=DD
         IF(K1.EQ.0)T=TT
         GO TO 8
      42 PIID=P
         D=17.3987
         IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
         K4=3
         GO TO 8
      43 PIIDC=P
         I=1
         K4=4
         GO TO 8
      44 PIDC=P
         P=PIDC+(PIID-PIIDC)
         K4=1
         IF(K1.EQ.3)GO TO 30
```

```
D Line# 1      7
    119     50 entrop=p
    120 C         RETURNS ENTROP - J/GM-K
    121        return
    122        END
```

Name	Type	Offset	P	Class
A	REAL	2		
ALOG				INTRINSIC
B	REAL	0		/COEFF /
D	REAL	138		
D2	REAL	142		
D3	REAL	146		
D4	REAL	150		
D5	REAL	154		
DD	REAL	110		
DGC	REAL	0	*	
EX	REAL	170		
EXP				INTRINSIC
F	REAL	218		
GAMMA	REAL	166		
I	INTEGER*4	130		
K	INTEGER*4	118		
K1	INTEGER*4	106		
K4	INTEGER*4	126		
KH	INTEGER*4	114		
KK	INTEGER*4	122		
M	INTEGER*4	162		
N	INTEGER*4	182		
P	REAL	206		
PIDC	REAL	234		
PIID	REAL	222		
PIIDC	REAL	230		
PTI	REAL	210		
PTIII	REAL	214		
R	REAL	158		
SINT	REAL	186		
SQRT				INTRINSIC
T	REAL	134		
T12	REAL	174		
T14	REAL	178		
TT	REAL	4	*	

Name	Type	Size	Class
COEFF		324	COMMON
DBDT	REAL		FUNCTION
DSATL	REAL		FUNCTION
ENTROP	REAL		FUNCTION
VIRB	REAL		FUNCTION

```
 Pass One     No Errors Detected
             122 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function enthal(dgc,tt)
      2 C         INPUT UNITS ARE DENSITY IN GM/CC (DGC) AND DEG. K (TT)
      3        COMMON/COEFF/B(27,3)
      4        DIMENSION A(26)
      5 C         ENTRY AT ENTHAL RETURNS ENTHALPY OF HELIUM AS A FUNCTION
      6 C         OF DENSITY AND TEMPERATURE
      7        k1=0
      8 C         CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
      9        DD=DGC/4.0026E-3
     10        KH=1
     11        K=1
     12        kk=1
     13        K4=1
     14        IF(TT.GE.15.)GO TO 20
     15        IF(TT.GT.10.)GO TO 30
     16        IF(DD.GT.17.3987) GO TO 40
     17        i=1
     18        T=TT
     19        D=DD
     20      8 kh=0
     21        d2=d*d
     22        D3=D2*D
     23        D4=D3*D
     24        D5=D4*D
     25        GAMMA=B(27,I)
     26        EX=EXP (D2*GAMMA)
     27        EXD3=EX*D3
     28        EXD5=EX*D5
     29        M=I
     30        a(1)=d5*d
     31        a(2)=a(1)/t
     32        t12=sqrt(t)
     33        t14=t**0.250
     34        a(3)=d5*t12
     35        a(4)=d5*t14
     36        a(5)=d5
     37        a(6)=d5/t14
     38        a(7)=d5/t12
     39        a(8)=a(7)/t14
     40        a(9)=d4*t12
     41        a(10)=d4/t12
     42        a(11)=a(10)/t
     43        a(12)=a(11)/t
     44        a(13)=d3*t
     45        a(14)=d3*t12
     46        a(15)=d3
     47        a(16)=d3/t12
     48        a(17)=d3/t
     49        a(18)=a(16)/t
     50        a(19)=a(17)/t
     51        a(20)=a(18)/t
     52        a(21)=exd3
     53        a(22)=a(21)/t
     54        a(23)=a(22)/t
     55        a(24)=exd5
     56        a(25)=a(24)/t
     57        a(26)=a(25)/t
     58        n=26
     59        I=M
```

```
D Line# 1      7
     60      7 P=0
     61        DO 15 J=1,N
1    62     15 P=P+B(J,I)*A(J)
     63        P=P+.0820558*D*T*(1+VIRB(T)*D)
     64        IF(KH.LT.1)GO TO 413
     65        GO TO(50,50,30,40),K
     66    413 PP=P
     67        KH=1
     68        D2=D*D
     69        D3=D*D2
     70        D4=D3*D
     71        d5=d4*d
     72        R=.0820558
     73        M=I
     74        GAMMA=B(27,M)
     75        EX=EXP (D2*GAMMA)
     76        a(1)=d5/5.
     77        a(2)=a(1)*2./t
     78        t12=sqrt(t)
     79        t14=t**0.250
     80        a(3)=d4/4./2.*t12
     81        a(4)=d4/4.*3./4.*t14
     82        a(5)=d4/4.
     83        a(6)=a(5)*5./4./t14
     84        a(7)=a(6)*6./5./t14
     85        a(8)=a(7)*7./6./t14
     86        a(9)=d3/3./2.*t12
     87        a(10)=d3/2./t12
     88        a(11)=a(10)*5./3./t
     89        a(12)=a(11)*7./5./t
     90        a(13)=0.0
     91        a(14)=d2/2./2.*t12
     92        a(15)=d2/2.
     93        a(16)=d2*3./4./t12
     94        a(17)=d2/t
     95        a(18)=a(16)*5./3./t
     96        a(19)=a(17)*6./4./t
     97        a(20)=a(18)*7./5./t
     98        a(21)=ex/(2.*gamma)
     99        a(22)=a(21)*2./t
    100        a(23)=a(22)*3./2./t
    101        a(24)=ex/(2.*gamma)*(d2-1./gamma)
    102        a(25)=a(24)*2./t
    103        a(26)=a(25)*3./2./t
    104        n=26
    105        hint=-r*t*t*d*dbdt(t)
    106        do 406 i=1,n
1   107    406 HINT=HINT+B(I,M)*A(I)
    108        D2=0
    109        EX=1.
    110        a(21)=ex/(2.*gamma)
    111        a(22)=a(21)*2./t
    112        a(23)=a(22)*3./2./t
    113        a(24)=ex/(2.*gamma)*(d2-1./gamma)
    114        a(25)=a(24)*2./t
    115        a(26)=a(25)*3./2./t
    116        n=26
    117        DO 412 I=21,N
1   118    412 HINT=HINT-B(I,M)*A(I)
```

```
      P=36.5582+5.193043*(T-4.2144)+25.31469*(HINT+PP/D-R*T)
      I=M
      GO TO(50,50,30,40),K
   20 K=2
      I=3
      T=TT
      D=DD
      GO TO 8
   30 K=3
      GO TO(33,34,35),KK
   33 D=DD
      T=TT
      KK=2
      IF(DD.GT.17.3987) GO TO 40
      I=1
      GO TO 8
   34 PTI=P
      I=3
      KK=3
      T=TT
      D=DD
      GO TO 8
   35 PTIII=P
      f=(15.-t)/5.
            P=F*PTI+(1.-F)*PTIII
      IF(KH.LT.1)GO TO 413
      GO TO 50
   40 IF(K.EQ.3)K1=3
      GO TO(41,42,43,44),K4
   41 K=4
      I=2
      K4=2
      D=DD
      IF(K1.EQ.0)T=TT
      GO TO 8
   42 PIID=P
      D=17.3987
      IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
      K4=3
      GO TO 8
   43 PIIDC=P
      I=1
      K4=4
      GO TO 8
   44 PIDC=P
      P=PIDC+(PIID-PIIDC)
      K4=1
      IF(K1.EQ.3)GO TO 30
   50 enthal=p
C        RETURNS ENTHAL- J/GM
      RETURN
      END
```

Name	Type	Offset	P	Class
A	REAL	2		
B	REAL	0		/COEFF /
D	REAL	138		
D2	REAL	142		

```
D Line# 1      7
D3      REAL            146
D4      REAL            150
D5      REAL            154
DD      REAL            110
DGC     REAL              0 *
EX      REAL            162
EXD3    REAL            166
EXD5    REAL            170
EXP                             INTRINSIC
F       REAL            238
GAMMA   REAL            158
HINT    REAL            214
I       INTEGER*4       130
J       INTEGER*4       194
K       INTEGER*4       118
K1      INTEGER*4       106
K4      INTEGER*4       126
KH      INTEGER*4       114
KK      INTEGER*4       122
M       INTEGER*4       174
N       INTEGER*4       186
P       REAL            190
PIDC    REAL            254
PIID    REAL            242
PIIDC   REAL            250
PP      REAL            206
PTI     REAL            230
PTIII   REAL            234
R       REAL            210
SQRT                            INTRINSIC
T       REAL            134
T12     REAL            178
T14     REAL            182
TT      REAL              4 *


Name     Type           Size    Class

COEFF                    324    COMMON
DBDT    REAL                    FUNCTION
DSATL   REAL                    FUNCTION
ENTHAL  REAL                    FUNCTION
VIRB    REAL                    FUNCTION


 Pass One     No Errors Detected
             170 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        function cv(dgc,tt)
      2 C         INPUT UNITS ARE DENSITY IN.GM/CC (DGC) AND DEG. K (TT)
      3        COMMON/COEFF/B(27,3)
      4        DIMENSION A(26)
      5 C         ENTRY AT CV RETURNS THE SPECIFIC HEAT OF HELIUM AT CONSTANT
      6 C         VOLUME AS A FUNCTION OF DENSITY AND TEMPERATURE
      7        k1=0
      8 C         CHANGE INPUT DENSITY UNITS FROM GM/CC TO MOL/LITER
      9        DD=DGC/4.0026E-3
     10        KH=1
     11        K=1
     12        KK=1
     13        K4=1
     14        IF(TT.GE.15.)GO TO 20
     15        IF(TT.GT.10.)GO TO 30
     16        IF(DD.GT.17.3987) GO TO 40
     17        i=1
     18        T=TT
     19        D=DD
     20      8 d2=d*d
     21        D3=D2*D
     22        D4=D3*D
     23        d5=d4*d
     24        R=.0820558
     25        M=I
     26        GAMMA=B(27,M)
     27        EX=EXP (D2*GAMMA)
     28        SINT=T*D*R*(2.*DBDT(T  )+D2DBDT2(T)*T)
     29        a(1)=0.0
     30        a(2)=d5/5.*2./t/t
     31        t12=sqrt(t)
     32        t14=t**0.250
     33        a(3)=-d4/16./t12
     34        a(4)=a(3)*3./4./t14
     35        a(5)=0.0
     36        a(6)=d4/4.*5./16./t/t14
     37        a(7)=a(6)*12./5./t14
     38        a(8)=a(7)*21./12./t14
     39        a(9)=-d3/3./4./t12
     40        a(10)=d3/4./t/t12
     41        a(11)=a(10)*5./t
     42        a(12)=a(11)*7./3./t
     43        a(13)=0.0
     44        a(14)=-d2/2./4./t12
     45        a(15)=0.0
     46        a(16)=d2/2.*3./4./t/t12
     47        a(17)=d2/t/t
     48        a(18)=a(16)*5./t
     49        a(19)=a(17)*3./t
     50        a(20)=a(18)*7./3./t
     51        a(21)=0.0
     52        a(22)=ex/(2.*gamma)*2./t/t
     53        a(23)=a(22)*3./t
     54        a(24)=0.0
     55        a(25)=ex/(2.*gamma)*(d2-1./gamma)*2./t/t
     56        a(26)=a(25)*3./t
     57        n=26
     58        DO 506 I=1,N
1    59    506 SINT=SINT+B(I,M)*A(I)
```

```
      P=SINT
      EX=1.
      D2=0
      a(21)=0.0
      a(22)=ex/(2.*gamma)*2./t/t
      a(23)=a(22)*3./t
      a(24)=0.0
      a(25)=ex/(2.*gamma)*(d2-1./gamma)*2./t/t
      a(26)=a(25)*3./t
      n=26
      DO 512 I=21,N
1   512 P=P-B(I,M)*A(I)
      P=5.193043*(3.0/5.0)-P*25.31549
      I=M
      GO TO(50,50,30,40),K
   20 K=2
      I=3
      T=TT
      D=DD
      GO TO 8
   30 K=3
      GO TO(33,34,35),KK
   33 D=DD
      T=TT
      KK=2
      IF(DD.GT.17.3987) GO TO 40
      I=1
      GO TO 8
   34 PTI=P
      I=3
      KK=3
      T=TT
      D=DD
      GO TO 8
   35 PTIII=P
      f=(15.-t)/5.
            P=F*PTI+(1.-F)*PTIII
      GO TO 50
   40 IF(K.EQ.3)K1=3
      GO TO(41,42,43,44),K4
   41 K=4
      I=2
      K4=2
      D=DD
      IF(K1.EQ.0)T=TT
      GO TO 8
   42 PIID=P
      D=17.3987
      IF(T.LT.5.2014) D=DSATL(T)*1000./4.0026
      K4=3
      GO TO 8
   43 PIIDC=P
      I=1
      K4=4
      GO TO 8
   44 PIDC=P
      P=PIDC+(PIID-PIIDC)
      K4=1
      IF(K1.EQ.3)GO TO 30
```

```
D Line# 1      7
    119     50 cv=p
    120 C         RETURNS CV - J/GM-K
    121        RETURN
    122        END
```

Name	Type	Offset	P	Class
A	REAL	2		
B	REAL	0		/COEFF /
D	REAL	138		
D2	REAL	142		
D3	REAL	146		
D4	REAL	150		
D5	REAL	154		
DD	REAL	110		
DGC	REAL	0	*	
EX	REAL	170		
EXP				INTRINSIC
F	REAL	218		
GAMMA	REAL	166		
I	INTEGER*4	130		
K	INTEGER*4	118		
K1	INTEGER*4	106		
K4	INTEGER*4	126		
KH	INTEGER*4	114		
KK	INTEGER*4	122		
M	INTEGER*4	162		
N	INTEGER*4	194		
P	REAL	202		
PIDC	REAL	234		
PIID	REAL	222		
PIIDC	REAL	230		
PTI	REAL	210		
PTIII	REAL	214		
R	REAL	158		
SINT	REAL	174		
SQRT				INTRINSIC
T	REAL	134		
T12	REAL	186		
T14	REAL	190		
TT	REAL	4	*	

Name	Type	Size	Class
COEFF		324	COMMON
CV	REAL		FUNCTION
D2DBDT	REAL		FUNCTION
DBDT	REAL		FUNCTION
DSATL	REAL		FUNCTION

```
 Pass One     No Errors Detected
            122 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION CP(D,T)
      2 C         CALCULATES SPECIFIC HEAT AT CONSTANT PRESSURE(J/GM-K) FOR
      3 C         INPUTS OF DENSITY(GM/CC) AND TEMPERATURE(K)
      4        CP=CV(D,T)+(T*(DPDT(D,T)**2)/((D**2)*DPDD(D,T)))*.1013278
      5        RETURN
      6        END
```

Name	Type	Offset	P	Class
D	REAL	0	*	
T	REAL	4	*	

Name	Type	Size	Class
CP	REAL		FUNCTION
CV	REAL		FUNCTION
DPDD	REAL		FUNCTION
DPDT	REAL		FUNCTION

```
 Pass One     No Errors Detected
               6 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION VISCDT(D,T)
      2 C      W.G.STEWARD,S DATA  23 JUNE 71  GOOD FOR 3 K LE. T LE. 300 K AND
      3 C      0 LE. P LE. 100 ATMOS
      4 C      INPUT UNITS IN GM/CC AND DEGREE K
      5 C      RETURNS VISCOSITY IN POISE (GM/CM-S)
      6        tl=alog(t)
      7        TL2=TL*TL
      8        TL3=TL**3
      9        ANNE= -0.135311743/TL + 1.00347841 + 1.20654649*TL
     10       1 -0.149564551*TL2 + 0.0125208416*TL3
     11        BETTY=D*(-47.5295259/TL + 87.6799309 -42.0741589*TL
     12       1+8.33128289*TL2 -0.589252385*TL3)
     13        CAROL= D*D*(547.309267/TL - 904.870586 + 431.404928*TL
     14       1-81.4504854*TL2 + 5.37008433*TL3)
     15        DAGMAR=D**3*(-1684.39324/TL + 3331.08630 - 1632.19172*TL
     16       1+308.804413*TL2 - 20.2936367*TL3)
     17        VISCDT=1.0E-06*EXP(ANNE + BETTY + CAROL + DAGMAR)
     18        RETURN
     19        END

Name     Type         Offset P Class

ALOG                           INTRINSIC
ANNE     REAL            14
BETTY    REAL            18
CAROL    REAL            22
D        REAL             0 *
DAGMAR   REAL            26
EXP                            INTRINSIC
T        REAL             4 *
TL       REAL             2
TL2      REAL             6
TL3      REAL            10


Name     Type           Size   Class

VISCDT   REAL                  FUNCTION


 Pass One     No Errors Detected
              19 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION THCON(D,T)
      2 C         INPUTS REQUIRED ARE DENSITY(D) IN GM/CC AND TEMP. (T) IN K
      3 C         CALCULATES THE THERMAL CONDUCTIVITY OF HELIUM USING THE
      4 C         EQUATIONS GIVEN IN NBS 631 -- UNITS ARE MW/CM-K
      5 C
      6 C         PROGRAMMED ONLY FOR TEMPERATURES OF 300K OR BELOW
      7 C                    D.P. BROWN 1/30/78
      8        X=ALOG(T)
      9        A=EXP(-4.3612+(1.9250*X)-(0.52544*X*X)+(0.090046*X**3)-
     10       1(0.0054774*X**4))
     11        B=EXP(4.7471-(5.3641*X)+(3.4640*X*X)-(1.0702*X**3)+(0.15713*X**4)
     12       1-(0.0089214*X**5))
     13        C=2.2109+187.74/T-1281.1/T**2+3645.2/T**3-3986.7/T**4
     14        BC=EXP(B*D+C*D*D)
     15        crit=0.0
     16        if(t.ge.11.83)go to 12
     17        if(d.gt.0.12)go to 12
     18        CP1=CP(D,T)
     19        CPCR=CP(D,11.83)
     20        CRIT=0.0026*(CP1-CPCR)
     21        if(crit.lt.0.0)crit=0.0
     22 12     THCON=A*BC+CRIT
     23        RETURN
     24        END
```

Name	Type	Offset	P	Class
A	REAL	6		
ALOG				INTRINSIC
B	REAL	10		
BC	REAL	18		
C	REAL	14		
CP1	REAL	26		
CPCR	REAL	34		
CRIT	REAL	22		
D	REAL	0	*	
EXP				INTRINSIC
T	REAL	4	*	
X	REAL	2		

Name	Type	Size	Class
CP	REAL		FUNCTION
THCON	REAL		FUNCTION

Pass One No Errors Detected
24 Source Lines

```
D Line# 1      7                                Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION FINDD(PI,TI,DI)
      2 C         RETURNS DENSITY IN GM/CC FOR INPUTS OF PRESSURE IN ATM AND TEMP
      3 C         IN DEGREE K.  FUNCTION ITERATES TO FIND DENSITY USING THE
      4 C         D,P,T RELATIONS FROM DTPRESS.  DI IS THE ESTIMATED APPROXIMATE
      5 C         DENSITY AT WHICH THE ITERATION WILL START.  IF DI IS SET TO
      6 C         -1., THE FUNCTION WILL MAKE ITS OWN INITIAL ESTIMATE OF
      7 C         DENSITY.  IN THE REGION OF LIQUID/VAPOR EQUILIBRIUM, THIS
      8 C         ESTIMATE BY THE FUNCTION GIVES INDETERMINANT RESULTS AND
      9 C         DI SHOULD BE SET BY THE PROGRAM TO GIVE DESIRED OUTPUT (DSATL
     10 C         FOR LIQUID AND DSATV FOR VAPOR PROPERTIES)
     11        key=0
     12        CONST=4.0026E-03
     13        T=TI
     14        P=PI
     15        JAN=0
     16        if(p.le.0.)go to 80
     17        if(t.le.0.)go to 81
     18        if(p.le.25.)go to 35
     19        if(p.ge.pmelt(t))go to 82
     20 35     if(t.ge.2.172)go to 39
     21        if(p.lt.vpa(t))go to 50
     22        if(t.lt.1.762)go to 83
     23        if(p.lt.plamda(t))go to 83
     24    39  d=di
     25        if(di.gt.0.0)go to 7
     26     1  if(t.le.5.2014)go to 2
     27        if(t.le.13.0)go to 60
     28        dml=10.0*p/t
     29        if((t.le.25.0).and.(p.ge.(2.5*t)))dml=45.0
     30        d=dml*const
     31        go to 7
     32    60  tl=alog(t)
     33        ptc=exp(-18.9831+22.6628*tl-8.2979*tl*tl+1.1116*tl**3)
     34        if((p/ptc.lt.0.8).or.(p/ptc.gt.1.25))go to 56
     35        dml=17.4+11.6*alog10(p)
     36        d=dml*const
     37        go to 7
     38    56  dml=45.0
     39        if(p.le.ptc)dml=10.0*p/t
     40        d=dml*const
     41        go to 7
     42     2  if(p.gt.vpa(t))go to 5
     43        dml=12.5*p/t
     44        d=dml*const
     45        go to 7
     46     5  dml=36.0*p**0.077
     47        d=dml*const
     48        if(p.gt.2.5)go to 7
     49        s=(1.0-t/5.2014)**(1.0/3.0)
     50        d=0.12874326*s-0.43128217*s**2+1.7851911*s**3
     51       1-3.3509624*s**4+3.0344216*s**5-1.098129*s**6+0.06964
     52 7      dint=d
     53        do 10 i=1,20
1    54        P2=DTPRESS(D,T)
1    55        if(abs(p-p2).le.(1.e-4*p))go to 20
1    56        dp=dpdd(d,t)
1    57        CORR=(P2-P)/DP
1    58        if(abs(corr).le.(1.e-4*d))go to 20
1    59        if(i.eq.1)go to 46
```

```
D Line# 1      7
1     60        if((corr/c1).ge.-0.8)go to 46
1     61        corr=-0.8*c1
1     62     46 C1=CORR
1     63        D=D-CORR
1     64        if(key.eq.1)write(*,171)i,d,p2,dp
1     65 171    format(' i=',i5,' d=',e10.3,' p2=',f6.3,' dp=',f8.4)
1     66     10 CONTINUE
      67 c      addition of biscetion iterative scheme
      68 c      should Newton-Rapson's method fails to converge
      69 172    format(' bisectional iterative method is being introduced')
      70        max=20
      71        iter=0
      72        d1=dint
      73        p1=dtpress(d1,t)
      74        if(p1.gt.p)go to 130
      75        do 125 i=1,15
1     76        d2=d1+i*0.05
1     77        p2=dtpress(d2,t)
1     78        if(p2.gt.p)go to 140
1     79 125    continue
      80        go to 140
      81 130    continue
      82        d2=d1
      83        do 135 i=1,15
1     84        d1=0.9*d1
1     85        p1=dtpres(d1,t)
1     86        if(p1.lt.p)go to 140
1     87 135    continue
      88 140    continue
      89        d3=(d1+d2)/2.
      90        p3=dtpress(d3,t)
      91        iter=iter+1
      92        findd=d3
      93        if(key.eq.1)write(*,173)iter,p,p1,p2,p3,d1,d2,d3
      94 173    format(' iter=',i3,' p=',f6.3,' p1=',f6.3,' p2=',f6.3,
      95       c' p3=',f6.3,/'        d1=',e10.3,' d2=',e10.3,' d3=',e10.3)
      96        if((abs(p3-p).lt.1.e-4*p).or.(abs(d3-d2).lt.1.e-4*d3))return
      97        if(iter.gt.max)go to 150
      98        if(iter.gt.max-3)write(*,173)iter,p,p1,p2,p3,d1,d2,d3
      99        if(p3.gt.p)p2=p3
     100        if(p3.gt.p)d2=d3
     101        if(p3.lt.p)p1=p3
     102        if(p3.lt.p)d1=d3
     103        go to 140
     104 150    write(*,170)p,t,di,d3
     105    170 format('  the iterative program to find the density did not conve
     106       lrge',/,'  p,t, trial density, final density =',4f10.3)
     107        return
     108     20 FINDD=D
     109        RETURN
     110     50 findd=12.188*p/t*const
     111        return
     112     80 write(*,180)
     113        findd=.1601
     114        return
     115    180 format(' pressure less than zero, cannot calculate density, set de
     116       lnsity = .1601 grams/cc')
     117     81 write(*,181)
     118        findd=.1601
```

```
    119        return
    120   181 format(' temperature less than zero, cannot calculate density, set
    121      2 density = .1601 grams/cc')
    122    82 write(*,182)p,t
    123       findd=.24015
    124       return
    125   182 format('   helium is solid at p,t = ',2f10.3,' density set =.24015
    126      3 grams/cc')
    127    83 write(*,183)p,t
    128       findd=.1601
    129       return
    130   183 format(' helium is superfluid at p, t =',2f10.3,' density set =
    131      4.1601 grams/cc')
    132       END
```

Name	Type	Offset	P	Class
ABS				INTRINSIC
ALOG				INTRINSIC
ALOG10				INTRINSIC
C1	REAL	86		
CONST	REAL	6		
CORR	REAL	82		
D	REAL	34		
D1	REAL	200		
D2	REAL	212		
D3	REAL	224		
DI	REAL	8	*	
DINT	REAL	58		
DML	REAL	38		
DP	REAL	74		
EXP				INTRINSIC
I	INTEGER*4	62		
ITER	INTEGER*4	196		
JAN	INTEGER*4	18		
KEY	INTEGER*4	2		
MAX	INTEGER*4	192		
P	REAL	14		
P1	REAL	204		
P2	REAL	66		
P3	REAL	228		
PI	REAL	0	*	
PTC	REAL	46		
S	REAL	54		
T	REAL	10		
TI	REAL	4	*	
TL	REAL	42		

133

Name	Type	Size	Class
DPDD	REAL		FUNCTION
DTPRES	REAL		FUNCTION
FINDD	REAL		FUNCTION
PLAMDA	REAL		FUNCTION
PMELT	REAL		FUNCTION
VPA	REAL		FUNCTION

```
D Line# 1      7
 Pass One     No Errors Detected
            133 Source Lines
```

```
D Line# 1      7                                   Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION FINDH(PT,TT)
      2 C         RETURNS ENTHALPY(J/GM) FOR INPUTS OF PRESSURE(ATM) AND TEMP(K)
      3        COMMON DI
      4        DT=FINDD(PT,TT,DI)
      5        FINDH=ENTHAL(DT,TT)
      6        RETURN
      7        END
```

Name	Type	Offset	P	Class
DI	REAL	0		/COMMQQ/
DT	REAL	2		
PT	REAL	0	*	
TT	REAL	4	*	

Name	Type	Size	Class
COMMQQ		4	COMMON
ENTHAL	REAL		FUNCTION
FINDD	REAL		FUNCTION
FINDH	REAL		FUNCTION

```
 Pass One     No Errors Detected
               7 Source Lines
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION VPH(P,H)
      2 C         OUTPUT IS LITERS/MOE, = 1.0/DML
      3 C         EQUATION  AS OF OCTOBER 1973
      4        DIMENSION F(36),C(36)
      5        DATA  C/-266.93513035,    1280.2153408,    -2138.6936689,
      6       1    1556.9746112,    -492.87815978,     55.369549758,
      7       2   569.40420832,     -2772.2878447,     4585.9849389,
      8       3    -3248.2021750,     975.00193915,     -99.693986960,
      9       4-474.12071656,     2312.0935094,     -3748.2352037,
     10       5    2554.7370305,     -710.47752015,     61.733962941,
     11       6180.72899664,     -869.47722083,     1371.0065073,
     12       7-890.38871888,     223.08026374,     -14.262360338,
     13       8     -26.175173305,     123.09822037,     -187.68556180,
     14       9    114.84668088,     -24.755912805,     0.65638085005,
     15       A     -0.018096064160,     0.0013453499824,     -0.0048916728215,
     16       B  -0.0036528943910,     -0.00085167781862,   -0.36035220794/
     17        mh=5
     18        mp=6
     19        r=2.0772258
     20        ra=20.500625
     21        HTC=16.3713+2.21774*P-0.002606*P*P
     22        X1=2.82394+0.064469*P-5.615E-06*P*P
     23        x=h/htc
     24        tih=0.4*h/r-x1
     25        PX=P**0.2
     26        P2=SQRT(P)
     27        DENOM=1.+X*X*X*X
     28        XI=1.0
     29        XX=1.0/SQRT(X)
     30        F(1)=XI/DENOM
     31        DO 30  I=2,MH
1    32        IA=(I-1)*MP+1
1    33        IB=(I-2)*MP+1
1    34     30 F(IA)=F(IB)*XX
     35        DO 40  I=1,MH
1    36        DO 36  J=2,MP
2    37        IC=(I-1)*MP+J
2    38        ID=(I-1)*MP+J-1
2    39     36 F(IC)=F(ID)*PX
1    40     40 CONTINUE
     41        F(31)=ALOG((1.+TIH/20.)/21.)
     42        f(32)=f(31)*f(31)
     43        f(33)=f(32)*f(32)
     44        f(34)=f(33)*f(31)
     45        F(35)=F(34)*F(31)
     46        DO 41  K=31,35
1    47     41 F(K)=F(K)*P
     48        F(36)=P2/(1.+X*X)
     49        YCALC=0.0
     50        DO 45  I=1,36
1    51     45 YCALC=YCALC+C(I)*F(I)
     52        TCALC=TIH+YCALC
     53        VCALC=TCALC*(RA/P-5.636E-08*RA*P)+0.14022*RA
     54        VPH=VCALC
     55        return
     56        end

Name     Type         Offset P Class
```

D Line# 1 7

ALOG				INTRINSIC
C	REAL	146		
DENOM	REAL	330		
F	REAL	2		
H	REAL	4	*	
HTC	REAL	306		
I	INTEGER*4	342		
IA	INTEGER*4	350		
IB	INTEGER*4	354		
IC	INTEGER*4	370		
ID	INTEGER*4	374		
J	INTEGER*4	362		
K	INTEGER*4	378		
MH	INTEGER*4	290		
MP	INTEGER*4	294		
P	REAL	0	*	
P2	REAL	326		
PX	REAL	322		
R	REAL	298		
RA	REAL	302		
SQRT				INTRINSIC
TCALC	REAL	386		
TIH	REAL	318		
VCALC	REAL	390		
X	REAL	314		
X1	REAL	310		
XI	REAL	334		
XX	REAL	338		
YCALC	REAL	382		

Name	Type	Size	Class
VPH	REAL		FUNCTION

Pass One No Errors Detected
56 Source Lines

```
      FUNCTION TPH(P,H)
      DIMENSION F(33), C(33)
C        EQUATION  AS OF OCTOBER 1973
C******************MODIFIED THIS FUNCTION 11/21/77 TO  CORRECT FOR
C******************DISAGREEMENT WITH FUNCTION DTPRESS.  THIS FUNCTION
C******************NOW ESTIMATES T AND THEN CALLS PHTD FOR ITERATIVE
C******************SOLUTION USING DTPRESS. -- D.P.BROWN
      DATA C/13.878902654,   -9.1294555014,     1.4753132952,
     A-0.064635371239,    -14.085703051,      -4.3222209397,
     B1.4429523076,    -0.29763970504,     18.815813408,
     C-3.3611959286,    3.2417847568,     5.8053275951,
     D-4.6708598724,    6.2461795885,     -26.313269947,
     E-2.3614901630,    4.6428182119,     -2.9101156545,
     F0.90838981816,     -76.900659027,     47.935651420,
     G34.867402925,     -0.23446642474,     -1.8712988904,
     H-4.1613166989,     -6.5729084580,     2.0788999721,
     I-0.17257458796,     0.10532918090,     -0.40806745000,
     J-1.1548143734,     4.5591407995,     21.976664669/
C                  FOLLOWING TEST EXTENDS RANGE OF THIS FUNCTION
C                  INTO LIQUID-VAPOR AND SUBCOOLED LIQUID REGION---
C                  ADDED 4/24/78--D.P.BROWN
      if(p.gt.2.245)go to 20
      tliq=vptemp(p)
      dliq=dsatl(tliq)
      dvap=dsatv(tliq)
      hliq=enthal(dliq,tliq)
      hvap=enthal(dvap,tliq)
      if(h.gt.hvap)go to 20
      if(h.lt.hliq)go to 12
      tph=tliq
      RETURN
12    tph=tliq-(hliq-h)/3.
      di=dliq+1.8*(tliq-tph)
      call phtd(p,h,tph,dph)
      di=-1.0
      RETURN
20    CONTINUE
      r=2.0772258
      cp=2.5*r
      al=0.137115
      wl=0.6
      A2=4.0
      TI=H/CP-2.82394-0.064469*P+5.615E-06*P*P
      HTC=16.3713+2.21774*P-0.002606*P*P
      HCPI=8.6467+1.3220*P-0.001823*P*P
      z=h/hcpi
      x=alog(z)
      x2=x*x
      vl=alog(h/htc)
      PL=ALOG(1.0+P)
      pl2=pl*pl
      pl3=pl*pl2
      pl4=pl*pl3
      a=(z/wl)**2.10
      b=wl**(-2.10)
      fa=1.0-((1.+b)/(1.+a)-1.)**2
      a=(z/wl)**1.85
      b=wl**(-1.85)
      fb=1.0-((1.+b)/(1.+a)-1.)**2
```

```
      a=(z/w1)**1.60
      b=w1**(-1.60)
      fc=1.0-((1.+b)/(1.+a)-1.)**2
      a=(z/w1)**1.35
      b=w1**(-1.35)
      fk=1.0-((1.+b)/(1.+a)-1.)**2
      FE=  (1.0+A1*ALOG(Z/2.0))/(1.0+(Z/2.0)**5)
      A4=10.*VL*VL
      IF (A4.GE.30.) A4=30.
      FF=VL*EXP(-1.0*A4)*EXP(-P/4.0)
      FG=VL*EXP(-1.0*A2*VL*VL)*EXP(-P/4.0)
      FI=X2/(1.0+1.0*Z**3)
      fh=x*fi
      fm=x*fh
      F(1)=FA*PL
      F(2)=FA*PL2
      F(3)=FA*PL3
      F(4)=FA*PL4
      F(5)=FB*PL
      F(6)=FB*PL2
      F(7)=FB*PL3
      F(8)=FI*PL4
      F(9)=FC*PL2
      F(10)=FC*PL3
      F(11)=FK*PL
      F(12)=FI*PL3
      F(13)=FF
      F(14)=FF*PL
      F(15)=FI*PL2
      F(16)=FG
      F(17)=FG*PL
      F(18)=FG*PL2
      F(19)=FG*PL3
      F(20)=FE*EXP(-1.0*P/3.50)
      F(21)=FE*EXP(-1.0*P/4.00)
      F(22)=FE*EXP(-1.0*P/3.00)
      F(23)=FI
      F(24)=FC*PL
      F(25)=FK*PL2
      F(26)=FH*PL2
      F(27)=FH*PL3
      F(28)=FH*PL4
      F(29)=FK*PL4
      F(30)=FF*PL2
      F(31)=FF*PL3
      F(32)=FH*PL
      F(33)=FI*PL
      ycalc=0.0
      do 30 k=1,33
1   30 YCALC=YCALC+C(K)*F(K)
      TPH=TI+YCALC
      CALL PHTD(P,H,TPH,DPH)
      return
      end
```

Name	Type	Offset	P	Class
A	REAL	378		
A1	REAL	322		

A2	REAL	330		
A4	REAL	406		
ALOG				INTRINSIC
B	REAL	382		
C	REAL	134		
CP	REAL	318		
DI	REAL	306		
DLIQ	REAL	274		
DPH	REAL	310		
DVAP	REAL	282		
EXP				INTRINSIC
F	REAL	2		
FA	REAL	386		
FB	REAL	390		
FC	REAL	394		
FE	REAL	402		
FF	REAL	410		
FG	REAL	414		
FH	REAL	422		
FI	REAL	418		
FK	REAL	398		
FM	REAL	426		
H	REAL	4	*	
HCPI	REAL	342		
HLIQ	REAL	290		
HTC	REAL	338		
HVAP	REAL	298		
K	INTEGER*4	434		
P	REAL	0	*	
PL	REAL	362		
PL2	REAL	366		
PL3	REAL	370		
PL4	REAL	374		
R	REAL	314		
TI	REAL	334		
TLIQ	REAL	266		
VL	REAL	358		
W1	REAL	326		
X	REAL	350		
X2	REAL	354		
YCALC	REAL	430		
Z	REAL	346		

Name	Type	Size	Class
DSATL	REAL		FUNCTION
DSATV	REAL		FUNCTION
ENTHAL	REAL		FUNCTION
PHTD			SUBROUTINE
TPH	REAL		FUNCTION
VPTEMP	REAL		FUNCTION

Pass One No Errors Detected
113 Source Lines

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION HPS(P,S)
      2 C         OUTPUT IS ENTHALPY IN J/GM FOR INPUTS OF PRESSURE IN ATM AND
      3 C         ENTROPY IN J/GM-K
      4 C         EQUATION  AS OF OCTOBER 1973
      5 C
      6        DIMENSION F(35), C(35)
      7        DATA  C/0.620 5652653,     -24.55 275204,     22.2 7261674,
      8       A-6.7492782747,     0.40121573691,     -0.052643461931,
      9       B362.16988376,     -327.37113461,     99.186574011,
     10       C-7.0819815035,     -52.181285811,     -1894.2905449,
     11       D1710.7602365,     -506.93398133,     38.921676371,
     12       E254.28630050,     4525.5682326,     -4118.9023813,
     13       F1182.8344037,     -93.218939174,     -449.98406724,
     14       G-5494.5892540,     4955.0341804,     -1381.9212763,
     15       H110.01849787,     297.26956809,     3267.8083338,
     16       I-2893.3124992,     787.24009925,     -62.871925138,
     17       J-66.277471159,     -754.78760174,     654.27817221,
     18       K-174.47708487,     13. 3 361536/
     19        r=2.0772258
     20        cp=2.5*r
     21        A1=3.6785757*P**0.4
     22        A2=0.0001077*P
     23        A3=14.6649+0.33479*P-0.00002913*P*P
     24 C
     25 C                    INTERPOLATES TO FIND H IF CONDITIONS ARE LIQ-VAPOR
     26 C                    MIXTURE
     27 C
     28        IF(P.GT.2.245) GO TO 20
     29        T=VPTEMP(P)
     30        if(t.ge.5.2014)go to 20
     31        dl=dsatl(t)
     32        DG=DSATV(T)
     33        SL=ENTROP(DL,T)
     34        SG=ENTROP(DG,T)
     35        fl=1.0-(s-sl)/(sg-sl)
     36        If((fl.le.0.).or.(fl.ge.1.))go to 20
     37        hl=enthal(dl,t)
     38        HG=ENTHAL(DG,T)
     39        HPS=FL*HL+(1.0-FL)*HG
     40        return
     41 C
     42 C                    CALCULATION FOR NON-LIQ-VAPOR MIXTURE
     43 C
     44     20 PL=ALOG(1.0+P)
     45        x=exp(s/cp)-1.0
     46        x3=x**(1.0/3.0)
     47        F(1)=X3*X3/(X+1.0)
     48        DO 16  J=2,5
1    49     16 F(J)=F(J-1)*PL
     50        DO 21  I=2,7
1    51        I1=1+(I-1)*5
1    52        I2=1+(I-2)*5
1    53        F(I1)=F(I2)/X3
1    54        DO 18  J=2,5
2    55        I3=(I-1)*5+J
2    56        I4=(I-1)*5+J-1
2    57     18 F(I3)=F(I4)*PL
1    58     21 CONTINUE
     59        ycalc=0.0
```

```
D Line# 1      7
     60        do 30 k=1,35
1    61     30 YCALC=YCALC+C(K)*F(K)
     62        HI=A1*EXP((S-A2)/CP)+A3
     63        hps=hi+ycalc
     64        return
     65        end
```

Name	Type	Offset	P	Class
A1	REAL	290		
A2	REAL	294		
A3	REAL	298		
ALOG				INTRINSIC
C	REAL	142		
CP	REAL	286		
DG	REAL	318		
DL	REAL	310		
EXP				INTRINSIC
F	REAL	2		
FL	REAL	342		
HG	REAL	354		
HI	REAL	406		
HL	REAL	346		
I	INTEGER*4	378		
I1	INTEGER*4	382		
I2	INTEGER*4	386		
I3	INTEGER*4	390		
I4	INTEGER*4	394		
J	INTEGER*4	374		
K	INTEGER*4	402		
P	REAL	0	*	
PL	REAL	362		
R	REAL	282		
S	REAL	4	*	
SG	REAL	334		
SL	REAL	326		
T	REAL	302		
X	REAL	366		
X3	REAL	370		
YCALC	REAL	398		

Name	Type	Size	Class
DSATL	REAL		FUNCTION
DSATV	REAL		FUNCTION
ENTHAL	REAL		FUNCTION
ENTROP	REAL		FUNCTION
HPS	REAL		FUNCTION
VPTEMP	REAL		FUNCTION

Pass One No Errors Detected
65 Source Lines

```
Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
    1        SUBROUTINE PHTD(P,H,T,D)
    2 C         ENTER THIS SUBROUTINE WITH P,H, AND ESTIMATED T
    3 C         SUBROUTINE CALCULATES CORRECT T AND D
    4        COMMON DI
    5        xt=t
    6        ti=t
    7        d=di
    8        DO 20 LIMIT=1,10
    9        D=FIND D(P,XT,D)
   10        XH=ENTHAL(D,XT)
   11        IF (ABS (H-XH)/H.LT.0.0001) GO TO 30
   12        XT=XT+(H-XH)/CP(D,XT)
   13     20 CONTINUE
   14        go to 40
   15     30 t=xt
   16        if(xt.ge.1.762)return
   17        write(*,80)p,h,xh,ti,xt
   18     80 format(' temp found in subroutine phtd is out of range',/,
   19       1' p,h,xh,ti,xt =',5f10.3)
   20        RETURN
   21 c      addition of biscetion iterative scheme
   22 c      should the Newton-Raphson's method fails
   23     40 CONTINUE
   24        max=25
   25        iter=0
   26        t1=t
   27        d1=findd(p,t1,di)
   28        h1=enthal(d1,t1)
   29        if(h1.gt.h)go to 130
   30        t2=t1
   31        do 125 i=1,15
   32        t2=1.5*t2
   33        d2=findd(p,t2,di)
   34        h2=enthal(d2,t2)
   35        if(h2.gt.h)go to 140
   36 125    continue
   37        write(*,81)p,t2,h2,h
   38 81     format(' phtd can not find h2.gt.h, p,t2,h2,h are',4f10.4)
   39        t=t2
   40        d=d2
   41        return
   42 130    continue
   43        t2=t1
   44        d2=d1
   45        h2=h1
   46        do 135 i=1,15
1  47        t1=0.8*t1
1  48        d1=findd(p,t1,d1)
1  49        h1=enthal(d1,t1)
1  50        if(h1.lt.h)go to 140
1  51 135    continue
   52        write(*,82)p,t1,h1,h
   53 82     format(' phtd can not find h1.lt.h, p,t1,h1,h are',4f10.4)
   54        t=t1
   55        d=d1
   56        return
   57 140    continue
   58        t=(t1+t2)/2.
   59        d=findd(p,t,d1)
```

```
      xh=enthal(d,t)
      iter=iter+1
      if(abs(xh-h)/h.lt.0.0001)return
      if(iter.gt.max)go to 150
      if(iter.gt.max-3)write(*,171)iter,t1,t2,t,d1,d2,d,h1,h2,xh,h,p
171   format(' phtd iter=',i5,' t1=',f9.3,' t2=',f9.3,' t3=',f9.3,
     c/' d1=',e10.3,' d2=',e10.3,' d3=',e10.3,
     c/' h1=',e10.3,' h2=',e10.3,' xh=',e10.3,' h=',e10.3,' p=',f7.3)
      if(xh.lt.h)go to 145
      t2=t
      d2=d
      h2=xh
      go to 140
145   continue
      t1=t
      d1=d
      h1=xh
      go to 140
150   continue
      write(*,90)p,h,xh,ti,xt
   90 format('  iteration to find t,d(p,h) did not converge',/
     1,' p, h, xh, ti, xt =',5f9.3)
      RETURN
      END
```

Name	Type	Offset	P	Class
ABS				INTRINSIC
D	REAL	12	*	
D1	REAL	128		
D2	REAL	152		
DI	REAL	0		/COMMQQ/
H	REAL	4	*	
H1	REAL	136		
H2	REAL	160		
I	INTEGER*4	148		
ITER	INTEGER*4	120		
LIMIT	INTEGER*4	10		
MAX	INTEGER*4	116		
P	REAL	0	*	
T	REAL	8	*	
T1	REAL	124		
T2	REAL	144		
TI	REAL	6		
XH	REAL	18		
XT	REAL	2		

84

Name	Type	Size	Class
COMMQQ		4	COMMON
CP	REAL		FUNCTION
ENTHAL	REAL		FUNCTION
FINDD	REAL		FUNCTION
PHTD			SUBROUTINE

Pass One No Errors Detected
84 Source Lines

```
      1        SUBROUTINE PSTD(P,S,T,D)
      2 C         ENTER THIS SUBROUTINE WITH P,S, AND ESTIMATED T
      3 C         SUBROUTINE CALCULATES CORRECT T AND D
      4        d=-1.0
      5        xt=t
      6        ti=t
      7        DO 20 LIMIT=1,15
1     8        D=FIND D(P,XT,D)
1     9        XS=ENTROP (D,XT)
1    10        IF (ABS (S-XS)/S.LT.0.0001) GO TO 30
1    11        C2=XT*(S-XS)/CP(D,XT)
1    12        if(limit.eq.1)go to 18
1    13        if((c2/c1).lt.-0.7)c2=-0.7*c1
1    14     18 XT=XT+C2
1    15        C1=C2
1    16     20 CONTINUE
     17        go to 40
     18     30 t=xt
     19        if(xt.ge.1.762)return
     20        write(*,80)p,s,xs,ti,xt
     21     80 format(' temp found in subroutine pstd is out of range',/,
     22       1' p,s,xs,ti,xt =',5f9.3)
     23        return
     24 c      Addition of bisection iterative scheme
     25 c      should the Newton-Raphson's method fails to converge
     26 40     continue
     27        max=25
     28        iter=0
     29        t1=t
     30        d1=findd(p,t1,d)
     31        s1=entrop(d1,t1)
     32        if(s1.gt.s)go to 130
     33        t2=t1
     34        d2=d1
     35        do 125 i=1,15
1    36        t2=1.5*t2
1    37        d2=findd(p,t2,d2)
1    38        s2=entrop(d2,t2)
1    39        if(s2.gt.s)go to 140
1    40 125    continue
     41        write(*,81)p,t2,s2,s
     42 81     format(' pstd cna not find s2.gt.s, p,t2,s2,s are',4f10.4)
     43        t=t2
     44        d=d2
     45        return
     46 130    continue
     47        t2=t1
     48        d2=d1
     49        h2=h1
     50        do 135 i=1,15
1    51        t1=0.8*t1
1    52        d1=findd(p,t1,d1)
1    53        s1=entrop(d1,t1)
1    54        if(s1.lt.s)go to 140
1    55 135    continue
     56        write(*,82)p,t1,s1,s
     57 82     format(' pstd can not find s1.lt.s, p,t1,s1,s are',4f10.4)
     58        t=t1
     59        d=d1
```

```
      return
140   continue
      t=(t1+t2)/2.
      d=findd(p,t,d1)
      xs=entrop(d,t)
      iter=iter+1
      if(abs(xs-s)/s.lt.0.0001)return
      if(iter.gt.max)go to 150
      if(iter.gt.max-3)write(*,171)iter,t1,t2,t,d1,d2,d,s1,s2,xs,s,p
171   format(' pstd iter=',i5,' t1=',f9.3,' t2=',f9.3,' t=',f9.3,
     c/' d1=',e10.3,' d2=',e10.3,' d=',e10.3,
     c/' s1=',e10.3,' s2=',e10.3,' xs=',e10.3,' s=',e10.3,' p=',f7.3)
      if(xh.lt.h)go to 145
      t2=t
      d2=d
      s2=xs
      go to 140
145   continue
      t1=t
      d1=d
      s1=xs
      go to 140
150   continue
      write(*,90)p,s,xs,ti,xt
   90 format(' iteration to find t,d(p,s) did not converge',/
     1,' p, s, xs, ti, xt =',5f9.3)
      RETURN
      END
```

Name	Type	Offset	P	Class
ABS				INTRINSIC
C1	REAL	34		
C2	REAL	26		
D	REAL	12	*	
D1	REAL	134		
D2	REAL	154		
H	REAL	496		
H1	REAL	230		
H2	REAL	226		
I	INTEGER*4	158		
ITER	INTEGER*4	126		
LIMIT	INTEGER*4	10		
MAX	INTEGER*4	122		
P	REAL	0	*	
S	REAL	4	*	
S1	REAL	142		
S2	REAL	166		
T	REAL	8	*	
T1	REAL	130		
T2	REAL	150		
TI	REAL	6		
XH	REAL	492		
XS	REAL	18		
XT	REAL	2		

88

Name	Type	Size	Class
CP	REAL		FUNCTION
ENTROP	REAL		FUNCTION
FINDD	REAL		FUNCTION
PSTD			SUBROUTINE

Pass One No Errors Detected
88 Source Lines

```
D Line# 1      7
      1        FUNCTION FLIQPH(P,H)
      2 C
      3 C                    RETURNS FRACTION LIQUID IN LIQUID - VAPOR MIXTURE
      4 C                    FOR INPUTS OF PRESSURE(ATM) AND ENTHALPY(J/GM)
      5 C                    OF THE MIXTURE
      6 C
      7        T=VPTEMP(P)
      8        DL=DSATL(T)
      9        DG=DSATV(T)
     10        HL=ENTHAL(DL,T)
     11        HG=ENTHAL(DG,T)
     12        fliqph=1.0-(h-hl)/(hg-hl)
     13        if(fliqph.ge.0.0)go to 12
     14        write(*,11)p,h,fliqph
     15     11 format(//' fraction liq is .lt. 0'/' p=',f8.3,' h=',f8.3,
     16       1' fliqph=',f8.3//)
     17        return
     18     12 if(fliqph.le.1.0)return
     19        write(*,15)p,h,fliqph
     20     15 format(//' fraction liq is .gt. 1'/' p=',f8.3,' h=',f8.3,
     21       1' fliqph=',f8.3//)
     22        return
     23        END
```

Name	Type	Offset	P	Class
DG	REAL	18		
DL	REAL	10		
H	REAL	4	*	
HG	REAL	34		
HL	REAL	26		
P	REAL	0	*	
T	REAL	2		

Name	Type	Size	Class
DSATL	REAL		FUNCTION
DSATV	REAL		FUNCTION
ENTHAL	REAL		FUNCTION
FLIQPH	REAL		FUNCTION
VPTEMP	REAL		FUNCTION

Pass One No Errors Detected
23 Source Lines

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.20 02/84
      1        FUNCTION FLIQPS(P,S)
      2 C
      3 C                    RETURNS FRACTION LIQUID IN LIQUID-VAPOR MIXTURE
      4 C                    FOR INPUTS OF PRESSURE(ATM) AND ENTROPY(J/GM-K) OF
      5 C                    THE MIXTURE
      6 C
      7        T=VPTEMP(P)
      8        DL=DSATL(T)
      9        DG=DSATV(T)
     10        SL=ENTROP(DL,T)
     11        SG=ENTROP(DG,T)
     12        fliqps=1.0-(s-sl)/(sg-sl)
     13        if((fliqps.lt.0.).or.(fliqps.gt.1.))write(*,15)p,s,t,fliqps
     14     15 format(//' fraction liq. is out of limits'/' p=',f8.3,' s=',f8.3,
     15       1' t=',f8.3,' fliqps=',f8.3//)
     16        return
     17        END
```

Name	Type	Offset	P	Class
DG	REAL	18		
DL	REAL	10		
P	REAL	0	*	
S	REAL	4	*	
SG	REAL	34		
SL	REAL	26		
T	REAL	2		

Name	Type	Size	Class
DSATL	REAL		FUNCTION
DSATV	REAL		FUNCTION
ENTROP	REAL		FUNCTION
FLIQPS	REAL		FUNCTION
VPTEMP	REAL		FUNCTION

Pass One No Errors Detected
17 Source Lines

```
D Line# 1      7                                   Microsoft FORTRAN77 V3.20 02/84
      1        BLOCK DATA
      2        common/coeff/aa(27),bb(27),cc(27)
      3        data cc/          -1.4802195348e-8,4.1721791119e-7,-2.3326553271
      4       1E-7,4.085511088E-7,1.0900567964E-5,-5.0060952775E-5,1.1312765043
      5       2E-4,-1.2539843287E-4,1.9661380688E-6,1.7122932666E-4,2.3051000563
      6       3E-4,-9.65647391E-4,-3.6027 735292E-5,1.6079946555E-3,-2.7441763615
      7       4E-2,.14739506957,-.43559344838,1.3447956078,-1.7040375125,.9026267
      8       54040,5.6875644111E-3,-1.4438146625E-1,3.3768874851E-3,1.0754201218
      9       6E-6,-4.5264622308E-5,3.8597388864E-5,-.0005/
     10        data bb/          -4.2287454626e-8,4.4529354413e-7,-1.0246150954
     11       1E-5,8.5254608956E-5,-2.5163069255E-4,3.2877709285E-4,-1.060195758
     12       2E-4,-1.0687738074E-4,-3.2120950632E-5,1.415901897E-4,1.4725630701
     13       4E-3,-2.618354941E-3,2.0461501117E-5,1.2746996288E-3,-2.0272929583
     14       5E-2,7.4648036615E-2,-.17217966521,.51053439738,-.40178202697,
     15       6.26829864632,7.906601204E-3,-8.9393485656E-2,-.15076580053,
     16       72.6882494327E-6,-3.3794316835E-5,-2.4495951195E-5, -.0005/
     17        data aa/-1.5096862619e-7,6.4640898904e-7,4.1362357367e-5,
     18       1-3.7910190353E-4,1.3806454049E-3,-2.5085412058E-3,2.3697560398E-3
     19       2,-9.5726461066E-4,3.7405931828E-5,-6.4103220333E-4,
     20       31.8579366177E-3,7.4007986606E-4,1.4792568148E-4,-3.2531355477E-3,
     21       41.9518739286E-2,-.10571817135,.33164944449,-.51130022535,
     22       53.9940004906E-1,-.15555244471,4.906264031E-3,-2.6148004377E-2,
     23       63.4221685545E-2,5.4159662622E-6,-1.0687806777E-5,-8.9484651869E-6,
     24       7-.0025/
     25        END
```

Name	Type	Offset	P	Class
AA	REAL	0		/COEFF /
BB	REAL	108		/COEFF /
CC	REAL	216		/COEFF /

Name	Type	Size	Class
BLKDQQ			BLOCK DATA
COEFF		324	COMMON

```
 Pass One     No Errors Detected
              25 Source Lines
```

Revised 1/23/86

DISTRIBUTION

BLDG. 911

AD Library
(H. Martin)
G. Bunce
R. Casey
H. Foelsche
J. Grisoli
P. Hughes (1) + 1
for each author
D. Lazarus
D. Lowenstein
T. Sluyters

FOR SSC PAPERS

FNAL

W. Fowler

LBL

R. Donaldson (2 copies)
P. Limon
C. Taylor
M. Tigner
R. Wolgast

TAC

F.R. Huson
Technical Library

BLDG. 902

D. Brown
J.G. Cottingham
P. Dahl
A. Flood
M. Garber
C. Goodzeit
A. Greene
R. Gupta
E. Kelly
R. Louttit
A. Prodell
W. Sampson
R. Shutt
J. Sondericker
P. Wanderer
E. Willen

BLDG. 460

N. Samios

BLDG. 510

H. Gordon
T. Kycia
L. Leipuner
S. Lindenbaum
R. Palmer
M. Sakitt
M. Tannenbaum
L. Trueman

BLDG. 1005S

J. Claus
E. Courant
F. Dell
E. Forsyth
H. Hahn
K. Jellett
S.Y. Lee
Z. Parsa
G. Parzen
P. Reardon
S. Tepekian

BLDG. 725

M. Barton
H. Halama

S&P in Magnet Division for all MD papers

S&P in Cryogenic Division for all Cryo Papers
(and J. Briggs, R. Dagradi, H. Hildebrand
W. Kollmer)