LBL-21859
UC-11

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

EARTH SCIENCES DIVISION

DOCUMENTATION FOR COMPUTER CODE NACL

O. Weres, J.C. Peiper, K.S. Pitzer,
and R. Pabalan

February 1987

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

LBL-21859

Documentation for Computer Code NACL

O. Weres

Earth Sciences Division, Lawrence Berkeley Laboratory, 50E
University of California, Berkeley, California 94720

and

J.C. Peiper, K.S. Pitzer, and R. Pabalan

Department of Chemistry and Lawrence Berkeley Laboratory
University of California, Berkeley, California 94720

February 1987

Introduction

The computer program NACL incorporates the empirical model of the thermodynamic properties of the system NaCl - H_2O recently published by Pitzer *et al.* (*1*). NACL is derived from the research codes used by Pitzer *et al.* to analyze the experimental data and fix the parameters in their model. NACL calculates values for all thermodynamic properties which are identical to values tabulated in Ref. 1.

NACL is written in VAX/VMS FORTRAN, and was developed on a VAX 8600 computer. Machine specific features have been avoided, and NACL should require few changes to compile and run with other compilers and computers.

A sample output and full code listing of NACL are appended to this document. For one year following the publication of this document, the code will be made available to interested users on 5.25" floppy diskette in MS-DOS 2.11 format. Please send a formatted diskette and a stamped, self-addressed mailer to Oleh Weres, Lawrence Berkeley Laboratory, 50E, Berkeley, CA 94720. Please put your name and address on the diskette.

Input and output

NACL accepts interactive input, and produces an output file (NACL.OUT) in print format. After invoking NACL, the user is asked to provide values of temperature, pressure, and NaCl molality. If the pressure specified is less than the vapor pressure of pure water at the temperature specified, the pressure is reset to equal the saturation pressure. NACL will then calculate many thermodynamic properties of the solution.

Two tabular formats are available for the output, and the user is interactively asked to select one or both of these. The first format ("Table 1") presents a listing of all quantities that have been calculated, properly identified and logically grouped. The symbols used in Table 1 are the variable names in the code. The second format ("Table 2"), presents just the quantities that are tabulated in Ref. 1. The values in Table 2 are explicitly related to the tables in Ref. 1. Table 2 may be extended by removing statement 105 in the subroutine TABLE2. With this statement removed, Table 2 will be extended to include "Class 2" and "Class 3" quantities, which are not tabulated in Ref. 1, and cannot be validated by comparison with Ref. 1. "Class 2" quantities have been used in calculating quantities tabulated in Ref.1, and have thereby been indirectly validated. "Class 3" quantities have not been validated by either means.

The calculation may be repeated, with output for additional values of t, P, c appended to the same output file.

Program structure

NACL is clearly structured and well commented. The code is provided as three files. The first file (NACL.FOR) contains the main program NACL which manages interactive input, and subroutines TABLE1 and TABLE2 which generate the corresponding output. The subroutines which actually calculate the thermodynamic properties are called from the main program. If so desired, the first file may be discarded, and the remainder of the program supplied with another main program or integrated into a larger program. The second file (NACLB.FOR) contains the heart of the model. The third file (NACLC.FOR) calculates the required properties of pure water based on the empirical formulas of Haar *et al.* (*2*). The dielectric constant of pure water and its derivatives are calculate in subroutine DIELEC using the empirical formula of Bradley and Pitzer (*3*).

Acknowledgements

This work was supported by the Office of Basic Energy Sciences of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Final validation and correction of NACL against the Tables in Ref. 1 were done by Mr. C.H. Jun.

References

1. K.S. Pitzer, J.C. Peiper, and R.H. Busey, Thermodynamic properties of aqueous sodium chloride solutions. J. Phys. Chem. Ref. Data, **13**, 1-102 (1984).
2. L. Haar, J.S. Gallagher, and G.S. Kell, Proc. 9th international conference on the properties of steam, J. Straub and K. Scheffler, eds., p. 69, Pergamon Press (Oxford, 1980).
3. D.J. Bradley and K.S. Pitzer, J. Phys. Chem. **83**, 1599 (1979).

List of Symbols

The following is a list of constants and variables used in the NaCl program. With few exceptions, constants and variables assigned to a common block in the program are grouped together in this list. The common block names are listed alphabetically.

The Gibbs energy, enthalpy, entropy, and heat capacity are dimensionless (divided by R or RT), whereas the volume and its derivatives with respect to temperature or pressure have dimensions of cc/mol, cc/mol/K, cc/mol/bar, except where noted otherwise.

Common Block/ cval2/

tr	reference temperature, 298.15 K
pr	reference pressure, 177 bars
mr	reference molality, 5.550825 moles NaCl/kg. water
mw	molar mass of water, 18.01534 g.
n1	moles of solvent

Common Block / debye/

ap,ah,as,	Debye-Hückel limiting law parameters for the
aj,av,aa,	osmotic coefficient, enthalpy, entropy, heat
ab	capacity, volume, expansivity, and compressibility, respectively.

Common Block/ delta/

v	specific volume of water, cc/g
vt	$= (\partial v/\partial T)_P$
vp	$= (\partial v/\partial P)_T$
vtt	$= (\partial^2 v/\partial T^2)_P$
vpp	$= (\partial^2 v/\partial P^2)_T$
vtp	$= (\partial^2 v/\partial T \partial P)$

Common Block/ dielec/

d	dielectric constant of water
dt	$= (\partial D/\partial T)_P$
dtt	$= (\partial^2 D/ \partial T^2)_P$
dp	$= (\partial D/ \partial P)_T$
dpp	$= (\partial^2 D/ \partial P^2)_T$
dtp	$= (\partial^2 D/ \partial T \partial P)$

Common Block /excess/

gex	excess Gibbs energy of the solution
g1ex, g2ex	partial molal excess Gibbs energy of solvent and solute
hex	excess enthalpy of the solution
h1ex, h2ex	partial molal excess enthalpy of solvent and solute
sex	excess entropy of the solution
s1ex,s2ex	partial molal excess entropy of solvent and solute
cpex	excess heat capacity of the solution
cp1ex,cp2ex	partial molal excess heat capacity of solvent and solute
vex	excess volume of the solution
v1ex,v2ex	partial molal excess volume of solvent and solute
vtex,vt1ex, vt2ex	temperature derivatives of vex, v1ex and v2ex, respectively
vpex, vp1ex, vp2ex	pressure derivatives of vex, v1ex and v2ex, respectively

Common Block/haar/ – standard state properties of pure liquid water at any T and P

f10	Helmholtz free energy
g10	Gibbs free energy
u10	internal energy
h10	enthalpy
s10	entropy
cv10	heat capacity at constant volume
cp10	heat capacity at constant pressure
cs10	heat capacity along saturation curve
v10	specific volume, cc/g
a10	expansivity, per Kelvin
b10	compressibility, per (10^5 pascals)

Common Block/ideal/ – thermodynamic properties of water as an ideal gas at T and P and referenced to 0 K

f0	Helmholtz free energy
g0	Gibbs free energy
u0	internal energy
h0	enthalpy
s0	entropy
cv0	heat capacity at constant volume
cp0	heat capacity at constant pressure

Common Block/partial/ – partial molal properties of water and NaCl(aq)

g1,g2	partial molal Gibbs energy of solvent and solute
s1,s2	partial molal entropy of solvent and solute
h1,h2	partial molal enthalpy of solvent and solute
cp1,cp2	partial molal heat capacity of solvent and solute
v1,v2	partial molal volume of solvent and solute
vt1,vt2	temperature derivatives of v1 and v2, respectively
vp1,vp2	pressure derivatives of v1 and v2, respectively

Common block/pg/

phi	osmotic coefficient of NaCl(aq)
gam	activity coefficient of NaCl(aq)

Common Block /ref/ – thermodynamic properties of NaCl(aq) in its standard state of the hypothetical one molal ideal solution at any T and P.

g20	standard state Gibbs energy of NaCl(aq), $(\overline{G}^{o}_{2(T,P)} - \overline{H}_{2(298,1\ atm)})/RT$
s20	standard state entropy of NaCl(aq)
h20	standard state enthalpy of NaCl(aq)
cp20	standard state heat capacity of NaCl(aq)
v20	standard state molal volume of NaCl(aq)
vt20,vp20	temperature and pressure derivatives of v20, respectively

Common Block/solid/ – standard state properties of pure solid NaCl

g2s	Gibbs energy of solid NaCl, $(G^{o}_{T,P} - H_{298,1\ atm})/RT$
h2s	enthalpy of solid NaCl
s2s	entropy of solid NaCl
cp2s	heat capacity of solid NaCl
v2s	molal volume of solid NaCl
vt2s,vp2s	temperature and pressure derivatives of v2s, respectively.

Common Block/ soln/

dgs0	standard Gibbs energy of solution, ΔG_s°
dss0	standard entropy of solution, ΔS_s°
dhs0	standard enthalpy of solution, ΔH_s°
dcps0	standard heat capacity of solution, ΔCp_s°
dvs0	standard volume of solution, ΔV_s°
dvts0,dvps0	temperature and pressure derivatives of dvs0, respectively

Common Block /virial/ – virial coefficients for the Pitzer equations

b0,b1,cp b0L,b1L,cpL b0s,b1s,cps b0j,b1j,cpj b0v,b1v,cpv b0a,b1a,cpa b0b,b1b,cpb	b0,b1 are pair-wise ion-interaction parameters and cp is the triplet ion-interaction parameter for Pitzer's equations for Gibbs energy. Symbols with letters L,s,j,v,a and b refer to parameters for enthalpy, entropy, heat capacity, volume, expansivity, and compressibility, respectively.

Others

i	ionic strength
m	molality
p	pressure, bars
ps	saturation pressure at temperature T
r	gas constant = 8.31440 J/mol/K = 83.1440 cc-bar/mol/K = 4.61518 cc-bar/g/K
si	square root of ionic strength
t	temperature, Kelvin

PROGRAM LISTING

```
c
c
c
      program nacl
c
c     This program calculates the properties of NaCl solutions
c     using the accurate empirical model described by Pitzer,
c     Peiper, and Busey, J. Phys. Chem. Ref. Data, V.13, 1-102 (1984)
c     and in Lawrence Berkeley Laboratory Report LBL-15512.
c
c     The tables in the cited references were generated using
c     an earlier version of this program, which was written by
c     J.C. Peiper.
c
c     The present version of NACL was developed by O. Weres with
c     assistance from C.H. Jun.
c
c     Version 2.00.
c
c                                   - O. Weres, March 1986
c
c     The main program accepts interactive input, calls the subroutines
c     that perform the actual calculations, and calls the subroutines
c     that produce printed output.
c
      implicit real*8(a-h,o-z)
      character*1 ans
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      open(unit=6,file='nacl.out',status='unknown')
c
c     Read tc,p,c interactively
c
      write(*,101)
  101 format('0Welcome to program NACL, v. 2.00.'/
     1 ' Interactive input.  Output is written to file NACL.OUT'/)
   27 continue
      write(*,114)
  114 format(' Enter T(C).')
      accept 102,tc
      t = tc + 273.15
      if(tc.lt.0d0.or.tc.gt.300d0) write (*,112)
  112 format(' Caution.  T(C) < 0. or T(C) > 300.')
      call psat(t)
  102 format(f10.1)
      write(*,103)
  103 format(' Enter P(bar)')
      accept 102, p
      if(p.gt.1000d0) write (*,108)
  108 format(' Caution.  P > 1000 bar')
      if(p.lt.ps) then
      write(*,109) ps
  109 format(' Caution.  P < Psat(T).  P is set = Psat(T) =',f10.3)
      p = ps
      endif
      write(*,104)
  104 format(' Enter C(molal)')
      accept 102,c
      if(c.lt.1d-20) then
      write(*,110)
  110 format(' Caution. C must be positive.  C is reset to 1e-20')
```

```
      c = 1d-20
      endif
      if(c.gt.6d0) write (*,111)
  111 format(' Caution.  C > 6 molal')
c
      write(6,30)
      call haar(t,p)
      call nacl1(t,p)
      call nacl2(t,p,c)
      call nacl3(t,p,c)
      write (*,107)
  107 format(' Do you want Table 1 (y or n)?')
      accept 106, ans
      if(ans.eq.'y'.or.ans.eq.'Y') call table1(t,p,c)
      write (*,113)
  113 format(' Do you want Table 2 (y or n)?')
      accept 106, ans
      if(ans.eq.'y'.or.ans.eq.'Y') call table2(t,p,c)
      write(*,105)
  105 format(' Do you want to go again (y or n) ?')
      accept 106,ans
  106 format(a1)
      if(ans.eq.'y' .or. ans.eq.'Y') go to 27
   30 format('1NACL calculates the properties of NaCl solutions using'
     1/' the accurate empirical model published by Pitzer, Peiper, and'
     2/' Busey, J. Phys. Chem. Ref. Data v. 13, pp. 1-102 (1984).'
     3/' The valid range is 0-300 C, 0-6 molal, Psat(T)-1000 bar.')
      stop
      end
c
c
c
      subroutine table1(t,p,c)
c
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,n1
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      common /debye/ ap,ah,as,aj,av,aa,ab
      common/delta/v,vt,vp,vtt,vpp,vtp
      common /diel/ d,dt,dtt,dp,dpp,dtp
      common /excess/ gex,g1ex,g2ex,hex,h1ex,h2ex,sex,s1ex,
     1 s2ex,cpex,cp1ex,cp2ex,vex,v1ex,v2ex,vtex,vt1ex,vt2ex,
     2 vpex,vp1ex,vp2ex
      common /haar/ f10,g10,u10,h10,s10,cv10,cp10,cs10,v10,a10,b10
      common /ideal/ f0,g0,u0,h0,s0,cv0,cp0
      common /partial/ g1,g2,s1,s2,h1,h2,cp1,cp2,v1,v2,vt1,vt2,
     1 vp1,vp2
      common /pg/ phi,gam
      common /ref/ g20,s20,h20,cp20,v20,vt20,vp20
      common /solid/ g2s,h2s,s2s,cp2s,v2s,vt2s,vp2s
      common /soln/ dgs0,dss0,dhs0,dcps0,dvs0,dvts0,dvps0
      common /vir/ b0,b0l,b0s,b0j,b0v,b0a,b0b,b1,b1l,b1s,b1j,
     1 b1v,b1a,b1b,cp,cpl,cps,cpj,cpv,cpa,cpb
      tc = t - 273.15
      write(6,2) t,tc,p,c
      write(6,28)
      write(6,29)
      write(6,21) ps,dpdt
      write(6,7) f10,g10,u10,h10,s10,cv10,cp10,cs10,v10,a10,b10
      write(6,8) f0,g0,u0,h0,s0,cv0,cp0
```

```
      write(6,5) v,vt,vp,vtt,vpp,vtp
      write(6,6) d,dt,dtt,dp,dpp,dtp
      write(6,4)ap,ah,as,aj,av,aa,ab
      write(6,31)
      write(6,9) b0,b0l,b0s,b0j,b0v,b0a,b0b
      write(6,10) b1,b1l,b1s,b1j,b1v,b1a,b1b
      write(6,11) cp,cpl,cps,cpj,cpv,cpa,cpb
      write(6,12) g20,s20,h20,cp20,v20,vt20,vp20
      write(6,13) gex,g1ex,g2ex,hex,h1ex,h2ex
      write(6,14) sex,s1ex,s2ex,cpex,cp1ex,cp2ex
      write(6,15) vex,v1ex,v2ex,vtex,vt1ex,vt2ex
      write(6,26) vpex,vp1ex,vp2ex
      write(6,16) g1,g2,s1,s2,h1,h2,cp1,cp2
      write(6,17) v1,v2,vt1,vt2,vp1,vp2
      write(6,18) phi,gam
      write(6,19) g2s,h2s,s2s,cp2s,v2s,vt2s,vp2s
      write(6,20) dgs0,dss0,dhs0,dcps0,dvs0,dvts0,dvps0
      write (6,32)
   32 format('1')
      return
    2 format('0Temperature, pressure and molality are'//
     1 ' t,tc = ',2f10.2,'    p = ',f10.3,'   m =',f7.3/)
   28 format('0Table 1 contains all calculated quantities.'/' See comme
     1nts in HAAR, RH0, NACl1 and NACL3 for further explanation.')
   29 format('0Properties of pure water are calculated first.'/)
   21 format('0Saturation pressure of pure water and temperature derivat
     1ive from PSAT'//' ps,dpdt =',1p2e13.4/)
    7 format('0Reference state properties for liquid water from HAAR'//
     1 ' f10,g10 =',1p2e13.4,'   u10,h10,s10 =',1p3e13.4//
     2 ' cv10,cp10,cs10 =',1p3e13.4,'   v10,a10,b10 =',1p3e13.4/)
    8 format('0Properties of water as ideal gas from HAAR'//
     1 ' f0,g0 =',1p2e13.4,'   u0,h0,s0 =',1p3e13.4,
     2 '   cv0,cp0 =',1p2e13.4/)
    5 format('0Specific volume of pure water and its derivatives over
     1 T and P from RH0'//
     2 ' v,vt,vp =',1p3e13.4,'   vtt,vpp,vtp =',1p3e13.4/)
    6 format('0Dielectric constant and its derivatives from DIELEC'//
     1 ' d,dt,dtt =',1p3e13.4,'   dp,dpp,dtp =',1p3e13.4/)
    4 format('0Debye-Huckel coefficients from DEBYE'//
     1 ' ap,ah,as =',1p3e13.3,'   aj,av =',1p2e13.4,
     2 '   aa,ab =',1p2e13.4/)
   31 format('0Virial coefficients from NACL1')
    9 format('0b0,b0l =',1p2e13.4,'   b0s,b0j =',1p2e13.4,
     1 '   b0v,b0a,b0b =',1p3e13.4)
   10 format('0b1,b1l =',1p2e13.4,'   b1s,b1j =',1p2e13.4,
     1 '   b1v,b1a,b1b =',1p3e13.4)
   11 format('0cp,cpl =',1p2e13.4,'   cps,cpj =',1p2e13.4,
     1 '   cpv,cpa,cpb =',1p3e13.4/)
   12 format('0Reference state properties for NaCl from NACL1'//
     1 ' g20,s20 =',1p2e13.4,'   h20,cp20 =',1p2e13.4,
     2 '   v20,vt20,vp20 =',1p3e13.4/)
   13 format('0Excess properties from EXCESS'//
     1 ' gex,g1ex,g2ex =',1p3e13.4,'   hex,h1ex,h2ex =',1p3e13.4)
   14 format('0sex,s1ex,s2ex =',1p3e14.4,'   cpex,cp1ex,cp2ex =',
     1 1p3e13.4)
   15 format('0vex,v1ex,v2ex =',1p3e15.4,'   vtex,vt1ex,vt2ex =',
     1 1p3e13.4)
   26 format('0vpex,vp1ex,vp2ex =',1p3e13.4/)
   16 format('0Partial molal quantities from NACL2'//
     1 ' g1,g2 =',1p2e13.4,'   s1,s2 =',1p2e13.4,
```

```
     2 ' '//' h1,h2 =',1p2e13.4,'   cp1,cp2 =',1p2e13.4)
    17 format('0v1,v2 =',1p2e13.4,'   vt1,vt2 =',1p2e13.4,
     1 '   vp1,vp2 =',1p2e13.4/)
    18 format('00smotic coefficient and mean ion activity coefficient fro
     1m NACL2.'//' phi,gam =',1p2e13.4/)
    19 format('0Properties of solid NaCl from NACL3'//
     1 ' g2s,h2s =',1p2e13.4,'   s2s,cp2s =',1p2e13.4,//
     2 ' v2s,vt2s,vp2s =',1p3e13.4/)
    20 format('0Change of free energy and other thermodynamic functions a
     1ccompanying dissolution of NaCl in water from NACL3'//
     2 ' dgs0,dss0 =',1p2e13.4,'   dhs0,dcps0 =',1p2e13.4,//
     3 ' dvs0,dvts0,dvps0 =',1p3e13.4/)
      end
c
c
c
      subroutine table2(t,p,c)
c
c     This subroutine generates output which may be compared to
c     the tables in report LBL-15512.
c                                    created by c.h.jun
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,n1,m2
      common /debye/ ap,ah,as,aj,av,aa,ab
      common /excess/ gex,g1ex,g2ex,hex,h1ex,h2ex,sex,s1ex,
     1      s2ex,cpex,cp1ex,cp2ex,vex,v1ex,v2ex,vtex,vt1ex,
     2      vt2ex,vpex,vp1ex,vp2ex
      common /haar/ f10,g10,u10,h10,s10,cv10,cp10,cs10,v10,a10,b10
      common /partial/ g1,g2,s1,s2,h1,h2,cp1,cp2,v1,v2,vt1,vt2,
     1      vp1,vp2
      common /pg/ phi,gam
      common /ref/ g20,s20,h20,cp20,v20,vt20,vp20
      common /soln/ dgs0,dss0,dhs0,dcps0,dvs0,dvts0,dvps0
      common /vir/ b0,b01,b0s,b0j,b0v,b0a,b0b,b1,b11,b1s,b1j,
     1      b1v,b1a,b1b,cp,cp1,cps,cpj,cpv,cpa,cpb
      common /cval2/ tr,pr,mr,mw,n1
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      common /delta/ v,vt,vp,vtt,vpp,vtp
      common /diel/ d,dt,dtt,dp,dpp,dtp
      common /ideal/ f0,g0,u0,h0,s0,cv0,cp0
      common /solid/ g2s,h2s,s2s,cp2s,v2s,vt2s,vp2s
c
c     in this routine r = 8.3144 j/k-mol
c
      data r/8.3144/
      data m2/58.44/
c
c     This subroutine generates a table whose entries correspond to the
c     tables in Report LBL-15512.  The original purpose of this
c     subroutine was to check values generated by this version of the
c     program against tables in the report.  Class 1 values, which are
c     normally printed out are those presented in the report,
c     and have been directly verified.  Note: the values for specific
c     entropy and specific enthalpy (tables 16 and 17) were incorrect
c     in LBL-15512, but have been corrected in the program and in the
c     journal article.
c
      tc=t-273.15
c
      f15 = ( 1.e3*v10 + c*v20 + vex )/( 1.e3 + c*m2 )
```

```
      f16 = ( n1*s10 + c*s20 + sex + 2.e0*c*(1.e0-dlog(c)) )/
     1      ( 1.e3 + c*m2 )
      f17 = ( n1*h1 + c*h2 )/( 1.e3 + c*m2 )
c
      write(6,100) t,tc,p,c
      write(6,101) g10,g20,ap,b0,b1,1.e3*cp/2.
      write(6,102) s10,s20,as,b0s,b1s,1.e3*cps/2.
      write(6,103) h10,h20,ah,1.e3*b0l,1.e3*b1l,1.e3*cpl/2.
      write(6,104) cp10,cp20,aj,1.e6*b0j,1.e6*b1j,1.e6*cpj/2.
      write(6,105) mw*v10,v20,av,1.e6*b0v,1.e6*cpv/2.
      write(6,106) 1.e3*mw*v10*a10,vt20,aa,1.e6*b0a,1.e9*cpa/2.
      write(6,107) -1.e3*mw*v10*b10,1.e3*vp20,1.e3*ab,1.e9*b0b
      write(6,108) gam
      write(6,109) phi
      write(6,110) dss0
      write(6,111) dhs0
      write(6,112) sex/c
      write(6,113) hex/c
      write(6,114) cpex/c
      write(6,115) 1./f15
      write(6,116) r*f16
      write(6,117) r*t*f17
  100 format('0Table 2.  The following variables (class 1) are tabul
     1ated in report LBL-15512'/' and are presented here in the same ord
     2er.'//' t,tc =',2f9.2,'     p =',f10.3,'     m =',f8.3/)
  101 format(//1x,'table 1   g10,g20,ap,b0,b1,1e3*cp/2',
     1       25x,5f10.4,f10.2)
  102 format(//1x,'table 2   s10,s20,as,b0s,b1s,1e3*cps/2',
     1       22x,5f10.4,f10.2)
  103 format(//1x,'table 3   h10,h20,ah,1e3*b0l,1e3*b1l,1e3*cpl/2',
     1       14x,f10.4,5f10.3)
  104 format(//1x,'table 4   cp10,cp20,aj,1e6*b0j,1e6*b1j,1e6*cpj/2',
     1       12x,6f10.3)
  105 format(//1x,'table 5   mw*v10,v20,av,1e6*b0v,1e6*cpv/2',
     1       19x,5f10.3)
  106 format(//1x,'table 6   1e3*mw*v10*a10,vt20,aa,1e6*b0a,1e9*cpa/2',
     1       10x,5f10.3)
  107 format(//1x,'table 7   -1e3*mw*v10*b10,1e3*vp20,1e3*ab,1e9*b0b',
     1       11x,4f10.3)
  108 format(//1x,'table 8   gam',47x,f10.3)
  109 format(/1x,'table 9   phi',47x,f10.3)
  110 format(/1x,'table 10  dss0',46x,f10.3)
  111 format(/1x,'table 11  dhs0',46x,f10.3)
  112 format(/1x,'table 12  sex/m',45x,f10.3)
  113 format(/1x,'table 13  hex/m',45x,f10.3)
  114 format(/1x,'table 14  cpex/m',44x,f10.3)
  115 format(/1x,'table 15  density',43x,f10.5)
  116 format(/1x,'table 16  specific entropy(j/g-k)',27x,f10.3)
  117 format(/1x,'table 17  specific enthalpy(j/g)',28x,f10.1)
c
c  The remaining code is normally skipped over.  Class 2 and 3
c  variables were not verified directly.  Class 2 variables
c  were used in the calculation of class 1 variables and have
c  thereby been verified indirectly.
c
c  To execute the remaining code, remove the following statement.
c
      if(tc.lt.1d10) return
      write(6,200)
      write(6,201)v,vt,vp,vtt,vpp,vtp
```

```
      write(6,202)d,dt,dtt,dp,dpp,dtp
      write(6,203)f10,u10,cv10
      write(6,204)f0,u0,cv0
      write(6,205)b1v
      write(6,206)gex,h1ex,h2ex,vex
      write(6,207)h1,h2
      write(6,208)h2s,s2s
c
      write(6,300)
      write(6,301)cs10
      write(6,302)g0,h0,s0,cp0
      write(6,303)b1a,b1b,cpb
      write(6,304)g1ex,g2ex
      write(6,305)s1ex,s2ex,cp1ex,cp2ex
      write(6,306)v1ex,v2ex,vtex,vt1ex,vt2ex
      write(6,307)g1,g2,s1,s2,cp1,cp2
      write(6,308)v1,v2,vt1,vt2,vp1,vp2
      write(6,309)g2s,cp2s,v2s,vt2s,vp2s
      write(6,310)dgs0,dcps0,dvs0,dvts0,dvps0
      write(6,311)ps,dpdt
c
  200 format(1x/////
     1' Class 2; variables used in calculation of class 1 variables.'
     2 ///)
  201 format(/1x,'v,vt,vp,vtt,vpp,vtp',6x,1p6e13.4)
  202 format(/1x,'d,dt,dtt,dp,dpp,dtp',6x,1p6e13.4)
  203 format(/1x,'f10,u10,cv10',12x,1p3e13.4)
  204 format(/1x,'f0,u0,cv0',16x,1p3e13.4)
  205 format(/1x,'b1v',22x,1pe13.4)
  206 format(/1x,'gex,h1ex,h2ex,vex',7x,1p4e13.4)
  207 format(/1x,'h1,h2',20x,1p2e13.4)
  208 format(/1x,'h2s,s2s',18x,1p2e13.4)
c
  300 format(/////1x,'Class 3 ; values not verified'///)
  301 format(/1x,'cs10',21x,1pe13.4)
  302 format(/1x,'g0,h0,s0,cp0',13x,1p4e13.4)
  303 format(/1x,'b1a,b1b,cpb',14x,1p3e13.4)
  304 format(/1x,'g1ex,g2ex',16x,1p2e13.4)
  305 format(/1x,'s1ex,s2ex,cp1ex,cp2ex',3x,1p4e13.4)
  306 format(/1x,'v1ex,v2ex,vtex,vt1ex,vt2ex',12x,1p5e13.4)
  307 format(/1x,'g1,g2,s1,s2,cp1,cp2',6x,1p6e13.4)
  308 format(/1x,'v1,v2,vt1,vt2,vp1,vp2',4x,1p6e13.4)
  309 format(/1x,'g2s,cp2s,v2s,vt2s,vp2s',3x,1p5e13.4)
  310 format(/1x,'dgs0,dcps0,dvs0,dvts0,dvps0',11x,1p5e13.4)
  311 format(/1x,'psat,dpdt',16x,1p2e13.4)
      return
      end
```

```
c
c
c
      subroutine nacl1(t,p)
c
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,nl
      common /cval2/ tr,pr,mr,mw,nl
      common /excess/ gex,glex,g2ex,hex,hlex,h2ex,sex,slex,
     1 s2ex,cpex,cplex,cp2ex,vex,vlex,v2ex,vtex,vtlex,vt2ex,
     2 vpex,vplex,vp2ex
      common /haar/ f10,g10,u10,h10,s10,cv10,cp10,cs10,v10,a10,b10
      common /ref/ g20,s20,h20,cp20,v20,vt20,vp20
      common /vir/ b0,b0l,b0s,b0j,b0v,b0a,b0b,b1,b1l,b1s,b1j,
     1 b1v,b1a,b1b,cp,cpl,cps,cpj,cpv,cpa,cpb
      common /zz/ z(53),zl(53),zh(53)
c
c     This program calculates the properties of NaCl solutions
c     using the accurate empirical model published by Pitzer,
c     Peiper, and Busey, J. Phys. Chem. Ref. Data, v.13, 1-102 (1984)
c     and in Lawrence Berkeley Laboratory Report LBL-15512.
c
c     The tables in the cited references were generated using
c     an earlier version of this program, which was written by
c     J.C. Peiper.
c
c     The present version of NACL was developed by O. Weres with
c     assistance from C.H. Jun.
c
c                                   - O. Weres, March 1986
c
        data zl /-71659.53087, 2.348333613, -8.366848370e-5,
     1          2.401816892e-9, 624.8820839, -5.369711934e-4,
     2          3.512696524e-7, 0.0, -110.74702, 0.03890080081,
     3          2.697345512e-6, -6.274687539e-10, -1.5267612e-5, 0.0,
     4          516.99706, -5.9960301e6, -656.81518, 24.87918316,
     5          -2.155273113e-5, 5.016685500e-8, 0.0, -4.4640952,
     6          0.01108709891, -6.447976135e-8, -2.323403150e-10, 0.0,
     7          -5.219487066e-6, 2.444520990e-10, 2.852706550e-13,
     8          -1.569623077, 2.233786380e-3, -6.393389110e-7,
     9          4.527057333e-11, 5.4151933, 0.0, 0.0, 0.0, 119.31966,
     x          -0.48309327, 1.4068095e-3, -4.2345814, -6.1084589,
     1          0.4074380280, -6.8152430e-6, -0.075354649,
     2          1.260901375e-4, 6.2480692e-8, 1.899437268e-8,
     3          -1.0731284e-10, 0.3213657227e0, -2.5382945e-4, 0.0, 0.0/
      data zh/  -71637.20299, 2.220901258, -7.799139523e-5,
     5          -4.809927222e-9, 624.6812520, 6.015978715e-4,
     6          3.406907417e-7, 2.196204308e-11, -110.74702,
     7          0.03949447337, -6.531347462e-7, -6.478189444e-10,
     8          -1.584201250e-5, 3.245200554e-9, 516.99706,
     9          -5.9960301e6, -656.81518, 24.86912950, 5.381275267e-5,
     x          -5.588746990e-8, 6.589326333e-12, -4.4640952,
     1          0.01110991383, -2.657339906e-7, 1.746006963e-10,
     2          1.046261900e-14, -5.307012889e-6, 8.634023325e-10,
     3          -4.178596200e-13, -1.579365943, 2.202282079e-3,
     4          -1.310550324e-7, -6.381368333e-11, 9.706578079,
     5          -0.02686039622, 1.534474401e-5, -3.215398267e-9,
     6          119.31966, -0.48309327, 1.4068095e-3, -4.2345814,
     7          -6.1084589, 0.4021779279, 2.2902837e-5, -0.075354649,
     8          1.531767295e-4, -9.0550901e-8, -1.538600820e-8,
```

```
     9          8.6926600e-11, 0.3531041360, -4.3314252e-4,
     x          -0.09187145529, 5.1904777e-4/
c
c     In this subroutine r = 83.144 bar-cu.cm/k-mol
c
        data mw,n1 /18.01533, 55.50825/
        data tr,pr,mr,r /298.15, 1.01325, 5.550825, 83.144/
c
c     Set array of z values.
c
      if(t.ge.338.15) then
      do 1 k=1,53
    1 z(k)=zh(k)
      else
      do 2 k=1,53
    2 z(k)=zl(k)
      end if
c
c     Calculate virial coefficients.
c
        b0 = z(17)/t + z(18) + z(19)*p + z(20)*p**2
     1                  + z(21)*p**3 + z(22)*dlog(t)
     2          + ( z(23) + z(24)*p + z(25)*p**2
     3                  + z(26)*p**3 )*t
     4          + ( z(27) + z(28)*p + z(29)*p**2 )*t**2
     5          + ( z(30) + z(31)*p + z(32)*p**2
     6                  + z(33)*p**3 )/(t-227e0)
     7          + ( z(34) + z(35)*p + z(36)*p**2
     8                  + z(37)*p**3 )/(680e0-t)
        b0l = -z(17)/t**2 + z(22)/t + z(23) + z(24)*p
     1                  + z(25)*p**2 + z(26)*p**3
     2          + 2e0*( z(27) + z(28)*p + z(29)*p**2 )*t
     3          - ( z(30) + z(31)*p + z(32)*p**2
     4                  + z(33)*p**3 )/(t-227e0)**2
     5          + ( z(34) + z(35)*p + z(36)*p**2
     6                  + z(37)*p**3 )/(680e0-t)**2
        b0s = z(18) + z(19)*p + z(20)*p**2 + z(21)*p**3
     1          + z(22)*(1e0+dlog(t))
     2          + 2e0*( z(23) + z(24)*p + z(25)*p**2
     3                  + z(26)*p**3 )*t
     4          + 3e0*( z(27) + z(28)*p + z(29)*p**2 )*t**2
     5          + ( z(30) + z(31)*p + z(32)*p**2
     6                  + z(33)*p**3 )*(1e0-t/(t-227e0))/(t-227e0)
     7          + ( z(34) + z(35)*p + z(36)*p**2
     8                  + z(37)*p**3 )*(1e0+t/(680e0-t))/(680e0-t)
        b0j = z(22)/t**2 + 2e0*( z(23) + z(24)*p + z(25)*p**2
     1                  + z(26)*p**3 )/t
     2          + 6e0*( z(27) + z(28)*p + z(29)*p**2 )
     3          + 454e0*( z(30) + z(31)*p + z(32)*p**2
     4                  + z(33)*p**3 )/(t*(t-227e0)**3)
     5          + 1360e0*( z(34) + z(35)*p + z(36)*p**2
     6                  + z(37)*p**3 )/(t*(680e0-t)**3)
        b0v = z(19) + 2e0*z(20)*p + 3e0*z(21)*p**2
     1          + ( z(24) + 2e0*z(25)*p + 3e0*z(26)*p**2 )*t
     2          + ( z(28) + 2e0*z(29)*p )*t**2
     3          + ( z(31) + 2e0*z(32)*p + 3e0*z(33)*p**2 )/(t-227e0)
     4          + ( z(35) + 2e0*z(36)*p + 3e0*z(37)*p**2 )/(680e0-t)
        b0a = ( z(19) + 2e0*z(20)*p + 3e0*z(21)*p**2 )/t
     1          + 2e0*z(24) + 4e0*z(25)*p + 6e0*z(26)*p**2
     2          + 3e0*( z(28) + 2e0*z(29)*p )*t
```

```
     4          - 227e0*( z(31) + 2e0*z(32)*p + 3e0*z(33)*p**2 )
     5                 /(t*(t-227e0)**2)
     6          + 680e0*( z(35) + 2e0*z(36)*p + 3e0*z(37)*p**2 )
     7                 /(t*(680e0-t)**2)
        b0b = 2e0*z(20) + 6e0*z(21)*p
     1          + ( 2e0*z(25) + 6e0*z(26)*p )*t
     2          + 2e0*z(29)*t**2
     3          + ( 2e0*z(32) + 6e0*z(33)*p )/(t-227e0)
     4          + ( 2e0*z(36) + 6e0*z(37)*p )/(680e0-t)
        b1 = z(38)/t + z(39) + z(40)*t + z(41)/(t-227e0)
        b1l = -z(38)/t**2 + z(40) - z(41)/(t-227e0)**2
        b1s = z(39) + 2e0*z(40)*t + z(41)*(1e0-t/(t-227e0))/(t-227e0)
        b1j = 2e0*z(40)/t + 454e0*z(41)/(t*(t-227e0)**3)
       b1v=0.
       b1a=0.
       b1b=0.
        cp = z(42)/t + z(43) + z(44)*p + z(45)*dlog(t)
     1          + ( z(46) + z(47)*p )*t
     2          + ( z(48) + z(49)*p )*t**2
     3          + ( z(50) + z(51)*p )/(t-227e0)
     4          + ( z(52) + z(53)*p )/(680e0-t)
        cpl = -z(42)/t**2 + z(45)/t + z(46) + z(47)*p
     1          + 2e0*( z(48) + z(49)*p )*t
     2          - ( z(50) + z(51)*p )/(t-227e0)**2
     3          + ( z(52) + z(53)*p )/(680e0-t)**2
        cps = z(43) + z(44)*p + z(45)*(1e0+dlog(t))
     1          + 2e0*( z(46) + z(47)*p )*t
     2          + 3e0*( z(48) + z(49)*p )*t**2
     3          + ( z(50) + z(51)*p )*(1e0-t/(t-227e0))/(t-227e0)
     4          + ( z(52) + z(53)*p )*(1e0+t/(680e0-t))/(680e0-t)
        cpj = z(45)/t**2 + 2e0*( z(46) + z(47)*p )/t
     1          + 6e0*( z(48) + z(49)*p )
     2          + 454e0*( z(50) + z(51)*p )/(t*(t-227e0)**3)
     3          + 1360e0*( z(52) + z(53)*p )/(t*(680e0-t)**3)
        cpv = z(44) + z(47)*t + z(49)*t**2 + z(51)/(t-227e0)
     1          + z(53)/(680e0-t)
        cpa = z(44)/t + 2e0*z(47) + 3e0*z(49)*t
     1          - 227e0*z(51)/(t*(t-227e0)**2)
     2          + 680e0*z(53)/(t*(680e0-t)**2)
       cpb=0.
c
c      Calculate reference state properties.
c
c        This routine calculates the properties of nacl(aq) in its
c        reference state (hypothetical 1 mol/kg solution at t and p).
c        The gibbs energy, enthalpy, entropy and heat capacity are
c        dimensionless, whereas the volume and its derivatives have
c        dimensions of cc/mol, cc/mol-k and cc/mol-bar (e.g., a20 and b20
c        are the derivatives (d v20 /dt) and (d v20 /dp), not the usual
c        quantities (d ln v20 /dt) and (d ln v20 /dp).
c
       call excess(t,p,mr)
        g20 = -10e0*g10 - gex/mr
     1          + ( z(1) + z(2)*p + z(3)*p**2 + z(4)*p**3 )/t
     2          + z(5) + z(6)*p + z(7)*p**2 + z(8)*p**3
     3          + z(9)*dlog(t)
     4          + ( z(10) + z(11)*p + z(12)*p**2 )*t
     5          + ( z(13) + z(14)*p )*t**2
     6          + z(15)/(t*(t-227e0)) + z(16)/(t*(680e0-t)**3)
        s20 = -10e0*s10 - sex/mr
```

```
     1          - z(5) - z(6)*p - z(7)*p**2 - z(8)*p**3
     2          - z(9)*(1e0 + dlog(t))
     3          - 2e0*( z(10) + z(11)*p + z(12)*p**2 )*t
     4          - 3e0*( z(13) + z(14)*p )*t**2
     5          + z(15)/(t-227e0)**2 - 3e0*z(16)/(680e0-t)**4
        h20 = -10e0*h10 - hex/mr
     1          + ( z(1) + z(2)*p + z(3)*p**2 + z(4)*p**3 )/t
     2          - z(9) - ( z(10) + z(11)*p + z(12)*p**2 )*t
     3          - 2e0*( z(13) + z(14)*p )*t**2
     4          + z(15)*(2e0*t-227e0)/(t*(t-227e0)**2)
     5          - z(16)*(4e0*t-680e0)/(t*(680e0-t)**4)
        cp20 = -10e0*cp10 - cpex/mr
     1          - z(9) - 2e0*( z(10) + z(11)*p + z(12)*p**2 )*t
     2          - 6e0*( z(13) + z(14)*p )*t**2
     3          - 2e0*z(15)*t/(t-227e0)**3
     4          - 12e0*z(16)*t/(680e0-t)**5
        v20 = -10e0*mw*v10 - vex/mr
     1          + r*( z(2) + 2e0*z(3)*p + 3e0*z(4)*p**2 )
     2          + r*t*( z(6) + 2e0*z(7)*p + 3e0*z(8)*p**2 )
     3          + r*t**2*( z(11) + 2e0*z(12)*p ) + r*t**3*z(14)
        vt20 = -10e0*mw*v10*a10 - vtex/mr
     1          + r*( z(6) + 2e0*z(7)*p + 3e0*z(8)*p**2 )
     2          + 2e0*r*t*( z(11) + 2e0*z(12)*p ) + 3e0*r*t**2*z(14)
        vp20 = 10e0*mw*v10*b10 - vpex/mr
     1          + r*( 2e0*z(3) + 6e0*z(4)*p )
     2          + r*t*( 2e0*z(7) + 6e0*z(8)*p ) + 2e0*r*t**2*z(12)
      return
      end
c
c
c
      subroutine nacl2(t,p,m)
c
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,n1
      common /cval2/ tr,pr,mr,mw,n1
      common /debye/ ap,ah,as,aj,av,aa,ab
      common /excess/ gex,g1ex,g2ex,hex,h1ex,h2ex,sex,s1ex,
     1 s2ex,cpex,cp1ex,cp2ex,vex,v1ex,v2ex,vtex,vt1ex,vt2ex,
     2 vpex,vp1ex,vp2ex
      common /haar/ f10,g10,u10,h10,s10,cv10,cp10,cs10,v10,a10,b10
      common /partial/ g1,g2,s1,s2,h1,h2,cp1,cp2,v1,v2,vt1,vt2,
     1 vp1,vp2
      common /pg/ phi,gam
      common /ref/ g20,s20,h20,cp20,v20,vt20,vp20
      common /vir/ b0,b01,b0s,b0j,b0v,b0a,b0b,b1,b11,b1s,b1j,
     1 b1v,b1a,b1b,cp,cp1,cps,cpj,cpv,cpa,cpb
c
c      In this routine r = 83.144 bar-cu.cm/k-mol
c
      data r/83.144/
        i = m
        si = dsqrt(i)
c
c     Calculate partial molal quantities.
c
      call excess(t,p,i)
        g1 = g10 - 2e0*m/n1 + g1ex
        g2 = g20  - 2e0*dlog(m) + g2ex
        s1 = s10 + 2e0*m/n1 + s1ex
```

```
      s2 = s20  - 2e0*dlog(m) + s2ex
      h1 = h10 + h1ex
      h2 = h20  + h2ex
      cp1 = cp10 + cp1ex
      cp2 = cp20  + cp2ex
      v1 = mw*v10 + v1ex
      v2 = v20  + v2ex
      vt1 = mw*v10*a10 + vt1ex
      vt2 = vt20  + vt2ex
      vp1 = -mw*v10*b10 + vp1ex
      vp2 = vp20  + vp2ex
c
c     Calculate phi and gam
c
      b = 2e0*b0  + b1 *(1e0-(1e0+2e0*si-2e0*i)*dexp(-2e0*si))/(2e0*i)
      alng = -ap*( si/(1e0+1.2*si) + dlog(1e0+1.2*si)/0.6 )
     1          + b*m + 1.5*cp *m**2
      gam = dexp(alng)
      phi = 1e0 - ap*si/(1e0+1.2*si)
     1          + m*( b0  + b1 *dexp(-2e0*si) )
     2          + cp *m**2
      return
      end
c
c
c
      subroutine excess(t,p,i)
c
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,n1
      common /cval2/ tr,pr,mr,mw,n1
      common /debye/ ap,ah,as,aj,av,aa,ab
      common /excess/ gex,g1ex,g2ex,hex,h1ex,h2ex,sex,s1ex,
     1 s2ex,cpex,cp1ex,cp2ex,vex,v1ex,v2ex,vtex,vt1ex,vt2ex,
     2 vpex,vp1ex,vp2ex
      common /vir/ b0,b01,b0s,b0j,b0v,b0a,b0b,b1,b11,b1s,b1j,
     1 b1v,b1a,b1b,cp,cp1,cps,cpj,cpv,cpa,cpb
c
c     In this routine r=83.144 bar-cu.cm/k-mol
c
      data r/83.144/
c
c      All quantities are dimensionless, except
c      for the volume and its derivatives.
c
      m=i
      si=dsqrt(i)
      f0 = dexp(-2e0*si)
      f1 = dlog(1e0+1.2*si)
      f2 = (1e0-(1e0+2e0*si)*f0)/(2e0*i)
      f3 = (1e0-(1e0+2e0*si-2e0*i)*f0)/(4e0*i)
      f4 = i**1.5
      gex = -4e0*ap*i*f1/1.2
     1          + 2e0*m**2*( b0
     2                  + b1 *f2
     3                  + m*cp /2e0 )
      g1ex = 2e0*ap*f4/(1e0+1.2*si)
     1          - 2e0*m**2*( b0  + b1 *f0 + m*cp  )
      g2ex = -2e0*ap*( si/(1e0+1.2*si) + f1/0.6 )
     1          + 4e0*m*( b0
```

```
2                   + b1 *f3 )
3           + 3e0*m**2*cp
   hex = ah*i*f1/1.2
1           - 2e0*t*m**2*( b0l
2                   + b1l *f2
3                   + m*cpl /2e0 )
   h1ex = (-ah*f4/(2e0*(1e0+1.2*si))
1           + 2e0*t*m**2*( b0l  + b1l *f0
2                   + m*cpl  ))/n1
   h2ex = ah*( si/(1e0+1.2*si) + f1/0.6 )/2e0
1           - 4e0*t*m*( b0l
2                   + b1l *f3 )
3           - 3e0*t*m**2*cpl
 sex=hex-gex
 s1ex=h1ex-g1ex
 s2ex=h2ex-g2ex
   cpex = aj*i*f1/1.2
1           - 2e0*(t*m)**2*( b0j
2                   + b1j *f2
3                   + cpj *m/2e0 )
   cp1ex = -aj*f4/(2e0*(1e0+1.2*si))
1           + 2e0*(t*m)**2*( b0j  + b1j *f0
2                   + cpj *m )
   cp2ex = aj*( si/(1e0+1.2*si) + f1/0.6 )/2e0
1           - 4e0*t**2*m*( b0j
2                   +b1j *f3 )
3           - 3e0*(t*m)**2*cpj
   vex = av*i*f1/1.2
1           + 2e0*r*t*m**2*( b0v
2                   + b1v *f2
3                   + cpv *m/2e0 )
   v1ex = av*f4/(2e0*(1e0+1.2*si))
1           - 2e0*r*t*m**2*( b0v  + b1v *f0
2                   + cpv *m )
   v2ex = -av*( si/(1e0+1.2*si) + f1/0.6 )/2e0
1           + 4e0*r*t*m*( b0v
2                   + b1v *f3 )
3           + 3e0*r*t*m**2*cpv
   vtex = aa*i*f1/1.2
1           + 2e0*r*t*m**2*( b0a
2                   + b1a *f2
3                   + cpa *m/2e0 )
   vt1ex = aa*f4/(2e0*(1e0+1.2*si))
1           - 2e0*r*t*m**2*( b0a  + b1a *f0
2                   + cpa *m )
   vt2ex = -aa*( si/(1e0+1.2*si) + f1/0.6 )/2e0
1           + 4e0*r*t*m*( b0a
2                   + b1a *f3 )
3           + 3e0*r*t*m**2*cpa
   vpex = ab*i*f1/1.2
1           + 2e0*r*t*m**2*( b0b
2                   + b1b *f2
3                   + cpb *m/2e0 )
   vp1ex = ab*f4/(2e0*(1e0+1.2*si))
1           - 2e0*r*t*m**2*( b0b  + b1b *f0
2                   + cpb *m )
   vp2ex = -ab*( si/(1e0+1.2*si) + f1/0.6 )/2e0
1           + 4e0*r*t*m*( b0b
2                   + b1b *f3 )
3           + 3e0*r*t*m**2*cpb
```

```
      return
      end
c
c
c
      subroutine nacl3(t,p,m)
c
      implicit real*8(a-h,o-z)
      real*8 i,m,mr,mw,n1
      common /cval2/ tr,pr,mr,mw,n1
      common /ref/ g20,s20,h20,cp20,v20,vt20,vp20
      common /solid/ g2s,h2s,s2s,cp2s,v2s,vt2s,vp2s
      common /soln/ dgs0,dss0,dhs0,dcps0,dvs0,dvts0,dvps0
        data s2sr,a,b,aa,bb /8.676, -1734.4, 37.589, -5.5254, -9.81e-4/
        data v2r /26.993/
        data cc,dd,ee /4.5e-8, 9.30e-5, -0.031728/
        data dhs0r /466.99/
c
c      In this routine r = 83.144 bar-cu.cm/k-mol
c
       data r/83.144/
      save s2sr,a,b,aa,bb,v2r,cc,dd,ee,dhs0r
c
c       This routine calculates the thermal properties of NaCl(c) based
c       on information from K.K. Kelley (U.S. Bur. Mines Report #584,
c       1960) and from the American Institute of Physics Handbook (3rd
c       ed., McGraw-Hill, New York, 1972).  The gibbs energy, enthalpy,
c       entropy and heat capacity are dimensionless.  The volume and its
c       derivatives have units of cc/mol, cc/mol-k and cc/mol-bar, e.g.
c       a2s and b2s are the derivatives (dv/dt) and (dv/dp), not the
c       quantities (d lnv/dt) and (d lnv/dp).
c
        g2s = -s2sr + a/t + b + aa*dlog(t) + bb*t
     1          + v2r*(p-pr)*dexp(cc*t**2 + dd*t + ee)/(r*t)
        h2s = a/t - aa - bb*t + v2r*(p-pr)*(2e0*cc*t**2 + dd*t + 1)
     1                  *dexp(cc*t**2 + dd*t + ee)/(r*t)
        s2s = s2sr - b - aa*(1e0+dlog(t)) - 2e0*bb*t
     1          + v2r*(p-pr)*(2e0*cc*t + dd)*dexp(cc*t**2 + dd*t + ee)/r
        cp2s = -aa-2e0*bb*t+v2r*(p-pr)*(4e0*cc**2e0*t**3+4e0*cc*dd*t**2
     1          + (dd**2 + 6e0*cc)*t + 2e0*dd)*dexp(cc*t**2 + dd*t + ee)/r
        v2s = v2r*dexp(cc*t**2 + dd*t + ee)
        vt2s = v2r*(2e0*cc*t + dd)*dexp(cc*t**2 + dd*t + ee)
       vp2s=0
c
c      Calculate properties of solution.
c
        dgs0 = g20  - g2s  + dhs0r/t
        dss0 = s20  - s2s
        dhs0 = h20  - h2s  + dhs0r/t
        dcps0 = cp20  - cp2s
        dvs0 = v20  - v2s
        dvts0 = vt20  - vt2s
        dvps0 = vp20  - vp2s
      return
      end
```

```
c
c
c
      subroutine haar(t,p)
c
      implicit real*8 (a-h,o-z)
      real*8 q(18),lnt
      common /alpha/ a(9,7),al,bet,gam,b0,c,d,e,bb0,b1,b2,b4
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      common/delta/v,vt,vp,vtt,vpp,vtp
      common /haar/ f,g,u,h,s,cv,cp,cs,v10,alfa,beta
      common /ideal/ f0,g0,u0,h0,s0,cv0,cp0
      data q /20.9662681977,1973.0271018,-.483429455355,6.05743189245,
     1   -68.3010170092,-106.4285576,-4.3135538513,.458155781,
     2   -.047754901883,.0041238460633,-.00027929052852,1.4481695261e-5,
     3   -5.6473658748e-7,1.6200446e-8,-3.303822796e-10,
     4   4.51916067368e-12,-3.70734122708e-14,1.37546068238e-16/
c
c     In this routine r = 4.61518 bar-cu.cm/k-g
c
      data r/4.61518/
      save q,r
c
c
c ***********************************************************************
c *                                                                     *
c *   HAAR calculates the thermodynamic properties of pure water using *
c *   the equations of Haar, Gallagher and Kell (Ninth International    *
c *   Conference on the Properties of Steam, Munich, 1979). The temp-   *
c *   erature t (in kelvins) and pressure p (in 10**5 pa) must be set   *
c *   by the calling routine before entering HAAR. The equations used   *
c *   herein are valid for t .gt. 273.15 k and pressures to 1 gpa.      *
c *                                                                     *
c *   The quantities returned by HAAR are --                            *
c *                                                                     *
c *        f     helmholtz free energy                                  *
c *        g     gibbs free energy                                      *
c *        u     internal energy                                        *
c *        h     enthalpy                                               *
c *        s     entropy                                                *
c *        cv    heat capacity at constant volume                       *
c *        cp    heat capacity at constant pressure                     *
c *        cs    heat capacity along the saturation curve (cs is set    *
c *                  equal to cp for pressures greater than saturation  *
c *                  pressure)                                          *
c *        v     volume, in cubic centimeters per gram                  *
c *        alfa  expansivity, per kelvin                                *
c *        beta  compressibility, per (10**5 pa)                        *
c *                                                                     *
c *   Unless otherwise indicated, all quantities are dimensionless.     *
c *   The helmholtz, gibbs and internal energies, the enthalpy and      *
c *   entropy are calculated on the practical absolute scale, i.e. are  *
c *   referenced to 0 k and 101325 pa with neglect of nuclear spin      *
c *   effects. The thermodynamic properties of water as an ideal gas    *
c *   at (t,p) and referenced to 0 k are given in common /ideal/, with  *
c *   similar dimensionalities.                                         *
c *                                                                     *
c *                                   J. Christopher Peiper             *
c *                                   29 March, 1982                    *
c *                                                                     *
```

```
c ***********************************************************************
c
c
      call rho(t,p)
      call debye(t,p)
c
      v10=v
      rh=1.0/v
      tau=t/647.073
      tt=t/100e0
      lnt=dlog(t)
      gz=-(q(1)+q(2)/t)*lnt
      hz=q(1)+q(2)*(1e0-lnt)/t
      cpz=q(1)-q(2)/t
c
      do 7 i=3,18
      gz=gz-q(i)*tt**(i-6)
      hz=hz+q(i)*(i-6)*tt**(i-6)
    7 cpz=cpz+q(i)*(i-6)*(i-5)*tt**(i-6)
      f0=gz-1e0
      g0=gz
      u0=hz-1e0
      h0=hz
      s0=h0-g0
      cv0=cpz-1e0
      cp0=cpz
c
      fz = f0 -
     1 dlog(1e0-y) - (bet-1e0)/(1e0-y) + (1e0+al+bet)/(2e0*(1e0-y)**2)
     2   + rh*(bb-gam*b) - (al-bet+3)/2e0 + dlog(r*rh*t/1.01325)
      uz = u0 + t*
     1 ((1e0/(y-1e0)+(bet-1e0)/(1e0-y)**2-(1e0+al+bet)/(1e0-y)**3)
     2                *yt
     3   - rh*(bbt-gam*bt))
      cvz=cv0 + 2e0*t*((1e0/(y-1e0) + (bet-1e0)/(1e0-y)**2
     1      - (1e0+al+bet)/(1e0-y)**3)*yt - rh*(bbt-gam*bt))
     2   +t**2*((2e0*(bet-1e0)/(1e0-y)**3 - 1e0/(1e0-y)**2
     3      - 3e0*(1e0+al+bet)/(1e0-y)**4)*yt**2
     4 +(1e0/(y-1e0)+(bet-1e0)/(1e0-y)**2-(1e0+al+bet)/(1e0-y)**3)*ytt
     5      -rh*(bbtt-gam*btt))
c
      df= 0e0
      du= 0e0
      dcv= 0e0
      do 20 n=1,9
      s1=a(n,1)
      s2=a(n,1)
      s3= 0e0
c
      do 17 j=2,7
      s1=s1+a(n,j)/tau**(j-1)
      s2=s2+j*a(n,j)/tau**(j-1)
   17 s3=s3+j*(1e0-j)*a(n,j)/tau**(j-1)
c
      df=df+s1*(1e0-er)**n/n
      du=du+s2*(1e0-er)**n/n
   20 dcv=dcv+s3*(1e0-er)**n/n
c
      f=fz+df/(r*t)
      u=uz+du/(r*t)
```

```
      cv=cvz+dcv/(r*t)
      g=f+p*v/(r*t)
      h=u+p*v/(r*t)
      s=h-g
      cp=cv-t*vt**2/(vp*r)
      cs=cp
      if (p.eq.ps) cs=cp-t*vt*dpdt/r
      alfa=vt/v
      beta=-vp/v
c
      return
      end
c
c
c
      subroutine debye(t,p)
c
      implicit real*8(a-h,o-z)
      real*8 n0,k
      common /debye/ ap,ah,as,aj,av,aa,ab
      common/delta/v,vt,vp,vtt,vpp,vtp
      common /diel/ d,dt,dtt,dp,dpp,dtp
      data pi,n0,e,k/3.141592658,6.022045e23,4.803242e-10,1.380662e-16/
c
c     In this routine r = 83.144 bar-cu.cm/k-mol
c
      data r /83.144/
      save pi,n0,e,k,r
c
c     This subroutine calculates the Debye-Huckel coefficients.
c
      rh=1e0/v
      call dielec(t,p)
      ap=dsqrt(2.*pi*n0*rh/1000.)*e**3/(d*k*t)**1.5/3.
      ah=-6.*t*ap*(dt/d +1./t + vt/(3.*v))
      aj=ah*t*(2e0/t-1.5*(dt/d+1e0/t+vt/(3e0*v))) - 6e0*t**2*ap*(dtt/d
     1   -(dt/d)**2-1e0/t**2+vtt/(3e0*v)-(vt/v)**2/3e0)
      av=2.*r*t*ap*(3.*dp/d + vp/v)
      aa=av*(1.+ah/(4.*ap))/t
     1   +2.*r*t*ap*(3.*(dtp/d -dt*dp/d**2)
     2   +vtp/v  -vt*vp/v**2)
      ab=-av**2/(4.*r*t*ap)
     1   +2.*r*t*ap*(3.*(dpp/d -(dp/d)**2) +vpp/v -(vp/v)**2)
      as=ah+4.*ap
      return
      end
c
c
c
      subroutine dielec(t,p)
c
      implicit real*8(a-h,o-z)
      common /diel/ d,dt,dtt,dp,dpp,dtp
      dimension u(9)
      data u/342.79,-.0050866,9.4690e-7,-2.0525,3115.9,-182.89,
     1-8032.5,4.2142e6,2.1417/
      save u
c
c     This subroutine calculates the dielectric constant of water
c     and its derivatives using the empirical formula presented by
```

```
c     Bradley and Pitzer, J. Phys. Chem. v. 83, 1599 (1979).
c
      b=u(7) + u(8)/t +u(9)*t
      bt=u(9)-u(8)/t**2
      btt=2.*u(8)/t**3
      c=u(4)+u(5)/(u(6)+t)
      ct=-u(5)/(u(6)+t)**2
      ctt=2.*u(5)/(u(6)+t)**3
      d1k=u(1)*dexp(u(2)*t+u(3)*t**2)
      d1kt=(u(2) + 2.*u(3)*t)*d1k
      d1ktt=(2.*u(3)+(u(2)+2.*u(3)*t)**2)*d1k
      d=d1k+c*dlog((b+p)/(b+1000.))
      dt=d1kt+c*(1./(b+p)-1./(b+1000.))*bt + ct*dlog((b+p)/(b+1000.))
      dtt=d1ktt-c*(1./(b+p)**2-1./(b+1000.)**2)*bt**2+(1./(b+p)-1./
     1(b+1000.))*(2.*bt*ct + c*btt)+ctt*dlog((b+p)/(b+1000.))
      dp=c/(b+p)
      dpp=-c/(b+p)**2
      dtp=-c*bt/(b+p)**2 + ct/(b+p)
      return
      end
c
c
c
      subroutine rho(t,p)
c
      implicit real*8 (a-h,o-z)
      common /alpha/ a(9,7),al,bet,gam,b0,c,d,e,bb0,b1,b2,b4
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      common/delta/v,vt,vp,vtt,vpp,vtp
      data al,bet,gam /11.0,44.3333333,3.5/
      data b0,c,d,e /.747862916,-.354078223,.007159876,-.003528426/
      data bb0,b1,b2,b4 /1.127833441,-.594400123,-5.010995915,
     1    .636842555/
c
c     In this routine r = 4.61518 bar-cu.cm/k-mol
c
      data r /4.61518/
      data a /0.0,0.0,68833.257944332,0.0,0.0,0.0,0.0,0.0,0.0,
     1   -5306.2968529023,178638.32875422,-2563743.6613260,
     1      11797433.655832,-31377774.947767,46561826.115608,
     1      -31555231.392127,0.0,6962522.0862664,
     2   22744.901424408,-395147.31563338,4821257.5981415,
     2      -21734810.110373,52911910.757704,-72752773.275387,
     2      47929666.384584,0.0,-10834900.096447,
     3   -26627.944829770,0.0,217572.45522644,0.0,
     3      -707304.18082074,0.0,0.0,0.0,0.0,
     4   7877.9333020687,338038.84280753,-3418301.6969660,
     4      10829952.168620,-13802577.177877,4177424.6148294,
     4      4091266.4781209,0.0,-2272282.7401688,
     5   0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
     6   -698.30527374994,-138550.50202703,1222315.6417448,
     6      -2544199.8064049,-2510991.4369001,14016358.244614,
     6      -13626369.388386,0.0,3836548.6000660/
      save r
c
c
c ********************************************************************
c *                                                                  *
c *   RHO calculates the density of water at the specified temperature *
c *   (t, in kelvins) and pressure (p, in 10**5 pa), using the       *
```

```
c *   equations of Haar, Gallagher and Kell (Ninth Internaltional      *
c *   Conference on the Properties of Steam, Munich, 1979).  These     *
c *   equations are valid for 273.15 K .le. t .le. 623.15 K and        *
c *   saturation pressure .le. p .le. 100 mpa.  The volume (which is   *
c *   simply the reciprocal of the density) and its temperature and    *
c *   pressure derivatives are also computed.  The density has units   *
c *   of grams per cubic centimeter.                                   *
c *                                                                    *
c *                                    J. Christopher Peiper           *
c *                                    29 march, 1982                  *
c *                                                                    *
c ********************************************************************
c
c
      tau=t/647.073
      call psat(t)
      if (p.le.0) p=ps
      b=b0+c*dlog(tau)+d/tau**3+e/tau**5
      bt=(c-3e0*d/tau**3-5e0*e/tau**5)/t
      btt=-(c-12e0*d/tau**3-30e0*e/tau**5)/t**2
      bb=bb0+b1/tau+b2/tau**2+b4/tau**4
      bbt=-(b1/tau+2e0*b2/tau**2+4e0*b4/tau**4)/t
      bbtt=(2e0*b1/tau+6e0*b2/tau**2+20e0*b4/tau**4)/t**2
      rh=1.0
      if (p.lt.ps) rh=.001
c
      do 30 i=1,100
      y=rh*b/4e0
      er=dexp(-rh)
      pr = 0e0
      dpr = 0e0
c
      do 15 n=1,9
      s=a(n,1)
c
      do 10 j=2,7
   10 s=s+a(n,j)/tau**(j-1)
c
      pr=pr+s*(1e0-er)**(n-1)
   15 dpr=dpr+s*(1e0-er)**(n-1)*(2e0+rh*(n*er-1e0)/(1e0-er))
c
      pr=rh**2*er*pr+rh*r*t*((1e0+al*y+bet*y**2)/(1e0-y)**3+4e0*y*
     1 (bb/b-gam))
      dpr=rh*er*dpr + r*t*((1e0+2e0*al*y+3e0*bet*y**2)/(1e0-y)**3
     1    +3e0*y*(1e0+al*y+bet*y**2)/(1e0-y)**4 + 8e0*y*(bb/b-gam))
      rhn=rh+(p-pr)/dpr
      if (abs(rhn-rh).lt.abs(rhn/1.e6).and.abs(p-pr).lt.abs(p/1.e5))
     1   go to 35
   30 rh=rhn
c
      print 100, t,p,rhn
c
   35 rh=rhn
      y=rh*b/4e0
      yt=rh*bt/4e0
      ytt=rh*btt/4e0
      yr=b/4e0
      er=dexp(-rh)
      pr = 0e0
      dpr = 0e0
```

```
      dpt = 0e0
      dprr = 0e0
      dptt = 0e0
      dprt = 0e0
c
      do 50 n=1,9
      an = n
      s=a(n,1)
      s1 = 0e0
      s2 = 0e0
c
      do 40 j=2,7
      aj = j
      s=s+a(n,j)/tau**(j-1)
      s1=s1+(1e0-aj)*a(n,j)/tau**(j-1)
   40 s2=s2+aj*(aj-1e0)*a(n,j)/tau**(j-1)
c
      pr=pr+s*(1e0-er)**(n-1)
      dpr=dpr+s*(2e0+rh*(an*er-1e0)/(1e0-er))*(1e0-er)**(n-1)
      dpt=dpt+s1*(1e0-er)**(n-1)
      dprr=dprr+s*((1e0+rh*(an*er-1e0)/(1e0-er))*
     1 (2e0+rh*(an*er-1e0)/(1e0-er))
     2       *(1e0-er)**(n-1)+rh*(1e0-er)**(n-2)
     3    *(an*er*(1e0-rh)-1e0-rh*er*(an*er-1e0)/(1e0-er)))
      dptt=dptt+s2*(1e0-er)**(n-1)
   50 dprt=dprt+s1*(2e0+rh*(an*er-1e0)/(1e0-er))*(1e0-er)**(n-1)
c
      pr=rh**2*er*pr+rh*r*t*((1e0+al*y+bet*y**2)/(1e0-y)**3
     1    +rh*(bb-gam*b))
      dpr=rh*er*dpr+r*t*((1e0+2e0*al*y+3e0*bet*y**2)/(1e0-y)**3
     1    +3e0*y*(1e0+al*y+bet*y**2)/(1e0-y)**4
     2    +2e0*rh*(bb-gam*b))
      dpt=rh**2*er*dpt/t+rh*r*((1e0+al*y+bet*y**2)/(1e0-y)**3
     1       +rh*(bb-gam*b))
     2    +rh*r*t*((al+2e0*bet*y)/(1e0-y)**3
     3       +3e0*(1e0+al*y+bet*y**2)/(1e0-y)**4)*yt
     4    +rh**2*r*t*(bbt-gam*bt)
      dptt=rh**2*er*dptt/t**2
     1    +2e0*rh*r*((al+2e0*bet*y)/(1e0-y)**3
     2       +3e0*(1e0+al*y+bet*y**2)/(1e0-y)**4)*yt
     3    +2e0*rh**2*r*(bbt-gam*bt)
     4    +rh*r*t*(2e0*bet/(1e0-y)**3+6e0*(al+2e0*bet*y)/(1e0-y)**4
     5     +12e0*(1e0+al*y+bet*y**2)/(1e0-y)**5)*yt**2
     6    +rh*r*t*((al+2e0*bet*y)/(1e0-y)**3
     7       +3e0*(1e0+al*y+bet*y**2)/(1e0-y)**4)*ytt
     8    +rh**2*r*t*(bbtt-gam*btt)
      dprr=er*dprr + 2e0*r*t*((al+3e0*bet*y)/(1e0-y)**3
     1       +3e0*(1e0+2e0*al*y+3e0*bet*y**2)/(1e0-y)**4
     2       +6e0*y*(1e0+al*y+bet*y**2)/(1e0-y)**5)*yr
     3    +2e0*r*t*(bb-gam*b)
      dprt=rh*er*dprt/t + 2e0*rh*r*t*(bbt-gam*bt)
     1    +r*((1e0+2e0*al*y+3e0*bet*y**2)/(1e0-y)**3
     2     +3e0*y*(1e0+al*y+bet*y**2)/(1e0-y)**4 + 2e0*rh*(bb-gam*b))
     3    +2e0*r*t*((al+3e0*bet*y)/(1e0-y)**3
     4       +3e0*(1e0+2e0*al*y+3e0*bet*y**2)/(1e0-y)**4
     5       +6e0*y*(1e0+al*y+bet*y**2)/(1e0-y)**5)*yt
c
      v=1e0/rh
      vt=dpt/(rh**2*dpr)
      vp=-1e0/(rh**2*dpr)
```

```
      vtt=2e0*(dpt/dpr)**2/rh**3 + dptt/(dpr*rh**2)
     1   - 2.*dprt*dpt/(rh*dpr)**2
     3   +dprr*dpt**2/(rh**2*dpr**3)
      vpp=(2e0*dpr+rh*dprr)/(rh*dpr)**3
      vtp=-2e0*dpt/(rh**3*dpr**2) - dprr*dpt/(rh**2*dpr**3)
     1   + dprt/(rh*dpr)**2
      return
c
  100 format ('0',5x,'at',f7.2,'k and',f10.4,'bars convergence of ',
     1   'the density was not achieved --',/10x,'the density is',f8.6)
      end
c
c
c
      subroutine psat(t)
c
      implicit real*8(a-h,o-z)
      common /beta/ ps,dpdt,er,b,bt,btt,bb,bbt,bbtt,y,yr,yt,ytt
      dimension a(8)
      data a/-7.8889166e0,2.5514255e0,-6.716169e0
     1,33.239495e0,-105.38479e0,174.35319e0,-148.39348e0
     2,48.631602e0/
      save a
c
c     This subroutine calculates the saturation pressure
c     of pure water and its temperature derivative.
c
      if(t.gt.314.e0) go to 2
      pl=6.3573118e0-8858.843e0/t+607.56335/t**.6
      ps=dexp(pl)
      dpdt=dexp(pl)*(8858.843e0/t**2 - 0.6*607.56335/t**1.6)
       return
2     v=t/647.25
      w=1.-v
      c=0.
      r=0.
      do 4 i=1,8
      z=i
      y=a(i)*w**((z+1.)/2.)
      r=r+(.5*z+.5)*a(i)*w**((z-1.)/2.)
   4  c=c+y
      q=c/v
         dpdt=220.93*dexp(q)*(-r/v/647.25-c/v/v/647.25)
      ps=220.93*dexp(q)
      return
      end
```

NACL calculates the properties of NaCl solutions using
the accurate empirical model published by Pitzer, Peiper, and
Busey, J. Phys. Chem. Ref. Data v. 13, pp. 1-102 (1984).
The valid range is 0-300 C, 0-6 molal, Psat(T)-1000 bar.

Temperature, pressure and molality are

t,tc = 298.15 25.00 p = 1.000 m = 1.000

Table 1 contains all calculated quantities.
See comments in HAAR, RHO, NACL1 and NACL3 for further explanation.

Properties of pure water are calculated first.

Saturation pressure of pure water and temperature derivaive from PSAT

ps,dpdt = 3.1687E-02 1.8887E-03

Reference state properties for liquid water from HAAR

f10,g10 = -2.2170E+01 -2.2169E+01 u10,h10,s10 = -1.3757E+01 -1.3756E+01 8.4130E+00

cv10,cp10,cs10 = 8.9681E+00 9.0645E+00 9.0645E+00 v10,a10,b10 = 1.0029E+00 2.5942E-04 4.5218E-05

Properties of water as ideal gas from HAAR

f0,g0 = -1.9703E+01 -1.8703E+01 u0,h0,s0 = 2.9953E+00 3.9953E+00 2.2698E+01 cv0,cp0 = 3.0396E+00 4.0396E+00

Specific volume of pure water and its derivatives over T and P from RHO

v,vt,vp = 1.0029E+00 2.6019E-04 -4.5352E-05 vtt,vpp,vtp = 9.6606E-06 1.4528E-08 9.5806E-08

Dielectric constant and its derivatives from DIELEC

d,dt,dtt = 7.8384E+01 -3.5930E-01 1.5526E-03 dp,dpp,dtp = 3.7056E-03 -5.4966E-07 -9.9103E-06

Debye-Huckel coefficients from DEBYE

ap,ah,as = 3.915E-01 8.006E-01 2.366E+00 aj,av = 3.9449E+00 1.8749E+00 aa,ab = 1.6841E-02 -3.8752E-04

Virial coefficients from NACL1

b0,b0l = 7.5374E-02 7.3770E-04 b0s,b0j = 2.9532E-01 -1.3344E-05 b0v,b0a,b0b = 1.2343E-05 -3.1844E-07 -5.4612E-09

b1,b1l = 2.7703E-01 9.0102E-04 b1s,b1j = 5.4567E-01 -8.4653E-06 b1v,b1a,b1b = 0.0000E+00 0.0000E+00 0.0000E+00

cp,cpl = 1.4074E-03 -1.1004E-04 cps,cpj = -3.1401E-02 1.4694E-06 cpv,cpa,cpb = -1.2936E-06 4.4292E-08 0.0000E+00

Reference state properties for NaCl from NACL1

g20,s20 = -1.3886E+01 1.3886E+01 h20,cp20 = 3.4132E-06 -1.0077E+01 v20,vt20,vp20 = 1.6679E+01 7.7179E-02 4.3351E-03

Excess properties from EXCESS

gex,g1ex,g2ex = -7.1210E-01 1.2732E-01 -8.3942E-01 hex,h1ex,h2ex = -4.0604E-02 4.7746E-03 -3.0563E-01

sex,s1ex,s2ex = 6.7149E-01 -1.2254E-01 5.3379E-01 cpex,cp1ex,cp2ex = 5.2808E+00 -3.2115E+00 8.4923E+00

vex,v1ex,v2ex = 1.8117E+00 -1.2170E-01 -5.3030E-01 vtex,vt1ex,vt2ex = -3.6250E-03 1.7420E-02 -4.3175E-02

vpex,vp1ex,vp2ex = -5.2538E-04 1.8269E-04 -1.9883E-04

Partial molal quantities from NACL2

g1,g2 = -2.2078E+01 -1.4725E+01 s1,s2 = 8.3265E+00 1.4420E+01

h1,h2 = -1.3751E+01 -3.0563E-01 cp1,cp2 = 5.8531E+00 -1.5849E+00

v1,v2 = 1.7947E+01 1.6149E+01 vt1,vt2 = 2.2107E-02 3.4004E-02 vp1,vp2 = -6.3434E-04 4.1362E-03

Osmotic coefficient and mean ion activity coefficient frm NACL2.

phi,gam = 9.3634E-01 6.5724E-01

Properties of solid NaCl from NACL3

g2s,h2s = -8.6782E+00 6.6410E-04 s2s,cp2s = 8.6789E+00 6.1104E+00

v2s,vt2s,vp2s = 2.6993E+01 3.2347E-03 0.0000E+00

Change of free energy and other thermodynamic functions ccompanying dissolution of NaCl in water from NACL3

dgs0,dss0 = -3.6415E+00 5.2071E+00 dhs0,dcps0 = 1.5656E+00 -1.6188E+01

dvs0,dvts0,dvps0 = -1.0314E+01 7.3944E-02 4.3351E-03

Table 2. The following quantities (class 1) are tabulated in report LBL-15512 and are presented here in the same order.

t,tc = 298.15 25.00 p = 1.000 m = 1.000

table 1	g10,g20,ap,b0,b1,1e3*cp/2	-22.1689	-13.8860	0.3915	0.0754	0.2770	0.70
table 2	s10,s20,as,b0s,b1s,1e3*cps/2	8.4130	13.8860	2.3664	0.2953	0.5457	-15.70
table 3	h10,h20,ah,1e3*b0l,1e3*b1l,1e3*cpl/2	-13.7560	0.000	0.801	0.738	0.901	-0.055
table 4	cp10,cp20,aj,1e6*b0j,1e6*b1j,1e6*cpj/2	9.065	-10.077	3.945	-13.344	-8.465	0.735
table 5	mw*v10,v20,av,1e6*b0v,1e6*cpv/2	18.068	16.679	1.875	12.343	-0.647	
table 6	1e3*mw*v10*a10,vt20,aa,1e6*b0a,1e9*cpa/2	4.687	0.077	0.017	-0.318	22.146	
table 7	-1e3*mw*v10*b10,1e3*vp20,1e3*ab,1e9*b0b	-0.817	4.335	-0.388	-5.461		
table 8	gam	0.657					
table 9	phi	0.936					
table 10	dss0	5.207					
table 11	dhs0	1.566					
table 12	sex/m	0.671					
table 13	hex/m	-0.041					
table 14	cpex/m	5.281					
table 15	density	1.03623					
table 16	specific entropy(j/g-k)	3.798					
table 17	specific enthalpy(j/g)	-1788.4					

NACL calculates the properties of NaCl solutions using
the accurate empirical model published by Pitzer, Peiper, and
Busey, J. Phys. Chem. Ref. Data v. 13, pp. 1-102 (1984).
The valid range is 0-300 C, 0-6 molal, Psat(T)-1000 bar.

Table 2. The following quantities (class 1) are tabulated in report LBL-15512
and are presented here in the same order.

t,tc = 573.15 300.00 p = 85.832 m = 6.000

table 1	g10,g20,ap,b0,b1,1e3*cp/2	-17.1382	-11.3597	0.9595	0.0768	0.5192	-1.95
table 2	s10,s20,as,b0s,b1s,1e3*cps/2	14.6673	-5.1276	17.8896	0.0577	1.1376	2.83
table 3	h10,h20,ah,1e3*b0l,1e3*b1l,1e3*cpl/2	-2.4710	-16.487	14.052	-0.033	1.079	0.008
table 4	cp10,cp20,aj,1e6*b0j,1e6*b1j,1e6*cpj/2	12.449	-197.332	183.418	8.273	4.828	0.072
table 5	mw*v10,v20,av,1e6*b0v,1e6*cpv/2	25.288	-95.681	98.730	-51.668	1.583	
table 6	1e3*mw*v10*a10,vt20,aa,1e6*b0a,1e9*cpa/2	82.760	-2.882	2.715	-1.565	31.847	
table 7	-1e3*mw*v10*b10,1e3*vp20,1e3*ab,1e9*b0b	-8.120	261.187	-217.969	91.188		
table 8	gam	0.164					
table 9	phi	0.747					
table 10	dss0	-17.961					
table 11	dhs0	-18.788					
table 12	sex/m	18.455					
table 13	hex/m	15.348					
table 14	cpex/m	173.723					
table 15	density	0.99187					
table 16	specific entropy(j/g-k)	5.446					
table 17	specific enthalpy(j/g)	-508.0					

LAWRENCE BERKELEY LABORATORY
TECHNICAL INFORMATION DEPARTMENT
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720