Supplementary Material:
Derivative of the peak shape function with respect to S

The peak shape function $F\left(2\theta,\mu,\sigma,{}^3\kappa,{}^4\kappa\right)$, is defined by

$$F\left(2\theta,\mu,\sigma,{}^3\kappa,{}^4\kappa\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{\left(2\theta-\mu\right)^2}{2\sigma^2}\right]\left[1+\frac{{}^3\kappa}{6\sigma^3}H_3\left(\frac{2\theta-\mu}{\sigma}\right)+\frac{{}^4\kappa}{24\sigma^4}H_4\left(\frac{2\theta-\mu}{\sigma}\right)\right],$$

where

$$\begin{aligned}\mu &= 2\theta_{\mathrm{B}}-2\theta_0+\delta_1,\\ \sigma &= \sqrt{U\left(\tan\theta_{\mathrm{B}}-\tan\theta_m\right)^2+V\left(\tan\theta_{\mathrm{B}}-\tan\theta_m\right)+W+\delta_2-\delta_1^2},\\ {}^3\kappa &= \delta_3-3\delta_1\delta_2+2\delta_1^3,\\ {}^4\kappa &= \delta_4-4\delta_1\delta_3-3\delta_2^2+12\delta_2\delta_1^2-6\delta_1^4.\end{aligned}$$

Here H_3 and H_4 are the third and fourth degree Hermite polynomials,

$$\begin{aligned}H_3\left(x\right) &= x^3-3x,\\ H_4\left(x\right) &= x^4-6x^2+3.\end{aligned}$$

θ_m is an arbitrary offset that should be chosen close to the minmum of the resolution curve to minimize the correlations among U, V, and W.

Although the Finger, Cox & Jephcoat model for the peak shape function, and the Edgeworth series approximation to it, appears to have no adjustable parameters, it implicitly assumes that the incident beam has no axial divergence. This can be approximately accounted for by an 'effective' sample height, $2S$. To make this a refinable parameter we need to evaluate the partial derivative of F with respect to S. First we need the partial derivatives of F with respect to its arguments, μ, σ, ${}^3\kappa$, and ${}^4\kappa$. For convenience, let $t=\left(2\theta-\mu\right)/\sigma$; $\frac{\partial t}{\partial\mu}=-1/\sigma$; $\frac{\partial t}{\partial\sigma}=\left(\mu-2\theta\right)/\sigma^2$. Then

$$\begin{aligned}\frac{\partial F}{\partial t} &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{t^2}{2}\right)\left[-t\left(1+\frac{{}^3\kappa}{6\sigma^3}H_3\left(t\right)+\frac{{}^4\kappa}{24\sigma^4}H_4\left(t\right)\right)\right.\\ &\left.+\frac{{}^3\kappa}{6\sigma^3}H_3'\left(t\right)+\frac{{}^4\kappa}{24\sigma^4}H_4'\left(t\right)\right],\end{aligned}$$

where

$$\begin{aligned}H_3'\left(t\right) &= \frac{\mathrm{d}H_3\left(t\right)}{\mathrm{d}t}=3t^2-3,\\ H_4'\left(t\right) &= \frac{\mathrm{d}H_4\left(t\right)}{\mathrm{d}t}=4t^3-12t.\end{aligned}$$

The partial derivatives of F with respect to its arguments are then

$$\begin{aligned}
\frac{\partial F}{\partial \mu} &= \frac{\partial F}{\partial t}\frac{\partial t}{\partial \mu}; \\
\frac{\partial F}{\partial \sigma} &= \frac{\partial F}{\partial t}\frac{\partial t}{\partial \sigma} - \frac{F}{\sigma} - \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{t^2}{2}\right)\left[\frac{{}^3\kappa}{2\sigma^4}H_3\left(t\right) + \frac{{}^4\kappa}{6\sigma^5}H_4\left(t\right)\right]; \\
\frac{\partial F}{\partial\,{}^3\kappa} &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{t^2}{2}\right)\left[\frac{1}{6\sigma^3}H_3\left(t\right)\right]; \\
\frac{\partial F}{\partial\,{}^4\kappa} &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{t^2}{2}\right)\left[\frac{1}{24\sigma^4}H_4\left(t\right)\right].
\end{aligned}$$

The arguments of F, μ, σ, ${}^3\kappa$, and ${}^4\kappa$, are themselves functions of the moments, δ_1, δ_2, δ_3, and δ_4, of the function that modifies the peak shape due to the effects of the Debye-Scherrer ring. These partial derivatives are

$$\begin{aligned}
\frac{\partial \mu}{\partial \delta_1} &= 1; \\
\frac{\partial \sigma}{\partial \delta_1} &= \frac{\delta_1}{\sigma}; \\
\frac{\partial \sigma}{\partial \delta_2} &= -\frac{1}{2\sigma}; \\
\frac{\partial\,{}^3\kappa}{\partial \delta_1} &= -3\delta_2 + 6\delta_1^2; \\
\frac{\partial\,{}^3\kappa}{\partial \delta_2} &= -3\delta_1; \\
\frac{\partial\,{}^3\kappa}{\partial \delta_3} &= 1; \\
\frac{\partial\,{}^4\kappa}{\partial \delta_1} &= -4\delta_3 + 24\delta_1\delta_2 - 24\delta_1^3; \\
\frac{\partial\,{}^4\kappa}{\partial \delta_2} &= -6\delta_2 + 12\delta_1^2; \\
\frac{\partial\,{}^4\kappa}{\partial \delta_3} &= -4\delta_1; \\
\frac{\partial\,{}^4\kappa}{\partial \delta_4} &= 1.
\end{aligned}$$

Finally

$$
\begin{aligned}
\frac{\partial F}{\partial \delta_1} &= \frac{\partial F}{\partial \mu} + \frac{\partial F}{\partial \sigma}\frac{\partial \sigma}{\partial \delta_1} + \frac{\partial F}{\partial\,{}^3\kappa}\frac{\partial\,{}^3\kappa}{\partial \delta_1} + \frac{\partial F}{\partial\,{}^4\kappa}\frac{\partial\,{}^4\kappa}{\partial \delta_1};\\
\frac{\partial F}{\partial \delta_2} &= \frac{\partial F}{\partial \sigma}\frac{\partial \sigma}{\partial \delta_2} + \frac{\partial F}{\partial\,{}^3\kappa}\frac{\partial\,{}^3\kappa}{\partial \delta_2} + \frac{\partial F}{\partial\,{}^4\kappa}\frac{\partial\,{}^4\kappa}{\partial \delta_2};\\
\frac{\partial F}{\partial \delta_3} &= \frac{\partial F}{\partial\,{}^3\kappa} + \frac{\partial F}{\partial\,{}^4\kappa}\frac{\partial\,{}^4\kappa}{\partial \delta_3};\\
\frac{\partial F}{\partial \delta_4} &= \frac{\partial F}{\partial\,{}^4\kappa}.
\end{aligned}
$$

The partial derivatives of F with respect to the four moments, δ_1 to δ_4, are all functions of 2θ, and therefore they must be evaluated for every data point. The derivatives of the moments with respect to S, however, depend only on S and $2\theta_{\mathrm{B}}$, and they need to be evaluated only once per reflection. GSAS, in its implementation of the Finger-Cox-Jephcoat correction, uses the quantities $s = S/R$ and $h = H/R$, where R is the distance from the sample to the detector slit. We shall derive the derivatives of the moments with respect to s. First, let $\alpha = (h-s)^2$, and $\beta = (h+s)^2$. Then $\frac{\mathrm{d}\alpha}{\mathrm{d}s} = -2\,(h-s)$, and $\frac{\mathrm{d}\beta}{\mathrm{d}s} = 2\,(h+s)$. In terms of α and β,

$$
\begin{aligned}
\delta_1 &= -\frac{\beta+\alpha}{12\tan 2\theta_{\mathrm{B}}};\\
\frac{\partial \delta_1}{\partial \alpha} &= \frac{\partial \delta_1}{\partial \beta} = -\frac{1}{12\tan 2\theta_{\mathrm{B}}};\\
\frac{\mathrm{d}\delta_1}{\mathrm{d}s} &= \frac{\partial \delta_1}{\partial \alpha}\frac{\mathrm{d}\alpha}{\mathrm{d}s} + \frac{\partial \delta_1}{\partial \beta}\frac{\mathrm{d}\beta}{\mathrm{d}s}.
\end{aligned}
$$

$$
\begin{aligned}
\delta_2 &= \frac{\beta^2+\beta\alpha+\alpha^2}{60\tan^2 2\theta_{\mathrm{B}}};\\
\frac{\partial \delta_2}{\partial \alpha} &= \frac{\beta+2\alpha}{60\tan^2 2\theta_{\mathrm{B}}};\\
\frac{\partial \delta_2}{\partial \beta} &= \frac{2\beta+\alpha}{60\tan^2 2\theta_{\mathrm{B}}};\\
\frac{\mathrm{d}\delta_2}{\mathrm{d}s} &= \frac{\partial \delta_2}{\partial \alpha}\frac{\mathrm{d}\alpha}{\mathrm{d}s} + \frac{\partial \delta_2}{\partial \beta}\frac{\mathrm{d}\beta}{\mathrm{d}s}.
\end{aligned}
$$

$$
\begin{aligned}
\delta_3 &= -\frac{\beta^3+\beta^2\alpha+\beta\alpha^2+\alpha^3}{224\tan^3 2\theta_{\mathrm{B}}};\\
\frac{\partial \delta_3}{\partial \alpha} &= -\frac{\beta^2+2\beta\alpha+3\alpha^2}{224\tan^3 2\theta_{\mathrm{B}}};\\
\frac{\partial \delta_3}{\partial \beta} &= -\frac{3\beta^2+2\beta\alpha+\alpha^2}{224\tan^3 2\theta_{\mathrm{B}}};\\
\frac{\mathrm{d}\delta_3}{\mathrm{d}s} &= -\frac{\partial \delta_3}{\partial \alpha}\frac{\mathrm{d}\alpha}{\mathrm{d}s} + \frac{\partial \delta_3}{\partial \beta}\frac{\mathrm{d}\beta}{\mathrm{d}s}.
\end{aligned}
$$

$$\begin{aligned}
\delta_4 &= \frac{\beta^4 + \beta^3\alpha + \beta^2\alpha^2 + \beta\alpha^3 + \alpha^4}{720\tan^4 2\theta_\mathrm{B}};\\
\frac{\partial\delta_4}{\partial\alpha} &= \frac{\beta^3 + 2\beta^2\alpha + 3\beta\alpha^2 + 4\alpha^3}{720\tan^4 2\theta_\mathrm{B}};\\
\frac{\partial\delta_4}{\partial\beta} &= \frac{4\beta^3 + 3\beta^2\alpha + 2\beta\alpha^2 + \alpha^3}{720\tan^4 2\theta_\mathrm{B}};\\
\frac{\mathrm{d}\delta_4}{\mathrm{d}s} &= \frac{\partial\delta_4}{\partial\alpha}\frac{\mathrm{d}\alpha}{\mathrm{d}s} + \frac{\partial\delta_4}{\partial\beta}\frac{\mathrm{d}\beta}{\mathrm{d}s}.
\end{aligned}$$

Then

$$\frac{\partial F}{\partial s} = \frac{\partial F}{\partial\delta_1}\frac{\mathrm{d}\delta_1}{\mathrm{d}s} + \frac{\partial F}{\partial\delta_2}\frac{\mathrm{d}\delta_2}{\mathrm{d}s} + \frac{\partial F}{\partial\delta_3}\frac{\mathrm{d}\delta_3}{\mathrm{d}s} + \frac{\partial F}{\partial\delta_4}\frac{\mathrm{d}\delta_4}{\mathrm{d}s}.$$

Also

$$\begin{aligned}
\frac{\partial F}{\partial U} &= \frac{\partial F}{\partial\sigma}\frac{(\tan\theta - \tan\theta_m)^2}{2\sigma};\\
\frac{\partial F}{\partial V} &= \frac{\partial F}{\partial\sigma}\frac{\tan\theta - \tan\theta_m}{2\sigma};\\
\frac{\partial F}{\partial W} &= \frac{\partial F}{\partial\sigma}\frac{1}{2\sigma}.
\end{aligned}$$

```
C example Fortran code to accompany Prince & Toby, A Comparison of Methods for
C Modeling the Effect of Axial Divergence in Powder Diffraction
C

      PROGRAM PEAKS
      CHARACTER*20 STRIN

      DIMENSION VALUES(81,4)

      NVALUES = 81
      WRITE(*,*) 'Enter FWHM: '
      READ(*,*) FWHM
      WRITE(*,*) 'Enter Peak position: '
      READ(*,*) TWOTH
      XMIN=TWOTH-6.*FWHM
      XMAX=TWOTH+2.*FWHM
      WRITE(*,*) 'Enter H/R: '
      READ(*,*) HOVERR
      WRITE(*,*) 'Enter S/R: '
      READ(*,*) SOVERR
      NCURVE = 1
      FACTR = 0.
      CALL CALCPK(FWHM,TWOTH,0.0,0.0
     1                  ,FACTR,1,VALUES(1,NCURVE),NVALUES)
      NCURVE = 2
      CALL CALCPK(FWHM,TWOTH,HOVERR,SOVERR
     1                  ,FACTR,1,VALUES(1,NCURVE),NVALUES)
      NCURVE = 3
      CALL CALCPK(FWHM,TWOTH,HOVERR,SOVERR
     1                  ,FACTR,2,VALUES(1,NCURVE),NVALUES)
      OPEN(UNIT=8,FORM='FORMATTED'
     1              ,STATUS='UNKNOWN')
      TWOTHETA=XMIN
      WRITE(8,*) 'FWHM=',FWHM,', TWOTHETA=',TWOTH,', S/L=',SOVERR,
     $     ', H/L=',HOVERR
      WRITE(8,'(4A10)') 'TWOTHETA','No Asym','Edgeworth','FCJ'

      RINCR=(XMAX-XMIN)/(FLOAT(NVALUES)-1.)
      DO I=1,NVALUES
         WRITE(8,9000)TWOTHETA,(VALUES(I,J),J=1,NCURVE)
 9000    FORMAT(5F10.4)
         TWOTHETA=TWOTHETA+RINCR
      ENDDO
      CLOSE(UNIT=8)
      STOP
      END PROGRAM PEAKS

      SUBROUTINE CALCPK(FWHM,TWOTH,HOVERR,SOVERR,FACTR,ISET,VALUES,N)
C     CALCULATE AN ASYMMETRIC PEAK.
C     ARGUMENTS ARE:
C     FWHM   = FULL WIDTH AT HALF MAXIMUM OF UNDISTORTED PEAK.
C     TWOTH  = 2 THETA, AS CALCULATED FROM CELL CONSTANTS.
C     HOVERR = HALF HEIGHT OF RECEIVING SLIT DIVIDED BY DISTANCE FROM SAMPLE.
C     VALUES = ARRAY TO STORE CURVE.
C     N      = NUMBER OF POINTS FROM TWOTH-6*FWHM TO TWOTH+2*FWHM.
      DIMENSION VALUES(N),COEF(8),DPDF(6)
      PIO180=ACOS(-1.)/180.
```

```
      SIGMA=FWHM*42.4661
      COEF(1)=0.
      COEF(2)=0.
      COEF(3)=SIGMA**2
      COEF(4)=0.
      COEF(5)=SOVERR
      COEF(6)=HOVERR
      XLEN=8.*FWHM
      XMIN=TWOTH-6.*FWHM
      X=XMIN
      ETA=0.
      DO I=1,N
         IF(ISET.EQ.1)THEN
            CALL SIMGAUS(COEF,100.*TWOTH,100.*(X-TWOTH),TMP,DPDT
     1                    ,DPDF,0.)
            VALUES(I)=100.*TMP
         ELSE
            VALUES(I)=PROFVAL(ETA,FWHM,HOVERR,SOVERR,X
     1                 ,TWOTH,DUM1,DUM2,DUM3,DUM4,DUM5,.TRUE.)
         ENDIF
         X=XMIN+XLEN*FLOAT(I)/FLOAT(N-1)
      ENDDO
      RETURN
      END SUBROUTINE CALCPK
      SUBROUTINE SIMGAUS(COFF,TTHETA,DTTH,PRFUNC,DPRDT,DPRDCF,ZERO)

!PURPOSE: Compute function & derivatives for Edgeworth series model
! of axial divergence effects. Written by E. Prince, December, 2004.
! All calcs in centidegrees

!CALLING ARGUMENTS:

      REAL*4         COFF(1)              !Coefficients
      REAL*4         TTHETA               !2-theta in centideg
      REAL*4         DTTH                 !Delta 2-theta
      REAL*4         PRFUNC               !Value of function at DTTH
      REAL*4         DPRDT                !partial df(t)/dt
      REAL*4         DPRDCF(1)            !partial df(t)/dc
      REAL*4         ZERO                 !zero corrextion

!LOCAL VARIABLES:

      REAL*4         TH
      REAL*4         ARGS(4)              !Asymmetry coefficients
      REAL*4         DERVS(4)             !Derivatives wrt S/L
      REAL*4         SIG                  !Gaussian width
      REAL*4         CUM3                 !Third cumulant
      REAL*4         CUM4                 !Fourth cumulant
      REAL*4         DFDT
      REAL*4         DTDMU
      REAL*4         DTDSIG
      REAL*4         DFDMU
      REAL*4         DFDSIG
      REAL*4         DFDCM3
      REAL*4         DFDCM4
      REAL*4         DELTA1
      REAL*4         DMUDS
```

```
      REAL*4        DSIGDS
      REAL*4        DCM3DS
      REAL*4        DCM4DS
      EQUIVALENCE (DELTA1,ARGS(1))
      EQUIVALENCE (SIG,ARGS(2))
      EQUIVALENCE (CUM3,ARGS(3))
      EQUIVALENCE (CUM4,ARGS(4))
      EQUIVALENCE (DMUDS,DERVS(1))
      EQUIVALENCE (DSIGDS,DERVS(2))
      EQUIVALENCE (DCM3DS,DERVS(3))
      EQUIVALENCE (DCM4DS,DERVS(4))

!SUBROUTINES CALLED:

!     ASYMCOF

!DATA STATEMENTS:

      DATA PIO180/0.000174533/

!CODE:

      TH = 0.5*PIO180*(TTHETA-ZERO)
      TTH = TAN(TH)-COFF(4)
      SIG = COFF(1)*TTH*TTH+COFF(2)*TTH+COFF(3)


      CALL ASYMCOF(TTHETA-ZERO,COFF(6),COFF(5),ARGS,DERVS)
      IF ( SIG.LE.0.01) SIG = 0.01
      SIG3=SIG*SIG*SIG
      SIG4=SIG3*SIG
      DSIGDW=0.5/SIG
      TT = (DTTH-DELTA1)/SIG
      DTDMU=-1./SIG
      DTDSIG=-TT/SIG
      EXPTT=EXP(-0.5*TT**2)
      H3TT=TT**3-3.*TT
      DH3TT=3.*TT**2-3.
      H4TT=TT**4-6.*TT**2+3.
      DH4TT=4.*TT**3-12.*TT
      SCLNRM=1./(2.50663*SIG)
      PRFUNC=SCLNRM*EXPTT*(1.+ CUM3*H3TT/(6.*SIG3)
     1       + CUM4*H4TT/(24.*SIG4))
      DFDT=SCLNRM*EXPTT*(-TT*(1.+ CUM3*H3TT/(6.*SIG3)
     1     + CUM4*H4TT/(24.*SIG4))+ CUM3*DH3TT/(6.*SIG3)
     2     + CUM4*DH4TT/(24.*SIG4))
      DFDMU=DFDT*DTDMU
      DFDSIG=DFDT*DTDSIG-PRFUNC/SIG-SCLNRM*EXPTT
     1       *((CUM3/SIG3)*H3TT/(2.*SIG)+ (CUM4/SIG4)*H4TT/(6.*SIG))
      DFDW=DFDSIG*DSIGDW
      DFDCM3=SCLNRM*EXPTT*H3TT/(6.*SIG3)
      DFDCM4=SCLNRM*EXPTT*H4TT/(24.*SIG4)


      DPRDCF(1) = DFDW*TTH*TTH
      DPRDCF(2) = DFDW*TTH
      DPRDCF(3) = DFDW
```

```
      DPRDCF(4) = 0.
      DPRDCF(5) = DFDMU*DMUDS + DFDSIG*DSIGDS + DFDCM3*DCM3DS
     1            + DFDCM4*DCM4DS
      DPRDCF(6) = 0.
      DPRDT=-DFDMU

      RETURN
      END
      SUBROUTINE ASYMCOF(TTHETA,HOL,SOL,ARGS,DERVS)

!PURPOSE: Compute the coefficients of the Edgeworth series model
!         of the axial divergence effects and their derivatives.
!         Written by E. Prince, December, 2004.

!CALLING ARGUMENTS:

      REAL*4        TTHETA              !2-theta
      REAL*4        HOL                 !H/L
      REAL*4        SOL                 !S/L
      REAL*4        ARGS(4)             !Arguments of profile function
      REAL*4        DERVS(4)            !Derivatives wrt S

!LOCAL VARIABLES:

      REAL*4        SIG
      REAL*4        CUM3
      REAL*4        CUM4
      REAL*4        DMUDS
      REAL*4        DSIGDS
      REAL*4        DCM3DS
      REAL*4        DCM4DS
      REAL*4        A
      REAL*4        B
      REAL*4        DADS
      REAL*4        DBDS
      REAL*4        DELTAM
      REAL*4        DELTA1
      REAL*4        DELTA2
      REAL*4        DELTA3
      REAL*4        DELTA4
      REAL*4        DD1DS
      REAL*4        DD2DS
      REAL*4        DD3DS
      REAL*4        DD4DS
      REAL*4        PIO180

!DATA STATEMENTS:

      DATA PIO180/0.000174533/

!CODE:

      IF(ABS(9000.-TTHETA).LT.4500.)THEN
         DELTAM=-0.5*TAN((9000.-TTHETA)*PIO180)/PIO180
      ELSE
         DELTAM=-(0.5/PIO180)/TAN(TTHETA*PIO180)
      ENDIF
```

```
      A=(HOL-SOL)**2
      DADS=-2.*(HOL-SOL)
      B=(HOL+SOL)**2
      DBDS=2.*(HOL+SOL)
      DELTA1=((A+B)/6.)*DELTAM
      ARGS(1)=DELTA1
      DELTA2=((B**2+B*A+A**2)/15.)*DELTAM**2
      DELTA3=((B**3+B**2*A+B*A**2+A**3)/28.)*DELTAM**3
      DELTA4=((B**4+B**3*A+B**2*A**2+B*A**3+A**4)/45.)*DELTAM**4
      SIG=SQRT(ARGS(2)+DELTA2-DELTA1**2)
      ARGS(2)=SIG
      CUM3=(DELTA3-3.*DELTA1*DELTA2+2.*DELTA1**3)                     !
      ARGS(3)=CUM3
      CUM4=(DELTA4-4.*DELTA1*DELTA3-3.*DELTA2**2+12.*DELTA2*DELTA1**2
     1      -6.*DELTA1**4)
      ARGS(4)=CUM4
      DD1DS=(DELTAM/6.)*(DADS+DBDS)
      DD2DS=(DELTAM**2/15.)*(DADS*(2.*A+B)+DBDS*(A+2.*B))
      DD3DS=(DELTAM**3/28.)*(DADS*(3.*A**2+2.*A*B+B**2)
     1                       +DBDS*(A**2+2.*A*B+3.*B**2))
      DD4DS=(DELTAM**4/45.)
     1      *(DADS*(4.*A**3+3.*A**2*B+2.*A*B**2+B**3)
     2      +(DBDS*(A**3+2.*A**2*B+3.*A*B**2+4.*B**3)))
      DMUDS=DD1DS
      DERVS(1)=DMUDS
      DSIGDS=(0.5*DD2DS-DELTA1*DD1DS)/SIG
      DERVS(2)=DSIGDS
      DCM3DS=((6.*DELTA1**2-3.*DELTA2)*DD1DS-3.*DELTA1*DD2DS
     1        +DD3DS)
      DERVS(3)=DCM3DS
      DCM4DS=((24.*(DELTA1*DELTA2-DELTA1**3)-4.*DELTA3)*DD1DS
     1        +6.*(2.*DELTA1**2-DELTA2)*DD2DS-4.*DELTA1*DD3DS
     2        +DD4DS)
      DERVS(4)=DCM4DS

      RETURN
      END
C Implementation of Finger-Cox-Jephcoat peak asymmetry function
C subroutines taken from program PROFILE.FOR found at
C http://www.ccp14.ac.uk/ccp/ccp14/ftp-mirror/larryfinger/PROFVAL/
C
        real*4 function Profval( Eta , Gamma , S_L , D_L , TwoTH ,
     1   TwoTH0 , dPRdT, dPRdG, dPRdE , dPRdS , dPRdD , Use_Asym )
c
c Returns value of Profile
c   Eta is the mixing coefficient between Gaussian and Lorentzian
c   Gamma is the FWHM
c   S_L is source width/detector distance
c   D_L is detector width/detector distance
c   TwoTH is point at which to evaluate the profile
c   TwoTH0 is two theta value for peak
c   dPRdT is derivative of profile wrt TwoTH0
c   dPRdG is derivative of profile wrt Gamma
c   dPRdE is derivative of profile wrt Eta
c   dPRdS is derivative of profile wrt S_L
c   dPRdD is derivative of profile wrt D_L
c   Use_Asym is true if asymmetry to be used
```

```
c
c
c Asymmetry due to axial divergence using the method of Finger, Cox and
c    Jephcoat, J. Appl. Cryst. 27, 892, 1992.

      implicit none
      real*4 Eta , Gamma , S_L , D_L , TwoTH
      real*4 TwoTH0 , dPRdT, dPRdG, dPRdE , dPRdS , dPRdD
      logical Use_Asym
      integer*4 NTERMS(14)/6,10,20,40,60,80,100,150,200,300,400,
     1   600,800,1000/
      integer Fstterm(14)/0,3,8,18,38,68,108,158,233,333,483,
     1   683,983,1383/
      real*4 RAD/57.2957795/

      integer*4 ArrayNum , K , NGT, ngt2 , it, i
      real*4 CsTH                   ! cos(theta)
      real*4 TTH         ! tan(theta)
      real*4 SnTwoTH           ! sin(twoth)
      real*4 CsTwoTH           ! cos(twoth)
      real*4 ApB         ! (S + H)/L
      real*4 AmB         ! (S - H)/L
      real*4 ApB2              ! (ApB) **2
      real*4 Einfl               ! 2phi value for inflection point
      real*4 Emin                ! 2phi value for minimum
      real*4 dEmindA             ! derivative of Emin wrt A
      real*4 tmp , tmp1 , tmp2  ! intermediate values
      real*4 WP(1883) , XP(1883)! Storage for Gauss-Legendre weights and
intervals
      real*4 Delta               ! Angle of integration for comvolution
      real*4 dDELTAdA            ! derivative of DELTA wrt A (S/L)
      real*4 sinDELTA            ! sine of DELTA
      real*4 cosDELTA            ! cosine of DELTA
      real*4 tanDELTA            ! tangent of DELTA
      real*4 RcosDELTA           ! 1/cos(DELTA)
      real*4 F , dFdA
      real*4 G , dGdA , dGdB , PsVoigt
      real*4 sumWG , sumWRG , sumWdGdA , sumWRdGdA ,sumWdGdB , sumWRdGdB
      real*4 sumWGdRdG , sumWGdRdE , sumWGdRdA , sumWGdRdB , sumWGdRd2t
!
! Values for the abscissas and weights of the Gauss-Legendre
!  N-point quadrature formula have been precomputed using routine
!  Gauleg from "Numerical Recipes" (Press, Flannery, Teukolsky
!  and Vetterling, 1986, Cambridge University Press,
!  ISBN 0 521 30811 9), and are stored in the DATA statements
!  for XP and WP below.
!
      data (xp(i),i=   1,  40)/
     $.2386192E+00,.6612094E+00,.9324695E+00,.1488743E+00,.4333954E+00,
     $.6794096E+00,.8650634E+00,.9739065E+00,.7652652E-01,.2277859E+00,
     $.3737061E+00,.5108670E+00,.6360537E+00,.7463319E+00,.8391170E+00,
     $.9122344E+00,.9639719E+00,.9931286E+00,.3877242E-01,.1160841E+00,
     $.1926976E+00,.2681522E+00,.3419941E+00,.4137792E+00,.4830758E+00,
     $.5494671E+00,.6125539E+00,.6719567E+00,.7273183E+00,.7783057E+00,
     $.8246122E+00,.8659595E+00,.9020988E+00,.9328128E+00,.9579168E+00,
     $.9772599E+00,.9907262E+00,.9982377E+00,.2595977E-01,.7780933E-01/
      data (xp(i),i=  41,  80)/
```

```
$.1294491E+00,.1807400E+00,.2315436E+00,.2817229E+00,.3311428E+00,
$.3796701E+00,.4271737E+00,.4735258E+00,.5186014E+00,.5622789E+00,
$.6044406E+00,.6449728E+00,.6837663E+00,.7207165E+00,.7557238E+00,
$.7886937E+00,.8195375E+00,.8481720E+00,.8745199E+00,.8985103E+00,
$.9200785E+00,.9391663E+00,.9557223E+00,.9697018E+00,.9810672E+00,
$.9897879E+00,.9958405E+00,.9992101E+00,.1951138E-01,.5850444E-01,
$.9740840E-01,.1361640E+00,.1747123E+00,.2129945E+00,.2509524E+00,
$.2885281E+00,.3256644E+00,.3623048E+00,.3983934E+00,.4338754E+00/
 data (xp(i),i=  81, 120)/
$.4686966E+00,.5028041E+00,.5361459E+00,.5686713E+00,.6003306E+00,
$.6310758E+00,.6608599E+00,.6896376E+00,.7173652E+00,.7440003E+00,
$.7695024E+00,.7938327E+00,.8169541E+00,.8388315E+00,.8594314E+00,
$.8787226E+00,.8966756E+00,.9132631E+00,.9284599E+00,.9422428E+00,
$.9545908E+00,.9654851E+00,.9749091E+00,.9828486E+00,.9892913E+00,
$.9942275E+00,.9976499E+00,.9995538E+00,.1562898E-01,.4687168E-01,
$.7806858E-01,.1091892E+00,.1402031E+00,.1710801E+00,.2017899E+00,
$.2323025E+00,.2625881E+00,.2926172E+00,.3223603E+00,.3517885E+00/
 data (xp(i),i= 121, 160)/
$.3808730E+00,.4095853E+00,.4378974E+00,.4657816E+00,.4932108E+00,
$.5201580E+00,.5465970E+00,.5725019E+00,.5978475E+00,.6226089E+00,
$.6467619E+00,.6702830E+00,.6931492E+00,.7153381E+00,.7368281E+00,
$.7575981E+00,.7776279E+00,.7968979E+00,.8153892E+00,.8330839E+00,
$.8499645E+00,.8660147E+00,.8812187E+00,.8955616E+00,.9090296E+00,
$.9216093E+00,.9332885E+00,.9440559E+00,.9539008E+00,.9628137E+00,
$.9707858E+00,.9778094E+00,.9838775E+00,.9889844E+00,.9931249E+00,
$.9962951E+00,.9984920E+00,.9997137E+00,.1043694E-01,.3130627E-01/
 data (xp(i),i= 161, 200)/
$.5216195E-01,.7299491E-01,.9379607E-01,.1145563E+00,.1352667E+00,
$.1559181E+00,.1765016E+00,.1970082E+00,.2174290E+00,.2377550E+00,
$.2579774E+00,.2780874E+00,.2980762E+00,.3179352E+00,.3376556E+00,
$.3572289E+00,.3766466E+00,.3959001E+00,.4149811E+00,.4338813E+00,
$.4525925E+00,.4711065E+00,.4894151E+00,.5075106E+00,.5253849E+00,
$.5430303E+00,.5604390E+00,.5776036E+00,.5945165E+00,.6111703E+00,
$.6275579E+00,.6436720E+00,.6595056E+00,.6750519E+00,.6903041E+00,
$.7052554E+00,.7198995E+00,.7342299E+00,.7482404E+00,.7619248E+00/
 data (xp(i),i= 201, 240)/
$.7752773E+00,.7882919E+00,.8009631E+00,.8132853E+00,.8252531E+00,
$.8368613E+00,.8481049E+00,.8589789E+00,.8694787E+00,.8795996E+00,
$.8893372E+00,.8986874E+00,.9076460E+00,.9162090E+00,.9243729E+00,
$.9321340E+00,.9394890E+00,.9464346E+00,.9529678E+00,.9590857E+00,
$.9647858E+00,.9700655E+00,.9749225E+00,.9793548E+00,.9833603E+00,
$.9869373E+00,.9900843E+00,.9927999E+00,.9950829E+00,.9969323E+00,
$.9983473E+00,.9993274E+00,.9998723E+00,.7834291E-02,.2350095E-01,
$.3916184E-01,.5481311E-01,.7045093E-01,.8607145E-01,.1016708E+00/
 data (xp(i),i= 241, 280)/
$.1172453E+00,.1327909E+00,.1483040E+00,.1637806E+00,.1792170E+00,
$.1946095E+00,.2099541E+00,.2252472E+00,.2404850E+00,.2556638E+00,
$.2707798E+00,.2858293E+00,.3008086E+00,.3157141E+00,.3305421E+00,
$.3452890E+00,.3599510E+00,.3745247E+00,.3890065E+00,.4033927E+00,
$.4176799E+00,.4318646E+00,.4459432E+00,.4599124E+00,.4737686E+00,
$.4875086E+00,.5011288E+00,.5146260E+00,.5279969E+00,.5412382E+00,
$.5543465E+00,.5673188E+00,.5801518E+00,.5928424E+00,.6053874E+00,
$.6177838E+00,.6300285E+00,.6421185E+00,.6540509E+00,.6658228E+00/
 data (xp(i),i= 281, 320)/
$.6774311E+00,.6888732E+00,.7001461E+00,.7112472E+00,.7221736E+00,
$.7329227E+00,.7434919E+00,.7538786E+00,.7640801E+00,.7740941E+00,
$.7839181E+00,.7935496E+00,.8029862E+00,.8122257E+00,.8212659E+00,
```

```
$.8301044E+00,.8387391E+00,.8471679E+00,.8553887E+00,.8633995E+00,
$.8711983E+00,.8787832E+00,.8861524E+00,.8933041E+00,.9002364E+00,
$.9069477E+00,.9134364E+00,.9197008E+00,.9257394E+00,.9315507E+00,
$.9371333E+00,.9424859E+00,.9476071E+00,.9524956E+00,.9571503E+00,
$.9615700E+00,.9657536E+00,.9697002E+00,.9734086E+00,.9768781E+00/
 data (xp(i),i= 321, 360)/
$.9801078E+00,.9830968E+00,.9858445E+00,.9883502E+00,.9906132E+00,
$.9926330E+00,.9944091E+00,.9959410E+00,.9972285E+00,.9982712E+00,
$.9990687E+00,.9996210E+00,.9999281E+00,.5227245E-02,.1568116E-01,
$.2613337E-01,.3658271E-01,.4702806E-01,.5746827E-01,.6790220E-01,
$.7832871E-01,.8874665E-01,.9915490E-01,.1095523E+00,.1199377E+00,
$.1303101E+00,.1406682E+00,.1510109E+00,.1613371E+00,.1716456E+00,
$.1819354E+00,.1922054E+00,.2024543E+00,.2126811E+00,.2228846E+00,
$.2330638E+00,.2432175E+00,.2533446E+00,.2634441E+00,.2735147E+00/
 data (xp(i),i= 361, 400)/
$.2835555E+00,.2935652E+00,.3035429E+00,.3134874E+00,.3233976E+00,
$.3332725E+00,.3431110E+00,.3529120E+00,.3626744E+00,.3723971E+00,
$.3820792E+00,.3917194E+00,.4013169E+00,.4108705E+00,.4203792E+00,
$.4298420E+00,.4392578E+00,.4486255E+00,.4579443E+00,.4672130E+00,
$.4764306E+00,.4855961E+00,.4947086E+00,.5037670E+00,.5127704E+00,
$.5217177E+00,.5306079E+00,.5394402E+00,.5482135E+00,.5569269E+00,
$.5655795E+00,.5741702E+00,.5826982E+00,.5911624E+00,.5995621E+00,
$.6078963E+00,.6161639E+00,.6243643E+00,.6324964E+00,.6405594E+00/
 data (xp(i),i= 401, 440)/
$.6485524E+00,.6564744E+00,.6643248E+00,.6721025E+00,.6798068E+00,
$.6874367E+00,.6949916E+00,.7024704E+00,.7098725E+00,.7171970E+00,
$.7244432E+00,.7316101E+00,.7386971E+00,.7457033E+00,.7526281E+00,
$.7594705E+00,.7662300E+00,.7729057E+00,.7794970E+00,.7860030E+00,
$.7924232E+00,.7987567E+00,.8050030E+00,.8111612E+00,.8172308E+00,
$.8232111E+00,.8291014E+00,.8349011E+00,.8406095E+00,.8462260E+00,
$.8517501E+00,.8571811E+00,.8625184E+00,.8677614E+00,.8729095E+00,
$.8779623E+00,.8829191E+00,.8877794E+00,.8925427E+00,.8972084E+00/
 data (xp(i),i= 441, 480)/
$.9017761E+00,.9062452E+00,.9106152E+00,.9148857E+00,.9190563E+00,
$.9231263E+00,.9270955E+00,.9309634E+00,.9347295E+00,.9383934E+00,
$.9419548E+00,.9454132E+00,.9487683E+00,.9520197E+00,.9551671E+00,
$.9582100E+00,.9611482E+00,.9639814E+00,.9667092E+00,.9693313E+00,
$.9718476E+00,.9742575E+00,.9765610E+00,.9787578E+00,.9808476E+00,
$.9828302E+00,.9847054E+00,.9864729E+00,.9881326E+00,.9896844E+00,
$.9911279E+00,.9924632E+00,.9936899E+00,.9948081E+00,.9958175E+00,
$.9967181E+00,.9975097E+00,.9981923E+00,.9987659E+00,.9992302E+00/
 data (xp(i),i= 481, 520)/
$.9995854E+00,.9998313E+00,.9999680E+00,.3922075E-02,.1176598E-01,
$.1960917E-01,.2745115E-01,.3529144E-01,.4312955E-01,.5096502E-01,
$.5879735E-01,.6662606E-01,.7445067E-01,.8227070E-01,.9008566E-01,
$.9789509E-01,.1056985E+00,.1134954E+00,.1212853E+00,.1290678E+00,
$.1368423E+00,.1446083E+00,.1523655E+00,.1601134E+00,.1678513E+00,
$.1755790E+00,.1832958E+00,.1910013E+00,.1986951E+00,.2063767E+00,
$.2140456E+00,.2217013E+00,.2293434E+00,.2369713E+00,.2445847E+00,
$.2521830E+00,.2597658E+00,.2673327E+00,.2748830E+00,.2824165E+00/
 data (xp(i),i= 521, 560)/
$.2899326E+00,.2974308E+00,.3049108E+00,.3123719E+00,.3198139E+00,
$.3272362E+00,.3346383E+00,.3420199E+00,.3493804E+00,.3567194E+00,
$.3640365E+00,.3713311E+00,.3786029E+00,.3858515E+00,.3930762E+00,
$.4002768E+00,.4074528E+00,.4146037E+00,.4217291E+00,.4288285E+00,
$.4359016E+00,.4429478E+00,.4499667E+00,.4569580E+00,.4639212E+00,
$.4708558E+00,.4777615E+00,.4846377E+00,.4914841E+00,.4983003E+00,
```

```
$.5050859E+00,.5118403E+00,.5185633E+00,.5252543E+00,.5319131E+00,
$.5385391E+00,.5451319E+00,.5516912E+00,.5582166E+00,.5647076E+00/
 data (xp(i),i= 561, 600)/
$.5711639E+00,.5775851E+00,.5839707E+00,.5903203E+00,.5966337E+00,
$.6029103E+00,.6091498E+00,.6153519E+00,.6215161E+00,.6276420E+00,
$.6337293E+00,.6397777E+00,.6457866E+00,.6517559E+00,.6576850E+00,
$.6635737E+00,.6694215E+00,.6752281E+00,.6809932E+00,.6867164E+00,
$.6923974E+00,.6980357E+00,.7036311E+00,.7091832E+00,.7146916E+00,
$.7201561E+00,.7255763E+00,.7309518E+00,.7362823E+00,.7415676E+00,
$.7468072E+00,.7520008E+00,.7571482E+00,.7622490E+00,.7673029E+00,
$.7723096E+00,.7772688E+00,.7821801E+00,.7870433E+00,.7918581E+00/
 data (xp(i),i= 601, 640)/
$.7966241E+00,.8013412E+00,.8060089E+00,.8106271E+00,.8151953E+00,
$.8197134E+00,.8241811E+00,.8285980E+00,.8329640E+00,.8372787E+00,
$.8415419E+00,.8457533E+00,.8499127E+00,.8540198E+00,.8580743E+00,
$.8620760E+00,.8660247E+00,.8699201E+00,.8737620E+00,.8775501E+00,
$.8812842E+00,.8849641E+00,.8885896E+00,.8921603E+00,.8956762E+00,
$.8991369E+00,.9025424E+00,.9058923E+00,.9091864E+00,.9124246E+00,
$.9156067E+00,.9187324E+00,.9218016E+00,.9248141E+00,.9277697E+00,
$.9306682E+00,.9335094E+00,.9362932E+00,.9390194E+00,.9416878E+00/
 data (xp(i),i= 641, 680)/
$.9442982E+00,.9468506E+00,.9493447E+00,.9517803E+00,.9541574E+00,
$.9564759E+00,.9587354E+00,.9609360E+00,.9630774E+00,.9651596E+00,
$.9671823E+00,.9691456E+00,.9710493E+00,.9728932E+00,.9746772E+00,
$.9764012E+00,.9780652E+00,.9796690E+00,.9812125E+00,.9826957E+00,
$.9841183E+00,.9854805E+00,.9867820E+00,.9880227E+00,.9892027E+00,
$.9903218E+00,.9913800E+00,.9923771E+00,.9933133E+00,.9941882E+00,
$.9950021E+00,.9957547E+00,.9964460E+00,.9970760E+00,.9976447E+00,
$.9981519E+00,.9985978E+00,.9989822E+00,.9993052E+00,.9995666E+00/
 data (xp(i),i= 681, 720)/
$.9997666E+00,.9999050E+00,.9999820E+00,.2615810E-02,.7847359E-02,
$.1307869E-01,.1830967E-01,.2354014E-01,.2876997E-01,.3399902E-01,
$.3922713E-01,.4445417E-01,.4967999E-01,.5490445E-01,.6012741E-01,
$.6534873E-01,.7056825E-01,.7578585E-01,.8100137E-01,.8621467E-01,
$.9142561E-01,.9663405E-01,.1018398E+00,.1070429E+00,.1122429E+00,
$.1174399E+00,.1226337E+00,.1278242E+00,.1330111E+00,.1381944E+00,
$.1433739E+00,.1485495E+00,.1537210E+00,.1588884E+00,.1640513E+00,
$.1692098E+00,.1743636E+00,.1795127E+00,.1846569E+00,.1897960E+00/
 data (xp(i),i= 721, 760)/
$.1949299E+00,.2000585E+00,.2051816E+00,.2102991E+00,.2154108E+00,
$.2205166E+00,.2256164E+00,.2307101E+00,.2357974E+00,.2408782E+00,
$.2459525E+00,.2510200E+00,.2560807E+00,.2611343E+00,.2661808E+00,
$.2712201E+00,.2762519E+00,.2812761E+00,.2862926E+00,.2913013E+00,
$.2963021E+00,.3012947E+00,.3062790E+00,.3112550E+00,.3162225E+00,
$.3211813E+00,.3261313E+00,.3310724E+00,.3360045E+00,.3409273E+00,
$.3458408E+00,.3507449E+00,.3556393E+00,.3605240E+00,.3653989E+00,
$.3702637E+00,.3751184E+00,.3799629E+00,.3847969E+00,.3896204E+00/
 data (xp(i),i= 761, 800)/
$.3944333E+00,.3992353E+00,.4040264E+00,.4088065E+00,.4135754E+00,
$.4183329E+00,.4230790E+00,.4278136E+00,.4325364E+00,.4372474E+00,
$.4419464E+00,.4466333E+00,.4513080E+00,.4559703E+00,.4606202E+00,
$.4652574E+00,.4698819E+00,.4744936E+00,.4790923E+00,.4836778E+00,
$.4882502E+00,.4928091E+00,.4973546E+00,.5018864E+00,.5064046E+00,
$.5109088E+00,.5153991E+00,.5198753E+00,.5243372E+00,.5287848E+00,
$.5332179E+00,.5376364E+00,.5420402E+00,.5464292E+00,.5508032E+00,
$.5551622E+00,.5595059E+00,.5638343E+00,.5681473E+00,.5724448E+00/
 data (xp(i),i= 801, 840)/
```

```
$.5767266E+00,.5809926E+00,.5852427E+00,.5894768E+00,.5936947E+00,
$.5978964E+00,.6020817E+00,.6062506E+00,.6104028E+00,.6145384E+00,
$.6186571E+00,.6227589E+00,.6268437E+00,.6309113E+00,.6349616E+00,
$.6389945E+00,.6430100E+00,.6470079E+00,.6509880E+00,.6549504E+00,
$.6588948E+00,.6628211E+00,.6667294E+00,.6706194E+00,.6744910E+00,
$.6783442E+00,.6821788E+00,.6859947E+00,.6897919E+00,.6935702E+00,
$.6973295E+00,.7010697E+00,.7047907E+00,.7084924E+00,.7121748E+00,
$.7158376E+00,.7194809E+00,.7231044E+00,.7267082E+00,.7302921E+00/
 data (xp(i),i= 841, 880)/
$.7338560E+00,.7373998E+00,.7409234E+00,.7444268E+00,.7479097E+00,
$.7513722E+00,.7548142E+00,.7582355E+00,.7616360E+00,.7650157E+00,
$.7683744E+00,.7717121E+00,.7750287E+00,.7783241E+00,.7815982E+00,
$.7848508E+00,.7880821E+00,.7912917E+00,.7944797E+00,.7976459E+00,
$.8007903E+00,.8039128E+00,.8070132E+00,.8100916E+00,.8131479E+00,
$.8161818E+00,.8191934E+00,.8221826E+00,.8251493E+00,.8280935E+00,
$.8310149E+00,.8339136E+00,.8367895E+00,.8396425E+00,.8424725E+00,
$.8452794E+00,.8480632E+00,.8508238E+00,.8535611E+00,.8562750E+00/
 data (xp(i),i= 881, 920)/
$.8589656E+00,.8616325E+00,.8642760E+00,.8668957E+00,.8694918E+00,
$.8720640E+00,.8746124E+00,.8771368E+00,.8796372E+00,.8821136E+00,
$.8845658E+00,.8869937E+00,.8893975E+00,.8917768E+00,.8941318E+00,
$.8964623E+00,.8987683E+00,.9010496E+00,.9033063E+00,.9055383E+00,
$.9077455E+00,.9099278E+00,.9120852E+00,.9142177E+00,.9163252E+00,
$.9184075E+00,.9204648E+00,.9224968E+00,.9245036E+00,.9264851E+00,
$.9284412E+00,.9303720E+00,.9322772E+00,.9341570E+00,.9360111E+00,
$.9378397E+00,.9396426E+00,.9414198E+00,.9431712E+00,.9448967E+00/
 data (xp(i),i= 921, 960)/
$.9465965E+00,.9482703E+00,.9499181E+00,.9515400E+00,.9531358E+00,
$.9547056E+00,.9562492E+00,.9577666E+00,.9592578E+00,.9607228E+00,
$.9621615E+00,.9635738E+00,.9649597E+00,.9663193E+00,.9676524E+00,
$.9689590E+00,.9702391E+00,.9714927E+00,.9727196E+00,.9739199E+00,
$.9750936E+00,.9762406E+00,.9773609E+00,.9784544E+00,.9795211E+00,
$.9805610E+00,.9815741E+00,.9825603E+00,.9835197E+00,.9844521E+00,
$.9853575E+00,.9862360E+00,.9870876E+00,.9879120E+00,.9887095E+00,
$.9894799E+00,.9902232E+00,.9909394E+00,.9916285E+00,.9922904E+00/
 data (xp(i),i= 961,1000)/
$.9929252E+00,.9935328E+00,.9941132E+00,.9946664E+00,.9951924E+00,
$.9956911E+00,.9961626E+00,.9966068E+00,.9970238E+00,.9974135E+00,
$.9977758E+00,.9981109E+00,.9984186E+00,.9986990E+00,.9989521E+00,
$.9991778E+00,.9993762E+00,.9995472E+00,.9996909E+00,.9998072E+00,
$.9998962E+00,.9999577E+00,.9999920E+00,.1962267E-02,.5886772E-02,
$.9811186E-02,.1373545E-01,.1765950E-01,.2158328E-01,.2550673E-01,
$.2942978E-01,.3335238E-01,.3727447E-01,.4119598E-01,.4511686E-01,
$.4903704E-01,.5295647E-01,.5687508E-01,.6079282E-01,.6470962E-01/
 data (xp(i),i=1001,1040)/
$.6862542E-01,.7254017E-01,.7645380E-01,.8036625E-01,.8427746E-01,
$.8818738E-01,.9209594E-01,.9600308E-01,.9990874E-01,.1038129E+00,
$.1077154E+00,.1116162E+00,.1155154E+00,.1194128E+00,.1233083E+00,
$.1272019E+00,.1310936E+00,.1349832E+00,.1388708E+00,.1427562E+00,
$.1466395E+00,.1505204E+00,.1543991E+00,.1582754E+00,.1621492E+00,
$.1660205E+00,.1698893E+00,.1737555E+00,.1776190E+00,.1814798E+00,
$.1853377E+00,.1891928E+00,.1930450E+00,.1968942E+00,.2007404E+00,
$.2045835E+00,.2084235E+00,.2122602E+00,.2160937E+00,.2199238E+00/
 data (xp(i),i=1041,1080)/
$.2237505E+00,.2275738E+00,.2313936E+00,.2352099E+00,.2390225E+00,
$.2428314E+00,.2466366E+00,.2504380E+00,.2542355E+00,.2580292E+00,
$.2618188E+00,.2656044E+00,.2693859E+00,.2731633E+00,.2769365E+00,
```

```
$.2807054E+00,.2844699E+00,.2882301E+00,.2919859E+00,.2957372E+00,
$.2994839E+00,.3032259E+00,.3069634E+00,.3106961E+00,.3144240E+00,
$.3181470E+00,.3218652E+00,.3255784E+00,.3292866E+00,.3329897E+00,
$.3366877E+00,.3403805E+00,.3440681E+00,.3477503E+00,.3514272E+00,
$.3550987E+00,.3587648E+00,.3624253E+00,.3660802E+00,.3697295E+00/
 data (xp(i),i=1081,1120)/
$.3733731E+00,.3770109E+00,.3806429E+00,.3842691E+00,.3878893E+00,
$.3915036E+00,.3951118E+00,.3987140E+00,.4023100E+00,.4058998E+00,
$.4094834E+00,.4130607E+00,.4166316E+00,.4201960E+00,.4237541E+00,
$.4273055E+00,.4308504E+00,.4343887E+00,.4379203E+00,.4414451E+00,
$.4449632E+00,.4484743E+00,.4519786E+00,.4554759E+00,.4589662E+00,
$.4624494E+00,.4659255E+00,.4693945E+00,.4728562E+00,.4763106E+00,
$.4797577E+00,.4831973E+00,.4866296E+00,.4900543E+00,.4934715E+00,
$.4968812E+00,.5002831E+00,.5036774E+00,.5070638E+00,.5104425E+00/
 data (xp(i),i=1121,1160)/
$.5138133E+00,.5171762E+00,.5205312E+00,.5238781E+00,.5272169E+00,
$.5305477E+00,.5338702E+00,.5371846E+00,.5404906E+00,.5437884E+00,
$.5470777E+00,.5503587E+00,.5536311E+00,.5568951E+00,.5601504E+00,
$.5633972E+00,.5666352E+00,.5698645E+00,.5730851E+00,.5762968E+00,
$.5794996E+00,.5826936E+00,.5858785E+00,.5890544E+00,.5922213E+00,
$.5953790E+00,.5985276E+00,.6016669E+00,.6047970E+00,.6079177E+00,
$.6110291E+00,.6141311E+00,.6172236E+00,.6203066E+00,.6233801E+00,
$.6264440E+00,.6294982E+00,.6325427E+00,.6355775E+00,.6386024E+00/
 data (xp(i),i=1161,1200)/
$.6416176E+00,.6446229E+00,.6476182E+00,.6506036E+00,.6535789E+00,
$.6565442E+00,.6594994E+00,.6624444E+00,.6653792E+00,.6683037E+00,
$.6712180E+00,.6741219E+00,.6770155E+00,.6798986E+00,.6827712E+00,
$.6856333E+00,.6884849E+00,.6913259E+00,.6941562E+00,.6969758E+00,
$.6997847E+00,.7025828E+00,.7053701E+00,.7081465E+00,.7109120E+00,
$.7136666E+00,.7164102E+00,.7191427E+00,.7218642E+00,.7245746E+00,
$.7272737E+00,.7299617E+00,.7326385E+00,.7353039E+00,.7379581E+00,
$.7406008E+00,.7432322E+00,.7458521E+00,.7484606E+00,.7510575E+00/
 data (xp(i),i=1201,1240)/
$.7536428E+00,.7562165E+00,.7587786E+00,.7613290E+00,.7638676E+00,
$.7663945E+00,.7689096E+00,.7714129E+00,.7739043E+00,.7763837E+00,
$.7788512E+00,.7813067E+00,.7837502E+00,.7861816E+00,.7886009E+00,
$.7910080E+00,.7934030E+00,.7957857E+00,.7981562E+00,.8005144E+00,
$.8028602E+00,.8051937E+00,.8075148E+00,.8098234E+00,.8121196E+00,
$.8144033E+00,.8166744E+00,.8189330E+00,.8211789E+00,.8234122E+00,
$.8256328E+00,.8278407E+00,.8300358E+00,.8322182E+00,.8343877E+00,
$.8365444E+00,.8386882E+00,.8408191E+00,.8429370E+00,.8450420E+00/
 data (xp(i),i=1241,1280)/
$.8471339E+00,.8492128E+00,.8512786E+00,.8533313E+00,.8553709E+00,
$.8573972E+00,.8594104E+00,.8614104E+00,.8633970E+00,.8653704E+00,
$.8673304E+00,.8692771E+00,.8712104E+00,.8731303E+00,.8750367E+00,
$.8769297E+00,.8788091E+00,.8806750E+00,.8825274E+00,.8843662E+00,
$.8861913E+00,.8880028E+00,.8898006E+00,.8915847E+00,.8933550E+00,
$.8951117E+00,.8968545E+00,.8985835E+00,.9002987E+00,.9020000E+00,
$.9036874E+00,.9053609E+00,.9070204E+00,.9086660E+00,.9102976E+00,
$.9119152E+00,.9135187E+00,.9151081E+00,.9166835E+00,.9182447E+00/
 data (xp(i),i=1281,1320)/
$.9197918E+00,.9213247E+00,.9228435E+00,.9243480E+00,.9258383E+00,
$.9273143E+00,.9287760E+00,.9302235E+00,.9316566E+00,.9330754E+00,
$.9344797E+00,.9358697E+00,.9372453E+00,.9386065E+00,.9399532E+00,
$.9412854E+00,.9426031E+00,.9439063E+00,.9451950E+00,.9464691E+00,
$.9477286E+00,.9489735E+00,.9502038E+00,.9514195E+00,.9526205E+00,
$.9538069E+00,.9549785E+00,.9561355E+00,.9572777E+00,.9584052E+00,
```

```
$.9595179E+00,.9606159E+00,.9616990E+00,.9627673E+00,.9638208E+00,
$.9648595E+00,.9658833E+00,.9668922E+00,.9678863E+00,.9688654E+00/
 data (xp(i),i=1321,1360)/
$.9698296E+00,.9707788E+00,.9717132E+00,.9726325E+00,.9735369E+00,
$.9744262E+00,.9753006E+00,.9761600E+00,.9770043E+00,.9778335E+00,
$.9786477E+00,.9794468E+00,.9802309E+00,.9809998E+00,.9817537E+00,
$.9824924E+00,.9832160E+00,.9839244E+00,.9846177E+00,.9852958E+00,
$.9859587E+00,.9866065E+00,.9872390E+00,.9878564E+00,.9884585E+00,
$.9890455E+00,.9896171E+00,.9901736E+00,.9907148E+00,.9912407E+00,
$.9917514E+00,.9922468E+00,.9927269E+00,.9931917E+00,.9936412E+00,
$.9940754E+00,.9944943E+00,.9948979E+00,.9952862E+00,.9956591E+00/
 data (xp(i),i=1361,1400)/
$.9960167E+00,.9963590E+00,.9966859E+00,.9969974E+00,.9972936E+00,
$.9975745E+00,.9978400E+00,.9980901E+00,.9983248E+00,.9985442E+00,
$.9987482E+00,.9989368E+00,.9991100E+00,.9992678E+00,.9994103E+00,
$.9995373E+00,.9996489E+00,.9997452E+00,.9998261E+00,.9998915E+00,
$.9999416E+00,.9999762E+00,.9999955E+00,.1570010E-02,.4710016E-02,
$.7849975E-02,.1098986E-01,.1412963E-01,.1726926E-01,.2040873E-01,
$.2354799E-01,.2668702E-01,.2982579E-01,.3296426E-01,.3610241E-01,
$.3924020E-01,.4237761E-01,.4551459E-01,.4865113E-01,.5178719E-01/
 data (xp(i),i=1401,1440)/
$.5492274E-01,.5805775E-01,.6119218E-01,.6432601E-01,.6745921E-01,
$.7059174E-01,.7372358E-01,.7685468E-01,.7998504E-01,.8311460E-01,
$.8624334E-01,.8937123E-01,.9249824E-01,.9562434E-01,.9874950E-01,
$.1018737E+00,.1049969E+00,.1081190E+00,.1112401E+00,.1143601E+00,
$.1174789E+00,.1205966E+00,.1237131E+00,.1268284E+00,.1299424E+00,
$.1330552E+00,.1361666E+00,.1392767E+00,.1423855E+00,.1454928E+00,
$.1485987E+00,.1517031E+00,.1548060E+00,.1579074E+00,.1610073E+00,
$.1641055E+00,.1672022E+00,.1702971E+00,.1733905E+00,.1764821E+00/
 data (xp(i),i=1441,1480)/
$.1795719E+00,.1826600E+00,.1857463E+00,.1888308E+00,.1919134E+00,
$.1949941E+00,.1980728E+00,.2011497E+00,.2042245E+00,.2072973E+00,
$.2103681E+00,.2134368E+00,.2165035E+00,.2195679E+00,.2226302E+00,
$.2256904E+00,.2287482E+00,.2318039E+00,.2348572E+00,.2379083E+00,
$.2409569E+00,.2440033E+00,.2470472E+00,.2500886E+00,.2531276E+00,
$.2561641E+00,.2591981E+00,.2622295E+00,.2652584E+00,.2682846E+00,
$.2713082E+00,.2743291E+00,.2773473E+00,.2803628E+00,.2833755E+00,
$.2863854E+00,.2893925E+00,.2923967E+00,.2953980E+00,.2983965E+00/
 data (xp(i),i=1481,1520)/
$.3013920E+00,.3043845E+00,.3073740E+00,.3103605E+00,.3133439E+00,
$.3163243E+00,.3193015E+00,.3222756E+00,.3252465E+00,.3282141E+00,
$.3311786E+00,.3341398E+00,.3370977E+00,.3400522E+00,.3430035E+00,
$.3459513E+00,.3488957E+00,.3518367E+00,.3547742E+00,.3577082E+00,
$.3606387E+00,.3635657E+00,.3664890E+00,.3694087E+00,.3723248E+00,
$.3752373E+00,.3781460E+00,.3810510E+00,.3839522E+00,.3868497E+00,
$.3897433E+00,.3926331E+00,.3955190E+00,.3984010E+00,.4012791E+00,
$.4041533E+00,.4070234E+00,.4098896E+00,.4127517E+00,.4156097E+00/
 data (xp(i),i=1521,1560)/
$.4184636E+00,.4213134E+00,.4241591E+00,.4270006E+00,.4298378E+00,
$.4326708E+00,.4354996E+00,.4383241E+00,.4411442E+00,.4439600E+00,
$.4467714E+00,.4495784E+00,.4523810E+00,.4551791E+00,.4579727E+00,
$.4607618E+00,.4635464E+00,.4663264E+00,.4691018E+00,.4718726E+00,
$.4746387E+00,.4774001E+00,.4801569E+00,.4829089E+00,.4856561E+00,
$.4883986E+00,.4911362E+00,.4938690E+00,.4965969E+00,.4993199E+00,
$.5020381E+00,.5047512E+00,.5074594E+00,.5101626E+00,.5128607E+00,
$.5155538E+00,.5182418E+00,.5209247E+00,.5236025E+00,.5262750E+00/
 data (xp(i),i=1561,1600)/
```

```
$.5289425E+00,.5316046E+00,.5342616E+00,.5369133E+00,.5395597E+00,
$.5422007E+00,.5448365E+00,.5474668E+00,.5500918E+00,.5527113E+00,
$.5553254E+00,.5579340E+00,.5605371E+00,.5631347E+00,.5657267E+00,
$.5683131E+00,.5708940E+00,.5734692E+00,.5760387E+00,.5786026E+00,
$.5811608E+00,.5837132E+00,.5862599E+00,.5888008E+00,.5913360E+00,
$.5938652E+00,.5963886E+00,.5989062E+00,.6014178E+00,.6039235E+00,
$.6064233E+00,.6089170E+00,.6114048E+00,.6138865E+00,.6163622E+00,
$.6188318E+00,.6212953E+00,.6237527E+00,.6262040E+00,.6286490E+00/
 data (xp(i),i=1601,1640)/
$.6310879E+00,.6335205E+00,.6359469E+00,.6383670E+00,.6407808E+00,
$.6431883E+00,.6455895E+00,.6479843E+00,.6503727E+00,.6527547E+00,
$.6551303E+00,.6574994E+00,.6598620E+00,.6622181E+00,.6645677E+00,
$.6669107E+00,.6692472E+00,.6715770E+00,.6739003E+00,.6762169E+00,
$.6785268E+00,.6808300E+00,.6831266E+00,.6854163E+00,.6876994E+00,
$.6899756E+00,.6922451E+00,.6945077E+00,.6967635E+00,.6990124E+00,
$.7012544E+00,.7034895E+00,.7057176E+00,.7079388E+00,.7101530E+00,
$.7123603E+00,.7145605E+00,.7167536E+00,.7189397E+00,.7211187E+00/
 data (xp(i),i=1641,1680)/
$.7232906E+00,.7254553E+00,.7276129E+00,.7297634E+00,.7319066E+00,
$.7340426E+00,.7361714E+00,.7382929E+00,.7404071E+00,.7425141E+00,
$.7446137E+00,.7467060E+00,.7487909E+00,.7508684E+00,.7529385E+00,
$.7550013E+00,.7570565E+00,.7591043E+00,.7611446E+00,.7631774E+00,
$.7652027E+00,.7672204E+00,.7692306E+00,.7712332E+00,.7732282E+00,
$.7752155E+00,.7771953E+00,.7791673E+00,.7811317E+00,.7830884E+00,
$.7850373E+00,.7869785E+00,.7889120E+00,.7908376E+00,.7927555E+00,
$.7946656E+00,.7965678E+00,.7984622E+00,.8003486E+00,.8022273E+00/
 data (xp(i),i=1681,1720)/
$.8040979E+00,.8059607E+00,.8078155E+00,.8096624E+00,.8115013E+00,
$.8133321E+00,.8151550E+00,.8169698E+00,.8187765E+00,.8205752E+00,
$.8223658E+00,.8241483E+00,.8259227E+00,.8276889E+00,.8294470E+00,
$.8311968E+00,.8329385E+00,.8346720E+00,.8363972E+00,.8381142E+00,
$.8398229E+00,.8415234E+00,.8432156E+00,.8448994E+00,.8465749E+00,
$.8482421E+00,.8499009E+00,.8515513E+00,.8531933E+00,.8548269E+00,
$.8564521E+00,.8580688E+00,.8596771E+00,.8612769E+00,.8628682E+00,
$.8644510E+00,.8660253E+00,.8675910E+00,.8691482E+00,.8706968E+00/
 data (xp(i),i=1721,1760)/
$.8722369E+00,.8737683E+00,.8752911E+00,.8768053E+00,.8783108E+00,
$.8798077E+00,.8812959E+00,.8827754E+00,.8842463E+00,.8857083E+00,
$.8871617E+00,.8886063E+00,.8900422E+00,.8914692E+00,.8928875E+00,
$.8942970E+00,.8956977E+00,.8970895E+00,.8984725E+00,.8998466E+00,
$.9012119E+00,.9025683E+00,.9039157E+00,.9052543E+00,.9065839E+00,
$.9079046E+00,.9092164E+00,.9105192E+00,.9118130E+00,.9130978E+00,
$.9143736E+00,.9156404E+00,.9168982E+00,.9181469E+00,.9193866E+00,
$.9206172E+00,.9218387E+00,.9230511E+00,.9242545E+00,.9254487E+00/
 data (xp(i),i=1761,1800)/
$.9266338E+00,.9278098E+00,.9289766E+00,.9301343E+00,.9312828E+00,
$.9324221E+00,.9335522E+00,.9346731E+00,.9357848E+00,.9368872E+00,
$.9379805E+00,.9390645E+00,.9401392E+00,.9412046E+00,.9422608E+00,
$.9433077E+00,.9443453E+00,.9453735E+00,.9463925E+00,.9474021E+00,
$.9484024E+00,.9493933E+00,.9503749E+00,.9513471E+00,.9523099E+00,
$.9532633E+00,.9542073E+00,.9551420E+00,.9560672E+00,.9569829E+00,
$.9578893E+00,.9587862E+00,.9596736E+00,.9605516E+00,.9614201E+00,
$.9622791E+00,.9631287E+00,.9639687E+00,.9647992E+00,.9656203E+00/
 data (xp(i),i=1801,1840)/
$.9664318E+00,.9672338E+00,.9680262E+00,.9688091E+00,.9695824E+00,
$.9703462E+00,.9711004E+00,.9718451E+00,.9725801E+00,.9733056E+00,
$.9740215E+00,.9747278E+00,.9754244E+00,.9761115E+00,.9767889E+00,
```

```
$.9774567E+00,.9781148E+00,.9787633E+00,.9794022E+00,.9800314E+00,
$.9806509E+00,.9812608E+00,.9818610E+00,.9824515E+00,.9830323E+00,
$.9836035E+00,.9841649E+00,.9847166E+00,.9852586E+00,.9857909E+00,
$.9863135E+00,.9868264E+00,.9873295E+00,.9878229E+00,.9883066E+00,
$.9887805E+00,.9892447E+00,.9896991E+00,.9901437E+00,.9905786E+00/
 data (xp(i),i=1841,1880)/
$.9910037E+00,.9914191E+00,.9918247E+00,.9922205E+00,.9926065E+00,
$.9929827E+00,.9933492E+00,.9937058E+00,.9940527E+00,.9943897E+00,
$.9947169E+00,.9950344E+00,.9953420E+00,.9956398E+00,.9959278E+00,
$.9962060E+00,.9964743E+00,.9967328E+00,.9969815E+00,.9972204E+00,
$.9974494E+00,.9976686E+00,.9978780E+00,.9980775E+00,.9982672E+00,
$.9984471E+00,.9986171E+00,.9987772E+00,.9989275E+00,.9990680E+00,
$.9991986E+00,.9993193E+00,.9994302E+00,.9995313E+00,.9996225E+00,
$.9997038E+00,.9997753E+00,.9998369E+00,.9998886E+00,.9999306E+00/
 data (xp(i),i=1881,1883)/
$.9999626E+00,.9999848E+00,.9999971E+00/
 data (wp(i),i=   1,  40)/
$.4679139E+00,.3607616E+00,.1713245E+00,.2955242E+00,.2692667E+00,
$.2190864E+00,.1494513E+00,.6667134E-01,.1527534E+00,.1491730E+00,
$.1420961E+00,.1316886E+00,.1181945E+00,.1019301E+00,.8327674E-01,
$.6267205E-01,.4060143E-01,.1761401E-01,.7750595E-01,.7703982E-01,
$.7611036E-01,.7472317E-01,.7288658E-01,.7061165E-01,.6791205E-01,
$.6480401E-01,.6130624E-01,.5743977E-01,.5322785E-01,.4869581E-01,
$.4387091E-01,.3878217E-01,.3346020E-01,.2793701E-01,.2224585E-01,
$.1642106E-01,.1049828E-01,.4521277E-02,.5190788E-01,.5176794E-01/
 data (wp(i),i=  41,  80)/
$.5148845E-01,.5107016E-01,.5051418E-01,.4982204E-01,.4899558E-01,
$.4803703E-01,.4694899E-01,.4573438E-01,.4439648E-01,.4293889E-01,
$.4136555E-01,.3968070E-01,.3788887E-01,.3599490E-01,.3400389E-01,
$.3192122E-01,.2975249E-01,.2750356E-01,.2518048E-01,.2278952E-01,
$.2033712E-01,.1782990E-01,.1527462E-01,.1267817E-01,.1004756E-01,
$.7389931E-02,.4712730E-02,.2026812E-02,.3901781E-01,.3895840E-01,
$.3883965E-01,.3866176E-01,.3842499E-01,.3812971E-01,.3777636E-01,
$.3736549E-01,.3689771E-01,.3637375E-01,.3579439E-01,.3516053E-01/
 data (wp(i),i=  81, 120)/
$.3447312E-01,.3373321E-01,.3294194E-01,.3210050E-01,.3121017E-01,
$.3027232E-01,.2928837E-01,.2825982E-01,.2718823E-01,.2607524E-01,
$.2492254E-01,.2373188E-01,.2250509E-01,.2124403E-01,.1995061E-01,
$.1862681E-01,.1727465E-01,.1589618E-01,.1449351E-01,.1306876E-01,
$.1162411E-01,.1016177E-01,.8683945E-02,.7192905E-02,.5690922E-02,
$.4180313E-02,.2663534E-02,.1144950E-02,.3125542E-01,.3122488E-01,
$.3116384E-01,.3107234E-01,.3095048E-01,.3079838E-01,.3061619E-01,
$.3040408E-01,.3016227E-01,.2989098E-01,.2959049E-01,.2926108E-01/
 data (wp(i),i= 121, 160)/
$.2890309E-01,.2851685E-01,.2810276E-01,.2766120E-01,.2719261E-01,
$.2669746E-01,.2617622E-01,.2562940E-01,.2505754E-01,.2446120E-01,
$.2384096E-01,.2319742E-01,.2253122E-01,.2184300E-01,.2113344E-01,
$.2040323E-01,.1965309E-01,.1888374E-01,.1809594E-01,.1729046E-01,
$.1646809E-01,.1562962E-01,.1477588E-01,.1390771E-01,.1302595E-01,
$.1213146E-01,.1122511E-01,.1030780E-01,.9380420E-02,.8443871E-02,
$.7499073E-02,.6546948E-02,.5588428E-02,.4624450E-02,.3655961E-02,
$.2683925E-02,.1709393E-02,.7346345E-03,.2087312E-01,.2086402E-01/
 data (wp(i),i= 161, 200)/
$.2084584E-01,.2081857E-01,.2078223E-01,.2073683E-01,.2068240E-01,
$.2061896E-01,.2054653E-01,.2046515E-01,.2037486E-01,.2027568E-01,
$.2016767E-01,.2005088E-01,.1992534E-01,.1979113E-01,.1964829E-01,
$.1949689E-01,.1933700E-01,.1916867E-01,.1899200E-01,.1880705E-01,
```

```
     $.1861391E-01,.1841266E-01,.1820338E-01,.1798617E-01,.1776113E-01,
     $.1752835E-01,.1728792E-01,.1703997E-01,.1678459E-01,.1652190E-01,
     $.1625201E-01,.1597504E-01,.1569110E-01,.1540033E-01,.1510285E-01,
     $.1479879E-01,.1448828E-01,.1417146E-01,.1384846E-01,.1351943E-01/
      data (wp(i),i= 201, 240)/
     $.1318451E-01,.1284384E-01,.1249758E-01,.1214587E-01,.1178887E-01,
     $.1142673E-01,.1105962E-01,.1068768E-01,.1031109E-01,.9930004E-02,
     $.9544593E-02,.9155022E-02,.8761463E-02,.8364086E-02,.7963064E-02,
     $.7558573E-02,.7150788E-02,.6739888E-02,.6326051E-02,.5909457E-02,
     $.5490289E-02,.5068728E-02,.4644959E-02,.4219166E-02,.3791535E-02,
     $.3362252E-02,.2931504E-02,.2499479E-02,.2066366E-02,.1632357E-02,
     $.1197647E-02,.7624721E-03,.3276087E-03,.1566826E-01,.1566442E-01,
     $.1565672E-01,.1564519E-01,.1562981E-01,.1561059E-01,.1558755E-01/
      data (wp(i),i= 241, 280)/
     $.1556067E-01,.1552998E-01,.1549547E-01,.1545716E-01,.1541506E-01,
     $.1536917E-01,.1531950E-01,.1526608E-01,.1520891E-01,.1514800E-01,
     $.1508338E-01,.1501505E-01,.1494303E-01,.1486735E-01,.1478802E-01,
     $.1470505E-01,.1461848E-01,.1452832E-01,.1443459E-01,.1433731E-01,
     $.1423652E-01,.1413223E-01,.1402447E-01,.1391327E-01,.1379866E-01,
     $.1368065E-01,.1355929E-01,.1343460E-01,.1330661E-01,.1317535E-01,
     $.1304086E-01,.1290316E-01,.1276230E-01,.1261831E-01,.1247122E-01,
     $.1232106E-01,.1216788E-01,.1201172E-01,.1185260E-01,.1169058E-01/
      data (wp(i),i= 281, 320)/
     $.1152568E-01,.1135796E-01,.1118744E-01,.1101418E-01,.1083822E-01,
     $.1065959E-01,.1047835E-01,.1029454E-01,.1010820E-01,.9919373E-02,
     $.9728115E-02,.9534468E-02,.9338480E-02,.9140200E-02,.8939676E-02,
     $.8736957E-02,.8532093E-02,.8325134E-02,.8116132E-02,.7905137E-02,
     $.7692201E-02,.7477377E-02,.7260717E-02,.7042274E-02,.6822103E-02,
     $.6600256E-02,.6376790E-02,.6151757E-02,.5925215E-02,.5697218E-02,
     $.5467822E-02,.5237083E-02,.5005059E-02,.4771806E-02,.4537382E-02,
     $.4301844E-02,.4065249E-02,.3827657E-02,.3589125E-02,.3349711E-02/
      data (wp(i),i= 321, 360)/
     $.3109476E-02,.2868477E-02,.2626773E-02,.2384425E-02,.2141492E-02,
     $.1898033E-02,.1654108E-02,.1409777E-02,.1165101E-02,.9201405E-03,
     $.6749606E-03,.4296466E-03,.1845901E-03,.1045439E-01,.1045325E-01,
     $.1045097E-01,.1044754E-01,.1044297E-01,.1043726E-01,.1043041E-01,
     $.1042242E-01,.1041329E-01,.1040302E-01,.1039161E-01,.1037907E-01,
     $.1036539E-01,.1035058E-01,.1033464E-01,.1031758E-01,.1029938E-01,
     $.1028006E-01,.1025961E-01,.1023804E-01,.1021535E-01,.1019155E-01,
     $.1016663E-01,.1014060E-01,.1011347E-01,.1008523E-01,.1005588E-01/
      data (wp(i),i= 361, 400)/
     $.1002544E-01,.9993899E-02,.9961267E-02,.9927547E-02,.9892741E-02,
     $.9856855E-02,.9819891E-02,.9781854E-02,.9742747E-02,.9702576E-02,
     $.9661345E-02,.9619057E-02,.9575718E-02,.9531333E-02,.9485905E-02,
     $.9439441E-02,.9391946E-02,.9343424E-02,.9293880E-02,.9243321E-02,
     $.9191751E-02,.9139177E-02,.9085604E-02,.9031038E-02,.8975485E-02,
     $.8918951E-02,.8861442E-02,.8802965E-02,.8743525E-02,.8683130E-02,
     $.8621786E-02,.8559499E-02,.8496277E-02,.8432127E-02,.8367054E-02,
     $.8301068E-02,.8234174E-02,.8166380E-02,.8097693E-02,.8028121E-02/
      data (wp(i),i= 401, 440)/
     $.7957672E-02,.7886353E-02,.7814173E-02,.7741138E-02,.7667257E-02,
     $.7592538E-02,.7516989E-02,.7440619E-02,.7363435E-02,.7285447E-02,
     $.7206662E-02,.7127090E-02,.7046739E-02,.6965617E-02,.6883734E-02,
     $.6801099E-02,.6717721E-02,.6633608E-02,.6548770E-02,.6463217E-02,
     $.6376957E-02,.6290000E-02,.6202356E-02,.6114033E-02,.6025043E-02,
     $.5935394E-02,.5845096E-02,.5754159E-02,.5662594E-02,.5570409E-02,
     $.5477616E-02,.5384224E-02,.5290244E-02,.5195685E-02,.5100559E-02,
```

```
$.5004875E-02,.4908644E-02,.4811876E-02,.4714583E-02,.4616774E-02/
 data (wp(i),i= 441, 480)/
$.4518461E-02,.4419654E-02,.4320364E-02,.4220601E-02,.4120378E-02,
$.4019704E-02,.3918590E-02,.3817049E-02,.3715090E-02,.3612725E-02,
$.3509965E-02,.3406822E-02,.3303306E-02,.3199429E-02,.3095203E-02,
$.2990638E-02,.2885746E-02,.2780539E-02,.2675029E-02,.2569225E-02,
$.2463141E-02,.2356788E-02,.2250177E-02,.2143320E-02,.2036229E-02,
$.1928915E-02,.1821391E-02,.1713667E-02,.1605756E-02,.1497670E-02,
$.1389420E-02,.1281018E-02,.1172476E-02,.1063806E-02,.9550200E-03,
$.8461294E-03,.7371464E-03,.6280830E-03,.5189512E-03,.4097636E-03/
 data (wp(i),i= 481, 520)/
$.3005340E-03,.1912855E-03,.8217779E-04,.7844110E-02,.7843627E-02,
$.7842662E-02,.7841214E-02,.7839284E-02,.7836871E-02,.7833976E-02,
$.7830599E-02,.7826741E-02,.7822400E-02,.7817579E-02,.7812276E-02,
$.7806493E-02,.7800229E-02,.7793485E-02,.7786262E-02,.7778560E-02,
$.7770379E-02,.7761720E-02,.7752583E-02,.7742970E-02,.7732880E-02,
$.7722314E-02,.7711273E-02,.7699757E-02,.7687768E-02,.7675306E-02,
$.7662371E-02,.7648965E-02,.7635088E-02,.7620742E-02,.7605926E-02,
$.7590643E-02,.7574892E-02,.7558676E-02,.7541994E-02,.7524848E-02/
 data (wp(i),i= 521, 560)/
$.7507240E-02,.7489169E-02,.7470638E-02,.7451646E-02,.7432197E-02,
$.7412290E-02,.7391927E-02,.7371109E-02,.7349838E-02,.7328114E-02,
$.7305939E-02,.7283315E-02,.7260243E-02,.7236724E-02,.7212760E-02,
$.7188352E-02,.7163501E-02,.7138210E-02,.7112480E-02,.7086312E-02,
$.7059708E-02,.7032669E-02,.7005198E-02,.6977296E-02,.6948964E-02,
$.6920205E-02,.6891020E-02,.6861412E-02,.6831380E-02,.6800929E-02,
$.6770059E-02,.6738773E-02,.6707072E-02,.6674958E-02,.6642433E-02,
$.6609500E-02,.6576160E-02,.6542416E-02,.6508269E-02,.6473721E-02/
 data (wp(i),i= 561, 600)/
$.6438775E-02,.6403433E-02,.6367697E-02,.6331569E-02,.6295052E-02,
$.6258147E-02,.6220857E-02,.6183184E-02,.6145131E-02,.6106700E-02,
$.6067893E-02,.6028713E-02,.5989161E-02,.5949241E-02,.5908956E-02,
$.5868306E-02,.5827296E-02,.5785926E-02,.5744201E-02,.5702123E-02,
$.5659693E-02,.5616916E-02,.5573792E-02,.5530326E-02,.5486520E-02,
$.5442376E-02,.5397897E-02,.5353085E-02,.5307945E-02,.5262478E-02,
$.5216687E-02,.5170575E-02,.5124145E-02,.5077400E-02,.5030342E-02,
$.4982975E-02,.4935301E-02,.4887323E-02,.4839045E-02,.4790469E-02/
 data (wp(i),i= 601, 640)/
$.4741598E-02,.4692436E-02,.4642984E-02,.4593247E-02,.4543228E-02,
$.4492929E-02,.4442353E-02,.4391504E-02,.4340385E-02,.4288999E-02,
$.4237349E-02,.4185438E-02,.4133270E-02,.4080847E-02,.4028173E-02,
$.3975251E-02,.3922085E-02,.3868678E-02,.3815032E-02,.3761152E-02,
$.3707040E-02,.3652700E-02,.3598135E-02,.3543349E-02,.3488345E-02,
$.3433126E-02,.3377697E-02,.3322059E-02,.3266217E-02,.3210173E-02,
$.3153933E-02,.3097498E-02,.3040873E-02,.2984060E-02,.2927064E-02,
$.2869888E-02,.2812535E-02,.2755009E-02,.2697314E-02,.2639453E-02/
 data (wp(i),i= 641, 680)/
$.2581429E-02,.2523247E-02,.2464909E-02,.2406419E-02,.2347782E-02,
$.2289000E-02,.2230077E-02,.2171017E-02,.2111823E-02,.2052500E-02,
$.1993050E-02,.1933477E-02,.1873786E-02,.1813979E-02,.1754061E-02,
$.1694034E-02,.1633904E-02,.1573673E-02,.1513345E-02,.1452924E-02,
$.1392413E-02,.1331817E-02,.1271139E-02,.1210383E-02,.1149552E-02,
$.1088651E-02,.1027682E-02,.9666507E-03,.9055595E-03,.8444126E-03,
$.7832138E-03,.7219667E-03,.6606753E-03,.5993432E-03,.5379742E-03,
$.4765722E-03,.4151409E-03,.3536841E-03,.2922057E-03,.2307099E-03/
 data (wp(i),i= 681, 720)/
$.1692014E-03,.1076904E-03,.4626372E-04,.5231608E-02,.5231465E-02,
```

```
$.5231179E-02,.5230749E-02,.5230177E-02,.5229461E-02,.5228602E-02,
$.5227600E-02,.5226454E-02,.5225166E-02,.5223735E-02,.5222161E-02,
$.5220444E-02,.5218584E-02,.5216581E-02,.5214435E-02,.5212147E-02,
$.5209716E-02,.5207142E-02,.5204426E-02,.5201567E-02,.5198567E-02,
$.5195423E-02,.5192138E-02,.5188710E-02,.5185141E-02,.5181429E-02,
$.5177576E-02,.5173581E-02,.5169445E-02,.5165167E-02,.5160747E-02,
$.5156186E-02,.5151485E-02,.5146642E-02,.5141658E-02,.5136534E-02/
 data (wp(i),i= 721, 760)/
$.5131269E-02,.5125863E-02,.5120318E-02,.5114632E-02,.5108806E-02,
$.5102840E-02,.5096735E-02,.5090490E-02,.5084106E-02,.5077583E-02,
$.5070920E-02,.5064119E-02,.5057180E-02,.5050102E-02,.5042885E-02,
$.5035531E-02,.5028039E-02,.5020409E-02,.5012642E-02,.5004738E-02,
$.4996696E-02,.4988518E-02,.4980203E-02,.4971752E-02,.4963165E-02,
$.4954443E-02,.4945584E-02,.4936590E-02,.4927461E-02,.4918197E-02,
$.4908799E-02,.4899266E-02,.4889599E-02,.4879799E-02,.4869864E-02,
$.4859797E-02,.4849596E-02,.4839263E-02,.4828797E-02,.4818199E-02/
 data (wp(i),i= 761, 800)/
$.4807470E-02,.4796608E-02,.4785616E-02,.4774492E-02,.4763238E-02,
$.4751853E-02,.4740338E-02,.4728694E-02,.4716920E-02,.4705017E-02,
$.4692985E-02,.4680825E-02,.4668537E-02,.4656121E-02,.4643577E-02,
$.4630907E-02,.4618109E-02,.4605185E-02,.4592136E-02,.4578960E-02,
$.4565659E-02,.4552233E-02,.4538683E-02,.4525008E-02,.4511210E-02,
$.4497288E-02,.4483243E-02,.4469075E-02,.4454785E-02,.4440373E-02,
$.4425840E-02,.4411185E-02,.4396410E-02,.4381514E-02,.4366498E-02,
$.4351363E-02,.4336109E-02,.4320736E-02,.4305245E-02,.4289636E-02/
 data (wp(i),i= 801, 840)/
$.4273910E-02,.4258066E-02,.4242106E-02,.4226030E-02,.4209839E-02,
$.4193532E-02,.4177110E-02,.4160574E-02,.4143924E-02,.4127161E-02,
$.4110284E-02,.4093296E-02,.4076195E-02,.4058982E-02,.4041659E-02,
$.4024225E-02,.4006681E-02,.3989027E-02,.3971264E-02,.3953392E-02,
$.3935412E-02,.3917324E-02,.3899129E-02,.3880828E-02,.3862420E-02,
$.3843906E-02,.3825288E-02,.3806564E-02,.3787737E-02,.3768805E-02,
$.3749771E-02,.3730634E-02,.3711395E-02,.3692054E-02,.3672612E-02,
$.3653070E-02,.3633427E-02,.3613685E-02,.3593845E-02,.3573906E-02/
 data (wp(i),i= 841, 880)/
$.3553869E-02,.3533735E-02,.3513504E-02,.3493177E-02,.3472754E-02,
$.3452237E-02,.3431624E-02,.3410918E-02,.3390119E-02,.3369227E-02,
$.3348242E-02,.3327166E-02,.3305999E-02,.3284741E-02,.3263394E-02,
$.3241957E-02,.3220431E-02,.3198818E-02,.3177116E-02,.3155328E-02,
$.3133454E-02,.3111493E-02,.3089448E-02,.3067318E-02,.3045104E-02,
$.3022806E-02,.3000426E-02,.2977964E-02,.2955420E-02,.2932796E-02,
$.2910091E-02,.2887306E-02,.2864443E-02,.2841501E-02,.2818481E-02,
$.2795384E-02,.2772211E-02,.2748961E-02,.2725637E-02,.2702238E-02/
 data (wp(i),i= 881, 920)/
$.2678765E-02,.2655218E-02,.2631599E-02,.2607908E-02,.2584146E-02,
$.2560312E-02,.2536409E-02,.2512437E-02,.2488395E-02,.2464286E-02,
$.2440109E-02,.2415865E-02,.2391555E-02,.2367179E-02,.2342739E-02,
$.2318235E-02,.2293667E-02,.2269037E-02,.2244344E-02,.2219590E-02,
$.2194775E-02,.2169901E-02,.2144966E-02,.2119973E-02,.2094922E-02,
$.2069814E-02,.2044649E-02,.2019428E-02,.1994152E-02,.1968821E-02,
$.1943437E-02,.1917999E-02,.1892508E-02,.1866966E-02,.1841373E-02,
$.1815729E-02,.1790036E-02,.1764294E-02,.1738503E-02,.1712665E-02/
 data (wp(i),i= 921, 960)/
$.1686780E-02,.1660848E-02,.1634872E-02,.1608850E-02,.1582785E-02,
$.1556676E-02,.1530525E-02,.1504331E-02,.1478097E-02,.1451822E-02,
$.1425507E-02,.1399154E-02,.1372762E-02,.1346332E-02,.1319866E-02,
$.1293363E-02,.1266825E-02,.1240253E-02,.1213646E-02,.1187006E-02,
```

```
$.1160334E-02,.1133630E-02,.1106895E-02,.1080130E-02,.1053335E-02,
$.1026511E-02,.9996593E-03,.9727801E-03,.9458743E-03,.9189426E-03,
$.8919858E-03,.8650045E-03,.8379996E-03,.8109717E-03,.7839217E-03,
$.7568502E-03,.7297579E-03,.7026457E-03,.6755143E-03,.6483644E-03/
 data (wp(i),i= 961,1000)/
$.6211967E-03,.5940120E-03,.5668111E-03,.5395947E-03,.5123635E-03,
$.4851182E-03,.4578597E-03,.4305887E-03,.4033058E-03,.3760120E-03,
$.3487078E-03,.3213941E-03,.2940716E-03,.2667411E-03,.2394033E-03,
$.2120589E-03,.1847087E-03,.1573535E-03,.1299941E-03,.1026314E-03,
$.7526651E-04,.4790311E-04,.2057885E-04,.3924530E-02,.3924469E-02,
$.3924348E-02,.3924167E-02,.3923925E-02,.3923623E-02,.3923260E-02,
$.3922837E-02,.3922354E-02,.3921810E-02,.3921206E-02,.3920541E-02,
$.3919816E-02,.3919030E-02,.3918185E-02,.3917278E-02,.3916312E-02/
 data (wp(i),i=1001,1040)/
$.3915285E-02,.3914198E-02,.3913051E-02,.3911843E-02,.3910575E-02,
$.3909247E-02,.3907858E-02,.3906410E-02,.3904901E-02,.3903332E-02,
$.3901703E-02,.3900014E-02,.3898265E-02,.3896456E-02,.3894587E-02,
$.3892658E-02,.3890668E-02,.3888619E-02,.3886510E-02,.3884342E-02,
$.3882113E-02,.3879825E-02,.3877476E-02,.3875068E-02,.3872601E-02,
$.3870074E-02,.3867487E-02,.3864840E-02,.3862134E-02,.3859369E-02,
$.3856544E-02,.3853660E-02,.3850716E-02,.3847713E-02,.3844651E-02,
$.3841530E-02,.3838349E-02,.3835109E-02,.3831811E-02,.3828453E-02/
 data (wp(i),i=1041,1080)/
$.3825036E-02,.3821561E-02,.3818026E-02,.3814433E-02,.3810781E-02,
$.3807070E-02,.3803300E-02,.3799473E-02,.3795586E-02,.3791641E-02,
$.3787638E-02,.3783576E-02,.3779456E-02,.3775278E-02,.3771042E-02,
$.3766747E-02,.3762395E-02,.3757984E-02,.3753516E-02,.3748990E-02,
$.3744406E-02,.3739765E-02,.3735066E-02,.3730309E-02,.3725495E-02,
$.3720624E-02,.3715695E-02,.3710709E-02,.3705666E-02,.3700566E-02,
$.3695408E-02,.3690194E-02,.3684923E-02,.3679596E-02,.3674211E-02,
$.3668770E-02,.3663273E-02,.3657719E-02,.3652109E-02,.3646442E-02/
 data (wp(i),i=1081,1120)/
$.3640720E-02,.3634941E-02,.3629106E-02,.3623216E-02,.3617269E-02,
$.3611267E-02,.3605209E-02,.3599096E-02,.3592927E-02,.3586703E-02,
$.3580424E-02,.3574090E-02,.3567700E-02,.3561256E-02,.3554757E-02,
$.3548203E-02,.3541594E-02,.3534931E-02,.3528213E-02,.3521441E-02,
$.3514615E-02,.3507734E-02,.3500800E-02,.3493812E-02,.3486770E-02,
$.3479674E-02,.3472524E-02,.3465321E-02,.3458065E-02,.3450756E-02,
$.3443393E-02,.3435977E-02,.3428508E-02,.3420987E-02,.3413413E-02,
$.3405786E-02,.3398107E-02,.3390375E-02,.3382592E-02,.3374756E-02/
 data (wp(i),i=1121,1160)/
$.3366868E-02,.3358928E-02,.3350937E-02,.3342894E-02,.3334800E-02,
$.3326654E-02,.3318457E-02,.3310208E-02,.3301909E-02,.3293559E-02,
$.3285158E-02,.3276707E-02,.3268205E-02,.3259653E-02,.3251051E-02,
$.3242398E-02,.3233696E-02,.3224944E-02,.3216142E-02,.3207290E-02,
$.3198390E-02,.3189440E-02,.3180440E-02,.3171392E-02,.3162295E-02,
$.3153149E-02,.3143955E-02,.3134712E-02,.3125421E-02,.3116082E-02,
$.3106695E-02,.3097260E-02,.3087778E-02,.3078247E-02,.3068670E-02,
$.3059045E-02,.3049373E-02,.3039654E-02,.3029888E-02,.3020075E-02/
 data (wp(i),i=1161,1200)/
$.3010217E-02,.3000311E-02,.2990360E-02,.2980362E-02,.2970318E-02,
$.2960229E-02,.2950094E-02,.2939914E-02,.2929688E-02,.2919418E-02,
$.2909102E-02,.2898742E-02,.2888336E-02,.2877887E-02,.2867393E-02,
$.2856855E-02,.2846273E-02,.2835647E-02,.2824977E-02,.2814264E-02,
$.2803508E-02,.2792708E-02,.2781865E-02,.2770980E-02,.2760052E-02,
$.2749081E-02,.2738068E-02,.2727013E-02,.2715915E-02,.2704776E-02,
$.2693596E-02,.2682374E-02,.2671110E-02,.2659805E-02,.2648460E-02,
```

```
$.2637073E-02,.2625646E-02,.2614179E-02,.2602671E-02,.2591123E-02/
 data (wp(i),i=1201,1240)/
$.2579536E-02,.2567908E-02,.2556241E-02,.2544535E-02,.2532789E-02,
$.2521005E-02,.2509181E-02,.2497319E-02,.2485419E-02,.2473480E-02,
$.2461503E-02,.2449488E-02,.2437436E-02,.2425346E-02,.2413218E-02,
$.2401054E-02,.2388852E-02,.2376614E-02,.2364339E-02,.2352028E-02,
$.2339680E-02,.2327296E-02,.2314877E-02,.2302422E-02,.2289931E-02,
$.2277405E-02,.2264844E-02,.2252249E-02,.2239618E-02,.2226953E-02,
$.2214254E-02,.2201520E-02,.2188753E-02,.2175952E-02,.2163117E-02,
$.2150249E-02,.2137349E-02,.2124415E-02,.2111448E-02,.2098449E-02/
 data (wp(i),i=1241,1280)/
$.2085417E-02,.2072354E-02,.2059258E-02,.2046131E-02,.2032972E-02,
$.2019782E-02,.2006561E-02,.1993309E-02,.1980026E-02,.1966713E-02,
$.1953370E-02,.1939996E-02,.1926592E-02,.1913159E-02,.1899697E-02,
$.1886205E-02,.1872684E-02,.1859134E-02,.1845555E-02,.1831949E-02,
$.1818314E-02,.1804650E-02,.1790960E-02,.1777241E-02,.1763495E-02,
$.1749722E-02,.1735922E-02,.1722096E-02,.1708242E-02,.1694363E-02,
$.1680457E-02,.1666526E-02,.1652569E-02,.1638586E-02,.1624578E-02,
$.1610545E-02,.1596488E-02,.1582405E-02,.1568299E-02,.1554168E-02/
 data (wp(i),i=1281,1320)/
$.1540013E-02,.1525835E-02,.1511633E-02,.1497407E-02,.1483159E-02,
$.1468888E-02,.1454594E-02,.1440278E-02,.1425940E-02,.1411579E-02,
$.1397197E-02,.1382794E-02,.1368369E-02,.1353923E-02,.1339456E-02,
$.1324969E-02,.1310461E-02,.1295933E-02,.1281385E-02,.1266817E-02,
$.1252230E-02,.1237623E-02,.1222998E-02,.1208353E-02,.1193690E-02,
$.1179009E-02,.1164309E-02,.1149592E-02,.1134857E-02,.1120104E-02,
$.1105334E-02,.1090547E-02,.1075743E-02,.1060923E-02,.1046086E-02,
$.1031234E-02,.1016365E-02,.1001481E-02,.9865808E-03,.9716658E-03/
 data (wp(i),i=1321,1360)/
$.9567359E-03,.9417913E-03,.9268321E-03,.9118587E-03,.8968712E-03,
$.8818700E-03,.8668551E-03,.8518269E-03,.8367855E-03,.8217313E-03,
$.8066644E-03,.7915851E-03,.7764936E-03,.7613902E-03,.7462750E-03,
$.7311483E-03,.7160104E-03,.7008614E-03,.6857017E-03,.6705313E-03,
$.6553507E-03,.6401600E-03,.6249593E-03,.6097491E-03,.5945295E-03,
$.5793007E-03,.5640630E-03,.5488166E-03,.5335618E-03,.5182987E-03,
$.5030277E-03,.4877489E-03,.4724626E-03,.4571690E-03,.4418684E-03,
$.4265610E-03,.4112470E-03,.3959267E-03,.3806003E-03,.3652680E-03/
 data (wp(i),i=1361,1400)/
$.3499301E-03,.3345867E-03,.3192383E-03,.3038849E-03,.2885269E-03,
$.2731644E-03,.2577977E-03,.2424270E-03,.2270526E-03,.2116747E-03,
$.1962935E-03,.1809093E-03,.1655224E-03,.1501328E-03,.1347410E-03,
$.1193471E-03,.1039514E-03,.8855408E-04,.7315545E-04,.5775582E-04,
$.4235569E-04,.2695689E-04,.1158044E-04,.3140018E-02,.3139987E-02,
$.3139926E-02,.3139833E-02,.3139709E-02,.3139554E-02,.3139368E-02,
$.3139152E-02,.3138904E-02,.3138625E-02,.3138316E-02,.3137975E-02,
$.3137604E-02,.3137201E-02,.3136768E-02,.3136304E-02,.3135809E-02/
 data (wp(i),i=1401,1440)/
$.3135283E-02,.3134726E-02,.3134138E-02,.3133519E-02,.3132869E-02,
$.3132189E-02,.3131477E-02,.3130735E-02,.3129962E-02,.3129158E-02,
$.3128323E-02,.3127457E-02,.3126560E-02,.3125633E-02,.3124675E-02,
$.3123686E-02,.3122666E-02,.3121615E-02,.3120534E-02,.3119422E-02,
$.3118279E-02,.3117105E-02,.3115901E-02,.3114666E-02,.3113400E-02,
$.3112103E-02,.3110776E-02,.3109418E-02,.3108029E-02,.3106610E-02,
$.3105160E-02,.3103680E-02,.3102169E-02,.3100627E-02,.3099055E-02,
$.3097452E-02,.3095819E-02,.3094155E-02,.3092461E-02,.3090736E-02/
 data (wp(i),i=1441,1480)/
$.3088981E-02,.3087195E-02,.3085379E-02,.3083532E-02,.3081655E-02,
```

```
$.3079748E-02,.3077810E-02,.3075842E-02,.3073843E-02,.3071815E-02,
$.3069756E-02,.3067666E-02,.3065547E-02,.3063397E-02,.3061217E-02,
$.3059007E-02,.3056766E-02,.3054496E-02,.3052195E-02,.3049865E-02,
$.3047504E-02,.3045113E-02,.3042692E-02,.3040242E-02,.3037761E-02,
$.3035250E-02,.3032709E-02,.3030139E-02,.3027538E-02,.3024908E-02,
$.3022248E-02,.3019558E-02,.3016838E-02,.3014089E-02,.3011310E-02,
$.3008501E-02,.3005662E-02,.3002794E-02,.2999896E-02,.2996969E-02/
 data (wp(i),i=1481,1520)/
$.2994012E-02,.2991026E-02,.2988010E-02,.2984965E-02,.2981890E-02,
$.2978786E-02,.2975652E-02,.2972489E-02,.2969297E-02,.2966075E-02,
$.2962825E-02,.2959545E-02,.2956236E-02,.2952897E-02,.2949530E-02,
$.2946134E-02,.2942708E-02,.2939254E-02,.2935770E-02,.2932258E-02,
$.2928716E-02,.2925146E-02,.2921547E-02,.2917919E-02,.2914262E-02,
$.2910577E-02,.2906863E-02,.2903120E-02,.2899349E-02,.2895549E-02,
$.2891720E-02,.2887863E-02,.2883978E-02,.2880064E-02,.2876122E-02,
$.2872151E-02,.2868152E-02,.2864125E-02,.2860069E-02,.2855985E-02/
 data (wp(i),i=1521,1560)/
$.2851873E-02,.2847734E-02,.2843565E-02,.2839369E-02,.2835145E-02,
$.2830893E-02,.2826613E-02,.2822305E-02,.2817970E-02,.2813606E-02,
$.2809215E-02,.2804796E-02,.2800350E-02,.2795875E-02,.2791374E-02,
$.2786844E-02,.2782288E-02,.2777704E-02,.2773092E-02,.2768453E-02,
$.2763787E-02,.2759094E-02,.2754373E-02,.2749625E-02,.2744850E-02,
$.2740048E-02,.2735219E-02,.2730363E-02,.2725480E-02,.2720571E-02,
$.2715634E-02,.2710671E-02,.2705681E-02,.2700664E-02,.2695621E-02,
$.2690551E-02,.2685454E-02,.2680331E-02,.2675182E-02,.2670006E-02/
 data (wp(i),i=1561,1600)/
$.2664804E-02,.2659576E-02,.2654321E-02,.2649040E-02,.2643733E-02,
$.2638400E-02,.2633041E-02,.2627657E-02,.2622246E-02,.2616809E-02,
$.2611347E-02,.2605858E-02,.2600344E-02,.2594805E-02,.2589240E-02,
$.2583649E-02,.2578033E-02,.2572391E-02,.2566724E-02,.2561032E-02,
$.2555315E-02,.2549572E-02,.2543804E-02,.2538011E-02,.2532193E-02,
$.2526350E-02,.2520483E-02,.2514590E-02,.2508672E-02,.2502730E-02,
$.2496763E-02,.2490772E-02,.2484756E-02,.2478715E-02,.2472650E-02,
$.2466561E-02,.2460447E-02,.2454309E-02,.2448147E-02,.2441961E-02/
 data (wp(i),i=1601,1640)/
$.2435751E-02,.2429516E-02,.2423258E-02,.2416976E-02,.2410670E-02,
$.2404340E-02,.2397986E-02,.2391609E-02,.2385209E-02,.2378784E-02,
$.2372337E-02,.2365866E-02,.2359371E-02,.2352853E-02,.2346312E-02,
$.2339748E-02,.2333161E-02,.2326551E-02,.2319918E-02,.2313262E-02,
$.2306584E-02,.2299882E-02,.2293158E-02,.2286411E-02,.2279642E-02,
$.2272850E-02,.2266036E-02,.2259200E-02,.2252341E-02,.2245460E-02,
$.2238557E-02,.2231631E-02,.2224684E-02,.2217715E-02,.2210724E-02,
$.2203711E-02,.2196677E-02,.2189620E-02,.2182543E-02,.2175443E-02/
 data (wp(i),i=1641,1680)/
$.2168323E-02,.2161180E-02,.2154017E-02,.2146832E-02,.2139626E-02,
$.2132400E-02,.2125152E-02,.2117883E-02,.2110593E-02,.2103282E-02,
$.2095951E-02,.2088599E-02,.2081226E-02,.2073833E-02,.2066420E-02,
$.2058986E-02,.2051531E-02,.2044057E-02,.2036562E-02,.2029047E-02,
$.2021513E-02,.2013958E-02,.2006384E-02,.1998789E-02,.1991175E-02,
$.1983542E-02,.1975888E-02,.1968216E-02,.1960524E-02,.1952812E-02,
$.1945082E-02,.1937332E-02,.1929563E-02,.1921775E-02,.1913968E-02,
$.1906142E-02,.1898298E-02,.1890434E-02,.1882552E-02,.1874652E-02/
 data (wp(i),i=1681,1720)/
$.1866733E-02,.1858795E-02,.1850840E-02,.1842866E-02,.1834874E-02,
$.1826863E-02,.1818835E-02,.1810789E-02,.1802725E-02,.1794643E-02,
$.1786544E-02,.1778427E-02,.1770292E-02,.1762140E-02,.1753970E-02,
$.1745784E-02,.1737580E-02,.1729359E-02,.1721120E-02,.1712865E-02,
```

```
$.1704593E-02,.1696304E-02,.1687999E-02,.1679677E-02,.1671338E-02,
$.1662983E-02,.1654611E-02,.1646223E-02,.1637819E-02,.1629399E-02,
$.1620962E-02,.1612510E-02,.1604042E-02,.1595557E-02,.1587058E-02,
$.1578542E-02,.1570011E-02,.1561465E-02,.1552903E-02,.1544325E-02/
 data (wp(i),i=1721,1760)/
$.1535733E-02,.1527125E-02,.1518503E-02,.1509865E-02,.1501213E-02,
$.1492545E-02,.1483863E-02,.1475167E-02,.1466456E-02,.1457730E-02,
$.1448990E-02,.1440236E-02,.1431467E-02,.1422684E-02,.1413888E-02,
$.1405077E-02,.1396253E-02,.1387414E-02,.1378563E-02,.1369697E-02,
$.1360818E-02,.1351926E-02,.1343020E-02,.1334101E-02,.1325169E-02,
$.1316224E-02,.1307265E-02,.1298294E-02,.1289310E-02,.1280314E-02,
$.1271305E-02,.1262283E-02,.1253249E-02,.1244202E-02,.1235143E-02,
$.1226072E-02,.1216989E-02,.1207894E-02,.1198787E-02,.1189668E-02/
 data (wp(i),i=1761,1800)/
$.1180538E-02,.1171396E-02,.1162242E-02,.1153077E-02,.1143900E-02,
$.1134712E-02,.1125513E-02,.1116303E-02,.1107082E-02,.1097850E-02,
$.1088607E-02,.1079354E-02,.1070089E-02,.1060815E-02,.1051529E-02,
$.1042234E-02,.1032928E-02,.1023612E-02,.1014286E-02,.1004950E-02,
$.9956034E-03,.9862475E-03,.9768819E-03,.9675067E-03,.9581219E-03,
$.9487276E-03,.9393240E-03,.9299112E-03,.9204892E-03,.9110581E-03,
$.9016180E-03,.8921690E-03,.8827112E-03,.8732448E-03,.8637697E-03,
$.8542861E-03,.8447941E-03,.8352937E-03,.8257851E-03,.8162684E-03/
 data (wp(i),i=1801,1840)/
$.8067436E-03,.7972109E-03,.7876703E-03,.7781220E-03,.7685659E-03,
$.7590023E-03,.7494312E-03,.7398528E-03,.7302670E-03,.7206740E-03,
$.7110739E-03,.7014668E-03,.6918528E-03,.6822320E-03,.6726045E-03,
$.6629703E-03,.6533295E-03,.6436824E-03,.6340289E-03,.6243691E-03,
$.6147032E-03,.6050312E-03,.5953533E-03,.5856694E-03,.5759799E-03,
$.5662846E-03,.5565837E-03,.5468774E-03,.5371657E-03,.5274486E-03,
$.5177264E-03,.5079991E-03,.4982667E-03,.4885295E-03,.4787874E-03,
$.4690406E-03,.4592892E-03,.4495332E-03,.4397729E-03,.4300082E-03/
 data (wp(i),i=1841,1880)/
$.4202392E-03,.4104661E-03,.4006890E-03,.3909079E-03,.3811229E-03,
$.3713342E-03,.3615418E-03,.3517459E-03,.3419465E-03,.3321437E-03,
$.3223377E-03,.3125285E-03,.3027162E-03,.2929009E-03,.2830827E-03,
$.2732617E-03,.2634380E-03,.2536118E-03,.2437830E-03,.2339519E-03,
$.2241184E-03,.2142827E-03,.2044449E-03,.1946051E-03,.1847634E-03,
$.1749198E-03,.1650745E-03,.1552276E-03,.1453792E-03,.1355293E-03,
$.1256781E-03,.1158257E-03,.1059721E-03,.9611747E-04,.8626190E-04,
$.7640548E-04,.6654832E-04,.5669051E-04,.4683217E-04,.3697344E-04/
 data (wp(i),i=1881,1883)/
$.2711461E-04,.1725677E-04,.7413338E-05/

 CsTH = cos(TwoTH0 * 0.5/RAD)
 if (abs(CsTH) .lt. 1.0e-15) CsTH = 1.0e-15
 TTH = sin(TwoTH0 * 0.5/RAD)/CsTH
 CsTwoTH = cos(TwoTH0/RAD)
 SnTwoTH = sin(TwoTH0/RAD)
 ApB = S_L + D_L
 AmB = S_L - D_L
 ApB2 = ApB**2
 if (((S_L .ne. 0.0) .or. (D_L .ne. 0.0)) .and. Use_Asym) then
   tmp = sqrt(1.0 + AmB**2)*CsTwoTH
   if (abs(tmp) .gt. 1.0) then
     Einfl = acos(CsTwoTH)*RAD
   else
     Einfl = acos(tmp)*RAD
```

```
        endif
        tmp2 = 1.0 + ApB2
        tmp = sqrt(tmp2 ) * CsTwoTH

c If S_L or D_L are zero, set Einfl = 2theta

        if ((S_L .eq. 0.0) .or. (D_L .eq. 0.0)) Einfl = TwoTH0
        if (abs(tmp) .le. 1.0) then
          Emin = acos(tmp) * RAD
          tmp1 = tmp2 * (1.0 - tmp2 * CsTwoTH**2)
        else
          tmp1 = 0.0
          if (tmp .gt. 0.0) then
            Emin = 0.0
          else
            Emin = 180.0
          endif
        endif
        if ((tmp1 .gt. 0.0) .and. (abs(tmp) .le. 1.0)) then
          dEmindA = -ApB * CsTwoTH/sqrt(tmp1)
        else
          dEmindA = 0.0
        endif
        ArrayNum = 1
        K = 400.0 * (TwoTH0 - Emin)   ! Calculate number of terms needed
        do while ((ArrayNum .lt. 14) .and. (K .gt. NTERMS(ArrayNum)))
          ArrayNum = ArrayNum + 1
        enddo
        NGT = nterms(ArrayNum)              ! Save number of terms
        ngt2 = ngt / 2
c Clear terms needed for summations
        sumWG = 0.0
        sumWRG = 0.0
        sumWdGdA = 0.0
        sumWRdGdA = 0.0
        sumWdGdB = 0.0
        sumWRdGdB = 0.0
        sumWGdRd2t = 0.0
        sumWGdRdG = 0.0
        sumWGdRdE = 0.0
        sumWGdRdA = 0.0
        sumWGdRdB = 0.0
c Compute the convolution integral
        it = fstterm(arraynum)-ngt2
        do K = ngt2 , NGT
          delta = Emin + (TwoTH0 - Emin) * xp(K + it)
          dDeltadA = (1.0 - xp(k+it) ) * dEmindA
          sinDELTA = sin(Delta/RAD)
          cosDELTA = cos(Delta/RAD)
          if (abs(cosDELTA) .lt. 1.0e-15) cosDELTA = 1.0e-15
          RcosDELTA = 1.0 / cosDELTA
          tanDELTA = tan(Delta/RAD)
          tmp = cosDELTA**2 - CsTwoTH**2
          if (tmp .gt. 0.0) then
            tmp1 = sqrt(tmp)
            F = abs(CsTwoTH) / tmp1
            dFdA = cosDELTA * CsTwoTH * sinDELTA * dDELTAdA
```

```
     1         / (tmp1 * tmp1 * tmp1)
          else
            F = 0.0
            dFdA = 0.0
          endif
c  calculate G(Delta,2theta) , FCJ eq. 7a and 7b
          if ( abs(Delta - Emin) .gt. abs(Einfl - Emin)) then
            if (S_L .gt. D_L) then
!
! N.B. this is the only place where d()/dA <> d()/dB
!
              G = 2.0 * D_L * F * RcosDELTA
              dGdA = 2.0 * D_L * RcosDELTA * (dFdA +
     1                F*tanDELTA*dDELTAdA)
              dGdB = dGdA + 2.0 * F * RcosDELTA
            else
              G = 2.0 * S_L * F * RcosDELTA
              dGdB = 2.0 * S_L * RcosDELTA
     1                  *(dFdA + F * tanDELTA * dDELTAdA)
              dGdA = dGdB + 2.0 * F * RcosDELTA
            endif
          else
            G = (-1.0 + ApB * F) * RcosDELTA
            dGdA = RcosDELTA * (F - tanDELTA * dDELTAdA + ApB * dFdA
     1               + ApB * F * tanDELTA * dDELTAdA)
            dGdB = dGdA
          endif
          tmp = PsVoigt(TwoTh-DELTA+TwoTH0,TwoTH0,eta,Gamma,dPRdT
     1          ,dPRdG,dPRdE)
          sumWG = sumWG + wp(k+it) * G
          sumWRG = sumWRG + wp(k+it) * G * tmp
          sumWdGdA = sumWdGdA + wp(k+it) * dGdA
          sumWdGdB = sumWdGdB + wp(k+it) * dGdB
          sumWRdGdA = sumWRdGdA + wp(k+it) * dGdA * tmp
          sumWRdGdB = sumWRdGdB + wp(k+it) * dGdB * tmp
          sumWGdRd2t = sumWGdRd2t + wp(k+it) * G * dPRdT
          sumWGdRdG = sumWGdRdG + wp(k+it) * G * dPRdG
          sumWGdRdE = sumWGdRdE + wp(k+it) * G * dPRdE
          sumWGdRdA = sumWGdRdA + wp(k+it) * G * dPRdT * dDELTAdA * RAD
        enddo
        if (sumWG .eq. 0.0) sumWG = 1.0
        Profval = sumWRG / sumWG
        dPRdT = sumWGdRd2t/ sumWG
        dPRdG = sumWGdRdG / sumWG
        dPRdE = sumWGdRdE / sumWG
        dPRdS = (sumWRdGdA + sumWGdRdA/RAD) / sumWG - sumWRG *
     1          sumWdGdA/RAD/sumWG**2
        dPRdD = (sumWRdGdB + sumWGdRdA/RAD) / sumWG - sumWRG *
     1          sumWdGdB/RAD/sumWG**2
      else   ! here for no asymmetry }
        tmp = PsVoigt(TwoTH,TwoTH0,eta,Gamma,dPRdT,dPRdG,dPRdE)
        Profval = tmp
        dPRdS = 0.0
        dPRdD = 0.0
      endif
      return
      end
```

```fortran
      real*4 function Gauss(Pos , Pos0 , Gamma , dGdT , dGdG )

c  Return value of Gaussian at 'Pos' for peak at 'Pos0' and 'Gamma'.
c  dGdT is derivative of G wrt Pos0.
c  dGdG is derivative of G wrt Gamma.

      implicit none
      real*4 Pos , Pos0 , Gamma , dGdT , dGdG
      real*4   c / 1.6651092/
      real*4  cg / 0.939437279/
      real*4  delp , temp

      delp = Pos - Pos0
      if (abs(delp)/Gamma .gt. 6) then
        Gauss = 0.0
        dGdT = 0.0
        dGdG = 0.0
      else
        temp = cg * exp(-(delp * c /Gamma)**2)/Gamma
        Gauss = temp
        dGdG = temp * ( -1.0 + 2.0 * (delp * c/Gamma)**2) / Gamma
        dGdT = 2.0 * c**2 * delp * temp/Gamma**2
      endif
      return
      end

      real*4 function Lorentz(Pos , Pos0 , Gamma , dLdT , dLdG )

c  Return value of Lorentzian at 'Pos' for peak at 'Pos0' and 'Gamma'.
c  dLdT is derivative of L wrt Pos0.
c  dLdG is derivative of L wrt Gamma.

      implicit none
      real*4 Pos , Pos0 , Gamma , dLdT , dLdG
      real*4 cl / 0.636619772/

      real*4  delp , denom

      delp = Pos - Pos0
      denom = 4.0 * delp**2 + Gamma**2
      Lorentz = cl * Gamma / denom
      dLdT = 8.0 * cl * Gamma * delp / denom**2
      dLdG = cl * (4.0 * delp**2 - Gamma**2) / denom**2
      return
      end

      real*4 function PsVoigt(TwoTH , TwoTH0 , eta , Gamma,
     1         dPRdT , dPRdG , dPRdE )
c
c   Returns value of Pseudo Voigt
c   Eta is the mixing coefficient between Gaussian and Lorentzian
c   Gamma is the FWHM
c   TwoTH is point at which to evaluate the profile
c   TwoTH0 is two theta value for peak
c   dPRdT is derivative of profile wrt TwoTH0
```

```
c   dPRdG is derivative of profile wrt Gamma
c   dPRdE is derivative of profile wrt Eta

      implicit none
      real*4 TwoTH , TwoTH0 , eta , Gamma
      real*4 dPRdT , dPRdG , dPRdE
      real*4  G,Gauss         ! Gaussian part
      real*4  L,Lorentz       ! Lorentzian part
      real*4 dGdT , dGdG , dLdT , dLdG

      G = Gauss(TwoTH , TwoTH0 , Gamma , dGdT , dGdG )
      L = Lorentz(TwoTH , TwoTH0 , Gamma , dLdT , dLdG )
      PsVoigt = Eta * L + (1.0 - Eta) * G
      dPRdT = Eta * dLdT + (1.0 - Eta) * dGdT
      dPRdG = Eta * dLdG + (1.0 - Eta) * dGdG
      dPRdE = L - G
      return
      end
```