```
!       HOW TO USE THE SUBROUTINE
!
!       NY = no. of meshes in the angular direction
!       NZ = no. of meshes in the axial direction
!
!       DO J=1,NY+1
!          VY(J)= r theta = distance of vertex J along surface
!                           in the angular direction
!       ENDDO
!       DO K=1,NZ+1
!          VZ(K)= z = distance of vertex K along the surface
!                     in the axial direction
!       ENDDO
!       NBY=0  reflective boundary conditions in the y-direction
!          =1  periodic   boundary conditions in the y-direction
!       NBZ=0  reflective boundary conditions in the y-direction
!          =1  periodic   boundary conditions in the y-direction
!
!       NR=1              !RANDOM NUMBER SET (could choose any +ve integer)
!       SS=-2.0           !POWER SPECTRUM EXPONENT
!       XLMIN=0.01        !MINIMUM WAVELENGTH
!       XLMAX=0.2         !MAXIMUM WAVELENGTH
!       SD=0.00005        !STANDARD DEVIATION
!       CALL PERTINT(NR,SD,XLMIN,XLMAX,SS,ZETA,              &
!      &                   NY,NZ,NBY,NBZ,VY,VZ)
!
!       ZETA(J,K)= required perturbation, J=1,NY+1;K=1,NZ+1
!
       SUBROUTINE PERTINT(NR,SD,XLMIN,XLMAX,SS,ZETA,                       &
      &                    NY,NZ,NBY,NBZ,VY,VZ)
!
!      SUBROUTINE WRITTEN BY D.L.YOUNGS,AWE, ALDERNASTON, UK.
!      BRITISH CROWN COPYRIGHT 2008/MOD
!
       IMPLICIT NONE
!
       REAL*8 ZETA(NY+1,NZ+1),VY(NY+1),VZ(NZ+1),                            &
      &     AA(0:NY,0:NZ),BB(0:NY,0:NZ),                                    &
      &     CC(0:NY,0:NZ),DD(0:NY,0:NZ)
       REAL*8 COSJ(NY),SINJ(NY)
       REAL*8 PI,PI2,R,R1,R2,RA,RANF,RB,RC,RD,RR,S,SD,SS,WWY,WWZ,           &
      &     XL,XL1,XL2,XLMAX,XLMIN,XLMIN1,Y,YNN,Z,ZNN,ZZZ,XX,YY,            &
      &     YK1,YK2,ZK1,ZK2,XKMIN,XKMAX,FRC1,FRC2,FRC,FFF,WW
       INTEGER*4 J,K,IR,IRMAX,NY,NZ,NBY,NBZ,NNY,NNY1,NNZ,NNZ1,NR
!
!
!      PERTURB INTERFACE
!      NR=RANDOM NUMBER SET
```

```
!     VY(J),J=1,NY+1: MESH VERTICES IN Y-DIRECTION
!     VZ(K),K=1,NZ+1: MESH VERTICES IN Z-DIRECTION
!     ZETA(J,K); PERTURBATION AT Y=VY(J),Z=VZ(K)
!     NBY=0  REFLECTIVE BOUNDARY CONDITIONS IN Y-DIRECTION
!         =1  PERIODIC BOUNDARY CONDITIONS IN Y-DIRECTION
!     NBZ=0  REFLECTIVE BOUNDARY CONDITIONS IN Z-DIRECTIONS
!         =1  PERIODIC BOUNDARY CONDITIONS IN Z-DIRECTION
!
!     POWER SPECTRUM
!     P(KY,KZ) DKY.DKZ=K**(SS-1) DKY.DKZ     KMIN.LT.K.LT.KMAX
!     KMIN=2.0*PI/XLMAX
!     KMAX=2.0*PI/XLMIN
!     K=SQRT(KY**2+KZ**2)
!     SD**2=INTEGRAL P DKY.DKZ
!     RANF()=RANDOM NUMBER IN THE RANGE (0,1)
!
!
      PI=3.141592653590
      PI2=2.0*PI
!
!     CALCULATE RANDOM PERTURBATION
!
      IF(NBY.EQ.0) THEN
         WWY=2.0*(VY(NY+1)-VY(1))
         NNY1=NY
      ELSE
         WWY=(VY(NY+1)-VY(1))
         NNY1=NY/2
      ENDIF
      IF(NBZ.EQ.0) THEN
         WWZ=2.0*(VZ(NZ+1)-VZ(1))
         NNZ1=NZ
      ELSE
         WWZ=(VZ(NZ+1)-VZ(1))
         NNZ1=NZ/2
      ENDIF
      XLMIN1=MAX(XLMIN,1.0E-50)
      NNY1=MIN(INT(WWY/XLMIN1+0.5),NNY1)
      NNZ1=MIN(INT(WWZ/XLMIN1+0.5),NNZ1)
!
      WRITE(6,99)
 99   FORMAT('   *****SET INITIAL PERTURBATION*****')
      WRITE(6,98) SS,XLMIN,XLMAX,SD
 98   FORMAT('  S=',F8.4,' XLMIN=',F10.5,' XLMAX=',F10.5,                    &
     &        ' SD=',F10.5/)
      IRMAX=NR
      DO IR=1,IRMAX
         DO NNZ=0,NNZ1
```

```
            DO NNY=0,NNY1
!
               ZK2=(FLOAT(NNZ)+0.5)/WWZ
               YK2=(FLOAT(NNY)+0.5)/WWY
               ZK1=(MAX(FLOAT(NNZ)-0.5,0.0))/WWZ
               YK1=(MAX(FLOAT(NNY)-0.5,0.0))/WWY
               XKMIN=1.0/XLMAX
               XKMAX=1.0/XLMIN
               CALL FRACR(YK1,YK2,ZK1,ZK2,10,XKMIN,FRC1)
               CALL FRACR(YK1,YK2,ZK1,ZK2,10,XKMAX,FRC2)
               FRC=FRC2-FRC1
               ZNN=FLOAT(NNZ)
               YNN=FLOAT(NNY)
               XL=1.0/SQRT((YNN/WWY)**2+(ZNN/WWZ)**2+1.0E-50)
!
               IF(FRC.GE.1.0E-10) THEN
!
!     CHOOSE RA,RB,RC,RD FROM UNIT GAUSSIAN DISTRIBUTION LIMIT TO
!     RANGE -4.0 T0 +4.0
!
                  R1=RANF()
                  R2=RANF()
                  R=SQRT(-2.0*LOG(R1))*COS(2.0*PI*R2)
                  RR=ABS(R)
                  RA=R
                  IF(RR.GT.4.0) RA=4.0*R/RR
!
                  R1=RANF()
                  R2=RANF()
                  R=SQRT(-2.0*LOG(R1))*COS(2.0*PI*R2)
                  RR=ABS(R)
                  RB=R
                  IF(RR.GT.4.0) RB=4.0*R/RR
!
                  R1=RANF()
                  R2=RANF()
                  R=SQRT(-2.0*LOG(R1))*COS(2.0*PI*R2)
                  RR=ABS(R)
                  RC=R
                  IF(RR.GT.4.0) RC=4.0*R/RR
!
                  R1=RANF()
                  R2=RANF()
                  R=SQRT(-2.0*LOG(R1))*COS(2.0*PI*R2)
                  RR=ABS(R)
                  RD=R
                  IF(RR.GT.4.0) RD=4.0*R/RR
!
```

```
!     MODS FOR REFLECTIVE BOUNDARIES
!
                IF(NBY.EQ.0) THEN
                   RC=0.0
                   RD=0.0
                ENDIF
                IF(NBZ.EQ.0) THEN
                   RB=0.0
                   RD=0.0
                ENDIF
!
                ZZZ=(XLMIN/XL)**(0.5*(SS-1.0))*SQRT(FRC)
                AA(NNY,NNZ)=RA*ZZZ
                BB(NNY,NNZ)=RB*ZZZ
                CC(NNY,NNZ)=RC*ZZZ
                DD(NNY,NNZ)=RD*ZZZ
             ELSE
                AA(NNY,NNZ)=0.0
                BB(NNY,NNZ)=0.0
                CC(NNY,NNZ)=0.0
                DD(NNY,NNZ)=0.0
             ENDIF
          END DO
       END DO
!
!     ALLOW FOR HALVED POWER LEVELS AT EDGES OF THE K-SPACE MESH
!
          AA(0,0)=0.0
          BB(0,0)=0.0
          CC(0,0)=0.0
          DD(0,0)=0.0
          S=1.0/SQRT(2.0)
          IF(NBZ.EQ.0) S=0.5
          DO NNY=1,NNY1
             AA(NNY,0)=S*AA(NNY,0)
             BB(NNY,0)=0.0
             CC(NNY,0)=S*CC(NNY,0)
             DD(NNY,0)=0.0
          END DO
          S=1.0/SQRT(2.0)
          IF(NBY.EQ.0) S=0.5
          DO NNZ=1,NNZ1
             AA(0,NNZ)=S*AA(0,NNZ)
             BB(0,NNZ)=S*BB(0,NNZ)
             CC(0,NNZ)=0.0
             DD(0,NNZ)=0.0
          END DO
!
```

```
!      SET RANDOM PERTURBATION
!
          DO J=1,NY+1
             DO K=1,NZ+1
                ZETA(J,K)=0.0
             END DO
          END DO
!
          DO NNY=0,NNY1
             DO J=1,NY+1
                Y=PI2*(VY(J)-VY(1))/WWY
                COSJ(J)=COS(FLOAT(NNY)*Y)
                SINJ(J)=SIN(FLOAT(NNY)*Y)
             END DO
             DO NNZ=0,NNZ1
                DO K=1,NZ+1
                   Z=PI2*(VZ(K)-VZ(1))/WWZ
                   XX=AA(NNY,NNZ)*COS(FLOAT(NNZ)*Z)                    &
     &                  +BB(NNY,NNZ)*SIN(FLOAT(NNZ)*Z)
                   YY=CC(NNY,NNZ)*COS(FLOAT(NNZ)*Z)                    &
     &                  +DD(NNY,NNZ)*SIN(FLOAT(NNZ)*Z)
                   DO J=1,NY
                      ZETA(J,K)=ZETA(J,K)+XX*COSJ(J)+YY*SINJ(J)
                   END DO
                END DO
             END DO
          END DO
!
!      SCALE TO REQUIRED STANDARD DEVIATION(SD) AND ADD
!      FIXED LONGWAVE PERTURBATION ZETA0
!
          S=0.0
          DO J=1,NY+1
             DO K=1,NZ+1
                WW=1.0
                IF(J.EQ.1.OR.J.EQ.NY+1) WW=0.5*WW
                IF(K.EQ.1.OR.K.EQ.NZ+1) WW=0.5*WW
                S=S+WW*ZETA(J,K)**2
             END DO
          END DO
          S=SQRT(S/FLOAT(NY*NZ))
          FFF=SD/S
          DO J=1,NY+1
             DO K=1,NZ+1
                ZETA(J,K)=ZETA(J,K)*FFF
             END DO
          END DO
       END DO
```

```
!
      RETURN
      END
      SUBROUTINE FRACR(X1,X2,Y1,Y2,N,R,FRC)
      IMPLICIT NONE
!
!     FRC=FRACTION OF BOX (X1,X2) x (Y1,Y2)
!     WITHIN CIRCLE OF RADIUS R
!
      REAL*8 X1,X2,Y1,Y2,R,FRC,R1,R2,R3,R4,RMAX,RMIN
      REAL*8 F,FF,DXX,DYY,XX,YY,RR
      INTEGER*4 I,J,N
      IF(X1.GE.X2.OR.Y1.GE.Y2.OR.R.LT.0.0) THEN
      WRITE(6,999)
  999 FORMAT('     ERROR IN FRAC')
      STOP
      ENDIF
!
      R1=SQRT(X1**2+Y1**2)
      R2=SQRT(X2**2+Y1**2)
      R3=SQRT(X1**2+Y2**2)
      R4=SQRT(X2**2+Y2**2)
      RMAX=MAX(R1,R2,R3,R4)
      RMIN=MIN(R1,R2,R3,R4)
      IF(R.GE.RMAX) THEN
         F=1.0
         ELSE
         IF(R.LE.RMIN) THEN
            F=0.0
            ELSE
            F=0.0
            FF=1.0/FLOAT(N*N)
            DXX=(X2-X1)/FLOAT(N)
            DYY=(Y2-Y1)/FLOAT(N)
            DO I=1,N
               DO J=1,N
                  XX=X1+(FLOAT(I)-0.5)*DXX
                  YY=Y1+(FLOAT(J)-0.5)*DYY
                  RR=SQRT(XX**2+YY**2)
                  IF(RR.LE.R) F=F+FF
               ENDDO
            ENDDO
         ENDIF
      ENDIF
      FRC=F
      RETURN
      END
```