```fortran
C#########################################################################C
c                  SUBROUNTINE PROGRAM                                    #
C                  SOLID BOUNDARY CONDITION BASED ON DES                  #
c                                 VERSION 1.0 (16/02/2011)                #
C                  AUTHORIZED BY ZHANG JINGXIN                            #
C                                 SHANGHAI JIAO TONG UNIVERSITY           #
C                                 SHANGHAI, CHINA                         #
C#########################################################################
    Subroutine DESSST
      Include './Include/OCERM_INF'
      Parameter(BETASTAR=0.09,FKAPA=0.41,CDES = 0.65)
    Parameter(SIGMAK1=0.85, SIGMAO1=0.5, BETA1=0.075, A1=0.31,
     &            GAMA1=BETA1/BETASTAR-SIGMAO1*FKAPA**2./Sqrt(BETASTAR))
    Parameter(SIGMAK2=1.0, SIGMAO2=0.856, BETA2=0.0828,
     &            GAMA2=BETA2/BETASTAR-SIGMAO2*FKAPA**2./Sqrt(BETASTAR))
    Dimension CROSSDIFFUSION(IJM,KB),F1(IJM,KB), F2(IJM,KB), S1(IJM,KB)
    Dimension VAR_EDGE(IJE,KB),VAR_UD(IJM,KB),GRADTKX(IJM,KB),
     &            GRADTKY(IJM,KB),GRADTKZ(IJM,KB),GRADTKDX(IJM,KB),
     &            GRADTKDY(IJM,KB),GRADTKDZ(IJM,KB)
      Dimension SMOOTH(IJM,KB), I_SMOOTH(IJM,KB),F3(IJM,KB)
```

$* \ \phi = F_1\phi_1 + (1-F_1)\phi_2$

ϕ is coefficient of SST k-ω model

```fortran
c=========================================================================c
c              transport equation of working variable                     c
c=========================================================================c
c-------------------------------------------------------------------------c
C                                Gradation of TKE                         c
c-------------------------------------------------------------------------c
```

$k \quad \frac{\partial k}{\partial x_i}$

```fortran
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,IL,IR,UP,YP,TKE_WALL)
!$OMP DO
      Do I = 1, IJM
       Do K = 1, KB
          VAR_UD(I,K) = 0.0
          SMOOTH(I,K) = 0.0
          I_SMOOTH(I,K) = 0
       Enddo
    Enddo
!$OMP END DO
      Do K = 1, KBM
!$OMP DO
         Do I = 1, IJE
            VAR_EDGE(I,K) = 0.0
```

EDGE

IL → IR

0 (initial)

```
      IL = 0
      IR = 0
      IL = INDEX_EDGE(I,K,1)
      IR = INDEX_EDGE(I,K,2)
      If(CFM(I) .EQ. 1.0) Then
        VAR_EDGE(I,K) = (TKE(IL,K) + TKE(IR,K)) / 2.
      Endif
      If(CFM(I) .EQ. 0.0 .OR. CFM(I) .EQ. -1.0) Then
         If(ISLIP .EQ. 1) Then
            If(IL .GT. 0) Then
               If(CCM(IL) .EQ. 1.0) VAR_EDGE(I,K) = TKE(IL,K)
            Endif
            If(IR .GT. 0) Then
               If(CCM(IR) .EQ. 1.0) VAR_EDGE(I,K) = TKE(IR,K)
            Endif
         Else
            If(IL .GT. 0) Then
               If(CCM(IL) .EQ. 1.0) Then
                 UP = Sqrt(UR(IL,K) ** 2. +
   *                  VR(IL,K) ** 2. + WR(IL,K) ** 2.)
                 YP = D2D(IL)    distance to solid walls
                 YSTAR =
   *             Dmax1(Sqrt(0.3)*Sqrt(TKE(IL,K))*YP/1.E-6,15.0)

                TKE_WALL = 0.41*UP*Sqrt(0.3)*Sqrt(TKE(IL,K))/
   *                 Dlog(9.81*YSTAR) / 0.3

                VAR_EDGE(I,K) = TKE_WALL
               Endif
            Endif
            If(IR .GT. 0) Then
               If(CCM(IR) .EQ. 1.0) Then
                 UP = Sqrt(UR(IR,K) ** 2. +
   *                  VR(IR,K) ** 2. + WR(IR,K) ** 2.)
                 YP = D2D(IR)
                 YSTAR =
   *             Dmax1(Sqrt(0.3)*Sqrt(TKE(IR,K))*YP/1.E-6,15.0)

                TKE_WALL = 0.41*UP*Sqrt(0.3)*Sqrt(TKE(IR,K))/
   *                 Dlog(9.81*YSTAR) / 0.3
```

```
                       VAR_EDGE(I,K) = TKE_WALL
                     Endif
                  Endif
               Endif
            Endif
            If(CFM(I) .EQ. -3.0) Then
               If(IL .GT. 0) Then
                  If(CCM(IL) .EQ. 1.0) VAR_EDGE(I,K) = TKE(IL,K)
               Endif
               If(IR .GT. 0) Then
                  If(CCM(IR) .EQ. 1.0) VAR_EDGE(I,K) = TKE(IR,K)
               Endif
            Endif
         Enddo
!$OMP END DO
      Enddo
!$OMP DO
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
          Do K = 2, KBM
             GRADTKZ(I,K) = (TKE(I,K-1) - TKE(I,K)) / DC(I) / DZZ(K-1)
```

$$\frac{\partial k}{\partial z} = \frac{\Delta k}{\Delta \sigma}\frac{1}{H+\eta}$$

水深 $\Delta\sigma$

```
            Enddo
            GRADTKZ(I,1) = 0.0
            GRADTKZ(I,KB) = TKE(I,KBM) / (0.5 * DZ(KBM) * DC(I))

C            VAR_UD(I,K) = (TKE(I,K-1) + TKE(I,K)) / 2.0
C         Enddo
C         If(KB .GE. 3) Then
C            VAR_UD(I,1) = 2. * TKE(I,1) - TKE(I,2)
C            VAR_UD(I,KB) = 0.0
C         Else
C            VAR_UD(I,1) = TKE(I,1)
C         Endif
       Endif
     Enddo
!$OMP END DO
!$OMP END PARALLEL
     Call GRAD_XY(VAR_EDGE,GRADTKX,GRADTKY)
C    Call GRAD_Z(VAR_UD,GRADTKZ)
C-----   Boundary conditions
----------------------------------------------------c
```

```
      If(NUMEBC .NE. 0) Then
         Do N = 1, NUMEBC
          ID = IEBC(N)
          IS = IEBCINX(N)
          Do K = 1, KBM
             GRADTKX(ID,K) = 0.0
             GRADTKY(ID,K) = 0.0
          Enddo
       Enddo
      Endif
C-----  astrotidal boundary condition
      If(NUMAST .NE. 0) Then
       Do N = 1, NUMAST
          ID = IABC(N)
          IS = IABCINX(N)
          Do K = 1, KBM
             GRADTKX(ID,K) = 0.0
             GRADTKY(ID,K) = 0.0
          Enddo
       Enddo
      Endif
c-----  discharge boundary condition
      If(NUMQBC .NE. 0) Then
        Call BCOND(3)
        Do N = 1, NUMQBC
           ID = IQBC(N)
           IS = IQBCINX(N)
           ISS = CELL_SIDE(ID,IS,1)
           Do K = 1, KBM
              GRADTKX(ID,K) = 0.0
              GRADTKY(ID,K) = 0.0
           Enddo
        Enddo
      Endif
C-----  velocity boundary condition
      If(NUMVBC .NE. 0) Then
        Do N = 1, NUMVBC
           ID = IVBC(N)
           IS = IVBCINX(N)
           Do K = 1, KBM
              GRADTKX(ID,K) = 0.0
```

```
                GRADTKY(ID,K) = 0.0
             Enddo
          Enddo
       Endif
c-----------------------------------------------------------------------------c
C                             Gradation of TDISS  ω  ∂ω/∂x_j                  c
c-----------------------------------------------------------------------------c

!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,ID_CELL,IL,IR,
!$OMP&           M, COS_WALL, SIN_WALL, TW_STAR, UW_STAR,
!$OMP&           RKS_PLUS,TDISS_PLUS, TDISS_WALL)
!$OMP DO
      Do I = 1, IJM
       Do K = 1, KB
          VAR_UD(I,K) = 0.0
       Enddo
    Enddo
!$OMP END DO
      Do K = 1, KBM
!$OMP DO
         Do I = 1, IJE
            VAR_EDGE(I,K) = 0.0
            IL = 0
            IR = 0
            IL = INDEX_EDGE(I,K,1)
            IR = INDEX_EDGE(I,K,2)
            If(CFM(I) .EQ. 1.0) Then
               VAR_EDGE(I,K) = (TDISS(IL,K) + TDISS(IR,K)) / 2.
            Endif
            If(CFM(I) .EQ. 0.0 .OR. CFM(I) .EQ. -1.0) Then
               If(ISLIP .EQ. 1) Then
                  If(IL .GT. 0) Then
                     If(CCM(IL) .EQ. 1.0) VAR_EDGE(I,K) = TDISS(IL,K)
                  Endif
                  If(IR .GT. 0) Then
                     If(CCM(IR) .EQ. 1.0) VAR_EDGE(I,K) = TDISS(IR,K)
                  Endif
               Else
                  If(IL .GT. 0) Then
                     If(CCM(IL) .EQ. 1.0)
     &               VAR_EDGE(I,K) = Sqrt(TKE(IL,K)) /
```

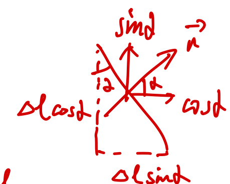
```fortran
     &                               Sqrt(0.3) / 0.41 / D2D(IL)
                  Endif
                  If(IR .GT.  0) Then
                     If(CCM(IR) .EQ.  1.0)
     &               VAR_EDGE(I,K) = Sqrt(TKE(IR,K)) /
     &                               Sqrt(0.3) / 0.41 / D2D(IR)
                  Endif
               Endif
            Endif
            If(CFM(I) .EQ.  -2.0) Then
               If(IL .GT.  0) Then
                  If(CCM(IL) .EQ.  1.0) THEN
C-------------------------BC of TDISS at wall(by WangJian)-----------------------
                     IF (IBC_TDISS_WALL == 1) THEN
                     VAR_EDGE(I,K) = 60.*UMOL/BETA1/(D2D(IL)**2.0)
                     ELSEIF (IBC_TDISS_WALL == 2) THEN

                     DO M = 1, CELL_POLYGEN(IL)
                       IF (CELL_SIDE(IL,M,1).EQ.I) THEN
                           COS_WALL= CELL_CUV(IL,M,7)
                           SIN_WALL= CELL_CUV(IL,M,8)
                       ENDIF
                     ENDDO
                     TW_STAR = 2.*UMOL*1000.*(SIN_WALL*(STRESS(IL,K,1)
     &                         +0.5*STRESS(IL,K,2) + 0.5*STRESS(IL,K,4)
     &                         +0.5*STRESS(IL,K,3) + 0.5*STRESS(IL,K,7))
     &                         -COS_WALL*(STRESS(IL,K,5)
     &                         +0.5*STRESS(IL,K,2) + 0.5*STRESS(IL,K,4)
     &                       +0.5*STRESS(IL,K,6) + 0.5*STRESS(IL,K,8)))
                     UW_STAR = SQRT(ABS(TW_STAR/1000.0))
                     RKS_PLUS = MAX(1.0, RKS*UW_STAR/UMOL)
                     IF (RKS_PLUS.LT.25.0)THEN
                      TDISS_PLUS = MIN(  (50.0/RKS_PLUS)**2 ,
     &                            6.0/0.09/(D2D(IL)*UW_STAR/UMOL)**2)
                     ELSE
                      TDISS_PLUS = MIN(  100.0/RKS_PLUS ,
     &                            6.0/0.09/(D2D(IL)*UW_STAR/UMOL)**2)
                     ENDIF

                     VAR_EDGE(I,K) = MAX(1.0E-10,
     &                TW_STAR*TDISS_PLUS/(UMOL*1000.))
```

```
                   ENDIF
C--------------------------BC of TDISS at wall---------------------------
                ENDIF
             Endif
             If(IR .GT. 0) Then
                If(CCM(IR) .EQ. 1.0) THEN
C-------------------------BC of TDISS at wall(by WangJian)------------------------
                   IF (IBC_TDISS_WALL == 1) THEN
                   VAR_EDGE(I,K) = 60.*UMOL/BETA1/(D2D(IR)**2.0)
                   ELSEIF (IBC_TDISS_WALL == 2) THEN

                   DO M = 1, CELL_POLYGEN(IR)
                     IF (CELL_SIDE(IR,M,1).EQ.I) THEN
                         COS_WALL= CELL_CUV(IR,M,7)
                         SIN_WALL= CELL_CUV(IR,M,8)
                     ENDIF
                   ENDDO
                   TW_STAR = 2.*UMOL*1000.*(SIN_WALL*(STRESS(IR,K,1)
     &                       +0.5*STRESS(IR,K,2) + 0.5*STRESS(IR,K,4)
     &                       +0.5*STRESS(IR,K,3) + 0.5*STRESS(IR,K,7))
     &                       -COS_WALL*(STRESS(IR,K,5)
     &                       +0.5*STRESS(IR,K,2) + 0.5*STRESS(IR,K,4)
     &                     +0.5*STRESS(IR,K,6) + 0.5*STRESS(IR,K,8)))
                   UW_STAR = SQRT(ABS(TW_STAR/1000.0))
                   RKS_PLUS = MAX(1.0, RKS*UW_STAR/UMOL)
                   IF (RKS_PLUS.LT.25.0)THEN
                    TDISS_PLUS = MIN(  (50.0/RKS_PLUS)**2 ,
     &                            6.0/0.09/(D2D(IR)*UW_STAR/UMOL)**2)
                   ELSE
                    TDISS_PLUS = MIN(  100.0/RKS_PLUS ,
     &                            6.0/0.09/(D2D(IR)*UW_STAR/UMOL)**2)
                   ENDIF

                   VAR_EDGE(I,K) = MAX(1.0E-10,
     &              TW_STAR*TDISS_PLUS/(UMOL*1000.))
                   ENDIF
C-------------------------BC of TDISS at wall------------------------
                ENDIF
             Endif
          Endif
          If(CFM(I) .EQ. -3.0) Then
```

```fortran
                If(IL .GT. 0) Then
                   If(CCM(IL) .EQ. 1.0) VAR_EDGE(I,K) = TDISS(IL,K)
                Endif
                If(IR .GT. 0) Then
                   If(CCM(IR) .EQ. 1.0) VAR_EDGE(I,K) = TDISS(IR,K)
                Endif
             Endif
          Enddo
!$OMP END DO
       Enddo
!$OMP DO
     Do I = 1, IJM
        If(CCM(I) .EQ. 1.0) Then
           Do K = 2, KBM
              GRADTKDZ(I,K) = (TDISS(I,K-1)-TDISS(I,K))/DC(I)/DZZ(K-1)
           Enddo
           GRADTKDZ(I,1) = 0.0
           GRADTKDZ(I,KB) = (TDISS(I,KBM) - 60. * UMOL / 0.075 /
     &                       (0.5 * DZ(KBM) * DC(I)) ** 2.) /
     &                       (0.5 * DZ(KBM) * DC(I))

C             VAR_UD(I,K) = (TDISS(I,K-1) + TDISS(I,K)) / 2.0
C          Enddo
C          If(KB .GE. 3) Then
C             VAR_UD(I,1) = 2. * TDISS(I,1) - TDISS(I,2)
C             VAR_UD(I,KB) = 60. * UMOL / 0.075 /
C     &                       (0.5 * DZ(KBM) * DC(I)) ** 2.
C          Else
C             VAR_UD(I,1) = TDISS(I,1)
C             VAR_UD(I,KB) = 60. * UMOL / 0.075 /
C     &                       (0.5 * DZ(KBM) * DC(I)) ** 2.
C          Endif
        Endif
     Enddo
!$OMP END DO
!$OMP END PARALLEL
     Call GRAD_XY(VAR_EDGE,GRADTKDX,GRADTKDY)
C    Call GRAD_Z(VAR_UD,GRADTKDZ)
C-----   Boundary conditions
----------------------------------------------------c
        If(NUMEBC .NE. 0) Then
```

```
          Do N = 1, NUMEBC
           ID = IEBC(N)
           IS = IEBCINX(N)
           Do K = 1, KBM
              GRADTKDX(ID,K) = 0.0
              GRADTKDY(ID,K) = 0.0
           Enddo
        Enddo
      Endif
C-----  astrotidal boundary condition
      If(NUMAST .NE. 0) Then
        Do N = 1, NUMAST
           ID = IABC(N)
           IS = IABCINX(N)
           Do K = 1, KBM
              GRADTKDX(ID,K) = 0.0
              GRADTKDY(ID,K) = 0.0
           Enddo
        Enddo
      Endif
c-----  discharge boundary condition
      If(NUMQBC .NE. 0) Then
         Call BCOND(3)
         Do N = 1, NUMQBC
            ID = IQBC(N)
            IS = IQBCINX(N)
            ISS = CELL_SIDE(ID,IS,1)
            Do K = 1, KBM
               GRADTKDX(ID,K) = 0.0
               GRADTKDY(ID,K) = 0.0
            Enddo
         Enddo
      Endif
C-----  velocity boundary condition
      If(NUMVBC .NE. 0) Then
         Do N = 1, NUMVBC
            ID = IVBC(N)
            IS = IVBCINX(N)
            Do K = 1, KBM
               GRADTKDX(ID,K) = 0.0
               GRADTKDY(ID,K) = 0.0
```

```
          Enddo
        Enddo
      Endif
C---------------------------------------------------------------------------------
---C
c                 calculate the cross-diffusion term
C
C---------------------------------------------------------------------------------
---C

C----------------correct the gradient of k and
w------------------------------------c
c -----------------added by
WangJian,2009/11/17--------------------------------------c
c---------------------------------------------------------------------------------
---c

!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,ELFX,ELFY,FXH,FYH)
!$OMP DO
      Do I = 1, IJM
        If(CCM(I) .EQ. 1.0) Then
          ELFX = 0.0
          ELFY = 0.0
          FXH = 0.0
          FYH = 0.0
          Do J = 1, CELL_POLYGEN(I)
               If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
```



Δl_J

$\frac{\eta_L+\eta_R}{2}$

```
                 ELFX = ELFX + CELL_CUV(I,J,6) *
     &                (ELF(CELL_SIDE(I,J,2)) + ELF(I)) / 2. *
     &                 CELL_CUV(I,J,7)
```

→ cosα

```
                 ELFY = ELFY + CELL_CUV(I,J,6) *
     &                (ELF(CELL_SIDE(I,J,2)) + ELF(I)) / 2. *
     &                 CELL_CUV(I,J,8)
               Else
                ELFX = ELFX + CELL_CUV(I,J,6) *
     &                ELF(I) * CELL_CUV(I,J,7)
                ELFY = ELFY + CELL_CUV(I,J,6) *
     &                ELF(I) * CELL_CUV(I,J,8)
```

m^2

静水深 $H \cdot \Delta l \cdot \cos\alpha$

```
               Endif
               FXH = FXH + HS(CELL_SIDE(I,J,1)) *
     &                    CELL_CUV(I,J,7) * CELL_CUV(I,J,6)
```

```
                 FYH = FYH + HS(CELL_SIDE(I,J,1)) *
     &                          CELL_CUV(I,J,8) * CELL_CUV(I,J,6)
              Enddo
              ELFX = ELFX / AREA(I)
              ELFY = ELFY / AREA(I)
              FXH = FXH / AREA(I)
              FYH = FYH / AREA(I)
              Do K = 1, KB
```

$\frac{\partial k_x}{\partial x}$

```
                 GRADTKX(I,K) = GRADTKX(I,K)-GRADTKZ(I,K) / DC(I)
     &                          * ( (1. + ZZ(K)) * ELFX + ZZ(K) * FXH )
                 GRADTKY(I,K) = GRADTKY(I,K)-GRADTKZ(I,K) / DC(I)
     &                          * ( (1. + ZZ(K)) * ELFY + ZZ(K) * FYH )
                 GRADTKDX(I,K) = GRADTKDX(I,K)-GRADTKDZ(I,K) / DC(I)
     &                          * ( (1. + ZZ(K)) * ELFX + ZZ(K) * FXH )
                 GRADTKDY(I,K) = GRADTKDY(I,K)-GRADTKDZ(I,K) / DC(I)
     &                          * ( (1. + ZZ(K)) * ELFY + ZZ(K) * FYH )
              Enddo
           Endif
        Enddo
!$OMP END DO
!$OMP END PARALLEL

!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,D1,FY1,CDKO,FY11,FY12,FY13)
!$OMP DO
        Do I = 1, IJM
           If(CCM(I) .EQ. 1.0) Then
              Do K = 1, KBM
```

$\sigma_{w2} \cdot \frac{1}{\omega} \cdot \frac{\partial k}{\partial x_j} \cdot \frac{\partial w}{\partial x_j}$

ω

```
                 CROSSDIFFUSION(I,K) = SIGMA02 / TDISS(I,K) *
     &                 (GRADTKX(I,K)*GRADTKDX(I,K) + GRADTKY(I,K)*GRADTKDY(I,K) +
     &                  (GRADTKZ(I,K)+GRADTKZ(I,K+1))/2.*
     &                  (GRADTKDZ(I,K)+GRADTKDZ(I,K+1))/2.)
              Enddo
           Endif
       Enddo
!$OMP END DO
!$OMP DO
        Do I = 1, IJM
           If(CCM(I) .EQ. 1.0) Then
              Do K = 1, KBM
C                CROSSDIFFUSION(I,K) = SIGMA02 / (TDISS(I,K)+1.E-6) *
C     &                 (GRADTKX(I,K)*GRADTKDX(I,K) + GRADTKY(I,K)*GRADTKDY(I,K) +
```

```fortran
C     &                  (GRADTKZ(I,K)+GRADTKZ(I,K+1))/2.*
C     &                  (GRADTKDZ(I,K)+GRADTKDZ(I,K+1))/2.)

                D1 = DC(I) * (1. + ZZ(K))
                D1 = Min(D1,D2D(I))

                CDKO = Max(2.*(CROSSDIFFUSION(I,K)),1.0E-15)
```

$CD = \max(2CRD, 10^{-15})$

```fortran
c               FY1 = Max(Sqrt(TKE(I,K)) / 0.09 / (TDISS(I,K)+1.E-10)
c     &                     / D1,500. * UMOL / D1**2. / (TDISS(I,K)+1.E-10))

c-----------This term "4.*SIGMAO2*TKE(I,K)/CDKO/D1**2." may impede the convergence
C-----------this bug has been fixed on 07/06/2019, by WangJian
                IF (TDISS(I,K) .LT. 1.0E-6) THEN
              FY1 = DSqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1
                ELSE
              FY1 =
```

$\phi_1 = \min\left(\max\left(\frac{\sqrt{k}}{0.09\omega y}, \frac{500\nu}{y^2\omega}\right), \frac{4\sigma_{\omega 2} k}{CD\, y^2}\right)$

```fortran
     &          Min( Min( Max( DSqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1,
     &                         500. * UMOL / D1**2. / TDISS(I,K) ),
     &                      4.*SIGMAO2*(TKE(I,K))/CDKO/D1**2. ),
     &                 10.0)
                ENDIF
!------------Test F1, added by WangJian
                !FY11 =DSqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1
                !FY12 =500. * UMOL / D1**2. / (TDISS(I,K))
                !FY13 =4.*SIGMAO2*(TKE(I,K))/CDKO/D1**2.
                !If( FY11.GT. FY12 .AND.  FY11.LT.FY13 )THEN
                !   RRF1(I,K) = FY11
                !Elseif(FY12.GE. FY11 .AND.  FY12.LT.FY13)THEN
                !   RRF1(I,K) = FY12
                !Elseif(FY13.LE. FY11 .AND.  FY13.LE.FY12)THEN
                !   RRF1(I,K) = FY13
                !Endif
                  F1(I,K) = DTanh(FY1 ** 4.)
```

$F_1 = \tanh(\phi_1^4)$

```fortran
                  RRF1(I,K) = F1(I,K)
                CROSSDIFFUSION(I,K) = 2. * (1. - F1(I,K)) * CROSSDIFFUSION(I,K)
```

$2(1-F_1)\,\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\cdot\frac{\partial \omega}{\partial x_j}$

```fortran
             Enddo
          Endif
       Enddo
!$OMP END DO
!$OMP END PARALLEL
```

本质：同流速动量方程 $\Rightarrow A^*_{ix} K^*_{xi} = BB^*_i$

```
      Call ADVTK(S1,F1,GRADTKX,GRADTKY,GRADTKZ,CDES)
    Call PROFTK(F1)   TKE (k)
      Call ADVTKD(CROSSDIFFUSION,F1,S1,GRADTKDX,GRADTKDY,GRADTKDZ)
    Call PROFTKD(F1)  TDISS (ε,ω)
c==============================================================================c
c                natural eddy viscosity                                        c
c==============================================================================c
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(D1,FY2,I,K,ZSTAR,EP,FY21,FY22)
!$OMP DO
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            Do K = 1, KBM
               D1 = DC(I) * (1. + ZZ(K))
               D1 = Min(D1,D2D(I))
c               D1 = Min(D1,D2D(I),-DC(I)*ZZ(K))
               ZSTAR = D1 * 1.E6 *    Cf·ρU = τw
     &              (Sqrt(Sqrt(TBX(I)**2.+TBY(I)**2.)/RMEAN(I,KBM)))  dz
                                 from CBC (Cf)      ρ=1000
              IF (TDISS(I,K) .LT. 1.0E-6) THEN
               FY2=2.*Sqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1
              Else
              FY2 = Min( Max(2.*Sqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1,
     &                       500. * UMOL / D1**2. / (TDISS(I,K))    ),
     &                     100.)
              Endif
```

$\phi_2 = \max\left(2\frac{\sqrt{k}}{0.09\,\omega y}, \frac{500\nu}{y^2\omega}\right)$

```
               F2(I,K) = Tanh(FY2 ** 2.)
              F3(I,K) = 1. - Tanh(150.*UMOL/D1**2./(TDISS(I,K)) )
```

$F_2 = \tanh(\phi_2^2)$

```
              RRF2(I,K) = F2(I,K)
              RRF3(I,K) = F3(I,K)
              IF (TDISS(I,K) .LT. 1.0E-6) THEN
               VIS(I,K) = 1.0E-10
              Else
              VIS(I,K) = A1 * TKE(I,K) /
     &                   Max(A1*(TDISS(I,K)),TENSOR(I,K)*F2(I,K))
              Endif
               VIS(I,K) = MAX(VIS(I,K),1.0E-10)
C                VIS(I,K) = TKE(I,K) / (TDISS(I,K)+1.E-6)
!-------------Test F2,added by WangJian----------------------------------c
              !FY21 =2.*Sqrt(TKE(I,K))/0.09/(TDISS(I,K))/D1
```

☆ $\nu_t = \frac{a_1 k}{\max(a_1\omega, \Omega F_2)}$

```
                 !FY22 =500. * UMOL / D1**2. / (TDISS(I,K))
                 !If( FY21.GE. FY22  )Then
                 !     RRF2(I,K) = 1.0
                 !Else
                 !     RRF2(I,K) = 2.0
                 !Endif
                 !If(A1*(TDISS(I,K)).GE.TENSOR(I,K)*F2(I,K)) Then
                 !    RRF3(I,K) = 1.0
                 !Else
                 !    RRF3(I,K) = 2.0
                 !Endif
!-------------Test F2,added by WangJian--------------------------------c
              Enddo
             Endif
        Enddo
!$OMP END DO
c============================================================================c
c                horizontal and vertical eddy viscosity                     c
c============================================================================c
      Do K = 1, KBM
!$OMP DO
         Do I = 1, IJM
            If(VERTMIX .EQ. 'CONSTANT  ') Then
               AAM(I,K) = VIS(I,K)
               KM(I,K)  = VIS(I,K)
            Else
    νt         KM(I,K)  = VIS(I,K)                νt
                      If(HORZMIX .NE. 'CLOSURE   ') AAM(I,K) = VIS(I,K)
            Endif
         Enddo                 νt = f(k,ω)
!$OMP END DO
      Enddo
!$OMP END PARALLEL
        DO I = 1, IJM
C         PRINT*, TBX(I),KM(I,KBM),UMOL
        ENDDO
c=============================================================================c
c                      open boundary treatments                                c
c=============================================================================c
C-----    elevation boundary condition
         If(NUMEBC .NE. 0) Then
```

```
        Do N = 1, NUMEBC
           ID = IEBC(N)
           Do K = 1, KBM
              AAM(ID,K) = 0.0
C             KM(ID,K) = 0.0
           Enddo
        Enddo
      Endif
C-----  astrotidal boundary condition
      If(NUMAST .NE. 0) Then
        Do N = 1, NUMAST
           ID = IABC(N)
           Do K = 1, KBM
              AAM(ID,K) = 0.0
C             KM(ID,K) = 0.0
           Enddo
        Enddo
      Endif
c-----  discharge boundary condition
      If(NUMQBC .NE. 0) Then
         Do N = 1, NUMQBC
            ID = IQBC(N)
            Do K = 1, KBM
               AAM(ID,K) = 0.0
C              KM(ID,K) = 0.0
            Enddo
         Enddo
      Endif
C-----  velocity boundary condition
      If(NUMVBC .NE. 0) Then
         Do N = 1, NUMVBC
            ID = IVBC(N)
            Do K = 1, KBM
               AAM(ID,K) = 0.0
C              KM(ID,K) = 0.0
            Enddo
         Enddo
      Endif
c----------------------------------------------------------------------------c
   Return
   End
```