```
C#############################################################################
c                                                                            #
c                   SUBROUTINE PROGRAM                                       #
C                 VERSION 1.0 (25/05/2009)                                   #
C                 AUTHORIZED BY ZHANG JINGXIN                                #
C                               SHANGHAI JIAO TONG UNIVERSITY                #
C                               SHANGHAI, CHINA                              #
c----------------------------------------------------------------------------#
c              TVD scheme for surface variants                               #
c                                                                            #
c#############################################################################
      Subroutine TVDSCHEMEV(QSUR,Q,VAR,GRADZ,LIMTER)
      Include './Include/OCERM_INF'
C   Parameter(LIMTER = 8)
    Integer LIMTER
    Dimension QSUR(IJM,KB),Q(IJM,KB),GRADZ(IJM,KB),VAR(IJM,-1:KB+1)
C   Dimension TVDCOE(IJM,KB)
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,RF,FLUX1,FLUX2,FLUX3,
!$OMP&         SI1,SI2,SI3,ALF1,ALF2,ALF3,OME1,OME2,OME3)
CC  Do K = 1, KB
CC!$OMP DO
CC     Do I = 1, IJM
CC        TVDCOE(I,K) = 0.0
c         TVDCOE(I,K) = 0.0
CC     Enddo
CC!$OMP END DO NOWAIT
CC  Enddo
CC  Do K = 2, KBM - 1
CC!$OMP DO
CC     Do I = 1, IJM
CC        If(CCM(I) .EQ. 1.0) Then
CC           TVDCOE(I,K) = .5 * ((Q(I,K-1) - Q(I,K)) / DZZ(K-1) +
CC     &                         (Q(I,K) - Q(I,K+1)) / DZZ(K))
CC        Endif
CC     Enddo
CC!$OMP END DO NOWAIT
CC  Enddo
CC  If(KBM .GT. 1) Then
CC!$OMP DO
CC  Do I = 1, IJM
CC     If(CCM(I) .EQ. 1.0) Then
CC        TVDCOE(I,1) = (Q(I,1) - .5 * (Q(I,1) + Q(I,2))) / (.5*DZ(1))
CC        TVDCOE(I,KBM) = (.5 * (Q(I,KBM)+Q(I,KBM-1)) - Q(I,KBM)) /
CC     &                      (.5*DZ(KBM))
CC     Endif
CC   Enddo
CC!$OMP END DO NOWAIT
CC   Endif
C============================================================================C
C                5TH WENO SCHEME                                             C
C============================================================================C
      If(LIMTER .EQ. 0) Then
!$OMP DO
         Do I = 1, IJM
            If(CCM(I) .EQ. 1.0) Then
               Do K = 2, KBM
                  If(W(I,K) .GE. 0.0) Then   !   POSITIVE VEL.
                                              !   SMOOTHING FACTOR
                     SI1 = 13./12.*
     &                        (VAR(I,K+2)-2.*VAR(I,K+1)+VAR(I,K))**2.+
     &                        1./4.  *
     &                        (VAR(I,K+2)-4.*VAR(I,K+1)+3.*VAR(I,K))**2.
                     SI2 = 13./12.*
     &                        (VAR(I,K+1)-2.*VAR(I,K)+VAR(I,K-1))**2.+
     &                        1./4.  *
     &                        (VAR(I,K+1)-VAR(I,K-1))**2.
                     SI3 = 13./12.*
     &                        (VAR(I,K)-2.*VAR(I,K-1)+VAR(I,K-2))**2.+
     &                        1./4.  *
     &                        (3.*VAR(I,K)-4.*VAR(I,K-1)+VAR(I,K-2))**2.

                     ! ALF(K) = C(K)/(EPS +SI(K))

                     ALF1= C_PLUX(1) / (SI1 + 1.E-6)
                     ALF2= C_PLUX(2) / (SI2 + 1.E-6)
                     ALF3= C_PLUX(3) / (SI3 + 1.E-6)

                     ! OME = ALF / SUM(ALF)

                     OME1 = ALF1 / (ALF1 + ALF2 + ALF3)
                     OME2 = ALF2 / (ALF1 + ALF2 + ALF3)
                     OME3 = ALF3 / (ALF1 + ALF2 + ALF3)
```

```
        !  FLUX_WENO = SUM (OME*FLUX)

        FLUX1 = ALF_PLUX(K,1,1) * VAR(I,K+2) +
&                   ALF_PLUX(K,1,2) * VAR(I,K+1) +
&                   ALF_PLUX(K,1,3) * VAR(I,K)

        FLUX2 = ALF_PLUX(K,2,1) * VAR(I,K+1) +
&                   ALF_PLUX(K,2,2) * VAR(I,K) +
&                   ALF_PLUX(K,2,3) * VAR(I,K-1)

        FLUX3 = ALF_PLUX(K,3,1) * VAR(I,K) +
&                   ALF_PLUX(K,3,2) * VAR(I,K-1) +
&                   ALF_PLUX(K,3,3) * VAR(I,K-2)

        !  WENO FLUX

        QSUR(I,K) = OME1*FLUX1 + OME2*FLUX2 + OME3*FLUX3

      Else                 ! NEGATIVE VEL.

        SI1 = 13./12.*
&                    (VAR(I,K-3)-2.*VAR(I,K-2)+VAR(I,K-1))**2.+
&                    1./4.  *
&                  (VAR(I,K-3)-4.*VAR(I,K-2)+3.*VAR(I,K-1))**2.
        SI2 = 13./12.*
&                    (VAR(I,K-2)-2.*VAR(I,K-1)+VAR(I,K))**2.+
&                    1./4.  *
&                    (VAR(I,K-2)-VAR(I,K))**2.
        SI3 = 13./12.*
&                    (VAR(I,K-1)-2.*VAR(I,K)+VAR(I,K+1))**2.+
&                    1./4.  *
&                    (3.*VAR(I,K-1)-4.*VAR(I,K)+VAR(I,K+1))**2.

        ! ALF(K) = C(K)/(EPS +SI(K))

        ALF1= C_MINU(1) / (SI1 + 1.E-6)
        ALF2= C_MINU(2) / (SI2 + 1.E-6)
        ALF3= C_MINU(3) / (SI3 + 1.E-6)

        ! OME = ALF / SUM(ALF)
```

```
                     OME1 = ALF1 / (ALF1 + ALF2 + ALF3)
                     OME2 = ALF2 / (ALF1 + ALF2 + ALF3)
                     OME3 = ALF3 / (ALF1 + ALF2 + ALF3)

                     !  FLUX_WENO = SUM (OME*FLUX)

                     FLUX1 = ALF_MINU(K,1,1) * VAR(I,K-3) +
     &                       ALF_MINU(K,1,2) * VAR(I,K-2) +
     &                       ALF_MINU(K,1,3) * VAR(I,K-1)
                     FLUX2 = ALF_MINU(K,2,1) * VAR(I,K-2) +
     &                       ALF_MINU(K,2,2) * VAR(I,K-1) +
     &                       ALF_MINU(K,2,3) * VAR(I,K)
                     FLUX3 = ALF_MINU(K,3,1) * VAR(I,K-1) +
     &                       ALF_MINU(K,3,2) * VAR(I,K) +
     &                       ALF_MINU(K,3,3) * VAR(I,K+1)

                     !  WENO FLUX

                     QSUR(I,K) = OME1*FLUX1 + OME2*FLUX2 + OME3*FLUX3

                  Endif
               Enddo
            Endif
         Enddo
 !$OMP END DO
      Endif
C===========================================================================C
    If(LIMTER .GT. 0) Then
    Do K = 2, KBM
!$OMP DO
       Do I = 1, IJM
          If(CCM(I) .EQ. 1.0) Then
              If(W(I,K) .GE. 0.0) Then    !#输入变量统一为梯度#, WJ, 2019-12-13
11:10:19 !
              RF = (GRADZ(I,K)+GRADZ(I,K+1)) * (ZZ(K-1) - ZZ(K)) * DC(I)
     &             / (Q(I,K-1)-Q(I,K) + Sign(1.E-10,(Q(I,K-1)-Q(I,K))) )
     &             - 1.0
```

Δr

$$r_f = \frac{(\nabla_z\phi_k + \nabla_z\phi_{k-1})\cdot\Delta r}{\phi_{k-1} - \phi_k}$$

```
c                RF = GRADZ(I,K+1) / (GRADZ(I,K)+Sign(1.E-15,GRADZ(I,K)))
              QSUR(I,K) = Q(I,K) + 0.5 * FUNLIMTER(LIMTER,RF) *
     &                       (Q(I,K-1) - Q(I,K))
```

$$\phi_f = \phi_c + \frac{1}{2}\psi(r_f)(\phi_{k-1} - \phi_k)$$

```
            Else
              RF = (GRADZ(I,K)+GRADZ(I,K-1)) * (ZZ(K) - ZZ(K-1)) * DC(I)
     &             / (Q(I,K)-Q(I,K-1) + Sign(1.E-10,(Q(I,K)-Q(I,K-1))) )
     &             -  1.0
c                RF = GRADZ(I,K-1) / (GRADZ(I,K)+Sign(1.E-15,GRADZ(I,K)))
              QSUR(I,K) = Q(I,K-1) + 0.5 * FUNLIMTER(LIMTER,RF) *
     &                       (Q(I,K) - Q(I,K-1))
            Endif
         Endif
       Enddo
!$OMP END DO
    Enddo
    Endif
!$OMP END PARALLEL
    Return
    End
C   Function FUNLIMTER(LIMTER,RF)
C   Double precision RF
C     Goto (1,2,3,4,5,6,7) LIMTER
C1  Continue                    ! SUPERBEE
C     FUNLIMTER = Dmax1(0.0,Dmin1(1.0,2.*RF), Dmin1(2.0,RF))
C   Goto 100
C2  Continue                    ! Van Leer
C     FUNLIMTER = (RF + Abs(RF)) / (1.0 + RF)
C   Goto 100
C3  Continue                    ! Van Albada
C     FUNLIMTER = (RF + RF ** 2.) / (1. + RF ** 2.)
C   Goto 100
C4  Continue                    ! Min-Mod
C     If(RF .GT. 0.0) Then
C       FUNLIMTER = Dmin1(RF,1.0)
C     Else
C       FUNLIMTER = 0.0
C     Endif
C   Goto 100
C5  Continue                    ! Sweby
C       FUNLIMTER = Dmax1(0.0, Dmin1(1.0,1.5*RF), Dmin1(1.5,RF))
C   Goto 100
C6     Continue                    ! QUICK
C     FUNLIMTER = Dmax1(0.0, Dmin1(2.0*RF,(3.+RF)/4.,2.))
C   Goto 100
```

```
C7  Continue                    ! UMIST
C     FUNLIMTER = Dmax1(0.0, Dmin1(2.0*RF,(1.+3.*RF)/4.,
C     &                 (3.+RF)/4.,2.))
C   Goto 100
C100    Continue
C   Return
C   End
```