```
C###############################################################################
c                                                                              #
c                    SUBROUTINE PROGRAM                                        #
C                  VERSION 1.0 (12/07/2009)                                    #
C                  AUTHORIZED BY ZHANG JINGXIN                                 #
C                               SHANGHAI JIAO TONG UNIVERSITY                  #
C                               SHANGHAI, CHINA                                #
c------------------------------------------------------------------------------#
c                Vertical eddy coefficient based on the Subgrid model          #
c                                                                              #
c###############################################################################

      Subroutine SUBGRIDV

      Include './Include/OCERM_INF'
    Parameter(KEY = 1)

C   If (VERTMIX .EQ. 'SMAGMODEL ') Goto 11
C   If (VERTMIX .EQ. 'SAMODEL   ') Goto 12
      If(VERTMIX .EQ. 'SMAGMODEL ') Then
C11 Continue
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,K,DIVU,DIVV)
!$OMP DO
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
C         Goto (1,2) KEY
C1        Continue
            If(KEY .EQ. 1) Then
          Do K = 2, KBM - 1
             DIVU = .5 * (UR(I,K-1) - UR(I,K)) / DZZ(K-1) +
     &                .5 * (UR(I,K) - UR(I,K+1)) / DZZ(K)
             DIVV = .5 * (VR(I,K-1) - VR(I,K)) / DZZ(K-1) +
     &                .5 * (VR(I,K) - VR(I,K+1)) / DZZ(K)

c            If((1.+ZZ(K)) .LE. 0.1 / 0.41) Then
               KM(I,K) = 0.41 ** 2. * DC(I) * (1.+ZZ(K)) **2. *
     &                     Sqrt(DIVU **2. + DIVV **2.)
c            Else
c              KM(I,K) = UMOL + 0.1 ** 2. * DC(I) * (1.+ZZ(K)) **2. *
c     &                     DSqrt(DIVU **2. + DIVV **2.)
c            Endif
```

```
          Enddo
C         If(KBM .GT. 1) Then
C           DIVU = (.5 * (UR(I,KBM-1) + UR(I,KBM)) - UR(I,KBM)) /
C     &             (0.5 * DZ(KBM))
C           DIVV = (.5 * (VR(I,KBM-1) + VR(I,KBM)) - VR(I,KBM)) /
C     &             (0.5 * DZ(KBM))
C         Else
            DIVU = UR(I,KBM) / (0.5 * DZ(KBM))
            DIVV = VR(I,KBM) / (0.5 * DZ(KBM))
C         Endif
c           If((1.+ZZ(KBM)) .LE. 0.1 / 0.41) Then
          KM(I,KBM) = 0.41 ** 2. * DC(I) * (1.+ZZ(KBM))**2. *
     &                Sqrt(DIVU **2. + DIVV **2.)
c         Else
c           KM(I,KBM) = UMOL + 0.1 ** 2. * DC(I) * (1.+ZZ(KBM))**2. *
c     &                 DSqrt(DIVU **2. + DIVV **2.)
c         Endif
          If(KBM .GT. 1) Then
C           DIVU = (UR(I,1) - .5 * (UR(I,1) + UR(I,2))) / (.5*DZ(1))
C           DIVV = (VR(I,1) - .5 * (VR(I,1) + VR(I,2))) / (.5*DZ(1))
            USURF = UR(I,1) * (0.5 * DZ(1) + DZZ(1)) / DZZ(1) -
     &              UR(I,2) * 0.5 * DZ(1) / DZZ(1)
            VSURF = VR(I,1) * (0.5 * DZ(1) + DZZ(1)) / DZZ(1) -
     &              VR(I,2) * 0.5 * DZ(1) / DZZ(1)
            DIVU = (USURF - 0.5 * (UR(I,1) + UR(I,2))) / DZ(1)
            DIVV = (VSURF - 0.5 * (VR(I,1) + VR(I,2))) / DZ(1)
c             If((1.+ZZ(1)) .LE. 0.1 / 0.41) Then
              KM(I,1) = 0.41 ** 2. * DC(I) * (1.+ZZ(1))**2. *
     &                  Sqrt(DIVU **2. + DIVV **2.)
c           Else
c             KM(I,1) = UMOL + 0.1 ** 2. * DC(I) * (1.+ZZ(1))**2. *
c     &                 DSqrt(DIVU **2. + DIVV **2.)
c           Endif
          Endif
          Endif
C         Goto 100
C2        Continue
            If(KEY .EQ. 2) Then
          Do K = 2, KBM - 1
            DIVU = .5 * (UR(I,K-1) - UR(I,K)) / DZZ(K-1) +
     &             .5 * (UR(I,K) - UR(I,K+1)) / DZZ(K)
```

```
            DIVV = .5 * (VR(I,K-1) - VR(I,K)) / DZZ(K-1) +
     &                .5 * (VR(I,K) - VR(I,K+1)) / DZZ(K)

            KM(I,K) = VERCON * DC(I) * DZ(K) ** 2. *
     &                  Sqrt(DIVU **2. + DIVV **2.)
          Enddo
          If(KBM .GT. 1) Then
            DIVU = (.5 * (UR(I,KBM-1) + UR(I,KBM)) - UR(I,KBM)) /
     &             (0.5 * DZ(KBM))
            DIVV = (.5 * (VR(I,KBM-1) + VR(I,KBM)) - VR(I,KBM)) /
     &             (0.5 * DZ(KBM))
          Else
              DIVU = UR(I,KBM) / (0.5 * DZ(KBM))
            DIVV = VR(I,KBM) / (0.5 * DZ(KBM))
          Endif
          KM(I,KBM) = VERCON * DC(I) * DZ(KBM) ** 2. *
     &                  Sqrt(DIVU **2. + DIVV **2.)
          If(KBM .GT. 1) Then
            DIVU = (UR(I,1) - .5 * (UR(I,1) + UR(I,2))) / (.5*DZ(1))
            DIVV = (VR(I,1) - .5 * (VR(I,1) + VR(I,2))) / (.5*DZ(1))
C           USURF = UR(I,1) * (0.5 * DZ(1) + DZZ(1)) / DZZ(1) -
C     &                 UR(I,2) * 0.5 * DZ(1) / DZZ(1)
C           VSURF = VR(I,1) * (0.5 * DZ(1) + DZZ(1)) / DZZ(1) -
C     &                 VR(I,2) * 0.5 * DZ(1) / DZZ(1)
C           DIVU = (USURF - 0.5 * (UR(I,1) + UR(I,2))) / DZ(1)
C           DIVV = (VSURF - 0.5 * (VR(I,1) + VR(I,2))) / DZ(1)
            KM(I,1) = VERCON * DC(I) * DZ(1) ** 2. *
     &                  Sqrt(DIVU **2. + DIVV **2.)
          Endif
          Endif
C         Goto 100
C100       Continue
        Endif
      Enddo
!$OMP END DO NOWAIT
C-------------------------------------------------------------------------C
C              Diffusion of mass transport                                c
c-------------------------------------------------------------------------c
!$OMP DO
      Do I = 1, IJM
        If(CCM(I) .EQ. 1.0) Then
```

```
          Do K = 1, KBM
             KQ(I,K) = KM(I,K)
          Enddo
       Endif
    Enddo
!$OMP END DO
!$OMP END PARALLEL
    Endif
CC  Goto 1000
c-------------------------------------------------------------------------------c
C12 Continue
      If (VERTMIX .EQ. 'SAMODEL    ') Then
c-----   S-A modle  ----------------------------------------------------------c
         CALL DESSA
      Endif
      If (VERTMIX .EQ. 'SSTMODEL   ') Then
c-----   S-A modle  ----------------------------------------------------------c
         CALL DESSST
      Endif
C   Goto 1000
c-------------------------------------------------------------------------------c
C1000   Continue
    Return
    End
```