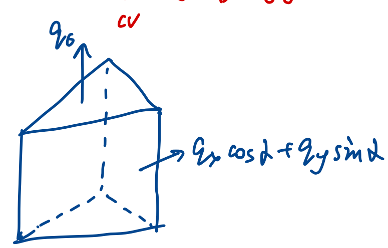
```
C###########################################################################
c                                                                          #
c                      SUBROUTINE PROGRAM                                  #
C                    VERSION 1.0 (12/07/2009)                              #
C                    AUTHORIZED BY ZHANG JINGXIN                           #
C                                  SHANGHAI JIAO TONG UNIVERSITY           #
C                                  SHANGHAI, CHINA                         #
c--------------------------------------------------------------------------#
c              computes the velocity in the Sigma direction                #
c                                                                          #
c###########################################################################
      Subroutine VERTVL
      Include './Include/OCERM_INF'
    Dimension FLUX(IJM,KB)
c--------------------------------------------------------------------------c
c                    Calculate new vertical velocity QW                    C
c--------------------------------------------------------------------------c
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K,IL,IR)
!$OMP DO
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
          W(I,1) = 0.0
          W(I,KB) = 0.0
       Endif
       Do K = 1, KB
          FLUX(I,K) = 0.0
       Enddo
    Enddo
!$OMP END DO
!$BARRIER
c----    flux of every cell  -------------------------------------------------c
    Do K = 1, KBM
!$OMP DO
       Do I = 1, IJM
          If(CCM(I) .EQ. 1.0) Then
             Do J = 1, CELL_POLYGEN(I)
                If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
                     IL = I
                   IR = CELL_SIDE(I,J,2)
                     FLUX(I,K) = FLUX(I,K) + THITA * CELL_CUV(I,J,6) *
     &                               PORE_HF(CELL_SIDE(I,J,1),K) *
```

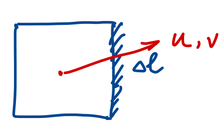
$$\int_{CV} \frac{q_{zi}^* - q_{zi}^n}{\Delta t}\, dV = \int_{CV} F q_{zi}^n\, dV + \int_{CV} \frac{\partial}{D\partial\sigma}\left(\frac{\nu_t}{D}\frac{\partial q_z^*}{\partial\sigma}\right)_i dV$$



$$\theta\cdot\Delta l\cdot\left[\frac{u_i^*\sqrt{A_i}+u_{i+1}^*\sqrt{A_{i+1}}}{\sqrt{A_i}+\sqrt{A_{i+1}}}\cos\alpha + \frac{v_i^*\sqrt{A_i}+v_{i+1}^*\sqrt{A_{i+1}}}{\sqrt{A_i}+\sqrt{A_{i+1}}}\sin\alpha\right] + (1-\theta)\,\Delta l\,(h+\eta)\left[u_i^n\cos\alpha + v_i^n\sin\alpha\right]$$

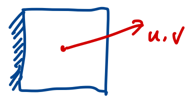
```
     &                          ((U(IL,K) * Sqrt(AREA(IR)) +
     &                            U(IR,K) * Sqrt(AREA(IL))) /
     &                           (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
     &                           CELL_CUV(I,J,7) +
     &                           (V(IL,K) * Sqrt(AREA(IR)) +
     &                            V(IR,K) * Sqrt(AREA(IL)))/
     &                           (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
     &                           CELL_CUV(I,J,8)) + (1. - THITA) *
     &                           PORE_HF(CELL_SIDE(I,J,1),K) *
     &                           CELL_CUV(I,J,6) *
     &                           DS(CELL_SIDE(I,J,1)) *
     &                         (UN(CELL_SIDE(I,J,1),K) * CELL_CUV(I,J,7)+
     &                          VN(CELL_SIDE(I,J,1),K) * CELL_CUV(I,J,8))

C                  FLUX(I,K) = FLUX(I,K) + THITA * CELL_CUV(I,J,6) *
C     &                         (0.5 * (U(I,K)+U(CELL_SIDE(I,J,2),K)) *
C     &                             CELL_CUV(I,J,7) +
C     &                             0.5 * (V(I,K)+V(CELL_SIDE(I,J,2),K)) *
C     &                             CELL_CUV(I,J,8)) +
C     &                             (1. - THITA) * CELL_CUV(I,J,6) *
C     &                             DS(CELL_SIDE(I,J,1)) *
C     &                            (UN(CELL_SIDE(I,J,1),K)*CELL_CUV(I,J,7) +
C     &                             VN(CELL_SIDE(I,J,1),K)*CELL_CUV(I,J,8))
                Endif
              Enddo
            Endif
          Enddo
!$OMP END DO NOWAIT
        Enddo
C==========================================================================c
c            open boundary                                                c
c==========================================================================c
!$OMP MASTER
C-----  elevation boundary condition
    If(NUMEBC .NE. 0) Then
      Do N = 1, NUMEBC
        ID = IEBC(N)
        IS = IEBCINX(N)
        Do K = 1, KBM
c          FLUX(ID,K) = 0.0
           FLUX(ID,K) = FLUX(ID,K) + CELL_CUV(ID,IS,6) *
```



$\Delta l(u\cos\alpha + v\sin\alpha)$

```
     &                          (U(ID,K) * CELL_CUV(ID,IS,7) +
     &                           V(ID,K) * CELL_CUV(ID,IS,8))
          Enddo
        Enddo
      Endif
c-----   discharge boundary condition
      If(NUMQBC .NE. 0) Then
        Do N = 1, NUMQBC
           ID = IQBC(N)
             IS = IQBCINX(N)
           ISS = CELL_SIDE(ID,IS,1)
          Do K = 1, KBM
             FLUX(ID,K) = FLUX(ID,K) + CELL_CUV(ID,IS,6)* DS(ISS) *
     &                           (UN(ISS,K) * CELL_CUV(ID,IS,7) +
     &                            VN(ISS,K) * CELL_CUV(ID,IS,8))
          Enddo
        Enddo
      Endif
```



$\Delta l\,(h+\eta)(u^n\cos\alpha + v^n\sin\alpha)$

```
c-----   velocity boundary condition
      If(NUMVBC .NE. 0) Then
        Do N = 1, NUMVBC
           ID = IVBC(N)
             IS = IVBCINX(N)
           ISS = CELL_SIDE(ID,IS,1)
          Do K = 1, KBM
             FLUX(ID,K) = FLUX(ID,K) + CELL_CUV(ID,IS,6)* DS(ISS) *
     &                           (UN(ISS,K) * CELL_CUV(ID,IS,7) +
     &                            VN(ISS,K) * CELL_CUV(ID,IS,8))
          Enddo
        Enddo
      Endif
c-----   offshore discharge boundary condition
      If(NUMDBC .NE. 0) Then
        Do N = 1, NUMDBC
           ID = IDBC(N)
          Do K = 1, KBM
             FLUX(ID,K) = FLUX(ID,K) -
     &                         QDIFF(N) * VDDIST(N,K) / 100. * RAMP
          Enddo
        Enddo
      Endif
```

```
c-----   astrotidle boundary condition
    If(NUMAST .NE. 0) Then
       Do N = 1, NUMAST
             ID = IABC(N)
           IS = IABCINX(N)
          Do K = 1, KBM
             FLUX(ID,K) = 0.0
c            FLUX(ID,K) + CELL_CUV(ID,IS,6) *
c     &                        (U(ID,K) * CELL_CUV(ID,IS,7) +
c     &                         V(ID,K) * CELL_CUV(ID,IS,8))
          Enddo
       Enddo
    Endif
C=========================================================================C
C               numerical wave flume                                     c
C=========================================================================C
    If(IFLUME .EQ. 1) Then
       Call WAVEGEN(1)
       Do I = 1, N_SOURCE
          II = IGEN(I)
        Do K = 1, KBM
             FLUX(II,K) = FLUX(II,K) - WGEN(I,K) * AREA(II)
            ENDDO
       Enddo
    Endif
!$OMP END MASTER
!$OMP BARRIER
C------------------------------------------------------------------------C
C              calculating the vertical velocity                         c
c------------------------------------------------------------------------c
!$OMP DO
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
          Do K = KBM, 2, -1
             W(I,K) = PORE_VF(I,K+1) * W(I,K+1) / PORE_VF(I,K) -
    &        qσ          DZ(K) * PORE_VF(I,K) * (ELF(I) - EL(I)) /
    &                    DTI / PORE_VF(I,K) -
    &                    FLUX(I,K) * DZ(K) / AREA(I) / PORE_VF(I,K)
          Enddo

C         Do K = 2, KBM
```

$$\int_{cv} \frac{\partial q_\sigma}{\partial \sigma} dv = \int_{cv} \left(-\frac{\partial \zeta}{\partial t} - \frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y}\right) dv$$

$$q_\sigma = -\frac{\zeta^{n+1} - \zeta^n}{\Delta t} \Delta\sigma - \underbrace{q_x \Delta l \cos\alpha \cdot \frac{\Delta\sigma}{A} - q_y \Delta l \sin\alpha \frac{\Delta\sigma}{A}}_{FLUX \cdot \frac{\Delta\sigma}{A}}$$

```
C            W(I,K) = W(I,K-1) + DZ(K-1) * (ELF(I) - EL(I)) / DTI +
C     &                 FLUX(I,K-1) * DZ(K-1) / AREA(I)
C         Enddo
       Endif
    Enddo
!$OMP END DO
!$OMP END PARALLEL
C============================================================================C
C   Do K = 1, KB
C      Do I = 1, IJM
C         If(CCM(I) .EQ. 1.0) Then
C            W(I,K) = W(I,K) / (HC(I) + ELF(I))
C         Endif
C      Enddo
C   Enddo
c-------------------------------------------------------------------------c
       Return
        End
```