核心：有限体积法离散！

```
c**********************************************************************
c                                                                     *
c                  begin numerical integration                        *
c                                                                     *
c**********************************************************************
                   Do 100 NSTEP = ISTART, IEND
c=======================================================================c
      Print*,'NUM. OF CAL.TSTEP=', NSTEP
        ITENPCT = (IEND - ISTART+1)/10
C         If (Mod(NSTEP-ISTART+1,ITENPCT).EQ.0) Then
C           Write(*,'(f12.2,a8)')
C     &       (NSTEP-ISTART+1)*100./(IEND-ISTART+1),'% Done.'
C         Endif
流量步数  RAMP = Tanh(Float(NSTEP)/Float(IRAMP+1)) / Tanh(1.0)
      If(NSTEP.GT.IRAMP) RAMP=1.0
c     RAMP = Float(NSTEP) / Float(IRAMP + 1)
c     If(NSTEP .GT. IRAMP) RAMP=1.0
        TIME = Float(NSTEP-ISTART+1) * DAYI * DTI
        THOUR = TIME * 24. + HOURSTAR
        Call BCOND(10)  边界条件                    !  boundary conditions
c-----------------------------------------------------------------------c
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I)
      Do I = 1, IJM
         ELF(I) = EL(I)
CCC      HCO(I) = HC(I)
      Enddo
!$OMP END PARALLEL DO
c-----------------------------------------------------------------------c
        If (DEM .NE. 'NEGLECT') THEN
            !Print*, '     NUM. OF DEM.TSTEP=',NNDEM
            Call DEMM
        ENDIF
        Call PORECAL  多孔介质模型
c-----------------------------------------------------------------------c
c             computing wind stress on the surface
c-----------------------------------------------------------------------c
        If (WIND.NE.'NEGLECT') Then
CCC     Call WINDUV                              !  surface wind stress
        Endif
c-----------------------------------------------------------------------C
c             computing baroclinic pressure gradient
```

```
c-----------------------------------------------------------------------C
CCC    Call BAROCLINIC                              !  baroclinic pressure
c-----------------------------------------------------------------------C
c             computing viscosity coefficients
c-----------------------------------------------------------------------C
      If (HORZMIX .EQ. 'CLOSURE   ') Call SUBGRIDH   !  horizontal edd vis.
        If (VERTMIX .NE. 'CONSTANT  ') Call SUBGRIDV   !  vertical edd vis.
      If (SGSTYPE .EQ. 'INCLUDE   ')  Call SGSMODEL  !  LES modle
      If (DES .EQ. 'SADES  ' .OR.
     *       DES .EQ. 'SADDES ' .OR.
     *       DES .EQ. 'SAIDDES' .OR.
     *       DES .EQ. 'SAZDES ' )  Call DESSA   !  DES modle
      If (DES .EQ. 'SSTDES ')   Call DESSST          !  DES modle
c   test
        If(WAVE_BREAKING .EQ. 1.) Then
          Do I = 1, IJM
            If(CCM(I) .EQ. 1.0) Then
              Do K = 1, KBM
                AAM(I,K) = AAM(I,K) + VIS_BW(I,K)
                KM(I,K) = KM(I,K) + VIS_BW(I,K)
              Enddo
            Endif
          Enddo
        Endif
C-----------------------------------------------------------------------C
c             computing bed friction coefficient
c-----------------------------------------------------------------------C
      If(WAVEDYN .EQ. 'NEGLECT') Then
           If(WFBC .EQ. '      FUN1') Then
             If (VERTMIX .EQ. 'SSTMODEL  ') Then
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(IJ,ZSTAR)
               Do IJ = 1, IJM
                 If (DC(IJ) .GT. 0.0) Then
                 IF( SQRT( UR(IJ,KBM)**2. + VR(IJ,KBM)**2. )  !!#Avoid 0
velocity#, WangJian, 2020-3-6 !
     &                    .LT. 1.0E-10 )THEN
                   CBC(IJ)=( 0.41 /Log( (ZZ(KBM)-Z(KB) )*DC(IJ)
     &                          /Z01(IJ) ) )**2.
                 ELSE
                   ZSTAR = Sqrt(0.3)*
     *                   Sqrt(TKE(IJ,KBM)+1.E-20)*DC(IJ)*DZZ(KBM)/1.E-6
```

$C_1\sqrt{k\cdot(H+\eta)}\cdot\Delta\sigma$

$C_2 \frac{k}{\lg 9.81 z^*}$

```
                        CBC(IJ) = 0.41*Sqrt(0.3)*
     *                          Sqrt(TKE(IJ,KBM)+1.E-20)/Log(9.81*ZSTAR)
                     Endif
                     Endif
                  Enddo
!$OMP END PARALLEL DO
               Else
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(IJ,Z0,CBCMIN)
               Do 120 IJ = 1, IJM
                    If (DC(IJ) .GT. 0.0) Then
                     Z0 = Z0B
                     CBCMIN = BFRIC
                    CBC(IJ)=(0.41/Log((ZZ(KBM)-Z(KB))*DC(IJ)/Z01(IJ)))**2.
C-----    manning coefficients
CCCC                CBC(IJ) = 9.8 * Z01(IJ) **2. / DC(IJ) ** 0.333
c               CBC(IJ)=Dmax1(CBCMIN,CBC(IJ))
                  Endif
120               Continue
!$OMP END PARALLEL DO
               Endif
             Endif
C        Else
C           If(WFBC .EQ. '      FUN2') Call WALLDRAG
C        Endif
       Endif
C-------------------------------------------------------------------------C
C      Feeding velocity fluctuation for DES                               c
c-------------------------------------------------------------------------c
        If(DES .EQ. 'SAZDES ') Then

           TIME1 = NSTEP * DTI

           Call TurGenerator(TIME1,NSTEPS)

           Do J = 1, NUM_CELL
           Do K = 1, NUM_VER
              UDIS(J,K) = U_FLU(J,K) * R11(J,K)
              WDIS(J,K) = U_FLU(J,K) * R21(J,K) + V_FLU(J,K) * R22(J,K)
              VDIS(J,K) = U_FLU(J,K) * R31(J,K) + V_FLU(J,K) * R32(J,K)
     &                  + W_FLU(J,K) * R33(J,K)
           Enddo
```

核心 $\int_{cv}\frac{q^*_{xi}-q^n_{xi}}{\Delta t}dv=\int_{cv}F q^n_{xi}dv-\int_{cv}gD\theta\left(\frac{\partial\xi^*}{\partial x}\right)_i dv-\int_{cv}gD(1-\theta)\left(\frac{\partial\xi^n}{\partial x}\right)_i dv+\int_{cv}\frac{\partial}{D\partial\sigma}\left(\frac{\nu_t}{D}\frac{\partial q^n_x}{\partial\sigma}\right)_i dv+\int_{cv}F^n_{zg,t}dv$

Enddo $\int_{cs}\vec{n}\cdot(\rho\phi\vec{u})ds$ convective $\int_{cs}\vec{n}\cdot(\Gamma\nabla\phi)ds$ diffusion

Endif

$\int_{cv}\nabla\phi\,dv=\int_{cs}\vec{n}\phi\,ds\Rightarrow\sum\nabla\phi\Delta v=\sum\phi\vec{n}\Delta s$ ↓ $A_{ij}\phi_j$ ↓ $a_{ij}\phi_{ij}$

```
c-----------------------------------------------------------------------C
c             computing u,v,w,elf,el;
c-----------------------------------------------------------------------C
c     IKEY=0
        Call ADVU                                  !  momentum in x direction
        Call ADVV                                  !  momentum in y direction
      If (HYDTYPE .EQ. 'NONSTATIC') Call ADVW  !  momentum in z direction
cc    If (TFIELD .NE. 'NEGLECT')  Call ADVT     !  momentum of tem.
cc    If (SFIELD .NE. 'NEGLECT')  Call ADVSAL   !  momentum of sal.
cc    If (SEDTRAN .NE. 'NEGLECT') Call ADVSED   !  momentum of sed.
      If (WAQ .NE. 'NEGLECT')     Call ADVC     !  momentum of mass.
cc    If (DEM .NE. 'NEGLECT')     Call DEMM     !  dem model
      Call ELTION                                !  compute the elevation
      Call PROFV                                 !   compute the velocities
cc    If (TFIELD .NE. 'NEGLECT')  Call PROFT    !  momentum of tem.
cc    If (SFIELD .NE. 'NEGLECT')  Call PROFSAL  !  momentum of sal.
cc    If (SEDTRAN .NE. 'NEGLECT') Call PROFSED  !  momentum of sed.
      If (WAQ .NE. 'NEGLECT')     Call PROFC    !  momentum of mass.
c-----------------------------------------------------------------------C
c      hydrostatic case
c-----------------------------------------------------------------------C
      If (HYDTYPE .NE. 'NONSTATIC') Then
           Call VERTVL                           !  compute qw
         Call WREAL                              !  compute w
      Else
c-----------------------------------------------------------------------c
c        nodhydrodynamic simulation                                     c
c-----------------------------------------------------------------------c
           Call PROFW                            !  compute the DW
           Call DYN                              !  hydrodynamic model
           Call UPDATEFLOW                       !  update after DYN
         Call WREAL
      Endif
c-----------------------------------------------------------------------c
c       Immersed boundary method                                        c
c-----------------------------------------------------------------------c
       !If (IBMETHOD .NE. 'NEGLECT')   Call IBM  !  IBM moudle
c-----------------------------------------------------------------------c
      Call REUV                                  !  compute u v
```

$A_{Pi}\xi^*_i-\sum_{s:i}^{NS}A_{Pis}\xi^*_{is}=\langle\tilde{B}_i\rangle$

核心：$\int_{cv}\xi^*_i dv-\int_{cv}Z_{ii}\left\{\frac{\partial}{\partial x}\left[A^{n-1}B^n\left(\frac{\partial\xi^*}{\partial x}\right)\right]\right\}_i dv-\int_{cv}Z_{2i}\left\{\frac{\partial}{\partial y}\left[A^{n-1}B^n\left(\frac{\partial\xi^*}{\partial y}\right)\right]\right\}_i dv=$

$\int_{cv}\xi^n_i dv+\int_{cv}Z_{ii}\left\{\frac{\partial}{\partial x}\left[A^{n-1}G^n_{xi}\right]\right\}_i dv+\int_{cv}Z_{ii}\left\{\frac{\partial}{\partial y}\left[A^{n-1}G^n_{yi}\right]\right\}_i dv-\int_{cv}Z_{ii}\left(\frac{\partial Q^n_x}{\partial x}\right)_i dv-\int_{cv}Z_{2i}\left(\frac{\partial Q^n_y}{\partial y}\right)_i dv$

```
      Call UVFN                                    !  vel. on sides of cells
c-----------------------------------------------------------------------c
c         update water depths
c-----------------------------------------------------------------------c
        Do 130 I = 1, IJM
           EL(I) = ELF(I)
130     Continue
c-----------------------------------------------------------------------C
c              check whether to go on                                   C
C-----------------------------------------------------------------------C
      UVMAX = 0.0
C!$OMP PARALLEL DO REDUCTION(MAX:UVMAX,IJMAX)
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            UVTEMP = Sqrt(UAVE(I) ** 2. + VAVE(I) ** 2.)
            If(UVTEMP .GT. UVMAX) Then
               UVMAX = UVTEMP
               IJMAX = I
            Endif
               IJNAN = 1    !#check "NAN"#, WangJian, 2020-3-6 19:27:43 !
               IF(ISNAN(ELF(I)))THEN
                  IJNAN = I
                  goto 101
               ENDIF
         Endif
        Enddo
      If(UVMAX .GT. 100.0) GOTO 101
c        Write(*,'(4F8.4)') CXY(IJMAX,1),CXY(IJMAX,2),
c     &                     UAVE(IJMAX),VAVE(IJMAX)
c-----------------------------------------------------------------------C
C          saving                                                       C
C-----------------------------------------------------------------------C
c-----  saving time series of water elevation at fixed observations  -----c
      Call ARCHIVE(DAYI,HOURSTAR)                  !  save in disk
        !#hotstart for averaging#, WangJian, 2020-3-6 19:27:40 !
        IF (IAVESAVE == 1) THEN
            IF (IAVEHOT == 0) THEN
              Call STATISTICS(DAYI,HOURSTAR)
            ELSEIF(IAVEHOT==1 .AND. NSTEP .GT. NSTEPAVE)THEN
              Call STATISTICS(DAYI,HOURSTAR)
            ENDIF
```

```
        ENDIF
c-------------------------------------------------------------------------c
c                   temporal saving for hot start                         c
c-------------------------------------------------------------------------c
      If(MOD(NSTEP,IHOTSTART).EQ.0.0) Then
        WRITE(FN,'(F10.4)') NSTEP*DTI/3600.
        Open(IUSTA,Form='unformatted',File='./Restart/'//FN//'.sta',
     *       Status='unknown')
        Write (IUSTA) NSTEP, DZR, Z,ZZ,DZ, DZZ, HC, HS, HP, DC, DS,
     *       EL, ELFV, ELFM, COR, WTSURF, WTBOT, AAM, KM, KH, DT,
     *       KQ, U, V, W, UR, VR, WR, UV, VV, WV, UN, VN, CBC, VIS,
     *       T, SED, RMEAN, Z01, PN, QZ, CCM, CFM, INDEX_EDGE,
     *       THOUR, TBX, TBY, TKE, TDISS, STRESS,
     *       VIS, TENSOR, VORTICITY,C,QZ,
     *       NNDEM,TTTDEM, XPDEM,YPDEM,ZPDEM,
     *       UPDEM,VPDEM,WPDEM,OMXDEM,OMYDEM,OMZDEM,
     *       PORE,GRADP,
     *       DELTAT_PP1X,DELTAT_PP1Y,DELTAT_PP1Z,N_P2,N_PP2
        Close(IUSTA)
          !#hot restart for averaging#, WangJian, 2020-3-6 19:36:24 !
          IF(IAVESAVE==1)THEN
          Open(IAVESTA,Form='unformatted',File='./Restart/AVE.STA',
     *       Status='unknown')
        Write (IAVESTA) NSTEP-1, AVEURV, AVEVRV,AVEWRV, AVEUTRV,
     &                  RYNUU, RYNUV, RYNUW, RYNVV, RYNVW,RYNWW
        Close(IAVESTA)
          ENDIF
      Endif
c=========================================================================c
100                 Continue
c*************************************************************************
c                                                                        *
c                end numerical simulation                                *
c                                                                        *
c*************************************************************************

1001  WRITE(*,*) 'Job successfully completed !'
```