```fortran
C##########################################################################
c                                                                         #
c                    SUBROUTINE PROGRAM                                   #
C                  VERSION 1.0 (28/07/2009)                               #
C                  AUTHORIZED BY ZHANG JINGXIN                            #
C                                SHANGHAI JIAO TONG UNIVERSITY            #
C                                SHANGHAI, CHINA                          #
c-------------------------------------------------------------------------#
c                      computes the hydrodynamic pressure                 #
c                                                                         #
c##########################################################################
    Subroutine DYN
      Include './Include/OCERM_INF'
      Include './Include/VORGEN_INF'

    Common/DYNBLK/AS(IJM,KB,IPOLYGEN),AB(IJM,KB),AT(IJM,KB),
     &             AP(IJM,KB),BB(IJM,KB),X(IJM,KB),XINI(IJM,KB)
c      Parameter(WB_ANGLE_I = 2.*3.14*30./360.,
c     &           WB_ANGLE_E = 2.*3.14*10./360.)
C      Common/SOURCE/BB(IJM,KB)
    Dimension UT(IJM,KB),VT(IJM,KB),WT(IJM,KB),ET(IJM,KB),
     &          WW(IJM,KB),WGENDEL(N_SOURCE,KB),FLUX(IJM,KB),QZT(IJM,KB)
    Dimension TEMP(IJM,KB),ELFX(IJM),ELFY(IJM),WB_MASK2(IJM),
     &             HX(IJM),HY(IJM),WTT(KB,2)
      Dimension PP(KBM),PV(KBM),DZP(KBM)
        Dimension FLU_SOURCE(NUM_CELL,NUM_VER)

      IIII = 0
```

$$\frac{q_{xi}^{n+1} - q_{xi}^{*}}{\Delta t} = -\frac{D}{P_0}\left(\frac{\partial p_n^{n+1}}{\partial x}\right)_i$$

```fortran
      IJM_B = IJM_DYN_B + IIII
      IJM_E = IJM_DYN_E + IIII
c========================================================================c
C              initializing the arrays                                   c
c========================================================================c
!$OMP PARALLEL DEFAULT(SHARED)
!$OMP&         PRIVATE(I,J,K,IR,IL,PV,DY0,DY1,DZP,UNEW,VNEW,
!$OMP&                 P1U,P2U,P1V,P2V,PUSUR,PVSUR,ZSIGMA)
    Do K = 1, KB
!$OMP DO
       Do I = 1, IJM
          Do J = 1, CELL_POLYGEN(I)
```

```fortran
             AS(I,K,J) = 0.0
          Enddo
            AB(I,K) = 0.0
            AT(I,K) = 0.0
            AP(I,K) = 1.0
            BB(I,K) = 0.0
            X(I,K)  = 0.0
            XINI(I,K) = PN(I,K)
C         PT(I,K) = 0.0
          UT(I,K) = 0.0
          VT(I,K) = 0.0
          WT(I,K) = 0.0
          QZT(I,K) = 0.0
          FLUX(I,K) = 0.0
       Enddo
!$OMP END DO
      Enddo
!$OMP DO
    Do I = IJM_B, IJM_E
       If(CCM(I) .EQ. 1.0) Then
          ELFX(I) = 0.0
          ELFY(I) = 0.0
C         ELFXNO(I) = 0.0
C         ELFYNO(I) = 0.0
          HX(I) = 0.0
          HY(I) = 0.0
C         HXNO(I) = 0.0
C         HYNO(I) = 0.0
            WB_MASK2(I) = 0.0
            Do J = 1, CELL_POLYGEN(I)
               HX(I) = HX(I) +
     &               HS(CELL_SIDE(I,J,1)) *
     &               CELL_CUV(I,J,7) * CELL_CUV(I,J,6)
             HY(I) = HY(I) +
     &               HS(CELL_SIDE(I,J,1)) *
     &               CELL_CUV(I,J,8) * CELL_CUV(I,J,6)
               If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
                  ELFX(I) = ELFX(I) +
     &                  WIX(I,J) * (ELF(CELL_SIDE(I,J,2)) - ELF(I))
                ELFY(I) = ELFY(I) +
     &                  WIY(I,J) * (ELF(CELL_SIDE(I,J,2)) - ELF(I))
```

```
C                 ELFXNO(I) = ELFXNO(I) +
C     &                 WIX(I,J) * (EL(CELL_SIDE(I,J,2)) - EL(I))
C               ELFYNO(I) = ELFYNO(I) +
C     &                 WIY(I,J) * (EL(CELL_SIDE(I,J,2)) - EL(I))

             Endif
            Enddo
            HX(I) = HX(I) / AREA(I)
            HY(I) = HY(I) / AREA(I)
          Endif
     Enddo
!$OMP END DO
C--------  TEMPORARY VELOCITIES
c!$OMP DO
c      Do I = 1, IJM
c         If(CCM(I) .EQ. 1.0) Then
c            Do K = 1, KBM
c               UNEW = 0.0
c            VNEW = 0.0
c            Do J = 1, CELL_POLYGEN(I)
c               If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
c                    UNEW = UNEW +
c     &                    WIX(I,J) * (PN(CELL_SIDE(I,J,2),K) - PN(I,K))
c                     VNEW = VNEW +
c     &                    WIY(I,J) * (PN(CELL_SIDE(I,J,2),K) - PN(I,K))
c               Endif
c            Enddo
c              UT(I,K) = U(I,K) - DTI * DC(I) / RMEAN(I,K) * UNEW
c               VT(I,K) = V(I,K) - DTI * DC(I) / RMEAN(I,K) * VNEW
c            Enddo
c         Endif
c      Enddo
c!$OMP END DO
!$OMP DO
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            Do K = 2, KBM
               TEMP(I,K) = (PN(I,K-1) - PN(I,K)) / DZZ(K-1)
            Enddo
            TEMP(I,1) = (0.0 - PN(I,1)) / (0.5 * DZ(1))
C           temp(i,1) = 0.0
```

$$\frac{P_{n,k-1} - P_{n,k}}{\Delta G} = \rho_0 \cdot \frac{q_{zi}^{n+1} - q_{zi}^{*}}{\Delta t}$$

q^*

```
            TEMP(I,KB) = 0.0
            Do K = 1, KBM
```

$\Delta t \cdot \frac{p_n^*}{\rho}[(1+\sigma)\cdot \eta^* + \sigma h]$

```
               UT(I,K) = DTI * (TEMP(I,K)+TEMP(I,K+1))/2./ RMEAN(I,K) *
     &                    ((1. + ZZ(K)) * ELFX(I) + ZZ(K) * HX(I))
               VT(I,K) = DTI * (TEMP(I,K)+TEMP(I,K+1))/2./ RMEAN(I,K) *
     &                    ((1. + ZZ(K)) * ELFY(I) + ZZ(K) * HY(I))
            Enddo
         Endif
      Enddo
!$OMP END DO
!$OMP DO
      Do I = IJM_B, IJM_E
         If(CCM(I) .EQ. 1.0) Then
```

$q_z^* - [(1+\sigma)\eta^* \cdot q_x^* + \sigma h q_x^*] - [(1+\sigma)\eta^* q_y^* + \sigma h q_y^*]$

```
            Do K = 2, KBM
               TEMP(I,K) = QZ(I,K) * PORE_VF(I,K) -
     &                      (U(I,K-1) + U(I,K)) / 2. * PORE_VF(I,K) *
     &                     ((1.+Z(K)) * ELFX(I) + Z(K) * HX(I)) -
     &                      (V(I,K-1) + V(I,K)) / 2. * PORE_VF(I,K)*
     &                     ((1.+Z(K)) * ELFY(I) + Z(K) * HY(I))
C               TEMP(I,K) = TEMP(I,K) -
C     &                       .5 * (UR(I,K-1) + UR(I,K)) * DC(I) *
C     &                      ((1.+Z(K)) * ELFXNO(I) + Z(K) * HXNO(I)) -
C     &                       .5 * (VR(I,K-1) + VR(I,K)) * DC(I) *
C     &                      ((1.+Z(K)) * ELFYNO(I) + Z(K) * HYNO(I)) -
C     &                       DC(I) *
C     &                     ((1.+Z(K)) * (ELF(I) - EL(I)) / DTI)
            Enddo
CC            TEMP(I,1) = QZ(I,1) - U(I,1) * ELFX(I) - V(I,1) * ELFY(I)
C     &                    DC(I) * (ELF(I) - EL(I)) / DTI
            TEMP(I,1) = 0.0
            TEMP(I,KB) = 0.0
         Endif
      Enddo
!$OMP END DO
c============================================================================c
c                   arrays for the hydrodynamic pressure based on the        c
c          continuity equation: deta/dt+dQx/dx+dQy/dy+dw/dz=0                c
c============================================================================c
    Do K = 1, KBM
!$OMP DO
       Do I = IJM_B, IJM_E
```

C.E. $\frac{\partial \zeta}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_\sigma}{\partial \sigma} = 0$

$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{D\partial \sigma} - \frac{\partial}{D\partial \sigma}\left[\frac{\partial \zeta}{\partial x}(1+\sigma) q_x + \frac{\partial \zeta}{\partial y}(1+\sigma) q_y + \frac{\partial h}{\partial x}\sigma q_x + \frac{\partial h}{\partial y}\sigma q_y\right] = 0$

$$\Delta t \Delta \sigma_k \frac{D}{\rho_0} \cdot \frac{\Delta l_{is}}{J_{is} \Delta \xi_{is}} (\cos\alpha_{is} y_\eta - \sin\alpha_{is} x_\eta)$$

```fortran
        If(CCM(I) .EQ. 1.0) Then
          AP(I,K) = 0.0
          Do J = 1, CELL_POLYGEN(I)
             If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
                AS(I,K,J) = DTI * DZ(K) * DS(CELL_SIDE(I,J,1)) /
     &                 (.5 * (RMEAN(I,K) + RMEAN(CELL_SIDE(I,J,2),K)))*
     &                 (DISCOE(I,J,1) - DISCOE(I,J,8))*
     &                  PORE_HF(CELL_SIDE(I,J,1),K)

               AP(I,K) = AP(I,K) + AS(I,K,J)

                IL = I
               IR = CELL_SIDE(I,J,2)

c              BB(I,K) = BB(I,K) + THITA * DZ(K) * 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)) + DZ(K) *
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))

C                 BB(I,K) = BB(I,K) + DZ(K) * CELL_CUV(I,J,6) *
C     &                 ((U(IL,K) * Sqrt(AREA(IL)) +
C     &                    U(IR,K) * Sqrt(AREA(IR)))/
C     &                    (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
C     &                    CELL_CUV(I,J,7) +
C     &                   (V(IL,K) * Sqrt(AREA(IL)) +
C     &                    V(IR,K) * Sqrt(AREA(IR))) /
C     &                    (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
C     &                    CELL_CUV(I,J,8))

                  BB(I,K) = BB(I,K) + DZ(K) * CELL_CUV(I,J,6) *
     &                       PORE_HF(CELL_SIDE(I,J,1),K) *
     &                  ((U(IL,K) + U(IR,K) + UT(IL,K) + UT(IR,K)) / 2.*
     &                     CELL_CUV(I,J,7) +
     &              (V(IL,K) + V(IR,K) + VT(IL,K) + VT(IR,K))/2.*
     &                     CELL_CUV(I,J,8))
```

显式项集合

```
c                    BB(I,K) = BB(I,K) + DTI*DZ(K)*DS(CELL_SIDE(I,J,1)) /
c      &                  (.5 * (RMEAN(I,K) + RMEAN(CELL_SIDE(I,J,2),K))*
c      &                  (DISCOE(I,J,2) - DISCOE(I,J,7)) *
c      &              (PNV(CELL_SIDE(I,J,4),K) - PNV(CELL_SIDE(I,J,3),K))
                   Else
CC                     BB(I,K) = BB(I,K) + DTI*DZ(K)*DS(CELL_SIDE(I,J,1)) /
CC      &                  RMEAN(I,K)*
CC      &                  (DISCOE(I,J,2) - DISCOE(I,J,7)) *
CC      &              (PNV(CELL_SIDE(I,J,4),K) - PNV(CELL_SIDE(I,J,3),K))
C                  BB(I,K) = BB(I,K) + THITA * DZ(K) * CELL_CUV(I,J,6) *
C      &                  ((U(IL,K) * Sqrt(AREA(IR)) +
C      &                     U(IR,K) * Sqrt(AREA(IL)))/
C      &                     (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
C      &                     CELL_CUV(I,J,7) +
C      &                    (V(IL,K) * Sqrt(AREA(IR)) +
C      &                     V(IR,K) * Sqrt(AREA(IL))) /
C      &                     (Sqrt(AREA(IL)) + Sqrt(AREA(IR))) *
C      &                     CELL_CUV(I,J,8)) +
C      &                    (1. - THITA) * DZ(K) * 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
CC           BB(I,K) = BB(I,K) + AREA(I) / DC(I) *
CC      &                  (TEMP(I,K) - TEMP(I,K+1))
           Endif
         Enddo
!$OMP END DO
     Enddo
!$OMP BARRIER
!$OMP DO
```

$$AP^{T}_{i,k} = \frac{\Delta t \Delta S_i}{D\rho_0 \Delta \sigma_{k-1/2}}$$

```
     Do I = IJM_B, IJM_E
       If(CCM(I) .EQ. 1.0) Then
           Do K = 2, KBM - 1
             AT(I,K) = DTI * AREA(I) / (HC(I)+ELF(I)) / DZZ(K-1) /
     &                    RMEAN(I,K) * PORE_VF(I,K)
             AB(I,K) = DTI * AREA(I) / (HC(I)+ELF(I)) / DZZ(K) /
     &                    RMEAN(I,K) * PORE_VF(I,K+1)
```

$$AP^{B}_{i,k} = \frac{\Delta t \Delta S_i}{D\rho_0 \Delta \sigma_{k+1/2}}$$

```
             AP(I,K) = AP(I,K) + AT(I,K) + AB(I,K)
```

$$AP_{i,k} = \sum_{s=1}^{NS} AP^{s}_{i,k} + AP^{T}_{i,k} + AP^{B}_{i,k}$$

$AP_{i,k}\, p_n{}^{n+1}_{i,k} - AP^{T}_{i,k}\, p_n{}^{n+1}_{i,k-1} - AP^{B}_{i,k}\, p_n{}^{n+1}_{i,k+1} - \sum AP^{S}_{i,k}\, p_n{}^{n+1}_{i,k} = \widetilde{BP}_{i,k}$

```
             BB(I,K) = BB(I,K) + AREA(I) / DC(I) *
     &                  (TEMP(I,K) - TEMP(I,K+1))
          Enddo
C-----   surface layer  ----------------------------------------------------c
C        AT(I,1) = DTI * AREA(I) / (HC(I)+ELF(I)) / DZ(1) /RMEAN(I,1)
           AT(I,1) = 0.0
         AB(I,1) = DTI * AREA(I) / (HC(I)+ELF(I)) / DZZ(1) / RMEAN(I,1) *
     &                 PORE(I,1)
         AP(I,1) = AP(I,1) + AT(I,1) + AB(I,1)
     &               + AREA(I) / RMEAN(I,1) / GRAV / DTI
C           BB(I,1) = BB(I,1) + AREA(I) / DC(I) *
C    &                 (TEMP(I,1) - TEMP(I,2))
         BB(I,1) = BB(I,1) + AREA(I) * (ELF(I) - EL(I)) / DTI -
     &                          AREA(I) / DC(I) * TEMP(I,2)
C        BB(I,1) = BB(I,1) + AREA(I) * WT(I,1) * DZ(1)
c-----   bottom layer  ----------------------------------------------------c
         AT(I,KBM) = DTI * AREA(I) / (HC(I)+ELF(I))/DZZ(KBM-1)/RMEAN(I,KBM)
     &                 * PORE(I,KBM)
         AB(I,KBM) = 0.0
         AP(I,KBM) = AP(I,KBM) + AT(I,KBM) + AB(I,KBM)
         BB(I,KBM) = BB(I,KBM) + AREA(I) / DC(I) * TEMP(I,KBM)
C        BB(I,KBM) = BB(I,KBM) + AREA(I) * WT(I,KBM) * DZ(KBM)
       Endif
    Enddo
!$OMP END DO
!$OMP DO
      Do I = IJM_B, IJM_E
         If(CCM(I) .EQ. 1.0) Then
            Do K = 1, KBM
               BB(I,K) = -BB(I,K)
            Enddo
         Endif
      Enddo
!$OMP END DO
c-----   implementing the source terms
C!$OMP DO
C      Do I = 1, IJM
C      If(CCM(I) .EQ. 1.0) Then
C           Do K = 1, KBM
C            BB(I,K) = -(BB(I,K) + AREA(I) * DZ(K) *
C    &                  (ELF(I) - EL(I)) / DTI)
```

```fortran
C          Enddo
C       Endif
C    Enddo
C!$OMP END DO
!$OMP END PARALLEL
C============================================================================C
C                 boundary condition                                        c
c============================================================================c
c-----    discharge boundary conditions
    If(NUMQBC .NE. 0) Then
       Do N = 1, NUMQBC
            ID = IQBC(N)
          IS = IQBCINX(N)
          Do K = 1, KBM
             Do J = 1, CELL_POLYGEN(ID)
                  If(CFM(CELL_SIDE(ID,J,1)) .EQ. 1.0) Then
                     AS(ID,K,J) = 0.0
                Endif
               Enddo
               AP(ID,K) = 1.0
             AB(ID,K) = 0.0
             AT(ID,K) = 0.0
             BB(ID,K) = 0.0

CC           BB(ID,K) = BB(ID,K) + QDIS(N) * VQDIST(N,K) / 100. * RAMP
c            BB(ID,K) = BB(ID,K) - DZ(K) * CELL_CUV(ID,IS,6) *
c     &                DS(CELL_SIDE(ID,IS,1)) *
c     &                (UN(CELL_SIDE(ID,IS,1),K) * CELL_CUV(ID,IS,7) +
c     &                    VN(CELL_SIDE(ID,IS,1),K) * CELL_CUV(ID,IS,8))
cc           BB(ID,K) = 0.0
          Enddo
       Enddo
    Endif
c-----    velocity boundary conditions
    If(NUMVBC .NE. 0) Then
       Do N = 1, NUMVBC
          ID = IVBC(N)
          IS = IVBCINX(N)
          Do K = 1, KBM
             BB(ID,K) = BB(ID,K) - DZ(K) * CELL_CUV(ID,IS,6) *
     &                DS(CELL_SIDE(ID,IS,1)) *
```

```fortran
     &                (UN(CELL_SIDE(ID,IS,1),K) * CELL_CUV(ID,IS,7) +
     &                   VN(CELL_SIDE(ID,IS,1),K) * CELL_CUV(ID,IS,8))
          Enddo
       Enddo
    Endif
c-----    elevation
    If(NUMEBC .NE. 0) Then
      Do N = 1, NUMEBC
         ID = IEBC(N)
         IS = IEBCINX(N)
         Do K = 1, KBM
C           UNEBC = UR(ID,K) * CELL_CUV(ID,IS,7) +
C     &                 VR(ID,K) * CELL_CUV(ID,IS,8)
C           If(UNEBC .LE. 0.0) Then
             Do J = 1, CELL_POLYGEN(ID)
                If(CFM(CELL_SIDE(ID,J,1)) .EQ. 1.0) Then
                 AS(ID,K,J) = 0.0
              Endif
             Enddo
             AP(ID,K) = 1.0
           AB(ID,K) = 0.0
           AT(ID,K) = 0.0
           BB(ID,K) = 0.0
C           Else
C                 BB(ID,K) = BB(ID,K) - DZ(K) * CELL_CUV(ID,IS,6) *
C     &                     (U(ID,K) * CELL_CUV(ID,IS,7) +
C     &                           V(ID,K) * CELL_CUV(ID,IS,8))
C           Endif
           Enddo
      Enddo
    Endif
c-----    astrotide boundary
    If(NUMAST .NE. 0) Then
      Do N = 1, NUMAST
         ID = IABC(N)
         IS = IABCINX(N)
           Do K = 1, KBM
              BB(ID,K) = BB(ID,K) - DZ(K) * CELL_CUV(ID,IS,6) *
     &                 (U(ID,K) * CELL_CUV(ID,IS,7) +
     &                  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
       Do I = 1, N_SOURCE
          Do K = 1, KBM
             WGENDEL(I,K) = WGEN(I,K)
          Enddo
       Enddo
       Call WAVEGEN(1)
       Do I = 1, N_SOURCE
          II = IGEN(I)
          Do K = 1, KBM
             BB(II,K) = BB(II,K) +
     &                  DZ(K) * WGEN(I,K) * AREA(II)
            ENDDO
       Enddo
    Endif
C----------------------------------------------------------------------C
C                Fluctuation generating                                 c
C----------------------------------------------------------------------C
    If(DES .EQ. 'SAZDES1') Then
         Do I = 1, NUM_CELL
            ID = ID_CELL(I)
            IS = ID_CELL_EDGE(I)
            Do K = 1, NUM_VER - 1
               FLU_SOURCE(I,K) = DS(CELL_SIDE(ID,IS,1)) * DZ(K) *
     &              CELL_CUV(ID,IS,6) *
     &             (UDIS(I,K) * CELL_CUV(ID,IS,7) +
     &              VDIS(I,K) * CELL_CUV(ID,IS,8)) +
     &              AREA(ID) * (WDIS(I,K) - WDIS(I,K+1))
               BB(ID,K) = BB(ID,K) + 2. * FLU_SOURCE(I,K) +
     &                    (WDIS(I,K) - WDIS(I,K+1)) * AREA(ID)
            Enddo
         Enddo
      Endif
c----------------------------------------------------------------------c
c            surface boundary condition                                 c
c----------------------------------------------------------------------c
```

```
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,WB_C)
!$OMP DO
    Do I = IJM_B, IJM_E
       If(CCM(I) .EQ. 1.0) Then
          Do J = 1, CELL_POLYGEN(I)
             If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
C                AS(I,1,J) = 0.0
              Endif
          Enddo
C         AP(I,1) = 1.0
C         AB(I,1) = 0.0
C         AT(I,1) = 0.0
C         BB(I,1) = 0.0
C            BB(I,1) = -DC(I) * RMEAN(I,1) * DZ(1) / 2. * W(I,1) / DTI
       Endif
    Enddo
!$OMP END DO
!$OMP END PARALLEL
c============================================================================c
c       Switch from Non-hydrostatic model to NSE                             c
c       in order to model the wave breaking                                  c
c============================================================================c
      If(WAVE_BREAKING .EQ. 1.0) Then
         Call WAVEBREAKING
      Endif
C============================================================================C
C        solving the equation by Bi-CGSTAB method                          c
c============================================================================c
C      Call SMOOTHING(BB)
C!$    begin1 = OMP_GET_WTIME()
    Call SOLVE3DPOLCG        双共轭梯度法求 Pn
C   Call SOLVEDYNICCG
C!$    END1 = OMP_GET_WTIME()
C      PRINT*, END1-BEGIN1
C      STOP
C   Call SOLVEDYNICBICG
C   Call SOLVEDYNICCG
ccc Call SOLVEDYNPRESOR
CCC Call SOLVEDYNSORCG
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I,K)
    Do I = IJM_B, IJM_E
```

```
        Do K = 1, KBM
           PN(I,K) = X(I,K)
        Enddo
     Enddo
!$OMP END PARALLEL DO
C----   SMOOTHING THE PN            -----------------------------------------C
C       If(Mod(NSTEP,10) .EQ. 0.0 ) Then
c        Call SMOOTHINGNOR(PN)
C       Endif
C========================================================================C
C           calculate the velocity at time step n+1                       c
C========================================================================C
!$OMP PARALLEL DEFAULT(SHARED)
!$OMP&          PRIVATE(I,J,K,KK,II,UNEW,VNEW,ELTEMP,
!$OMP&                  ZK,ZKU,ZKD,PNSUR,ZSIGMA,Z1,Z2,Z3,P1,P2,P3,
!$OMP&                  PV,DY0,DY1,DZP,IL,IR,WB_C)

     Do K = 1, KBM
!$OMP DO
        Do I = IJM_B, IJM_E
           If(CCM(I) .EQ. 1.0) Then
              UNEW = 0.0
              VNEW = 0.0
              Do J = 1, CELL_POLYGEN(I)
c               If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
c                    UNEW = UNEW +
c      &                 WIX(I,J) * (PN(CELL_SIDE(I,J,2),K) - PN(I,K))
c                     VNEW = VNEW +
c      &                 WIY(I,J) * (PN(CELL_SIDE(I,J,2),K) - PN(I,K))
c               Endif

                 If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
                    UNEW = UNEW + (PN(I,K)+PN(CELL_SIDE(I,J,2),K))/2. *
     &                       CELL_CUV(I,J,7) * CELL_CUV(I,J,6)
                    VNEW = VNEW + (PN(I,K)+PN(CELL_SIDE(I,J,2),K))/2. *
     &                       CELL_CUV(I,J,8) * CELL_CUV(I,J,6)
                  Else
                    UNEW = UNEW + PN(I,K)*CELL_CUV(I,J,7)*CELL_CUV(I,J,6)
                    VNEW = VNEW + PN(I,K)*CELL_CUV(I,J,8)*CELL_CUV(I,J,6)
                  Endif
              Enddo
```

$$\int_{cv} \frac{\partial P_n^{n+1}}{\partial x} dv = \int_{cs} n_x \cdot P_n^{n+1} ds$$

$$= \cos\alpha \cdot P_n^{n+1} \Delta l \cdot \cancel{\Delta\theta}$$

$$\frac{\Delta\theta}{\Delta V} = \frac{1}{\Delta S}$$

$$q_{xi}^{n+1} = q_{xi}^{*} - \frac{\Delta t \cdot D}{\rho_0}\left(\frac{\partial p_n^{n+1}}{\partial x}\right)_i$$

```
              UT(I,K) = - DTI*DC(I) / RMEAN(I,K) * UNEW / AREA(I) + UT(I,K)

              VT(I,K) = - DTI*DC(I) / RMEAN(I,K) * VNEW / AREA(I) + VT(I,K)

           Endif
        Enddo
!$OMP END DO
     Enddo
!$OMP DO
     Do I = IJM_B, IJM_E
        If(CCM(I) .EQ. 1.0) Then
           Do K = 2, KBM
               QZT(I,K) = - DTI / (0.5 *(RMEAN(I,K) + RMEAN(I,K-1)) *
     &              (PN(I,K-1) - PN(I,K)) / DZZ(K-1)
           Enddo
           QZT(I,1) = - DTI / RMEAN(I,1) *
     &                 (0.0 - PN(I,1)) / (.5 * DZ(1))
C             QZT(I,1) = 0.0
             QZT(I,KB) = 0.0
C             QZT(I,1) = UT(I,1) * ELFX(I) + VT(I,1) * ELFY(I)
        Endif
      Enddo
!$OMP END DO
C-----------------------------------------------------------------------------C
C              variables at time step n+1                                     c
c-----------------------------------------------------------------------------c
!$OMP DO
     Do I = IJM_B, IJM_E
        If(CCM(I) .EQ. 1.0) Then
C          ELF(I) = ELF(I) + PN(I,1) / RMEAN(I,1) / GRAV
C            WB_C = 0.6 * Sqrt(9.8 * HC(I))
C             If((ELF(I) - EL(I)) / DTI .LT. WB_C) Then

              Do K = 1, KB
      qx^n+1    U(I,K) = U(I,K) + UT(I,K)
      qy^n+1    V(I,K) = V(I,K) + VT(I,K)
      qz^n+1    QZ(I,K) = QZ(I,K) + QZT(I,K)
C                 PN(I,K) = PN(I,K) - PN(I,1)
              Enddo
C          Endif
```

```
      Endif
   Enddo
!$OMP END DO
!$OMP DO
   Do I = IJM_B, IJM_E
      If(CCM(I) .EQ. 1.0) Then
         ELF(I) = ELF(I) + QZT(I,1) / DC(I) * DTI
      Endif
   Enddo
!$OMP END DO

1000   Continue
   Return
   End
```