```
C###############################################################################
c                                                                              #
c                    SUBROUTINE PROGRAM                                        #
C                  VERSION 1.0 (11/01/2010)                                    #
C                  AUTHORIZED BY ZHANG JINGXIN                                 #
C                               SHANGHAI JIAO TONG UNIVERSITY                  #
C                               SHANGHAI, CHINA                                #
c------------------------------------------------------------------------------#
c                      specifying the boundary condition at each step          #
c                                                                              #
c###############################################################################

      Subroutine BCOND(IDX)
      Include './Include/OCERM_INF'
     Parameter(UMEAN = 1.0, GVARIANCE = 1.0, TINT = 0.00)
c=============================================================================c
c          astrotidle frequency:   s2     m2     n2     k1     p1     o1     c
c=============================================================================c
      Data PERIOD /43200., 44712., 45570., 86164., 86637., 92950.,
     &           0.,0.,0.,0.,0./
      Data PI2 /6.283185307/
c=============================================================================c
c              idx identifies which variables are considered                  c
c              1=u,v and surface elevation                                    c
c              2=open bc's for u and v                                        c
c              3=internal mode u,v                                            c
c              4=temp,sal for open b.c.s                                      c
c              5=w velocity                                                   c
c              6= km,kh,q2,q2l,l                                              c
c              7=surface forcing and temporal cycling                         c
c=============================================================================c

      Go To (10, 20, 30, 40, 50, 60, 70, 80, 90, 1000), IDX
                                                          1-10
C=============================================================================c
C                   Flow module boundary conditions                           c
C=============================================================================c
c-----------------------------------------------------------------------------c
c        Elevation boundary condition                                          c
c-----------------------------------------------------------------------------c
10    Continue
```

```
      Do N = 1, NUMEBC
         ID = IEBC(N)
         ELF(ID) = EBDRY(N) * RAMP
      Enddo
      Return
c-----------------------------------------------------------------------c
c        Velocity boundary condition                                    c
c-----------------------------------------------------------------------c
20    Continue
      Do N = 1, NUMVBC
         ID = IVBC(N)
         IS = IVBCINX(N)
         ISS = CELL_SIDE(ID,IS,1)
         Do K = 1, KBM
            UN(ISS,K) = UBDRY(N) * VVDIST(N,K) * RAMP
            VN(ISS,K) = VBDRY(N) * VVDIST(N,K) * RAMP
         Enddo
      Enddo
      Return
c-----------------------------------------------------------------------c
c        Discharge / onshore boundary condition                         c
c-----------------------------------------------------------------------c
30    Continue
      Do N = 1, NUMQBC
         ID = IQBC(N)
         IS = IQBCINX(N)
         ISS = CELL_SIDE(ID,IS,1)
         Do K = 1, KBM
C     if(mod(nstep,10).eq. 0.) then
C          Call Random_number(RSEED)
C            Call NGRN1(UMEAN,GVARIANCE,RSEED,REDDY)
C     endif
            FRESH = -QDIS(N) * VQDIST(N,K) / 100. * RAMP
     &
c           FRESH = -QDIS(N) * VQDIST(N,K) / 100. * RAMP
            UN(ISS,K) = FRESH * CELL_CUV(ID,IS,7) /
     &                  CELL_CUV(ID,IS,6) / DZ(K) / DS(ISS)
            VN(ISS,K) = FRESH * CELL_CUV(ID,IS,8) /
     &                  CELL_CUV(ID,IS,6) / DZ(K) / DS(ISS)
          Enddo
```

```
    Enddo
c-------------------------------------------------------------------------c
C        Discharge / offshore boundary condition                          c
c-------------------------------------------------------------------------c
40  Continue
    Do N = 1, NUMDBC
       ID = IDBC(N)
       Do K = 1, KBM
          FRESH = QDIFF(N) * VDDIST(N,K) / 100. * RAMP
          UF(ID,K) = UF(ID,K) + DTI * FRESH * QU(N)
          VF(ID,K) = VF(ID,K) + DTI * FRESH * QV(N)
       Enddo
    Enddo
    Return
C-----   Astrotide boundary condition
50  Continue
c      Call GETCOR(IYR,IMO,IDA,AMP0,PHASE0,FREQ,11)
    Do N = 1, NUMAST
         ID = IABC(N)
       FORCE = 0.0
         Do I = 1, IAST
          FORCE = FORCE + AMP0(I) * AMP(N,I) *
     &      COS((FREQ(I) * THOUR - PHASE(N,I)) * PI2 / 360. + PHASE0(I))
         Enddo
       FORCE = FORCE + EMEAN(N)
         ELF(ID) = FORCE * RAMP
    Enddo
    Return
C=========================================================================c
C                  Sediment module boundary conditions                    c
C=========================================================================c
60  Continue


    Return
C=========================================================================c
C                  Salinity module boundary conditions                    c
C=========================================================================c
70  Continue
```

```
      Return
C=====================================================================c
C                    Temperature module boundary conditions           c
C=====================================================================c
80    Continue


      Return
C=====================================================================c
C                    Wave module boundary conditions                  c
C=====================================================================c
90    Continue


      Return
c=====================================================================c
c                    Variables on the boundary cells                  c
c=====================================================================c
1000    Continue
C---------------------------------------------------------------------C
C         Read elevation boundary condition at new time step           c
c---------------------------------------------------------------------c
      If (NUMEBC .NE. 0) Then
        If (THOUR .GE. T2E) Then   如果计算时间超过9999s
          T1E = T2E
          Do N = 1, NUMEBC
            DEBDRY(N,1) = DEBDRY(N,2)
          Enddo                    中间步存储文件
          Read (IUT90,5000,End=420) T2E
          Read (IUT90,5000) (DEBDRY(N,2), N = 1, NUMEBC)
        End If                          0 from IUT90
        FACT = (THOUR-T1E) / (T2E-T1E)
        Do N = 1, NUMEBC
          EBDRY(N) = DEBDRY(N,1) + FACT * (DEBDRY(N,2) - DEBDRY(N,1))
      Enddo    0   出口水位边界条件 from oute.EBC
      End If
C---------------------------------------------------------------------C
C         Read velocity boundary condition at new time step            c
c---------------------------------------------------------------------c
      If (NUMVBC .NE. 0) Then    NUMVBC=0
        If (THOUR .GE. T2V) Then
```

```
            T1V = T2V
            Do N = 1, NUMVBC
               DUBDRY(N,1) = DUBDRY(N,2)
               DVBDRY(N,1) = DVBDRY(N,2)
            Enddo
            Read (IUT91,5000,End=420) T2V
            Read (IUT91,5000) (DUBDRY(N,2), N = 1, NUMVBC)
            Read (IUT91,5000) (DVBDRY(N,2), N = 1, NUMVBC)
         End If
         FACT = (THOUR-T1V) / (T2V-T1V)
         Do N = 1, NUMVBC
            UBDRY(N) = DUBDRY(N,1) + FACT * (DUBDRY(N,2) - DUBDRY(N,1))
            VBDRY(N) = DVBDRY(N,1) + FACT * (DVBDRY(N,2) - DVBDRY(N,1))
       Enddo
      End If
C-----------------------------------------------------------------------C
C         Read discharge / onshore boundary condition at new time step      c
c-----------------------------------------------------------------------c
      If (NUMQBC .NE. 0) Then
         If (THOUR .GE. T2Q) Then
            T1Q = T2Q
            Do N = 1, NUMQBC
               DQDIS(N,1) = DQDIS(N,2)
            Enddo
            Read (IUT92,5000,End=420) T2Q
            Read (IUT92,5000) (DQDIS(N,2), N = 1, NUMQBC)
         End If
         FACT = (THOUR-T1Q) / (T2Q-T1Q)
         Do N = 1, NUMQBC
            QDIS(N) = DQDIS(N,1) + FACT * (DQDIS(N,2) - DQDIS(N,1))
       Enddo
      End If
C-----------------------------------------------------------------------C
C         Read discharge / offshore boundary condition at new time step     c
c-----------------------------------------------------------------------c
      If (NUMDBC .NE. 0) Then
         If (THOUR .GE. T2D) Then
            T1D = T2D
            Do N = 1, NUMDBC
               DQDIFF(N,1) = DQDIFF(N,2)
               DQU(N,1) = DQU(N,2)
```

VQDIST (层流量比例)

from IUT92

入口流量边界 from infl.dat

```
            DQV(N,1) = DQV(N,2)
          Enddo
          Read (IUT93,5000,End=420) T2D
          Read (IUT93,5000) (DQDIFF(N,2), N = 1, NUMDBC)
          Read (IUT93,5100) (DQU(N,2), N = 1, NUMDBC)
          Read (IUT93,5100) (DQV(N,2), N = 1, NUMDBC)
        End If
        FACT = (THOUR-T1D) / (T2D-T1D)
        Do N = 1, NUMDBC
          QDIFF(N) = DQDIFF(N,1) + FACT * (DQDIFF(N,2) - DQDIFF(N,1))
          QU(N) = DQU(N,1) + FACT * (DQU(N,2) - DQU(N,1))
          QV(N) = DQV(N,1) + FACT * (DQV(N,2) - DQV(N,1))
       Enddo
      End If
C-------------------------------------------------------------------------------C
C         Read viscosity / onshore boundary condition at new time step      c
c-------------------------------------------------------------------------------c
C----- One equation (SA)
    If (VERTMIX .EQ. 'SAMODEL    ' .OR.
     *    DES .EQ. 'SADES  ' .OR. DES .EQ. 'SADDES ' .OR.
     *    DES .EQ. 'SAIDDES' .OR. DES .EQ. 'SAZDES ') Then
        If (NUMEBC .NE. 0) Then
          If (THOUR .GE. T2VISE) Then
            T1VISE = T2VISE
            Do N = 1, NUMEBC
              VISEBDRY(N,1) = VISEBDRY(N,2)
            Enddo
            Read (IUT96,5000,End=420) T2VISE
            Read (IUT96,5000) (VISEBDRY(N,2), N = 1, NUMEBC)
           End If
           FACT = (THOUR - T1VISE) / (T2VISE - T1VISE)
           Do N = 1, NUMEBC
              VISE(N) = VISEBDRY(N,1) +
     &                  FACT * (VISEBDRY(N,2) - VISEBDRY(N,1))
         Enddo
        End If
        If (NUMQBC .NE. 0) Then
          If (THOUR .GE. T2VISQ) Then
            T1VISQ = T2VISQ
            Do N = 1, NUMQBC
              Do K = 1, KBM
```

```
                    VISQBDRY(N,1,K) = VISQBDRY(N,2,K)
                  Enddo
                Enddo
                Read (IUT97,5000,End=420) T2VISQ
              Do N = 1, NUMQBC
                   Read (IUT97,5000) (VISQBDRY(N,2,K), K = 1, KBM)
              Enddo
              End If
              FACT = (THOUR - T1VISQ) / (T2VISQ - T1VISQ)
              Do N = 1, NUMQBC
               Do K = 1, KBM
                    VISQ(N,K) = VISQBDRY(N,1,K) +
     &               FACT * (VISQBDRY(N,2,K) - VISQBDRY(N,1,K))
               Enddo
             Enddo
           End If
    Endif
c-----Two equation (SST k-w)
    If (VERTMIX .EQ. 'SSTMODEL  ' .OR. DES .EQ. 'SSTDES ') Then
         If (NUMEBC .NE. 0) Then
            If (THOUR .GE. T2VISE) Then
               T1VISE = T2VISE
               Do N = 1, NUMEBC
                  TKEEBDRY(N,1) = TKEEBDRY(N,2)
                  TDISSEBDRY(N,1) = TDISSEBDRY(N,2)
               Enddo
               Read (IUT96,5000,End=420) T2VISE
               Read (IUT96,5000) (TKEEBDRY(N,2), N = 1, NUMEBC)
               Read (IUT96,5000) (TDISSEBDRY(N,2), N = 1, NUMEBC)
             End If
             FACT = (THOUR - T1VISE) / (T2VISE - T1VISE)
             Do N = 1, NUMEBC
  outl.EBC   k  TKEE(N) = TKEEBDRY(N,1) +
     &                    FACT * (TKEEBDRY(N,2) - TKEEBDRY(N,1))
             w  TDISSE(N) = TDISSEBDRY(N,1) +
     &                    FACT * (TDISSEBDRY(N,2) - TDISSEBDRY(N,1))

           Enddo
         End If
         If (NUMQBC .NE. 0) Then
           If (THOUR .GE. T2VISQ) Then
```

```
              T1VISQ = T2VISQ
              Do N = 1, NUMQBC
                 Do K = 1, KBM
                    TKEQBDRY(N,1,K) = TKEQBDRY(N,2,K)
                    TDISSQBDRY(N,1,K) = TDISSQBDRY(N,2,K)
               Enddo
              Enddo
             Read (IUT97,5000,End=420) T2VISQ
            Do N = 1, NUMQBC
                 Read (IUT97,5000) (TKEQBDRY(N,2,K), K = 1, KBM)
                 Read (IUT97,5000) (TDISSQBDRY(N,2,K), K = 1, KBM)
            Enddo
            End If
            FACT = (THOUR - T1VISQ) / (T2VISQ - T1VISQ)
            Do N = 1, NUMQBC
             Do K = 1, KBM
    infl.QBC  k  TKEQ(N,K) = TKEQBDRY(N,1,K) +
     &                FACT * (TKEQBDRY(N,2,K) - TKEQBDRY(N,1,K))
              w  TDISSQ(N,K) = TDISSQBDRY(N,1,K) +
     &                FACT * (TDISSQBDRY(N,2,K) - TDISSQBDRY(N,1,K))
             Enddo
          Enddo
         End If
    Endif
C-------------------------------------------------------------------------------C
C         Read concentration boundary condition at new time step              c
c-------------------------------------------------------------------------------c
    If (WAQ .NE. 'NEGLECT') Then
         If (NUMEBC .NE. 0) Then
            If (THOUR .GE. T2CEBC) Then
               T1CEBC = T2CEBC
               Do N = 1, NUMEBC
                  DCEBC(N,1) = DCEBC(N,2)
               Enddo
               Read (IUT98,5000,End=420) T2CEBC
               Read (IUT98,5000) (DCEBC(N,2), N = 1, NUMEBC)
            End If
            FACT = (THOUR - T1CEBC) / (T2CEBC - T1CEBC)
            Do N = 1, NUMEBC
               CEBC(N) = DCEBC(N,1) +
     &                   FACT * (DCEBC(N,2) - DCEBC(N,1))
```

```
          Enddo
        End If
        If (NUMQBC .NE. 0) Then
          If (THOUR .GE. T2CQBC) Then
            T1CQBC = T2CQBC
            Do N = 1, NUMQBC
               Do K = 1, KBM
                  DCQBC(N,1,K) = DCQBC(N,2,K)
             Enddo
            Enddo
            Read (IUT99,5000,End=420) T2CQBC
          Do N = 1, NUMQBC
               Read (IUT99,5000) (DCQBC(N,2,K), K = 1, KBM)
          Enddo
          End If
          FACT = (THOUR - T1CQBC) / (T2CQBC - T1CQBC)
          Do N = 1, NUMQBC
           Do K = 1, KBM
                CQBC(N,K) = DCQBC(N,1,K) +
     &               FACT * (DCQBC(N,2,K) - DCQBC(N,1,K))
           Enddo
         Enddo
        End If
        If (NUMDBC .NE. 0) Then
          If (THOUR .GE. T2CDBC) Then
            T1CDBC = T2CDBC
            Do N = 1, NUMDBC
               DCDBC(N,1) = DCDBC(N,2)
            Enddo
            Read (IUT101,5000,End=420) T2CDBC
            Read (IUT101,5000) (DCDBC(N,2), N = 1, NUMDBC)
          End If
          FACT = (THOUR - T1CDBC) / (T2CDBC - T1CDBC)
          Do N = 1, NUMDBC
             CDBC(N) = DCDBC(N,1) +
     &                 FACT * (DCDBC(N,2) - DCDBC(N,1))
         Enddo
        End If
      Endif
        Return
C=======================================================================C
```

```
C=====================================================================C
610   Write (IUPRT,5100) THOUR
      Go To 420
620   Write (IUPRT,5200) THOUR
      Go To 420
630   Write (IUPRT,5300) THOUR
      Go To 420
640   Write (IUPRT,5400) THOUR
      Go To 420
650   Write (IUPRT,5500) THOUR
    Go To 420
660   Write (IUPRT,5500) THOUR
    Go To 420
420   Continue
      Close (IUT90)
      Close (IUT91)
      Close (IUT92)
      Close (IUT93)
      Close (IUT94)
 5000 Format (8E14.7)
 5100 Format (//' the model has run out of elevation data at time 'F10
     *    .4,' hours'/,'       revise input deck and resubmit '//)
 5200 Format (//' the model has run out of velocity data at time
     *    'F10.4,' hours'/,'       revise input deck and resubmit',//)
 5300 Format (//' the model has run out of discharge / onshore data
     *      at time 'F10
     *    .4,' hours'/,'       revise input deck and resubmit '//)
 5400 Format (//' the model has run out of discharge / offshore data
     *      at time 'F10.4,
     *    ' hours'/,'       revise input deck and resubmit '//)
 5500 Format (//' the model has run out of astrotidle data at time '
     *    F10.4,' hours'/,'       revise input deck and resubmit '//)
      End
```