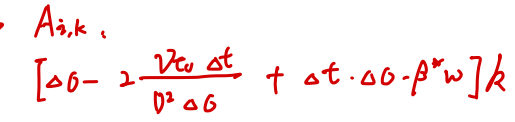
```
C#############################################################################
c                                                                             #
c                    SUBROUTINE PROGRAM                                       #
C                  VERSION 1.0 (16/02/2011)                                   #
C                  AUTHORIZED BY ZHANG JINGXIN                                #
C                               SHANGHAI JIAO TONG UNIVERSITY                 #
C                               SHANGHAI, CHINA                               #
c-----------------------------------------------------------------------------#
c                      computes the velocity                                  #
c                                                                             #
c#############################################################################
      Subroutine PROFTK(F1)
      Include './Include/OCERM_INF'
      Common/VISKW/VISSOURCE(IJM,KB),TKEMARK(IJM,KB)
    Parameter(SIGMAK1=0.85,SIGMAK2=1.0)
        Dimension F1(IJM,KB),VISCOE(IJM,KB)
        Dimension AAAA(KBM,KBM), BBBB(KBM)
    Dimension AA(3*KBM-2), BB(KBM)

!!!###########################################################################!!!
!!!   Feature    :Type of BC at free surface.                                 !!!
!!!               (1: zeore-gradient, 2: value)                               !!!
!!!   UpdatedBy  :WangJian                                                    !!!
!!!   UpdatedDate:2020-3-9 22:47:29                                           !!!
!!!###########################################################################!!!
      Itype_BCtop = 2
      !#用零梯度算出的床面摩擦系数比赋值形式的大#, WJ, 2020-1-9 !
c=============================================================================c
C              initializing the arrays                                        c
c=============================================================================c
    Do K1 = 1, KBM
         Do K2 = 1, KBM
          AAAA(K1,K2) = 0.0
       Enddo
    Enddo
    Do K1 = 1, 3 * KBM - 2
       AA(K1) = 0.0
    Enddo
    Do K1 = 1, KBM
       BB(K1) = 0.0
       BBBB(K1) = 0.0
```

$A_{ix}\,\phi_{xi} = B_i$

$\Downarrow$

$\phi \to k$

```
          RKM_TOP(K1)= 0.0
      Enddo
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J,K)
      Do K = 1, KB
!$OMP DO
        Do I = 1, IJM
          VISCOE(I,K) = F1(I,K) * SIGMAK1 + (1. - F1(I,K)) * SIGMAK2
        Enddo
!$OMP END DO
      Enddo
!$OMP END PARALLEL
c=======================================================================c
c                  arrays for the turbulent kinetic governing equation      c
c=======================================================================c
!$OMP PARALLEL DO DEFAULT(SHARED)
!$OMP&PRIVATE(I,J,K,K1,K2,AAAA,BBBB,AA,BB,L,D1)  !#Extented to 2 lines#, WJ,
2020-3-5 23:14:08 !
      Do I = 1, IJM
        IF(CCM(I) .EQ. 1.0) Then
          Do K = 2, KBM
            AAAA(K,K-1) = -DTI * (UMOL + (KM(I,K) + KM(I,K-1)) / 2. *
     &                    (VISCOE(I,K) + VISCOE(I,K-1)) / 2.) /
     &                    DC(I) ** 2. / DZZ(K-1) * TKEMARK(I,K)
            AAAA(K-1,K) = AAAA(K,K-1)
          Enddo
          Do K = 2, KBM - 1
            AAAA(K,K) = DZ(K) - AAAA(K,K-1) - AAAA(K,K+1) +
     &                  DTI * DZ(K) * VISSOURCE(I,K)
          Enddo
          Do K = 1, KBM
            BBBB(K) = VISF(I,K) / AREA(I)
          Enddo
c----       Boundary condition   ------------------------------------------c
c----       at the surface (zero gradient)-----------------------------------c
           IF (Itype_BCtop .EQ. 1 )THEN
          If(KBM .GT. 1) Then
              AAAA(1,1) = DZ(1) - AAAA(1,2) +
     &                    DTI * DZ(1) *  VISSOURCE(I,1)     !For zero gradient on
surface
c               AAAA(1,1) = DZ(1) - 2.0*AAAA(1,2) +
```



```
c     &                   DTI * DZ(1) *  VISSOURCE(I,1)     !For a small value on
surface
C              IF(IBC_CONSISTENT .EQ. 1) THEN
C                AAAA(1,1) = DZ(1) - AAAA(1,2) + DTI*( UMOL + KM_TOP(I,1)
C     &                     * VISCOE(I,1) )/ DC(I)**2. * DIVTKEZ_TOP(I,1)
C     &                     + DTI * DZ(1)* VISSOURCE(I,1)
C              ENDIF
          Else
             AAAA(1,1) = DZ(1) +
     &                   DTI * (KM(I,1)+UMOL) * VISCOE(I,1) /
     &                   (.5 * DZ(1)) / DC(I) ** 2. +
     &                   DTI * DZ(KBM) * 0.09 * TDISS(I,KBM)
          Endif
           Endif
c---------at the surface (a given value)-----------------------------------c
           IF (Itype_BCtop .EQ. 2 )THEN
          If(KBM .GT. 1) Then
c---------------value at top layer center------------------------------
C              AAAA(1,1) = 1.0
C              AAAA(1,2) = 0.0
c              D1 = Min(DC(I)*(1. + ZZ(1)), D2D(I))
c             BBBB(1) = (0.1 * D1) ** 2. * TENSOR(I,1) ** 2. / 0.3
c---------------value at top layer-------------------------------------

             AAAA(1,1) = DZ(1) - 2. * AAAA(1,2) +
     &                   DTI * DZ(1) *  VISSOURCE(I,1)
               BBBB(1) = BBBB(1) - AAAA(1,2) *
     &                 (0.1 *Dmin1(D2D(I),DC(I)))**2.*TENSOR(I,1)**2./0.3
          Else
             AAAA(1,1) = DZ(1)
             BBBB(1) = UMOL * TENSOR(I,1) / 0.3
            Endif
          Endif
c-----     at the bottom  ---------------------------------------c
           If(WFBC .EQ. '      FUN1') Then      ! wall function for RANS
              AAAA(KBM,KBM) = 1.0
             BBBB(KBM) = Sqrt(TBX(I)** 2. + TBY(I)** 2.) / RMEAN(I,KBM) /
     &                   0.3
          Endif
          If(WFBC .EQ. '      FUN2') Then      ! wall function for LES
```

```
          Endif
          If(WFBC .EQ. '       FUN3') Then       ! no-slip B.C.
            If(KBM .GT. 1) Then
                AAAA(KBM,KBM) = DZ(KBM) - AAAA(KBM,KBM-1) * TKEMARK(I,KBM) +
     &             DTI*(UMOL+KM(I,KBM)*VISCOE(I,KBM))*TKEMARK(I,KBM)/
     &                          DC(I) ** 2. / DZZ(KBM) +
     &                        DTI * DZ(KBM) * VISSOURCE(I,KBM)
            Else
                  AAAA(KBM,KBM) = 1.0
C               AAAA(KBM,1) = 0.0
                BBBB(KBM) = UMOL * TENSOR(I,1) / 0.3
            Endif
          Endif
C=======================================================================C
C            solve by forward elimination and back-substitution         c
C=======================================================================C
          AA(1) = AAAA(1,1)
          If(KBM .GT. 1) AA(2) = AAAA(1,2)
            Do K = 2, KBM - 1
               AA(2*(K-1)+K-1) = AAAA(K,K-1)
              AA(2*(K-1)+K)    = AAAA(K,K)
               AA(2*(K-1)+K+1) = AAAA(K,K+1)
            Enddo
          If(KBM .GT. 1) AA(3*KBM-3) = AAAA(KBM,KBM-1)
          AA(3*KBM-2) = AAAA(KBM,KBM)
            Do K = 1, KBM
             BB(K) = BBBB(K)
          Enddo
            Call ATRDE(AA,KBM,3*KBM-2,BB,L)
          Do K = 1, KBM
             TKE(I,K) = Max(BB(K),1.0E-20)
          Enddo
        Endif
      Enddo
!$OMP END PARALLEL DO
C-----------------------------------------------------------------------C
      Return
      End
```