```fortran
C#########################################################################
C                       BI-CGSTAB method for equations solving           #
c                              VERSION 1.0 (25/05/2009)                  #
C                              AUTHORIZED BY ZHANG JINGXIN               #
C                              SHANGHAI JIAO TONG UNIVERSITY             #
C                                   SHANGHAI, CHINA                      #
c#########################################################################
      Subroutine SOLVEELFPOLCG
      Include './Include/OCERM_INF'
      Parameter(EPSON=1.E-20,EPSI=1.E-6)
      Common/ELFBLK/CS(IJM,IPOLYGEN),CB(IJM),CP(IJM),X(IJM)
      Dimension R1(IJM),U1(IJM),V1(IJM),P1(IJM)
      Dimension TRACE(IJM)
C========================================================================C
C                 optimazing the Matrix                                  c
c========================================================================c
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(I,J)
!$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
                  CS(I,J) = CS(I,J) /
     &                     Sqrt(CP(I)) / Sqrt(CP(CELL_SIDE(I,J,2)))
               Endif
            Enddo
            CB(I) = CB(I) / Sqrt(CP(I))
         Endif
      Enddo
!$OMP END DO
!$OMP DO
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            TRACE(I) = CP(I)
            CP(I) = 1.0
         Endif
      Enddo
!$OMP END DO
!$OMP DO
      Do I=1,IJM
         If(CCM(I) .EQ. 1.0) Then
```

$$\frac{AP_i\xi_i}{\sqrt{AP_i}} + \sum \frac{AP_{is}}{\sqrt{AP_i}}\cdot\frac{1}{\sqrt{AP_{is}}}\xi_{is}\sqrt{AP_{is}} = \frac{\langle BB_i\rangle}{\sqrt{AP_i}}$$

$$Ax = b$$

$$x = \sqrt{AP_i}\,\xi_i$$

$$AP_i\xi_i + \sum AP_{is}\xi_{is} = \langle BB_i\rangle$$

$$\Downarrow$$

$$A\xi_i = b$$

$$\Downarrow$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1j} & & \\ \vdots & a_{22} & & a_{2j} & \\ a_{j1} & & \ddots & & \\ & a_{j2} & & a_{ii} & \\ & & & \ddots & a_{nn} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_i \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

```
        U1(I) = 0.0
        V1(I) = 0.0
        P1(I) = 0.0
        X(I) = 0.0
      Endif
    Enddo
!$OMP END DO
C-----   INITIAL VALUES
!$OMP DO
    Do I = 1, IJM
      If(CCM(I) .EQ. 1.0) Then
        X(I)  = ELF(I) * Sqrt(TRACE(I))
      Endif
    Enddo
!$OMP END DO
!$OMP BARRIER
!$OMP DO
    Do I = 1, IJM
      If(CCM(I) .EQ. 1.0) Then
        R1(I) = 0.0
        Do J = 1, CELL_POLYGEN(I)
          If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
            R1(I) = R1(I) +
     &                      CS(I,J) * X(CELL_SIDE(I,J,2))
          Endif
         Enddo
        R1(I) = CB(I) + R1(I) - CP(I) * X(I)
        P1(I) = R1(I)
      Endif
    Enddo
!$OMP END DO
!$OMP END PARALLEL
C--------------------------------------------------------------------------C
    ROU = 1.0
    ALPHA = 1.0
    OMEGA = 1.0
      KNUM = 0
 10   Continue
      KNUM = KNUM+1
      BETA = ROU
    ROU = 0.0
```

$$\boxed{\begin{aligned} r^{(0)} &= b - Ax^{(0)} \\ p^{(0)} &= r^{(0)} \end{aligned}}$$

```
CD     ROU=(P,B)
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I) REDUCTION(+:ROU)
      Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
         ROU = ROU + R1(I) * R1(I)
```

$(r^{(k)}, r^{(k)})$

```
       Endif
    Enddo
!$OMP END PARALLEL DO
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I,J)
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
         V1(I) = 0.0
         Do J = 1, CELL_POLYGEN(I)
            If(CFM(CELL_SIDE(I,J,1)) .EQ. 1.0) Then
               V1(I) = V1(I) + CS(I,J) * P1(CELL_SIDE(I,J,2))
            Endif
         Enddo
         V1(I) = -V1(I) + CP(I) * P1(I)
```

$Ap^{(k)}$

```
       Endif
    Enddo
!$OMP END PARALLEL DO
    BV = 0.0
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I) REDUCTION(+:BV)
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
         BV = BV + P1(I) * V1(I)
```

$(p^{(k)}, Ap^{(k)})$

```
       Endif
    Enddo
!$OMP END PARALLEL DO
    ALPHA = ROU / (BV + Sign(EPSON,BV))
```

$$\boxed{\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(p^{(k)}, Ap^{(k)})}}$$

```
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I)
      Do I = 1, IJM
          If(CCM(I) .EQ. 1.0) Then
         X(I) = X(I) + ALPHA * P1(I)
```

$$\boxed{x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}}$$

```
       Endif
    Enddo
!$OMP END PARALLEL DO
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I)
    Do I = 1, IJM
       If(CCM(I) .EQ. 1.0) Then
         R1(I) = R1(I) - ALPHA * V1(I)
```

$$\boxed{r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}}$$

```
         Endif
      Enddo
!$OMP END PARALLEL DO
      AR = 0.0
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I) REDUCTION(+:AR)
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            AR = AR + R1(I) * R1(I)
         Endif
      Enddo
!$OMP END PARALLEL DO
      AR = Sqrt(AR)
```

$r^{(k+1)} \to 0$

```
        If(AR .LT. EPSI .OR. KNUM .GE. 200) Goto 1000
      UR1 = 0.0
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I) REDUCTION(+:UR1)
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            UR1 = UR1 + R1(I) * R1(I)
```

$(r^{(k+1)}, r^{(k+1)})$

```
         Endif
      Enddo
!$OMP END PARALLEL DO
      BETA = UR1 / (ROU + EPSON)
```

$$\boxed{\beta_k = \frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})}}$$

```
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I)
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            P1(I) = R1(I) + BETA * P1(I)
```

$$\boxed{p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}}$$

```
         Endif
      Enddo
!$OMP END PARALLEL DO
      Goto 10
C----------------------------------------------------------------------C
1000    Continue
!$OMP PARALLEL DO DEFAULT(SHARED) PRIVATE(I)
      Do I = 1, IJM
         If(CCM(I) .EQ. 1.0) Then
            X(I) = X(I) / Sqrt(TRACE(I))
         Endif
      Enddo
!$OMP END PARALLEL DO
      Return
      End
```