```
***********************************************************************
* ファイル名  ：MAT2D.FOR                                             *
* タイトル    ：線形システム解法プログラム                            *
* 製作者      ：平野　博之                                            *
* 所属        ：岡山理科大学 工学部 応用化学科                        *
* 製作日      ：2003.12.25                                            *
* 言語        ：FORTRAN                                               *
***********************************************************************
*［内容］                                                             *
*    本プログラムは本書の15.6節にある2次元の例題で作成された線形      *
*    システム解法の1例である.                                         *
*    解法は，直接法，反復法，そしてクリロフ部分空間法を用いている.   *
*    解法アルゴリズムとそのサブルーチン名は以下に示す通りである.      *
*    ---------------------------------------------------------------  *
*    直接法.....(15.3)節                  サブルーチン名              *
*    ---------------------------------------------------------------  *
*    LU分解法(軸選択付)              ---> DLUP..(15.3.1),(15.3.2)項*
*    Gaussの消去法(軸選択付)         ---> DFGP...(15.3.2)項          *
*    Gaussの消去法(軸選択無)         ---> DFG....(15.3.2)項          *
*    Gauss-Jordan法(軸選択付)        ---> DGJP...(15.3.3)項          *
*    Gaussの消去法(2次元(5点)差分    ---> DGBND..(15.3.4)項          *
*    バンドマトリックス用：軸選択無)                                  *
*    ---------------------------------------------------------------  *
*    反復法                               サブルーチン名              *
*    ---------------------------------------------------------------  *
*    JACOBI法                        ---> HJACOB...(15.4.1)項        *
*    point-SOR法                     ---> HSOR..(15.4.2),(15.4.3)項*
*    point-SOR法(バンドマトリックス用)---> SORBND(15.4.2),(15.4.3)項*
*    line-SOR法(バンドマトリックス用) ---> LBLBND...(15.3.5)項       *
*    注意：SOR法で，ω=1とするとGauss-Seidel法となる.                 *
*    ---------------------------------------------------------------  *
*    クリロフ部分空間法...(15.5)節        サブルーチン名              *
*    ---------------------------------------------------------------  *
*    共役残差法                      ---> KCR.....(15.5.3)項         *
*    共役残差法(バンドマトリックス用) ---> KCRBND..(15.5.3)項        *
*    Bi-CGSTAB                       ---> KBICG...(15.5.4)項         *
*    Bi-CGSTAB(バンドマトリックス用)  ---> KBIBND..(15.5.4)項        *
*［ 例  題 ］.....(15.6)節                                            *
*    2次元偏微分方程式 ∇^{2}f + ∇_{x}f + ∇_{y}f = b について，2次*
*  精度中心差分近似を用いて離散化せよ．そして以下に示す計算領域を考 *
*  え，通し番号をつけられた計算格子の番号そのものが解となるような   *
*  線形システム(AX=B)を作り，各種の解法を用いて解け.                 *
*        +------------------------+  (注)                             *
* (NY)5  |  5 | 10 | 15 | 20 | 25 |  f の定義点は各格子の中心である *
*        +------------------------+  とする．f の定義点間距離および *
*     4  |  4 |  9 | 14 | 19 | 24 |  各計算格子の幅はいずれもΔ=1と *
*        +------------------------+  する.                            *
*     3  |  3 |  8 | 13 | 18 | 23 |  また，境界条件に関しては，計算 *
*        +------------------------+  領域外の定義点の値をゼロとして *
*     2  |  2 |  7 | 12 | 17 | 22 |  扱う.                            *
*        +------------------------+  通し番号:k=(i-1)*NY+j           *
```

```
*     1  |  1 |  6 | 11 | 16 | 21 |                               *
*        +------------------------+                              *
*      j   i=1     2    3    4    5(NX)                          *
*  x,y方向それぞれ5等分し，通し番号をk=1から25までつけ，f{i,j}=kが  *
*  解となるようにする．                                          *
*[  解  答  例  ]                                                *
*( 1 )  離  散  化                                               *
*   ∇^{2}f + ∇_{x}f + ∇_{y}f = b                                 *
*   ===> ( f_{i-1,j  }-2f_{i,j}+f_{i+1,j  } ) / Δ^{2}            *
*       +( f_{i  ,j-1}-2f_{i,j}+f_{i  ,j+1} ) / Δ^{2}            *
*       +( f_{i+1,j  }-f_{i-1,j  } ) / (2Δ)                      *
*       +( f_{i  ,j+1}-f_{i  ,j-1} ) / (2Δ)                      *
*       = b_{k}   where k=(i-1)*NY + j                           *
*   ===> (  1/Δ^{2} - 1/(2Δ)  ) f_{i-1,j  }                      *
*         ~~~~~~~~~~~~~~~~~~~~~~~~-> AT1(k) = A(i-1,j)           *
*       +( -2/Δ^{2} - 2/Δ^{2} ) f_{i  ,j  }                      *
*         ~~~~~~~~~~~~~~~~~~~~~~~~-> AT3(k) = A(i,j)             *
*       +(  1/Δ^{2} + 1/(2Δ)  ) f_{i+1,j  }                      *
*         ~~~~~~~~~~~~~~~~~~~~~~~~-> AT5(k) = A(i+1,j)           *
*       +(  1/Δ^{2} - 1/(2Δ) )  f_{i  ,j-1}                      *
*         ~~~~~~~~~~~~~~~~~~~~~~~~-> AT2(k) = A(i,j-1)           *
*       +(  1/Δ^{2} + 1/(2Δ) )  f_{i  ,j+1}                      *
*         ~~~~~~~~~~~~~~~~~~~~~~~~-> AT4(k) = A(i,j+1)           *
*       = b_{k}   where k=(i-1)*NY + j  : 注意：Δ=1とする         *
*   ===> Af = B ===> AX = B                                      *
*( 2 )  マトリックスの設定                                        *
*  2-1 Aの設定                                                   *
*    Aは65行から73行にあるように設定する．AT1,AT2,AT3,AT4,AT5 は   *
*    バンドマトリックス用の解法のためのものである．これ以外の，ゼロ *
*    の要素を含む全(フル)マトリックスを扱う場合は，Aを用いる．      *
*  2-2 Bの設定                                                   *
*    65行から73行において，解となるfの値を代入したものをBとすればよ *
*    い．具体的には，以下のようにして順にマトリックスを求めてゆく． *
*    k= 1 (i=1,j=1)->f_{i-1,j  }:領域外となるので0とする          *
*                    f_{i  ,j  }:(i-1)*NY+jを代入                 *
*                    f_{i+1,j  }:((i+1)-1)*NY+jを代入             *
*                    f_{i  ,j-1}:領域外となるので0とする          *
*                    f_{i  ,j+1}:(i-1)*NY+(j+1)を代入             *
*                    b_{k}:以上のfの値とΔ(=1)から求まる          *
*     k= 2 (i=1,j=2), k= 3 (i=1,j=3), k= 4 (i=1,j=4),            *
*     k= 5 (i=1,j=5), k= 6 (i=2,j=1), k= 7 (i=2,j=2),            *
*     ...... ,          k=25 (i=5,j=5)                           *
*   i=3, j=3 (k=13における，AT1,AT2,AT3,AT4,AT5とAの関係           *
*        +------------------------+                              *
* (NY)5  |    |    |    |    |    |                              *
*        +------------------------+                              *
*     4  |    |    |AT4 |    |    |    k=(i-1)*NY+j              *
*        +------------------------+                              *
*     3  |    | AT1|AT3 |AT5 |    |    AT1(k) -> A(k,k-NY)       *
*        +---------+--------------+    AT2(k) -> A(k,k- 1)       *
*     2  |    |    |AT2 |    |    |    AT3(k) -> A(k,k   )       *
```

```
*        +-------------------------+    AT4(k) -> A(k,k+ 1)          *
*     1  |    |    |    |    |    |    AT5(k) -> A(k,k+NY)          *
*        +-------------------------+    B(k)   -> B(k)              *
*     j   i=1    2    3    4    5(NX)                               *
********************************************************************
      PROGRAM MAT2D
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
* パラメータ変数(NX0と0がついているのはサブルーチン引数にはPARAMETER
* 変数であるNX0を直接渡せないため. 後でNX=NX0と代入し直してこれを渡し
* ている. )  NX:x方向格子数, NY:y方向格子数, NE:全格子数=NX*NY
      PARAMETER ( NX0=5, NY0=5, NE0=25 )
      DIMENSION A(NE0,NE0),B(NE0),X1(NE0),X2(NX0,NY0)
* バンドマトリックス用配列の定義
      DIMENSION AT1(NE0),AT2(NE0),AT3(NE0),AT4(NE0),AT5(NE0)
* 作業用配列
      DIMENSION W1(NE0),W2(NE0),W3(NE0),W4(NE0),W5(NE0),W6(NE0),
     $          W7(NE0),W8(NE0),AGB(-NY0:NY0,NE0)
      DIMENSION IPV(NE0)* 上記離散化方程式のΔ=1を設定
      DELTA = 1.0D0
* PARAMETER変数の値をSUBROUTINE引数に渡すために代入
      NX = NX0
      NY = NY0
      NE = NE0
* 戻り点 : 解法の変更毎の戻り点
  700 CONTINUE
* A,B を設定するためにあらかじめX2の値を格子番号と同じになるようにする
      I = 1
      DO 10 IX = 1,NX
        DO 20 IY = 1,NY
          X2(IX,IY) = DBLE(I)
          I =I + 1
   20   CONTINUE
   10 CONTINUE
* X2 以外の配列のゼロクリア
      DO 30 I = 1,NE
        AT1(I) = 0.0D0
        AT2(I) = 0.0D0
        AT3(I) = 0.0D0
        AT4(I) = 0.0D0
        AT5(I) = 0.0D0
        B(I) = 0.0D0
        X1(I) = 0.0D0
        W1(I)=0.0D0
        W2(I)=0.0D0
        W3(I)=0.0D0
        W4(I)=0.0D0
        W5(I)=0.0D0
        W6(I)=0.0D0
        W7(I)=0.0D0
        W8(I)=0.0D0
        DO 40 II = 1,NE
```

```
          A(I,II) = 0.0D0
   40   CONTINUE
        DO 45 III = -NY,NY
          AGB(III,I) = 0.0D0
   45   CONTINUE
   30 CONTINUE
* X_{i,j}=k=(i-1)*NY+j となるように A,B を求める
      I = 1
      DO 50 IX = 1,NX
        DO 60 IY = 1,NY
*         f(i-1,j)が計算領域外（左側）の境界条件の処理
          IF (IX.EQ.1) THEN
*           f(0,j)=0として処理する
            AT1(I) = 0.0D0
*           AT1*f(0,j)をB1とする
            B1 = 0.0D0
*         f(i-1,j)が計算領域内の場合
          ELSE
            AT1(I) = 1.0D0/DELTA**2 - 1.0D0/(2.0D0*DELTA)
            A(I,I-NY) = AT1(I)
            B1 = X2(IX-1,IY  )*( 1.0D0/DELTA**2 - 1.0D0/(2.0D0*DELTA) )
          END IF
*         f(i,j)は常に計算領域内
          AT3(I) = -2.0D0/DELTA**2 - 2.0D0/DELTA**2
          A(I,I) = AT3(I)
*         AT3*f(i,j)をB3とする
          B3 = X2(IX  ,IY  )*( -2.0D0/DELTA**2 - 2.0D0/DELTA**2 )
*         f(i+1,j)が計算領域外（右側）の境界条件の処理
          IF (IX.EQ.NX) THEN
*           f(NX,j)=0として処理
            AT5(I) = 0.0D0
*           AT5*f(NX,j)=B5とする
            B5 = 0.0D0
*         f(i+1,j)が計算領域内の場合
          ELSE
            AT5(I) = 1.0D0/DELTA**2 + 1.0D0/(2.0D0*DELTA)
            A(I,I+NY) = AT5(I)
            B5 = X2(IX+1,IY  )*( 1.0D0/DELTA**2 + 1.0D0/(2.0D0*DELTA) )
          END IF
*         f(i,j-1)が計算領域外（下側）の境界条件の処理
          IF (IY.EQ.1) THEN
*           f(i,j-1)=0として処理
            AT2(I) = 0.0D0
*           AT2*f(i,j-1)=B2とする
            B2 = 0.0D0
*         f(i,j-1)が計算領域内の場合
          ELSE
            AT2(I) = 1.0D0/DELTA**2 - 1.0D0/(2.0D0*DELTA)
            A(I,I-1) = AT2(I)
            B2 = X2(IX  ,IY-1)*( 1.0D0/DELTA**2 - 1.0D0/(2.0D0*DELTA) )
          END IF
```

```
*         f(i,j+1)が計算領域外（上側）の境界条件の処理
          IF (IY.EQ.NY) THEN
*           f(i,NY)=0として処理
            AT4(I) = 0.0D0
*           AT4*f(i,NY)=B4とする
            B4 = 0.0D0
*         f(i,j+1)が計算領域内の場合
          ELSE
            AT4(I) = 1.0D0/DELTA**2 + 1.0D0/(2.0D0*DELTA)
            A(I,I+1) = AT4(I)
            B4 = X2(IX  ,IY+1)*( 1.0D0/DELTA**2 + 1.0D0/(2.0D0*DELTA) )
          END IF
*         Bの計算
          B(I) = B1 + B2 + B3 + B4 + B5
          I = I + 1
   60   CONTINUE
   50 CONTINUE
* 戻り点 : 解法の選択が無効なときの戻り点
  710 CONTINUE
* 各サブルーチンを用いて計算する
      WRITE (6,2000)
 2000 FORMAT ( '  1 : LU Decomposition '/,
     $         '  2 : Gauss Elimination '/,
     $         '  3 : Gauss Elimination without Pivotting '/,
     $         '  4 : Gauss Jordan '/,
     $         '  5 : Gauss Elimination for Band Matrix '/,
     $         '  6 : Jacobi '/,
     $         '  7 : point-SOR '/,
     $         '  8 : point-SOR for Band matrix '/,
     $         '  9 : line-SOR for Band Matrix '/,
     $         ' 10 : Conjugate Residual '/,
     $         ' 11 : Conjugate Residual for Band Matrix '/,
     $         ' 12 : Bi-CGSTAB '/,
     $         ' 13 : Bi-CGSTAB for Band Matrix '/,
     $         ' 14 : END '/,
     $         ' Input Number ---> ')
      READ (5,*) IJ
*   直接法
      IF (IJ.EQ.1) THEN
        CALL DLU (A,B,X1,NE,
     $            W1,IPV)
      ELSE IF (IJ.EQ.2) THEN
        CALL DFGP (A,B,X1,NE,
     $             IPV)
      ELSE IF (IJ.EQ.3) THEN
        CALL DFG (A,B,X1,NE)
      ELSE IF (IJ.EQ.4) THEN
        CALL DGJP (A,B,X1,NE,
     $             IPV)
      ELSE IF (IJ.EQ.5) THEN
        CALL DGBND (AT1,AT2,AT3,AT4,AT5,B,X1,NY,NE,
```

```
     $              AGB)
*   反復法
      ELSE IF (IJ.EQ.6) THEN
        CALL HJACOB (A,B,X1,NE,
     $               W1)
      ELSE IF (IJ.EQ.7) THEN
        CALL HSOR (A,B,X1,NE)
      ELSE IF (IJ.EQ.8) THEN
        CALL SORBND (AT1,AT2,AT3,AT4,AT5,B,X1,NY,NE)
      ELSE IF (IJ.EQ.9) THEN
        CALL LBLBND (AT1,AT2,AT3,AT4,AT5,B,X1,NX,NY,NE,
     $               W1,W2,W3,W4,W5,W6,W7,W8)
*   クリロフ部分空間法
      ELSE IF (IJ.EQ.10) THEN
        CALL KCR (A,B,X1,NE,
     $            W1,W2,W3,W4)
      ELSE IF (IJ.EQ.11) THEN
        CALL KCRBND (AT1,AT2,AT3,AT4,AT5,B,X1,NY,NE,
     $               W1,W2,W3,W4)
      ELSE IF (IJ.EQ.12) THEN
        CALL KBICG (A,B,X1,NE,
     $              W1,W2,W3,W4,W5,W6)
      ELSE IF (IJ.EQ.13) THEN
        CALL KBIBND (AT1,AT2,AT3,AT4,AT5,B,X1,NY,NE,
     $               W1,W2,W3,W4,W5,W6)
      ELSE IF (IJ.EQ.14) THEN
        GO TO 900
      ELSE
        GO TO 710
      END IF
      GO TO 700
  900 CONTINUE
* 計算終了
      STOP
      END
*********************************************************************
*                  直接法 : 解法 1 - 5
*********************************************************************
*    LU分解による非対称行列 A を含む線形システム解法サブルーチン     *
*    (部分軸選択付)        AX=B                                      *
*    LU分解では次のような2段階で解を求める                           *
*    第1段階 : LY = B から Y を求める                                *
*    第2段階 : UX = Y から X を求める                                *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数     *
*    A(NE,NE) : AX=B のマトリックス A                                *
*    B(NE) : 線形システム AX=B のマトリックスB                       *
*    X(NE) : 線形システム AX=B のマトリックスX                       *
*            これは第2段階の UX=Y の解 X                             *
*            (このサブルーチンでこれを求める)                        *
*    Y(NE) : LY=B の解 Y                                             *
*    IPV(NE) : 部分軸選択用の配列                                    *
```

```
*             全部で NE 回の消去操作を行うが, 各消去段階で行う除算  *
*             について分母が最も大きくなるような列の番号が入る      *
**********************************************************************
      SUBROUTINE DLU (A,B,X,NE,
     $                Y,IPV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE),Y(NE)
      DIMENSION IPV(NE)
* 配列のゼロクリア
      DO 10 I = 1,NE
        Y(I)  = 0.0D0
   10 CONTINUE
* 部分軸選択用配列の初期設定
      DO 20 I = 1, NE
        IPV(I) = I
   20 CONTINUE
* 部分軸選択による特異性の判定値
      EPS = 1.0D-50
      DO 30 K = 1, NE
        L = K
        APV = ABS( A(IPV(L),K) )
        DO 40 I = K+1, NE
*         部分軸選択
          IF ( ABS( A(IPV(I),K) ). GT. APV ) THEN
            L = I
            APV = ABS( A(IPV(L),K) )
          END IF
   40   CONTINUE
*       部分軸選択を行った方がよいと判断したら入れ替えを行う
        IF (L.NE.K) THEN
          IPVEX = IPV(K)
          IPV(K) = IPV(L)
          IPV(L) = IPVEX
        END IF
*       部分軸選択を行っても計算不可能なとき : 行列は特異(singular)
        IF ( ABS( A(IPV(K),K) ). LE. EPS ) THEN
          WRITE (*,2000) K
 2000     FORMAT (' Matrix is singular at K = ', I4)
          STOP
        END IF
* U を求める..........式(15.12)
        A(IPV(K),K) = 1.0D0 / A(IPV(K),K)
        DO 50 I = K+1, NE
          A(IPV(I),K) = A(IPV(I),K) * A(IPV(K),K)
          DO 60 J = K+1, NE
            A(IPV(I),J) = A(IPV(I),J) - A(IPV(I),K)*A(IPV(K),J)
   60     CONTINUE
   50   CONTINUE
   30 CONTINUE
* 第1段階 LY = B を解く..........Lは式(15.17)で定義される
      Y(1) = B(IPV(1))
```

```
      DO 70 I = 2, NE
        T = B(IPV(I))
        DO 80 J = 1, I-1
          T = T - A(IPV(I),J) * Y(J)
   80   CONTINUE
        Y(I) = T
   70 CONTINUE
* 第2段階 UX = Y を解く
      X(NE) = Y(NE) * A(IPV(NE),NE)
      DO 90 I = NE-1, 1, -1
        T = Y(I)
        DO 100 J = I+1, NE
          T = T - A(IPV(I),J) * X(J)
  100   CONTINUE
        X(I) = T * A(IPV(I),I)
   90 CONTINUE
* 結果出力
      DO 110 I = 1,NE
        WRITE(6,2010) I,X(I)
 2010   FORMAT(' X_1_( ',I3,' ) = ',1PD13.5)
  110 CONTINUE
      RETURN
      END
*********************************************************************
*    Gaussの消去法(部分軸選択付)による非対称行列Aを含む線形システム *
*    解法サブルーチン     AX=B                                       *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数     *
*    A(NE,NE) : AX=B のマトリックス A                                *
*    B(NE) : 線形システム AX=B のマトリックスB                       *
*    X(NE) : 線形システム AX=B のマトリックスX                       *
*            (このサブルーチンでこれを求める)                        *
*    IPV(NE) : 部分軸選択用の配列                                    *
*             全部で NE 回の消去操作を行うが, 各消去段階で行う除算  *
*             について分母が最も大きくなるような列の番号が入る      *
*********************************************************************
      SUBROUTINE DFGP (A,B,X,NE,
     $                 IPV)
      IMPLICIT DOUBLE PRECISION (A,B,D-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
      DIMENSION IPV(NE)
* 部分軸選択による特異性の判定値
      EPS = 1.0D-50
* 部分軸選択用配列の初期設定
      DO 10 I = 1, NE
        IPV(I) = I
   10 CONTINUE
      DO 20 K = 1, NE
        L = K
        APV = ABS( A(IPV(L),K) )
*       部分軸選択
        DO 30 I = K+1, NE
```

```
          IF ( ABS( A(IPV(I),K) ). GT. APV ) THEN
            L = I
            APV = ABS( A(IPV(L),K) )
          END IF
   30   CONTINUE
*       部分軸選択を行った方がよいと判断したら入れ替えを行う
        IF (L.NE.K) THEN
          IPVEX = IPV(K)
          IPV(K) = IPV(L)
          IPV(L) = IPVEX
        END IF
*       部分軸選択を行っても計算不可能なとき : 行列は特異
        IF ( ABS( A(IPV(K),K) ). LE. EPS ) THEN
          WRITE (*,2000) K
 2000     FORMAT (' Matrix is singular at K= ', I4)
          STOP
        END IF
* 前進消去
        DO 40 I = K+1, NE
          A(IPV(K),I)=A(IPV(K),I)/A(IPV(K),K)
   40   CONTINUE
        B(IPV(K)) = B(IPV(K))/A(IPV(K),K)
        DO 50 I = K+1, NE
          DO 60 J = K+1, NE
            A(IPV(I),J)=A(IPV(I),J)-A(IPV(I),K)*A(IPV(K),J)
   60     CONTINUE
          B(IPV(I))=B(IPV(I))-A(IPV(I),K)*B(IPV(K))
   50   CONTINUE
   20 CONTINUE
* 後退代入
      X(NE) = B(IPV(NE))
      DO 70 I = NE-1, 1, -1
        T = B(IPV(I))
        DO 80 J = I+1, NE
          T = T - A( IPV(I), J ) * X(J)
   80   CONTINUE
        X(I) = T
   70 CONTINUE
* 結果出力
      DO 90 I = 1,NE
        WRITE(6,2010) I,X(I)
 2010   FORMAT(' X_2_( ',I3,' ) = ',1PD13.5)
   90 CONTINUE
      RETURN
      END
**********************************************************************
*    Gaussの消去法(部分軸選択無)による非対称行列Aを含む線形システム *
*    解法サブルーチン    AX=B                                        *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数     *
*    A(NE,NE) : AX=B のマトリックス A                                *
*    B(NE) : 線形システム AX=B のマトリックスB                       *
```

```
*    X(NE) : 線形システム AX=B のマトリックスX ---> これを求める    *
**********************************************************************
      SUBROUTINE DFG (A,B,X,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
* 特異性の判定値
      EPS = 1.0D-50
* 前進消去
      DO 10 K = 1, NE
*       除算を行う係数が小さく計算不可能なとき : 行列は特異
        IF ( ABS( A(K, K) ). LE. EPS ) THEN
          WRITE (*,2000) K
 2000     FORMAT (' Matrix is singular at K = ', I4)
          STOP
        END IF
        DO 20 I = K+1, NE
          A(K,I)=A(K,I)/A(K,K)
   20   CONTINUE
        B(K) = B(K)/A(K,K)
        DO 30 I = K+1, NE
          DO 40 J = K+1, NE
            A(I,J)=A(I,J)-A(I,K)*A(K,J)
   40     CONTINUE
          B(I)=B(I)-A(I,K)*B(K)
   30   CONTINUE
   10 CONTINUE
* 後退代入
      X(NE) = B(NE)
      DO 50 I = NE-1, 1, -1
        T = B(I)
        DO 60 J = I+1, NE
          T = T - A( I, J ) * X(J)
   60   CONTINUE
        X(I) = T
   50 CONTINUE
* 結果出力
      DO 70 I = 1,NE
        WRITE(6,2010) I,X(I)
 2010   FORMAT(' X_3_( ',I3,' ) = ',1PD13.5)
   70 CONTINUE
      RETURN
      END
**********************************************************************
*    Gauss-Jordan法(部分軸選択付)による非対称行列Aを含む線形システム*
*    解法サブルーチン    AX=B                                        *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数    *
*    A(NE,NE) : AX=B のマトリックス A                                *
*    B(NE) : 線形システム AX=B のマトリックスB                       *
*    X(NE) : 線形システム AX=B のマトリックスX ---> これを求める    *
*    IPV(NE) : 部分軸選択用の配列                                    *
*             全部で NE 回の消去操作を行うが, 各消去段階で行う除算  *
```

```
*              について分母が最も大きくなるような列の番号が入る       *
**********************************************************************
      SUBROUTINE DGJP (A,B,X,NE,
     $                 IPV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
      DIMENSION IPV(NE)
* 部分軸選択による特異性の判定値
      EPS = 1.0D-50
* 部分軸選択用配列の初期設定
      DO 10 I = 1, NE
        IPV(I) = I
   10 CONTINUE
      DO 20 K = 1, NE
        L = K
        AIPV = ABS( A(IPV(L),K) )
*       部分軸選択
        DO 30 I = K+1, NE
          IF ( ABS( A(IPV(I),K) ). GT. AIPV ) THEN
            L = I
            AIPV = ABS( A(IPV(L),K) )
          END IF
   30   CONTINUE
*       部分軸選択を行った方がよいと判断したら入れ替えを行う
        IF (L.NE.K) THEN
          IPVEX = IPV(K)
          IPV(K) = IPV(L)
          IPV(L) = IPVEX
        END IF
*       部分軸選択を行っても計算不可能なとき
        IF ( ABS( A(IPV(K),K) ). LE. EPS ) THEN
          WRITE (*,2000) K
 2000     FORMAT (' Matrix is singular at K= ', I4)
          STOP
        END IF
* 消去過程
        DO 40 I = K+1, NE
          A(IPV(K),I)=A(IPV(K),I)/A(IPV(K),K)
   40   CONTINUE
        B(IPV(K)) = B(IPV(K))/A(IPV(K),K)
        DO 50 I = 1, NE
          IF (I.EQ.K) GO TO 50
          DO 60 J = K+1, NE
            A(IPV(I),J)=A(IPV(I),J)-A(IPV(I),K)*A(IPV(K),J)
   60     CONTINUE
          B(IPV(I))=B(IPV(I))-A(IPV(I),K)*B(IPV(K))
   50   CONTINUE
   20 CONTINUE
* 上の消去過程が終了すると, IX=Bとなり解はすぐ求まる..........式(15.21)
      DO 70 I = 1, NE
        X(I) = B(IPV(I))
```

```
   70 CONTINUE
* 結果出力
      DO 80 I = 1,NE
        WRITE(6,2010) I,X(I)
 2010   FORMAT(' X_4_( ',I3,' ) = ',1PD13.5)
   80 CONTINUE
      RETURN
      END
**********************************************************************
*     2次元(5点)差分用バンドマトリックス解法サブルーチン AX=B        *
*     2次元(5点)差分にて得られた規則的非対称行列Aを含んだ線形システム*
*     をGaussの消去法を用いて解くサブルーチン.（部分軸選択無）      *
*     高速解法のためにバンドマトリックス用にしてある.                 *
*     NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数    *
*     AT1(NE),AT2(NE),AT3(NE),AT4(NE),AT5(NE)                         *
*              i=3, j=3 (k=13)における, AT1,AT2,AT3,AT4,AT5           *
*                +-------------------------+                         *
*          (NY)5 |    |    |    |    |    |                          *
*                +-------------------------+                         *
*              4 |    |    |AT4 |    |    |    k=(I-1)*NY+J          *
*                +-------------------------+                         *
*              3 |    | AT1|AT3 |AT5 |    |    AT1(k)->AT(-NY,k)     *
*                +-------------------------+   AT2(k)->AT(- 1,k)     *
*              2 |    |    |AT2 |    |    |    AT3(k)->AT(  0,k)     *
*                +-------------------------+   AT4(k)->AT(+ 1,k)     *
*              1 |    |    |    |    |    |    AT5(k)->AT(+NY,k)     *
*                +-------------------------+   B(k)                  *
*      j   i=1     2    3    4    5(NX)                              *
*     AT(-NY:NY,NE) : 2次元差分近似による規則的非対称行列            *
*           -NY:NY->上の図で考えるとk=13のときAT1からAT5の通し番号は *
*                   AT1は k"-NY" : AT2は k"-1"   : AT3は k"+0"       *
*                   AT4は k"+1"  : AT5は k"+NY"                      *
*                   となる. この"と"で囲まれた値が-NY:NYである.      *
*                   なお, これ以外のAT((-4,-3,-2,2,3,4),NE)はゼロ    *
*                   となる                                           *
*           NE->上述のようなkは全部で1からNE                         *
*     B(NE) : 線形システム AX=B のマトリックスB                      *
*     X(NE) : 線形システム AX=B のマトリックスX ---> これを求める    *
**********************************************************************
      SUBROUTINE DGBND (AT1,AT2,AT3,AT4,AT5,B,X,NY,NE,
     $                  AT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AT1(NE),AT2(NE),AT3(NE),AT4(NE),AT5(NE)
      DIMENSION AT(-NY:NY,NE),B(NE),X(NE)
* マトリックスATのゼロクリア
      DO 10 INE = 1,NE
        DO 20 I = -NY,NY
          AT(I,INE) = 0.0D0
   20   CONTINUE
   10 CONTINUE
* AT1からAT5を AT に格納
```

```
      DO 30 INE = 1,NE
        AT(-NY,INE) = AT1(INE)
        AT( -1,INE) = AT2(INE)
        AT(  0,INE) = AT3(INE)
        AT(  1,INE) = AT4(INE)
        AT( NY,INE) = AT5(INE)
   30 CONTINUE
* 前進消去
      DO 40 I = 1,NE-1
        IF ( I.LE.NE-NY ) THEN
          DO 50 J = 1,NY
            AA = AT(-J,I+J)/AT(0,I)
            B(I+J) = B(I+J) - B(I)*AA
            N = 1
            DO 60 K = -J+1,NY-J
              AT(K,I+J) = AT(K,I+J)-AT(N,I)*AA
              N = N + 1
   60       CONTINUE
   50     CONTINUE
        ELSE
          DO 70 J = 1,NE-I
            AA = AT(-J,I+J)/AT(0,I)
            B(I+J) = B(I+J) - B(I)*AA
            N = 1
            DO 80 K = -J+1,NE-I-J
              AT(K,I+J) = AT(K,I+J)-AT(N,I)*AA
              N = N + 1
   80       CONTINUE
   70     CONTINUE
        END IF
   40 CONTINUE
*係数行列の特異性を判定
      IF ( DABS(AT(0,NE)).LE.1.0D-20 ) THEN
        WRITE (6,*) ' Matrix is singular : |A(0,NE)| < 1E-20 '
      END IF
* 後退代入
      X(NE) = B(NE) / AT(0,NE)
      DO 90 I = NE-1,1,-1
        S = 0.0D0
        IF ( I.GT.NE-NY ) THEN
          DO 100 N = 1,NE-I
            S = S + AT(N,I)* X(I+N)
  100     CONTINUE
          X(I) = ( B(I)-S ) / AT(0,I)
        ELSE
          DO 110 N = 1,NY
            S = S + AT(N,I) * X(I+N)
  110     CONTINUE
          X(I) = ( B(I)-S ) / AT(0,I)
        END IF
   90 CONTINUE
```

```
* 結果出力
      DO 120 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_5_( ',I3,' ) = ',1PD13.5)
  120 CONTINUE
      RETURN
      END
*********************************************************************
*                  反復法 : 解法 6 - 9
*********************************************************************
*    Jacobi法による非対称行列 A を含む線形システム解法サブルーチン  *
*                        AX=B                                      *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数    *
*    A(NE,NE) : AX=B のマトリックス A                               *
*    B(NE) : 線形システム AX=B のマトリックスB                      *
*    X(NE) : 旧値 ; XN(NE) : 新値 ---> これを求める                 *
*********************************************************************
      SUBROUTINE HJACOB (A,B,XN,NE
     $                    X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE),XN(NE)
* 最大繰り返し回数(NITR)と収束判定値(EITR)の設定
      NITR = 200
      EITR = 1.0D-9
* X のゼロクリア
      DO 10 I = 1,NE
        X(I) = 0.0D0
   10 CONTINUE
* Bの2乗ノルムの計算
      BNORM = 0.0D0
      DO 20 I = 1,NE
        BNORM = BNORM + B(I)**2
   20 CONTINUE
* ITR : 繰り返し回数のカウンタ
      ITR = 0
* 繰り返しのための戻り点
  700 CONTINUE
* 先に得られた値を旧値に設定し直す
      ITR = ITR + 1
      DO 30 I = 1,NE
        X(I) = XN(I)
   30 CONTINUE
* 新値の計算
      DO 40 I = 1, NE
        SUM = 0.0D0
        DO 50 J = 1, NE
          IF (I.EQ.J) GO TO 50
          SUM = SUM + A(I,J)*X(J)
   50   CONTINUE
        XN(I) = ( B(I)-SUM )/A(I,I)
   40 CONTINUE
```

```
* 収束判定のためのノルムの計算
*     RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
      RNORM = 0.0D0
      DO 60 I = 1,NE
        SUM = 0.0D0
        DO 70 J = 1,NE
          SUM = SUM + A(I,J)*XN(J)
   70   CONTINUE
        RNORM = RNORM + (B(I)-SUM)**2
   60 CONTINUE
*     残差の計算 ZANSA = || R || / || B ||
      ZANSA = DSQRT(RNORM/BNORM)
* 収束判定
      IF (ITR.LE.NITR) THEN
        IF (ZANSA.GT.EITR) THEN
          GO TO 700
        ELSE
          WRITE (6,*) 'Converged : Total ITR = ',ITR
        END IF
      ELSE
        WRITE (6,*) ' Not converged ! '
      END IF
* 結果出力
      DO 80 I = 1,NE
        WRITE(6,2000) I,XN(I)
 2000   FORMAT(' X_6_( ',I3,' ) = ',1PD13.5)
   80 CONTINUE
      RETURN
      END
*********************************************************************
*   point-SOR法による非対称行列 A を含む線形システム解法サブルーチン*
*    (緩和係数OMGを1とするとGauss-Seidel法)  AX=B                  *
*    NX : x方向格子分割数; NY : y方向格子分割数; NE : 総格子点数    *
*    A(NE,NE) : AX=B のマトリックス A                               *
*    B(NE) : 線形システム AX=B のマトリックスB                      *
*    X(NE) : マトリックス X の1次元配列                             *
*********************************************************************
      SUBROUTINE HSOR (A,B,X,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
* 最大繰り返し回数(NITR)と収束判定値(EITR)の設定
      NITR = 200
      EITR = 1.0D-9
* 緩和係数の設定 ( OMG=1.0D0ならGauss-Seidel法 )
      WRITE (6,*) ' Input OMG ( if Gauss-Seidel, OMG=1.0D0 ) ---> '
      READ (5,*) OMG
* Bの2乗ノルムの計算
      BNORM = 0.0D0
      DO 10 I = 1,NE
        BNORM = BNORM + B(I)**2
   10 CONTINUE
```

```
* ITR : 繰り返し回数のカウンタ
      ITR = 0
* 繰り返しのための戻り点
  700 CONTINUE
      ITR = ITR + 1
      DO 20 I = 1, NE
        XOLD = X(I)
        SUM = 0.0D0
        DO 30 J = 1, NE
          IF (I.EQ.J) GO TO 30
          SUM = SUM + A(I,J)*X(J)
   30   CONTINUE
        XNEW = ( B(I)-SUM )/A(I,I)
        X(I) = XOLD + OMG * ( XNEW - XOLD )
   20 CONTINUE
* 収束判定のためのノルムの計算
*     RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
      RNORM = 0.0D0
      DO 40 I = 1,NE
        SUM = 0.0D0
        DO 50 J = 1,NE
          SUM = SUM + A(I,J)*X(J)
   50   CONTINUE
        RNORM = RNORM + (B(I)-SUM)**2
   40 CONTINUE
*     残差の計算 ZANSA = || R || / || B ||
      ZANSA = DSQRT(RNORM/BNORM)
* 収束判定
      IF (ITR.LE.NITR) THEN
        IF (ZANSA.GT.EITR) THEN
          GO TO 700
        ELSE
          WRITE (6,*) 'Converged : Total ITR = ',ITR
        END IF
      ELSE
        WRITE (6,*) ' Not converged ! '
      END IF
* 結果出力
      DO 60 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_7_( ',I3,' ) = ',1PD13.5)
   60 CONTINUE
      RETURN
      END
*******************************************************************
*  point-SOR 法による非対称行列 A を含む線形システム解法サブルーチン*
*  (バンドマトリックス用)                                        *
*    線形システム ---> A_{i,j} X_{i} = B_{i}                       *
*    A_{i,j} ---> A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)             *
*    B : 既知ベクトル                                             *
*    X : 未知ベクトル ---> これを求める                           *
```

```
* [変数の説明]                                                       *
*    NX:x方向格子分割数; NY:y方向格子分割数; NE:総格子点数= NX*NY    *
*    NITR : 最大反復回数 ; EITR : 収束判定値                         *
*********************************************************************
      SUBROUTINE SORBND (A1,A2,A3,A4,A5,B,X,NY,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),X(NE)
* 最大繰り返し回数(NITR)と収束判定値(EITR)の設定
      NITR = 200
      EITR = 1.0D-9
* 緩和係数の設定 ( OMG=1.0D0ならGauss-Seidel法 )
      WRITE (6,*) ' Input OMG ( if Gauss-Seidel, OMG=1.0D0 ) ---> '
      READ (5,*) OMG
* Bの2乗ノルムの計算
      BNORM = 0.0D0
      DO 10 I = 1,NE
        BNORM = BNORM + B(I)**2
   10 CONTINUE
* ITR : 繰り返し回数のカウンタ
      DO 20 ITR = 1,NITR
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
        RNORM = 0.0D0
*       A3,A4,A5の範囲
        I=1
          XOLD = X(I)
          SUM =  A4(I)*X(I+1)+A5(I)*X(I+NY)
          XNEW = ( B(I)-SUM )/A3(I)
          X(I) = XOLD + OMG * ( XNEW - XOLD )
*       A2,A3,A4,A5の範囲
        DO 30 I=2,NY
          XOLD = X(I)
          SUM =  A2(I)*X(I-1)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          XNEW = ( B(I)-SUM )/A3(I)
          X(I) = XOLD + OMG * ( XNEW - XOLD )
   30   CONTINUE
*       A1 - A5 の範囲
        DO 40 I=NY+1,NE-NY
          XOLD = X(I)
          SUM = A1(I)*X(I-NY)+A2(I)*X(I-1)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          XNEW = ( B(I)-SUM )/A3(I)
          X(I) = XOLD + OMG * ( XNEW - XOLD )
   40   CONTINUE
*       A1 - A4 の範囲
        DO 50 I=NE-NY+1,NE-1
          XOLD = X(I)
          SUM = A1(I)*X(I-NY)+A2(I)*X(I-1)+A4(I)*X(I+1)
          XNEW = ( B(I)-SUM )/A3(I)
          X(I) = XOLD + OMG * ( XNEW - XOLD )
   50   CONTINUE
*       A1 - A3 の範囲
        I=NE
```

```
            XOLD = X(I)
            SUM = A1(I)*X(I-NY)+A2(I)*X(I-1)
            XNEW = ( B(I)-SUM )/A3(I)
            X(I) = XOLD + OMG * ( XNEW - XOLD )
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
          RNORM= 0.0D0
          DO 60 I = 1,NE
            SUM = 0.0D0
            IF (I.EQ.1) THEN
              SUM=A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
            ELSE IF (I.GE.2. AND. I.LE.NY) THEN
              SUM=A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
            ELSE IF (I.GE.NY+1. AND. I.LE.NE-NY) THEN
              SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
     $           +A4(I)*X(I+1)+A5(I)*X(I+NY)
            ELSE IF (I.GE.NE-NY+1. AND. I.LE.NE-1) THEN
              SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)
            ELSE IF (I.EQ.NE) THEN
              SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
            END IF
            RNORM= RNORM + (B(I)-SUM)**2
   60     CONTINUE
*       残差の計算 ZANSA = || R || / || B ||
          ZANSA = DSQRT(RNORM/BNORM)
*       収束判定
          IF (ZANSA.LE.EITR) THEN
            WRITE (6,*) 'Converged : Total ITR = ',ITR
            GO TO 700
          END IF
   20 CONTINUE
* NITRまで計算しても収束せず
        WRITE (6,*) ' Not converged ! '
* 収束と判定されたときの分岐点
  700 CONTINUE
* 結果出力
        DO 70 I = 1,NE
          WRITE(6,2000) I,X(I)
 2000     FORMAT(' X_8_( ',I3,' ) = ',1PD13.5)
   70 CONTINUE
        RETURN
        END
*********************************************************************
*  line-SOR 法による非対称行列 A を含む線形システム解法サブルーチン *
*  (バンドマトリックス用)                                           *
*    線形システム ---> A_{i,j} X_{i} = B_{i}                        *
*    A_{i,j} ---> AT1(NE),AT2(NE),AT3(NE),AT4(NE),AT5(NE)           *
*    B -> BX(NE) : 既知ベクトル                                     *
*    X -> XN(NE) : 未知ベクトル ---> これを求める                   *
* [トーマス法のための係数行列]                                      *
*    x,y方向に陰的に離散化された結果を以下のように表す.             *
*  |B(1) C(1) 0    0 ...            |X(1)   |  |D(1)   |            *
```

```
*  |A(2) B(2) C(2) 0 ...             |X(2)   |  |D(2)    |            *
*  |0    A(3) B(3) C(3) 0 ...        |X(3)   |= |D(3)    |            *
*  |              ...                |...    |  |...     |            *
*  |0   ...   A(NE-1) B(NE-1) C(NE-1)|X(NE-1)|  |D(NE-1)|             *
*  |0    0    0 ....  A(NE  ) B(NE)  |X(NE)  |  |D(NE)   |            *
* [変数の説明]                                                        *
*    NX:x方向格子分割数; NY:y方向格子分割数; NE:総格子点数=NX*NY      *
*    NITR : 最大反復回数; EITR : 収束判定値                           *
*    OMG : 緩和係数. 1.0で十分.                                       *
*           注意：Point-SORと異なり, あまり大きくしすぎると発散する *
* [配列の説明]                                                        *
* XN...各方向への掃引後のX(番号付けは不変)                            *
*      はじめにこのサブルーチンへ渡されるXでもある                    *
* X1...各方向への掃引後のX(番号付けは軸方向に異なる)                  *
* X0...各方向への掃引前のX(番号付けは不変)                            *
**********************************************************************
      SUBROUTINE LBLBND (AT1,AT2,AT3,AT4,AT5,BX,XN,NX,NY,NE,
     $                   X1,X0,A,B,C,D,U,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AT1(NE),AT2(NE),AT3(NE),AT4(NE),AT5(NE)
      DIMENSION XN(NE),X1(NE),BX(NE),X0(NE)
      DIMENSION A(NE),B(NE),C(NE),D(NE),U(NE),Y(NE)
* 最大繰り返し回数(NITR)と収束判定値(EITR)の設定
      NITR = 200
      EITR = 1.0D-9
* 緩和係数の設定 ( OMG=1.0D0ならGauss-Seidel法 )
      WRITE (6,*) ' Input OMG ( if Gauss-Seidel, OMG=1.0D0 ) ---> '
      READ (5,*) OMG
* Bの2乗ノルムの計算
      BNORM = 0.0D0
      DO 10 I = 1,NE
        BNORM = BNORM + BX(I)**2
   10 CONTINUE
*ITR : 繰り返し回数のカウンタ
      DO 20 ITR=1,NITR
*       x 軸方向への掃引 : トーマス法による.........................式(15.33)
        INX = 1
        DO 100 IY = 1,NY
          DO 110 IX = 1,NX
            INY = IY + (IX-1)*NY
*           トーマス法のための係数A,B,C,Dの設定 : XNは最新のX
            A(INX) = AT1(INY)
            B(INX) = AT3(INY)
            C(INX) = AT5(INY)
            D(INX) = BX(INY)
            IF (INY-1.GE.1) THEN
              D(INX)=D(INX)-AT2(INY)*XN(INY-1)
            END IF
            IF (INY+1.LE.NE) THEN
              D(INX)=D(INX)-AT4(INY)*XN(INY+1)
            END IF
```

```
*           トーマス法で答えを求める前のXNをX0に保存
            X0(INY) = XN(INY)
            INX = INX + 1
  110     CONTINUE
  100   CONTINUE
*       Ly=b を解く
        U(1) = C(1) / B(1)
        DO 120 J = 2,NE-1
        U(J) = C(J) / ( B(J)-A(J)*U(J-1) )..........................式(15.28)
  120   CONTINUE
        Y(1) = D(1) / B(1)
        DO 130 J = 2,NE
          Y(J) = ( D(J)-A(J)*Y(J-1) ) / ( B(J)-A(J)*U(J-1) ).........式(15.29)
  130   CONTINUE
*       Ux=y を解く
        X1(NE) = Y(NE)
        DO 140 J = NE-1,1,-1
          X1(J) = Y(J) - U(J)*X1(J+1)...............................式(15.30)
  140   CONTINUE
        INX = 1
        DO 150 IY = 1,NY
          DO 160 IX = 1,NX
            INY = IY + (IX-1)*NY
*           得られたX1と反復前のX0により最新のXNを緩和
            XN(INY)=(1.0D0-OMG)*X0(INY)+OMG*X1(INX)
            INX = INX + 1
  160     CONTINUE
  150   CONTINUE
*       y 軸方向への掃引 : トーマス法による.........................式(15.34)
        INY = 1
        DO 200 IX = 1,NX
          DO 210 IY = 1,NY
            INX = IX + (IY-1)*NX
*           トーマス法のための係数A,B,C,Dの設定 : XNは最新のX
            A(INY) = AT2(INY)
            B(INY) = AT3(INY)
            C(INY) = AT4(INY)
            D(INY) = BX(INY)
            IF (INY-NY.GE.1) THEN
              D(INY)=D(INY)-AT1(INY)*XN(INY-NY)
            END IF
            IF (INY+NY.LE.NE) THEN
              D(INY)=D(INY)-AT5(INY)*XN(INY+NY)
            END IF
*           トーマス法で答えを求める前のXNをX0に保存
            X0(INY) = XN(INY)
            INY = INY + 1
  210     CONTINUE
  200   CONTINUE
*       Ly=b を解く
        U(1) = C(1) / B(1)
```

```
        DO 220 J = 2,NE-1
          U(J) = C(J)/( B(J)-A(J)*U(J-1) )..........................式(15.28)
  220   CONTINUE
        Y(1) = D(1) / B(1)
        DO 230 J = 2,NE
          Y(J) = ( D(J)-A(J)*Y(J-1) ) / ( B(J)-A(J)*U(J-1) ).........式(15.29)
  230   CONTINUE
*       Ux=y を解く
        X1(NE) = Y(NE)
        DO 240 J = NE-1,1,-1
          X1(J) = Y(J) - U(J)*X1(J+1)................................式(15.30)
  240   CONTINUE
        INY = 1
        DO 250 IX = 1,NX
          DO 260 IY = 1,NY
*           得られたX1と反復前のXOにより最新のXNを緩和
            XN(INY)=(1.0D0-OMG)*XO(INY)+OMG*X1(INY)
            INY = INY + 1
  260     CONTINUE
  250   CONTINUE
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム : サブルーチンSORBNDの場合分けを参照
        RNORM= 0.0D0
        DO 310 I = 1,NE
          SUM = 0.0D0
          IF (I.EQ.1) THEN
            SUM=AT3(I)*XN(I)+AT4(I)*XN(I+1)+AT5(I)*XN(I+NY)
          ELSE IF (I.GE.2. AND. I.LE.NY) THEN
          SUM=AT2(I)*XN(I-1)+AT3(I)*XN(I)+AT4(I)*XN(I+1)+AT5(I)*XN(I+NY)
          ELSE IF (I.GE.NY+1. AND. I.LE.NE-NY) THEN
            SUM=AT1(I)*XN(I-NY)+AT2(I)*XN(I-1)+AT3(I)*XN(I)
     $         +AT4(I)*XN(I+1)+AT5(I)*XN(I+NY)
          ELSE IF (I.GE.NE-NY+1. AND. I.LE.NE-1) THEN
          SUM=AT1(I)*XN(I-NY)+AT2(I)*XN(I-1)+AT3(I)*XN(I)+AT4(I)*XN(I+1)
          ELSE IF (I.EQ.NE) THEN
            SUM=AT1(I)*XN(I-NY)+AT2(I)*XN(I-1)+AT3(I)*XN(I)
          END IF
          RNORM= RNORM + (BX(I)-SUM)**2
  310   CONTINUE
*       残差の計算 ZANSA = || R || / || B ||
        ZANSA = DSQRT(RNORM/BNORM)
*       収束判定
        IF (ZANSA.LE.EITR) THEN
          WRITE (6,*) 'Converged : Total ITR = ',ITR
          GO TO 900
        END IF
   20 CONTINUE
*     NITRまで計算しても収束せず
      WRITE (6,*) ' Not converged ! '
* 収束と判定されたときの分岐点
  900 CONTINUE
* 結果出力
```

```
      DO 320 I = 1,NE
        WRITE(6,2000) I,XN(I)
 2000   FORMAT(' X_9_( ',I3,' ) = ',1PD13.5)
  320 CONTINUE
      RETURN
      END
*********************************************************************
*              クリロフ部分空間法 : 解法 10 - 13
*********************************************************************
*    共役残差(Conjugate Residual)法による非対称行列 A を含む       *
*  線形システム解法サブルーチン AX=B                                *
*［変数の説明］                                                     *
*    NX:x方向格子分割数; NY:y方向格子分割数; NE:総格子点数=NX*NY    *
*    NITR : 最大反復回数; EPSP : 収束判定値                         *
*［配列の説明］                                                     *
*    A(NE,NE) : AX=B のフルマトリックス A                           *
*    B(NE) : 線形システム AX=B のベクトル B                         *
*    X(NE) : 線形システム AX=B のベクトル X ---> これを求める       *
*    R(NE) : r_{k} = B - A x_{k}                                    *
*    P(NE) :  p_{k+1} = r_{k+1} + β_{k} p_{k}, p_{0} = r_{0}        *
*    AP(NE) : A * P , AR(NE) : A * R                                *
*********************************************************************
      SUBROUTINE KCR (A,B,X,NE,
     $                R,P,AP,AR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
* 作業用配列
      DIMENSION R(NE),P(NE),AP(NE),AR(NE)
* 最大繰り返し回数の設定
      NITR = 200
* 収束判定値(変数 ZANSA がこの値以下になれば NITR 以下でも収束と判定する)
      EITR = 1.0D-9
* 配列のゼロクリア
      DO 10 J=1, NE
        R(J) = 0.0D0
        P(J) = 0.0D0
        AP(J) = 0.0D0
        AR(J) = 0.0D0
   10 CONTINUE
* R に AX を代入
      CALL PROFMV (A,X,R,NE)
* Bのノルムの計算
      BNORM = 0.0D0
      DO 20 I = 1,NE
        BNORM = BNORM + B(I)**2
   20 CONTINUE
* r_{0}とp_{0}(初期値)の設定
      DO 30 I = 1, NE
*       r_{0} = B - AX の計算........................................式(15.103)
        R(I) = B(I) - R(I)
*       p_{0} = r_{0}................................................式(15.104)
```

```
        P(I) = R(I)
   30 CONTINUE
* APに A p_{0} を代入(以降のAPは 式(A) で求める)
      CALL PROFMV (A,P,AP,NE)
* 繰り返し計算
      DO 40 K = 1,NITR
*       ( r_{k}, A p_{k} )の計算 => RAP
        CALL PROVV (R,AP,RAP,NE)
*       ( A p_{k}, A p_{k} )の計算 => APAP
        CALL PROVV (AP,AP,APAP,NE)
*       α_{k} = ( r_{k}, A p_{k} ) / ( A p_{k}, A p_{k} )...........式(15.105)
        ALP = RAP / APAP
        DO 50 I = 1,NE
*         x_{k+1}=x_{k}+α_{k}p_{k}.................................式(15.106)
          X(I) = X(I) + ALP*P(I)
*         r_{k+1}=r_{k}-α_{k}Ap_{k}................................式(15.107)
          R(I) = R(I) - ALP*AP(I)
   50   CONTINUE
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
        RNORM = 0.0D0
        DO 60 I = 1,NE
          SUM = 0.0D0
          DO 70 J = 1,NE
            SUM = SUM + A(I,J)*X(J)
   70     CONTINUE
          RNORM = RNORM + (B(I)-SUM)**2
   60   CONTINUE
*       残差の計算 ZANSA = || R || / || B ||
        ZANSA = DSQRT(RNORM/BNORM)
*       ZANSA が EITR 以下なら収束とみなして結果を表示して 900 へ
        IF (ZANSA.LE.EITR) THEN
          WRITE (6,*) ' Converged : Total ITR = ',K
          GO TO 900
        ELSE
*         A r_{k+1} の計算 => AR(NE)
          CALL PROFMV (A,R,AR,NE)
*         ( A r_{k+1}, A p_{k} )の計算 => ARAP
          CALL PROVV( AR,AP,ARAP,NE)
*         β_{k} = - ( A r_{k+1}, A p_{k} ) / ( A p_{k}, A p_{k} )...式(15.108)
          BETA = - ARAP / APAP
          DO 80 I = 1, NE
*           p_{k+1} = r_{k+1} + β_{k} p_{k}.........................式(15.109)
            P(I) = R(I) + BETA*P(I)
*           A p_{k+1} = A r_{k+1} + β_{k}A p_{k}<-------式(A)
            AP(I) = AR(I) + BETA*AP(I)
   80     CONTINUE
        END IF
   40 CONTINUE
*     NITRまで計算しても収束せず
      WRITE (6,*) ' Not Converged ! '
*     収束と判定されたときの分岐点
```

```
  900 CONTINUE
*結果出力
      DO 90 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_10_( ',I3,' ) = ',1PD13.5)
   90 CONTINUE
      RETURN
      END
*********************************************************************
*    フルマトリックス A とベクトル B の積の計算サブルーチン AB=C    *
*    NE : 総格子点数                                                *
*********************************************************************
      SUBROUTINE PROFMV (A,B,C,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),C(NE)
      DO 10 I = 1, NE
        S = 0.0D0
        DO 20 J = 1, NE
          S = S + A(I,J)*B(J)
   20   CONTINUE
        C(I) = S
   10 CONTINUE
      RETURN
      END
*********************************************************************
*    ベクトル A とベクトル B の積の計算サブルーチン                 *
*                        AB=C                                       *
*［変数の説明］                                                     *
*    NE : 総格子点数(ベクトル A,B のサイズ)                          *
*    C  : A と B の積(スカラー)                                      *
*********************************************************************
      SUBROUTINE PROVV(A,B,C,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  A(NE),B(NE)
      C = 0.0D0
      DO 10 I=1,NE
        C = C + A(I)*B(I)
   10 CONTINUE
      RETURN
      END
*********************************************************************
*    共役残差(Conjugate Residual)法による非対称行列 A を含む        *
*  線形システム解法サブルーチン  (バンドマトリックス用)             *
*                        AX=B                                       *
*    A_{i,j} ---> A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)                *
*  B : 既知ベクトル                                                 *
*  X : 未知ベクトル ---> これを求める                               *
*［変数の説明］                                                     *
*    NX:x方向格子分割数; NY:y方向格子分割数; NE:総格子点数=NX*NY    *
*    NITR : 最大反復回数; EPSP : 収束判定値                         *
*［配列の説明］                                                     *
```

```
*    R(NE) : r_{k} = B - A x_{k}                                         *
*    P(NE) :  p_{k+1} = r_{k+1} + β_{k} p_{k}, p_{0} = r_{0}           *
*    AP(NE) : A * P                                                      *
*    AR(NE) : A * R                                                      *
*********************************************************************
      SUBROUTINE KCRBND (A1,A2,A3,A4,A5,B,X,NY,NE,
     $                   R,P,AP,AR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),X(NE)
* 作業用配列
      DIMENSION R(NE),P(NE),AP(NE),AR(NE)
* 最大繰り返し回数の設定
      NITR = 200
* 収束判定条件(変数 ZANSA がこの値以下になれば NITR 以下でも収束と判定する)
      EITR = 1.0D-9
* 配列のゼロクリア
      DO 10 J=1, NE
        R(J) = 0.0D0
        P(J) = 0.0D0
        AP(J) = 0.0D0
        AR(J) = 0.0D0
   10 CONTINUE
* R に AX を代入
      CALL PROBMV (A1,A2,A3,A4,A5,X,R,NY,NE)
* Bのノルムの計算
      BNORM = 0.0D0
      DO 20 I = 1,NE
        BNORM = BNORM + B(I)**2
   20 CONTINUE
* r_{0}とp_{0}(初期値)の設定
      DO 30 I = 1, NE
*       r_{0} = B - AX の計算........................................式(15.103)
        R(I) = B(I) - R(I)
*       p_{0} = r_{0}................................................式(15.104)
        P(I) = R(I)
   30 CONTINUE
* APに A p_{0} を代入(以降のAPは 式(A) で求める)
      CALL PROBMV (A1,A2,A3,A4,A5,P,AP,NY,NE)
* 繰り返し計算
      DO 40 K = 1,NITR
*       ( r_{k}, A p_{k} )の計算 => RAP
        CALL PROVV (R,AP,RAP,NE)
*       ( A p_{k}, A p_{k} )の計算 => APAP
        CALL PROVV (AP,AP,APAP,NE)
*       α_{k} = ( r_{k}, A p_{k} ) / ( A p_{k}, A p_{k} )...........式(15.105)
        ALP = RAP / APAP
        DO 50 I = 1,NE
*         x_{k+1}=x_{k}+α_{k}p_{k}...................................式(15.106)
          X(I) = X(I) + ALP*P(I)
*         r_{k+1}=r_{k}-α_{k}Ap_{k}..................................式(15.107)
          R(I) = R(I) - ALP*AP(I)
```

```
   50   CONTINUE
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム : サブルーチンSORBNDの場合分けを参照
        RNORM= 0.0D0
        DO 60 I = 1,NE
          SUM = 0.0D0
          IF (I.EQ.1) THEN
            SUM=A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.2. AND. I.LE.NY) THEN
            SUM=A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.NY+1. AND. I.LE.NE-NY) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
     $         +A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.NE-NY+1. AND. I.LE.NE-1) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)
          ELSE IF (I.EQ.NE) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
          END IF
          RNORM= RNORM + (B(I)-SUM)**2
   60   CONTINUE
*       残差の計算 ZANSA = || R || / || B ||
        ZANSA = DSQRT(RNORM/BNORM)
*       ZANSA が EITR 以下なら収束とみなして 700 へ
        IF (ZANSA.LE.EITR) THEN
          WRITE (6,*) ' Converged : Total ITR = ',K
          GO TO 700
        END IF
*       収束せずの場合
*       A r_{k+1} の計算 => AR(NE)
        CALL PROBMV (A1,A2,A3,A4,A5,R,AR,NY,NE)
*       ( A r_{k+1}, A p_{k} )の計算 => ARAP
        CALL PROVV( AR,AP,ARAP,NE)
*       β_{k} = - ( A r_{k+1}, A p_{k} ) / ( A p_{k}, A p_{k} ).....式(15.108)
        BETA = - ARAP / APAP
        DO 70 I = 1, NE
*         p_{k+1} = r_{k+1} + β_{k} p_{k}..........................式(15.109)
          P(I) = R(I) + BETA*P(I)
*         A p_{k+1} = A r_{k+1} + β_{k}A p_{k}<-------式(A)
          AP(I) = AR(I) + BETA*AP(I)
   70   CONTINUE
   40 CONTINUE
* NITR まで計算しても収束せず
      WRITE (6,*) ' Not Converged ! '
* 収束と判定されたときの分岐点
  700 CONTINUE
* 結果出力
      DO 80 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_11_( ',I3,' ) = ',1PD13.5)
   80 CONTINUE
      RETURN
      END
```

```
*********************************************************************
*    バンドマトリックス A とベクトル B の積の計算サブルーチン       *
*                         AB=C                                      *
*    A_{i,j} ---> A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)               *
*[変数の説明]                                                       *
*    NY : y方向格子分割数                                           *
*                 ---> バンドマトリックスとベクトルの積の計算で使用 *
*    NE : 総格子点数                                                *
*    C  : A と B の積(ベクトル)                                     *
*********************************************************************
      SUBROUTINE PROBMV (A1,A2,A3,A4,A5,B,C,NY,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),C(NE)
* サブルーチンSORBNDの場合分けを参照
      I=1
        C(I) = A3(I)*B(I)+A4(I)*B(I+1)+A5(I)*B(I+NY)
      DO 10 I=2,NY
        C(I) = A2(I)*B(I-1)+A3(I)*B(I)+A4(I)*B(I+1)+A5(I)*B(I+NY)
   10 CONTINUE
      DO 20 I=NY+1,NE-NY
        C(I) = A1(I)*B(I-NY)+A2(I)*B(I-1)+A3(I)*B(I)
     $      +A4(I)*B(I+1)+A5(I)*B(I+NY)
   20 CONTINUE
      DO 30 I=NE-NY+1,NE-1
        C(I) = A1(I)*B(I-NY)+A2(I)*B(I-1)+A3(I)*B(I)+A4(I)*B(I+1)
   30 CONTINUE
      I=NE
        C(I) = A1(I)*B(I-NY)+A2(I)*B(I-1 )+A3(I)*B(I)
      RETURN
      END
*********************************************************************
*  Bi-CGSTAB 法による非対称行列 A を含む線形システム解法サブルーチン*
*                         AX=B                                      *
*    B : 既知ベクトル                                               *
*    X : 未知ベクトル ---> これを求める                             *
*[変数の説明]                                                       *
*    NX : x方向格子分割数                                           *
*    NY : y方向格子分割数                                           *
*    NE : 総格子点数 = NX * NY                                      *
*    NITR : 最大反復回数                                            *
*    EPSP : 収束判定条件                                            *
*[配列の説明]                                                       *
*    T(NE) : t_{k} = r_{k} - α_{k} A p_{k}                          *
*    X(NE) : x_{k+1} = x_{k} + α_{k} p_{k} + ξ_{K} t_{k}           *
*    R(NE) : r_{k+1} = t_{k} - ξ_{k} A t_{k}                        *
*            r_{0} = B - A x_{0}                                    *
*    P(NE) : p_{k+1} = r_{k+1} + β_{k} ( p_{k}-ξ_{k} A p_{k} )     *
*            p_{0} = r_{0}                                          *
*    AP(NE) : A * P                                                 *
*    AR(NE) : A * T                                                 *
*********************************************************************
```

```
      SUBROUTINE KBICG (A,B,X,NE,
     $                  R,AP,AT,P,S,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NE,NE),B(NE),X(NE)
* 作業用配列
      DIMENSION R(NE),AP(NE),AT(NE),P(NE),S(NE),T(NE)
* 最大繰り返し回数の設定
      NITR = 200
* 収束判定条件(変数 ZANSA がこの値以下になれば NITR 以下でも収束と判定する)
      EITR = 1.0D-9
* Bのノルムの計算
      BNORM = 0.0D0
      DO 10 I = 1,NE
        BNORM = BNORM + B(I)**2
   10 CONTINUE
* 配列のゼロクリア
      DO 20 J=1, NE
        R(J) = 0.0D0
        AP(J) = 0.0D0
        AT(J) = 0.0D0
        P(J) = 0.0D0
        S(J) = 0.0D0
        T(J) = 0.0D0
   20 CONTINUE
* R に AX を代入
      CALL PROFMV (A,X,R,NE)
*r_{0}とp_{0}(初期値), そして s=r_{0} の設定
      DO 30 I=1,NE
        R(I) = B(I) - R(I)..........................................式(15.110)
        P(I) = R(I).................................................式(15.111)
        S(I) = R(I)
   30 CONTINUE
* 繰り返し計算
      DO 40 J =1,NITR
*       ( s, r_{k} ) の計算 => SR1
        CALL PROVV (S,R,SR1,NE)
*       A p_{k} の計算 => AP(NE)
        CALL PROFMV (A,P,AP,NE)
*       ( s, A p_{k} ) の計算 => SAP
        CALL PROVV (S,AP,SAP,NE)
*       α_{k} = ( s, r_{k} ) / ( s, A p_{k} )......................式(15.112)
        ALPHA = SR1/SAP
          DO 50 I=1,NE
*           t_{k} = r_{k} - α_{k} A p_{k}..........................式(15.113)
            T(I) = R(I) - ALPHA*AP(I)
   50     CONTINUE
*       A t_{k} の計算 => AT(NE)
        CALL PROFMV (A,T,AT,NE)
*       ( A t_{k}, t_{k} ) の計算 => ATT
        CALL PROVV (AT,T,ATT,NE)
*       ( A t_{k}, A t_{k} ) の計算 => ATAT
```

```
      CALL PROVV (AT,AT,ATAT,NE)
*       ξ_{k} = ( A t_{k}, t_{k} ) / ( A t_{k}, A t_{k} )...........式(15.114)
        XI = ATT/ATAT
        DO 60 I=1,NE
*         x_{k+1} = x_{k} + α_{k} p_{k} + ξ_{k} t_{k}..............式(15.115)
          X(I) = X(I) + ALPHA*P(I) + XI*T(I)
*         r_{k+1} = t_{k} - ξ_{k} A t_{k}..........................式(15.116)
          R(I) = T(I) - XI*AT(I)
   60   CONTINUE
* RNORM : 残差ベクトル(R=AX-B)の2乗ノルム
      RNORM = 0.0D0
      DO 70 I = 1,NE
        SUM = 0.0D0
        DO 80 M = 1,NE
          SUM = SUM + A(I,M)*X(M)
   80   CONTINUE
        RNORM = RNORM + (B(I)-SUM)**2
   70 CONTINUE
*     残差の計算 ZANSA = || R || / || B ||
        ZANSA = DSQRT(RNORM/BNORM)
*       ZANSA が EITR 以下なら収束とみなして 900 へ
        IF (ZANSA.LE.EITR) THEN
          WRITE (6,*) ' Converged : Total ITR = ',J
          GO TO 900
        END IF
*       収束せずの場合 : β_{k}と p_{k+1} を求めて繰り返し計算
*       ( s, r_{k+1} ) の計算 => SR2
        CALL PROVV (S,R,SR2,NE)
*       β_{k} = ( α_{k} / ξ_{k} ) * ( s, r_{k+1} )/( s, r_{k} )...式(15.117)
        BETA = (ALPHA / XI) * (SR2 / SR1)
        DO 90 I=1,NE
*         p_{k+1} = r_{k+1} + β_{k}( p_{k}-ξ_{k} A p_{k} ).........式(15.118)
          P(I) = R(I) + BETA * ( P(I) - XI*AP(I) )
   90   CONTINUE
   40 CONTINUE
*     NITR まで計算しても収束せず
      WRITE (6,*) ' Not Converged ! '
* 収束と判定されたときの分岐点
  900 CONTINUE
* 結果出力
      DO 100 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_12_( ',I3,' ) = ',1PD13.5)
  100 CONTINUE
      RETURN
      END
**********************************************************************
*  Bi-CGSTAB 法による非対称行列 A を含む線形システム解法サブルーチン*
*  (バンドマトリックス用)                                             *
*                            AX=B                                    *
*    A_{i,j} ---> A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)                  *
```

```
*    B : 既知ベクトル                                              *
*    X : 未知ベクトル ---> これを求める                            *
*[変数の説明]                                                      *
*    NX : x方向格子分割数                                          *
*    NY : y方向格子分割数                                          *
*    NE : 総格子点数 = NX * NY                                     *
*    NITR : 許容反復回数(in2d.mac)にて設定                         *
*    EPSP : 収束判定条件で用いる値(in2d.mac)にて設定               *
*[配列の説明]                                                      *
*    T(NE) : t_{k} = r_{k} - α_{k} A p_{k}                         *
*    X(NE) : x_{k+1} = x_{k} + α_{k} p_{k} + ξ_{K} t_{k}           *
*    R(NE) : r_{k+1} = t_{k} - ξ_{k} A t_{k}                       *
*            r_{0} = B - A x_{0}                                   *
*    P(NE) : p_{k+1} = r_{k+1} + β_{k} ( p_{k}-ξ_{k} A p_{k} )     *
*            p_{0} = r_{0}                                         *
*    AP(NE) : A * P                                                *
*    AT(NE) : A * T                                                *
********************************************************************
      SUBROUTINE KBIBND (A1,A2,A3,A4,A5,B,X,NY,NE,
     $                   R,AP,AT,P,S,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),X(NE)
* 作業用配列
      DIMENSION R(NE),AP(NE),AT(NE),P(NE),S(NE),T(NE)
* 最大繰り返し回数の設定
      NITR = 200
* 収束判定条件(変数 ZANSA がこの値以下になれば NITR 以下でも収束と判定する)
      EITR = 1.0D-9
* Bのノルムの計算
      BNORM = 0.0D0
      DO 10 I = 1,NE
        BNORM = BNORM + B(I)**2
   10 CONTINUE
* 配列のゼロクリア
      DO 20 J=1, NE
        R(J) = 0.0D0
        AP(J) = 0.0D0
        AT(J) = 0.0D0
        P(J) = 0.0D0
        S(J) = 0.0D0
        T(J) = 0.0D0
   20 CONTINUE
* R に AX を代入
      CALL PROBMV (A1,A2,A3,A4,A5,X,R,NY,NE)
* r_{0}とp_{0}(初期値), そして s=r_{0} の設定
      DO 30 I=1,NE
        R(I) = B(I) - R(I)........................................式(15.110)
        P(I) = R(I)...............................................式(15.111)
        S(I) = R(I)
   30 CONTINUE
* 繰り返し計算
```

```
      DO 40 J =1,NITR
*       ( s, r_{k} ) の計算 => SR1
        CALL PROVV (S,R,SR1,NE)
*       A p_{k} の計算 => AP(NE)
        CALL PROBMV (A1,A2,A3,A4,A5,P,AP,NY,NE)
*       ( s, A p_{k} ) の計算 => SAP
        CALL PROVV (S,AP,SAP,NE)
*       α_{k} = ( s, r_{k} ) / ( s, A p_{k} )......................式(15.112)
        ALPHA = SR1/SAP
        DO 50 I=1,NE
*         t_{k} = r_{k} - α_{k} A p_{k}............................式(15.113)
          T(I) = R(I) - ALPHA*AP(I)
   50   CONTINUE
*       A t_{k} の計算 => AT(NE)
        CALL PROBMV (A1,A2,A3,A4,A5,T,AT,NY,NE)
*       ( A t_{k}, t_{k} ) の計算 => ATT
        CALL PROVV (AT,T,ATT,NE)
*       ( A t_{k}, A t_{k} ) の計算 => ATAT
        CALL PROVV (AT,AT,ATAT,NE)
*       ξ_{k} = ( A t_{k}, t_{k} ) / ( A t_{k}, A t_{k} )...........式(15.114)
        XI = ATT/ATAT
        DO 60 I=1,NE
*         x_{k+1} = x_{k} + α_{k} p_{k} + ξ_{k} t_{k}..............式(15.115)
          X(I) = X(I) + ALPHA*P(I) + XI*T(I)
*         r_{k+1} = t_{k} - ξ_{k} A t_{k}..........................式(15.116)
          R(I) = T(I) - XI*AT(I)
   60   CONTINUE
*       RNORM : 残差ベクトル(R=AX-B)の2乗ノルム : サブルーチンSORBNDの場合分けを参照
        RNORM= 0.0D0
        DO 70 I = 1,NE
          SUM = 0.0D0
          IF (I.EQ.1) THEN
            SUM=A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.2. AND. I.LE.NY) THEN
            SUM=A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.NY+1. AND. I.LE.NE-NY) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
     $           +A4(I)*X(I+1)+A5(I)*X(I+NY)
          ELSE IF (I.GE.NE-NY+1. AND. I.LE.NE-1) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)+A4(I)*X(I+1)
          ELSE IF (I.EQ.NE) THEN
            SUM=A1(I)*X(I-NY)+A2(I)*X(I-1)+A3(I)*X(I)
          END IF
          RNORM= RNORM + (B(I)-SUM)**2
   70   CONTINUE
*       残差の計算 ZANSA = || R || / || B ||
        ZANSA = DSQRT(RNORM/BNORM)
*       ZANSA が EITR 以下なら収束とみなして 900 へ
        IF (ZANSA.LE.EITR) THEN
          WRITE (6,*) ' Converged : Total ITR = ',J
          GO TO 900
```

```
        END IF
*       収束せずの場合
*       ( s, r_{k+1} ) の計算 => SR2
        CALL PROVV (S,R,SR2,NE)
*       β_{k} = ( α_{k} / ξ_{k} ) * ( s, r_{k+1} )/( s, r_{k} )...式(15.117)
        BETA = (ALPHA / XI) * (SR2 / SR1)
        DO 90 I=1,NE
*         p_{k+1} = r_{k+1} + β_{k}( p_{k}-ξ_{k} A p_{k} ).........式(15.118)
          P(I) = R(I) + BETA * ( P(I) - XI*AP(I) )
   90   CONTINUE
   40 CONTINUE
* NITR まで計算しても収束せず
      WRITE (6,*) ' Not Converged ! '
* 収束と判定されたときの分岐点
  900 CONTINUE
* 結果出力
      DO 100 I = 1,NE
        WRITE(6,2000) I,X(I)
 2000   FORMAT(' X_13_( ',I3,' ) = ',1PD13.5)
  100 CONTINUE
      RETURN
      END
```