```
**********************************************************************
* ファイル名  : SIMPLE2D.FOR                                         *
* タイトル    : SIMPLE法による2次元熱流動解析プログラム              *
* 製作者      : 平野　博之                                           *
* 所属        : 岡山理科大学 工学部 バイオ・応用化学科               *
* 製作日      : 2011.11.01                                           *
* 言語        : FORTRAN (FORTRAN77でも実行可能)                      *
**********************************************************************
*                                                                    *
*   本プログラムでは, SIMPLE法のアルゴリズムに基づき,対流項・拡散項*
*  風上法あるいはべき乗法を用いて離散化している. 高精度の計算を行う *
*  場合などは, 適宜変更のこと.                                       *
*  格子分割数を変更するときは, PARAMETER文中のNX0,NY0,NEU0,NEV0,NEP0*
*  をすべて変更.                                                     *
*                                                                    *
*  ●変数                                                            *
*          →                                                        *
*   速度 V=(U,V,W),   圧力 P,   温度 T                               *
*                                                                    *
*  ●基礎方程式について                                              *
*         →                                                         *
*   ∇・V ＝０                                                       *
*      →                                                            *
*   ＤＶ                    ２→                                     *
*   ーー＝ー∇Ｐ＋ＶＩＳ（∇  Ｖ）＋ＢＵＯ×Ｔ(0,1)                  *
*   Ｄｔ         ~~~~~~           ~~~~~~                             *
*   ＤＴ          ２                                                 *
*   ーー＝ＡＬＰ（∇  Ｔ）                                           *
*   Ｄｔ ~~~~~~                                                      *
*   方程式に応じて,"BIS","BUO","ALP"を定義して与える.                *
*                                                                    *
*  ●格子分割について (NX=3, NY=3の例)                               *
*       仮想セル 左側境界                  右側境界 仮想セル         *
*             ↓      ↓                          ↓  ↓             *
*           +--------+-------+-------+-------+-------+               *
*           |P(0,NY+1)       |       |       ||P(NX+1,NY+1)          *
* 仮想セル→|   ・   →  ・   |  ・   |  ・  →  ・  |←仮想セル*
*           |      U(0,NY+1) |       |     U(NX,NY+1) |              *
* 上方境界→+===↑===+=======+=======+=======+==↑===+←上方境界*
* (IY=NY)   |V(0,NY) ||      |       |       ||V(NX+1,NY)            *
*           |   ・   ||  ・  |  ・   |  ・   ||  ・  |               *
*           |        ||      |       |       ||      |               *
*           +--------+-------+-------+-------+-------+               *
*           |        ||      |       |       ||      |               *
*           |   ・   ||  ・  |  ・   |  ・   ||  ・  |               *
*           |        ||      |       |       ||      |               *
*           +--------+-------+-------+-------+-------+               *
*           |        ||      |       |       ||      |               *
*           |   ・   ||  ・  |  ・   |  ・   ||  ・  |               *
*           | V(0,0) ||      |       |       ||V(NX+1,0)             *
* 下方境界→+===↑===+=======+=======+=======+==↑===+←下方境界*
```

```
* (IY=0)     | P(0,0)  ||         |        |          ||P(NX+1,0)          *
* 仮想セル→|    ・    →   ・   |    ・   |    ・    →   ・    |←仮想セル*
*            |       U(0,0)       |        |      U(NX,0)     |            *
*      y     +--------+-------+--------+--------+-------+            *
*      ↑         ↑      ↑                          ↑     ↑               *
*   ↓  |   仮想セル  左側境界                    右側境界 仮想セル      *
*   g  |              (IX=0)                     (IX=NX)                 *
*      +ー→x                                                           *
*                                                                        *
*  ●スタッガードメッシュについて                                        *
*                               ←     DX     →                         *
*    +-----------+-----------+------------+                             *
*    |           |           |            | ↑                           *
*    |           |  P(i,j+1) |            |                             *
*    |           |     ・    |            | DY                          *
*    |           |           |            |                             *
*    |           |  V(i,j)   |            | ↓                           *
*    +-----------+-----↑-----+------------+                             *
*    |           |           |            |                             *
*    |  P(i-1,j) |  P(i,j)   |  P(i+1,j)  | → U 定義点                 *
*    |     ・    →     ・    →     ・     | ↑ V 定義点                 *
*    |        U(i-1,j)    U(i,j)          | ・ P,T 定義点              *
*    |           |  V(i,j-1) |            | (注)                        *
*    +-----------+-----↑-----+------------+ プログラム中のTは,         *
*    |           |           |            | 本文中ではΘとなって*
*    |           |  P(i,j-1) |            | いる.                       *
*    |           |     ・    |            |                             *
*    |           |           |            |                             *
*    |           |           |            |                             *
*    +-----------+-----------+------------+                             *
*                                                                        *
*    上の格子分割図を参考に, たとえば, NX=20,NY=20の場合, PARAMETER *
*  文の変数は以下のように計算する.                                      *
*  NEUO: 境界を除く未知のUの数  : (NX-1) *  NY     = 380                *
*  NEVO: 境界を除く未知のVの数  :  NX    * (NY-1) = 380                *
*  NEPO: 境界を除く未知のP,Tの数:  NX    *  NY     = 400                *
*                                                                        *
*  ●パラメーターファイル(ｉｎ２ｄ．ｓｉｍ)について                    *
*    プログラムを実行すると,”ｉｎ２ｄ．ｓｉｍ”というパラメータ   *
*    ファイルを読みに行くので, あらかじめ作成しておく.                 *
*                                                                        *
*  ”in2d.sim”のリスト                                                  *
* 1: U.1......U の計算結果を出力するファイル名                          *
* 2: V.1......V の計算結果を出力するファイル名                          *
* 3: P.1......PS(推測した圧力P*)とPD(圧力補正値P')の計算結果を出力 *
*             するファイル名                                            *
* 4: T.1......Θの計算結果を出力するファイル名                          *
* 5: U.0......継続計算用のUの入力データ        新規の計算を行う場合は *
* 6: V.0......継続計算用のVの入力データ        これらのファイルは不要 *
* 7: P.0......継続計算用のPとPDの入力データ   であるが, 削除は不可.  *
* 8: T.0......継続計算用のΘの入力データ                                *
```

```
* 9: UVT.1....Tecplot(第3部で述べる可視化ソフト)の入力用データ       *
*10: +==========================+                                    *
*11: | ITYPE   1===>   isothermal |                                  *
*12: |         2===>nonisothermal |                                  *
*13: +==========================+                                    *
*14: ------------------------------------------------------------    *
*15: ITYPE   ICYCLE   NITR    NCYCLE   NCNT                          *
*16: 2       0        100000  10000    1000                          *
*17: ------------------------------------------------------------    *
*18: EPSP    EPSC    OMG     ALPHAP  ALPHAU  ALPHAV  ALPHAT          *
*19: 1.0e-6  1.0e-3  1.1e+0  0.5e0   0.5e0   0.5e0   0.5e0           *
*20: ------------------------------------------------------------    *
*21: DT       RE      PR       GR                                    *
*22: 1.0e-4   0.0e0   7.1e-1   1.0e5                                 *
*23: ------------------------------------------------------------    *
*24: DLX    DLY    IRELP  METHOD  NMT(1:UPWIND,ELSE:SHISUU)          *
*25: 1.0e0  1.0e0  2      5       2                                 *
*26: ------------------------------------------------------------    *
*                                                                   *
*    ITYPE....1:等温場(浮力項と温度場の計算をしない)   2:非等温場   *
*    ICYCLE...計算開始のサイクル数                                  *
*             0なら新規でプログラムにある初期条件にしたがって計算開始*
*             0以外の値なら継続計算                                 *
*              時刻 TIME = ICYCLE * DT                              *
*    NITR.....(U*,V*,P',T)の各線形システム解法(反復法とクリロフ部分 *
*              空間法)のための最大反復回数.                         *
*    NCYCLE...計算終了サイクル数                                    *
*    NCNT.....連続の式の収束判定の許容最大反復回数                  *
*    EPSP.....収束判定値（速度，圧力，温度の線形システム解法で      *
*              反復法とクリロフ部分空間法において使用．）           *
*    EPSC.....連続の式の収束判定値                                  *
*    OMG........線形システム解法のうち，(point,line - SOR)を        *
*                用いるときの緩和係数                               *
*    ALPHAP...圧力の緩和係数 -+  ある時間ステップ(ICYCLE)において   *
*    ALPHAU...Uの緩和係数     |->速度，圧力，温度を修正しながら     *
*    ALPHAV...Vの緩和係数     |  連続の式を満足する値を求めるとき   *
*    ALPHAT...温度の緩和係数 -+  に使用する(NCNT以下であれば収束)   *
*    DT.......時間刻み                                              *
*    RE.......レイノルズ数                                          *
*    PR.......プラントル数                                          *
*    GR.......グラスホフ数                                          *
*    DLX......解析領域の横幅(DLX=NX*DX)                             *
*                NX,DX...x方向の格子数，格子幅(等間隔)              *
*    DLY......解析領域の高さ(DLY=NY*DY)                             *
*                NY,DY...y方向の格子数，格子幅(等間隔)              *
*    IRELP....圧力の基準値と解の1次独立性の設定                     *
*              0 : 行わない                                         *
*                   直接法では特異行列の問題に遭遇                  *
*                   数学的には正しくない.                           *
*                   丸め誤差がなければ解は得られない.               *
*              1 : 圧力基準を反映した1次独立な解を求める.           *
```

```
*              2 : 圧力基準を設定 P(1,1)=0                            *
*                   1次従属の解のうちの1つを求めて,                   *
*                   圧力基準値P(1,1)=0を設定する.                     *
*   METHOD...圧力補正の線形システム解法に用いるアルゴリズム           *
*              すべてバンドマトリックス用に最適化してある             *
*              1: 直接法 -> ガウスの消去法                            *
*                  IRELP=1とする必要がある.(丸め誤差により,IREP=0,2 *
*                  としても解を得られる場合もあるが, 数学的には誤り)*
*              2: 反復法1 -> point-SOR 法                             *
*              3: 反復法2 -> line-SOR 法                              *
*                                 OMG=1で十分.大きくしすぎると発散.   *
*              4: クリロフ部分空間法1 -> 共役残差法                   *
*              5: クリロフ部分空間法2 -> Bi-CGSTAB法                  *
*   NMT......Simple法における対流・拡散項の離散化方法                 *
*              1:風上法 / 2:指数法                                    *
*   パラメータファイルの数値について,                                 *
*   FORTRAN プログラム.....できれば倍精度実数で与える"1.0D0,1.0d0" *
*             Cプログラムと共用させて"1.0E0,1.0e0"としても問題はない *
*   C プログラム....."1.0E0,1.0e0"として与える                       *
*   TECPLOT用データを除いた入出力ファイルは書式なし形式で, 使用      *
*   するコンパイラーに依存する. コンパイラーに依存しない形式にする *
*   には, 容量は増えるが書式付き形式に変更すればよい.                 *
*                                                                     *
*  ●圧力の相対性について                                             *
*   圧力補正の線形システムを解くにあたり, 本問題のような, 境界にお *
*   ける速度が既知の場合, 係数行列が特異(matrix singular)となり,    *
*   得られる解は一次従属となり, 無数の解が存在することとなる.       *
*   これに対する方策として,                                           *
*   (1) 直接法を用いるときは, 方程式の数を減らす必要がある.         *
*   サブルーチンPRESSにおいて, 基準として δP(1,1)=P(1,1)=0 と固定 *
*   できるようになっている.                                           *
*   (2) クリロフ部分空間法を含む反復法においては, 意識しなくてよい *
*   場合が多い.                                                       *
*   数学的に厳密にいえば, 一次従属な解のうちの1つを見つけるという   *
*   ことは, 正しいとはいえないが, 圧力の値は, 絶対値ではなく相対値 *
*   のみが問題となる.                                                 *
*   ただし, 計算を進行させてゆくにつれ, 圧力の絶対値が大きくなって *
*   ゆくような場合は, 一次従属な解のうちの1つを求めてから, 基準値を*
*   設定したほうがよい. はじめから, 基準値を設定して一次独立な解を *
*   得るには, 通常, 計算時間が長くなる.                             *
*                                                                     *
*  ●変数・配列の説明                                                 *
*   ICYCLE  -----> 時間進行のためのカウンタ                           *
*                   NCYCLEになれば計算を終了                          *
*   ITR     -----> 速度, 圧力, 温度の線形システム解法のための        *
*                   反復回数のカウンタ                                *
*                  (クリロフ部分空間法を含む反復法において使用)      *
*   ITR     -----> 線形システム解法のための反復回数のカウンタ         *
*   ITRCNT  -----> ある時刻における連続の式の評価回数のカウンタ       *
*                   NCNT回以下でEPSCの精度で連続の式を満たせば次の    *
*                   時間ステップへ                                    *
```

```
*     IX,IY    -----> 上の図を参照                                    *
*     UO,VO,TO-----> 前の時刻の値                                     *
*     U ,V ,T -----> 新たな圧力補正を用いて計算された値               *
*     PS       -----> 圧力推測値，あるいは補正を反映した圧力           *
*     PD       -----> 圧力補正                                        *
*                                                                     *
*   ●本プログラムは，HSMAC法のプログラムhsmac2d.forや，SMAC法のプロ *
*     グラムsmac2d.forと可能な限り対応するように作成してあり，これら *
*     プログラムとの違いをわかりやすくするため，SIMPLE法において変更 *
*     あるいは新たに付け加えた個所には,"--- SIMPLE ---"としてコメント*
*     を挿入してある.                                                 *
*                                                                     *
*   ●SIMPLE法における線形システム解法について                        *
*     SIMPLE法の線形システムの係数行列は速度が影響するため，静止状態 *
*     直後の場合など，速度場がゼロであったりすると，METHOD=1,4,5にお *
*     いて解を得られない場合がある.                                   *
*     METHOD=1: 一次独立な解を求めるのが数学的に正しいのでIRELP=1と  *
*               とすること．コンパイラーによっては，IRELP=0,2として  *
*               も丸め誤差などにより，解が得られる場合もある.        *
*     METHOD=4: 初期値ベクトル¥vec{X}_{0}がゼロとなったりすると，探索*
*               ベクトルの計算がゼロ除算となるので適当に初期値を設定 *
*               する必要がある．本プログラムでは，初期値として最新の *
*               値を用いるようにして，可能な限り繰り返し回数が少なく *
*               なることを優先としており，ゼロ除算のときは POINT-SOR *
*               法に切り替えるようにしてあるので，このときは，OMGの  *
*               値を用いることとなる.                                 *
*     METHOD=5: METHOD=4に同じ.                                       *
*                                                                     *
*   ●SIMPLE法における収束解について                                  *
*     もともと線形システム解法における収束判定値は，それぞれの解法の *
*     途中で，解がその判定値”以下”となるかどうかを判定するものであ *
*     る．したがって，得られる解はその収束判定値に依存する場合がある.*
*     SIMPLE法ではとくに，線形システム解法で得られる速度場が係数行列 *
*     にも影響するので，線形システム解法の種類によって解が若干異なる *
*     場合がある.                                                     *
*                                                                     *
*   ●ほか                                                            *
*     この計算プログラム例では，計算終了時に，                        *
*       1. 圧力補正の線形システム解法を行った回数の合計               *
*       2. 連続の式を満たすのにかかった回数の合計                     *
*     を画面表示しますので，各解法による違いを確認できます.          *
*                                                                     *
* SIMPLE法については以下の文献を参照                                  *
* [1] スハス.V.パタンカー 著，水谷幸夫・香月正司 共訳                 *
*      コンピュータによる熱移動と流れの数値解析，森北出版（1985).    *
*                                                                     *
* SIMPLE法を用いたより汎用的なプログラミングについては以下の文献を   *
* 参照                                                                *
* [2] 香月正司・中山　顕 共著                                         *
*      熱流動の数値シミュレーション，森北出版（1990)                  *
*                                                                     *
```

```
**********************************************************************
      PROGRAM SMPL2D
**********************************************************************
*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $         US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $         PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $         T(0:NX0+1,0:NY0+1),
     $         XX(NEP0)
      COMMON / ARRAY2 /
     $         U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
      CHARACTER FNAME(20)*20
**********************************************************************
* コンパイラーによっては，以下の2つのDIMENSION文は，メインプログラム*
* で必ずしも宣言されていなくてもよく，これらの配列を用いるSUBROUTINE*
* においてのみ宣言すればよい．この場合，これに対応して，CALL文や     *
* SUBROUTINE文の引数にあるこれらの配列を削除する．                   *
**********************************************************************
*     作業用配列
      DIMENSION W1(NEP0),W2(NEP0),W3(NEP0),W4(NEP0),W5(NEP0),W6(NEP0),
     $          W7(NEP0),W8(NEP0),W9(NEP0)
*     バンドガウス消去法のための係数行列配列
      DIMENSION AGB(-NY0:NY0,NEP0)
**********************************************************************
*
*x方向の格子分割数
      NX = NX0
*y方向の格子分割数
      NY = NY0
*方程式の数( U* に関する未知数の数) --- SIMPLE ---
      NEU = NEU0
*方程式の数( V* に関する未知数の数) --- SIMPLE ---
      NEV = NEV0
*方程式の数(圧力補正に関する未知数の数) --- SIMPLE ---
      NEP = NEP0
```

```
*圧力計算の線形システム解法の総反復回数のための変数
      ITRALL = 0
      ICNTAL = 0
*パラメータファイルのオープン
      OPEN (10,FILE='IN2D.SIM',STATUS='OLD')
*出力ファイル名の読み込み
      DO 20 I = 1,9
        READ (10,'(A20)') FNAME(I)
   20 CONTINUE
*Uの計算結果出力用ファイルオープン(書式なし形式)
*書式なし形式はコンパイラーに依存するので注意
      OPEN (11, FILE=FNAME(1), STATUS='NEW', FORM='UNFORMATTED')
*Vの計算結果出力用ファイルオープン(書式なし形式)
      OPEN (12, FILE=FNAME(2), STATUS='NEW', FORM='UNFORMATTED')
*Pの計算結果出力用ファイルオープン(書式なし形式)
      OPEN (13, FILE=FNAME(3), STATUS='NEW', FORM='UNFORMATTED')
*Tの計算結果出力用ファイルオープン(書式なし形式)
      OPEN (14, FILE=FNAME(4), STATUS='NEW', FORM='UNFORMATTED')
*パラメータファイルからパラメータを読み込む
  700 CONTINUE
      READ (10,*,ERR=700) ITYPE,ICYCLE,NITR,NCYCLE,NCNT
  710 CONTINUE
      READ (10,*,ERR=710) EPSP,EPSC,OMG,ALPHAP,ALPHAU,ALPHAV,ALPHAT
  720 CONTINUE
      READ (10,*,ERR=720) DT,RE,PR,GR
  730 CONTINUE
      READ (10,*,ERR=730) DLX,DLY,IRELP,METHOD,NMT
*継続の計算の場合
      IF (ICYCLE.NE.0) THEN
*Uデータファイルのオープン(書式なし形式)
        OPEN (15, FILE=FNAME(5), STATUS='OLD', FORM='UNFORMATTED')
*Vデータファイルのオープン(書式なし形式)
        OPEN (16, FILE=FNAME(6), STATUS='OLD', FORM='UNFORMATTED')
*Pデータファイルのオープン(書式なし形式)
        OPEN (17, FILE=FNAME(7), STATUS='OLD', FORM='UNFORMATTED')
*Tデータファイルのオープン(等温場でもT=0.0のデータを読み込む)(書式なし形式)
        OPEN (18, FILE=FNAME(8), STATUS='OLD', FORM='UNFORMATTED')
      END IF
*x方向の格子幅
      DX = DLX / FLOAT(NX)
*y方向の格子幅
      DY = DLY / FLOAT(NY)
*運動方程式中の拡散項の係数(ここではPr)..........基礎式に応じて変更
      VIS = PR
*浮力項の係数(ここでは Gr * Pr**2)..............基礎式に応じて変更
      BUO = GR * ( PR**2 )
*エネルギー方程式中の拡散項の係数(ここでは1).....基礎式に応じて変更
      ALP = 1.0D0
*等温場なら浮力項の係数はゼロに設定
      IF (ITYPE.EQ.1) BUO = 0.0D0
*文献[1]にしたがった離散化を考慮してR0を定義
```

```
*本書では無次元化をおこなってから計算しており，RO=1とする
      RO = 1.0D0
*初期値の設定
      CALL CINITI (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
*時間進行のための戻り点
  750 CONTINUE
*時間進行
      CALL ADV
      write(6,*) icycle
*連続の式の収束判定の反復回数を表す変数
      ITRCNT = 0
*速度・圧力・温度補正のための戻り点
  760 CONTINUE
      ITRCNT = ITRCNT + 1
*速度，圧力，温度の線形システム解法が収束したかどうかのパラメータ(IFLG)を初期化
*これは，クリロフ部分空間法を含む反復法で用いる
*IFLG -> 0:収束 1:発散(設定された許容回数NITR以下で解が得られない)
      IFLG  = 0
*速度，圧力，温度を同時に補正しながら連続の式を満たすまで計算するが，
*この際，連続の式を満たすかどうかを判定する反復回数を意味する
*パラメータ(IFLGC)を初期化
*IFLGC-> 0:連続の式を満たす 1:発散(許容回数NCNT以下で解が得られない)
      IFLGC = 0
*速度場の計算
C     --- 注意 ---
C     本計算においては，ICYCLE=0のときの初期条件は速度場と圧力場をゼロと
C     しているので，これらに関わる線形システムを解くのが困難となるため，
C     ICYCLE=1のときだけは温度の計算にジャンプするようにする．
C     計算条件や問題に応じて適宜削除あるいは変更
      IF (ICYCLE.EQ.1.AND.ITRCNT.EQ.1) GOTO 770
      CALL CALVEL (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
*     速度計算の線形システム解法が収束していないとき
      IF ( IFLG.EQ.1 ) THEN
        WRITE (6,*) ' NOT CONVERGE ! '
C       データを出力して強制終了
        CALL PROUT
        GO TO 900
      END IF
*圧力補正の計算と，圧力場と速度場の更新
      CALL PRESS (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
*     圧力計算の線形システム解法の総反復回数
      ITRALL = ITRALL + ITR
C     速度場と圧力場がゼロとなるときのスキップ先
  770 CONTINUE
*     圧力計算の線形システム解法が収束していないとき
      IF ( IFLG.EQ.1 ) THEN
        WRITE (6,*) ' NOT CONVERGE ! '
C       データを出力して強制終了
        CALL PROUT
        GO TO 900
      END IF
```

```
*     非等温場計算の場合
      IF ( ITYPE.EQ.2 ) THEN
*       温度場を計算
        CALL CALTEM (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
*       温度計算の線形システム解法が収束していないとき
        IF ( IFLG.EQ.1 ) THEN
          WRITE (6,*) ' NOT CONVERGE ! '
C         データを出力して強制終了
          CALL PROUT
          GO TO 900
        END IF
      END IF
*     速度，圧力，温度場の計算が収束して連続の式をEPSP以下で満足していないとき(IFLGC=1)
      IF ( IFLGC.EQ.1 ) THEN
*       予め設定された許容回数NCNTに達していないときは再計算
        IF (ITRCNT.LT.NCNT) THEN
          GO TO 760
*       予め設定された許容回数NCNTに達したらデータを出力して強制終了
        ELSE
          WRITE (6,*) ' NOT CONVERGE ! '
C         データを出力して強制終了
          CALL PROUT
          GO TO 900
        END IF
      END IF
*     速度，圧力，温度場の計算が収束して連続の式をEPSP以下で満足し(IFLGC=0)
*     時間進行カウンタ(ICYCLE)がNCYCLEより小さい時
      ICNTAL = ICNTAL + ITRCNT
      IF ( ICYCLE. LT. NCYCLE ) THEN
        GO TO 750
*     時間進行カウンタがNCYCLEになったら計算終了
      ELSE
        CALL PROUT
      END IF
  900 CONTINUE
*Tecplot用データの出力
      CALL TECPLT(FNAME(9))
      CLOSE (10)
      CLOSE (11)
      CLOSE (12)
      CLOSE (13)
      CLOSE (14)
*
*     連続の式を満たすための反復回数の合計(ICNTAL)と,
*     圧力補正の線形システム解法のための反復回数の合計(ITRALL)を表示
      WRITE (6,*) 'Total Iteration Number'
      WRITE (6,*) 'To solve Continuity Eq.      ',ICNTAL
      WRITE (6,*) 'To solve Linear System of PD',ITRALL
*
      STOP
      END
```

```
************************************************************************
*                          初期設定
************************************************************************
      SUBROUTINE CINITI (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $         US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $         PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $         T(0:NX0+1,0:NY0+1),
     $         XX(NEP0)
      COMMON / ARRAY2 /
     $         U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*     作業用配列
      DIMENSION W1(NEP0),W2(NEP0),W3(NEP0),W4(NEP0),W5(NEP0),W6(NEP0),
     $          W7(NEP0),W8(NEP0),W9(NEP0)
*     バンドガウス消去法のための係数行列配列
      DIMENSION AGB(-NY0:NY0,NEP0)
*
*新規計算の場合
      IF ( ICYCLE. EQ. 0 ) THEN
*       Uの初期値設定.................計算開始時は静止(あるいはごく小さな値)
        DO 10 IX = 0,NX
          DO 20 IY = 0,NY+1
*           METHOD=1,4,5のとき適宜修正
            U(IX,IY) = 0.0D0
   20     CONTINUE
   10   CONTINUE
*       Uの初期値設定.................計算開始時は静止(あるいはごく小さな値)
        DO 30 IY = 0,NY
          DO 40 IX = 0,NX+1
*           METHOD=1,4,5のとき適宜修正
            V(IX,IY) = 0.0D0
   40     CONTINUE
   30   CONTINUE
*       P* と P’の初期値設定.........計算開始時は静止しているので変動圧力はゼロ
        DO 50 IY = 0,NY+1               (12.3節を参照)
```

```
          DO 60 IX = 0,NX+1          変動圧力(静止状態からのずれ:以降圧力とよぶ)
C           --- SIMPLE ---           をゼロに初期化
            PS(IX,IY) = 0.0D0
            PD(IX,IY) = 0.0D0
   60     CONTINUE
   50   CONTINUE
*-----------------------------------------------------------------------*
*（注意）浮力項の計算で温度の配列を使用しているので等温場でもT=0として *
* 初期条件だけは設定する必要がある. ゼロ以外の値を入れると浮力項が計算 *
* される可能性があるので注意.                                          *
*-----------------------------------------------------------------------*
*       Tの初期値設定(領域内は高温(+0.5)と低温(-0.5)の中間温度)
        DO 70 IY = 0,NY+1
          DO 80 IX = 0,NX+1
            T(IX,IY) = 0.0D0
   80     CONTINUE
   70   CONTINUE
........温度は境界で定義できないので仮想セルを用いる
        ここでは1次精度(前進あるいは後退)差分を使用(式(13.11),(13.12)を参照)
*       Tの境界：右面（冷却）T=-0.5
        DO 90 IY = 0,NY+1.................仮想セルの温度を設定(式(14.56)を参照)
          T(NX+1,IY) = 2.0D0 * ( -0.5D0 ) - T(NX,IY)
   90   CONTINUE
*       Tの境界：左面（加熱）T=+0.5
        DO 100 IY = 0,NY+1................仮想セルの温度を設定(式(14.56)を参照)
          T(0,IY) = 2.0D0 * ( +0.5D0 ) - T(1,IY)
  100   CONTINUE
*       Tの境界：上面（断熱）
        DO 110 IX = 1,NX..................仮想セルの温度を用いて勾配ゼロを設定
          T(IX,NY+1) = T(IX,NY)
  110   CONTINUE
*       Tの境界：下面（断熱）
        DO 120 IX = 1,NX..................仮想セルの温度を用いて勾配ゼロを設定
          T(IX,0) = T(IX,1)
  120   CONTINUE
*継続計算（すでにある計算結果からスタート）の場合
      ELSE
*       Uデータファイルからの読み込み[Unit No.=15](書式なし形式)
        READ (15) U
*       Vデータファイルからの読み込み[Unit No.=16](書式なし形式)
        READ (16) V
*       Pデータファイルからの読み込み[Unit No.=17](書式なし形式)
C       --- SIMPLE ---
        READ (17) PS,PD
*----------------------------------------------------------------------*
*    (注意) 等温場の計算でもT(=0)のファイルを読み込む必要がある        *
*----------------------------------------------------------------------*
*       Tデータファイルからの読み込み[Unit No.=18](書式なし形式)
        READ (18) T
        CLOSE (15)
        CLOSE (16)
```

```
         CLOSE (17)
         CLOSE (18)
      END IF
*線形システム解法に関わる配列も初回のみ初期化
*U*, V*, P', Tに関わる線形システム解法の解が配列XXに入る
      DO 130 I=1,NEP
         ATP1(I)=0.0D0
         ATP2(I)=0.0D0
         ATP3(I)=0.0D0
         ATP4(I)=0.0D0
         ATP5(I)=0.0D0
         BP(I)=0.0D0
* METHOD=1,4,5のとき適宜修正(--- SIMPLE ---)
* XX をここでのみ初期化するか線形システム解法の度に直前で初期化するなどする
         XX(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
         W9(I)=0.0D0
         DO 140 II = -NY,NY
           AGB(II,I)=0.0D0
  140    CONTINUE
  130 CONTINUE
      RETURN
      END
**********************************************************************
*                            時間進行
**********************************************************************
      SUBROUTINE ADV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
```

```
     $          US(0:NX0,   0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $          PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $          T(0:NX0+1,0:NY0+1),
     $          XX(NEP0)
      COMMON / ARRAY2 /
     $          U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*
      TIME   = DT*FLOAT(ICYCLE)
      ICYCLE = ICYCLE + 1
*時間進行カウンタ(ICYCLE)を100回毎に表示
      IF (MOD(ICYCLE,100).EQ.0) THEN
        WRITE (6,2000) ICYCLE
 2000   FORMAT ('  CYC = ',2I8)
      END IF
*----------------------------------------------------------------------
* U  -> U0 : 必要なら入れ替える前にUとU0から変動量を求める
* U  : 前の時間ステップにおいて最終的に得られた値
*       速度・圧力補正の繰り返しで U が更新されるので,
* U0 : 前の時間ステップの値を保存しておく
*----------------------------------------------------------------------
      DO 10 IX = 0,NX
        DO 20 IY = 0,NY+1
          U0(IX,IY) = U(IX,IY)
   20   CONTINUE
   10 CONTINUE
*----------------------------------------------------------------------
* V  -> V0 : 必要なら入れ替える前にVとV0から変動量を求める
* V  : 前の時間ステップにおいて最終的に得られた値
*       速度・圧力補正の繰り返しで V が更新されるので,
* V0 : 前の時間ステップの値を保存しておく
*----------------------------------------------------------------------
      DO 30 IX = 0,NX+1
        DO 40 IY = 0,NY
          V0(IX,IY) = V(IX,IY)
   40   CONTINUE
   30 CONTINUE
*----------------------------------------------------------------------
* T  -> T0 : 必要なら入れ替える前にTとT0から変動量を求める
* T  : 前の時間ステップにおいて最終的に得られた値
*       速度・圧力補正の繰り返しで T が更新されるので,
* T0 : 前の時間ステップの値を保存しておく
*----------------------------------------------------------------------
      DO 110 IX = 0,NX+1
        DO 120 IY = 0,NY+1
          T0(IX,IY) = T(IX,IY)
  120   CONTINUE
  110 CONTINUE
      RETURN
      END
**********************************************************************
*                  仮の速度場 (U*, V*) の計算
```

```
**********************************************************************
*                                                                    *
* [仮の速度場(U*, V*)の線形システムに関する配列の説明]               *
*                                                                    *
* (U*, V*)の線形システムを ATP_{i,j} US(VS)_{i} = BP_{i} とする.     *
* 有限差分近似を用いて離散化していることから明らかなように,          *
* 1. ATP_{i,j}の大部分はゼロ                                         *
* 2. 非ゼロ要素は疎であり, 規則的に並んでいる                        *
* したがって,ATP_{i,j}すべてを記憶させるのは計算容量のムダであるので*
* 本プログラムでは, 以下のような規則にしたがって記憶する.            *
* 2次元問題であれば, 近隣の5点が関わる.                              *
*                                                                    *
* ●用いる配列(1次元配列に格納):クリロフ部分空間法を含む反復法      *
*   係数行列     ---> ATP1, ATP2, ATP3, ATP4, ATP5                   *
*   既知ベクトル ---> BP                                             *
*   未知ベクトル ---> XX                                             *
*   未知数の数(未知のUS,VSの数)                                     *
*   ---> US: NEU = (NX-1) * NY,   VS: NEV = NX * (NY-1)              *
*   注意:未知数はNEU, NEVであるが、係数行列の配列のサイズは、       *
*         圧力と温度の配列と共用するためにNEPを用いる                *
*                                    ~~~~~~~~~~~~~~~~~~~~~~~~~~      *
*                                                                    *
*    ATP1(NEP),ATP2(NEP),ATP3(NEP),ATP4(NEP),ATP5(NEP)               *
*              i=3, j=3 (k=13) : US(3,3) における ATP1,ATP2,ATP3     *
*                 +-------------------------+       ATP4,ATP5        *
*         (NY)5   |    |    |    |    |    |                         *
*                 +-------------------------+     通し番号の規則     *
*             4   |    |    |ATP4|    |    |      k=(i-1)*NY+j       *
*                 +-------------------------+                        *
*             3   |    |ATP1|ATP3|ATP5|    |      ATP1(k)            *
*                 +-------------------------+     ATP2(k)            *
*             2   |    |    |ATP2|    |    |      ATP3(k)            *
*                 +-------------------------+     ATP4(k)            *
*             1   |    |    |    |    |    |      ATP5(k)            *
*                 +-------------------------+     BP(k)              *
*             j    i=1   2    3    4    5(NX)                        *
*                                                                    *
*    BP(NEP) : 既知ベクトル                                          *
*                                                                    *
*    XX(NEP) : 未知ベクトル(各サブルーチンでこれが求まる)           *
*                                                                    *
* ●用いる配列:ガウスの消去法による直接法(バンド行列として格納)     *
*    AGB(-NY:NY,NEP) : 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 ATP1 から ATP5 までの k は*
*    USを求める場合であれば                                          *
*                       ATP1のkは K"-NY"                             *
*                       ATP2のkは K"-1 "                             *
*                       ATP3のkは K"+0 "                             *
*                       ATP4のkは K"+1 "                             *
*                       ATP5のkは K"+NY"                             *
*                  このように, kを固定したとき, " "で囲まれた5個の値 *
*                  (-NY,-1,0,1,NY)                                   *
```

```
*     NEP -> 上述のような k は全部で NEP 個定義される                *
*            (実際は NEU あるいは NEV 個)                           *
*                                                                    *
* ●線形システムを構成する際の注意                                   *
* 線形システムを構成する際，境界条件をどこで反映させるかによって，  *
* プログラミングが異なる．本プログラムでは，境界条件は，係数行列を  *
* 作成する際に反映させ，線形システムの解法においては専ら AX=B のみ  *
* に着目する立場をとる．                                            *
* 上述のような立場とは異なり，線形システムの解法において境界条件を  *
* 反映させることもできるが，ここでは，線形システムの解法のサブルー  *
* チンに汎用性をもたせることを優先させた．                          *
*                                                                    *
**********************************************************************
      SUBROUTINE CALVEL (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $         US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $         PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $         T(0:NX0+1,0:NY0+1),
     $         XX(NEP0)
      COMMON / ARRAY2 /
     $         U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*     作業用配列
      DIMENSION W1(NEP0),W2(NEP0),W3(NEP0),W4(NEP0),W5(NEP0),W6(NEP0),
     $          W7(NEP0),W8(NEP0),W9(NEP0)
*     バンドガウス消去法のための係数行列配列
      DIMENSION AGB(-NY0:NY0,NEP0)
*
*----------------------------------------------------------------*
* 時間微分:オイラー陽解法(1次精度前進差分)...13.2.1項参照       *
* 拡散 + 対流項: コントロール・ボリューム法..14.6節参照         *
*----------------------------------------------------------------*
*--------------------------------------------*
*             U* : US(IX,IY) の計算          *
*        位置を表す"1,2,3,4,5"の説明         *
*      1:w   2:es   3:e   4:en   5:ee        *
```

```
*------------------------------------------------*
* I:線形システム行列用の通し番号を意味する変数
* 2次元配列(IX,IY):離散化用 -> 1次元配列(I):線形システム用
      I = 1
      DO 10 IX = 1,NX-1
        DO 20 IY = 1,NY
*         Uに関するDの計算............................式(14.103)
          DU5 = VIS * DY / DX.........................D_E
          DU1 = VIS * DY / DX.........................D_P
          DU4 = VIS * DX / DY.........................D_en
          DU2 = VIS * DX / DY.........................D_es
*         Uに関するFの計算
*         速度・圧力・温度場の繰り返し計算となっても線形システムの係数となる
*         ここの速度は更新されないように(U,V)には前の時刻の収束値(U0,V0)を用いる
          UU5 = ( U0(IX  ,IY  ) + U0(IX+1,IY  ) ) / 2.0D0...u_E
          UU1 = ( U0(IX-1,IY  ) + U0(IX  ,IY  ) ) / 2.0D0...u_P
          VU4 = ( V0(IX  ,IY  ) + V0(IX+1,IY  ) ) / 2.0D0...v_en
          VU2 = ( V0(IX  ,IY-1) + V0(IX+1,IY-1) ) / 2.0D0...v_es
          FU5 = RO * UU5 * DY...F_E
          FU1 = RO * UU1 * DY...F_P          +--------+--------+--------+
          FU4 = RO * VU4 * DX...F_en         |        |P(i,j+1)|        |
          FU2 = RO * VU2 * DX...F_es         |        |   ・   |        |
*         Uに関するPeの計算                  |        | V(i,j) |        |
          PEU5 = FU5 / DU5...Pe_E            +--------+---↑---+--------+
          PEU1 = FU1 / DU1...Pe_P            |P(i-1,j)| P(i,j) |P(i+1,j)|
          PEU4 = FU4 / DU4...Pe_en           |   ・   →   ・   →   ・   |
          PEU2 = FU2 / DU2...Pe_es           |     U(i-1,j)  U(i,j)     |
*         関数A(|Pe|)の計算...式(14.105)     +--------+---↑---+--------+
*         NMT=1なら 風上法                   |        |V(i,j-1)|        |
          IF ( NMT.EQ.1 ) THEN               |        |   ・   |        |
            APEU5 = 1.0D0...A(Pe_E)          |        |P(i,j-1)|        |
            APEU1 = 1.0D0...A(Pe_P)          +--------+--------+--------+
            APEU4 = 1.0D0...A(Pe_en)
            APEU2 = 1.0D0...A(Pe_es)
*         NMT=2なら べき乗法
          ELSE
            APEU5 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEU5))**5 ) )...A(Pe_E)
            APEU1 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEU1))**5 ) )...A(Pe_P)
            APEU4 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEU4))**5 ) )...A(Pe_en)
            APEU2 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEU2))**5 ) )...A(Pe_es)
          END IF
*         U* の線形システムの係数の計算.............式(14.101)
          AU5 = ( DU5*APEU5 + DMAX1(-FU5,0.0D0) )...a_E
          AU1 = ( DU1*APEU1 + DMAX1( FU1,0.0D0) )...a_P
          AU4 = ( DU4*APEU4 + DMAX1(-FU4,0.0D0) )...a_en
          AU2 = ( DU2*APEU2 + DMAX1( FU2,0.0D0) )...a_es
*         生成項..................................式(14.108)
          SCU = 0.0D0
          SPU = 0.0D0
*         U* の線形システムのBの計算................式(14.109)
          BP(I) = SCU*DX*DY+( PS(IX,IY)-PS(IX+1,IY) )*DY...圧力項もBに入れてある
```

```
     $          + RO*DX*DY/DT*UO(IX,IY)
*境界の処理
* 1. 境界のU_BND についても変数とみなしてAU3(IX,IY)を求める
* 2. その後, 境界の項 AU(1,2,4,5)*U_BND をBに吸収させる
* 3. 2に対応して(吸収の後), 境界のAU(1,2,4,5)をゼロとしてリンクを断つ
          AU3(IX,IY) = AU1+AU2+AU4+AU5 + RO*DX*DY/DT - SPU*DX*DY
*U（右面）
          IF ( IX.EQ.NX-1 ) THEN
            BP(I) = BP(I)+AU5*U(NX,IY)
            AU5 = 0.0D0
          END IF
*U（左面）
          IF ( IX.EQ.1 ) THEN
            BP(I) = BP(I)+AU1*U(0,IY)
            AU1 = 0.0D0
          END IF
*U（上面）
          IF ( IY.EQ.NY ) THEN
            BP(I) = BP(I) + AU4*U(IX,NY+1)
            AU4 = 0.0D0
          END IF
*U（下面）
          IF ( IY.EQ.1 ) THEN
            BP(I) = BP(I) + AU2*U(IX,0)
            AU2 = 0.0D0
          END IF
* 線形システム計算用に係数行列を計算
* ALPHAU : U* 計算のための緩和係数
          ATP3(I) = AU3(IX,IY) / ALPHAU
          ATP5(I) = - AU5
          ATP1(I) = - AU1
          ATP4(I) = - AU4
          ATP2(I) = - AU2
          BP(I) = BP(I) + (1.0D0-ALPHAU)*ATP3(I)*UO(IX,IY)
* 線形システム解法前のための初期化(任意) (--- SIMPLE ---)
* METHOD=1,4,5のとき適宜修正
          XX(I) = U(IX,IY)
          I = I + 1
   20   CONTINUE
   10 CONTINUE
*
*サブルーチンに一般性を持たせるため, NX,NY,NEを引数として渡す
*COMMON文で定義されている値でもあるので, そのまま渡せない！
      NNX = NX-1
      NNY = NY
      NNE = NEUO
*以下の線形システム解法の後, 配列XXにUS(U*)が入る
C   1. 直接法 : バンドマトリックスによるガウスの消去法
      IF (METHOD.EQ.1)
     $  CALL GB      (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                AGB)
```

```
C   2. 反復法1 : point-SOR 法
       IF (METHOD.EQ.2)
      $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
C   3. 反復法2 : line-SOR 法 : OMG=1で十分. 大きくしすぎると発散する
       IF (METHOD.EQ.3)
      $  CALL LSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
      $                W1,W2,W3,W4,W5,W6,W7,W8,W9)
C   4. クリロフ部分空間法1 : 共役残差法
       IF (METHOD.EQ.4)
      $  CALL CRB     (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
      $                W1,W2,W3,W4,W5,W6)
C   5. クリロフ部分空間法2 : BiCGSTAB
       IF (METHOD.EQ.5)
      $  CALL BICGB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
      $                W1,W2,W3,W4,W5,W6,W7)
C   METHOD=4,5のときの探索ベクトル計算時のゼロ除算対策
       IF (IFLG.EQ.2)
      $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
*線形システム解法で得られた1次元配列の解を2次元配列に置き換える
      DO 30 IX = 1,NX-1
        DO 40 IY = 1,NY
          K = IY + (IX-1)*NY
          US(IX,IY) = XX(K)
   40   CONTINUE
   30 CONTINUE
*U*の境界条件の処理
      CALL UBND (US)
*
*-----------------------------------------------*
*           V* : VS(IX,IY) の計算               *
*         位置を表す"1,2,3,4,5"の説明           *
*        1:wn   2:s   3:n   4:nn   5:en         *
*-----------------------------------------------*
* I:線形システム行列用の通し番号を意味する変数
* 2次元配列(IX,IY):離散化用 -> 1次元配列(I):線形システム用
      I = 1
      DO 50 IX = 1,NX
        DO 60 IY = 1,NY-1
*         Vに関するDの計算.............................式(14.104)
          DV5 = VIS * DY / DX............................D_en
          DV1 = VIS * DY / DX............................D_wn
          DV4 = VIS * DX / DY............................D_N
          DV2 = VIS * DX / DY............................D_P
*         Vに関するFの計算
*         速度・圧力・温度場の繰り返し計算となっても線形システムの係数となる
*         ここの速度は更新されないように(U,V)には前の時刻の収束値(U0,V0)を用いる
          UV5 = ( U0(IX  ,IY  ) + U0(IX  ,IY+1) ) / 2.0D0...u_en
          UV1 = ( U0(IX-1,IY  ) + U0(IX-1,IY+1) ) / 2.0D0...u_wn
          VV4 = ( V0(IX  ,IY  ) + V0(IX  ,IY+1) ) / 2.0D0...v_N
          VV2 = ( V0(IX  ,IY-1) + V0(IX  ,IY  ) ) / 2.0D0...v_P
          FV5 = R0 * UV5 * DY...F_en
```

```
          FV1 = RO * UV1 * DY...F_wn          +--------+--------+--------+
          FV4 = RO * VV4 * DX...F_N           |        |P(i,j+1)|        |
          FV2 = RO * VV2 * DX...F_P           |        |   ・   |        |
*         Vに関するPeの計算                   |        | V(i,j) |        |
          PEV5 = FV5 / DV5...Pe_en            +--------+---↑---+--------+
          PEV1 = FV1 / DV1...Pe_wn            |P(i-1,j)| P(i,j) |P(i+1,j)|
          PEV4 = FV4 / DV4...Pe_N             |   ・   →   ・   →   ・   |
          PEV2 = FV2 / DV2...Pe_P             |     U(i-1,j)  U(i,j)     |
*         関数A(|Pe|)の計算                   +--------+---↑---+--------+
*         NMT=1なら 風上法                    |        |V(i,j-1)|        |
          IF ( NMT.EQ.1 ) THEN                |        |   ・   |        |
            APEV5 = 1.0D0...A(Pe_en)          |        |P(i,j-1)|        |
            APEV1 = 1.0D0...A(Pe_wn)          +--------+--------+--------+
            APEV4 = 1.0D0...A(Pe_N)
            APEV2 = 1.0D0...A(Pe_P)
*         NMT=2なら べき乗法
          ELSE
            APEV5 = DMAX1(0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEV5))**5 ) )...A(Pe_en)
            APEV1 = DMAX1(0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEV1))**5 ) )...A(Pe_wn)
            APEV4 = DMAX1(0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEV4))**5 ) )...A(Pe_N)
            APEV2 = DMAX1(0.0D0,DBLE( (1.0D0-0.1D0*ABS(PEV2))**5 ) )...A(Pe_P)
          END IF
*         V* の線形システムの係数の計算.............式(14.102)
          AV5 = ( DV5*APEV5 + DMAX1(-FV5,0.0D0) )...a_en
          AV1 = ( DV1*APEV1 + DMAX1( FV1,0.0D0) )...a_wn
          AV4 = ( DV4*APEV4 + DMAX1(-FV4,0.0D0) )...a_N
          AV2 = ( DV2*APEV2 + DMAX1( FV2,0.0D0) )...a_P
*         速度・圧力・温度場の繰り返し計算となった場合，更新されたT
*         により浮力も更新される
          TV = ( T(IX,IY) + T(IX,IY+1) ) / 2.0D0
*         生成項(浮力項)の計算.....................式(14.110)
          SCV = BUO * TV
          SPV = 0.0D0
*         V* の線形システムのBの計算................式(14.110)
          BP(I) = SCV*DX*DY+( PS(IX,IY)-PS(IX,IY+1) )*DX...圧力項もBに入れてある
     $          + RO*DX*DY/DT*VO(IX,IY)
*境界の処理
* 1. 境界のV_BND についても変数とみなしてAV3(IX,IY)を求める
* 2. その後，境界の項 AV(1,2,4,5)*V_BND をBに吸収させる
* 3. 2に対応して(吸収の後)，境界のAV(1,2,4,5)をゼロとしてリンクを断つ
          AV3(IX,IY) = AV1+AV2+AV4+AV5 + RO*DX*DY/DT - SPV*DX*DY
*V（右面）
          IF ( IX.EQ.NX ) THEN
            BP(I) = BP(I) + AV5 * V(NX+1,IY)
            AV5 = 0.0D0
          END IF
*V（左面）
          IF ( IX.EQ.1 ) THEN
            BP(I) = BP(I) + AV1 * V(0,IY)
            AV1 = 0.0D0
          END IF
```

```
*V（上面）
          IF ( IY.EQ.NY-1 ) THEN
            BP(I) = BP(I) + AV4*V(IX,NY)
            AV4 = 0.0D0
          END IF
*V（下面）
          IF ( IY.EQ.1 ) THEN
            BP(I) = BP(I) + AV2*V(IX,0)
            AV2 = 0.0D0
          END IF
* 線形システム計算用に係数行列を計算
* ALPHAV : V* 計算のための緩和係数
          ATP3(I) = AV3(IX,IY) / ALPHAV
          ATP5(I) = - AV5
          ATP1(I) = - AV1
          ATP4(I) = - AV4
          ATP2(I) = - AV2
          BP(I) = BP(I) + (1.0D0-ALPHAV)*ATP3(I)*V0(IX,IY)
* 線形システム解法前のための初期化(任意) (--- SIMPLE ---)
* METHOD=1,4,5のとき適宜修正
          XX(I) = V(IX,IY)
          I = I + 1
   60   CONTINUE
   50 CONTINUE
*
*サブルーチンに一般性を持たせるため，NX,NY,NEを引数として渡す
*COMMON文で定義されている値でもあるので，そのまま渡せない！
      NNX = NX
      NNY = NY-1
      NNE = NEV
*以下の線形システム解法の後，配列XXにVS(V*)が入る
C   1. 直接法 : バンドマトリックスによるガウスの消去法
      IF (METHOD.EQ.1)
     $  CALL GB      (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                AGB)
C   2. 反復法1 : point-SOR 法
      IF (METHOD.EQ.2)
     $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
C   3. 反復法2 : line-SOR 法 : OMG=1で十分. 大きくしすぎると発散する
      IF (METHOD.EQ.3)
     $  CALL LSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6,W7,W8,W9)
C   4. クリロフ部分空間法1 : 共役残差法
      IF (METHOD.EQ.4)
     $  CALL CRB     (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6)
C   5. クリロフ部分空間法2 : BiCGSTAB
      IF (METHOD.EQ.5)
     $  CALL BICGB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6,W7)
C   METHOD=4,5のときの探索ベクトル計算時のゼロ除算対策
```

```
      IF (IFLG.EQ.2)
     $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
*線形システム解法で得られた1次元配列の解を2次元配列に置き換える
      DO 70 IX = 1,NX
        DO 80 IY = 1,NY-1
          K = IY + (IX-1)*(NY-1)
          VS(IX,IY) = XX(K)
   80   CONTINUE
   70 CONTINUE
*V*の境界条件の処理
      CALL VBND (VS)
*
      RETURN
      END
*********************************************************************
*                           圧力場の計算
*********************************************************************
*                                                                   *
* [圧力補正の線形システムに関する配列の説明]                        *
*                                                                   *
* 圧力補正(PD)の線形システムを ATP_{i,j} PD_{i} = BP_{i} とする.    *
* 有限差分近似を用いて離散化していることから明らかなように,         *
* 1. ATP_{i,j}の大部分はゼロ                                        *
* 2. 非ゼロ要素は疎であり, 規則的に並んでいる                       *
* したがって,ATP_{i,j}すべてを記憶させるのは計算容量のムダであるので*
* 本プログラムでは, 以下のような規則にしたがって記憶する.           *
* 2次元問題であれば, 近隣の5点が関わる.                             *
*                                                                   *
* ●用いる配列(1次元配列に格納) : クリロフ部分空間法を含む反復法    *
*   係数行列     ---> ATP1, ATP2, ATP3, ATP4, ATP5                  *
*   既知ベクトル ---> BP                                            *
*   未知ベクトル ---> XX                                            *
*   未知数の数(未知のPDの数)   ---> NEP = NX * NY                   *
*                                                                   *
*    ATP1(NE),ATP2(NE),ATP3(NE),ATP4(NE),ATP5(NE)                   *
*            i=3, j=3 (k=13) : PD(3,3) における ATP1,ATP2,ATP3      *
*               +-------------------------+        ATP4,ATP5        *
*         (NY)5 |    |    |    |    |    |                          *
*               +-------------------------+      通し番号の規則     *
*             4 |    |    |ATP4|    |    |      k=(i-1)*NY+j        *
*               +-------------------------+                         *
*             3 |    |ATP1|ATP3|ATP5|    |      ATP1(k)             *
*               +-------------------------+      ATP2(k)            *
*             2 |    |    |ATP2|    |    |      ATP3(k)             *
*               +-------------------------+      ATP4(k)            *
*             1 |    |    |    |    |    |      ATP5(k)             *
*               +-------------------------+      BP(k)              *
*             j   i=1   2    3    4    5(NX)                        *
*                                                                   *
*    BP(NE) : 既知ベクトル                                          *
*                                                                   *
```

```
*    XX(NE) : 未知ベクトル(各サブルーチンでこれが求まる)              *
*                                                                     *
* ●用いる配列：ガウスの消去法による直接法(バンド行列として格納)      *
*    AGB(-NY:NY,NEP) : 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 ATP1 から ATP5 までの k は*
*                    ATP1のkは K"-NY"                                 *
*                    ATP2のkは K"-1 "                                 *
*                    ATP3のkは K"+0 "                                 *
*                    ATP4のkは K"+1 "                                 *
*                    ATP5のkは K"+NY"                                 *
*               このように，kを固定したとき，" "で囲まれた5個の値    *
*               (-NY,-1,0,1,NY)                                       *
*     NEP -> 上述のような k は全部で NEP 個定義される                 *
*                                                                     *
* ●線形システムを構成する際の注意                                    *
* 線形システムを構成する際，境界条件をどこで反映させるかによって，   *
* プログラミングが異なる．本プログラムでは，境界条件は，係数行列を   *
* 作成する際に反映させ，線形システムの解法においては専ら AX=B のみ   *
* に着目する立場をとる．                                              *
* 上述のような立場とは異なり，線形システムの解法において境界条件を   *
* 反映させることもできるが，ここでは，線形システムの解法のサブルー   *
* チンに汎用性をもたせることを優先させた．                            *
*                                                                     *
***********************************************************************
      SUBROUTINE PRESS (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $         US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $         PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $         T(0:NX0+1,0:NY0+1),
     $         XX(NEP0)
      COMMON / ARRAY2 /
     $         U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*     作業用配列
      DIMENSION W1(NEP0),W2(NEP0),W3(NEP0),W4(NEP0),W5(NEP0),W6(NEP0),
     $          W7(NEP0),W8(NEP0),W9(NEP0)
```

```
*     バンドガウス消去法のための係数行列配列
      DIMENSION AGB(-NY0:NY0,NEP0)
*
*     速度・圧力・温度を連立させて繰り返し計算を行って，連続の式を満たす
*     ( U*, V* ) となったかどうかを判定する際の最大値の初期設定
      DMAX = 0.0D0
*     線形システム解法のため，2次元配列を1次元配列にするための変数
      I = 1
*-------------------------------------------------*
*         離散化における位置を表す番号              *
*          1:W   2:S   3:P   4:N   5:E            *
*-------------------------------------------------*
      DO 10 IX = 1,NX
        DO 20 IY = 1,NY
*速度場(U*, V*)が連続の式を満たすかどうかを計算し，満たしていれば，
*この(U*,V*)をもとに，ここで計算するP'(PD)から得られる(U,V,P)を
*この時間ステップの収束解とする．なお，(U,V)は常に連続の式を満たす
*ことが保証される．したがって，(U*, V*)が連続の式を満たすということは，
*前回の反復において得られた(U,V)を用いると，もはやこれを補正する必要が
*なくなるということを意味する．
*         P'(PD) の線形システムのBの計算..............式(14.123)
          BP(I)=RO*( ( US(IX-1,IY  )-US(IX,IY) )*DY
     $              + ( VS(IX  ,IY-1)-VS(IX,IY) )*DX )
*         (U*, V*)による連続の式の収束性をチェックし，div ¥VEC{v}の
*         最大値がDMAXとなるようにする
          IF ( DABS(BP(I)). GT. DMAX ) THEN
            DMAX = DABS(BP(I))
          END IF
*P'（右面）
          IF ( IX.EQ.NX ) THEN          +--------+--------+--------+
            AP5 = 0.0D0                 |        |P(i,j+1)|        |
          ELSE                          |        |   ・   |        |
            APU5 = AU3(IX,IY)           |        | V(i,j) |        |
            DP5 =  (DY*1.0D0) / APU5    +--------+---↑---+--------+
            AP5 = RO * DP5 * DY         |P(i-1,j)| P(i,j) |P(i+1,j)|
          END IF                        |   ・   →   ・   →   ・   |
*P'（左面）                              |     U(i-1,j)  U(i,j)     |
          IF ( IX.EQ.1 ) THEN           +--------+---↑---+--------+
            AP1 = 0.0D0                 |        |V(i,j-1)|        |
          ELSE                          |        |   ・   |        |
            APU1 = AU3(IX-1,IY)         |        |P(i,j-1)|        |
            DP1 =  (DY*1.0D0) / APU1    +--------+--------+--------+
            AP1 = RO * DP1 * DY
          END IF
*P'（上面）
          IF ( IY.EQ.NY ) THEN
            AP4 = 0.0D0
          ELSE
            APV4 = AV3(IX,IY)
            DP4 = (DX*1.0D0) / APV4
            AP4 = RO * DP4 * DX
```

```
          END IF
*P'（下面）
          IF ( IY.EQ.1 ) THEN
            AP2 = 0.0D0
          ELSE
            APV2 = AV3(IX,IY-1)
            DP2 = (DX*1.0D0) / APV2
            AP2 = RO * DP2 * DX
          END IF
          AP3 = AP1 + AP2 + AP4 + AP5
* 線形システム計算用に係数行列を計算
          ATP3(I) =  AP3
          ATP5(I) = -AP5
          ATP1(I) = -AP1
          ATP4(I) = -AP4
          ATP2(I) = -AP2
*１次独立な解を得るための処理 : IRELP=1 : 直接法においては必須
*(IX=1,IY=1 ---> K=1を基準点とし，常にP(1,1)=PD(1,1)=0とする)
          IF (IRELP.EQ.1. AND. IX.EQ.1. AND. IY.EQ.1) THEN
            ATP4(1) = 0.0D0
            ATP5(1) = 0.0D0
            BP(1)   = 0.0D0
*           K=2 の点の処理(K=1とのリンクを断つ)
            ATP2(2) = 0.0D0
*           K=1+NY の点の処理(K=1とのリンクを断つ)
            ATP1(1+NY) = 0.0D0
          END IF
* 線形システム解法前のための初期化(任意) (--- SIMPLE ---)
* METHOD=1,4,5のとき適宜修正
          XX(I) = PD(IX,IY)
          I =  I + 1
   20   CONTINUE
   10 CONTINUE
* --- SIMPLE --- 圧力補正 P'(PD) に関する解法
*サブルーチンに一般性を持たせるため，NX,NY,NEを引数として渡す
*COMMON文で定義されている値でもあるので，そのまま渡せない！
      NNX = NX
      NNY = NY
      NNE = NEP
C   1. 直接法 : バンドマトリックスによるガウスの消去法
      IF (METHOD.EQ.1)
     $  CALL GB      (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                AGB)
C   2. 反復法1 : point-SOR 法
      IF (METHOD.EQ.2)
     $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
C   3. 反復法2 : line-SOR 法 : OMG=1で十分．大きくしすぎると発散する
      IF (METHOD.EQ.3)
     $  CALL LSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6,W7,W8,W9)
C   4. クリロフ部分空間法1 : 共役残差法
```

```
      IF (METHOD.EQ.4)
     $  CALL CRB     (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6)
C   5. クリロフ部分空間法2 : BiCGSTAB
      IF (METHOD.EQ.5)
     $  CALL BICGB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                W1,W2,W3,W4,W5,W6,W7)
C   METHOD=4,5のときの探索ベクトル計算時のゼロ除算対策
      IF (IFLG.EQ.2)
     $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
*
      DO 30 IX = 1,NX
        DO 40 IY = 1,NY
          K = IY + (IX-1)*NY
*         圧力の相対性の処理 : 基準値の設定
*         1次独立な解を求める場合は圧力の基準点が強制的に設定される
*         圧力の基準点を設けない場合 (IRELP=0,1)
          IF (IRELP.EQ.0. OR. IRELP.EQ.1) PD(IX,IY) = XX(K)
*         1次従属な解のうちの1つを求めた後, 圧力基準を設ける場合 (IRELP=2)
*         P'(1,1)=0 ---> P(1,1)=0
          IF (IRELP.EQ.2) PD(IX,IY) = XX(K)-XX(1)
   40   CONTINUE
   30 CONTINUE
*圧力補正の境界条件の処理
      CALL PDBND (PD)
*速度の修正.........................................式(14.118, 119)
      DO 50 IY = 1,NY
        DO 60 IX = 1,NX-1
          DU = DY*1.0D0 / AU3(IX,IY)
          U(IX,IY)=US(IX,IY)+DU*(PD(IX,IY)-PD(IX+1,IY))
   60   CONTINUE
   50 CONTINUE
      DO 70 IY = 1,NY-1
        DO 80 IX = 1,NX
          DV = DX*1.0D0 / AV3(IX,IY)
          V(IX,IY)=VS(IX,IY)+DV*(PD(IX,IY)-PD(IX,IY+1))
   80   CONTINUE
   70 CONTINUE
*新たに得られた速度を用いて境界条件を処理する
      CALL UBND (U)
      CALL VBND (V)
*P* の修正: P = P* + P': ただしPの値はPSに代入........式(14.124)
      DO 90 IY = 1,NY
        DO 100 IX = 1,NX
          PS(IX,IY) = PS(IX,IY) + ALPHAP*PD(IX,IY)
  100   CONTINUE
   90 CONTINUE
*新たに得られたP*を用いて境界条件を処理する
      CALL PSBND (PS)
*速度場 (U*,V*) が EPSC 以下で連続の式を満たすかどうかを判定
      IF ( DMAX. GT. EPSC ) IFLGC=1
```

```
      RETURN
      END
**********************************************************************
*                        温度場の計算
**********************************************************************
*                                                                    *
* [温度の線形システムに関する配列の説明]                                 *
*                                                                    *
* 温度 (T) の線形システムを ATP_{i,j} T_{i} = BP_{i} とする.             *
* 有限差分近似を用いて離散化していることから明らかなように,                  *
* 1. ATP_{i,j}の大部分はゼロ                                            *
* 2. 非ゼロ要素は疎であり, 規則的に並んでいる                              *
* したがって,ATP_{i,j}すべてを記憶させるのは計算容量のムダであるので*
* 本プログラムでは, 以下のような規則にしたがって記憶する.                   *
* 2次元問題であれば, 近隣の5点が関わる.                                  *
*                                                                    *
* ●用いる配列(1次元配列に格納) : クリロフ部分空間法を含む反復法           *
*   係数行列     ---> ATP1, ATP2, ATP3, ATP4, ATP5                    *
*   既知ベクトル ---> BP                                               *
*   未知ベクトル ---> XX                                               *
*   未知数の数(未知のPDの数)   ---> NEP = NX * NY                       *
*                                                                    *
*    ATP1(NE),ATP2(NE),ATP3(NE),ATP4(NE),ATP5(NE)                    *
*              i=3, j=3 (k=13) : T(3,3) における ATP1,ATP2,ATP3        *
*                 +-------------------------+      ATP4,ATP5          *
*          (NY)5  |    |    |    |    |    |                          *
*                 +-------------------------+     通し番号の規則        *
*              4  |    |    |ATP4|    |    |     k=(i-1)*NY+j          *
*                 +-------------------------+                        *
*              3  |    |ATP1|ATP3|ATP5|    |     ATP1(k)              *
*                 +-------------------------+     ATP2(k)              *
*              2  |    |    |ATP2|    |    |     ATP3(k)              *
*                 +-------------------------+     ATP4(k)              *
*              1  |    |    |    |    |    |     ATP5(k)              *
*                 +-------------------------+     BP(k)                *
*              j   i=1  2    3    4    5(NX)                          *
*                                                                    *
*    BP(NE) : 既知ベクトル                                              *
*                                                                    *
*    XX(NE) : 未知ベクトル(各サブルーチンでこれが求まる)                     *
*                                                                    *
* ●用いる配列 : ガウスの消去法による直接法(バンド行列として格納)           *
*    AGB(-NY:NY,NEP) : 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 ATP1 から ATP5 までの k は*
*                    ATP1のkは K"-NY"                                 *
*                    ATP2のkは K"-1 "                                 *
*                    ATP3のkは K"+0 "                                 *
*                    ATP4のkは K"+1 "                                 *
*                    ATP5のkは K"+NY"                                 *
*                このように, kを固定したとき, " "で囲まれた5個の値       *
*                (-NY,-1,0,1,NY)                                      *
```

```
*     NEP -> 上述のような k は全部で NEP 個定義される                   *
*                                                                        *
* ●線形システムを構成する際の注意                                       *
* 線形システムを構成する際, 境界条件をどこで反映させるかによって,       *
* プログラミングが異なる. 本プログラムでは, 境界条件は, 係数行列を     *
* 作成する際に反映させ, 線形システムの解法においては専ら AX=B のみ     *
* に着目する立場をとる.                                                  *
* 上述のような立場とは異なり, 線形システムの解法において境界条件を     *
* 反映させることもできるが, ここでは, 線形システムの解法のサブルー     *
* チンに汎用性をもたせることを優先させた.                                *
*                                                                        *
**************************************************************************
        SUBROUTINE CALTEM (AGB,W1,W2,W3,W4,W5,W6,W7,W8,W9)
**************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $           U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $          US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $          PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $          T(0:NX0+1,0:NY0+1),
     $          XX(NEP0)
      COMMON / ARRAY2 /
     $          U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*     作業用配列
      DIMENSION W1(NEP0),W2(NEP0),W3(NEP0),W4(NEP0),W5(NEP0),W6(NEP0),
     $           W7(NEP0),W8(NEP0),W9(NEP0)
*     バンドガウス消去法のための係数行列配列
      DIMENSION AGB(-NY0:NY0,NEP0)
*
*     線形システム解法のため, 2次元配列を1次元配列にするための変数
      I = 1
      DO 10 IX = 1,NX
        DO 20 IY = 1,NY
*         Tに関するDの計算..........................式(14.85)
          DT5 = ALP * DY / DX...D_e
          DT1 = ALP * DY / DX...D_w
          DT4 = ALP * DX / DY...D_n
```

```
          DT2 = ALP * DX / DY...D_s
*         Tに関するFの計算
*         速度・圧力・温度場の繰り返し計算の際のUとVを線形システムの係数に反映させる
          UT5 = U(IX  ,IY  )...u_e
          UT1 = U(IX-1,IY  )...u_w             +--------+--------+--------+
          VT4 = V(IX  ,IY  )...v_n             |        |P(i,j+1)|        |
          VT2 = V(IX  ,IY-1)...v_s             |        |   ・   |        |
          FT5 = RO * UT5 * DY...F_e            |        | V(i,j) |        |
          FT1 = RO * UT1 * DY...F_w            +--------+---↑---+--------+
          FT4 = RO * VT4 * DX...F_n            |P(i-1,j)| P(i,j) |P(i+1,j)|
          FT2 = RO * VT2 * DX...F_s            |   ・   →   ・   →   ・   |
*         Tに関するPeの計算                     |     U(i-1,j)  U(i,j)     |
          PET5 = FT5 / DT5...Pe_e              +--------+---↑---+--------+
          PET1 = FT1 / DT1...Pe_w              |        |V(i,j-1)|        |
          PET4 = FT4 / DT4...Pe_n              |        |   ・   |        |
          PET2 = FT2 / DT2...Pe_s              |        |P(i,j-1)|        |
*         関数A(|Pe|)の計算                     +--------+--------+--------+
*         NMT=1なら 風上法
          IF ( NMT.EQ.1 ) THEN
            APET5 = 1.0D0...A(Pe_e)
            APET1 = 1.0D0...A(Pe_w)
            APET4 = 1.0D0...A(Pe_n)
            APET2 = 1.0D0...A(Pe_s)
*         NMT=2なら べき乗法
          ELSE
            APET5 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PET5))**5 ) )...A(Pe_e)
            APET1 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PET1))**5 ) )...A(Pe_w)
            APET4 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PET4))**5 ) )...A(Pe_n)
            APET2 = DMAX1( 0.0D0,DBLE( (1.0D0-0.1D0*ABS(PET2))**5 ) )...A(Pe_s)
          END IF
*         T の線形システムの係数の計算..............式(14.84)
          AT5 = ( DT5*APET5 + DMAX1(-FT5,0.0D0) )...a_e
          AT1 = ( DT1*APET1 + DMAX1( FT1,0.0D0) )...a_w
          AT4 = ( DT4*APET4 + DMAX1(-FT4,0.0D0) )...a_n
          AT2 = ( DT2*APET2 + DMAX1( FT2,0.0D0) )...a_s
*         生成項..................................式(14.88)
          SCT = 0.0D0
          SPT = 0.0D0
*         T の線形システムのBの計算.................式(14.88)
          BP(I) = SCT*DX*DY + RO*DX*DY/DT*T0(IX,IY)
*境界の処理
* 1. 境界のT_BND についても変数とみなしてAT3を求める
* 2. その後, 境界の項 AT3*T_BND をBに吸収させる
* 3. 2に対応して(吸収の後), 境界のAT(1,2,4,5)をゼロとしてリンクを断つ
          AT3 = AT1+AT2+AT4+AT5 + RO*DX*DY/DT - SPT*DX*DY
*T（右面）
          IF (IX.EQ.NX) THEN
            BP(I) = BP(I) + AT5 * T(NX+1,IY)
            AT5 = 0.0D0
*T（左面）
          ELSE IF (IX.EQ.1) THEN
```

```
            BP(I) = BP(I) + AT1 * T(0,IY)
            AT1 = 0.0D0
          END IF
*T（上面）
          IF (IY.EQ.NY) THEN
            BP(I) = BP(I) + AT4 * T(IX,NY+1)
            AT4 = 0.0D0
*T（下面）
          ELSE IF (IY.EQ.1) THEN
            BP(I) = BP(I) + AT2 * T(IX,0)
            AT2 = 0.0D0
          END IF
* 線形システム計算用に係数行列を計算
* ALPHAT : T 計算のための緩和係数
          ATP3(I) =   AT3 / ALPHAT
          ATP5(I) = - AT5
          ATP1(I) = - AT1
          ATP4(I) = - AT4
          ATP2(I) = - AT2
          BP(I) = BP(I) + (1.0D0-ALPHAT)*ATP3(I)*T0(IX,IY)
* 線形システム解法前のための初期化(任意) (--- SIMPLE ---)
* METHOD=1,4,5のとき適宜修正
          XX(I) = T(IX,IY)
          I = I + 1
   20   CONTINUE
   10 CONTINUE
*
*サブルーチンに一般性を持たせるため，NX,NY,NEを引数として渡す
*COMMON文で定義されている値でもあるので，そのまま渡せない！
      NNX = NX
      NNY = NY
      NNE = NEP
C    1. 直接法 : バンドマトリックスによるガウスの消去法
      IF (METHOD.EQ.1)
     $  CALL GB       (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                 AGB)
C    2. 反復法1 : point-SOR 法
      IF (METHOD.EQ.2)
     $  CALL PSORB    (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
C    3. 反復法2 : line-SOR 法 : OMG=1で十分．大きくしすぎると発散する
      IF (METHOD.EQ.3)
     $  CALL LSORB    (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                 W1,W2,W3,W4,W5,W6,W7,W8,W9)
C    4. クリロフ部分空間法1 : 共役残差法
      IF (METHOD.EQ.4)
     $  CALL CRB      (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                 W1,W2,W3,W4,W5,W6)
C    5. クリロフ部分空間法2 : BiCGSTAB
      IF (METHOD.EQ.5)
     $  CALL BICGB    (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE,
     $                 W1,W2,W3,W4,W5,W6,W7)
```

```
C   METHOD=4,5のときの探索ベクトル計算時のゼロ除算対策
       IF (IFLG.EQ.2)
      $  CALL PSORB   (ATP1,ATP2,ATP3,ATP4,ATP5,BP,XX,NNX,NNY,NNE)
*線形システム解法で得られた1次元配列の解を2次元配列に置き換える
      DO 30 IX = 1,NX
        DO 40 IY = 1,NY
          K = IY + (IX-1)*NY
          T(IX,IY) = XX(K)
   40   CONTINUE
   30 CONTINUE
*境界条件の処理
      CALL TBND (T)
      RETURN
      END
*********************************************************************
*                速度 ( U* と U ) の境界条件の処理
*********************************************************************
      SUBROUTINE UBND (XU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D1 / NX,NY
      DIMENSION XU(0:NX,0:NY+1)
*U (左面) ......................................................境界で定義可能
      DO 20 IY = 1,NY
        XU(0,IY) = 0.0D0
   20 CONTINUE
*U (右面) ......................................................境界で定義可能
      DO 30 IY = 1,NY
        XU(NX,IY) = 0.0D0
   30 CONTINUE
*U (下面) ..............................境界で定義できないので仮想セルを用いる
      DO 40 IX = 0,NX
        XU(IX,0) = -XU(IX,1).........................................式(14.57)
   40 CONTINUE
*U (上面) ..............................境界で定義できないので仮想セルを用いる
      DO 50 IX = 0,NX
        XU(IX,NY+1) = -XU(IX,NY).....................................式(14.57)
   50 CONTINUE
*
      RETURN
      END
*********************************************************************
*                速度 ( V* と V ) の境界条件の処理
*********************************************************************
      SUBROUTINE VBND (XV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D1 / NX,NY
      DIMENSION XV(0:NX+1,0:NY)
*V (左面) ..............................境界で定義できないので仮想セルを用いる
      DO 20 IY = 1,NY-1
        XV(0,IY) = -XV(1,IY).........................................式(14.58)
   20 CONTINUE
```

```
*V（右面）................................境界で定義できないので仮想セルを用いる
      DO 40 IY = 1,NY-1
        XV(NX+1,IY) = -XV(NX,IY)......................................式(14.58)
   40 CONTINUE
*V（上面）.....................................................境界で定義可能
      DO 50 IX = 0,NX+1
        XV(IX,NY) = 0.0D0
   50 CONTINUE
*V（下面）.....................................................境界で定義可能
      DO 60 IX = 0,NX+1
        XV(IX,0) = 0.0D0
   60 CONTINUE
*
      RETURN
      END
**********************************************************************
*                  温度の境界条件の処理
**********************************************************************
      SUBROUTINE TBND (XT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D1 / NX,NY
      DIMENSION XT(0:NX+1,0:NY+1)
.........温度は境界で定義できないので仮想セルを用いて境界条件を与える
         ここでは1次精度(前進あるいは後退)差分を使用(式(13.11),(13.12)を参照)
* 左面
      DO 20 IY = 1,NY
        XT(0,IY) = 2.0D0 * ( +0.5D0 ) - XT(1,IY)..................式(14.56)
   20 CONTINUE
* 右面
      DO 30 IY = 1,NY
        XT(NX+1,IY) = 2.0D0 * ( -0.5D0 ) - XT(NX,IY)..............式(14.56)
   30 CONTINUE
* 下面.............................................................勾配ゼロ
      DO 40 IX = 0,NX+1
        XT(IX,0) = XT(IX,1)
   40 CONTINUE
* 上面.............................................................勾配ゼロ
      DO 50 IX = 0,NX+1
        XT(IX,NY+1) = XT(IX,NY)
   50 CONTINUE
*
      RETURN
      END
**********************************************************************
*                       圧力補正の境界条件                           *
**********************************************************************
      SUBROUTINE PDBND (XP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D1 / NX,NY
      DIMENSION XP(0:NX+1,0:NY+1)
........圧力補正は境界で定義できないので仮想セルを用いる
```

```
         ここでは1次精度(前進あるいは後退)差分を使用(式(13.11),(13.12)を参照)
* 左面..................境界において法線方向の勾配ゼロを1次精度前進差分で与える
      DO 10 IY = 1,NY
        XP(0,IY) = XP(1,IY)
   10 CONTINUE
* 右面..................境界において法線方向の勾配ゼロを1次精度後退差分で与える
      DO 20 IY = 1,NY
        XP(NX+1,IY) = XP(NX,IY)
   20 CONTINUE
* 下面..................境界において法線方向の勾配ゼロを1次精度前進差分で与える
      DO 30 IX = 0,NX+1
        XP(IX,0) = XP(IX,1)
   30 CONTINUE
* 上面..................境界において法線方向の勾配ゼロを1次精度後退差分で与える
      DO 40 IX = 0,NX+1
        XP(IX,NY+1) = XP(IX,NY)
   40 CONTINUE
*
      RETURN
      END
**********************************************************************
*                       圧力の境界条件                              *
**********************************************************************
      SUBROUTINE PSBND (XP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D1 / NX,NY
      DIMENSION XP(0:NX+1,0:NY+1)
........圧力は境界で定義できないので仮想セルを用いる
         ここでは1次精度(前進あるいは後退)差分を使用(式(13.11),(13.12)を参照)
* 左面..................境界において法線方向の勾配ゼロを1次精度前進差分で与える
      DO 10 IY = 1,NY
        XP(0,IY) = XP(1,IY)
   10 CONTINUE
* 右面..................境界において法線方向の勾配ゼロを1次精度後退差分で与える
      DO 20 IY = 1,NY
        XP(NX+1,IY) = XP(NX,IY)
   20 CONTINUE
* 下面..................境界において法線方向の勾配ゼロを1次精度前進差分で与える
      DO 30 IX = 0,NX+1
        XP(IX,0) = XP(IX,1)
   30 CONTINUE
* 上面..................境界において法線方向の勾配ゼロを1次精度後退差分で与える
      DO 40 IX = 0,NX+1
        XP(IX,NY+1) = XP(IX,NY)
   40 CONTINUE
*
      RETURN
      END
**********************************************************************
*                        データ出力
**********************************************************************
```

```
      SUBROUTINE PROUT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
*     圧力計算の総反復回数を出力できるようにした 3.16,2000
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $          U(0:NX0,  0:NY0+1),  V(0:NX0+1,0:NY0  ),
     $         US(0:NX0,  0:NY0+1), VS(0:NX0+1,0:NY0  ),
     $         PS(0:NX0+1,0:NY0+1), PD(0:NX0+1,0:NY0+1),
     $         T(0:NX0+1,0:NY0+1),
     $         XX(NEP0)
      COMMON / ARRAY2 /
     $         U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
*
      WRITE (11) U
      WRITE (12) V
      WRITE (13) PS,PD
      WRITE (14) T
      RETURN
      END
**********************************************************************
*                 Tecplot用データ出力
**********************************************************************
      SUBROUTINE TECPLT (FNAME)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX0=20,NY0=20,NEU0=380,NEV0=380,NEP0=400)
      COMMON / D1 / NX,NY
      COMMON / D2 / DX,DY,DT
      COMMON / D3 / VIS,ALP,BUO
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
C     --- SIMPLE ---
*     圧力計算の総反復回数を出力できるようにした 3.16,2000
      COMMON / D8 / NEU,NEV,NEP
      COMMON / D9 / NITR, NCNT
      COMMON / D10 / RO
```

```
      COMMON / ARRAY1 / AU3(0:NX0,0:NY0+1),AV3(0:NX0+1,0:NY0),
     $ATP1(NEP0),ATP2(NEP0),ATP3(NEP0),ATP4(NEP0),ATP5(NEP0),BP(NEP0),
     $           U(0:NX0,   0:NY0+1),   V(0:NX0+1,0:NY0  ),
     $          US(0:NX0,   0:NY0+1),  VS(0:NX0+1,0:NY0  ),
     $          PS(0:NX0+1,0:NY0+1),  PD(0:NX0+1,0:NY0+1),
     $          T(0:NX0+1,0:NY0+1),
     $          XX(NEP0)
      COMMON / ARRAY2 /
     $          U0(0:NX0,0:NY0+1),V0(0:NX0+1,0:NY0),T0(0:NX0+1,0:NY0+1)
      CHARACTER FNAME*20
*
      OPEN (21,FILE=FNAME,STATUS='NEW')
      WRITE (21,*) 'VARIABLES = "X", "Y", "U", "V", "T"'
      NX1 = NX+1
      NY1 = NY+1
      WRITE (21,4000) NX1,NY1
 4000 FORMAT (1H ,'ZONE I=',I3,',J=',I3,' F=POINT')
      DO 10 IY = 0,NY
        DO 20 IX = 0,NX                                      ^
           X = DX * FLOAT(IX)                         +-----|-----+
           Y = DY * FLOAT(IY)                         |           |
           UU = ( U(IX  ,IY)+U(IX  ,IY+1) )/2.0E0     |           |
           VV = ( V(IX  ,IY)+V(IX+1,IY  ) )/2.0E0     ->    .     ->
           TT = ( T(IX  ,IY)  +T(IX+1,IY  )           |           |
     $            +T(IX  ,IY+1)+T(IX+1,IY+1) )/4.0E0  |     ^     |
          WRITE (21,4010) X,Y,UU,VV,TT                +-----|-----+
 4010     FORMAT (1H ,5(1PE11.3))              各種データを格子分割図の交わる点
   20   CONTINUE                            (上の図の四隅の点)における値に変換して出力
   10 CONTINUE
      CLOSE(21)
      RETURN
      END
*********************************************************************
*                       各種の線形システム解法                       *
*                                                                    *
*  1. ガウスの消去法                                                 *
*  2. point-SOR 法                                                   *
*  3. line-SOR 法                                                    *
*  4. 共役残差法                                                     *
*  5. Bi-CGSTAB法                                                    *
*                                                                    *
*  いずれも，2次元のポアソン方程式を5点差分近似にて離散化した        *
*  線形システムを解くためのもので，最適化してある                    *
*  いずれのサブルーチンも同一引数としてある                          *
*                                                                    *
* [注意]                                                             *
*  1. 収束判定条件は適宜変更のこと.                                  *
*  2. 引数の NX,NY,NE と，配列宣言文の A1,A2,A3,A4,A5,B,X は，       *
*     いずれもこのサブルーチンがコールされている PRESS において対応 *
*     するものと同じ名前としてあるが，COMMON文で定義していないので，*
*     計算機の中では異なる変数として定義される．本計算プログラムに  *
```

```
*      おいては，できるだけ線形システム解法のサブルーチンに汎用性を  *
*      もたせるため，あえて，COMMON文は使用していない．また，分かり  *
*      やすくするため，サブルーチンがコールされている個所と同じ名前で*
*      それぞれの引数を定義してある．以降同様．                    *
*                                                                  *
********************************************************************
*
********************************************************************
*                                                                  *
*    2次元ラプラシアン離散化による5点差分近似                      *
*    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              *
*                         AX=B                                     *
*    をGaussの消去法を用いて解くサブルーチン．（軸選択無）          *
*    Aはバンドマトリックス                                         *
*                                                                  *
*     線形システム --->  A_{i,j} X_{i} = B_{i}                     *
*                                                                  *
*     係数行列の計算容量節約：詳細はサブルーチン PRESS を参照      *
*     A_{i,j} ---> A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)              *
*             ---> A(-NY:NY,NE)に格納しなおす                      *
*                                                                  *
*     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                                    *
*                                                                  *
********************************************************************
      SUBROUTINE GB (A1,A2,A3,A4,A5,B,X,NX,NY,NE,
     $               A)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE)
      DIMENSION A(-NY:NY,NE)
      DIMENSION B(NE)
      DIMENSION X(NE)
*
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
C     --- SIMPLE ---
      COMMON / D9 / NITR, NCNT
*
*直接法のときは(係数行列が特異でなければ)反復なしで，必ず解を得る
      IFLG = 0
*
*マトリックスAのゼロクリア
      DO 10 INE = 1,NE
        DO 20 I = -NY,NY
          A(I,INE) = 0.0D0
```

```
   20   CONTINUE
   10 CONTINUE
*
*必要なところにA1からA5までを格納する
      DO 30 INE = 1,NE
        A(-NY,INE) = A1(INE)
        A( -1,INE) = A2(INE)
        A(  0,INE) = A3(INE)
        A(  1,INE) = A4(INE)
        A( NY,INE) = A5(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 = A(-J,I+J)/A(0,I)
            B(I+J) = B(I+J) - B(I)*AA
            N = 1
            DO 60 K = -J+1,NY-J
              A(K,I+J) = A(K,I+J)-A(N,I)*AA
              N = N + 1
   60       CONTINUE
   50     CONTINUE
        ELSE
          DO 70 J = 1,NE-I
            AA = A(-J,I+J)/A(0,I)
            B(I+J) = B(I+J) - B(I)*AA
            N = 1
            DO 80 K = -J+1,NE-I-J
              A(K,I+J) = A(K,I+J)-A(N,I)*AA
              N = N + 1
   80       CONTINUE
   70     CONTINUE
        END IF
   40 CONTINUE
*
*後退代入
*     係数行列の特異性を判定
*.....係数行列が特異なら計算終了する (--- SIMPLE ---)
      IF ( DABS(A(0,NE)).LE.1.0D-50 ) THEN
        WRITE (6,*) ' Matrix singular : |A(0,NE)| < 1E-50 '
        IFLG = 1
        RETURN
      END IF
      X(NE) = B(NE) / A(0,NE)
*
      DO 90 I = NE-1,1,-1
        S = 0.0D0
        IF ( I.GT.NE-NY ) THEN
          DO 100 N = 1,NE-I
```

SIMPLE2D.FOR 36

```
              S = S + A(N,I)* X(I+N)
  100       CONTINUE
            X(I) = ( B(I)-S ) / A(0,I)
          ELSE
            DO 110 N = 1,NY
              S = S + A(N,I)* X(I+N)
  110       CONTINUE
            X(I) = ( B(I)-S ) / A(0,I)
          END IF
   90 CONTINUE
*
*.....線形システムの総反復回数出力のために追加 (--- SIMPLE ---)
      ITR = 1
*
*サブルーチン終了
      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*
*  2次元ラプラシアン離散化による5点差分近似用                        *
*                                                                    *
*    線形システム --->  A_{i,j} X_{i} = B_{i}                        *
*                                                                    *
*    係数行列の計算容量節約：詳細はサブルーチン PRESS を参照         *
*    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                 *
*                                                                    *
* [収束判定条件]                                                     *
*  (¥vec{x}^{new}-¥vec{x}^{old})^{2} < EPSP                          *
*  ¥vec{x}^{old} : 前の反復による値                                  *
*  ¥vec{x}^{new} : 新しい反復による値                                *
*                                                                    *
**********************************************************************
      SUBROUTINE PSORB (A1,A2,A3,A4,A5,B,X,NX,NY,NE)...........(15.4.3)項を参照
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
C     --- SIMPLE ---
      COMMON / D9 / NITR, NCNT
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),X(NE)
```

```
*
*     IFLGの初期値は"収束せず"
      IFLG=1
*
      DO 10 J = 1,NITR
*
        RNORM1 = 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 )
          RNORM1 = RNORM1 + ( XNEW - XOLD )**2
*...........................................................A2,A3,A4,A5の範囲
        DO 20 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 )
          RNORM1 = RNORM1 + ( XNEW - XOLD )**2
   20   CONTINUE
*........................................................A1,A2,A3,A4,A5の範囲
        DO 30 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 )
          RNORM1 = RNORM1 + ( XNEW - XOLD )**2
   30   CONTINUE
*...........................................................A1,A2,A3,A4の範囲
        DO 40 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 )
          RNORM1 = RNORM1 + ( XNEW - XOLD )**2
   40   CONTINUE
*..............................................................A1,A2,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 )
          RNORM1 = RNORM1 + ( XNEW - XOLD )**2
*
*       収束判定; 収束なら IFLG=0 に設定
        IF (RNORM1.LE.EPSP) THEN
          IFLG=0
          ITR = J
          GO TO 700
```

```
        END IF
   10 CONTINUE
*
* 収束と判定されたときの分岐点
  700 CONTINUE
*
*サブルーチン終了
      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 *
*  2次元ラプラシアン離散化による5点差分近似用                       *
*                                                                    *
*    線形システム --->  A_{i,j} X_{i} = B_{i}                        *
*                                                                    *
*    係数行列の計算容量節約：詳細はサブルーチン PRESS を参照         *
*    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                    *
*                                                                    *
* [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                 *
*    OMG  : 緩和係数(IN2D.MAC)にて設定. 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 *
*                                                                    *
* [収束判定条件]                                                     *
*  (¥vec{x}^{new}-¥vec{x}^{old})^{2} < EPSP                          *
*  ¥vec{x}^{old} : 前の反復による値                                  *
*  ¥vec{x}^{new} : 新しい反復による値                                *
*                                                                    *
* [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)                 *
* XO...各方向への掃引前のX(番号付けは不変)                           *
* XOLD...このサブルーチンに入る前のX(番号付けは不変)                 *
*                                                                    *
**********************************************************************
      SUBROUTINE LSORB (AT1,AT2,AT3,AT4,AT5,BX,XN,NX,NY,NE, ...........(15.3.5)項を参照
     $                  X1,XO,XOLD,A,B,C,D,U,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
C     --- SIMPLE ---
```

```
      COMMON / D9 / NITR, NCNT
      DIMENSION AT1(NE),AT2(NE),AT3(NE),AT4(NE),AT5(NE)
      DIMENSION XN(NE),X1(NE),BX(NE),XO(NE),XOLD(NE)
      DIMENSION A(NE),B(NE),C(NE),D(NE),U(NE),Y(NE)
*
*     IFLGの初期値は”収束せず”
      IFLG=1
*
      DO 10 K=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XOに保存
          XO(INY) = XN(INY)
*         x方向へのトーマス法で答えを求める前のXNをXOLDに保存
          XOLD(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(15.31)
  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XOにより最新のXNを緩和
```

```
          XN(INY)=(1.0D0-OMG)*XO(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XOに保存
          XO(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= 0.0D0
      DO 300 I = 1,NE
        RNORM= RNORM + (XN(I)-XOLD(I))**2
```

```
  300 CONTINUE
*
*     収束判定
      IF (RNORM.LE.EPSP) THEN
        IFLG=0
        ITR=K
        GO TO 900
      END IF
*
   10 CONTINUE
*
* 収束と判定されたときの分岐点
  900 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                                      *
*                         AX=B                                       *
*                                                                    *
*    係数行列の計算容量節約：詳細はサブルーチン PRESS を参照         *
*    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 ---> ここでは便宜上配列 XP(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 : 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]                                                       *
*    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                                                  *
*                                                                    *
* [収束判定条件]                                                     *
*  (¥vec{x}^{new}-¥vec{x}^{old})^{2} < EPSP                          *
*  ¥vec{x}^{old} : 前の反復による値                                  *
*  ¥vec{x}^{new} : 新しい反復による値                                *
*                                                                    *
**********************************************************************
      SUBROUTINE CRB (A1,A2,A3,A4,A5,B,XP,NX,NY,NE, ....(15.5.3)項を参照
     $                 R,P,AP,AR,X,XOLD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
```

```
     $               ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
C     --- SIMPLE ---
      COMMON / D9 / NITR, NCNT
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),XP(NE)
* 作業用配列
      DIMENSION R(NE),P(NE),AP(NE),AR(NE),X(NE),XOLD(NE)
*
* R に AX を代入
      CALL PROMV (A1,A2,A3,A4,A5,X,R,NX,NY,NE)
*
      DO 40 I = 1, NE
*       r_{0}とp_{0}(初期値)の設定
        R(I) = B(I) - R(I)..........................................式(15.103)
        P(I) = R(I).................................................式(15.104)
*       前の時刻のXをXOLDに代入
        XOLD(I) = XP(I)
   40 CONTINUE
*
* APに A p_{0} を代入
      CALL PROMV (A1,A2,A3,A4,A5,P,AP,NX,NY,NE)
*
* 反復計算
      DO 50 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)
***********************************************************
*.......探索方向のための計算が 0除算 なら計算終了(--- SIMPLE ---)
        IF (DABS(APAP).LT.1.0D-50) THEN
          WRITE (6,*) ' 0 division : ALPHA_{K} in Conjugate Residual '
          IFLG = 2
          RETURN
        ELSE
          ALP = RAP / APAP
        END IF
***********************************************************
*
        RNORM = 0.0D0
*
        DO 70 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)
*         前の反復との差のノルムの計算
          RNORM = RNORM + (X(I)-XOLD(I))**2
*         得られたXをXOLDに代入
          XOLD(I) = X(I)
```

```
   70   CONTINUE
*
* RNORM が EPSP 以下なら収束とみなして 700 へ
        IF (RNORM.LE.EPSP) THEN
          IFLG=0
          ITR=K
          GO TO 700
        END IF
*
* 収束せずの場合
*       A r_{k+1} の計算 => AR(NE)
        CALL PROMV (A1,A2,A3,A4,A5,R,AR,NX,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)
**********************************************************
*.......探索方向のための計算が 0除算 なら計算終了(--- SIMPLE ---)
        IF (DABS(APAP).LT.1.0D-50) THEN
          WRITE (6,*) ' 0 division : BETA_{K} in Conjugate Residual '
          IFLG = 2
          RETURN
        ELSE
          BETA = -ARAP / APAP
        END IF
**********************************************************
*
        DO 90 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}
          AP(I) = AR(I) + BETA*AP(I)
   90   CONTINUE
*
   50 CONTINUE
*     NITR まで計算しても収束せず
      IFLG=1
*
* 収束と判定されたときの分岐点
  700 CONTINUE
*
      DO 100 I = 1,NE
        XP(I) = X(I)
  100 CONTINUE
*
*サブルーチン終了
      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
*
***********************************************************************
*                                                                     *
*    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                                      *
*                                                                     *
*［変数の説明］                                                       *
*    NE : 総格子点数(正方マトリックス A,B,C のサイズ)                 *
*    C  : A と B の積(ベクトル)                                       *
*                                                                     *
***********************************************************************
      SUBROUTINE PROMV (A1,A2,A3,A4,A5,B,C,NX,NY,NE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A1(NE),A2(NE),A3(NE),A4(NE),A5(NE),B(NE),C(NE)
*.............................................................A3,A4,A5の範囲
      I=1
        C(I) = A3(I)*B(I)
     $      +A4(I)*B(I+1)+A5(I)*B(I+NY)
*..........................................................A2,A3,A4,A5の範囲
      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
*.......................................................A1,A2,A3,A4,A5の範囲
      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
*..........................................................A1,A2,A3,A4の範囲
      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
*.............................................................A1,A2,A3の範囲
```

```
      I=NE
        C(I) = A1(I)*B(I-NY)+A2(I)*B(I-1 )+A3(I)*B(I)
*
*サブルーチン終了
      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                                      *
*                                                                   *
*    係数行列の計算容量節約：詳細はサブルーチン PRESS を参照            *
*    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                                                *
*    AR(NE) ： A * T                                                *
*                                                                   *
* [収束判定条件]                                                      *
*  (¥vec{x}^{new}-¥vec{x}^{old})^{2} < EPSP                         *
*  ¥vec{x}^{old} ： 前の反復による値                                   *
*  ¥vec{x}^{new} ： 新しい反復による値                                 *
*                                                                   *
*********************************************************************
      SUBROUTINE BICGB (A1,A2,A3,A4,A5,B,X,NX,NY,NE, ....(15.5.4)項を参照
     $                  R,AP,AT,P,S,T,XOLD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP,EPSC,
     $              ALPHAP,ALPHAU,ALPHAV,ALPHAT
      COMMON / D5 / ICYCLE,ITR,IFLG,IFLGC,IRELP,METHOD,NMT
C     --- SIMPLE ---
      COMMON / D9 / NITR, NCNT
*
```

```
      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),XOLD(NE)
*
      DO 5 I = 1,NE
        XOLD(I) = X(I)
    5 CONTINUE
*
* R に AX を代入
      CALL PROMV (A1,A2,A3,A4,A5,X,R,NX,NY,NE)
* r_{0}とp_{0}(初期値), そして s=r_{0} の設定
      DO 10 I=1,NE
        R(I) = B(I) - R(I)........................................式(15.110)
        P(I) = R(I)...............................................式(15.111)
        S(I) = R(I)
   10 CONTINUE
*
* 繰り返し計算
      DO 20 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 PROMV (A1,A2,A3,A4,A5,P,AP,NX,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)
*********************************************************
*.......探索方向のための計算が 0除算 なら計算終了(--- SIMPLE ---)
        IF (DABS(SAP).LT.1.0D-50) THEN
          WRITE (6,*) ' 0 division : ALPHA_{K} in Bi-CGSTAB '
          IFLG = 2
          RETURN
        ELSE
          ALPHA = SR1/SAP
        END IF
*********************************************************
        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 PROMV (A1,A2,A3,A4,A5,T,AT,NX,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)
*********************************************************
*.......探索方向のための計算が 0除算 なら計算終了(--- SIMPLE ---)
        IF (DABS(ATAT).LT.1.0D-50) THEN
          WRITE (6,*) ' 0 division : XI_{K} in Bi-CGSTAB '
```

```
          IFLG = 2
          RETURN
        ELSE
          XI = ATT/ATAT
        END IF
**********************************************************
        RNORM = 0.0D0
        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)
*         前の反復との差のノルムの計算
          RNORM = RNORM + (X(I)-XOLD(I))**2
*         得られたXをXOLDに代入
          XOLD(I) = X(I)
   60   CONTINUE
* RNORM が EPSP 以下なら収束とみなして 900 へ
        IF (RNORM.LE.EPSP) THEN
          IFLG=0
          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 70 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) )
   70   CONTINUE
   20 CONTINUE
*     NITR まで計算しても収束せず
      IFLG=1
*
  900 CONTINUE
      RETURN
      END
*
```