

```

*****
* ファイル名   : SMAC2D.FOR
* タイトル    : SMAC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、対流項は非保存形で1次精度風上差分を、拡散項
*   は2次精度中心差分を用いて離散化している。さらに高精度の近似を行う
*   場合は、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
*   →
*    $\nabla \cdot \mathbf{V} = 0$ 
*   →
*   
$$\frac{D\mathbf{V}}{Dt} = -\nabla P + \nu \nabla^2 \mathbf{V} + \mathbf{B} \times \mathbf{T}(0,1)$$

*   
$$\frac{DT}{Dt} = \text{ALP} (\nabla^2 T)$$

*   方程式に応じて、"BIS", "BU0", "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
*   (1Y=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)          *
*          +---→ x

```

●スタaggerドメッシュについて

```

*          ←      DX      →
*  +-----+-----+-----+
*  |          |          |          |          |          |
*  |          |          P(i,j+1) |          |          |          |
*  |          |          .          |          |          |          |
*  |          |          V(i,j)   |          |          |          |
*  |          |          ↑          |          |          |          |
*  |          |          +-----+-----+-----+
*  |          |          |          |          |          |          |
*  | P(i-1,j) | P(i,j) | P(i+1,j) |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          U(i-1,j)  U(i,j) |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          V(i,j-1) |          |          |          |
*  |          |          ↑          |          |          |          |
*  |          |          +-----+-----+-----+
*  |          |          P(i,j-1) |          |          |          |
*  |          |          .          |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          |          |          |          |          |
*  |          |          |          |          |          |          |
*  +-----+-----+-----+

```

→ U 定義点
↑ V 定義点
・ P,T 定義点
(注)

プログラム中のTは、
本文中ではθとなっている。

●パラメーターファイル (in2d. mac) について

プログラムを実行すると," in2d. mac" 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. mac" のリスト

- * 1: U. NEW Uの計算結果の出力ファイル名
- * 2: V. NEW Vの計算結果の出力ファイル名
- * 3: P. NEW Pの計算結果の出力ファイル名
- * 4: T. NEW Tの計算結果の出力ファイル名
- * 5: U. OLD 継続計算のUの入力データ
- * 6: V. OLD 継続計算のVの入力データ
- * 7: P. OLD 継続計算のPの入力データ
- * 8: T. OLD 継続計算のTの入力データ
- * 9: UVT. NEW Tecplot用データ
- *10: +=====+

```

*11: | ITYPE  1==> isothermal | *
*12: |          2==>nonisothermal | *
*13: +=====+ *
*14: ----- *
*15: ITYPE      ICYCLE      NITR      NCYCLE *
*16: 2          0          10000    1000 *
*17: ----- *
*18: EPSP        OMG *
*19: 1.0e-3      1.7e+0 *
*20: ----- *
*21: DT          RE          PR          GR *
*22: 1.0e-4      0.0e+0      7.1e-1      1.0e+5 *
*23: ----- *
*24: DLX          DLY          IREL      METHOD *
*25: 1.0e+0      1.0e+0      2          5 *
* *
* ITYPE.....1:等温計算 2:非等温計算(1 1, 1 2行を参照) *
* ICYCLE.....計算開始のサイクル数(時刻T=ICYCLE*DT) *
*          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.... 圧力補正の線形システム解法(反復法・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 *
* EPSP.... 収束判定値 *
*          圧力補正の線形システム解法(反復法, クリロフ部分空間法) *
*          の収束評価に使用 *
* OMG.... 圧力補正のための緩和係数: (point, line-)SOR法の加速係数 *
* 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):DXは一定で等間隔格子 *
* DLY.... 解析領域の高さ(DLY=NY*DY):DYは一定で等間隔格子 *
*          格子幅(等間隔)DX,DYはプログラムの中で求める. *
* IREL... 圧力の基準値と解の1次独立性の設定 *
*          0: 行わない *
*          <SMACにおける圧力補正の線形システム解法に関して> *
*          -> 1次従属な解の1つを求めるのみ *
*          -> 直接法では特異行列の問題に遭遇. 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1つを求め, 圧力基準値P(1,1)=0を設定する. *
*              (IRELP=0の計算でPDを求め, PD(1,1)を差し引きP(1,1)=0とする)*
* METHOD.. 圧力補正の線形システム解法のアルゴリズム *
*          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 *
*          IREL=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正しくない.)*

```

```

*      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する *
*      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 *
*      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PRESSIにおいて, 基準として  $\delta P(1, 1) = P(1, 1) = 0$  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. *
*      数学的に厳密に言えば, 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 *
*      ITR -----> 圧力補正計算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) *
*      IX, IY -----> 上の図を参照 *
*      U0, V0, T0-----> 圧力補正計算を行う前の値 *
*      UN, VN, TN-----> 新たな圧力補正を用いて計算された値 *
*      PD -----> 圧力補正 *
*
* ●SMAC法における線形システム解法について *
*      静止状態直後など, 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。 *
* *
* 本プログラムは、HAMAC法のプログラムhsmac2d.forをもとにして作成 *
* してある。HSMACとSMACが本質的に同等であるので、プログラムの変更 *
* は、圧力補正をニュートン法で行う(HSMAC法)代わりに、線形システム *
* 解法で行うようにすればよい。なお、できる限り、hsmac2d.forとの違 *
* いをわかりやすくするため、SMAC法において新たに付け加えた個所に *
* は、"--- SMAC ---"としてコメントを挿入してある。 *
* *
*****
PROGRAM SMAC2D
*****
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER ( NX0=20, NY0=20, NEO=400 )
COMMON / D1 / NX,NY
COMMON / D2 / DX,DY,DT
COMMON / D3 / VIS,ALP,BUO
COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP
COMMON / D5 / ICYCLE,ITR,IFLG,IREL,METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
C --- SMAC ---
COMMON/ARRAY2/DIV(NX0,NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
* 圧力補正の線形システム用
DIMENSION A1(NEO), A2(NEO), A3(NEO), A4(NEO), A5(NEO),
$          B(NEO), X(NEO)
* 作業用配列
DIMENSION W1(NEO), W2(NEO), W3(NEO), W4(NEO), W5(NEO), W6(NEO),
$          W7(NEO), W8(NEO), W9(NEO)
* バンドガウス消去法のための係数行列配列
DIMENSION AGB(-NY0:NY0, NEO)
C
CHARACTER FNAME(10)*20
*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(圧力補正に関する未知数の数) --- SMAC ---

NE = NE0

*パラメータファイルのオープン

OPEN (10, FILE='IN2D.MAC', STATUS='OLD')

*出力ファイル名の読み込み

DO 10 I = 1, 9

READ (10, '(A20)') FNAME(I)

10 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')

*in2d.mac中のコメント行(10-15行目)のスキップ

READ (10, '(A)')

READ (10, '(A)')

READ (10, '(A)')

READ (10, '(A)')

READ (10, '(A)')

READ (10, '(A)')

READ (10, *) ITYPE, ICYCLE, NITR, NCYCLE

*in2d.mac中のコメント行(17-18行目)のスキップ

READ (10, '(A)')

READ (10, '(A)')

READ (10, *) EPSP, OMG

*in2d.mac中のコメント行(20-21行目)のスキップ

READ (10, '(A)')

READ (10, '(A)')

READ (10, *) DT, RE, PR, GR

*in2d.mac中のコメント行(23-24行目)のスキップ

READ (10, '(A)')

READ (10, '(A)')

READ (10, *) DLX, DLY, IRELP, METHOD

*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
*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 = 1.0D+0
*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 )
*等温場なら浮力項の係数はゼロに設定
      IF (ITYPE.EQ.1) BUO = 0.0D0
*初期値の設定
      CALL CINITI
      DO 20 I=1, NE
        A1(I)=0.0D0
        A2(I)=0.0D0
        A3(I)=0.0D0
        A4(I)=0.0D0
        A5(I)=0.0D0
        B(I)=0.0D0
        X(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 25 II = -NY, NY
        AGB(II, I)=0.0D0
      25 CONTINUE
      20 CONTINUE
*時間進行のための戻り点
      700 CONTINUE
*時間進行
      CALL ADV
*圧力補正の計算が収束したかどうかのパラメータ(IFLG)を初期化(反復法でのみ有効)
*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 <--- 第2版のプログラムより位置を移動
C      --- SMAC ---
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、圧力補正の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へ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) GOTO 720

```

```

*速度場の計算
  CALL CALVEL
*圧力場の計算
  CALL PRESS (A1, A2, A3, A4, A5, B, X,
    $          AGB, W1, W2, W3, W4, W5, W6, W7, W8, W9)
C    速度場と圧力場がゼロとなる時のスキップ先--- SMAC ---
  720 CONTINUE
*    圧力場の計算が収束したとき
  IF ( IFLG. EQ. 0 ) THEN
*      非等温場計算の場合
  IF ( ITYPE. EQ. 2 ) THEN
*        温度場を計算
        CALL CALTEM
      END IF
*    圧力場の計算が収束していないとき
  ELSE 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
*    時間進行カウンタ(ICYCLE)がNCYCLEより小さい時
  IF ( ICYCLE. LT. NCYCLE ) THEN
    GO TO 700
*    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)
  STOP
  END

*
*****
*          初期設定
*****
SUBROUTINE CINITI
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
  PARAMETER ( NX0=20, NY0=20, NEO=400 )
  COMMON / D1 / NX, NY
  COMMON / D2 / DX, DY, DT
  COMMON / D3 / VIS, ALP, BUO

```



```

COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
C      --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
*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
          DO 10 IX = 0, NX
              DO 20 IY = 0, NY+1
                  UN(IX, IY) = 0.0D0
20          CONTINUE
10          CONTINUE
*          Vの初期値設定..... 計算開始時は静止
          DO 30 IY = 0, NY
              DO 40 IX = 0, NX+1
                  VN(IX, IY) = 0.0D0
40          CONTINUE
30          CONTINUE
*          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
                                (12.3節を参照)
                                圧力補正(PD)と変動圧力(PN)をゼロに初期化
                                変動圧力 -> 静止状態からのずれ(以降、圧力とよぶ)
C          --- SMAC ---
          PD(IX, IY) = 0.0D0
          PN(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 61 IY = 0, NY+1
              DO 62 IX = 0, NX+1
                  TN(IX, IY) = 0.0D0
62          CONTINUE
61          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 70 IY = 0, NY+1..... 仮想セルの温度を設定(式(14.56)を参照)
  TN(NX+1, IY) = 2.0D0 * ( -0.5D0 ) - TN(NX, IY)
70 CONTINUE
*   Tの境界：左面（加熱）T=+0.5
DO 80 IY = 0, NY+1..... 仮想セルの温度を設定(式(14.56)を参照)
  TN(0, IY) = 2.0D0 * ( +0.5D0 ) - TN(1, IY)
80 CONTINUE
*   Tの境界：上面（断熱）
DO 90 IX = 1, NX..... 仮想セルの温度を用いて勾配ゼロを設定
  TN(IX, NY+1) = TN(IX, NY)
90 CONTINUE
*   Tの境界：下面（断熱）
DO 95 IX = 1, NX..... 仮想セルの温度を用いて勾配ゼロを設定
  TN(IX, 0) = TN(IX, 1)
95 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) UN
*   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) VN
*   Pデータファイルからの読み込み[Unit No.=17]（書式なし形式）
  READ (17) PN, PD
  C --- SMAC ---
*-----*
*   （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) TN
  CLOSE (15)
  CLOSE (16)
  CLOSE (17)
  CLOSE (18)
  END IF
  RETURN
  END
*****
*                               時間進行
*****
SUBROUTINE ADV
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
  PARAMETER ( NX0=20, NY0=20, NEO=400 )
  COMMON / D1 / NX, NY
  COMMON / D2 / DX, DY, DT
  COMMON / D3 / VIS, ALP, BUO
  COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
  COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
  COMMON / D6 / DMAX

```

```

COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
C    --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
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 = ', I8)
END IF

*-----
* UN -> UO : 必要なら入れ替える前にUNとUOから変動量を求める
* UN : 前の時間ステップにおいて最終的に得られた値, 圧力補正の度に更新される
* UO : 新しい時間ステップでの初期値. UNを保存.
*-----
DO 70 IX = 0, NX
    DO 80 IY = 0, NY+1
        UO(IX, IY) = UN(IX, IY)
80  CONTINUE
70  CONTINUE

*-----
* VN -> V0 : 必要なら入れ替える前にVNとV0から変動量を求める
* VN : 前の時間ステップにおいて最終的に得られた値, 圧力補正の度に更新される
* V0 : 新しい時間ステップでの初期値, VNを保存.
*-----
DO 90 IX = 0, NX+1
    DO 100 IY = 0, NY
        V0(IX, IY) = VN(IX, IY)
100  CONTINUE
90  CONTINUE

*-----
* TN -> T0 : 必要なら入れ替える前にTNとT0から変動量を求める
* TN : 前の時間ステップでの値
* T0 : 新しい時間ステップでの初期値. TNを保存.
* PN -> P0 : 必要なら入れ替える前にPNとP0から変動量を求める
* PN : 前の時間ステップでの値
* P0 : 新しい時間ステップでの初期値. PNを保存.
*-----
DO 110 IX = 0, NX+1
    DO 120 IY = 0, NY+1
        T0(IX, IY) = TN(IX, IY)
C    --- SMAC ---

```

```

      PO(IX,IY) = PN(IX,IY)
120  CONTINUE
110  CONTINUE
      RETURN
      END

```

* 速度場の計算

```

SUBROUTINE CALVEL
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER ( NX0=20, NY0=20, NEO=400 )
COMMON / D1 / NX,NY
COMMON / D2 / DX,DY,DT
COMMON / D3 / VIS,ALP,BUO
COMMON / D4 / RE,PR,GR,TIME,OMG,EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$                VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$                PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$                TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)

```

C --- SMAC ---

```

COMMON/ARRAY2/DIV(NX0,NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR

```

* 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 *

* 対流項:1次精度風上差分..... 13. 3. 3項, 14. 2節参照 *

* 拡散項:2次精度中心差分..... 13. 1. 1項, 14. 2節参照 *

* U(IX,IY)の計算..... 式(14. 10), (14. 11), (14. 23), (14. 25)

DO 10 IX = 1,NX-1	+-----+-----+-----+
DO 20 IY = 1,NY	P(i, j+1)
* VVはU(IX,IY)におけるVの補間値	.
VV = (VO(IX,IY) + VO(IX+1,IY))	V(i, j)
\$ + VO(IX,IY-1) + VO(IX+1,IY-1)) / 4. 0D0	+-----+-----+-----+
* 対流項(CNVUX,CNVUY)を1次精度風上差分にて計算	P(i-1, j) P(i, j) P(i+1, j)
IF (UO(IX,IY). GE. 0. 0D0) THEN	. → . → .
CNVUX = UO(IX,IY)*(UO(IX,IY) - UO(IX-1,IY)) / DX	U(i-1, j) U(i, j)
ELSE IF (UO(IX,IY). LT. 0. 0D0) THEN	+-----+-----+-----+
CNVUX = UO(IX,IY)*(UO(IX+1,IY) - UO(IX,IY)) / DX	V(i, j-1)
END IF	.
IF (VV. GE. 0. 0D0) THEN	P(i, j-1)
CNVUY = VV*(UO(IX,IY) - UO(IX,IY-1)) / DY	+-----+-----+-----+
ELSE IF (VV. LT. 0. 0D0) THEN	
CNVUY = VV*(UO(IX,IY+1) - UO(IX,IY)) / DY	

```

      END IF
*      x方向の浮力項(BUOU)はゼロ
      TU = 0.0D0
      BUOU = BUO * TU
*      拡散項(DIFU)の計算
      DIFU = VIS*(
$          ( UO(IX-1, IY)-2.0D0*UO(IX, IY)+UO(IX+1, IY) )/DX**2
$          +( UO(IX, IY-1)-2.0D0*UO(IX, IY)+UO(IX, IY+1) )/DY**2
$          )
*      仮の速度(U)の計算
      UN(IX, IY) = UO(IX, IY)
$      + DT*( -CNVUX-CNVUY+DIFU+BUOU+( PO(IX, IY)-PO(IX+1, IY) )/DX )
20  CONTINUE
10  CONTINUE
*-----
*      V(IX, IY)の計算.....式(14.12), (14.13), (14.14), (14.24), (14.26)
*-----
      DO 30 IX = 1, NX
      DO 40 IY = 1, NY-1
*      UUはV(IX, IY)におけるUの補間値
      UU = ( UO(IX-1, IY) + UO(IX, IY)
$          + UO(IX+1, IY) + UO(IX, IY+1) )/4.0D0
*      対流項(CNVVX, CNVVY)を1次精度風上差分にて計算
      IF ( UU. GE. 0.0D0 ) THEN
          CNVVX = UU*( VO(IX, IY) - VO(IX-1, IY) ) / DX
      ELSE IF ( UU. LT. 0.0D0 ) THEN
          CNVVX = UU*( VO(IX+1, IY) - VO(IX, IY) ) / DX
      END IF
      IF ( VO(IX, IY). GE. 0.0D0 ) THEN
          CNVVY = VO(IX, IY)*( VO(IX, IY) - VO(IX, IY-1) ) / DY
      ELSE IF ( VO(IX, IY). LT. 0.0D0 ) THEN
          CNVVY = VO(IX, IY)*( VO(IX, IY+1) - VO(IX, IY) ) / DY
      END IF
*      浮力項(BUOV)の計算
      TV = ( TO(IX, IY) + TO(IX, IY+1) )/2.0D0
      BUOV = BUO*TV
*      拡散項(DIFV)の計算
      DIFV = VIS*(
$          ( VO(IX-1, IY)-2.0D0*VO(IX, IY)+VO(IX+1, IY) )/DX**2
$          +( VO(IX, IY-1)-2.0D0*VO(IX, IY)+VO(IX, IY+1) )/DY**2
$          )
*      仮の速度(V)の計算
      VN(IX, IY) = VO(IX, IY)
$      + DT*( -CNVVX-CNVVY+DIFV+BUOV+( PO(IX, IY)-PO(IX, IY+1) )/DY )
40  CONTINUE
30  CONTINUE
*速度の境界条件の処理
      CALL VELBND

```

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  $A_{[i, j]} PD_{[i]} = B_{[i]}$  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.  $A_{[i, j]}$  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,  $A_{[i, j]}$  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, 圧力補正に関するポアソン方程式において, 2階の
* 微分項は近隣の5点で表せる.
*
* ●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 ----> A1, A2, A3, A4, A5
* 既知ベクトル ----> B
* 未知ベクトル ----> X
* 未知数の数 (未知のPDの数) ----> NX * NY
*
* A1 (NE), A2 (NE), A3 (NE), A4 (NE), A5 (NE)
* i=3, j=3 (k=13) : PD (3, 3) における A1, A2, A3, A4, A5
*
*      +-----+
* (NY) 5 |   |   |   |   |   |
*      +-----+
*      4 |   |   | A4 |   |   |   通し番号の規則
*      +-----+   k=(i-1)*NY+j
*      3 |   | A1 | A3 | A5 |   |   A1 (k)
*      +-----+   A2 (k)
*      2 |   |   | A2 |   |   |   A3 (k)
*      +-----+   A4 (k)
*      1 |   |   |   |   |   |   A5 (k)
*      +-----+   B (k)
*      j  i=1  2   3   4   5 (NX)
*
* B (NE) : 既知ベクトル
*
* X (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)
* A (-NY:NY, NE) : 2次元差分近似による規則的非対称行列
* -NY:NY -> 上の図で考えると K=13 のとき A1 から A5 までの k は
*      A1のkは K"-NY"
*      A2のkは K"-1 "
*      A3のkは K"+0 "

```

```

*           A4のkは K"+1 "           *
*           A5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)           *
*           NE -> 上述のような k は全部で NE 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 (A1, A2, A3, A4, A5, B, X,
$               AGB, W1, W2, W3, W4, W5, W6, W7, W8, W9)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER ( NX0=20, NY0=20, NE0=400 )
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D3 / VIS, ALP, BUO
COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$               VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$               PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$               TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
* --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
* 圧力補正の線形システム用
DIMENSION A1(NEO), A2(NEO), A3(NEO), A4(NEO), A5(NEO), B(NEO), X(NEO)
* 作業用配列
DIMENSION W1(NEO), W2(NEO), W3(NEO), W4(NEO), W5(NEO),
$       W6(NEO), W7(NEO), W8(NEO), W9(NEO)
* バンドガウス消去法のための係数行列配列
DIMENSION AGB(-NY:NY, NEO)
*

```

```

*線形システム関係の配列の初期化

```

```

DO 1 IX = 1, NX
DO 2 IY = 1, NY
K = IY + (IX-1)*NY
B(K) = 0.0D0

```

```

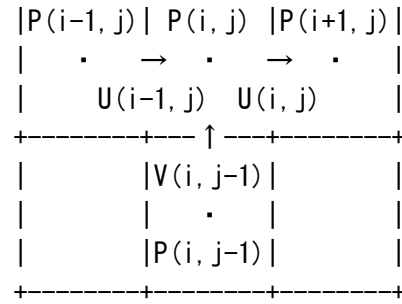
+-----+-----+-----+
|           |P(i, j+1)|           |
|           |   .   |           |
|           |V(i, j)|           |
+-----+-----+-----+
                    ↑

```

```

      A1(K) = 0.0D0
      A2(K) = 0.0D0
      A3(K) = 0.0D0
      A4(K) = 0.0D0
      A5(K) = 0.0D0
      X(K) = 0.0D0
2    CONTINUE
1    CONTINUE
*P(IX, IY)の計算
  DO 10 IX = 1, NX
    DO 20 IY = 1, NY
C      --- SMAC ---
      DIV(IX, IY) = ( UN(IX, IY) - UN(IX-1, IY) )/DX
$      + ( VN(IX, IY) - VN(IX, IY-1) )/DY
20    CONTINUE
10    CONTINUE
*  係数行列の作成.....式(14.39)
*  計算領域の1点さらに内側の点（仮想セルを含めて考えると2点を除く内側の点）
*  に関して係数行列を作成. 残りは境界条件を反映させて PDBNDC で設定する
  DO 11 IX = 2, NX-1
    DO 12 IY = 2, NY-1
      K = IY + (IX-1)*NY
      A3(K) = DT*( 2.0D0/DX**2 + 2.0D0/DY**2 )
      A4(K) = - 1.0D0 / DY**2 * DT
      A2(K) = - 1.0D0 / DY**2 * DT
      A1(K) = - 1.0D0 / DX**2 * DT
      A5(K) = - 1.0D0 / DX**2 * DT
      B(K) = -DIV(IX, IY)
      X(K) = PD(IX, IY)
12    CONTINUE
11    CONTINUE
* --- SMAC --- 境界条件を係数行列に反映させる
  CALL PDBNDC (A1, A2, A3, A4, A5, B)
* --- SMAC --- 圧力補正 P' (PD) に関するポアソン方程式の解法
*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 = NE
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      (A1, A2, A3, A4, A5, B, X, NNX, NNY, NNE,
$                                AGB)
C  2. 反復法1 : point-SOR 法
  IF (METHOD.EQ.2) CALL PSORB   (A1, A2, A3, A4, A5, B, X, 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   (A1, A2, A3, A4, A5, B, X, 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      (A1, A2, A3, A4, A5, B, X, 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   (A1, A2, A3, A4, A5, B, X, 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    (A1, A2, A3, A4, A5, B, X, 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) = X (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) = X (K)-X (1)
      40  CONTINUE
      30  CONTINUE
*圧力補正の境界条件の処理
      CALL PDBND
*速度の修正..... HSMAC法とは異なり, 一度に圧力補正を計算した後,
      DO 50 IX = 1, NX-1      速度補正を行うので, 式 (14. 31-34, 35-38) は以下の
      DO 60 IY = 1, NY      ように計算できる.
      UN (IX, IY) = UN (IX, IY) + ( PD (IX, IY)-PD (IX+1, IY) )/DX*DT
      60  CONTINUE                      ..... 式 (14. 28, 31, 32, 35, 36)
      50  CONTINUE
      DO 70 IX = 1, NX
      DO 80 IY = 1, NY-1
      VN (IX, IY) = VN (IX, IY) + ( PD (IX, IY)-PD (IX, IY+1) )/DY*DT
      80  CONTINUE                      ..... 式 (14. 29, 33, 34, 37, 38)
      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 VELBND
*圧力の修正
      DO 90 IX = 1, NX
      DO 100 IY = 1, NY
      PN (IX, IY) = P0 (IX, IY) + PD (IX, IY)..... 式 (14. 30)
      100  CONTINUE
      90  CONTINUE
*新たに得られた圧力を用いて境界条件を処理する
      CALL PBND
      RETURN
      END
*****
*      温度場の計算..... 式 (14. 15), (14. 16), (14. 27)
*****
      SUBROUTINE CALTEM

```

```

IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER ( NX0=20, NY0=20, NEO=400 )
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D3 / VIS, ALP, BUO
COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)

```

C --- SMAC ---

```

COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR

```

```

*-----*
* 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 *
* 対流項:1次精度風上差分..... 13. 3. 3項, 14. 2節参照 *
* 拡散項:2次精度中心差分..... 13. 1. 1項, 14. 2節参照 *
*-----*

```

*T(IX, IY)の計算

```
DO 10 IX = 1, NX
```

```
DO 20 IY = 1, NY
```

* UUT, VVTはそれぞれT(IX, IY)におけるU, Vの補間値
 $UUT = (UO(IX, IY) + UO(IX-1, IY)) / 2.0D0$
 $VVT = (VO(IX, IY) + VO(IX, IY-1)) / 2.0D0$

* 対流項(CNVTX, CNVTY)を1次精度風上差分にて計算
IF (UUT. GE. 0.0D0) THEN

```
CNVTX = UUT*( TO(IX, IY) - TO(IX-1, IY) ) / DX
```

```
ELSE IF ( UUT. LT. 0.0D0 ) THEN
```

```
CNVTX = UUT*( TO(IX+1, IY) - TO(IX, IY) ) / DX
```

```
END IF
```

```
IF ( VVT. GE. 0.0D0 ) THEN
```

```
CNVTY = VVT*( TO(IX, IY) - TO(IX, IY-1) ) / DY
```

```
ELSE IF ( VVT. LT. 0.0D0 ) THEN
```

```
CNVTY = VVT*( TO(IX, IY+1) - TO(IX, IY) ) / DY
```

```
END IF
```

* 拡散項(DIFT)の計算

```
DIFT = ALP*(
```

```
$      +( TO(IX-1, IY)-2.0D0*TO(IX, IY)+TO(IX+1, IY) )/DX**2
```

```
$      +( TO(IX, IY-1)-2.0D0*TO(IX, IY)+TO(IX, IY+1) )/DY**2
```

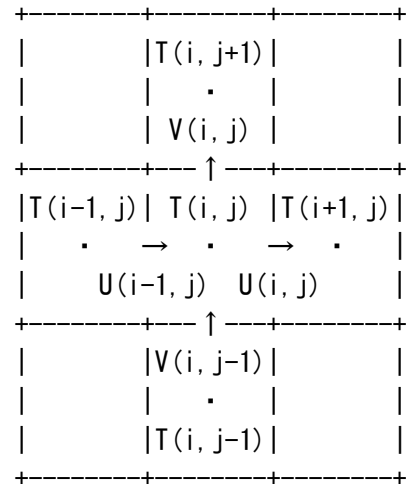
```
$      )
```

* 次の時間のTの計算

```
TN(IX, IY) = TO(IX, IY) + DT*( -CNVTX-CNVTY+DIFT )
```

```
20 CONTINUE
```

```
10 CONTINUE
```



*境界条件の処理

```
CALL TBND
RETURN
END
```

```
*****
```

* 速度の境界条件の処理

```
*****
```

```
SUBROUTINE VELBND
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER (NX0=20, NY0=20, NE0=400)
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D3 / VIS, ALP, BUO
COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$ VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$ PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$ TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
```

```
C --- SMAC ---
```

```
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
```

..... 以降の仮想セルを用いた境界条件(境界で定義できない場合)はいずれも

..... 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)

*U (右面) 境界で定義可能

```
DO 10 IY = 1, NY
UN(NX, IY) = 0.0D0
10 CONTINUE
```

*U (左面) 境界で定義可能

```
DO 20 IY = 1, NY
UN(0, IY) = 0.0D0
20 CONTINUE
```

*U (上面) 境界で定義できないので仮想セルを用いる

```
DO 30 IX = 0, NX
UN(IX, NY+1) = -UN(IX, NY) ..... 式(14.57)
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
UN(IX, 0) = -UN(IX, 1) ..... 式(14.57)
40 CONTINUE
```

*V (右面) 境界で定義できないので仮想セルを用いる

```
DO 50 IY = 1, NY-1
VN(NX+1, IY) = -VN(NX, IY) ..... 式(14.58)
50 CONTINUE
```

*V (左面) 境界で定義できないので仮想セルを用いる

```

      DO 60 IY = 1, NY-1
        VN(0, IY) = -VN(1, IY) ..... 式(14.58)
      60 CONTINUE
*V (上面) ..... 境界で定義可能
      DO 70 IX = 0, NX+1
        VN(IX, NY) = 0.0D0
      70 CONTINUE
*V (下面) ..... 境界で定義可能
      DO 80 IX = 0, NX+1
        VN(IX, 0) = 0.0D0
      80 CONTINUE
      RETURN
      END

```

* 温度の境界条件の処理

```

      SUBROUTINE TBND
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      PARAMETER ( NX0=20, NY0=20, NEO=400 )
      COMMON / D1 / NX, NY
      COMMON / D2 / DX, DY, DT
      COMMON / D3 / VIS, ALP, BUO
      COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
      COMMON / D5 / ICYCLE, ITR, IFLG, IREL, METHOD
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
      COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)
C      --- SMAC ---
      COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
      COMMON / D8 / NE
      COMMON / D9 / NITR
      ..... 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)
*右面
      DO 10 IY = 0, NY+1
        TN(NX+1, IY) = 2.0D0 * ( -0.5D0 ) - TN(NX, IY) ..... 式(14.56)
      10 CONTINUE
*左面
      DO 20 IY = 0, NY+1
        TN(0, IY) = 2.0D0 * ( +0.5D0 ) - TN(1, IY) ..... 式(14.56)
      20 CONTINUE
*上面 ..... 勾配ゼロ
      DO 30 IX = 1, NX
        TN(IX, NY+1) = TN(IX, NY)
      30 CONTINUE

```

*下面.....勾配ゼロ

```

DO 40 IX = 1, NX
    TN(IX, 0) = TN(IX, 1)
40 CONTINUE
RETURN
END

```

```

*
*           圧力補正の係数行列と境界条件
*
*
* 本計算プログラムでは、圧力補正の境界条件を、係数行列作成時に考慮
* する。また、解の1次独立性の確保もここで処理する。
*
*

```

```

SUBROUTINE PDBNDC (A1, A2, A3, A4, A5, B)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER ( NX0=20, NY0=20, NEO=400 )
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D5 / ICYCLE, ITR, IFLG, IREL, METHOD
*
* --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
DIMENSION A1(NEO), A2(NEO), A3(NEO), A4(NEO), A5(NEO), B(NEO)
*
C --- SMAC ---
* 係数行列に境界条件を反映させる
* 計算領域の左側の係数行列 (境界条件を反映)
IX = 1
DO 13 IY = 2, NY-1
    K = IY + (IX-1)*NY
    A3(K) = DT*( 1.0D0/DX**2 + 2.0D0/DY**2 )
    A4(K) = - 1.0D0 / DY**2 * DT
    A2(K) = - 1.0D0 / DY**2 * DT
    A5(K) = - 1.0D0 / DX**2 * DT
    B(K) = -DIV(IX, IY)
13 CONTINUE
* 計算領域の右側の係数行列 (境界条件を反映)
IX = NX
DO 14 IY = 2, NY-1
    K = IY + (IX-1)*NY
    A3(K) = DT*( 1.0D0/DX**2 + 2.0D0/DY**2 )
    A4(K) = - 1.0D0 / DY**2 * DT
    A2(K) = - 1.0D0 / DY**2 * DT
    A1(K) = - 1.0D0 / DX**2 * DT
    B(K) = -DIV(IX, IY)
14 CONTINUE
* 計算領域の下側の係数行列 (境界条件を反映)
IY = 1

```

```

DO 15 IX = 2, NX-1
  K = IY + (IX-1)*NY
  A3(K) = DT*( 2.0D0/DX**2 + 1.0D0/DY**2 )
  A4(K) = - 1.0D0 / DY**2 * DT
  A1(K) = - 1.0D0 / DX**2 * DT
  A5(K) = - 1.0D0 / DX**2 * DT
  B(K) = -DIV(IX, IY)

```

15 CONTINUE

* 計算領域の上側の係数行列（境界条件を反映）
IY = NY

```

DO 16 IX = 2, NX-1
  K = IY + (IX-1)*NY
  A3(K) = DT*( 2.0D0/DX**2 + 1.0D0/DY**2 )
  A2(K) = - 1.0D0 / DY**2 * DT
  A1(K) = - 1.0D0 / DX**2 * DT
  A5(K) = - 1.0D0 / DX**2 * DT
  B(K) = -DIV(IX, IY)

```

16 CONTINUE

* 左下点の係数行列（境界条件を反映）

```

IX = 1
IY = 1
K = IY + (IX-1)*NY
A3(K) = DT*( 1.0D0/DX**2 + 1.0D0/DY**2 )
A4(K) = - 1.0D0 / DY**2 * DT
A5(K) = - 1.0D0 / DX**2 * DT
B(K) = -DIV(IX, IY)

```

* 左上点の係数行列（境界条件を反映）

```

IX = 1
IY = NY
K = IY + (IX-1)*NY
A3(K) = DT*( 1.0D0/DX**2 + 1.0D0/DY**2 )
A2(K) = - 1.0D0 / DY**2 * DT
A5(K) = - 1.0D0 / DX**2 * DT
B(K) = -DIV(IX, IY)

```

* 右下点の係数行列（境界条件を反映）

```

IX = NX
IY = 1
K = IY + (IX-1)*NY
A3(K) = DT*( 1.0D0/DX**2 + 1.0D0/DY**2 )
A4(K) = - 1.0D0 / DY**2 * DT
A1(K) = - 1.0D0 / DX**2 * DT
B(K) = -DIV(IX, IY)

```

* 右上点の係数行列（境界条件を反映）

```

IX = NX
IY = NY
K = IY + (IX-1)*NY
A3(K) = DT*( 1.0D0/DX**2 + 1.0D0/DY**2 )
A2(K) = - 1.0D0 / DY**2 * DT

```

```

      A1(K) = - 1.0D0 / DX**2 * DT
      B(K) = -DIV(IX, IY)
* 1 次独立な解を得るための処理 : IREL=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PN(1, 1)=PD(1, 1)=0とする)
      IF (IREL.EQ.1) THEN
        A3(1) = 1.0D0
        A4(1) = 0.0D0
        A5(1) = 0.0D0
        B(1) = 0.0D0
*      K=2 の点の処理 (K=1とのリンクを断つ)
        A3(2) = DT*( 1.0D0/DX**2 + 2.0D0/DY**2 )
        A4(2) = - 1.0D0 / DY**2 * DT
        A2(2) = 0.0D0
        A5(2) = - 1.0D0 / DX**2 * DT
*      K=1+NY の点の処理 (K=1とのリンクを断つ)
        A3(1+NY) = DT*( 2.0D0/DX**2 + 1.0D0/DY**2 )
        A4(1+NY) = - 1.0D0 / DY**2 * DT
        A1(1+NY) = 0.0D0
        A5(1+NY) = - 1.0D0 / DX**2 * DT
      END IF
*
      RETURN
      END
*****
*
*      圧力補正の境界条件の処理
*
*
* 本計算プログラムでは, 係数行列作成時に境界条件を考慮しているので,
* 実質的には, 最終的に仮想セルの値を配列に格納しているにすぎない.
* このルーチンはなくともよい.
* 反復過程で境界条件を考慮する場合は必須で, 反復のたびにこの処理を
* 行う必要が生じる.
*
*****
      SUBROUTINE PDBND
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER ( NX0=20, NY0=20, NEO=400 )
      COMMON / D1 / NX, NY
      COMMON / D2 / DX, DY, DT
      COMMON / D3 / VIS, ALP, BUO
      COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
      COMMON / D5 / ICYCLE, ITR, IFLG, IREL, METHOD
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
      COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$      VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$      PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$      TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)

```

```

C      --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
..... 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)
*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 10 IY = 1, NY
            PD(NX+1, IY) = PD(NX, IY)
        10 CONTINUE
*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 20 IY = 1, NY
            PD(0, IY) = PD(1, IY)
        20 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 30 IX = 0, NX+1
            PD(IX, NY+1) = PD(IX, NY)
        30 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 40 IX = 0, NX+1
            PD(IX, 0) = PD(IX, 1)
        40 CONTINUE
        RETURN
        END

```

* 圧力の境界条件の処理

```

SUBROUTINE PBNB
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER (NX0=20, NY0=20, NEO=400)
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D3 / VIS, ALP, BUO
COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$                VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$                PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$                TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)

```

```

C      --- SMAC ---
COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
COMMON / D8 / NE
COMMON / D9 / NITR
..... 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)
*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 10 IY = 1, NY
        PN(NX+1, IY) = PN(NX, IY)
10    CONTINUE
*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 20 IY = 1, NY
        PN(0, IY) = PN(1, IY)
20    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 30 IX = 0, NX+1
        PN(IX, NY+1) = PN(IX, NY)
30    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 40 IX = 0, NX+1
        PN(IX, 0) = PN(IX, 1)
40    CONTINUE
      RETURN
      END

```

```

*****

```

```

*                      データ出力

```

```

*****

```

```

      SUBROUTINE PROUT
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      PARAMETER ( NX0=20, NY0=20, NEO=400 )
      COMMON / D1 / NX, NY
      COMMON / D2 / DX, DY, DT
      COMMON / D3 / VIS, ALP, BUO
      COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
      COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
      COMMON / D6 / DMAX
      COMMON / D7 / ITYPE
      COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$                      VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$                      PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$                      TO(0:NX0+1, 0:NY0+1), TN(0:NX0+1, 0:NY0+1)

```

```

C    --- SMAC ---

```

```

      COMMON/ARRAY2/DIV(NX0, NY0), PD(0:NX0+1, 0:NY0+1)
      COMMON / D8 / NE
      COMMON / D9 / NITR
      WRITE (11) UN
      WRITE (12) VN
      WRITE (13) PN, PD
      WRITE (14) TN
      RETURN
      END

```

```

*****

```

```

*                      Tecplot用データ出力

```

```

*****

```

```

      SUBROUTINE TECPLT (FNAME)

```

```

IMPLICIT DOUBLE PRECISION (A-H, O-Z)
PARAMETER ( NX0=20, NY0=20, NEO=400 )
COMMON / D1 / NX, NY
COMMON / D2 / DX, DY, DT
COMMON / D3 / VIS, ALP, BUO
COMMON / D4 / RE, PR, GR, TIME, OMG, EPSP
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
COMMON / D6 / DMAX
COMMON / D7 / ITYPE
COMMON / ARRAY1 / UO(0:NX0, 0:NY0+1), UN(0:NX0, 0:NY0+1),
$          VO(0:NX0+1, 0:NY0), VN(0:NX0+1, 0:NY0),
$          PO(0:NX0+1, 0:NY0+1), PN(0:NX0+1, 0:NY0+1),
$          TO(0:NX0+1, 0:NY0+1), TN(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)
    U = ( UN (IX, IY)+UN (IX, IY+1) )/2.0D0
    V = ( VN (IX, IY)+VN (IX+1, IY) )/2.0D0
    T = ( TN (IX, IY) +TN (IX+1, IY)
$      +TN (IX, IY+1)+TN (IX+1, IY+1) )/4.0D0
    WRITE (21, 4010) X, Y, U, V, T
4010  FORMAT (1H, 5(1PE11.3))
20  CONTINUE
10 CONTINUE
CLOSE (21)
RETURN
END

```

各種データを格子分割図の交わる点
(上の図の四隅の点)における値に変換して出力

.....以降で用いる線形システム解法のための各種サブルーチンは
.....15章のプログラムを利用している。
.....(クリロフ部分空間法を含む)反復法において必要となる収束判定条件は、各々の解法
.....の各反復において、 $\sum (i=1, NE) |X(i)_{NEW} - X(i)_{OLD}| < EPSP$.
.....EPSPはパラメータファイルにて設定する。
.....最大反復回数はNITRで、これもパラメータファイルにて設定する。

```

*          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, . . . . (15. 3. 2), (15. 3. 4) 項を参照
$ 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
        COMMON / D5 / ICYCLE, ITR, IFLG, IREL, METHOD
C      --- SMAC ---
        COMMON / D9 / NITR
*
*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
      IF ( DABS(A(0,NE)).LE.1.0D-50 ) THEN
        WRITE (6,*) ' Matrix singular : |A(0,NE)| < 1E-50 '
        IFLG = 1
      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
            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
*
*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]
*  $(\forall \text{vec}\{x\}^{\text{new}} - \forall \text{vec}\{x\}^{\text{old}})^2 < \text{EPSP}$ 
*  $\forall \text{vec}\{x\}^{\text{old}}$  : 前の反復による値
*  $\forall \text{vec}\{x\}^{\text{new}}$  : 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
      COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
C    --- SMAC ---
      COMMON / D9 / NITR
      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]
*  $(\forall \text{vec}\{x\}^{\text{new}} - \forall \text{vec}\{x\}^{\text{old}})^2 < \text{EPSP}$ 
*  $\forall \text{vec}\{x\}^{\text{old}}$  : 前の反復による値
*  $\forall \text{vec}\{x\}^{\text{new}}$  : 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)
* X0... 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)
*
* [トーマス法のための係数行列]
*   x, y方向に陰的に離散化された結果を以下のように表す.
*   | B(1) C(1) 0   0 ...           | X(1) | | D(1) |
*   | A(2) B(2) C(2) 0 ...           | X(2) | | D(2) |
*   | 0   A(3) B(3) C(3) 0 ...       | X(3) | = | D(3) |
*   |                               | ...  | | ...  |
*   | 0   ...   A(NE-1) B(NE-1) C(NE-1) | X(NE-1) | | D(NE-1) |
*   | 0   0   0 ...   A(NE ) B(NE)   | X(NE)  | | D(NE)  |
*
*****
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
COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
C --- SMAC ---
COMMON / D9 / NITR
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X0に保存
      X0(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X0により最新のXNを緩和
          XN(INY)=(1. 0DO-OMG)*X0(INY)+OMG*X1(INX)
          INX = INX + 1
160  CONTINUE
150 CONTINUE
*      y 軸方向への掃引 : トーマス法による..... 式(15. 34)
      INY = 1
      DO 200 IX = 1, NX
        DO 210 IY = 1, NY
          INX = IX + (IY-1)*NX
*      トーマス法のための係数A, B, C, Dの設定 : XNは最新のX
          A(INY) = AT2(INY)
          B(INY) = AT3(INY)
          C(INY) = AT4(INY)
          D(INY) = BX(INY)
          IF (INY-NY.GE. 1) THEN
            D(INY)=D(INY)-AT1(INY)*XN(INY-NY)

```

```

      END IF
      IF (INY+NY.LE. NE) THEN
        D(INY)=D(INY)-AT5(INY)*XN(INY+NY)
      END IF
*      トーマス法で答えを求める前のXNをX0に保存
      X0(INY) = XN(INY)
      INY = INY + 1
210  CONTINUE
200 CONTINUE
*      Ly=b を解く
      U(1) = C(1) / B(1)
      DO 220 J = 2, NE-1
        U(J) = C(J)/( B(J)-A(J)*U(J-1) )..... 式(15. 28)
220 CONTINUE
      Y(1) = D(1) / B(1)
      DO 230 J = 2, NE
        Y(J) = ( D(J)-A(J)*Y(J-1) ) / ( B(J)-A(J)*U(J-1) )..... 式(15. 29)
230 CONTINUE
*      Ux=y を解く
      X1(NE) = Y(NE)
      DO 240 J = NE-1, 1, -1
        X1(J) = Y(J) - U(J)*X1(J+1)..... 式(15. 30)
240 CONTINUE
      INY = 1
      DO 250 IX = 1, NX
        DO 260 IY = 1, NY
*          得られたX1と反復前のX0により最新のXNを緩和
          XN(INY)=(1. 0DO-OMG)*X0(INY)+OMG*X1(INY)
          INY = INY + 1
260  CONTINUE
250 CONTINUE
*
      RNORM= 0. 0DO
      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
800 CONTINUE
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) :  $p_{\{k+1\}} = r_{\{k+1\}} + \beta_{\{k\}} p_{\{k\}}$ ,  $p_{\{0\}} = r_{\{0\}}$ 
*      AP (NE) :  $A * P$ 
*      AR (NE) :  $A * R$ 
*
* [収束判定条件]
*       $(\forall \text{vec}\{x\}^{\text{new}} - \forall \text{vec}\{x\}^{\text{old}})^2 < \text{EPSP}$ 
*       $\forall \text{vec}\{x\}^{\text{old}}$  : 前の反復による値
*       $\forall \text{vec}\{x\}^{\text{new}}$  : 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
      COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
C      --- SMAC ---
      COMMON / D9 / NITR
      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
*
*   API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 PROV (R, AP, RAP, NE)
*       ( A p_{k}, A p_{k} ) の計算 => APAP
          CALL PROV (AP, AP, APAP, NE)
*        $\alpha_k = (r_k, A p_k) / (A p_k, A p_k)$  ..... 式 (15. 105)
*****
*       ..... 探索方向のための計算が 0除算 なら計算終了(―― SMAC ――)
          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
*        $x_{k+1} = x_k + \alpha_k p_k$  ..... 式 (15. 106)
          X(I) = X(I) + ALP*P(I)
*        $r_{k+1} = r_k - \alpha_k A p_k$  ..... 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)
*       $\beta_{[k]} = - ( A r_{[k+1]}, A p_{[k]} ) / ( A p_{[k]}, A p_{[k]} )$  ..... 式(15.108)
*****
*.....探索方向のための計算が 0除算 なら計算終了(--- SMAC ---)
*      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
*           $p_{[k+1]} = r_{[k+1]} + \beta_{[k]} p_{[k]}$  ..... 式(15.109)
*          P(I) = R(I) + BETA*P(I)
*           $A p_{[k+1]} = A r_{[k+1]} + \beta_{[k]} A p_{[k]}$ 
*          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 PROV (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) :  $t_{\{k\}} = r_{\{k\}} - \alpha_{\{k\}} A p_{\{k\}}$ 
* X(NE) :  $x_{\{k+1\}} = x_{\{k\}} + \alpha_{\{k\}} p_{\{k\}} + \xi_{\{K\}} t_{\{k\}}$ 
* R(NE) :  $r_{\{k+1\}} = t_{\{k\}} - \xi_{\{k\}} A t_{\{k\}}$ 
*           $r_{\{0\}} = B - A x_{\{0\}}$ 
* P(NE) :  $p_{\{k+1\}} = r_{\{k+1\}} + \beta_{\{k\}} (p_{\{k\}} - \xi_{\{k\}} A p_{\{k\}})$ 
*           $p_{\{0\}} = r_{\{0\}}$ 
* AP(NE) :  $A * P$ 
* AR(NE) :  $A * T$ 
*
* [収束判定条件]
*  $(\forall \text{vec}\{x\}^{\text{new}} - \forall \text{vec}\{x\}^{\text{old}})^2 < \text{EPSP}$ 
*  $\forall \text{vec}\{x\}^{\text{old}}$  : 前の反復による値
*  $\forall \text{vec}\{x\}^{\text{new}}$  : 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
      COMMON / D5 / ICYCLE, ITR, IFLG, IRELP, METHOD
C      --- SMAC ---
      COMMON / D9 / NITR
*

```

```

      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 PROV V (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 PROV V (S, AP, SAP, NE)
*       $\alpha_{[k]} = (s, r_{[k]}) / (s, A p_{[k]})$ ..... 式 (15. 112)
*****
*..... 探索方向のための計算が 0除算 なら計算終了(--- SMAC ---)
          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
*       $t_{[k]} = r_{[k]} - \alpha_{[k]} A p_{[k]}$ ..... 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 PROV V (AT, T, ATT, NE)
*      ( A t_{k}, A t_{k} ) の計算 => ATAT
          CALL PROV V (AT, AT, ATAT, NE)
*       $\xi_{[k]} = (A t_{[k]}, t_{[k]}) / (A t_{[k]}, A t_{[k]})$ ..... 式 (15. 114)
*****

```



```

*.....探索方向のための計算が 0除算 なら計算終了(--- SMAC ---)
      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} + \alpha_{k} p_{k} + \xi_{k} t_{k}.....式(15.115)
        X(I) = X(I) + ALPHA*P(I) + XI*T(I)
*       r_{k+1} = t_{k} - \xi_{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)
* \beta_{k} = ( \alpha_{k} / \xi_{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} + \beta_{k} ( p_{k} - \xi_{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
*

```