

第3章 連立1次方程式の解法—ベクトルコンピュータに適した反復法

前章に述べた SOR 法と ADI 法は典型的な反復法であるが，その点過緩和または線過緩和の計算は並列処理することができないので，科学技術計算に広く用いられているベクトルコンピュータには向かない．一方 Jacobi 過緩和法は並列処理可能であるが収束性が良くない．本章には並列処理可能でかつ収束性も良いいくつかの反復法について述べる．本章では読者が前章の内容を理解していることを前提に解説する．

3.1 Chebyshev SOR 法

Chebyshev SOR 法 (red-black point relaxation) は SOR 法をベクトルコンピュータ向きに改良したもので，2次元問題の5点差分方程式，3次元問題の7点差分方程式の境界値問題に適用することができる．ここでは2次元 Poisson 方程式の5点差分方程式

$$c_{ij}^1 u_{i-1,j} + c_{ij}^2 u_{i+1,j} + c_{ij}^3 u_{i,j-1} + c_{ij}^4 u_{i,j+1} + c_{ij}^0 u_{i,j} = f_{i,j} \tag{3.1}$$

の Dirichlet 問題から説明する．計算領域は図 3.1 に示すように長方形で，緩和計算は $1 \leq i \leq i_1$, $1 \leq j \leq j_1$ の内点に対して行われる．内点を $i+j$ が偶数になる点 (red point, 図に \circ 印で示す) と奇数になる点 (black point, \times 印で示す) に分ける．このとき偶数点の緩和はほかの偶数点のデータと無関係に行えるので，偶数点のみを同時に陽的¹に緩和することができる．奇数点についても同様のことが言える．Chebyshev SOR 法はすべての偶数点の点過緩和と，すべての奇数点の点過緩和を交互に解が収束するまで反復するものである．

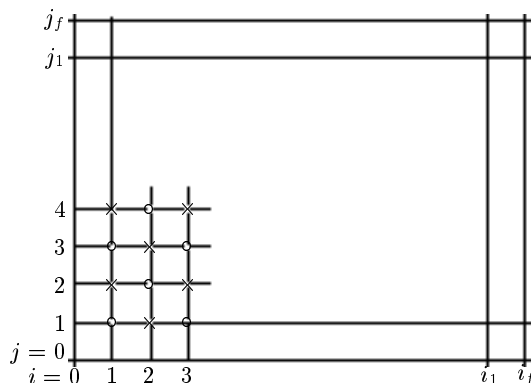


図 3.1: Chebyshev SOR 法の偶数点 \circ と奇数点 \times

¹ 差分方程式を個々に計算するときに陽的 (explicit) に解くといい，その解法を陽解法と言う．また連立方程式として計算するときに陰的 (implicit) に解くといいその解法を陰解法と言う．

以下には，式 (3.1) の Dirichlet 問題を Chebyshev SOR 法で解くプログラムを示す．

```

!*****
! Solution of Dirichlet problem of elliptic pde by red-black point relaxation
!*****
SUBROUTINE REDBLK1(u,c,f,r,i1,if,j1,jf,me,nef,mo,nof,al,na,naf,rmean,e)
DIMENSION u(0:if,0:jf),c(i1,j1,0:4),f(i1,j1),r(i1,j1),me(2,nef),mo(2,nof)
IF(al<=1.)al=1.5; naf=MAX0(naf,100)
e=AMAX1(e,1.E-5)
ne=0; no=0
DO i=1,i1; DO j=1,j1
  IF(MOD(i+j,2)==0)THEN                                !for red points
    ne=ne+1; me(1,ne)=i; me(2,ne)=j                    !input ij in me
  ELSE                                                  !for black points
    no=no+1; mo(1,no)=i; mo(2,no)=j                    !input ij in mo
  ENDIF
ENDDO; ENDDO
na=0; 10 na=na+1
DO n=1,nef; i=me(1,n); j=me(2,n)                      !red points
  r(i,j) = c(i,j,1)*u(i-1,j)+c(i,j,2)*u(i+1,j) &      !compute residuals
          +c(i,j,3)*u(i,j-1)+c(i,j,4)*u(i,j+1) &
          +c(i,j,0)*u(i,j)-f(i,j)
  u(i,j) = u(i,j)-al*r(i,j)/c(i,j,0)                  !point over-relaxation
ENDDO
DO n=1,nof; i=mo(1,n); j=mo(2,n)                      !black points
  r(i,j) = c(i,j,1)*u(i-1,j)+c(i,j,2)*u(i+1,j) &      !compute residuals
          +c(i,j,3)*u(i,j-1)+c(i,j,4)*u(i,j+1) &
          +c(i,j,0)*u(i,j)-f(i,j)
  u(i,j) = u(i,j)-al*r(i,j)/c(i,j,0)                  !point over-relaxation
ENDDO
rmean=0.
DO i=1,i1; DO j=1,j1
  rmean = rmean+ABS(r(i,j))                            !compute mean residual
ENDDO; ENDDO
rmean=rmean/FLOAT(i1*j1)
IF(rmean>e.AND.na<naf)                                GOTO 10          !decide convergence
END

```

このサブルーチンは，配列 u に u_{ij} の予測値と境界値，配列 c, f に式 (3.1) の係数 c_{ij}^k と右辺 f_{ij} の値，引数 al, naf, e に緩和係数 α ，最大反復数 n_f ，収束判定用の小さい正数 ε の値， i_1, if, j_1, jf, nr, nb にも値を入れて引用する．ただし $i_1 = i_f - 1$ ， $j_1 = j_f - 1$ ，また nr と nb は偶数点の数 n_{red} または奇数点の数 n_{black} で， n_{red} は n_{black} と同じか 1 つ多くなる． $n_{red} + n_{black} = i_1 \times j_1$ ．このプログラムでは，まずすべての偶数点またはすべての奇数点を簡単に呼出せるように，配列 mr, mb を用意し，これらの配列にあらかじめ偶数点または奇数点の i, j を入れておく．反復計算は，偶数点の点過緩和，奇数点の点過緩和の順に行われ，平均残差値 $r_{mean} \equiv (\sum |r_{ij}|) / (i_1 j_1) \leq \varepsilon$ または反復回数 $n = n_f$ になるまで繰り返される．計算終了時に，配列 u に関数値 u_{ij} ，配列 r に残差値 r_{ij} ，引数 $na, rmean$ に n と r_{mean} の値が入る．このプログラムは，すべての偶数点またはすべての奇数点の緩和計算が並列に行えるように作られているので，ベクトル化率だけでなく並列度も十分高くなっている．なお念のため書けば，この解法は，パソコンなどのスカラ演算のみを行うコンピュータでは SOR 法よりもかなり劣るもので，ベクトルコンピュータを使用する大規模計算においてはじめて威力を発揮するものである．

問 1 上記問題で，図 3.2 に示すように，右側境界 ($i = i_f$) を Neumann 条件，上下境界 ($j = 0, j_f$) を周期条件に変更した混合境界値問題の Chebyshev SOR 法のプログラムを示せ．

Neumann 条件 $(\partial u / \partial x)_{i_f j} = g_j$ を考慮した差分方程式

$$c_{i_f j}^1 u_{i_f-1, j} + c_{i_f j}^3 u_{i_f, j-1} + c_{i_f j}^4 u_{i_f, j+1} + c_{i_f j}^0 u_{i_f j} = f_{i_f j}$$

と周期条件 $u_{i j_f} = u_{i 0} + \phi$, $(\partial u / \partial y)_{i j_f} = (\partial u / \partial y)_{i 0}$ を考慮した式

$$c_{i j_f}^1 u_{i-1, j_f} + c_{i j_f}^2 u_{i+1, j_f} + c_{i j_f}^3 u_{i, j_f-1} + c_{i j_f}^4 u_{i, j_f+1} + c_{i j_f}^0 u_{i j_f} = f_{i, j_f},$$

$$u_{i j_f} = u_{i 0} + \phi$$

を用いることにする。またここでは、 j_f を偶数に取り、点過緩和の計算を下方の境界点を除外して行うことにする。このとき奇数点と偶数点の数は等しくなる。次のプログラムは上記のサブルーチン REDBLK1 に倣って書かれている。

```

*****
! Solution of mixed bound value problem of elliptic pde by red-black point relaxation
*****
SUBROUTINE REDBLK2(u, c, f, r, if, jf, phi, me, nef, mo, nof, al, na, naf, rmean, e)
DIMENSION u(0:if, 0:jf), c(if, jf, 0:4), f(if, jf), r(if, jf), me(2, nef), mo(2, nof)
IF(al<=1.)al=1.5; naf=MAX0(naf, 1000)
e=AMAX1(e, 1.E-5)
ne=0; no=0
DO i=1, if; DO j=1, jf
  IF(MOD(i+j, 2)==0)THEN                                !for red points
    ne=ne+1; me(1, ne)=i; me(2, ne)=j                  !input ij in me
  ELSE                                                    !for black points
    no=no+1; mo(1, no)=i; mo(2, no)=j                  !input ij in mo
  ENDIF
ENDDO; ENDDO
na=0; 10 na=na+1
DO n=1, nef; i=me(1, n); j=me(2, n)                    !red points
  ip1=MIN0(i+1, if)                                     !Neumann cond
  jp1=j+1; IF(j==jf)jp1=1                               !periodic cond
  r(i, j) = c(i, j, 1)*u(i-1, j)+c(i, j, 2)*u(ip1, j) & !compute residuals
            +c(i, j, 3)*u(i, j-1)+c(i, j, 4)*u(i, jp1) &
            +c(i, j, 0)*u(i, j)-f(i, j)
  u(i, j) = u(i, j)-al*r(i, j)/c(i, j, 0)              !point over-relaxation
ENDDO
DO n=1, nof; i=mo(1, n); j=mo(2, n)                    !black points
  ip1=MIN0(i+1, if)                                     !Neumann cond
  jp1=j+1; IF(j==jf)jp1=1                               !periodic cond
  r(i, j) = c(i, j, 1)*u(i-1, j)+c(i, j, 2)*u(ip1, j) & !compute residuals
            +c(i, j, 3)*u(i, j-1)+c(i, j, 4)*u(i, jp1) &
            +c(i, j, 0)*u(i, j)-f(i, j)
  u(i, j) = u(i, j)-al*r(i, j)/c(i, j, 0)              !point over-relaxation
ENDDO
FORALL(i=1:if)u(i, 0)=u(i, jf)-phi
rmean=0.
DO i=1, if; DO j=1, jf
  rmean = rmean+ABS(r(i, j))                            !compute mean residual
ENDDO; ENDDO
rmean = rmean/FLOAT(if*jf)
IF(rmean>e.AND.na<naf)                                GOTO 10          !decide convergence
END

```

以下には、ポアソン方程式 $u_{xx} + u_{yy} = -2$ の図 3.2 の境界条件の課された 2 つの境界値問題を、出発値 $u = 0$ を与えて解いた結果を示す。計算には格子間隔 0.05 の正方形格子が用いられた。解 u_{ij} は、 $\varepsilon = 10^{-5}$

で小数点以下 3 桁まではほぼ正確に求めることができる。表 3.1 は緩和係数 α と最終反復数 n ，その時点における平均残差値 r_{mean} の関係を示したものである。解は $\alpha = 1.75$ 程度の大き目の緩和係数に対して良く収束する。また計算領域の一辺のみに関数値の与えられる混合境界値問題では，収束までにはかなりの反復を要することが分かる。なおこの混合境界値問題の厳密解は $u(x, y) = -x^2 + 2x + y$ である。

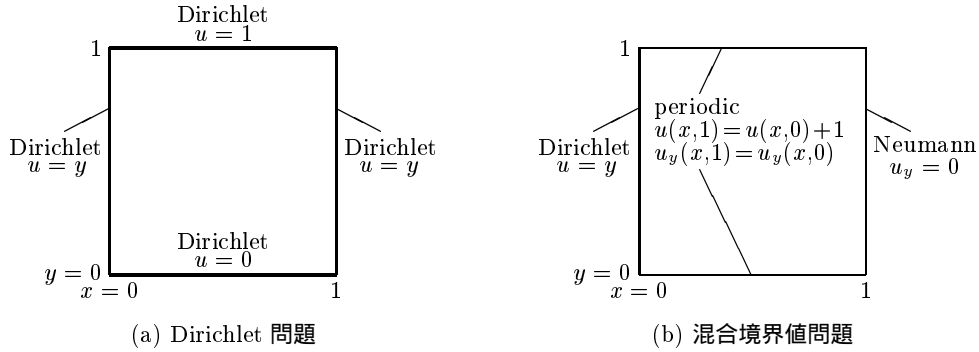


図 3.2: Chebyshev SOR 法の例題

表 3.1: Chebyshev SOR 法の解の収束状況

α	Dirichlet 問題		混合境界値問題	
	n	r_{mean}	n	r_{mean}
1.00	339	10^{-5}	1000	63×10^{-5}
1.25	209	10^{-5}	1000	11×10^{-5}
1.50	118	10^{-5}	850	10^{-5}
1.75	40	10^{-5}	385	10^{-5}
1.80	50	10^{-5}	300	10^{-5}

Chebyshev SOR 法は，3 次元の 7 点差分方程式にそのまま拡張することができる。すなわち $i+j+k$ が偶数の点と奇数の点に分ければ，各々の点過緩和の計算を並列処理することができる。図 3.3(a) に示す 2 次元の 9 点高次差分方程式の場合には， $\text{mod}(i+j, 3)$ が 0 の点，1 の点，2 の点に分ければ，点過緩和の計算を並列処理することができる。また (b) 図の 9 点高次差分方程式の場合には，表に示すように 4 色の点に分ければ，各色ごとの点過緩和を並列処理することができる。

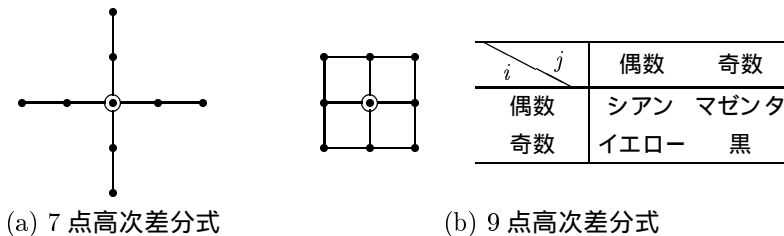


図 3.3: 2 次元高次差分演算子

3.2 Chebyshev SLOR法

Chebyshev SLOR法 (zebra line overrelaxation) は線過緩和法 (SLOR iterative scheme) をベクトルコンピュータ向きに改良したものである。5点差分方程式 (3.1) の場合には、偶数列の線緩和は互いに独立に行えるので、すべての偶数列を並べて同時に線緩和することができる。奇数列についても同様のことが言える。Chebyshev SLOR法は偶数列の線過緩和と奇数列の線過緩和の並列処理を交互に解が収束するまで反復するものである。線過緩和については2.4節のADI法の項を参照されたい。

次に、Chebyshev SLOR法のプログラムの簡単な例として、前節に示したポアソン方程式のDirichlet問題を解くサブルーチンを示す。

```

!*****
! Solution of Dirichlet problem of elliptic pde by zebra line overrelaxation
!*****
SUBROUTINE ZEBRA1(u,c,f,r,i1,if,j1,jf,a,b,n,ie,io,bt,na,naf,rmean,e)
DIMENSION u(0:if,0:jf),c(0:if,0:jf,0:4),f(0:if,0:jf),r(0:if,0:jf),a(io,n,3),b(io,n)
IF(bt<.75)bt=.8; naf=MAX0(naf,1000)
e=AMAX1(e,1.E-5)
na=0; 10 na=na+1
DO ii=1,ie; i=2*ii; DO j=0,jf           !even columns
  a(ii,j+1,1) = c(i,j,3)                !setup coef matrix
  a(ii,j+1,2) = c(i,j,0)*bt
  a(ii,j+1,3) = c(i,j,4)
  b(ii,j+1) = f(i,j)-c(i,j,1)*u(i-1,j)-c(i,j,0)*u(i,j) & !setup rhs
             *(1.-bt)-c(i,j,2)*u(i+1,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,io,n,ie)                !solve sets of linear eqns
DO ii=1,ie; i=2*ii; DO j=0,jf
  u(i,j) = b(ii,j+1)
ENDDO; ENDDO
DO ii=1,io; i=2*ii-1; DO j=0,jf         !odd columns
  a(ii,j+1,1) = c(i,j,3)                !setup coef matrix
  a(ii,j+1,2) = c(i,j,0)*bt
  a(ii,j+1,3) = c(i,j,4)
  b(ii,j+1) = f(i,j)-c(i,j,1)*u(i-1,j)-c(i,j,0)*u(i,j) & !setup rhs
             *(1.-bt)-c(i,j,2)*u(i+1,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,io,n,io)                !solve sets of linear eqns
DO ii=1,io; i=2*ii-1; DO j=0,jf
  u(i,j) = b(ii,j+1)
ENDDO; ENDDO
rmean=0.
DO i=1,i1; DO j=1,j1
  r(i,j) = c(i,j,1)*u(i-1,j)+c(i,j,2)*u(i+1,j) & !compute residuals
          +c(i,j,3)*u(i,j-1)+c(i,j,4)*u(i,j+1) &
          +c(i,j,0)*u(i,j)-f(i,j)
  rmean = rmean+ABS(r(i,j))/FLOAT(i1*j1) !compute mean residual
ENDDO; ENDDO
IF(rmean>e.AND.na<naf)                   GOTO 10 !decide convergence
END

!***** Subroutine GAUSSZ *****
! Solution of sets of linear eqns with tri-diagonal matrix by Gaussian elimination
SUBROUTINE GAUSSZ(a,b,m,n,if)
DIMENSION a(m,n,3),b(m,n)
DO k=1,n-1; DO i=1,if

```

```

b(i,k)=b(i,k)/a(i,k,2); a(i,k,3)=a(i,k,3)/a(i,k,2)
b(i,k+1)=b(i,k+1)-a(i,k+1,1)*b(i,k)
a(i,k+1,2)=a(i,k+1,2)-a(i,k+1,1)*a(i,k,3)
ENDDO; ENDDO
FORALL (i=1:if) b(i,n)=b(i,n)/a(i,n,2)
DO k=n-1,1,-1; DO i=1,if
  b(i,k)=b(i,k)-a(i,k,3)*b(i,k+1)
ENDDO; ENDDO
END

```

このプログラムは偶数列の線過緩和，奇数列の線過緩和，平均残差値による収束判定を繰り返し実行するものである。このプログラムでは， $i_1 = i_f - 1$ ， $j_1 = j_f - 1$ ， i_e, i_o はそれぞれ偶数列の数 i_{even} または奇数列の数 i_{odd} で， $i_{\text{even}} + i_{\text{odd}} = i_1$ である。この場合の i_{odd} は i_{even} と同数か 1 つ多くなる。a と b は i_{odd} 個の 3 重対角行列の連立 1 次方程式の係数行列と右边を記憶する配列で，n は各連立 1 次方程式の未知変数の数 $n = j_f + 1$ である。偶数列の計算では， i_{even} 個の n 元連立 1 次方程式 (2.23a) の係数と右边を作業用配列 a, b に並べて入力し，サブルーチン GAUSSZ を用いて並べて解く。奇数列の計算も同様である。GAUSSZ は，1.4 節に述べた 1 つの 3 重対角行列の連立 1 次方程式を解くサブルーチン GAUSS3 を，if 個の方程式を並べて解くように拡張したものである。並列度を上げるためには，i に関する DO 文を元のサブルーチンの DO 文の内側に入れなければならない。

問 2 2.4 節の混合境界値問題の解を Chebyshev SLOR 法で求めよ。

この混合境界値問題のプログラムを上記サブルーチン ZEBRA1 に倣って作ったものを以下に示す。

```

!*****
! Sol of mixed bound value problem of elliptic pde by zebra line overrelaxation
!*****
SUBROUTINE ZEBRA2(u,c,f,r,i0,i1,if,j1,jf,a,b,m,n,bt,na,naf,rmean,e)
REAL*4 u(0:if,0:jf),c(0:if,0:jf,0:4),f(0:if,0:jf),r(0:if,0:jf),a(m,n,3),b(m,n)
IF(bt<.75)bt=.8; naf=MAX0(naf,1000)
e=AMAX1(e,1.E-5)
ie0=(i0+1)/2; io0=i0-ie0; ie=if/2+1; io=if+1-ie
je=jf/2+1; jo=jf+1-je
na=0; 10 na=na+1
DO ii=2,ie; i=2*ii-2; ip1=i+1; DO j=0,jf
  IF(i==if)ip1=if-2
  a(ii,j+1,1) = c(i,j,3)
  a(ii,j+1,2) = c(i,j,0)*bt
  a(ii,j+1,3) = c(i,j,4)
  b(ii,j+1) = f(i,j)-c(i,j,1)*u(i-1,j)-c(i,j,0)*u(i,j) &
    *(1.-bt)-c(i,j,2)*u(ip1,j)
ENDDO; ENDDO
CALL GAUSSMZ(a,b,m,n,2,ie0,jf+1,3,3)
CALL GAUSSMZ(a,b,m,n,ie0+1,ie,jf+1,1,1)
DO ii=2,ie; i=2*ii-2; DO j=0,jf
  u(i,j) = b(ii,j+1)
ENDDO; ENDDO
DO ii=1,io; i=2*ii-1; ip1=i+1; DO j=0,jf
  IF(i==if)ip1=if-2
  a(ii,j+1,1) = c(i,j,3)
  a(ii,j+1,2) = c(i,j,0)*bt
  a(ii,j+1,3) = c(i,j,4)
  b(ii,j+1) = f(i,j)-c(i,j,1)*u(i-1,j)-c(i,j,0)*u(i,j) &
    *(1.-bt)-c(i,j,2)*u(ip1,j)
ENDDO; ENDDO

```

```

CALL GAUSSMZ(a,b,m,n,1,io0,jf+1,3,3)           !solve sets of linear eqns
CALL GAUSSMZ(a,b,m,n,io0+1,io,jf+1,1,1)
DO ii=1,io; i=2*ii-1; DO j=0,jf
  u(i,j) = b(ii,j+1)
ENDDO; ENDDO
                                           GOTO 11

na=na+1
DO jj=1,je; j=2*(jj-1); jm1=j-1; jp1=j+1; DO i=0,if           !even rows
  IF(i<i0)THEN; IF(j==0)jm1=jf-1; IF(j==jf)jp1=1           !ib1=ibn=3
  ELSE;           IF(j==0)jm1=0; IF(j==jf)jp1=jf           !ib1,ibn=1
  ENDF
  a(jj,i+1,1) = c(i,j,1)           !setup coef matrices
  a(jj,i+1,2) = c(i,j,0)*bt
  a(jj,i+1,3) = c(i,j,2)
  b(jj,i+1) = f(i,j)-c(i,j,3)*u(i,jm1)-c(i,j,0)*u(i,j) &           !setup rhs
              *(1.-bt)-c(i,j,4)*u(i,jp1)
ENDDO; ENDDO
CALL GAUSSMZ(a,b,m,n,1,je,if+1,1,2)           !solve sets of linear eqns
DO jj=1,je; j=2*(jj-1)
  FORALL(i=0:if)u(i,j)=b(jj,i+1)
ENDDO
DO jj=1,jo; j=2*jj-1; DO i=0,if           !odd rows
  a(jj,i+1,1) = c(i,j,1)           !setup coef matrices
  a(jj,i+1,2) = c(i,j,0)*bt
  a(jj,i+1,3) = c(i,j,2)
  b(jj,i+1) = f(i,j)-c(i,j,3)*u(i,j-1)-c(i,j,0)*u(i,j) &           !setup rhs
              *(1.-bt)-c(i,j,4)*u(i,j+1)
ENDDO; ENDDO
CALL GAUSSMZ(a,b,m,n,1,jo,if+1,1,2)           !solve sets of linear eqns
DO jj=1,jo; j=2*jj-1
  FORALL(i=0:if)u(i,j)=b(jj,i+1)
ENDDO
11 rmean=0.
DO i=1,if-1; im1=i-1; ip1=i+1; DO j=0,jf; jm1=j-1; jp1=j+1
  IF(i<i0)THEN; IF(j==0)jm1=jf-1; IF(j==jf)jp1=1           !ib1=ibn=3
  ELSE;           IF(j==0)jm1=0; IF(j==jf)jp1=jf           !ib1,ibn=1
  ENDF
  IF(i==if.AND.j>0.AND.j<jf)ip1=if-2           !ibn=2
  r(i,j) = c(i,j,1)*u(im1,j)+c(i,j,2)*u(ip1,j) &           !compute residuals
          +c(i,j,3)*u(i,jm1)+c(i,j,4)*u(i,jp1) &
          +c(i,j,0)*u(i,j)-f(i,j)
  rmean=rmean+ABS(r(i,j))/FLOAT(if*jf)           !compute mean residual
ENDDO; ENDDO
IF(rmean>e.AND.na<naf)           GOTO 10           !decide convergence
END

```

!***** Subroutine GAUSSMZ *****

! Sol of sets of linear eqns with modified tri-diagonal matrix based on Gaussian elimination

SUBROUTINE GAUSSMZ(a,b,m1,n1,is,ie,n,ib1,ibn)

REAL*4 a(m1,n1,3),b(m1,n1)

IF(ib1==2)THEN !ib1=2

FORALL(i=is:ie)a(i,1,1)=a(i,1,1)/a(i,1,2)

FORALL(i=is:ie)a(i,2,3)=a(i,2,3)-a(i,2,1)*a(i,1,1)

ENDIF

DO k=1,n-1

IF(ib1==3.AND.k==n-2) & !ib1=ibn=3

FORALL(i=is:ie)a(i,k,3)=a(i,k,3)+a(i,k,1)

```

FORALL (i=is:ie)
  b(i,k)=b(i,k)/a(i,k,2); a(i,k,3)=a(i,k,3)/a(i,k,2)
  b(i,k+1)=b(i,k+1)-a(i,k+1,1)*b(i,k)
  a(i,k+1,2)=a(i,k+1,2)-a(i,k+1,1)*a(i,k,3)
ENDFORALL
IF (ibn==2 .AND. k==n-2) THEN
  ! ibn=2
  FORALL (i=is:ie) b(i,n)=b(i,n)-a(i,n,3)*b(i,k)
  FORALL (i=is:ie) a(i,n,1)=a(i,n,1)-a(i,n,3)*a(i,k,3)
ENDIF
IF (ib1==3) THEN
  ! ib1=ibn=3
  IF (k<n-2) THEN
    FORALL (i=is:ie) a(i,k,1)=a(i,k,1)/a(i,k,2)
    FORALL (i=is:ie) a(i,k+1,1)=-a(i,k+1,1)*a(i,k,1)
  ENDIF
  IF (k>1 .AND. k<n-2) THEN
    FORALL (i=is:ie) b(i,n)=b(i,n)-a(i,n,3)*b(i,k)
    FORALL (i=is:ie) a(i,n,1)=a(i,n,1)-a(i,n,3)*a(i,k,1)
    FORALL (i=is:ie) a(i,n,3)=-a(i,n,3)*a(i,k,3)
  ELSEIF (k==n-2) THEN
    FORALL (i=is:ie) b(i,n)=b(i,n)-a(i,n,3)*b(i,k)
    FORALL (i=is:ie) a(i,n,1)=a(i,n,1)-a(i,n,3)*a(i,k,3)
  ENDIF
ENDIF
ENDDO
FORALL (i=is:ie) b(i,n)=b(i,n)/a(i,n,2)
DO k=n-1,1,-1
  FORALL (i=is:ie) b(i,k)=b(i,k)-a(i,k,3)*b(i,k+1)
  IF (ib1==3 .AND. k<n-2) &
    FORALL (i=is:ie) b(i,k)=b(i,k)-a(i,k,1)*b(i,n-1)
  ! ib1=ibn=3
ENDDO
IF (ib1==2) FORALL (i=is:ie) b(i,1)=b(i,1)-a(i,1,1)*b(i,3)
! ib1=2
END

```

このプログラムでは、偶数 i 列の線過緩和、奇数 i 列の線過緩和、偶数 j 行の線過緩和、奇数 j 行の線過緩和、平均残差値による収束判定が繰り返される。 i 列の線過緩和では、連立 1 次方程式は周期条件の課されている部分と Dirichlet 条件の課されている部分に分けて解かれる。また偶数 j 行の線過緩和でもこれらの境界条件を考慮するなど必要である。偶数列または奇数列の連立 1 次方程式はサブルーチン GAUSSMZ を用い並列処理される。GAUSSMZ は 2.4 節のサブルーチン GAUSSM を並列処理できるように書換えたもので、ここに出てくるすべての境界条件に対応できる。その引数 $ie0, io0, ie, io, je, jo$ は周期条件の課されている区間の偶数列の数 $i_{\text{even}0}$ 、奇数列の数 $i_{\text{odd}0}$ 、全緩和区間の偶数列の数 i_{even} 、奇数列の数 i_{odd} 、偶数行の数 j_{even} 、奇数行の数 j_{odd} である。また $m1, n1$ は同時に解く連立 1 次方程式の最大数、その未知数の最大数、 is, ie は同時に線緩和する初めと終わりの列番号または行番号、 $ib1, ibn$ については 2.4 節参照、メインプログラムについても 2.4 節参照。

以下には、ポアソン方程式 $u_{xx} + u_{yy} = -2$ の Dirichlet 問題とラプラス方程式 $u_{xx} + u_{yy} = 0$ の混合境界値問題を、図 3.4 の境界条件のもとで出発値 $u = 0$ を与えて解いた結果を示す。計算には格子間隔 0.05 の正方形格子を用いた。次表は $\varepsilon = 10^{-5}$ における係数 β と最終反復数 n の関係を示したものである。なお収束しないときには $n_f = 1000$ における平均残差値 r_{mean} が示してある。i-column は i 列の線過緩和のみの場合、j-row は j 行の線過緩和のみの場合である。 i 列の線過緩和と j 行の線過緩和を交互に行う ADI の場合は、予想に反し反復数は多くなった。括弧内の数字は i 列と j 行の線過緩和を 4 回ずつ交互に行った場合で反復数は少なくなる。この表から i 列の線過緩和または j 行の線過緩和のどちらかで十分なことが分かる。

Chebyshev SLOR 法は、図 3.3(b) に示す 2 次元の 9 点高次差分方程式に対してはほとんどそのまま適用

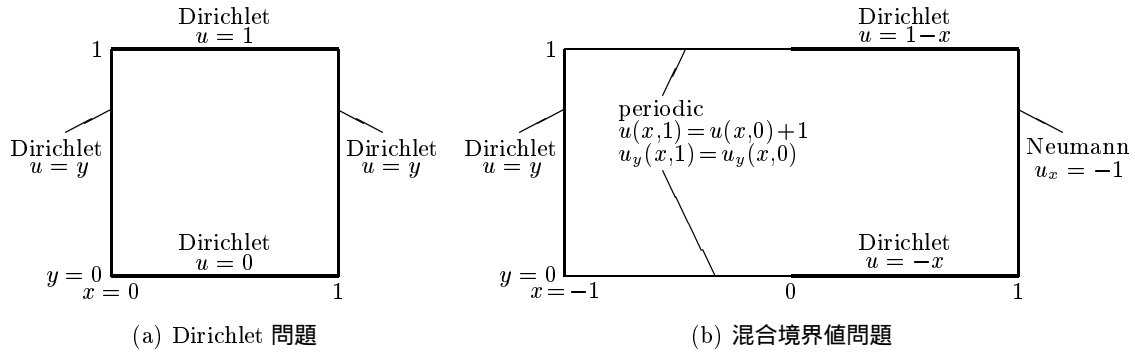


図 3.4: Chebyshev SLOR 法の例題

表 3.2: Chebyshev SLOR 法の解の収束状況

β	Dirichlet 問題	混合境界値問題		
	i-column	i-column	j-row	ADI
0.75	391	41×10^{-5}	diverge	260 (104)
0.80	24	99	100	306 (152)
1.00	157	426	430	536 (464)
1.25	313	853	856	926 (888)
1.50	470	4×10^{-5}	4×10^{-5}	5×10^{-5} (5×10^{-5})

することができる。(a) 図の 9 点高次差分方程式の場合には, $\text{mod}(i, 3)$ が 0 の列, 1 の列, 2 の列に分け線緩和を並列処理するように書き換えることが必要である。3 次元の 7 点差分方程式の場合には, xy 面内の Chebyshev SLOR 法によるの計算を z 方向に掃引することも考えられるが, Chebyshev SLOR 法と Chebyshev SOR 法を組み合わせ, z 方向に並ぶ点列を $i+j$ が偶数のものと奇数のものに分け線緩和を並列に行えばいっそう高並列度になる。

3.3 多重格子法

多重格子 (multigrid, MG) 法は, スカラーコンピュータの時代に, 楕円型偏微分方程式の境界値問題の効率的な数値解法として考案されたもので, その後多くの問題に適用できるように拡張されてきた. 多重格子法は, 格子間隔 h の基本格子のほかに, それに重なる格子間隔 $2h, 4h, \dots$ の格子を用い緩和計算を行うものである. 前述のように, 短波長の残差の波は容易に緩和できるが, 長波長のは容易でない. 多重格子法は, 異なる寸法の格子を用い, すべての波長の残差を効率よく緩和しようとするものである. 多重格子法では, 格子間隔 h の細かい格子の残差は限定集計 (restriction) によって 1 つずつ粗い格子のものに移され, 粗い格子の修正値は延長補間 (prolongation) によって 1 つずつ細かい格子のものに移される. また点緩和は適宜各格子で行われるが, 過緩和の必要はなく専ら Gauss 反復法が用いられる. 最も粗い格子では直接法も用いられる. 限定集計, 延長補間, Gauss 反復法はすべて並列処理可能で多重格子法はベクトルコンピュータにも適する計算技法である.

3.3.1 楕円型線形微分方程式の多重格子法

始めに Poisson 方程式 $u_{xx} + u_{yy} = -2$ に対して多重格子法を説明する. 格子間隔 h の正方形格子の場合には, 差分方程式は

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij} = -2h^2$$

となり, その残差の式は

$$r_{ij} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij} + 2h^2$$

となる. 今出発値 $u_{ij} = 0$ を与えて計算を始めることにすればその初期残差は $r_{ij} = 2h^2$ となる. また格子間隔 $H = 2h$ の粗い格子に対しては初期残差は $r_{IJ} = 2H^2 = 8h^2$ となる. これより初期残差値は, 格子セルの面積に比例することが分かる. 格子間隔 $\Delta x = h, \Delta y = h/\lambda$ の長方形格子の場合には, 差分方程式は

$$u_{i-1,j} - 2u_{ij} + u_{i+1,j} + \lambda^2(u_{i,j-1} - 2u_{ij} + u_{i,j+1}) = -2h^2 \quad (3.2)$$

また残差の式は

$$r_{ij} = u_{i-1,j} - 2u_{ij} + u_{i+1,j} + \lambda^2(u_{i,j-1} - 2u_{ij} + u_{i,j+1}) + 2h^2 \quad (3.3)$$

となる. 多重格子法では, まず既知の h 格子の残差値 r_{ij} に対し H 格子の残差値 r_{IJ} を見積もることが必要である.

上記の初期残差の例では, 残差値が計算領域を通して一様で, また計算領域内の残差値を集計したものは格子の大きさによらず一定であった. この性質を一般化すれば h 格子と H 格子の残差の関係は次のようになる. 図 3.5(a) は多重格子を示したもので, h 格子の格子点を \bullet 印, H 格子の格子点を \circ 印で, また h 格子の格子点番号を ij , H 格子のものを IJ で示してある. 計算領域内の残差の集計値が一定になるように, 残差値 r_{ij} はすべて r_{IJ} に, $r_{i+1,j}$ は 2 分して r_{IJ} と $r_{I+1,J}$ に, $r_{i+1,j+1}$ は 4 分して r_{IJ} , $r_{I+1,J}$, $r_{I+1,J+1}$, $r_{I,J+1}$ に集計することにする. このとき, 残差値 r_{IJ} は図に \bullet 印を付けた格子点の残差値 r_{ij}, \dots を太線枠内の面積を勘案して集計したものになる. このような操作を限定集計 (restriction) と呼び, 式で表せば次のようになる.

$$r_{IJ} = r_{ij} + \frac{1}{2}(r_{i-1,j} + r_{i+1,j} + r_{i,j-1} + r_{i,j+1}) + \frac{1}{4}(r_{i-1,j-1} + r_{i+1,j-1} + r_{i-1,j+1} + r_{i+1,j+1}) \quad (3.4)$$

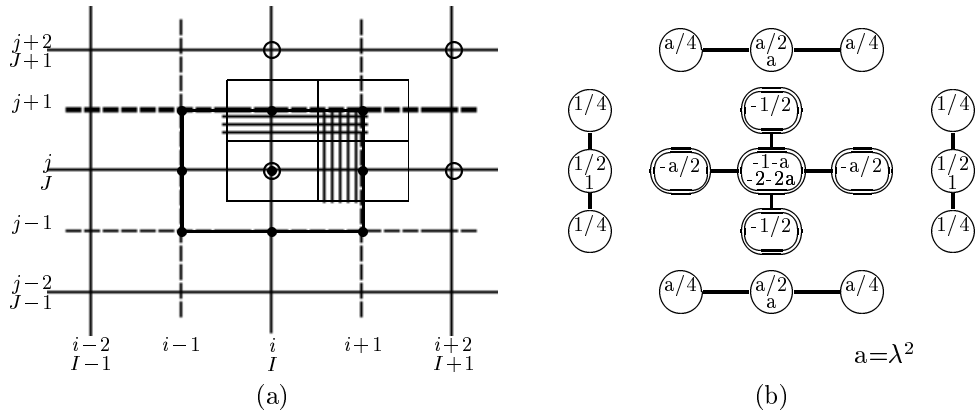


図 3.5: 長方形格子の限定集計

式 (3.3) を限定集計すれば、左辺はかなり長い式になるが、その係数を整理したものは (b) 図のようになる。すなわち、この図の位置関係は (a) 図と同じで、図中の円またはオーバル内の上段の数字になる。また下段の数字はこの係数を H 格子の格子点に集めたものである。これより h 格子のラプラス演算子を限定集計したものは、当然のことながら、 H 格子のラプラス演算子で近似できることが分かる。結局 式 (3.3) を限定集計した H 格子の式は次のようになる²。

$$r_{IJ} = u_{I-1,J} - 2u_{IJ} + u_{I+1,J} + \lambda^2(u_{I,J-1} - 2u_{IJ} + u_{I,J+1}) + 2H^2 \quad (3.5)$$

多重格子法の計算の途中の段階を考え、その u の近似値を u_{ij} 、また解を $u_{ij} + d_{ij}$ とする。この d_{ij} は理想的な修正値で、すべての残差を 0 にするものである。したがって

$$0 = (u_{i-1,j} + d_{i-1,j}) - 2(u_{ij} + d_{ij}) + (u_{i+1,j} + d_{i+1,j}) + \lambda^2 \{ (u_{i,j-1} + d_{i,j-1}) - 2(u_{ij} + d_{ij}) + (u_{i,j+1} + d_{i,j+1}) \} + 2h^2$$

この式に残差の式 (3.3) を用いれば次の h 格子の修正値 d_{ij} の差分方程式が導かれる。

$$d_{i-1,j} - 2d_{ij} + d_{i+1,j} + \lambda^2(d_{i,j-1} - 2d_{ij} + d_{i,j+1}) + 2h^2 = -r_{ij} \quad (3.6)$$

この式を限定集計すれば、次の H 格子の修正値 d_{IJ} の差分方程式が導かれる。

$$d_{I-1,J} - 2d_{IJ} + d_{I+1,J} + \lambda^2(d_{I,J-1} - 2d_{IJ} + d_{I,J+1}) + 2h^2 = -r_{IJ} \quad (3.7)$$

通常この差分方程式はガウス反復法で解かれ H 格子の u_{IJ} の修正値 d_{IJ} が求められる。この修正値は h 格子の修正値 d_{ij} にそのまままたは Lagrange 補間によって移される。

$$d_{ij} = d_{IJ}, \quad d_{ij} = (-d_{i,j-3} + 9d_{i,j-1} + 9d_{i,j+1} - d_{i,j+3})/16, \quad (3.8)$$

$$d_{i2} = (3d_{i1} + 6d_{i3} - d_{i5})/8$$

²なお差分方程式と残差の式が

$$\frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\Delta y^2} = -2$$

$$r_{ij} = \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\Delta y^2} + 2$$

のようにしている場合には、残差の限定集計ではなく重み平均を取るべきである。

多重格子法では H 格子の修正値からこのようにして h 格子の修正値を求める操作を延長補間 (prolongation) という。多重格子法では、各格子レベルで緩和計算を行い、格子サイズに合った残差の波の成分を緩和することになる。小サイズの格子の修正値は、その格子で求めた修正値にそれよりも大きいサイズの格子で求めた修正値を延長補間したものすべてを合算したものになる。

以上の説明をもとに、次に多重格子法を一般的に説明する。線形微分方程式は

$$L(u) = f \quad (3.9)$$

またその近似差分方程式は

$$L^h(u^h) = f^h \quad (3.10)$$

のように書くことができる。上添え字 h は格子間隔を意味するものとする。その第 n 近似解を $u^{h(n)}$ とすれば、式 (3.10) の残差は

$$r^{h(n)} = L^h(u^{h(n)}) - f^h \quad (3.11)$$

となる。今式 (3.10) の残差を零にする理想的な $u^{h(n)}$ の修正値を d^h とする。すなわち $L^h(u^{h(n)} + d^h) = f^h$ 。 L^h は線形演算子であるから、この式と式 (3.11) から、次の細かい格子の修正値 d^h の差分方程式が得られる。

$$L^h(d^h) = -r^{h(n)} \quad (3.12)$$

ここで格子間隔 $H = 2h$ の粗い格子を考え、式 (3.12) の限定集計を取れば次式が得られる。

$$L^H(d^H) = -r^H \quad (3.13)$$

ただし

$$r^H = I_h^H r^h \quad (3.14)$$

I_h^H は限定集計演算子 (restriction operator) と呼ばれるものである。式 (3.13) は H 格子の修正値 d^H の差分方程式で、 d^H の値は緩和計算によって求められる。修正値 d^H から h 格子の修正値 d^h は延長補間によって求められる。

$$d^h = I_H^h d^H \quad (3.15)$$

ただし I_H^h は延長補間演算子 (prolongation operator) と呼ばれるものである。 u の第 $n+1$ 近似値は

$$u^{h(n+1)} = u^{h(n)} + \tilde{d}^h \quad (3.16)$$

となる。ただし \tilde{d}^h は、 h 格子の緩和計算で求めた修正値に、 $2h, 4h, \dots$ 格子の緩和計算で求めた修正値を延長補間したものを加えた修正値である。

多重格子法の計算の手順、すなわち残差の計算、限定集計、修正値の計算、延長補間、修正値の加算をどのような順番で反復するのは各種のものが提案されている。その最適手順は解くべき問題、微分方程式と境界条件、格子や数値解法によるものと思われる。図 3.6 に 2 つの 1 サイクルの計算手順を示す。この図の説明は次項にする。

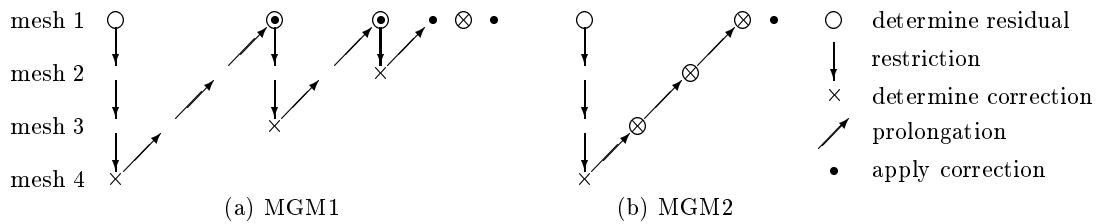


図 3.6: 多重格子法の計算手順

3.3.2 Poisson 方程式の Dirichlet 問題の多重格子法のプログラム

ここでは多重格子法のプログラムの一例を示す．次の Poisson 方程式の Dirichlet 問題

$$u_{xx} + u_{yy} = -20 \quad (0 < x < 2, 0 < y < 1.2) \quad (3.17a)$$

$$u(0, y) = u(2, y) = u(x, 0) = u(x, 1.2) = 0 \quad (3.17b)$$

を考える．計算領域は長方形で，計算格子は4重の正方形格子でそれぞれを mesh 1, mesh 2, mesh 3, mesh 4 と呼ぶことにする．mesh 1 は格子間隔 $h = 0.05$ の基本格子である．プログラム中の記号の意味はだいたい分かると思うが，各格子の残差と修正値は次のように定義する．

$$\text{mesh 1: } (41 \times 25) \quad R_1 = r^1, \quad D_1 = d^1$$

$$\text{mesh 2: } (21 \times 13) \quad R_2 = r^2, \quad D_2 = d^2$$

$$\text{mesh 3: } (11 \times 7) \quad R_3 = r^3, \quad D_3 = d^3$$

$$\text{mesh 4: } (6 \times 4) \quad R_4 = r^4, \quad D_4 = d^4$$

また a と b は，修正値 d^4 を Gauss 消去法で計算するために連立1次方程式の係数行列と右辺を入れる配列である．この問題は対称であるが，ここでは全領域を計算することにする．サブルーチン MGM1 は図 3.6(a) に示す標準的な計算手順，また MGM2 は (b) 図に示す簡略化した手順のプログラムである．

```

program MAIN
REAL u(0:40,0:24),f(0:40,0:24)
DATA u/1025*0./ f/1025*-20./                                !predictor+bound values, rhs
DATA i1,j1,h/40, 24, .05/ naf,e/100, 1.E-5/
OPEN(20,FILE='OUTPUT.dat')
CALL MGM2(u,f,i1,j1,h,na,naf,rmean,e)                    !call MGM1 or MGM2
WRITE(20,'(I5, F6.3)')(u(i,j),i=0,i1)na,rmean           !output na,rmean
DO j=j1,0,-1
  WRITE(20,'(41F6.3)')(u(i,j),i=0,i1)                    !output u
ENDDO
CLOSE(20)
STOP
END

!***** Subroutine MGM1
SUBROUTINE MGM1(u,f,i1,j1,h,na,naf,rmean,e)
REAL u(0:i1,0:j1),f(0:i1,0:j1),a(24,-4:4),b(24), &
  r1(0:40,0:24),r2(0:20,0:12),r3(0:10,0:6),r4(0:5,0:3), &
  d1(0:40,0:24),d2(0:20,0:12),d3(0:10,0:6),d4(0:5,0:3)
i2=i1/2; i3=i2/2; i4=i3/2
j2=j1/2; j3=j2/2; j4=j3/2
kf=(i4+1)*(j4+1); lf=j4+1
na=0; 100 na=na+1
n =5; 101 n=n-1
! Determine residuals and restrictions

```

```

FORALL(i=1:i1-1,j=1:j1-1)r1(i,j) = &
  u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)-4.*u(i,j)-h*h*f(i,j) !comp r1
                                IF(n==1)GOTO 11
CALL RESTRC(r1,i1,j1,r2,i2,j2);      IF(n==2)GOTO 12 !restriction(r1 r2)
CALL RESTRC(r2,i2,j2,r3,i3,j3);      IF(n==3)GOTO 13 !restriction(r2 r3)
CALL RESTRC(r3,i3,j3,r4,i4,j4);      IF(n==4)GOTO 14 !restriction(r3 r4)
! Determin corrections by point relaxation
11 FORALL(i=1:i1-1,j=1:j1-1)d1(i,j)=r1(i,j)/4.; GOTO 21 !comp d1
12 FORALL(i=1:i2-1,j=1:j2-1)d2(i,j)=r2(i,j)/4.; GOTO 22 !comp d2
13 FORALL(i=1:i3-1,j=1:j3-1)d3(i,j)=r3(i,j)/4.; GOTO 23 !comp d3
! Determine corrections by Gaussian elimination
14 DO i=0,i4; DO j=0,j4; k=i*lf+j+1 !comp d4
  FORALL(l=-lf:lf)a(k,l)=0.; a(k,0)=1.
  IF(i>0.AND.i<i1.AND.j>0.AND.j<j1)THEN
    a(k,-lf)=1.; a(k,-1)=1.; a(k,0)=-4.; a(k,1)=1.; a(k,lf)=1.
  ENDIF
  b(k)=-r4(i,j)
ENDDO; ENDDO
CALL GAUSSB(a,b,kf,-lf,lf,det)
FORALL(i=0:i4,j=0:j4)d4(i,j) = b(i*lf+j+1)
! Prolongations and apply corrections
  CALL PROLON(d4,i4,j4,d3,i3,j3) !prolongation(d4 d3)
23 CALL PROLON(d3,i3,j3,d2,i2,j2) !prolongation(d3 d2)
22 CALL PROLON(d2,i2,j2,d1,i1,j1) !prolongation(d2 d1)
21 FORALL(i=1:i1-1,j=1:j1-1)u(i,j)=u(i,j)+d1(i,j) !apply d1
! Decide convergence by mean residual
rmean=0.
DO i=1,i1-1; DO j=1,j1-1
  rmean = rmean+ABS(r1(i,j)) !comp mean residual
ENDDO; ENDDO
rmean = rmean/FLOAT(i1*j1)
IF(n>1) GOTO 101
IF(rmean>e.AND.na<naf) GOTO 100 !decide convergence
END
!***** Subroutine RESTRC
SUBROUTINE RESTRC(r1,i1,j1,r2,i2,j2)
REAL r1(0:i1,0:j1),r2(0:i2,0:j2)
DO ii=1,i2-1; i=2*ii; DO jj=1,j2-1; j=2*jj
  r2(ii,jj) = r1(i,j)+(r1(i-1,j)+r1(i+1,j)+r1(i,j-1)+r1(i,j+1))/2. &
    +(r1(i-1,j+1)+r1(i+1,j-1)+r1(i-1,j-1)+r1(i+1,j+1))/4.
ENDDO; ENDDO
END
!***** Subroutine PROLON
SUBROUTINE PROLON(d2,i2,j2,d1,i1,j1)
REAL d2(0:i2,0:j2),d1(0:i1,0:j1)
FORALL(i=1:i2-1,j=1:j2-1)d1(2*i,2*j)=d2(i,j)
DO ii=1,i2-1; i=2*ii
  d1(i,1)=(3.*d1(i,0)+6.*d1(i,2)-d1(i,4))/8.
  FORALL(j=3:j1-3:2)d1(i,j)=(-d1(i,j-3)+9.*d1(i,j-1)+9.*d1(i,j+1)-d1(i,j+3))/16.
  d1(i,j1-1)=(-d1(i,j1-4)+6.*d1(i,j1-2)+3.*d1(i,j1))/8.
ENDDO
DO j=1,j1-1
  d1(1,j)=(3.*d1(0,j)+6.*d1(2,j)-d1(4,j))/8.
  FORALL(i=3:i1-3:2)d1(i,j)=(-d1(i-3,j)+9.*d1(i-1,j)+9.*d1(i+1,j)-d1(i+3,j))/16.
  d1(i1-1,j)=(-d1(i1-4,j)+6.*d1(i1-2,j)+3.*d1(i1,j))/8.
ENDDO
END

```

```

!**** Subroutine GAUSSB
SUBROUTINE GAUSSB(a,b,n,m1,m2,det)
DIMENSION a(n,m1:m2),b(n)
det=1.
cycle_1: DO k=1,n
  det=a(k,0)*det
  b(k)=b(k)/a(k,0); IF(k==n)CYCLE cycle_1
  j2=MINO(m2,n-k)
  FORALL(j=1:j2)a(k,j)=a(k,j)/a(k,0)
  i2=MINO(n,k-m1)
  DO i=k+1,i2; j0=k-i
    b(i)=b(i)-a(i,j0)*b(k)
    DO jk=1,j2; j=jk+k-i
      a(i,j)=a(i,j)-a(i,j0)*a(k,jk)
    ENDDO
  ENDDO
ENDDO cycle_1
DO i=n-1,1,-1; j2=MINO(m2,n-i)
  DO j=1,j2; k=i+j
    b(i)=b(i)-a(i,j)*b(k)
  ENDDO
ENDDO
END
!***** Subroutine MGM2
SUBROUTINE MGM2(u,f,i1,j1,h,na,naf,rmean,e)
REAL u(0:i1,0:j1),f(0:i1,0:j1),a(24,-4:4),b(24), &
  r1(0:40,0:24),r2(0:20,0:12),r3(0:10,0:6),r4(0:5,0:3), &
  d1(0:40,0:24),d2(0:20,0:12),d3(0:10,0:6),d4(0:5,0:3)
i2=i1/2; i3=i2/2; i4=i3/2
j2=j1/2; j3=j2/2; j4=j3/2
kf=(i4+1)*(j4+1); lf=j4+1
na=0; 100 na=na+1
! Determine residuals and restrictions
FORALL(i=1:i1-1,j=1:j1-1)r1(i,j) = &
  u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)-4.*u(i,j)-h*h*f(i,j)
CALL RESTRC(r1,i1,j1,r2,i2,j2)
CALL RESTRC(r2,i2,j2,r3,i3,j3)
CALL RESTRC(r3,i3,j3,r4,i4,j4)
! Determine corrections by Gaussian elimination
DO i=0,i4; DO j=0,j4; k=i*lf+j+1
  FORALL(l=-lf:lf)a(k,l)=0.; a(k,0)=1.
  IF(i>0.AND.i<i1.AND.j>0.AND.j<j1)THEN
    a(k,-lf)=1.; a(k,-1)=1.; a(k,0)=-4.; a(k,1)=1.; a(k,lf)=1.
  ENDDO
  b(k)=-r4(i,j)
ENDDO; ENDDO
CALL GAUSSB(a,b,kf,-lf,lf,det)
FORALL(i=0:i4,j=0:j4)d4(i,j) = b(i*lf+j+1)
! Prolongation, correct corrections by point relaxation
CALL PRORLX(d4,i4,j4,d3,i3,j3,r3) !prolongation(d4 d3),d3
CALL PRORLX(d3,i3,j3,d2,i2,j2,r2) !prolongation(d3 d2),d2
CALL PRORLX(d2,i2,j2,d1,i1,j1,r1) !prolongation(d2 d1),d1
FORALL(i=1:i1-1,j=1:j1-1)u(i,j)=u(i,j)+d1(i,j) !apply d1
! Decide convergence
rmean=0.
DO i=1,i1-1; DO j=1,j1-1
  rmean = rmean+ABS(r1(i,j))

```

```

ENDDO; ENDDO
rmean = rmean/FLOAT(i1*j1)
IF(rmean>e.AND.na<naf)          GOTO 100
END
!***** Subroutine PRORLX
SUBROUTINE PRORLX(d2,i2,j2,d1,i1,j1,r1)
REAL d2(0:i2,0:j2),d1(0:i1,0:j1),r1(0:i1,0:j1)

```

[この間にサブルーチン PROLON と同じ延長補間に関する 11 行の文が入る]

```

FORALL (i=1:i1-1,j=1:j1-1) &
  r1(i,j)=r1(i,j)+d1(i-1,j)+d1(i+1,j)+d1(i,j-1)+d1(i,j+1)-4.*d1(i,j)
FORALL (i=1:i1-1,j=1:j1-1)d1(i,j)=d1(i,j)+r1(i,j)/4.
END

```

メインプログラムではサブルーチン MGM1, MGM2 のいずれかを CALL しなければならない。サブルーチン MGM1 は、図 3.6(a) に示す計算手順によるもので、下図のフローチャートのように組まれている。一般に緩和計算では残差の計算と修正値の計算は対をなすが、MGM1 では常に基本格子で残差 r^1 を計算し、限定集計によって r^2, r^3, r^4 を求めている。また各格子で修正値 d^4, d^3, d^2 を計算し延長補間によって基本格子の d^1 を求め解 u に加えている。最後に r^1 から直接 d^1 を求め解 u に加えている。 n は緩和計算を行う格子を示す指標である。図 (a) の手順は解が十分に収束するまで反復される。 n_a はサイクル数で、計算は残差の平均値が十分小さくなったとき $r_{\text{mean}} \leq \varepsilon$, または所定のサイクル数に達したとき $n_a = n_{af}$ に終了する。プログラムの内容はその中のコメントと下図から理解できるものと思う。

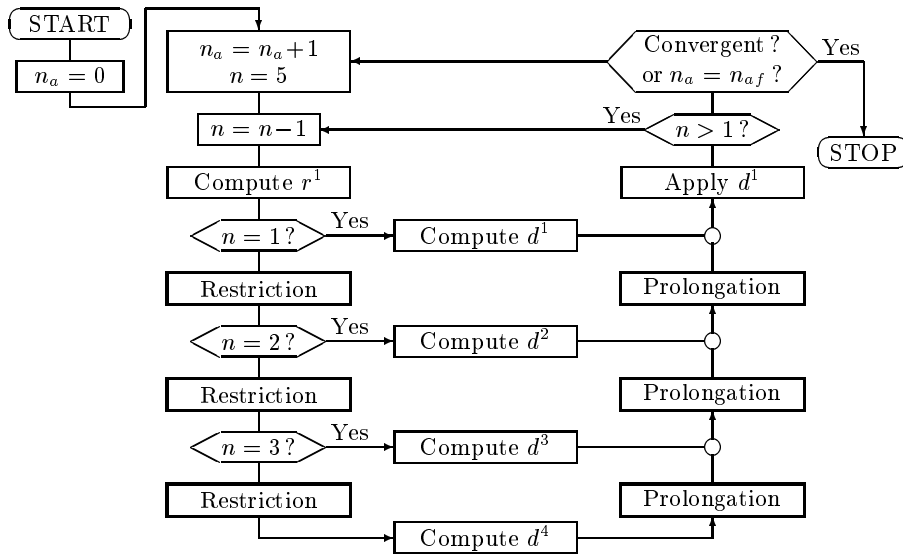


図 3.7: MGM1 のフローチャート

サブルーチン MGM2 はサブルーチン MGM1 の計算手順をコンパクトに纏めたものである。MGM1 では粗い mesh 4 で計算した修正値 d^4 は延長補間によって一気に mesh 1 の修正値 d^1 まで移したが、MGM2 では延長補間によって mesh 3 の修正値 d^3 に移した時点で、先に限定集計によって求めた残差値 r^3 がその影響を受けることに注目し、

$$r_{ij} = r_{ij} + d_{i-1,j} + d_{i+1,j} + d_{i,j-1} + d_{i,j+1} - 4d_{ij}$$

のように修正する．この式の右辺の残差 r^3 は修正前の u に対する残差で，また左辺の残差 r^3 は修正後の $u+d^3$ に対する残差である．サブルーチン PRORLX では，まず PROLON と同じ延長補間によって d^4 から d^3 を求め，次に r^3 を修正し，それからこの修正された r^3 に対し点緩和の修正値 $r_{ij}/4$ を求めこれを延長補間で移した d^3 に加えている．この一連の操作は mesh 2 と mesh 1 に関しても同様に行われる．MGM2 ではこのようにして，1 つのループの中で各格子レベルで残差の修正，修正値の計算と加算が効率よく行われることになる．

計算の結果は次表に示すように MGM1 と MGM2 による計算はともにサイクル数 $n_a = 8$ 程度で収束し，昔のベクトルコンピュータによる計算では，MGM2 の CPU 時間は MGM1 のものの約 $1/2$ である．

n_a	MGM1		MGM2	
	r_{mean}	ms	r_{mean}	ms
1	0.009063	20	0.009046	8
2	0.002929	39	0.002901	17
3	0.000608	57	0.000602	26
4	0.000251	75	0.000248	34
5	0.000068	93	0.000068	41
6	0.000032	111	0.000031	49
7	0.000013	128	0.000014	57
8	0.000007	145	0.000007	65

3.3.3 準線形 2 階楕円型微分方程式の混合境界値問題への適用

式 (3.9) が 2 独立変数 2 階の準線形楕円型微分方程式¹の場合について述べる．

$$L(u) \equiv au_{xx} + bu_{xy} + cu_{yy} + d = 0 \quad (3.18)$$

ただし a, b, c, d は一般に x, y, u, u_x, u_y の関数である．準線形の場合には， a, b, c, d 中の u^h を暫時定数とみなし，線形の場合の式をそのまま用いて計算を進め，随時 a, b, c, d 中の u^h を新しいものに換えていく方法が取られる．多くの問題では，微分方程式中の最高階数の微分が主役を演じるので，この方法で収束解を得ることができる．

式 (3.18) の近似差分方程式は一般に

$$\begin{aligned} & c_{ij}^0 u_{ij} + c_{ij}^1 u_{i-1,j} + c_{ij}^2 u_{i+1,j} + c_{ij}^3 u_{i,j-1} + c_{ij}^4 u_{i,j+1} \\ & + c_{ij}^5 u_{i-1,j-1} + c_{ij}^6 u_{i+1,j-1} + c_{ij}^7 u_{i-1,j+1} + c_{ij}^8 u_{i+1,j+1} = f_{ij} \end{aligned} \quad (3.19)$$

のように書かれる．ただし $c_{ij}^0 + c_{ij}^1 + c_{ij}^2 + c_{ij}^3 + c_{ij}^4 = 0$ ， $c_{ij}^5 + c_{ij}^6 + c_{ij}^7 + c_{ij}^8 = 0$ で，左辺の係数は優対角の条件を満足またはあまり逸脱しないものとする．また右辺 f_{ij} は格子間隔の 2 乗 h^2 またはこれに相当の因子を含むものとする．式 (3.19) の場合には残差の式 (3.11) は

$$\begin{aligned} r_{ij}^{(n)} = & c_{ij}^0 u_{ij}^{(n)} + c_{ij}^1 u_{i-1,j}^{(n)} + c_{ij}^2 u_{i+1,j}^{(n)} + c_{ij}^3 u_{i,j-1}^{(n)} + c_{ij}^4 u_{i,j+1}^{(n)} \\ & + c_{ij}^5 u_{i-1,j-1}^{(n)} + c_{ij}^6 u_{i+1,j-1}^{(n)} + c_{ij}^7 u_{i-1,j+1}^{(n)} + c_{ij}^8 u_{i+1,j+1}^{(n)} - f_{ij} \end{aligned} \quad (3.20)$$

また修正値を求める式 (3.12) は

$$\begin{aligned} & c_{ij}^0 d_{ij} + c_{ij}^1 d_{i-1,j} + c_{ij}^2 d_{i+1,j} + c_{ij}^3 d_{i,j-1} + c_{ij}^4 d_{i,j+1} \\ & + c_{ij}^5 d_{i-1,j-1} + c_{ij}^6 d_{i+1,j-1} + c_{ij}^7 d_{i-1,j+1} + c_{ij}^8 d_{i+1,j+1} = -r_{ij}^{(n)} \end{aligned} \quad (3.21)$$

¹準線形とは最高階数に関して線形であることをいう．例えば 2 階の微分方程式の場合には 2 階微分の線形結合，1 階の連立微分方程式の場合には各式が 1 階微分の線形結合であることをいう．

となる．次にこの式に上記の限定集計の式 (3.4) を適用し，粗い格子の修正値を求める式を導出する．その式は初めかなり冗長なものになる．しかしながら，係数 c は式 (3.18) の 2 階微分に由来し $c^1 \approx c^2$ ， $c^3 \approx c^4$ ， $c^1 + c^3 + c^0/2 \approx 0$ ， $c^5 \approx c^8 \approx -c^6 \approx -c^7$ のように近似可能で²，また通常なめらかに変化することを考慮すれば，限定集計した式の d_{ij} ， $d_{i-1,j}$ ， $d_{i-2,j}$ ， $d_{i-1,j-1}$ の係数は次のように簡単になる．

$$\begin{aligned} c_{ij}^0 + (c_{i+1,j}^1 + c_{i-1,j}^2 + c_{i,j+1}^3 + c_{i,j-1}^4)/2 + (c_{i+1,j+1}^5 + c_{i-1,j+1}^6 + c_{i+1,j-1}^7 + c_{i-1,j-1}^8)/4 &\approx c_{ij}^0/2 \\ c_{i-1,j}^0/2 + c_{ij}^1 + (c_{i-1,j+1}^3 + c_{i-1,j-1}^4)/4 + (c_{i,j+1}^5 + c_{i,j-1}^7)/2 &\approx c_{i-1,j}^0/4 + c_{ij}^1/2, \\ c_{i-1,j}^1/2 + (c_{i-1,j+1}^5 + c_{i-1,j-1}^7)/4 &\approx c_{i-1,j}^1/2, \\ c_{i-1,j-1}^0/4 + (c_{i,j-1}^1 + c_{i-1,j}^3)/2 + c_{ij}^5 &\approx c_{ij}^5 \end{aligned}$$

$d_{i-2,j-1}$ と $d_{i-2,j-2}$ の係数はそれぞれ $c_{i-1,j-1}^1/4 + c_{i-1,j}^5/2$ ， $c_{i-1,j-1}^5/4$ となる．ほかの係数についても同様である．また延長補間で得られた修正値 d^h がかなりなめらかに変化することに注意すれば，近似的に

$$\begin{aligned} c_{ij}^5 d_{i-1,j-1} &\approx c_{ij}^5 d_{i-1,j-1}/2 + (c_{ij}^5 d_{i-2,j-2} + c_{ij}^5 d_{ij})/4, \\ c_{i-1,j}^5 d_{i-2,j-1} &\approx (c_{i-1,j}^5 d_{i-2,j-2} + c_{i-1,j}^5 d_{i-2,j})/2 \end{aligned}$$

のように分配し，また線形の場合と同様に

$$\begin{aligned} (c_{i-1,j}^0 d_{i-1,j} + c_{i-1,j}^0 d_{i-1,j} + c_{i-1,j}^0 d_{i-1,j} + c_{i-1,j}^0 d_{i-1,j})/4 &\approx c_{IJ}^0 d_{IJ}, \\ c_{i-1,j}^1 d_{i-2,j}/2 + (c_{i-1,j-1}^1 d_{i-2,j-1} + c_{i-1,j+1}^1 d_{i-2,j+1})/4 &\approx c_{i-1,j}^1 d_{I-1,J}, \\ (c_{ij}^5 + c_{i-1,j}^5 + c_{i,j-1}^5 + c_{i-1,j-1}^5) d_{i-2,j-2} &\approx c_{i-1,j-1}^5 d_{I-1,J-1} \end{aligned}$$

のように結合することができる．結局限定集計した式は次のようになる．

$$\begin{aligned} c_{ij}^0 d_{IJ} + c_{i-1,j}^1 d_{I-1,J} + c_{i+1,j}^2 d_{I+1,J} + c_{i,j-1}^3 d_{I,J-1} + c_{i,j+1}^4 d_{I,J+1} \\ + c_{i-1,j-1}^5 d_{I-1,J-1} + c_{i+1,j-1}^6 d_{I+1,J-1} + c_{i-1,j+1}^7 d_{I-1,J+1} + c_{i+1,j+1}^8 d_{I+1,J+1} \\ + (c_{ij}^0 d_{ij} + c_{ij}^1 d_{i-1,j} + c_{ij}^2 d_{i+1,j} + c_{ij}^3 d_{i,j-1} + c_{ij}^4 d_{i,j+1} \\ + c_{ij}^5 d_{i-1,j-1} + c_{ij}^6 d_{i+1,j-1} + c_{ij}^7 d_{i-1,j+1} + c_{ij}^8 d_{i+1,j+1})/2 = -r_{IJ} \end{aligned}$$

この式の括弧の中は， h 格子の修正値の差分方程対式 (3.21) から $-r_{ij}$ になり，この項の大きさは一様残差値に対して r_{IJ} の $1/8$ になる．今この部分を無視すれば，粗い格子の修正値を求める式は保存形で次のよう表わされる．

$$\begin{aligned} c_{i-1,j}^1 (d_{I-1,J} - d_{IJ}) + c_{i+1,j}^2 (d_{I+1,J} - d_{IJ}) + c_{i,j-1}^3 (d_{I,J-1} - d_{IJ}) + c_{i,j+1}^4 (d_{I,J+1} - d_{IJ}) \\ + c_{i-1,j-1}^5 (d_{I-1,J-1} - d_{IJ}) + c_{i+1,j-1}^6 (d_{I+1,J-1} - d_{IJ}) \\ + c_{i-1,j+1}^7 (d_{I-1,J+1} - d_{IJ}) + c_{i+1,j+1}^8 (d_{I+1,J+1} - d_{IJ}) = -r_{IJ} \end{aligned} \quad (3.22)$$

この式に解釈を与えれば，式 (3.18) の 2 階微分の項に関し 1 階差分を，式 (3.19)(3.21) では $\{c_{ij}^2(u_{i+1,j} - u_{ij})\}/h$ のように差分近似したのに対し，この式では $\{c_{i+1,j}^2(u_{i+1,J} - u_{IJ})\}/H$ のように差分近似しているということである．限定集計はこの式を用い次々に行われる．また延長補間は式 (3.8) を用いて行われる．

ここで準線形微分方程式 (3.19) の 1 つの混合境界値問題を考える．その計算領域と計算格子は前節と同じであるが，左側境界に Dirichlet 条件，右側境界に Neumann 条件，上下境界に周期条件が課されるもの

²なめらかな構造格子に対し写像空間の正方形格子上で計算する場合にはこのような近似は概ね妥当である．

とする．ただしここでは微分方程式 (3.18) の項 bu_{xy} は上の説明とは異なり d の中に含めるものとする．このとき差分方程式 (3.19) の係数は c^0, c^1, c^2, c^3, c^4 のみとなり，その連立1次方程式は優対角の条件を常に満足する．右側境界では，Neumann 条件 $(u_x)_{i_f j} = g_j$ を式 (3.19) に考慮すれば，差分方程式は

$$(c_{i_f j}^1 + c_{i_f j}^2)u_{i_f-1,j} + c_{i_f j}^3 u_{i_f,j-1} + c_{i_f j}^4 u_{i_f,j+1} + c_{i_f j} u_{i_f j} = f_{i_f j} - 2hc_{i_f j}^2 g_j$$

のようになる．あらかじめメインプログラムで $f_{i_f j}$ に $-2hc_{i_f j}^2 g_j$ を加えておく必要がある．またこの境界上の限定集計の式は，長波長の残差を想定し $r_{i_f+1,j-1} = r_{i_f,j-1}$, $r_{i_f+1,j} = r_{i_f j}$, $r_{i_f+1,j+1} = r_{i_f,j+1}$ と置けば，

$$r_{I_f J} = 1.5r_{i_f j} + (r_{i_f-1,j} + 1.5r_{i_f,j-1} + 1.5r_{i_f,j+1})/2 + (r_{i_f-1,j-1} + r_{i_f-1,j+1})/4$$

のようになる．上下境界では，周期条件の課される上下の境界点を共に計算している．境界条件に関する文にはコメントを付けてある．

次のプログラムは，上記のサブルーチン MGM2 にならって作られたものである． c, f は差分方程式の係数 c^0, c^1, \dots と右辺 f の配列， a は最も粗い格子の修正値を Gauss 消去法で解くときに用いられる配列である．なお Gauss 消去法による連立1次方程式の計算は 1.3 節に述べたサブルーチン GAUSS を引用して行われ， wa はこのときに必要な作業用配列である．MGM2 で用いたサブルーチン PRORLX は，ここでは延長補間のサブルーチン PROLON と Gauss 反復法による緩和計算の部分に分けられる．

```

*****
! Sol of mixed bound value problem of elliptic pde in rectangul domain by MGM
*****
SUBROUTINE MGM3(u,c,f,i1,j1,phi,h,na,naf,rmean,e)
REAL u(0:i1,0:j1),c(0:i1,0:j1,0:4),f(0:i1,0:j1),a(24,25),wa(25), &
  r1(0:40,0:24),r2(0:20,0:12),r3(0:10,0:6),r4(0:5,0:3), &
  d1(0:40,0:24),d2(0:20,0:12),d3(0:10,0:6),d4(0:5,0:3)
i2=i1/2; i3=i2/2; i4=i3/2
j2=j1/2; j3=j2/2; j4=j3/2
n=(i4+1)*(j4+1); n1=n+1
na=0; 100 na=na+1
! Determine residuals r1 and restrictions
DO i=1,i1; ip1=i+1; IF(i==i1)ip1=i-1 !Neumann cond
  DO j=0,j1; jm1=j-1; IF(j==0)jm1=j1-1 !periodic cond
    jp1=j+1; IF(j==j1)jp1=1 !periodic cond
    r1(i,j) = c(i,j,1)*u(i-1,j)+c(i,j,2)*u(ip1,j) & !comp residuals r1
      +c(i,j,3)*u(i,jm1)+c(i,j,4)*u(i,jp1) &
      +c(i,j,0)*u(i,j)-f(i,j)
  ENDDO; ENDDO
CALL RESTRC(r1,i1,j1,r2,i2,j2) !restriction(r1 r2)
CALL RESTRC(r2,i2,j2,r3,i3,j3) !restriction(r2 r3)
CALL RESTRC(r3,i3,j3,r4,i4,j4) !restriction(r3 r4)
! Determine corrections d4 by Gaussian elimination
l=j4+1
DO i=0,i4; i0=8*i
  cycle_1: DO j=0,J4; j0=8*j; k=i*1+j+1
    FORALL(m=1:n)a(k,m)=0.; a(k,k)=1.; a(k,n1)=-r4(i,j) !setup rhs
    IF(i==0)CYCLE cycle_1 !Dirichlet cond
    kml=k-1; kpl=k+1; im4=i0-4; ip4=i0+4
    km1=k-1; kp1=k+1; jm4=j0-4; jp4=j0+4
    IF(i==i4)THEN; kpl=k-1; ip4=i1; ENDIF !Neumann cond
    IF(j==0)THEN; km1=k+1-2; jm4=j1-4; ENDIF !periodic cond
    IF(j==j4)THEN; kp1=k-1+2; jp4=4; ENDIF !periodic cond

```

```

a(k,km1) = c(im4,j0,1) !setup coef matrix
a(k,km1) = c(i0,jm4,3)
a(k,k ) = -c(im4,j0,1)-c(ip4,j0,2)-c(i0,jm4,3)-c(i0,jp4,4)
a(k,kp1) = c(i0,jp4,4)
a(k,kp1) = c(ip4,j0,2)+a(k,kp1)
ENDDO cycle_1; ENDDO
CALL GAUSS(a,wa,n,n1,det,1.E-5) !solve linear equations
FORALL (i=0:i4,j=0:j4)d4(i,j) = a(i*1+j+1,n1) !determine d4
! Prolongation, determine residuals and corrections by point relaxation
CALL PROLON(d4,i4,j4,d3,i3,j3) !prolongation(d4 d3)
DO i=1,i3; i0=4*i; DO j=0,j3; j0=4*j
  im1=i-1; ip1=i+1; jm1=j-1; jp1=j+1
  im2=i0-2; ip2=i0+2; jm2=j0-2; jp2=j0+2
  IF (i==i3) THEN; ip1=i3-1; ip2=i1; ENDF !Neumann cond
  IF (j== 0) THEN; jm1=j3-1; jm2=j1-2; ENDF !periodic cond
  IF (j==j3) THEN; jp1=1; jp2=2; ENDF !periodic cond
  r3(i,j) = r3(i,j)+c(im2,j0,1)*(d3(im1,j)-d3(i,j)) & !determine r3
                +c(ip2,j0,2)*(d3(ip1,j)-d3(i,j)) &
                +c(i0,jm2,3)*(d3(i,jm1)-d3(i,j)) &
                +c(i0,jp2,4)*(d3(i,jp1)-d3(i,j))
  c0 = -c(im2,j0,1)-c(ip2,j0,2)-c(i0,jm2,3)-c(i0,jp2,4)
  r3(i,j) = -r3(i,j)/c0 !d3 by point relaxation
ENDDO; ENDDO
FORALL (i=1:i3,j=0:j3)d3(i,j)=d3(i,j)+r3(i,j) !determine d3
CALL PROLON(d3,i3,j3,d2,i2,j2) !prolongation(d3 d2)
DO i=1,i2; i0=2*i; DO j=0,j2; j0=2*j
  im1=i-1; ip1=i+1; im0=i0-1; ip0=i0+1
  jm1=j-1; jp1=j+1; jm0=j0-1; jp0=j0+1
  IF (i==i2) THEN; ip1=i2-1; ip0=i1; ENDF !Neumann cond
  IF (j==0 ) THEN; jm1=j2-1; jm0=j1-1; ENDF !periodic cond
  IF (j==j2) THEN; jp1=1; jp0=1; ENDF !periodic cond
  r2(i,j) = r2(i,j)+c(im0,j0,1)*(d2(im1,j)-d2(i,j)) & !determine r2
                +c(ip0,j0,2)*(d2(ip1,j)-d2(i,j)) &
                +c(i0,jm0,3)*(d2(i,jm1)-d2(i,j)) &
                +c(i0,jp0,4)*(d2(i,jp1)-d2(i,j))
  c0 = -c(im0,j0,1)-c(ip0,j0,2)-c(j0,jm0,3)-c(i0,jp0,4)
  r2(i,j) = -r2(i,j)/c0 !d2 by point relaxation
ENDDO; ENDDO
FORALL (i=1:i2,j=0:j2)d2(i,j)=d2(i,j)+r2(i,j) !determine d2
CALL PROLON(d2,i2,j2,d1,i1,j1) !prolongation(d2 d1)
DO i=1,i1; DO j=0,j1
  im1=i-1; ip1=i+1;jm1=j-1; jp1=j+1;
  IF (i==i1) ip1=i-1 !Neumann cond
  IF (j==j1) jp1=1 !periodic cond
  IF (j==0 ) jm1=j1-1 !periodic cond
  r1(i,j) = r1(i,j)+c(i,j,0)*d1(i,j) & !determine r1
                +c(i,j,1)*d1(im1,j)+c(i,j,2)*d1(ip1,j) &
                +c(i,j,3)*d1(i,jm1)+c(i,j,4)*d1(i,jp1)
ENDDO; ENDDO
FORALL (i=1:i1,j=0:j1)d1(i,j) = d1(i,j)-r1(i,j)/c(i,j,0) !determine d1
! Apply corrections
DO i=1,i1
  FORALL (j=0:j1)u(i,j)=u(i,j)+d1(i,j) !apply d1
  um=u(i,0)+u(i,j1); u(i,0)=(um-phi)/2.; u(i,j1)=(um+phi)/2. !periodic cond
ENDDO
! Decide convergence
rmean=0.

```

```

DO i=1,i1; DO j=0,j1
  rmean = rmean+ABS(r1(i,j))/(i1*j1)          !comp mean residual
ENDDO; ENDDO
WRITE(20,'(I3,F12.6)')na,rmean              !output converging process
IF(rmean>e.AND.na<naf)                      GOTO 100          !decide convergence
END
!***** Subroutine RESTRC
SUBROUTINE RESTRC(r1,i1,j1,r2,i2,j2)
REAL r1(0:i1,0:j1),r2(0:i2,0:j2)
DO jj=0,j2; j=2*jj; jm1=j-1; jp1=j+1
  IF(j==0)jm1=j1-1; IF(j==j1)jp1=1          !periodic cond
  DO ii=1,i2-1; i=2*ii
    r2(ii,jj)=r1(i,j)+r1(i-1,j)+r1(i+1,j)+r1(i,jm1)+r1(i,jp1))/2. &
      +(r1(i-1,jm1)+r1(i+1,jp1)+r1(i+1,jm1)+r1(i-1,jp1))/4.
  ENDDO
  r2(i2,jj)=1.5*r1(i1,j)+r1(i1-1,j)+1.5*r1(i1,jm1) &          !Neumann cond
    +1.5*r1(i1,jp1))/2.+(r1(i1-1,jm1)+r1(i1-1,jp1))/4.
ENDDO
END

```

このサブルーチンについて、!Determine corrections d4 by Gaussian elimination のブロックで $a_{k,kpl} = c_{ip4,j}^2 + a_{k,kpl}$ について、右辺に $a_{k,kpl}$ を加えることはその上で $a_{k,m} = 0$, ($m = 1, \dots$) と置いているので一般には意味のないことであるが、Neumann 条件の課される右側境界では必要なことに注意。またこの文の前に $a_{k,kml} = c_{im4,j}^1$ を置かなければならない。!Apply corretions のブロックで周期条件を確実に満足させる操作は必要なもので、省略すると解が発散またはほとんど収束しても最後に発散することになる。副サブルーチン PROLON は前項のものと同じである。

最後に、サブルーチン MGM3 を引用する計算例として、極座標系 r, θ における Poisson 方程式の混合境界値問題

$$u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 1.0 \quad (1 < r < 3, 0 < \theta < \pi/4) \quad (3.23a)$$

$$u(1, \theta) = 10 + \phi \theta / (\pi/4), \quad u_r(3, \theta) = -0.5, \quad (3.23b)$$

$$u(r, \pi/4) = u(r, 0) + \phi, \quad u_\theta(r, \pi/4) = u_\theta(r, 0)$$

を解くことにする。ただし $\phi = -1.0$ 。基本格子は r 方向に 40 等分、 θ 方向に 24 等分したものとする。式 (3.23) の差分方程式は、前の説明とは異なり左辺に 1 階微分の項も含めることにすれば次のようになる。

$$(1 - g_{ij})u_{i-1,j} + (1 + g_{ij})u_{i+1,j} + h_{ij}u_{i,j-1} + h_{ij}u_{i,j+1} - 2(1 + h_{ij})u_{ij} = \Delta r^2 \quad (3.24)$$

ただし $g_{ij} \equiv (\Delta r)/2r_{ij}$, $h_{ij} \equiv (\Delta r/r_{ij}\Delta\theta)^2$, $\Delta r = 1/20$, $\Delta\theta = \pi/4/24$ である。この差分方程式の係数 $1 \pm g_{ij}$ は本来格子間隔によるが、ここでは粗い格子で修正値を求める式にもそのまま使うことにする。この混合境界値問題のメインプログラムは次のようになる。

```

program MAIN
REAL u(0:40,0:24),c(0:40,0:24,0:4),f(0:40,0:24)
DATA i1,j1,grad,phi,naf,e/40,24,-.5,-1.,100,1.E-5/
dr=2./FLOAT(i1); dt=3.14159/4./FLOAT(j1)
DO I=1,I1; DO J=0,J1
  r=1.+dr*i; g=dr/2./r; h=dr*dr/(r*dt*r*dt)
  c(i,j,1)=1.-g; c(i,j,2)=1.+g; c(i,j,3)=h; c(i,j,4)=h; c(i,j,0)=-2.*(1.+h)
  f(i,j)=1.*dr*dr
ENDDO; ENDDO

```

```

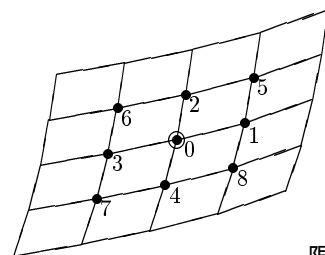
FORALL (i=1:i1) f(i, 0)=f(i, 0)+c(i, 0,3)*phi           !periodic cond
FORALL (i=1:i1) f(i, j1)=f(i, j1)-c(i, j1,4)*phi
DO j=0, j1
  u(0, j)=10.+j/FLOAT(j1)*phi                          !Dirichlet cond
  f(i1, j)=f(i1, j)-grad*2*dr*c(i1, j, 2)             !Neumann cond
  FORALL (i=1:i1) u(i, j)=u(0, j)                     !starting values
ENDDO
OPEN (20, FILE='OUTPUT.dat')
CALL MGM3(u, c, f, i1, j1, phi, h, na, naf, rmean, e)
DO j=j1, 0, -1
  WRITE(20, '(41F8.3)')(u(i, j), i=0, i1)
ENDDO
CLOSE(20)
STOP; END

```

計算の結果は、出発値 $u \equiv 0$ を与えた場合にはサイクル数 $n_a = 64$ で収束し、また Dirichlet 条件と同じ周期条件を満足する出発値 $u(r, \theta) = 10 + \phi \theta / (\pi/4)$ を全領域に与えた場合には $n_a = 8$ で収束した。

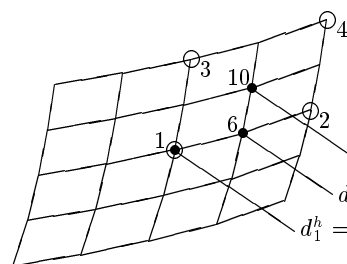
3.4 多重格子法の各種問題への拡張

多重格子法は、差分法と同様に、有限要素法 (FEM, finite element method) の計算効率をあげるために用いられている。通常 四辺形双 1 次要素または四辺形ラグランジュ要素が用いられ、有限要素方程式 (節点値 u_i の連立 1 次方程式) が多重格子法で解かれる。その限定集計と延長補間が差分法と異なるところで、差分法のを大らかに利用する方法と FEM らしく行う方法がある。前者の方法でも、最も細かい基本メッシュに対しては、もちろん有限要素方程式から残差を正確に計算しすべての残差が 0 になるまで反復計算を履行するのであるが、その上に重なる粗いメッシュに対しては大らかに計算するということである。その限定集計と延長補間は下図に示すように大に行われる。メッシュのゆがみや間隔の変化が大きいところでは、限定集計と延長補間に伴う誤差が大きくなり、サイクル数は増えると思われるが、結果の精度には無関係である。



$$\begin{aligned}
 r_0^{2h} &= r_0^h + \frac{1}{2}(r_1^h + r_2^h + r_3^h + r_4^h) \\
 &\quad + \frac{1}{4}(r_5^h + r_6^h + r_7^h + r_8^h) \\
 &\approx 4r_0^h
 \end{aligned}$$

限定集計



延長補間

$$\begin{aligned}
 d_{10}^h &= (d_1^{2h} + d_2^{2h} + d_3^{2h} + d_4^{2h})/4 \\
 d_6^h &= (d_1^{2h} + d_2^{2h})/2 \\
 d_1^h &= d_1^{2h}
 \end{aligned}$$

後者の方法は，メッシュのゆがみや間隔の変化が大きいところ，また四辺形や六面体以外の要素にもそのまま適用できる．形状関数を N_i とすれば，限定集計の式は

$$r_i^{2h} = [I_h^{2h} r^h]_i = \int_{\Omega^{e^{2h}}} N_i^{2h}(\mathbf{x}) N_j^h(\mathbf{x}) r_j^h d\mathbf{x} \quad (3.25)$$

また延長補間の式は

$$d_j^h = [I_{2h}^h d^{2h}]_j = N_i^{2h}(\mathbf{x}_j) d_i^{2h} \quad (3.26)$$

となる．ここで蛇足を加えれば，FEMでは要素の節点 \mathbf{x}_i に節点値 $u_i(r_i, d_i, \dots)$ が定義され，要素内の任意の点 \mathbf{x} の u は補間関数 $u(\mathbf{x}) = N_i(\mathbf{x})u_i$ から求められる． u の微分値，積分値は補間関数を微分または積分することによって求められる．ある節点の微分値はその節点を共有する要素のものが一般に異なる値を示すことになる．このように異なる値を示すものは形状関数 N_i の重みを付けて積分し平均したものが取られる．延長補間の式 (3.26) は，細かいメッシュ h の節点 \mathbf{x}_j における補間値 d_j^h は，その点が粗いメッシュ $2h$ の n 番目の要素内にあれば，その要素の補間関数 $N_i^{2h}(\mathbf{x})d_i^{2h}$ から求められるということである．また式 (3.25) の $N_j^h(\mathbf{x})r_j^h$ は細かいメッシュ h の残差の補間値で，この限定集計の式は，粗いメッシュ $2h$ の残差 r_i^{2h} がこの補間値に形状関数 $N_i^{2h}(\mathbf{x})$ の重みを付けて積分したものであることを意味している．粗いメッシュのすべて節点にわたってこの残差の積分値を合計したものは，任意の点の形状関数の合計が1であることから，細かい格子の残差値を計算領域にわたって積分したものである．このことから，式 (3.25) は限定集計の式として妥当なものであることが分かる．

多重格子法の実用型方程式への適用例として，Jameson(1979)の遷音速ポテンシャル流れ方程式の解法の概要を紹介する³．定常2次元遷音速流れの完全ポテンシャル方程式は，大部分の亜音速流れ域では楕円型であるが一部の超音速流れ域では双曲型になる．超音速域ではある点の状態はその点から上流側に広がる円錐状の依存域 (domain of dependence) 内の状態にのみ依存し，逆にその点の状態はその点から下流側に広がる円錐状の影響域 (domain of influence) 内の状態にのみ影響を及ぼす．なお亜音速域では依存域，影響域は流れ場全域となる．離散方程式はこの物理現象を反映したものであるべきで，超音速域で中心差分を上流差分に切替える方法が用いられている．また超音速域に人工粘性を付加する方法も用いられているが，上流差分は中心差分に拡散項を付加したものであることからこれら2つの方法は本質的に同じものである．この流れの微分方程式は線形化の後

$$L\phi \equiv a\phi_{xx} + b\phi_{yy} = 0 \quad (3.27)$$

となる．この完全ポテンシャル方程式 (full potential equation) は， Δ 形陰解法の式

$$(\alpha - A\delta_x^2)(\alpha - B\delta_y^2)\Delta\phi = \omega\alpha L\phi \quad (3.28)$$

に書換え，ADI法を適用し解かれている．ただし $A = a/\Delta x^2$ ， $B = b/\Delta y^2$ ， δ_x ， δ_y は中心差分の演算子， $\alpha \sim 4 \min(A, B)$ ， $\omega \sim 1.5$ ，不等間隔の曲線座標格子を使用．MGMでは波長 $2\Delta x$ の残差の波を減衰させれば十分で，それには α をこのように選ぶのが適当である．この方法は亜音速流れには有効であるが，流れが超音速になり，すなわち $A < 0$ ，式 (3.27) が双曲型になるところでは不安定になり，このままでは遷音速流れには適用できない．

Jamesonは，遷音速ポテンシャル流れの方程式として，式 (3.27) の代わりに双曲形偏微分方程式

$$\beta_0\phi_t + \beta_1\phi_{xt} + \beta_2\phi_{yt} = a\phi_{xx} + b\phi_{yy} \quad (3.29)$$

³Jameson, A., Acceleration of transonic potential flow calculations on arbitrary meshes by the Multiple grid method, AIAA 4th CFD Conf., (1979-7), 122-146.

を用いる解法を提案している．この式は解が収束すれば $\partial/\partial t = 0$ になるので式 (3.27) と等価になる．式 (3.29) を ADI 法で解くための Δ 形陰解法の式は次のようになる．

$$(S - A\delta_x^2)(S - B\delta_y^2)\Delta\phi = \omega S L\phi \quad (3.30)$$

ただし $S \equiv \alpha_0 + \alpha_1\delta_x^- + \alpha_2\delta_y^-$, $\alpha_0, \alpha_1, \alpha_2$ は $\beta_0, \beta_1, \beta_2$ による定数 , δ_x^-, δ_y^- は上流差分の演算子である . $A < 0$ のところでは δ_x^2 も上流差分のものに切替えることが必要である . MGM の計算手順は , 最細格子 最粗格子 二番目に細かい格子を 1 サイクルとし , 各格子レベルで式 (3.30) を ADI 法で 1 回計算する . 1 サイクルの計算量は最細格子の計算量の約 $1\frac{2}{3}$ になる . 遷音速翼まわりの流れの計算例では , 192×32 最細格子 , 5 レベルの多重格子の場合に 10 サイクルで収束解が得られる .

多重格子法は差分法と FEM では離散方程式の効率的な解法としてかなり用いられているが , 境界積分法ではあまりない . 最後に多重格子法のパネル法への適用について述べた Slooff(1982) の解説⁴を参考までに引用する . Atkinson(1973) は , 次の第 2 種 Fredholm 型積分方程式の解法に多重格子法を用いている .

$$\lambda f(x) - \int_0^1 K(x, \xi) f(\xi) d\xi = g(x) \quad (3.31)$$

この方程式の計算式は

$$\lambda f(x_i) - \sum_j W_j K(x_i, x_j) f_j h_j = g(x_i) \quad (3.32)$$

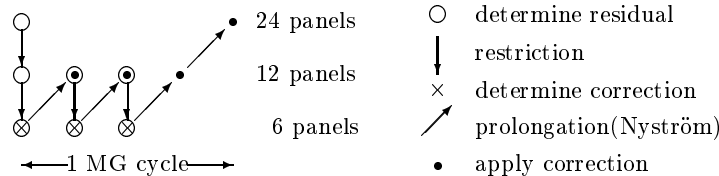
となるが , 行列 $[I - W_j K_{ij} h_j] = A$, ベクトル $\{f(x_j)\} = x$, $\{g(x_i)\} = b$ と置けば連立 1 次方程式

$$Ax = b \quad (3.33)$$

になる . 1 サイクルの計算は次のようなものである . 細かい格子 G^2 上で , $A^2 x^2 = b^2$ を Jacobi 緩和法で解いて x^2 を求める . $r^2 = A^2 x^2 - b^2$ を計算する . 粗い格子 G^1 に対し , $r^1 = I_2^1 r^2$ を計算する . $A^1 \delta x^1 = r^1$ を直接法で解いて δx^1 を求める . $\delta x^2 = I_1^2 \delta x^1$ を Nyström 補間すなわち

$$f(x) = \frac{1}{\lambda} [g(x) + \sum_j W_j K(x, x_j) f_j h_j] \quad (3.34)$$

から計算する . x^2 を修正する . この方法では G^1 に十分粗い格子を用いているので直接法が適用できる . また式 (3.34) の計算には平滑化操作が入るので , Jacobi の緩和計算を省略することもできる . 第 1 種 Fredholm 積分方程式は , 特異核関数の特異性を分離し , 第 2 種の形に帰着させて解かれる . 揚力の作用する平板翼まわりの流れに関する積分方程式を , 24 パネルを用いて解く場合に , 多重格子法では 6 サイクルで収束する . なおこの問題は Jacobi 緩和法のみでは発散する .



⁴Slooff, J.W., Requirements and developments shaping a next generation of integral methods, *Numerical Methods in Aeronautical Fluid Dynamics*, Roe, P.L., ed., p.517, 1982, Academic Press.