

第2章 連立1次方程式の解法—反復法

反復法 (iteration) は，楕円型偏微分方程式の境界値問題を解くときにできるような，大形疎行列の連立1次方程式を解くために用いられる．疎行列 (sparse matrix) とは，大部分の要素の値がゼロの行列のことである．このような行列はまた，優対角行列であることが多い．優対角行列 (diagonally dominant matrix) は式次で定義されるものである．

$$|a_{ii}| \geq \left(\sum_{j=1}^{i-1} + \sum_{j=i+1}^n \right) |a_{ij}| \quad (i = 1, 2, \dots, n) \quad (2.1)$$

すなわち各主対角要素の絶対値がその行の残りすべての要素の絶対値の和に等しいかまたはそれよりも大きい行列である．反復法の計算は係数行列が優対角の条件を満足するときによく収束する．反復法の特徴は直接法に比べ，計算に必要なメモリが著しく少ないこと，また少ない桁数の計算でよい結果が得られることである．

2.1 反復法の基本操作

はじめに，簡単な例題を計算のプロセスの見える手計算で解き，反復法の基本操作の意味を十分納得のいくように説明する．先を急ぐ読者は次節から読んでもよい．

2.1.1 2点境界値問題

始めに次の常微分方程式の2点境界値問題を取り上げる．

$$u''(x) = 0 \quad (0 < x < 1.0) \quad (2.2a)$$

$$u(0) = 0, \quad u(1.0) = 1.0 \quad (2.2b)$$

この方程式は幾何学的に解釈すれば解曲線 $u(x)$ の曲率 u'' がいたるところゼロということで，この問題の解は2つの境界値を結ぶ直線になり，この問題は解くまでもないということになる．ここでは式 (2.2a) の境界値問題を差分法により数値的に解くことにする．差分法 (finite-difference method) では，関数 $u(x)$ の常微分方程式 $L(u) = 0$ が，有限個の計算点 x_i ($i = 0, 1, \dots, n$) 上の関数値 u_i ($0, 1, \dots, n$) の差分方程式 (finite-difference equations) で近似される．差分方程式は一般には u_i の代数方程式であるが，通常はその連立1次方程式である．ここでは計算領域を8等分するように計算点を取り，2次中心差分を用いた差分方程式を，出発値ゼロを与え，緩和法で解くことにする．式 (2.2a) の差分方程式は2次中心差分を用いれば次のようになる．

$$(u_{i-1} - 2u_i + u_{i+1})/h^2 = 0 \quad (i = 1, 2, \dots, 7) \quad (2.3)$$

ただし $h \equiv x_i - x_{i-1}$ は計算点の間隔である．内点 x_1, x_2, \dots, x_7 に対するこれらの差分方程式に境界条件 (2.2b) を考慮したものは次の 7 元連立 1 次方程式になる．

$$\begin{pmatrix} -2 & 1 & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ 0 & & & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1.0 \end{pmatrix} \quad (2.4)$$

この連立 1 次方程式の係数行列は優対角の条件式 (2.1) を満足する．

ここでは緩和法 (relaxation method) で解くことにする．緩和法ではまず u_i に適当な予測値を与える．予測値はもちろん解に近いものを与えるべきであるが，ここではあえて $u_i = 0$ ($i = 1, 2, \dots, 7$) を与えることにする．またひとつの試みとして，基本的な点緩和 (point relaxation) で修正計算を行うことにする．点緩和ではそれまでに分かっている u_j の値をもとに差分方程式の残差 (residual)

$$r_i = u_{i-1} - 2u_i + u_{i+1} \quad (2.5)$$

の値を計算し，この残差値をゼロにするように u_i に修正値 (correction)

$$\delta u_i = r_i / 2 \quad (2.6)$$

を加える．

連立 1 次方程式を点緩和で解くには，端の点から順番に点緩和の操作をすべての内点に対して行い，更にこれを反復する．点緩和の操作を一通り行うことを掃引 (sweep) という．係数行列が優対角の条件を満足するときには，このような点緩和による掃引を反復することによって収束解を得ることができる．解の収束は，例えば最大残差値 $\max |r_i^{(k)}| \equiv r_{\max}$ と十分小さい正数 ε を比較することによって判定され， $r_{\max} < \varepsilon$ になった時点の解 $u_i^{(k)}$ をもって求める解とする．

手計算ではまずすべての内点の残差値を計算し，通常は大きい残差値のところから点緩和を始める．1 つの計算点に修正値 1 を加えたときの残差の変動を示したものを緩和パターン (relaxation pattern) という．この場合緩和パターンは図 2.1(a) のようになる．つまりその点の残差が -2 変動すると同時に隣接点の残差も $+1$ 変動する．手計算では緩和パターンをよく頭に入れておくことが必要で，1 つの計算点に修正値を加えるつど緩和パターンをもとにその点と近傍点の残差値を修正し，図 2.1(b) に示すように記録していく．すな

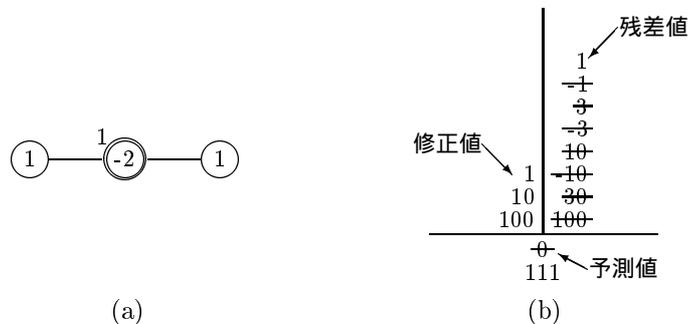


図 2.1: 緩和パターン (a) と修正値，残差値の記録 (b)

る．第2点については，次の点過緩和 (point over-relaxation) が考えられている．この方法では修正値は

$$\delta u_i = \alpha r_i / 2 \quad (2.7)$$

によって大きめにとられる．ただし α は緩和係数 (relaxation factor) あるいは加速係数 (acceleration parameter) などと呼ばれるもので， $1 < \alpha < 2$ である． α の最適値は長波長の残差の波に対しては大きめに，隣合う残差の符号が異なるような短波長の残差の波に対しては1よりも小さくなる．しかしながら短波長の波の成分はすぐに緩和され長波長のものが残るので，長波長のものを重視し $\alpha = 1.5$ 程度に取るのが無難である．緩和係数の最適値を数値実験で求め，最適値を選ぶことによって計算がいかに加速されるかを述べている論文は多数ある．しかし最適値は問題，数値解法，初期残差によっても違ってくるので，例えば $\alpha = 1.35$ のような推奨値はある問題には最適であっても普遍妥当性があるとは言えない．いくつかの α の値を用意し循環して使う方法も提案されている．優対角の条件が満足されず解が発散するような場合には， α を1よりも小さく取りむりやり収束させることもある．この場合の α は減衰係数 (damping factor) と呼ばれる．

図 2.2(b) は，(a) 図と同じ2点境界値問題を，緩和係数 $\alpha = 1.5$ の点過緩和法を用い，掃引の向きを交互にかえて計算したものである．同じ計算点で点過緩和を2度続けて行うと緩和係数の効果が失われる．点過緩和の計算を境界のところで折り返すときにはこのことに十分注意しなければならない．(b) 図は点過緩和法による4回の掃引過程を示したものである．(a) 図と同じ掃引回数で得られた (b) 図の結果は，かなり厳密解に近づいており，(a) 図に比べ格段の改善のあとが見られる．緩和係数を1.5に取り修正値を50%大きく取るとは，計算を単に50%加速するのではなく，相乗作用によって大幅に加速するのである．

2.1.2 Poisson 方程式の Dirichlet 境界値問題

次に Poisson 方程式の Dirichlet 境界値問題を取上げる．

$$u_{xx} + u_{yy} = -2 \quad (0 < x < 1.0, 0 < y < 0.8) \quad (2.8a)$$

$$u(0, y) = u(1.0, y) = u(x, 0) = u(x, 0.8) = 0 \quad (2.8b)$$

ここでは，正方形格子，格子間隔 $h = 0.1$ を用いることにする．式 (2.8a) の差分方程式は2次中心差分を用いれば次のようになる．

$$(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{ij})/h^2 = -2 \quad (2.9)$$

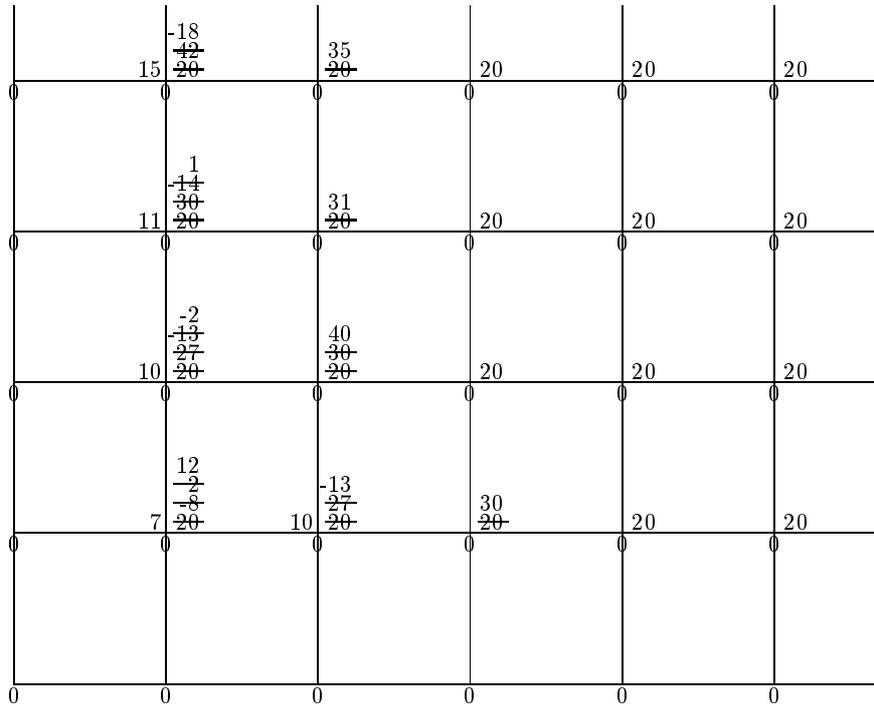
ただし $u_{ij} = u_{i,j} = u(x_i, y_j)$ である．この方程式の残差と点過緩和法の修正値は次のようになる．

$$r_{ij} = u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{ij} + 2h^2 \quad (2.10)$$

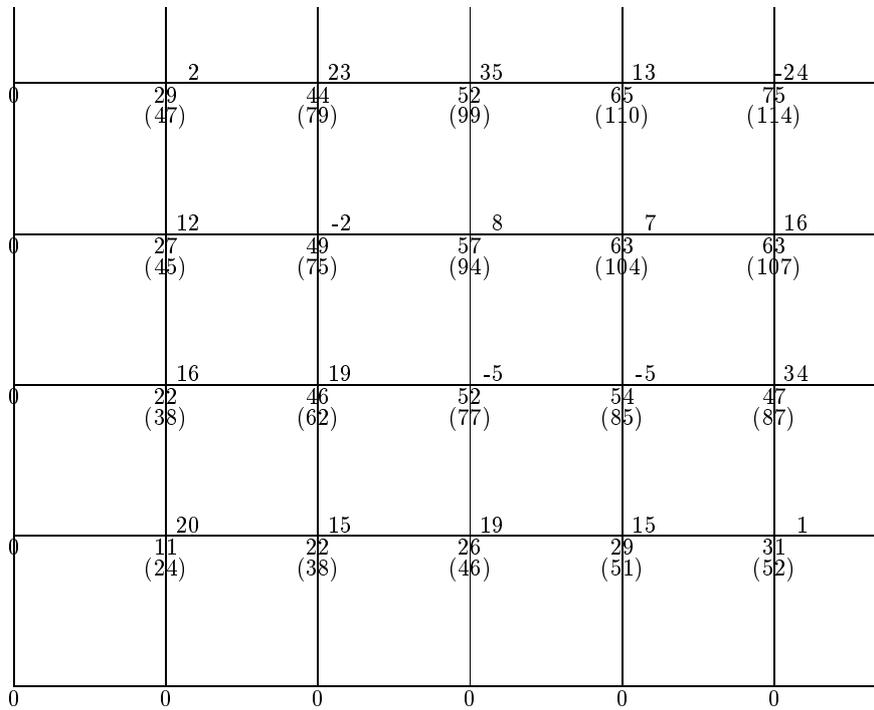
$$\delta u_{ij} = \alpha r_{ij} / 4 \quad (2.11)$$

また緩和パターンは下記の図 2.4(a) のようになる．問題の対称性より， $0 \leq i \leq 5$ ， $0 \leq j \leq 4$ の範囲だけを対象に計算を進めることにする．

予測値 $u_{ij}^{(0)} = 0$ を与えれば初期残差値は式 (2.10) より一様に $r_{ij}^{(0)} = 0.02$ になる．緩和係数 1.5 を用いた点過緩和で，奇数回には左下から右上に，偶数回には右上から左下に向かって掃引することにする．点過緩和を2度続けてしないという考えから，偶数回の掃引では境界の隣接点と対称軸上の点を除外する．図 2.3(a) は計算を始めた直後の状態，(b) 図は4回掃引した後の状態である (数値は1000倍されている)．(a) 図では，まず u_{11} に式 (2.11) により修正値 $\delta u_{11} = 7$ を加え，図 2.4(a) の緩和パターンにより残差値 $r_{11} = -8$ ，



(a)



(b)

図 2.3: 点過緩和による計算

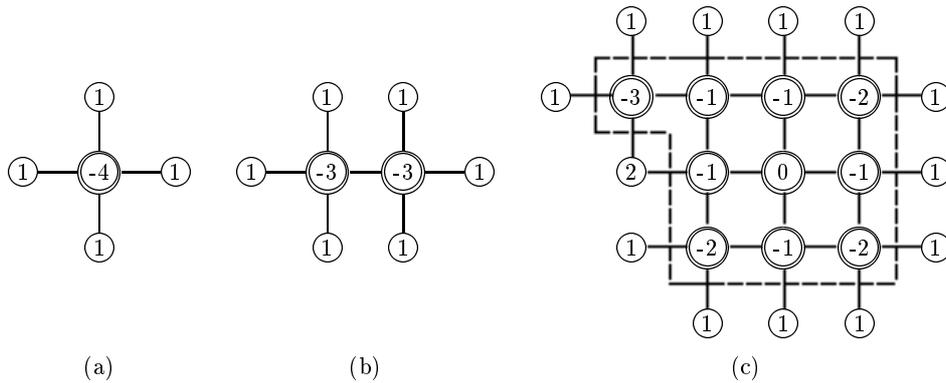


図 2.4: Laplace 演算子の緩和パターン

$r_{21} = r_{12} = 27$ に書替えている． u_{12} 以降の緩和も同様に行われるが， u_{13} の緩和では対称性により $r_{14} = 42$ のように 2 倍修正する．この図は u_{21} まで緩和したものである．(b) 図には 4 回掃引後の残差値 r_{ij} と修正された変数値 u_{ij} を示す．計算は着実に収束へ向かっているものの，() 内の収束解に達するまでにはこの先かなりの計算が必要である．

緩和法 (relaxation method) は英国の Southwell らによって 1935 年頃までに確立された楕円型偏微分方程式の数値解法である．Southwell の緩和法は，点過緩和とともに“ブロック緩和”を用い，粗い格子から部分的に細かくなっている格子に計算を進めるものである．ブロック緩和 (block relaxation) は，残差の同符号の複数の隣接格子点に同じ修正値を加え，まとめて緩和するものである．2 つの隣接格子点に修正値 1 を加えれば，そのときの残差の変動すなわち緩和パターンは図 2.4(b) のようになる．これは (a) 図の緩和パターンを 2 つ重ね合わせたものである．また例えば (c) 図に破線で示すブロックに対しては緩和パターンは同図のようになる．このような緩和パターンをすべて記憶することはできないが，緩和パターンは簡単に作ることができ，またなくてもブロック緩和はできる．(c) 図の緩和パターンでは，ブロック内の格子点ではその点から外側の点に繋がる格子線の数だけマイナスになり，また外側の点では内側の点に繋がる格子線の数だけプラスになる．ブロック内の格子点に加える適切な修正値は，ブロック内の残差の平均値をゼロにするもので，それはブロック内の残差の和をブロックの内外を繋ぐ格子線の数で割ったものである．ブロック緩和を併用することにより計算の労力は大幅に節減される．緩和計算は，始めに大きいブロックに対してブロック緩和

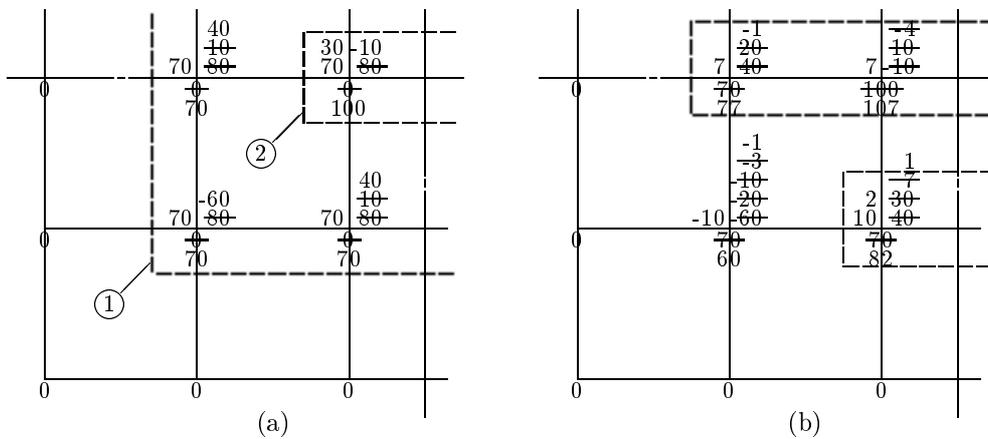
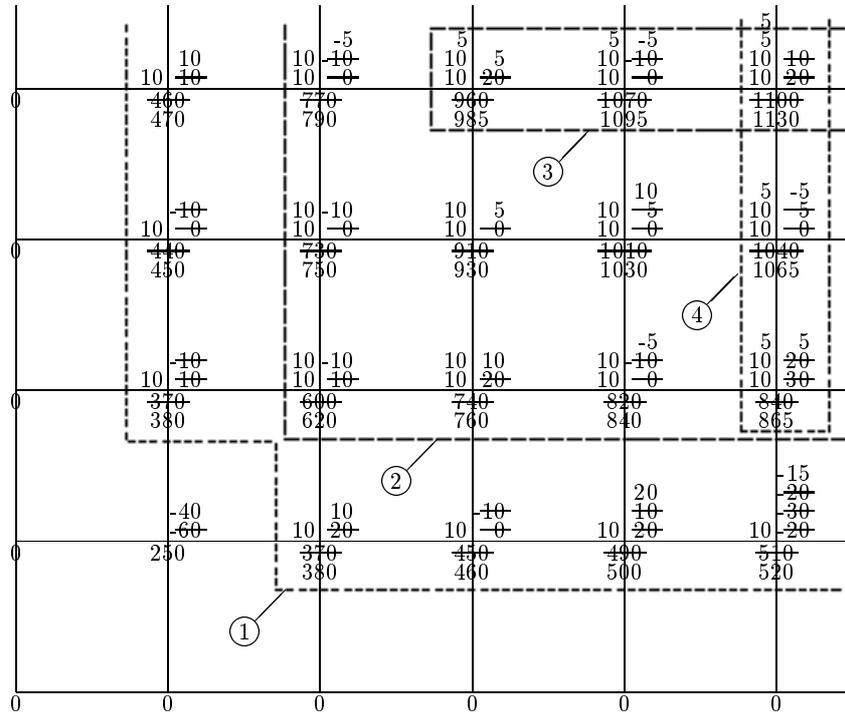
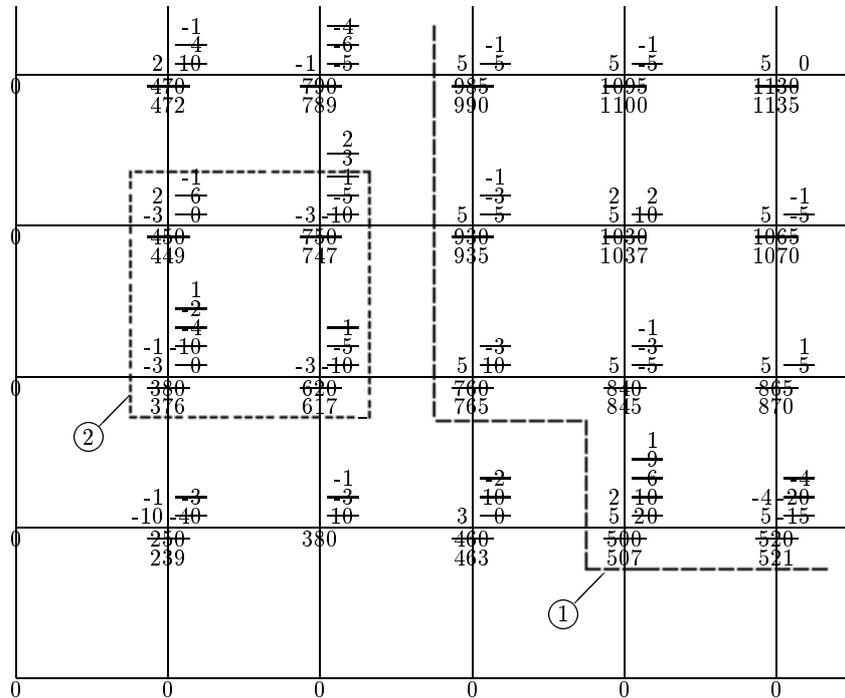


図 2.5: Southwell の緩和法による計算 (粗い格子)



(a)



(b)

図 2.6: Southwell の緩和法による計算 (細かい格子)

を行い計算領域全体の残差値をできるだけゼロに近付けてから、通常の点緩和にブロック緩和を織り交ぜて行うべきである。

ここでは上記の問題をまず $h = 0.2$ の粗い格子に対して大まかに解き、次に $h = 0.1$ の細かい格子に対して、粗い格子の結果をもとに予測値を与え、より正確に解くこととする。粗い格子に対しては、予測値 $u_{ij}^{(0)} = 0$ を与えれば残差値は一様に 0.08 になる。粗い格子では連立 1 次方程式の未知数の数が少ないので直接法で解くこともできるが、ここでは緩和法で解くことにする。図 2.5 は緩和法による計算の過程を示したものである(数値は 1 000 倍されている)。今簡単に説明すれば、始めに (a) 図に示す 2 つのブロック緩和を順に行う。ブロックの修正値は、省略されている対称の部分も一所に考えれば $960 \div 14 \approx 70$ 、またブロックの修正値は $160 \div 6 \approx 30$ になる。この 2 つのブロック緩和により長波長の残差は消え、以後の計算は (b) 図に示すように楽になる。なおここでは修正値 $\delta u_{11} = -10$ 、 $\delta u_{21} = 10$ 、 $\delta u_{12} = \delta u_{22} = 7$ 、 $\delta u_{21} = 2$ を順に加え、破線で囲んだ 2 つブロック緩和を行っている²。

細かい格子の計算の過程は図 2.6 に示すようになる(図の数値は 10 000 倍されている)。その予測値は粗い格子の結果をもとに、補間公式から求めることもできるが、ここでは差分方程式

$$(u_{i+1,j+1} + u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j-1} + u_{i,j})/2h^2 = -2 \quad (2.12)$$

と式 (2.9) から求めている。初期残差値は各格子点の右上の下端に示すように小さく、その値は粗い格子の計算をしない場合の 200 から平均 2.9 に減少している。それでもなお初めにブロック緩和を行うことは必要である。(a) 図は 4 つのブロック緩和を順に行ったもので、ブロックにそれぞれ修正値 10, 10, 5, 5 を加えている。

続いて (b) 図では、まず最大残差 $r_{11} = -40$ を修正値 $\delta u_{11} = -10$ を加えて点緩和し、次に 2 つのブロックにそれぞれ修正値 5, -3 を加えブロック緩和を行い、平均的残差値を 0 に十分近付けてから、次のように修正値を加え緩和を行っている。

$$\begin{aligned} \delta u_{51} &= -4, \delta u_{31} = 3, \delta u_{43} = 2, \delta u_{41} = 2, \\ \delta u_{13} &= \delta u_{14} = 2, \delta u_{11} = \delta u_{12} = -1, \delta u_{24} = -1 \end{aligned}$$

収束解を各格子点の下に示す。最終残差値は、この桁数ではこれ以上緩和することのできないところまで小さくなっている。図 2.5 と 2.6 の計算量を図 2.3 のものと比較すれば、Southwell の緩和法が単純な点過緩和法に対していかに優れているかが分かるであろう。

上記の手計算による数値例は、Southwell 緩和法の 2 つの手法、すなわち粗い格子から細かい格子に進む方法とブロック緩和が計算量を大幅に軽減することを示している。しかしながらこれら二つの手法がそのままコード化されているわけではない。粗い格子から細かい格子に進む方法は多重格子法 (multigrid method) となって洗練化されている。多重格子法については次章に詳しく述べるが、その主な相違点は、粗い格子から細かい格子に一方的に進むのではなく、いくつかの粗さの異なる格子の間を何度も往復することと、細かい格子の残差をもとに粗い格子の残差を求め、粗い格子の修正値をもとに細かい格子の修正値を求めることである。他方ブロック緩和については、人は一目で同符号で大きい残差のブロックを見出せるが、コンピュータはこのような一種のパターン認識は苦手でコード化がむずかしい。そのため筆者の知る限りではブロックは通常残差の状態とは無関係に画一的に取られることになり、そのため計算量の大幅軽減には繋がっていない。

²これらの図を描くための pL^AT_EX₂ ϵ のプログラムは、実際の緩和計算の手順通りに作られている。

2.2 連立1次方程式の反復解法

連立1次方程式

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (2.13a)$$

または

$$Ax = b \quad (2.13b)$$

が優対角の条件式 (2.1) を満足するものとする。各式をそれぞれの主対角要素で割算すれば、

$$\begin{pmatrix} 1 & \hat{a}_{12} & \cdots & \hat{a}_{1n} \\ \hat{a}_{21} & 1 & \cdots & \hat{a}_{2n} \\ \vdots & \vdots & & \vdots \\ \hat{a}_{n1} & \hat{a}_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{pmatrix} \quad (2.14a)$$

または

$$\hat{A}x = \hat{b} \quad (2.14b)$$

のようになる。ただし $\hat{a}_{ij} = a_{ij}/a_{ii}$, $\hat{b}_i = b_i/a_{ii}$ である。Gauss 反復法または Jacobi 法といわれる解法は、通常 予測値 $x^{(0)} = 0$ を与え、上式を書換えた式

$$x^{(k)} = \hat{b} - (\hat{A} - I)x^{(k-1)} \quad (2.15)$$

によって反復計算を行い、その収束解を求めるものである。解の収束は、例えば $\max |x_i^{(k)} - x_i^{(k-1)}| < \varepsilon$, ε は十分小さい正数、によって判定される。係数行列が正則で優対角ならばこのようにして収束解を得ることができる。この方法では配列 a, b, x のほかに配列 x0 を用意し、第 k 近似計算では第 k-1 近似値 $x^{(k-1)}$ を配列 x0 に移してから、第 k 近似値 $x^{(k)}$ を計算し配列 x に記憶する。この方法は DO 文の中を並列処理することができるので、ベクトルコンピュータに向いている。

また Gauss-Seidel 法といわれる解法は、式 (2.14b) を書換えた式

$$x^{(k)} = \hat{b} - \hat{U}x^{(k-1)} - \hat{L}x^{(k)} \quad (2.16)$$

を用い反復計算を行うものである。ただし $\hat{A} = \hat{L} + I + \hat{U}$, I は単位行列, \hat{U} と \hat{L} は主対角要素ゼロの上下の三角行列である。式 (2.16) は、第 k 近似計算を x_1, x_2, \dots, x_n の順に行い、 $x_i^{(k)}$ を計算する時点では、 x_1, \dots, x_{i-1} はすでに第 k 近似値になっているが x_{i+1}, \dots, x_n はまだ第 k-1 近似値のままであることを意味する。この解法は、緩和の過程で x_i の値を逐一新しいものに替えていくことによって、Jacobi 法の収束性を改善したものであるが、これによって DO 文の中を並列処理することができなくなるので、スカラーコンピュータに適した方法といえる。

これら 2 つの解法はもとの連立 1 次方程式 (2.13b) に対してはそれぞれ次式によって反復計算を行うことになる。

$$\begin{aligned} x^{(k)} &= D^{-1}\{b - (A - D)x^{(k-1)}\} \\ x^{(k)} &= D^{-1}(b - Ux^{(k-1)} - Lx^{(k)}) \end{aligned}$$

ただし $A = L + D + U$, D は対角行列, U と L は主対角要素ゼロの上下の三角行列である. なお A の要素を a_{ij} とすれば L, D, U の要素は

$$l_{ij} = \begin{cases} a_{ij} & (j < i) \\ 0 & (j \geq i) \end{cases}, \quad d_{ij} = \begin{cases} a_{ij} & (j = i) \\ 0 & (j \neq i) \end{cases}, \quad u_{ij} = \begin{cases} a_{ij} & (j > i) \\ 0 & (j \leq i) \end{cases}$$

また D^{-1} は対角行列でその i 行目の要素は $1/a_{ii}$ である. 前節の点緩和法は A が疎行列の場合の Gauss-Seidel 法である.

Yacobi 過緩和法 (Jacobi overrelaxation) と Gauss-Seidel 過緩和法 (Gauss-Seidel overrelaxation) は, それぞれ Yacobi 法または Gauss-Seidel 法に緩和係数 α を導入し, 修正値を大きく取ることによって解の収束を加速するもので, その反復計算は次式によって行われる.

$$x^{(k)} = x^{(k-1)} - \alpha D^{-1} (Ax^{(k-1)} - b) \quad (2.17)$$

$$x^{(k)} = x^{(k-1)} - \alpha D^{-1} \{ (D+U)x^{(k-1)} + Lx^{(k)} - b \} \quad (2.18)$$

緩和係数については前節を参照されたい. 前節の点過緩和法は A が疎行列の場合の Gauss-Seidel 過緩和法である. Gauss-Seidel 過緩和法は, 緩和の過程で x の値を逐一新しいものに替えていくことによって, Jacobi 過緩和法の収束性を改善したものであるが, このことによって DO 文の中を並列処理できなくなる. そのため, 並列度にもよるが, ベクトルコンピュータでは Gauss-Seidel 過緩和法よりも Jacobi 過緩和法の方が CPU 時間が少なくなることも多い.

2.3 SOR 法

SOR (successive over-relaxation) 法は優対角行列の連立 1 次方程式を点過緩和で解くもので, Gauss-Seidel 過緩和法とも言われる. ここでは 2 階楕円型方程式の境界値問題を考え, その差分方程式が

$$c_{ij}^1 u_{i-1,j} + c_{ij}^2 u_{i+1,j} + c_{ij}^3 u_{i,j-1} + c_{ij}^4 u_{i,j+1} + c_{ij}^0 u_{ij} = f_{ij} \quad (2.19)$$

で与えられるものとする. この方程式の残差と点過緩和の修正値は次のようになる.

$$r_{ij} = c_{ij}^1 u_{i-1,j} + \dots + c_{ij}^0 u_{ij} - f_{ij} \quad (2.20a)$$

$$\delta u_{ij} = -\alpha r_{ij} / c_{ij}^0 \quad (2.20b)$$

ただし $u_{ij}^{(k)} = u_{ij}^{(k-1)} + \delta u_{ij}$, k は反復回数, また α は緩和係数で $1 < \alpha < 2$ に取られる.

長方形領域 $0 \leq i \leq i_f$, $0 \leq j \leq j_f$ を考え, すべての境界 ($i = 0, i_f$, $j = 0, j_f$) に Dirichlet 条件を与えることにする. 次にこの Dirichlet 問題を SOR 法で解く FORTRAN 95/90 で書かれたプログラムを示す.

```
!*****
! Solution of Dirichlet problem of elliptic pde by SOR-method
!*****
SUBROUTINE SOR001(u,c,f,if,jf,al,k,kf,rmax,e) !pde=partial differential equation
DIMENSION u(0:if,0:jf),c(0:if,0:jf,0:4),f(0:if,0:jf)
IF(al<1.)al=1.5
kf=MAX0(kf,1000)
e=AMAX1(e,1.E-5)
i1=if-1; j1=jf-1
k=0; 10 k=k+1
```

```

rmax=0.
ie=i1; IF(MOD(k,2)==0)ie=i1-2
je=j1; IF(MOD(k,2)==0)je=j1-2
DO ii=1,ie; i=ii; IF(MOD(k,2)==0)i=i1-i           !alternate sweep direction
  DO jj=1,je; j=jj; IF(MOD(k,2)==0)j=j1-j         !alternate sweep direction
    r = 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)-a1*r/c(i,j,0)                   !point over-relaxation
    rmax = AMAX1(rmax,ABS(r))                         !max residual
  ENDDO
ENDDO
IF(rmax>e.AND.k<kf)                                GOTO 10           !decide convergence
END

```

このサブルーチンは、配列 u に u の予測値と Dirichlet 条件の境界値、配列 c と f に差分方程式の係数 c とその右辺 f の値、引数 $a1$, kf , e にそれぞれ緩和係数 α , 最大反復数 k_f , 小さい正数 ε を入力して計算を実行すれば、 u に解 $u_{ij}^{(k)}$, k と $rmax$ に最終反復回数 k とそのときの残差値 $r_{max} = \max |r_{ij}^{(k)}|$ が出力される。このプログラムでは、反復ごとに掃引方向を反転させ、また境界またはその隣接点で重ねて点過緩和しないように配慮している。左側境界の Dirichlet 条件 $u_{0j} = f_{0j}$ の課される境界では、メインプログラムで $c_{0j}^1 = c_{0j}^2 = c_{0j}^3 = c_{0j}^4 = 0$, $c_{0j}^0 = 1$ と u_{0j} の値を与えることが必要である。他の境界も同じである。計算は $r_{max} \leq \varepsilon$ または $k = k_f$ で終了する。解の収束状況は r_{max} と k から判定される。

次に参考までにこのサブルーチンのメインプログラムの簡単な例を示そう。Poisson 方程式の境界値問題

$$u_{xx} + u_{yy} = -2, \quad (0 < x < 1, 0 < y < 1) \quad (2.21a)$$

$$u(x, 0) = 0, \quad u(x, 1) = 1, \quad u(0, y) = u(1, y) = y \quad (2.21b)$$

を考える。ここではこの問題を、1 辺の長さ $h = 0.1$ の正方形格子を用い、出発値 0 を与え、サブルーチン SOR001 を引用して解く。解曲面 $u(x, y)$ は、右辺 0 の Laplace 方程式の場合には平面 $u(x, y) = y$ になるが、この場合には平均曲率が負であるからこの平面よりも全体に特に中央部の盛り上がった曲面になる。

```

program MAIN
DIMENSION u(0:10,0:10),c(0:10,0:10,0:4),f(0:10,0:10)
DATA u/121*0./,c/121*-4.,121*1.,121*1.,121*1.,121*1./,f/121*-.02/
DATA if,jf/10,10/,a1,kf,e/0.,1,0./
OPEN(20,FILE='OUTPUT.dat')
FORALL(i=0:if)u(i,jf)=1.
FORALL(j=0:jf-1)u(0,j)=.1*j
FORALL(j=0:jf-1)u(if,j)=.1*j
CALL SOR001(u,c,f,if,jf,a1,kf,e)
WRITE(20,'(I3,F12.6)')k,rmax
DO j=jf,0,-1
  WRITE(20,'(11F8.3)')(u(i,j),i=0,10)
ENDDO
CLOSE(20)
STOP
END

```

問1 上記の SOR 法のサブルーチン SOR001 を、左側境界 ($i = 0$) に Dirichlet 条件、右側境界 ($i = i_f$) に Neumann 条件、上下境界 ($j = 0, j_f$) に周期境界条件 $u_{ij_f} = u_{i0} + \phi$, $(\partial u / \partial y)_{ij_f} = (\partial u / \partial y)_{i0}$ が課される場合に適用できるように書換えよ。ただし ϕ は定数である。

右側境界点 (i_f, j) における Neumann 条件 $\partial u / \partial x = g$ の差分式は

$$(u_{i_f+1,j} - u_{i_f-1,j}) / 2h = g_{i_f j}$$

ただし h は格子間隔である。この点の差分方程式は、式 (2.19) の計算領域外の $u_{i_f+1,j}$ をこの Neumann 条件の式を用いて消去したもので、形式的に次のように書かれる。

$$c_{i_f j}^1 u_{i_f-1,j} + c_{i_f j}^3 u_{i_f,j-1} + c_{i_f j}^4 u_{i_f,j+1} + c_{i_f j}^0 u_{i_f j} = f_{i_f j} \quad (2.22)$$

メインプログラムで $c_{i_f j}^1$ の値を変更し $c_{i_f j}^2 = 0$ と置く。また周期条件の課される上下の境界の差分方程式は形式的に次のように書かれる。

$$c_{i_0}^1 u_{i-1,0} + c_{i_0}^2 u_{i+1,0} + c_{i_0}^3 u_{i,j_f-1} + c_{i_0}^4 u_{i1} + c_{i_0}^0 u_{i0} = f_{i0}, \quad (2.23a)$$

$$c_{i_j f}^1 u_{i-1,j_f} + c_{i_j f}^2 u_{i+1,j_f} + c_{i_j f}^3 u_{i,j_f-1} + c_{i_j f}^4 u_{i1} + c_{i_j f}^0 u_{i j_f} = f_{i j_f} \quad (2.23b)$$

下方境界では領域外の $u_{i,-1}$ の代わりに $u_{i,j_f-1} - \phi$ を使い、メインプログラムで右辺に $c_{i_0}^3 \phi$ を加え、同様に上方境界では領域外の u_{i,j_f+1} の代わりに $u_{i1} + \phi$ を使い、メインプログラムで右辺に $-c_{i_j f}^4 \phi$ を加えることが必要である。次に上記のサブルーチン SOR001 にならって作られたこの混合境界値問題のプログラムを示す。

```

!*****
! Solution of mixed bound value problem of elliptic pde by SOR-method
!*****
SUBROUTINE SOR002(u,c,f,if,jf,phi,al,k,kf,rmax,e)
DIMENSION u(0:if,0:jf),c(0:if,0:jf,0:4),f(0:if,0:jf)
IF(al<1.)al=1.5
kf=MAX0(kf,1000)
e=AMAX1(e,1.E-5)
k=0; 10 k=k+1
rmax=0.
ie=if; IF(MOD(k,2)==0)ie=if-2
DO ii=1,ie; i=ii; IF(MOD(k,2)==0)i=if-i           !alternate sweep direction
  ip1=MIN0(i+1,if)                                !Neumann cond
  DO j=1,jf
    jp1=j+1; IF(j==jf)jp1=1                        !periodic cond
    r = 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/c(i,j,0)                 !point over-relaxation
    rmax = AMAX1(rmax,ABS(r))                       !max residual
  ENDDO
  u(i,0)=u(i,jf)-phi
ENDDO
IF(rmax>e.AND.k<kf)                                GOTO 10           !decide convergence
END

```

このプログラムに関して注意すべき点をあげれば、Dirichlet 条件は周期条件と両立するようにメインプログラムで与えられなければならない。Neumann 条件の課されている右側境界では計算領域外の $u_{i_f+1,j}$ を引いてはならない。また右側境界の上下の点には Neumann 条件と共に周期条件も与えなければならない。上記の式 (2.21) で与えられる Poisson 方程式の Dirichlet 問題と、その境界条件を

$$u(0, y) = y, \quad u_x(1, y) = 0, \quad u(x, 1) = u(x, 0) + 1, \quad u_y(x, 1) = u_y(x, 0) \quad (2.24)$$

に変えた混合境界値問題の計算結果を次に示す．計算格子は 11×11 で，収束判定には $\varepsilon = 10^{-5}$ を用いた．過緩和係数 (加速パラメータ) α と最終反復数 k の関係を次表に示す．Dirichlet 問題では一般的な $\alpha = 1.5$ で良く収束するが，この混合境界値問題ではそれよりも大きめの α で良く収束することが分かる．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．

α	Dirichlet 問題	混合境界値問題
1.00	119	810
1.25	75	504
1.50	50	296
1.75	74	138
1.80	98	112
2.00	diverge	diverge

2.4 ADI法

長方形の計算領域を考える． i 一定格子点列または j 一定格子点列に対する緩和計算をまとめて同時に行うことを線緩和 (line relaxation) という．線緩和の式は差分方程式 (2.19) に対しては次のようになる．

i 一定格子点列の線緩和：

$$c_{ij}^3 u_{i,j-1} + \beta c_{ij}^0 u_{ij} + c_{ij}^4 u_{i,j+1} = c_{ij}^1 u_{i-1,j} - (1 - \beta) c_{ij}^0 u_{ij} - c_{ij}^2 u_{i+1,j} + f_{ij} \quad (2.25a)$$

$$(j = 1, 2, \dots)$$

j 一定格子点列の線緩和：

$$c_{ij}^1 u_{i-1,j} - \beta c_{ij}^0 u_{ij} + c_{ij}^2 u_{i+1,j} = c_{ij}^3 u_{i,j-1} - (1 - \beta) c_{ij}^0 u_{ij} - c_{ij}^4 u_{i,j+1} + f_{ij} \quad (2.25b)$$

$$(i = 1, 2, \dots)$$

ただし β は係数である．これらの式は，1つの格子点列上の残差をまとめて緩和するもので，右辺に既知の値を代入すれば3重対角行列の連立1次方程式になるので，Gauss 消去法によって容易に解くことができる．ADI (alternating direction implicit, iterative) 法³は， i 一定格子点列の線緩和の i 方向掃引と， j 一定格子点列の線緩和の j 方向掃引を交互に反復することによって，疎行列の連立1次方程式を解くものである．ADI法は連立1次方程式を解かなければならないという煩わしさはあるが，2つの方向に線緩和するので安定しており，SOR法と並んで広く用いられている．

次に，2階楕円型微分方程式の差分方程式 (2.19) に対しその Dirichlet 問題を ADI法で解く FORTRAN 95/90 のプログラムを示す．

```
!*****
! Solution of Dirichlet problem of elliptic pde by ADI-method
!*****
SUBROUTINE ADI001(u,c,f,i1,if,j1,jf,a,b,n,bt,k,kf,rmax,e)
DIMENSION u(0:if,0:jf),c(0:if,0:jf,0:4),f(0:if,0:jf),a(n,3),b(n)
```

³Peaceman, D.W. and Rachford, H.H., Jr., The numerical solution of parabolic and elliptic difference equations, *J. Soc. Indust. Appl. Mathematics*, **3**(1955), 28-41.

この論文では Poisson 方程式に対する次の差分方程式が解かれている．

$$u_{i-1,j} - (2 + \rho)u_{ij} + u_{i+1,j} = -b^2(u_{i,j-1} - (2 - \rho)u_{ij} + u_{i,j+1}) - \Delta x^2 f_{ij} \quad (i = 1, 2, \dots)$$

$$u_{i,j-1} - (2 + \rho)u_{ij} + u_{i,j+1} = -\frac{1}{b^2}(u_{i-1,j} - (2 - \rho)u_{ij} + u_{i+1,j}) - \Delta y^2 f_{ij} \quad (j = 1, 2, \dots)$$

ただし $b = \Delta x / \Delta y$ である．

```

IF(bt<.75)bt=1.
kf=MAXO(kf,1000)
e=AMAX1(e,1.E-5)
k=0; 10 k=k+1
cycle_1: DO i=0,if                                     !i-sweep
  IF(i==0.OR.i==if)CYCLE cycle_1                     !Dirichlet cond
  im1=MAXO(i-1,0); ip1=MINO(i+1,if)                 !Neumann conds
  FORALL(j=0:jf)
    a(j+1,1) = c(i,j,3)
    a(j+1,2) = c(i,j,0)*bt                           !setup coefs
    a(j+1,3) = c(i,j,4)
    b(j+1) = f(i,j)-c(i,j,1)*u(im1,j)-c(i,j,2)*u(ip1,j) & !setup rhs
              -(1.-bt)*c(i,j,0)*u(i,j)
  ENDFORALL
  CALL GAUSS3(a,b,n,jf+1)                             !solve linear eqs
  FORALL(j=0:jf)u(i,j) = b(j+1)
ENDDO cycle_1
k=k+1
cycle_2: DO j=0,jf;                                   !j-sweep
IF(j==0.OR.j==jf)CYCLE cycle_2                       !Dirichlet cond
  jm1=MAXO(j-1,0); jp1=MINO(j+1,jf)                 !Neumann cond
  FORALL(i=0:if)
    a(i+1,1) = c(i,j,1)
    a(i+1,2) = c(i,j,0)*bt                           !setup coefs
    a(i+1,3) = c(i,j,2)
    b(i+1) = f(i,j)-c(i,j,3)*u(i,jm1)-c(i,j,4)*u(i,jp1) & !setup rhs
              -(1.-bt)*c(i,j,0)*u(i,j)
  ENDFORALL
  CALL GAUSS3(a,b,n,if+1)                             !solve linear eqs
  FORALL(i=0:if)u(i,j) = b(i+1)
ENDDO cycle_2
rmax=0.
DO i=0,if; im1=MAXO(i-1,0); ip1=MINO(i+1,if)
  DO j=0,jf; jm1=MAXO(j-1,0); jp1=MINO(j+1,jf)
    r = 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)
    rmax = AMAX1(rmax,ABS(r))                         !max residual
  ENDDO
ENDDO
IF(rmax>e.AND.k<kf) GOTO 10                           !decide convergence
END

```

```

!***** Subroutine GAUSS3 *****
! Solution of system of linear eqs with tri-diagonal matrix
SUBROUTINE GAUSS3(a,b,n1,n)
DIMENSION a(n1,3),b(n1)
DO k=1,n-1
  b(k)=b(k)/a(k,2); a(k,3)=a(k,3)/a(k,2)
  b(k+1)=b(k+1)-a(k+1,1)*b(k)
  a(k+1,2)=a(k+1,2)-a(k+1,1)*a(k,3)
ENDDO; b(n)=b(n)/a(n,2)
DO k=n-1,1,-1
  b(k)=b(k)-a(k,3)*b(k+1)
ENDDO
END

```

このサブルーチンは、 i 方向掃引、 j 方向掃引、残差 r_{ij} による収束判定からなる。このサブルーチンの引用に際しては、配列 u に予測値 $u_{ij}^{(0)}$ と Dirichlet 条件の境界値、配列 c と f にそれぞれ差分方程式の係数 c_{ij}^1, \dots と右辺 f_{ij} の値、また bt , kf , e にそれぞれ係数 β , 最大反復数 k_f , 収束判定用の小さい正数 ε を入力する。また a と b は 3 重対角行列の連立 1 次方程式の係数と右辺を入れる作業用配列で、 n に $\max(i_f+1, j_f+1)$ を入力する。左側境界の Dirichlet 条件 $u_{0j} = f_{0j}$ は、メインプログラムで $c_{0j}^1 = c_{0j}^2 = c_{0j}^3 = c_{0j}^4 = 0$, $c_{0j}^0 = 1$, u_{0j} の値, $f_{0j} = u_{0j}$ のように与えられる。残りの境界についても同様である。 i 方向掃引では u_{i0}, \dots, u_{i,j_f} の線緩和を $i = 1$ から $i_f - 1$ まで行い、3 重対角行列の連立 1 次方程式の計算は 1.4 節に述べたサブルーチン GAUSS3 を用いて行われる。なおこの際に ADI001 では j が 0 から始まるのに対し GAUSS3 では k が 1 から始まることに留意しなければならない。 j 方向掃引についても同様である。計算終了時には、 u に解 $u_{ij}^{(k)}$, k , r_{\max} にそれぞれ最終掃引回数 k とそのときの最大残差値 $r_{\max} = \max |r_{ij}^{(k)}|$ が出力される。解の収束状況は k と r_{\max} から判定できる。

ところで SOR 法の緩和係数 α と ADI 法の係数 β はどのように対応するのであろうか。ここでは簡単な Laplace 演算子、最も緩和し難い一様残差値、一定修正値を加える線緩和を考えることにする。ADI 法の線緩和の式 (2.25a) から残差値 $r_{ij}^{(k-1)}$ は

$$r_{ij}^{(k-1)} = u_{i,j-1} - \beta 4u_{ij} + u_{i,j+1} + u_{i-1,j} - (1-\beta)4u_{ij} + u_{i+1,j} - f_{ij}$$

となる。今連立 1 次方程式を解いて緩和すれば、その直後の残差値はゼロになり、また修正値 δu は仮定により一定で次式が成立する。

$$\begin{aligned} 0 &= (u_{i,j-1} + \delta u) - \beta 4(u_{ij} + \delta u) + (u_{i,j+1} + \delta u) + u_{i-1,j} - (1-\beta)4u_{ij} + u_{i+1,j} - f_{ij} \\ &= r_{ij}^{(k-1)} + (2-4\beta)\delta u \end{aligned}$$

一方 i 一定格子点列をブロック緩和するときの修正値 δu は

$$\delta u = \alpha^{\text{block}} r_{ij}^{(k-1)} / 2$$

である。これら 2 つの式から次の係数 α と β の概略の対応付けができる。

$$\beta = \frac{1}{2} \left(1 + \frac{1}{\alpha^{\text{block}}} \right) \quad (2.26)$$

α^{block}	β
0	∞
0.5	1.5
1	1
1.5	0.8333
2	0.75

なおこの線緩和の α^{block} と点過緩和の α は、これらの係数が 0 より大で緩和が行われ、2 より小で収束という観点からは同等ともいえる。しかし緩和計算した後に周りの点の緩和によって残差値が元の状態に戻されるという観点からはブロック緩和の緩和係数は、 $\alpha^{\text{block}} = 1 + (\alpha - 1)/2$ のように 1 に近く選ぶべきものであろうか。

問2 計算領域 $0 \leq i \leq i_f$, $0 \leq j \leq j_f$ において、左側境界 ($i = 0$) に Dirichlet 条件、上下境界 ($j = 0, j_f$) の $1 \leq i < i_0$ の部分に周期条件 $u_{i,j_f} = u_{i0} + \phi$, $(\partial u / \partial y)_{i,j_f} = (\partial u / \partial y)_{i0}$, 残りの部分に Dirichlet 条件、右側境界 $i = i_f$ に Neumann 条件が課されているものとする。これらの境界条件を満足する差分方程式 (2.19)

の解をADI法で求めよ。ただし境界点 (i_f, j) 上では, Neumann 条件を考慮した差分方程式

$$c_{i_f j}^1 u_{i_f-1, j} + c_{i_f j}^3 u_{i_f, j-1} + c_{i_f j}^4 u_{i_f, j+1} + c_{i_f j}^* u_{i_f-2, j} + c_{i_f j}^0 u_{i_f j} = f_{i_f j} \quad (2.27)$$

を用いるものとする。

先にこの楕円型微分方程式の混合境界値問題を解くADI法のプログラムを示す。

```

*****
! Solution of mixed bound value problem of elliptic pde by ADI-method
*****
SUBROUTINE ADI002(u, c, f, i0, i1, if, j1, jf, a, b, n, bt, k, kf, rmax, e)
DIMENSION u(0:if,0:jf), c(0:if,0:jf,0:4), f(0:if,0:jf), a(n,3), b(n), r(0:if,0:jf)
IF(bt<.75)bt=1.
kf=MAX0(kf,1000)
e=AMAX1(e,1.E-5)
k=0; 10 k=k+1
cycle_1: DO i=0,if;                                     ! i-sweep
  IF(i==0)CYCLE cycle_1                                 ! ib1=1
  im1=i-1; ip1=i+1; IF(i==if)ip1=if-2                 ! ibn=2
  FORALL(j=0:jf)
    a(j+1,1) = c(i,j,3)
    a(j+1,2) = c(i,j,0)*bt                             ! setup coef matrix
    a(j+1,3) = c(i,j,4)
    b(j+1) = f(i,j)-c(i,j,1)*u(im1,j)-c(i,j,2)*u(ip1,j) & ! setup rhs
              -(1.-bt)*c(i,j,0)*u(i,j)
  ENDFORALL
  IF(i<i0)THEN; CALL GAUSSM(a,b,n,jf+1,3,3)           ! solve linear eqs
  ELSE;        CALL GAUSSM(a,b,n,jf+1,1,1)
  ENDIF
  FORALL(j=0:jf)u(i,j) = b(j+1)
ENDDO cycle_1
                                         GOTO 11

k=k+1
cycle_2: DO j=0,jf; jml=j-1; jpl=j+1                 ! j-sweep
  DO i=0,if
    IF(i<i0)THEN; IF(j==0)jml=jf-1; IF(j==jf)jpl=1   ! ib1=ibn=3
  ELSE;        IF(j==0)jml=0; IF(j==jf)jpl=jf         ! ib1, ibn=1
  ENDIF
  a(i+1,1) = c(i,j,1)
  a(i+1,2) = c(i,j,0)*bt                             ! setup coef matrix
  a(i+1,3) = c(i,j,2)
  b(i+1) = f(i,j)-c(i,j,3)*u(i,jml)-c(i,j,4)*u(i,jpl) & ! setup rhs
              -(1.-bt)*c(i,j,0)*u(i,j)
  ENDDO
  CALL GAUSSM(a,b,n,if+1,1,2)                         ! solve linear eqs
  FORALL(i=0:if)u(i,j) = b(i+1)
ENDDO cycle_2
11 rmax=0.
DO i=1,if; im1=i-1; ip1=i+1
  DO j=0,jf; jml=j-1; jpl=j+1
    IF(i<i0)THEN; IF(j==0)jml=jf-1; IF(j==jf)jpl=1   ! ib1=ibn=3
  ELSE;        IF(j==0)jml=0; IF(j==jf)jpl=jf         ! ib1, ibn=1
  ENDIF
  IF(i==if)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,jml)+c(i,j,4)*u(i,jpl) &

```

```

      +c(i,j,0)*u(i,j)-f(i,j)
      rmax = AMAX1(rmax,ABS(r(i,j)))           !max residual
    ENDDO
  ENDDO
  IF(rmax>e.AND.k<kf)           GOTO 10       !decide convergence
  END

!***** Subroutine GAUSSM *****
! Solution of linear eqs with modified tri-diagonal matrix
! based on Gaussian elimination
SUBROUTINE GAUSSM(a,b,n1,n,ib1,ibn)
DIMENSION a(n1,3),b(n1)
IF(ib1==2)THEN                                     ! ib1=2
  a(1,1)=a(1,1)/a(1,2); a(2,3)=a(2,3)-a(2,1)*a(1,1)
ENDIF
DO k=1,n-1
  IF(ib1==3.AND.k==n-2)a(k,3)=a(k,3)+a(k,1)       ! ib1=ibn=3
  b(k)=b(k)/a(k,2); a(k,3)=a(k,3)/a(k,2)
  b(k+1)=b(k+1)-a(k+1,1)*b(k)
  a(k+1,2)=a(k+1,2)-a(k+1,1)*a(k,3)
  IF(ibn==2.AND.k==n-2)THEN                       ! ibn=2
    b(n)=b(n)-a(n,3)*b(k); a(n,1)=a(n,1)-a(n,3)*a(k,3)
  ENDIF
  IF(ib1==3)THEN                                   ! ib1=ibn=3
    IF(k<n-2)THEN
      a(k,1)=a(k,1)/a(k,2); a(k+1,1)=-a(k+1,1)*a(k,1)
    ENDIF
    IF(k>1.AND.k<n-2)THEN
      b(n)=b(n)-a(n,3)*b(k); a(n,1)=a(n,1)-a(n,3)*a(k,1)
      a(n,3)=-a(n,3)*a(k,3)
    ELSEIF(k==n-2)THEN
      b(n)=b(n)-a(n,3)*b(k); a(n,1)=a(n,1)-a(n,3)*a(k,3)
    ENDIF
  ENDIF
ENDIF
ENDDO
b(n)=b(n)/a(n,2)
DO k=n-1,1,-1
  b(k)=b(k)-a(k,3)*b(k+1)
  IF(ib1==3.AND.k<n-2)b(k)=b(k)-a(k,1)*b(n-1)       ! ib1=ibn=3
ENDDO
  IF(ib1==2)b(1)=b(1)-a(1,1)*b(3)                 ! ib1=2
END

```

このサブルーチンは前記のサブルーチン ADI001 に倣って作られている。相違点は式 (2.27) の Neumann 条件と周期条件の部分である。副サブルーチン GAUSSM は、基本的には GAUSS3 と同様に 3 重対角行列の連立 1 次方程式を Gauss 消去法で解くもので、これらの境界条件に対処できるようになっている。プログラム中の $ib1$ は左方または下方境界、 ibn は右方または上方境界を意味し、また $=1$ は Dirichlet 条件、 $=2$ は Neumann 条件、 $=3$ は周期条件を意味する。左側境界の Dirichlet 条件は $ib1=1$ 、右側境界の Neumann 条件は $ibn=2$ 、上下境界の周期条件は $ib1=ibn=3$ で与えられる。以下にはコメント $ib1=2$ 、 $ibn=2$ 、 $ib1=ibn=3$ の付いている部分、Neumann 条件と周期条件の処理について詳しく説明する。

Neumann 条件の課されている境界点 $ib1=2$ の計算から説明する。図 2.7 (a) は ADI 法で解かれる連立 1 次方程式の係数行列と右辺を示したものである。3 重対角行列の連立 1 次方程式を Gauss 消去法で解くときには、その初めのステップ $k=1$ で、係数行列の第 1 列を 1, 0 にするために ● で示す 4 箇所を計算する

が、式(2.27)では係数 a_{13} があるので、 \odot で示す 2 箇所の計算を追加することが必要になる。プログラムでは係数 a_{13} を空いている配列要素 $a(1,1)$ に記憶している。ステップ $k=2$ 以降は 3 重対角行列の場合と変わらないが、ガウス消去法の後半の計算で最後に a_{13} が 0 でないことによる計算が追加される。境界点 $ibn=2$ の計算では、係数 $a_{n,n-2}$ があるので、ステップ $k=n-2$ で (b) 図に示すように \odot で示す 2 箇所の計算を追加することが必要になる。 $a_{n,n-2}$ も空いている $a(n,3)$ に記憶している。他は 3 重対角行列の場合と同じである。

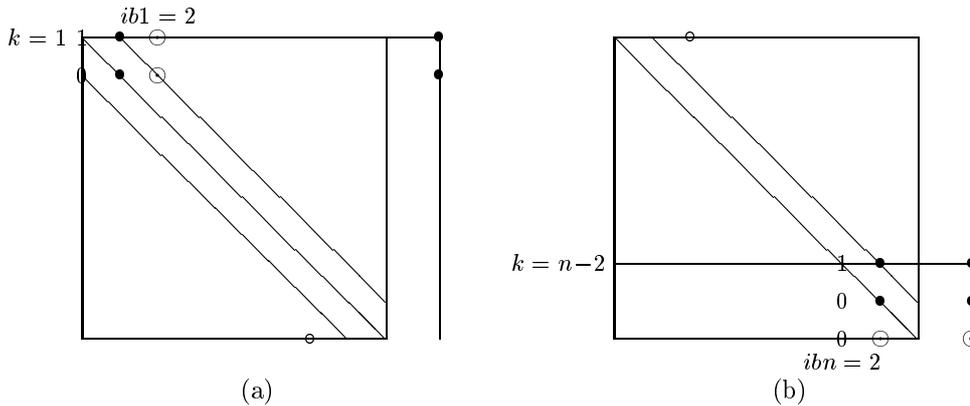


図 2.7: GAUSSM の Neumann 条件

次に周期条件の課されている境界点 $ib1=ibn=3$ の計算について説明する。ガウス消去法のステップ $k=1$ の計算では、図 2.8 (a) に示すように、3 重対角行列の連立 1 次方程式の \bullet で示す 4 箇所の計算に加え、係数 $a_{1,n-1}$ があるために \odot で示す 2 箇所の計算が必要になる。プログラムでは係数 $a_{1,n-1}$ は空いている配列要素 $a(1,1)$ に記憶し、 $a_{2,n-1}$ は用済みの $a(2,1)$ に記憶する。 $k=2$ では、(b) 図に示すように、 $a_{1,n-1}$ に由来する 2 箇所の計算の他に、 a_{n2} を 0 にするために更に 3 箇所の計算が加わる。 a_{n2} は空いている $a(n,3)$ に記憶し、 a_{n3} も同じところに上書きする。一般の $k=k$ では、(c) 図に示すように、3 重対角行列の計算に \odot で示す 5 箇所の計算が加わる。 $k=n-2$ では、(d) 図に示すように、初めに $a_{1,n-1}$ に由来する $a(n-2,1)$ を $a(n-2,3)$ に加え、3 重対角行列の計算を行い、それから $a_{n,n-2}$ を 0 にするために 2 箇所の計算を行う。 $k=n-1$ と $k=n$ の計算は 3 重対角行列の場合と同じである。ガウス消去法の後半では、 $a_{i,n-1}$ ($i=n-3, \dots, 1$) が 0 でないことによる計算が追加される。

最後にサブルーチン ADI002 を用いる簡単なメインプログラムを示そう。Laplace 方程式の境界値問題

$$u_{xx} + u_{yy} = 0 \quad (-1 < x < 1, 0 < y < 1) \quad (2.28a)$$

$$\begin{aligned} u(0, y) &= y+1, & u_x(0, y) &= -1, \\ u(x, 1) &= u(x, 0)+1, & u_y(x, 1) &= u_y(x, 0), & (-1 \leq x < 0) \\ u(x, 0) &= -x, & u(x, 1) &= 1-x & (0 \leq x \leq 1) \end{aligned} \quad (2.28b)$$

を解くことにする。格子間隔 $h = 0.1$ の正方形格子を用いる。右側境界の Neumann 条件が差分式

$$(u_{i_f-2,j} - 6u_{i_f-1,j} + 3u_{i_f,j} + 2u_{i_f+1,j})/6h = -1.0$$

与えられるときに、この境界上の差分方程式は次のようになる。

$$-0.5u_{i_f-2,j} + 4u_{i_f-1,j} + u_{i_f,j-1} + u_{i_f,j+1} - 5.5u_{i_f,j} = 0.3$$

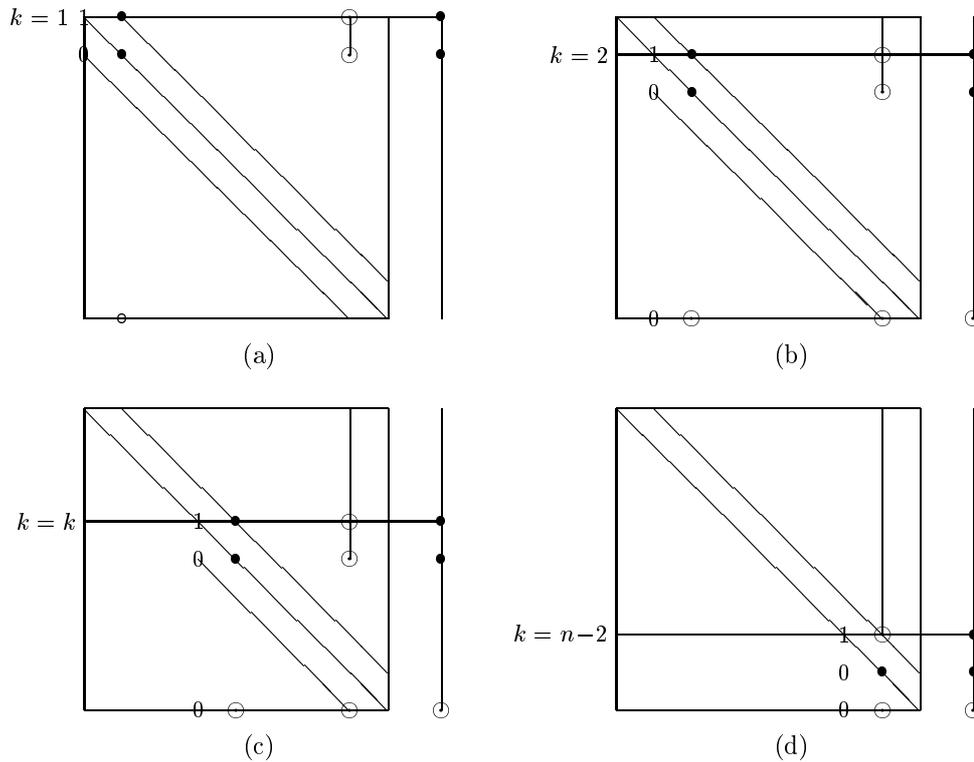


図 2.8: GAUSSM の周期条件

また上下の周期境界上の差分方程式は .

$$u_{i-1,0} + u_{i+1,0} + u_{i,j_f-1} + u_{i1} - 4u_{i0} = 1$$

$$u_{i-1,j_f} + u_{i+1,j_f} + u_{i,j_f-1} + u_{i1} - 4u_{i j_f} = -1$$

ここでは予測値 $u_{ij} = 0$ を与えこの問題を解く . なおこの問題の厳密解は $u(x, y) = y - x$ である .

```

program MAIN
DIMENSION u(0:20,0:10),c(0:20,0:10,0:4),f(0:20,0:10),a(21,3),b(21),r(0:20,0:10)
DATA u/231*0./,c/231*-4.,231*1.,231*1.,231*1.,231*1./,f/231*0./
DATA i0,i1,if,j1,jf,n/10,19,20,9,10,21/,phi,bt,kf,e/1.,0.,1,0./
OPEN(20,FILE='OUTPUT.DAT')
FORALL(i=1:i0-1)
  f(i,0)=c(i,0,3)*phi; f(i,jf)=-c(i,jf,4)*phi          !ib1=ibn=3
ENDFORALL
FORALL(i=i0:if)
  u(i,0)=-.1*(i-i0); f(i,0)=u(i,0)                      !ib1=1
  c(i,0,0)=1.; c(i,0,1)=0.; c(i,0,2)=0.; c(i,0,3)=0.; c(i,0,4)=0.
  u(i,jf)=1.-.1*(i-i0); f(i,jf)=u(i,jf)                !ibn=1
  c(i,jf,0)=1.; c(i,jf,1)=0.; c(i,jf,2)=0.; c(i,jf,3)=0.; c(i,jf,4)=0.
ENDFORALL
FORALL(j=0:jf)
  u(0,j)=1.+1*j; f(0,j)=u(0,j)                          !ib1=1
  c(0,j,0)=1.; c(0,j,1)=0.; c(0,j,2)=0.; c(0,j,3)=0.; c(0,j,4)=0.
ENDFORALL
FORALL(j=1:j1)
  c(if,j,0)=-5.5; c(if,j,1)=4.; c(if,j,2)=-.5; f(if,j)=.3  !ibn=2

```

```

ENDFORALL
CALL ADI002(u,c,f,i0,i1,if,j1,jf,a,b,n,bt,k,kf,rmax,e)
WRITE(20,'(I4,F12.6)')k,rmax
DO j=jf,0,-1
  WRITE(20,'(21F8.3)')(u(i,j),i=0,if)
ENDDO
CLOSE(20)
STOP
END

```

最後に式 (2.21) の Dirichlet 問題をサブルーチン ADI001 を用いて計算した結果と、式 (2.28) の混合境界値問題を ADI002 を用いて計算したものを示す。計算格子は 11×11 または 21×11 で、収束判定は $\varepsilon = 10^{-5}$ に対して行った。次表に係数 β と最終反復数 k の関係を示す。

β	Dirichlet 問題	混合境界値問題		
	ADI	ADI	i -sweep only	j -sweep only
0.75	22	42	417	diverge
0.80	20	44	29	43
1.00	56	152	135	152
1.25	104	286	273	291
1.50	156	422	411	428

混合境界値問題の場合には、参考までに i 掃引のみと j 掃引のみの場合も示してある。大きい β に対してはこれら 3 方法は同程度に収束し、片方の掃引のみで十分といえる。しかしながら小さい β では ADI 法は安定しているが片方の掃引のみの場合には発散の虞もある。

ここに示した ADI 法のプログラムは、連立 1 次方程式を $n \times 3$ 縮小配列を使って計算するところに一つの特徴がある。しかしそのために、やや理解しがたいサブルーチン GAUSSM を引用しなければならないことになった。なおこのサブルーチンは、各種の境界条件に対応できるもので、作り変える必要はない。ADI 法のプログラムを $n \times n$ 配列を使って連立 1 次方程式を計算するよう大らかに作れば、もちろん普通のガウス消去法のサブルーチンで済むことになる。

流れや波動の問題では、線過緩和 (successive line overrelaxation, SLOR) 法も良く用いられている。この解法では、線過緩和の掃引が物理量の対流・伝播する 1 つの主方向に対してのみ行われる。

最後に、ここに載せたいいくつかのプログラムはすべて FORTRAN 95/90 で compile し error や warning のないことを確認し、更に簡単な例題を解き妥当な結果の得られることを確かめている。しかしながらこれらのプログラムに誤りのないことを保障するものではない。