第３回 方程式の代数化と連立一次方程式の解法

筑波大学システム情報工学科
構造エネルギー工学専攻
田中聖三

基礎方程式の離散化 => 連立1次方程式 (flow)

Q. 数値解析とは？
A. 基礎方程式を離散化して最終的に得られる連立1次方程式を解く.

基礎方程式（１次元 Laplace 方程式）

$$\frac{\partial^2 \boldsymbol{u}}{\partial x^2} = 0 \qquad \begin{aligned} \boldsymbol{u} &= \hat{\boldsymbol{u}} \quad \text{on } \Gamma_d \\ \boldsymbol{u}_{,\mathbf{n}} &= \hat{\boldsymbol{h}} \quad \text{on } \Gamma_n \end{aligned}$$

↓ 差分法で離散化

差分方程式

$$\frac{\boldsymbol{u}_{i+1} - 2\boldsymbol{u}_i + \boldsymbol{u}_{i-1}}{\Delta x^2} + O(\Delta x^2 = 0$$

↓

差分方程式の代数表示

$$\mathbf{K}\boldsymbol{u} = 0$$

→ 計算領域モデル
境界条件

連立1次方程式

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$$

基礎方程式の離散化 => 連立1次方程式 (実作業)

差分方程式

境界条件の処理が必要

$$\frac{1}{\Delta x^2}\left(\boldsymbol{u}_0 - 2\boldsymbol{u}_1 + \boldsymbol{u}_2\right) = 0 \qquad \frac{1}{\Delta x^2}\left(\boldsymbol{u}_1 - 2\boldsymbol{u}_2 + \boldsymbol{u}_3\right) = 0$$

$$\frac{1}{\Delta x^2}\left(\boldsymbol{u}_2 - 2\boldsymbol{u}_3 + \boldsymbol{u}_4\right) = 0$$

$$\vdots$$

$$\frac{1}{\Delta x^2}\left(\boldsymbol{u}_{n-1} - 2\boldsymbol{u}_n + \boldsymbol{u}_{n+1}\right) = 0$$

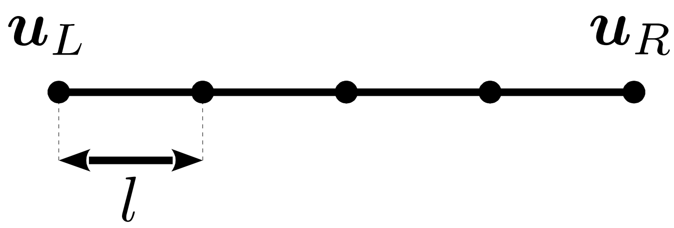
境界条件の処理が必要

差分方程式の代数表示

$$\frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \vdots \\ \boldsymbol{u}_{n-1} \\ \boldsymbol{u}_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

基礎方程式の離散化 => 連立1次方程式 (実作業)

問題設定



差分方程式の代数表示

$$\frac{1}{l^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \boldsymbol{u}_4 \\ \boldsymbol{u}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

境界条件の導入 (古典的、正統派なやり方)

$$\frac{1}{l^2}\begin{bmatrix} 1 & 0 & & & \\ 0 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 0 \\ & & & 0 & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \boldsymbol{u}_4 \\ \boldsymbol{u}_5 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_L/l^2 \\ -\boldsymbol{u}_L/l^2 \\ 0 \\ -\boldsymbol{u}_R/l^2 \\ \boldsymbol{u}_R/l^2 \end{bmatrix}$$

連立１次方程式の解法

直接法 (Direct Method)

•行列の変形や、逆行列に相当するものを計算する。
•Gauss の消去法、Gauss-Jordan 法、LU 分解等
•長所：
  • 安定 (とりあえず解ける)
  • 疎行列、密行列いずれにも適用可能
•短所：
  • 反復法よりも記憶容量、計算時間が必要
  • 桁落ち誤差、丸め誤差が大きくなる可能性

* 数値誤差は？

丸め誤差：

桁落ち誤差：

情報落ち誤差：

連立１次方程式の解法

反復法 (Iterative Method)

- 適当な初期解から繰り返し計算により真の解へ収束させる。
- 長所：
  - 直接法よりもメモリ使用量、計算量が少ない
  - 並列計算に適している
- 短所：
  - 収束性が問題の影響を受けやすい (収束しない場合もある)
  - 収束性が重要 (前処理の導入)

定常法： 反復計算中に、解ベクトル以外の変数は変化しない。
Jacobi 法、Gauss-Seidel 法、SOR 法など。
単純だが、収束性能が悪い

非定常法： 拘束、最適化条件が加わる。
CG (Conjugate Gradient: 共役勾配法)
Bi-CG 法 (Bi-Conjugate Gradient: 双共役勾配法)
GMRES(Generalized Minimal RESidual: 一般化残差最小法)

直接法 1 : Gauss の消去法

連立 1 次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

を解を変えないように変形し、以下のような形 (上三角行列) に変形する。
(この変形を前進消去 (Forward Elimination) と言う。)

$$\begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a'_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix}$$

解は、後退代入 (Backward Substitution) により求められる。

直接法 1：Gauss の消去法

例題:

$$\begin{bmatrix} 2 & 2 & 4 \\ 1 & 3 & 2 \\ 3 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 18 \\ 13 \\ 14 \end{bmatrix} \quad \Rightarrow$$

直接法 1：Gauss の消去法

プログラム

```
    do i = 1, ndof-1  !Frontward Elimination
      ai = 1.d0 / a(i,i)
      do j = i+1, ndof
         cc = a(j,i) * ai
         a(j,i) = 0.0d0
         do k = i+1, ndof
           a(j,k) = a(j,k) - a(i,k) * cc
         enddo
         x(j) = x(j) - x(i) * cc
      enddo
    enddo
!
    do i = 1, ndof    !Backward Substitution
      i1 = ndof + 1 - i
      do j = i1+1, ndof
        x(i1) = x(i1) - a(i1,j) * x(j)
        a(i1,j) = 0.0d0
      enddo
      x(i1) = x(i1) / a(i1,i1)
      a(i1,i1) = 0.0d0
    enddo
```

直接法 2：Gauss-Jordan 法

連立 1 次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

を解を変えないように変形し、以下のような形 (単位対角行列) に変形する。

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix}$$

解は、b' である。

直接法 2：Gauss-Jordan 法

例題:

$$\begin{bmatrix} 2 & 2 & 4 \\ 1 & 3 & 2 \\ 3 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 18 \\ 13 \\ 14 \end{bmatrix} \quad \Rightarrow$$

直接法 2：Gauss-Jordan 法

プログラム

```
      do i = 1, ndof
!
         ai = 1.0d0 / a(i,i)
         x(i) = x(i) * ai
         do j = 1, ndof
            a(i,j) = a(i,j) * ai
         enddo
!
         do j = 1, ndof
           if( i == j ) cycle
           cc = a(j,i)
           do k = 1, ndof
             a(j,k) = a(j,k) - cc * a(i,k)
           enddo
           x(j) = x(j) - cc * x(i)
         enddo
!
      enddo
```

直接法 3：完全 LU 分解

連立 1 次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

を解を変えないように変形し、次頁のような形 (下三角、上三角行列) に変形する。

直接法 3 :完全 LU 分解

簡単のため 4x4 の正方行列とする.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$A = LU$ となる LU 行列を求める. L , U はそれぞれ下･上三角行列である.

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \qquad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

L, U の成分の計算

$a_{11} = 1 \cdot u_{11}$ $\quad a_{12} = 1 \cdot u_{12}$ $\quad a_{13} = 1 \cdot u_{13}$ $\quad a_{14} = 1 \cdot u_{14}$

$a_{21} = l_{21} \cdot u_{11}$ $\quad a_{22} = l_{21} \cdot u_{12} + 1 \cdot u_{22}$ $\quad a_{23} = l_{21} \cdot u_{13} + 1 \cdot u_{23}$ $\quad a_{24} = l_{21} \cdot u_{14} + 1 \cdot u_{24}$

$a_{31} = l_{31} \cdot u_{11}$ $\quad a_{32} = l_{31} \cdot u_{12} + l_{32} \cdot u_{22}$ $\quad a_{33} = l_{31} \cdot u_{13} + l_{32} \cdot u_{23} + 1 \cdot u_{33}$ . . . . .

. . . .

直接法：完全LU分解

例題：

$$\begin{bmatrix} 2 & 2 & 4 \\ 1 & 3 & 2 \\ 3 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 18 \\ 13 \\ 14 \end{bmatrix} \Rightarrow$$

直接法 3 : 完全 LU 分解

L, U の成分の計算

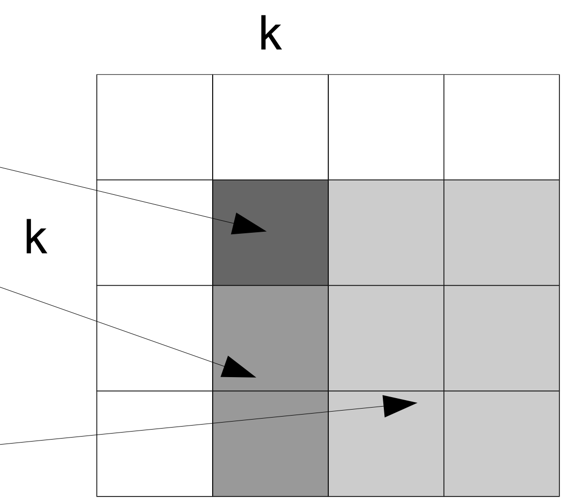
$$a_{11} = 1 \cdot u_{11} \quad a_{12} = 1 \cdot u_{12} \quad a_{13} = 1 \cdot u_{13} \quad a_{14} = 1 \cdot u_{14}$$

$$a_{21} = l_{21} \cdot u_{11} \quad a_{22} = l_{21} \cdot u_{12} + 1 \cdot u_{22} \quad a_{23} = l_{21} \cdot u_{13} + 1 \cdot u_{23} \quad a_{24} = l_{21} \cdot u_{14} + 1 \cdot u_{24}$$

$$a_{31} = l_{31} \cdot u_{11} \quad a_{32} = l_{31} \cdot u_{12} + l_{32} \cdot u_{22} \quad a_{33} = l_{31} \cdot u_{13} + l_{32} \cdot u_{23} + 1 \cdot u_{33} \quad \cdots\cdots$$

$$a_{41} = l_{41} \cdot u_{11} \quad a_{42} = l_{41} \cdot u_{12} + l_{42} \cdot u_{22} \quad a_{43} = l_{41} \cdot u_{13} + l_{42} \cdot u_{23} + l_{43} \cdot u_{33} \quad \cdots\cdots$$

```
do k = 1, ndof
  dtmp = 1.d0 / a(k,k)
  do i = k+1, ndof
    a(i,k) = a(i,k) * dtmp
  enddo
  do j = k+1, ndof
    dakj = a(k,j)
    do i = k+1, ndof
      a(i,j) = a(i,j) - a(i,k) * dakj
    enddo
  enddo
enddo
```



* a の下半分に L(対角は省略) , 上半分に U が格納されている.

直接法3：完全LU分解

解くべき連立1次方程式

$$LUx = b$$

$Ux = y$ と置くと， $Ly = b$ となり、y について前進代入により解く。

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

前進代入プログラム:

```
do i = 1, ndof  !Forward substitution for Ly=b
  dtmp = 0.d0
  do j = 1, i-1
    dtmp = dtmp + a(i,j) * x(j)
  enddo
  x(i) = x(i) - dtmp
enddo
```

直接法3：完全LU分解

解くべき連立1次方程式

$$LUx = b$$

$Ux = y$ と置くと，$Ly = b$ となり、y について前進代入により解く。

y が求まれば，$Ux = y$ となり、x について後退代入により解く。

後退代入プログラム:

```
do k = 1, ndof  !Backward substitution for Ux=y
  i = ndof - k + 1
  dtmp = 0.d0
  do j = i+1, ndof
    dtmp = dtmp + a(i,j) * x(j)
  enddo
  x(i) = ( x(i) - dtmp ) / a(i,i)
enddo
```

反復法1：Jacobi 法

連立1次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$Ax = b$$

$$\Downarrow$$

$$(L + D + U)x = b$$

L: 下三角、D: 対角、U: 上三角

$$\Downarrow$$

$$Dx = b - (L + U)x$$

$$\Downarrow$$

$$x^{k+1} = D^{-1}\left\{b - (L + U)x^k\right\}$$

Jacobi 法では、以下のようにk回目の反復解を用いて、k+1回目の推定値を求める。

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \cdots - a_{1n}x_n^{(k)}\right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} \cdots - a_{2n}x_n^{(k)}\right)$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$x_n^{(k+1)} = \frac{1}{a_{nn}}\left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} \cdots - a_{n\,n-1}x_{n-1}^{(k)}\right)$$

反復法１：Jacobi 法

k+1 ステップの推定値

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} a_{ij} x_j^{(k)} \right) \quad (1 \leq i \leq n)$$

プログラム

```
x(1:ndof) = 0.0d0
do k = 1, kmax

  do i = 1, ndof
    dtmp = 0.0d0
    do j = 1, ndof
      if(i==j) cycle
      dtmp = dtmp + a(i,j) * x(j)
    enddo
    xk(i) = (b(i) - dtmp) / a(i,i)
  enddo
  x(1:ndof) = xk(1:ndof)
enddo
```

反復法２：Gauss-Seidel 法

連立１次方程式

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$Ax = b$$

$$(L + D + U)x = b$$

L: 下三角、D: 対角、U: 上三角

$$Dx = b - (L + U)x$$

$$x^{k+1} = D^{-1}\left(b - Lx^{k+1} - Ux^k\right)$$

Gauss-Jordan 法では、以下のように k 回目の反復解を用いて、
ただし、Jacobi 法と違い、すでに求めた k+1 回目の推定値を使用する。

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \cdots - a_{1n}x_n^{(k)}\right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} \cdots - a_{2n}x_n^{(k)}\right)$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$x_n^{(k+1)} = \frac{1}{a_{nn}}\left(b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} \cdots - a_{n\,n-1}x_{n-1}^{(k+1)}\right)$$

反復法２：Gauss-Seidel 法

Jacobi 法

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \quad (1 \le i \le n)$$

Gauss-Seidel 法

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \quad (1 \le i \le n)$$

プログラム (Jacobi)

```
x(1:ndof) = 0.0d0
do k = 1, kmax
  do i = 1, ndof
    dtmp = b(i)
    do j = 1, ndof
      if(i==j) cycle
      dtmp = dtmp - a(i,j) * x(j)
    enddo
    xk(i) =  dtmp / a(i,i)
  enddo
  x(1:ndof) = xk(1:ndof)
enddo
```

プログラム (Gauss-Seidel)

```
x(1:ndof) = 0.0d0
do k = 1, kmax
  do i = 1, ndof
    dtmp = b(i)
    do j = 1, ndof
      if(i==j) cycle
      dtmp = dtmp - a(i,j) * x(j)
    enddo
    x(i) = dtmp / a(i,i)
  enddo

enddo
```

反復法3：SOR 法

SOR(Successive Over-Relaxation) 法

$$\tilde{x}_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$$

$$x_i^{(k+1)} = x_i^{(k)} + \omega \left(\tilde{x}_i^{(k+1)} - x_i^{(k+1)} \right)$$

プログラム (SOR)

```
x(1:ndof) = 0.0d0
do k = 1, kmax
  do i = 1, ndof
    dtmp = b(i)
    do j = 1, ndof
      if(i==j) cycle
      dtmp = dtmp - a(i,j) * x(j)
    enddo
    xt = dtmp / a(i,i)
    x(i) = x(i)+omega*(xt - x(i))
  enddo
enddo
```

Gauss-Seidel 法の修正量に加速パラメータ ω を乗じて修正量を拡大する。ω は 1 以上の値となるが、大きくしすぎると収束性能が悪くなる。
1.1 ~ 1.3 程度。
問題によっては、Gauss-Seidel 法 (ω=1.0) が良かったりもする。

反復法4：CG法

CG(Conjugate Gradient，共役勾配）法

y を厳密解とするとき、以下の式を最小とする x を求める:

$$\min((x-y),[A](x-y))$$

つまり、下記の $f(x)$ を最小とする　を求める。

$$f(x)=\frac{1}{2}(x,Ax)-(x,b)$$

CG法は任意の x_0 から始めて、$f(x)$ の最小値を逐次探索する。
探索方向 p が決まったとすると、

$$x_{k+1}=x_k+\alpha_k p_k$$

$f(x_{k+1})$ を最小とするには:

$$f(x_k+\alpha_k p_k)=\frac{1}{2}\alpha^2(p_k,Ap_k)-\alpha_k(p_k,b-Ax_k)+f(x_k)$$

$$\frac{\partial f(x_k+\alpha_k p_k)}{\partial \alpha_k}=0\Rightarrow \alpha_k=\frac{(p_k,b-Ax_k)}{(p_k,Ap_k)}=\frac{(p_k,r_k)}{(p_k,Ap_k)}$$

残差 r_k は以下の式により更新できる:

$$r_{k+1}=r_k-\alpha_k Ap_k$$

筑波大学
University of Tsukuba

反復法4：CG法

CG(Conjugate Gradient，共役勾配）法　(2/3)

探索方向は次の漸化式によって求める:

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

＊探索方向の更新を決めるβを決めたい。

ここで、収束した解が求まっているとすると、

$$y = x_{k+1} + \alpha_{k+1} p_{k+1}$$

以下のような直交条件がある:

$$(Ap_k, y - x_{k+1}) = 0$$

よって以下が成り立つ:

$$(Ap_k, y - x_{k+1}) = (Ap_k, \alpha_{k+1} p_{k+1}) = 0 \Rightarrow (p_{k+1}, Ap_k) = 0$$

$*$ $(p_{k+1}, Ap_k) = 0$　とは　p_kと　p_{k+1}が行列　Aに関して共役

$$(p_{k+1}, Ap_k) = (r_{k+1} + \beta_k p_k, Ap_k) = (r_{k+1}, Ap_k) + \beta_k (p_k, Ap_k) = 0$$

$$\Rightarrow \beta_k = \frac{-(r_{k+1}, Ap_k)}{(p_k, Ap_k)} = \frac{(r_{k+1}, r_{k+1})}{r_k, r_k}$$

反復法4：CG 法

CG(Conjugate Gradient，共役勾配）法

Initial guess

$$r_0 = b - Ax_0$$

$$p_1 = r_0$$

for k = 0, 1, …, do:

$$\alpha_k = \frac{(r_k, r_k)}{(p_k, Ap_k)}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k + \alpha_k A p_k$$

if $||r_{k+1}|| \le \varepsilon$ exit

$$\beta_k = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

end

連立１次方程式の解法まとめ

直接法 (Gauss の消去法 ,Gauss-Jordan, LU 分解)

使用制限:

対角項がゼロではない。(Pivoting による回避が必要)

「定常」反復法 (Jacobi, Gauss-Seidel, SOR)

使用制限:

対角優位 (第 i 行の対角項の絶対値がそれ以外の成分の絶対値の和より大きい)

$$|a_{ii}| > \sum_{j=1,j\neq i}^{n} |a_{ij}|$$

「非定常」反復法 (CG)

(CG 法の) 使用制限:

対称正定値行列 (Symmetric Positive Definite)

* 非対称行列では多項漸化式を用いる Bi-CG 法
残差を Krylov 部分空間内で最小化する GMRES 法などがある。

課題 ST1(提出日 :2016 年 11 月 11 日)

$$
\begin{bmatrix}
10 & 3 & 1 & 2 & 4 \\
3 & 18 & 2 & -1 & 5 \\
1 & 2 & 12 & 1 & 1 \\
2 & -1 & 1 & 9 & -4 \\
4 & 5 & 1 & -4 & 14
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5
\end{bmatrix}
=
\begin{bmatrix}
19 \\ 2 \\ 34 \\ -49 \\ 83
\end{bmatrix}
$$

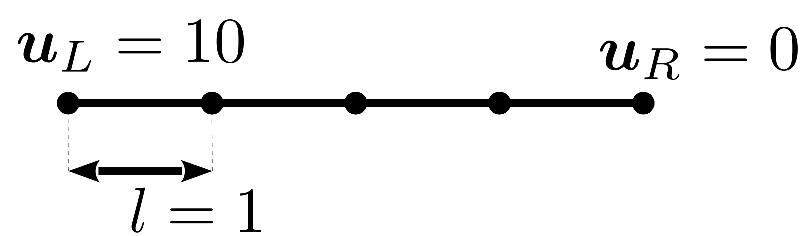
1. 上の連立 1 次方程式を Gauss の消去法、Gauss-Jordan 法、LU 分解法で解け。
2. 上の連立 1 次方程式を Jacobi 法、Gauss-Seidel 法、SOR 法で解け。
3. 本講義で解説した解法の中から一つ選び、プログラミングし動作確認せよ。
4. 下の問題を 3. のプログラムを利用し解け。

基礎方程式（1 次元 Laplace 方程式）

$$\frac{\partial^2 \boldsymbol{u}}{\partial x^2} = 0$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \quad \text{on } \Gamma_d$$

$$\boldsymbol{u}_{,\mathbf{n}} = \hat{\boldsymbol{h}} \quad \text{on } \Gamma_n$$



レポートは計算過程を詳細に記すこと。(Gauss の消去法なら、上三角化の過程など)
3. のプログラミングの言語はしていしない。コードと動作確認根拠を示すこと。