線形方程式の解法：反復法

中島 研吾
東京大学情報基盤センター
同　大学院情報理工学系研究科数理情報学専攻

数値解析（科目番号 500080）

直接法（Direct Method）

- Gaussの消去法，完全LU分解他
  - 行列の変形，逆行列に相当するものの計算

- 利点
  - 安定，幅広いアプリケーションに適用可能
    - Pivoting
  - 疎行列，密行列いずれにも適用可能
- 欠点
  - 反復法よりもメモリ，計算時間を必要とする
    - 密行列の場合，O（N^3）の計算量
  - 大規模な計算向けではない
    - O（N^2）の記憶容量，O（N^3）の計算量

反復法とは・・・

連立一次方程式

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

$$\mathbf{A} \qquad \mathbf{x} \qquad \mathbf{b}$$

初期解

$$\mathbf{x}^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{pmatrix}$$

適当な初期解 $\mathbf{x}^{(0)}$から始めて，繰り返し計算によって真の解に収束(converge)させていく

$$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots$$

反復法（Iterative Method）

- 定常（stationary）法
  - 反復計算中，解ベクトル以外の変数は変化せず
  - SOR，Gauss-Seidel，Jacobiなど
  - 概して遅い

$$\mathbf{Ax} = \mathbf{b} \Rightarrow$$
$$\mathbf{x}^{(k+1)} = \mathbf{Mx}^{(k)} + \mathbf{Nb}$$

- 非定常（nonstationary）法
  - 拘束，最適化条件が加わる
  - Krylov部分空間（subspace）への写像を基底として使用するため，Krylov部分空間法とも呼ばれる
  - CG（Conjugate Gradient：共役勾配法）
  - BiCGSTAB（Bi-Conjugate Gradient Stabilized）
  - GMRES（Generalized Minimal Residual）

反復法（Iterative Method）（続き）

- 利点
  - 直接法と比較して，メモリ使用量，計算量が少ない。
  - 並列計算には適している。

- 欠点
  - 収束性が，アプリケーション，境界条件の影響を受けやすい。
    - 収束しない（答えが得られない）可能性がある
  - 前処理（preconditioning）が重要。

- 定常反復法(1)
  - ヤコビ法, ガウス･ザイデル法
- 非定常反復法
  - 共役勾配法
- 定常反復法(2)
  - SOR法

ヤコビ法（Jacobi）

連立一次方程式

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

k回目の反復における解の推定値(k)

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix}$$

次のステップの推定値(k+1)

右辺の x_i は全て(k)における値

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

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

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

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

$$\rightarrow \quad \mathbf{x}^{(k+1)} = \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix}$$

何をやっているのか?

連立一次方程式：*n*個の未知数，*n*個の方程式

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

***n*個の方程式を一つずつ解いていく**
***i*番目の方程式を解くときは，$x_i^{(k+1)}$ のみが未知数，あとは既知の値として解く**

$$
a_{i1}x_1 + a_{i2}x_2 \cdots + a_{i\,i-1}x_{i-1} + a_{i\,i}x_i + a_{i\,i+1}x_{i+1} \cdots + a_{in}x_n = b_i
$$

$$
a_{i1}x_1^{(k)} + a_{i2}x_2^{(k)} \cdots + a_{i\,i-1}x_{i-1}^{(k)} + a_{i\,i}x_i^{(k+1)} + a_{i\,i+1}x_{i+1}^{(k)} \cdots + a_{in}x_n^{(k)} = b_i
$$

$$
x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - a_{i1}x_1^{(k)} - a_{i2}x_2^{(k)} \cdots - a_{i\,i-1}x_{i-1}^{(k)} - a_{i\,i+1}x_{i+1}^{(k)} \cdots - a_{in}x_n^{(k)}\right)
$$

ガウス・ザイデル法（Gauss-Seidel）

次のステップの推定値（$k+1$）

右辺の x_i は既に計算されている場合は（$k+1$）における値を使用（最新の値）

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - a_{14}x_4^{(k)} \cdots - a_{1n}x_n^{(k)}\right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - a_{24}x_4^{(k)} \cdots - a_{2n}x_n^{(k)}\right)$$

$$x_3^{(k+1)} = \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)} \cdots - a_{3n}x_n^{(k)}\right)$$

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

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

通常，ヤコビ法よりも速く収束する
（大体2倍くらいの速さと言われている）

$$\rightarrow \quad \mathbf{x}^{(k+1)} = \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix}$$

ヤコビ法とガウス・ザイデル法

連立一次方程式

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

*k*回目の反復における解の推定値(*k*)

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix}$$

次のステップの推定値(*k*+1)

ヤコビ法

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

ガウス・ザイデル法

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

幾何学的解釈：直線の交点

解くべき連立一次方程式

$$-2x_1 + x_2 = 0$$
$$x_1 + x_2 = 3$$

真の解

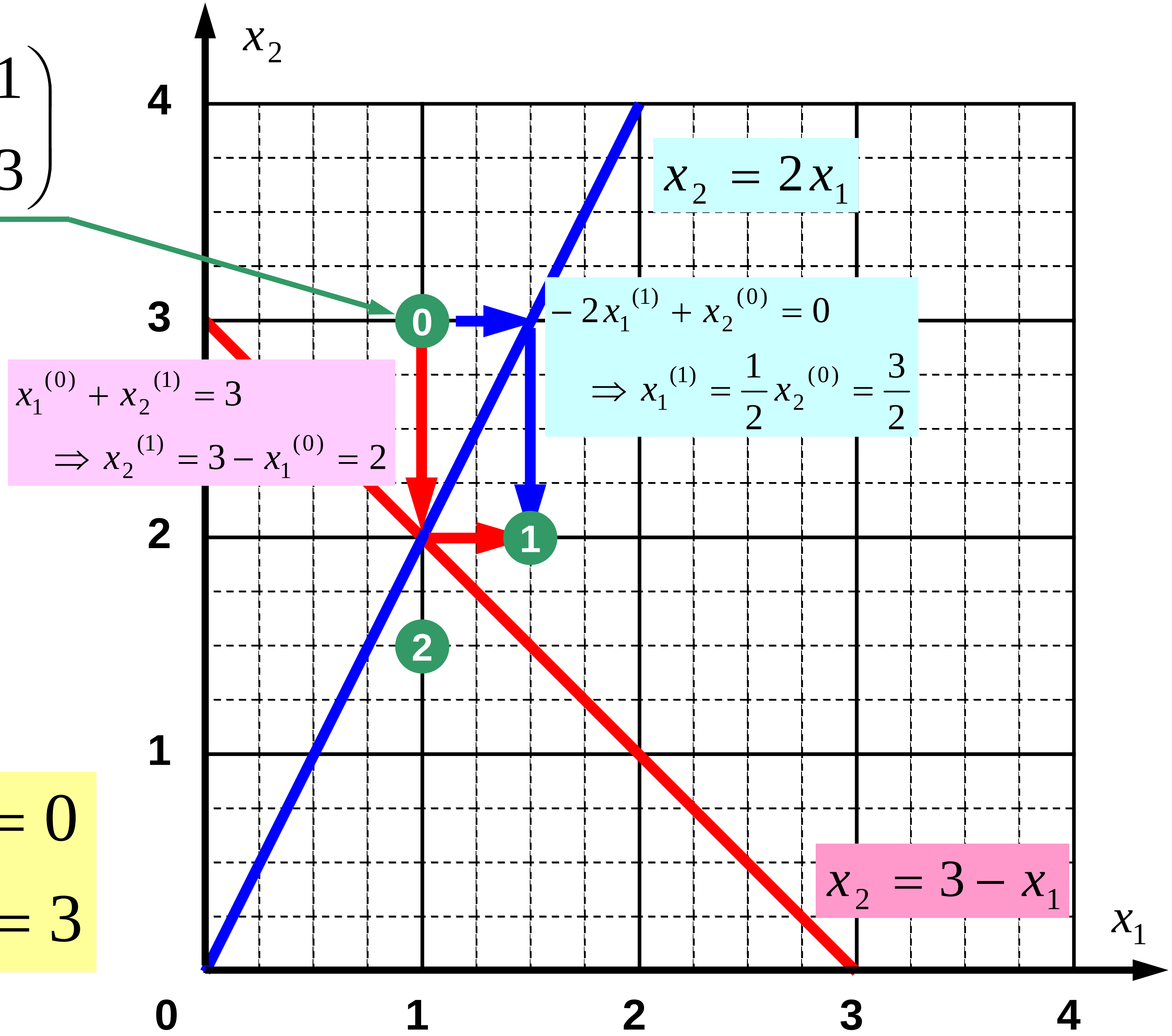
$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

初期解：例えば

$$\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

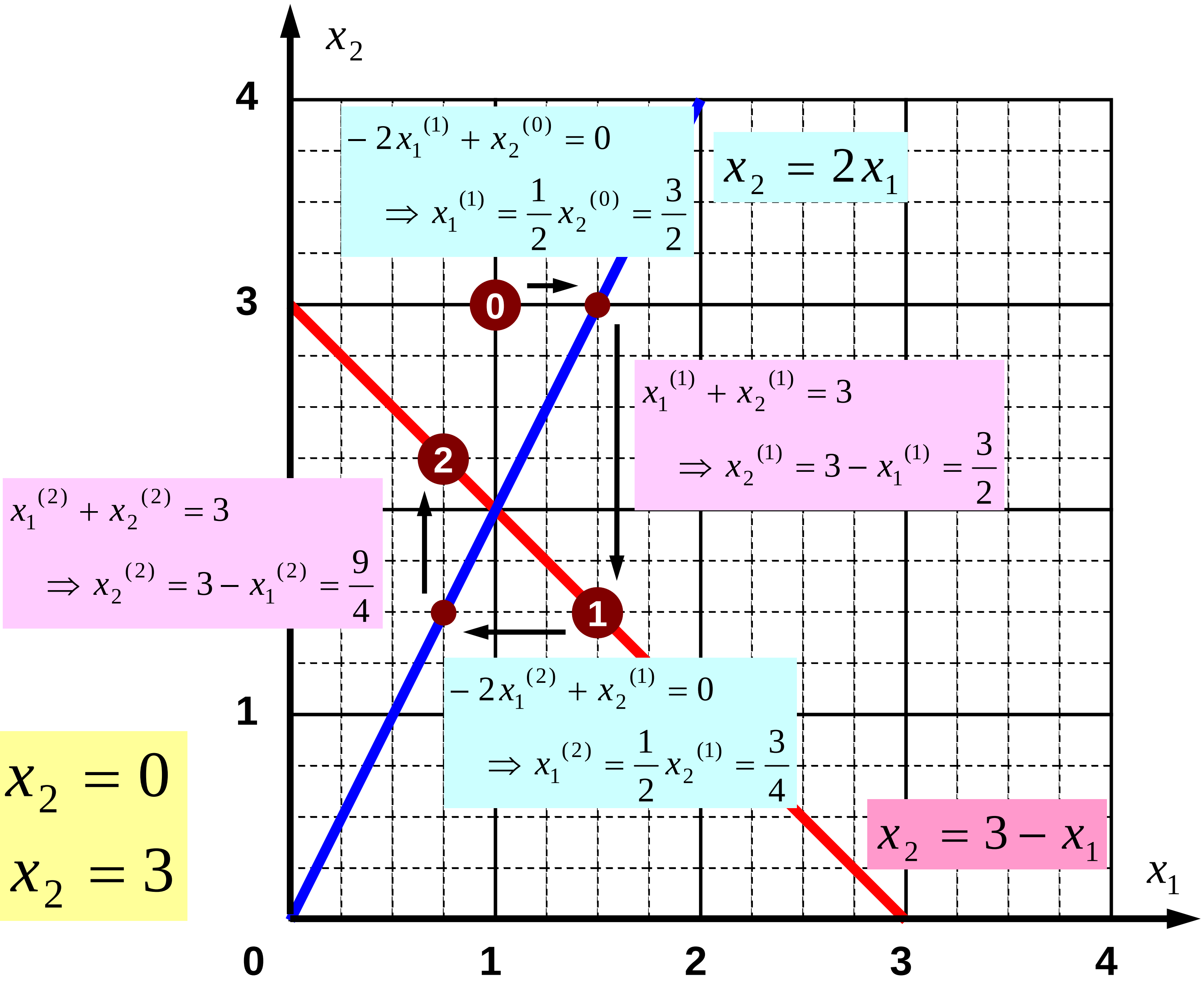
ヤコビ法：2直線の交点が解

$$\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$



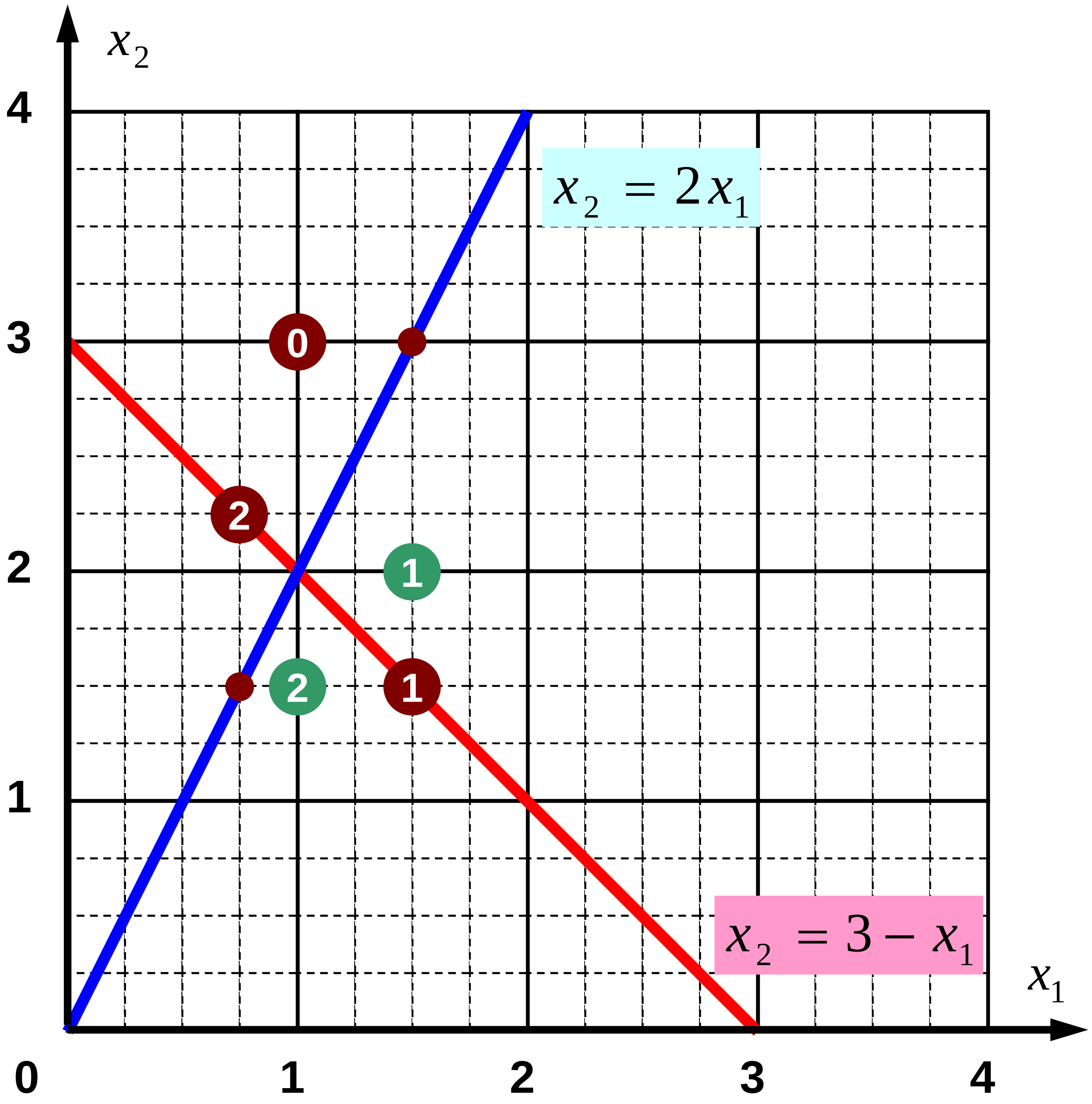
$$\begin{aligned} -2x_1 + x_2 &= 0 \\ x_1 + x_2 &= 3 \end{aligned}$$

Gauss-Seidel法：2直線の交点が解



$$
\begin{aligned}
-2x_1 + x_2 &= 0 \\
x_1 + x_2 &= 3
\end{aligned}
$$

ヤコビ法とガウス・ザイデル法



$$-2x_1 + x_2 = 0$$
$$x_1 + x_2 = 3$$

反復法の収束判定（「解を得られた」という判定）

解の推定値 $\mathbf{x}^{(k)}$

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix}$$

- 適切な条件のもとで，$k \Rightarrow k+1$のプロセスを繰り返すことによって，$\mathbf{x}^{(k)}$は正しい解に収束していく。
- $[\mathbf{A}]\{\mathbf{x}\}=\{\mathbf{b}\}$という方程式を解いているので，$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2 \sim 0$となれば収束したとみなすことができる。
- 通常は$\|\mathrm{b}\|_2$で無次元化した「残差ノルム」が予め設定した値 ε より小さくなった場合に収束したとみなす。**εの値は要求される精度によって異なる。**

残差ノルム

$$\frac{\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{b}\right\|_2} < \varepsilon$$

$$\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right\|_2 = \sqrt{\sum_{i=1}^{n}\left|b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k)}\right|^2}, \quad \left\|\mathbf{b}\right\|_2 = \sqrt{\sum_{i=1}^{n}\left|b_i\right|^2}$$

プログラム例（ヤコビ法）

```
do iter= 1, N*100
  do i= 1, N
    X0(i)= X(i)
  enddo

  do i= 1, N
    RESID= B(i)
    do j= 1, N
      if (j.ne.i) then
        RESID= RESID - A(i,j)*X0(j)
      endif
    enddo
    X(i)= RESID/A(i,i)
  enddo

  （収束判定）

enddo
```

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

A(i,j): **A**のa_{ij}成分
B(i):　**b**の各成分
X(i):　**x**の各成分
X0(i): **x**の各成分（1ステップ前のX(i)）
注：A(i,j)と書くのはメモリアクセス上は効率がとても悪い（Fortranの場合）

プログラム例（ガウス・ザイデル法）

```
do iter= 1, N*100
  do i= 1, N
    RESID= B(i)
    do j= 1, N
      if (j.ne.i) then
        RESID= RESID - A(i,j)*X(j)
      endif
    enddo
    X(i)= RESID/A(i,i)
  enddo

   （収束判定）

enddo
```

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

$$(1 \le i \le n)$$

A(i,j): $\mathbf{A}$のa_{ij}成分
B(i): $\mathbf{b}$の各成分
X(i): $\mathbf{x}$の各成分

プログラムはヤコビ法より実は簡単

プログラム例（収束判定）

```
 BNRM= 0.d0
 do i= 1, N
   BNRM= BNRM + B(i)**2
 enddo
 BNRM= dsqrt(BNRM)

...

do iter= 1, N*100

   (反復計算)

  VAL= 0.d0
  do i= 1, N
    RESID= B(i)
    do j= 1, N
      RESID= RESID - A(i,j)*X(j)
    enddo
    VAL= VAL + RESID**2
  enddo

  VAL= dsqrt(VAL)/BNRM

  if (VAL.lt.EPS) exit
enddo
```

$$\left\|\mathbf{b}\right\|_2 = \sqrt{\sum_{i=1}^{n}\left|b_i\right|^2}$$

$$\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right\|_2 = \sqrt{\sum_{i=1}^{n}\left|b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k)}\right|^2}$$

$$\frac{\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{b}\right\|_2} < \varepsilon$$

反復計算と同じくらいのコスト：結構時間がかかる
（数回に1回の判定でも良い）

プログラム例（収束判定：代替法）

```
 BNRM= 0.d0
 do i= 1, N
   BNRM= BNRM + B(i)**2
 enddo
 BNRM= dsqrt(BNRM)

...

do iter= 1, N*100

   (反復計算)

  VAL= 0.d0
  do i= 1, N
    VAL= VAL + (X(i)-X0(i))**2
  enddo

  VAL= dsqrt(VAL)/BNRM

  if (VAL.lt.EPS) exit
enddo
```

$$\|\mathbf{b}\|_2 = \sqrt{\sum_{i=1}^{n} |b_i|^2}$$

$$\frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2}{\|\mathbf{b}\|_2} < \varepsilon$$

計算量は減少，必ずしも正しい解に収束していない場合がある

数値例（1/2）

例

$$3x_1 + x_2 - x_3 = 0$$

$$x_1 - 4x_2 + 2x_3 = 24$$

$$2x_1 - x_2 + 5x_3 = 14$$

ヤコビ法

$$x_1^{(k+1)} = \frac{1}{3}(-x_2^{(k)} + x_3^{(k)})$$

$$x_2^{(k+1)} = -\frac{1}{4}(24 - x_1^{(k)} - 2x_3^{(k)})$$

$$x_3^{(k+1)} = \frac{1}{5}(14 - 2x_1^{(k)} + x_2^{(k)})$$

ガウス･ザイデル法

$$x_1^{(k+1)} = \frac{1}{3}(-x_2^{(k)} + x_3^{(k)})$$

$$x_2^{(k+1)} = -\frac{1}{4}(24 - x_1^{(k+1)} - 2x_3^{(k)})$$

$$x_3^{(k+1)} = \frac{1}{5}(14 - 2x_1^{(k+1)} + x_2^{(k+1)})$$

数値例（2/2）

$$(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}) = (0,0,0)$$

正解 $(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}) = (2,-5,1)$

k	ヤコビ法		ガウス・ザイデル法	
	$(x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$	$\lVert \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \rVert / \lVert \mathbf{b} \rVert$	$(x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$	$\lVert \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \rVert / \lVert \mathbf{b} \rVert$
1	+0.000000E+00 -6.000000E+00 +2.800000E+00	4.330875E-01	+0.000000E+00 -6.000000E+00 +1.600000E+00	2.967876E-01
2	+2.933333E+00 -4.600000E+00 +1.600000E+00	1.869982E-01	+2.533333E+00 -4.566667E+00 +8.733333E-01	9.369901E-02
3	+2.066667E+00 -4.466667E+00 +7.066667E-01	1.224674E-01	+1.813333E+00 -5.110000E+00 +1.052667E+00	2.903653E-02
4	+1.724444E+00 -5.130000E+00 +1.080000E+00	4.005661E-02	+2.054222E+00 -4.960111E+00 +9.862889E-01	9.133105E-03
5	+2.070000E+00 -5.028889E+00 +1.084222E+00	2.500786E-02	+1.982133E+00 -5.011322E+00 +1.004882E+00	2.846575E-03

ヤコビ法，ガウス・ザイデル法の収束条件と対角優位性

係数行列Aが以下の性質を満たしていれば、どの初期点から始めても真の解に収束する.

$$\left|a_{ii}\right| > \sum_{j=1,\, j\neq i}^{n} \left|a_{ij}\right| \quad i = 1,2,\cdots,n \qquad \text{（（狭義行）対角優位）}$$

対角優位

第 i 行の、対角項以外の成分の絶対値の和よりも対角項の絶対値が大きい場合

証明：定常反復法の構成（1/2）

- 連立一次方程式:　$\mathbf{Ax} = \mathbf{b}$
- 定常反復法:　$\mathbf{x}^{(k+1)} = \mathbf{Mx}^{(k)} + \mathbf{Nb}$（$\mathbf{M}$:反復行列）
- 以下のように考える:$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$

$$
\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad
\mathbf{D} = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}
$$

$$
\mathbf{L} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{pmatrix}, \quad
\mathbf{U} = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}
$$

証明：定常反復法の構成（2/2）

ヤコビ法

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

$$\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}), \quad \mathbf{N} = \mathbf{D}^{-1}$$

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

ガウス・ザイデル法

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

$$\mathbf{M} = -(\mathbf{D}+\mathbf{L})^{-1}\mathrm{U}, \quad \mathbf{N} = (\mathbf{D}+\mathbf{L})^{-1}$$

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

証明：ヤコビ法の収束（1/2）

$$
\begin{aligned}
\mathbf{x} &= \mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{b} \\
\mathbf{x}^{(k)} &= \mathbf{M}\mathbf{x}^{(k-1)} + \mathbf{N}\mathbf{b} \\
\mathbf{x}^{(k-1)} &= \mathbf{M}\mathbf{x}^{(k-2)} + \mathbf{N}\mathbf{b} \\
&\cdots \\
\mathbf{x}^{(1)} &= \mathbf{M}\mathbf{x}^{(0)} + \mathbf{N}\mathbf{b}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x} - \mathbf{x}^{(k)} &= \mathbf{M}\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right) = \cdots \\
&= \mathbf{M}^{k}\left(\mathbf{x} - \mathbf{x}^{(0)}\right)
\end{aligned}
$$

$$
\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \le \left\|\mathbf{M}\right\|^{k} \left\|\mathbf{x} - \mathbf{x}^{(0)}\right\|
$$

- k→∞の時，任意の初期値$\mathbf{x}^{(0)}$に対して，$\mathbf{x}^{(k)}$が解$\mathbf{x}$に収束するためには，$||\mathbf{M}||<1$である必要がある。
- スペクトル半径（絶対値最大固有値）$\rho(\mathbf{M}) \leqq ||\mathbf{M}||$より，$\rho(\mathbf{M})<1$のとき，$\mathbf{x}^{(k)}$は解$\mathbf{x}$に収束する

証明：ヤコビ法の収束（2/2）

- 対角優位な行列のスペクトル半径は1より小さい：

$$\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$$

$$\|\mathbf{A}\|_1 = \max_j\left(\sum_{i=1}^{n}|a_{ij}|\right)$$ 列方向の絶対値の和の最大値

修正 p.16

$$\|\mathbf{A}\|_\infty = \max_i\left(\sum_{j=1}^{n}|a_{ij}|\right)$$ 行方向の絶対値の和の最大値

$$\therefore \left\|\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\right\|_\infty = \max_i\left(\sum_{j=1}^{n}\left|\frac{a_{ij}}{a_{ii}}\right|\right) < 1$$

$$\rho\left(\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\right) \le \left\|\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\right\|_\infty = \max_i\left(\sum_{j=1}^{n}\left|\frac{a_{ij}}{a_{ii}}\right|\right) < 1$$

収束条件と対角優位性(1/2)

$$\left|a_{ii}\right| > \sum_{j=1,\, j\neq i}^{n} \left|a_{ij}\right| \quad i = 1,2,\cdots,n$$

解くべき連立一次方程式

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

真の解

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

初期解

$$\begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

ガウス・ザイデル法

$$\begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} = \begin{pmatrix} 5/2 \\ 9/4 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix} = \begin{pmatrix} 11/8 \\ 45/16 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^{(3)} \\ x_2^{(3)} \end{pmatrix} = \begin{pmatrix} 35/32 \\ 189/64 \end{pmatrix}$$

収束条件と対角優位性(2/2)

$$\left|a_{ii}\right| > \sum_{j=1,\, j\neq i}^{n} \left|a_{ij}\right| \quad i = 1,2,\cdots,n$$

解くべき連立一次方程式

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

真の解

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

初期解

$$\begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

ガウス・ザイデル法

$$\begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} = \begin{pmatrix} 5 \\ -3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix} = \begin{pmatrix} 11 \\ -15 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^{(3)} \\ x_2^{(3)} \end{pmatrix} = \begin{pmatrix} 35 \\ -63 \end{pmatrix}$$

収束条件と対角優位性(3/3)

$$|a_{ii}| > \sum_{j=1,\, j\neq i}^{n} |a_{ij}| \quad i = 1,2,\cdots,n$$

解くべき連立一次方程式

$$\begin{bmatrix} \boxed{3} & 1 & -2 \\ 1 & -4 & 2 \\ 2 & -1 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 24 \\ 14 \end{pmatrix}$$

a_{11}を変えて計算してみよう(自習課題)
$\{x_1,x_2,x_3\}=\{2,-5,1\}$となるように右辺調節

真の解

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -5 \\ 1 \end{pmatrix}$$

初期解

$$\begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$\{x_1,x_2,x_3\}=\{2,-5,1\}$となるよう右辺を調節

$$\begin{bmatrix} a_{11} & 1 & -2 \\ 1 & -4 & 2 \\ 2 & -1 & 5 \end{bmatrix} \begin{pmatrix} 2 \\ -5 \\ 1 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\begin{bmatrix} a_{11} & 1 & -2 \\ 1 & -4 & 2 \\ 2 & -1 & 5 \end{bmatrix} \begin{pmatrix} 2 \\ -5 \\ 1 \end{pmatrix} = \begin{pmatrix} 2a_{11}-7 \\ 24 \\ 14 \end{pmatrix}$$

- 定常反復法(1)
  - ヤコビ法, ガウス･ザイデル法
- **非定常反復法**
  - **共役勾配法**
- 定常反復法(2)
  - SOR法

非定常反復法：クリロフ部分空間法（1/2）
Krylov Subspace Method

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{x} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}$$

以下の反復式を導入し$\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$を求める：

$$\begin{aligned}\mathbf{x}_k &= \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}_{k-1} \\ &= (\mathbf{b} - \mathbf{Ax}_{k-1}) + \mathbf{x}_{k-1} \\ &= \mathbf{r}_{k-1} + \mathbf{x}_{k-1}\end{aligned}$$

$where\ \mathbf{r}_k = \mathbf{b} - \mathbf{Ax}_k$ ：残差ベクトル（residual）

$$\mathbf{x}_k = \mathbf{x}_0 + \sum_{i=0}^{k-1} \mathbf{r}_i$$

$$\begin{aligned}\mathbf{r}_k &= \mathbf{b} - \mathbf{Ax}_k = \mathbf{b} - \mathbf{A}(\mathbf{r}_{k-1} + \mathbf{x}_{k-1}) \\ &= (\mathbf{b} - \mathbf{Ax}_{k-1}) - \mathbf{Ar}_{k-1} = \mathbf{r}_{k-1} - \mathbf{Ax}_{k-1}\mathbf{r}_{k-1} = (\mathbf{I} - \mathbf{A})\mathbf{r}_{k-1}\end{aligned}$$

非定常反復法：クリロフ部分空間法（2/2）
Krylov Subspace Method

$$\mathbf{x}_k = \mathbf{x}_0 + \sum_{i=0}^{k-1} \mathbf{r}_i = \mathbf{x}_0 + \mathbf{r}_0 + \sum_{i=0}^{k-2} (\mathbf{I} - \mathbf{A})\mathbf{r}_i = \mathbf{x}_0 + \mathbf{r}_0 + \sum_{i=1}^{k-1} (\mathbf{I} - \mathbf{A})^i \mathbf{r}_0$$

$$\mathbf{z}_k = \mathbf{r}_0 + \sum_{i=1}^{k-1} (\mathbf{I} - \mathbf{A})^i \mathbf{r}_0 = \left[\mathbf{I} + \sum_{i=1}^{k-1} (\mathbf{I} - \mathbf{A})^i \right] \mathbf{r}_0$$

$\mathbf{z_k}$はk次のクリロフ部分空間（Krylov Subspace）に属するベクトル，問題はクリロフ部分空間からどのようにして解の近似ベクトル$\mathbf{x_k}$を求めるかにある：

$$\left[\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0 \right]$$

代表的な非定常反復法：共役勾配法

- Conjugate Gradient法，略して「CG」法
  - 最も代表的な「非定常」反復法
- 対称正定値行列（Symmetric Positive Definite：SPD）
  - 任意のベクトル$\{x\}$に対して$\{x\}^T[A]\{x\}>0$
  - 全対角成分>0，全固有値>0，全部分行列式（主小行列式・首座行列式）>0と同値

$$
\det\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2n} \\
a_{31} & a_{32} & a_{33} & a_{34} & \cdots & a_{3n} \\
a_{41} & a_{42} & a_{43} & a_{44} & \cdots & a_{4n} \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
a_{n1} & a_{n2} & a_{n3} & a_{n4} & \cdots & a_{nn}
\end{bmatrix}
$$

- アルゴリズム
  - 最急降下法（Steepest Descent Method）の変種
  - $\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_i \mathbf{p}^{(i)}$
    - $x^{(i)}$：反復解，$p^{(i)}$：探索方向，α_i：定数）
  - 厳密解をyとするとき $\{x-y\}^T[A]\{x-y\}$を最小とするような$\{x\}$を求める。
  - 詳細は参考文献参照
    - 例えば：森正武「数値解析（第2版）」（共立出版）

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減（DAXPY）

$x^{(i)}$: Vector

α_i : Scalar

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減（DAXPY）

$x^{(i)}$: Vector

α_i : Scalar

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減（DAXPY）

$x^{(i)}$: Vector
α_i : Scalar

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減（DAXPY）
  - Double
  - {y}= a{x} + {y}

$x^{(i)}$: Vector

α_i : Scalar

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

$x^{(i)}$: Vector

α_i : Scalar

CG法アルゴリズムの導出(1/5)

yを厳密解($Ay=b$)とするとき，下式を最小にするxを求める:

$$(x-y)^T[A](x-y)$$

$$(x-y)^T[A](x-y)=(x,Ax)-(y,Ax)-(x,Ay)+(y,Ay)$$
$$=(x,Ax)-2(x,Ay)+(y,Ay)=(x,Ax)-2(x,b)+\underline{(y,b)}$$

定数

従って，下記 $f(x)$ を最小にするxを求めればよい:

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

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

任意のベクトル h

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

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

• 任意のベクトルh

$$f(x+h) = \frac{1}{2}(x+h, A(x+h)) - (x+h, b)$$

$$= \frac{1}{2}(x+h, Ax) + \frac{1}{2}(x+h, Ah) - (x, b) - (h, b)$$

$$= \frac{1}{2}(x, Ax) + \underline{\frac{1}{2}(h, Ax)} + \underline{\frac{1}{2}(x, Ah)} + \frac{1}{2}(h, Ah) - (x, b) - (h, b)$$

$$= \frac{1}{2}(x, Ax) - (x, b) + (h, Ax) - (h, b) + \frac{1}{2}(h, Ah)$$

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

CG法アルゴリズムの導出(2/5)

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。
今，k番目の近似値 $x^{(k)}$と探索方向$p^{(k)}$が決まったとすると:

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$$

$f(x^{(k+1)})$ を最小にするためには:

$$f\left(x^{(k)} + \alpha_k p^{(k)}\right) = \frac{1}{2}\alpha_k{}^2\left(p^{(k)}, Ap^{(k)}\right) - \alpha_k\left(p^{(k)}, b - Ax^{(k)}\right) + f\left(x^{(k)}\right)$$

$$\frac{\partial f\left(x^{(k)} + \alpha_k p^{(k)}\right)}{\partial \alpha_k} = 0 \Rightarrow \alpha_k = \frac{\left(p^{(k)}, b - Ax^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} = \frac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} \qquad \textbf{(1)}$$

$r^{(k)} = b - Ax^{(k)}$ は第k近似に対する残差

CG法アルゴリズムの導出(3/5)

残差 $r^{(k)}$も以下の式によって計算できる:

$$r^{(k+1)} = b - Ax^{(k+1)},\ r^{(k)} = b - Ax^{(k)}$$

$$r^{(k+1)} - r^{(k)} = Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ap^{(k)}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)} \qquad \textbf{(2)}$$

探索方向を以下の漸化式によって求める:

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)},\ r^{(0)} = p^{(0)} \qquad \textbf{(3)}$$

本当のところは下記のように(k+1)回目に厳密解 y が求まれば良いのであるが，解がわかっていない場合は困難・・・

$$y = x^{(k+1)} + \alpha_{k+1} p^{(k+1)}$$

CG法アルゴリズムの導出(4/5)

ところで，下式のような都合の良い直交関係がある：

$$\left(Ap^{(k)}, y - x^{(k+1)}\right) = 0$$

$$\left(Ap^{(k)}, y - x^{(k+1)}\right) = \left(p^{(k)}, Ay - Ax^{(k+1)}\right) = \left(p^{(k)}, b - Ax^{(k+1)}\right)$$
$$= \left(p^{(k)}, b - A\left[x^{(k)} + \alpha_k p^{(k)}\right]\right) = \left(p^{(k)}, b - Ax^{(k)} - \alpha_k Ap^{(k)}\right)$$
$$= \left(p^{(k)}, r^{(k)} - \alpha_k Ap^{(k)}\right) = \left(p^{(k)}, r^{(k)}\right) - \alpha_k \left(p^{(k)}, Ap^{(k)}\right) = 0$$

$$\because \alpha_k = \frac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$$

従って以下が成立する：

$$\left(Ap^{(k)}, y - x^{(k+1)}\right) = \left(Ap^{(k)}, \alpha_{k+1} p^{(k+1)}\right) = 0 \Rightarrow \left(p^{(k+1)}, Ap^{(k)}\right) = 0$$

CG法アルゴリズムの導出(5/5)

$$\left(p^{(k+1)}, Ap^{(k)}\right) = \left(r^{(k+1)} + \beta_k p^{(k)}, Ap^{(k)}\right) = \left(r^{(k+1)}, Ap^{(k)}\right) + \beta_k \left(p^{(k)}, Ap^{(k)}\right) = 0$$

$$\Rightarrow \beta_k = \frac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} \qquad \textbf{(4)}$$

$\left(p^{(k+1)}, Ap^{(k)}\right) = 0$ $p^{(k)}$ と $p^{(k+1)}$ が行列Aに関して共役（conjugate）

```
Compute p(0)=r(0)= b-[A]x(0)
for i= 1, 2, …
    calc. αi-1
    x(i)= x(i-1) + αi-1p(i-1)
    r(i)= r(i-1) – αi-1[A]q(i-1)

    check convergence |r|
    (if not converged)
    calc. βi-1
    p(i)= r(i) + βi-1 p(i-1)
end
```

$$\alpha_{i-1} = \frac{\left(p^{(i-1)}, r^{(i-1)}\right)}{\left(p^{(i-1)}, Ap^{(i-1)}\right)}$$

$$\beta_{i-1} = \frac{-\left(r^{(i)}, Ap^{(i-1)}\right)}{\left(p^{(i-1)}, Ap^{(i-1)}\right)}$$

CG法アルゴリズム

任意の(i,j)に対して以下の共役関係が得られる:

$$\left(p^{(i)}, Ap^{(j)}\right) = 0 \ (i \neq j)$$

探索方向$p^{(k)}$，残差ベクトル$r^{(k)}$についても以下の関係が成立する:

$$\left(r^{(i)}, r^{(j)}\right) = 0 \ (i \neq j), \quad \left(p^{(k)}, r^{(k)}\right) = \left(r^{(k)}, r^{(k)}\right)$$

N次元空間で互いに直交で一次独立な残差ベクトル $r^{(k)}$ はN個しか存在しない，従って共役勾配法は未知数がN個のときにN回以内に収束する
⇒ 実際は丸め誤差の影響がある（条件数が大きい場合）

Top 10 Algorithms in the 20th Century (SIAM)
http://www.siam.org/news/news.php?id=637
モンテカルロ法，シンプレックス法，クリロフ部分空間法，行列分解法，最適化Fortranコンパイラ，QR法，クイックソート，FFT，整数関係アルゴリズム，FMM（高速多重極法）

Proof (1/3)

Mathematical Induction
数学的帰納法

$$\left(r^{(i)}, r^{(j)}\right) = 0 \ (i \neq j)$$ 直交性

$$\left(p^{(i)}, Ap^{(j)}\right) = 0 \ (i \neq j)$$ 共役性

(1) $$\alpha_k = \frac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$$

(2) $$r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$$

(3) $$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}, \ r^{(0)} = p^{(0)}$$

(4) $$\beta_k = \frac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$$

Proof (2/3)

Mathematical Induction
数学的帰納法

$$\left(r^{(i)}, r^{(j)}\right) = 0 \ (i \neq j)$$
$$\left(p^{(i)}, Ap^{(j)}\right) = 0 \ (i \neq j) \qquad (*)$$

(＊) is satisfied for $i \leq k,\ j \leq k$ where $i \neq j$ または $0 \leq i < j \leq k$

if $i < k$

$$\left(r^{(k+1)}, r^{(i)}\right) = \left(r^{(i)}, r^{(k+1)}\right) \overset{(2)}{=} \left(r^{(i)}, r^{(k)} - \alpha_k Ap^{(k)}\right)$$
$$\overset{(*)}{=} -\alpha_k \left(r^{(i)}, Ap^{(k)}\right) \overset{(4)}{=} -\alpha_k \left(p^{(i)} - \beta_{i-1} p^{(i-1)}, Ap^{(k)}\right)$$
$$= -\alpha_k \left(p^{(i)}, Ap^{(k)}\right) + \alpha_k \beta_{i-1} \left(p^{(i-1)}, Ap^{(k)}\right) \overset{(*)}{=} 0$$

if $i = k$

$$\left(r^{(k+1)}, r^{(k)}\right) \overset{(2)}{=} \left(r^{(k)}, r^{(k)}\right) - \left(r^{(k)}, \alpha_k Ap^{(k)}\right)$$
$$\overset{(3)}{=} \left(r^{(k)}, r^{(k)}\right) - \left(p^{(k)} - \beta_{k-1} p^{(k-1)}, \alpha_k Ap^{(k)}\right)$$
$$\overset{(*)}{=} \left(r^{(k)}, r^{(k)}\right) - \alpha_k \left(p^{(k)}, Ap^{(k)}\right) \overset{(1)}{=} \left(r^{(k)}, r^{(k)}\right) - \left(p^{(k)}, r^{(k)}\right)$$
$$\overset{(3)}{=} \left(r^{(k)}, r^{(k)}\right) - \left(\beta_{k-1} p^{(k-1)} + r^{(k)}, r^{(k)}\right)$$
$$= -\beta_{k-1} \left(p^{(k-1)}, r^{(k)}\right) \overset{(2)}{=} -\beta_{k-1} \left(p^{(k-1)}, r^{(k-1)} - \alpha_{k-1} Ap^{(k-1)}\right)$$
$$= -\beta_{k-1} \left\{\left(p^{(k-1)}, r^{(k-1)}\right) - \alpha_{k-1} \left(p^{(k-1)}, Ap^{(k-1)}\right)\right\} \overset{(1)}{=} 0$$

(1) $\alpha_k = \dfrac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

(2) $r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$

(3) $p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$

(4) $\beta_k = \dfrac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

Proof (3/3)

Mathematical Induction
数学的帰納法

$$\left(r^{(i)}, r^{(j)}\right) = 0 \; (i \neq j)$$
$$\left(p^{(i)}, Ap^{(j)}\right) = 0 \; (i \neq j) \qquad (*)$$

(*) is satisfied for $i \leq k,\ j \leq k$ where $i \neq j$ または $0 \leq i < j \leq k$

if $i < k$

$$\left(p^{(k+1)}, Ap^{(i)}\right) \overset{(3)}{=} \left(r^{(k+1)} + \beta_k p^{(k)}, Ap^{(i)}\right) \overset{(*)}{=} \left(r^{(k+1)}, Ap^{(i)}\right) \overset{(2)}{=} \frac{1}{\alpha_k}\left(r^{(k+1)}, r^{(i)} - r^{(i-1)}\right) = 0$$

if $i = k$

$$\left(p^{(k+1)}, Ap^{(k)}\right) \overset{(3)}{=} \left(r^{(k+1)}, Ap^{(k)}\right) + \beta_k \left(p^{(k)}, Ap^{(k)}\right) \overset{(4)}{=} 0$$

(1) $\alpha_k = \dfrac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

(2) $r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$

(3) $p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$

(4) $\beta_k = \dfrac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

$$\left(p^{(k)}, r^{(k)}\right) \overset{(3)}{=} \left(r^{(k)} + \beta_{k-1} p^{(k-1)}, r^{(k)}\right)$$
$$= \left(\beta_{k-1} p^{(k-1)}, r^{(k)}\right) + \left(r^{(k)}, r^{(k)}\right) = \left(r^{(k)}, r^{(k)}\right)$$

(1) $\alpha_k = \dfrac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

(2) $r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$

(3) $p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$

(4) $\beta_k = \dfrac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$

$$\because \left(Ap^{(k)}, y - x^{(k+1)}\right) = \left(p^{(k)}, Ay - Ax^{(k+1)}\right)$$
$$= \left(p^{(k)}, b - Ax^{(k+1)}\right) = \left(p^{(k)}, r^{(k+1)}\right) = 0$$

$$\left(p^{(i)}, r^{(k+1)}\right) = 0, \quad i = 0,1,\ldots,k$$

$$x^{(k+1)} = x^{(i+1)} + \sum_{j=i+1}^{k} \alpha_j p^{(j)}$$

$$r^{(k+1)} = b - Ax^{(k+1)} = b - A\left[x^{(i+1)} + \sum_{j=i+1}^{k} \alpha_j p^{(j)}\right]$$

$$= \left[b - Ax^{(i+1)}\right] - \sum_{j=i+1}^{k} \alpha_j Ap^{(j)} = r^{(i+1)} - \sum_{j=i+1}^{k} \alpha_j Ap^{(j)}$$

$$\left(p^{(i)}, r^{(k+1)}\right) = \left(p^{(i)}, r^{(i+1)} - \sum_{j=i+1}^{k} \alpha_j Ap^{(j)}\right)$$

$$= \underbrace{\left(p^{(i)}, r^{(i+1)}\right)}_{=0} - \underbrace{\left(p^{(i)}, \sum_{j=i+1}^{k} \alpha_j Ap^{(j)}\right)}_{=0} = 0$$

α_k, β_k

実際はα_k, β_kはもうちょっと簡単な形に変形できる:

$$\alpha_k = \frac{\left(p^{(k)}, b - Ax^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} = \frac{\left(p^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} = \frac{\left(r^{(k)}, r^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)}$$

$$\because \left(p^{(k)}, r^{(k)}\right) = \left(r^{(k)}, r^{(k)}\right)$$

$$\beta_k = \frac{-\left(r^{(k+1)}, Ap^{(k)}\right)}{\left(p^{(k)}, Ap^{(k)}\right)} = \frac{\left(r^{(k+1)}, r^{(k+1)}\right)}{\left(r^{(k)}, r^{(k)}\right)}$$

$$\because \left(r^{(k+1)}, Ap^{(k)}\right) = \frac{\left(r^{(k+1)}, r^{(k)} - r^{(k+1)}\right)}{\alpha_k} = -\frac{\left(r^{(k+1)}, r^{(k+1)}\right)}{\alpha_k}$$

共役勾配法（CG法）のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    ρi-1= r(i-1) r(i-1)
    if i=1
      p(1)= r(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= r(i-1) + βi-1 p(i-1)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

$x^{(i)}$: Vector

α_i : Scalar

$$\beta_{i-1} = \frac{\left(r^{(i-1)}, r^{(i-1)}\right)}{\left(r^{(i-2)}, r^{(i-2)}\right)} \quad \begin{array}{l} \left(= \rho_{i-1}\right) \\ \left(= \rho_{i-2}\right) \end{array}$$

$$\alpha_i = \frac{\left(r^{(i-1)}, r^{(i-1)}\right)}{\left(p^{(i)}, Ap^{(i)}\right)} \quad \left(= \rho_{i-1}\right)$$

プログラム例（CG法）（1/3）

```
do i= 1, N
  R(i)= B(i)
  do j= 1, N
    R(i)= R(i) - AMAT(i,j)*X(j)
  enddo
enddo

BNRM2= 0.0D0
do i= 1, N
  BNRM2= BNRM2 + B(i) **2
enddo
```

AMAT(i,j): **A**のa_{ij}成分
B(i): **b**の各成分
X(i): **x**の各成分

P(i): **p**の各成分
Q(i): **q**の各成分
R(i): **r**の各成分

Compute $r^{(0)}= b-[A]x^{(0)}$

<u>for</u> i= 1, 2, …
  $\rho_{i-1}= r^{(i-1)}\, r^{(i-1)}$
  <u>if</u> i=1
    $p^{(1)}= r^{(0)}$
  <u>else</u>
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= r^{(i-1)} + \beta_{i-1}\, p^{(i-1)}$
  <u>endif</u>
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
<u>end</u>

プログラム例（CG法）（2/3）

```
   do iter= 1, ITERmax
     RHO= 0.d0
     do i= 1, N
       RHO= RHO + R(i)*R(i)
     enddo

     if ( iter.eq.1 ) then
       do i= 1, N
         P(i)= R(i)
       enddo
      else
       BETA= RHO / RHO1
       do i= 1, N
         P(i)= R(i) + BETA*P(i)
       enddo
     endif

     do i= 1, N
       Q(i)= 0.d0
       do j= 1, N
         Q(i)= Q(i) + AMAT(i,j)*P(j)
       enddo
     enddo

...
   enddo
```

Compute $r^{(0)}= b-[A]x^{(0)}$
<u>for</u> i= 1, 2, …
$\rho_{i-1}= r^{(i-1)} r^{(i-1)}$
<u>**if**</u> **i=1**
$p^{(1)}= r^{(0)}$
<u>**else**</u>
$\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
$p^{(i)}= r^{(i-1)} + \beta_{i-1} p^{(i-1)}$
<u>**endif**</u>
$q^{(i)}= [A]p^{(i)}$
$\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
$x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
$r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
check convergence |r|
<u>end</u>

プログラム例（CG法）（3/3）

```
  do iter= 1, ITERmax
...

    C1= 0.d0
    do i= 1, N
      C1= C1 + P(i)*Q(i)
    enddo
    ALPHA= RHO / C1

    do i= 1, N
      X(i)= X(i) + ALPHA * P(i)
      R(i)= R(i) - ALPHA * Q(i)
    enddo

    DNRM2 = 0.0
    do i= 1, N
      DNRM2= DNRM2 + R(i)**2
    enddo

    RESID= dsqrt(DNRM2/BNRM2)

    if ( RESID.le.EPS) exit

    RHO1 = RHO

  enddo
```

$$\frac{\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{b}\right\|_2} < \varepsilon$$

$\rho_{i-1} = \rho_{i-2}$

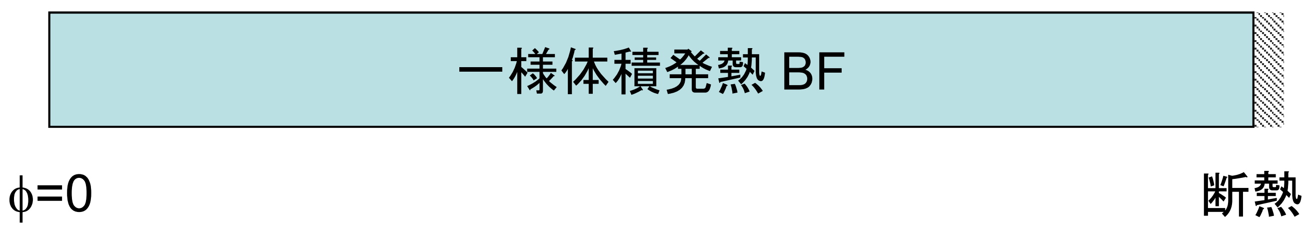
Compute $r^{(0)} = b - [A]x^{(0)}$
for $i = 1, 2, \ldots$
  $\rho_{i-1} = r^{(i-1)} r^{(i-1)}$
  if $i=1$
    $p^{(1)} = r^{(0)}$
  else
    $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
    $p^{(i)} = r^{(i-1)} + \beta_{i-1}\, p^{(i-1)}$
  endif
  $q^{(i)} = [A]p^{(i)}$
  $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence $|r|$
end

一次元熱伝導方程式

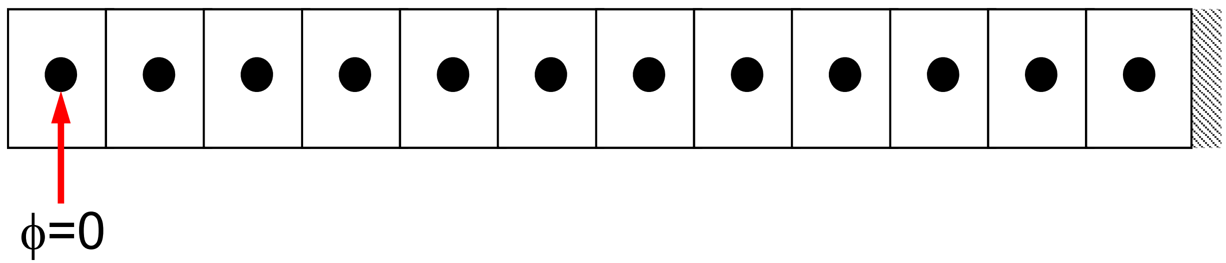
支配方程式:熱伝導率=1(一様)

$$\frac{d^2\phi}{dx^2}+BF=0,\quad \phi=0@x=0,\frac{d\phi}{dx}=0@x=x_{\max}$$

$$\phi=-\frac{1}{2}BF\,x^2+BF\,x_{\max}\,x$$



以下のような離散化(要素中心で従属変数を定義)をしている

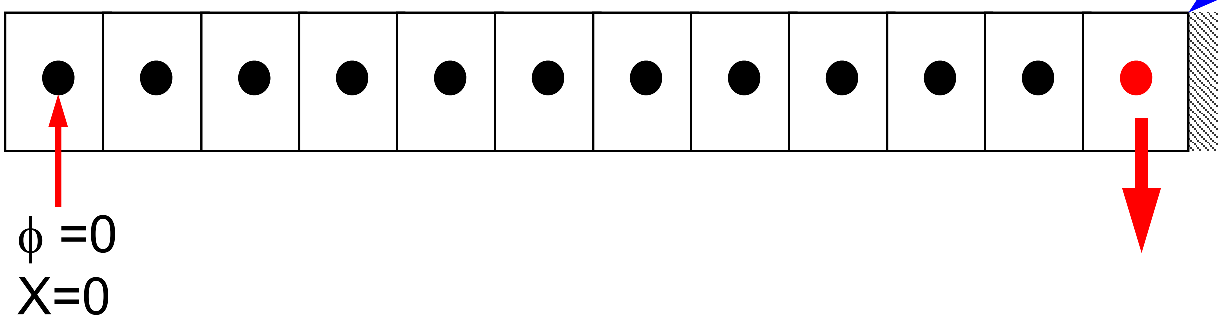


一次元熱伝導方程式

解析解

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{max}\, x$$

断熱となって
いるのはこの面，
しかし温度は計算
されない（$X=X_{max}$）。



Δx=1.d0，メッシュ数=50，とすると，X_{max}=49.5，
●の点のX座標は49.0となる。BF=1.0d0とすると●での温度は：

$$\phi = -\frac{1}{2} 49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

計算例（N=50）：Jacobi法

```
 1000 iters, RESID=    5.443248E-01 PHI(N)=     4.724513E+02
 2000 iters, RESID=    3.255667E-01 PHI(N)=     7.746137E+02
 3000 iters, RESID=    1.947372E-01 PHI(N)=     9.555996E+02
...
34000 iters, RESID=    2.347113E-08 PHI(N)=     1.225000E+03
35000 iters, RESID=    1.403923E-08 PHI(N)=     1.225000E+03
35661 iters, RESID=    9.999053E-09 PHI(N)=     1.225000E+03

1        0.000000E+00     0.000000E+00
2        4.899999E+01     4.900000E+01
3        9.699999E+01     9.700000E+01
4        1.440000E+02     1.440000E+02
5        1.900000E+02     1.900000E+02
...

41       1.180000E+03     1.180000E+03
42       1.189000E+03     1.189000E+03
43       1.197000E+03     1.197000E+03
44       1.204000E+03     1.204000E+03
45       1.210000E+03     1.210000E+03
46       1.215000E+03     1.215000E+03
47       1.219000E+03     1.219000E+03
48       1.222000E+03     1.222000E+03
49       1.224000E+03     1.224000E+03
50       1.225000E+03     1.225000E+03
```

反復回数
最大残差
$\phi(50)$

数値解，解析解

$$\phi = -\frac{1}{2}49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

計算例（N=50）：Gauss-Seidel 法

```
 1000 iters, RESID=    3.303725E-01 PHI(N)=     7.785284E+02
 2000 iters, RESID=    1.182010E-01 PHI(N)=     1.065259E+03
 3000 iters, RESID=    4.229019E-02 PHI(N)=     1.167848E+03
...
16000 iters, RESID=    6.657001E-08 PHI(N)=     1.225000E+03
17000 iters, RESID=    2.381754E-08 PHI(N)=     1.225000E+03
17845 iters, RESID=    9.993196E-09 PHI(N)=     1.225000E+03

1        0.000000E+00     0.000000E+00
2        4.899999E+01     4.900000E+01
3        9.699999E+01     9.700000E+01
4        1.440000E+02     1.440000E+02
5        1.900000E+02     1.900000E+02
...

41       1.180000E+03     1.180000E+03
42       1.189000E+03     1.189000E+03
43       1.197000E+03     1.197000E+03
44       1.204000E+03     1.204000E+03
45       1.210000E+03     1.210000E+03
46       1.215000E+03     1.215000E+03
47       1.219000E+03     1.219000E+03
48       1.222000E+03     1.222000E+03
49       1.224000E+03     1.224000E+03
50       1.225000E+03     1.225000E+03
```

反復回数
最大残差
$\phi(50)$

数値解，解析解

計算例（N=50）：CG法

```
    49 iters, RESID=    0.000000E-00 PHI(N)=     1.225000E+03

1        0.000000E+00     0.000000E+00
2        4.899999E+01     4.900000E+01
3        9.699999E+01     9.700000E+01
4        1.440000E+02     1.440000E+02
5        1.900000E+02     1.900000E+02
...

41       1.180000E+03     1.180000E+03
42       1.189000E+03     1.189000E+03
43       1.197000E+03     1.197000E+03
44       1.204000E+03     1.204000E+03
45       1.210000E+03     1.210000E+03
46       1.215000E+03     1.215000E+03
47       1.219000E+03     1.219000E+03
48       1.222000E+03     1.222000E+03
49       1.224000E+03     1.224000E+03
50       1.225000E+03     1.225000E+03
```

数値解，解析解

反復回数
最大残差
$\phi(50)$

49回目に収束していることに注意（未知数は49個）

$$\phi = -\frac{1}{2}49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

- 定常反復法(1)
  - ヤコビ法, ガウス・ザイデル法
- 非定常反復法
  - 共役勾配法
- 定常反復法(2)
  - SOR法

SOR法（Successive Over-Relaxation）

逐次加速緩和法

ヤコビ法

$$\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}), \quad \mathbf{N} = \mathbf{D}^{-1}$$

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

ガウス・ザイデル法

$$\mathbf{M} = -(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}, \quad \mathbf{N} = (\mathbf{D}+\mathbf{L})^{-1}$$

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

SOR法

$$\mathbf{M} = (\mathbf{D}+\omega\mathbf{L})^{-1}\left\{(1-\omega)\mathbf{D} - \omega\mathbf{U}\right\}, \quad \mathbf{N} = \omega(\mathbf{D}+\omega\mathbf{L})^{-1}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega\left[\xi^{(k+1)} - \mathbf{x}^{(k)}\right] \quad (0 < \omega < 2)$$

$$\xi^{(k+1)} = \mathbf{D}^{-1}\left[\mathbf{b} - \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)}\right]$$ ガウス・ザイデル法の解

ω=1の場合，ガウス・ザイデル法と一致

プログラム例（ガウス・ザイデル法）

```
do iter= 1, N*100
  do i= 1, N
    RESID= B(i)
    do j= 1, N
      if (j.ne.i) then
        RESID= RESID - A(i,j)*X(j)
      endif
    enddo
    X(i)= RESID/A(i,i)
  enddo
```

$$\mathbf{M} = (\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}, \quad \mathbf{N} = (\mathbf{D}+\mathbf{L})^{-1}$$

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

A(i,j): $\mathbf{A}$のa_{ij}成分
B(i): $\mathbf{b}$の各成分
X(i): $\mathbf{x}$の各成分

プログラム例（SOR法）

```
do iter= 1, N*100
  do i= 1, N
    RESID= B(i)
    do j= 1, N
      if (j.ne.i) then
        RESID= RESID - A(i,j)*X(j)
      endif
    enddo
    X(i)= OMEGA*(RESID/A(i,i)-X(i)) + X(i)
  enddo
```

$$\mathbf{M} = (\mathbf{D} + \omega\mathbf{L})^{-1}\{(1-\omega)\mathbf{D} - \omega\mathbf{U}\}$$

$$\mathbf{N} = \omega(\mathbf{D} + \omega\mathbf{L})^{-1}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega\left[\boldsymbol{\xi}^{(k+1)} - \mathbf{x}^{(k)}\right]$$

$$\boldsymbol{\xi}^{(k+1)} = \mathbf{D}^{-1}\left[\mathbf{b} - \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)}\right]$$

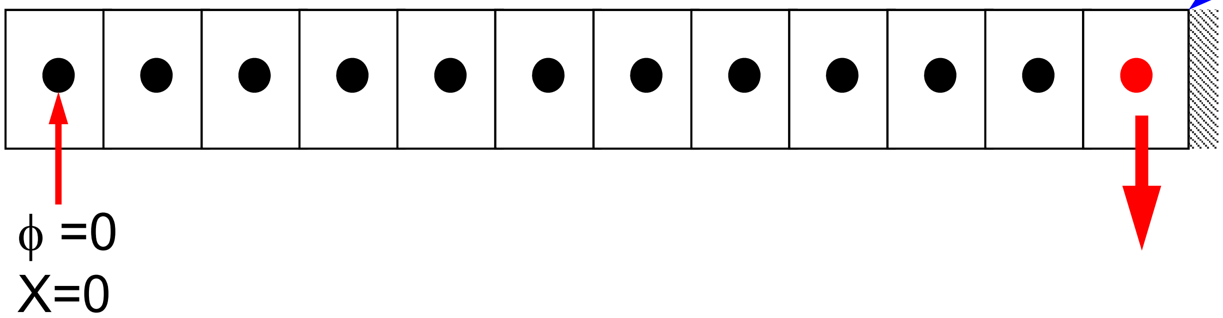
A(i,j): $\mathbf{A}$のa_{ij}成分
B(i): $\mathbf{b}$の各成分
X(i): $\mathbf{x}$の各成分

一次元熱伝導方程式

解析解

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{max}\, x$$

断熱となっているのはこの面，しかし温度は計算されない（$X=X_{max}$）。



Δx=1.d0，メッシュ数=50，とすると，X_{max}=49.5，

●の点のX座標は49.0となる。BF=1.0d0とすると●での温度は：

$$\phi = -\frac{1}{2} 49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

	反復回数
Jacobi	35,561
Gauss-Seidel	17,845
CG	49
SOR（ω=0.70）	33,131
SOR（ω=0.80）	26,762
SOR（ω=0.90）	21,808
SOR（ω=1.00）	17,845
SOR（ω=1.30）	9,614
SOR（ω=1.50）	5,955
SOR（ω=1.60）	4,469
SOR（ω=1.70）	3,155
SOR（ω=1.80）	1,980
SOR（ω=1.90）	886

	反復回数
SOR（ω=1.91）	773
SOR（ω=1.92）	653
SOR（ω=1.93）	520
SOR（ω=1.94）	342
SOR（ω=1.95）	392
SOR（ω=1.96）	497
SOR（ω=1.97）	682
SOR（ω=1.98）	1,020
SOR（ω=1.99）	2,028
SOR（ω=2.00）	NA

ωの範囲

$$\mathbf{M} = -(\mathbf{D}+\omega\mathbf{L})^{-1}\{(1-\omega)\mathbf{D}-\omega\mathbf{U}\},\quad \mathbf{N} = \omega(\mathbf{D}+\omega\mathbf{L})^{-1}$$

$$\det(\mathbf{M}) = \prod_{i=1}^{N}\lambda_i \qquad \lambda_i:\mathbf{M}\text{の固有値}(i=1\sim N)$$

$$\begin{aligned}\det(\mathbf{M}) &= \det\left((\mathbf{D}+\omega\mathbf{L})^{-1}\{(1-\omega)\mathbf{D}-\omega\mathbf{U}\}\right)\\ &= \det\left((\mathbf{D}+\omega\mathbf{L})^{-1}\right)\cdot\det\left((1-\omega)\mathbf{D}-\omega\mathbf{U}\right)\\ &= \det\left(\mathbf{D}^{-1}\right)\cdot\det\left((1-\omega)\mathbf{D}-\omega\mathbf{U}\right)\\ &= \det\left((1-\omega)\mathbf{I}-\omega\mathbf{D}^{-1}\mathbf{U}\right) = \det\left((1-\omega)\mathbf{I}\right) = (1-\omega)^N\end{aligned}$$

$$\rho(\mathbf{M}) = \max_i|\lambda_i| \ge \sqrt[N]{|(1-\omega)|^N} = |1-\omega| \quad \text{再びスペクトル半径}$$

$|1-\omega|<1$となるので，$0<\omega<2$，通常は$1<\omega<2$

反復法（Iterative Method）

- 利点
  - 直接法と比較して，メモリ使用量，計算量が少ない。
  - 並列計算には適している。

- 欠点
  - 収束性が，アプリケーション，境界条件の影響を受けやすい。
    - 収束しない（答えが得られない）可能性がある
  - 前処理（preconditioning）が重要。
    - 条件数（condition number）の大きい問題

- 今日紹介した例は非常に解きやすい問題，難しい問題の対処法については最終回（1月24日）に「さわり」を紹介する。