

連立方程式の数値解法

春日 悠

2024 年 9 月 29 日

目次

1	はじめに	2
2	直接法	2
2.1	連立方程式の解法の基本	2
2.1.1	Gauss の消去法	2
2.1.2	軸選択	3
2.1.3	階数	4
2.1.4	Gauss-Jordan の消去法	6
2.1.5	逆行列	7
2.1.6	スケーリング	8
2.1.7	行列式	8
2.1.8	条件数	8
2.2	分解法	9
2.2.1	LU 分解	9
2.2.2	Cholesky 分解	10
2.2.3	修正 Cholesky 分解	12
2.2.4	修正 Cholesky 分解 2	13
2.2.5	三重対角行列アルゴリズム	14
2.3	直交化	14
2.3.1	Gram-Schmidt の直交化	14
2.3.2	Householder 法	17
2.3.3	Givens 回転	20
2.3.4	Hessenberg 変換	22
3	反復法	24
3.1	反復法の基本	24
3.1.1	Jacobi 法	24
3.1.2	Gauss-Seidel 法	26

3.1.3	SOR 法	27
3.1.4	Richardson 反復	28
3.1.5	反復計算の止め方	29
3.2	勾配法	30
3.2.1	最急降下法	30
3.2.2	最小残差反復	31
3.3	共役勾配法	33
3.4	前処理付き共役勾配法	36
3.4.1	スケーリング	37
3.4.2	不完全 Cholesky 分解	37
3.5	共役残差法	39
3.6	Krylov 部分空間法	43
3.6.1	直交化	44
3.6.2	共役勾配法	46
3.6.3	共役残差法	52
3.6.4	一般化最小残差法	54
参考文献		60

1 はじめに

連立方程式の数値解法について整理する．ここでは係数行列として主に正方行列である実行列 (実数の行列) を想定する．

2 直接法

連立方程式を手で解くのと同様の方法で数値的に解く方法を，直接法 (direct method) という．

2.1 連立方程式の解法の基本

2.1.1 Gauss の消去法

以下のような連立方程式を考える．

$$\begin{aligned} 2x_1 + x_2 &= 4 \dots (a) \\ 3x_1 + 2x_2 &= 7 \dots (b) \end{aligned} \tag{1}$$

x_1, x_2 について解く．まず，(b) から x_1 を消すことを考える．(a) を 2 で割って 3 をかけて，(b) から引けばよいので， $(b) - 3 \times (a)/2$ で

$$\begin{aligned} (2 - 3 \times 1/2)x_2 &= (7 - 3 \times 4/2) \\ \rightarrow (1/2)x_2 &= 1 \end{aligned} \tag{2}$$

これから x_2 が計算できる．

$$x_2 = 2 \tag{3}$$

Algorithm 1 前進消去 (Gauss の消去法)

```
for  $k = 1, 2, \dots, n-1$  do
  for  $i = k+1, k+2, \dots, n$  do
     $h = \frac{a_{ik}}{a_{kk}}$ 
    for  $j = k+1, k+2, \dots, n$  do
       $a_{ij} = a_{ij} - a_{kj}h$ 
    end for
     $b_i = b_i - b_k h$ 
     $a_{ik} = 0$ 
  end for
end for
```

(a) に x_2 を代入すれば x_1 が求まる.

$$x_2 = (4 - 2)/2 = 1 \quad (4)$$

前半の変数を消去する過程を前進消去 (forward elimination), 後半の変数に代入していく過程を後退代入 (back substitution) という.

前進消去の過程を行列で表すと次のようになる.

$$\begin{aligned} & \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \end{bmatrix} \\ \longrightarrow & \begin{bmatrix} 2 & 1 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \end{aligned} \quad (5)$$

係数行列が上三角行列に変換されている. 一番下の行から上に向かって処理していけば変数の値が次々と定まる. これが後退代入である.

連立方程式を行列で次式のように表す.

$$A\mathbf{x} = \mathbf{b} \quad (6)$$

A は $n \times n$ の係数行列, $\mathbf{x}$ は解ベクトル, $\mathbf{b}$ は右辺ベクトルである. A の成分を a_{ij} , $\mathbf{x}$ の成分を x_i , $\mathbf{b}$ の成分を b_i とすると, 前進消去および後退代入のの手順はそれぞれアルゴリズム 1, 2 のように表される. これを Gauss の消去法 (Gaussian elimination) という.

ここでは説明のために下三角成分に 0 を入れているが ($a_{ik} = 0$), 計算には使われないので実装上は放っておいても問題ない.

2.1.2 軸選択

Gauss の消去法では, 係数行列の対角成分で割る部分がある. この対角成分を軸 (pivot, ピボット) という. 前進消去の過程でこの軸が 0 になることがあり, 方程式としては解けるはずなのに, 前進消去が進められなくなることがある. これを避けるために軸選択 (pivoting) という方法が使われる.

具体的には, 今, k 行に注目しているとすると, 軸は a_{kk} となるが, これをそのまま使う代わりに, $a_{kk} \cdots a_{nk}$ の中から絶対値が最大のものを探す. これが仮に i 行の成分 a_{ik} だとすると, これを軸にするために k 行と i 行を入れ替える. そして前進消去を行う. これを繰り返す.

Algorithm 2 後退代入 (Gauss の消去法)

```

$$x_n = \frac{b_n}{a_{nn}}$$
for  $i = n - 1, n - 2, \dots, 1$  do  
    
$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$
end for
```

たとえば、次の方程式を考える。

$$\begin{bmatrix} 2 & 4 & 2 \\ 1 & 2 & 3 \\ 4 & 6 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 9 \\ 8 \end{bmatrix} \quad (7)$$

この例の場合、軸選択なしでは最初の前進消去で 2 行目 2 列目の成分が 0 になってしまう。そこで、軸選択を用いる。1 行目に注目すると、1 列目の絶対値の最大値は 3 行目なので、1 行目と 3 行目を入れ替える。

$$\begin{bmatrix} 4 & 6 & 2 \\ 1 & 2 & 3 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \\ 9 \end{bmatrix} \quad (8)$$

前進消去する。

$$\begin{bmatrix} 4 & 6 & 2 \\ 0 & 0.5 & 2.5 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 7 \\ 2 \end{bmatrix} \quad (9)$$

2 行目に注目すると、2 列目の 2 行目以降の絶対値の最大値は 3 行目なので、2 行目と 3 行目を入れ替える。

$$\begin{bmatrix} 4 & 6 & 2 \\ 0 & 1 & 1 \\ 0 & 0.5 & 2.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 7 \end{bmatrix} \quad (10)$$

前進消去する。

$$\begin{bmatrix} 4 & 6 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 6 \end{bmatrix} \quad (11)$$

実装上は、実際に行を入れ替えるのではなく、添字配列を介して行を参照するようにして、添字配列上の添字だけを入れ替える。手順をアルゴリズム 3, 4 に示す。ここで、`index` は添字の順番を管理する添字配列、`argmax( $x$ )` は配列 x の最大値の要素の添字を返す関数である。

ここでは行だけを入れ替えており、部分的軸選択 (partial pivoting) という。行だけでなく列も入れ替える完全軸選択 (complete pivoting) もある。その場合、解ベクトルの成分の順番も変わる。ただ、そこまでやるメリットはあまりない。

2.1.3 階数

方程式 $Ax = b$ が解けるためには、係数行列 A が正則行列 (逆行列が存在する行列) でなければならない。連立方程式として見た場合、未知数が n 個あるなら、独立した方程式が n 個必要である。独立というのは、ある方程式が他の方程式をかけたり足したりして表現できないという意味である。もしそういうものがあると、消去法の途中ですべての要素の値が 0 になる行が出てくる。つまり、軸が 0 になってしまう。

Algorithm 3 前進消去 (軸選択付き)

```
index = {1, ..., n}
for k = 1, 2, ..., n - 1 do
  I = {indexk, indexk+1, ..., indexn}
  imax = argmax(|ai∈I,k|) + k - 1
  indexk, indeximax ← indeximax, indexk
  k' = indexk
  for i = k + 1, k + 2, ..., n do
    i' = indexi
    h =  $\frac{a_{i'k}}{a_{k'k}}$ 
    for j = k + 1, k + 2, ..., n do
      ai'j = ai'j - ak'jh
    end for
    bi' = bi' - bk'h
    ai'k = 0
  end for
end for
```

Algorithm 4 後退代入 (軸選択付き)

```
i' = indexn
xn =  $\frac{b_{i'}}$ 
for i = n - 1, n - 2, ..., 1 do
  i' = indexi
  xi =  $\frac{b_{i'} - \sum_{j=i+1}^n a_{i'j}x_j}{a_{i'i}}$ 
end for
```

連立方程式のそれぞれの方程式が独立であるというのは、係数行列の列ベクトル同士あるいは行ベクトル同士が線形独立であることを意味する。というのも、 $A\mathbf{x} = \mathbf{0}$ の場合、 A が正則なら $\mathbf{x} = \mathbf{0}$ であるが、 $A\mathbf{x}$ は $\mathbf{x}$ の要素 x_i を係数とした A の列ベクトル $\mathbf{a}_i$ の線形結合 $\sum_i x_i \mathbf{a}_i$ と見ることができるので、これは A の列ベクトルが線形独立であることを意味するからである。また、 $A^T \mathbf{y} = \mathbf{0}$ を考えれば、 A が正則なら A^T も正則なので (行列式 $|A| = |A^T| \neq 0$)、同様に A の行ベクトルが線形独立であることを示すことができる。

A の列ベクトルあるいは行ベクトルの線形独立なものの個数の最大値を A の階数 (rank) という。正方行列とは限らない行列 A の階数を $\text{rank}(A)$ と表す。 $n \times n$ の係数行列 A の階数が n であるとき、 A は正則である。

A の階数と A^T の階数は一致する。列ベクトルの作る空間の次元は階数と同じである。 A の列ベクトルを $\mathbf{a}_i$ 、行ベクトルを $\mathbf{a}'_i$ とする。 A^T の階数を $r = \text{rank}(A^T)$ として、 A^T の列ベクトル空間の基底を $\mathbf{v}_1, \mathbf{v}_2,$

$\dots, \mathbf{v}_r$ とすると, A^T の列ベクトルは次のように表される.

$$\mathbf{a}'_i = \sum_{k=1}^r u_{ik} \mathbf{v}_k \quad (12)$$

成分で表すと

$$a_{ij} = \sum_{k=1}^r u_{ik} v_{kj} \quad (13)$$

これは, 次のように表すことができる.

$$a_{ij} = \sum_{k=1}^r v_{kj} u_{ik} \quad (14)$$

$$\mathbf{a}_j = \sum_{k=1}^r v_{kj} \mathbf{u}_k \quad (15)$$

A の列ベクトル空間は r 個のベクトル $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r$ で表されるので, 少なくとも $\text{rank}(A) \leq \text{rank}(A^T)$ ということになる. 上の A^T 側からの議論を A 側からも展開でき, その場合は $\text{rank}(A^T) \leq \text{rank}(A)$ ということになる. ということは, $\text{rank}(A) = \text{rank}(A^T)$ である.

A が正則かどうかは, (部分) 軸選択付き Gauss の消去法を最後まで実行できるか (途中で軸が 0 にならないか) どうかでわかる. A の階数は, 完全軸選択付き Gauss の消去法における軸の数を数えれば得られる.

2.1.4 Gauss-Jordan の消去法

Gauss の消去法では, 前進消去で係数行列を上三角行列にして, あとは後退代入で解を求める方法であった. その代わりに, 消去で係数行列を単位行列にして, 直接的に解を求める方法もある. これを Gauss-Jordan の消去法 (Gauss-Jordan elimination), あるいは掃き出し法 (sweep method) という.

Gauss-Jordan の消去法の手順をアルゴリズム 5 に示す. ここでは, 方程式の $A\mathbf{x} = \mathbf{b}$ の $A, \mathbf{b}$ を並べて, $m \times n$ の行列 $A \leftarrow [A|\mathbf{b}]$ として消去法を適用する.

以下の方程式を考える.

$$\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \end{bmatrix} \quad (16)$$

これから次のような行列を構成する.

$$\left[\begin{array}{cc|c} 2 & 1 & 4 \\ 3 & 2 & 7 \end{array} \right] \quad (17)$$

これに対して消去法を適用すると, 次のようになる.

$$\begin{aligned} & \left[\begin{array}{cc|c} 2 & 1 & 4 \\ 3 & 2 & 7 \end{array} \right] \\ \rightarrow & \left[\begin{array}{cc|c} 1 & 1/2 & 2 \\ 0 & 1/2 & 1 \end{array} \right] \\ \rightarrow & \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array} \right] \end{aligned} \quad (18)$$

右端の列が解になる.

Gauss-Jordan の消去法は, Gauss の消去法に比べると計算量が増える. この方法の利点は, 逆行列を計算するのに利用できることである.

Algorithm 5 Gauss-Jordan の消去法

```
for  $k = 1, 2, \dots, m$  do
   $p = a_{kk}$ 
  for  $j = k, k + 1, \dots, n$  do
     $a_{kj} = \frac{a_{kj}}{p}$ 
  end for
  for  $i = 1, 2, \dots, m$  do
    if  $i \neq k$  then
       $h = a_{ik}$ 
      for  $j = k, k + 1, \dots, n$  do
         $a_{ij} = a_{ij} - a_{kj}h$ 
      end for
    end if
  end for
end for
end for
```

2.1.5 逆行列

正則行列 A の逆行列 A^{-1} を計算することを考える. $AA^{-1} = I$ なので, A^{-1} の列ベクトルをそれぞれ $\mathbf{a}'_i$, 単位行列 I の列ベクトルをそれぞれ $\mathbf{e}_i$ とすると, それぞれの列について $A\mathbf{a}'_i = \mathbf{e}_i$ を解けば逆行列が求まる. したがって, $[A|I]$ に Gauss-Jordan の消去法を適用すると, 逆行列が求まることになる.

以下の行列の逆行列を求める.

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \quad (19)$$

横に単位行列を並べて Gauss-Jordan の消去法を適用する.

$$\begin{aligned} & \left[\begin{array}{cc|cc} 2 & 1 & 1 & 0 \\ 3 & 2 & 0 & 1 \end{array} \right] \\ & \rightarrow \left[\begin{array}{cc|cc} 1 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & -3/2 & 1 \end{array} \right] \\ & \rightarrow \left[\begin{array}{cc|cc} 1 & 0 & 2 & -1 \\ 0 & 1 & -3 & 2 \end{array} \right] \end{aligned} \quad (20)$$

したがって

$$A^{-1} = \begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix} \quad (21)$$

実際

$$\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (22)$$

となる.

2.1.6 スケーリング

$AA^{-1} = I$ なので、逆行列を計算してもとの行列にかけた結果は、対角成分以外 0 になるはずだが、実際には微妙に数値が残ることがある。直接法は手計算と同じ方法とはいえ、数値的に計算するため誤差が生じる。誤差を軽減する方法のひとつとして、スケーリングがある。

方程式 $Ax = b$ について、係数行列の i 行の絶対値の最大値を s_i とすると、消去法の適用前に係数行列および右辺ベクトル成分をそれぞれ $a_{ij} \leftarrow a_{ij}/s_i$, $b_i \leftarrow b_i/s_i$ とする。

以下の例を考える。

$$\left[\begin{array}{cc|cc} 2 & 1 & 1 & 0 \\ 3 & 2 & 0 & 1 \end{array} \right] \quad (23)$$

スケーリングすると、次のようになる。

$$\left[\begin{array}{cc|cc} 1 & 1/2 & 1/2 & 0 \\ 1 & 2/3 & 0 & 1/3 \end{array} \right] \quad (24)$$

スケーリングを行うと、係数行列の行同士の絶対値が近づいて、誤差の減少が期待できる。

2.1.7 行列式

正方行列 A に Gauss の消去法や Gauss-Jordan の消去法を適用する過程を利用すると、行列式 (determinant) を計算できる。行列式の性質として、ある行を c 倍すると c 倍になる、行を交換したら符号が変わる、というものがある。また、対角行列や上三角行列、下三角行列の行列式は、対角要素をすべてかけ合わせたものになる。 A の行列式を $|A|$ とすると、 A に消去法を適用する際、行をそれぞれ $d_1, d_2, \dots, d_m$ で割り、 l 回だけ行の交換を行ったとすると、その結果単位行列になるので、行列式の性質から次式が得られる。

$$\frac{(-1)^l |A|}{d_1 d_2 \cdots d_m} = 1 \quad (25)$$

これより

$$|A| = (-1)^l d_1 d_2 \cdots d_m \quad (26)$$

要するに、消去法の過程で軸をかけ合わせつつ軸の交換回数を数えておいて、上式を計算すれば行列式が得られる。

2.1.8 条件数

消去法の観点からすると、行列にはよい条件 (well condition) のものと悪い条件 (ill condition) のものがある。それを表すのが条件数 (condition number) である。行列 A (正方行列とは限らない) の条件数 $\text{cond}(A)$ は次式で定義される。

$$\text{cond}(A) = \|A\| \|A^{-1}\| \quad (27)$$

ここで、 $\|A\|$ は A のノルム (norm) である。ここでは $\|A\|$ は $p = 2$ ノルムとして、 $A^T A$ の最大固有値の平方根 (A の最大特異値) とする。 $(A^T A)^{-1}$ の最大固有値は $A^T A$ の最小固有値の逆数である。したがって、 $A^T A$ の最大固有値と最小固有値をそれぞれ $\lambda_{\max}$, $\lambda_{\min}$ とすると、条件数は次式で計算できる。

$$\text{cond}(A) = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \quad (28)$$

条件数は誤差の増大率に当たり、できるだけ1に近いほうがよい。たとえば、 A を $A^T A$ にただけで条件は悪くなる。ノルムの性質から $\|A^T A\| \leq \|A^T\| \|A\|$ および $\|A^{-T} A^{-1}\| \leq \|A^{-T}\| \|A^{-1}\|$ なので、条件数が大きくなる。

2.2 分解法

2.2.1 LU 分解

同じ係数行列を使って、異なる右辺ベクトルに対する解を次々に求めたい場合がある。その場合はLU分解 (LU factorization or decomposition) を用いる。LU分解は、係数行列 A を下三角行列 L と上三角行列 U に分解して $A = LU$ とする。

L と U に分解できれば、 $A\mathbf{x} = \mathbf{b}$ は、 $LU\mathbf{x} = \mathbf{b}$ なので、次のように代入操作だけで計算できる。

$$\begin{aligned} L\mathbf{y} &= \mathbf{b} \\ U\mathbf{x} &= \mathbf{y} \end{aligned} \tag{29}$$

1つめの代入を前進代入 (forward substitution), 2つめの代入を後退代入 (これは Gauss の消去法と同じ) という。

$A = LU$ を成分で書くと、次のようになる。

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{30}$$

U の対角成分は1にしてある。たとえば、 a_{32} , a_{33} , a_{34} についてそれぞれ考えると、次のようになる。

$$\begin{aligned} a_{32} &= l_{31}u_{12} + l_{32} \\ a_{33} &= l_{31}u_{13} + l_{32}u_{23} + l_{33} \\ a_{34} &= l_{31}u_{14} + l_{32}u_{24} + l_{33}u_{34} \end{aligned} \tag{31}$$

したがって、 a_{ij} について、次のように表される。

$$\begin{cases} a_{ij} = \sum_{k=1}^{j-1} l_{ik}u_{kj} + l_{ij} & (i \geq j) \\ a_{ij} = \sum_{k=1}^{i-1} l_{ik}u_{kj} + l_{ii}u_{ij} & (i < j) \end{cases} \tag{32}$$

これより、次式が得られる。

$$\begin{cases} l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} & (i \geq j) \\ u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}} & (i < j) \end{cases} \tag{33}$$

LU分解および前進代入、後退代入の手順をそれぞれアルゴリズム 6, 7 に示す。

ここでは L と U を別の配列として扱っているが、処理上はお互いの要素の場所は重なっていないため、同じ $n \times n$ の配列に入れても問題ない (その場合は分解時の $u_{ii} = 1$ の処理を省く)。

たとえば、次のように分解される。

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 3 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \tag{34}$$

Algorithm 6 LU 分解

```
for  $i = 1, 2, \dots, n$  do
   $l_{i1} = a_{i1}$ 
  for  $j = 2, 3, \dots, i$  do
     $l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}$ 
  end for
   $u_{ii} = 1$ 
  for  $j = i + 1, i + 2, \dots, n$  do
     $u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}}$ 
  end for
end for
```

Algorithm 7 前進代入/後退代入 (LU 分解)

```
 $y_1 = \frac{b_1}{l_{11}}$ 
for  $i = 2, 3, \dots, n$  do
   $y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij}y_j}{l_{ii}}$ 
end for
 $x_n = y_n$ 
for  $i = n - 1, n - 2, \dots, 1$  do
   $x_i = y_i - \sum_{j=i+1}^n u_{ij}x_j$ 
end for
```

L と U を同じ配列 (行列) に入れると、次のようになる。

$$\begin{bmatrix} 2 & 1/2 \\ 3 & 1/2 \end{bmatrix} \quad (35)$$

A が正方行列でなくても LU 分解できる。 A のサイズが $m \times n$ だとすると、 $m > n$ のときには L , U のサイズはそれぞれ $m \times n$, $n \times n$ に、 $m < n$ のときにはそれぞれ $m \times m$, $m \times n$ になる。

2.2.2 Cholesky 分解

係数行列 A が対称行列の場合、LU 分解すると $A = LL^T$ になるので、分解は L だけ計算すればよいことになる。この方法を Cholesky 分解 (Cholesky factorization or decomposition) という。 $a_{ij} = a_{ji}$, $u_{ij} = l_{ji}$ として、LU 分解同様に考えればよい。 $i \geq j$ だけ考えればよいので、 a_{ij} は次式で表される。

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik}l_{jk} + l_{ij}l_{jj} \quad (36)$$

Algorithm 8 Cholesky 分解

```
 $l_{11} = \sqrt{a_{11}}$ 
for  $i = 2, 3, \dots, n$  do
   $l_{i1} = \frac{a_{i1}}{l_{11}}$ 
  for  $j = 2, 3, \dots, i-1$  do
     $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}}{l_{jj}}$ 
  end for
   $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}l_{ik}}$ 
end for
```

Algorithm 9 前進代入/後退代入 (Cholesky 分解)

```
 $y_1 = \frac{b_1}{l_{11}}$ 
for  $i = 2, 3, \dots, n$  do
   $y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij}y_j}{l_{ii}}$ 
end for
 $x_n = \frac{y_n}{l_{nn}}$ 
for  $i = n-1, n-2, \dots, 1$  do
   $x_i = \frac{y_i - \sum_{j=i+1}^n l_{ji}x_j}{l_{ii}}$ 
end for
```

これより、次式を得る。

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}}{l_{jj}} \quad (37)$$

対角成分 l_{ii} は、

$$l_{ii} = \frac{a_{ii} - \sum_{k=1}^{i-1} l_{ik}l_{ik}}{l_{ii}} \quad (38)$$

より

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}l_{ik}} \quad (39)$$

Cholesky 分解および前進代入、後退代入の手順をアルゴリズム 8, 9 に示す。この方法では、 A は正定値行列である必要がある。そうでないと平方根の中身が負になりえるからである。なお、 A が正定値行列の場合、軸選択なしでも計算自体は常にできる。

Algorithm 10 修正 Cholesky 分解

```
 $d_1 = a_{11}$   
 $l_{11} = 1$   
for  $i = 2, 3, \dots, n$  do  
  for  $j = 2, 3, \dots, i-1$  do  
     $l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k}{d_j}$   
  end for  
   $d_i = a_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik} d_k$   
   $l_{ii} = 1$   
end for
```

2.2.3 修正 Cholesky 分解

上述の Cholesky 分解で平方根を使わない方法が提案されている。これを修正 Cholesky 分解という。修正 Cholesky 分解では、下三角行列 L と対角行列 D に $A = LDL^T$ の形で分解する。 $a_{ij} = a_{ji}$, $l_{ii} = 1$, $u_{ij} = d_i l_{ji}$ として、LU 分解同様に考える。 $i \geq j$ について、 a_{ij} は次式で表される。

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k + l_{ij} d_j \quad (40)$$

これより、次式を得る。

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k}{d_j} \quad (41)$$

対角成分 d_i は、

$$l_{ii} = \frac{a_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik} d_k}{d_i} = 1 \quad (42)$$

より

$$d_i = a_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik} d_k \quad (43)$$

前進代入と後退代入は次式で行う。

$$\begin{aligned} L^T \mathbf{y} &= \mathbf{b} \\ LD\mathbf{x} &= \mathbf{y} \end{aligned} \quad (44)$$

修正 Cholesky 分解の手順をアルゴリズム 10, 11 に示す。

Algorithm 11 前進代入/後退代入 (修正 Cholesky 分解)

```
y1 =  $\frac{b_1}{d_1}$ 
for  $i = 2, 3, \dots, n$  do
     $y_i = b_i - \sum_{j=1}^{i-1} l_{ij}y_j$ 
end for
 $x_n = \frac{y_n}{d_n}$ 
for  $i = n-1, n-2, \dots, 1$  do
     $x_i = \frac{y_i}{d_i} - \sum_{j=i+1}^n l_{ji}x_j$ 
end for
```

Algorithm 12 修正 Cholesky 分解 2

```
 $l_{11} = a_{11}$ 
 $d_1 = \frac{1}{l_{11}}$ 
for  $i = 2, 3, \dots, n$  do
     $l_{i1} = a_{i1}$ 
    for  $j = 2, 3, \dots, i$  do
         $l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}d_k$ 
    end for
     $d_i = \frac{1}{l_{ii}}$ 
end for
```

2.2.4 修正 Cholesky 分解 2

上の方法では $l_{ii} = 1$ としたが, $l_{ii}d_i = 1$ とする方法もある. この場合, a_{ij} は次式で表される.

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik}l_{jk}d_k + l_{ij} \quad (45)$$

これより, 次式を得る.

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}d_k \quad (46)$$

l_{ii} は次式で計算できる.

$$l_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}l_{ik}d_k \quad (47)$$

d_i は, $l_{ii}d_i = 1$ より $d_i = 1/l_{ii}$ で求まる.

修正 Cholesky 分解 2 の手順をアルゴリズム 12, 13 に示す. 1つ目の修正版より少し単純になっている.

Algorithm 13 前進代入/後退代入 (修正 Cholesky 分解 2)

```

 $y_1 = \frac{b_1}{d_1}$ 
for  $i = 2, 3, \dots, n$  do
     $y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij}y_j}{l_{ii}}$ 
end for
 $x_n = \frac{y_n}{d_n}$ 
for  $i = n-1, n-2, \dots, 1$  do
     $x_i = y_i - d_i \sum_{j=i+1}^n l_{ji}x_j$ 
end for
```

2.2.5 三重対角行列アルゴリズム

対角要素とその両隣の要素のみ非 0 である行列を三重対角行列 (tridiagonal matrix) という。次のような形の行列である。

$$\begin{bmatrix} t_{11} & t_{12} & 0 & 0 \\ t_{21} & t_{22} & t_{23} & 0 \\ 0 & t_{32} & t_{33} & t_{34} \\ 0 & 0 & t_{43} & t_{44} \end{bmatrix} \quad (48)$$

係数行列が三重対角行列であるとき、特別なアルゴリズムが使える。それを三重対角行列アルゴリズム (tridiagonal matrix algorithm, TDMA) という。

単純に言えば、 $A\mathbf{x} = \mathbf{b}$ を $U\mathbf{x} = L^T\mathbf{b}$ に変形して後退代入で解くだけである。ほとんどの要素の値が 0 なので、LU 分解における総和の範囲が狭くて済む。

三重対角行列アルゴリズムの手順をアルゴリズム 14 に示す。ポイントは、 $l_{i,i-1} = a_{i,i-1}$ となるので $l_{i,i-1}$ は計算しなくてよくて、 l_{ij} で必要なのは対角要素のみであること、そのまま前進代入するので l_{ij} を取っておく必要はないこと、 u_{ij} については対角要素が 1 なので対角要素の上側だけ取っておけばよいことである。

2.3 直交化

2.3.1 Gram-Schmidt の直交化

あるデータを表すように関数のパラメータを決定する際、最小二乗法が使われるが、そのとき $A^T A \mathbf{x} = A^T \mathbf{b}$ といった形の方程式を解くことになる (これを正規方程式 normal equation という)。この場合、一般に A 自体の条件があまりよくないが、それがさらに $A^T A$ となるので、条件はかなり悪くなる。この問題への対策として、直交化がある。ここでは A をとりあえず正方行列として、 $A = QR$ のように分解する。この場合 Q は直交行列であり、 $Q^T Q = Q Q^T = I$ である。 R は上三角行列である。これを QR 分解 (QR factorization or

Algorithm 14 三重対角行列アルゴリズム

```

 $l_1 = a_{11}$ 
 $u_1 = \frac{a_{12}}{l_1}$ 
 $y_1 = \frac{b_1}{l_1}$ 
for  $i = 2, 3, \dots, n-1$  do
     $l_i = a_{ii} - a_{i,i-1}u_{i-1}$ 
     $u_i = \frac{a_{i,i+1}}{l_i}$ 
     $y_i = \frac{b_i - a_{i,i-1}y_{i-1}}{l_i}$ 
end for
 $l_n = a_{nn} - a_{n,n-1}u_{n-1}$ 
 $y_n = \frac{b_n - a_{n,n-1}y_{n-1}}{l_n}$ 
 $x_n = y_n$ 
for  $i = n-1, n-2, \dots, 1$  do
     $x_i = y_i - u_i x_{i+1}$ 
end for
```

decomposition) という。これを用いると

$$\begin{aligned} A^T A x &= A^T b \\ \longrightarrow (QR)^T QR x &= (QR)^T b \\ \longrightarrow R^T Q^T QR x &= R^T Q^T b \\ \longrightarrow R^T R x &= R^T Q^T b \\ \longrightarrow R x &= Q^T b \end{aligned} \tag{49}$$

となり、計算が簡単になる。これは数値的に安定である。

直交行列を作るには、行列の列ベクトル同士を直交化する。直交化の方法として、Gram-Schmidt の直交化 (Gram-Schmidt orthogonalization) がある。 A の列ベクトル $\mathbf{a}_i$ から正規直交化したベクトル列 $\mathbf{q}_i$ を作ることを考える。まず、 $\mathbf{a}_1$ を自身のノルム (大きさ) で割って正規化し $\mathbf{q}_1$ を作る。

$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} \tag{50}$$

次に、 $\mathbf{q}_1$ に直交するように、 $\mathbf{a}_2$ から $\mathbf{q}_1$ 方向の成分を引く。

$$\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - (\mathbf{a}_2 \cdot \mathbf{q}_1) \mathbf{q}_1 \tag{51}$$

そして正規化する。

$$\mathbf{q}_2 = \frac{\tilde{\mathbf{q}}_2}{\|\tilde{\mathbf{q}}_2\|} \tag{52}$$

さらに、 $\mathbf{q}_1$ と $\mathbf{q}_2$ に直交するように、 $\mathbf{a}_3$ から $\mathbf{q}_1$ 方向の成分と $\mathbf{q}_2$ 方向の成分を引く。

$$\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - (\mathbf{a}_3 \cdot \mathbf{q}_1) \mathbf{q}_1 - (\mathbf{a}_3 \cdot \mathbf{q}_2) \mathbf{q}_2 \tag{53}$$

正規化する.

$$\mathbf{q}_3 = \frac{\tilde{\mathbf{q}}_3}{\|\tilde{\mathbf{q}}_3\|} \quad (54)$$

これを繰り返す. $\mathbf{q}_i$ は次のようになる.

$$\tilde{\mathbf{q}}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} (\mathbf{a}_i \cdot \mathbf{q}_j) \mathbf{q}_j \quad (55)$$

$$\mathbf{q}_i = \frac{\tilde{\mathbf{q}}_i}{\|\tilde{\mathbf{q}}_i\|} \quad (56)$$

$A = QR$ は次のように表される.

$$[\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_n] = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_n] R \quad (57)$$

直交化の手順から, 次のように書くことができる.

$$\begin{aligned} \mathbf{a}_1 &= \|\mathbf{a}_1\| \mathbf{q}_1 \\ \mathbf{a}_2 &= (\mathbf{a}_2 \cdot \mathbf{q}_1) \mathbf{q}_1 + \|\tilde{\mathbf{q}}_2\| \mathbf{q}_2 \\ \mathbf{a}_3 &= (\mathbf{a}_3 \cdot \mathbf{q}_1) \mathbf{q}_1 + (\mathbf{a}_3 \cdot \mathbf{q}_2) \mathbf{q}_2 + \|\tilde{\mathbf{q}}_3\| \mathbf{q}_3 \\ &\vdots \\ \mathbf{a}_i &= \sum_{j=1}^{i-1} (\mathbf{a}_i \cdot \mathbf{q}_j) \mathbf{q}_j + \|\tilde{\mathbf{q}}_i\| \mathbf{q}_i \end{aligned} \quad (58)$$

R の成分は $\mathbf{q}_i$ にかかる係数である.

$$r_{ji} = \begin{cases} \mathbf{a}_i \cdot \mathbf{q}_j & (j \leq i-1) \\ \|\tilde{\mathbf{q}}_i\| & (j = i) \\ 0 & (j > i) \end{cases} \quad (59)$$

ただし, $\tilde{\mathbf{q}}_1 = \mathbf{a}_1$ とする. 行列で表すと次のような上三角行列になる.

$$R = \begin{bmatrix} \|\tilde{\mathbf{q}}_1\| & \mathbf{a}_2 \cdot \mathbf{q}_1 & \mathbf{a}_3 \cdot \mathbf{q}_1 & \cdots \\ 0 & \|\tilde{\mathbf{q}}_2\| & \mathbf{a}_3 \cdot \mathbf{q}_2 & \cdots \\ 0 & 0 & \|\tilde{\mathbf{q}}_3\| & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (60)$$

R の列ベクトル $\mathbf{r}_i$ は, $\mathbf{a}_i$ について $\mathbf{q}_1, \dots, \mathbf{q}_{i-1}$ を基底として表したときの成分に当たるものを成分として持っているので, $\mathbf{r}_i$ は $\mathbf{a}_i$ を回転したものになっている.

Gram-Schmidt の直交化に基づいた QR 分解の手順をアルゴリズム 15 に示す. $R\mathbf{x} = Q^T \mathbf{b}$ を解く場合は, 後退代入を使えばよい.

Gram-Schmidt の直交化には改良版があり, 修正 Gram-Schmidt 法 (modified Gram-Schmidt method, MGS) などと呼ばれる. 手順をアルゴリズム 16 に示す. r_{ji} の計算のところで $\mathbf{a}_i$ が $\mathbf{v}$ に変わっているだけだが, $\mathbf{v}$ はループのあいだに値を変えていくことに注意. このようにしても結果は変わらない. というのも, $\mathbf{v}$ は $\tilde{\mathbf{q}}_i$ に当たるが, $\mathbf{q}_i$ どうしは直交しているはずなので, 結局 $\tilde{\mathbf{q}}_i \cdot \mathbf{q}_j = \mathbf{a}_i \cdot \mathbf{q}_j$ となるからである. こうしたほうが数値的に性質がよいらしい (数値的には $\mathbf{q}_i$ どうしが直交するとは限らないから?).

一般的な最小二乗法では, A は正方行列とは限らない. A を $m \times n$ (普通は $m \geq n$) 行列とすると, QR 分解において Q の大きさは $m \times n$, R の大きさは $n \times n$ となる. Q は直交行列とは限らないが, $Q^T Q = I$ とすることができる (ただし $Q Q^T = I$ とは限らない).

Algorithm 15 QR 分解 (Gram-Schmidt の直交化)

```
for  $i = 1, 2, \dots, n$  do
   $\mathbf{v} = \mathbf{a}_i$ 
  for  $j = 1, 2, \dots, i - 1$  do
     $r_{ji} = \mathbf{a}_i \cdot \mathbf{q}_j$ 
     $\mathbf{v} = \mathbf{v} - r_{ji}\mathbf{q}_j$ 
  end for
   $r_{ii} = \|\mathbf{v}\|$ 
   $\mathbf{q}_i = \frac{\mathbf{v}}{r_{ii}}$ 
end for
```

Algorithm 16 QR 分解 (修正 Gram-Schmidt 法)

```
for  $i = 1, 2, \dots, n$  do
   $\mathbf{v} = \mathbf{a}_i$ 
  for  $j = 1, 2, \dots, i - 1$  do
     $r_{ji} = \mathbf{v} \cdot \mathbf{q}_j$ 
     $\mathbf{v} = \mathbf{v} - r_{ji}\mathbf{q}_j$ 
  end for
   $r_{ii} = \|\mathbf{v}\|$ 
   $\mathbf{q}_i = \frac{\mathbf{v}}{r_{ii}}$ 
end for
```

2.3.2 Householder 法

実用的な直交化には、数値的な安定性の点で Householder 変換 (Householder transformation) あるいは Householder 鏡映 (Householder reflection) をもとにした Householder 法も用いられる。Householder 鏡映とは、 $H = I - 2\mathbf{u}\mathbf{u}^T$ の形をした変換である。ここで $\mathbf{u}$ は単位ベクトルである。 $H^T = H$ であり、 $H^T H = H H = I$ である。 $H\mathbf{u} = -\mathbf{u}$ なので、 H は $\mathbf{u}$ を $-\mathbf{u}$ に鏡映する。 $\mathbf{u}$ に直交するベクトル $\mathbf{v}$ については、 $H\mathbf{v} = \mathbf{v}$ となる。したがって、 c を係数として、 $\mathbf{v} + c\mathbf{u}$ は H によって $\mathbf{v} - c\mathbf{u}$ に写される。任意のベクトルは $\mathbf{v} + c\mathbf{u}$ の形で表されるので、 H は $\mathbf{v}$ の乗った面を鏡映面として任意のベクトルを $\mathbf{u}$ の反対側へ鏡写しにする。実際、あるベクトル $\mathbf{x}$ について

$$H\mathbf{x} = \mathbf{x} - 2\mathbf{u}\mathbf{u}^T\mathbf{x} = \mathbf{x} - 2(\mathbf{x} \cdot \mathbf{u})\mathbf{u} \quad (61)$$

なので、 $\mathbf{x}$ から $\mathbf{u}$ 方向成分を 2 回引いていることになり (1 回なら直交するだけだが)、 $\mathbf{x}$ は $\mathbf{u}$ の反対方向に写ることになる。

R の列ベクトル $\mathbf{r}_i$ は、 A の列ベクトル $\mathbf{a}_i$ の Householder 鏡映 $H\mathbf{a}_i$ で作ることができる。 $\mathbf{r}_i$ は $\mathbf{a}_i$ を回転したものだからである。本当は回転したいところだが、求めるべき $\mathbf{q}_i$ がわからないことには回転できないの

で、鏡映変換で $\mathbf{a}_i$ をちょこちょこ移動させて最終的な $\mathbf{r}_i$ に近づけていくのである。

Householder 法では、 A に H をかけ続けて R に変換する。つまり

$$R = H_n H_{n-1} \cdots H_1 A \quad (62)$$

とする。 $A = QR$, $Q^T Q = I$ なので

$$Q^T = H_n H_{n-1} \cdots H_1 \quad (63)$$

である。 A が R に近づくような H を見つけていって、それらをかけ合わせれば Q^T が得られるということである。漸化式で表すと、 $R_0 = A$, $Q_0 = I$ として

$$R_k = H_k R_{k-1} \quad (64)$$

$$Q_k^T = H_k Q_{k-1}^T \quad (65)$$

と表現できる。

H_k については、QR 分解の一意性に頼って、最終的に R さえ得られればよいものとして、機械的に R に近づけるような操作とする。 $\mathbf{r}_i$ を $\mathbf{a}_i$ から鏡映で作るとすると、 $\mathbf{u}$ は $\mathbf{w} = \mathbf{a}_i - \mathbf{r}_i$ の方向であり、 $\mathbf{u} = \mathbf{w}/\|\mathbf{w}\|$ である。 $\mathbf{r}_i$ は次式で計算される。

$$\mathbf{r}_i = H \mathbf{a}_i = \mathbf{a}_i - 2\mathbf{u}\mathbf{u}^T \mathbf{a}_i \quad (66)$$

ただし、右辺の $\mathbf{u}$ の中に求めるべき $\mathbf{r}_i$ が含まれている。これを漸化式の形にすると、反復回数を k , R_k の列ベクトルを $\mathbf{r}_i^{(k)}$ として

$$\mathbf{r}_i^{(k)} = H_k \mathbf{r}_i^{(k-1)} = \mathbf{r}_i^{(k-1)} - 2\mathbf{u}_k \mathbf{u}_k^T \mathbf{r}_i^{(k-1)} \quad (67)$$

ここで、 $\mathbf{u}_k = \mathbf{w}_k/\|\mathbf{w}_k\|$, $\mathbf{w}_k = \mathbf{r}_i^{(k-1)} - \mathbf{r}_k$ である。問題は $\mathbf{r}_k$ をどうするかである。

とりあえず、 $\mathbf{r}_1$ を以下のように決める。

$$\mathbf{r}_1 = \begin{bmatrix} \|\mathbf{a}_1\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (68)$$

これを用いて $\mathbf{w}_1 = \mathbf{r}_1^{(0)} - \mathbf{r}_1 = \mathbf{a}_1 - \mathbf{r}_1$ を計算し、 H_1 を求める。 R_1 は次のようになる。

$$R_1 = \begin{bmatrix} \|\mathbf{a}_1\| & r_{12}^{(1)} & r_{13}^{(1)} & \cdots \\ 0 & r_{22}^{(1)} & r_{23}^{(1)} & \cdots \\ 0 & r_{32}^{(1)} & r_{33}^{(1)} & \cdots \\ 0 & r_{42}^{(1)} & r_{43}^{(1)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (69)$$

次は、 $\mathbf{r}_2$ を次のように決める。

$$\tilde{\mathbf{r}}_2 = \begin{bmatrix} r_{12}^{(1)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (70)$$

$$\mathbf{r}_2 = \tilde{\mathbf{r}}_2 + \begin{bmatrix} 0 \\ \|\mathbf{r}_2^{(1)} - \tilde{\mathbf{r}}_2\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (71)$$

ややこしいことをしていそうに見えるが、 $\|\mathbf{r}_2\| = \|\mathbf{r}_2^{(1)}\|$ となるように決めているだけである。 $\mathbf{w}_2 = \mathbf{r}_2^{(1)} - \mathbf{r}_2$ として H_2 から R_2 を計算する。

$$R_2 = \begin{bmatrix} \|\mathbf{a}_1\| & r_{12}^{(1)} & r_{13}^{(1)} & \cdots \\ 0 & \|\mathbf{r}_2^{(1)} - \tilde{\mathbf{r}}_2\| & r_{23}^{(2)} & \cdots \\ 0 & 0 & r_{33}^{(2)} & \cdots \\ 0 & 0 & r_{43}^{(2)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (72)$$

続いて、 $\mathbf{r}_3$ を次のように決める。

$$\tilde{\mathbf{r}}_3 = \begin{bmatrix} r_{13}^{(2)} \\ r_{23}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (73)$$

$$\mathbf{r}_3 = \tilde{\mathbf{r}}_3 + \begin{bmatrix} 0 \\ 0 \\ \|\mathbf{r}_3^{(2)} - \tilde{\mathbf{r}}_3\| \\ \vdots \\ 0 \end{bmatrix} \quad (74)$$

R_3 は

$$R_3 = \begin{bmatrix} \|\mathbf{a}_1\| & r_{12}^{(1)} & r_{13}^{(1)} & \cdots \\ 0 & \|\mathbf{r}_2^{(1)} - \tilde{\mathbf{r}}_2\| & r_{23}^{(2)} & \cdots \\ 0 & 0 & \|\mathbf{r}_3^{(2)} - \tilde{\mathbf{r}}_3\| & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (75)$$

これを繰り返す。 $\mathbf{w}$ をうまく選んであるので、 R の k 行 k 列を含む右下以外の値は変わらない。 実際の計算では右下だけを更新していく。

Householder 法による QR 分解の手順をアルゴリズム 17 に示す。 A は $m \times n$ 行列を想定している。 Householder 法の場合、 Q は $m \times m$ 、 R は $m \times n$ である。 部分的な計算を表現するために、特殊な表記を使っている。 $R(i, j)$ は r_{ij} 、 $\mathbf{w}(i)$ は w_i の意味であるが、ここでは配列のイメージである。 $R(i; m, i)$ は、 R の i 列目の i 行目から m 行目までの部分ベクトルの意味である。

手順の $\mathbf{w}(1) = \mathbf{w}(1) - \|\mathbf{w}\|$ の計算について、右辺第 2 項の符号は任意であるから、数値計算上は桁落ちを避けるために $\mathbf{w}(1) = \mathbf{w}(1) + \text{sign}(\mathbf{w}(1))\|\mathbf{w}\|$ とされる。 ここで $\text{sign}(x)$ は、 x が正なら 1、負なら -1 を返す関数である。

A が正方行列でない場合、 Q 、 R のサイズが不便なので、Gram-Schmidt 法に合わせるなら、 m と n の小さい方を l として、 Q を $m \times l$ 、 R を $l \times n$ のサイズにすればよい。

Algorithm 17 QR 分解 (Householder 法)

```
 $R = A$   
 $Q_t = I$   
for  $i = 1, 2, \dots, n$  do  
   $w = 0$   
   $w(i : m) = R(i : m, i)$   
   $w(i) = w(i) - \|w\|$   
   $u = \frac{w}{\|w\|}$   
   $P = 2uu^T$   
   $R(:, i : n) = R(:, i : n) - PR(:, i : n)$   
   $Q_t = Q_t - PQ_t$   
end for  
 $Q \leftarrow Q_t^T$ 
```

2.3.3 Givens 回転

QR 分解にはもっと単純で直接的な方法もある。それは Givens 回転 (Givens rotation) を用いるものである。Givens 回転は次のような行列による変換である。

$$G = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos \theta & \sin \theta & & \\ & & -\sin \theta & \cos \theta & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \quad (76)$$

ここで、 θ は回転角度だが、直接は扱わない。この変換により、行列の非 0 部分の端っこが 0 になるようにちまちま回転して R に近づけていく。 A に G をかけ続けて

$$R = G_n G_{n-1} \cdots G_1 A \quad (77)$$

なので

$$Q^T = G_n G_{n-1} \cdots G_1 \quad (78)$$

である。漸化式は

$$R_k = G_k R_{k-1} \quad (79)$$

$$Q_k^T = G_k Q_{k-1}^T \quad (80)$$

である。

具体的な変換方法だが、 R_k が次のようになっているとする。

$$R_k = \begin{bmatrix} \vdots & & \vdots & \vdots & \vdots & \\ 0 & \cdots & 0 & r_{ij}^{(k)} & r_{i,j+1}^{(k)} & \cdots \\ 0 & \cdots & 0 & r_{i+1,j}^{(k)} & r_{i+1,j+1}^{(k)} & \cdots \\ 0 & \cdots & 0 & 0 & \vdots & \\ \vdots & & \vdots & \vdots & \vdots & \end{bmatrix} \quad (81)$$

これについて、Givens 回転で $r_{i+1,j}$ を 0 にしたい。

$$r_{i+1,j}^{(k+1)} = -r_{ij}^{(k)} \sin \theta + r_{i+1,j}^{(k)} \cos \theta \quad (82)$$

これが 0 になればよいので

$$\begin{aligned} \cos \theta &= \frac{r_{ij}^{(k)}}{\sqrt{\left(r_{ij}^{(k)}\right)^2 + \left(r_{i+1,j}^{(k)}\right)^2}} \\ \sin \theta &= \frac{r_{i+1,j}^{(k)}}{\sqrt{\left(r_{ij}^{(k)}\right)^2 + \left(r_{i+1,j}^{(k)}\right)^2}} \end{aligned} \quad (83)$$

とすればよい。 r_{ij} については

$$r_{ij}^{(k+1)} = r_{ij}^{(k)} \cos \theta + r_{i+1,j}^{(k)} \sin \theta \quad (84)$$

ということになるが、これは結局

$$r_{ij}^{(k+1)} = \sqrt{\left(r_{ij}^{(k)}\right)^2 + \left(r_{i+1,j}^{(k)}\right)^2} \quad (85)$$

である。

Givens 変換行列 G は、その成分を g_{ij} とすると、 $G = I$ とした上で、ターゲットの $r_{i+1,j}$ に対して次のように値を設定すればよい。

$$\begin{aligned} g_{ii} &= \cos \theta \\ g_{i,i+1} &= \sin \theta \\ g_{i+1,i} &= -\sin \theta \\ g_{i+1,i+1} &= \cos \theta \end{aligned} \quad (86)$$

といっても、必ずしも行列全体を作って計算する必要はなく、以下について $l = j \cdots n$ の範囲で計算すればよい。

$$\begin{aligned} r_{il}^{(k+1)} &= r_{il}^{(k)} \cos \theta + r_{i+1,l}^{(k)} \sin \theta \\ r_{i+1,l}^{(k+1)} &= -r_{il}^{(k)} \sin \theta + r_{i+1,l}^{(k)} \cos \theta \end{aligned} \quad (87)$$

Q^T の計算も同様だが、 $l = 1 \cdots m$ の範囲で計算する。

Givens 回転による QR 分解の手順を 18 に示す。 Q^T の計算の部分は、成分で計算するので、添字を入れ替えて直接 Q を計算している。Householder 法と同様に、 A が正方行列でない場合は Q , R のサイズが不便なので、Gram-Schmidt 法に合わせるなら、 m と n の小さい方を l として、 Q を $m \times l$, R を $l \times n$ のサイズにすればよい。

Givens 回転による QR 分解の利点は、上三角行列に近い形の行列 (具体的には、次に出てくる Hessenberg 行列など) の QR 分解を高速に行えることである。

Algorithm 18 QR 分解 (Givens 回転)

```
 $R = A$ 
 $Q = I$ 
for  $j = 1, 2, \dots, n$  do
  for  $i = m - 1, m - 2, \dots, j$  do
     $t = \sqrt{r_{ij}^2 + r_{i+1,j}^2}$ 
     $c = \frac{r_{ij}}{t}$ 
     $s = \frac{r_{i+1,j}}{t}$ 
     $r_{ij} = t$ 
     $r_{i+1,j} = 0$ 
    for  $k = j + 1, j + 2, \dots, n$  do
       $r_1 = cr_{ik} + sr_{i+1,k}$ 
       $r_{i+1,k} = -sr_{ik} + cr_{i+1,k}$ 
       $r_{ik} = r_1$ 
    end for
    for  $k = 1, 2, \dots, m$  do
       $q_1 = cq_{ki} + sq_{k,i+1}$ 
       $q_{k,i+1} = -sq_{ki} + cq_{k,i+1}$ 
       $q_{ki} = q_1$ 
    end for
  end for
end for
```

2.3.4 Hessenberg 変換

上三角行列 R の代わりに、上三角行列より 1 つ分だけ広い Hessenberg 行列がほしい場合がある。

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,n-1} & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2,n-1} & h_{2n} \\ 0 & h_{32} & \cdots & h_{3,n-1} & h_{3n} \\ 0 & 0 & \cdots & h_{4,n-1} & h_{4n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & h_{n,n-1} & h_{nn} \end{bmatrix} \quad (88)$$

この場合、 A を QH に分解するのではなくて、 $H = Q^T A Q$ と変換する、つまり AQ を QH に分解する。これを Hessenberg 変換という。こうすることで、行列の固有値を変えないまま簡単な行列に変換できるので、行列の固有値計算などに利用される。

A の代わりに AQ を直交化する。Gram-Schmidt の直交化を用いる場合は、 $\mathbf{q}_1$ を適当に決めて、 $A\mathbf{q}_2$, $A\mathbf{q}_3$ と直交化することを考える。この方法を Arnoldi 法という。詳細は Krylov 部分空間法のところで出てくるので、ここでは手順のみアルゴリズム 19 に示す。ここで、 $\mathbf{x}_0$ は適当な初期値で、たとえば第 1 成分の値が 1 で

Algorithm 19 Hessenberg 変換 (Arnoldi 法)

```
 $\mathbf{q}_1 = \frac{\mathbf{x}_0}{\|\mathbf{x}_0\|}$ 
for  $i = 1, 2, \dots, n$  do
   $\mathbf{w} = A\mathbf{q}_i$ 
  for  $j = 1, 2, \dots, i$  do
     $h_{ji} = \mathbf{w} \cdot \mathbf{q}_j$ 
     $\mathbf{w} = \mathbf{w} - h_{ji}\mathbf{q}_j$ 
  end for
  if  $i < n$  then
     $h_{i+1,i} = \|\mathbf{w}\|$ 
     $\mathbf{q}_{i+1} = \frac{\mathbf{w}}{h_{i+1,i}}$ 
  end if
end for
```

残りが 0 のベクトル (標準基底の第 1 基底ベクトル) などでもよい.

Householder 法を用いる場合は, Householder 鏡映を P とすると

$$H = P_n P_{n-1} \cdots P_1 A P_1 \cdots P_{n-1} P_n \quad (89)$$

として H に変換する. 本来は $P^T A P$ なのだが, $P^T = P$ なのでこうなる. 上式が $H = Q^T A Q$ となるので

$$Q = P_1 \cdots P_{n-1} P_n \quad (90)$$

である. 漸化式は, $H_0 = A$, $Q_0 = I$ として

$$H_k = P_k H_{k-1} P_k \quad (91)$$

$$Q_k = Q_{k-1} P_k \quad (92)$$

である.

実際の計算では, 計算量を減らす工夫がされる. PHP は

$$\begin{aligned} H &= PHP \\ &= (I - 2\mathbf{u}\mathbf{u}^T)H(I - 2\mathbf{u}\mathbf{u}^T) \\ &= H - 2H\mathbf{u}\mathbf{u}^T - 2\mathbf{u}\mathbf{u}^T H + 4\mathbf{u}\mathbf{u}^T H\mathbf{u}\mathbf{u}^T \\ &= H - 2\mathbf{u} \{ \mathbf{u}^T H - (\mathbf{u}^T H\mathbf{u})\mathbf{u}^T \} - 2 \{ H\mathbf{u} - \mathbf{u}(\mathbf{u}^T H\mathbf{u}) \} \mathbf{u}^T \end{aligned} \quad (93)$$

ここで

$$\begin{aligned} \mathbf{v}_1 &= 2 \{ \mathbf{u}^T H - (\mathbf{u}^T H\mathbf{u})\mathbf{u}^T \} \\ \mathbf{v}_2 &= 2 \{ H\mathbf{u} - \mathbf{u}(\mathbf{u}^T H\mathbf{u}) \} \end{aligned} \quad (94)$$

として

$$H = H - \mathbf{u}\mathbf{v}_1 - \mathbf{v}_2\mathbf{u}^T \quad (95)$$

Algorithm 20 Hessenberg 変換 (Householder 法)

```
 $H = A$   
 $Q = I$   
for  $i = 1, 2, \dots, n-1$  do  
     $w = 0$   
     $w(i+1:n) = H(i+1:n, i)$   
     $w(i+1) = w(i+1) - \|w\|$   
     $u = \frac{w}{\|w\|}$   
     $P = I - 2uu^T$   
     $H = PHP$   
     $Q = QP$   
end for
```

の形で計算するほうが、乗算回数が少なく済む。対称行列の場合、さらに $v_1 = v_2$ とできる。

Householder 法による Hessenberg 変換の手順をアルゴリズム 20 に示す。

なお、Hessenberg 行列は、Givens 回転により簡単に QR 分解できる (アルゴリズム 21)。

3 反復法

連立方程式を反復計算で解く方法を反復法 (iterative method) という。係数行列が大規模な疎行列の場合、反復法のほうが直接法よりも有利な場合がある。

3.1 反復法の基本

3.1.1 Jacobi 法

単純な反復法を考えると、連立方程式 $Ax = b$ を $x = g(x)$ の形にして繰り返し計算することが考えられる。方程式 $Ax = b$ は、成分で書くと $\sum_{j=1}^n a_{ij}x_j = b_i$ である。これを x_i の形にすると、次のように表される。

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}} \quad (96)$$

反復計算の回数を $k = 0, 1, \dots$ で表し、 k 回計算した結果を x_i^k などと表すと、Jacobi 法は次のような計算となる。

$$x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^n a_{ij}x_j^k}{a_{ii}} \quad (97)$$

初期値 x_i^0 は適当に決める。

行列の形で考えると、これは A を対角行列 D とそれ以外 $A - D$ に分けて考えていることになる。

$$Dx + (A - D)x = b \quad (98)$$

Algorithm 21 Hessenberg 行列の QR 分解 (Givens 回転)

```
 $R = A$ 
 $Q = I$ 
for  $i = 1, 2, \dots, n$  do
  if  $i < m$  then
     $t = \sqrt{r_{ii}^2 + r_{i+1,i}^2}$ 
     $c = \frac{r_{ii}}{t}$ 
     $s = \frac{r_{i+1,i}}{t}$ 
     $r_{ii} = t$ 
     $r_{i+1,i} = 0$ 
    for  $k = i + 1, i + 2, \dots, n$  do
       $r_1 = cr_{ik} + sr_{i+1,k}$ 
       $r_{i+1,k} = -sr_{ik} + cr_{i+1,k}$ 
       $r_{ik} = r_1$ 
    end for
    for  $k = 1, 2, \dots, m$  do
       $q_1 = cq_{ki} + sq_{k,i+1}$ 
       $q_{k,i+1} = -sq_{ki} + cq_{k,i+1}$ 
       $q_{ki} = q_1$ 
    end for
  end if
end for
```

これを整理すると

$$\mathbf{x} = -D^{-1}(A - D)\mathbf{x} + D^{-1}\mathbf{b} \quad (99)$$

あるいは

$$\mathbf{x} = \mathbf{x} + D^{-1}(\mathbf{b} - A\mathbf{x}) \quad (100)$$

反復計算の形では

$$\mathbf{x}_{k+1} = -D^{-1}(A - D)\mathbf{x}_k + D^{-1}\mathbf{b} \quad (101)$$

あるいは

$$\mathbf{x}_{k+1} = \mathbf{x}_k + D^{-1}(\mathbf{b} - A\mathbf{x}_k) \quad (102)$$

残差ベクトルを $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ とすると

$$\mathbf{x}_{k+1} = \mathbf{x}_k + D^{-1}\mathbf{r}_k \quad (103)$$

これは、 $\mathbf{x}_k$ や $\mathbf{r}_k$ に一定のものをかけて $\mathbf{x}_k$ を修正する形になっており、この種の反復法を定常法 (stationary method) という。

Jacobi 法の収束性について見てみる．十分に計算を繰り返したら，値が変化しなくなるはずである．その回数を K とすると

$$\mathbf{x}_K = -D^{-1}(A - D)\mathbf{x}_K + D^{-1}\mathbf{b} \quad (104)$$

である．もとの反復式から上式を引くと

$$\mathbf{x}_{k+1} - \mathbf{x}_K = -D^{-1}(A - D)(\mathbf{x}_k - \mathbf{x}_K) \quad (105)$$

このノルムを考えると，次のように書ける．

$$\|\mathbf{x}_{k+1} - \mathbf{x}_K\| \leq \|D^{-1}(A - D)\|^{k+1} \|\mathbf{x}_0 - \mathbf{x}_K\| \quad (106)$$

反復計算が収束するには，左辺が小さくなっていかないといけないので

$$\|D^{-1}(A - D)\| < 1 \quad (107)$$

であればよい． $D^{-1}(A - D)$ を成分で書くと

$$D^{-1}(A - D) = \begin{bmatrix} 1/a_{11} & 0 & 0 & \cdots \\ 0 & 1/a_{22} & 0 & \cdots \\ 0 & 0 & 1/a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots \\ a_{21} & 0 & a_{23} & \cdots \\ a_{31} & a_{32} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (108)$$

となるので， $p = \infty$ ノルムを用いると

$$\|D^{-1}(A - D)\|_{\infty} = \max_i \sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| \quad (109)$$

要するに

$$\sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1 \quad (110)$$

であれば十分である．これを変形すると

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad (111)$$

である．これは， A のすべての行について，対角要素の絶対値が他の要素の絶対値よりもずっと大きいということ．この性質を対角優位 (diagonally dominant) という．

つまり，Jacobi 法はとりあえず A が対角優位であれば収束する (そうでなくても収束することはあるが)．この性質は，一般的に他の手法でも有利に働く．

3.1.2 Gauss-Seidel 法

Jacobi 法には単純な改良法がある．Jacobi 法の計算では x_i の計算に他の x_j の値が必要だが，すでに計算済みの値があるため，それをそのまま用いるのである．

$$x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k}{a_{ii}} \quad (112)$$

x_i の場合, $j = 1, 2, \dots, i-1$ については x_j の値が計算済みなので, それを用いる. これにより収束が早まることが期待できる. これを Gauss-Seidel 法 (Gauss-Seidel method) という.

実装上は特別なことはなくて, $\mathbf{x}$ のメモリを k 回目と $k+1$ 回目で共有していたら勝手にそうなる.

反復回数の点では一般的に Gauss-Seidel 法のほうが Jacobi 法より有利だが, 並列計算が可能な場合は, 処理に依存性のない Jacobi 法のほうが有利な場合がある.

行列の形で考えると, A を対角行列 D , 下三角行列 L , 上三角行列 U に分けるとすると

$$D\mathbf{x} + (L + U)\mathbf{x} = \mathbf{b} \quad (113)$$

整理して

$$\mathbf{x} = -D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b} \quad (114)$$

ここまでは Jacobi 法と同じなのだが ($A - D = L + U$ なので), Gauss-Seidel 法では反復計算を次のように表す.

$$\mathbf{x}_{k+1} = D^{-1}(-L\mathbf{x}_{k+1} - U\mathbf{x}_k + \mathbf{b}) \quad (115)$$

普通の Gauss-Seidel 法は, $i = 1, 2, \dots, n$ という順番で x_i^{k+1} を求めていくが, 対称行列用に, $i = 1, 2, \dots, n$ と計算したあと逆に $i = n, n-1, \dots, 1$ と計算するところまでを 1 セットとする対称 Gauss-Seidel 法 (symmetric Gauss-Seidel method) というものもある.

3.1.3 SOR 法

Gauss-Seidel 法の収束を加速する方法として, SOR 法 (successive over-relaxation method, 加速緩和法) がある. 緩和法 (relaxation method) というのは, 反復計算で新しい値が $\mathbf{x}_{new}$ と提案されたとき, k 回目の値 $\mathbf{x}_k$ と比べて変化が大きすぎて問題になるとき, この 2 つを混ぜて $k+1$ 回目の値 $\mathbf{x}_{k+1}$ を決める方法である.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha(\mathbf{x}_{new} - \mathbf{x}_k) \quad (116)$$

ここで, α を緩和係数 (relaxation factor) という. これは次式のようにも表される.

$$\mathbf{x}_{k+1} = (1 - \alpha)\mathbf{x}_k + \alpha\mathbf{x}_{new} \quad (117)$$

α の値は 0 から 1 のあいだで取り, $\alpha = 1$ なら新しい値をそのまま採用, $\alpha = 0$ なら全く変化を許容しないことになる (したがって $\alpha = 0$ とすることはない). これは本来, 扱いにくい問題をなんとか扱うためのものである. SOR 法は, これを逆に使って計算を加速させようという方法である. すなわち, α に当たるものに 1 以上の値を設定する.

Gauss-Seidel 法に SOR 法を適用すると, 次のようになる.

$$\begin{aligned} x_i^{new} &= \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k}{a_{ii}} \\ x_i^{k+1} &= x_i^k + \omega(x_i^{new} - x_i^k) \end{aligned} \quad (118)$$

ここで, ω は加速緩和係数 (successive over-relaxation factor) である. ω の値は 0 から 2 のあいだで取られる. ω の最適値は問題によるが, だいたい 1.2 くらいの値が用いられる.

行列の形で書くと, 以下のようになる.

$$\mathbf{x}_{k+1} = (1 - \omega)\mathbf{x}_k + \omega D^{-1}(-L\mathbf{x}_{k+1} - U\mathbf{x}_k + \mathbf{b}) \quad (119)$$

対称 Gauss-Seidel 法同様に、逆順に解くまでを 1 セットにした対称 SOR 法 (symmetric SOR method, SSOR 法) というものもある。

3.1.4 Richardson 反復

もっと単純な反復法を考えることができる。

$$\begin{aligned}
 Ax &= b \\
 \longrightarrow x + Ax &= x + b \\
 \longrightarrow x &= x + b - Ax \\
 \longrightarrow x &= (I - A)x + b
 \end{aligned} \tag{120}$$

これを反復形式にすると

$$x_{k+1} = (I - A)x_k + b \tag{121}$$

あるいは、残差ベクトルを $r = b - Ax$ とすると

$$x_{k+1} = x_k + r_k \tag{122}$$

このままだとたぶん収束しないので、係数 α をかけて $\alpha Ax = \alpha b$ の形で考える。

$$x_{k+1} = x_k + \alpha r_k \tag{123}$$

この形式の反復法を Richardson 反復 (Richardson iteration) という。

残差ベクトルは次のように表される。

$$\begin{aligned}
 r_{k+1} &= b - Ax_{k+1} \\
 &= b - A(x_k + \alpha r_k) \\
 &= b - Ax_k - \alpha Ar_k \\
 &= r_k - \alpha Ar_k
 \end{aligned} \tag{124}$$

形式的には

$$r_{k+1} = (I - \alpha A)r_k \tag{125}$$

となるが、初期残差ベクトルを r_0 とすると

$$\begin{aligned}
 r_{k+1} &= (I - \alpha A)r_k \\
 &= (I - \alpha A)^2 r_{k-1} \\
 &= (I - \alpha A)^3 r_{k-2} \\
 &\dots \\
 &= (I - \alpha A)^{k+1} r_0
 \end{aligned} \tag{126}$$

なので、 A の多項式を $P_{k+1}(A) = (I - \alpha A)^{k+1}$ とすると、残差ベクトルは初期残差ベクトルに A の多項式をかけた形 $r_{k+1} = P_{k+1}(A)r_0$ で表されることになる。この $P_{k+1}(A)$ を残差多項式という。また

$$\begin{aligned}
 x_1 &= x_0 + \alpha r_0 \\
 x_2 &= x_1 + \alpha r_1 = x_0 + \alpha r_0 + \alpha r_1 \\
 &\dots \\
 x_{k+1} &= x_0 + \alpha r_0 + \dots + \alpha r_k
 \end{aligned} \tag{127}$$

Algorithm 22 Richardson 反復

```
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$   
for  $i = 1, 2, \dots, i_{max}$  do  
     $\mathbf{x} = \mathbf{x} + \alpha\mathbf{r}$   
     $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{r}$   
    if  $\|\mathbf{r}\| < \varepsilon$  then  
        loop end  
    end if  
end for
```

なので、解ベクトルと初期値の差 $\mathbf{x}_{k+1} - \mathbf{x}_0$ もまた A の多項式と初期残差ベクトル $\mathbf{r}_0$ で表される。これは、 $\mathbf{r}_k$ および $\mathbf{x}_{k+1} - \mathbf{x}_0$ が $\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^k\mathbf{r}_0$ の線形結合で表されることを意味する。 $\mathbf{x}_0 = \mathbf{0}$ の場合、 $\mathbf{x}_{k+1}$ が $\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^k\mathbf{b}$ で表されることになる。

Richardson 反復の手順はアルゴリズム 22 のようになる。ここで、 i_{max} は反復計算の最大回数、 ε は収束判定の定数である。 $\|\mathbf{r}\|$ はノルム $\|\mathbf{r}\| = \sqrt{\mathbf{r} \cdot \mathbf{r}}$ である。次で述べるように、反復計算の止め方にはいくつかの方法がある。

ここで α を一定にせず、 α_k として反復ごとに変わるようにすると、非定常法 (nonstationary method) が得られる。

3.1.5 反復計算の止め方

反復法には、反復計算をいつ止めるかという問題がある。解ベクトルがある値に収束したことを数値的に判断する必要がある。論点として、何を見て止めるかという点と、それをどう評価するかという点がある。

何を見て止めるかについては、方程式の残差ベクトル $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ が小さくなったかどうかを見ることが考えられる。ただ、問題によっては残差が思ったほど小さくならないことがあり、その場合は $\mathbf{x}$ の変化が小さくなったかどうかで止めることが考えられる。

残差ベクトルにしても解ベクトルの変化にしても、ベクトルであるので、これが小さくなったことをどう確認するかという問題がある。ひとつは、絶対値が最大の要素の値を取ることが考えられる。また、ベクトルの大きさ (ノルム) を取ることも考えられる。

値が小さくなったかどうかをどう判断するかという点も考える必要がある。絶対値が小さくなったかどうかを見るのが基本だが、どれぐらいなら十分かというのを決めかねることがある。また、問題によっては絶対値が小さくならないことがあり、その場合は相対値で見ることもある。

また、ある程度計算が進んだ段階では残差や解の変化は小さくなる方向に進むと期待できるなら、それらが反転して大きくなったら (計算精度の限界に達したと見て) 計算を止めるという方法も考えられる。ただ、常に限界の精度が必要とも限らず、ある程度計算できればよいときもあり (非線形方程式の線形化方程式を解く場合など)、その場合は適当な判定で早めに切り上げることもある。

結局は好みの問題になるが、一般的には、収束の問題で停止判定に悩む必要が生じたら、問題の設定か解法の選択かどちらかを見直したほうがよいと思う。

なお、どうしたって停止判定に至らない場合のために、反復回数には最大値を設定しておく必要がある。

3.2 勾配法

3.2.1 最急降下法

方程式 $Ax = b$ を解くということは、別の言い方をすると、残差ベクトル $r = b - Ax$ が 0 になるような x を見つけるということである。ずばり $r = 0$ となるような x を求めるのは無理なので (できるなら最初からやってる), r ができるだけ小さくなるような x を探すことにする。そこで、ここでは A を (対称) 正定値行列 ($x \cdot Ax > 0$) と仮定して、以下の関数 $F(x)$ の最小化問題を考える。

$$F(x) = \frac{1}{2} x \cdot Ax - x \cdot b \quad (128)$$

話の流れからすると $F(x) = r \cdot r / 2$ とすべきだが、結局同じなので上のようにする。

任意のベクトル $\delta x \neq 0$ について、 x が $x + \delta x$ に変化したとき

$$F(x + \delta x) = F(x) + \frac{1}{2} \delta x \cdot A \delta x + \delta x \cdot (b - Ax) \quad (129)$$

において、 $F(x)$ が最小値だとすると、右辺第 2 項は正なので、 $b - Ax = 0$ でなければならない。というのも、 δx が任意なので、もし $b - Ax$ が値を持っていたら、 $F(x + \delta x) < F(x)$ となるような δx を選べてしまうからである。というわけで、 $F(x)$ の最小化と $Ax = b$ を解くことは同じ問題である。

ちなみに、 $F(x) = r \cdot r / 2$ の場合は

$$F(x + \delta x) = F(x) + \frac{1}{2} A \delta x \cdot A \delta x - A \delta x \cdot (b - Ax) \quad (130)$$

$F(x)$ が最小値だとすると、右辺第 2 項は正なので、やはり $b - Ax = 0$ でなければならない。

さて、 x の探索方向のベクトルを p とする。それに係数 α をかけて x を修正するものとする。反復回数を k とすると

$$x_{k+1} = x_k + \alpha_k p_k \quad (131)$$

とりあえず p はわかっているものとして、最適な α を決める。 $F(x_{k+1})$ は次のように書ける。

$$\begin{aligned} F(x_{k+1}) &= \frac{1}{2} x_{k+1} \cdot A x_{k+1} - x_{k+1} \cdot b \\ &= \frac{1}{2} (x_k + \alpha_k p_k) \cdot A (x_k + \alpha_k p_k) - (x_k + \alpha_k p_k) \cdot b \\ &= F(x_k) + \frac{1}{2} \alpha_k x_k \cdot A p_k + \frac{1}{2} \alpha_k p_k \cdot A x_k + \frac{1}{2} \alpha_k^2 p_k \cdot A p_k - \alpha_k p_k \cdot b \end{aligned} \quad (132)$$

$F(x_{k+1})$ が最小になるように α_k を決めるとすると

$$\frac{\partial F(x_{k+1})}{\partial \alpha_k} = \frac{1}{2} x_k \cdot A p_k + \frac{1}{2} p_k \cdot A x_k + \alpha_k p_k \cdot A p_k - p_k \cdot b = 0 \quad (133)$$

これより

$$\begin{aligned} \alpha_k &= \frac{p_k \cdot b - x_k \cdot A p_k}{p_k \cdot A p_k} \\ &= \frac{p_k \cdot b - p_k \cdot A x_k}{p_k \cdot A p_k} \\ &= \frac{p_k \cdot (b - A x_k)}{p_k \cdot A p_k} \end{aligned} \quad (134)$$

したがって、次式が得られる.

$$\alpha_k = \frac{\mathbf{p}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (135)$$

残差ベクトルは次式のように計算できる.

$$\begin{aligned} \mathbf{r}_{k+1} &= \mathbf{b} - A\mathbf{x}_{k+1} \\ &= \mathbf{b} - A(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \\ &= \mathbf{b} - A\mathbf{x}_k - \alpha_k A\mathbf{p}_k \\ &= \mathbf{r}_k - \alpha_k A\mathbf{p}_k \end{aligned} \quad (136)$$

以上のような方法を勾配法 (gradient method) という.

問題は $\mathbf{p}$ の決め方だが, $\mathbf{x}$ について $F(\mathbf{x})$ が最小になる方向というのは

$$-\frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{b} - A\mathbf{x} = \mathbf{r} \quad (137)$$

である. つまり, $\mathbf{p} = \mathbf{r}$ とするのがひとつの手である. この場合は最急降下法 (steepest descent method) という. これは単純に関数の変化が一番急な方向を選択しているだけであり, あまり効率的な方法ではない.

残差ベクトル $\mathbf{r}$ と探索方向ベクトル $\mathbf{p}$ には, 次の関係がある.

$$\begin{aligned} \mathbf{r}_{k+1} \cdot \mathbf{p}_k &= (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot \mathbf{p}_k \\ &= \mathbf{r}_k \cdot \mathbf{p}_k - \frac{\mathbf{p}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \mathbf{p}_k \cdot A\mathbf{p}_k \\ &= \mathbf{r}_k \cdot \mathbf{p}_k - \mathbf{r}_k \cdot \mathbf{p}_k \\ &= 0 \end{aligned} \quad (138)$$

最急降下法の場合, $\mathbf{p} = \mathbf{r}$ なので, $\mathbf{r}_{k+1} \cdot \mathbf{r}_k = 0$ ということになる. 残差ベクトルの更新は次式で行われる.

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{r}_k \cdot A\mathbf{r}_k} A\mathbf{r}_k \quad (139)$$

この式は, $A\mathbf{r}_k$ を用いて $\mathbf{r}_k$ に直交するように次の残差ベクトル $\mathbf{r}_{k+1}$ を構成するようになっている. $F(\mathbf{x}_{k+1})$ が最小になるように α_k を決める代わりに, 最初から直交条件を設定して α_k を決めても同じということである.

勾配法および最急降下法の手順をそれぞれアルゴリズム 23, 24 に示す. 最急降下法は勾配法で $\mathbf{p} = \mathbf{r}$ としたもののなので単純になっているが, これは Richardson 反復の非定常版になっている. 勾配法は Richardson 反復を一般化したものといえる.

3.2.2 最小残差反復

$F(\mathbf{x})$ の最小化を考える代わりに, $F(\mathbf{r})$ の最小化を考えることもできる. 以下の関数 $F(\mathbf{r})$ の最小化問題を考える.

$$F(\mathbf{r}) = \frac{1}{2} \mathbf{r} \cdot \mathbf{r} \quad (140)$$

任意のベクトル $\delta \mathbf{r} \neq \mathbf{0}$ について, $\mathbf{r}$ が $\mathbf{r} + \delta \mathbf{r}$ に変化したとき

$$F(\mathbf{r} + \delta \mathbf{r}) = F(\mathbf{r}) + \frac{1}{2} \delta \mathbf{r} \cdot \delta \mathbf{r} + \delta \mathbf{r} \cdot \mathbf{r} \quad (141)$$

において, $F(\mathbf{r})$ が最小値だとすると, 右辺第 2 項は正なので, $\mathbf{r} = \mathbf{0}$ でなければならない. $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ なので, $F(\mathbf{r})$ の最小化と $A\mathbf{x} = \mathbf{b}$ を解くことは同じ問題である (ここでは A の正定値性が利用されていないことに注意).

Algorithm 23 勾配法

```
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$   
 $\mathbf{p}$  を計算する  
for  $i = 1, 2, \dots, i_{max}$  do  
   $\alpha$  を計算する  
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$   
   $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{p}$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
   $\mathbf{p}$  を計算する  
end for
```

Algorithm 24 最急降下法

```
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$   
for  $i = 1, 2, \dots, i_{max}$  do  
   $\alpha = \frac{\mathbf{r} \cdot \mathbf{r}}{\mathbf{r} \cdot A\mathbf{r}}$   
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{r}$   
   $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{r}$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
end for
```

$F(\mathbf{r}_{k+1})$ は次のように書ける.

$$\begin{aligned} F(\mathbf{r}_{k+1}) &= \frac{1}{2} \mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1} \\ &= \frac{1}{2} (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \\ &= F(\mathbf{r}_k) - \alpha_k \mathbf{r}_k \cdot A\mathbf{p}_k + \frac{1}{2} \alpha_k^2 A\mathbf{p}_k \cdot A\mathbf{p}_k \end{aligned} \tag{142}$$

$F(\mathbf{r}_{k+1})$ が最小になるように α_k を決めるとする

$$\frac{\partial F(\mathbf{r}_{k+1})}{\partial \alpha_k} = -\mathbf{r}_k \cdot A\mathbf{p}_k + \alpha_k A\mathbf{p}_k \cdot A\mathbf{p}_k = 0 \tag{143}$$

これより

$$\alpha_k = \frac{\mathbf{r}_k \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} \tag{144}$$

Algorithm 25 最小残差反復

```
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ 
for  $i = 1, 2, \dots, i_{max}$  do
   $\alpha = \frac{\mathbf{r} \cdot A\mathbf{r}}{A\mathbf{r} \cdot A\mathbf{r}}$ 
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{r}$ 
   $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{r}$ 
  if  $\|\mathbf{r}\| < \varepsilon$  then
    loop end
  end if
end for
```

$F(\mathbf{x})$ 最小化版では $\mathbf{r}_{k+1} \cdot \mathbf{p}_k = 0$ だったが, $F(\mathbf{r})$ 最小化版でこれに当たる関係は $\mathbf{r}_{k+1} \cdot A\mathbf{p}_k = 0$ である.

$$\begin{aligned}\mathbf{r}_{k+1} \cdot A\mathbf{p}_k &= (\mathbf{r}_k - \alpha_k \cdot A\mathbf{p}_k) \cdot A\mathbf{p}_k \\ &= \mathbf{r}_k \cdot A\mathbf{p}_k - \frac{\mathbf{r}_k \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} A\mathbf{p}_k \cdot A\mathbf{p}_k \\ &= \mathbf{r}_k \cdot A\mathbf{p}_k - \mathbf{r}_k \cdot A\mathbf{p}_k \\ &= 0\end{aligned}\tag{145}$$

$F(\mathbf{r})$ 最小化版の最急降下法は

$$-\frac{\partial F(\mathbf{r})}{\partial \mathbf{r}} = \mathbf{r}\tag{146}$$

なので, 結局 $\mathbf{p} = \mathbf{r}$ であり, $F(\mathbf{x})$ 最小化版との違いは α_k だけである.

$$\alpha_k = \frac{\mathbf{r}_k \cdot A\mathbf{r}_k}{A\mathbf{r}_k \cdot A\mathbf{r}_k}\tag{147}$$

この方法は最小残差反復 (minimul residual iteration) などと呼ばれる (最小残差法と言っても, 別の方法とごっちゃになる).

最小残差反復の手順をアルゴリズム 25 に示す. 最急降下法とは α の式が異なるだけなので, これも Richardson 反復の一種である.

3.3 共役勾配法

勾配法における探索方向ベクトル $\mathbf{p}$ をもう少しきちんと決めるものとして, 共役勾配法 (conjugate gradient method, CG 法) がある. A は (対称) 正定値行列とし, $F(\mathbf{x})$ の最小化を考える.

$\mathbf{p}_{k+1}$ の第一候補は $\mathbf{r}_{k+1}$ だが, 以前の $\mathbf{p}$ を用いて補正するものとして, $\mathbf{p}_{k+1}$ を次のように表す.

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i\tag{148}$$

ところで, 残差ベクトル $\mathbf{r}$ は次のように書ける.

$$\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1} = A(A^{-1}\mathbf{b} - \mathbf{x}_{k+1}) = A\mathbf{e}_{k+1}\tag{149}$$

ここで、 $A^{-1}\mathbf{b} = \mathbf{x}$ は方程式の真の解なので、 $\mathbf{e}_{k+1} = A^{-1}\mathbf{b} - \mathbf{x}_{k+1}$ というのは誤差を意味する。

$\mathbf{r}_{k+1} \cdot \mathbf{p}_k = 0$ なので

$$\mathbf{r}_{k+1} \cdot \mathbf{p}_k = A\mathbf{e}_{k+1} \cdot \mathbf{p}_k = 0 \quad (150)$$

A は対称なので

$$\mathbf{e}_{k+1} \cdot A\mathbf{p}_k = 0 \quad (151)$$

$\mathbf{p}_{k+1}$ は誤差の方向に取りたいので、上式を参考にして次の関係を想定する。

$$\mathbf{p}_{k+1} \cdot A\mathbf{p}_k = 0 \quad (152)$$

これより

$$\begin{aligned} \mathbf{p}_{k+1} \cdot A\mathbf{p}_k &= \left(\mathbf{r}_{k+1} + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i \right) \cdot A\mathbf{p}_k \\ &= \mathbf{r}_{k+1} \cdot A\mathbf{p}_k + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i \cdot A\mathbf{p}_k = 0 \end{aligned} \quad (153)$$

さらに以下の関係も想定する。

$$\mathbf{p}_i \cdot A\mathbf{p}_j = 0 \quad (154)$$

ここで、 i, j は任意の反復回数 (ただし $i \neq j$) である。これを $\mathbf{p}_i$ と $\mathbf{p}_j$ は A について共役 (conjugate) であるという。これにより

$$\mathbf{r}_{k+1} \cdot A\mathbf{p}_k + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i \cdot A\mathbf{p}_k = \mathbf{r}_{k+1} \cdot A\mathbf{p}_k + \beta_{kk} \mathbf{p}_k \cdot A\mathbf{p}_k = 0 \quad (155)$$

したがって

$$\beta_{kk} = -\frac{\mathbf{r}_{k+1} \cdot A\mathbf{p}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \equiv \beta_k \quad (156)$$

共役性により $\beta_{ik} \mathbf{p}_i$ の総和は k の項以外消えるので、 $\mathbf{p}_{k+1}$ を次式で考える。

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \quad (157)$$

ところで

$$\alpha_k = \frac{\mathbf{p}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (158)$$

であったが

$$\begin{aligned} \mathbf{p}_k \cdot \mathbf{r}_k &= (\mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1}) \cdot \mathbf{r}_k \\ &= \mathbf{r}_k \cdot \mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1} \cdot \mathbf{r}_k \end{aligned} \quad (159)$$

$\mathbf{p}_{k-1} \cdot \mathbf{r}_k = 0$ なので

$$\mathbf{p}_k \cdot \mathbf{r}_k = \mathbf{r}_k \cdot \mathbf{r}_k \quad (160)$$

したがって

$$\alpha_k = \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (161)$$

また

$$\begin{aligned} \mathbf{p}_k \cdot A\mathbf{p}_k &= (\mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1}) \cdot A\mathbf{p}_k \\ &= \mathbf{r}_k \cdot A\mathbf{p}_k + \beta_{k-1} \mathbf{p}_{k-1} \cdot A\mathbf{p}_k \end{aligned} \quad (162)$$

$p_{k-1} \cdot Ap_{k-1} = 0$ なので

$$p_k \cdot Ap_k = r_k \cdot Ap_k \quad (163)$$

ここで、残差ベクトルどうしの内積を考えてみる。まず $r_{k+1} \cdot r_k$ を考えると

$$\begin{aligned} r_{k+1} \cdot r_k &= (r_k - \alpha_k Ap_k) \cdot r_k \\ &= r_k \cdot r_k - \frac{r_k \cdot r_k}{p_k \cdot Ap_k} r_k \cdot Ap_k \\ &= r_k \cdot r_k - \frac{r_k \cdot r_k}{p_k \cdot Ap_k} p_k \cdot Ap_k \\ &= r_k \cdot r_k - r_k \cdot r_k \\ &= 0 \end{aligned} \quad (164)$$

また、 i, j を任意の異なる反復回数 (ただし $i \neq j+1$) として

$$\begin{aligned} r_i \cdot r_j &= (r_{i-1} - \alpha_{i-1} Ap_{i-1}) \cdot r_j \\ &= r_{i-1} \cdot r_j - \alpha_{i-1} r_j \cdot Ap_{i-1} \end{aligned} \quad (165)$$

$r_j \cdot Ap_{i-1}$ が問題だが、 $p_i \cdot Ap_j$ を展開すると

$$\begin{aligned} p_i \cdot Ap_j &= (r_i - \beta_{i-1} p_{i-1}) \cdot Ap_j \\ &= r_i \cdot Ap_j - \beta_{i-1} p_{i-1} \cdot Ap_j \end{aligned} \quad (166)$$

$p_{i-1} \cdot Ap_j = 0$ なので (念を押すが $i \neq j+1$)

$$p_i \cdot Ap_j = r_i \cdot Ap_j = 0 \quad (167)$$

これより、次の関係が得られる。

$$r_i \cdot r_j = r_{i-1} \cdot r_j \quad (168)$$

数学的帰納法により、 $i \neq j, i \neq j+1$ のときに、任意の i, j について $r_i \cdot r_j = 0$ であることを証明する。まず、 $r_1 \cdot r_0 = 0$ である。ここで、 $r_k \cdot r_0 = 0$ と仮定すると、上で示した関係から $r_{k+1} \cdot r_0 = 0$ である。また、 $r_k \cdot r_l = 0$ (ただし $l \neq k, l \neq k+1$) を仮定すると $r_{k+1} \cdot r_l = 0$ 。また $r_k \cdot r_{l+1} = 0$ でもあるので、 $r_{k+1} \cdot r_{l+1} = 0$ 。以上の議論から、数学的帰納法の原理より、任意の i, j (ただし $i \neq j, i \neq j+1$) について

$$r_i \cdot r_j = 0 \quad (169)$$

が成り立つ。 $r_{k+1} \cdot r_k = 0$ と合わせると、任意の異なる残差ベクトルどうしはお互いに直交することになる。これはどういうことかというと、 $r_{k+1} \cdot r_k = 0$ なので

$$r_{k+1} = r_k - \alpha_k Ap_k \quad (170)$$

は、 Ap_k を用いて r_k に直交するように r_{k+1} を構成していることになる。さらに任意の残差ベクトルどうしが直交しているということは、 n 次元ベクトルの場合、 n 個の線形独立な残差ベクトルが構成されるということであり、反復計算の過程で $r_0, \dots, r_{n-1}$ が構成されると、 r_n はもはや線形独立ではないので、任意の r_i ($i \neq n$) との内積が 0 になるには $r_n = 0$ でなければならない。これは、理論的には n 回の計算で解が求まることを意味する。

α_k から

$$p_k \cdot Ap_k = \frac{r_k \cdot r_k}{\alpha_k} \quad (171)$$

Algorithm 26 共役勾配法

```
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}$   
 $\mathbf{p} = \mathbf{r}_0$   
for  $i = 1, 2, \dots, i_{max}$  do  
   $\alpha = \frac{\mathbf{r}_0 \cdot \mathbf{r}_0}{\mathbf{p} \cdot A\mathbf{p}}$   
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$   
   $\mathbf{r} = \mathbf{r}_0 - \alpha A\mathbf{p}$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
   $\beta = \frac{\mathbf{r} \cdot \mathbf{r}}{\mathbf{r}_0 \cdot \mathbf{r}_0}$   
   $\mathbf{p} = \mathbf{r} + \beta\mathbf{p}$   
   $\mathbf{r}_0 = \mathbf{r}$   
end for
```

また

$$\begin{aligned}\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1} &= (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot \mathbf{r}_{k+1} \\ &= \mathbf{r}_k \cdot \mathbf{r}_{k+1} - \alpha_k \mathbf{r}_{k+1} \cdot A\mathbf{p}_k \\ &= -\alpha_k \mathbf{r}_{k+1} \cdot A\mathbf{p}_k\end{aligned}\tag{172}$$

より

$$\mathbf{r}_{k+1} \cdot A\mathbf{p}_k = -\frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{\alpha_k}\tag{173}$$

これらより、 β_k は次のように表される。

$$\beta_k = \frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{\mathbf{r}_k \cdot \mathbf{r}_k}\tag{174}$$

以上を整理すると、共役勾配法の手順はアルゴリズム 26 のようになる。

3.4 前処理付き共役勾配法

共役勾配法は、係数行列が $n \times n$ の場合、理論的には n 回計算すれば解が得られるという意味で、厳密には反復計算の形をした直接法的一种である。ただ、実用的には n 回よりもはるかに小さな回数での収束を目指すという意味で、反復法的一种とみなされている。収束性を改善する方法の一つとして、前処理 (preconditioning) がある。前処理を用いた共役勾配法を前処理付き共役勾配法 (preconditioned conjugate gradient method, PCG 法) という。

前処理行列 M という正則行列を考えて、方程式の両辺に M^{-1} をかける。

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}\tag{175}$$

$M^{-1}A$ が単位行列に近くなるようにできれば、反復計算の収束が速くなると期待できる。係数行列の右側からかけるように

$$AM^{-1}(M\mathbf{x}) = \mathbf{b}\tag{176}$$

とすることもある。また、両方用いて

$$M^{-1}AM^{-1}(M\mathbf{x}) = M^{-1}\mathbf{b} \quad (177)$$

とする場合もある。上式は形式的なもので、実際にはもっと柔軟に対応する。

共役勾配法の場合、(対称) 正定値性が保たれるように係数行列を変形する必要がある。

3.4.1 スケーリング

単純な前処理として、スケーリングがある。次の成分をもつ対角行列 D を考える。

$$d_{ij} = \sqrt{a_{ij}}\delta_{ij} \quad (178)$$

ここで δ_{ij} はクロネッカーのデルタである。要するに A の対角成分を取り出して平方根を取ったものが D である。前処理行列として $M = D$ とする。方程式を以下のように変形する。

$$(D^{-1}AD^{-1})(D\mathbf{x}) = D^{-1}\mathbf{b} \quad (179)$$

D^{-1} の成分は $(1/\sqrt{a_{ij}})\delta_{ij}$ なので、簡単に計算できる。 $A' = D^{-1}AD^{-1}$, $\mathbf{x}' = D\mathbf{x}$, $\mathbf{b}' = D^{-1}\mathbf{b}$ として

$$A'\mathbf{x}' = \mathbf{b}' \quad (180)$$

を解くことになる。最終結果は $\mathbf{x} = D^{-1}\mathbf{x}'$ で得られる。

3.4.2 不完全 Cholesky 分解

前処理として不完全 Cholesky 分解を用いる共役勾配法を ICCG 法 (incomplete Cholesky conjugate gradient method) という。Cholesky 分解は対称行列を分解するあれだが、不完全 (incomplete) というのは、行列の値が 0 の要素が分解の過程で非 0 になってしまうことをフィルイン (fill-in) というが、このフィルインを無視するという意味で不完全ということである。

修正 Cholesky 分解を考えると、係数行列は $A = LDL^T$ と分解される。不完全 Cholesky 分解なら $A \approx LDL^T$ である。 $D^{-1}L^{-1}AL^{-T} \approx I$ なので、 A をこれに近い形にすれば収束性が改善されそうである。係数行列は対称にしたいので、

$$A \approx LDL^T = LD^{1/2}D^{1/2}L^T = (LD^{1/2})(LD^{1/2})^T \quad (181)$$

と考える。つまり、 $(LD^{1/2})^{-1}A(LD^{1/2})^{-T} \approx I$ と変形する。

方程式を次の形で表す。

$$(LD^{1/2})^{-1}A(LD^{1/2})^{-T}(LD^{1/2})^T\mathbf{x} = (LD^{1/2})^{-1}\mathbf{b} \quad (182)$$

$A' = (LD^{1/2})^{-1}A(LD^{1/2})^{-T}$, $\mathbf{x}' = (LD^{1/2})^T\mathbf{x}$, $\mathbf{b}' = (LD^{1/2})^{-1}\mathbf{b}$ とすると $A'\mathbf{x}' = \mathbf{b}'$ の形になる。これに共役勾配法を適用する。実際には、方程式を変換して共役勾配法を適用するのではなく、変換を考慮した形の共役勾配法を用いる。

変形後の残差ベクトルを $\mathbf{r}'$ とすると

$$\mathbf{r}' = \mathbf{b}' - A'\mathbf{x}' = (LD^{1/2})^{-1}\mathbf{r} \quad (183)$$

変形後の探索ベクトル $\mathbf{p}'$ は、 $\mathbf{p}'_i \cdot A'\mathbf{p}'_j = \mathbf{p}_i \cdot A\mathbf{p}_j = 0$ ($i \neq j$) となるように、以下のようにする。

$$\mathbf{p}' = (LD^{1/2})^T\mathbf{p} \quad (184)$$

$\mathbf{p}'_0 = \mathbf{r}'_0$ なので

$$(LD^{1/2})^T \mathbf{p}_0 = (LD^{1/2})^{-1} \mathbf{r}_0 \quad (185)$$

これより

$$\begin{aligned} (LD^{1/2})^T \mathbf{p}_0 &= (LD^{1/2})^{-1} \mathbf{r}_0 \\ \longrightarrow LD^{1/2} (LD^{1/2})^T \mathbf{p}_0 &= \mathbf{r}_0 \\ \longrightarrow LD^{1/2} D^{1/2} L^T \mathbf{p}_0 &= \mathbf{r}_0 \\ \longrightarrow LDL^T \mathbf{p}_0 &= \mathbf{r}_0 \end{aligned} \quad (186)$$

したがって

$$\mathbf{p}_0 = (LDL^T)^{-1} \mathbf{r}_0 \quad (187)$$

$\mathbf{r}'_k \cdot \mathbf{r}'_k$ は

$$\begin{aligned} \mathbf{r}'_k \cdot \mathbf{r}'_k &= (LD^{1/2})^{-1} \mathbf{r}_k \cdot (LD^{1/2})^{-1} \mathbf{r}_k \\ &= \left\{ (LD^{1/2})^{-1} \mathbf{r}_k \right\}^T (LD^{1/2})^{-1} \mathbf{r}_k \\ &= (\mathbf{r}_k)^T (LD^{1/2})^{-T} (LD^{1/2})^{-1} \mathbf{r}_k \end{aligned} \quad (188)$$

ここで $\mathbf{z} = (LD^{1/2})^{-T} (LD^{1/2})^{-1} \mathbf{r}$ とおくと

$$\begin{aligned} \mathbf{z} &= (LD^{1/2})^{-T} (LD^{1/2})^{-1} \mathbf{r} \\ \longrightarrow (LD^{1/2})^T \mathbf{z} &= (LD^{1/2})^{-1} \mathbf{r} \\ \longrightarrow LD^{1/2} (LD^{1/2})^T \mathbf{z} &= \mathbf{r} \\ \longrightarrow LD^{1/2} D^{1/2} L^T \mathbf{z} &= \mathbf{r} \\ \longrightarrow LDL^T \mathbf{z} &= \mathbf{r} \\ \longrightarrow \mathbf{z} &= (LDL^T)^{-1} \mathbf{r} \end{aligned} \quad (189)$$

したがって

$$\mathbf{r}'_k \cdot \mathbf{r}'_k = \mathbf{r}_k \cdot (LDL^T)^{-1} \mathbf{r}_k \quad (190)$$

また, $\mathbf{p}'_k \cdot A' \mathbf{p}'_k = \mathbf{p}_k \cdot A \mathbf{p}_k$ なので, 変換後の α'_k は

$$\alpha'_k = \frac{\mathbf{r}'_k \cdot \mathbf{r}'_k}{\mathbf{p}'_k \cdot A' \mathbf{p}'_k} = \frac{\mathbf{r}_k \cdot (LDL^T)^{-1} \mathbf{r}_k}{\mathbf{p}_k \cdot A \mathbf{p}_k} \quad (191)$$

同様に, 変換後の β'_k は

$$\beta'_k = \frac{\mathbf{r}'_{k+1} \cdot \mathbf{r}'_{k+1}}{\mathbf{r}' \cdot \mathbf{r}'} = \frac{\mathbf{r}_{k+1} \cdot (LDL^T)^{-1} \mathbf{r}_{k+1}}{\mathbf{r}_k \cdot (LDL^T)^{-1} \mathbf{r}_k} \quad (192)$$

$\mathbf{x}$ の更新は

$$\begin{aligned} \mathbf{x}'_{k+1} &= \mathbf{x}'_k + \alpha'_k \mathbf{p}'_k \\ \longrightarrow (LD^{1/2})^T \mathbf{x}_{k+1} &= (LD^{1/2})^T \mathbf{x}_k + \alpha'_k (LD^{1/2})^T \mathbf{p}_k \\ \longrightarrow \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha'_k \mathbf{p}_k \end{aligned} \quad (193)$$

これより, 残差ベクトルの更新は

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha'_k A \mathbf{p}_k \quad (194)$$

Algorithm 27 ICCG 法

不完全 Cholesky 分解 $A \rightarrow LDL^T$

$$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}$$

$$\mathbf{u}_0 = (LDL^T)^{-1}\mathbf{r}_0$$

$$\mathbf{p} = \mathbf{u}_0$$

for $i = 1, 2, \dots, i_{max}$ **do**

$$\alpha = \frac{\mathbf{r}_0 \cdot \mathbf{u}_0}{\mathbf{p} \cdot A\mathbf{p}}$$

$$\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$$

$$\mathbf{r} = \mathbf{r}_0 - \alpha A\mathbf{p}$$

if $\|\mathbf{r}\| < \varepsilon$ **then**

 loop end

end if

$$\mathbf{u} = (LDL^T)^{-1}\mathbf{r}$$

$$\beta = \frac{\mathbf{r} \cdot \mathbf{u}}{\mathbf{r}_0 \cdot \mathbf{u}_0}$$

$$\mathbf{p} = \mathbf{u} + \beta\mathbf{p}$$

$$\mathbf{r}_0 = \mathbf{r}$$

$$\mathbf{u}_0 = \mathbf{u}$$

end for

$\mathbf{p}$ の更新は

$$\begin{aligned} \mathbf{p}'_{k+1} &= \mathbf{r}'_k + \beta'_k \mathbf{p}'_k \\ \longrightarrow (LD^{1/2})^T \mathbf{p}_{k+1} &= (LD^{1/2})^{-1} \mathbf{r}_k + \beta'_k (LD^{1/2})^T \mathbf{p}_k \\ \longrightarrow \mathbf{p}_{k+1} &= (LDL^T)^{-1} \mathbf{r}_k + \beta'_k \mathbf{p}_k \end{aligned} \quad (195)$$

以上より、ICCG 法の手順はアルゴリズム 27 のようになる． $(LDL^T)^{-1}\mathbf{r}$ の計算は前進代入，後退代入を用いる．不完全 Cholesky 分解については， A の要素の値が 0 の位置の計算を飛ばせばよい．

3.5 共役残差法

共役勾配法と似た方法で，共役残差法 (conjugate residual method, CR 法) がある． A は正定値とは限らない行列とし， $F(\mathbf{r})$ の最小化を考える．

共役勾配法と同様に，最急降下法の探索方向ベクトル $\mathbf{p}$ の更新を次式で想定する．

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i \quad (196)$$

$F(\mathbf{r})$ の最小化の場合， $\mathbf{r}_{k+1} \cdot A\mathbf{p}_k = 0$ が成り立つが， $\mathbf{r}_{k+1} = A\mathbf{e}_{k+1}$ だったので

$$A\mathbf{e}_{k+1} \cdot A\mathbf{p}_k = 0 \quad (197)$$

が成り立つ． $\mathbf{p}_{k+1}$ を誤差の方向に取りたいので，以下の関係を仮定する．

$$A\mathbf{p}_{k+1} \cdot A\mathbf{p}_k = 0 \quad (198)$$

さらに、共役勾配法同様、以下の関係も仮定する。

$$A\mathbf{p}_i \cdot A\mathbf{p}_j = 0 \quad (199)$$

ここで、 i, j は任意の反復回数 (ただし $i \neq j$) である。これにより

$$\begin{aligned} A\mathbf{p}_{k+1} \cdot A\mathbf{p}_i &= A \left(\mathbf{r}_{k+1} + \sum_{l=0}^k \beta_{lk} \mathbf{p}_l \right) \cdot A\mathbf{p}_i \\ &= A\mathbf{r}_{k+1} \cdot A\mathbf{p}_i + \sum_{l=0}^k \beta_{lk} A\mathbf{p}_l \cdot A\mathbf{p}_i \\ &= A\mathbf{r}_{k+1} \cdot A\mathbf{p}_i + \beta_{ik} A\mathbf{p}_i \cdot A\mathbf{p}_i = 0 \end{aligned} \quad (200)$$

ここで i は任意の反復回数 (ただし $i \leq k$) である。したがって

$$\beta_{ik} = -\frac{A\mathbf{r}_{k+1} \cdot A\mathbf{p}_i}{A\mathbf{p}_i \cdot A\mathbf{p}_i} \quad (201)$$

実際には、過去の $\mathbf{p}$ を $p (\leq k+1)$ 個だけ取ることがある。すなわち

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=k-p+1}^k \beta_{ik} \mathbf{p}_i \quad (202)$$

また、共役残差法では $A\mathbf{r}_{k+1}$ を計算する必要があるが、これを $A\mathbf{p}_{k+1}$ の計算に利用できる。

$$A\mathbf{p}_{k+1} = A\mathbf{r}_{k+1} + \sum_{i=k-p+1}^k \beta_{ik} A\mathbf{p}_i \quad (203)$$

$p = 1$ の場合を考える。

$$\alpha_k = \frac{\mathbf{r}_k \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (204)$$

であったが

$$\begin{aligned} \mathbf{r}_k \cdot A\mathbf{p}_k &= \mathbf{r}_k \cdot A(\mathbf{r}_k + \beta_{kk} \mathbf{p}_{k-1}) \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k + \beta_{kk} \mathbf{r}_k \cdot A\mathbf{p}_{k-1} \end{aligned} \quad (205)$$

$\mathbf{r}_k \cdot A\mathbf{p}_{k-1} = 0$ なので

$$\mathbf{r}_k \cdot A\mathbf{p}_k = \mathbf{r}_k \cdot A\mathbf{r}_k \quad (206)$$

したがって

$$\alpha_k = \frac{\mathbf{r}_k \cdot A\mathbf{r}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (207)$$

また

$$\begin{aligned} \mathbf{r}_{k+1} \cdot A\mathbf{r}_{k+1} &= (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot A\mathbf{r}_{k+1} \\ &= \mathbf{r}_k \cdot A\mathbf{r}_{k+1} - \alpha_k A\mathbf{r}_{k+1} \cdot A\mathbf{p}_k \end{aligned} \quad (208)$$

$\mathbf{r}_k \cdot A\mathbf{r}_{k+1} = 0$ なので

$$\mathbf{r}_{k+1} \cdot A\mathbf{r}_{k+1} = -\alpha_k A\mathbf{r}_{k+1} \cdot A\mathbf{p}_k \quad (209)$$

したがって

$$\alpha_k = -\frac{\mathbf{r}_{k+1} \cdot A\mathbf{r}_{k+1}}{A\mathbf{r}_{k+1} \cdot A\mathbf{p}_k} \quad (210)$$

Algorithm 28 共役残差法

```
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}$   
 $\mathbf{p} = \mathbf{r}_0$   
for  $i = 1, 2, \dots, i_{max}$  do  
   $\alpha = \frac{\mathbf{r}_0 \cdot A\mathbf{p}}{A\mathbf{p} \cdot A\mathbf{p}}$   
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$   
   $\mathbf{r} = \mathbf{r}_0 - \alpha A\mathbf{p}$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
   $\beta = \frac{\mathbf{r} \cdot A\mathbf{r}}{\mathbf{r}_0 \cdot A\mathbf{r}_0}$   
   $\mathbf{p} = \mathbf{r} + \beta\mathbf{p}$   
   $\mathbf{r}_0 = \mathbf{r}$   
end for
```

これらより, β_{kk} は次のように表すことができる.

$$\beta_{kk} = -\frac{A\mathbf{r}_{k+1} \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} = \frac{\mathbf{r}_{k+1} \cdot A\mathbf{r}_{k+1}}{\mathbf{r}_k \cdot A\mathbf{r}_k} \quad (211)$$

共役残差法の手順について, $p = 1$ の場合と任意の p を取った場合をそれぞれアルゴリズム 28, 29 に示す. 一般的には, $p > 1$ の場合を切断 (truncated) 版 Orthomin 法, 特に $p = i$ の場合を一般化共役残差法 (generalized conjugate residual method, GCR 法) という.

共役勾配法では $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ ($i \neq j$) という性質があったが, 共役残差法ではその代わりに $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ ($i > j$) が成り立つ. まず, $\mathbf{r}_{k+1} \cdot A\mathbf{r}_k = 0$ についてだが, いくつか準備がいる. $\mathbf{r}_{k+1} \cdot A\mathbf{p}_i$ ($i < k$) について

$$\begin{aligned} \mathbf{r}_{k+1} \cdot A\mathbf{p}_i &= (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot A\mathbf{p}_i \\ &= \mathbf{r}_k \cdot A\mathbf{p}_i - \alpha_k A\mathbf{p}_k \cdot A\mathbf{p}_i \\ &= \mathbf{r}_k \cdot A\mathbf{p}_i \end{aligned} \quad (212)$$

これを繰り返し適用すると

$$\mathbf{r}_{k+1} \cdot A\mathbf{p}_i = \mathbf{r}_{i+1} \cdot A\mathbf{p}_i = 0 \quad (213)$$

である. $\mathbf{r}_{k+1} \cdot A\mathbf{p}_k = 0$ と合わせて $\mathbf{r}_{k+1} \cdot A\mathbf{p}_i = 0$ ($i < k+1$), 要するに $\mathbf{r}_i \cdot A\mathbf{p}_j = 0$ ($i > j$) である. また

$$\begin{aligned} A\mathbf{p}_k \cdot A\mathbf{p}_k &= A \left(\mathbf{r}_k + \sum_{i=0}^{k-1} \beta_{i,k-1} \mathbf{p}_i \right) \cdot A\mathbf{p}_k \\ &= A\mathbf{r}_k \cdot A\mathbf{p}_k + \sum_{i=0}^{k-1} \beta_{i,k-1} A\mathbf{p}_i \cdot A\mathbf{p}_k \\ &= A\mathbf{r}_k \cdot A\mathbf{p}_k \end{aligned} \quad (214)$$

Algorithm 29 一般化共役残差法

```
 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$   
 $\mathbf{p} = \mathbf{r}$   
for  $i = 1, 2, \dots, i_{max}$  do  
   $\alpha = \frac{\mathbf{r} \cdot A\mathbf{p}}{A\mathbf{p} \cdot A\mathbf{p}}$   
   $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$   
   $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{p}$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
   $j_0 = \max(i - p, 0)$   
  for  $j = j_0, \dots, i - 1$  do  
     $\beta_j = -\frac{A\mathbf{r} \cdot A\mathbf{p}_j}{A\mathbf{p}_j \cdot A\mathbf{p}_j}$   
  end for  
   $\mathbf{p} = \mathbf{r} + \sum_{j=j_0}^{i-1} \beta_j \mathbf{p}_j$   
end for
```

さらに

$$\begin{aligned} \mathbf{r}_k \cdot A\mathbf{p}_k &= \mathbf{r}_k \cdot A \left(\mathbf{r}_k + \sum_{i=0}^{k-1} \beta_{i,k-1} \mathbf{p}_i \right) \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k + \sum_{i=0}^{k-1} \beta_{i,k-1} \mathbf{r}_k \cdot A\mathbf{p}_i \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k \end{aligned} \tag{215}$$

以上より

$$\begin{aligned} \mathbf{r}_{k+1} \cdot A\mathbf{r}_k &= (\mathbf{r}_k - \alpha_k A\mathbf{p}_k) \cdot A\mathbf{r}_k \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k - \frac{\mathbf{r}_k \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} A\mathbf{p}_k \cdot A\mathbf{r}_k \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k - \frac{\mathbf{r}_k \cdot A\mathbf{r}_k}{A\mathbf{r}_k \cdot A\mathbf{p}_k} A\mathbf{r}_k \cdot A\mathbf{p}_k \\ &= \mathbf{r}_k \cdot A\mathbf{r}_k - \mathbf{r}_k \cdot A\mathbf{r}_k \\ &= 0 \end{aligned} \tag{216}$$

つぎは $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ ($i > j$) についてだが, まず, $i > j$ として

$$\begin{aligned} A\mathbf{p}_i \cdot A\mathbf{p}_j &= A\mathbf{p}_i \cdot A \left(\mathbf{r}_j + \sum_{l=0}^{j-1} \beta_{l,j-1} \mathbf{p}_l \right) \\ &= A\mathbf{p}_i \cdot A\mathbf{r}_j + \sum_{l=0}^{j-1} \beta_{l,j-1} A\mathbf{p}_i \cdot A\mathbf{p}_l \\ &= A\mathbf{p}_i \cdot A\mathbf{r}_j = 0 \end{aligned} \tag{217}$$

これより

$$\begin{aligned} \mathbf{r}_i \cdot A\mathbf{r}_j &= (\mathbf{r}_{i-1} - \alpha_{i-1}A\mathbf{p}_{i-1}) \cdot A\mathbf{r}_j \\ &= \mathbf{r}_{i-1} \cdot A\mathbf{r}_j - \alpha_{i-1}A\mathbf{p}_{i-1} \cdot A\mathbf{r}_j \end{aligned} \quad (218)$$

したがって, $i > j + 1$ について, 次式の関係が得られる.

$$\mathbf{r}_i \cdot A\mathbf{r}_j = \mathbf{r}_{i-1} \cdot A\mathbf{r}_j \quad (219)$$

あとは, 共役勾配法における $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ ($i \neq j + 1$) の場合と同様に, $i > j + 1$ について $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ がいえる. $\mathbf{r}_{k+1} \cdot A\mathbf{r}_k = 0$ と合わせると, $i > j$ なる任意の i, j について $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ が成り立つ.

これはどういうことかという, 初期残差ベクトル $\mathbf{r}_0$ を考えると, それ以外のすべての残差ベクトルが $A\mathbf{r}_0$ に直交するということである. n 次元ベクトルの場合, $A\mathbf{r}_0$ に直交するように異なる残差ベクトル $\mathbf{r}_1, \dots, \mathbf{r}_{n-1}$ を構成していくと, $\mathbf{r}_n = 0$ となるので, そのうち解が求まるということである (たぶん).

$p = 1$ の場合は, 任意の i, j ($i \neq j$) で $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ が成り立つ. 任意の i, j ($i \neq j$) で $A\mathbf{p}_i \cdot A\mathbf{p}_j = A\mathbf{p}_i \cdot A\mathbf{r}_j = 0$ が成り立つためである.

共役残差法は, $\mathbf{r} \cdot \mathbf{r}/2$ を最小化する方向に計算を進めるため, (理屈の上では) 残差が単調に減少する. また, 係数行列に (対称) 正定値性を仮定していないので, 対称非正定値や非対称の係数行列をもつ方程式も解ける可能性がある. ただし, p をそれなりの値にする必要があり, その分計算量とメモリが多くなる. 共役勾配法と比べると, $p = 1$ でも少し計算量が大きい.

3.6 Krylov 部分空間法

Richardson 反復では, 残差ベクトル $\mathbf{r}_k$ および解ベクトル $\mathbf{x}_{k+1} - \mathbf{x}_0$ が, 係数行列 A と初期残差ベクトル $\mathbf{r}_0$ により $\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^k\mathbf{r}_0$ の線形結合で表されるのであった. 勾配法は Richardson 反復の一種であり, 共役勾配法と共役残差法は勾配法から派生したのであった. したがって, 共役勾配法や共役残差法を $\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^k\mathbf{r}_0$ の線形結合の枠組みで考えることができる. この枠組みで作られた手法を総称して Krylov 部分空間法 (Krylov subspace method) という.

ベクトル列 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ の線形結合で作られる空間を

$$\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\} \quad (220)$$

と表す. ベクトル $\mathbf{v}$ に行列 A のべき乗をかけて生成されるベクトル列 $\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots$ によって張られる空間の部分空間

$$\mathcal{K}_k(A, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^{k-1}\mathbf{v}\} \quad (221)$$

を Krylov 部分空間という. Krylov 部分空間法は

$$\mathcal{K}_k(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0\} \quad (222)$$

を考えて, $\mathbf{r}_k, \mathbf{x}_{k+1} - \mathbf{x}_0 \in \mathcal{K}_{k+1}(A, \mathbf{r}_0)$ となるように $\mathbf{x}_{k+1}$ を求める手順を構成する手法である.

$\mathbf{x}_{k+1}$ を求める手順の構成方法にはいくつかの方式がある. まず, 残差ベクトルと部分空間の直交条件

$$\mathbf{r}_k \perp \mathcal{K}_k(A, \mathbf{r}_0) \quad (223)$$

を想定する Ritz-Galerkin 方式がある. 共役勾配法がこれに当たる.

残差ベクトルのノルムを最小化する最小残差方式もある。これは、以下の直交条件を考えるのと同等である。

$$\mathbf{r}_k \perp A\mathcal{K}_k(A, \mathbf{r}_0) \quad (224)$$

共役残差法がこれに当たる。

また、残差ベクトルと別の適当な部分空間との直交条件を考える Petrov-Galerkin 方式というものもある。これは、共役勾配法の非対称化に利用される。

3.6.1 直交化

$\mathcal{K}_k(A, \mathbf{r}_0)$ を作るベクトル列 $\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0\}$ が互いに直交していると都合がよいのだが、一般にはそうならないので、これらを正規直交化する。それには Gram-Schmidt 法などを使えばよいが、そのまま $\mathbf{r}_0$ に A をかけまくることになるので、直交化ベクトルの 1 つ目を $\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$ として、基にするベクトル列を $\{A\mathbf{v}_1, A\mathbf{v}_2, \dots, A\mathbf{v}_{k-1}\}$ と再帰的に表す。

Gram-Schmidt の直交化を適用すると、まず、 $\mathbf{r}_0$ から $\mathbf{v}_1$ を作る。

$$\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|} \quad (225)$$

次に、 $\mathbf{v}_1$ に直交するように、 $A\mathbf{v}_1$ から $\mathbf{v}_1$ 方向の成分を引き、正規化する。

$$\tilde{\mathbf{v}}_2 = A\mathbf{v}_1 - (A\mathbf{v}_1 \cdot \mathbf{v}_1)\mathbf{v}_1 \quad (226)$$

$$\mathbf{v}_2 = \frac{\tilde{\mathbf{v}}_2}{\|\tilde{\mathbf{v}}_2\|} \quad (227)$$

同様に

$$\tilde{\mathbf{v}}_3 = A\mathbf{v}_2 - (A\mathbf{v}_2 \cdot \mathbf{v}_1)\mathbf{v}_1 - (A\mathbf{v}_2 \cdot \mathbf{v}_2)\mathbf{v}_2 \quad (228)$$

$$\mathbf{v}_3 = \frac{\tilde{\mathbf{v}}_3}{\|\tilde{\mathbf{v}}_3\|} \quad (229)$$

これを繰り返す。 $\mathbf{v}_i$ は

$$\tilde{\mathbf{v}}_i = A\mathbf{v}_{i-1} - \sum_{j=1}^{i-1} (A\mathbf{v}_{i-1} \cdot \mathbf{v}_j)\mathbf{v}_j \quad (230)$$

$$\mathbf{v}_i = \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|} \quad (231)$$

以上から、次のように書くことができる。

$$\begin{aligned} A\mathbf{v}_1 &= (A\mathbf{v}_1 \cdot \mathbf{v}_1)\mathbf{v}_1 + \|\tilde{\mathbf{v}}_2\|\mathbf{v}_2 \\ A\mathbf{v}_2 &= (A\mathbf{v}_2 \cdot \mathbf{v}_1)\mathbf{v}_1 + (A\mathbf{v}_2 \cdot \mathbf{v}_2)\mathbf{v}_2 + \|\tilde{\mathbf{v}}_3\|\mathbf{v}_3 \\ &\dots \\ A\mathbf{v}_{i-1} &= \sum_{j=1}^{i-1} (A\mathbf{v}_{i-1} \cdot \mathbf{v}_j)\mathbf{v}_j + \|\tilde{\mathbf{v}}_i\|\mathbf{v}_i \end{aligned} \quad (232)$$

A の QR 分解の例と比べると、1 つ目の式がなくなっていて、いきなり 2 成分から始まるので、これによってできる行列は上三角行列にはならない (近い形にはなる)。 k を反復回数として、直交化された列ベクトルから作られる $n \times k$ の行列を $V_k = [\mathbf{v}_1 \cdots \mathbf{v}_k]$ と表すと、上の手順から次式が得られる。

$$AV_{k-1} = V_k \tilde{H}_{k-1} \quad (233)$$

ここで、 $\tilde{H}_{k-1}$ は次のような $k \times k-1$ の行列である.

$$\tilde{H}_{k-1} = \begin{bmatrix} A\mathbf{v}_1 \cdot \mathbf{v}_1 & A\mathbf{v}_2 \cdot \mathbf{v}_1 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_1 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_1 \\ \|\tilde{\mathbf{v}}_2\| & A\mathbf{v}_2 \cdot \mathbf{v}_2 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_2 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_2 \\ 0 & \|\tilde{\mathbf{v}}_3\| & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_3 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_3 \\ 0 & 0 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_4 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_4 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & \|\tilde{\mathbf{v}}_{k-1}\| & A\mathbf{v}_{k-1} \cdot \mathbf{v}_{k-1} \\ 0 & 0 & \cdots & 0 & \|\tilde{\mathbf{v}}_k\| \end{bmatrix} \quad (234)$$

$\tilde{H}_{k-1}$ から k 行目を取り除くと、 $k-1 \times k-1$ の上三角行列を 1 つ分だけ広くしたような Hessenberg 行列 H_{k-1} が得られる.

$$H_{k-1} = \begin{bmatrix} A\mathbf{v}_1 \cdot \mathbf{v}_1 & A\mathbf{v}_2 \cdot \mathbf{v}_1 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_1 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_1 \\ \|\tilde{\mathbf{v}}_2\| & A\mathbf{v}_2 \cdot \mathbf{v}_2 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_2 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_2 \\ 0 & \|\tilde{\mathbf{v}}_3\| & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_3 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_3 \\ 0 & 0 & \cdots & A\mathbf{v}_{k-2} \cdot \mathbf{v}_4 & A\mathbf{v}_{k-1} \cdot \mathbf{v}_4 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & \|\tilde{\mathbf{v}}_{k-1}\| & A\mathbf{v}_{k-1} \cdot \mathbf{v}_{k-1} \end{bmatrix} \quad (235)$$

これを用いると

$$\begin{aligned} AV_{k-1} &= V_k \tilde{H}_{k-1} \\ &= V_{k-1} H_{k-1} + \begin{bmatrix} 0 & 0 & \cdots & \|\tilde{\mathbf{v}}_k\| \mathbf{v}_k \end{bmatrix} \\ &= V_{k-1} H_{k-1} + \|\tilde{\mathbf{v}}_k\| \mathbf{v}_k \left(\mathbf{e}_{k-1}^{(k-1)} \right)^T \end{aligned} \quad (236)$$

ここで、 $\mathbf{e}_{k-1}^{(k-1)}$ は $k-1$ 番目が 1 でそれ以外は 0 の $k-1$ 次元ベクトルである (右肩の $(k-1)$ が次元). $\mathbf{v}_k$ は V_{k-1} を構成する列ベクトルすべてと直交するはずなので、 $V_{k-1}^T \mathbf{v}_k = \mathbf{0}$ である. したがって、次式が得られる.

$$H_{k-1} = V_{k-1}^T AV_{k-1} \quad (237)$$

これは、部分空間への A の射影になっている.

以上の手順を修正 Gram-Schmidt 法で表したものをアルゴリズム 30 に示す. これを Arnoldi 過程 (Arnoldi process) という. この過程は、部分空間をどんどん広げて行くようなイメージである.

係数行列 A が対称の場合、 H_n は Hessenberg 行列ではなく対称三重対角行列になる. というのも、たとえば H_n の $(1, 3)$ の要素を考えると $A\mathbf{v}_3 \cdot \mathbf{v}_1$ だが、 A が対称だと $A\mathbf{v}_3 \cdot \mathbf{v}_1 = A\mathbf{v}_1 \cdot \mathbf{v}_3$ が成り立つ. $A\mathbf{v}_1$ には $\mathbf{v}_1$ と $\mathbf{v}_2$ の成分しかないのだから、 $A\mathbf{v}_1 \cdot \mathbf{v}_3 = 0$. したがって、 $A\mathbf{v}_3 \cdot \mathbf{v}_1 = 0$ となる. また、 $(1, 2)$ の要素を考えると、同様に $A\mathbf{v}_2 \cdot \mathbf{v}_1 = A\mathbf{v}_1 \cdot \mathbf{v}_2 = \|\tilde{\mathbf{v}}_2\|$ となる.

対称三重対角行列を仮定できるなら、Arnoldi 過程を少し簡単にできる. これを Lanczos 過程 (Lanczos process) という. 三重対角行列であれば、 $A\mathbf{v}_{i-1}$ は 3 項で計算できる.

$$A\mathbf{v}_{i-1} = (A\mathbf{v}_{i-1} \cdot \mathbf{v}_{i-2})\mathbf{v}_{i-2} + (A\mathbf{v}_{i-1} \cdot \mathbf{v}_{i-1})\mathbf{v}_{i-1} + \|\tilde{\mathbf{v}}_i\|\mathbf{v}_i \quad (238)$$

対称性から $A\mathbf{v}_{i-1} \cdot \mathbf{v}_{i-2} = A\mathbf{v}_{i-2} \cdot \mathbf{v}_{i-1}$ だが

$$A\mathbf{v}_{i-2} = (A\mathbf{v}_{i-2} \cdot \mathbf{v}_{i-3})\mathbf{v}_{i-3} + (A\mathbf{v}_{i-2} \cdot \mathbf{v}_{i-2})\mathbf{v}_{i-2} + \|\tilde{\mathbf{v}}_{i-1}\|\mathbf{v}_{i-1} \quad (239)$$

Algorithm 30 Arnoldi 過程

```
 $v_1 = \frac{r_0}{\|r_0\|}$ 
for  $i = 1, 2, \dots, n$  do
   $w = Av_i$ 
  for  $j = 1, 2, \dots, i$  do
     $h_{ji} = w \cdot v_j$ 
     $w = w - h_{ji}v_j$ 
  end for
  if  $i < n$  then
     $h_{i+1,i} = \|w\|$ 
     $v_{i+1} = \frac{w}{h_{i+1,i}}$ 
  end if
end for
```

なので, $Av_{i-1} \cdot v_{i-2} = \|\tilde{v}_{i-1}\|$. したがって

$$Av_{i-1} = \|\tilde{v}_{i-1}\|v_{i-2} + (Av_{i-1} \cdot v_{i-1})v_{i-1} + \|\tilde{v}_i\|v_i \quad (240)$$

これを次のように表すと, 係数を2つずつ計算していけばよいことになる.

$$\begin{aligned} Av_{i-1} &= \beta_i v_{i-2} + \alpha_i v_{i-1} + \beta_{i+1} v_i \\ \alpha_i &= Av_{i-1} \cdot v_{i-1} \\ \beta_{i+1} &= \|\tilde{v}_i\| \end{aligned} \quad (241)$$

v_i の計算は

$$\tilde{v}_i = Av_{i-1} - \alpha_i v_{i-1} - \beta_i v_{i-2} \quad (242)$$

$$v_i = \frac{\tilde{v}_i}{\beta_{i+1}} \quad (243)$$

となる.

Lanczos 過程の手順をアルゴリズム 31 に示す.

3.6.2 共役勾配法

Krylov 部分空間法の枠組みで共役勾配法を導出する. Ritz-Galerkin 方式を想定し, 係数行列 A は対称行列として, 部分空間の基底の直交化には Lanczos 過程を用いる. 反復 k 回目の近似解を x_k , 近似解の初期値を x_0 とすると, この差が $\mathcal{K}_k(A, r_0)$ に属するようにする.

$$x_k - x_0 \in \mathcal{K}_k(A, r_0) \quad (244)$$

部分空間の正規直交基底を $\{v_1, \dots, v_k\}$, $V_k = [v_1 \cdots v_k]$ とすると, $x_k - x_0$ が $\{v_1, \dots, v_k\}$ の線形結合で表されればよいので, 係数ベクトルを y_k とすると

$$x_k - x_0 = V_k y_k \quad (245)$$

Algorithm 31 Lanczos 過程

```
 $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$   
 $\mathbf{v}_0 = \mathbf{0}$   
 $\beta = 0$   
for  $i = 1, 2, \dots, n$  do  
   $\mathbf{w} = A\mathbf{v}_i$   
   $\alpha = \mathbf{w} \cdot \mathbf{v}_i$   
   $\mathbf{w} = \mathbf{w} - \alpha\mathbf{v}_i - \beta\mathbf{v}_0$   
   $h_{ii} = \alpha$   
  if  $i \geq 2$  then  
     $h_{i-1,i} = \beta$   
  end if  
  if  $i < n$  then  
     $\beta = \|\mathbf{w}\|$   
     $\mathbf{v}_{i+1} = \frac{\mathbf{w}}{\beta}$   
     $h_{i+1,i} = \beta$   
     $\mathbf{v}_0 = \mathbf{v}_i$   
  end if  
end for
```

残差を $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ とすると

$$\begin{aligned}\mathbf{r}_k &= \mathbf{b} - A\mathbf{x}_k \\ &= \mathbf{b} - A\mathbf{x}_0 + A\mathbf{x}_0 - A\mathbf{x}_k \\ &= \mathbf{r}_0 - A(\mathbf{x}_k - \mathbf{x}_0)\end{aligned}\tag{246}$$

これより

$$\mathbf{r}_k = \mathbf{r}_0 - AV_k\mathbf{y}_k\tag{247}$$

Ritz-Galerkin 方式では、残差ベクトルと部分空間の直交条件

$$\mathbf{r}_k \perp \mathcal{K}_k(A, \mathbf{r}_0)\tag{248}$$

を考えるが、これは $V_k^T \mathbf{r}_k = \mathbf{0}$ を意味する。

$$V_k^T \mathbf{r}_k = V_k^T \mathbf{r}_0 - V_k^T AV_k \mathbf{y}_k = \mathbf{0}\tag{249}$$

直交化の議論から、 $H_k = V_k^T AV_k$ より

$$V_k^T \mathbf{r}_0 - H_k \mathbf{y}_k = \mathbf{0}\tag{250}$$

したがって

$$\mathbf{y}_k = H_k^{-1} V_k^T \mathbf{r}_0\tag{251}$$

ここで, $\mathbf{r}_0 = \|\mathbf{r}_0\|\mathbf{v}_1 = \|\mathbf{r}_0\|V_k\mathbf{e}_1^{(k)}$ より $V_k^T\mathbf{r}_0 = \|\mathbf{r}_0\|\mathbf{e}_1^{(k)}$ なので, 次のようにも書ける.

$$\mathbf{y}_k = \|\mathbf{r}_0\|H_k^{-1}\mathbf{e}_1^{(k)} \quad (252)$$

改めて残差について考えると

$$AV_k = V_kH_k + \|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \quad (253)$$

より

$$\begin{aligned} \mathbf{r}_k &= \mathbf{r}_0 - AV_k\mathbf{y}_k \\ &= \mathbf{r}_0 - \left\{ V_kH_k + \|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \right\} \mathbf{y}_k \\ &= \mathbf{r}_0 - V_kH_k\mathbf{y}_k - \|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \mathbf{y}_k \\ &= \mathbf{r}_0 - V_kH_k \left(\|\mathbf{r}_0\|H_k^{-1}\mathbf{e}_1^{(k)} \right) - \|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \mathbf{y}_k \\ &= \mathbf{r}_0 - \mathbf{r}_0 - \|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \mathbf{y}_k \\ &= -\|\tilde{\mathbf{v}}_{k+1}\|\mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)}\right)^T \mathbf{y}_k \end{aligned} \quad (254)$$

したがって

$$\mathbf{r}_k = -\|\tilde{\mathbf{v}}_{k+1}\| \left(\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k \right) \mathbf{v}_{k+1} \quad (255)$$

上の式は, $\mathbf{r}_k$ と $\mathbf{v}_{k+1}$ が同じ方向を向いていることを意味しているのもので, ただちに残差ベクトル同士の直交性 $\mathbf{r}_i \cdot \mathbf{r}_j = 0 (i \neq j)$ を示すことができる. また, $\mathbf{r}_i \cdot \mathbf{r}_i$ を考えると

$$\mathbf{r}_i \cdot \mathbf{r}_i = \|\tilde{\mathbf{v}}_{i+1}\|^2 \left(\mathbf{e}_i^{(i)} \cdot \mathbf{y}_i \right)^2 \mathbf{v}_{i+1} \cdot \mathbf{v}_{i+1} = \|\tilde{\mathbf{v}}_{i+1}\|^2 \left(\mathbf{e}_i^{(i)} \cdot \mathbf{y}_i \right)^2 \quad (256)$$

$\mathbf{r}_i \cdot \mathbf{r}_i = \|\mathbf{r}_i\|^2$ とすると

$$\|\mathbf{r}_k\| = \pm \|\tilde{\mathbf{v}}_{k+1}\| \left(\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k \right) \quad (257)$$

式 (255) は

$$\mathbf{r}_k = \pm \|\mathbf{r}_k\| \mathbf{v}_{k+1} \quad (258)$$

となるので

$$\mathbf{v}_{k+1} = \pm \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|} \quad (259)$$

つまり, 部分空間の基底ベクトルは残差を正規化したものである. 正負の符号はどちらでもよいが (実際, Gram-Schmidt の直交化では $\mathbf{v}_1$ 以外はどちらにもなりうる), ここではわかりやすく正のほうをとる.

ここでは Lanczos 過程を想定するので, H_k は三重対角行列である. $\mathbf{v}_{k+1}$ の漸化式は, Lanczos 過程の議論から

$$\tilde{\mathbf{v}}_{k+1} = A\mathbf{v}_k - (A\mathbf{v}_k \cdot \mathbf{v}_{k-1})\mathbf{v}_{k-1} - (A\mathbf{v}_k \cdot \mathbf{v}_k)\mathbf{v}_k \quad (260)$$

$$\mathbf{v}_{k+1} = \frac{\tilde{\mathbf{v}}_{k+1}}{\|\tilde{\mathbf{v}}_{k+1}\|} \quad (261)$$

式 (259) を代入すると

$$\begin{aligned}\tilde{\mathbf{v}}_{k+1} &= \frac{1}{\|\mathbf{r}_{k-1}\|} A\mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\|\mathbf{r}_{k-1}\| \|\mathbf{r}_{k-2}\|^2} \mathbf{r}_{k-2} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\|\mathbf{r}_{k-1}\|^3} \mathbf{r}_{k-1} \\ &= \frac{1}{\|\mathbf{r}_{k-1}\|} \left(A\mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} \mathbf{r}_{k-2} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \mathbf{r}_{k-1} \right)\end{aligned}\quad (262)$$

式 (255) に上式を代入すると

$$\mathbf{r}_k = \frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left(A\mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} \mathbf{r}_{k-2} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \mathbf{r}_{k-1} \right) \quad (263)$$

内積を取ると

$$\mathbf{r}_k \cdot \mathbf{r}_k = \frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} A\mathbf{r}_{k-1} \cdot \mathbf{r}_k \quad (264)$$

これより

$$\frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} = \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{A\mathbf{r}_{k-1} \cdot \mathbf{r}_k} = \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{A\mathbf{r}_k \cdot \mathbf{r}_{k-1}} \quad (265)$$

式 (246) を式 (263) の一部 ($A\mathbf{r}_{k-1}$ 以外) に代入すると

$$\begin{aligned}\mathbf{r}_k &= -\frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left(\frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \right) \mathbf{r}_0 \\ &\quad + \frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} A \left\{ \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-2} - \mathbf{x}_0) + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} (\mathbf{x}_{k-1} - \mathbf{x}_0) + \mathbf{r}_{k-1} \right\}\end{aligned}\quad (266)$$

式 (246) と比較すると

$$-\frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left(\frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \right) = 1 \quad (267)$$

$$\mathbf{x}_k - \mathbf{x}_0 = -\frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left\{ \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-2} - \mathbf{x}_0) + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} (\mathbf{x}_{k-1} - \mathbf{x}_0) + \mathbf{r}_{k-1} \right\} \quad (268)$$

ここで、第 2 式の右辺の括弧の中の 2 項について

$$\begin{aligned}&\frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-2} - \mathbf{x}_0) + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} (\mathbf{x}_{k-1} - \mathbf{x}_0) = \\ &\left(\frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \right) (\mathbf{x}_{k-1} - \mathbf{x}_0) - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-1} - \mathbf{x}_{k-2})\end{aligned}\quad (269)$$

なので

$$\begin{aligned}\mathbf{x}_k - \mathbf{x}_0 &= \mathbf{x}_{k-1} - \mathbf{x}_0 - \frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left\{ \mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-1} - \mathbf{x}_{k-2}) \right\} \\ \longrightarrow \mathbf{x}_k - \mathbf{x}_{k-1} &= -\frac{\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k}{\|\mathbf{r}_{k-1}\|} \left\{ \mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} (\mathbf{x}_{k-1} - \mathbf{x}_{k-2}) \right\}\end{aligned}\quad (270)$$

ここで、 $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ と想定する。上式から $\mathbf{p}_k$ の漸化式を作りたいのだが、式が簡単になるように次のように決める。

$$\alpha_k = -\frac{\mathbf{e}_{k+1}^{(k+1)} \cdot \mathbf{y}_{k+1}}{\|\mathbf{r}_k\|} = -\frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{A\mathbf{r}_{k+1} \cdot \mathbf{r}_k} \quad (271)$$

これを用いると

$$\begin{aligned}
\alpha_{k-1}\mathbf{p}_{k-1} &= \alpha_{k-1} \left(\mathbf{r}_{k-1} - \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} \alpha_{k-2}\mathbf{p}_{k-2} \right) \\
\longrightarrow \mathbf{p}_{k-1} &= \mathbf{r}_{k-1} + \frac{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} \frac{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{A\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-2}} \mathbf{p}_{k-2} \\
\longrightarrow \mathbf{p}_{k-1} &= \mathbf{r}_{k-1} + \frac{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-2} \cdot \mathbf{r}_{k-2}} \mathbf{p}_{k-2}
\end{aligned} \tag{272}$$

したがって、次式が得られる.

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{\mathbf{r}_k \cdot \mathbf{r}_k} \mathbf{p}_k \tag{273}$$

$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_{k+1}\mathbf{p}_k$ とすると

$$\beta_k = \frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{\mathbf{r}_k \cdot \mathbf{r}_k} \tag{274}$$

つまり、 $\mathbf{p}_k$ の漸化式は Lanczos 過程の漸化式から出てくる.

ところで、 $\mathbf{p}_k$ は

$$\mathbf{p}_k = \frac{1}{\alpha_k} (\mathbf{x}_{k+1} - \mathbf{x}_k) = \frac{1}{\alpha_k} \{ (\mathbf{x}_{k+1} - \mathbf{x}_0) - (\mathbf{x}_k - \mathbf{x}_0) \} \tag{275}$$

なので、 $\mathbf{p}_k \in \mathcal{K}_{k+1}(A, \mathbf{r}_0)$ である. したがって、直交条件から $\mathbf{r}_{k+1} \cdot \mathbf{p}_k = 0$ である. $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$ なので

$$\begin{aligned}
\mathbf{r}_{k+1} \cdot \mathbf{p}_k &= \mathbf{r}_k \cdot \mathbf{p}_k - \alpha_k A\mathbf{p}_k \cdot \mathbf{p}_k = 0 \\
\longrightarrow \alpha_k &= \frac{\mathbf{p}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k}
\end{aligned} \tag{276}$$

あとはもとの共役勾配法の導出と同じだが

$$\begin{aligned}
\mathbf{p}_k \cdot \mathbf{r}_k &= \left(\mathbf{r}_k + \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \mathbf{p}_{k-1} \right) \cdot \mathbf{r}_k \\
&= \mathbf{r}_k \cdot \mathbf{r}_k + \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \mathbf{p}_{k-1} \cdot \mathbf{r}_k
\end{aligned} \tag{277}$$

$\mathbf{p}_{k-1} \cdot \mathbf{r}_k = 0$ なので $\mathbf{p}_k \cdot \mathbf{r}_k = \mathbf{r}_k \cdot \mathbf{r}_k$. したがって、次式が得られる.

$$\alpha_k = \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot A\mathbf{p}_k} \tag{278}$$

以上で共役勾配法は導かれたことになるが、別の導き方もある. $V_k^T \mathbf{r}_k = 0$ から $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ ($i \neq j$) であり、 $\mathbf{v}_k$ は $\mathbf{r}_{k-1}$ を正規化したものであることがわかったので、 $V_k = [\mathbf{v}_1 \cdots \mathbf{v}_k]$ の代わりに $R_k = [\mathbf{r}_0 \cdots \mathbf{r}_{k-1}]$ を考えて $\mathbf{x}_k - \mathbf{x}_0 = R_k \mathbf{z}_k$ としてもよい. そうすると

$$\begin{aligned}
R_k^T \mathbf{r}_k &= R_k^T (\mathbf{r}_0 - AR_k \mathbf{z}_k) \\
&= R_k^T \mathbf{r}_0 - R_k^T \left\{ R_k H_k + \|\tilde{\mathbf{v}}_{k+1}\| \mathbf{v}_{k+1} \left(\mathbf{e}_k^{(k)} \right)^T \right\} \mathbf{z}_k \\
&= R_k^T \mathbf{r}_0 - R_k^T R_k H_k \mathbf{z}_k = 0
\end{aligned} \tag{279}$$

より

$$R_k^T R_k H_k \mathbf{z}_k = R_k^T \mathbf{r}_0 \tag{280}$$

ここで、 $R_k^T R_k$ は残差ベクトルノルムの 2 乗を対角要素に持った対角行列であり、その逆行列は残差ベクトルノルムの 2 乗の逆数を対角要素に持った対角行列である。 $R_k^T \mathbf{r}_0$ は第 1 成分に $\|\mathbf{r}_0\|^2$ を持ったベクトルである。したがって

$$\begin{aligned} R_k^T R_k H_k \mathbf{z}_k &= \|\mathbf{r}_0\|^2 \mathbf{e}_1^{(k)} \\ \longrightarrow H_k \mathbf{z}_k &= \mathbf{e}_1^{(k)} \end{aligned} \quad (281)$$

これより $\mathbf{x}_k - \mathbf{x}_0$ は

$$\mathbf{x}_k - \mathbf{x}_0 = R_k \mathbf{z}_k = R_k H_k^{-1} \mathbf{e}_1^{(k)} \quad (282)$$

と表される。これを $\mathbf{x}_k - \mathbf{x}_0 = P_k \mathbf{q}_k$ と表す。ここで、 $P_k = [\mathbf{p}_0 \cdots \mathbf{p}_{k-1}]$ 、 $\mathbf{q}_k = [q_0 \cdots q_{k-1}]^T$ とする。 H_k を LU 分解して $H_k = L_k U_k$ として、 $P_k = R_k U_k^{-1}$ 、 $\mathbf{q}_k = L_k^{-1} \mathbf{e}_1^{(k)}$ とする。 $R_k = P_k U_k$ だが、 A が対称行列の場合、 H_k は三重対角行列であり

$$R_k = [\mathbf{p}_0 \cdots \mathbf{p}_{k-1}] \begin{bmatrix} 1 & u_0 & & & \\ & 1 & u_1 & & \\ & & \ddots & \ddots & \\ & & & & u_{k-2} \\ & & & & 1 \end{bmatrix} \quad (283)$$

となるので

$$\mathbf{r}_{k-1} = u_{k-2} \mathbf{p}_{k-2} + \mathbf{p}_{k-1} \quad (284)$$

したがって

$$\mathbf{p}_{k-1} = \mathbf{r}_{k-1} - u_{k-2} \mathbf{p}_{k-2} \quad (285)$$

$\mathbf{x}_k - \mathbf{x}_0$ については

$$\mathbf{x}_k - \mathbf{x}_0 = P_k \mathbf{q}_k = [\mathbf{p}_0 \cdots \mathbf{p}_{k-1}] \begin{bmatrix} q_0 \\ \vdots \\ q_{k-1} \end{bmatrix} \quad (286)$$

から

$$\mathbf{x}_k - \mathbf{x}_0 = P_{k-1} \mathbf{q}_{k-1} + q_{k-1} \mathbf{p}_{k-1} \quad (287)$$

なので

$$\mathbf{x}_k = \mathbf{x}_{k-1} + q_{k-1} \mathbf{p}_{k-1} \quad (288)$$

以上を $\alpha_{k-1} = q_{k-1}$ 、 $\beta_{k-2} = -u_{k-2}$ として整理すると、共役勾配法の基本漸化式が得られる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (289)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A \mathbf{p}_k \quad (290)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \quad (291)$$

いずれにしても、 $\mathbf{p}_k$ の漸化式が短くて済むのは A の対称性のゆえであることがわかる。

α_k は上述のとおりとして、 β_k はどうするか。 $\mathbf{p}_k$ の漸化式から

$$A \mathbf{p}_k = A \mathbf{r}_k + \beta_{k-1} A \mathbf{p}_{k-1} \quad (292)$$

$\mathbf{r}_{k+1}$ の漸化式から

$$A\mathbf{p}_{k-1} = \frac{1}{\alpha_{k-1}}(\mathbf{r}_{k-1} - \mathbf{r}_k) \quad (293)$$

以上の準備をした上で

$$\begin{aligned} \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A\mathbf{p}_k \\ &= \mathbf{r}_k - \alpha_k (A\mathbf{r}_k + \beta_{k-1} A\mathbf{p}_{k-1}) \\ &= \mathbf{r}_k - \alpha_k A\mathbf{r}_k - \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} (\mathbf{r}_{k-1} - \mathbf{r}_k) \end{aligned} \quad (294)$$

$\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}$ を作ると

$$\begin{aligned} \mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1} &= -\alpha_k A\mathbf{r}_k \cdot \mathbf{r}_{k+1} \\ \longrightarrow \alpha_k &= -\frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{A\mathbf{r}_k \cdot \mathbf{r}_{k+1}} = -\frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{A\mathbf{r}_{k+1} \cdot \mathbf{r}_k} \end{aligned} \quad (295)$$

$\mathbf{r}_{k+1} \cdot \mathbf{r}_{k-1}$ を作ると

$$\begin{aligned} \mathbf{r}_{k+1} \cdot \mathbf{r}_{k-1} &= -\alpha_k A\mathbf{r}_k \cdot \mathbf{r}_{k-1} - \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} \mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1} = 0 \\ \longrightarrow \frac{\beta_{k-1}}{\alpha_{k-1}} \mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1} &= -A\mathbf{r}_k \cdot \mathbf{r}_{k-1} \\ \longrightarrow \beta_{k-1} &= -\alpha_{k-1} \frac{A\mathbf{r}_k \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} = \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{A\mathbf{r}_k \cdot \mathbf{r}_{k-1}} \frac{A\mathbf{r}_k \cdot \mathbf{r}_{k-1}}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \\ \longrightarrow \beta_{k-1} &= \frac{\mathbf{r}_k \cdot \mathbf{r}_k}{\mathbf{r}_{k-1} \cdot \mathbf{r}_{k-1}} \end{aligned} \quad (296)$$

したがって

$$\beta_k = \frac{\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}}{\mathbf{r}_k \cdot \mathbf{r}_k} \quad (297)$$

探索方向ベクトル $\mathbf{p}$ の A 共役性 ($\mathbf{p}_i \cdot A\mathbf{p}_j = 0$, $i \neq j$) は陽には仮定されていないが, $\mathbf{p}_k$ の漸化式を短く取った時点でそれは導入されていて, それは係数行列 A に対称性を想定したことにより Lanczos 過程から自然に出てきたのであった. また, A の正定値性を仮定していないため, A が非正定値でも解ける可能性があるが, Hessenberg 行列 H_k が LU 分解可能であることは想定していることになるので, 仮定はその中に含まれているといえる. 結局, 漸化式さえ得られれば共役勾配法は導けるが, 漸化式を決定した時点でかなりの仮定を導入していることになる.

3.6.3 共役残差法

Krylov 部分空間法の枠組みで共役残差法を導出したい. 最小残差方式を考える. 残差ベクトルのノルム $\|\mathbf{r}\|$ の最小化を考えることになるが, これは $\mathbf{r} \cdot \mathbf{r}$ の最小化を考えるのと同じなので, 勾配法の最小残差反復で考えた $F(\mathbf{r}) = \mathbf{r} \cdot \mathbf{r}/2$ の最小化と同じである. その場合, 残差ベクトル $\mathbf{r}$ が $\mathbf{r} + \delta\mathbf{r}$ に変化したとすると, $\delta\mathbf{r} \cdot \mathbf{r} = 0$ が成り立つ. $\delta\mathbf{r} = -A\delta\mathbf{x}$ とすると, $(A\delta\mathbf{x}) \cdot \mathbf{r} = 0$ となる. これは $A^T A\mathbf{x} = A^T \mathbf{b}$ を考えることと同じである. Krylov 部分空間法においては, $\delta\mathbf{r}_k = -AV_k \delta\mathbf{y}_k$ より, $(AV_k)^T \mathbf{r}_k = \mathbf{0}$ を考えることになる.

$(AV_k)^T \mathbf{r}_k = \mathbf{0}$ を考えると

$$(AV_k)^T \mathbf{r}_k = (AV_k)^T (\mathbf{r}_0 - AV_k \mathbf{y}_k) \quad (298)$$

直交化の議論から, $A_k V_k = V_{k+1} \tilde{H}_k$ より

$$(AV_k)^T (\mathbf{r}_0 - AV_k \mathbf{y}_k) = \tilde{H}_k^T V_{k+1}^T \mathbf{r}_0 - \tilde{H}_k^T \tilde{H}_k \mathbf{y}_k = 0 \quad (299)$$

したがって

$$\tilde{H}_k^T \tilde{H}_k \mathbf{y}_k = \tilde{H}_k^T V_{k+1}^T \mathbf{r}_0 \quad (300)$$

$\mathbf{r}_i \cdot A\mathbf{r}_j$ を考えると

$$\begin{aligned} \mathbf{r}_i \cdot A\mathbf{r}_j &= \mathbf{r}_i^T A(\mathbf{r}_0 - AV_j \mathbf{y}_j) \\ &= \mathbf{r}_i^T A\mathbf{r}_0 - \mathbf{r}_i^T AV_{j+1} \tilde{H}_j \mathbf{y}_j \end{aligned} \quad (301)$$

$\mathbf{r}_0 = \|\mathbf{r}_0\| \mathbf{v}_1$ なので、右辺第 1 項は 0 である。 $i > j$ のとき $(AV_{j+1})^T \mathbf{r}_i = \mathbf{0}$ なので、右辺第 2 項も 0 である。したがって、 $i > j$ のとき $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ である。

$V_k = [\mathbf{v}_1 \cdots \mathbf{v}_k]$ の代わりに $R_k = [\mathbf{r}_0 \cdots \mathbf{r}_{k-1}]$ を考える。 $\mathbf{r}_i \cdot A\mathbf{r}_j = 0$ ($i > j$) なので $(AR_k)^T \mathbf{r}_k = 0$ が成り立つ。 $\mathbf{x}_k - \mathbf{x}_0 = R_k \mathbf{z}_k$ とすると、次式が得られる。

$$\tilde{H}_k^T R_{k+1}^T R_{k+1} \tilde{H}_k \mathbf{z}_k = \tilde{H}_k^T R_{k+1}^T \mathbf{r}_0 \quad (302)$$

これを $\hat{H}_k \mathbf{z}_k = \hat{\mathbf{b}}$ と表すことにすると

$$\mathbf{x} - \mathbf{x}_0 = R_k \mathbf{z}_k = R_k \hat{H}_k^{-1} \hat{\mathbf{b}} \quad (303)$$

となる。これを $\mathbf{x} - \mathbf{x}_0 = P_k \mathbf{q}_k$ と表す。あとは共役勾配法の場合と同様にして (ただし、直交条件として $\mathbf{r}_k \cdot A\mathbf{p}_{k-1} = 0$ が成り立つ)、 A を対称行列として Lanczos 過程を想定して進めたいところだが、あいにく $\hat{H}_k$ は三重対角行列ではない。 $\tilde{H}_k$ を LU 分解して (この場合 U は正方行列)、 $P_k = R_k U_k^{-1}$ とし、残りは $\mathbf{q}_k$ に突っ込むことにすれば (たぶん) 行けそうである。以下が得られる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (304)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k \quad (305)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \quad (306)$$

$$\alpha_k = \frac{\mathbf{r}_k \cdot A\mathbf{p}_k}{A\mathbf{p}_k \cdot A\mathbf{p}_k} \quad (307)$$

$$\beta_k = \frac{\mathbf{r}_{k+1} \cdot A\mathbf{r}_{k+1}}{\mathbf{r}_k \cdot A\mathbf{r}_k} \quad (308)$$

つまり、 $\mathbf{p}_k$ の漸化式に $\beta_{ik} \mathbf{p}_i$ の項が 1 つしかない共役残差法は、 (はっきりとは要求されていないが) 対称行列が想定されていることになる。

では、 A を対称行列に限らず、Arnoldi 過程を考えると一般化共役残差法が出てくるのか。 $\mathbf{x}_k - \mathbf{x}_0 = P_k \mathbf{q}_k$ として、 $P_k = R_k U_k^{-1}$ とすると、 $R_k = P_k U_k$ で、この場合は U_k には何の仮定も置かないので

$$R_k = [\mathbf{p}_0 \cdots \mathbf{p}_{k-1}] \begin{bmatrix} 1 & u_{12} & \cdots & & u_{1k} \\ & 1 & u_{23} & \cdots & u_{2k} \\ & & \ddots & \ddots & \\ & & & & u_{k-1,k} \\ & & & & 1 \end{bmatrix} \quad (309)$$

これより

$$\mathbf{r}_{k-1} = u_{1k} \mathbf{p}_0 + \cdots u_{k-1,k} \mathbf{p}_{k-2} + \mathbf{p}_{k-1} \quad (310)$$

したがって

$$\mathbf{p}_{k-1} = \mathbf{r}_{k-1} - \sum_{i=0}^{k-2} u_{i+1,k} \mathbf{p}_i \quad (311)$$

$\beta_{i-1,k-2} = -u_{ik}$ とすると

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=0}^k \beta_{ik} \mathbf{p}_i \quad (312)$$

直交条件から $\mathbf{r}_{i+1} \cdot \mathbf{A}\mathbf{p}_j = 0$ ($i > j$) なので

$$\mathbf{r}_{i+1} \cdot \mathbf{A}\mathbf{p}_j = \mathbf{r}_i \cdot \mathbf{A}\mathbf{p}_j - \alpha_i \mathbf{A}\mathbf{p}_i \cdot \mathbf{A}\mathbf{p}_j = 0 \quad (313)$$

したがって, $\mathbf{A}\mathbf{p}_i \cdot \mathbf{A}\mathbf{p}_j = 0$ ($i > j$) である. $\mathbf{A}\mathbf{p}_i \cdot \mathbf{A}\mathbf{p}_j = \mathbf{A}\mathbf{p}_j \cdot \mathbf{A}\mathbf{p}_i$ なので, 結局 $\mathbf{A}\mathbf{p}_i \cdot \mathbf{A}\mathbf{p}_j = 0$ ($i \neq j$) である. $\mathbf{A}\mathbf{p}_{k+1} \cdot \mathbf{A}\mathbf{p}_i = 0$ より β_{ik} が求まる.

$$\beta_{ik} = -\frac{\mathbf{A}\mathbf{r}_{k+1} \cdot \mathbf{A}\mathbf{p}_i}{\mathbf{A}\mathbf{p}_i \cdot \mathbf{A}\mathbf{p}_i} \quad (314)$$

つまり, 共役残差法において $\mathbf{p}_k$ の漸化式の $\beta_{ik}\mathbf{p}_i$ の項数を十分に取れば, 非対称の係数行列をもつ方程式も解けるはずということになる.

3.6.4 一般化最小残差法

Krylov 部分空間法の枠組みで共役勾配法や共役残差法を導出できるということは, より直接的な方法で似たような手法を導けるということである. 例えば, Ritz-Galerkin 方式で Arnoldi 過程を使えば, 共役勾配法の非対称行列版のようなものが得られる.

共役勾配法の議論から

$$\mathbf{x}_k - \mathbf{x}_0 = V_k \mathbf{y}_k \quad (315)$$

$$\mathbf{y}_k = \|\mathbf{r}_0\| H_k^{-1} \mathbf{e}_1^{(k)} \quad (316)$$

だったので, 単純に考えれば, Arnoldi 過程で H_k を構成して上式を解けば $\mathbf{x}_k$ が得られることになる. この方法を full orthogonalization method (FOM) という. H_k は Hessenberg 行列なので, $\mathbf{y}_k$ は Givens 回転による QR 分解と後退代入で計算できる. 手順をアルゴリズム 32 に示す. ここで $n_s \leq n$ は部分空間の次元である.

実用上は残差を見て反復計算を早めに切り上げることになるが, 残差ベクトルのノルムは $\|\mathbf{w}\| |y_i|$ で計算できる. というのも

$$\mathbf{r}_k = -\|\tilde{\mathbf{v}}_{k+1}\| \left(\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k \right) \mathbf{v}_{k+1} \quad (317)$$

だったので

$$\|\mathbf{r}_k\| = \|\tilde{\mathbf{v}}_{k+1}\| \left| \mathbf{e}_k^{(k)} \cdot \mathbf{y}_k \right| \quad (318)$$

であり, $\|\tilde{\mathbf{v}}_{k+1}\|$ は $\tilde{H}_k$ の端っこの成分 $h_{k+1,k}$ なので, アルゴリズム上は $h_{i+1,i} = \|\mathbf{w}\|$ に当たる. $\mathbf{e}_k^{(k)} \cdot \mathbf{y}_k = y_k$ より, $\|\mathbf{r}_i\| = \|\mathbf{w}\| |y_i|$ となる. 毎度 H_k を QR 分解するのは無駄なので, 手順に QR 分解自体も組み込んで, H_k の拡大された部分だけ処理を行う. $\mathbf{y}$ の方程式の右辺も同時に処理する. 方程式を $R\mathbf{y} = \hat{\mathbf{b}}$ とすると, y_i は後退代入の初っ端に計算されるものなので, $\mathbf{y}$ 全体を計算する必要はなく, $y_i = \hat{b}_i / r_{ii}$ で求まる. 結局, 残差ベクトルノルムは $\|\mathbf{r}_i\| = \|\mathbf{w}\| |\hat{b}_i / r_{ii}|$ で計算できる.

Algorithm 32 FOM

```
 $v_1 = \frac{r_0}{\|r_0\|}$ 
for  $i = 1, 2, \dots, n_s$  do
   $w = Av_i$ 
  for  $j = 1, 2, \dots, i$  do
     $h_{ji} = w \cdot v_j$ 
     $w = w - h_{ji}v_j$ 
  end for
  if  $i < n_s$  then
     $h_{i+1,i} = \|w\|$ 
     $v_{i+1} = \frac{w}{h_{i+1,i}}$ 
  end if
end for
 $Q, R \leftarrow H_{n_s}$ 
 $y \leftarrow Ry = \|r_0\|Q^T e_1^{(n_s)}$ 
 $x = x_0 + V_{n_s}y$ 
```

残差判定版の手順をアルゴリズム 33 に示す。ごちゃごちゃして見えるが、Arnoldi 過程のあとに Givens 回転の処理があるだけである。ここでは H を直接 R に変換している。Arnoldi 過程の $h_{i+1,i}$, v_{i+1} の部分の i の判定が外してあるが、判定を避けるために H と V のサイズを 1 つ余分に大きくしている想定である。

FOM の最小残差方式版に当たるものが一般化最小残差法 (generalized minimal residual method, GMRES 法) である。残差ベクトルノルム $\|r_k\|$ の最小化を考える。これは $r_k \cdot r_k$ の最小化を考えるということなので、 $\delta r_k \cdot r_k = 0$ を考える。残差ベクトル r_k は

$$r_k = r_0 - AV_k y_k \quad (319)$$

だったので、 $\delta r_k = -AV_k \delta y_k$ として、 $\delta y_k^T (AV_k)^T r_k = 0$, $\delta y_k \neq 0$ なので $(AV_k)^T r_k = 0$ である。直交化の議論から、 $AV_k = V_{k+1} \tilde{H}_k$ より

$$\begin{aligned} (AV_k)^T r_k &= (AV_k)^T (r_0 - AV_k y_k) \\ &= \tilde{H}_k^T V_{k+1}^T (r_0 - V_{k+1} \tilde{H}_k y_k) \\ &= \tilde{H}_k^T V_{k+1}^T r_0 - \tilde{H}_k^T \tilde{H}_k y_k = 0 \end{aligned} \quad (320)$$

ここで、 $r_0 = \|r_0\|v_1 = \|r_0\|V_{k+1}e_1^{(k+1)}$ より $V_{k+1}^T r_0 = \|r_0\|e_1^{(k+1)}$ なので

$$\tilde{H}_k^T \tilde{H}_k y_k = \|r_0\| \tilde{H}_k^T e_1^{(k+1)} \quad (321)$$

である。これは正規方程式の形なので、 $\|\|r_0\|e_1^{(k+1)} - \tilde{H}_k y_k\|$ の最小化を考えていることになる。上式は QR 分解を利用して解ける。

Algorithm 33 FOM(残差判定版)

```

$$\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$$

$$\hat{\mathbf{b}} = \|\mathbf{r}_0\| \mathbf{e}_1^{(i_{max}+1)}$$

$$t = 0$$

$$n_s = i_{max}$$
for  $i = 1, 2, \dots, i_{max}$  do  
   $\mathbf{w} = A\mathbf{v}_i$   
  for  $j = 1, 2, \dots, i$  do  
     $h_{ji} = \mathbf{w} \cdot \mathbf{v}_j$   
     $\mathbf{w} = \mathbf{w} - h_{ji}\mathbf{v}_j$   
  end for  
   $h_{i+1,i} = \|\mathbf{w}\|$   
   $\mathbf{v}_{i+1} = \frac{\mathbf{w}}{h_{i+1,i}}$   
  if  $i > 1$  then  
     $h_{i-1,i-1} = t$   
     $h_{i,i-1} = 0$   
     $\hat{b}_i = -s_{i-1}\hat{b}_{i-1}$   
     $\hat{b}_{i-1} = c_{i-1}\hat{b}_{i-1}$   
  end if  
  for  $j = 2, 3, \dots, i$  do  
     $r_1 = c_{j-1}h_{j-1,i} + s_{j-1}h_{j,i}$   
     $h_{ji} = -s_{j-1}h_{j-1,i} + c_{j-1}h_{ji}$   
     $h_{j-1,i} = r_1$   
  end for  
   $t = \sqrt{h_{ii}^2 + h_{i+1,i}^2}$   
   $c_i = \frac{h_{ii}}{t}$   
   $s_i = \frac{h_{i+1,i}}{t}$   
   $\|\mathbf{r}\| \leftarrow \|\mathbf{w}\| \left| \frac{\hat{b}_i}{h_{ii}} \right|$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
     $n_s = i$   
  loop end  
  end if  
end for  
 $\mathbf{y} \leftarrow H(1 : n_s, 1 : n_s)\mathbf{y} = \hat{\mathbf{b}}(1 : n_s)$   
 $\mathbf{x} = \mathbf{x}_0 + V_{n_s}\mathbf{y}$ 

---


```

Algorithm 34 一般化最小残差法

```

 $v_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$ 
for  $i = 1, 2, \dots, n_s$  do
   $\mathbf{w} = A\mathbf{v}_i$ 
  for  $j = 1, 2, \dots, i$  do
     $h_{ji} = \mathbf{w} \cdot \mathbf{v}_j$ 
     $\mathbf{w} = \mathbf{w} - h_{ji}\mathbf{v}_j$ 
  end for
   $h_{i+1,i} = \|\mathbf{w}\|$ 
  if  $i < n_s$  then
     $\mathbf{v}_{i+1} = \frac{\mathbf{w}}{h_{i+1,i}}$ 
  end if
end for
 $Q, R \leftarrow \tilde{H}_{n_s}$ 
 $\mathbf{y} \leftarrow R\mathbf{y} = \|\mathbf{r}_0\|Q^T \mathbf{e}_1^{(n_s+1)}$ 
 $\mathbf{x} = \mathbf{x}_0 + V_{n_s}\mathbf{y}$ 

```

もっと直接的には

$$\begin{aligned}
 \mathbf{r}_k &= \mathbf{r}_0 - AV_k \mathbf{y}_k \\
 &= \|\mathbf{r}_0\| V_{k+1} \mathbf{e}_1^{(k+1)} - V_{k+1} \tilde{H}_k \mathbf{y}_k \\
 &= V_{k+1} \left(\|\mathbf{r}_0\| \mathbf{e}_1^{(k+1)} - \tilde{H}_k \mathbf{y}_k \right)
 \end{aligned} \tag{322}$$

なので

$$\begin{aligned}
 \|\mathbf{r}_k\| &= \|V_{k+1}\| \|\|\mathbf{r}_0\| \mathbf{e}_1^{(k+1)} - \tilde{H}_k \mathbf{y}_k\| \\
 &= \|\|\mathbf{r}_0\| \mathbf{e}_1^{(k+1)} - \tilde{H}_k \mathbf{y}_k\|
 \end{aligned} \tag{323}$$

である。つまり、これ自体が残差ベクトルノルムなのである。

一般化最小残差法の手順をアルゴリズム 34 に示す。見かけ上は FOM とほとんど変わらず、 H_k が $\tilde{H}_k$ になったくらいである。ただし、 Q , R のサイズはそれぞれ $n_s + 1 \times n_s$, $n_s \times n_s$ である。

残差判定版の手順をアルゴリズム 35 に示す。残差ベクトルノルムの計算については、この場合は $\|\mathbf{r}\| = |\hat{b}_{i+1}|$ である。 $\tilde{H}_k$ を Givens 回転で $\tilde{Q}$, $\tilde{R}$ に分解するものとする。ただし、それぞれのサイズは $k+1 \times k+1$,

$k+1 \times k$ とする. $\hat{\mathbf{b}} = \|\mathbf{r}_0\| \tilde{Q}^T \mathbf{e}_1^{(k+1)}$ とすると, 残差ベクトルノルムは

$$\begin{aligned}
\|\mathbf{r}_k\| &= \|\|\mathbf{r}_0\| \mathbf{e}_1^{(k+1)} - \tilde{H}_k \mathbf{y}_k\| \\
&= \|\|\mathbf{r}_0\| \mathbf{e}_1^{(k+1)} - \tilde{Q} \tilde{R} \mathbf{y}_k\| \\
&= \|\tilde{Q}(\|\mathbf{r}_0\| \tilde{Q}^T \mathbf{e}_1^{(k+1)} - \tilde{R} \mathbf{y}_k)\| \\
&= \|\tilde{Q}\| \|\hat{\mathbf{b}} - \tilde{R} \mathbf{y}_k\| \\
&= \|\hat{\mathbf{b}} - \tilde{R} \mathbf{y}_k\| \\
&= \|\|\mathbf{r}_0\| Q^T \mathbf{e}_1^{(k+1)} - R \mathbf{y}_k + \hat{b}_{k+1} \mathbf{e}_{k+1}^{(k+1)}\| \\
&= |\hat{b}_{k+1}|
\end{aligned} \tag{324}$$

となる.

あとは FOM と比べて計算の順番が少し変わったくらいだが, 性能は異なる. 実際に計算してみるとわかるが, FOM は $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ となるが, 一般化最小残差法では $\mathbf{r}_i \cdot A \mathbf{r}_j = 0$ である. 一般化最小残差法の場合は残差ベクトルノルムが単調に減少するが, FOM はそうとは限らない.

この種類の手法には, 反復回数が増えるに応じて行列のサイズが大きくなっていくという問題がある. つまり, 計算量とメモリが多くなる. これに対処するために, 何回か計算したら (言い換えると, ある大きさの部分空間で計算したら), 得られた解を初期値にして最初からやり直すという方法 (restart) が考えられている.

ところで, FOM が共役勾配法の非対称版なのはよいが, 一般化最小残差法については一般化共役残差法と同じものなのではないか, という疑問が浮かぶかもしれない. 理屈上はそうでも, 実際に計算するとそうとはかぎらない. 数学的に持っていると言われる性質 (ベクトルの共役性とか直交性など) は, 問題によっては数値計算の過程であつという間に失われることがあり, その差異が各手法の性能の差となって現れることになる. また, 計算量やメモリの使用量もそれぞれ異なる.

一般化最小残差法の対称版を考えてみる. これは MINRES 法と呼ばれるが, 共役残差法と等価な方法である. Arnoldi 過程の代わりに Lanczos 過程を考える. 係数行列が対称だと Hessenberg 行列は三重対角行列になるが, それを QR 分解する場合は, R は対角要素から 3 つの要素だけ考えればよい.

一般化最小残差法で解くべき方程式は

$$\tilde{H}_k^T \tilde{H}_k \mathbf{y}_k = \|\mathbf{r}_0\| \tilde{H}_k^T \mathbf{e}_1^{(k+1)} \tag{325}$$

であつた. QR 分解を用いると

$$R \mathbf{y}_k = \|\mathbf{r}_0\| Q^T \mathbf{e}_1^{(k+1)} \tag{326}$$

である. これより

$$\begin{aligned}
\mathbf{x}_k - \mathbf{x}_0 &= V_k \mathbf{y}_k \\
&= V_k R^{-1} \left(\|\mathbf{r}_0\| Q^T \mathbf{e}_1^{(k+1)} \right) \\
&= P_k \hat{\mathbf{b}}
\end{aligned} \tag{327}$$

Algorithm 35 一般化最小残差法 (残差判定版)

```

 $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$ 
 $\hat{\mathbf{b}} = \|\mathbf{r}_0\| \mathbf{e}_1^{(i_{max}+1)}$ 
 $n_s = i_{max}$ 
for  $i = 1, 2, \dots, i_{max}$  do
     $\mathbf{w} = A\mathbf{v}_i$ 
    for  $j = 1, 2, \dots, i$  do
         $h_{ji} = \mathbf{w} \cdot \mathbf{v}_j$ 
         $\mathbf{w} = \mathbf{w} - h_{ji}\mathbf{v}_j$ 
    end for
     $h_{i+1,i} = \|\mathbf{w}\|$ 
     $\mathbf{v}_{i+1} = \frac{\mathbf{w}}{h_{i+1,i}}$ 
    for  $j = 2, 3, \dots, i$  do
         $r_1 = c_{j-1}h_{j-1,i} + s_{j-1}h_{j,i}$ 
         $h_{ji} = -s_{j-1}h_{j-1,i} + c_{j-1}h_{j,i}$ 
         $h_{j-1,i} = r_1$ 
    end for
     $t = \sqrt{h_{ii}^2 + h_{i+1,i}^2}$ 
     $c_i = \frac{h_{ii}}{t}$ 
     $s_i = \frac{h_{i+1,i}}{t}$ 
     $h_{ii} = t$ 
     $h_{i+1,i} = 0$ 
     $\hat{b}_{i+1} = -s_i\hat{b}_i$ 
     $\hat{b}_i = c_i\hat{b}_i$ 
     $\|\mathbf{r}\| \leftarrow |\hat{b}_{i+1}|$ 
    if  $\|\mathbf{r}\| < \varepsilon$  then
         $n_s = i$ 
    loop end
    end if
end for
 $\mathbf{y} \leftarrow H(1:n_s, 1:n_s)\mathbf{y} = \hat{\mathbf{b}}(1:n_s)$ 
 $\mathbf{x} = \mathbf{x}_0 + V_{n_s}\mathbf{y}$ 

```

ここで $P_k = V_k R^{-1}$ とした. $V_k = P_k R$ という事なので

$$V_k = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_k] \begin{bmatrix} r_{11} & r_{12} & r_{13} & & & & & \\ & r_{22} & r_{23} & r_{24} & & & & \\ & & r_{33} & r_{34} & & & & \\ & & & r_{44} & & & & \\ & & & & \ddots & & & \\ & & & & & r_{k-2,k} & & \\ & & & & & r_{k-1,k} & & \\ & & & & & & r_{kk} & \end{bmatrix} \quad (328)$$

より

$$\mathbf{v}_k = r_{k-2,k} \mathbf{p}_{k-2} + r_{k-1,k} \mathbf{p}_{k-1} + r_{k,k} \mathbf{p}_k \quad (329)$$

したがって

$$\mathbf{p}_k = \frac{\mathbf{v}_k - r_{k-2,k} \mathbf{p}_{k-2} - r_{k-1,k} \mathbf{p}_{k-1}}{r_{kk}} \quad (330)$$

$\mathbf{x}_{k+1}$ は次式で求められる.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tilde{b}_k \mathbf{p}_k \quad (331)$$

MINRES 法の手順をアルゴリズム 36 に示す. ここでは $\alpha = r_{kk}$, $\beta = r_{k-1,k}$, $\gamma = r_{k-2,k}$ としている. 少しややこしい感じがするが, 一般化最小残差法を Arnoldi 過程から Lanczos 過程に変えて, 対称行列用に最適化してあるだけである.

参考文献

- [1] 戸川隼人, 永坂秀子 監修, 佐藤次男, 中村理一郎 著: よくわかる数値計算, 日刊工業新聞社 (2001).
- [2] 小国力 編著, 村田武郎, 三好俊郎, J.J. ドンガラ, 長谷川秀彦 著: 行列計算ソフトウェア, 丸善 (1991).
- [3] 戸川隼人: 新装版 UNIX ワークステーションによる科学技術計算ハンドブック 基礎編 C 言語版, サイエンス社 (1992).
- [4] ギルバート・ストラング: 世界標準 MIT 教科書 ストラング: 線形代数イントロダクション, 近代科学社 (2015).
- [5] ギルバート・ストラング: 世界標準 MIT 教科書 ストラング: 計算理工学, 近代科学社 (2017).
- [6] 一般社団法人 日本計算工学会 編, 藤野清次, 阿部邦美, 杉原正顕, 中嶋徳正 著: 線型方程式の反復解法, 丸善 (2013).
- [7] Henk A. van der Vorst: Iterative Krylov Methods for Large Linear Systems, Cambridge University Press (2003).
- [8] Yousef Saad: Iterative Methods for Sparse Linear Systems (Second Edition), SIAM (2003).

Algorithm 36 MINRES 法

```
 $\mathbf{x} = \mathbf{x}_0$   
 $\mathbf{p} = \mathbf{p}_0 = \mathbf{p}_{00} = \mathbf{0}$   
 $\mathbf{v} = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$   
 $\hat{b} = \|\mathbf{r}_0\|, \hat{b}_1 = 0$   
 $\mathbf{v}_0 = \mathbf{0}$   
 $\beta = 0$   
 $c = c_0 = 1$   
 $s = s_0 = 1$   
for  $i = 1, 2, \dots, i_{max}$  do  
   $\mathbf{w} = A\mathbf{v}$   
   $\alpha = \mathbf{w} \cdot \mathbf{v}$   
   $\mathbf{w} = \mathbf{w} - \alpha\mathbf{v} - \beta\mathbf{v}_0$   
   $\beta_1 = \|\mathbf{w}\|$   
   $\mathbf{v}_1 = \frac{\mathbf{w}}{\beta_1}$   
   $\hat{\alpha} = c\alpha - c_0s\beta$   
   $\hat{\beta} = s\alpha + c_0c\beta$   
   $\gamma = s_0\beta$   
   $c_0 = c, s_0 = s$   
   $t = \sqrt{\hat{\alpha}^2 + \beta_1^2}$   
   $c = \frac{\hat{\alpha}}{t}, s = \frac{\beta_1}{t}$   
   $\hat{b}_1 = -s\hat{b}, \hat{b} = c\hat{b}$   
   $\mathbf{p} = \frac{\mathbf{v} - \beta\mathbf{p}_0 - \gamma\mathbf{p}_{00}}{t}$   
   $\mathbf{x} = \mathbf{x} + \hat{b}\mathbf{p}$   
   $\|\mathbf{r}\| \leftarrow |\hat{b}_1|$   
  if  $\|\mathbf{r}\| < \varepsilon$  then  
    loop end  
  end if  
   $\mathbf{p}_{00} = \mathbf{p}_0, \mathbf{p}_0 = \mathbf{p}$   
   $\mathbf{v}_0 = \mathbf{v}, \mathbf{v} = \mathbf{v}_1$   
   $\beta = \beta_1$   
   $\hat{b} = \hat{b}_1$   
end for
```
