

ハイパフォーマンス コンピューティング： ガウスの消去法

目的

- 数値シミュレーションなどでは、最終的に連立一次方程式を解くことが多く、その取り扱いが重要。
- 連立一次方程式については、多くの解法(アルゴリズム)が開発されている。本講義では、代表的な「ガウスの消去法」とそれを用いた「LU分解法」について学習し、プログラム作成の実習を行う。

ガウスの消去法

- 例として、3元連立一次方程式を考える

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

書き換えると



$$Ax = b$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

前進消去(1)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \cdots (1) \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \cdots (2) \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \cdots (3) \end{cases}$$

式(1)の a_{11} が0でないとき、(2),(3)の x_1 の係数を消去する。

- (2)から(1)* (a_{21}/a_{11}) を引くと、(2)の x_1 の係数は0になる。
- 同様に、(3)から(1)* (a_{31}/a_{11}) を引くと、(3)の x_1 の係数は0になる。

前進消去(2)

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \cdots (1) \\ \boxed{a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \cdots (2)'} \\ \hline \boxed{a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = b_3^{(1)} \cdots (3)'} \end{array} \right.$$

式(2)'の a_{22} が0でないとき、(3)'の x_2 の係数を消去する。

- (3)'から(2)'*(a_{32}/a_{22})を引くと、(3)'の x_2 の係数は0になる

※注意: 式(2),(3)は、はじめの式から更新されるため、式(2)',(3)'とし、係数に添字を付けている。

前進消去(3)

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \cdots (1) \\ \begin{array}{c} \text{blue triangle} \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \cdots (2)' \\ a_{33}^{(2)}x_3 = b_3^{(2)} \cdots (3)'' \end{array} \end{array} \right.$$

- この形式を**上三角方程式**とよび、以上の過程を**前進消去**と呼ぶ。

(※)途中で a_{ii} の係数が0になる場合、この過程は失敗する
→対処法は軸交換(後で説明)

後退代入

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \cdots (1) \\ \begin{array}{c} \text{▲} \\ \text{▲} \\ \text{▲} \end{array} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \cdots (2)' \\ a_{33}^{(2)}x_3 = b_3^{(2)} \cdots (3)'' \end{array} \right.$$

- 上三角化された方程式は、後退代入により、下から順番に容易に解を求めることができる。
- $x_3 = b_3^{(2)} / a_{33}^{(2)}$
- $x_2 = (b_2^{(1)} - a_{23}^{(1)}x_3) / a_{22}^{(1)}$
- $x_1 = (b_1 - a_{12}x_2 - a_{13}x_3) / a_{11}$

練習: ガウスの消去法の手計算

- ガウスの消去法を用いて、以下の方程式の解を紙とペンを使って計算してみよう。

$$\begin{cases} 2x_1 - x_2 + x_3 = 0 \cdots (1) \\ -x_1 + 2x_2 - x_3 = 1 \cdots (2) \\ 2x_1 - 2x_2 - x_3 = -3 \cdots (3) \end{cases}$$

アルゴリズムを理解しないとプログラムは書けないので、しっかり頑張りましょう！

練習: ガウスの消去法のプログラム

- 3元連立一次方程式を解くプログラムをMATLABで実行する。
- 網掛け部分は各自で考えて記述すること。
- 完成したら、前のページの例題で試してみよう。

```
function x = pregauss(A,b)
x = zeros(3,1);
```

```
%前進消去
```

```
%1段目
```

```
%2行目の1列目を消去
```

```
w = A(2,1)/A(1,1);
```

```
A(2,1) = A(2,1) - w*A(1,1);
```

```
A(2,2) = A(2,2) - w*A(1,2);
```

```
A(2,3) = A(2,3) - w*A(1,3);
```

```
b(2) = b(2) - w*b(1);
```

```
%1段目
```

```
3行目の1列目を消去
```

```
%2段目
```

```
3行目の2列目を消去
```

```
%後退代入
```

```
x(3) = b(3)/A(3,3);
```

```
x(2) = (b(2) - A(2,3)*x(3))/A(2,2);
```

```
X(1)の後退代入
```

```
end
```

実習：前進消去のプログラムを作成する

- n元連立一次方程式の解を求める前進消去プログラムを作成しなさい。

$$\begin{cases}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \cdots (1) \\
 a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}x_4 = b_2^{(1)} \cdots (2)' \quad \text{k行目} \\
 \text{消去} \left\{ \begin{array}{l}
 a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}x_4 = b_3^{(1)} \cdots (3)' \\
 a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}x_4 = b_4^{(1)} \cdots (4)'
 \end{array} \right\} \text{i=k+1} \sim \text{n行} \\
 \quad \quad \quad \underbrace{\hspace{10em}}_{\text{j=k+1} \sim \text{n列}}
 \end{cases}$$

k列目

これを一般的に表すと

$k=1, 2, \dots, n-1$ に対して

$$w_{ik} = A(i, k) / A(k, k) \quad (i=k+1 \sim n)$$

$$A(i, j) = A(i, j) - w_{ik} \times A(k, j) \quad (i=k+1 \sim n, j=k+1 \sim n)$$

$$b(i) = b(i) - w_{ik} \times b(k) \quad (i=k+1 \sim n)$$

実習：後退代入のプログラムを作成する

- n元連立一次方程式の解を求める後退代入のプログラムを作成しなさい。

$$x(n) = b(n) / A(n, n)$$

$$x(n-1) = (b(n-1) - A(n-1, n) * x(n)) / A(n-1, n-1)$$

...

$$x(1) = \left(b(1) - \sum_{j=2}^n A(1, j) * x(j) \right) / A(1, 1)$$

i=n~1行

以上を一般的に表すと、 $i=n, n-1, \dots, 1$ に対して

$$x(i) = \left(b(i) - \sum_{j=i+1}^n A(i, j) * x(j) \right) / A(i, i)$$

実習: ガウスの消去法のプログラム作成

- n元連立一次方程式を解けるようにする。
- ファイル名は「mygauss1.m」とする。
- 関数は $x = \text{mygauss1}(A,b)$ のように利用できるようにする。

【実行例】

```
>> A=[1 2 3; 4 5 6; 7 8 10];  
>> b=[1; 1; 1];  
>> x=mygauss1(A,b)  
x =  
-1.0000  
1.0000  
0.0000
```

☆答えが合っているか、確認☆

```
>> A\b  
ans =  
-1.0000  
1.0000  
0.0000
```

演習問題:

ガウスの消去法で連立一次方程式の解を求める

- 以下の連立一次方程式をガウスの消去法で解きなさい。

$$(1) \begin{cases} -2x + y = -3 \\ x + y = 3 \end{cases}$$

$$(2) \begin{cases} 10x - 7y = 7 \\ -3x + 2y + 6z = 4 \\ 5x + y + 5z = 6 \end{cases}$$

$$(3) \begin{cases} x + 4y - z + w = 2 \\ 2x + 7y + z - 2w = 16 \\ x + 4y - z + 2w = -15 \\ 3x - 10y - 2z + 5w = -15 \end{cases}$$

※解が「NaN」となり失敗する式は、なぜ解けないか考えてみましょう。

部分軸交換

- 解が「NaN」となり失敗するのはなぜか？

• **k=1**

$$\begin{cases} x + 4y - z + w = 2 \\ 2x + 7y + z - 2w = 16 \\ x + 4y - z + 2w = -15 \\ 3x - 10y - 2z + 5w = -15 \end{cases}$$

• **k=2**

$$\begin{cases} x + 4y - z + w = 2 \\ -y + 3z - 4w = 12 \\ w = -17 \\ -22y + z + 2w = -21 \end{cases}$$

• **k=3**

$$\begin{cases} x + 4y - z + w = 2 \\ -y + 3z - 4w = 12 \\ \square w = -17 \\ -65z + 90w = -285 \end{cases}$$

$a_{33} = 0$

- 前進消去の過程(k段目)で a_{kk} の係数が0になる場合、0での除算が発生して、失敗する。
- 対処法 ⇒ 部分軸交換(部分ピボット選択):
同じ列で、k行目以降から絶対値が最大の要素を探し
その行と交換する

部分軸交換

- 部分軸交換を適用すると

$$\begin{cases} x + 4y - z + w = 2 \\ 2x + 7y + z - 2w = 16 \\ x + 4y - z + 2w = -15 \\ 3x - 10y - 2z + 5w = -15 \end{cases} \quad \text{部分軸交換後}$$



$$\begin{cases} 3x - 10y - 2z + 5w = -15 \\ 2x + 7y + z - 2w = 16 \\ x + 4y - z + 2w = -15 \\ x + 4y - z + w = 2 \end{cases}$$

```
>> x=mygauss1(A,b)
```

```
x =
    NaN
    NaN
    NaN
    NaN
```

```
>> x=mygauss2(A,b)
```

```
x =
    5.6923
   -1.4615
  -19.1538
  -17.0000
```

答えが出ました!

部分軸交換の具体例

• 部分軸交換のアルゴリズム 具体例

• $k=1$

$$\begin{array}{l}
 \boxed{x + 4y - z + w = 2} \\
 \left. \begin{array}{l}
 2x + 7y + z - 2w = 16 \\
 x + 4y - z + 2w = -15 \\
 3x - 10y - 2z + 5w = -15
 \end{array} \right\} i=k+1 \sim n \text{行}
 \end{array}
 \xrightarrow{\substack{\text{4行目と} \\ \text{1行目を交換}}}
 \begin{array}{l}
 \boxed{3x - 10y - 2z + 5w = -15} \\
 2x + 7y + z - 2w = 16 \\
 x + 4y - z + 2w = -15 \\
 \boxed{x + 4y - z + w = 2}
 \end{array}$$

- 絶対値が最大となる係数の絶対値の候補を c 、その行番号を r とする。

初期値: $c = |A(1,1)| = 1, r = 1$

$i=2$: $|A(i,k)| = 2 \Rightarrow c$ より大きい $\dots c = 2, r = 2$ に更新

$i=3$: $|A(i,k)| = 1 \Rightarrow c$ より小さい $\dots c = 2, r = 2$ のまま更新しない

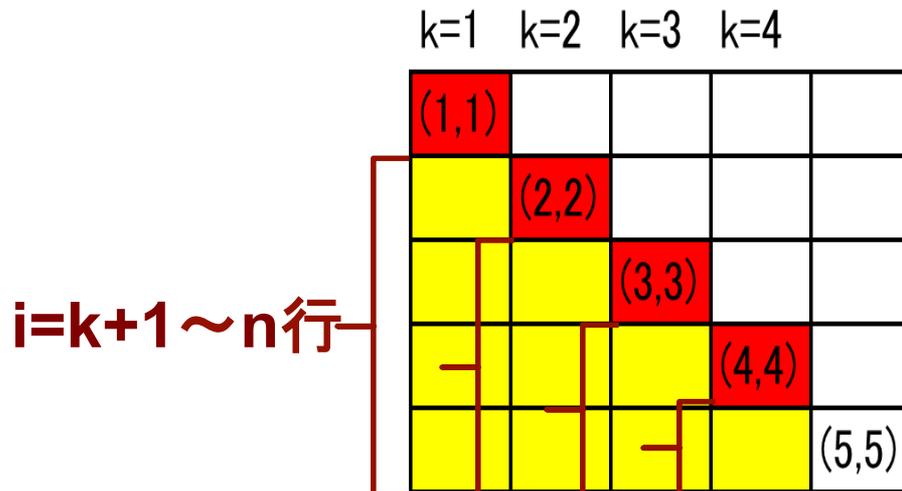
$i=4$: $|A(i,k)| = 3 \Rightarrow c$ より大きい $\dots c = 3, r = 4$ に更新

よって、 $k=1$ について、 k 列目の k 行目以降で絶対値最大の係数は、 $r = 4$ 行目にある。

\Rightarrow 4行目と1行目を交換

実習：部分軸交換を追加

- N次元連立一次方程式の解を求める、ガウスの消去法のプログラムに部分軸交換を追加しなさい。



前ページのアルゴリズムをk=1~n-1まで繰り返します。

実習：部分軸交換を追加

```
function x = mygauss2(A,b)
% ガウスの消去法(部分軸交換あり)

n = length(b);
x = zeros(n,1);

% 前進消去
for k=1:n-1
    % 部分軸交換開始
    % 係数の絶対値が最大のものを探索
    c = abs(A(k,k));
    r = k;
    for i=k+1:n
        
    end
end
```

続く



```
if c == 0 % 対角成分以下がすべて0
    error('A is singular.');
```

elseif r ~= k % k行とr行を交換



```
end
% 部分軸交換終了
```

■実行例

```
>> x=mygauss1(A,b) % 部分軸交換なしのガウスの消去法
x =
    NaN
    NaN
    NaN
    NaN
>> x=mygauss2(A,b)
x =
    5.6923
   -1.4615
  -19.1538
  -17.0000
```

何をしたら良いか、よくわからない人は・・・

まず、以下の問題を考えてみよう。

1. 2つの数 a と b を与えたら、絶対値が大きい方の数を入力する関数 `mymax` を作成する。

`function c = mymax(a,b)` のような形で作成。

2. ベクトル v を与えたら、絶対値が一番大きい要素 $c = v_k$ とその要素番号 k を出力する関数 `maxvec` を作成する。

`function [c,k] = maxvec(v)` のような形で作成。

3. n 次の行列 A と正の整数 $k, r < n$ を与えたら、その行列の k 行目と r 行目を交換した行列 B を出力する関数 `mymat1` を作成する。

`function B = mymat1(A,k,r)` のような形で作成。