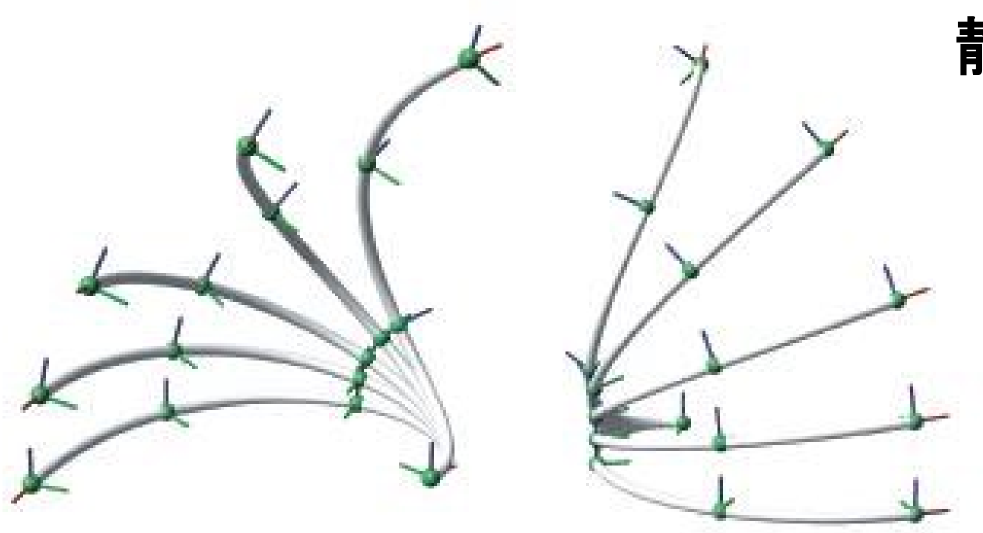
数値解析

2022年度前期　第8週 [6月9日]



静岡大学
工学研究科機械工学専攻
ロボット・計測情報講座
創造科学技術大学院
情報科学専攻

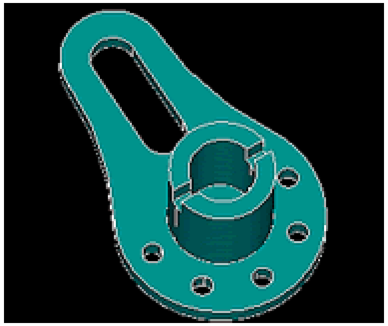
三浦 憲二郎

講義アウトライン [6月9日]

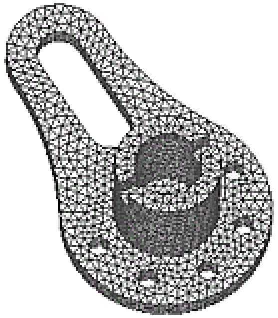
- 連立1次方程式の直接解法
  - ガウス消去法

連立1次方程式

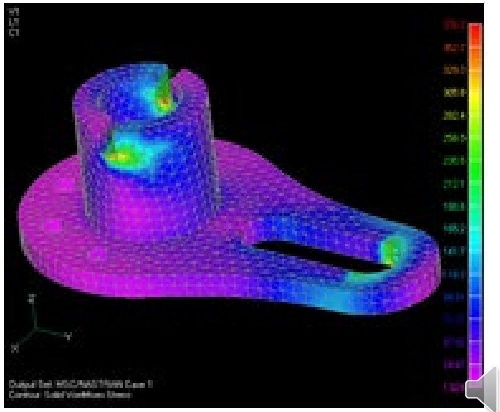
$$Ax = b$$



- 連立1次方程式の重要性



- 非線形の問題は基本的には解けない.
- 非線形問題を線形化して解く.
  - 複雑な構造物，単純な要素に分解
  - 各要素に対して線形方程式を立てる.
  - 重ね合わせの原理により統合
  - ただし，変数の数は膨大
  - コンピュータにより数値的に解く.



ガウス消去法

連立1次方程式の解法

$x-y=1$ (1)

$x+2y=4$ (2)

１．代入法

式(1)より　$y=x-1$, 式(2)に代入

$x+2(x-1)=4$, したがって$3x=6$, よって$x=2$, 式(1)より$y=1$

２．加減法

式(2)より式(1)を引く: $3y=3$, したがって$y=1$, 式(1)より$x=2$

ガウス消去法は，加減法をコンピュータに適した方法で行う.

ガウス消去法:3元

$$
\begin{aligned}
3x_0 + x_1 + 2x_2 &= 13 \\
5x_0 + x_1 + 3x_2 &= 20 \\
4x_0 + 2x_1 + x_2 &= 13
\end{aligned}
\qquad
\begin{bmatrix} 3 & 1 & 2 \\ 5 & 1 & 3 \\ 4 & 2 & 1 \end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}
=
\begin{bmatrix} 13 \\ 20 \\ 13 \end{bmatrix}
$$

(1) 第1列の2行目と3行目を0にする.
「第1行×(−5/3)+第2行目」
「第1行×(−4/3)+第3行目」

$$
\begin{bmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & \frac{2}{3} & -\frac{5}{3} \end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}
=
\begin{bmatrix} 13 \\ -\frac{5}{3} \\ -\frac{13}{3} \end{bmatrix}
$$

(2) 第2列の3行目を0にする.
「第2行×1+第3行」
(1), (2) : 前進消去

$$
\begin{bmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & 0 & -2 \end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}
=
\begin{bmatrix} 13 \\ -\frac{5}{3} \\ -6 \end{bmatrix}
$$

(3) x3, x2, x1の順に代入して答えを求める.
(3) : 後退代入

ガウス消去法:n元

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,2} & \cdots & a_{n-1,n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

(前進消去)

If $a_{00} \neq 0$

$$\alpha_{i0} = -\frac{a_{i0}}{a_{00}}, \quad i = 1, 2, \ldots, n-1$$

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ 0 & a_{11}^{(1)} & \cdots & a_{1,n-1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n-1,2}^{(1)} & \cdots & a_{n-1,n-1}^{(1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1^{(1)} \\ \vdots \\ b_{n-1}^{(1)} \end{bmatrix}$$

$$a_{ij}^{(1)} = a_{ij} + \alpha_{i0} a_{0j}, b_i^{(1)} = b_i + \alpha_{i0} b_0, i = 1,2,\cdots, n-1$$

ガウス消去法：n元

（後退代入）

$$
\begin{aligned}
x_{n-1} &= \frac{b_{n-1}^{(n-1)}}{a_{n-1,n-1}^{(n-1)}} \\
x_{n-2} &= \frac{b_{n-2}^{(n-2)} - a_{n-2,n-1}^{(n-2)} x_{n-1}}{a_{n-2,n-2}^{(n-2)}} \\
x_k &= \frac{b_k^{(k)} - \sum_{j=k+1}^{n-1} a_{kj}^{(k)} x_j}{a_{kk}^{(k)}}
\end{aligned}
$$

ガウス消去法:アルゴリズム

行列Aとベクトルbの入力

/* 前進消去 */

for $k = 0, 1, \cdots, n-2$
  for $i = k+1, k+2, \cdots, n-1$
    $\alpha \leftarrow -\frac{a_{ik}}{a_{kk}}$
    for $j = k+1, k+2, \cdots, n-1$
      $a_{ij} \leftarrow a_{ij} + \alpha a_{kj}$
    end for
    $b_i = b_i + \alpha b_k$
  end for
end for

/* 後退代入 */

for $k = n-1, n-2, \cdots, 0$
  $b_k \leftarrow \frac{b_k - \sum_{j=k+1}^{n-1} a_{kj} b_j}{a_{kk}}$
end for

bを出力 /* 答えはbに上書き*/

ガウス消去法:プログラム

```
#include <stdio.h>
#include <stdlib.h>
#define N 4 /* N次正方行列 */
void input_matrix(double a[N][N],char c,FILE* fin, FILE* fout);
void input_vector(double b[N],char c,FILE* fin,FILE* fout);
void simple_gauss(double a[N][N],double b[N]);
int main(void){
    FILE *fin, *fout;
    double a[N][N], b[N];
    int i;
    if((fin=fopen("input.dat","r"))==NULL) exit(1);
    if((fout=fopen("output.dat","w"))==NULL) exit(1);
    input_matrix(a,'A',fin,fout); input_vector(b,'b',fin,fout);
    simple_gauss(a,b);
    fprintf(fout,"Ax=bの解は次の通りです¥n");
    for(i=0;i<N;i++){ fprintf(fout,"%f¥n",b[i]);}
    fclose(fin); fclose(fout);
    return 0;
}
```

ガウス消去法：プログラム

```
void simple_gauss(double a[N][N],double b[N]){
    int i,j,k;
    double alpha, tmp;
    for(k=0;k<N-1;k++){ /* 前進消去 */
        for(i=k+1;i<N;i++){
            alpha=-a[i][k]/a[k][k];
            for(j=k+1;j<N;j++){
                a[i][j]=a[i][j]+alpha*a[k][j];
            }
            b[i]=b[i]+alpha*b[k];
        }
    }
    b[N-1]=b[N-1]/a[N-1][N-1]; /* 後退代入 */
    for(k=N-2;k>=0;k--){
        tmp=b[k];
        for(j=k+1;j<N;j++){
            tmp=tmp-a[k][j]*b[j];
        }
        b[k]=tmp/a[k][k];
    }
}
```

データ入出力　p.19「C言語入門」p.282)

プログラム2.3改

```
#include <stdio.h>
#include <stdlib.h>
#define N 4
void input_matrix(double a[N][N],char c,FILE* fin,FILE* fout);
void input_vector(double b[N],char c,FILE* fin,FILE* fout);
int main(void)
{
    FILE *fin,*fout;
    double a[N][N],b[N];
    if((fin=fopen("input.dat","r"))==NULL){
        printf("ファイルは見つかりません:input.dat ¥n");
        exit(1);
    }
    if((fout=fopen("output.dat","w"))==NULL){
        printf("ファイルが作成できません:output.dat ¥n");
        exit(1);
    }
```

データ入出力　p.19「C言語入門」p.282)

プログラム2.3改　続き

```
    input_matrix(a,'A',fin,fout); /* 行列Aの入出力 */
    input_vector(b,'b',fin,fout); /* ベクトルbの入出力 */

    fclose(fin); fclose(fout);
}
void input_matrix(double a[N][N],char c,FILE* fin,FILE* fout){
    int i,j;
    fprintf(fout,"行列%cは次の通りです¥n",c);
    for(i=0;i<N;++i){
        for(j=0;j<N;++j){
            fscanf(fin,"%lf",&(a[i][j]));
            fprintf(fout,"%5.2f¥t",a[i][j]);
        }
        fprintf(fout,"¥n");
    }
}
```

データ入出力　p.19「C言語入門」p.282)

プログラム2.3改 続き

```
void input_vector(double b[N],char c,FILE* fin,FILE* fout){
    int i;
    fprintf(fout,"ベクトル%cは次の通りです¥n",c);
    for(i=0;i<N;++i){
        fscanf(fin,"%lf",&(b[i]));
        fprintf(fout,"%5.2f¥t",b[i]);
        fprintf(fout,"¥n");
    }
}
```

input.datの内容

```
1.0  2.0  1.0  1.0
4.0  5.0 -2.0  4.0
4.0  3.0 -3.0  1.0
2.0  1.0  1.0  3.0
-1.0 -7.0 -12.0 2.0
```

データ入出力 p.19「C言語入門」p.282)

プログラム2.3改 続き

output.datの内容

```
行列Aは次の通りです.
1.0  2.0  1.0  1.0
4.0  5.0 -2.0  4.0
4.0  3.0 -3.0  1.0
2.0  1.0  1.0  3.0
ベクトルbは次の通りです.
-1.0
-7.0
-12.0
 2.0
```

まとめ

- 連立1次方程式の直接解法
  - ガウス消去法
- C言語の基礎
  - 2次元配列
  - データ入出力