２０２３年前期　数値解析　第９回講義ノート

２０２３年６月１５日　三浦憲二郎

スライド１　講義アウトライン

- 連立方程式の反復解法
  - 「　　」行列と「　　」行列
  - 反復法の原理：「　　」の原理
  - 「　　」法
  - 「　　」法

スライド２　連立１次方程式　p.93

- 行列のほとんどの要素が 0　　$Ax = b$

  「　　」行列　　sparse matrix
- 逆にそのほとんどが 0 でない

  「　　」行列　　dense matrix
- 疎行列を効率よく解く方法

  「　　」解法

スライド３　反復法の原理

連立１次方程式

$\boldsymbol{Ax=b}$　(5.1)

を同値な方程式

$\boldsymbol{x=\phi(x)=Mx+Nb}$

に変形し，「　　」法により解を求める．$\boldsymbol{M}$を「　　」行列という．

⇩

「　　」写像の原理

スライド４ 縮小写像の原理　定理 5.1　p.94

$\boldsymbol{R^n}$　上で定義された関数$\boldsymbol{g(x)}$ が

(1) $\boldsymbol{x\in R^n \rightarrow g(x)\in R^n}$

(2) $\boldsymbol{x, y\in R\text{^}n \rightarrow ||g(x)-g(y)|| \leq L||x-y|| \in R\text{^}n}$

(3) $\boldsymbol{0\leq L<1}$

を満たすとき，方程式

$\boldsymbol{x=g(x)}$

の解$\boldsymbol{x=\alpha}$は$\boldsymbol{R^n}$においてただ１つ存在する．そして，$\boldsymbol{x^{(0)}}$ を「　　」とする反復に

$\boldsymbol{x^{(k+1)}=g(x^{(k)})}$

によって，生成される列$\boldsymbol{〚\{x〛\text{^}((k))\}}$は$\boldsymbol{k\rightarrow\infty}$のとき$\boldsymbol{\alpha}$に「　　」する．

スライド５　縮小写像の原理　定理 5.2　p.94

あるノルム||・||について|(|**M**|)|<**1**ならば反復法

$$\boldsymbol{x}^{((\boldsymbol{k}+\boldsymbol{1}))}=\boldsymbol{\phi}(\boldsymbol{x}^{((\boldsymbol{k})})=\boldsymbol{M}\boldsymbol{x}^{((\boldsymbol{k})})+\boldsymbol{N}\boldsymbol{b}$$

によって作られる列〖{$\boldsymbol{x}$}〗^(($\boldsymbol{k}$))}は(5.1)の解$\boldsymbol{x}$に収束する

(証明)

任意の$\boldsymbol{x}$, $\boldsymbol{y}\in\boldsymbol{R}^{\boldsymbol{n}}$に対して,

$$|(|\boldsymbol{\phi}(\boldsymbol{x})-\boldsymbol{\phi}(\boldsymbol{y})|)|=|(|\boldsymbol{M}\boldsymbol{x}-\boldsymbol{M}\boldsymbol{y}|)|\leq|(|\boldsymbol{M}|)|\cdot||\boldsymbol{x}-\boldsymbol{y}||(\leq||\boldsymbol{x}-\boldsymbol{y}|)|$$

が成り立つので，$\boldsymbol{\phi}(\boldsymbol{x})$は定理 5.1 の仮定を満たす．よって，　〖{$\boldsymbol{x}$}〗^(($\boldsymbol{k}$))}は解$\boldsymbol{x}$に収束する．

スライド６ 行列 A の分解

$$\boldsymbol{A}=\boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}$$

$$\boldsymbol{D}=\begin{bmatrix}\boldsymbol{a_{11}} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \boldsymbol{a_{nn}}\end{bmatrix}\quad \boldsymbol{L}=\begin{bmatrix}\boldsymbol{0} & \cdots & \boldsymbol{0}\\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{n1}} & \cdots & \boldsymbol{0}\end{bmatrix}\quad \boldsymbol{U}=\begin{bmatrix}\boldsymbol{0} & \cdots & \boldsymbol{a_{1n}}\\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0}\end{bmatrix}$$

スライド７　ヤコビ法　p.95

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{\phi}(\boldsymbol{x})\\ &= \boldsymbol{M}\boldsymbol{x}+\boldsymbol{N}\boldsymbol{b}\end{aligned}$$

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

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{N}^{-1}(\boldsymbol{I}-\boldsymbol{M})\\ &= \boldsymbol{D}(\boldsymbol{I}+\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}))\\ &= \boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}\end{aligned}$$

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

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}\\ \boldsymbol{D} &= \begin{bmatrix}a_0 0 & & \\ & \ddots & \\ & & a_{n-1,n-1}\end{bmatrix}\\ \boldsymbol{L} &= \begin{bmatrix}0 & \cdots & \cdots & 0\\ a_{10} & \ddots & & \vdots\\ \vdots & \ddots & \ddots & \vdots\\ a_{n-1,0} & \cdots & a_{n-1,n-2} & 0\end{bmatrix}\\ \boldsymbol{U} &= \begin{bmatrix}0 & a_0 1 & \cdots & a_{0,n-1}\\ \vdots & \ddots & \ddots & \vdots\\ \vdots & & \ddots & a_{n-2,n-1}\\ 0 & \cdots & \cdots & 0\end{bmatrix}\end{aligned}$$

スライド８　ヤコビ法：プログラム　　p.96

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N       10                          /* N 元方程式 */
#define EPS    pow(10.0, -8.0)          /* epsilon の設定 */
#define KMAX 100                          /* 最大反復回数 */
int double_comp(const void *s1 , const void *s2 );
/* 最大値ノルムの計算 a[0,...,n-1] */
double vector_norm_max(double a[N]);
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);
```

```
/* ヤコビ法 */
void jacobi_lin(double a[N][N], double b[N], double x[N]);
int main(void)
{
   FILE *fin, *fout;
   double a[N][N], b[N], x[N];
   int i;
   /* ファイルのオープン */
   if ( (fin = fopen( "input_sp.dat", "r")) == NULL )
   {
        printf("ファイルが見つかりません ：input_sp.dat ¥n");
        exit(1);
   }
   if( (fout = fopen( "output_sp.dat", "w")) == NULL )
   {
      printf("ファイルが作成できません ：output_sp.dat ¥n");
      exit(1);
    }
   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 */
   input_vector( x, 'x', fin, fout ); /* 初期ベクトル x0 の入出力*/
   jacobi_lin( a, b, x );                /* ヤコビ法 */
   /* 結果の出力 */
   fprintf( fout, "Ax=b の解は次の通りです¥n");
   for( i = 0 ; i < N ; i++)
   {
      fprintf(fout, "%f¥n", x[i]);
   }
   fclose(fin); fclose(fout);   /* ファイルのクローズ */
   return 0;
}
void jacobi_lin(double a[N][N], double b[N], double x[N])
{
   double eps, xn[N];
   int i, j, k=0;
   /* x <- x_k,    xn <- x_{k+1} */
   do{
      for(i = 0; i < N; i++)
      {
         xn[i] = b[i];
         for ( j = 0; j < N; j++)
         {
            xn[i] -= a[i][j]*x[j];
         }
         xn[i] += a[i][i]*x[i];       /* 余分に引いた分を加える */
         xn[i] /= a[i][i];
    }
      for( i = 0; i < N; i++)    x[i] = xn[i]-x[i];
      eps = vector_norm_max(x);
      for( i = 0; i < N; i++)    x[i] = xn[i]; /* 値を更新 */
      k++;
 }while(eps > EPS && k < KMAX);
   if ( k == KMAX ){
      printf("答えが見つかりませんでした¥n");
      exit(1);
   }
   else{
      printf("反復回数は%d 回です¥n", k);
```

input_sp.dat

```
5.0  2.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
2.0  5.0  2.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  2.0  5.0  2.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0  2.0  5.0  2.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0  0.0  2.0  5.0  2.0  0.0  0.0  0.0  0.0
0.0  0.0  0.0  0.0  2.0  5.0  2.0  0.0  0.0  0.0
0.0  0.0  0.0  0.0  0.0  2.0  5.0  2.0  0.0  0.0
0.0  0.0  0.0  0.0  0.0  0.0  2.0  5.0  2.0  0.0
0.0  0.0  0.0  0.0  0.0  0.0  0.0  2.0  5.0  2.0
0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  2.0  5.0
3.0  1.0  4.0  0.0  5.0  -1.0  6.0  -2.0  7.0  -15.0
1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
```

output_sp.dat

```
行列 A は次の通りです
 5.00      2.00      0.00      0.00      ...  0.00
 2.00      5.00      2.00      0.00      ...  0.00
      （省略）

ベクトル b は次の通りです
 3.00
 1.00
 4.00
 0.00
 5.00
-1.00
 6.00
-2.00
 7.00
-15.00
ベクトル x は次の通りです
 1.00
 1.00
 1.00
 1.00
 1.00
 1.00
 1.00
 1.00
 1.00
 1.00
Ax=b の解は次の通りです
1.000000
-1.000000
2.000000
-2.000000
```

```
      /* 反復回数を画面に表示 */
    }
}
/* 最大値ノルムの計算 a[0,...,n-1] */
double vector_norm_max( double a[N])
{
    int i;
    for ( i = 0 ; i < N; i++) a[i] = fabs(a[i]);
    /* 並べ換え：ソートする関数 qsort() */
    qsort(a, N, sizeof(a[0]), double_comp);
    /*    for(i=m;i<=n;i++) printf("a[%d]=%f¥n",i,a[i]);*/
    return a[0];
}
int double_comp( const void *s1 , const void *s2 )
{
    const double a1 = *((double *)s1); /* (double *)へキャスト */
    const double a2 = *((double *)s2); /* (double *)へキャスト */
    if( a1 < a2 )
    {
        return -1;
    }
    else if( a1 == a2 )
    {
        return 0;
    }
    else
    {
        return 1;
    }
}                                                    /*   反復回数   66 回   */
```

スライド 9　ガウス・ザイデル法　　p.100

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

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{N}^{-1}(\boldsymbol{I}-\boldsymbol{M}) \\ &= (\boldsymbol{D}+\boldsymbol{L})(\boldsymbol{I}-(\boldsymbol{D}+\boldsymbol{L}\boldsymbol{U})^{-1} \\ &= \boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}\end{aligned}$$

反復法　$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}\{\boldsymbol{b}-(L+U)\boldsymbol{x}^{(k)}\}$

反復回数　34 回

スライド 10 ガウスザイデル法：プログラム p.101

```
/* ガウス・ザイデル法 */
double* gauss_seidel(double** a, double* b, double* x)
{
        double eps, * xo, s, t;
        int i, j, k = 0;
        xo = dvector(1, N); /* xo[1...N] */
        do{
                /* xo <- x_k, x <- x_{k+1} */
                for (i = 1; i <= N; i++) xo[i] = x[i]; /* x_k に x_(k+1)を代入 */
                /* i=1 の処理 */
                t = 0.0;
```

```
                for (j = 2; j <= N; j++) t += a[1][j] * xo[j];
                x[1] = (b[1] - t) / a[1][1];
                /* i=2,3, . . .N の処理 */
                for (i = 2; i <= N; i++){
                        s = 0.0; t = 0.0;
                        for (j = 1; j < i; j++) s += a[i][j] * x[j];        /* i-1 列までの和 */
                        for (j = i+1; j <= N; j++) t += a[i][j] * x[j];     /* i+1 列以降の和 */
                        x[i] = (b[i] - s - t) / a[i][i];
                }
                for (i = 1; i <= N; i++) xo[i] = xo[i] - x[i];
                eps = vector_norm_max(xo, 1, N);
                k++;
        } while (eps > EPS && k < KMAX);
        free_dvector(xo, 1); /* */
        if (k == KMAX){
                printf("答えはがみつかりませんでした¥n");
                exit(1);
        }
        else{
                printf("反復回数は%d です¥n", k); /* 反復回数を画面に表示 */
                return x;
        }
}
```

スライド 11 データ入出力 p.19
プログラム 2.3

```
#include <stdio.h>
#include <stdlib.h>
#define N 4
void input_matrix(double **a,char c,FILE* fin,FILE* fout);
void input_vector(double *b,char c,FILE* fin,FILE* fout);
double **dmatrix(double **a,int nr1,int nr2,int nl1,int nl2);
Void free_dmatrix(double **a,int nr1,int nr2,int nl1,int nl2);
double *dvector(double *a, int i);
Void free_dvector(double *a, int i);
int main(void)
{
    FILE *fin,*fout;
    double **a,*b;
    a = dmatrix(1, N, 1, N); /* A[1…N][1…N] */
    b = dvector(1,N); /* b[1…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);
    }
    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,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");
	}
}
void input_vector(double *b,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");
	}
}
```