# ベクトル・行列の計算をしよう+反復法で 1 次方程式

樋口さぶろお

龍谷大学理工学部数理情報学科

数値計算法 L12(2010-07-09)

今日の目標

1. 1 次元, 2 次元配列を利用してベクトル・行列の計算ができるようになろう.
2. 反復法を使って 1 次方程式の解を求めるプログラムを書けるようになろう.

hig3.net

# 復習:数列=1 次元配列

数列 $\{x_n\}_{n=0,1,2,\dots}$ は 1 次元配列 $x[0], x[1], x[2], \dots$ で表現することもできた.

```
int i;
int n=10;
int x[10]; /* x[0],x[1],...,x[9] が使える */

x[0]=2.0;
for(i=0;i<n-1;i++){
    x[i+1]=2*x[i]+1;
}
```

- x: 配列の 名前
- i: 配列の , 添え字,
- x[3]: 配列の 要素
- n=10: 配列の , 長さ

# ベクトル=1 次元配列

$\boldsymbol{x} = (x_0, x_1, \ldots) \leftrightarrow$ x[i]
$n$ 次元ベクトルはサイズ $n$ の 1 次元配列で.
縦ベクトル横ベクトルどっちでも同じ.

```
double a[]={0,1,2,8,8,9,0};/*宣言と同時に初期化するならサイズ指定不要*/
double b[]={0,1,2,8,8,9,0};
int i;    /* ベクトルの成分番号=配列のインデックス*/
int n=7; /* ベクトルの次元=配列のサイズ */
double inner=0.0;

for(i=0;i<n;i++){
    inner+=a[i]*b[i];
}
print("%f\n",inner);
```

# 行列=2 次元配列

$A = (A_{ij}) \leftrightarrow$ a[i][j] a[行][列]
$n \times m$ 行列は, サイズ $n, m$ の 2 次元配列で.
$i$ $\boxed{行}$ $i$-th row 上から数えて何番目?
$j$ $\boxed{列}$ $j$-th column 左から数えて何番目?

```
double a[2][3] ={{2,3,4},
                 {2,3,4}};  /* 2行3列の行列 */
int nrow=2, mcolumn=3;
int i,j;     /*行, 列のインデックス*/
for(i=0;i<nrow;i++){     /*行列を出力*/
    for(j=0;j<mcolumn;j++){
        printf("%f ",a[i][j]); /* i行j列 */
    }
    printf("\n");
}
```

横ベクトル=$1 \times n$ 行列, 縦ベクトル=$n \times 1$ 行列.2 次元配列じゃないの?

# ベクトル・行列の積の成分表示

下の例は $n = 3$.
$n \times m$ 行列 $A = (A_{ij})$, $m$ 次元ベクトル $\boldsymbol{x} = (x_j), \boldsymbol{w} = (w_i)$

**$\boldsymbol{w} = A\boldsymbol{x}$ の第 $i$ 成分 $w_i$ は?**

だって〜

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} A_{00}x_0 + A_{01}x_1 + A_{02}x_2 + A_{03}x_3 \\ \cdots \\ \cdots \end{pmatrix}$$

$\cdots$ のところを埋めてみよう

## のプログラム例

```c
int  nrow=3,  mcolumn=4;
double  a[3][4]  ={{2,3,4,5},
                   {3,2,3,4},
                   {0,2,0,4}};        /* 3行4列の行列 */
double  x[4]={1,2,1,3};               /* 4次元ベクトル */
double  w[3];
int  i,j,k;

for(i=0;i<nrow;i++){
     w[i]=0.0;
     for(k=0;k<mcolumn;k++){
          w[i]=w[i]+a[i][k]*x[k];
     }
}
```

$n \times n$ 正方行列 $A = (A_{ij}), B = (B_{ij}), C = (C_{ij})$.

**$C = AB$ の $(i,j)$ 成分 $C_{ij}$ は?**

$$\begin{pmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{pmatrix}$$

$$= \begin{pmatrix} A_{00}B_{00} + A_{01}B_{10} + A_{02}B_{20} & A_{00}B_{01} + A_{01}B_{11} + A_{02}B_{21} & \cdots \\ A_{10}B_{00} + A_{11}B_{10} + A_{12}B_{21} & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

$\cdots$ のところを埋めてみよう

## のプログラム例

```
double a[3][3] ={{2,8,4},
                 {2,0,9}
                 {8,8,4}};        /* 3行3列の行列 */
double b[3][3] ={{3,3,4},
                 {2,9,4}
                 {6,3,4}};
double c[3][3];
double w[3];
int i,j,k;
int n=3;

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        c[i][j]=0.0;
        for(k=0;k<n;k++){
            c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }
    }
}
```

# 1 次元,2 次元配列を引数にとる関数

```
double f(double v[]){
  int i,j;
  for(i=0;i<5;i++){
      v[i]=...;
  }
  return 0.0;
}


double g(double x[][5]){   /*サイズを省略していいのは最初のインデックスだけ */
  int i,j;
  for(i=0;i<5;i++){
     for(j=0;j<5;j++){
        x[i][j]=...;
     }
  }
  return 0.0;
}
```

## 配列と関数にまつわる, 重要な情報

- プログラミング・演習
- したがって, f(v) で呼び出すと, 配列 v の中身 (成分) が変更されて main に返ってくる. f の中での v[i] の変更が外に影響する.

## 配列と関数にまつわる, 今回の演習では不要な情報

- (上のことがあるので必要性は高くないが) 配列を返り値にすることは ‘できない.’
- 配列を引数として与えただけではサイズは関数に伝わらない. サイズが可変な場合, サイズも引数として与える, グローバル変数にするなど.

# (復習) 非線形方程式を解くための反復法

$f(x) = 0 \Leftrightarrow x = g(x)$

$A \neq 0$ は調節する定数.

漸化式 $x_{n+1} = g(x_n)$ で定まる数列 $\{x_n\}$ が $x = x_*$ に収束
$\Longrightarrow f(x_*) = 0.$

L06, プチテスト

# 連立 1 次方程式を解くための反復法

栗原 §3.4 をもっと単純にしたもの
$\boldsymbol{x}, \boldsymbol{b}$: $m$ 次元ベクトル, $M$: $m \times m$ 行列 ($n$ は別の用途).
未知数ベクトル $\boldsymbol{x}$ の満たす 1 次方程式

$$M\boldsymbol{x} = \boldsymbol{b}$$

を考える. $\boldsymbol{f}(\boldsymbol{x}) = M\boldsymbol{x} - \boldsymbol{b}$ とおくと, この方程式は, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$.
非線形方程式の反復法と同様ののりで,

$$\begin{aligned} \boldsymbol{x} =& \boldsymbol{x} - \frac{1}{A}\boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{x} =& \boldsymbol{g}(\boldsymbol{x}) \end{aligned}$$

と書きかえられる

($E$ は単位行列, $A$ はスカラー).
ベクトルの漸化式

$$\boldsymbol{x}_{n+1} = \boldsymbol{g}(\boldsymbol{x}_n)$$

の定めるベクトルの列 $\boldsymbol{x}_n$ が $\boldsymbol{x}_*$ に収束 $\Longrightarrow$ $\boldsymbol{x}_*$ は 1 次方程式の解

# いつ収束するの?

非線形方程式の場合 $f'(x_*)$ と関係していた. 今回は 1 次方程式なので, 条件がもう少し詳しくわかる.
$\boldsymbol{x}_*$ に収束する場合,

$$\boldsymbol{x}_{n+1} - \boldsymbol{x}_* = (E - \tfrac{1}{A}M)(\boldsymbol{x}_n - \boldsymbol{x}_*)$$

すなわち一般項は

$$\boldsymbol{x}_n - \boldsymbol{x}_* = (E - \tfrac{1}{A}M)^n(\boldsymbol{x}_0 - \boldsymbol{x}_*)$$

よって, 行列 $E - \frac{1}{A}M$ のすべての　　　　　の絶対値が 1 未満になるようにスカラー $A$ が調節できれば, 初期値 $\boldsymbol{x}_0$ によらずに $\boldsymbol{x}_*$ に収束する.
だって, $(E - \frac{1}{A}M)^n = P^{-1}\mathrm{diag}(\lambda_1^n, \ldots, \lambda_m^n)P$.
大注意: そういう $A$ が取れないような行列 $M$ もある. というかそのほうが普通. 例:⋯

## in 疑似コード

- 配列 x[] に適当な初期値で初期化
- 以下反復
  - x を更新する (x に g(x) を代入する)
  - | 誤差 $|^2 =$ | 今回の x − 前回の x$|^2$ が定数より小さければ break (反復を抜け出す)
- x を出力

## 注意

- わざとはっきり書いてないけど, Newton 法の収束のときみたいに, 変数として がいるよね.
- g(x) の計算のところが行列やベクトルの積だよね.
- 大注意 $\boldsymbol{x}_n$ の $n$ は, ベクトルのインデックスでなく, 数列 (ベクトル列) の添え字. したがってプログラム中には現れない.

# 来週の Quiz 計画!

来週は予習 (復習?) 問題. 次の連立 1 次方程式を, (拡大係数行列だけの操作で)Gauss-Jordan の消去法 (前進消去+後退消去) で解け. 線形代数 II

$$
\begin{aligned}
2x+3y+z&=1\\
x\quad\;\; +z&=3\\
x-y\quad\;\; &=2
\end{aligned}
$$

の類似問題.

**お知らせ**

- **演習** e ラーニングシステムの評価ページの合計にはこれまで意味のない点数やパーセントが表示されましたが, 現在は, 科目の点数 100 点のうち, これまでに獲得した点数を表示するようにしました.
- **演習** 2010-07-17 土 2 の補講は自由参加. この日 12:15 が E11,E12,E13 の課題の提出の最終チャンス (病気, 公務欠席などによる延長はありません)
- **講義** レポート R11 課題 → L11 資料参照

# ファイナルトライアル計画!

科目の成績 100 点中 50 点. 外部記憶ペーパー使用可能. 別紙参照.
準備としては, まずは各回の quiz を確実に解けるようになること, 演習課題を心から理解することをお奨めします. 模範解答を作ろうプロジェクトはない予定.
現在のところの出題計画 (2010-07-16 に更新されます)

- 台形公式で数値積分しよう!
- シンプソン公式で数値積分しよう!
- データの平均, 分散, 標準偏差を求めよう!
- 2 変量データの相関係数を求めよう!
- 2 分法で非線形方程式を解こう!
- 反復法で連立 1 次方程式や非線形方程式を解こう!(一部プチテスト範囲再出題)
- 連立 1 次方程式を Gauss-Jordan の消去法計算機バージョンで解こう!
- 行列やベクトルを成分で計算するプログラムを書こう!
- ニュートン法のプログラムを書こう!(プチテスト範囲再出題)
- 漸化式で定義される数列 $a_k$ について $\sum_{k=0}^{n} a_k$ を求めるプログラムを書こう!(プチテスト範囲再出題)