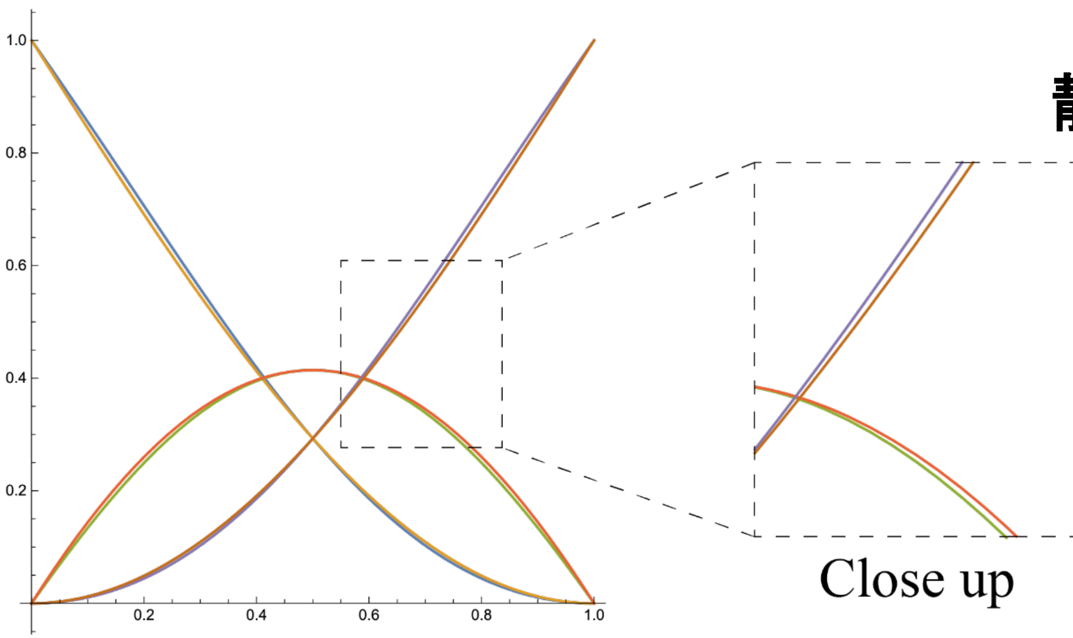
数値解析

2023年度前期　第10週 [6月22日]

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

三浦 憲二郎



2次有理Bernstein基底 ($\sigma = 1/\sqrt{2}$) vs 1次一般化三角関数基底

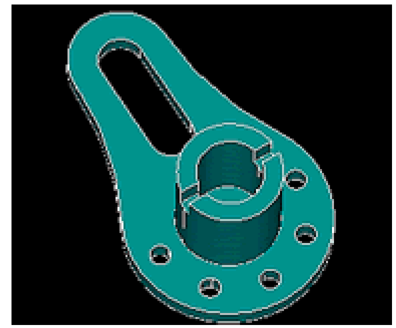
講義アウトライン [6月15日]

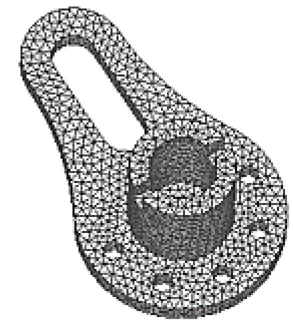
- 連立1次方程式の反復解法
    - 前回の復習
    - 共役勾配法

連立一次方程式の反復解法 p.93

- 連立1次方程式

  $$Ax = b$$



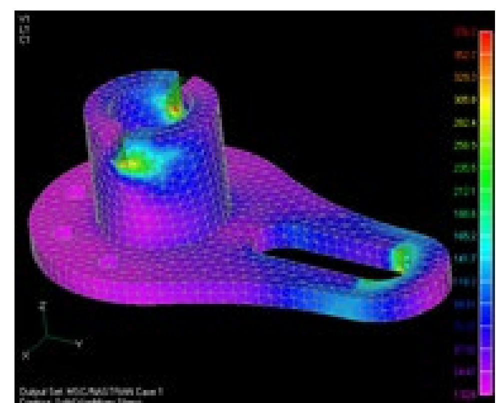


- 密(dense)行列と疎（sparse）行列

  密行列：行列の要素のすべて，あるいはほとんどゼロでない行列

  疎行列：行列の要素のほとんどがゼロである行列

- 疎行列
  - 反復解法を用いるほうが速い．



- ヤコビ法とガウスザイデル法
- 共役勾配法

反復法の原理 p.93

連立1次方程式

$$\boldsymbol{Ax} = \boldsymbol{b} \quad (5.1)$$

を同値な方程式

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

に変形し，反復法により解を求める．$\boldsymbol{M}$を反復行列という．

↓

縮小写像の原理

縮小写像の原理　定理5.1　p.94

R^n 上で定義された関数$g(x)$ が

(1) $x \in R^n \to g(x) \in R^n$

(2) $x, y \in R^n \to ||g(x) - g(y)|| \leq L||x - y|| \in R^n$

(3) $0 \leq L < 1$

を満たすとき，方程式

$$x = g(x)$$

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

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

によって，生成される列$\{x^{(k)}\}$は$k \to \infty$のときαに収束する．

縮小写像の原理　定理5.2　p.94

あるノルム‖・‖について$||\mathbf{M}|| < \mathbf{1}$ならば反復法

$$x^{(k+1)} = \phi\left(x^{(k)}\right) = Mx^{(k)} + Nb$$

によって作られる列$\{x^{(k)}\}$は(5.1)の解xに収束する

(証明)

任意の$x, y \in R^n$に対して,

$$||\phi(x) - \phi(y)|| = ||Mx - My|| \leq ||M|| \cdot ||x - y|| < ||x - y||$$

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

行列Aの分解　p.95

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

$$\boldsymbol{D} = \begin{bmatrix} a_{11} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & a_{nn} \end{bmatrix}$$

$$\boldsymbol{L} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} \mathbf{0} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

ヤコビ法とガウス･ザイデル法

- **ヤコビ法**

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

とおいた反復法

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

- **ガウス･ザイデル法**

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

とおいた反復法

$$x^{(k+1)} = D^{-1}\{b - Lx^{(k+1)} - Ux^{(k)}\}$$

共役勾配法　p.109

行列Aを対称正定値行列と仮定

正定値行列 p.57

n次正方行列Aが，　$x \neq 0$である任意のベクトル$x \in R^n$

$$(x, Ax) > 0.$$

- Aが対称でない場合

$$A^t A x = A^t b$$

とすれば共役勾配法を使える.

共役勾配法　p.109

定理5.7 Aをn次対称正定値行列とする．このとき，連立一次方程式

$$Ax = b$$

の解$\tilde{x}$は関数

$$f(x) = \frac{1}{2}(x, Ax) - (x, b)$$

を最小にする．ここで，　$(\cdot,\cdot)$はベクトルの内積を表す．

共役勾配法 p.109

定理5.7の証明

$\forall \boldsymbol{h} \in \boldsymbol{R}^n(\boldsymbol{h} \neq \boldsymbol{0})$に対して

$$\boldsymbol{f}(\widetilde{\boldsymbol{x}}+\boldsymbol{h}) = \frac{1}{2}(\widetilde{\boldsymbol{x}}+\boldsymbol{h}, \boldsymbol{A}(\widetilde{\boldsymbol{x}}+\boldsymbol{h})) - (\widetilde{\boldsymbol{x}}+\boldsymbol{h}, \boldsymbol{b})$$

$$= \frac{1}{2}\{(\widetilde{\boldsymbol{x}}, \boldsymbol{A}\widetilde{\boldsymbol{x}}) + (\widetilde{\boldsymbol{x}}, \boldsymbol{A}\boldsymbol{h}) + (\boldsymbol{h}, \boldsymbol{A}\widetilde{\boldsymbol{x}}) + (\boldsymbol{h}, \boldsymbol{A}\boldsymbol{h})\} - (\widetilde{\boldsymbol{x}}, \boldsymbol{b}) - (\boldsymbol{h}, \boldsymbol{b})$$

$$= \boldsymbol{f}(\widetilde{\boldsymbol{x}}) + \frac{1}{2}\{(\widetilde{\boldsymbol{x}}, \boldsymbol{A}\boldsymbol{h}) + (\boldsymbol{h}, \boldsymbol{A}\widetilde{\boldsymbol{x}}) + (\boldsymbol{h}, \boldsymbol{A}\boldsymbol{h})\} - (\boldsymbol{h}, \boldsymbol{b})$$

ここで，$\boldsymbol{A}$は対称なので， $(\widetilde{\boldsymbol{x}}, \boldsymbol{A}\boldsymbol{h}) = (\boldsymbol{A}\widetilde{\boldsymbol{x}}, \boldsymbol{h}) = (\boldsymbol{h}, \boldsymbol{A}\widetilde{\boldsymbol{x}})$

よって,

$$\boldsymbol{f}(\widetilde{\boldsymbol{x}}+\boldsymbol{h}) = \boldsymbol{f}(\widetilde{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{h}, \boldsymbol{A}\boldsymbol{h}) + (\boldsymbol{h}, \boldsymbol{A}\widetilde{\boldsymbol{x}}) - (\boldsymbol{h}, \boldsymbol{b}) = \boldsymbol{f}(\widetilde{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{h}, \boldsymbol{A}\boldsymbol{h})$$

$\boldsymbol{A}$の正定値性より$(\boldsymbol{h}, \boldsymbol{A}\boldsymbol{h}) > \boldsymbol{0}$なので,

$$\boldsymbol{f}(\widetilde{\boldsymbol{x}}+\boldsymbol{h}) - \boldsymbol{f}(\widetilde{\boldsymbol{x}}) > \boldsymbol{0}, \text{ つまり } \boldsymbol{f}(\widetilde{\boldsymbol{x}}+\boldsymbol{h}) > \boldsymbol{f}(\widetilde{\boldsymbol{x}})$$

共役勾配法　p.109,110

共役勾配法の原理

ベクトル $\boldsymbol{p}_i$ と $\boldsymbol{p}_j$ が

$$(\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_j) = \boldsymbol{0}, \qquad \boldsymbol{i} \neq \boldsymbol{j}$$

を満たすとき，ベクトル $\boldsymbol{p}_i$ と $\boldsymbol{p}_j$は行列Aに関して共役である，あるいはAに関して直交である，という.

$\boldsymbol{f}(\boldsymbol{x})$を最小にする$\tilde{x}$を求める

⇩

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\alpha}_k \boldsymbol{p}_k, \qquad (\boldsymbol{k} = \boldsymbol{0}, \boldsymbol{1}, \boldsymbol{2}, \dots)$$

$\boldsymbol{f}(\boldsymbol{x}_{k+1})$は　$\boldsymbol{\alpha}_k = \dfrac{(\boldsymbol{p}_k, \boldsymbol{r}_k)}{(\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}$ のとき最小となる.

ただし，誤差ベクトル$\boldsymbol{r}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_k + \boldsymbol{\alpha}_k \boldsymbol{p}_k) = \boldsymbol{r}_k - \boldsymbol{\alpha}_k \boldsymbol{A}\boldsymbol{p}_k$

共役勾配法　p.110

探索方向$\boldsymbol{p}_{k+1}$は

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \boldsymbol{\beta}_k \boldsymbol{p}_k$$

で定める.

$$\boldsymbol{\beta}_k = -\frac{(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}$$

共役勾配法　Conjugate Gradient method: CG法

定理5.8 p.114　共役勾配法によって計算される誤差ベクトル$\boldsymbol{r}_k$および方向ベクトル$\boldsymbol{p}_k$について，　$\boldsymbol{i} \neq \boldsymbol{j}$であれば，　$(\boldsymbol{r}_i, \boldsymbol{r}_j) = \boldsymbol{0}, (\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_j) = \boldsymbol{0}$.

定理5.9 p.115 Aは正則なn次対称正定値行列で，$\boldsymbol{b}$ はn次元ベクトルとする.

このとき，共役勾配法の反復をn回実行するするると$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$の解が得られる！

共役勾配法:プログラム その1

```
/* 1ノルムの計算 */
double vector_norm1(double* a, int m, int n)
{
    int i;
    double norm = 0.0;
    for (i = m; i <= n; i++){
        norm += fabs(a[i]);
    }
    return norm;
}
/* Aベクトル a[m...n]とb[m...n]の内積を計算する */
double inner_product(int m, int n, double* a, double* b)
{
    int i;
    double s = 0.0;
    for (i = m; i <= n; i++) s += a[i] * b[i];
    return s;
}
/* 行列a[1...N][1...]とベクトルb[1...N]との積 c <- Ab */
void matrix_vector_product(double** a, double* b, double* c)
{
    double wk;
    int i, j;
    for (i = 1; i <= N; i++){
        wk = 0.0;
        for (j = 1; j <= N; j++){
            wk += a[i][j] * b[j];
        }
        c[i] = wk;
    }
}
```

共役勾配法：プログラム その2

```
/* 共役勾配法 */
double* cg(double** a, double* b, double* x)
{
	double eps, * r, * p, * tmp, alpha, beta, work;
	int i, k = 0;
	r = dvector(1, N); /* r[1...N] */
	p = dvector(1, N); /* p[1...N] */
	tmp = dvector(1, N); /* tmp[1...N] */
	matrix_vector_product(a, x, tmp); /* tmp <- A b */
	for( i = 1; i <= N; i++){
		p[i] = b[i] - tmp[i]; r[i] = p[i];
	}
	do{	/* alpha の計算 */
		matrix_vector_product(a, p, tmp); /* tmp <- A p_k */
		work = inner_product(1, N, p, tmp); /* work <- (p,Ap_k)*/
		alpha = inner_product(1, N, p, r) / work;
		/* x_{k+1} と r_{k+1}の計算 */
		for (i = 1; i <= N; i++) x[i] = x[i] + alpha * p[i];
		for (i = 1; i <= N; i++) r[i] = r[i] - alpha * tmp[i];
		/* 収束判定 */
		eps = vector_norm1(r, 1, N);
		k++; /* 反復回数の更新 */
		if (eps < EPS) goto OUTPUT;
		/* beta と p_{k+1}の計算 */
		beta = -inner_product(1, N, r, tmp) / work;
		for (i = 1; i <= N; i++) p[i] = r[i] + beta * p[i];
	}while (k < KMAX);
OUTPUT:;
	/* 領域の解放 */
	free_dvector(r, 1); free_dvector(p, 1); free_dvector(tmp, 1);
	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です¥n", k); /* 反復回数を画面に表示 */
		return x;
	}
}
```

まとめ

- 連立1次方程式の反復解法
  - 前回の復習
  - 共役勾配法