数理アルゴリズム　演習課題 4　提出日：2017 年 10 月 26 日
201611350 江畑 拓哉

1　課題 1

1.1

行列を A を正定値行列対称行列とし、$\boldsymbol{x}x^\star$ を $A\boldsymbol{x} = \boldsymbol{b}$ の解とする。関数 $f(x)$ を

$$f(\boldsymbol{x}) = 1/2(\boldsymbol{x}, A\boldsymbol{x}) - (\boldsymbol{x}, \boldsymbol{b})$$

として、以下の課題を行うこと。

1.1.1　ベクトル $\boldsymbol{h}$ は零ベクトルでないとする。このとき、以下の不等式を示せ。

$$f(\boldsymbol{x} + \boldsymbol{h}) > f(\boldsymbol{x}^\star)$$

$$\begin{aligned}
&f(\boldsymbol{x} + \boldsymbol{h}) - f(\boldsymbol{x}^\star) \\
&\quad = 1/2(\boldsymbol{x}^\star + \boldsymbol{h}, A(\boldsymbol{x}^\star + \boldsymbol{h})) - (\boldsymbol{x}^\star + \boldsymbol{h}, \boldsymbol{b}) - 1/2(\boldsymbol{x}^\star, A\boldsymbol{x}^\star) + (\boldsymbol{x}^\star, \boldsymbol{b}) \\
&\quad = 1/2(\boldsymbol{x}^\star, A\boldsymbol{h}) + 1/2(\boldsymbol{h}, A\boldsymbol{x}^\star) + 1/2(\boldsymbol{h}, A\boldsymbol{h}) - (\boldsymbol{h}, A\boldsymbol{x}^\star) \\
&\quad = 1/2(\boldsymbol{h}, A\boldsymbol{x}^\star) + 1/2(\boldsymbol{h}, A\boldsymbol{x}^\star) + 1/2(\boldsymbol{h}, A\boldsymbol{h}) - (\boldsymbol{h}, A\boldsymbol{x}^\star) \\
&\quad = 1/2(\boldsymbol{h}, A\boldsymbol{h}) > 0
\end{aligned}$$

よって

$$f(\boldsymbol{x} + \boldsymbol{h}) > f(\boldsymbol{x}^\star)$$

1.1.2　ベクトル $\boldsymbol{x}_k + 1$ は $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$ で与えられるとする。このとき

$$f(\boldsymbol{x}_{k+1}) = 1/2\alpha_k^2(\boldsymbol{p}_k, A\boldsymbol{p}_k) - \alpha(\boldsymbol{p}_k, \boldsymbol{r}_k) + f(\boldsymbol{x}_k)$$

となることを示せ。ここで、$\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ である。

$$
\begin{aligned}
&f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \\
&\quad = f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) - f(\boldsymbol{x}_k) \\
&\quad = 1/2(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, A(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)) - (\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \boldsymbol{b}) - 1/2(\boldsymbol{x}_k, A\boldsymbol{x}_k) + (\boldsymbol{x}_k, \boldsymbol{b}) \\
&\quad = 1/2\alpha_k(\boldsymbol{x}_k, A\boldsymbol{p}_k) + 1/2\alpha_k(\boldsymbol{p}_k, A\boldsymbol{x}_k) + 1/2\alpha_k^2(\boldsymbol{p}_k, A\boldsymbol{p}_k) - \alpha_k(\boldsymbol{p}_k, \boldsymbol{r}_k + A\boldsymbol{x}_k) \\
&\quad = 1/2\alpha_k(\boldsymbol{p}_k, A\boldsymbol{x}_k) + 1/2\alpha_k(\boldsymbol{p}_k, A\boldsymbol{x}_k) + 1/2\alpha_k^2(\boldsymbol{p}_k, A\boldsymbol{p}_k) - \alpha_k(\boldsymbol{p}_k, \boldsymbol{r}_k) - \alpha_k(\boldsymbol{p}_k, A\boldsymbol{x}_k) \\
&\quad = 1/2\alpha_k^2(\boldsymbol{p}_k, A\boldsymbol{p}_k) - \alpha_k(\boldsymbol{p}_k, \boldsymbol{r}_k)
\end{aligned}
$$

よって

$$f(\boldsymbol{x}_{k+1}) = 1/2\alpha_k^2(\boldsymbol{p}_k, a\boldsymbol{p}_k) - \alpha(\boldsymbol{p}_k, \boldsymbol{r}_k) + f(\boldsymbol{x}_k)$$

1.2　CG 法は以下の漸化式で構成される. このとき, 行列 A は正定値対称行列とする.

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k \\
\boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k \\
\boldsymbol{p}_{k+1} &= \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k
\end{aligned} \tag{1}
$$

1.2.1　以下の関数を最小化するように係数 α_k を設定する. α_k の導出過程を示せ.

$$f(\boldsymbol{x}_{k+1}) = 1/2\alpha_k^2(\boldsymbol{p}_k, A\boldsymbol{p}_k) - \alpha_k(\boldsymbol{p}_k, \boldsymbol{r}_k) + f(\boldsymbol{x}_k)$$

両辺を微分して、

$$\frac{\partial}{\partial \alpha_k} f(\boldsymbol{x}_k) = \alpha_k(\boldsymbol{p}_k, A\boldsymbol{p}_k) - (\boldsymbol{p}_k, \boldsymbol{r}_k)$$

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

1.2.2　式 (1) から β_k を求めたい.

$\boldsymbol{p}_i$ と $\boldsymbol{p}_j$ は互いに共役であるという条件から, β_k を導出の過程を示せ.

$$
\begin{aligned}
(\boldsymbol{p}_{k+1}, A\boldsymbol{p}_k) &= (\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k, A\boldsymbol{p}_k) \\
&= (\boldsymbol{r}_{k+1}, A\boldsymbol{p}_k) + \beta_k(\boldsymbol{p}_k, A\boldsymbol{p}_k) \\
&= 0
\end{aligned}
$$

$$
\beta_k = -\frac{(\boldsymbol{r}_{k+1}, A\boldsymbol{p}_k)}{(\boldsymbol{p}_k, A\boldsymbol{p}_k)}
$$

2　課題 2

ラプラス方程式の差分近似の式から, $n = 3$ のときの連立一次方程式の係数行列 A および右辺ベクトル b を図 1 を参考に作成し, A, b の各要素をレポートに示せ.

$$
\begin{aligned}
4u_{1,1} - (u_{2,1} + u_{0,1} + u_{1,2} + u_{1,0}) &= 0 \\
4u_{1,2} - (u_{2,2} + u_{0,2} + u_{1,3} + u_{1,1}) &= 0 \\
4u_{1,3} - (u_{2,3} + u_{0,3} + u_{1,4} + u_{1,2}) &= 0 \\
4u_{2,1} - (u_{3,1} + u_{1,1} + u_{2,2} + u_{2,0}) &= 0 \\
4u_{2,2} - (u_{3,2} + u_{1,2} + u_{2,3} + u_{2,1}) &= 0 \\
4u_{2,3} - (u_{3,3} + u_{1,3} + u_{2,4} + u_{2,2}) &= 0 \\
4u_{3,1} - (u_{4,1} + u_{2,1} + u_{3,2} + u_{3,0}) &= 0 \\
4u_{3,2} - (u_{4,2} + u_{2,2} + u_{3,3} + u_{3,1}) &= 0 \\
4u_{3,3} - (u_{4,3} + u_{2,3} + u_{3,4} + u_{3,2}) &= 0
\end{aligned}
$$

$$
\begin{aligned}
4u_{1,1} - (u_{2,1} + u_{1,2}) &= u_{0,1} + u_{1,0} \\
4u_{1,2} - (u_{2,2} + u_{1,3} + u_{1,1}) &= u_{0,2} \\
4u_{1,3} - (u_{2,3} + u_{1,4} + u_{1,2}) &= u_{0,3} \\
4u_{2,1} - (u_{3,1} + u_{1,1} + u_{2,2}) &= u_{2,0} \\
4u_{2,2} - (u_{3,2} + u_{1,2} + u_{2,3} + u_{2,1}) &= 0 \\
4u_{2,3} - (u_{3,3} + u_{1,3} + u_{2,2}) &= u_{2,4} \\
4u_{3,1} - (u_{2,1} + u_{3,2}) &= u_{4,1} + u_{3,0} \\
4u_{3,2} - (u_{2,2} + u_{3,3} + u_{3,1}) &= u_{4,2} \\
4u_{3,3} - (u_{2,3} + u_{3,2}) &= u_{4,3} + u_{3,4}
\end{aligned}
$$

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{3,1} \\ u_{3,2} \\ u_{3,3} \end{pmatrix} = \begin{pmatrix} u_{0,1} + u_{1,0} \\ u_{0,2} \\ u_{0,3} \\ u_{2,0} \\ 0 \\ u_{2,4} \\ u_{4,1} + u_{3,0} \\ u_{4,2} \\ u_{4,3} + u_{3,4} \end{pmatrix}$$

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

3　課題 3

3.1　連立一次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を解く CG 法のプログラムを作成し, 解を求めよ.

以下が CG 法を計算するための関数である。

```
function [hatX, R] = cg_function(A, b)
    size_ = size(A)(1);

    x = zeros(size_, 2);
    r = zeros(size_, 2);
    p = zeros(size_, 2);
    a = zeros(1, 2);
    be = zeros(1, 2);

    idx = 1;
    r(:, idx) = b - A * x(:, idx);
    p(:, idx) = r(:, idx);
    R = zeros(1, 2)
    R(:, idx) = 100;
    while R(:, idx) >= 10^-4,
        tmp_ = (A * p(:, idx))
        a(idx) = ((p(:, idx)' * r(:, idx)) / (p(:, idx)' * tmp_));
        x(:, idx + 1) = (x(:, idx) + (a(idx) * p(:, idx)));
        r(:, idx + 1) = (r(:, idx) - (a(idx) * tmp_));
        be(idx) = (-1 * ((r(:, idx + 1)' * tmp_) / (p(:, idx)' * tmp_)))
        p(:, idx + 1) = r(:, idx + 1) + be(idx) * p(:, idx);
```

```
        idx = idx + 1;
        R(:, idx) = norm(r(:, idx));
    end
    hatX = x(:, idx);
    R = R(2:$)
endfunction
```

実行方法は次の課題に合わせて示す。以下に実行結果を示す。

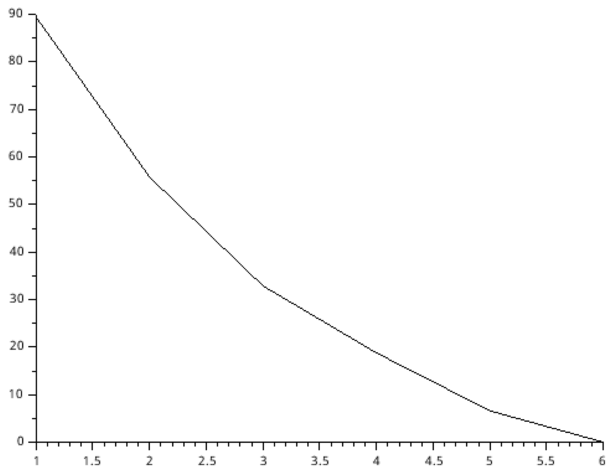

図 1: n_4.mat 線形グラフ

3.2 (3-1) で求めた解 x を n 次正方行列に変換し, その要素の値を Scilab の surf 関数を用いてグラフに描け.

以下が正方行列に変換するための関数である。

```
function [mat] = createSquare(x)
    size_ = size(x)(1)^(1/2);
    mat = zeros(size_, size_);
    idx_ = 1;
    for idx = 1:size_,
      mat(:, idx) = x(idx_:idx*size_, 1)
      idx_ = idx * size_ + 1
    end
endfunction
```

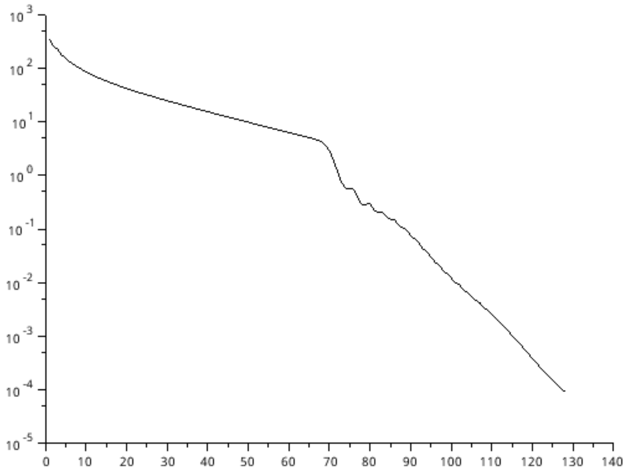

図 2: n_50.mat 縦軸を対数とした片対数グラフ

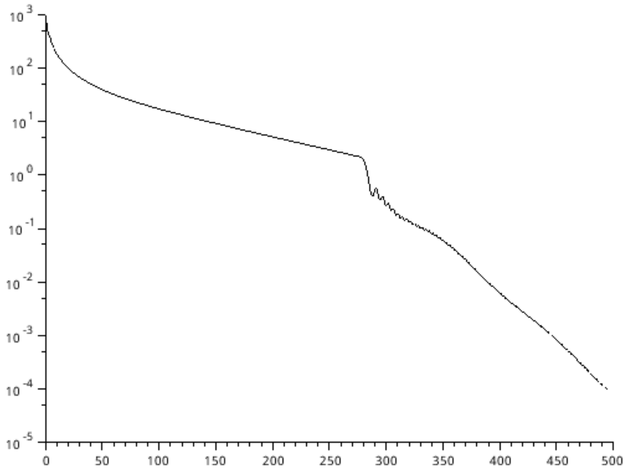

図 3: n_200.mat 縦軸を対数とした片対数グラフ

実行は以下のようになる。

```
// n = 4
loadmatfile('/path/to/n_4.mat')
[X, R] = cg_function(A, b);
plot2d(R)
surf(createSquare(X))

// n = 50
loadmatfile('/path/to/n_50.mat')
[X, R] = cg_function(A, b);
plot2d('nl', R)
surf(createSquare(X))

// n = 200
loadmatfile('/path/to/n_200.mat')
[X, R] = cg_function(A, b);
plot2d('nl', R)
surf(createSquare(X))
```

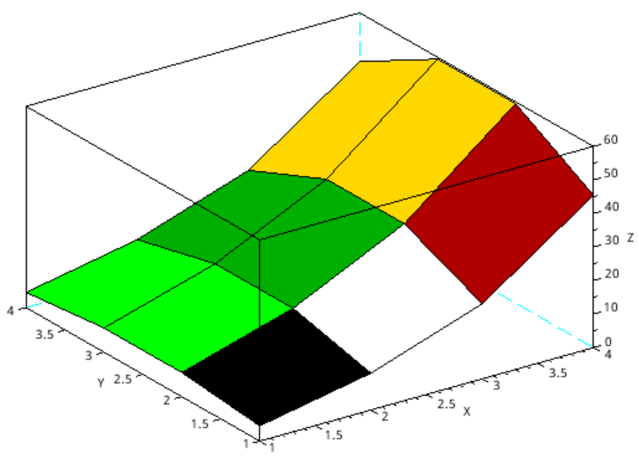

図 4: n_4.mat

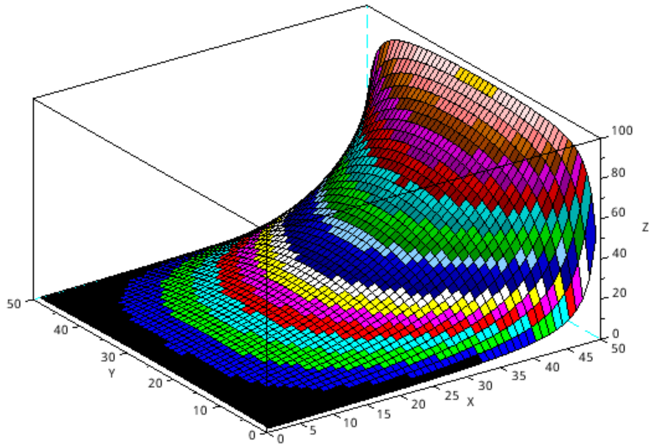

図 5: n_50.mat

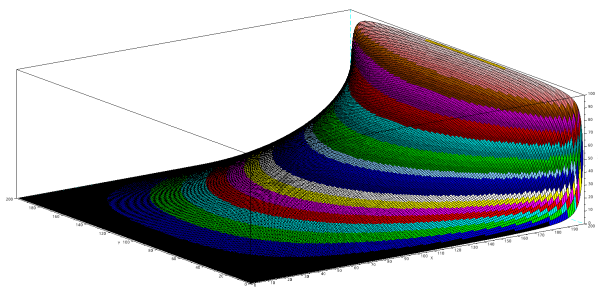

図 6: n_200.mat

4　課題 4

CG 法を用いてソースコード 1 で生成される行列 A と乱数ベクトル b からなる連立一次方程式を解け. このとき, 収束条件は r_k の 2 ノルムが $10^{-}8$ より小さくなった時とする. また, 行列サイズ n を n = 10, 11, . . . , 100 の間で変更し, 収束までの反復回数をグラフに描画せよ.

以下が新しい CG 法を解くための関数である。以前のそれとの変更点は、収束条件が $10^{-}8$ になっている点である。

```
function [hatX, R] = cg_function2(A, b)
    size_ = size(A)(1);

    x = zeros(size_, 2);
    r = zeros(size_, 2);
    p = zeros(size_, 2);
    a = zeros(1, 2);
    be = zeros(1, 2);

    idx = 1;
    r(:, idx) = b - A * x(:, idx);
    p(:, idx) = r(:, idx);
    R = zeros(1, 2)
    R(:, idx) = 100;
    while R(:, idx) >= 10^-8,
        tmp_ = (A * p(:, idx))
        a(idx) = ((p(:, idx)' * r(:, idx)) / (p(:, idx)' * tmp_));
        x(:, idx + 1) = (x(:, idx) + (a(idx) * p(:, idx)));
        r(:, idx + 1) = (r(:, idx) - (a(idx) * tmp_));
        be(idx) = (-1 * ((r(:, idx + 1)' * tmp_) / (p(:, idx)' * tmp_)))
        p(:, idx + 1) = r(:, idx + 1) + be(idx) * p(:, idx);
        idx = idx + 1;
        R(:, idx) = norm(r(:, idx));
    end
    hatX = x(:, idx);
    R = R(2:$)
endfunction
```

以下が実行である。

```
l = zeros(1, 1);
idx = 1
for n = 10:100,
    A = rand(n,n);
    A = (A + A')/2;
    A = A + 5* eye(n,n);
    b = rand(n,1);
    [X, R] = cg_function2(A, b);
    l(:, idx) = size(R)(2);
    idx = idx + 1;
end
plot2d(linspace(10,  100, 91), l)
```

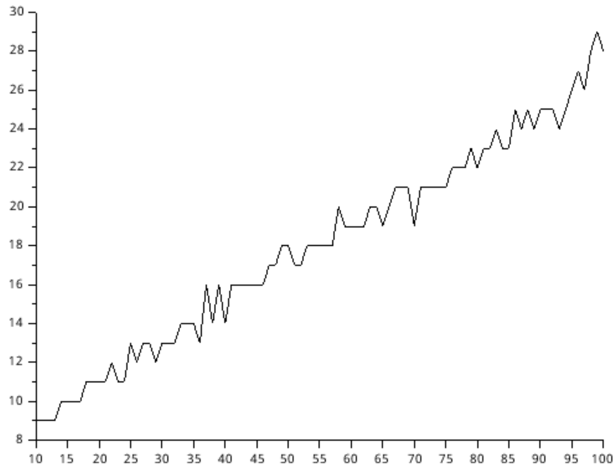

図 7: 収束までの反復回数