数理アルゴリズムとシミュレーション　演習課題5　 提出日：2020年11月12日
201811319 永崎遼太

課題2

$\lambda_i \neq \lambda_j$ を固有値とし、q_i および q_j を対応する固有ベクトルとすると、

$$A q_i = \lambda_i q_i \ , \ A q_j = \lambda_j q_j$$

2式それぞれにおいて以下が成り立つ。

$$q_j^T A q_i = \lambda_i q_j^T q_i$$

$$q_i^T A q_j = \lambda_j q_i^T q_j$$

他方、

$$(q_j^T A q_i)^T = q_i^T A^T q_j = q_i^T A q_j$$

より、

$$\lambda_j q_i^T q_j = \lambda_i q_i^T q_j$$

したがって

$$(\lambda_j - \lambda_i) q_i^T q_j = 0$$

から、

$$q_i^T q_j = 0$$

よって、実対称行列の相異なる固有値の固有ベクトルは直交する。

課題3
(3-1)

次のプログラムを作成した。

ーーーーー

```
A = [ 2 1 0;
     1 2 1;
     0 1 2];
x = [ 1; 1; 1];
maxit = 2000;
tol = 1e-05;
for it = 1:maxit
   r = 0;
   w = A*x;
   x = w / norm(w);
   l1 = dot(x, A*x);
   l2 = dot(x,x);
   lambda = l1/l2;
   r = norm(A*x - lambda*x);
   if r < tol
      break;
   end
   T_(it)=r;
   t(it)=it;
end
lambda
x
```

ーーーーー

これを実行すると、次の結果が得られる。

ーーーーー

```
lambda =

   3.4142

x =

   0.5000

   0.7071

   0.5000
```

ーーーーー

絶対値最大固有値とそれに対応する固有ベクトルを求めることができている。

(3-2)

次のプログラムを作成した。

ーーーーー

```
n = 100;
x3 = 3 * ones(n, 1);
x2 = 2 * ones(n-1, 1);
x1 = ones(n-2,1);
A3 = diag(x3);
A2_1 = diag(x2, -1);
A2_2 = diag(x2, 1);
A1_1 = diag(x1, -2);
A1_2 = diag(x1, 2);
A = A3+A2_1+A2_2+A1_1+A1_2;

x = ones(n, 1);
```

```
maxit = 2000;
tol = 1e-05;

for it = 1:maxit
    r = 0;
    w = A*x;
    x = w / norm(w);
    l1 = dot(x, A*x);
    l2 = dot(x,x);
    lambda = l1/l2;
    r = norm(A*x - lambda*x);
    if r < tol
        break;
    end
end
lambda
```
ーーーーー

これを実行すると次の結果が得られる。

ーーーーー

```
lambda =
    8.9942
```
ーーーーー

これより、絶対値最大固有値を求めることができた。

課題4

(4-1)

次のプログラムを作成した。

ーーーーー

```
n = 100;

x3 = 3 * ones(n, 1);

x2 = 2 * ones(n-1, 1);

x1 = ones(n-2,1);

A3 = diag(x3);

A2_1 = diag(x2, -1);

A2_2 = diag(x2, 1);

A1_1 = diag(x1, -2);

A1_2 = diag(x1, 2);

A = A3+A2_1+A2_2+A1_1+A1_2;

A_ = inv(A);

x = ones(n, 1);

maxit = 2000;

tol = 1e-05;

for it = 1:maxit

    r = 0;

    w = A\x;

    x = w / norm(w);

    l1 = dot(x, x);

    l2 = dot(x,A\x);

    lambda = l1/l2;

    r = norm(A*x - lambda*x);

    if r < tol

        break;
```

```
    end
    T_(it)=r;
    t(it)=it;
end
x
lambda
semilogy(t,T_)
xlabel('Number of Iterations')
ylabel('r')
```

ーーーーー

これを実行すると、次の結果が得られた。

ーーーーー

```
x =
   0.0091
  -0.0089
  -0.0093
   0.0273
  -0.0178
  -0.0186
   0.0453
  -0.0266
  -0.0276
   0.0629
  -0.0351
  -0.0364
   0.0800
  -0.0434
  -0.0449
   0.0963
  -0.0513
  -0.0529
```

0.1119

-0.0588

-0.0605

0.1265

-0.0659

-0.0675

0.1400

-0.0724

-0.0740

0.1524

-0.0782

-0.0798

0.1634

-0.0835

-0.0849

0.1730

-0.0880

-0.0892

0.1812

-0.0918

-0.0928

0.1878

-0.0948

-0.0956

0.1928

-0.0971

-0.0976

0.1961

-0.0985

-0.0987

0.1978

-0.0990

-0.0990

0.1978

-0.0987

-0.0985

0.1961

-0.0976

-0.0971

0.1928

-0.0956

-0.0948

0.1878

-0.0928

-0.0918

0.1812

-0.0892

-0.0880

0.1730

-0.0849

-0.0835

0.1634

-0.0798

-0.0782

0.1524

-0.0740

-0.0724

0.1400

-0.0675

-0.0659

0.1265

-0.0605

-0.0588

0.1119

-0.0529

-0.0513

0.0963

-0.0449

-0.0434

0.0800

-0.0364

-0.0351

0.0629

-0.0276

-0.0266

0.0453

-0.0186

-0.0178

0.0273

-0.0093

-0.0089

0.0091

lambda =

0.0028

ーーーーー

Aの絶対値最小の固有値とそれに対応する固有ベクトルを求めることができている。

また、図1の反復ごとの残差の値のグラフが得られた。

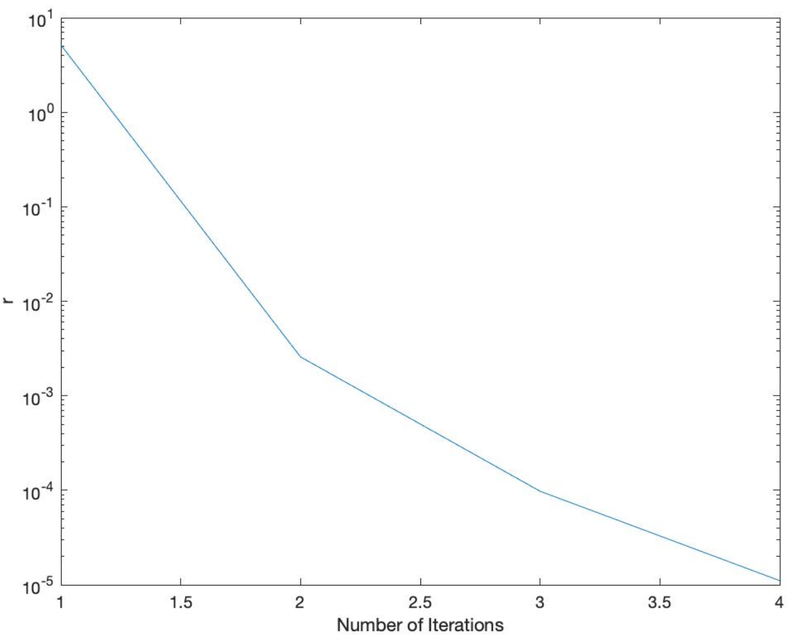

図1 反復ごとの残差

(4-2)

次のプログラムを作成した。

ーーーーー

```
n = 100;
x3 = 3 * ones(n, 1);
x2 = 2 * ones(n-1, 1);
x1 = ones(n-2,1);
A3 = diag(x3);
A2_1 = diag(x2, -1);
A2_2 = diag(x2, 1);
A1_1 = diag(x1, -2);
A1_2 = diag(x1, 2);
A = A3+A2_1+A2_2+A1_1+A1_2;
x = ones(n, 1);

interval = 0.01;
x2 = 3;
x1 = 2;
q = (x2-x1)/ interval +1;
sigma = linspace(2, 3, q);
I_ = ones(1, n);
I = diag(I_);

maxit = 2000;
tol = 1e-05;

for i = 1:numel(sigma)
    shift = sigma(i);
```

```
    x = ones(100, 1);
    for it = 1:maxit
        r = 0;
        w = (shift*I-A)\x;
        x = w / norm(w);
        l1 = dot(x, x);
        l2 = dot(x,(shift*I-A)\x);
        lambda = shift -l1/l2;
        r = norm(A*x - lambda*x);
        if r < tol
            break;
        end
        t = it;
    end
    s(i) = shift;
    c(i) = t;
end

plot(s,c)
xlabel('Value of Shift')
ylabel('Number of Iterations')
```

ーーーーー

これを実行すると、図2が得られる。

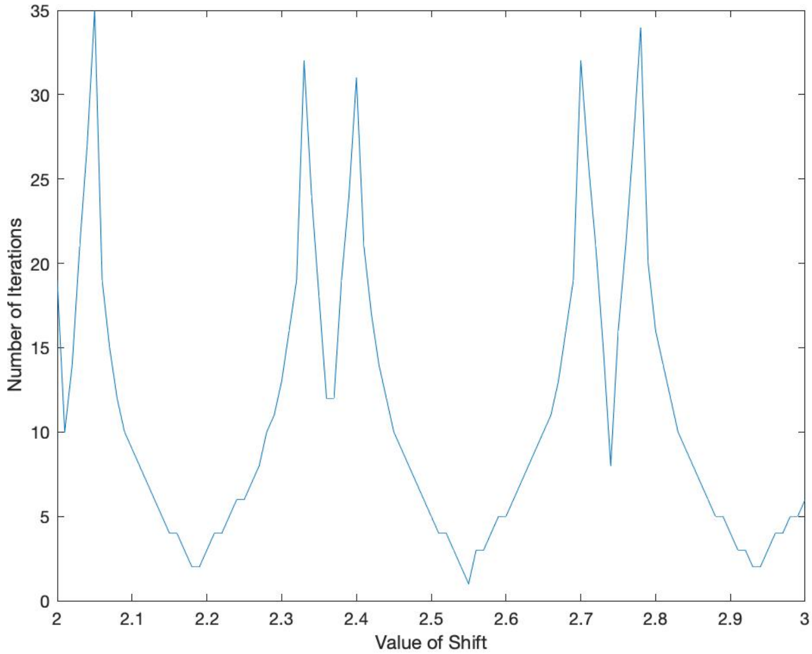

図2 シフトの値ごとの反復回数

それそれぞれのシフトでの反復回数を図示することができている。

(4-3)

次のプログラムを作成した。

ーーーーー

n = 100;

x3 = 3 * ones(n, 1);

x2 = 2 * ones(n-1, 1);

x1 = ones(n-2,1);

A3 = diag(x3);

A2_1 = diag(x2, -1);

A2_2 = diag(x2, 1);

A1_1 = diag(x1, -2);

A1_2 = diag(x1, 2);

```
A = A3+A2_1+A2_2+A1_1+A1_2;
x = ones(n, 1);
interval = 0.01;
x2 = 3;
x1 = 2;
q = (x2-x1)/ interval +1;
sigma = linspace(2, 3, q);
I_ = ones(1, n);
I = diag(I_);


maxit = 2000;
tol = 1e-05;


for i = 1:numel(sigma)
    shift = sigma(i);
    x = ones(100, 1);
    for it = 1:maxit
        r = 0;
        w = (shift*I-A)\x;
        x = w / norm(w);
        l1 = dot(x, x);
        l2 = dot(x,(shift*I-A)\x);
        lambda = shift -l1/l2;
        r = norm(A*x - lambda*x);
        if r < tol
            break;
        end
        t = it;
```

```
    end
    s(i) = shift;
    l(i) = lambda;
end
plot(s,l)
xlabel('Value of shift')
ylabel('Value of Lambda')
% semilogy(t,T_)
```

ーーーーー

これを実行すると、図3が得られる。

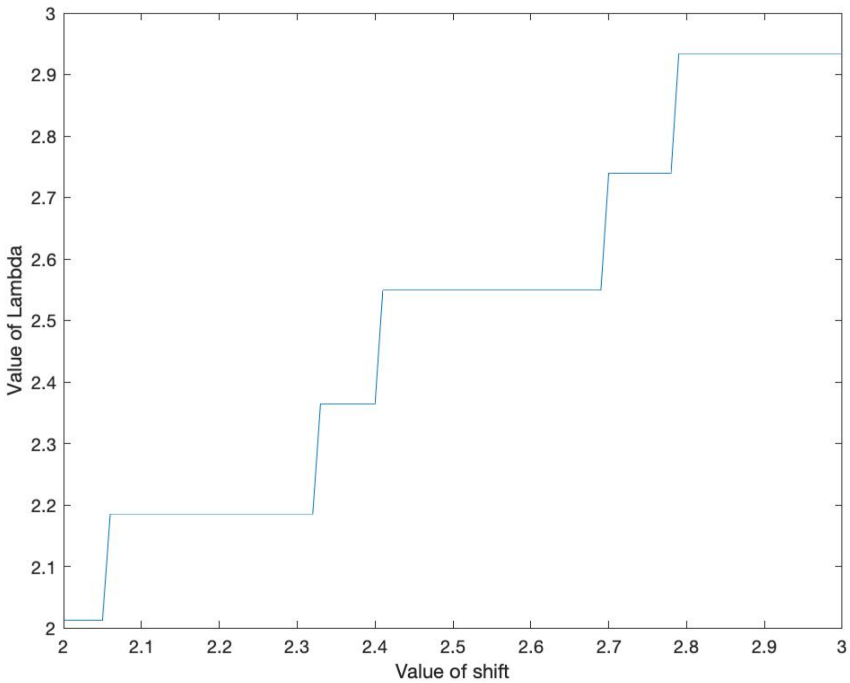

図3 シフトの値ごとの近似固有値

x 軸をシフトの値, y 軸を収束した近似固有値の値として、それぞれのシフトで収束した近似固有値をグラフに描画することができた。