

課題2

(2-1)

次のようなコードを作成した。

```
-----  
A = [1 1 1 1 1;  
      1 2 2 2 2;  
      1 2 3 3 3;  
      1 2 3 4 4;  
      1 2 3 4 5];  
  
B = [2 1 0 0 0;  
      1 2 1 0 0;  
      0 1 2 1 0;  
      0 0 1 2 1;  
      0 0 0 1 2];  
  
v = ones(5, 1);  
shift = 2;  
R = chol(B);  
maxit = 2000;  
tol = 1e-10;  
  
for it = 1:maxit  
    tmp1 = transpose(R)*v;  
    tmp2 = A-shift*B;  
    tmp3 = tmp2\tmp1;  
    w = R*tmp3;  
    lambda = 1/dot(v, w) + shift;  
    v = w/norm(w);  
    x = R\v;  
    x = x/norm(x);  
    tmp3 = A*x;  
    tmp4 = lambda*B*x;  
    r = norm(tmp3 - tmp4);  
    if r < tol  
        break;  
    end  
    T_(it)=r;  
    t(it)=it;  
end  
x  
lambda  
semilogy(t,T_)  
-----
```

これを実行すると、固有値と固有ベクトルが次のように出力される。

x =

0.2673
-0.5345
0.5345
-0.5345
0.2673

lambda =

1.0000

eig関数を用いてこの問題の一般化固有値を求めると次のようになる。

ans =

0.2679
0.3333
0.5000
1.0000
3.7321

これより、作成したプログラムで2付近の固有値を正しく求めていることがわかる。
また、横軸を反復数、縦軸を残差とした図1が描かれた。

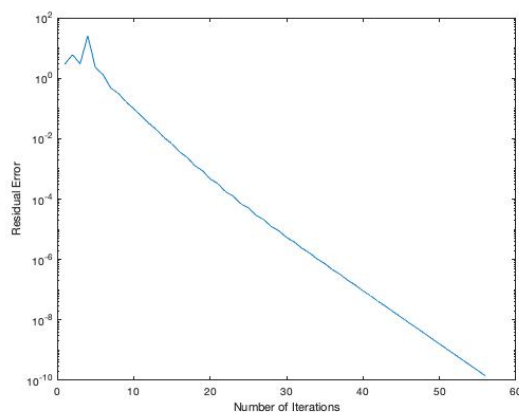


図1 反復回数と残差のグラフ

(2-2)

次のようなコードを作成した。

```
-----  
A = [1 1 1 1 1;  
     1 2 2 2 2;  
     1 2 3 3 3;  
     1 2 3 4 4;  
     1 2 3 4 5];  
  
B = [2 1 0 0 0;  
     1 2 1 0 0;  
     0 1 2 1 0;  
     0 0 1 2 1;  
     0 0 0 1 2];  
  
shift = 2;  
R = chol(B);  
  
interval = 0.01;  
x2 = 1.995;  
x1 = 0.005;  
  
q = (x2-x1)/ interval +1;  
  
sigma = linspace(x1, x2, q);  
  
maxit = 2000;  
tol = 1e-10;  
  
for i = 1:numel(sigma)  
    shift = sigma(i);  
    v = ones(5, 1);  
    for it = 1:maxit  
        tmp1 = transpose(R)*v;  
        tmp2 = A-shift*B;  
        tmp3 = tmp2\tmp1;  
        w = R*tmp3;  
        lambda = 1/dot(v, w) + shift;  
        v = w/norm(w);  
  
        x = R\v;  
        x = x/norm(x);  
  
        tmp3 = A*x;  
        tmp4 = lambda*B*x;
```

```
r = norm(tmp3 - tmp4);  
if r < tol  
    break;  
end  
tmp_l = lambda;  
tmp_t = it;  
end  
s(i) = shift;  
l(i) = tmp_l;  
c(i) = tmp_t;  
end  
  
plot(s,l);  
% plot(s,c);  
xlabel('Value of Shift')  
ylabel('Eigenvalue')  
% ylabel('Number of Iterations')  
-----
```

これを実行すると、各シフト量で得られる固有値の変化および反復回数の変化をそれぞれグラフに描画した図2、図3が得られる。

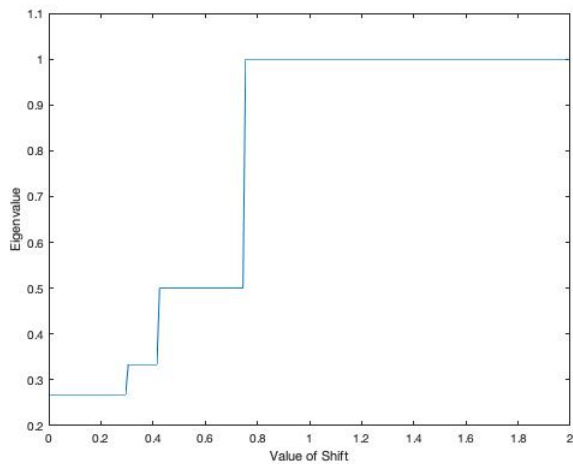


図2 各シフト量で得られる固有値の変化

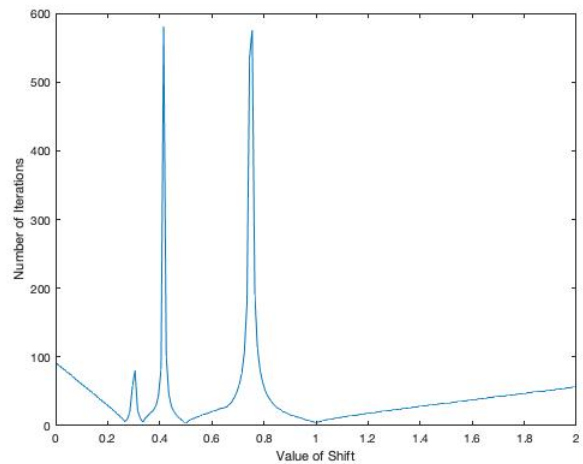


図3 各シフト量で得られる反復回数の変化

課題3

(3-1)

次のようなコードを作成した。

```
—————  
sigma = 0.4;  
[W,D] = make_WD(U,sigma);  
  
L = D - W;  
  
maxit = 2000;  
tol = 1e-02;  
  
A = L;  
B = D;  
  
v = ones(300, 1);  
shift = 0.006;  
R = chol(B);  
  
for it = 1:maxit  
    tmp1 = transpose(R)*v;  
    tmp2 = A-shift*B;  
    tmp3 = tmp2\tmp1;  
    w = R*tmp3;  
    lambda = 1/dot(v, w) + shift;  
    v = w/norm(w);  
    x = R\v;  
    x = x/norm(x);  
    tmp3 = A*x;  
    tmp4 = lambda*B*x;  
    r = norm(tmp3 - tmp4);  
    if r < tol  
        break;  
    end  
end  
  
plot(x,'ro');  
xlabel('添字番号i')  
ylabel('非ゼロ最小固有値に対応する固有ベクトルの第i要素x_i')  
—————
```

これを実行すると、次の図4が描画される。

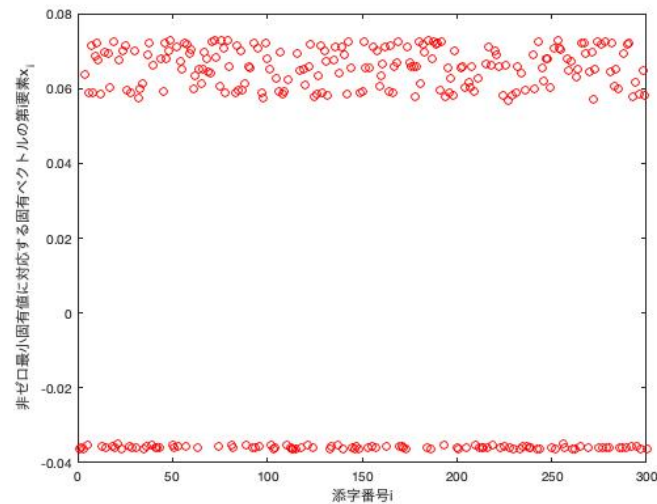


図4 非ゼロ最小固有値に対応する固有ベクトル

(3 -2)

次のようなコードを作成した。

```
-----  
sigma = 0.4;  
[W,D] = make_WD(U,sigma);  
  
L = D - W;  
  
maxit = 2000;  
tol = 1e-02;  
  
A = L;  
B = D;  
  
v = ones(300, 1);  
shift = 0.006;  
R = chol(B);  
  
for it = 1:maxit  
    tmp1 = transpose(R)*v;  
    tmp2 = A-shift*B;  
    tmp3 = tmp2\tmp1;  
    w = R*tmp3;  
    lambda = 1/dot(v, w) + shift;  
    v = w/norm(w);  
    x = R\v;  
    x = x/norm(x);  
end
```

```
tmp3 = A*x;  
tmp4 = lambda*B*x;  
r = norm(tmp3 - tmp4);  
if r < tol  
    break;  
end  
end
```

```
M = mean(x);
```

```
hold on  
for i = 1:300  
    if x(i)>M  
        plot(U(1,i),U(2,i),'or');  
    else  
        plot(U(1,i),U(2,i),'ob');  
    end  
end  
hold off  
-----
```

これを実行すると、次の図5が描画される。

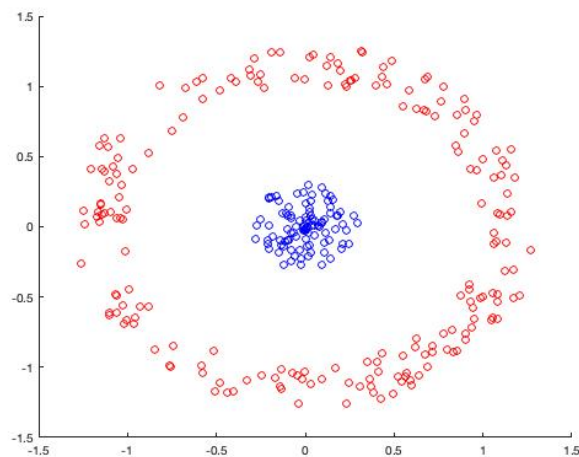


図5 クラスタリング後のデータの分布

図5より、正しくクラスタリングができていることがわかる。