201811319 永崎遼太

# 課題3

$Ax = \lambda x,\ x \neq 0 \iff \det(A - \lambda I) = 0$
を証明する。

$x$ を固有ベクトルとすると、($x \neq 0$)

$$Ax = \lambda x$$

より、

$$(A - \lambda I)x = 0$$

ここで、$A - \lambda I$ に逆行列が存在すると仮定する。左から逆行列をかけると

$$(A - \lambda I)^{-1}(A - \lambda I)x = 0$$

$$\iff \quad x = 0$$

となり、$x \neq 0$ という条件に反し、矛盾が生じる。
したがって $A - \lambda E$ に逆行列は存在しない。
ゆえに

$$\det(A - \lambda I) = 0$$

である。//

# 課題4

## (4-1)

次のようなコードを作成した。

ーーーーー

```
% make P
rng(20201023, 'twister')
n = 1000;
avg = 5;
ind = randi(n, avg*n, 1);
jnd = randi(n, avg*n, 1);
P = sparse(ind, jnd, ones(avg*n, 1));

for j = 1:n
    non0 = nnz(P(:, j));
    if non0 > 0
        P(P(:, j)~=0, j) = 1 / non0;
    else
        P(:, j) = 1 / n;
    end
end
clear ind jnd avg j n non0;

%%%%%
n = 1000;
e = ones(n,1);
```

```
v = e/n;
x = v;
alpha = 0.85;
maxit = 2000;
tol = 1e-04;

for it = 1:maxit
    R = 0;
    l = 1/norm(x);
    ll1 = alpha*P + (1-alpha)*v*transpose(e);
    ll2 = norm(ll1*x - x);
    R = l*ll2;
    if R < tol
        break;
    end
    x = alpha*P*x + (1-alpha)*v;
    T_(it)=R;
    t(it)=it;
end

semilogy(t, T_);
xlim([0 12]);
ylim([1e-05 10]);
xlabel('Number of Iterations')
```

ーーーーー

これを実行すると、図1が描画される。

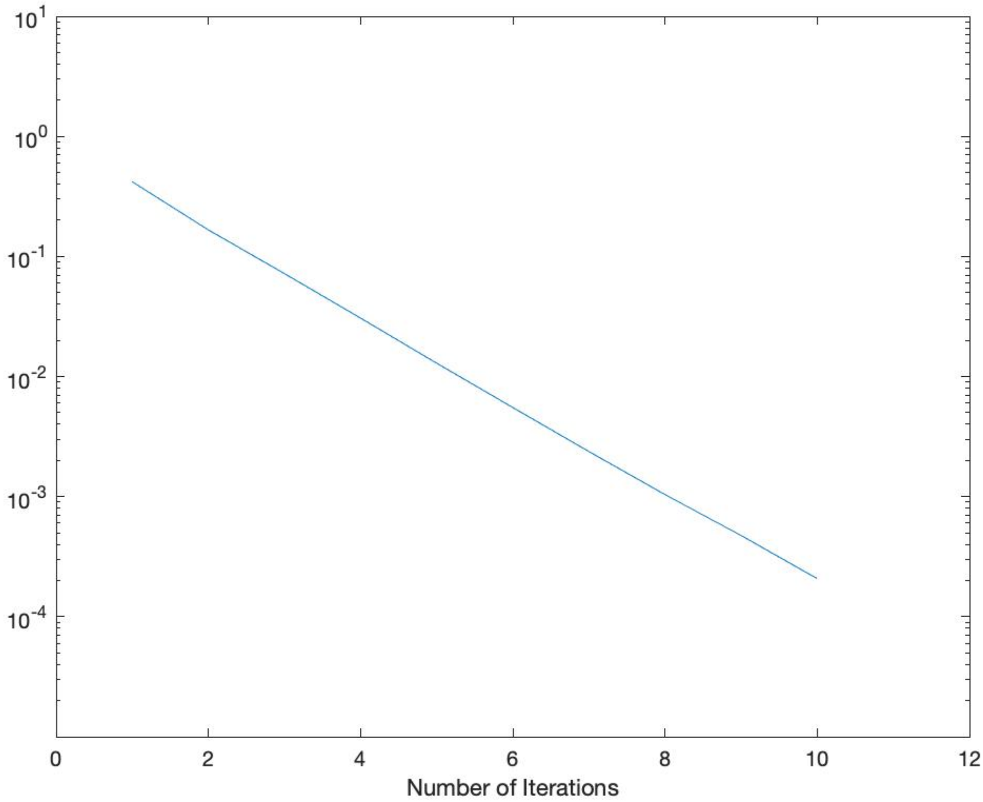


図1 各反復での値

横軸に反復回数、縦軸に各反復での

$$\frac{1}{\|\boldsymbol{x}^{(k)}\|_2}\left\|\left[\alpha P+(1-\alpha)\boldsymbol{v}\boldsymbol{e}^{\mathsf{T}}\right]\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k)}\right\|_2$$

の値を取っている。

(4-2)

次のようなコードを作成した。

ーーーーー

```
% make P
rng(20201023, 'twister')
n = 1000;
avg = 5;
ind = randi(n, avg*n, 1);
jnd = randi(n, avg*n, 1);
P = sparse(ind, jnd, ones(avg*n, 1));

for j = 1:n
    non0 = nnz(P(:, j));
    if non0 > 0
        P(P(:, j)~=0, j) = 1 / non0;
    else
        P(:, j) = 1 / n;
    end
end
clear ind jnd avg j n non0;

%%%%%
n = 1000;
e = ones(n,1);
v = e/n;
x = v;
```

```
alpha = 0.85;
maxit = 2000;
tol = 1e-04;


for it = 1:maxit
    R = 0;
    l = 1/norm(x);
    ll1 = alpha*P + (1-alpha)*v*transpose(e);
    ll2 = norm(ll1*x - x);
    R = l*ll2;
    x = alpha*P*x + (1-alpha)*v;
    if R < tol
        break;
    end
end


k = 3;
M = maxk(x, k);
for i = 1:k
    q(i) = find(x==M(i));
    X = [q(i),M(i)];
    disp(X);
end
```
ーーーーー

**M = maxk(x, k);**　でランクが高い順に上からk(=3)番目まで値をMに格納している。そして、それぞれのランクのページ番号を検索し、それらを組みとして出力するようにしている。

実際にこれを実行すると、次のような結果が出力される。

―――――

```
>>

    1.0000    0.0034

  448.0000    0.0032

   81.0000    0.0032
```

―――――

これより、ページ番号をランクが高い順に上から3つ挙げると、1, 448, 81である。

## (4-3)

次のようなコードを作成した。

―――――

```
% make P
rng(20201023, 'twister')
n = 1000;
avg = 5;
ind = randi(n, avg*n, 1);
jnd = randi(n, avg*n, 1);
P = sparse(ind, jnd, ones(avg*n, 1));
for j = 1:n
    non0 = nnz(P(:, j));
    if non0 > 0
        P(P(:, j)~=0, j) = 1 / non0;
    else
        P(:, j) = 1 / n;
```

```
    end
end
clear ind jnd avg j n non0;

%%%%%
n = 1000;
alpha = 0.5;

for j = 1:10
    alpha = 0.5+0.05*(j-1);
    e = ones(n,1);
    v = e/n;
    x = v;
    maxit = 2000;
    tol = 1e-04;

    for it = 1:maxit
        R = 0;
        l = 1/norm(x);
        ll1 = alpha*P + (1-alpha)*v*transpose(e);
        ll2 = norm(ll1*x - x);
        R = l*ll2;
        if R < tol
            break;
        end
        x = alpha*P*x + (1-alpha)*v;
        T_(it)=R;
        t=it;
```

```
    end
    aa(j) = alpha;
    c(j) = t;
end
plot(aa,c)
xlim([0.45 1]);
ylim([5 13]);
xlabel('alpha')
ylabel('Number of Iterations')
```

ーーーーー

これを実行すると、図2が描画される。

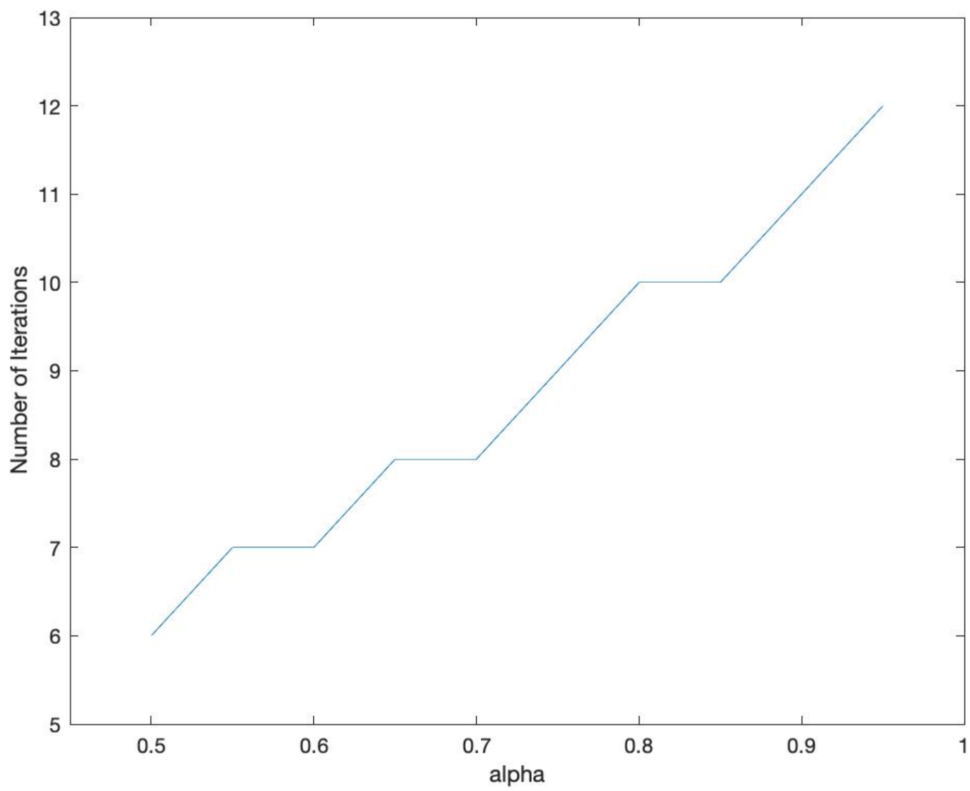


図2 alphaの値ごとの反復回数