Contents

- CS375 HW11
- Problem 1.C
- Problem 1.E
- Problem 1.F
- Problem 2.A and 2.B

CS375 HW11

Juan Alejandro Ormaza November 16 2021

```
clc ;clear all; close all;
format long g
```

Problem 1.C

```
e= sqrt(eps)/4;
A=[1 e; e 1];

charPolynomial=charpoly(A) %characteristic polynomial

eigenvalues=roots(charPolynomial) %lambda 1 and lambda 2
```

```
charPolynomial =

     1    -2     1


eigenvalues =

     1
     1
```

Problem 1.E

See attachments

Problem 1.F

```
e= sqrt(eps)/4;
A=[1 e; e 1];
x=[3;4];
tol1=1e-8;
tol2=1e-9;
tol3=1e-10;

[eval1,evec1]=power_method(A,x,tol1);
[eval2,evec2]=power_method(A,x,tol2);

% to run code just eliminate percentage sign:
```

```
%[eval3,evec3]=power_method(A,x,tol3);

eval1
evec1

eval2
evec2
```

```
eval1 =

          1.00000000279397


evec1 =

         0.750000001629815
                         1


eval2 =

           1.0000000031863


evec2 =

         0.855315474301313
                         1
```

Problem 2.A and 2.B

see attachments

Published with MATLAB® R2021a

CS375: HW11
Juan Alejandro Ormaza
Nov 16 2021

1.

$$A = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$$

a) Characteristic polynomial of A

$$\det(A - \lambda I) = \begin{vmatrix} 1-\lambda & \epsilon \\ \epsilon & 1-\lambda \end{vmatrix} = (1-\lambda)^2 - \epsilon^2$$
$$= \lambda^2 - 2\lambda + 1 - \epsilon^2$$

b) What are the eigenvalues and eigenvectors of A.

$$\lambda_{1,2} = \frac{2 \pm \sqrt{4 - 4(1-\epsilon^2)}}{2}$$

$$\lambda_{1,2} = 1 \pm \sqrt{\cancel{1} - \cancel{1} + \epsilon^2}$$

$$\lambda_{1,2} = 1 \pm \epsilon$$

$$\boxed{\lambda_1 = 1+\epsilon \quad \lambda_2 = 1-\epsilon}$$

$$A - \lambda_1 I = \begin{bmatrix} \cancel{1}-\cancel{1}-\epsilon & \epsilon \\ \epsilon & -\epsilon \end{bmatrix} = \begin{bmatrix} -\epsilon & \epsilon \\ \epsilon & -\epsilon \end{bmatrix}$$

↳ Singular matrix
∞ Solutions

$$(A - \lambda_1 I)v = 0$$

$$\boxed{V_{\lambda_1 = 1+\epsilon} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad V_{\lambda = 1-\epsilon} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}}$$

$$A - \lambda_2 I = \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}$$

c. See Matlab code...

d. Since ϵ is $\sqrt{\epsilon_m}/4$

let's plug this in our equation for the characteristic polynomial

$\lambda^2 - 2\lambda + 1 - \epsilon^2$

$\lambda^2 - 2\lambda + 1 - \frac{\epsilon_m}{16}$

Since $\frac{\epsilon_m}{16} < \epsilon_m$

It essentially can be thought of as 0 in the Computer, yielding:

$\lambda^2 - 2\lambda + 1$

$(\lambda - 1)(\lambda - 1)$

$\lambda_1 = 1 = \lambda_2$

$(A - \lambda_{1,2} I)v = \begin{pmatrix} 0 & \epsilon \\ \epsilon & 0 \end{pmatrix} v$

real eigenvalues and eigenvectors are not being calculated because of.

e.
f. } See MatLab code...

2.

a) $B = A - \frac{\lambda_1}{v^{(1)t} v^{(1)}} v^{(1)} (v^{(1)})^t \qquad (1)$

$Bv^{(1)} = 0$

$(v^{(i)})^t v^{(i)} = 0 \quad \text{for } i \neq j \qquad (2)$

let's multiply both sides of eq (1) by $V^{(1)}$

$$BV^{(1)} = \left(A - \frac{\lambda_1}{V^{(1)t} V^{(1)}} V^{(1)} (V^{(1)})^t \right) V^{(1)}$$

$$BV^{(1)} = \left(AV^{(1)} - \frac{\lambda_1}{\cancel{V^{(1)t} V^{(1)}}} V^{(1)} \cancel{(V^{(1)t} V^{(1)})} \right)$$

$$BV^{(1)} = AV^{(1)} - \lambda_1 V^{(1)}$$

by definition:

$$BV^{(1)} = \cancel{AV^{(1)}} - \cancel{AV^{(1)}} = 0$$

$$BV^{(1)} = 0$$

let us try another number now $\neq 1$ and let's call it j

let us plug $V^{(j)}$ on both sides

$$BV^{(j)} = AV^{(j)} - \frac{\lambda_1}{V^{(1)t} V^{(1)}} V^{(1)} \cancelto{0}{\underbrace{V^{(1)t} V^{(j)}}_{\text{using orthogonality.}}} \qquad 1 \neq j$$

$$BV^{(j)} = AV^{(j)}$$ This means A will have the same eigenValues as B for all $\lambda_{j \neq 1}$

6. See MatLab code...

```
%Juan Alejandro Ormaza
% Algorithm based on Burden's Numerical Analysis 9th edition

function [eval, evec] = power_method(A,x,tol)

    k=1;
    N=1e100;

    x=x/norm(x,inf); %normalized

    while k<N
        y=A*x;

        lambda=norm(y,inf);

        error=norm((x-(y/norm(y,inf))),2);
        x=y/norm(y,inf);

        if error<tol
            break;
        end

        k=k+1;
    end

    eval=lambda;
    evec=x;



end
```

```
Not enough input arguments.

Error in power_method (line 9)
    x=x/norm(x,inf); %normalized
```

Published with MATLAB® R2021a

```matlab
% Juan Alejandro Ormaza
% Algorithm based on Burden's Numerical Analysis 9th edition

function [eval2] = power_method_mod(A,x,tol)

    k=1;
    N=1e100;

    x=x/norm(x,inf); %normalized


    while k<N
        y=A*x;

        lambda=norm(y,inf);

        error=norm((x-(y/norm(y,inf))),2);
        x=y/norm(y,inf);

        if error<tol
            break;
        end

        k=k+1;
    end

    evec=x;
    eval=lambda;

    B= A-(eval/(evec'*evec))*(evec*evec');
    eval2=eig(B);
    eval2=eval2(2);



end
```

```
Not enough input arguments.

Error in power_method_mod (line 9)
    x=x/norm(x,inf); %normalized
```

Published with MATLAB® R2021a