بسمه تعالی

عنوان : مقایسه تعداد تکرار تا رسیدن به جواب میان روش های محاسباتی برای ماتریس های تصادفی چهار و پنج و شش بعدی

مراحل پروژه به صورت زیر می باشند :

1- تولید ماتریس تصادفی مربعی $n * n$ براساس مراحل زیر :

1-1 درایه های روی قطر اصلی یک باشند یعنی رابطه $a_{i,j} = 1 \ \ ,\forall\ i = j, i, j = 1,2, \dots, n$ برقرار است.

1-2 درایه های زرد رنگ به صورت اعداد تصادفی شامل یکی از اعداد $1,2,3,4,5,6,7,8,9,\frac{1}{2},\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6},\frac{1}{7},\frac{1}{8},\frac{1}{9}$ تعیین گردند.

1-3 درایه های بالای قطر اصلی براساس رابطه $a_{i,j} * a_{j,k} = a_{i,k}\ , i, j, k = 1,2, \dots, n$ (به عنوان مثال $a_{2,4} * a_{4,5} = a_{2,5}$) بدست آیند بگونه ای که لازم است به ترتیب ابتدا درایه های سبز رنگ، سپس درایه های بنفش رنگ، سپس درایه های قرمز رنگ و ... محاسبه گردند زیرا تعیین مقدار درایه های سبز رنگ پیش نیاز محاسبه مقدار درایه های بنفش رنگ و مقدار درایه های بنفش رنگ پیش نیاز محاسبه مقدار درایه های قرمز رنگ و ... می باشند.

1-4 پس از محاسبه درایه های بالای قطر اصلی، درایه های پایین قطر اصلی از رابطه $a_{i,j} * a_{j,i} = 1$ قابل محاسبه خواهند بود.

$$A = \begin{array}{cccccccccc}
1 & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & & & & & \\
 & 1 & a_{2,3} & a_{2,4} & a_{2,5} & \dots & & & & \\
 & & 1 & a_{3,4} & a_{3,5} & \dots & a_{i-4,j} & & & \\
 & & & 1 & \dots & \dots & a_{i-3,j} & \dots & & \\
 & & & & 1 & \dots & a_{i-2,j} & \dots & \dots & \\
 & & & & & 1 & a_{i-1,j} & \dots & \dots & a_{n-4,n} \\
 & & & & & & 1 & \dots & \dots & a_{n-3,n} \\
 & & & & & & & 1 & \dots & a_{n-2,n} \\
 & & & & & & & & 1 & a_{n-1,n} \\
 & & & & & & & & & 1
\end{array}$$

2- تعیین چندجمله مشخصه Characteristic polynomial ماتریس تصادفی A مرحله اول که کد آن برای یک ماتریس سه بعدی مربعی نمونه به شکل زیر می باشد :

Characteristic polynomial of the matrix `A` in terms of `x`:

```
syms x
A = sym([1 1 0; 0 1 0; 0 0 1]);
polyA = charpoly(A,x)
polyA =
x^3 - 3*x^2 + 3*x - 1
```

لازم به ذکر است که کد فوق باید برای این پروژه بازنویسی شود و فقط syntax دستورات آن برای راحتی کار اشاره شده است.

3- بکارگیری الگوریتم پنج روش «دوبخشی»-«نابجایی»-«نیوتن رافسون»-«وتری»-«نقطه ثابت» جهت حل چندجمله مشخصه ماتریس تصادفی مرحله دوم با لحاظ میزان خطای 10^{-4}.

به جواب چندجمله مشخصه «مقدار ویژه» یا eigenvalue گفته می شود و کدهای الگوریتم های مذکور به شرح زیر برای محاسبه مقدار ویژه نوشته شده و تست گردیده است و برای راحتی کار به شما ارائه شده است.

3-1- روش دوبخشی

به جای نقطه چین در کد زیر باید چندجمله مشخصه از مرحله دوم جایگزین شود.

```
function [x, fx] = BisectionMethod(f, interval, MinRelErr)

a = interval (1);
b = interval (2);
fa = f(a);
fb = f(b);
it = 0;
     While true
           it = it + 1;
           m = (a+b)/2;
           fm = f(m);
           if fm*fa>0
             a = m;
             fa = fm;

           elseif fm*fb>0
             b = m;
             fb = fm;

           else
             break;
```

```
            end
            EstRelErr = abs((a-b)/((a+b)/2));
            if EstRelErr <MinRelErr
             break;
            end

      end

      x = m;

end

clc;
clear;
close all;
%% The function f(x)
f=@(x) …………………;
[x, fx] = BisectionMethod(f, [-8 8], 1e-4);
disp('Final Solution:');
disp(['eigenvalue= ' num2str(x)]);
disp(['Number of Iteration= ' num2str(it)]);
```

3-2- روش نابجایی

به جای نقطه چین در کد زیر باید چندجمله مشخصه از مرحله دوم جایگزین شود.

```
function [x, fx] = FalsePositionMethod(f, interval, MinRelErr)

a = interval (1);
b = interval (2);
fa = f(a);
fb = f(b);
it = 0;
      While true
        it = it + 1;
        m = (a*fb-b*fa)/(fb-fa);
        fm = f(m);
        if fm*fa>0
            a = m;
            fa = fm;

        elseif fm*fb>0
            b = m;
            fb = fm;

        else
            break;

        end
        EstRelErr = abs((a-b)/((a+b)/2));
        if EstRelErr <MinRelErr
            break;
```

```
        end

      end

      x = m;

end
clc;
clear;
close all;
%% The function f(x)
f=@(x) …………………;
[x, fx] = FalsePositionMethod(f, [-8 8], 1e-4);
disp('Final Solution:');
disp(['eigenvalue= ' num2str(x)]);
disp(['Number of Iteration= ' num2str(it)]);
```

3-3- روش نیوتن رافسون

به جای نقطه چین در کد زیر باید چندجمله مشخصه از مرحله دوم جایگزین شود.

```
function df = Derivative(f, x)
      dx = 1e-4;
      df = (f(x+dx)-f(x-dx))/(2*dx);
end
function [x, fx] = NewtonRaphsonMethod(f, x0, MinRelErr)
x = x0;
fx = f(x);
      while true
            f1x = Derivative(f,x);
            xnew = x - fx/f1x;
            fxnew = f(xnew);
            EstRelErr = abs((xnew-x)/x);
            x = xnew;
            fx = fxnew;
            if EstRelErr <MinRelErr
              break;
            end
      end
end
clc;
clear;
close all;
%% The function f(x)
f=@(x) …………………;
[x, fx] = NewtonRaphsonMethod(f, 1, 1e-4);
disp('Final Solution:');
disp(['eigenvalue= ' num2str(x)]);
disp(['Number of Iteration= ' num2str(it)]);
```

3-4- روش وتری

به جای نقطه چین در کد زیر باید چندجمله مشخصه از مرحله دوم جایگزین شود.

```
function [x, fx] = SecantMethod(f, interval, MinRelErr)
x1= interval(1);
x2= interval(2);
fx1 = f(x1);
fx2 = f(x2);
      while true
      xnew = x2 - fx2*(x2-x1)/(fx2-fx1);
      fxnew = f(xnew);
      EstRelErr = abs((xnew-x2)/x2);
      x1 = x2;
      x2 = xnew;
      fx1 = fx2;
      fx2 = fxnew;
      if EstRelErr <MinRelErr
       break;
      end

      end
x = x2;
fx = fx2;

end
clc;
clear;
close all;
%% The function f(x)
f=@(x) …………………;
[x, fx] = SecantMethod(f, [-8 8], 1e-4);
disp('Final Solution:');
disp(['eigenvalue= ' num2str(x)]);
disp(['Number of Iteration= ' num2str(it)]);
```

3-5- روش نقطه ثابت

به جای نقطه چین در کد زیر باید چندجمله مشخصه از مرحله دوم جایگزین شود.

```
function [x, fx] = FixedPointMethod(f, x0, MinRelErr)
g=@(x) fx+x;
x = x0;
fx = f(x);
gx = g(x);
      while true
      xnew = gx;
      RelErr = abs((xnew-x)/x);
      x = xnew;
      fx = f(x);
      gx = g(x);
```

```
      if RelErr <MinRelErr
        break;
      end

      end
end
clc;
clear;
close all;
%% The function f(x)
f=@(x) ………………;
[x, fx] = FixedPointMethod(f, 1, 1e-4);
disp('Final Solution:');
disp(['eigenvalue= ' num2str(x)]);
disp(['Number of Iteration= ' num2str(it)]);
```

4- بکارگیری ماتریس تصادفی A مرحله اول در الگوریتم روش «توانی» با لحاظ میزان خطای 10^{-4} براساس مراحل زیر.

لازم به ذکر است که در این روش برخلاف پنج روش قبلی نیازی به چندجمله مشخصه نمی باشد.

4-1- بردار ستونی $U_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{n*1}$ در نظر گرفته می شود.

4-2- مقدار بردار U_1 از رابطه $U_1 = \frac{\mathrm{A}U_0}{\|\mathrm{A}U_0\|_2}$ محاسبه می گردد که در آن نرم اقلیدسی هر بردار مانند $W = \begin{bmatrix} a \\ b \\ c \\ . \\ . \\ . \\ . \end{bmatrix}_{n*1}$ از رابطه $\| W \|_2 = \sqrt{a^2 + b^2 + c^2 + \cdots}$ بدست می آید.

4-3- مقدار λ_1 از رابطه $\lambda_1 = \frac{U_1^T \mathrm{A} U_1}{U_1^T U_1}$ محاسبه می گردد که در آن ترانهاده هر بردار مانند

$W = \begin{bmatrix} a \\ b \\ c \\ . \\ . \\ . \\ . \end{bmatrix}_{n*1}$ از رابطه $W^T = [a \quad b \quad c \quad . \quad . \quad . \quad .]_{1*n}$ بدست می آید.

4-4- مقدار بردار U_2 از رابطه $U_2 = \frac{\mathrm{A}U_1}{\|\mathrm{A}U_1\|_2}$ محاسبه می گردد.

4-5- مقدار λ_2 از رابطه $\lambda_2 = \frac{U_2^T \mathrm{A} U_2}{U_2^T U_2}$ محاسبه می گردد.

4-6- مقدار $\left| \frac{\lambda_2 - \lambda_1}{\lambda_2} \right|$ محاسبه می گردد و چنانچه $\left|\frac{\lambda_2 - \lambda_1}{\lambda_2}\right| \leq \varepsilon = 10^{-4}$ باشد آنگاه الگوریتم خاتمه می یابد و مقدار λ_1 به عنوان eigenvalue در نظر گرفته می شود، در غیر این صورت مرحله بعد اجرا می شود.

4-7- مقدار بردار U_3 از رابطه $U_3 = \frac{\mathrm{A}U_2}{\|\mathrm{A}U_2\|_2}$ محاسبه می گردد.

4-8- مقدار λ_3 از رابطه $\lambda_3 = \frac{U_3^T \mathrm{A} U_3}{U_3^T U_3}$ محاسبه می گردد.

4-9- مقدار $\left| \frac{\lambda_3 - \lambda_2}{\lambda_3} \right|$ محاسبه می گردد و چنانچه $\left|\frac{\lambda_3 - \lambda_2}{\lambda_3}\right| \leq \varepsilon = 10^{-4}$ باشد آنگاه الگوریتم خاتمه می یابد و مقدار λ_2 به عنوان «مقدار ویژه» یا eigenvalue در نظر گرفته می شود، در غیر این صورت مرحله بعد اجرا می شود.

4-10- مقدار بردار U_4 از رابطه $U_4 = \frac{\mathrm{A}U_3}{\|\mathrm{A}U_3\|_2}$ محاسبه می گردد.

....

....

....

5- خروجی پروژه به صورت نمایش «مقدار ویژه» و «تعداد تکرار تا رسیدن به جواب» در شش روش «دوبخشی»-«نابجایی»-«نیوتن رافسون»-«وتری»-«نقطه ثابت»-«توانی» برای صد ماتریس تصادفی چهار و پنج و شش بعدی در قالب جدول زیر مورد نظر می باشد :

			روش نیوتن رافسون	روش دوبخشی	روش نقطه ثابت	روش وتری	روش نابجایی	روش توانی
ماتریس 4 بعدی	ماتریس تصادفی اول	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	. . . .	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	ماتریس تصادفی صدم	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
ماتریس 5 بعدی	ماتریس تصادفی اول	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	. . . .	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	ماتریس تصادفی صدم	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
ماتریس 6 بعدی	ماتریس تصادفی اول	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	. . . .	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						
	ماتریس تصادفی صدم	مقدار ویژه						
		تعداد تکرار تا به رسیدن به جواب						