Roots of Polynomials, Roots location with Software Packages [1-6]

References:

1. Chapra S.C. and Canale R.P. "Numerical Methods for Engineers", Sixth Edition,McGraw Hill, International Edition 2010.

2. Chapra S.C. and Canale R. P. "Yazılım ve programlama Uygulamalarıyla Mühendisler için Sayısal Yöntemler" 4.Basımdan Çevirenler: Hasan Heperkan ve Uğur Kesgin 2003.

3.Chapra S.C. "Applied Numerical Methods with MATLAB for engineers and Scientists" Third Edition, McGraw Hill, International Edition 2012.

4. Mathews J.H. and Fınk K.D. "Numerical Methods using MATLAB", Fourth Edition, Pearson P. Hall, International Edition 2004.

5. Fausett L.V. "Applied Numerical Analysis Using MATLAB, Second Edition, PearsonP. Hall, International Edition, 2008.

6. Gilat A. And Subramaniam V. "Numerical Methods, An introduction with Applications Using MATLAB", Second Edition,John Wiley and Sons. Inc. 2011.

Müller's method with guesses of x_0, x_1, $x_2 = 0.5, 0.4, 0.3$ respectively to determine a root of the equation

$$f(x) = x^3 - 3x + 1$$

$$h_0 = x_1 - x_0 \quad h_1 = x_2 - x_1 \qquad \delta_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad \delta_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$a = \frac{\delta_1 - \delta_0}{h_1 + h_0} \quad b = ah_1 + \delta_1 \quad c = f(x_2)$$

$$x_3 = x_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

1st iteration

$$f(x_0) = f(0.5) = 0.5^3 - 3 \times 0.5 + 1 = -0.375$$

$$f(x_1) = f(0.4) = -0.136$$

$$f(x_2) = f(0.3) = 0.127$$

$$c = f(x2) = 0.127$$

$$h_0 = 0.4 - 0.5 = -0.1$$

$$h_1 = 0.3 - 0.4 = -0.1$$

$$\delta_0 = (-0.136 - (-0.375) / (-0.1) = -2.39$$

$$\delta_1 = (0.127 - (-0.136) / (0.3 - 0.4) = -2.63$$

$$a = \frac{(-2.63 - (-2.39))}{(-0.1 - 0.1)} = 1.2$$

$$b = 1.2(-0.1) + (-2.63) = -2.75$$

$$c = 0.127$$

$$\sqrt{b^2 - 4ac} = \sqrt{(-2.75)^2 - 4 \times 1.2 \times 0.127} = 2.6368$$

$$|-2.75 - 2.6368| > |-2.75 + 2.6368|$$

$$x_3 = 0.3 + \frac{(-2 \times 0.127)}{(-2.75 - 2.6368)} = 0.34715$$

$$f(0.34715) = 0.000386$$

the Newton-Raphson method to determine one real root for the equation

using an initial guess of 3.0

$$f(x) = 2x^3 - 11.7x^2 + 17.7x - 5$$

- $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

$f(x) = 2x^3 - 11.7x^2 + 17.7x - 5$

$f'(x) = 6x^2 - 23.4x + 17.7$

$$x_{i+1} = x_i - \frac{2x_i^3 - 11.7x_i^2 + 17.7x_i - 5}{6x_i^2 - 23.4x_i + 17.7}$$

Iteration	x_i	ε_a (%)
1	3	
2	5.1333	[(5.1333 – 3) / 5.1333] 100% = 41.5581
3	4.2698	[(4.2698 – 5.1333) / 4.2698] 100% = 20.2234
4	3.7929	[(3.7929 – 4.2698) / 3.7929] 100% = 12.5735
5	3.5998	[(3.5998 – 3.7959) / 3.5998] 100% = 5.4503

x = 3.5998

```
for i=1:kmax                    'bisection'
    ya=f(xl);
    yb=f(xu);
    xr=0.5*(xl+xu);
    yr=f(xr);
    fprintf(fid,'%4.1f %7.4f %7.4f %7.4f   %7.4f   %7.4f   %7.4f\n',i,xl,xu,xr,ya,yb,yr);
    if ya*yr<0
        xu=xr;
    else xl=xr;
    end
    if abs(((sqrt(508)-xr)/(sqrt(508)))*100)<tol;
        break
    end
end
```

'falseposition'

```
for k=1:kmax
    xr=xu-yb*(xu-xl)/(yb-ya);
    y=f(xr);
    iter=k;
    fprintf(fid,'%4.1f %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f\n',iter,xl,xu,xr,ya,yb,y);
    if abs(((sqrt(508)-xr)/(sqrt(508)))*100)<tol;
       disp('false position has converged');break;
    end
   if sign(y)~=sign(ya)
       xu=xr;
       yb=y;
   else
       xl=xr;
       ya=y;
   end
   if iter>=kmax
       disp('zero not found to desired tolerance')
   end
end
```

```
for i=1:kmax
    ya=f(xl);
    yb=f(xu);
    xr=xl-(ya*(xl-xu))/(ya-yb);
    yr=f(xr);
    fprintf(fid,'%4.1f %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f \n',i,xl,xu,xr,ya,yb,yr);
    if abs(((sqrt(508)-xr)/(sqrt(508)))*100)<tol;
        break
    end
    xu=xl;
    xl=xr;
end
```

'secant'

$$f(x) = X^2 - 12X + 27 = (x-9)(x-3)$$

```
>> sys=[1 -12 27]

sys =

1   -12    27

>> roots(sys)

ans =

9

3

>> fx=(ans).^2-12*(ans)+27

fx =

0

0
```