Contents


```
% Midterm Problem 3 (e)
% Polynomial deflation
% This script works up to 6th order

clear
clc
close all
```

Order & Coefficients of the poly

```
o = 5; % Order of the poly
itn = o;
A(:,1) = [1;-15;85;-225;274;-120]; % Coefficients of the poly (Decreasing order)
```

Setup for iteration

```
maxit = 1000000;
tol = 1e-6;
x0 = 0.5;   % Initial guess for the root from the graph

if 2 < o && o < 7
```

```
    for k = 1 : o-2
```

Newton's approx. method to find the 1st root

```
        if itn == 6
            F = @(x) A(1,k)*x^(o+1-k) + A(2,k)*x^(o-k) + A(3,k)*x^(o-1-k) + A(4,k)*x^(o-2-k) + A(5,k)*x^(o-3-k) + A(6,k)*x^(o-4-k) + A(7,k)*x^(o-5-k);
        elseif itn == 5
            F = @(x) A(1,k)*x^(o+1-k) + A(2,k)*x^(o-k) + A(3,k)*x^(o-1-k) + A(4,k)*x^(o-2-k) + A(5,k)*x^(o-3-k) + A(6,k)*x^(o-4-k);
        elseif itn == 4
            F = @(x) A(1,k)*x^(o+1-k) + A(2,k)*x^(o-k) + A(3,k)*x^(o-1-k) + A(4,k)*x^(o-2-k) + A(5,k)*x^(o-3-k);
        elseif itn == 3
            F = @(x) A(1,k)*x^(o+1-k) + A(2,k)*x^(o-k) + A(3,k)*x^(o-1-k) + A(4,k)*x^(o-2-k);
        end % if
        [roots(k,1),it(k,1),success(k,1)] = newton_approx(F,x0,maxit,tol);
        table(roots,it,success)
```

```
ans =

  1×3 table

    roots    it    success
    _____    __    _______

      1      59     true



ans =

  2×3 table

    roots    it     success
    _____    ___    _______

      1       59     true
      2      216     true



ans =

  3×3 table
```

```
roots    it     success
_____    ___    _______

  1       59     true
  2      216     true
  3      237     true
```

Division

```
        N = roots(k,1)  % Difine the divisor
        for i = 1 : itn+1   % Reordering the coefficients
            Ap(i,k) = A(itn-i+2,k);    % Increasing order
        end % for
        % Division
        Qi(itn+1,k) = Ap(itn+1,k);
        for i = itn : -1 : 1
            Qi(i,k) = Ap(i,k) + N*Qi(i+1,k);    % Increasing order
        end % for
        R = Qi(1,k);   % Remainder
        % Coefficients for Q
        for i = 1 : itn
            Q(i,k) = Qi(itn-i+2,k);    % Decreasing order
        end % for
        for i = itn : -1 : 1
            A(i,k+1) = Q(i,k);
        end % for
        itn = itn - 1;  % Update for the order of Qn-1

disp('Qn-1 = ')
disp(Q)
```

```
N =

    1.0000

Qn-1 =
    1.0000
  -14.0000
   71.0000
 -154.0000
  120.0000



N =

    2.0000

Qn-1 =
    1.0000    1.0000
  -14.0000  -12.0000
   71.0000   47.0000
 -154.0000  -60.0000
  120.0000         0



N =

    3.0000

Qn-1 =
    1.0000    1.0000    1.0000
  -14.0000  -12.0000   -9.0000
   71.0000   47.0000   20.0000
 -154.0000  -60.0000         0
  120.0000         0         0
```

```
    end % for
```

Coefficients of the quadratic equation

```
    P2 = [A(1,k+1);A(2,k+1);A(3,k+1)];
    x = solveqe(P2)
```

```
x =

    5.0000
    4.0000
```

```
elseif o == 2
    k = 2;
    x = solveqe(A)
elseif o == 1
    k = 1;
    [roots,it,success] = newton_approx(F,x0,maxit,tol)
end % if
```

Published with MATLAB® R2020b