Table of Contents


```
% Pierce Zhang, CMOR220, FALL 2023, Compentency complex values and vector
% polynomials
% competency_complex.m
% Answers to competency on complex values and vector polynomials
% Last modified: 23 September 2023

% Driver to answer the problems
function competency_complex
    problem1();
    problem2();
    problem3();
    problem4();
end
```

PROBLEM 1

Inputs: none Outputs: none

```
function problem1
    % Initializes several complex numbers
    z1 = 3 + 2i
    z2 = 60 - 90i
    z3 = 45 - 34i
    z4 = -2 + 7i
end
```

```
z1 =

   3.0000 + 2.0000i


z2 =

  60.0000 -90.0000i


z3 =

  45.0000 -34.0000i
```

```
z4 =

  -2.0000 + 7.0000i
```

PROBLEM 2

Inputs: x Outputs: val, the evaluation of f2 at x

```
function [val] = f2(x)
    val = 0.1 * x^3 - x^2 + 5;
end

% Inputs: x
% Outputs: val, the evaluation of the derivative of f2 at x
function [val] = f2dx(x)
    val = 0.3*x^2 - 2*x;
end

% Inputs: f, the function, fprime, the derivative of the function, x, the
% initial value, tol, the tolerance interval
% Outputs: x, the root estimation, iter, the number of iterations it took
function[x,iter] = denewt(f,fprime,x,tol)
    % Newton's methods
    iter = 0;
    while (abs(f(x)) > tol)
        x = x - f(x)/fprime(x);
        iter = iter + 1;
    end
end

% Inputs: none
% Outputs: none
function problem2
    % Runs Newton's method on various starting values for f2
    tol = 0.01;
    x1 = denewt(@f2,@f2dx,-5,tol)
    x2 = denewt(@f2,@f2dx,1,tol)
    x3 = denewt(@f2,@f2dx,8,tol)
end
```

```
x1 =

   -2.0381


x2 =

    2.5993


x3 =
```

```
    9.4388
```

PROBLEM 3

Inputs: none Outputs: none

```
function problem3
    % Runs polyval to evaluate various polynomials at x=3
    x = 3;
    y1 = polyval([1,0,-4,-2,7,-8],x)
    y2 = polyval([7,0,0,-30,700],x)
    y3 = polyval([8i,-2,3i,-5],x)
end
```

```
y1 =

   130


y2 =

        1177


y3 =

  -2.3000e+01 + 2.2500e+02i
```

PROBLEM 4

Inputs: x Outputs: val, the evaluation of f4 at x

```
function [val]=f4(z)
    val = z^4 - 0.84*z^2 - 0.16;
end

% Inputs:f, the function to find roots for, z, first bound, zprev, second
% bound, iter, number of iterations to use
% Outputs: root, the estimation of the zero of f
function [root] = secmethod(f, z, zprev, iter)
    % Implements the secant method a function f using two estimations z and
    % zprev
    for i=1:iter
        znext = z - f(z) * ((z - zprev)/(f(z) - f(zprev))); % update
        zprev = z;
        z = znext; %change z
    end
    root = z; % declare
end
```

```
% Inputs: none
% Outputs: none
function problem4
    % Executes the secant method for the function f4 on 12 iterations
    root = secmethod(@f4, 1+2i,4+i,12)
end
```

```
root =

   0.0010 + 0.3987i
```

Published with MATLAB® R2023a