Assignment 6

1. Find approximations for the two roots of the polynomial $0.0002358x^2 - 5535.0x + 0.00003513$ using the quadratic formula you learned in secondary school. Then, find the same roots, but choosing the appropriate formula for each.

2. Find approximations for the two roots of the polynomial $0.0002358x^2 + 5535.0x - 0.00003513$ using the quadratic formula you learned in secondary school. Then, find the same roots, but choosing the appropriate formula for each.

- 3. Given the function $f(x) = x^3 x^2 x 1$, approximate the real root using two steps of each of:
 - a. Newton's method starting with $x_0 = 2.0$,
 - b. the bisection method starting with [1, 2],
 - c. the bracketed secant method starting with [1, 2] (optional), and
 - d. the secant method starting with $x_0 = 2.0$ and $x_1 = 1.9$.

```
% Newton's method
f = @(x)(x^3 - x^2 - x - 1.0);
df = (a(x)(3*x^2 - 2*x - 1.0);
x0 = 2.0;
x1 = x0 - f(x0)/df(x0)
        x1 = 1.857142857142857
x2 = x1 - f(x1)/df(x1)
        x2 = 1.839544513457557
% Bisection method
a = 1;
b = 2;
m = (a + b)/2.0;
if sign(f(a)) == sign(f(m)); a = m else b = m end
         a = 1.5000000000000000
m = (a + b)/2.0;
if sign(f(a)) == sign(f(m)); a = m else b = m end
         a = 1.7500000000000000
% Bracketed secant method (optional)
a = 1;
b = 2;
m = (a*f(b) - b*f(a))/(f(b) - f(a));
if sign(f(a)) == sign(f(m)); a = m else b = m end
         m = (a*f(b) - b*f(a))/(f(b) - f(a));
if sign(f(a)) == sign(f(m)); a = m else b = m end
         a = 1.816326530612245
% Secant method
x0 = 2.0;
x1 = 1.9;
x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0))
        x2 = 1.846390168970814
x3 = (x1*f(x2) - x2*f(x1))/(f(x2) - f(x1))
        x3 = 1.839628859068081
```

- 4. Given the same function as in Question 3, approximate the first positive root using one step of each of:
 - a. Muller's method starting with $x_0 = 2.0$, $x_1 = 1.9$ and $x_2 = 1.8$ (optional), and
 - b. inverse quadratic interpolation with the same three points.

```
% Muller's method (optional)
x0 = 2.0;
x1 = 1.9;
x2 = 1.8;
% Find the polynomial passing through
          (x0 - x2, f(x0)), (x1 - x2, f(x1)), (x2 - x2, f(x2))
p = polyfit([x0 x1 x2] - x2, [f(x0) f(x1) f(x2)], 2)
        delta = (-2*p(3))/(p(2) + sqrt(p(2)^2 - 4*p(1)*p(3)))
    delta = 3.935683999680209e-02
x3 = x2 + delta
       x3 = 1.839356839996802
% Inverse quadratic interpolation
x0 = 2.0;
x1 = 1.9;
x2 = 1.8;
% Find the polynomial passing through (f(x0), x0), (f(x1), x1), (f(x2), x2)
p = polyfit([f(x0) f(x1) f(x2)], [x0 x1 x2], 2)
        p = -0.02145975382064491 0.1825590389332351 1.838900714887409
% Get the constant coefficient
x3 = p(3)
       x3 = 1.838900714887409
```

- 5. Given the function $f(x) = x^2 \cos(0.4x)e^{-0.3x}$, approximate the first positive root using two steps of each of:
 - a. Newton's method starting with $x_0 = 4.0$,
 - b. the bisection method starting with [3, 4],
 - c. the bracketed secant method starting with [3, 4] (optional), and
 - d. the secant method starting with $x_0 = 3.8$ and $x_1 = 3.9$.

```
% Newton's method
f = @(x)(x^2*\cos(0.4*x)*\exp(-0.3*x));
df = @(x)(2.0*x *cos(0.4*x)*exp(-0.3*x)
        -0.4*x^2*sin(0.4*x)*exp(-0.3*x)
        - 0.3*x^2*\cos(0.4*x)*\exp(-0.3*x);
x0 = 4.0;
x1 = x0 - f(x0)/df(x0)
       x1 = 3.928021373533735
x2 = x1 - f(x1)/df(x1)
       x2 = 3.926991039021064
% Bisection method
a = 3.0;
b = 4.0;
m = (a + b)/2.0;
if sign(f(a)) == sign(f(m)); a = m else b = m end
        a = 3.500000000000000
m = (a + b)/2.0;
if sign(f(a)) == sign(f(m)); a = m else b = m end
        a = 3.7500000000000000
% Bracketed secant method (optional)
a = 3.0;
b = 4.0:
m = (a*f(b) - b*f(a))/(f(b) - f(a));
if sign(f(a)) == sign(f(m)); a = m else b = m end
        a = 3.904055035798684
m = (a*f(b) - b*f(a))/(f(b) - f(a));
if sign(f(a)) == sign(f(m)); a = m else b = m end
        a = 3.926649703297110
% Secant method
x0 = 3.8;
x1 = 3.9;
x2 = (x0*f(x1) - x1*f(x0))/(f(x1) - f(x0))
       x2 = 3.927770674823227
x3 = (x1*f(x2) - x2*f(x1))/(f(x2) - f(x1))
       x3 = 3.926986348481126
```

- 6. Given the same function as in Question 5, approximate the first positive root using one step of each of:
 - a. Muller's method starting with $x_0 = 3.8$, $x_1 = 4.0$ and $x_2 = 3.9$ (optional), and
 - b. inverse quadratic interpolation with the same three points.

```
% Muller's method (optional)
x0 = 3.8;
x1 = 4.0;
x2 = 3.9;
p = polyfit([x0 x1 x2] - x2, [f(x0), f(x1), f(x2)], 2)
         p = -0.4079818564029011 -1.876008417378137 0.05096502657785155
delta = (-2*p(3))/(p(2) - sqrt(p(2)^2 - 4*p(1)*p(3)))
    delta = 2.700810336615210e-02
x3 = x2 + delta
        x3 = 3.927008103366152
x0 = 3.8;
x1 = 4.0;
x2 = 3.9;
p = polyfit([f(x0) f(x1) f(x2)], [x0 x1 x2], 2)
         p = -0.06182184308358357 -0.5272495888523211 3.927031867462113
x3 = p(3)
       x3 = 3.927031867462113
```

7. Describe the issues with using root-finding techniques for finding all the roots of a polynomial.

Once you find one root, you must factor that root out of the polynomial using an algorithm such as synthetic division. Unfortunately, this introduces errors into the remaining coefficients, so the second root you find will only be the root of an approximation of the remaining polynomial. This accumulation of error will mean that subsequent roots will be accurate.