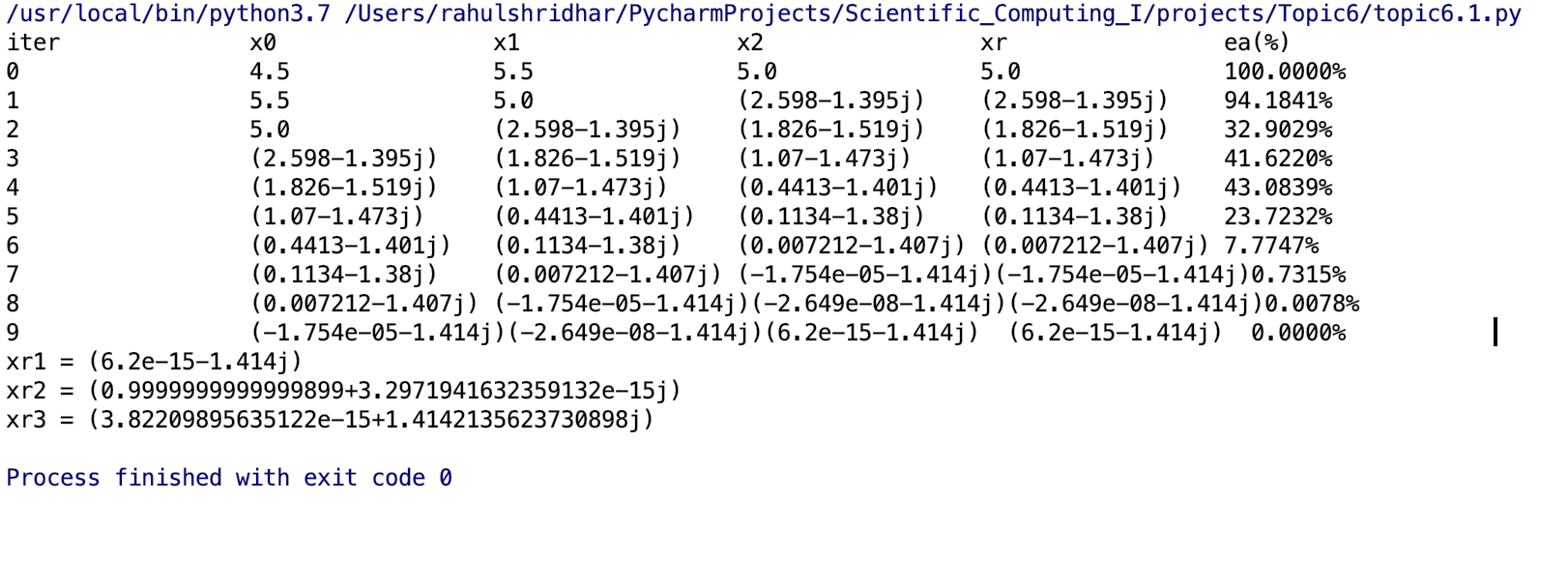
Hands On Topic 6

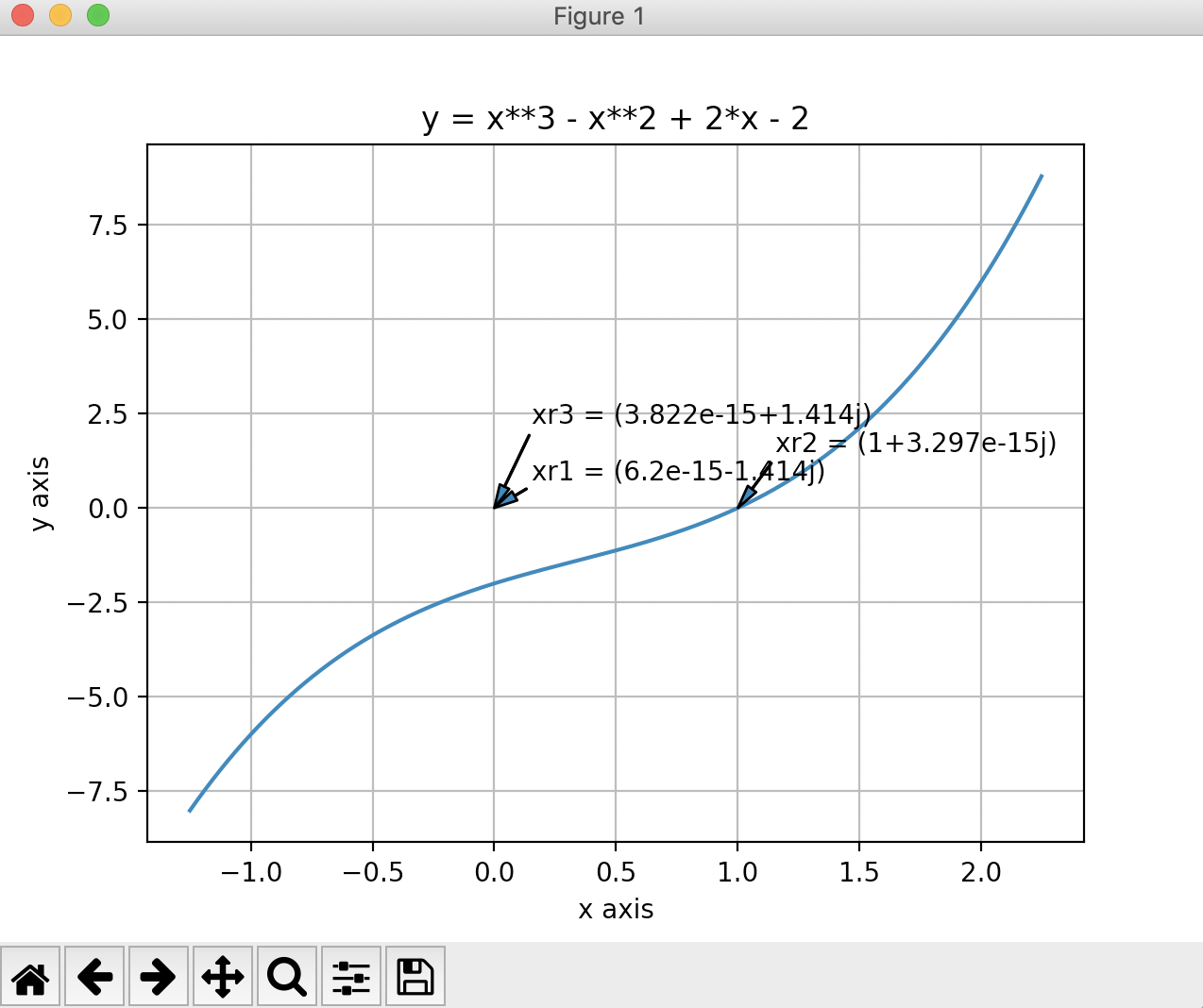
For each of the following polynomials, apply either the Muller's method or the Bairstow's method to find all real or complex roots. Analyze and contrast performance between the two methods. Plot the functions to choose root guesses appropriately. You can apply any simplifications you deem appropriate prior to applying the numerical methods.  
  
1. f(x) = x3 - x2 + 2x - 2

Solution:

f(x) = x3 - x2 + 2x – 2

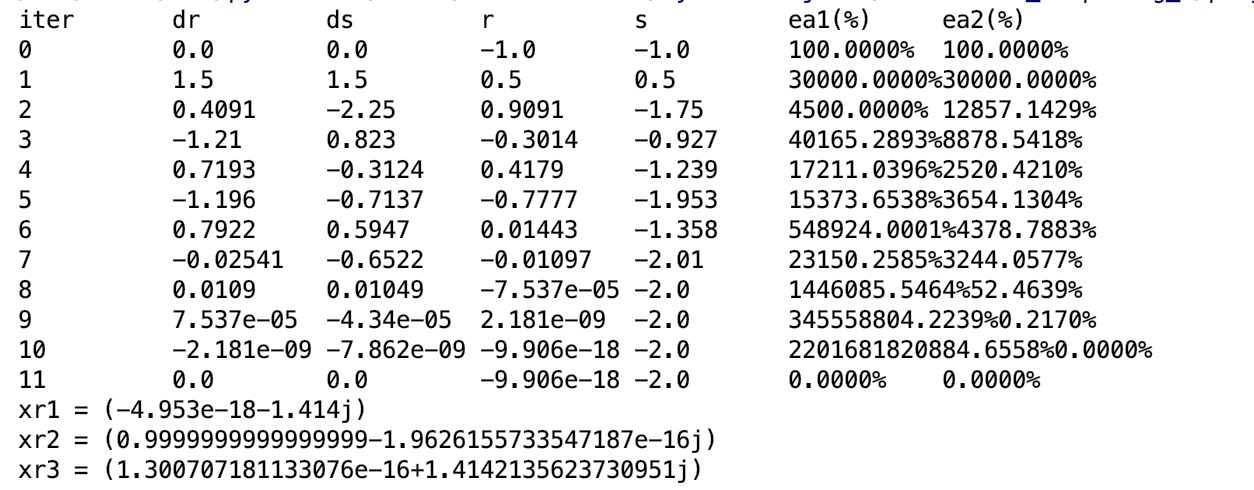
Program topic6.1.py provides the solution to the above equation by Muller’s method. This program finds all the roots of the cubic equation.

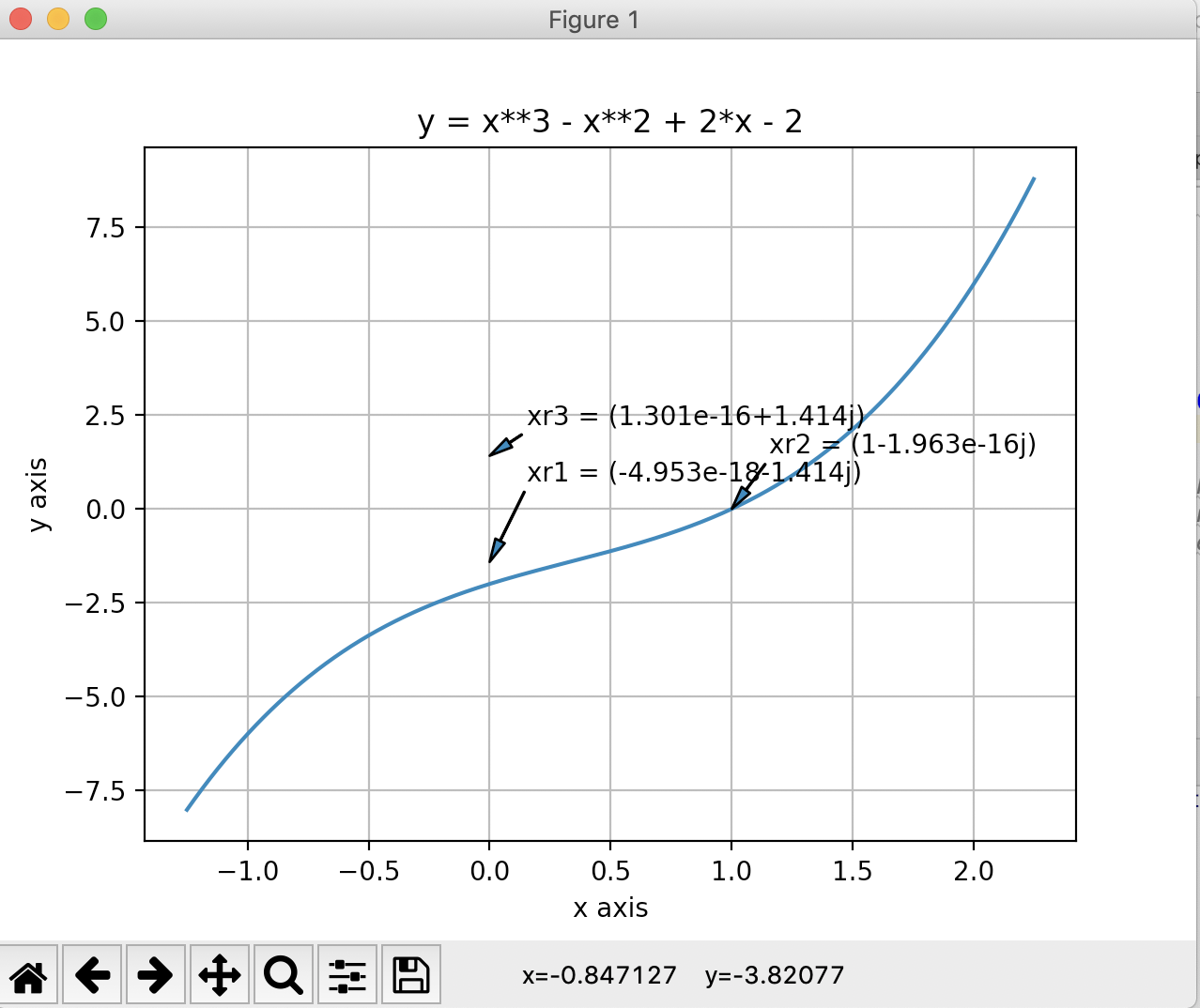




As can be seen above, the Muller’s method converges to the first root in 10 iterations. I used this root to deflate the original polynomial, hence making it a quadratic equation. I used my modified version of the quadratic function to find the remaining two roots. The roots of the equation are as follows:- xr1=-1.414j, xr2=1.414j and xr3 = 1.

Program topic-6.1.b.py solves the above cubic equation by Bairstow’s method to find all its roots.

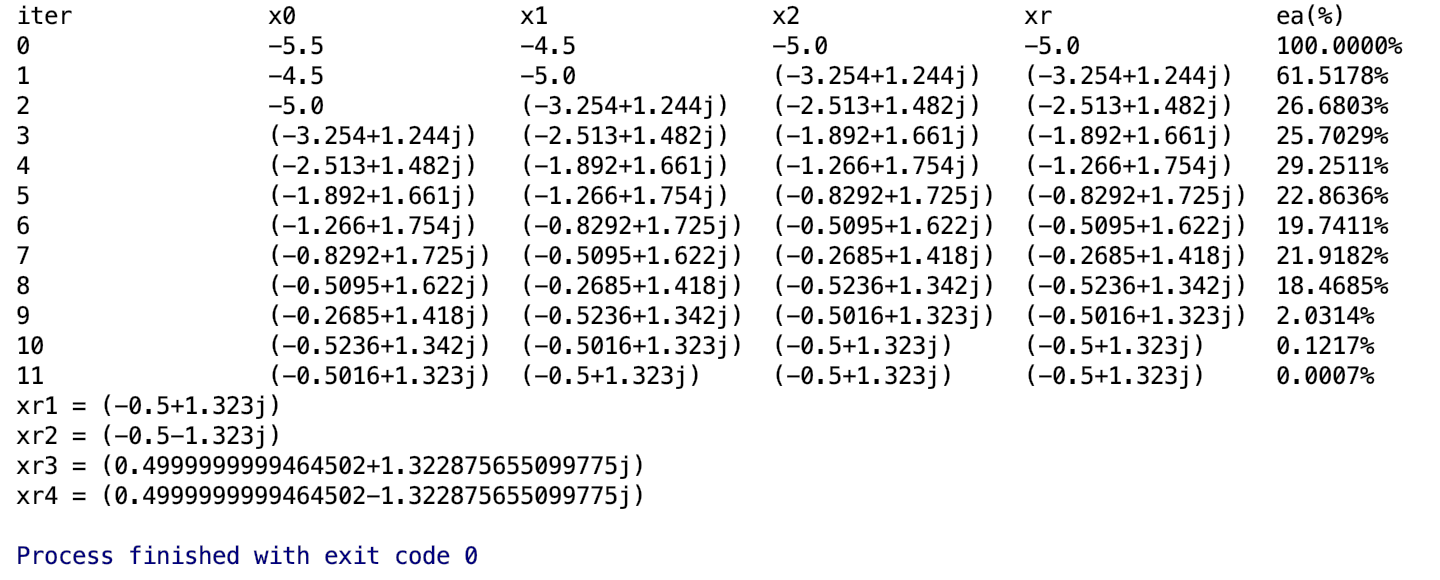




From above, we can conclude the Bairstow’s method converges to the first two roots in 10 iterations. Whereas Muller’s method finds just 1 root in 10 iterations. So I think that Bairstow’s method is the better method to adopt for higher order equations. The roots of the equation are as follows:- xr1=-1.414j, xr2=1 and xr3 = 1.414j.

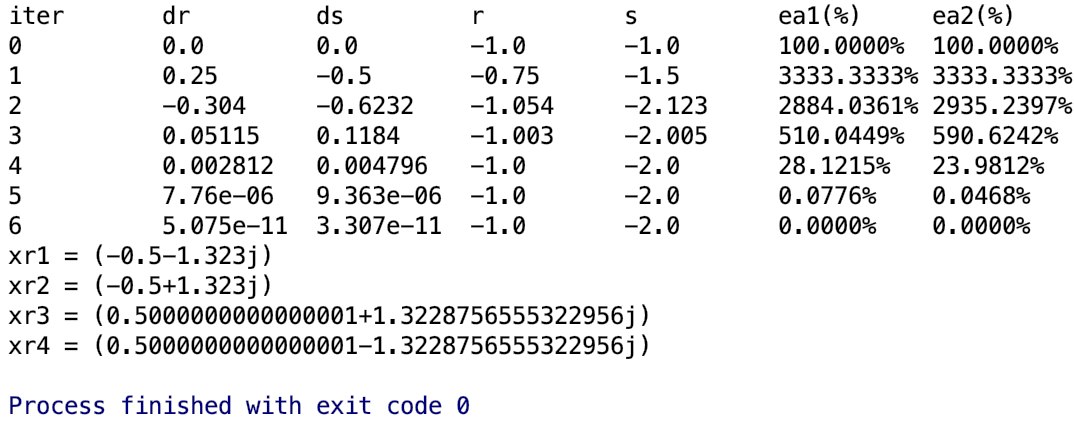
2. f(x) = 2x4 + 6x2 + 8

topic-6.2.py finds the roots of the above equation using Muller’s method. It finds the solution in 11 iterations.



After finding the first root, I find the conjugate of this root. Then I deflate the original polynomial by using the above two roots. This leads to a quadratic equation which can be easily solved using my modified quadratic function. The roots of this equation are as follows:- xr1=-.5+1.323j, xr2=-.5-1.323j, xr3=0.5+1.322j and xr4=0.5-1.322j.

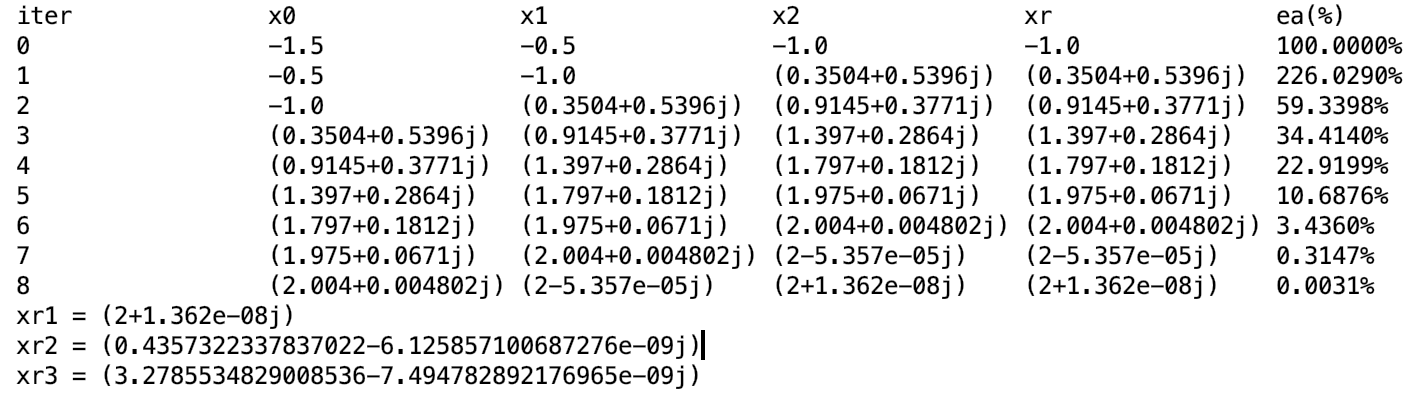
topic-6.2.b.py finds the roots of the above equation using Bairstow’s method.

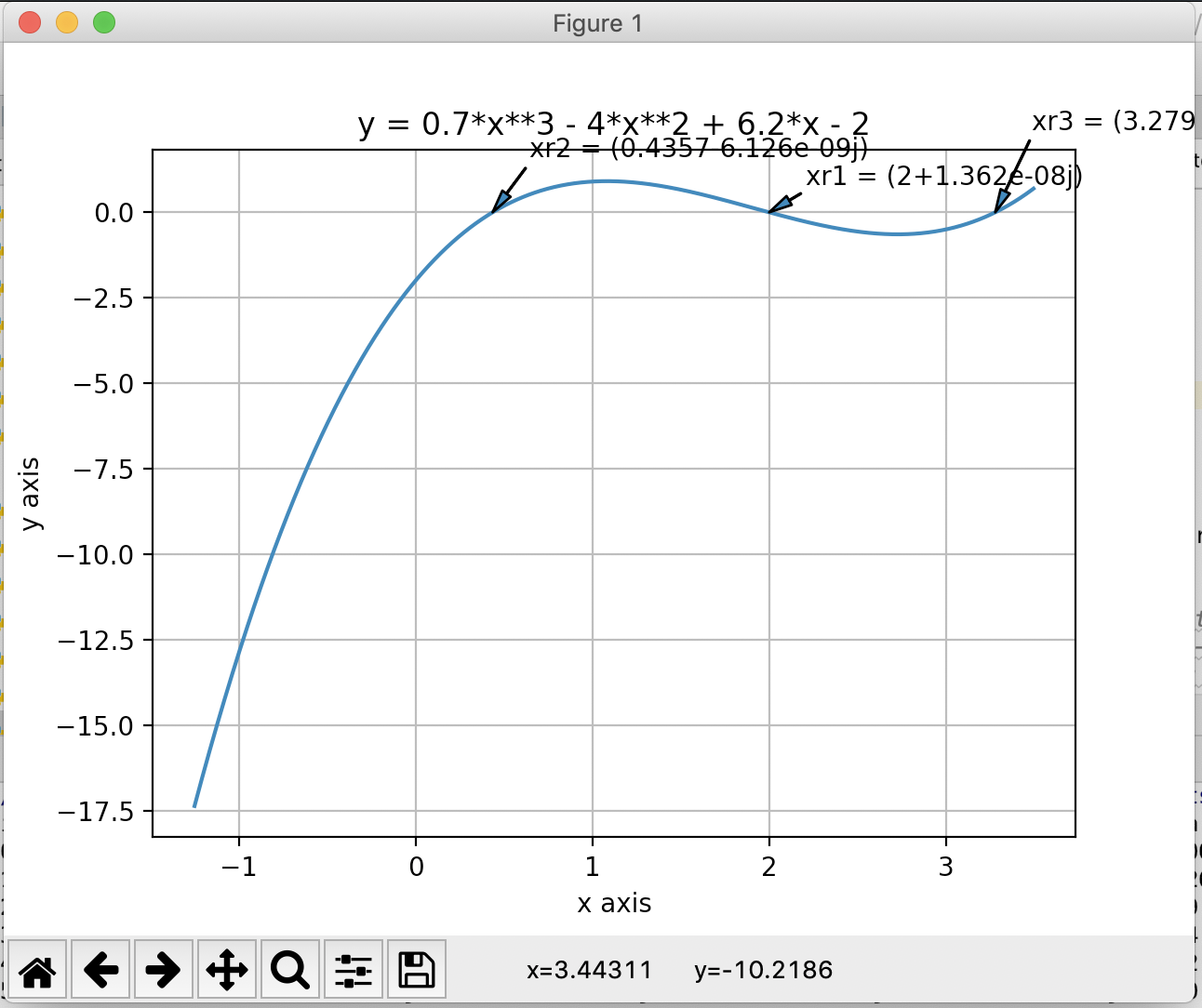


Bairstow’s method performs exceptionally for this problem. It finds the first two roots in 7 iterations which is quicker than Muller that takes 11 iterations to find the first root. After finding the first two roots, I deflate the polynomial using these two roots and solve the resulting quadratic equation.

3. f(x) = -2 + 6.2x - 4x2 + 0.7x3

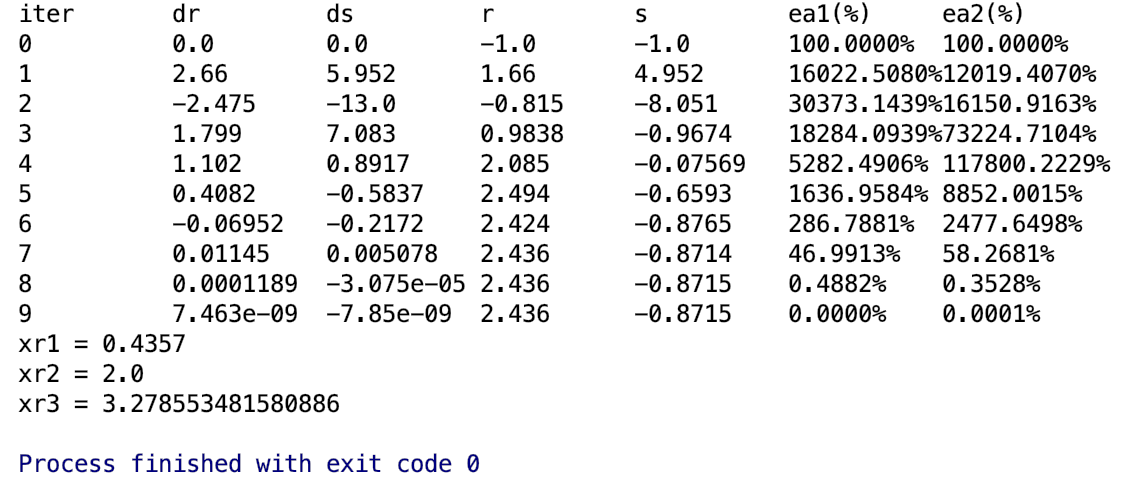
topic-6.3.py contains the solution to the above cubic equation using Muller’s method. It finds all the roots of the cubic equation.

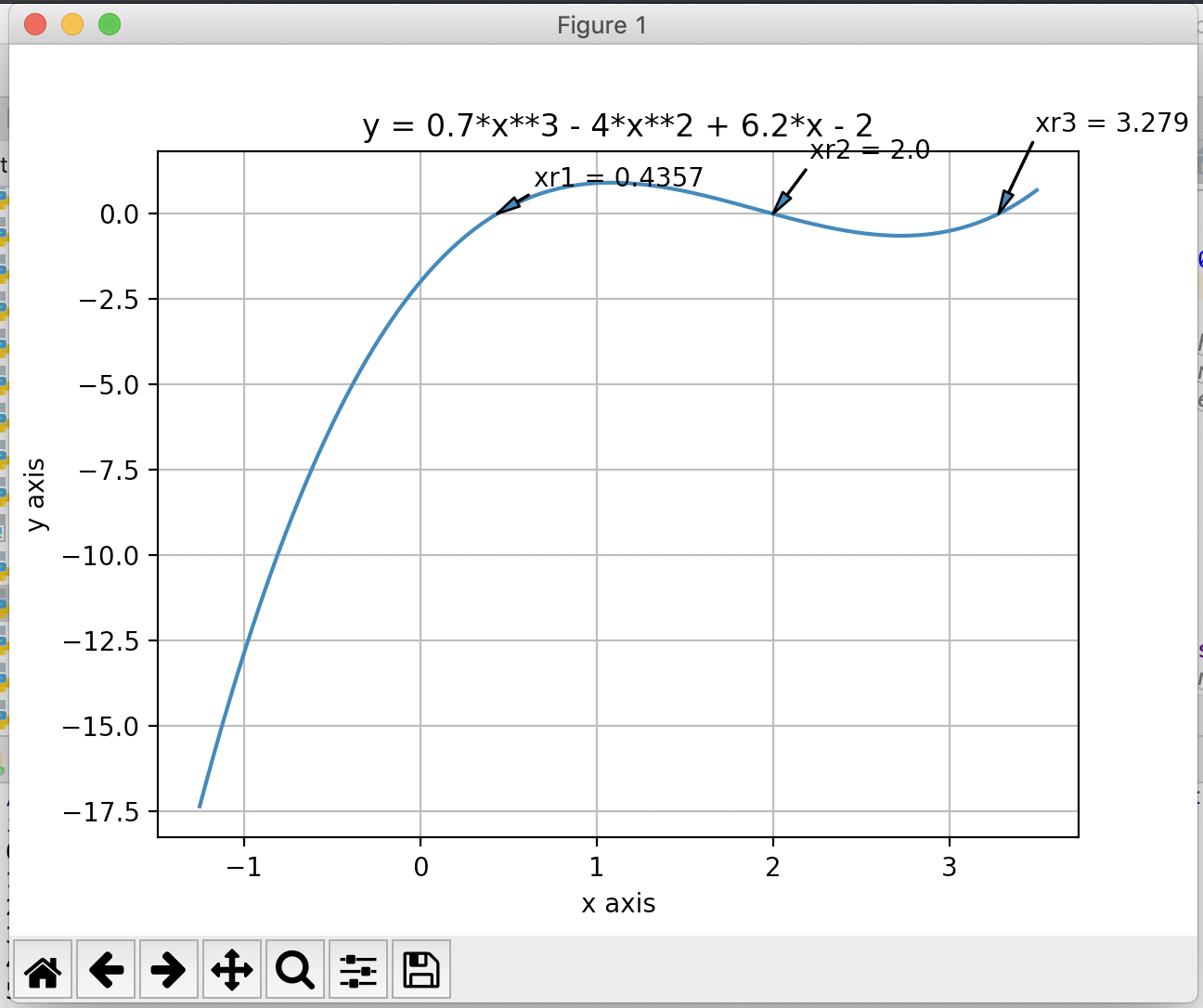




The Muller’s method converges to the first root in 9 iterations. The roots of the cubic equation are as follows:- xr1=2, xr2=0.438 and xr3=3.27.

topic-6.3.b.py solves the above equation using Bairstow’s method.



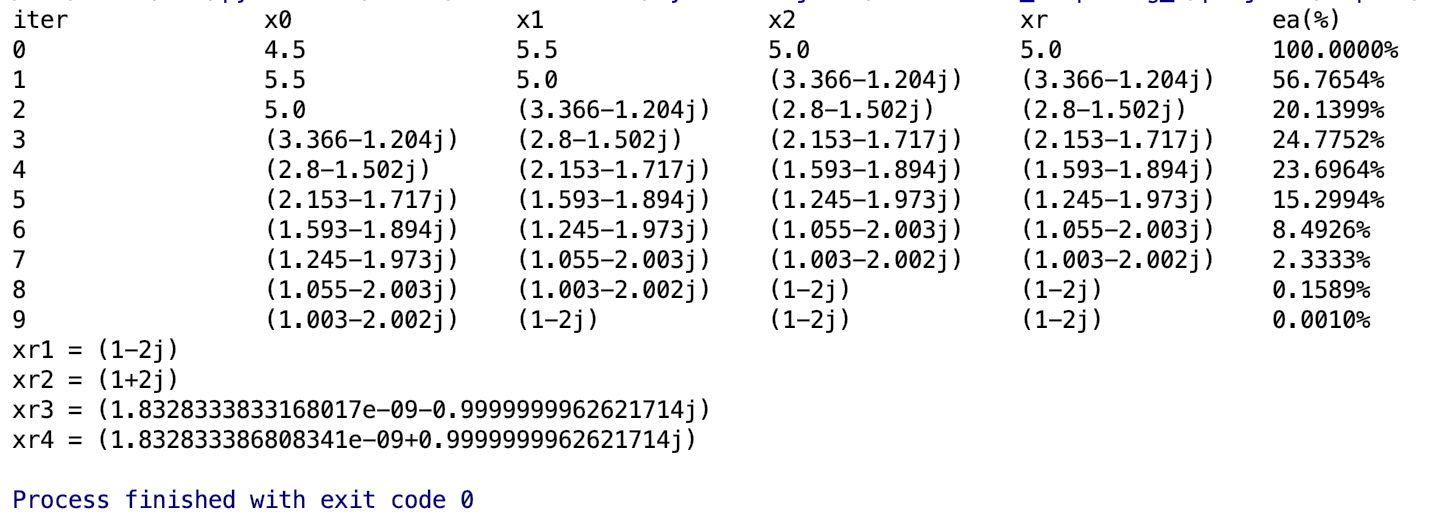


The Bairstow’s method finds the solution of the cubic equation in 9 iterations. I used the first root to deflate the polynomial into a quadratic equation. Then I found the remaining roots of the equation. However I could have divided the original equation with an equation of the form (x-xr1)(x-xr2) to find the third equation. It looks like Bairstow is faster than the Muller’s method since it employs the same number of iterations to predict double the number of roots. The final roots are as follows:-

xr1 = 0.43, xr2=2, xr3=3.27

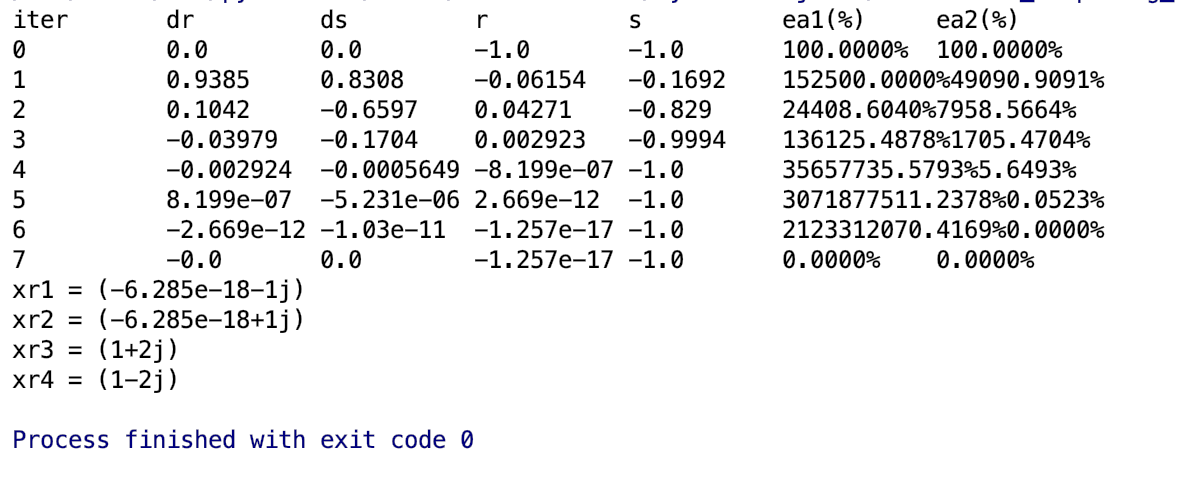
4. f(x) = x4 - 2x3 + 6x2 - 2x + 5

topic-6.4.py finds the roots of the equation using Muller’s method. It finds the first root in 10 iterations.



After finding the first root, I find the conjugate of this root. Then I deflate the original polynomial by using the above two roots. This leads to a quadratic equation which can be easily solved using my modified quadratic function. The roots of this equation are as follows:- xr1=1-2j, xr2=1+2j, xr3=-j and xr4=j.

topic-6.4.b.py solves the above equation using Bairstow’s method.



The Bairstow’s method finds the solution of the cubic equation in 8 iterations. I used the first two roots to deflate the polynomial into a quadratic equation. Then I found the remaining roots of the equation. It looks like Bairstow is faster than the Muller’s method since it requires less number of iterations to predict double the number of roots. The final roots are as follows:-

xr1 = -j, xr2=j, xr3=1+2j, xr4=1-2j