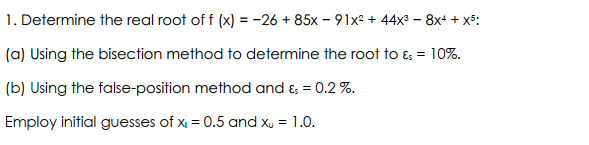
Lê Thanh Phương Nam

ITITWE19025

Report Lab 2 TMC



Step:

1. Import libs (numpy, math)
2. Create bisection function:

* Input(function, lower, upper, Es(tol))
* Output: c, #iteration

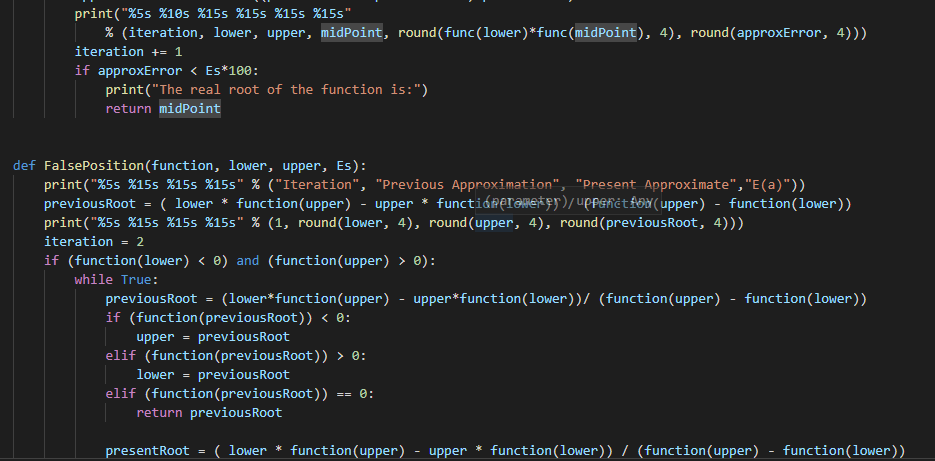
1. Create false-position function

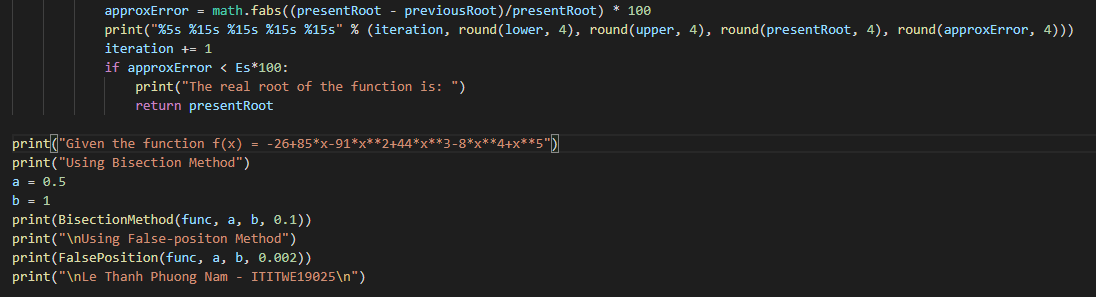
* Input(function, lower, upper, Es)
* Output: c, #iteration

1. Define the given expression
2. Call functions bisection, false-position and print the results out
3. Assessment the results

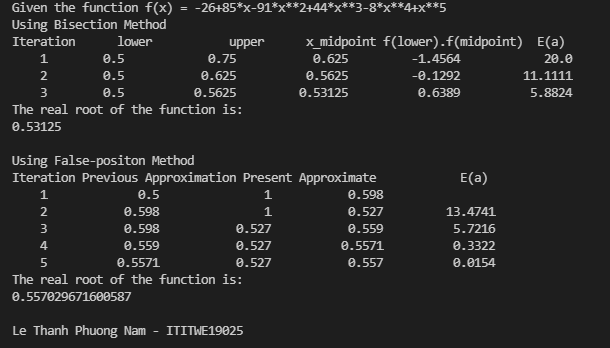
Code:

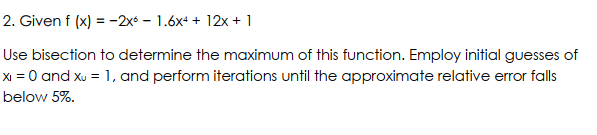






Result:



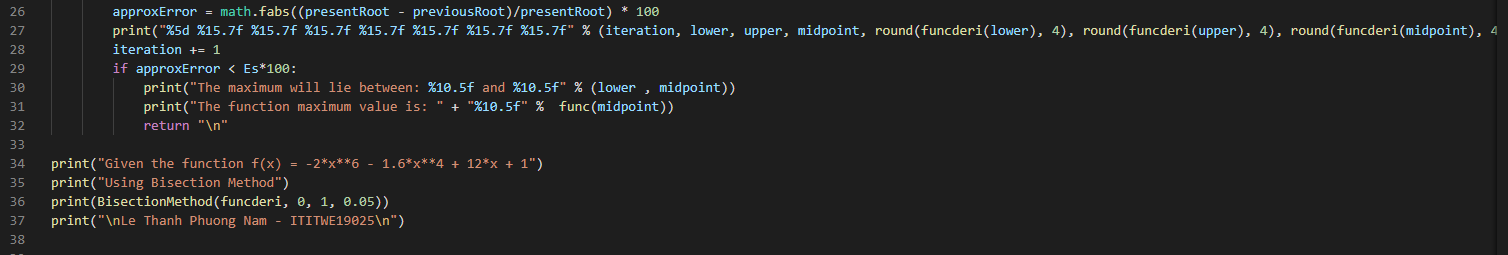


Step:

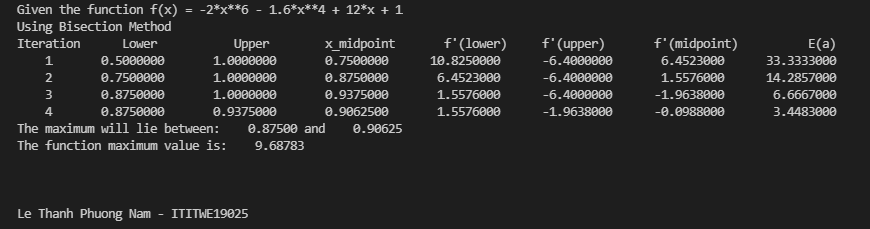
1. Import libs (numpy, math)
2. Define the given function
3. Define the given function 1st derivative
4. Create bisection function:
   * 1. Input(function, lower, upper, Es(tol))
     2. Output: c, #iteration
5. Calculate
6. Print the results

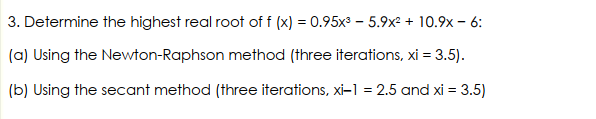
Code:





Result:

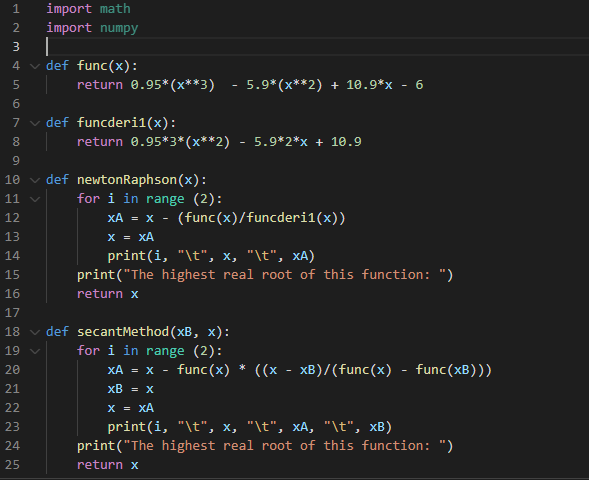


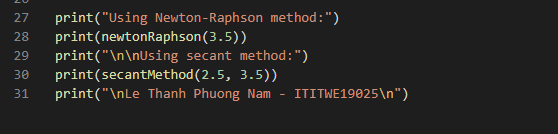


Step:

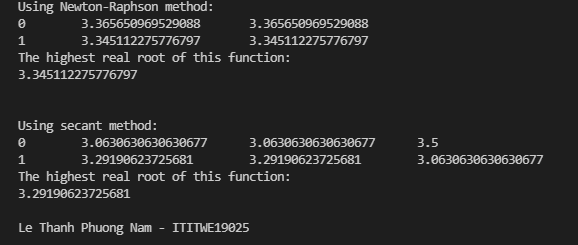
1. Import libs (numpy, math)
2. Define the given function
3. Define the given function 1st derivative
4. Create Newton-Raphson method function
5. Calculate Newton-Raphson method function
6. Create Secant method function
7. Calculate Secant method function
8. Print the results

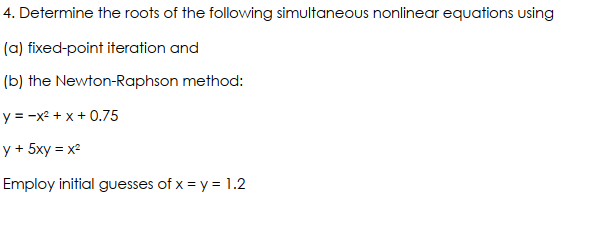
Code:





Result:

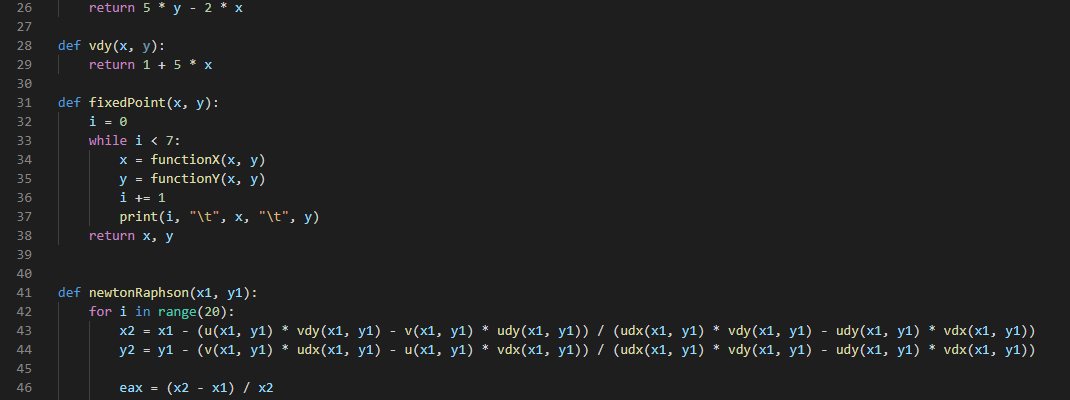
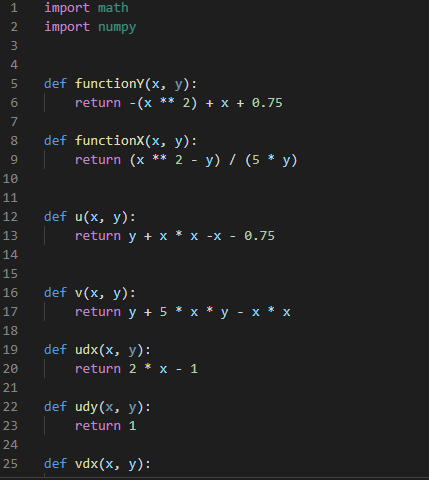


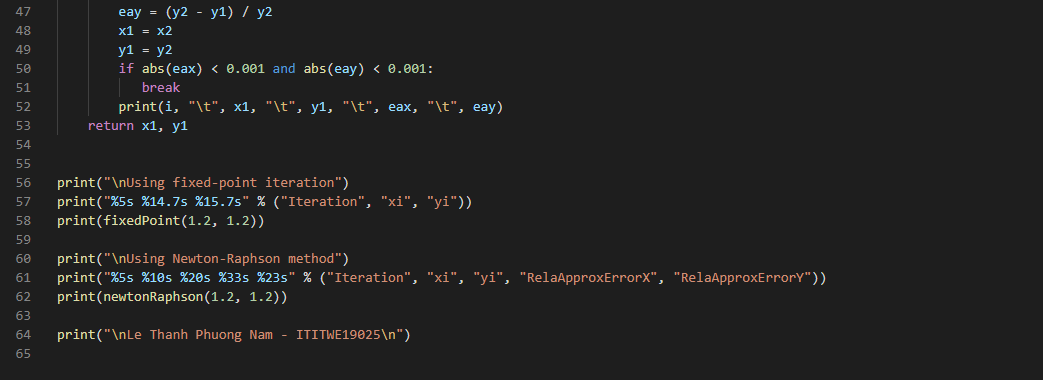


Step:

1. Import libs (numpy, math)
2. Define the given functionY, functionX, u(x, y), v(x, y), udx(x, y), udy(x, y), vdx(x, y), vdy(x, y)
3. Create Fixed-point function
4. Calculate Fixed-point method function
5. Create Newton-Raphson method function
6. Calculate Newton-Raphson method function
7. Print the results

Code:





Result:

