5633A - Numerical Methods for High-performance Computing

Programming Assignment 3

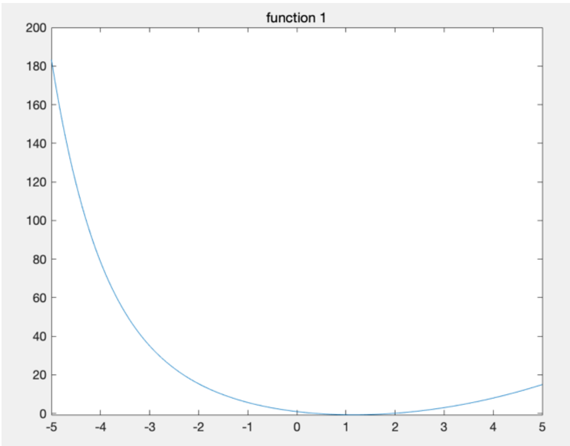
Min Zhang

Question 1:

A)

Functions 1:

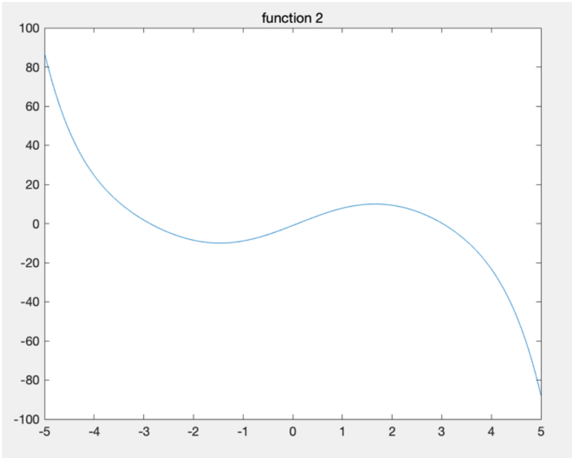
$$e^{-x} = x(2-x)$$



	Bisection method	Newton's method
X0	0.416382	0.416403
X1	1.924072	1.924119

Functions 2:

$$10sin(x - \frac{1}{10}) = \frac{x^7}{1000}$$



	Bisection method	Newton's method
X0	-2.877506	-2.877506
X1	0.100006	0.100006
X2	3.013786	3.013786

B) Non-plot solution:

Newton's method and the Bisection method make use of a general principle of numerical analysis. When trying to solve a problem, if there is no direct or simple solution, another problem which is easier to solve is used to approximate the original problem. Both numerical methods replace the solution of f(x)=0 with a very easy linear equation to solve the root problem. When we cannot use plot method. We can follow the general guidelines. When dealing with a problem with a differentiable function f(x), approximate f(x) with a linear function, and get a similar problem.

Question 2:

A)

Newton's interpolation method is based on the difference quotient formula:

$$F[x1,x2,\ldots xn] = (f[x1,x2,\ldots xn-1]-f[x2,x3,\ldots xn]) / (x0 - xn)$$

Newton interpolation formula is obtained:

$$N(x)= f(x0)+f[x0,x1](x-x0)+f[x0,x1,x2](x-x0)(x-x1)+...+f[x0,x1,x2,..xn](x-x0)\ldots(x-xn-1)$$

Newtoninterp.m's code performs the difference quotient operation of the function in line 5-9 of the for loop, and saves the result in the matrix Nd. Where the ith column of nd represents the ith difference quotient of f(x). According to Newton interpolation formula, line 19 of the code represents the calculation result of Newton interpolation.

Output for interpolated polynomial on function 1 using 2,3,4 points (bisection method) :

points	[0.5 1.5]	[0.5 1.5 2]	[0.5 1.5 2 2.5]
X0	b1_1 = 0.1260	b1_1 = 0.4116	b1_1 = 0.4146
X1	b1_2 = 2.5000	b1_2 = 1.9251	b1_2 = 1.9242

Output for interpolated polynomial on function 1 using 2,3,4 points (bisection method) :

points	[-3 0]	[-3 0 3]	[-3 0 3 6]
X0	b2_1 = -1.0815	b2_1 = -1.6161	b2_1 = -2.9461
X1	b2_2 = 2	b2_2 = 2	b2_2 = 0.0632
X2	b2_3 = 5	b2_3 = 2.7983	b2_3 = 3.0066

It is observed that in the two equations, the error between the zero point of the interpolation polynomial and the actual zero point is closer as the interpolation point increases. Therefore, when the interpolation points are 4 points, it is best to approximate the solution of the actual equation with the interpolation polynomial. Since the interpolation polynomial is approximated by the selected interpolation point, when the interpolation node is selected as the actual zero point of the equation or near the zero point, the result of the polynomial fitting will be closer to the actual situation and the effect will be the best

Functon1	Function2
root_1_2 = 0.4167	root_2_2 = 0.1000
root_1_3 = 1.9241 0.4164	root_2_3 = -2.8800 0.1000
root_1_4 = 31.8072 1.9241 0.4164	root_2_4 = 3.0137 -2.8774 0.1000

In order to verify the conclusion, this paper selects the actual zero point of the equation as the interpolation point for polynomial fitting. The obtained polynomial zero point is shown in the figure. It is observed that the calculated result is very close to the actual zero point, which further demonstrates the accuracy of the argument in this paper.

B)

If the degree of interpolation polynomials is increased properly, the accuracy of calculation results may be improved. However, the higher the frequency, the more nodes participate in the interpolation. The accuracy of the result can be improved by increasing the number of interpolation polynomials. However, the higher the number of times, the more nodes are involved in interpolation:

(1) Large amount of calculation;
(2) The curve of interpolation function oscillates violently on part of the interval (at both ends), and the truncation error of interpolation polynomial/calculation remainder term is larger.

Runge phenomenon indicates that encrypted nodes cannot guarantee that the interpolation polynomial obtained can better approximate F (x), and the effect of higher-order interpolation is not necessarily better than that of lower-order interpolation. When polynomial nodes are uniformly distributed, their endpoints tend to be maximum, which is a manifestation of runge phenomenon. And chebyshev nodes compensate for the size of polynomials to some extent on the interval. In general, Chebyshev interpolation is a specific optimal point spacing selection method. Compared with Lagrange interpolation and Newton difference quotient formula, chebyshev interpolation can avoid the appearance of runge phenomenon more effectively and fit the original function better. In other words, its result prediction is relatively more accurate.