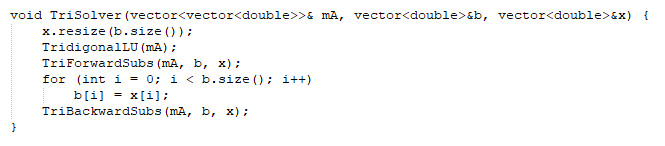
수치 해석학 과제4

1. About code

I made the Assignment4 class to solve the Exercise2.5.4. In Exercise2.5.6 (c), the problem is to interpolate the Runge function f(x) = 1 / (1+x^2) by the natural cubic spline. In Exercise2.5.6 (d), the problem is to interpolate an exponential function f(x) = exp(0.8\*x) by the natural cubic spline.

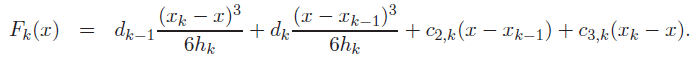
First, the function **void makeLinearSystem(double(\*f)(double x), const vector<double>& interval, int order, vector<vector<double>>& mA, vector<double>&vF, vector<double>& points, vector<double>& y, double& mesh\_size)** initialize the linear system Ax=k for the natural cubic spline. More precisely, the matrix mA and the vector vF are update so that they are satisfied the answer of Exercise2.5.5 (a). Here, mA is an (order+1)\*(order+1) matrix and vF is a vector of length (order+1) so that d=(,,…,) is the solution of Ax=k where A=mA and k=vF. The input points and y change into the vector of uniform grids and the vector of function values respectively for the given interval and given order. The mesh\_size changes to the mesh size of the uniform grids.

Second, **void TriSolver(vector<vector<double>>& mA, vector<double>&b, vector<double>&x)** solves mA\*x=b by Tridiagonal matrix algorithm which is a type of Gaussian Elimination solver. I capsulized the function **Trisolver** as follow:



The function **Trisolver** is just a combination of LU-decomposition, Forward-substitution and Backward-substitution with the property of the tridiagonal matrix.

The function **double CubicSpline(vector<double>& d, vector<double>& y,vector<double>& points, double mesh\_size, double x)** returns the value obtained from the following equation :



,where

Here, =points[k] and and .

Similarly, the function **double CubicSpline\_derivative(vector<double>& d, vector<double>& y, vector<double>& points, double mesh\_size, double x)** returns the value obtained from the following equation:



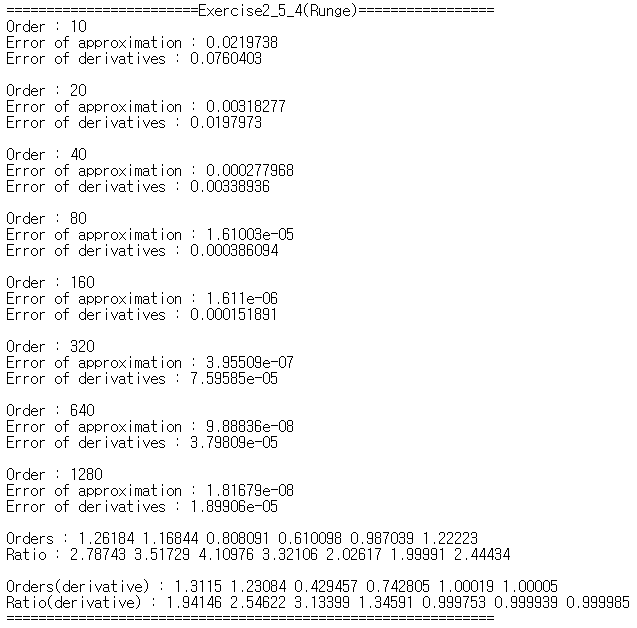
, where

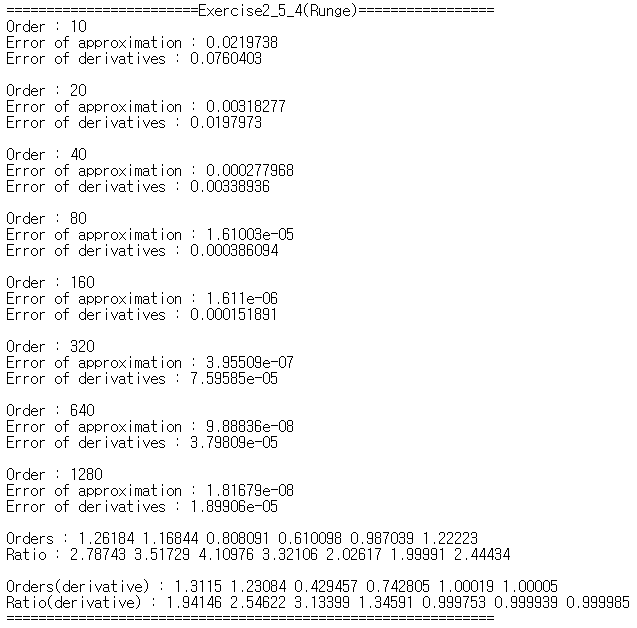
Finally, **void Exercise2\_5\_4\_Runge()** and **Exercise2\_5\_4\_Exponential()** show some result of cubic spline interpolation of the Runge function and the exponential function with some orders.

The graphs of the results are in the Section 3 and 4.

1. Runge Function

I tested the cubic spline interpolation of the Runge function with. The result of the code is as follow:





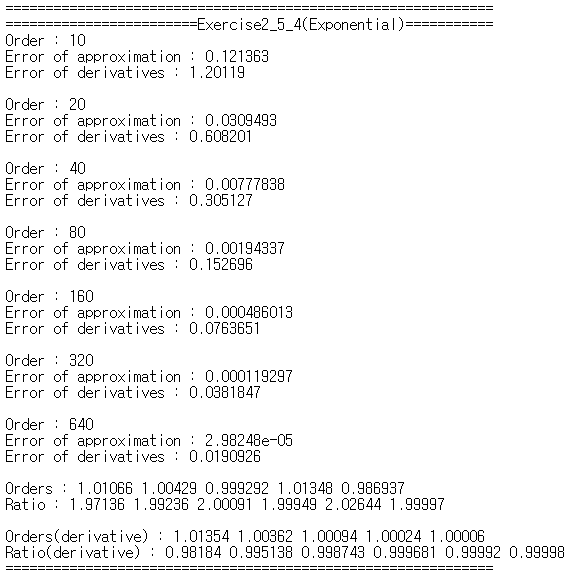
Therefore, the table of “Error of Cubic spline” is as follow:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Runge |  | ratio |  | ratio |
|  | 0.0216738 |  | 0.0760403 |  |
|  | 0.00318277 | 2.78743 | 0.0197973 | 1.94146 |
|  | 0.000277968 | 3.51729 | 0.00338936 | 2.54622 |
|  | 1.61003e-05 | 4.10976 | 0.000386094 | 3.13399 |
|  | 1.611e-06 | 3.32106 | 0.000151891 | 1.34591 |
|  | 3.95509e-07 | 2.02617 | 7.59585e-05 | 0.999753 |
|  | 9.88836e-08 | 1.99991 | `3.79809e-05 | 0.999939 |
|  | 1.81679e-08 | 2.44434 | 1.89906e-05 | 0.999985 |

As the order(n) increases, the ratio goes to 1. However, the ratio is not consistent and the orders of the infinite norms of the absolute errors are around 1 mostly.

1. Exponential Function

I tested the cubic spline interpolation of the exponential function with. The result of the code is as follow:

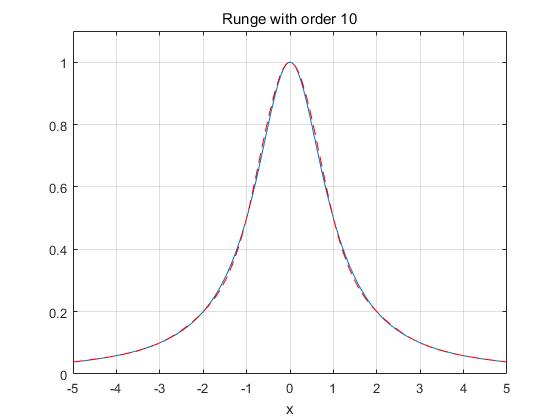


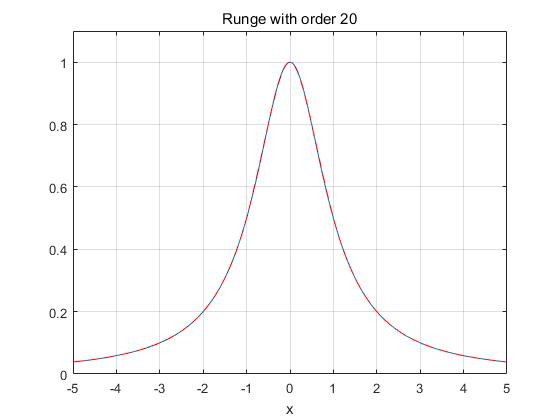
Therefore, the table of “Error of Cubic spline” is as follow:

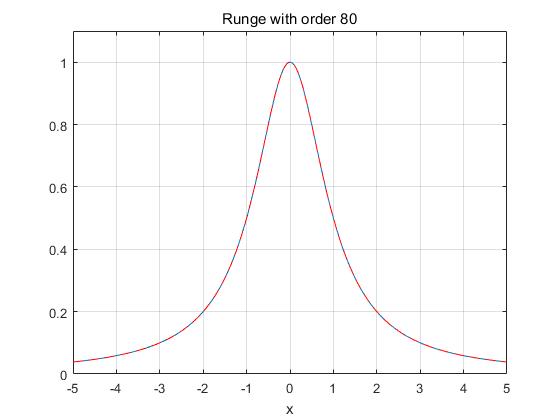
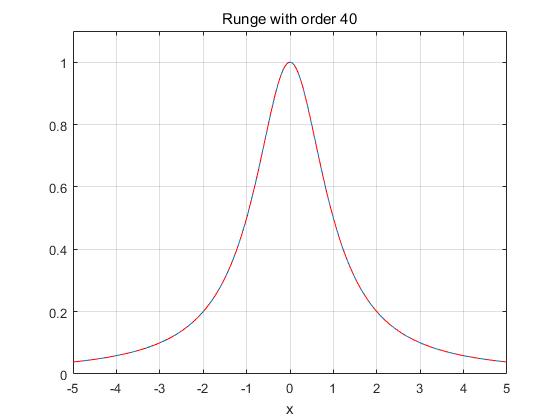
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Runge |  | ratio |  | ratio |
|  | 0.121363 |  | 1.20119 |  |
|  | 0.0309493 | 1.97136 | 0.608201 | 0.98184 |
|  | 0.00777838 | 1.99236 | 0.305127 | 0.995138 |
|  | 0.00194337 | 2.00091 | 0.152696 | 0.998743 |
|  | 0.000486013 | 1.99949 | 0.0763651 | 0.999681 |
|  | 0.00194337 | 2.02644 | 0.0381847 | 0.99992 |
|  | 2.98248e-05 | 1.99997 | 0.0190926 | 0.99998 |

As the order(n) increases, the ratio goes to 2 and the ratio goes to 1.

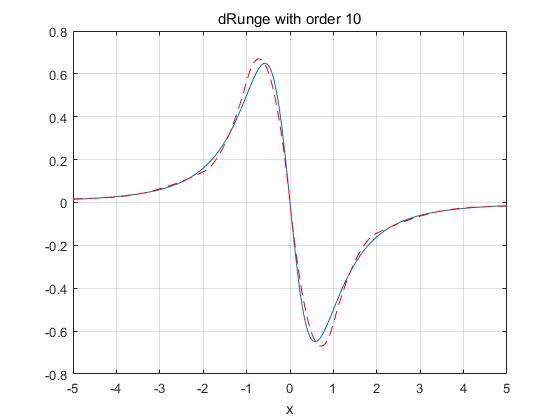
1. Graphs of Runge function
2. Approximation

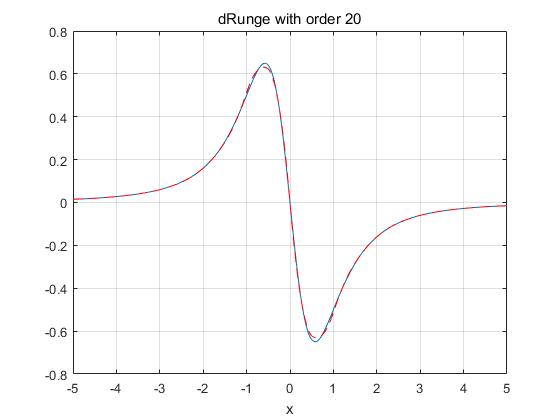


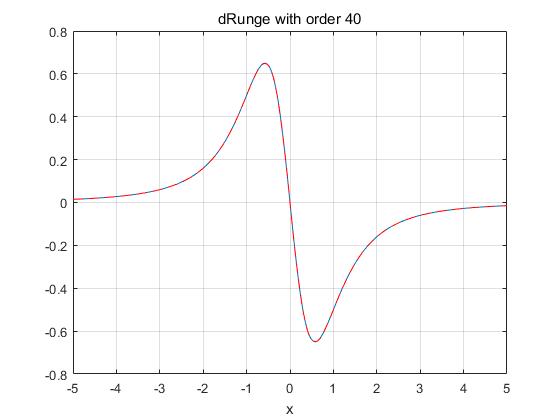


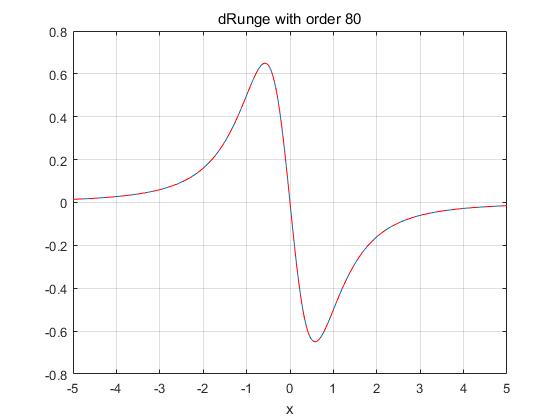


1. Derivatives

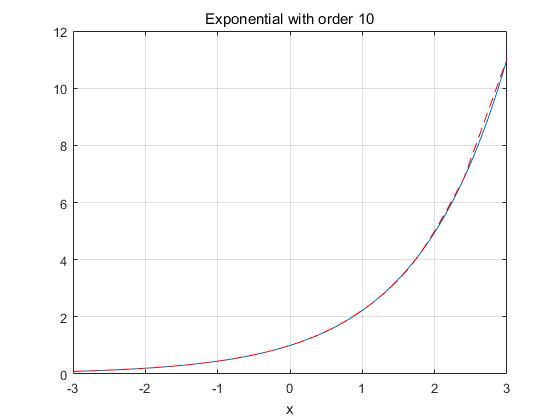


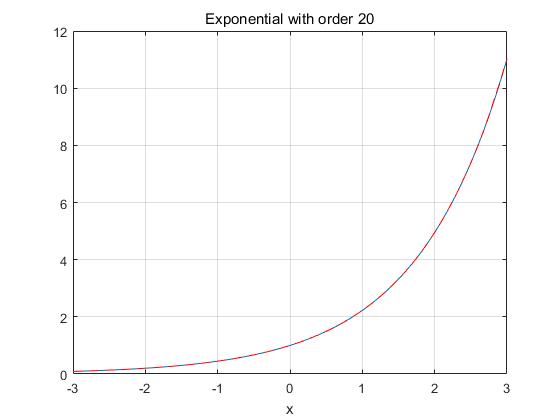


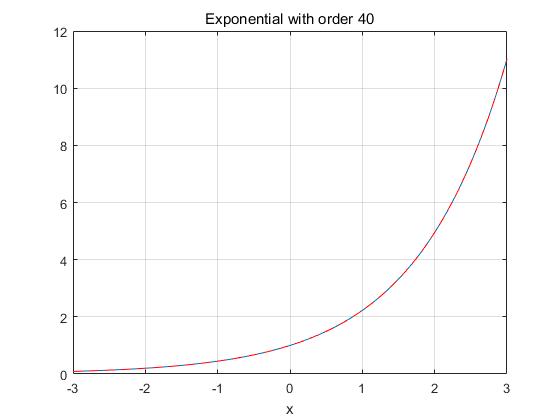


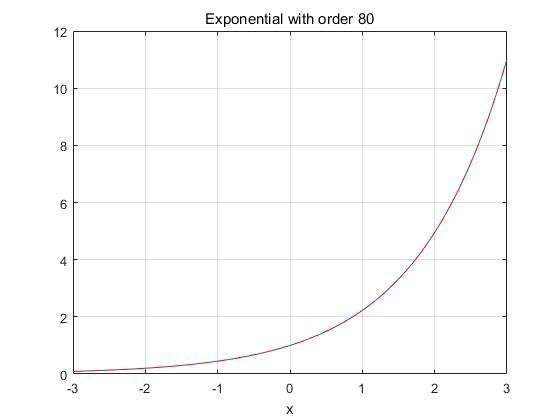


1. Graphs of Exponential function
2. Approximation









1. Derivatives

