

Second Project, MAE 185. Cubic spline determination using the improved method

Nasser M. Abbasi

May 15, 2003

Contents

| | | |
|----------|-------------------------------|-----------|
| 1 | Goal of project | 1 |
| 2 | How to run the program | 1 |
| 3 | Program Design | 2 |
| 4 | Program Output | 2 |
| 4.1 | First example run | 2 |
| 4.2 | Second Example run | 5 |
| 4.3 | Third Example run | 7 |
| 4.4 | Fourth Example run | 9 |
| 5 | appendix | 11 |
| 5.1 | source code | 11 |

1 Goal of project

The goal of this project is to write a program to find the all the cubic splines once coordinates are given.

2 How to run the program

This program consists of 2 matlab files. The first file is called `nma_proj_2.m` which is the driver which calls a second file called `nma_spline.m` to do the actual calculation of the splines.

To run the program, simply copy these 2 files (contained in the floppy disk attached) to your MATLAB *work* folder on your C: drive. The MATLAB work folder will be located under the main MATLAB folder. The name of the MATLAB\ main folder depends on the version of MATLAB you have installed.

For example, for MATLAB 6.5, it is called

C:\MATLAB6p5

Once the files are copied to the work folder below the above MATLAB main folder, then start matlab itself, and from the matlab console, type the command:

nama_proj_2

Now the program will ask the user to enter the coordinates of the points. There is no limit to the number of the points that can be given. Each points is entered by typing the x value and then the y value. Each is entered on a separate line. When the user have decided that no more points needed, then simply hit the RETURN key when the program is asking for the next x coordinates of a new point.

3 Program Design

This program uses the improved cubic spline determination method, which requires $n - 1$ equations to be solved. End conditions for the first and last splines uses the natural spline conditions. I have implemented this algorithm in the file `nama_spline.m` For solving the generated $n - 1$ equations for the unknowns (the coefficients a, b, c, d).

4 Program Output

The program have 2 types of output. One output is send to the MATLAB console. It will print the cubic splines generated, and the x range over which each spline is defined.

The second type of output is a graphical output. A plot is made of all the splines, and the equation of each spline is displayed next to the spline.

Examples of theses outputs are show below to help illustrate how to use the program.

4.1 First example run

For this run, I used the points from HW 2, problem 18.1 since I know what the answer should be, this verifies the program output.

Notices how input is handled. The user hits a RETURN for the 6th point. This tells the program to terminate input and to use the first 5 points.

| x | y |
|---|-------|
| 1 | 4.75 |
| 2 | 4 |
| 3 | 5.25 |
| 5 | 19.75 |
| 6 | 36 |

```
>> nama_proj_2
```

```
Please enter the X and Y coordinates of each point separately.
```

```
Enter as many points as you wish. There is no hardcoded limit.
```

```
When done, simply hit RETURN without typing anything on the line.
```

```
x[1] ? >1
```

```
y[1] ? >4.75
```

```
x[2] ? >2
```

```
y[2] ? >4
```

```
x[3] ? >3
```

y[3] ? >5.25

x[4] ? >5

y[4] ? >19.75

x[5] ? >6

y[5] ? >36

x[6] ? >

The cubic splines are:

$+0.377049*(x-1.000000)^3 + 0.000000*(x-1.000000)^2 - 1.127049*(x-1.000000) + 4.750000$
Range is from x=1.000000 to x=2.000000

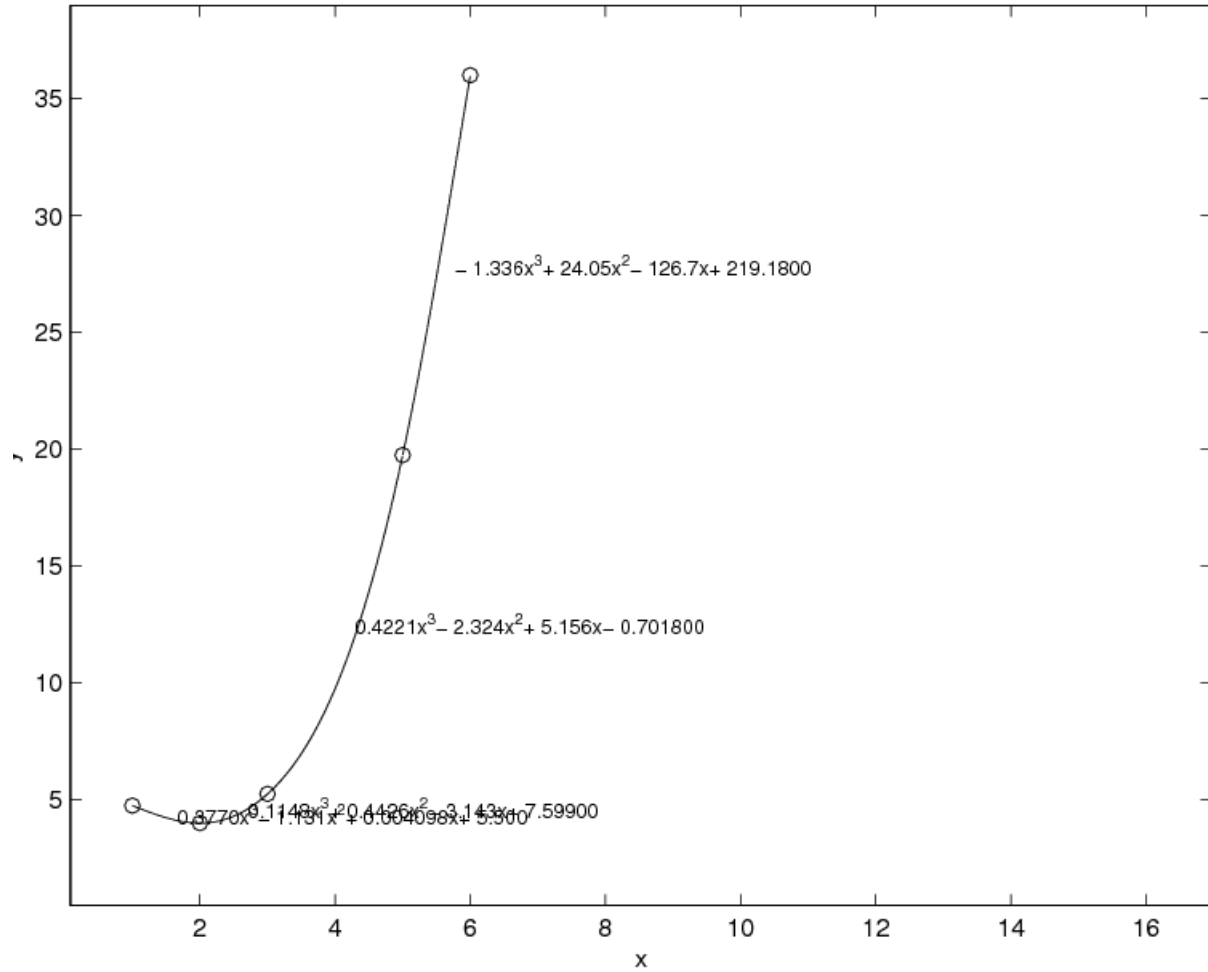
$+0.114754*(x-2.000000)^3 + 1.131148*(x-2.000000)^2 + 0.004098*(x-2.000000) + 4.000000$
Range is from x=2.000000 to x=3.000000

$+0.422131*(x-3.000000)^3 + 1.475410*(x-3.000000)^2 + 2.610656*(x-3.000000) + 5.250000$
Range is from x=3.000000 to x=5.000000

$-1.336066*(x-5.000000)^3 + 4.008197*(x-5.000000)^2 + 13.577869*(x-5.000000) + 19.750000$
Range is from x=5.000000 to x=6.000000

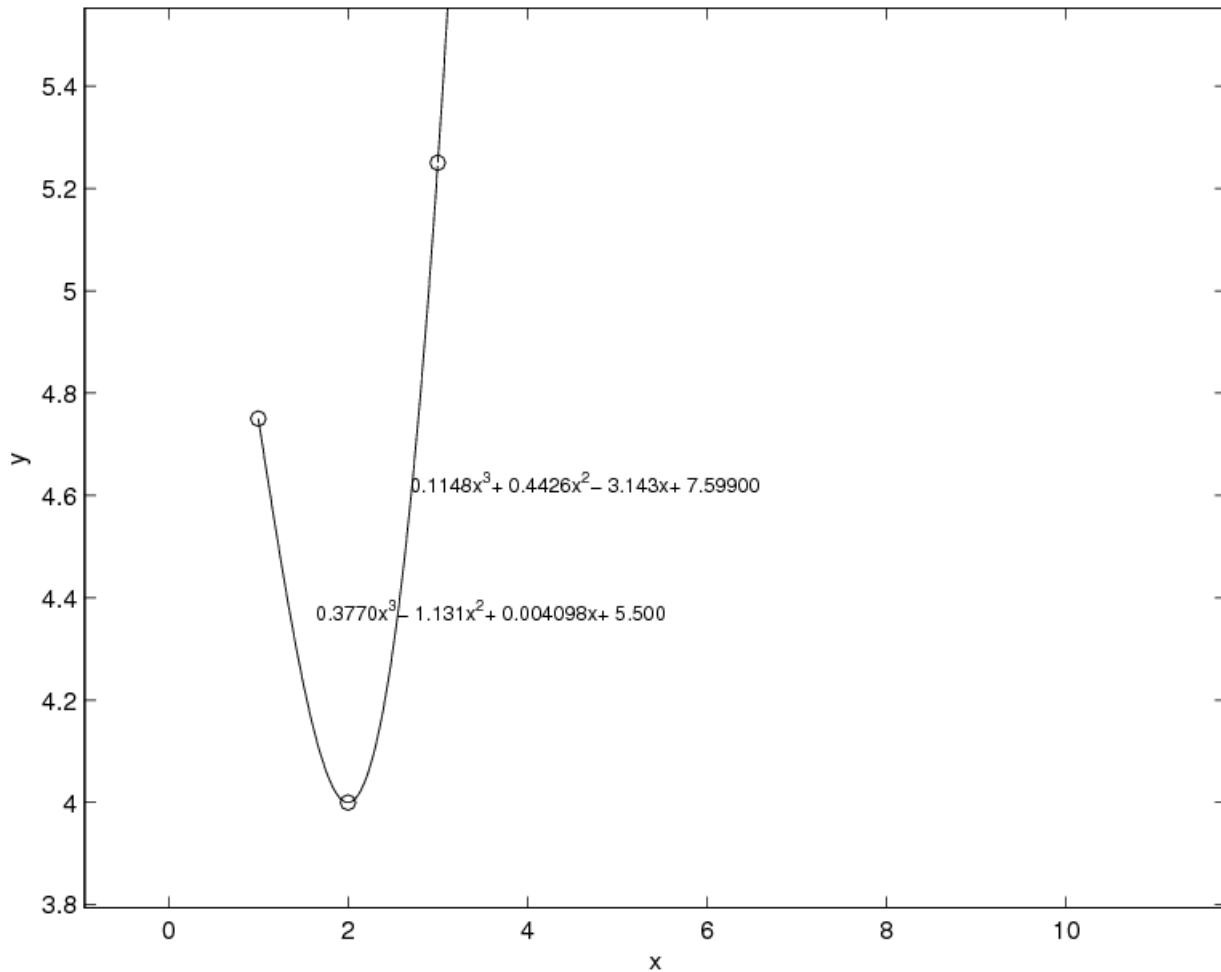
>>

Cubic splines by the improved spline method. Project 2 MAE 185



Since I display the actual spline equation on the plot, due to limitation of space, some equations might be hard to read if the other spline is close to it. Simply use the ZOOM feature in the plot window to look at the area in question closer, as shown below:

Cubic splines by the improved spline method. Project 2 MAE 185



Notice the following: In the plot, When I display the spline equations, I did a symbolic simplification on them to transform the cubic equation from the original form, such as

$$+0.377049 * (x - 1.000000)^3 + 0.000000 * (x - 1.000000)^2 - 1.127049 * (x - 1.000000) + 4.750000$$

$$0.377x^3 - 1.131x^2 + 0.004098x + 5.5$$

The above 2 equations are the same, but the second one takes less space, and hence easier to fit on the plotting screen. I have used MATLAB symbolic toolbox to do the conversion.

4.2 Second Example run

For this example, I will use the data given on page 245, of our text book (Applied Numerical Analysis). The data is

| x | y |
|------|---------|
| 0 | 2 |
| 1 | 4.4366 |
| 1.5 | 6.7134 |
| 2.25 | 13.9130 |
| | |

```
>> nma_proj_2
```

Please enter the X and Y coordinates of each point separately.

Enter as many points as you wish. There is no hardcoded limit.

When done, simply hit RETURN without typing anything on the line.

```
x[1] ? >0
```

```
y[1] ? >2
```

```
x[2] ? >1
```

```
y[2] ? >4.4366
```

```
x[3] ? >1.5
```

```
y[3] ? >6.7134
```

```
x[4] ? >2.25
```

```
y[4] ? >13.9130
```

```
x[5] ? >
```

The cubic splines are:

```
+0.382009*(x-0.000000)^3 +0.000000*(x-0.000000)^2 +2.054591*(x-0.000000) +2.000000
```

Range is from x=0.000000 to x=1.000000

```
+3.119871*(x-1.000000)^3 +1.146028*(x-1.000000)^2 +3.200618*(x-1.000000) +4.436600
```

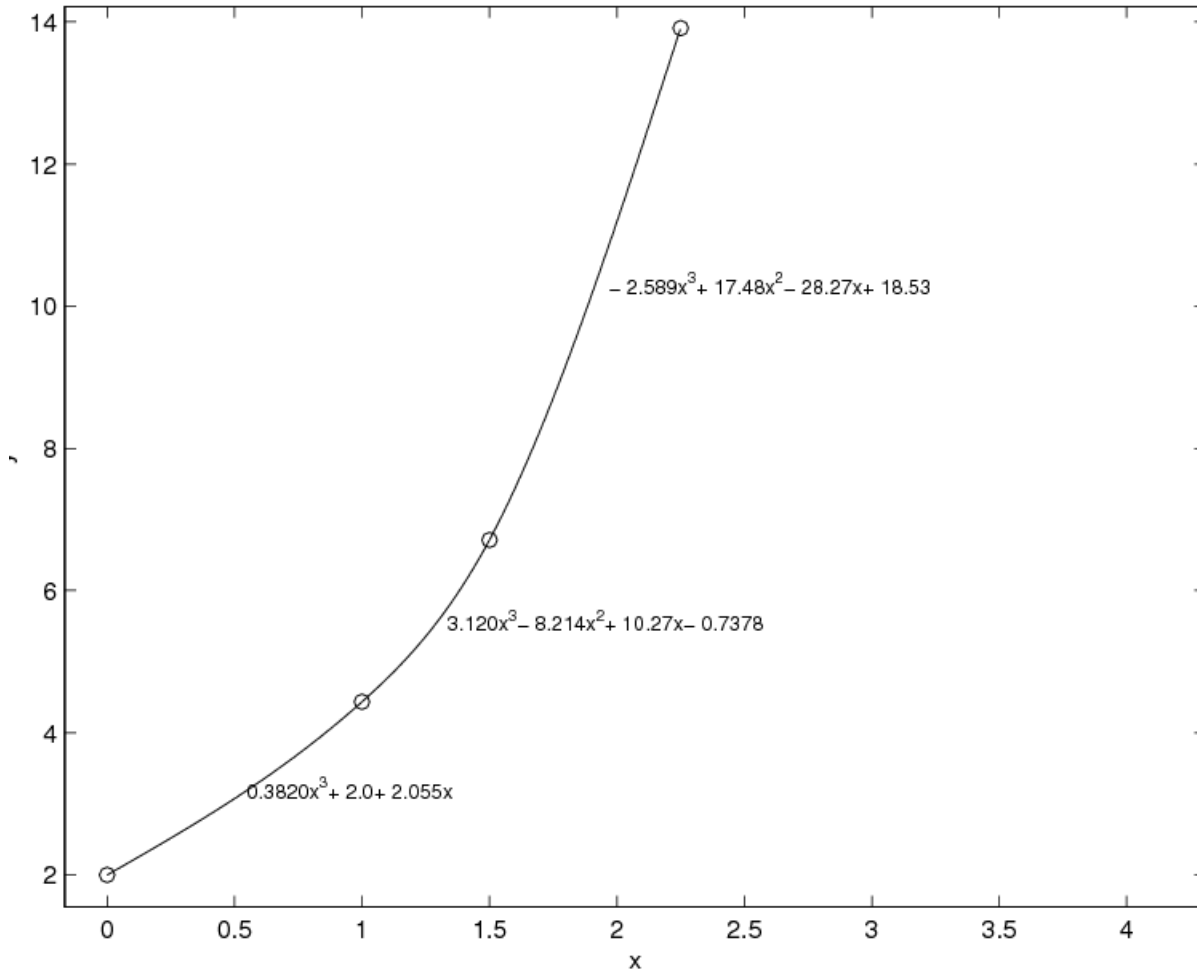
Range is from x=1.000000 to x=1.500000

```
-2.589260*(x-1.500000)^3 +5.825834*(x-1.500000)^2 +6.686549*(x-1.500000) +6.713400
```

Range is from x=1.500000 to x=2.250000

```
>>
```

Cubic splines by the improved spline method. Project 2 MAE 185



4.3 Third Example run

This example is taken from the text *Numerical methods for Engineers*, second edition, page 396

| x | y |
|-----|-----|
| 3 | 2.5 |
| 4.5 | 1 |
| 7 | 2.5 |
| 9 | 0.5 |
| | |

```
>> nma_proj_2
```

```
Please enter the X and Y coordinates of each point separately.
```

```
Enter as many points as you wish. There is no hardcoded limit.
```

```
When done, simply hit RETURN without typing anything on the line.
```

```
x[1] ? >3
```

```
y[1] ? >2.5
```

```
x[2] ? >4.5
```

y[2] ? >1

x[3] ? >7

y[3] ? >2.5

x[4] ? >9

y[4] ? >0.5

x[5] ? >

The cubic splines are:

$+0.186565*(x-3.00000)^3 + 0.000000*(x-3.00000)^2 - 1.419772*(x-3.00000) + 2.500000$

Range is from x=3.000000 to x=4.500000

$-0.214144*(x-4.50000)^3 + 0.839544*(x-4.50000)^2 - 0.160456*(x-4.50000) + 1.000000$

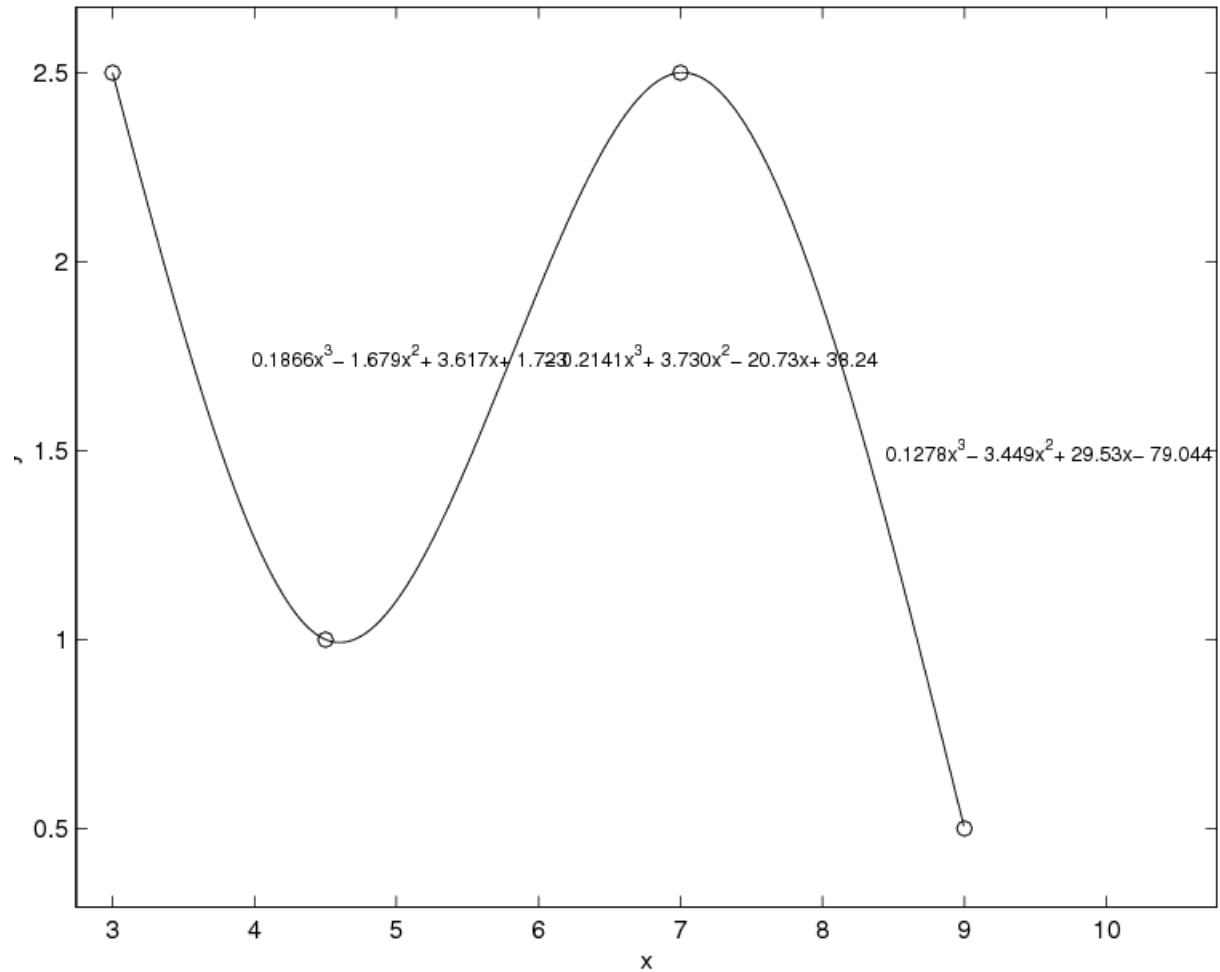
Range is from x=4.500000 to x=7.000000

$+0.127757*(x-7.00000)^3 - 0.766540*(x-7.00000)^2 + 0.022053*(x-7.00000) + 2.500000$

Range is from x=7.000000 to x=9.000000

>>

Cubic splines by the improved spline method. Project 2 MAE 185



4.4 Fourth Example run

This example I made up, to show many points being used. I used 10 points, which gives 9 splines.

```
>> nma_proj_2
```

Please enter the X and Y coordinates of each point separately.

Enter as many points as you wish. There is no hardcoded limit.

When done, simply hit RETURN without typing anything on the line.

```
x[1] ? >1
```

```
y[1] ? >2
```

```
x[2] ? >1.5
```

```
y[2] ? >3
```

```
x[3] ? >2.1
```

```
y[3] ? >4
```

```
x[4] ? >3.3
```

y[4] ? >5.5

x[5] ? >5.2

y[5] ? >1.4

x[6] ? >6.4

y[6] ? >-3

x[7] ? >9

y[7] ? >2

x[8] ? >15

y[8] ? >5.4

x[9] ? >20

y[9] ? >12

x[10] ? >24

y[10] ? >10

x[11] ? >

The cubic splines are:

$-0.343424*(x-1.000000)^3 + 0.000000*(x-1.000000)^2 + 2.085856*(x-1.000000) + 2.000000$
Range is from x=1.000000 to x=1.500000

$+0.409610*(x-1.500000)^3 - 0.515135*(x-1.500000)^2 + 1.828288*(x-1.500000) + 3.000000$
Range is from x=1.500000 to x=2.100000

$-0.464653*(x-2.100000)^3 + 0.222163*(x-2.100000)^2 + 1.652505*(x-2.100000) + 4.000000$
Range is from x=2.100000 to x=3.300000

$+0.116296*(x-3.300000)^3 - 1.450589*(x-3.300000)^2 + 0.178394*(x-3.300000) + 5.500000$
Range is from x=3.300000 to x=5.200000

$+0.939532*(x-5.200000)^3 - 0.787699*(x-5.200000)^2 - 4.074354*(x-5.200000) + 1.400000$
Range is from x=5.200000 to x=6.400000

$-0.431490*(x-6.400000)^3 + 2.594617*(x-6.400000)^2 - 1.906053*(x-6.400000) - 3.000000$
Range is from x=6.400000 to x=9.000000

$+0.065482*(x-9.000000)^3 - 0.771006*(x-9.000000)^2 + 2.835334*(x-9.000000) + 2.000000$
Range is from x=9.000000 to x=15.000000

$-0.054950*(x-15.000000)^3 + 0.407678*(x-15.000000)^2 + 0.655368*(x-15.000000) + 5.400000$
Range is from x=15.000000 to x=20.000000

$+0.034715*(x-20.000000)^3 - 0.416577*(x-20.000000)^2 + 0.610873*(x-20.000000) + 12.000000$

Range is from $x=20.000000$ to $x=24.000000$

5 appendix

5.1 source code

nma_proj_2.m.txt