

# The Report for Programming Assignments in Chapter Two

Wenchong Huang

*Date: October 10, 2022*

## 1 How to Test

Enter the folder `Programming-Chapter2/src` with terminal, make here, you will see some executable files whose names are corresponding assignments. Run them directly and you will see the results.

## 2 Manual

This package is for Newton's polynomial interpolation.

### 2.1 Newton's Interpolation

#### 2.1.1 Initializing

You should include the header `interpolation.h`. Then you should give your interpolating points and interpolating values with two `std::vector<double>`. Here is an example where  $x_k = -5 + 10\frac{k}{n}$  ( $k = 0, 1, \dots, n$ ) and  $f(x) = \frac{1}{1+x^2}$ .

```
class F{
public:
    double operator () (const double &x) const{
        return 1.0 / (1.0 + x * x);
    }
} func;
std::vector<double> x, f;
for(int i = 0; i <= n; i++){
    x.push_back(-5.0 + 10.0 * i / n);
    f.push_back(func(x[i]));
}
```

After that, you can get the interpolation polynomial by:

```
NewtonInterpolation poly(x, f);
```

You can use the following code to get the value of  $p_n(f; x)$  at point  $x$ .

```
double v = poly(x);
```

### 2.1.2 Add an Interpolating Point

One of the advantage of Newton's interpolation is that it can add interpolating points conveniently. In this package, you can add one with the following code.

```
poly.addPoint(x_new);
```

It will increase the order of the polynomial `poly` by one to make it coincides with  $f$  at  $x_{\text{new}}$  in an  $O(n)$  time.

### 2.1.3 Output

This package supports three output modes: Normal, Latex and Tikz.

The Normal mode is for human to read directly, as following

```
2+0.1*(x+5)-0.03*(x+5)*x
```

The Latex mode is for showing the result as a formula in a  $\text{\LaTeX}$  document, as following

```
2+0.1\pi_{0}(x)-0.03\pi_{1}(x)
```

The Tikz mode is for drawing the image of the result in the **Tikz** package, as following

```
2+0.1*(\x+5)-0.03*(\x+5)*\x
```

The default output mode is Normal. To change it, using one of the following codes.

```
NewtonInterpolation::setOutput(NewtonInterpolation::OUTPUT_NORMAL);  
NewtonInterpolation::setOutput(NewtonInterpolation::OUTPUT_LATEX);  
NewtonInterpolation::setOutput(NewtonInterpolation::OUTPUT_TIKZ);
```

## 2.2 Hermite's Interpolation

Hermite's Interpolation depends not only on function values, but also derivate values, even high order derivate values.

### 2.2.1 Initializing

You should include the header `interpolation.h`. Then define an Hermite interpolation polynomial.

```
HermiteInterpolation poly;
```

You should give your interpolating conditions like  $f^{(k)}(x) = y$  with the following code.

```
poly.addCondition(x, k, y);
```

You can get the value of the interpolation polynomial at point  $x$  with the following code.

```
double v = poly(x);
```

You can get the derivate function of  $k$ th order with the following code.

```
HermiteInterpolation dpoly = poly.diff(k);
```

## 2.2.2 Output

This package supports two output modes: Latex and Tikz.

The Latex mode is for showing the result as a formula in a  $\text{\LaTeX}$  document, as following

```
3+75x^{\scriptstyle 1}+7.16191x^{\scriptstyle 2}
```

The Tikz mode is for drawing the image of the result in the **Tikz** package, as following

```
3+75*\x+7.16191*\x*\x
```

The default output mode is Latex. To change it, using one of the following codes.

```
HermiteInterpolation::setOutput(HermiteInterpolation::OUTPUT_LATEX);
HermiteInterpolation::setOutput(HermiteInterpolation::OUTPUT_TIKZ);
```

## 3 Results

### 3.1 Assignment B

Here are the Newton's interpolation results.

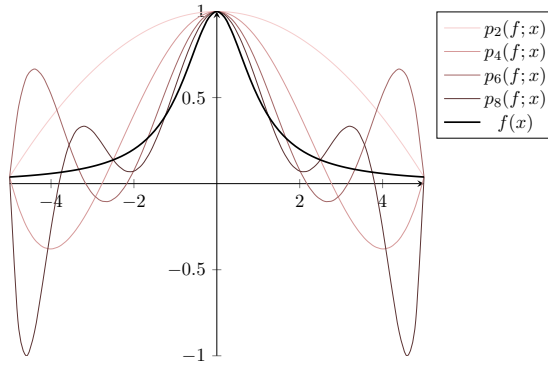
$$p_2(f; x) = 0.0384615 + 0.192308\pi_0(x) - 0.0384615\pi_1(x)$$

$$p_4(f; x) = 0.0384615 + 0.0397878\pi_0(x) + 0.061008\pi_1(x) - 0.0265252\pi_2(x) + 0.00530504\pi_3(x)$$

$$p_6(f; x) = 0.0384615 + 0.0264644\pi_0(x) + 0.0248454\pi_1(x) + 0.0149446\pi_2(x) - 0.0131699\pi_3(x) \\ + 0.00420316\pi_4(x) - 0.000840633\pi_5(x)$$

$$p_8(f; x) = 0.0384615 + 0.0223428\pi_0(x) + 0.013956\pi_1(x) + 0.0117043\pi_2(x) + 0.000674338\pi_3(x) \\ - 0.00489646\pi_4(x) + 0.00243964\pi_5(x) - 0.000687223\pi_6(x) + 0.000137445\pi_7(x)$$

The following figure shows the images of the interpolating polynomials and the original function.



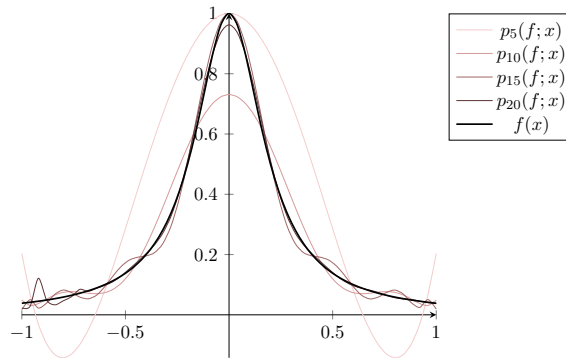
This figure illustrates the Runge phenomenon significantly.

### 3.2 Assignment C

In this assignment, we choose the interpolating points to be the zeros of Chebyshev polynomials  $T_n$ , which are

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k = 1, 2, \dots, n$$

The interpolation results are too long to show. Please run the corresponding program directly to see the results. The following figure shows the images of the interpolating polynomials and the original function.



This figure illustrates that the Chebyshev interpolation is free of the wide oscillations in assignment B.

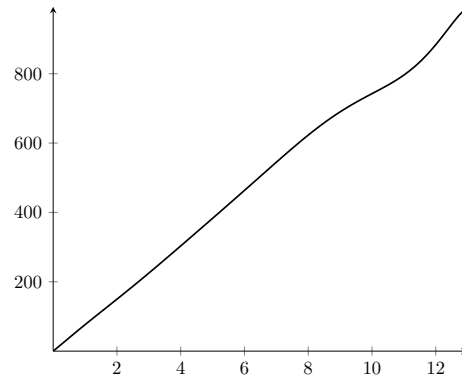
## 4 Assignment D

Let  $f(x)$  be the position at time  $x$ , then the velocity at time  $x$  is  $f'(x)$ .

Interpolating with the function values and derivate values at 0, 3, 5, 8, 13, we can get an polynomial of order 9, as following.

$$\begin{aligned} f(x) = & 75x^1 + 7.16191x^2 - 10.0953x^3 + 5.50812x^4 - 1.5383x^5 + 0.243041x^6 \\ & - 0.0218757x^7 + 0.00104059x^8 - 2.02236 \times 10^{-5}x^9 \end{aligned}$$

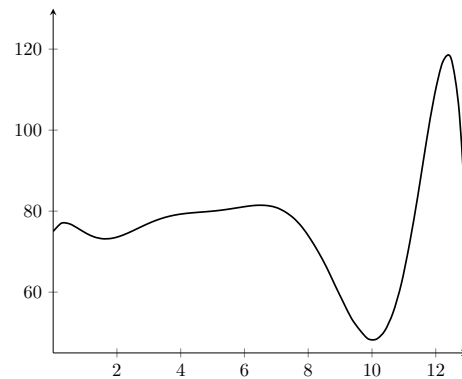
The image of the function sees the following figure.



To get the vilocity image, we can calculate the derivate of  $f$ , as following.

$$f'(x) = 75 + 14.3238x^1 - 30.2859x^2 + 22.0325x^3 - 7.69148x^4 + 1.45825x^5 \\ - 0.15313x^6 + 0.00832472x^7 - 0.000182013x^8$$

The image of vilocity function sees the following figure.



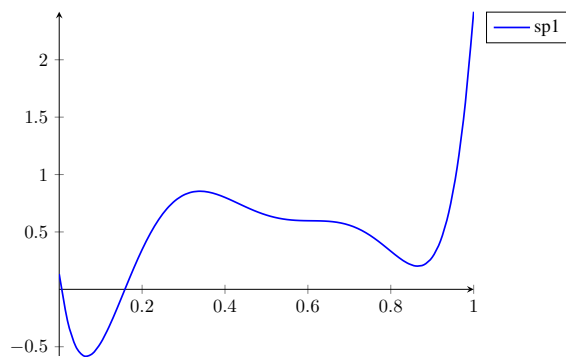
Now we can answer the problems:

- (a) The position of time  $t = 10s$  is  $f(10) = 742.503$  (feet).
- (b) As the vilocity image shows, it could be easily observed that the vilocity around time  $t = 12s$  reaches about 120 feet per second. That's an excessive speeding.

## 4.1 Assignment E

Here are the interpolating results.

The images see the following figure.



## 5 Summary

The figures in this report are drawn with **Tikz**.

This template is designed by Elegant $\text{\LaTeX}$  Program.

Many appreciations for your carefully reading. If you found any mistakes, please contact me directly.  
Have fun with your loving one (boyfriend, girlfriend or coding)!