SI 211: Numerical Analysis **Prof. Boris Houska**

Solution of Homework 2

scribe: Provided by *Yiming Mao* *Due Date: Oct 15, 2018*

Question-1

Assume that a function $f : \mathbb{R} \to \mathbb{R}$ satisfies $f(0) = 1, f(1) = 3$, and $f(2) = 19$. Construct a polynomial of the form $p(x) = a_0 + a_1 x + a_2 x^2$ such that p interpolates f at $x \in \{0, 1, 2\}$. What are a_0, a_1, a_2?

Solution: By using Lagrange interpolation, the corresponding Lagrange polynomials are

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{1}{2}x^2 - \frac{3}{2} + 1$$
$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = -x^2 + 2x$$
$$L_2(x) = \frac{(x - x_1)(x - x_0)}{(x_1 - x_1)(x_1 - x_0)} = \frac{1}{2}x^2 - \frac{1}{2}x$$

Thus, the affine given function passing through the points is

$$p(x) = \sum_{i \in \{0,1,2\}} f(x_i) L_i(x) = 7x^2 - 5x + 1$$

and $a_0 = 1, a_1 = -5, a_2 = 7$

be cautions! The substitution method by substituting the given points into p(x) and then solving the equations is not recommended .

Question-2

Assume that a function $f : \mathbb{R}^2 \to \mathbb{R}$ satisfies

$$f(0,0) = 1, f(0,1) = 3, f(0,2) = 19, f(1,0) = 3, f(2,0) = 19, f(1,1) = 0$$

Construct a polynomial $p : \mathbb{R}^2 \to \mathbb{R}$ of the form

$$p(\boldsymbol{x}) = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2 + a_4x_2^2 + a_5x_1x_2$$

such that p interpolates f at all 6 points. What are $a_0, a_1, a_2, a_3, a_4, a_5$?
Solution: By observation, the $x_1 = 0$ show out 3 times, so we can let $x_1 = 0$, and we got:

$$p(0, x_2) = a_0 + a_3x_2 + a_4x_2^2$$

and refer from Question-1 the coefficients are $a_0 = 1, a_3 = 5, a_4 = 7$ And similarly, $a_1 = -5, a_2 = 7$ Then we got:

$$p(x_1, x_2) = 1 - 5x_1 - 5x_2 + 7x_1^2 + 7x_2^2 + a_5x_1x_2$$

Applying the last equation

$$f(1,1) = 0$$

to this: Solve the equation: $a_5 = -5$.
After all, $a_0 = 1, a_1 = -5, a_2 = 7, a_3 = -5, a_4 = 7, a_5 = -5$.

be cautions! The substitution method by substituting the given points into p(x) and then solving the equations is not recommended .

Question-3

Implement a computer program that interpolates a function $f(x)$ at the points

$$x_1 = 5, x_2 = 4, x_3 = 3, ..., x_{10} = 4, x_{11} = 5$$

with a polynomial p of order 10. Test your program for

(a) the function $f(x) = \sin(x)$ and

(b) the function $f(x) = \frac{1}{1+x^2}$.

Plot the functions as well as their interpolating polynomials. How big are the approximation errors?
Solution: (a)

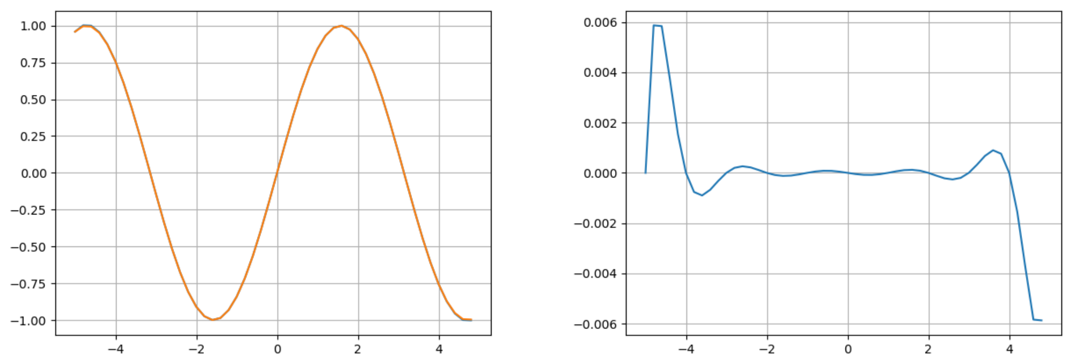

Figure 1: *left*: Sin(x) and p(x) by lagrange interpolation. *right*: error values

refer from ppt. the error is about:

$$|f(x) - p(x)| = \frac{1}{(n+1)!} \prod_{j=1}^{n} (x - x_j) \leq \frac{1}{(n+1)!} * 10^n \leq 250.52108385441718$$

for each x. (b)

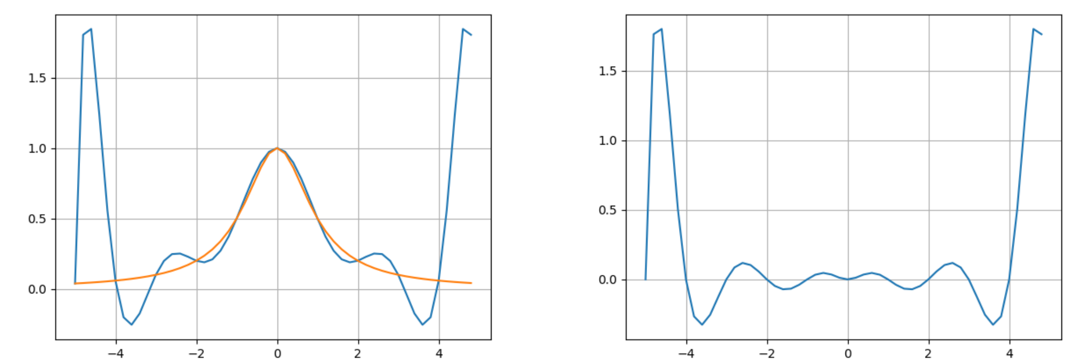

Figure 2: *left*: f(x) and p(x) by lagrange interpolation. *right*: error values

refer from ppt. the error is about:

$$|f(x) - p(x)| = \frac{1}{(n+1)!}\prod_{j=1}^{n}(x - x_j) \leq \frac{1}{(n+1)!} * 10^n \leq 250.52108385441718$$

for each x.

```
import numpy as np
import math
import matplotlib.pyplot as plt
x = list(range(-5,6,1))
y = [math.sin(x_i) for x_i in x]
def lagrange(x, x_values, y_values):
    # eval f(x) based on xs ys
    k = len(x_values)
    y = 0
    for j in range(k):
        p = 1.0
        for i in range(k):
            if i == j:
                continue
            p *= (x - x_values[i]) / (x_values[j] - x_values[i])
        y += y_values[j]*p
    return y
def f(x):
    return 1 / (1 + x*x)
def diff(f, x, given_x):
    y = [f(x_i) for x_i in x]
    return (lagrange(given_x, x, y) - f(given_x))
def cal_error(f, x, y):
    x_ = [i/5.0 for i in range(-int(5*5), int(5*5))]
    err =[lagrange(i / 5.0, x, y) for i in range(-int(5*5), int(5*5))]
    y_ = [diff(f, x, i/5.0) for i in range(-int(5*5), int(5*5))]
    print(np.mean(err))
    return x_, err, y_
x_, result1, y_ = cal_error(math.sin, x, y)
plt.plot(x_, result1)
plt.grid()
plt.show()
print("====================")
y = [f(x_i) for x_i in x]
x_, result2, y_ = cal_error(f, x, y)
plt.plot(x_, result2)
plt.grid()
plt.show()
```

References

[1] Boris Houska, *ShanghaiTech SIST Numerical Errors Lecture 2 Slides*, available at http://sist.shanghaitech.edu.cn/faculty/boris/downloads/TF502/lecture2.pdf.