SI 211：Numerical Analysis
Homework 2

付敏 72677006

October 16, 2018

1 problem1

Assume that a function $f : R \rightarrow R$ 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.

$$
\begin{aligned}
N_0 &= 1 \\
N_1 &= (x - x_0) \\
N_2 &= (x - x_0)(x - x_1) \\
p(x) &= \sum_{i=0}^{2} b_i N_i(x)
\end{aligned}
$$

Because

$$
\begin{aligned}
f(0) &= p(x_0) \\
f(1) &= P(x_1) \\
f(3) &= P(x_2)
\end{aligned}
$$

we can get that

$$
\begin{aligned}
b_0 &= f(0) = 1 \\
b_1 &= \frac{f(1) - f(0)}{x_1 - x_0} = 2 \\
b_{21} &= \frac{f(1) - b_0}{x_2 - x_1} - b_1 \frac{x_2 - x_0}{x_2 - x_1} = 7
\end{aligned}
$$

So

$$
p(x) = 1 + 2(x - 0) + 7(x - 0)(x - 1)
$$

$\Rightarrow a_0 = 1, a_1 = -5, a_2 = 7$

2 problem2

Assume that a function $f : R^2 \to R$ 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 $p : R^2 R$ of the form

$$p(x) = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_2 + a_4 x_2^2 + a_5 x_1 x_2$$

such that p interpolates f at all 6 points. What are $a_0, a_1, a_2, a_3, a_4, a_5$?

Solution.

When $x_1 = 0$, $p(x) = a_0 + a_3 x_2 + a_4 x_2^2$ This problem is the same as problem 1. So we can get that $a_0 = 1, a_3 = -5, a_4 = 7$.

When $x_2 = 0$
$p(x) = a_0 + a_1 x_1 + a_1 x_1^2$

$$\begin{array}{l|l|l|l} x_1 = 0, x_2 = 0 & f(0,0) = 1 & d_{01} = 2 & d_{02} = 7 \\ x_1 = 1, x_2 = 0 & f(1,0) = 3 & d_{12} = 16 & \\ x_1 = 2, x_2 = 0 & f(2,0) = 19 & & \end{array}$$

From the above table, we can get that $p(x) = f(0,0) + d_{01}(x_1 - 0) + d_{02}(x_1 - 0)(x_1 - 1)$.

$\Rightarrow a_1 = -5, a_2 = 7$. And we have $f(1,1) = 0$. So we can get that $a_5 = -5$.
In conclusion, $a_0 = 1, a_1 = -5, a_2 = 7, a_3 = -5, a_4 = 7, a_5 = -5$.

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

$$x_1 = -5, x_2 = -4, x_3 = -3, ..., x_1 0 = 4, x_1 1 = 5$$

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. I use python to complete this program. And I use Lagrange basis to interpolate polynomials.

```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
```

```
In [3]: def lagrange_basis(x, data):
            basis =np.ones((data.shape[0], x.shape[0]))
            for i in range(data.shape[0]):
                for j in range(data.shape[0]):
                    if i != j:
                        basis[i, :] *= (x - data[j, 0]) / (data[i, 0] - data[j, 0])
            return basis
```

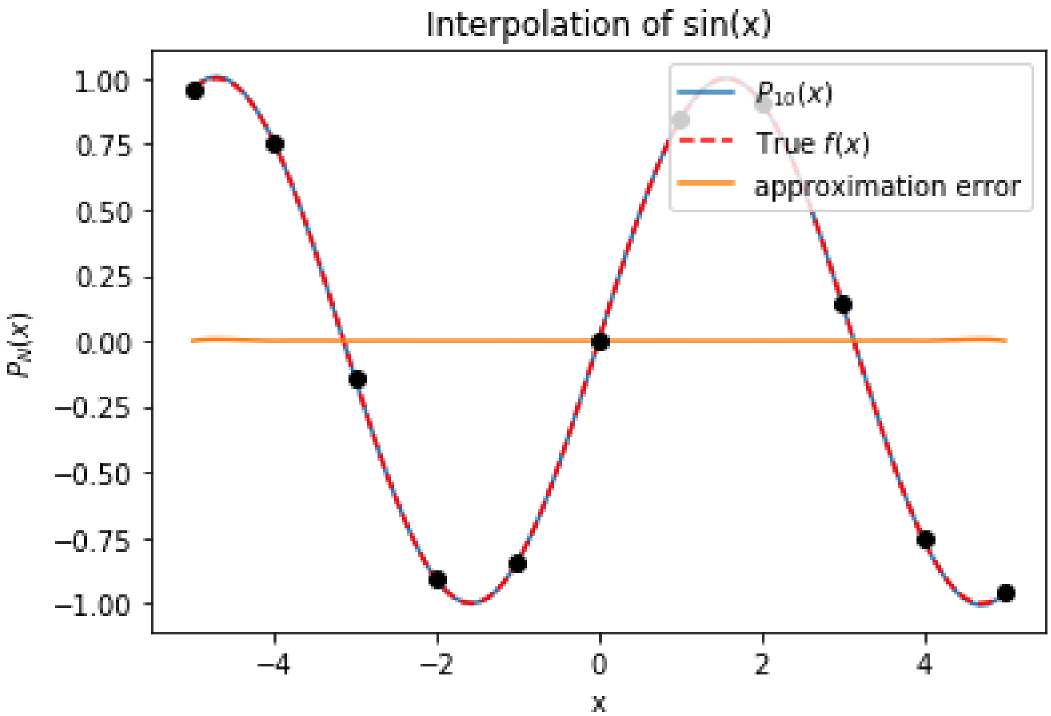
```
def poly_interpolant(x, data):
    P = np.zeros(x.shape[0])
    basis = lagrange_basis(x, data)
    for n in range(data.shape[0]):
        P += basis[n, :] * data[n, 1]
    return P
def f(x):
    return 1.0 / (1.0 + x**2)
```

```
In [7]: num_points = 11
        data_a = np.empty((num_points, 2))
        data_a[:, 0] = np.array([i for i in range(-5, 6)])
        data_a[:, 1] = np.sin(data_a[:, 0])
        data_b = np.empty((num_points, 2))
        data_b[:, 0] = np.array([i for i in range(-5, 6)])
        data_b[:, 1] = f(data_b[:, 0])

        x_a = np.linspace(-5, 5, 100)
        x_b = np.linspace(-5, 5, 100)
```

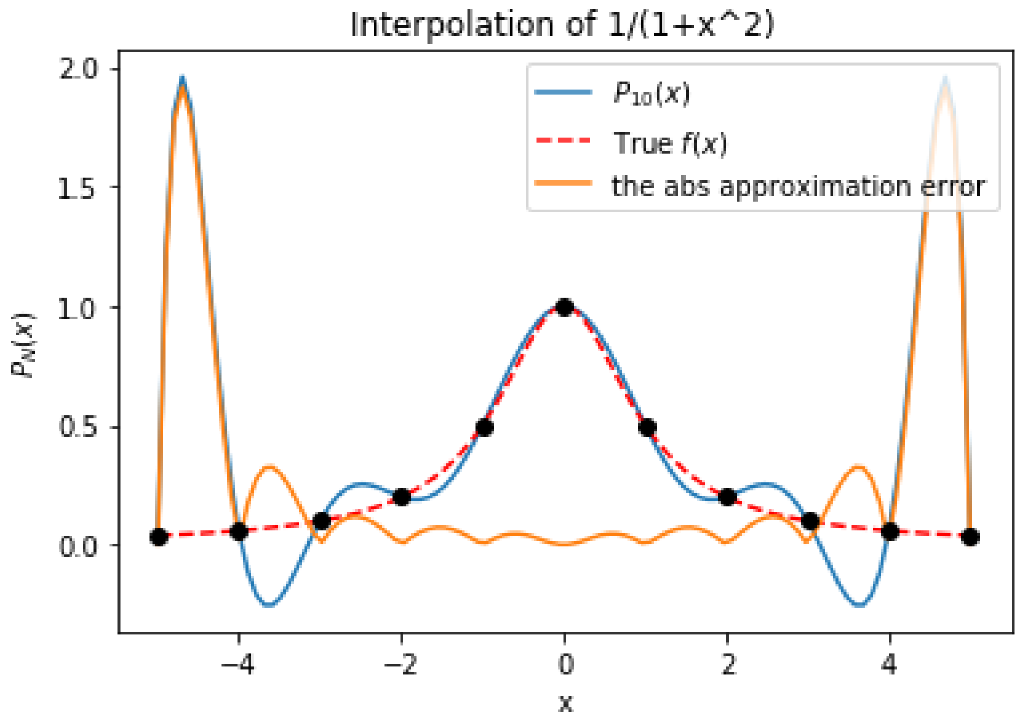
(a)

```
In [9]: fig = plt.figure()
        axes = fig.add_subplot(1, 1, 1)
        axes.plot(x_a, poly_interpolant(x_a, data_a), label="$P_{%s}(x)$" % 10)
        axes.plot(x_a, np.sin(x_a), 'r--', label="True $f(x)$")
        axes.plot(x_a,abs(np.sin(x_a)- poly_interpolant(x_a, data_a)),  label="approximation e
        for point in data_a:
            axes.plot(point[0], point[1], 'ko')
        axes.set_title("Interpolation of sin(x)")
        axes.set_xlabel("x")
        axes.set_ylabel("$P_N(x)$")
        axes.legend(loc=1)
        plt.show()
```



(b)

```
In [10]: fig = plt.figure()
         axes = fig.add_subplot(1, 1, 1)
         axes.plot(x_b, poly_interpolant(x_b, data_b), label="$P_{%s}(x)$" % 10)
         axes.plot(x_b, f(x_b), 'r--', label="True $f(x)$")
         axes.plot(x_b,abs(f(x_b)- poly_interpolant(x_b, data_b)),label="the abs approximation
         for point in data_b:
             axes.plot(point[0], point[1], 'ko')
         axes.set_title("Interpolation of 1/(1+x^2)")
         axes.set_xlabel("x")
         axes.set_ylabel("$P_N(x)$")
         axes.legend(loc=1)
         plt.show()
```



Interpolation of 1/(1+x^2)
$P_{10}(x)$
True $f(x)$
the abs approximation error
2.0
1.5
1.0
0.5
0.0
$P_N(x)$
−4
−2
0
2
4
x