Лабораторная работа №3

Ларин Егор. 4 группа 2 курс

4 апреля 2022 г.

Теория

$$N = 10, a = -2, b = 2, h = \frac{b-a}{N}, f(x) = e^{\cos x}, f'(x) = -\sin(x)e^{\cos x}$$

$$\begin{cases} \frac{h}{6}M_{i-1} + \frac{2h}{3}M_i + \frac{h+1}{6}Mi+1 = \frac{f_{i+1}-f_i}{h} - \frac{f_i - fi-1}{h}, i = \overline{2, N-2} \\ \frac{h}{3}M_0 + \frac{h}{6}M_1 = \frac{f_1 - f_0}{h} - f'(a) \\ \frac{h}{3}M_N + \frac{h}{6}M_{N-1} = \frac{f_{N-1}-f_N}{h} + f'(b) \end{cases}$$

$$S_3(x) = \left\{Pi, 3(x) = M_{i-1}\frac{(x_i - x)^3}{6h} + M_i\frac{(x - x_{i-1})^3}{6h} + (f_{i-1} - \frac{h_i^2}{6}M_{i-1})\frac{x_i - x}{h} + (f_i - \frac{h^2}{6}M_i)\frac{x - x_{i-1}}{h} | x \in [x_{i-1}, x_i], i = \overline{1, N}\right\}$$

Листинг кода

```
from math import exp, cos, sin
import numpy as np
from matplotlib import pyplot as plt

a = -2
b = 2
N = 10
h = (b-a) / N
f = lambda x: exp(cos(x))
dfdx = lambda x: -sin(x) * exp(cos(x))
# f = lambda x: abs(x)
# dfdx = lambda x: -1 if x== -2 else 1
xss = [a + i*h for i in range(0, N+1)]
yss = [f(x) for x in xss]

def TDMA(a, b, c, d):
    n = len(d)
    w=  [.0] * (n-1)
    g= [.0] * n
    p= [.0] * n

    w[0] = c[0]/b[0]
    g[0] = d[0]/b[0]

    for i in range(1,n-1):
        w[i] = c[i]/(b[i] - a[i-1]*w[i-1])
    for i in range(1,n):
        g[i] = (d[i] - a[i-1]*g[i-1])/(b[i] - a[i-1]*w[i-1])
    p[n-1] = g[n-1]
    for i in range(n-1,0,-1):
        p[i-1] = g[i-1] - w[i-1]*p[i]
    return p

css = []
```

```
ass = []
bss = []
dss = []
for i in range(0, N+1):
    if i== 0:
        ass.append(h/3)
        bss.append(h/6)
        dss.append((yss[1] - yss[0])/h - dfdx(a))
    elif i== N:
        css.append(h/6)
        ass.append(h/3)
        dss.append(dfdx(b) - (yss[N] - yss[N-1])/h)
    else:
        css.append(h/6)
        ass.append(2*h/3)
        bss.append(h/6)
        dss.append((yss[i+1] - yss[i])/h - (yss[i] - yss[i-1])/h)

mss = TDMA(css,ass,bss,dss)
sss = []

def wrap(i):
    def s(x):
        return (xss[i] - x)**3 * mss[i-1] / (6*h)  + (x-xss[i-1])**3 * mss[i] / (6
    return s

for i in range(1, N+1):
    sss.append(wrap(i))

def S(x):
    i  = int((x - a)/h)
    if i == 10:
        i = 9
    return sss[i](x)

def compare(n):
    d = (b-a)/n
    xs = [a + i * d for i in range(n+1)]
    plt.plot(
            xs, [S(x) for x in xs], "b",
            xs, [f(x) for x in xs],  "r",
            xss, [f(x) for x in xss], "ro",
    )
    plt.legend(["S(x)", "f(x)"])
    plt.show()

def error():
    d = (b-a)/100
    xs = [a + i * d for i in range(101)]
    return max([abs(f(x) - S(x)) for x in xs])

print(error())
compare(100)
```

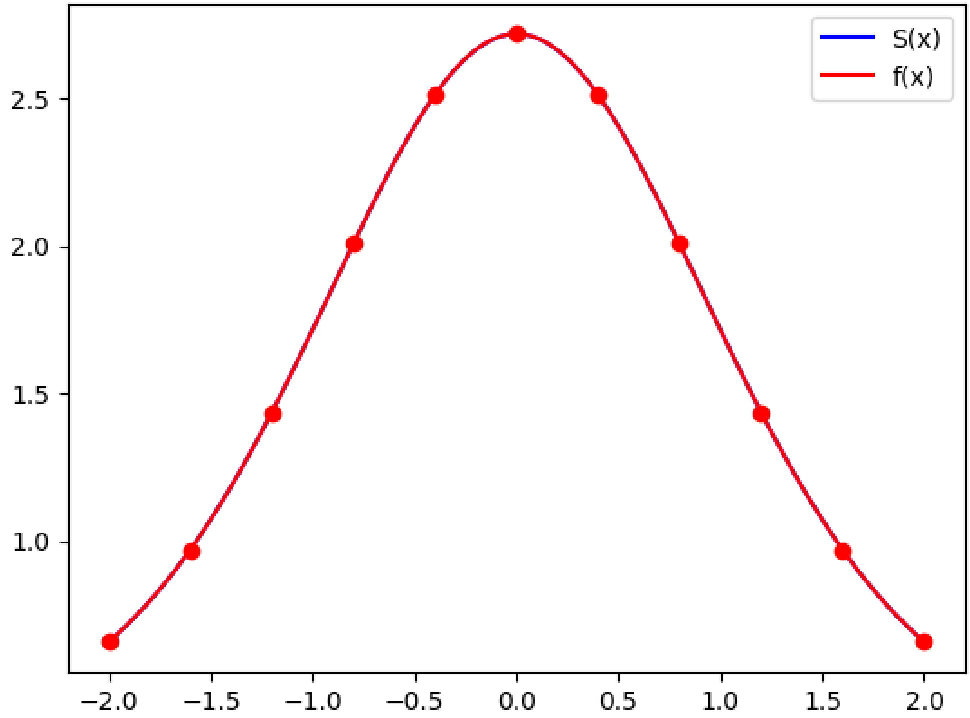
Результаты численного эксперимента

Оценка погрешности

$$max_{i=0,100} |S_3(\overline{x_i} - f(\overline{x_i})| = 0.0008154280452394858$$

График

Рис. 1:



Выводы

Кубический сплайн хорошо приближает функции с непрерывной производной.