МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

МЕТОДЫ ЧИСЛЕННОГО АНАЛИЗА

Лабараторная работа №2

Приближение функций с помощью кубического сплайна

Студент
2 курса 2 группы
Царик Виталий
Александрович

Преподаватель
Никифоров Иван
Васильевич

Минск 2019

1 Условие

Отрезок $[a, b]$ разбить на 10, 20, 40 отрезков. На каждом отрезке приблизить функцию кубическим сплайном по узлам Чебышева. Для каждого из 3 случаев (n=10,20,40) построить график.

2 Вариант

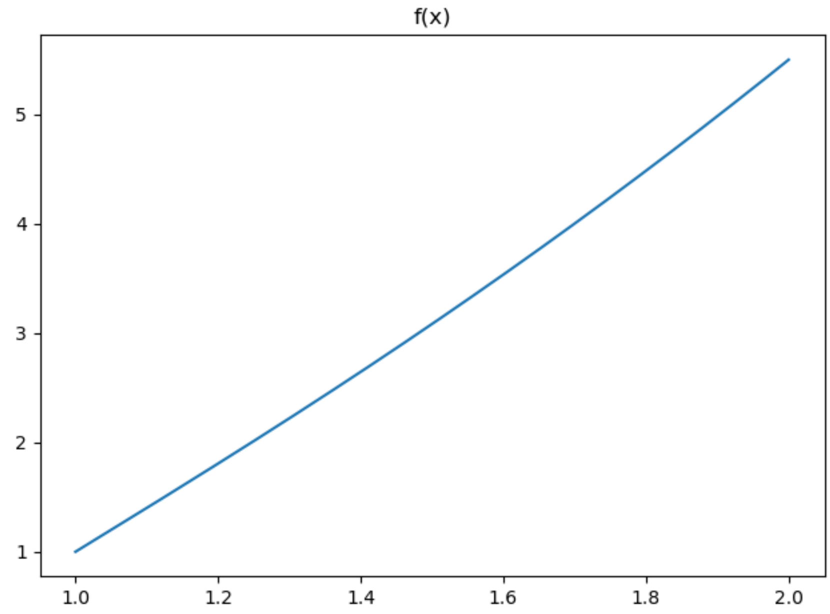
$$f(x) = -\frac{1}{x} + x + x^2, x \in [1, 2] \tag{1}$$

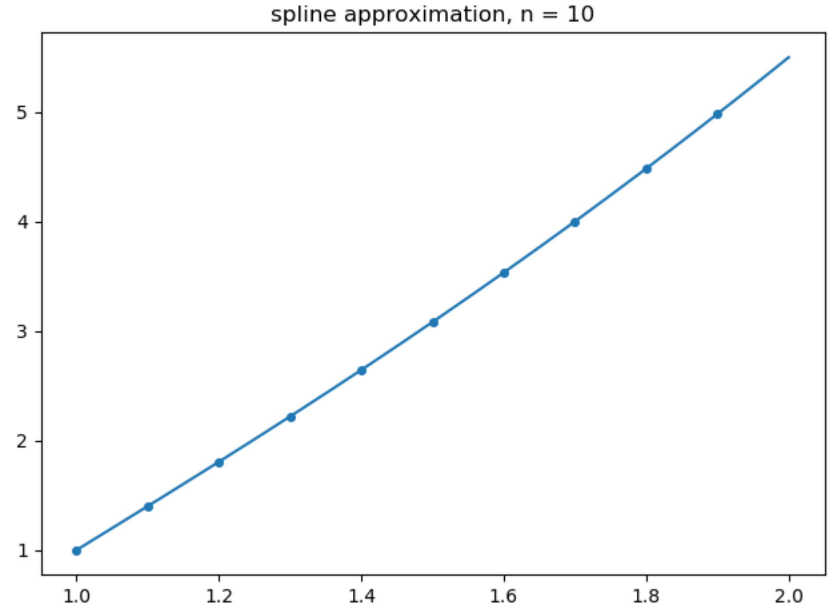
3 Теория

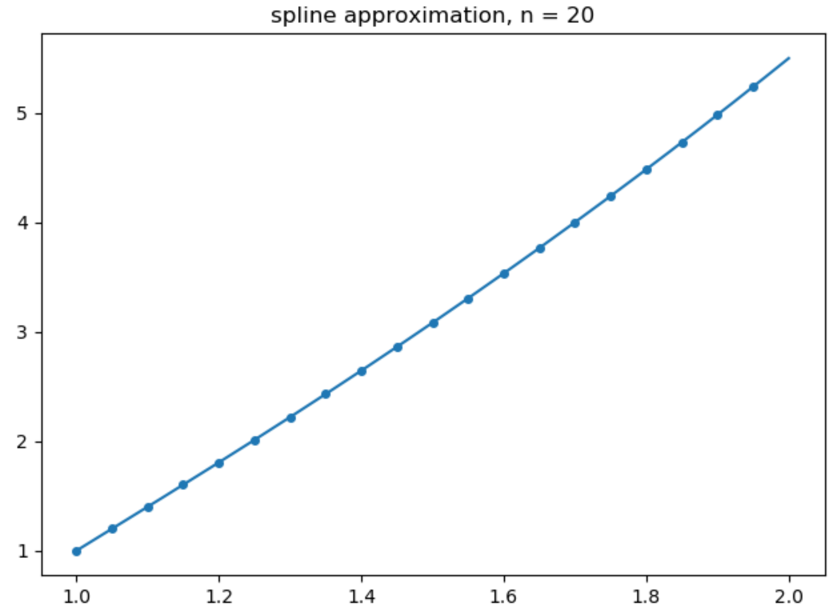
Узлы Чебышева вычисляются по формуле:

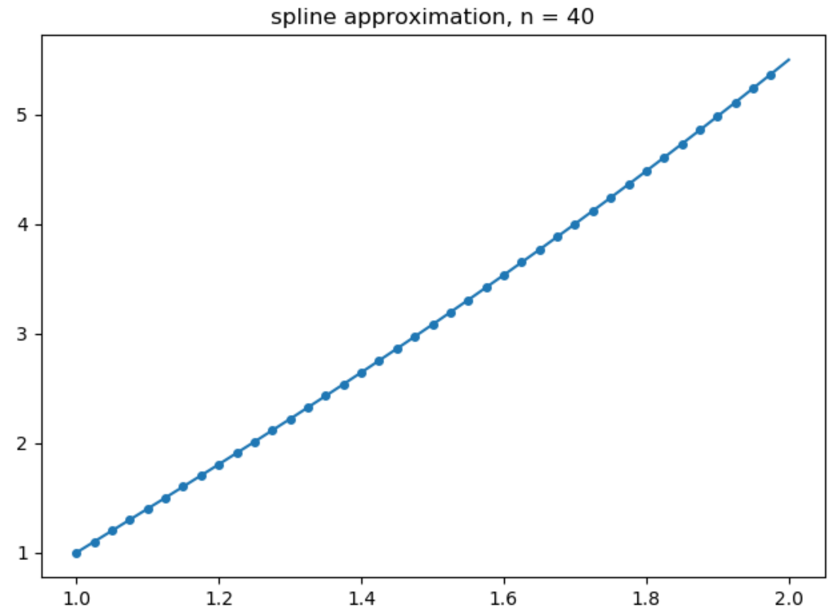
$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2k+1}{2(n+1)}\pi\right), k = \overline{1, n} \tag{2}$$

4 Графики




spline approximation, n = 10
1
2
3
4
5
1.0
1.2
1.4
1.6
1.8
2.0


spline approximation, n = 20
1
2
3
4
5
1.0
1.2
1.4
1.6
1.8
2.0


spline approximation, n = 40
1
2
3
4
5
1.0
1.2
1.4
1.6
1.8
2.0

5 Исходный код

```
import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline

n = (10, 20, 40)
A = 1
B = 2
RANK = 3
NUMBER_OF_NODES = RANK + 1


def f(x):
  return -1/x + x + x*x


def chebyshev_nodes(a, b, n):
  return np.array(
    ([(a + b) / 2.0 + (b - a) / 2.0 * math.cos((2 * i + 1) / (2.0 * n + 2.0) * math.pi)
    for i in range(n)]), dtype='float')


if __name__ == '__main__':
  x = np.linspace(A, B, 10000)
  plt.plot(x, f(x))
  plt.title('f(x)')
  plt.show()

  for n_i in n:
    beg = A
    step = (B - A) / n_i
    end = beg

    x = []
    y = []
    for i in range(n_i):
    beg = end
    end += step

    nodes = chebyshev_nodes(beg, end, NUMBER_OF_NODES)
    nodes.sort()
    polynom = CubicSpline(nodes, f(nodes))

    x_i = np.linspace(beg, end, 100)
    x.extend(x_i)
    y.extend(polynom(x_i))

  plt.title('spline approximation, n = {}'.format(n_i))
  plt.plot(x, y, 'o', ls='-', ms=4, markevery=100, label='polynom')
  plt.show()
```

6 Выводы

Приближение с помощью кубического сплайна являются довольно точным для данной функции даже при относительно небольших значениях n