

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
Московский государственный технический университет имени Н.Э. Баумана

Лабораторная работа №3
по курсу «Численные методы»
«Интерполяция В-сплайнами»

Выполнил:
студент группы ИУ9-62Б
Головкин Дмитрий
Проверила:
Домрачева А.Б.

Москва, 2023

Цель:

Анализ метода интерполяции функции, основанный на построении кубического в контрольных точках.

Постановка задачи:

Дано: Функция $y_i = \phi(x_i), i = \overline{1, n}$ задана таблично, исходные данные включают ошибки измерения.

x_1	x_2	\dots	x_{n-1}	x_n
y_1	y_2	\dots	y_{n-1}	y_n

Найти: Функцию (интерполанту) $f(x)$, совпадающую с значениями $y_i, i = \overline{1, n}$ в контрольных точках $x_i, i = \overline{1, n}$:

$$f(x_i) = y_i$$

Тестовый пример:

Зададим некоторую функцию $\phi(x) = \exp(x)$. Представим значения в виде таблицы:

0.0	0.055	0.111	0.166	0.222	0.277	0.333	0.388	0.444	0.5
1.0	1.057	1.117	1.181	1.248	1.320	1.395	1.475	1.559	1.648

0.555	0.611	0.666	0.722	0.777	0.833	0.888	0.944	1.0
1.742	1.842	1.947	2.059	2.176	2.300	2.432	2.571	2.718

Теоретические сведения:

Интерполяция, интерполирование — в вычислительной математике способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.

Основная цель интерполяции — получить быстрый (экономичный) алгоритм вычисления значений $y(x)$ для значений x , не содержащихся в таблице данных. Интерполирующие функции $f(x)$, как правило строятся в виде линейных комбинаций некоторых элементарных функций:

$$f(x) = \sum_{k=0}^N c_k \Phi_k(x)$$

где $\{\Phi_k(x)\}$ — фиксированный линейно независимые функции, c_0, c_1, \dots, c_n — не определенные пока коэффициенты.

Один из способов интерполирования на всем отрезке $[a, b]$ является интерполирование сплайнами. Сплайном называется кусочно-полиномиальная функция, определенная на отрезке $[a, b]$ и имеющая на этом отрезке некоторое количество непрерывных производных. Преимущества интерполяции сплайнами по сравнению с обычными методами интерполяции — в сходимости и устойчивости вычислительного процесса.

Рассмотрим один из наиболее распространенных в практике случаев – интерполирование функции кубическим сплайном.

Пусть на отрезке $[a, b]$ задана непрерывная функция $y(x)$. Введем разбиение отрезка:

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$$

и обозначим $y_i = y(x_i), i = \overline{1, n}$

Сплайном, соответствующим данной функции узлам интерполяции называется функция $s(x)$, удовлетворяющая следующим условиям:

1. На каждом отрезке $[x_{i-1}; x_i], i = \overline{2, n}$ функция $s(x)$ является кубическим многочленом;
2. Функция $s(x)$, а также ее первая и вторая производные непрерывны на отрезке $[a, b]$;
3. $s(x_i) = y_i, i = \overline{2, n}$ - условие интерполирования.

Сплайн, определяемый условиями 1 – 3, называется интерполяционным кубическим сплайном и имеет вид:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Заметим несколько важных условий: $S'_i(x_i) = S'_{i+1}(x_i)$, $S'_i(x_i) = S'_{i+1}(x_i)$, $S''_1(a) = 0$, $S''_n(b) = 0$. Последние два условия называют условиями гладкости на краях. Пусть n - число разбиений на отрезке $[a; b]$. Тогда зададим параметр h следующим образом : $h = \frac{b-a}{n}$

Теперь, используя данные условия, можем получить следующие соотношения:

1. $d_i = \frac{c_{i+1} - c_i}{3h}, i = \overline{1, n};$
2. Используя условия гладкости на краях получаем: $c_1 = 0$, $c_{n+1} = 0$;
3. $a_i = y_{i-1}, i = \overline{1, n};$
4. $b_i = \frac{y_i - y_{i-1}}{h} - \frac{h}{3}(c_{i+1} + 2c_i);$

Следующим шагом необходимо вычислить коэффициенты c_i . Преобразуя далее функцию сплайна, получаем следующую систему:

$$\begin{cases} 4c_2 + c_3 = \frac{3}{h^2}(y_2 - 2y_1 - y_0) \\ \dots \\ c_i + 4c_{i+1} + c_{i+2} = \frac{3}{h^2}(y_{i+1} - 2y_i - y_{i-1}) \\ \dots \\ c_{n-1} + 4c_n = \frac{3}{h^2}(y_n - 2y_{n-1} - y_{n-2}) \end{cases}$$

Данные коэффициенты будем искать с помощью метода прогонки.

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ: Листинг 1. Интерполяция кубическими сплайнами

```

#!/python
# -*- coding: utf-8 -*-
import numpy as np

def calculate_x_n(a, b, c, d, n):
    #alpha = [0.0] * n
    #beta = [0.0] * n
    alpha = np.zeros(n)
    beta = np.zeros(n)
    alpha[0] = -c[0] / b[0]
    beta[0] = d[0] / b[0]
    for i in range(1,n-1):
        #print(i)
        alpha[i] = -c[i] / (a[i-1]*alpha[i-1] + b[i])
        beta[i] = (d[i] - a[i-1]*beta[i-1]) / (a[i-1]*alpha[i-1] + b[i])
    x_n = (d[n-1] - a[n-2]*beta[n-2]) / (a[n-2]*alpha[n-2] + b[n-1])
    #print(alpha, beta)
    return x_n, alpha, beta

def calculate_c_n_vector(x_n, alpha, beta, n):
    x = [0.0] * (n-1)
    x.append(x_n)
    for i in range(n-2, -1, -1):
        x[i] = alpha[i]*x[i+1] + beta[i]
    return x

def calculate_d(y_n, h, n):
    d = [y_n[0] / 3 * (h*h)]
    for i in range(2, n+1):
        d.append(y_n[i] - 2*y_n[i-1] + y_n[i-2])
    return (3/(h*h)) * np.asarray(d)

a = 0
b = 1
n = 9
a_coefs = [1.0] * (n-1)
b_coefs = [4.0] * n

```

```

c_coefs = [1.0] * (n-1)
x_n = np.arange(a, b + b/18, b/9)
y_n = np.exp(x_n)
print('X: ', x_n)
print('Y: ', y_n)

h = (b - a) / n
d_coefs = calculate_d(y_n, h, n)
#print("----", d_coefs)
spl, alpha, beta = calculate_x_n(a_coefs, b_coefs, c_coefs, d_coefs, n)
#print(spl, alpha, beta)

c_n = calculate_c_n_vector(spl, alpha, beta, n)
c_n.append(0)
c_n = np.asarray(c_n)
print('c_coefs: ', c_n)
d_n = (c_n[1:n+1] - c_n[:n]) / (3*h)
print('d_coefs: ', d_n)
a_n = y_n[:n]
print('a_coefs: ', a_n)
b_n = (y_n[1:n+1] - y_n[:n]) / h - (h / 3) * (c_n[1:n+1] + 2*c_n[:n])
print('b_coefs: ', b_n)

x_n_new = np.arange(a, b + b/9, b/18)
y_n_new = np.exp(x_n_new)
spline_extended = [a_n[(idx-1)//2] + b_n[(idx-1)//2] * (x_n_new[idx] - x_n[(idx-1)//2]) +
                   d_n[(idx-1)//2] * (x_n_new[idx] - x_n[(idx-1)//2])**3 for idx in range(1, 2*n+1)]
spline_extended = np.concatenate((a_n[:1], spline_extended))

spline = a_n + b_n*(x_n[1:n+1] - x_n[:n]) + c_n[:n]*(x_n[1:n+1] - x_n[:n])**2 +
         d_n*(x_n[1:n+1] - x_n[:n])**3
spline = np.concatenate((a_n[:1], spline))

for i in range(0, n+1):
    print("x=", x_n[i], " | f(x)=", y_n[i], " | spl(x)=", spline[i], " | delta=")

for i in range(0, 2*n+1):
    print("x=", x_n_new[i], " | f(x)=", y_n_new[i], " | spl(x)=", spline_extended[i], " | delta=")

```

Результаты:

Для тестирования полученной программы была задана функция $\phi(x) = \exp(x)$. Представим её в виде таблицы:

0.0	0.055	0.111	0.166	0.222	0.277	0.333	0.388	0.444	0.5
1.0	1.057	1.117	1.181	1.248	1.320	1.395	1.475	1.559	1.648

0.555	0.611	0.666	0.722	0.777	0.833	0.888	0.944	1.0
1.742	1.842	1.947	2.059	2.176	2.300	2.432	2.571	2.718

В результате работы программы (Листинг 1) получаем значения:

Значение x_n	Значение y_n	Значение сплайна $s^3(x_n)$	Разница значений $s^3(x_n) - y_n$
0.0	1.0	1.0	0.0
0.055	1.057	1.0575984147738515	0.0004706700136150044
0.111	1.117	1.1175190687418637	0.0
0.166	1.181	1.181233576490805	0.00012683637484078858
0.222	1.248	1.2488488690016821	0.0
0.277	1.320	1.3202266534113838	3.386497726354243e-05
0.333	1.395	1.3956124250860895	0.0
0.388	1.475	1.475328851159584	1.1764331038222053e-05
0.444	1.559	1.5596234976067807	0.0
0.5	1.648	1.6487309531466745	9.682446546310786e-06
0.555	1.742	1.7429089986334578	0.0
0.611	1.842	1.8424465712119993	3.0887835709814127e-05
0.666	1.947	1.9477340410546755	2.220446049250313e-16
0.722	2.059	2.0591131797740907	0.00010948556121936903
0.777	2.176	2.176629931716248	0.0
0.833	2.300	2.3005639380231275	0.00041195286969708533
0.888	2.432	2.4324254542872077	0.0
0.944	2.571	2.57291728658255	0.0015328517945203401
1.0	2.718	2.718281828459045	0.0

Выводы:

В ходе выполнения лабораторной работы был рассмотрен метод интерполяции функции, основанный на построении кубического сплайна в контрольных точках, так же для метода была написана реализация на языке программирования Python.

Анализируя результаты полученной программы, можно заметить то, что метод интерполяции функции, основанный на построении кубического в контрольных точках, имеет высокую

точность вследствие низкой погрешности. Повышая степень кусочно-интерполяционного многочлена, можно добиться еще лучших результатов аппроксимации функции.