A picture containing logo

Description automatically generated

Лабораторная работа №5

по Вычислительной Математики

Вариант 13

Выполнил:

Пурэвсурэн Билгуун

Группа Р3213

Преподователь:

Малышева Татьяна Алексеевна

г. Санкт-Петербург 2022г

1. **Цель работы:**

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

**Для исследования использовать:**

* многочлен Лагранжа;
* многочлен Ньютона;
* многочлен Гаусса.

1. **Вычислительная реализация задачи:**
   1. Используя первую или вторую интерполяционную формулу Ньютона, первую или вторую интерполяционную формулу Гаусса вычислить значения функции при данных значениях аргумента (для значения Х1 и Х2, см. табл. 1 - 4).
   2. Построить таблицу конечных разностей.
   3. **Подробные вычисления привести в отчете.**

Table

Description automatically generated

1. **Программная реализация задачи:**
   1. Исходные данные задаются в виде:
      1. Набора данных (таблицы x,y)
      2. На основе выбранной функции (например, sin x)
   2. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл.5)
   3. Построить графии заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами)
2. **Описание метода, расчетные формулы:**

Интерполяция — это метод нахождения неизвестных промежуточных значений некоторой функции по имеющемуся дискретному набору ее известных значений. Точки 𝑥0, 𝑥1, ... 𝑥𝑛 называются узлами интерполяции. Точка, в которой нужно найти значение функции – точкой интерполяции.

Процесс вычисления значений функции F(x) в точках отличных от узлов интерполирования называется интерполированием функции f(x). При этом различают интерполирование в узком смысле, когда x принадлежит интервалу [x0, xn], и экстраполирование, когда x находится за пределами отрезка. Алгебраический многочлен, удовлетворяющий условиям интерполяции, называется интерполяционным многочленом.

Многочлен Лагранжа:

A picture containing schematic

Description automatically generated

Интерполяционная формула Гаусса:

Text, letter

Description automatically generated

Text, letter

Description automatically generated

Интерполянионная формула Стирлинга:

Text

Description automatically generated

Формула Стирлинга представляет собой среднее арифметическое первой и второй интерполяционных формул Гаусса. Применяется для интерполирования при значениях 𝑡, близких к 0. На практике ее используютпри 𝒕 ≤𝟎,𝟐𝟓. Строится по нечетному числу узлов

Интерполяционная формула Беселя:

Text, letter

Description automatically generated

Формула Бесселя применяется для интерполирования при значениях 𝑡, близких к 0,5. На практике ее используют при 𝟎, 𝟐𝟓 ≤ 𝒕 ≤ 𝟎, 𝟕𝟓. Строится по четному числу узлов

1. **Реализация расчета:**

Вычислительная реализация:

**Используя первую или вторую интерполяционную формулу Ньютона найти приближенное значение функции для x=0,534 и х=0,384 по заданной таблице.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.50 | 0.55 |
| y | 1.2557 | 2.1764 | 3.1218 | 4.0482 | 5.9875 | 6.9195 | 7.8359 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | xi | Yi | Δyi | Δ2yi | Δ3yi | Δ4yi | Δ5yi | Δ6yi |
| 0 | 0.25 | 1.2557 | 0.9207 | 0.0247 | -0.0437 | 1.0756 | -4.1277 | 10.1917 |
| 1 | 0.30 | 2.1764 | 0.9454 | -0.019 | 1.0319 | -3.0521 | 6.064 |  |
| 2 | 0.35 | 3.1218 | 0.9264 | 1.0129 | -2.0202 | 3.0119 |  |  |
| 3 | 0.40 | 4.0482 | 1.9393 | -1.0073 | 0.9917 |  |  |  |
| 4 | 0.45 | 5.9875 | 0.9320 | -0.0156 |  |  |  |  |
| 5 | 0.50 | 6.9195 | 0.9164 |  |  |  |  |  |
| 6 | 0.55 | 7.8359 |  |  |  |  |  |  |

|  |
| --- |
| йцб |

* Использован в первой интерполяционной формулы Ньютона

|  |
| --- |
|  |

* Использован в второй интерполяционной формулы Ньютона

Используем вторую интерполяционной формулы Ньютона для x=0.534, т.к она лежит во правой половине отрезка.

xn = 0.55, h = 0.3 – 0.25 = 0.05

t =

----------------------------------------------------------------------------------------------------

Для x = 0.384 используем формулой Ньютона для интерполирования вперед, т.к она лежит в первой половине отрезка.

xn = 0.30, h = 0.3 – 0.25 = 0.05

t =

----------------------------------------------------------------------------------------------------

**Используя первую или вторую интерполяционную формулу Гаусса найти приближенное значение функции для x=0,534 и х=0,384 по заданной таблице.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.50 | 0.55 |
| y | 1.2557 | 2.1764 | 3.1218 | 4.0482 | 5.9875 | 6.9195 | 7.8359 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | xi | Yi | Δyi | Δ2yi | Δ3yi | Δ4yi | Δ5yi | Δ6yi |
| 0 | 0.25 | 1.2557 | 0.9207 | 0.0247 | -0.0437 | 1.0756 | -4.1277 | 10.1917 |
| 1 | 0.30 | 2.1764 | 0.9454 | -0.019 | 1.0319 | -3.0521 | 6.064 |  |
| 2 | 0.35 | 3.1218 | 0.9264 | 1.0129 | -2.0202 | 3.0119 |  |  |
| 3 | 0.40 | 4.0482 | 1.9393 | -1.0073 | 0.9917 |  |  |  |
| 4 | 0.45 | 5.9875 | 0.9320 | -0.0156 |  |  |  |  |
| 5 | 0.50 | 6.9195 | 0.9164 |  |  |  |  |  |
| 6 | 0.55 | 7.8359 |  |  |  |  |  |  |

|  |
| --- |
|  |

* Использован в первой интерполяционной формулы Гаусса

|  |
| --- |
| sad |

* Использован в второй интерполяционной формулы Гаусса

Для x = 0.534 будем использовать первую интерполяционную формулу Гаусса. x0 = 0.40 и x > x0

t =

Для x = 0.384 будем использовать вторую интерполяционную формулу Гаусса. x0 = 0.4 и x < x0

t =

= 3.63

Программная реализация:

Калькулятор параметра t:

def t\_calc(t, n, forward=True):

""" Вычислить параметр 't' """

result = t

j = 1

if forward:

for i in range(1,n):

if i % 2 != 0:

result \*= t - j

else:

result \*= t + j

j += 1

else:

for i in range(1, n):

if i % 2 != 0:

result \*= t + j

else:

result \*= t - j

j+=1

return result

Калькулятор параметра t для формулы Беселя:

def bessel\_t(t,n):

if (n == 0):

return 1

result = t

for i in range(1, int(n/2+1)):

result \*= (t-i)

for i in range(1, int(n/2)):

result \*= (t+i)

return result

Многочлен Лагранжа:

def lagrange\_polynomial(dots, x):

""" Многочлен Лагранжа """

result = 0

n = len(dots)

for i in range(n):

c1 = c2 = 1

for j in range(n):

if i != j:

c1 \*= x - dots[j][0]

c2 \*= dots[i][0] - dots[j][0]

result += dots[i][1] \* c1 / c2

return result

Интерполяционные формулы Гаусса:

def gauss\_polynomial(dots, x):

""" Интерполяционные формулы Гаусса"""

n = len(dots)

h = dots[1][0] - dots[0][0]

a = [[0] \* n for \_ in range(n)]

for i in range(n):

a[i][0] = dots[i][1]

for i in range(1, n):

for j in range(n - i):

a[j][i] = a[j + 1][i - 1] - a[j][i - 1]

if x <= dots[n // 2][0]:

# Первая интерполяционная формула Гаусса

x0 = n // 2

t = (x - dots[x0][0]) / h

result = a[x0][0]

temp = x0

for i in range(1, n):

if i % 2 == 0:

temp -= 1

result += (t\_calc(t, i) \* a[temp][i]) / factorial(i)

else:

# Вторая интерполяционная формула Гасса

x0 = n // 2

t = (x - dots[x0][0]) / h

temp = x0

result = a[x0][0]

for i in range(1, n):

if i % 2 != 0 :

temp -= 1

result += (t\_calc(t, i, False) \* a[temp][i]) / factorial(i)

return result

Интерполяционная формула Стирлинга:

def stirling\_polynomial(dots,x):

""" Интерполяционная формула Стирлинга"""

n = len(dots)

h = dots[1][0] - dots[0][0]

a = [[0] \* n for \_ in range(n)]

for i in range(n):

a[i][0] = dots[i][1]

for i in range(1, n):

for j in range(n - i):

a[j][i] = a[j + 1][i - 1] - a[j][i - 1]

# Первая интерполяционная формула Гаусса

x0 = n // 2

t1 = (x - dots[x0][0]) / h

result1 = a[x0][0]

temp1 = x0

for i in range(1, n):

if i % 2 == 0:

temp1 -= 1

result1 += (t\_calc(t1, i) \* a[temp1][i]) / factorial(i)

# Вторая интерполяционная формула Гасса

x0 = n // 2

t2 = (x - dots[x0][0]) / h

temp2 = x0

result2 = a[x0][0]

for i in range(1, n):

if i % 2 != 0 :

temp2 -= 1

result2 += (t\_calc(t2, i, False) \* a[temp2][i]) / factorial(i)

# И среднеарифметическое значение

result = (result1 +result2)/2

return result

Интерполяционная формула Беселя:

def bessel\_polynomial(dots,x):

""" Интерполяционная формула Беселя"""

n = len(dots)

h = dots[1][0] - dots[0][0]

a = [[0] \* n for \_ in range(n)]

for i in range(n):

a[i][0] = dots[i][1]

for i in range(1, n):

for j in range(n - i):

a[j][i] = a[j + 1][i - 1] - a[j][i - 1]

x0 = 0

if ((n%2) > 0):

x0 = int(n/2)

else:

x0 = int(n/2-1)

t = (x - dots[x0][0])/h

result = (a[x0][0] + a[x0+1][0])/2

for i in range(1,n):

if i % 2:

result += ((t-0.5) \* bessel\_t(t,i-1) \* a[x0][i])/factorial(i)

else:

result += (bessel\_t(t,i) \* (a[x0][i]+a[x0-1][i])/(factorial(i)\*2))

x0 -= 1

return result

1. **Результат программы**

Text

Description automatically generated

Chart, line chart

Description automatically generated

1. **Вывод**

В результате выполнения данной лабораторной работой я познакомился с методами интерполяции функции и реализовал их на языке программирования Python, закрепив знания.