Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

09.03.04 Программная инженерия

Системное и прикладное программное обеспечение

Лабораторная работа №5
По дисциплине «Вычислительная математика»
Вариант № 1

Выполнила студентка группы P3213:

Авшистер Ольга Аркадьевна

Преподаватель:

Машина Екатерина Алексеевна

г. Санкт-Петербург 2024 г.

Цель работы

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

Рабочие формулы

Многочлен Лагранжа:

$$L_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Многочлен Ньютона с разделенными разностями:

$$N_n(x) = f(x_0) + \sum_{k=1}^{n} f(x_0, x_1, \dots, x_k) \prod_{j=0}^{k-1} (x - x_j)$$

Первая интерполяционная формула Ньютона:

$$N_n(x) = y_i + t\Delta y_i + \frac{t(t-1)}{2!}\Delta^2 y_i + \dots + \frac{t(t-1)\dots(t-n+1)}{n!}\Delta^n y_i$$

Вторая интерполяционная формула Ньютона:

$$N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \dots + \frac{t(t+1)\dots(t+n-1)}{n!}\Delta^n y_0$$

Листинг программы

```
def lagrange_polynomial(x, y, x_cur):
    res = 0
    for i in range(len(x)):
        p = 1
        for j in range(len(y)):
            if i != j:
                p *= (x_cur - x[j]) / (x[i] - x[j])
        res += p * y[i]
    return res
```

```
def newton_coefficient(x, y):
    m = len(x)
    x = np.copy(x)
    y = np.copy(y)
    for k in range(1, m):
        y[k:m] = (y[k:m] - y[k - 1]) / (x[k:m] - x[k - 1])
    return y


def newton_polynomial(x, y, x_cur):
    a = newton_coefficient(x, y)
    n = len(x) - 1
    p = a[n]
    for k in range(1, n + 1):
        p = a[n - k] + (x_cur - x[n - k]) * p
    return p


def t_calc(t, n, forward=True):
    result = t
    for i in range(1, n):
        if forward:
            result *= t - i
        else:
            result *= t + i
    return result


def newton_interpolation(x, y, x_cur):
    n = len(x)
    is_equally_spaced = True
    h = x[1] - x[0]
    for i in range(1, n - 1):
```

```
        if round(x[i + 1] - x[i], 3) != h:
            is_equally_spaced = False
            break
    if not is_equally_spaced:
        return 'Узлы не являются равноотстоящими'
    a = [[0] * n for _ in range(n)]
    for i in range(n):
        a[i][0] = y[i]
    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_cur <= x[n // 2]:
        x0 = n - 1
        for i in range(n):
            if x_cur <= x[i]:
                x0 = i - 1
                break
        if x0 < 0:
            x0 = 0
        t = (x_cur - x[x0]) / h
        result = a[x0][0]
        for i in range(1, n):
            result += (t_calc(t, i) * a[x0][i]) / math.factorial(i)
    else:
        t = (x_cur - x[n - 1]) / h
        result = a[n - 1][0]
        for i in range(1, n):
            result += (t_calc(t, i, False) * a[n - i - 1][i]) / math.factorial(i)
    return result


def stirling_interpolation(x, y, x_cur):
    is_equally_spaced = True
```

```
h = round(x[1] - x[0], 3)
n = len(x)
for i in range(1, n - 1):
    if round(x[i + 1] - x[i], 3) != h:
        is_equally_spaced = False
        break
if not is_equally_spaced:
    return 'Узлы не являются равноотстоящими'
a = [[0] * n for _ in range(n)]
for i in range(n):
    a[i][0] = y[i]
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 n % 2 == 0:
    x_0 = int(n / 2 - 1)
else:
    x_0 = int(n / 2)
t = (x_cur - x[x_0]) / h
if abs(t) > 0.25:
    print('t > 0.25 => результат Стирлинга может быть неточным')
n = a[x_0][0]
comp_t1 = t
comp_t2 = t**2
pr_number = 0
for i in range(1, len(x)):
    if i % 2 == 0:
        n += (comp_t2 / math.factorial(i)) * a[x_0 - (i // 2)][i]
        comp_t2 *= (t**2 - pr_number**2)
    else:
        n += (comp_t1 / math.factorial(i)) * \
            ((a[x_0 - ((i + 1) // 2)][i] + a[x_0 - (((i + 1) // 2) - 1)][i]) / 2)
        pr_number += 1
```

```
        comp_t1 *= (t**2 - pr_number**2)
    return n


def bessel_interpolation(x, y, x_cur):
    is_equally_spaced = True
    h = round(x[1] - x[0], 3)
    n = len(x)
    a = [[0] * n for _ in range(n)]
    for i in range(n):
        a[i][0] = y[i]
    for i in range(1, n):
        for j in range(n - i):
            a[j][i] = a[j + 1][i - 1] - a[j][i - 1]
    for i in range(1, n - 1):
        if round(x[i + 1] - x[i], 3) != h:
            is_equally_spaced = False
            break
    if not is_equally_spaced:
        return 'Узлы не являются равноотстоящими'
    if n % 2 == 0:
        x_0 = int(n / 2 - 1)
    else:
        x_0 = int(n / 2)
    t = (x_cur - x[x_0]) / h
    if abs(t) < 0.25 or abs(t) > 0.75:
        print('t < 0.25 или t > 0.75 => результат Бесселя может быть неточным')
    n = (a[x_0][0] + a[x_0 + 1][0]) / 2
    n += (t - 0.5) * a[x_0][1]
    comp_t = t
    last_number = 0
    for i in range(2, len(x)):
        if i % 2 == 0:
```

```
            last_number += 1
            comp_t *= (t - last_number)
            n += (comp_t / math.factorial(i)) * ((a[x_0 - i // 2][i] + a[x_0 - ((i//2) - 1)][i]) / 2)
        else:
            n += (comp_t * (t - 0.5) / math.factorial(i)) * a[x_0 - ((i - 1)//2)][i]
            comp_t *= (t + last_number)
    return n
```

Примеры и результаты работы программы

Введите 1, если ввод данных будет происходить из файла. Введите 2, если с клавиатуры. Введите 3 для выбора уравнения 1

Конечные разности: [[1.0, -1.0, 2.0, 1.0], [-2.0, 3.0, -1.0], [5.0, -4.0], [-9.0]]

Введите значение аргумента 2

t < 0.25 или t > 0.75 => результат Бесселя может быть неточным

Полином Лагранжа дал ответ: 5.0

Полином Ньютона с разделенными разностями дал ответ: 5.0

Полином Ньютона с конечными разностями дал ответ: 5.0

Многочлен Стирлинга дал ответ: 5.0

Многочлен Бесселя дал ответ: 5.0

Введите 1, если ввод данных будет происходить из файла. Введите 2, если с клавиатуры. Введите 3 для выбора уравнения3

1. sin(x)

2. x ** 2

Выберите уравнение (1 или 2) 2

Введите исследуемый интервал 1 4

Введите количество точек на интервале 6

Конечные разности: [[1.25, 1.75, 2.25, 2.75, 3.25], [0.5, 0.5, 0.5, 0.5], [0.0, 0.0, 0.0], [0.0, 0.0], [0.0]]

Введите значение аргумента 2

t < 0.25 или t > 0.75 => результат Бесселя может быть неточным

Полином Лагранжа дал ответ: 4.0

Полином Ньютона с разделенными разностями дал ответ: 4.0

Полином Ньютона с конечными разностями дал ответ: 4.0

Многочлен Стирлинга дал ответ: 4.0

Многочлен Бесселя дал ответ: 4.0

Вычислительная часть

$x_0 = 0{,}25 \quad y_0 = 1{,}2557$

$x_1 = 0{,}3 \quad y_1 = 2{,}1764$

$x_2 = 0{,}35 \quad y_2 = 3{,}1218$

$x_3 = 0{,}4 \quad y_3 = 4{,}0482$

$x_4 = 0{,}45 \quad y_4 = 5{,}9875$

$x_5 = 0{,}5 \quad y_5 = 6{,}9195$

$x_6 = 0{,}55 \quad y_6 = 7{,}8359$

$X_1 = 0{,}251 \quad X_2 = 0{,}402$

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
0,25	1,2557	0,9207	0,0247	-0,0437	1,0756	-4,1277	10,1917
0,3	2,1764	0,9454	-0,019	1,0319	-3,0521	6,064	
0,35	3,1218	0,9264	1,0129	-3,0202	3,0119		
0,4	4,0482	1,9393	-1,0073	0,9917			
0,45	5,9875	0,932	-0,0156				
0,5	6,9195	0,9164					
0,55	7,8359						

X_1 находится левее середины отрезка

$$t = \frac{x - x_0}{h} = \frac{0{,}251 - 0{,}25}{0{,}05} = 0{,}02$$

$$N_6 = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 + \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)(t-5)}{6!}\Delta^6 y_0$$

$$Y_1 = 1{,}2557 + 0{,}02 \cdot 0{,}9207 + \frac{0{,}02 \cdot (-0{,}98)}{2} \cdot 0{,}0247 + \frac{0{,}02(-0{,}98)(-1{,}98)}{6} \cdot (-0{,}0437) +$$

$$+ \frac{0{,}02(-0{,}98)(-1{,}98)(-2{,}98)}{24} \cdot 1{,}0756 + \frac{0{,}02(-0{,}98)(-1{,}98)(-2{,}98)(-3{,}98)}{120} \cdot (-4{,}1277) +$$

$$+ \frac{0{,}02(-0{,}98)(-1{,}98)(-2{,}98)(-3{,}98)(-4{,}98)}{720} \cdot 10{,}1917 = 1{,}22$$

X_2 находится правее середины отрезка

$$P_6 = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{t(t-1)(t+1)}{3!}\Delta^3 y_{-1} + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} +$$

$$+ \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)(t-3)}{6!}\Delta^6 y_{-3}$$

$$t = \frac{0{,}402 - 0{,}4}{0{,}05} = 0{,}04$$

$$Y_2 = 4{,}0482 + 0{,}04 \cdot 1{,}9393 + \frac{0{,}04 \cdot (-0{,}96)}{2!} \cdot 1{,}0129 + \frac{1{,}04 \cdot 0{,}04 \cdot (-0{,}96)}{3!}(-3{,}0202)$$

$$+ \frac{1{,}04 \cdot 0{,}04 \cdot (-0{,}96)(-1{,}96)}{4!}(-3{,}0521) + \frac{2{,}04 \cdot 1{,}04 \cdot 0{,}04 \cdot (-0{,}96) \cdot (-1{,}96)}{5!} \cdot 6{,}064 +$$

$$+ \frac{2{,}04 \cdot 1{,}04 \cdot 0{,}04 \cdot (-0{,}96)(-1{,}96)(-2{,}96)}{6!} \cdot 10{,}1917 = 4{,}111$$

Вывод

В результате выполнения данной лабораторной работы были изучены различные методы интерполяции функций.