МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

Численное дифференцирование функций

Отчет

по лабораторной работе №5

по дисциплине

Вычислительная математика

РУКОВОДИТЕЛЬ:

________________ Суркова А. С.

СТУДЕНТ:

________________ Соляник Д. Р.

19-ИВТ-2

Работа защищена «___» _____________

С оценкой _________________________

Нижний Новгород 2021

1. Тема лабораторной работы:

Численное дифференцирование функций

Цель лабораторной работы

Закрепление знаний и умений по численному дифференцированию функций с помощью интерполяционного многочлена Ньютона и метода неопределенных коэффициентов.

2. Вариант задания на лабораторную работу

Вариант № 18

Найти первую и вторую производную функции в точках x, заданных таблицей, используя интерполяционные многочлены Ньютона. Сравнить со значениями производных, вычисленными по формулам, основанным на интерполировании многочленом Лагранжа (вычисление производных через значения функций).

18.

x	y
0.01	0.991824
0.06	0.951935
0.11	0.913650
0.16	0.876905
0.21	0.841638
0.26	0.807789
0.31	0.775301
0.36	0.744120
0.41	0.714193
0.46	0.685470
0.51	0.657902
0.56	0.631442

3. Теоретические сведения и описание лабораторной работы

- *Ньютон*

3. Использование интерполяционных формул. Предположим, что функция $f(x)$, заданная в виде таблицы с постоянным шагом $h = x_i - x_{i-1}$ $(i = 1, 2, \ldots, n)$, может быть аппроксимирована интерполяционным многочленом Ньютона (2.39):

$$y \approx N(x_0 + th) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \ldots$$

$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0, \quad t = \frac{x - x_0}{h}.$$

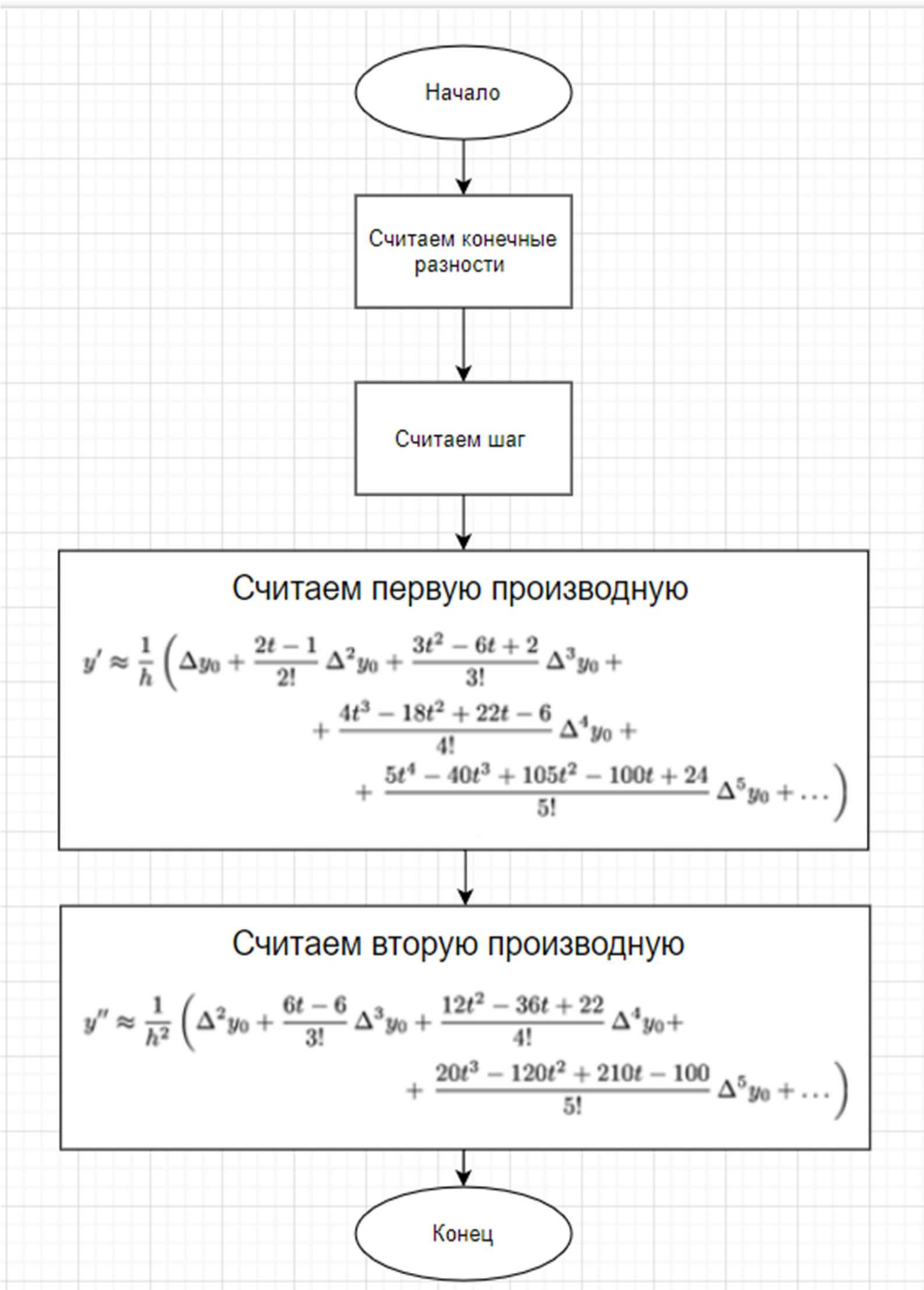
Дифференцируя этот многочлен по переменной x с учетом правила дифференцирования сложной функции:

$$\frac{dN}{dx} = \frac{dN}{dt}\frac{dt}{dx} = \frac{1}{h}\frac{dN}{dt},$$

можно получить формулы для вычисления производных любого порядка:

$$y' \approx \frac{1}{h}\left(\Delta y_0 + \frac{2t-1}{2!}\Delta^2 y_0 + \frac{3t^2-6t+2}{3!}\Delta^3 y_0 + \right.$$
$$+ \frac{4t^3-18t^2+22t-6}{4!}\Delta^4 y_0 +$$
$$\left. + \frac{5t^4-40t^3+105t^2-100t+24}{5!}\Delta^5 y_0 + \ldots\right),$$

$$y'' \approx \frac{1}{h^2}\left(\Delta^2 y_0 + \frac{6t-6}{3!}\Delta^3 y_0 + \frac{12t^2-36t+22}{4!}\Delta^4 y_0 + \right.$$
$$\left. + \frac{20t^3-120t^2+210t-100}{5!}\Delta^5 y_0 + \ldots\right),$$



- *Лагранж*

Вычисление производной на основе интерполяционного многочлена Лагранжа применяется, когда аналитическое выражение функции $y=f(x)$ не известно, а функция $y=f(x)$ задана таблично.

Пусть функция $y=f(x)$ определена на отрезке и в точках $\{x_i\}$ ($i=0,1,2,\ldots,n$) этого отрезка принимает значения $y_i=f(x_i)$.

Разность между соседними значениями аргумента x_i постоянна и является шагом $h=x_i-x_{i-1}$ ($i=1, \ldots,n$) разбиения отрезка на n частей, прием $a=x_0$ и $b=x_n$.

Найдем аппроксимации производной первого порядка с помощью значений функций y_i в узловых точках x_i.

Для того чтобы выразить значения производной через значения функции y_i в узлах интерполяции x_i, построим интерполяционный многочлен Лагранжа $L_m(x)$ степени m, удовлетворяющий условиям

$L_m(x)= f(x_k)= y_k$ ($k=i, i+1, \ldots, i+m$), $i+m£n$

Многочлен Лагранжа $L_m(x)$ интерполирует функцию $f(x)$ на отрезке $[x_i, x_{i+m}]$. Дифференцируя многочлен $L_m(x)$, получаем значения производной в точках $\{x_i\}$ ($k=i, i+1, \ldots, i+m$).

Если $m=2$, то график интерполяционного многочлена Лагранжа $L_2(x)$ – парабола, проходящая через три точки (x_i, y_i), (x_{i+1}, y_{i+1}) и (x_{i+2}, y_{i+2}).

Вычислим первую производную многочлена $L_2(x)$ на отрезке $[x_i, x_{i+2}]$:

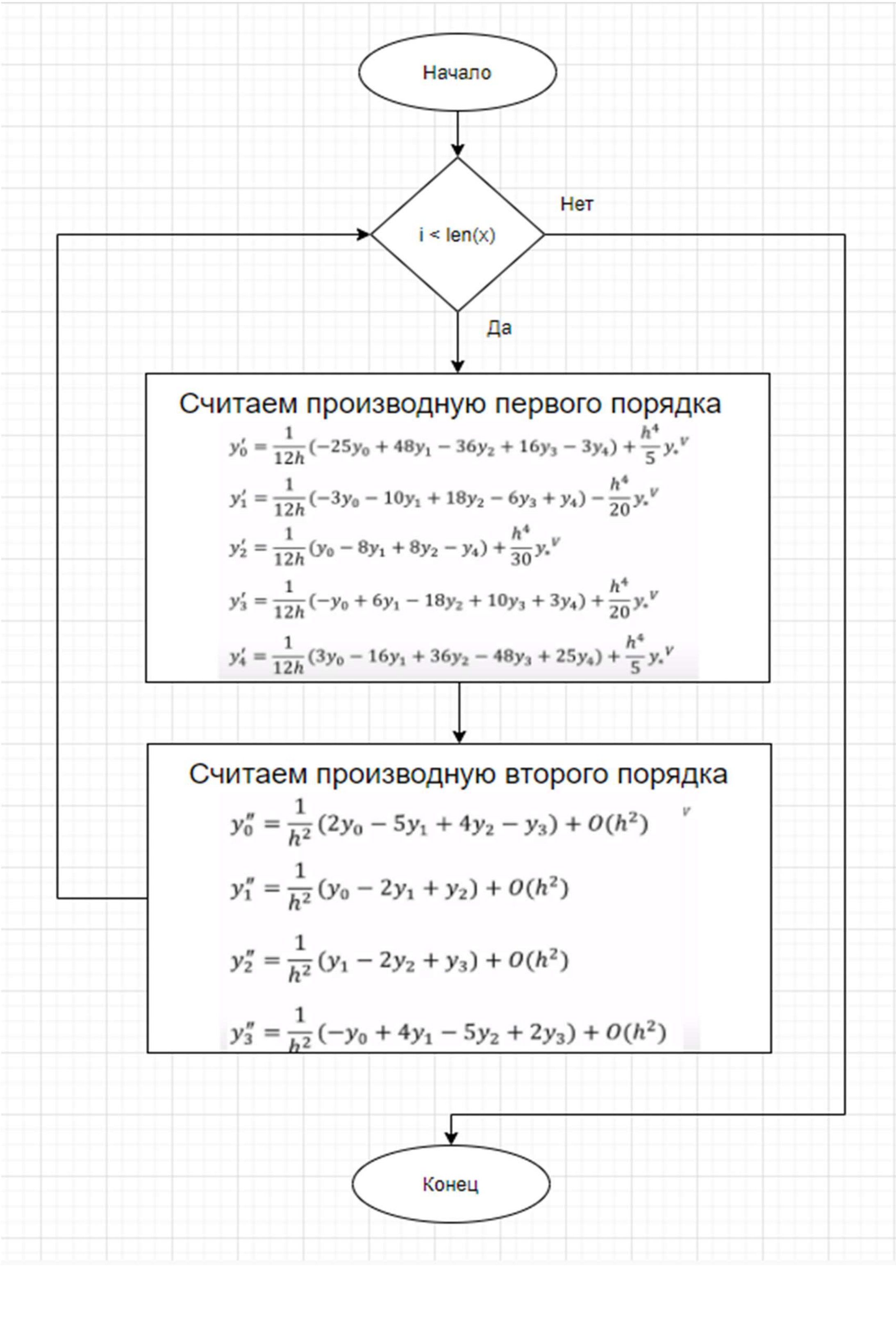
$$L_2(x)=\frac{1}{2h^2}\left[y_i(x-x_{i+1})(x-x_{i+2})-2y_{i+1}(x-x_i)(x-x_{i+2})+y_{i+2}(x-x_i)(x-x_{i+1})\right]$$

$$L_2'(x)=\frac{1}{2h^2}\left[y_i(2x-x_{i+1}-x_{i+2})-2y_{i+1}(2x-x_i-x_{i+2})+y_{i+2}(2x-x_i-x_{i+1})\right]$$

$$L_2''(x)=\frac{1}{h^2}\left[y_i-2y_{i+1}+y_{i+2}\right]$$

Производная многочлена $L_2(x)$ в точках x_i, x_{i+1}, x_{i+2} является приближением производной функции $f(x)$ в этих точках:

$$
\begin{cases}
y_i' = f'(x_i) \approx L_2'(x_i) = \dfrac{1}{2h}(-3y_i + 4y_{i+1} - y_{i+2}), \\
y_{i+1}' = f'(x_{i+1}) \approx L_2'(x_{i+1}) = \dfrac{1}{2h}(-y_i + y_{i+2}), \\
y_{i+2}' = f'(x_{i+2}) \approx L_2'(x_{i+2}) = \dfrac{1}{2h}(y_i - 4y_{i+1} + 3y_{i+2})
\end{cases}
$$



4. Листинг разработанной программы

- *Ньютон*

```
mport math


# Задаем массив значений x
x = [0.01, 0.06, 0.11, 0.16, 0.21, 0.26, 0.31, 0.36, 0.41, 0.46, 0.51, 0.56]


# Задаем массив значений y
y = [[0 for i in range(len(x))] for j in range(len(x))]
y[0][0] = 0.991824; y[1][0] =  0.951935; y[2][0] = 0.91365; y[3][0] = 0.876905;
y[4][0] = 0.841638; y[5][0] = 0.807789; y[6][0] = 0.775301; y[7][0] = 0.74412;
y[8][0] = 0.714193; y[9][0] = 0.68547; y[10][0] = 0.657902; y[11][0] = 0.631442;


def factorial(n):
    "Функция нахождения факториала"
    f = 1
    for i in range(2, n + 1):
        f *= i
    return f


def proiz_1(xx):
    "Функция вычисления первой производной интерполяцым многочленом Ньютона"
    h = x[1] - x[0] # Вычисляем шаг
    t = (xx - x[0]) / h

    sum = y[0][1]
    sum += ((2 * t - 1) / factorial(2)) * y[0][2]
    sum += (((3 * t ** 2 - 6 * t + 2) / factorial(3)) * y[0][3])
    sum += ((4 * t ** 3 - 18 * t ** 2 + 22 * t - 6) / factorial(4)) * y[0][4]
    sum += ((5 * t ** 4 - 40 * t ** 3 + 105 * t ** 2 - 100 * t + 24) / factorial(5)) * y[0][5]
    return sum / h
```

```
def proiz_2(xx):
    "Функция вычисления второй производной интерполяцым многочленом Ньютона"
    h = x[1] - x[0]
    t = (xx - x[0]) / h

    sum = y[0][2]
    sum += ((6 * t - 6) / factorial(3)) * y[0][3]
    sum += (((12 * t ** 2 - 36 * t + 22) / factorial(4)) * y[0][4])
    sum += ((20 * t ** 3 - 120 * t ** 2 + 210 * t - 100) / factorial(5)) * y[0][5]
    return sum / (h ** 2)

# Расчитываем значения прясых разниц
for i in range(1, len(x)):
    for j in range(len(x) - i):
        y[j][i] = y[j + 1][i - 1] - y[j][i - 1]

# for i in range(len(x)):
#     print("%.3f" %x[i], end = "\t")
#     for j in range(len(x) - i):
#         print("%.8f" %y[i][j], end = "\t")
#     print("")

print("  x   |   y    |     y'   |     y''  ")
for i in range(len(x)):
    print("", x[i], "|", "%.4f" % y[i][0], "|", "%.4f" % proiz_1(x[i]),"|", "%.4f" % proiz_2(x[i]))
```

- *Лагранж*

```
x = [0.01, 0.06, 0.11, 0.16, 0.21, 0.26, 0.31, 0.36, 0.41, 0.46, 0.51, 0.56]
y = [0.991824, 0.951935, 0.91365, 0.876905, 0.841638, 0.807789, 0.775301, 0.74412, 0.714193,
0.68547, 0.657902, 0.631442]

def proiz_1(xx, yy):
"Функция вычисления первой производной интерполяцым многочленом Ньютона"
```

```
    ans = []
    h = xx[1] - xx[0]

    # Вычисляем производную в начальных точках
    res = (-25 * yy[0] + 48 * yy[1] - 36 * yy[2] + 16 * yy[3] - 3 * yy[4]) / (12 * h)
    ans.append(res)
    res = (-3 * yy[0] - 10 * yy[1] + 18 * yy[2] - 6 * yy[3] + yy[4]) / (12 * h)
    ans.append(res)

    # Вычисляем производную в средних точках
    for i in range(2, 10):
        res = (yy[i - 2] - 8 * yy[i - 1] + 8 * yy[i + 1] - yy[i + 2]) / (12 * h)
        ans.append(res)

    # Вычисляем производную в последних точках
    res = (-yy[7] + 6 * yy[8] - 18 * yy[9] + 10 * yy[10] + 3 * yy[11]) / (12 * h)
    ans.append(res)
    res = (3 * yy[7] - 16 * yy[8] + 36 * yy[9] - 48 * yy[10] + 25 * yy[11]) / (12 * h)
    ans.append(res)
    return ans

def proiz_2(xx, yy):
"Функция вычисления второй производной интерполяцым многочленом Ньютона"
    ans = []
    h = xx[1] - xx[0]

    # Вычисляем производную в начальной точке
    res = (2.0 * yy[0] - 5.0 * yy[1] + 4.0 * yy[2] - yy[3]) / (h ** 2)
    ans.append(res)

    # Вычисляем производную в средних точках
    for i in range(1, 11):
```

```
            res = (yy[i - 1] - 2 * yy[i] + yy[i + 1]) / (h ** 2)
            ans.append(res)

        # Вычисляем производную в конечной точке
        res = (-yy[8] + 4 * yy[9] - 5 * yy[10] + 2 * yy[11]) / (h ** 2);
        ans.append(res)
        return ans

print("  x   |   y    |    y'   |   y''  ")
for i in range(len(x)):
    print("", x[i], "|", "%.4f" % y[i], "|", "%.4f" % proiz_1(x, y)[i],"|", "%.4f" % proiz_2(x, y)[i])
```

5. Результаты работы программы

- *Ньютон*

x	y	y'	y''
0.01	0.9918	-0.8143	0.6679
0.06	0.9519	-0.7815	0.6415
0.11	0.9136	-0.7501	0.6159
0.16	0.8769	-0.7199	0.5911
0.21	0.8416	-0.6910	0.5671
0.26	0.8078	-0.6632	0.5439
0.31	0.7753	-0.6366	0.5215
0.36	0.7441	-0.6110	0.4999
0.41	0.7142	-0.5865	0.4791
0.46	0.6855	-0.5631	0.4591
0.51	0.6579	-0.5406	0.4399
0.56	0.6314	-0.5191	0.4215

- *Лагранж*

x	y	y'	y''
0.01	0.9918	-0.8143	0.6672
0.06	0.9519	-0.7815	0.6416
0.11	0.9136	-0.7501	0.6160
0.16	0.8769	-0.7199	0.5912
0.21	0.8416	-0.6910	0.5672
0.26	0.8078	-0.6632	0.5444
0.31	0.7753	-0.6365	0.5228
0.36	0.7441	-0.6109	0.5016
0.41	0.7142	-0.5863	0.4816
0.46	0.6855	-0.5628	0.4620
0.51	0.6579	-0.5401	0.4432
0.56	0.6314	-0.5184	0.4244

6. Вывод

В ходе данной лабораторной работы были закреплены знания и умения по численному дифференцированию функций с помощью интерполяционного многочлена Ньютона и метода неопределенных коэффициентов