МІНІСТЕРСТВО НАУКИ ТА ОСВІТИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Розрахункова робота №2

З предмету: **“Чисельні методи ч1”**

на тему:

“Дослідження похибки інтерполяції функцій многочленом Ньютона”

Варіант №27

Виконав

студент групи ПМ-33

Венгринюк Олег

Перевірила

Тесак І.Є

Завдання:

Для функції $10\,ch(\frac{x}{2})$ заданої на рівномірній сітці в точках $a=x_0, x_{1,\dots}, x_n=b$ на інтервалі $[0,3]$

1. знайти аналітичний вираз залишкового члена інтерполяційного многочлена;
2. обчислити його максимальне значення і значення в точках (x', x'', x''' - не співпадають з вузлами інтерполяції) і в т. x_k – вузлі інтерполяції;
3. побудувати інтерполяційний многочлен Ньютона через розділені різниці (використати програму з лабораторної роботи №3);
4. обчислити похибки інтерполяції в точка x', x'', x''' , x_k і порівняти їх із значеннями залишкового члена.

Виконання:

1.Знайдемо аналітичний вираз залишкового члена інтерполяційного многочлена

Залишковий член: $R_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(x')*w_{(n+1)}(t)}{(n+1)!}, x' \in [a,b]$, де

$w_{(n+1)}(t) = (t - x_0)*(t - x_1)*(t - x_2)*...*(t - x_n)$

Задана функція: $f(x) = 10\text{ch}(\frac{x}{2})$

Похідна степеня n+1 (5-го) степеня $f^{(5)} = \frac{5*sh(\frac{x}{2})}{16}$

Многочлен $W_{(5)}(t) = (t-0)*(t-0.75)*(t-1.5)*(t-2.25)$

Залишнковий член у нашому випадку має вигляд:

$$R_n(x) = \frac{sh(\frac{x}{2})*(t-0)*(t-0.75)*(t-1.5)*(t-2.25)}{16*4!}$$

2. Використовуючи засоби Maple обчислимо максимальне значення і значення у точках , x', x'', x''' що не співпадають з вузлами інтерполяції і в точці – x_k вузлі інтерполяції;

```
> restart;
> f(t):=10*cosh(t/2);
```

$$f(t) := 10\cosh\left(\frac{1}{2}\,t\right) \quad \textbf{(1)}$$

```
> n:=4;
```

$$n := 4 \quad \textbf{(2)}$$

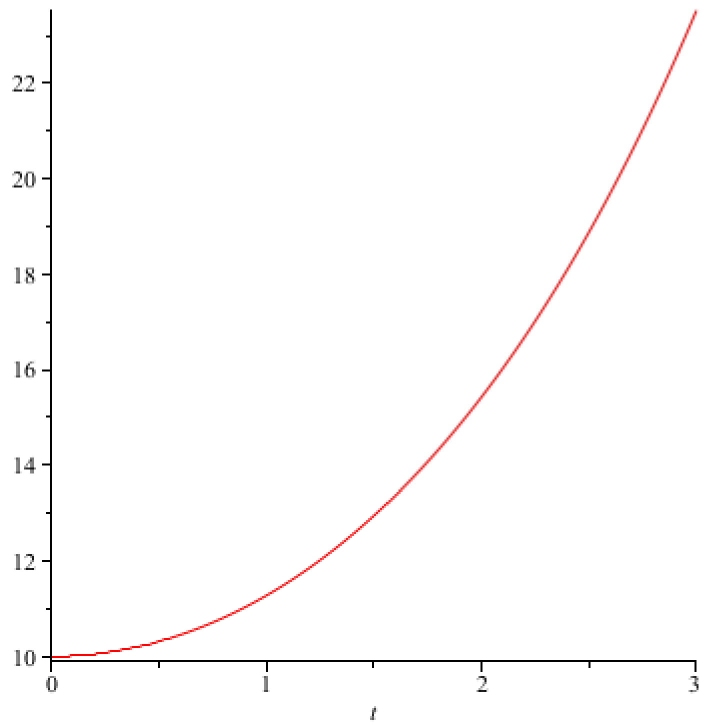
```
> a:=0;
```

$$a := 0 \quad \textbf{(3)}$$

```
> b:=3;
```

$$b := 3 \quad \textbf{(4)}$$

> $plot(f(t), t = a..b);$




```
> df(t):=diff(f(t),t$n+1);
```

Рис 1.ініціалізація функції та побудова графіку

$$df(t) := \frac{5}{16}\sinh\left(\frac{1}{2}\,t\right) \tag{5}$$

```
> plot(df(t),t=a..b);
```

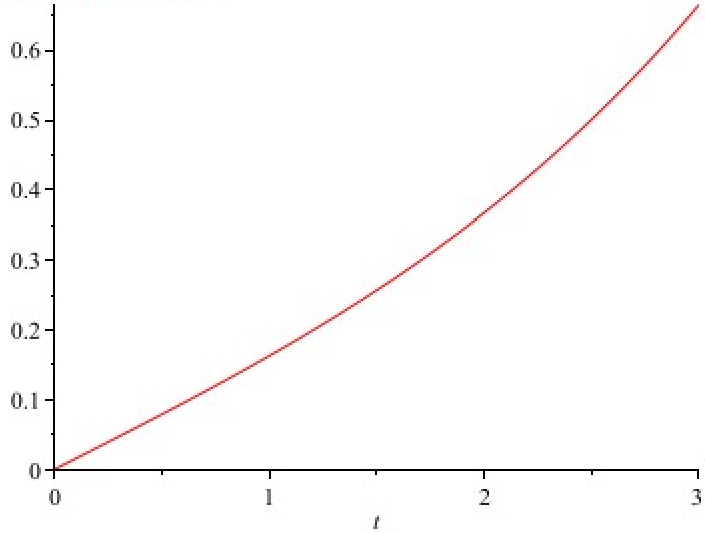


Рис 2. побудова графіку похідної 5-го степеня початкової функції

```
> M:=evalf(maximize(abs(diff(f(t),t$n+1)),t=a..b));
```

$$M := 0.6653998297 \tag{6}$$

```
> h:=(b-a)/n;
```

$$h := \frac{3}{4} \tag{7}$$

```
> w:=t-a;
```

$$w := t \tag{8}$$

```
> for i from 1 to n do w:=w*(t-a-i*h) end do;
```

$$w := t\left(t-\frac{3}{4}\right)$$

$$w := t\left(t-\frac{3}{4}\right)\left(t-\frac{3}{2}\right)$$

$$w := t\left(t-\frac{3}{4}\right)\left(t-\frac{3}{2}\right)\left(t-\frac{9}{4}\right)$$

$$w := t\left(t-\frac{3}{4}\right)\left(t-\frac{3}{2}\right)\left(t-\frac{9}{4}\right)(t-3) \tag{9}$$

Рис 3. пошук максимуму поіхдної та обчислення многочлена w

```
> plot(w,t=a..b);
```

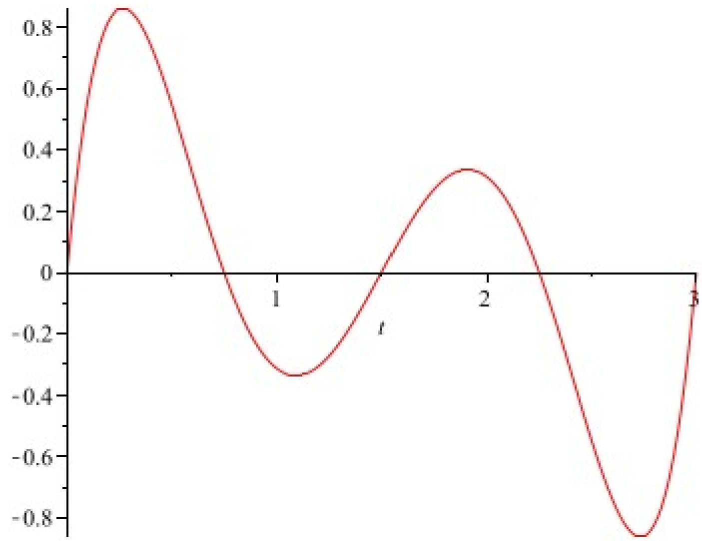


Рис 4. побудова графіку многочлена w

```
> Wmax:=evalf(maximize(abs(w),t=a..b));
```

$$Wmax := 0.8617558852$$

```
> R:=M*Wmax/(n+1)!;
```

$$R := 0.004778435161$$

Рис 5. обчислення максимум залишкового члена

> $R\_calc :=$ **proc**$(M,\ a, b, n, t\ )$
>
> $h := \frac{(b-a)}{n}$:
>
> $w := evalf(t-a)$:
>
> **for** i **from** 1 **to** n **do** $w := evalf(w * (t-a-i*h))$ **end do**:
>
> $R := \frac{M \cdot w}{(n+1)!}$
>
> **end proc**:

Рис 6. визначення процедури для обчислення значення залишкового члена

>

$Rn(0.3) = R\_calc(M, a, b, n, 0.3);$
$Rn(0.5) = R\_calc(M, a, b, n, 0.5);$
$Rn(1.75) = R\_calc(M, a, b, n, 1.75);$

$Rn(a+h) = R\_calc(M, a, b, n, a+h);$

$$Rn(0.3) = 0.004729495639$$
$$Rn(0.5) = 0.003032421099$$
$$Rn(1.75) = 0.001516210549$$
$$Rn\left(\frac{3}{4}\right) = -0.$$

.

Рис 7. обчислення значення залишкового члена у невузлових точка та у вузі a+h

3. Побудуємо інтерполяційний многочлен Ньютона через розділені різниці та порівняємо отримані похибки зі значенням аналітично знайденого залишкового члена. Використовуючи мову програмування Python3 та пакети numpy, matplotlib, виконаємо поставлене завдання.

Програмний код

```
import matplotlib.pyplot as plt
import numpy as np
import texttable as tt


def divided_diff(x: np.ndarray, y: np.ndarray) -> tuple:
    n = y.shape[0]

    result = [[] for i in range(n)]
    result[0] = y

    for k in range(n-1):
        for i in range(1, len(result[k])):
            divided_diff = (result[k][i]-result[k][i-1])/(x[i+k] - x[i-1])
            result[k+1].append(divided_diff)

    coefs = [diff[0] for diff in result]

    return coefs, result


def L(x: float, x_i: list, f_diff: list) -> float:
    x_i = np.insert(x_i, 0, x-1)
    res = f_diff[-1]
    res *= (x-x_i[-1])
    for i in range(len(f_diff)-1, -1, -1):
        res += f_diff[i]
        res *= x-x_i[i]

    return res


def print_divided_diff_table(x: np.ndarray, y: np.ndarray, f_coefs: list) -> None:
    f_coefs_copy = [[val for val in f_coef] for f_coef in f_coefs]

    tab = tt.Texttable()
```

```
    # ----------- set headers
    headings = ['X', 'Y']
    for i in range(1, y.shape[0]):
        head = 'f(x_i;...;x_i+{})'.format(i)
        headings.append(head)

    tab.header(headings)

    values = [x]

    # --------- cal divided differences

    for i in range(len(f_coefs_copy)):
        f_coefs_copy[i].extend([""]*(y.shape[0]-len(f_coefs_copy[i])))
        values.append(f_coefs_copy[i])

    for row in zip(*values):
        tab.add_row(row)

    s = tab.draw()
    print(s)


def f(x: float) -> float:
    return np.cosh(x/2)/10


def print_polynomial(X, f_coefs) -> None:

  polinomial = 'Ln(x) = {:.3f}+'

  template = '(x-{:.3f})({:.3f}*'
  template_end = '(x-{:.3f}){:.3f}'
  args = []

  for i in range(len(f_coefs)-2):
    polinomial += template

  polinomial += template_end
  polinomial += ")"*(len(f_coefs)-2)

  for i in range(0, len(f_coefs)):
    args.append(f_coefs[i])
```

```
        args.append(X[i])

    print(polinomial.format(*args))


def print_difference(X_test, analytic_difference, X, Y, x_diff) -> None:
    L_x = np.array([L(x, X, x_diff) for x in X_test])
    f_res = f(X_test)
    difference = abs(f_res - L_x)

    tab = tt.Texttable()
    headings = ['x', 'f(x)', 'L(x)', 'R(x)', 'teoretical R(x)']
    tab.header(headings)
    values = [X_test, f_res, L_x, difference, analytic_difference]
    for row in zip(*values):
        tab.add_row(row)
    s = tab.draw()
    print(s)


if __name__ == '__main__':

    a, b = 0, 3
    n = 4
    X = np.linspace(a, b, n+1)
    Y = f(X)

    f_coefs, divided_diffs = divided_diff(X, Y)

    X_test = np.array([0.3, 0.5, 1.75, 0.75])
    analytic_difference = [0.0047, 0.003,  0.0015, 0.]

    print("Netwon's polynomial")
    print_polynomial(X, f_coefs)
    print("Table of divided differences")
    print_divided_diff_table(X, Y, divided_diffs)
    print('Difference between function and interpolation polynomial')
    print_difference(X_test, analytic_difference, X, Y, f_coefs)

    debug = True
    if debug:
        fig = plt.figure()
        plt.plot(X, Y, "ob", markersize=5)
```

```
X_interp = np.linspace(X[0], X[-1], 100)
plt.plot(X_interp, [L(x, X, f_coefs) for x in X_interp], 'oy', markersize=2)
plt.legend(["Table point", "Interpolation"])
plt.show()
```

Результат роботи програми:

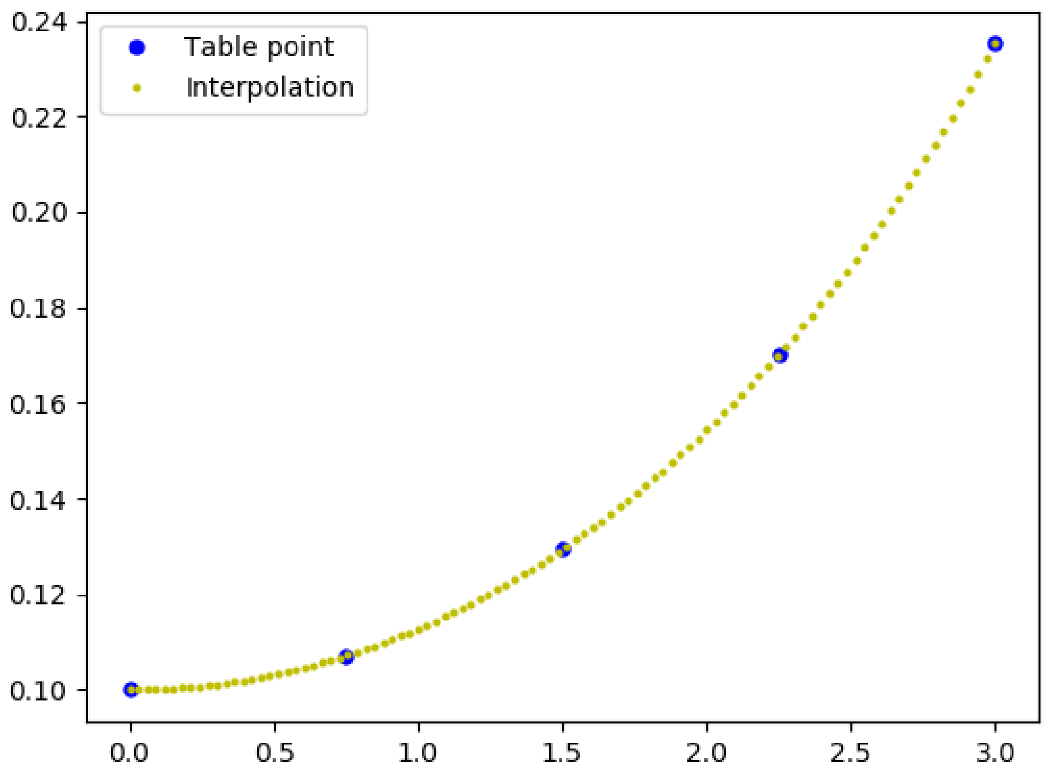


Рис 8. графік інтерполяційного многочлена та точок інтерполяції

```
Netwon's polynomial
Ln(x) = 0.100+(x-0.000)(0.009*(x-0.750)(0.014*(x-1.500)(0.001*(x-2.250)0.000)))
Table of divided differences
+-------+-------+---------------+---------------+---------------+--------------+
|   X   |   Y   | f(x_i;...;x_i | f(x_i;...;x_i | f(x_i;...;x_i | f(x_i;...;x_ |
|       |       |      +1)      |      +2)      |      +3)      |     i+4)     |
+=======+=======+===============+===============+===============+==============+
| 0     | 0.100 | 0.009         | 0.014         | 0.001         | 0.000        |
+-------+-------+---------------+---------------+---------------+--------------+
| 0.750 | 0.107 | 0.030         | 0.016         | 0.002         |              |
+-------+-------+---------------+---------------+---------------+--------------+
| 1.500 | 0.129 | 0.054         | 0.022         |               |              |
+-------+-------+---------------+---------------+---------------+--------------+
| 2.250 | 0.170 | 0.087         |               |               |              |
+-------+-------+---------------+---------------+---------------+--------------+
| 3     | 0.235 |               |               |               |              |
+-------+-------+---------------+---------------+---------------+--------------+
Difference between function and interpolation polynomial
+-------+-------+-------+-------+-----------------+
|   x   | f(x)  | L(x)  | R(x)  | teoretical R(x) |
+=======+=======+=======+=======+=================+
| 0.300 | 0.101 | 0.101 | 0.000 | 0.005           |
+-------+-------+-------+-------+-----------------+
| 0.500 | 0.103 | 0.103 | 0.000 | 0.003           |
+-------+-------+-------+-------+-----------------+
| 1.750 | 0.141 | 0.141 | 0.000 | 0.002           |
+-------+-------+-------+-------+-----------------+
| 0.750 | 0.107 | 0.107 | 0     | 0               |
+-------+-------+-------+-------+-----------------+
```

Рис 9. результат виконання програми

Висновок: в ході виконання розрахункової роботи було освоєно дослідження похибки інтерполяції функцій многочленом Ньютона. Було виведено вигляд многочлена із коефіцієнтами з певною кількістю знаків після коми, задля спрощення вигляду многочлена Ньютона. Було виведено таблицю розділених різниць, комірки, що не мали значення відповідні розділеної різниці, містили порожній рядок. Було виведено в таблицю, для наочності, похибку у точках спостереження, що відповідають точкам спостержень із завдання 2. Було навмисно обрано крайню точку спостерження x = 10 – точку поза межами відрізку інтерполяції, на якому було проведено спостереження, оскільки інтерполяційний многочлен давав нульову похибку. Похибка практичного характеру не перевищує значення аналітичного залишкового члена у розгляданих точках, що дає підгрунття для прийняття даної програми як правильної.