Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра програмного забезпечення

ЗВІТ

Про виконання лабораторної роботи № 8

«НАБЛИЖЕННЯ ДИСКРЕТНИХ (ТАБЛИЧНО ЗАДАНИХ) ФУНКЦІЙ»

з дисципліни «Чисельні методи»

Лектор:

доцент кафедри ПЗ

Мельник Н.Б.

Виконав:

студ. групи ПЗ-15

Бабіля О.О.

Прийняв:

асистент кафедри ПЗ

Гарматій Г.Ю.

«___» ______ 2022 р.

∑ = ____ ___________

Львів – 2022

Мета: ознайомитися з методом інтерполяції таблично заданих функцій.

Теоретичні відомості:

Найпростіша задача інтерполяції полягає в тому, що на відрізку [a,b] задано (n +1) точок $x_0, x_1, \dots, x_n$, які називають вузлами інтерполяції, і значення деякої функції у цих точках

$$f(x_0) = y_0, \quad f(x_1) = y_1, \quad \dots, \quad f(x_n) = y_n.$$

Необхідно побудувати інтерполяційну функцію F(x) , яка приймає у вузлах інтерполяції ті самі значення, що й функція f (x) . Тобто треба знайти таку функцію F(x) , щоб

$$F(x_0) = y_0, \quad F(x_1) = y_1, \quad \dots, \quad F(x_n) = y_n.$$

Геометрично це означає, що треба знайти криву y = F(x) певного типу, яка проходить через задану систему точок.

Зауважимо, що через задану множину точок можна провести безліч гладких кривих. Тому задача інтерполяції є неоднозначною. Вона стає однозначною тоді, коли за інтерполяційну функцію вибрати поліном $P_n(x)$ n-го степеня, де степінь полінома на одиницю менший від к-ті вузлів інтерполяції, і такий, що виконуються умови

$$P_n(x_0) = y_0, \quad P_n(x_1) = y_1, \quad \dots, \quad P_n(x_n) = y_n.$$

Інтерполяційний поліном Лагранжа

Один з методів знаходження інтерполяційного полінома запропонував Лагранж. Основна ідея цього методу полягає в пошуку полінома, який в одному довільному вузлі інтерполяції приймає значення один, а в усіх інших вузлах – нуль.

Наближену функцію y = F(x) розглянемо у вигляді

$$F(x) = L_n(x) = \sum_{i=0}^{n} P_i(x) f(x_i),$$

де $P_i(x)$ - такий многочлен, що

$$P_i(x_j) = \begin{cases} 0, & i \neq j, \\ 1, & i = j, \end{cases} \qquad i, j = \overline{0, n}.$$

Оскільки точки $x_0, x_1, \dots, x_n$ є коренями полінома, то його можна записати у такому вигляді

$$P_i(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)},$$

а наближена функція F(x) , яку називають інтерполяційним многочленом Лагранжа, матиме вигляд

$$F(x) = L_n(x) = \sum_{i=0}^{n} \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)} f(x_i)$$

Для запису інтерполяційного полінома Лагранжа зручно використовувати таблицю:

$\Pi_{n+1}(x)$					D_i	y_i
$x-x_0$	x_0-x_1	x_0-x_2	...	x_0-x_n	D_0	y_0
x_1-x_0	$x-x_1$	x_1-x_2	...	x_1-x_n	D_1	y_1
x_2-x_0	x_2-x_1	$x-x_2$	...	x_2-x_n	D_2	y_2
...	...	...	...	...	...	...
x_n-x_0	x_n-x_1	x_n-x_2	...	$x-x_n$	D_n	y_n

Тут D_i – добуток елементів і –го рядка, $\Pi_{n+1}(x)$ – добуток елементів головної діагоналі.

Тоді поліном Лагранжа можна записати у вигляді

$$L_n(x) = \Pi_{n+1}(x) \sum_{i=0}^{n} \frac{y_i}{D_i}.$$

У випадку рівновіддалених вузлів вираз набуде форми

$$L_n(x) = \frac{1}{n!} \Pi_{n+1}(t) \sum_{i=0}^{n} (-1)^{n-i} \frac{C_n^i}{t-i} y_i,$$

де t – крок інтерполяції.

Інтерполяційний поліном Ньютона

Інший спосіб розв’язування задачі інтерполяції запропонував Ньютон. Цей спосіб полягає в тому, що поліном $P_n(x)$ для загального випадку нерівновіддалених вузлів записують у вигляді

$$P_n(x) = f(x_0) + f(x_0, x_1)(x-x_0) + f(x_0, x_1, x_2)(x-x_0)(x-x_1) +$$
$$+ ... + f(x_0, x_1, ..., x_n)(x-x_0)(x-x_1)...(x-x_{n-1}),$$

де

$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ – розділена різниця 1-го порядку,

$f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}$ – розділена різниця 2-го порядку,

$f(x_0, x_1, \dots, x_n) = \frac{f(x_1, \dots, x_n) - f(x_0, \dots, x_{n-1})}{x_n - x_0}$ – розділена різниця n-го порядку.

Для випадку рівновіддалених вузлів маємо вираз

$$P_n(x) = f(x_0) + \frac{\Delta f(x_0)}{1!h}(x - x_0) + \frac{\Delta^2 f(x_0)}{2!h^2}(x - x_0)(x - x_1) +$$

$$+ \dots + \frac{\Delta^n f(x_0)}{n!h^n}(x - x_0)(x - x_1)\dots(x - x_{n-1}).$$

де $\Delta f(x_0)$ – скінченна різниця першого порядку і обчислюється наступним чином

$$\Delta f(x_i) = f(x_{i+1}) - f(x_i).$$

Для обчислення скінченної різниці другого порядку використовуємо скінченні різниці першого порядку

$$\Delta^2 f(x_i) = \Delta(\Delta f(x_i)) = \Delta f(x_{i+1}) - \Delta f(x_i).$$

Аналогічно запишемо скінченну різницю n -го порядку

$$\Delta^n f(x_i) = \Delta(\Delta^{n-1} f(x_i)) = \Delta^{n-1} f(x_{i+1}) - \Delta^{n-1} f(x_i)$$

Індивідуальне завдання

Використовуючи інтерполяційні поліноми Лагранжа та Ньютона, обчислити значення табличної заданої функції у точці x_0 .

x	1,415	1,420	1,425	1,430	1,435	1,440	1,445	1,450	1,455	1,460
y	0,88	0,889	0,890	0,891	0,892	0,893	0,894	0,895	0,896	0,897

$x_0 = 1{,}4161$

Хід роботи:

Поліном Лагранжа

x	1,415	1,420	1,425	1,430	1,435	1,440	1,445	1,450	1,455	1,460
y	0,88	0,889	0,890	0,891	0,892	0,893	0,894	0,895	0,896	0,897

Запишемо загальну формулу для знаходження інтерполяційного полінома Лагранжа:

$L_n(x) = \sum_{i=0}^{n} l_i(x) y_i$, оскільки точок 10, то $n=9$

Знайдемо $l_0(x)$:

$$l_0(x) = \frac{(x-1,420)\cdot(x-1,425)\cdot(x-1,430)\cdot(x-1,435)\cdot(x-1,440)}{(1,415-1,420)\cdot(1,415-1,425)\cdot(1,415-1,430)\cdot(1,415-1,435)\cdot(1,415-1,440)} \cdot \frac{(x-1,445)\cdot(x-1,450)\cdot(x-1,455)\cdot(x-1,460)}{(1,415-1,445)\cdot(1,415-1,450)\cdot(1,415-1,455)\cdot(1,415-1,460)}$$

Далі нам потрібно провести такі обчислення для інших точок, та кожен з $l_i(x)$ домножити на y_i.

Після того для знаходження полінома нам потрібно додати кожен член між собою.

Поліном Ньютона

x	1,415	1,420	1,425	1,430	1,435	1,440	1,445	1,450	1,455	1,460
y	0,88	0,885	0,890	0,891	0,892	0,893	0,894	0,895	0,896	0,897

Оскільки вузли рівновіддалені $h = 0{,}005$, то запишемо форму інтерполяційну формулу Ньютона для рівновіддалених вузлів.

$$P_n(x) = f(x_0) + \frac{\Delta f(x_0)}{1!\,h} \cdot q + \ldots + \frac{\Delta^n f(x_0)}{n!\,h^n} q(q-1)\ldots(q-n+1)$$

$$q = \frac{x - x_0}{h}$$

Знайдемо перші дві скінченні різниці:

$$\Delta f(x_0) = f(x_1) - f(x_0) = 0{,}885 - 0{,}88 = 0{,}009$$

$$\Delta^2 f(x_0) = \Delta(\Delta f(x_0)) = \Delta f(x_1) - \Delta f(x_0)$$

$$\Delta f(x_1) = f(x_2) - f(x_1) = 0{,}890 - 0{,}889 = 0{,}001$$

$$\Delta^2 f(x_0) = 0{,}001 - 0{,}009 = -0{,}008.$$

Далі нам потрібно обчислити інші скінченні різниці і підставити їх у формулу.

Код програми:

```
                                                }
void methodLagrange (double x0, double X[10], double Y[10]) {
    if (x0 == 0)  return;
    if (is(X) == false) {
        return;
    }
    printf("\n               Lagrangemethod:\n");
    double Q, deltaY[9][9], H = X[1] - X[0];
    for (int i = 0; i < 9; i++)
        for (int k = 0; k < 9; k++)
            if (k != 0)
                deltaY[i][k] = 0;
            else
                deltaY[i][0] = Y[i + 1] - Y[i];
    for (int i = 1; i < 9; i++)
        for (int k = 0; k < 9 - i; k++)
            deltaY[k][i] = deltaY[k + 1][i - 1] - deltaY[k][i - 1];

    for (int i = 0; i < 10; i++)
    {

        double d = 1;
        for (int j = 0; j < 10; j++)
        {
            if (i == j) {
                printf("%+2.3lf ", x0 - X[j]);
                d = d * (x0 - X[j]);
            }
            else {
                printf("%+2.3lf ", X[i] - X[j]);
                d = d * (X[i] - X[j]);
            }
        }
        cout << d;
        printf("\t%5.2lf \n", Y[i]);
    }
    double QRes = 1, factorial = 1, res = Y[0];
    Q = (x0 - X[0]) / H;
    for (int i = 0; i < 9; i++) {
        factorial *= i + 1;
        QRes *= (Q - i);
        res += (QRes * deltaY[0][i]) / factorial;
    }
    printf("\ny = %f\n", res);
                                                }
double newtone(const double x, const double h, const double* y, const int n, const double z)
       //метод Ньютона для рівновіддалених значень
{
       double** dy = (double**)malloc(n * sizeof(double*));
       for (int i = 0; i < n; ++i)
              dy[i] = (double*)malloc(n * sizeof(double));

       for (int i = 0; i < n; ++i)
              dy[0][i] = y[i];
       for (int i = 1; i < n; ++i)
              for (int j = 0; j < n; ++j)
                     if (j > (n - i - 1))
                            dy[i][j] = 0;
                     else
```

```
                    dy[i][j] = dy[i - 1][j + 1] - dy[i - 1][j];
printf("  x        y");
for (int i = 1; i < n; ++i)
       printf("       D%dy", i);
puts("\n");
for (int i = 0; i < n; ++i)
{
       printf("%5.3lf", x + i * h);
       for (int j = 0; j < n; ++j)
              if (dy[j][i] == 0)
                     break;
              else
                     printf("%10.6lf", dy[j][i]);
       puts("");
}
double ser = 0;
for (int i = 0; i < n; ++i)
       ser += (x + i * h);
ser /= n;
long long f = 1;
double q, k, res = 0;
if (z < ser)
{
       q = (z - x) / h;
       for (int i = 0; i < n; ++i)
       {


              if (i != 0)
                     f *= i;
              if (i == 0)
                     k = 1;
              else if (i == 1)
                     k = q;
              else
                     k *= q - i + 1;
              res += (dy[i][0] * k) / f;
       }
}
else
{
       for (int i = 0; i < n; ++i)
       {
              q = (z - (x + (n - 1) * h)) / h;
              if (i != 0)
                     f *= i;
              if (i == 0)
                     k = 1;
              else if (i == 1)
                     k = q;
              else
                     k *= q + i - 1;
              res += (dy[i][n - i - 1] * k) / f;
       }
}
for (int i = 0; i < n; ++i)
       free(dy[i]);
free(dy);
return res;
                                        }
```

Вигляд виконаної програми

```
Lagrange method:
                                             P(x0)                                         D[i]        Y[i]
 1 +0.001 -0.005 -0.010 -0.015 -0.020 -0.025 -0.030 -0.035 -0.040 -0.045 -7.79625e-19     0.88
 2 +0.005 -0.004 -0.005 -0.010 -0.015 -0.020 -0.025 -0.030 -0.035 -0.040 -3.07125e-19     0.89
 3 +0.010 +0.005 -0.009 -0.005 -0.010 -0.015 -0.020 -0.025 -0.030 -0.035 1.75219e-19      0.89
 4 +0.015 +0.010 +0.005 -0.014 -0.005 -0.010 -0.015 -0.020 -0.025 -0.030 -1.17281e-19     0.89
 5 +0.020 +0.015 +0.010 +0.005 -0.019 -0.005 -0.010 -0.015 -0.020 -0.025 1.06313e-19      0.89
 6 +0.025 +0.020 +0.015 +0.010 +0.005 -0.024 -0.005 -0.010 -0.015 -0.020 -1.34438e-19     0.89
 7 +0.030 +0.025 +0.020 +0.015 +0.010 +0.005 -0.029 -0.005 -0.010 -0.015 2.43844e-19      0.89
 8 +0.035 +0.030 +0.025 +0.020 +0.015 +0.010 +0.005 -0.034 -0.005 -0.010 -6.67406e-19     0.90
 9 +0.040 +0.035 +0.030 +0.025 +0.020 +0.015 +0.010 +0.005 -0.039 -0.005 3.06338e-18      0.90
10 +0.045 +0.040 +0.035 +0.030 +0.025 +0.020 +0.015 +0.010 +0.005 -0.044 -3.11141e-17     0.90

y = 0.884104
Newton`s method
  x        y        D1y       D2y       D3y       D4y       D5y       D6y       D7y       D8y       D9y

1.415  0.880000  0.009000 -0.008000  0.008000 -0.008000  0.008000 -0.008000  0.008000 -0.008000  0.008000
1.420  0.889000  0.001000         0         0         0         0         0         0         0
1.425  0.890000  0.001000         0         0         0         0         0         0
1.430  0.891000  0.001000         0         0         0         0         0
1.435  0.892000  0.001000         0         0         0         0
1.440  0.893000  0.001000         0         0         0
1.445  0.894000  0.001000         0         0
1.450  0.895000  0.001000         0
1.455  0.896000  0.001000
1.460  0.897000

       y = 0.884104
```

Висновки:

На даній лабораторній роботі ознайомився з методом інтерполяції таблично заданих функцій.. Розв’язав завдання, згідно до індивідуального варіанту:

x	1,415	1,420	1,425	1,430	1,435	1,440	1,445	1,450	1,455	1,460
y	0,88	0,889	0,890	0,891	0,892	0,893	0,894	0,895	0,896	0,897

$x_0 = 1{,}4161$

Також за допомогою знайдених поліномів , знайшов значення в точці x_0, отримав y_0=0.884104, в обох методах