УДК 004.415.3
Пех П.А. к.т.н. доц.,Мельник В.М. к.ф-м.н. доц., Островський М.В.
Луцький національний технічний університет

C++BUILDER ПРОЕКТ ІНТЕРПОЛЯЦІЇ ФУНКЦІЙ МНОГОЧЛЕНАМИ НЬЮТОНА ТА ЛАГРАНЖА

Пех П.А., Мельник В.М., Островський М.В. C++Builder проект інтерполяції функцій многочленами Ньютона та Лагранжа. В статті запропоновано C++Builder проект для побудови інтерполяційних многочленів Ньютона та Лагранжа.
Ключові слова: C++Builder проект, інтерполяція функцій, многочлен Ньютона, многочлен Лагранжа.

Пех П.А., Мельник В.М., Островский М.В. C++Builder проект интерполяции функций многочленами Ньютона и Лагранжа. В статье предложен C++Builder проект для построения интерполяционных многочленов Ньютона и Лагранжа..
Ключевые слова: C++Builder проект, интерполяция функций, многочлен Ньютона, многочлен Лагранжа.

Pekh P., Melnyk V., Ostrovsky M. C++Builder project for interpolation functions by polynomial Newton and Lagrange. In this article is designed the C++Builder project to build an interpolation functions by polynomial Newton and Lagrange.
Keywords: C + + Builder project, interpolation functions polynomial Newton, polynomial Lagrange.

Постановка задачі. Нехай невідому функцію $y=f(x)$ задано своїми числовими значеннями $y_0=f(x_0)$, $y_1=f(x_1)$, $\ldots, y_n=f(x_n)$ у точках $x_0, x_1, x_2, \Lambda, x_n$, .які називаються вузлами інтерполяції. Задача інтерполяції функції многочленом $P_n(x)$ степеня n полягає у знаходженні многочлена такого вигляду [4]:

$$P_n(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+a_3(x-x_0)(x-x_1)(x-x_2)+\Lambda \qquad (1)$$
$$+\Lambda+a_n(x-x_0)(x-x_1)\Lambda(x-x_{n-1}),$$

де $x_0, x_1, \Lambda, x_{n-1}$ — задані значення аргументу x, а a_0, a_1, Λ, a_n - невідомі коефіцієнти, причому многочлен $P_n(x)$ у вузлах інтерполяції має приймати такі ж значення, що й задана функція $y=f(x)$. Визначити коефіцієнти a_0, a_1, Λ, a_n і означає розв'язати задачу інтерполяції, оскільки знайдений за таких умов многочлен дозволить обчислити значення заданої функції у довільній точці.

Метою даного дослідження є розроблення засобами середовища C++Builder [1,2,3] проекту, який дозволяє автоматизувати процес побудови інтерполяційних многочленів за формулами Ньютона та Лагранжа.

Основна частина. З математичної точки зору задача інтерполяції функції многочленами може бути розв'язана багатьма методами. У даній статті аналізується, як ця задача розв'язується за допомогою першої та другої інтерполяційних формул Ньютона, у яких вузли інтерполяції є рівновіддаленими, а також за допомогою інтерполяційної формули Лагранжа для випадку нерівновіддалених вузлів інтерполяції.

Звичайно, побудова інтерполяційних формул значно спрощується, коли вузли інтерполяції $x_0, x_1, \ldots, x_n$ - рівновіддалені, тобто крок інтерполяції є постійним:

$$h=x_{i+1}-x_i=const \quad (i=0,1,\ldots,n-1). \qquad (2)$$

В інтерполяційних формулах з рівновіддаленими вузлами інтерполяції використовується поняття скінченних різниць. Назвемо скінченною різницею першого порядку різницю між значеннями функції у сусідніх вузлах інтерполяції:

$$y_1-y_0=\Delta y_0; \quad y_2-y_1=\Delta y_1; \quad \ldots y_n-y_{n-1}=\Delta y_{n-1}. \qquad (3)$$

Зі скінченних різниць першого порядку можна одержати скінченні різниці другого порядку:

$$\Delta^2 y_0=\Delta y_1-\Delta y_0; \quad \Delta^2 y_1=\Delta y_2-\Delta y_1; \quad \ldots\Delta^2 y_{n-2}=\Delta y_{n-1}-\Delta y_{n-2} \qquad (4)$$

Скінченні різниці n-го порядку визначаються формулами:

$$\Delta^n y_0 = \Delta^{n-1} y_1 - \Delta^{n-1} y_0;\ \Delta^n y_1 = \Delta^{n-1} y_2 - \Delta^{n-1} y_1;\ \ldots \Delta^n y_i = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i, \ldots \quad (5)$$

Різниці довільних порядків можуть бути виражені безпосередньо через значення функції:

$$\Delta y_i = y_{i+1} - y_i;$$

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i = (y_{i+2} - y_{i+1}) - (y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i;$$

$$\Delta^3 y_i = \Delta^2 y_{i+1} - \Delta^2 y_i = (y_{i+3} - 2y_{i+2} + y_{i+1}) - (y_{i+1} - 2y_{i+1} + y_i) =$$

$$= y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i. \quad (6)$$

Неважко довести, що для будь-якого цілого значення m мають місце формули:

$$\Delta^m y_i = y_{i+m} - my_{i+m-1} + \frac{m(m-1)}{2!} y_{i+m-2} -$$

$$- \frac{m(m-1)(m-2)}{3!} y_{i+m-3} + \ldots + (-1)^{m-1} my_{i+1} + (-1)^m y_i. \quad (7)$$

Скінченні різниці різних порядків зручно розташовувати у вигляді: горизонтальної (табл.1) або діагональної таблиць (табл. 2).

Таблиця 1 – Горизонтальна таблиця скінченних різниць функції

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_2$	
x_2	y_2	Δy_2		
x_3	y_3			

Таблиця 2 – Діагональна таблиця скінченних різниць функції

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
x_0	y_0			
		Δy_0		
x_1	y_1		$\Delta^2 y_0$	
		Δy_1		$\Delta^3 y_0$
x_2	y_2		$\Delta^2 y_2$	
		Δy_2		
x_3	y_3			

Перша інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції має такий вигляд:

$$P_n(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!\,h}(x - x_0)(x - x_1) + \quad (8)$$

$$+ \frac{\Delta^3 y_0}{3!\,h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots + \frac{\Delta^n y_0}{n!\,h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1}).$$

Її можна записати у вигляді більш зручному для практичного використання. Позначимо:

$$\frac{x - x_0}{h} = q. \quad (9)$$

Тоді

$$\frac{(x - x_1)}{h} = \frac{x - (x_0 + h)}{h} = q - 1; \quad (10)$$

де $\dfrac{x-x_2}{h}=q-2;\quad \ldots,\dfrac{x-x_{n-1}}{h}=q-n+1,$ (11)

і формула (8) набуває вигляду:

$$P_n(x)=y_0+q\Delta y_0+\frac{q(q-1)}{2!}\Delta^2 y_0+\frac{q(q-1)(q-2)}{3!}\Delta^3 y_0+\ldots+ \qquad (12)$$
$$+\frac{q(q-1)(q-2)\ldots(q-n+1)}{n!}\Delta^n y_0.$$

Формулу (12) доцільно використовувати для інтерполювання (екстраполювання) функції $y=f(x)$ в околі початкового значення x_0, де значення q є малим за абсолютною величиною.

Друга інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції має вигляд:

$$P_n(x)=y_n+\frac{\Delta y_{n-1}}{h}(x-x_n)+\frac{\Delta^2 y_{n-2}}{2!\,h^2}(x-x_n)(x-x_{n-1})+ \qquad (13)$$
$$+\frac{\Delta^3 y_{n-3}}{3!\,h^3}(x-x_n)(x-x_{n-1})(x-x_{n-2})+\ldots\frac{\Delta^n y_0}{n!\,h^n}(x-x_n)(x-x_{n-1})\ldots(x-x_1).$$

Запишемо її у вигляді, зручнішому для практичного використання. Позначивши $\dfrac{x-x_n}{h}=q$, дістанемо:

$$\frac{x-x_{n-1}}{h}=\frac{x-(x_n-h)}{h}=q+1;$$
$$\frac{x-x_{n-2}}{h}=\frac{x-(x_n-2h)}{h}=q+2; \qquad (14)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\frac{x-x_1}{h}=\frac{x-[x_n-(n-1)h]}{h}=q+n-1.$$

Після підстановок цих значень у формулу (13) вона набуває вигляду

$$P_n(x)=y_n+q\Delta y_{n-1}+\frac{q(q+1)}{2!}\Delta^2 y_{n-2}+ \qquad (15)$$
$$+\frac{q(q+1)(q+2)}{3!}\Delta^3 y_{n-3}+\ldots+\frac{q(q+1)\ldots(q+n-1)}{n!}\Delta^n y_0.$$

Цю формулу рекомендується використовувати для інтерполювання (екстраполювання) функції $y=f(x)$ для тих значень аргумента, які знаходяться у кінці таблиці.

Нарешті, нехай невідому функцію $y=f(x)$ задано своїми числовими значеннями $y_0=f(x_0),\ y_1=f(x_1),\ \ldots,y_n=f(x_n)$ у нерівновіддалених вузлах інтерполяції x_0,x_1,x_2,Λ,x_n. Тоді задача інтерполяції функції може бути розв'язана за допомогою інтерполяційного полінома Лагранжа:

$$L_n(x)=\sum_{k=0}^{n}\frac{(x-x_0)(x-x_1)(x-x_2)\Lambda(x-x_{k-1})(x-x_{k+1})\Lambda(x-x_{n-1})(x-x_n)}{(x_k-x_0)(x_k-x_1)(x_k-x_2)\Lambda(x_k-x_{k-1})(x_k-x_{k+1})\Lambda(x_k-x_{n-1})(x_k-x_n)}\cdot y_k \qquad (16)$$

Зупинимося на оцінках похибок інтерполяційних формул Лагранжа і Ньютона. Адже недостатньо лише обчислити значення функції у довільній точці, потрібно також вміти оцінити похибку знайденого значення.

Про похибку, що виникає у разі заміни функції $y=f(x)$ її інтерполяційним многочленом $P_n(x)$, можна судити за величиною залишкового члена:

$$R_n(x)=f(x)-P_n(x). \qquad 7)$$

Якщо для функції $y=f(x)$ відомий аналітичний вираз і можна знайти її похідні до $(n+1)-$го порядку включно у деякій області $a\le x\le b$, що містить вузли інтерполяції $x_0, x_1, x_2, \Lambda, x_n$, то величину залишкового члена $R_n(x)$ для інтерполяційної формули Лагранжа (3.16) визначають у такий спосіб:

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\ldots(x-x_n), \quad (18)$$

де змінна ξ лежить усередині відрізка $[a,b]$. Позначивши через

$$M_{n+1} = \max_{a \le x \le b}\left|f^{(n+1)}(x)\right|, \quad)$$

отримаємо оцінку для абсолютної похибки інтерполяційної формули Лагранжа:

$$\left|R_n(x)\right| \le \frac{M_{n+1}}{(n+1)!}\left|(x-x_0)(x-x_1)\ldots(x-x_n)\right|. \quad (20)$$

Якщо вузли інтерполяції $x_0, x_1, x_2, \Lambda, x_n$ рівновіддалені, то, вважаючи $q = \frac{x - x_0}{h}$, отримаємо залишковий член першої інтерполяційної формули Ньютона:

$$R_n(x) = h^{n+1}\frac{q(q-1)(q-2)\ldots(q-n)}{(n+1)!}f^{(n+1)}(\xi), \quad (21)$$

де ξ — деяке проміжне значення між вузлами інтерполяції. Аналогічно, вважаючи $q = \frac{x - x_n}{h}$, отримаємо залишковий член другої інтерполяційної формули Ньютона:

$$R_n(x) = h^{n+1}\frac{q(q+1)(q+2)\ldots(q+n)}{(n+1)!}f^{(n+1)}(\xi). \quad 22)$$

У практичних розрахунках аналітичний вигляд функції $y = f(x)$ не завжди відомий. Тоді, припускаючи, що в таблиці скінченних різниць для функції $y = f(x)$ різниці $(n+1)-$-го порядку $\Delta^{n+1}y$ у майже постійні і h досить мале, а такожураховуючи, що

$$f^{(n+1)}(x) = \lim_{h \to 0}\frac{\Delta^{n+1}y}{h^{n+1}}, \quad (23)$$

наближено можна вважати:

$$f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1}y_0}{h^{n+1}}. \quad (24)$$

У цьому разі залишковий член першої інтерполяційної формули Ньютона дорівнює:

$$R_n(x) \approx \frac{q(q-1)\ldots(q-n)}{(n+1)!}\Delta^{n+1}y_0, \quad (25)$$

а для другої інтерполяційної формули Ньютона залишковий член дорівнює:

$$R_n(x) \approx \frac{q(q+1)\ldots(q+n)}{(n+1)!}\Delta^{n+1}y_n. \quad (26)$$

Для реалізації мети дослідження розроблено C++Builder проект, який складається з головної форми Form1, програмний код якої знаходиться у файлі MainFormUnit1, та семи підлеглих форм, призначення яких показано у таблиці 3. На головній формі (рис. 1) розміщене головне меню проекту, яке забезпечує зв’язок з усіма іншими формами. Головне меню включає в себе шість команд, за допомогою яких можна перейти на всі підлеглі форми, кожна з яких вирішує ту чи іншу підзадачу проекту. Наприклад, вибравши першу команду головного меню, ми перейдемо на форму Form2, на якій розв'язується задача інтерполяції функції за допомогою першої інтерполяційної формули Ньютона.

Для створення головного меню проекту використано компонент MainMenu, кожна команда якого (пункт меню) забезпечує перехід на відповідну підлеглу форму, а з кожної підлеглої форми можна повернутись лише на головну.

Зв’язування головної та всіх підлеглих форм досягається тим, що у файлі головної форми MainFormUnit1.cpp включені директиви з іменами файлів усіх підлеглих форм:

```
#include "MainFormUnit1.h"
#include "TabNewtOneUnit2.h"
#include "TabNewtTwoUnit3.h"
#include "TabLagrUnit4.h"
#include "GrafNewt1Unit5.h"
#include "GrafNewt2Unit6.h"
#include "GrafLagrUnit7.h"
#include "ProectContUnit8.h",
```

а у заголовки підлеглих форм включаємо директиву звернення до головної форми. Наприклад, для форми Form2 це виглядає так:

```
#include "TabNewtOneUnit2.h"
#include "MainFormUnit1.h"
```

Звичайно, перехід з головної форми на одну з підлеглих забезпечується програмуванням відповідних елементів управління. Так, перша команда головного меню, яка забезпечує перехід на форму Form2, запрограмована так:

```
void __fastcall TForm1::N11Click(TObject *Sender)
{Form2->Show();}
```

Відповідно до цього коду, клацнувши по першій команді головного меню, ми перейдемо до форми Form2. Команда До головного меню форми Form2, про яку йтиметься далі, має такий програмний код:

```
void __fastcall TForm2::N3Click(TObject *Sender)
{Form2->Close();
 Form1->Show();}
```

Виконавши команду **До головного меню** на формі Form2, ми закриємо форму Form2 і перейдемо до головної форми.

Таблиця 3. Найменування та призначення форм і відповідних їм файлів.

з/п	Форма	Найменування файла	Призначення форми
	2	3	4
	Form1	MainFormUnit1	Головна форма проекту
	Form2	TabNewtOneUnit2	Інтерполяція функції за першою інтерполяційною формулою Ньютона
	Form3	TabNewtTwoUnit3	Інтерполяція функції за другою інтерполяційною формулою Н[illegible]
	Form4	TabLagrUnit4	Інтерполяція функції за інтерполяційною формулою Лагранжа
	Form5	GrafNewt1Unit5	Графік функції за першою інтерполяційною формулою Ньютона
	Form6	GrafNewt2Unit6	Графік функції за другою інтерполяційною формулою Ньютона
	Form7	GrafLagrUnit7	Графік функції за інтерполяційною формулою Лагранжа
	Form8	ProectContUnit8	Зміст задач проекту

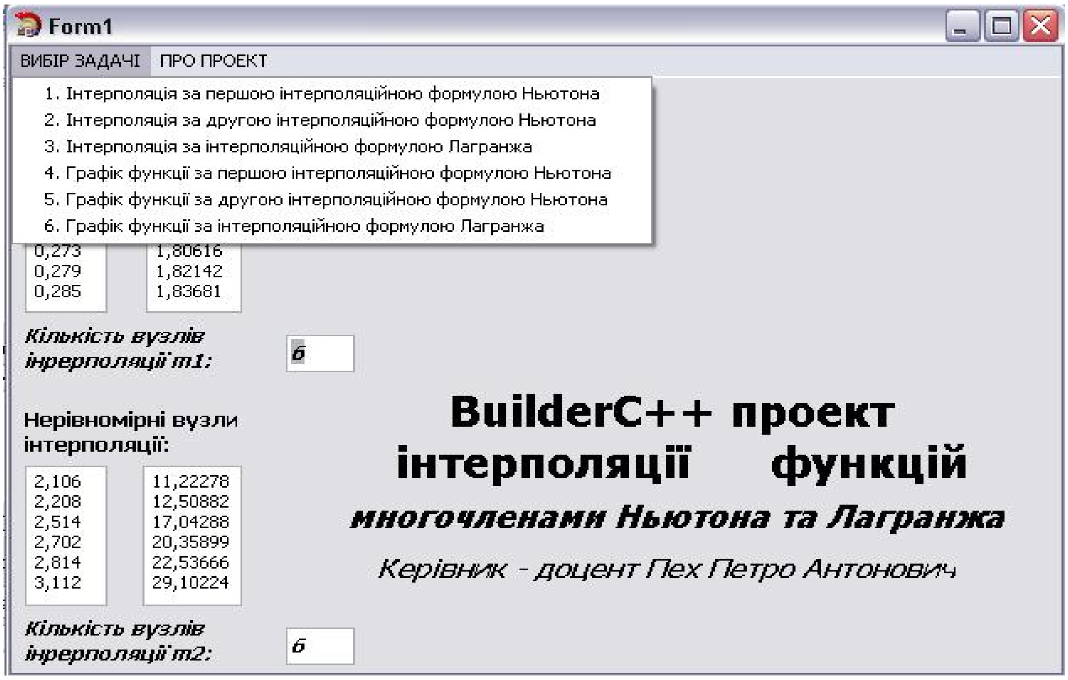

Рис. 1 – Вікно Form1 - головної форми проекту

На рис.2 наведено форму Form2, з результатами обчислень кінцевих різниць та коефіцієнтів інтерполяційного многочлена, а на рис.3 наведено ту ж форму Form2, з виведеним на неї інтерполяційним многочленом, знайденим за першою інтерполяційною формулою Ньютона.

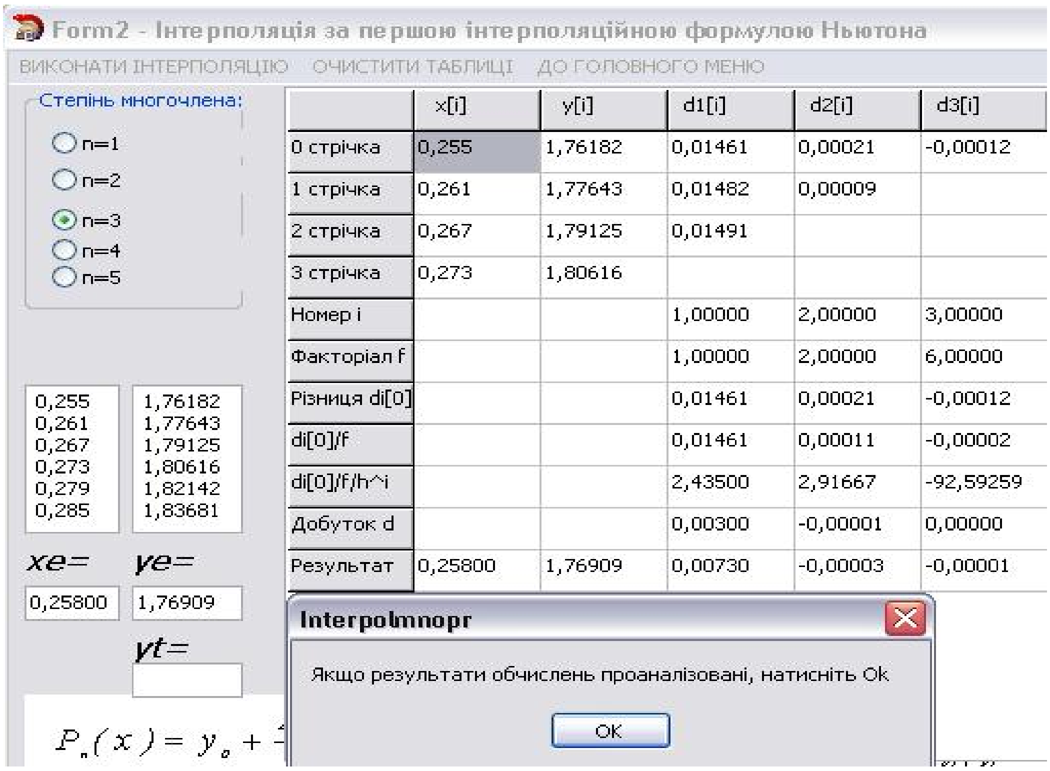

Рис. 2 – Вигляд вікна Form2 з таблицею проміжних та кінцевих результатів

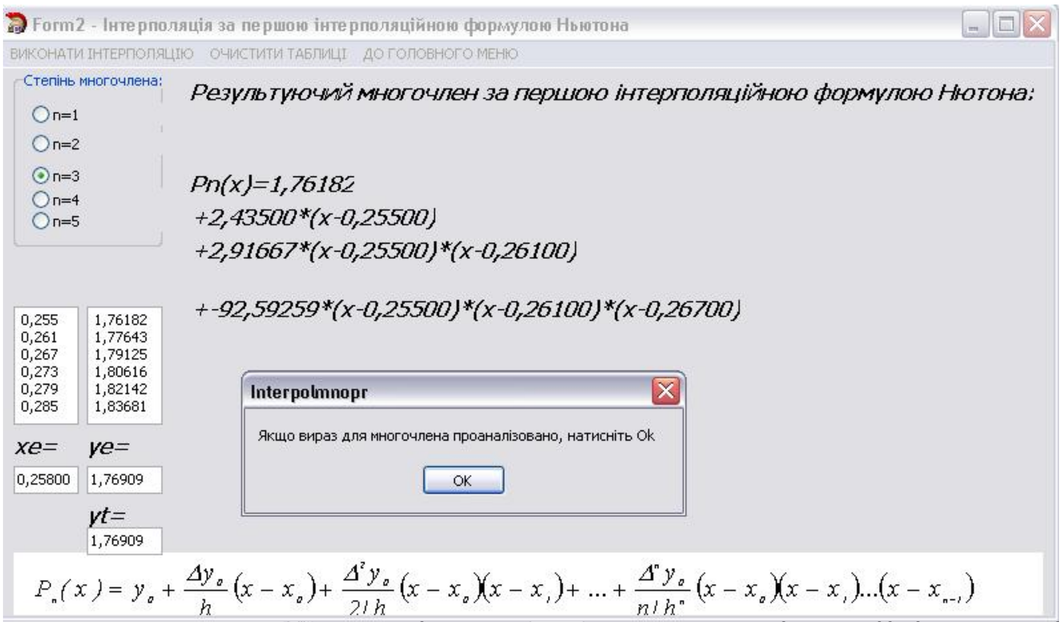

Рис. 3 – Вигляд вікна Form2 з першою інтерполяційною формулою Ньютона

Такі ж результати отримані стосовно другої інтерполяційної формули Ньютона та інтерполяційного многочлена Лагранжа. Передбачена також можливість побудови графіків за кожним зі знайдених многочленів.

Висновки

У роботі розроблено C++Builder проект, за допомогою якого засобами сучасних інформаційних технологій розв'язується актуальна задача автоматизації процесу побудови інтерполяційних многочленів за формулами Ньютона та Лагранжа.

1. Архангельский А.Я. Программирование в C++Builder. –М.:ООО «Бином-Пресс», 2010.-1298 с.
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на С. – М.:Издательство Бином, 2006.-912 с.
3. Шпак З.Я. Програмування мовою С. – Львыв: Орыяна-Нова, 2006. -432 с.
4. Хвищун І.О. Програмування та математичне моделювання. –К.:Видавничий Дім «Ін Юре», 2007. -544 с.