ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД

«ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»

МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ

КУРСОВА РОБОТА

|  |  |
| --- | --- |
| з | дисципліни «Численні методи» |
|  | (назва дисципліни) |
| на тему: | «Інтерполяція функцій» |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студента | | 2 | курсу, групи | | 4315 |
| напряму підготовки | | | | 6.040301 «Прикладна математика» | |
|  | | | | (шифр і назва напряму підготовки) | |
| А.С.Солодухін | | | | | |
| (ініціали та прізвище) | | | | | |
| Керівник | Полюга С.І. | | | | |
|  | (посада, вчене звання, науковий ступінь, прізвище та ініціали) | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Національна шкала: | |  | | |
| Кількість балів: |  | | Оцінка ECTS: |  |
|  |  | |  |  |
| Члени комісії: |  |  | |  | |
|  | (підпис) |  | | (ініціали та прізвище) | |
|  |  |  | |  | |
|  |  |  | |  | |
|  | (підпис) |  | | (ініціали та прізвище) | |
|  |  |  | |  | |
|  |  |  | |  | |
|  | (підпис) |  | | (ініціали та прізвище) | |

Запоріжжя – 2016

# РЕФЕРАТ

Курсова робота: 24 с., 4 джерела.

Об’єкт дослідження – методи інтерполяції функцій.

Метод дослідження – аналітичний.

Метою роботи є вивчення основних методів інтерполяції функції, та реалізація програмного продукту для інтерполяції многочленом Лагранжа.

У даній курсовій роботі розглядаються основні методи інтерполяції функцій (многочлен Лагранжа, многочлен Ньютона), а також програмно реалізовується метод многочлена Лагранжа для функцій однієї змінної. Програмний продукт реалізован на базі JavaScript-плагина Highcharts.js

**ЗМІСТ**

[РЕФЕРАТ 4](#_Toc449617574)

[ВСТУП 6](#_Toc449617575)

[1 ІНТЕРПОЛЯЦІЯ ФУНКЦІЙ 7](#_Toc449617576)

[1.1 Постановка задачі інтерполяції 7](#_Toc449617577)

[1.2 Локальна та глобальна інтерполяції 7](#_Toc449617578)

[2 МЕТОДИ ІНТЕРПОЛЯЦІЇ](#_Toc449617579) 8

[2.1 Інтерполяційний многочлен Лагранжа 8](#_Toc449617580)

[2.2 Інтерполяційний многочлен Ньютона 9](#_Toc449617581)

[2.2.1 Перша інтерполяційна формула Ньютона 9](#_Toc449617581)

[2.3 Порівняння інтерполяційних многочленів](#_Toc449617580) 10

[3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ 10](#_Toc449617576)

[3.1 Опис програми та її методів 10](#_Toc449617577)

[3.2 Приклад роботи програми 10](#_Toc449617578)

[ВИСНОВКИ 13](#_Toc449617582)

[ПЕРЕЛІК ПОСИЛАНЬ 14](#_Toc449617583)

# ВСТУП

У обчислювальній математиці вагому роль має інтерполяція функцій, тобто побудова по заданій функції, або по точкам (вузлам) іншої функції. Для реалізації цього процесу на практиці, вважається ефективним звести задану функцію до поліномів, або до дробно-раціональних функцій. Теорія інтерполювання використовується при побудові та дослідженні квадратурних формул для численного інтегрування, для отримання методів вирішення диференціальних та інтегральних рівнянь.

У даній курсовій роботі будуть описані методи поліномінальної інтерполяції, або інтерполяції многочленами.

Також у даній роботі буде представлено прототип програми для інтерполювання функції однієї змінної, виконаний на мові програмування JavaScript за допомогою бібліотек jQuery та Hightcharts.js.

# ІНТЕРПОЛЯЦІЯ

## Постановка задачі інтерполяції

Інтерполяція в [обчислювальній математиці](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) спосіб знаходження проміжних значень величини за наявним [дискретним](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C) набором відомих значень. Тобто ми оперуємо тільки із значеннями отриманими в ході деякого експерименту чи шляхом виборки. Як правило, на підставі цих наборів потрібно побудувати [функцію](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), зі значеннями якої могли б з високою точністю збігатися інші отримувані значення. Така задача називається [апроксимацією](https://uk.wikipedia.org/wiki/%D0%90%D0%BF%D1%80%D0%BE%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D1%86%D1%96%D1%8F) кривої. Інтерполяцією називають такий різновид апроксимації, при якій крива побудованої функції проходить точно через наявні точки даних.

Нехай маємо n значень *xi*, кожному з яких відповідає своє значення *yi*. Потрібно знайти таку функцію *F*, що:

*F(xi) = yi , I = 1, … , n* (1.1)

При цьому:

* *xi* називають вузлами інтерполяції
* пари (*xi*, *yi*) називають точками даних чи базовими точками
* різницю між «сусідніми» значеннями *xi* і- *xi-1* — кроком
* функцію *F (x)* — функцією, що інтерполює чи інтерполянтом.

Тобто, інтерполяція – це процес переходу табличних даних у вигляд функції.

Інтерполяційну функцію знаходять серед многочленів *Pn**x*, степінь  
яких не перевищує числа *n* за умови, що в таблиці заданий *n*1 вузол.  
Многочлен *Pn**x*, що набирає у вузлових точках значення  
*Pn* *xi**yi* , *i* 0,…,*n ,* що задано таблицею, називають інтерполяційним  
многочленом для функції *y**f**x*.Доведено, що існує тільки один многочлен *Pn**x*степені не вище *n* , що є інтерполяційним многочленом для *y**f**x*.Заміна функції її інтерполяційним многочленом *f* *x*≈*Pn**x*при *x* ∈*a*, *b*називається інтерполяцією функції. Звичайно, при цьому виникаєзапитання про оцінку похибок такої заміни. Геометричний зміст інтерполяції полягає в подаванні функції *y* *f**x*у вигляді параболи *y* *Pn**x*степені *n* , яка проходить через задані таблицею точки *x*0; *y*0 , *x*1; *y*1, …, *xn*; . *yn* Формулу одержання значення функції в точці *x належить* *x*0, *xn* , що знаходиться між вузлами інтерполяції, вважають інтерполяційною, якщо ж *x не належить* *x*0 , *xn* , тобто знаходиться поза відрізком, то формулу називають *екстраполяційною*.

## Локальна та глобальна інтерполяції

Якщо деяка функція *f(x)* інтерполюється на відрізку [a,b] за допомогою єдиного многочлена Pm(x), то таку інтерполяцію називають *глобальною.*У випадку *локальної інтерполяції* на кожному інтервалі [xi , xi+1] будується окремий інтерполяційний многочлен невисокого степені.

# 2 СПОСОБИ ІНТЕРПОЛЯЦІЇ

## 2.1 Інтерполяційний многочлен Лагранжа

Нехай на відрізку [*a; b*] дано (*n+1*) різних значень аргумента

Інтерполяційна формула Лагранжа (Інтерполяційна формула Лагранжа) для яких відомі відповідні значення функції Інтерполяційна формула Лагранжа. Необхідно побудувати поліном, степінь якого не перевищує *n*, і який у *вузлах інтерполяції* Інтерполяційна формула Лагранжа  приймає ті ж значення, що і функція  Інтерполяційна формула Лагранжа, тобто Інтерполяційний поліном Лагранжа . *Інтерполяційна формула**Лагранжа* дозволяє представити поліном Інтерполяційна формула Лагранжа у вигляді лінійної комбінації функції Інтерполяційна формула Лагранжа у *вузлах інтерполяції*:

Інтерполяційна формула Лагранжа (2.1)

де Інтерполяційна формула Лагранжа — поліном степені *n*, для якого виконується умова:

Інтерполяційна формула Лагранжа (2.2)

Врахувавши (2.1) поліном Інтерполяційна формула Лагранжа можна записати у наступному вигляді:

Інтерполяційна формула Лагранжа (2.3)

Де Інтерполяційна формула Лагранжа постійний коефіцієнт. Значення даного коефіцієнта можна знайти при Інтерполяційна формула Лагранжа.

Інтерполяційна формула Лагранжа (2.4)

З останнього співвідношення визначаєсо Інтерполяційна формула Лагранжа і підставляємо його у формулу (2.3):

Інтерполяційна формула Лагранжа

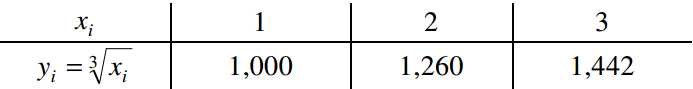
Інтерполяційна формула Лагранжа (2.5)

Тоді *інтерполяційний многочлен Лагранжа* буде мати наступний вигляд:

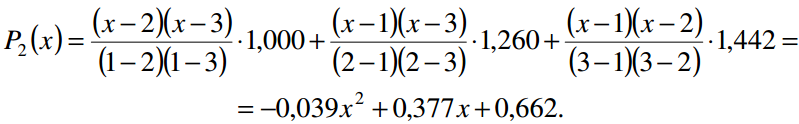
Інтерполяційна формула Лагранжа(2.6)

**Приклад 1.1**

Побудувати інтерполяційний многочлен Лагранжа для функції *f* *x*3 *x* з вузлами інтерполяції *x*0 , *x*1 , *x*2 .

**

Тут *n* 2, тоді інтерполяційний многочлен запишеться у вигляді:

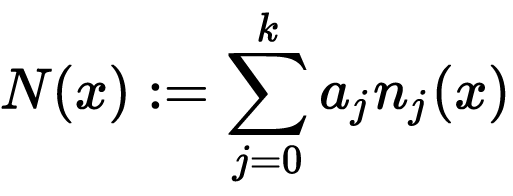
**

Як видно з ходу вирішення , кожен раз коли вузол *xk* змінюється, всі базові многочлени Лагранжа необхідно перерахувати. Найкращим варіантом інтерполяційного многочлена для практичних (або обчислювальних) цілей є барицентрична форма інтерполяції Лагранжа або [поліном Ньютона](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BD%D0%BE%D0%BC_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0).

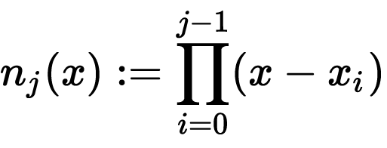
2.2 [Інтерполяційний многочлен Ньютона](#_Toc449617581)

## Маючи множину з k + 1 точок

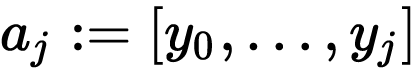
## де немає двох однакових xi, інтерполяційний многочлен у формі Ньютона — це [лінійна комбінація](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B0_%D0%BA%D0%BE%D0%BC%D0%B1%D1%96%D0%BD%D0%B0%D1%86%D1%96%D1%8F) базових многочленів Ньютона

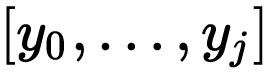
 (2.7)

Де базовий многочлен Ньютона задається так

 (2.8)

Для j > 0

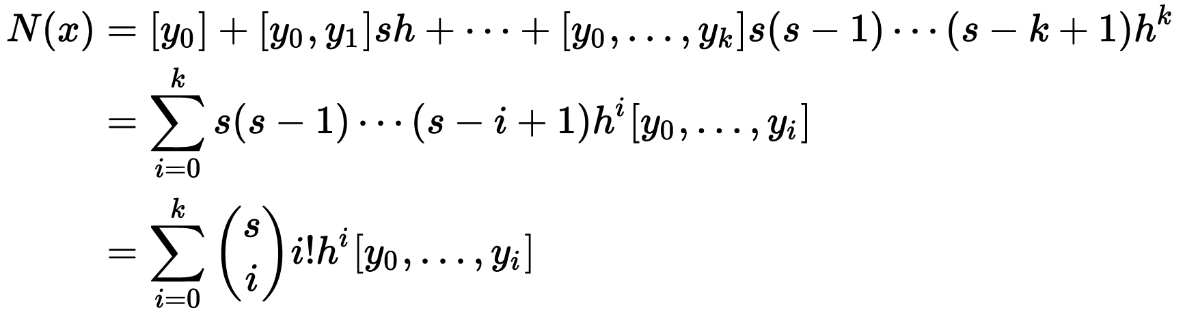
Коефіцієнти задаються 

де це позначення [розділеної різниці](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B0_%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D1%86%D1%8F). Отже інтерполяційний многочлен Ньютона можна записати як

 (2.9)

Інтерполяційний многочлен Ньютона можна представити у спрощеній формі якщо  *x0,x1,…,xk* впорядковані послідовно і з рівними проміжками. Позначаючи  *h =xi+1*  для кожного   *i = 0,1,…,k* та

*x = x0+sh* , різницю *x-xi* можна записати як *(s - i)h*. Так інтерполяційний многочлен Ньютона набуває форми:

(2.10)

така форма називається прямий інтерполяційний многочлен Ньютона.

**Приклад 2.1*.***

Побудувати інтерполяційний многочлен Ньютона для функції *f* *x*з вузлами інтерполяції *x*0 , 2 *x*1 , *x*2 .

*Розв'язування*. Вузлів інтерполяції три, так що будемо будувати многочлен Ньютона другого порядку. Для цього обчислимо відповідні кінцеві різниці, результати наведемо в таблиці

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | *xi* | *yi* | *Δyi* | *Δ2yi* |
| 0 | 1 | 1.000 | 0.260 | -0.078 |
| 1 | 2 | 1.260 | 0.182 |  |
| 2 | 3 | 1.442 |  |  |

Підставимо отримані значення в інтерполяційну формулу Ньютона, враховуючи, що *h* 1, одержимо



Отриманий многочлен тотожно збігається з інтерполяційним многочленом Лагранжа. Додатковий вузол не заданий, тому похибка визначається за тією ж формулою, що і для многочлена Лагранжа.

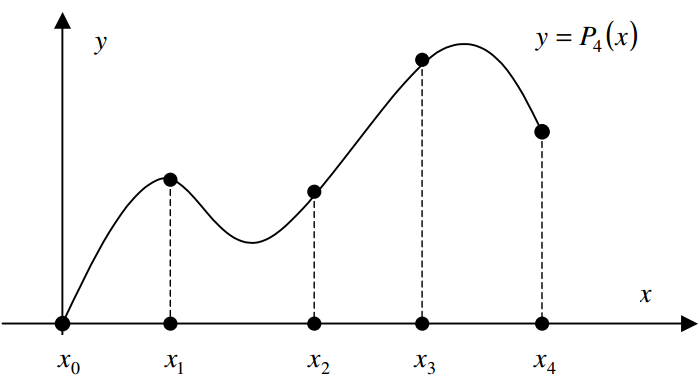
2.3 Порівняння інтерполяційних многочленів

Приведемо порівняльну характеристику розглянутих інтерполяційних многочленів Лагранжа, першої, другої формул Ньютона:

1. Інтерполяційні многочлени Лагранжа та Ньютона мають степінь, не вище *n* за умови, що визначення многочленів виконується за таблицею, у якій заданий *n* 1 вузол. При отриманні многочлена Лагранжа вузли можуть бути розташовані довільно, при отриманні многочленів Ньютона вузли повинні бути рівновіддаленими. 2. Многочлен Лагранжа складається із суми рівноправних доданків, кожний з яких є многочленом *n* -ої степені і повинен брати участь в усіх обчисленнях. У многочленів Ньютона степені їхніх доданків постійно підвищуються, починаючи від нульової в першому доданку до *n* -ої в останньому. Крім того, у формулах Ньютона знаменники коефіцієнтів містять величину *k*! Ці числа зі збільшенням *k* швидко зростають, отже, коефіцієнти зменшуються і при обчисленнях, починаючи з деякого номера, останніми доданками більш високої степені можна зневажити.

3. При додаванні додаткового вузла в таблицю кожний доданок многочлена Лагранжа змінюється, що впливає на значення многочлена в цілому. Додавання ж нового вузла інтерполяції у формулах Ньютона додасть лише новий доданок, але не змінить всіх попередніх доданків.

4. Якщо при отриманні многочленів Лагранжа і Ньютона вузли інтерполяції збігаються, то многочлени є тотожно рівними, тобто рівними є коефіцієнти при однакових степенях. Це випливає з існування тільки одного многочлена степені *n*

****

Рисунок(1.1) – Поведінка інтерполяціонної функції

# 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Опис програми та її методів

Програма реалізована на мові програмування JavaScript за допомогою бібліотеки jQuery та плагіну Hightcharts.js. Такий вибір було зроблено не випадково, тому що, таким чином доступ до програми мають усі користувачі інтернет ресурсів, це повністю виключає процес інсталяції програми на компьютері та дає змогу використовувати програму не залежно від характеристик ЕВМ.

Логіка програми, доволі, проста. Плагін Hightcharts.js дає змогу побудувати графік функції по її табличним значення, де точність розбиття відрізку аргументів вибирає сам користувач. Всі математичні операції та реалізація методу Лагранжа приведені у файлі math.js ( Дотаток А) процедурно. А використання цих методів та функцій приведена у файлі lagrange.js (Дотаток Б) у вигляді коду jQuery.

3.2 Приклад роботи програми

Побудуємо інтерполяційну функцію для *f(x) = tg(x),* що має наступні значення:

*x0 = -1.5 f(x0) = -14.1014*

*x1 = -0.75 f(x1) = -0.93*

*x2 = 0 f(x2) = 0*

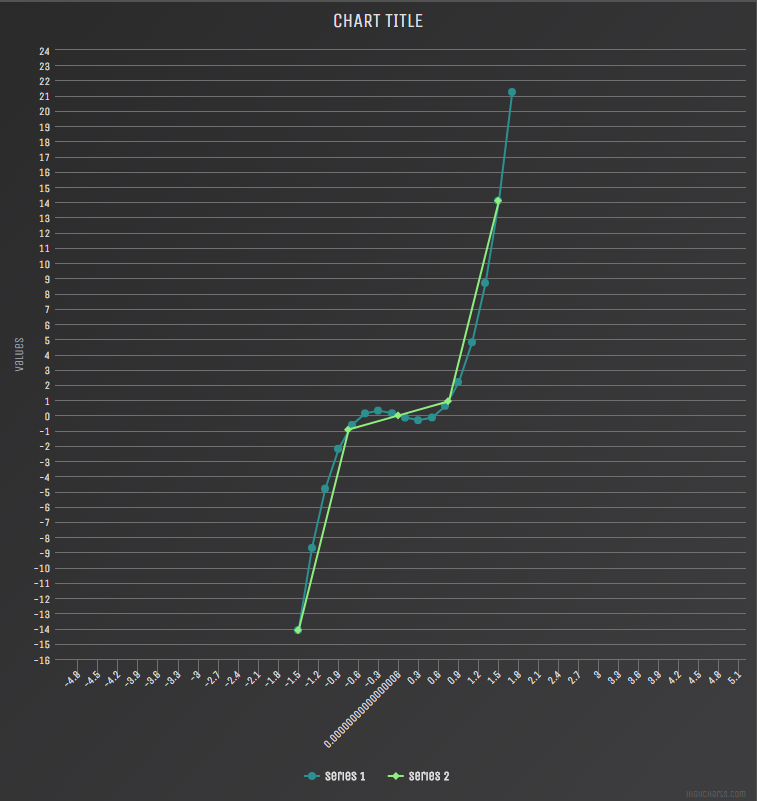
*x3 = 0.75 f(x3) = 0.93*

*x4 = 1.5 f(x4) = 14.1014*



Рисунок(1.2) – Ведення вузлів

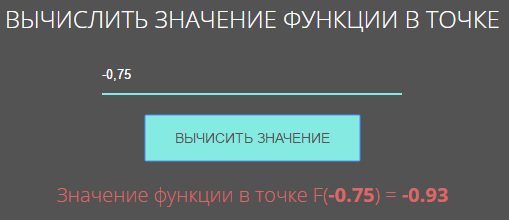
Після задання координат точок отримаємо:



Рисунок(1.3) – Результат роботи програми

Це інтерполяційна функція для *f(x) = tg(x),*  що проходить через задані точки та має високу точність значень.

Для перевірки цього використаємо метод, що визначає значення функції в точці вже заданої нам таблично:



Рис(1.4) – Вичислення значення функції в точці

Це відповідає значенню функції в точці *f(x1) = -0.93*

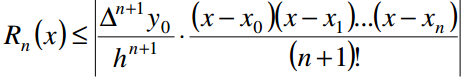
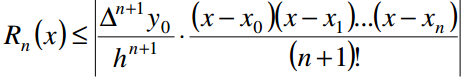
Отже, можна зробити висновок, що така програма точно знайде значення інтерполяційної функції.

# ВИСНОВКИ

Існує багато різних способів інтерполяції. Вибір найпридатнішого [алгоритму](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) залежить від відповідей на питання: наскільки точний обраний метод, які затрати на його використання, наскільки гладкою є інтерполяційна функція, яку кількість точок даних вона вимагає і т.д.

Найкращим варіантом інтерполяційного многочлена для практичних (або обчислювальних) цілей є барицентрична форма інтерполяції Лагранжа або [поліном Ньютона](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BD%D0%BE%D0%BC_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0) тому що:

1. Оцінка похибки використання інтерполяційної формули Ньютона  
визначається так само, як і для многочлена Лагранжа.  
Однак, якщо вузли інтерполяції рівновіддалені та є відомим  
додатковий вузол інтерполяції *xn*1, то для оцінки похибки *Rn**x*можна  
скористатися формулою



2. При додаванні додаткового вузла в таблицю кожний доданок  
многочлена Лагранжа змінюється, що впливає на значення многочлена  
в цілому. Додавання ж нового вузла інтерполяції у формулах Ньютона  
додасть лише новий доданок, але не змінить всіх попередніх доданків.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Численные методы, Книга 2, Методы математической физики, Калиткин Н.Н., Корякин П.В., 2013.

2. Численные методы А.А. Самарский, А.В. Гулин,

3. Интерполяция. Методы и компьютерные технологии их реализации Половко А., Бутусов П.

4. Численные методы [Електронний ресурс]. Режим доступу - <http://studopedia.su/13_102987_sutnist-personalu-pidpriiemstva-ta-yogo-struktura.html>

### 5. Чисельні методи Возняк Л. С., Шарин С. В. 2001

6. ЧИСЕЛЬНІ МЕТОДИ Задачин В. М. Конюшенко І. Г. 2014

7. Численные методы [Електронний ресурс]. Режим доступу - <https://uk.wikipedia.org/wiki/Численні_методи>

8. Численные методы [Електронний ресурс]. Режим доступу - <http://stud.com.ua/53355/informatika/interpolyatsiya_aproksimatsiya_eksperimentalnih_danih>

9. Н.Н. Калинтин, Численные методы Глава 2 Апроскимация функций

# 10. Методы аппроксимации функций [Електронний ресурс]. Режим доступу - <http://de.ifmo.ru/bk_netra/page.php?dir=3&tutindex=35&index=19&layer=1>

ДОДАТОК А

$(document).ready(function(){

$('#breakPoints').on('focusout',function(){

var numberOfBreakpoints = $(this).val();

$('.xAxis').empty();

$('.yAxis').empty();

for (var i = 0; i < numberOfBreakpoints; i++)

{

$('.xAxis').append('<input type="number" name="X'+i+'" placeholder="x' + i + '">');

$('.yAxis').append('<input type="number" name="Y'+i+'" placeholder="y' + i + '">');

}

// console.log(numberOfBreakpoints);

$('.xAxis, .yAxis').slideDown();

})

});

$(document).ready(function(){

$('button[name="submit"]').click(function(e){

e.preventDefault();

var numberOfBreakpoints = $('#breakPoints').val();

dataX = [];

dataY =[];

for(var i = 0; i < numberOfBreakpoints; i++)

{

dataX[i] = Number($('.xAxis').children('input[name=X'+i+']').val());

dataY[i] = Number($('.yAxis').children('input[name=Y'+i+']').val());

}

tempX = dataX;

tempY = dataY;

var temp = [];

temp[0] = -max(dataX);

for(var i = 0; i < dataX.length; i++)

{

temp[i+1] = dataX[i];

}

console.log(temp);

init();

//buildPoints(dataY,dataX);

dataX2 = divisionDataX(temp,15);

var Y = [];

console.log(dataX,dataY,dataX2);

setTimeout(function(){

for(var i = 0; i < dataX2.length; i++)

{

Y[i] = lagrange(dataX,dataY,dataX2[i]);

}

console.log(Y,dataX2);

},1000)

setTimeout(function(){

initData(dataX2,Y,tempY,tempX);

},1000);

})

})

$(document).ready(function(){

$('button[name="submitInPoint"]').on('click',function(e){

e.preventDefault();

var PointValue = $('input[name="valueInPoint"]').val();

PointValue = Number(PointValue);

//console.log(PointValue);

$('.answerPoint > .point').html(PointValue);

var numberOfBreakpoints = $('#breakPoints').val();

dataX = [];

dataY =[];

for(var i = 0; i < numberOfBreakpoints; i++)

{

dataX[i] = Number($('.xAxis').children('input[name=X'+i+']').val());

dataY[i] = Number($('.yAxis').children('input[name=Y'+i+']').val());

}

var Value = lagrange(dataX,dataY,PointValue);

//console.log(Value);

$('.answerPoint > .value').html(Value);

})

})

ДОТАТОК Б

//jQuery.fn.extend({

// polynome : function(){

//

// }

//});

function builtChart(dataY,dataX){

//chart.series[0].data.length = 0;

for(var i = 0; i < dataY.length; i++){

chart.series[0].addPoint([

dataY[i],

dataX[i],

]);

}

console.log(chart.series[0].data);

}

function initData(dataY,dataX,dataY2,dataX2){

init();

var dataY = dataY;

var dataX = dataX;

console.log(chart);

builtChart(dataY,dataX);

buildPoints(dataY2,dataX2);

}

function buildPoints(dataX,dataY)

{

for(var i = 0; i < dataY.length; i++){

chart.series[1].addPoint([

dataY[i],

dataX[i],

]);

}

}

function init(dataX,dataY)

{

chart = Highcharts.chart('container', {

chart:{

type: 'line',

},

yAxis:{

tickInterval : 0.5,

},

xAxis : {

tickLength : 20,

minRange : 10,

tickPixelInterval: 10

},

series :

[{

data: []

},

{

data: []

}],

});

}

function max(dataX){

var max = dataX[0];

for(var i = 0; i < dataX.length;i++)

{

if(dataX[i]>max)

{

max = dataX[i]

}

}

return max;

}

function divisionDataX(dataX,divisionNum)

{

var max = dataX[0];

var min = dataX[0];

for(var i = 0; i < dataX.length;i++)

{

if(dataX[i]>max)

{

max = dataX[i]

}

}

for(var i = 0; i < dataX.length;i++)

{

if(dataX[i]<min)

{

min = dataX[i]

}

}

var step=(max-min)/divisionNum;

var dataTemp = [];

dataTemp[0] = min;

var i = 0;

while(dataTemp[i] < max)

{

dataTemp[i+1]=dataTemp[i]+step;

i++;

}

//console.log(dataTemp);

return dataTemp;

//console.log(step);

}

function divisionDataY(dataY,divisionNum)

{

var max = dataY[0];

var min = dataY[0];

for(var i = 0; i < dataY.length;i++)

{

if(dataY[i]>max)

{

max = dataY[i]

}

}

for(var i = 0; i < dataY.length;i++)

{

if(dataY[i]<min)

{

min = dataY[i]

}

}

var step=(max-min)/divisionNum;

var dataTemp = [];

dataTemp[0] = min;

var i = 0;

while(dataTemp[i] < max)

{

dataTemp[i+1]=dataTemp[i]+step;

i++;

}

// console.log(dataTemp);

return dataTemp;

//console.log(step);

}

function lagrange(dataX,dataY,x)

{

var sum = 0;

for(var i = 0; i < dataY.length; i++)

{

var Lagrange = dataY[i]\*basispolynome(dataX,x,i);

sum = sum + Lagrange;

console.log(Lagrange/x);

}

return sum;

}

function basispolynome(dataX,x,n)

{

var res = 1;

for(var i = 0; i < dataX.length; i++)

{

var baseCh = x - dataX[i];

var baseZN = dataX[n]-dataX[i];

var arg = baseCh/baseZN;

if(n==i){

arg = 1;

}

res = res\*arg;

// console.log(res);

}

return res;

}