Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та методи обчислень

Лабораторна робота №3

«Інтерполяція функцій»

Виконала:

студентка групи ІВ-71

Молчанова В.С.

Залікова книжка № 7110

Номер у списку групи 10

Перевірив ст.вик. Порєв В. М.

Київ

2019 р.

**Тема:** «Інтерполяція функцій».

**Мета:** Ознайомлення з інтерполяційними формулами Лагранжа, Ньютона, рекурентним співвідношенням Ейткена, методами оцінки похибки інтерполяції.

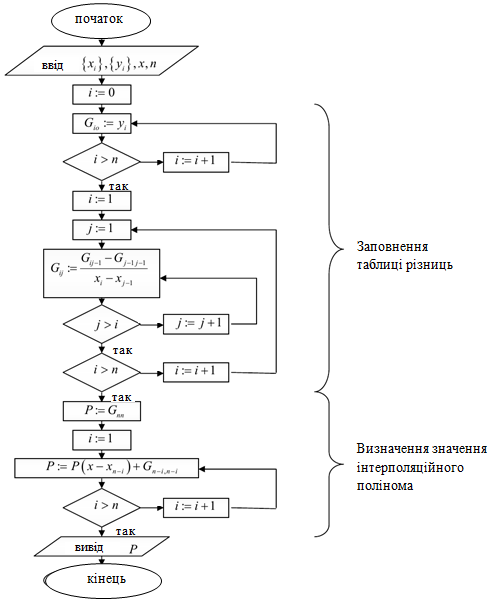
**Завдання:** Закріплення, поглиблення і розширення знань студентів при вирішенні практичних обчислювальних завдань. Оволодіння обчислювальними

методами і практичними методами оцінки похибки обчислень. Придбання умінь і навичок при програмуванні та налагодженні обчислювальних завдань на

комп'ютері.

|  |  |  |  |
| --- | --- | --- | --- |
| № Варіанта |  |  | Формула |
| 10 |  | [3,6] | 1.6 |

Блок-схема:



Текст програми:

Page {

id: *intSinPage*

width: 640

height: 400

background: Rectangle{

color: "white"

}

Row{

id: *row*

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *parent*.top

Button{

id: *countIntButton*

text: "Start"

anchors.top: *parent*.top

onClicked: {

*chart*.countInt()

}

}

Button{

id: *countErrorButton*

text: "Error"

anchors.top: *parent*.top

onClicked: {

*chart*.countError()

}

}

}

ChartView {

id: *chart*

width: *parent*.width

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *row*.bottom

anchors.bottom: *parent*.bottom

axes: [

ValueAxis{

id: *xAxis*

min: 3.0

max: 6.0

},

ValueAxis{

id: *yAxis*

min: -1.2

max: 1.2

}

]

function *countInt*() {

*chart*.removeAllSeries()

*yAxis*.min = -1.2

*yAxis*.max = 1.2

var *a* = 3;

var *b* = 6;

var *acc* = 10;

var *h* = (*b* - *a*) / 10;

var *numNodes* = 11;

var *xi* = [];

var *y* = [];

for (var *i* = 0; *i* < *numNodes*; *i*++) {

*xi*.push(*a* + *h*\**i*);

*y*.push(*Math*.sin(*xi*[*i*]));

}

var *constants* = []

var *t* = *numNodes* - 1;

var *constant* = 0;

while(*t* >= 0){

*constant* = 0;

for(*i* = 0; *i* < *numNodes*-*t*; *i*++){

var *znam* = 1;

for(var *j* = 0; *j* < *numNodes*-*t*; *j*++){

if (*i* != *j*) *znam* \*= (*xi*[*i*] - *xi*[*j*]);

}

*constant* += *y*[*i*] / *znam*;

}

*constants*.push(*constant*);

*t*--;

}

function *resSin*(x){

var *list* = [];

*list*.push(1);

for (var *i* = 0;*i* < *numNodes*;*i*++) {

*list*.push(*list*[*i*]\*(*x* - *xi*[*i*]));

}

var *f0* = 0;

for (*i* = 0; *i* < *numNodes*; *i*++){

*f0* += *constants*[*i*] \* *list*[*i*];

}

return *f0*;

}

var *seriesSin* = *chart*.createSeries(ChartView.SeriesTypeSpline, "Sin", *xAxis*, *yAxis*);

var *seriesSinInt* = *chart*.createSeries(ChartView.SeriesTypeSpline, "SinInt", *xAxis*, *yAxis*);

*seriesSin*.pointsVisible = true;

*seriesSin*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*seriesSin*.hovered.connect(function(point, state){ *console*.log(*point*); });

*seriesSinInt*.pointsVisible = true;

*seriesSinInt*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*seriesSinInt*.hovered.connect(function(point, state){ *console*.log(*point*); });

var *xRes* = []

var *yRes* = []

var *yIntRes* = []

for (*i* = 0; *i* < (*b*-*a*)\**acc*; *i*++){

var *x* = *a* + *i*/*acc*;

*xRes*.push(*x*);

*yRes*.push(*Math*.sin(*x*));

*yIntRes*.push(*resSin*(*x*));

*seriesSin*.append(*x*, *yRes*[*i*]);

*seriesSinInt*.append(*x*, *yIntRes*[*i*]);

}

var *result* = [*xRes*, *yRes*, *yIntRes*];

return *result*;

}

function *countError*() {

var *temp* = *countInt*();

*chart*.removeAllSeries();

*yAxis*.min = -0.01

*yAxis*.max = 0.01

var *series* = *chart*.createSeries(ChartView.SeriesTypeSpline, "Error", *xAxis*, *yAxis*);

*series*.pointsVisible = true;

*series*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*series*.hovered.connect(function(point, state){ *console*.log(*point*); });

for (var *i* = 0; *i* < *temp*[0].length; *i*++){

*series*.append(*temp*[0][*i*], *temp*[1][*i*] - *temp*[2][*i*]);

}

}

}

}

Page {

id: *intFuncPage*

width: 640

height: 400

background: Rectangle{

color: "white"

}

Row{

id: *row*

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *parent*.top

Button{

id: *countIntButton*

text: "Start"

anchors.top: *parent*.top

onClicked: {

*chart*.countInt()

}

}

Button{

id: *countErrorButton*

text: "Error"

anchors.top: *parent*.top

onClicked: {

*chart*.countError()

}

}

}

ChartView {

id: *chart*

width: *parent*.width

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *row*.bottom

anchors.bottom: *parent*.bottom

axes: [

ValueAxis{

id: *xAxis*

min: 3.0

max: 6.0

},

ValueAxis{

id: *yAxis*

min: -1.2

max: 1.2

}

]

function *countInt*() {

*chart*.removeAllSeries()

*yAxis*.min = -1.2

*yAxis*.max = 1.2

var *a* = 3;

var *b* = 6;

var *acc* = 10;

var *h* = (*b* - *a*) / 10;

var *numNodes* = 11;

var *xi* = [];

var *y* = [];

for (var *i* = 0; *i* < *numNodes*; *i*++) {

*xi*.push(*a* + *h*\**i*);

*y*.push(*Math*.cos(*xi*[*i*]+*Math*.exp(*Math*.cos(*xi*[*i*]))));

}

var *constants* = []

var *t* = *numNodes* - 1;

var *constant* = 0;

while(*t* >= 0){

*constant* = 0;

for(*i* = 0; *i* < *numNodes*-*t*; *i*++){

var *znam* = 1;

for(var *j* = 0; *j* < *numNodes*-*t*; *j*++){

if (*i* != *j*) *znam* \*= (*xi*[*i*] - *xi*[*j*]);

}

*constant* += *y*[*i*] / *znam*;

}

*constants*.push(*constant*);

*t*--;

}

function *resFunc*(x){

var *list* = [];

*list*.push(1);

for (var *i* = 0;*i* < *numNodes*;*i*++) {

*list*.push(*list*[*i*]\*(*x* - *xi*[*i*]));

}

var *f0* = 0;

for (*i* = 0; *i* < *numNodes*; *i*++){

*f0* += *constants*[*i*] \* *list*[*i*];

}

return *f0*;

}

var *seriesFunc* = *chart*.createSeries(ChartView.SeriesTypeSpline, "Func", *xAxis*, *yAxis*);

var *seriesFuncInt* = *chart*.createSeries(ChartView.SeriesTypeSpline, "FuncInt", *xAxis*, *yAxis*);

*seriesFunc*.pointsVisible = true;

*seriesFunc*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*seriesFunc*.hovered.connect(function(point, state){ *console*.log(*point*); });

*seriesFuncInt*.pointsVisible = true;

*seriesFuncInt*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*seriesFuncInt*.hovered.connect(function(point, state){ *console*.log(*point*); });

var *xRes* = []

var *yRes* = []

var *yIntRes* = []

for (*i* = 0; *i* < (*b*-*a*)\**acc*; *i*++){

var *x* = *a* + *i*/*acc*;

*xRes*.push(*x*);

*yRes*.push(*Math*.cos(*x*+*Math*.exp(*Math*.cos(*x*))));

*yIntRes*.push(*resFunc*(*x*));

*seriesFunc*.append(*x*, *yRes*[*i*]);

*seriesFuncInt*.append(*x*, *yIntRes*[*i*]);

}

var *result* = [*xRes*, *yRes*, *yIntRes*];

return *result*;

}

function *countError*() {

var *temp* = *countInt*();

*chart*.removeAllSeries();

*yAxis*.min = -0.05

*yAxis*.max = 0.05

var *series* = *chart*.createSeries(ChartView.SeriesTypeSpline, "Error", *xAxis*, *yAxis*);

*series*.pointsVisible = true;

*series*.color = *Qt*.rgba(*Math*.random(),*Math*.random(),*Math*.random(),1);

*series*.hovered.connect(function(point, state){ *console*.log(*point*); });

for (var *i* = 0; *i* < *temp*[0].length; *i*++){

*series*.append(*temp*[0][*i*], *temp*[1][*i*] - *temp*[2][*i*]);

}

}

}

}

import QtQuick 2.12

import QtQuick.Controls 2.2

import QtQuick.Window 2.12

import QtQuick.Layouts 1.3

ApplicationWindow {

visible: true

width: 640

height: 480

title: *qsTr*("Lab3")

background: Rectangle{

color: "white"

}

SwipeView{

id: *swipeView*

anchors.fill: *parent*

currentIndex: *tabBar*.currentIndex

HomePage{

}

IntSinPage{}

IntFuncPage{}

}

header: RowLayout{

Button{

text: "Home"

onClicked: *swipeView*.setCurrentIndex(0)

}

}

footer: TabBar{

id:*tabBar*

currentIndex: *swipeView*.currentIndex

width: *parent*.width

TabButton {

text: "Home"

width: 0

}

TabButton {

text: "Інтерполяція sin(x)"

font.pointSize: 11

}

TabButton {

text: "Інтерполяція f(x)"

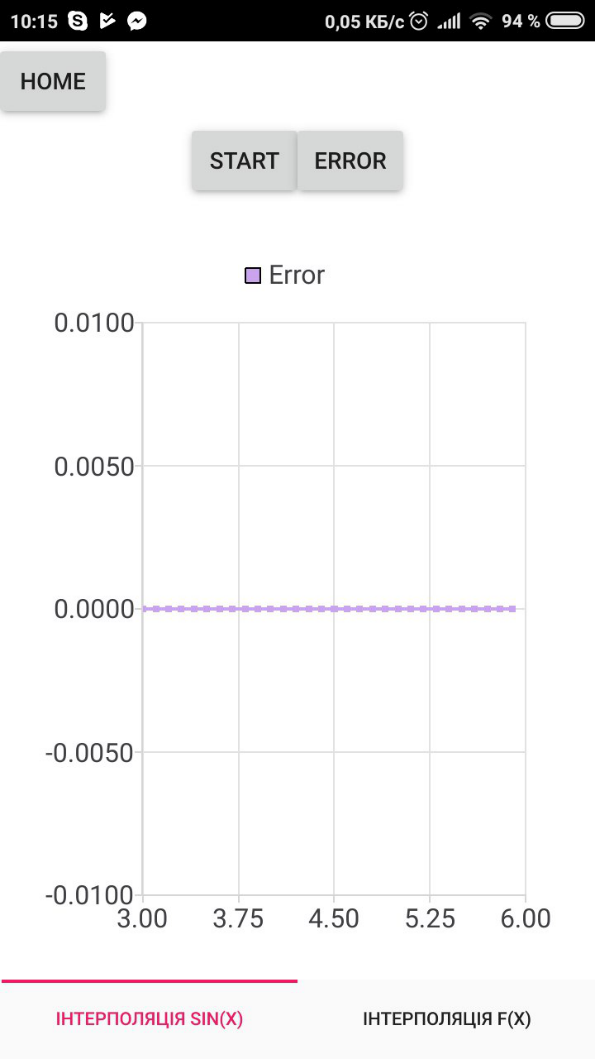
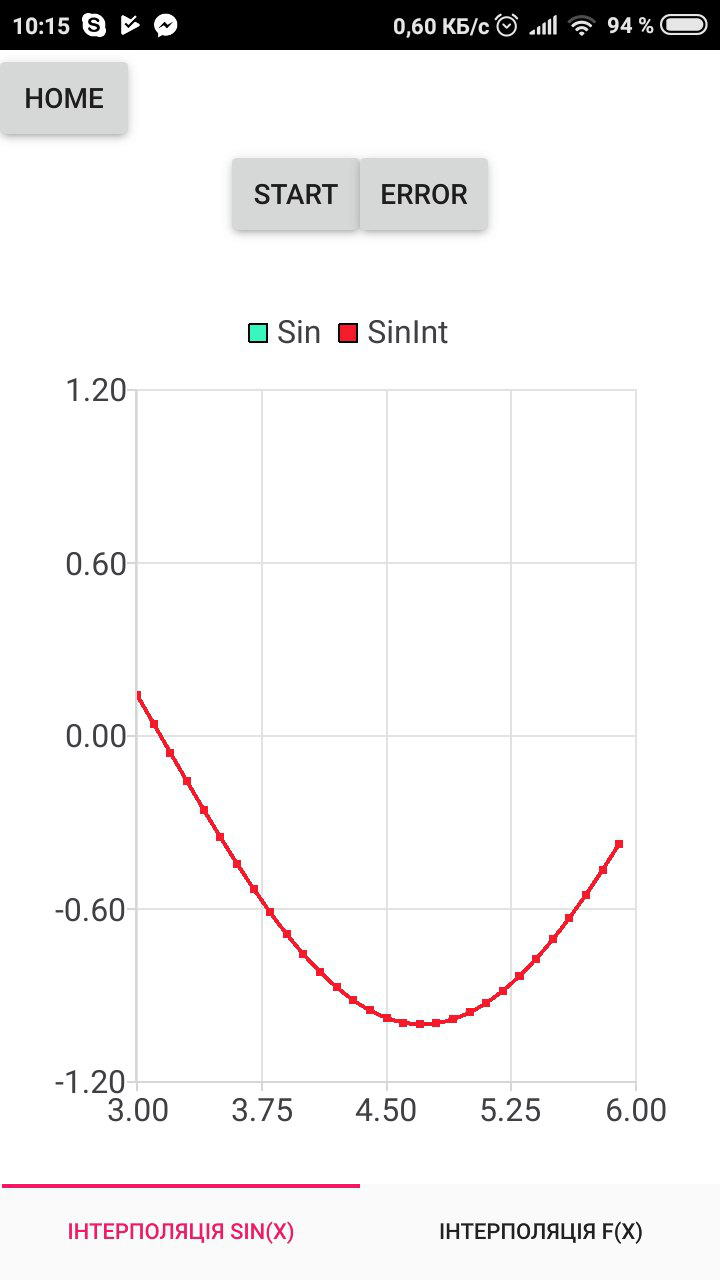
font.pointSize: 11

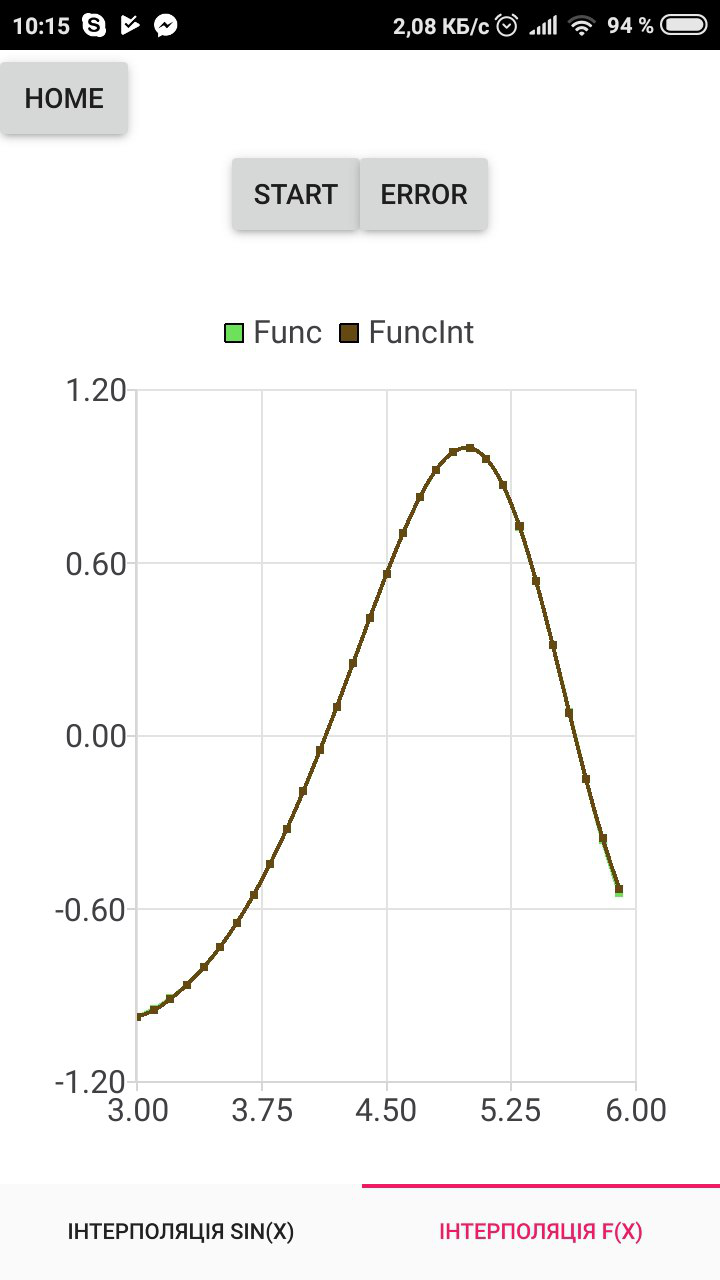
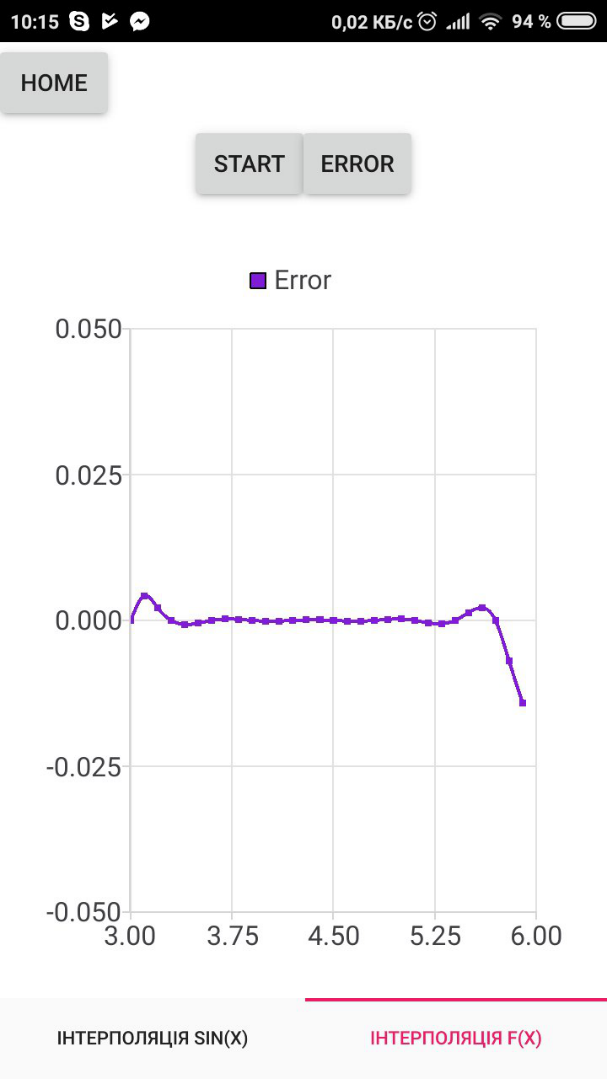
}

}

}

Результат роботи програми:



Висновок:

Під час даної лабораторної роботи я:

закріпила знання про інтерполяції, та методи її реалізації; розробила відповідну програму на основі алгоритму інтерполяції методом Ньютона;

Результати успішної роботи програми наведені вище підтверджують правильність обраних мною рішень.