**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний Технічний Університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №3**

*з дисципліни «Алгоритми та методи обчислень»*

*на тему: «Інтерполяція функцій»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Варіант:** 3

**Перевірив:**

Старший викладач

*Порєв В. М.*

**Київ – 2016**

**ЛАБОРАТОРНА РОБОТА №3**

*Інтерполяція функцій*

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

**I. Завдання**

y = esin(x), x є [0;5]

**ІІ. Код програми**

**package com.example.android.lab3;**

**import android.app.FragmentTransaction;**

**import android.content.Context;**

**import android.os.Bundle;**

**import android.support.design.widget.NavigationView;**

**import android.support.design.widget.Snackbar;**

**import android.support.v4.view.GravityCompat;**

**import android.support.v4.widget.DrawerLayout;**

**import android.support.v7.app.ActionBarDrawerToggle;**

**import android.support.v7.app.AppCompatActivity;**

**import android.support.v7.widget.Toolbar;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.view.inputmethod.InputMethodManager;**

**import android.widget.EditText;**

**public class MainActivity extends AppCompatActivity**

**implements NavigationView.OnNavigationItemSelectedListener {**

**public static double result = 0;**

**public static double x = 0;**

**public static double al = 0;**

**public static double bl = 0;**

**public static double an = 0;**

**public static double bn = 0;**

**public static int r = 0;**

**public static double delta = 0;**

**public static double delta0 = 0;**

**public static double l = 0;**

**public static double xt[];**

**public static double ft[];**

**LagrangeInterpolation lagrangeInterpolation;**

**NewtonInterpolation newtonInterpolation;**

**GraphView graphView;**

**Home home;**

**GraphViewError graphViewError;**

**/\*@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**// Inflate the menu; this adds items to the action bar if it is present.**

**getMenuInflater().inflate(R.menu.main, menu);**

**return true;**

**}\*/**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);**

**setSupportActionBar(toolbar);**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(**

**this, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);**

**drawer.setDrawerListener(toggle);**

**toggle.syncState();**

**NavigationView navigationView = (NavigationView) findViewById(R.id.nav\_view);**

**navigationView.setNavigationItemSelectedListener(this);**

**lagrangeInterpolation = new LagrangeInterpolation();**

**newtonInterpolation = new NewtonInterpolation();**

**graphView = new GraphView();**

**home = new Home();**

**graphViewError = new GraphViewError();**

**}**

**@Override**

**public void onBackPressed() {**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**if (drawer.isDrawerOpen(GravityCompat.START)) {**

**drawer.closeDrawer(GravityCompat.START);**

**} else {**

**super.onBackPressed();**

**}**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**// Handle action bar item clicks here. The action bar will**

**// automatically handle clicks on the Home/Up button, so long**

**// as you specify a parent activity in AndroidManifest.xml.**

**int id = item.getItemId();**

**//noinspection SimplifiableIfStatement**

**if (id == R.id.action\_settings) {**

**return true;**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**@SuppressWarnings("StatementWithEmptyBody")**

**@Override**

**public boolean onNavigationItemSelected(MenuItem item) {**

**int id = item.getItemId();**

**FragmentTransaction fragmentTransaction = getFragmentManager().beginTransaction();**

**if (id == R.id.home\_link) {**

**fragmentTransaction.replace(R.id.container, home);**

**} else if (id == R.id.lagrange) {**

**fragmentTransaction.replace(R.id.container, lagrangeInterpolation);**

**} else if (id == R.id.newton) {**

**fragmentTransaction.replace(R.id.container, newtonInterpolation);**

**} else if (id == R.id.graphview) {**

**fragmentTransaction.replace(R.id.container, graphView);**

**} else if (id == R.id.graphviewerror) {**

**fragmentTransaction.replace(R.id.container, graphViewError);**

**}**

**fragmentTransaction.commit();**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**drawer.closeDrawer(GravityCompat.START);**

**drawer.setDrawerListener(new DrawerLayout.DrawerListener() {**

**@Override**

**public void onDrawerSlide(View drawerView, float slideOffset) {**

**}**

**@Override**

**public void onDrawerOpened(View drawerView) {**

**}**

**@Override**

**public void onDrawerClosed(View drawerView) {**

**}**

**@Override**

**public void onDrawerStateChanged(int newState) {**

**InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT\_METHOD\_SERVICE);**

**imm.hideSoftInputFromWindow(findViewById(R.id.drawer\_layout).getWindowToken(), 0);**

**}**

**});**

**return true;**

**}**

**public static double[] getXt(double a, double b, int r) {**

**l = b - a;**

**delta0 = l / (r - 1);**

**delta = a;**

**double xt[] = new double[r + 1];**

**for (int i = 0; delta <= a + l; i++) {**

**xt[i] = delta;**

**delta += delta0;**

**}**

**xt[r] = b;**

**return xt;**

**}**

**public static double[] getFt(int r, double[] xt) {**

**double ft[] = new double[r + 1];**

**for (int i = 0; i < xt.length; i++) {**

**ft[i] = Math.exp(Math.sin(xt[i]));**

**delta += delta0;**

**}**

**ft[r] = Math.exp(Math.sin(xt[r]));**

**return ft;**

**}**

**public boolean emptinessCheck(String editTextX, String editTextA, String editTextB, String editTextR) {**

**return !editTextA.isEmpty() && !editTextB.isEmpty() && !editTextR.isEmpty() && !editTextX.isEmpty();**

**}**

**public void calculatelagrange(View view) {**

**String editTextX = ((EditText) findViewById(R.id.x\_variable)).getText().toString();**

**String editTextA = ((EditText) findViewById(R.id.a\_lagrange\_variable)).getText().toString();**

**String editTextB = ((EditText) findViewById(R.id.b\_lagrange\_variable)).getText().toString();**

**String editTextR = ((EditText) findViewById(R.id.t\_lagrange\_variable)).getText().toString();**

**if (emptinessCheck(editTextX, editTextA, editTextB, editTextR)) {**

**x = Double.valueOf(editTextX);**

**al = Double.valueOf(editTextA);**

**bl = Double.valueOf(editTextB);**

**r = Integer.valueOf(editTextR);**

**if (!(al == bl)) {**

**if (x < al || x > bl) {**

**Snackbar.make(view, R.string.AboveBorders, Snackbar.LENGTH\_LONG).show();**

**} else {**

**if (!(al > bl)) {**

**if (r == 0) {**

**Snackbar.make(view, R.string.ZeroDivision, Snackbar.LENGTH\_LONG).show();**

**} else {**

**InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT\_METHOD\_SERVICE);**

**imm.hideSoftInputFromWindow(view.getWindowToken(), 0);**

**result = 0;**

**xt = getXt(al, bl, r);**

**ft = getFt(r, xt);**

**double result = getInterpolationLagrange(x, r, xt, ft);**

**((EditText) findViewById(R.id.y\_variable)).setText(String.valueOf(result));**

**double error = getErrorLagrange(x);**

**((EditText) findViewById(R.id.er\_lagrange)).setText(String.valueOf(error));**

**double errorOfError = getErrorOfErrorLagrange(x);**

**((EditText) findViewById(R.id.blurrinessLagrange)).setText(String.valueOf(getBlurinessLagr()));**

**GraphView.getSeriesLagrange(xt, ft);**

**GraphViewError.getErrorSeriesLagrange();**

**}**

**} else {**

**Snackbar.make(view, R.string.BeginEndError, Snackbar.LENGTH\_LONG).show();**

**}**

**}**

**} else {**

**Snackbar.make(view, R.string.NotADot, Snackbar.LENGTH\_LONG).show();**

**}**

**} else {**

**Snackbar.make(view, R.string.EmptyError, Snackbar.LENGTH\_LONG).show();**

**}**

**}**

**public static double getInterpolationLagrange(double x, int r, double[] xt, double[] ft) {**

**result = 0;**

**for (int i = 0; i < r; i++) {**

**double temp = 1;**

**for (int j = 0; j < i; j++) {**

**temp = temp \* (x - xt[j]) / (xt[i] - xt[j]);**

**}**

**for (int j = i + 1; j < r; j++) {**

**temp = temp \* (x - xt[j]) / (xt[i] - xt[j]);**

**}**

**result = result + temp \* ft[i];**

**}**

**return result;**

**}**

**public void calculateNewton(View view) {**

**String editTextX = ((EditText) findViewById(R.id.x\_newton\_variable)).getText().toString();**

**String editTextA = ((EditText) findViewById(R.id.a\_newton\_variable)).getText().toString();**

**String editTextB = ((EditText) findViewById(R.id.b\_newton\_variable)).getText().toString();**

**String editTextR = ((EditText) findViewById(R.id.t\_newton\_variable)).getText().toString();**

**if (emptinessCheck(editTextX, editTextA, editTextB, editTextR)) {**

**x = Double.valueOf(editTextX);**

**an = Double.valueOf(editTextA);**

**bn = Double.valueOf(editTextB);**

**r = Integer.valueOf(editTextR);**

**if (!(an == bn)) {**

**if (!(an > bn)) {**

**if (x < an || x > bn) {**

**Snackbar.make(view, getString(R.string.AboveBorders), Snackbar.LENGTH\_LONG).show();**

**} else {**

**if (r == 0) {**

**Snackbar.make(view, getString(R.string.ZeroDivision), Snackbar.LENGTH\_LONG).show();**

**} else {**

**InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT\_METHOD\_SERVICE);**

**imm.hideSoftInputFromWindow(view.getWindowToken(), 0);**

**result = 0;**

**xt = getXt(an, bn, r);**

**ft = getFt(r, xt);**

**double result = getInterpolationNewton(x, xt, r, ft);**

**((EditText) findViewById(R.id.y\_newton\_variable)).setText(String.valueOf(result));**

**double error = getErrorNewton(x);**

**((EditText) findViewById(R.id.er\_newton)).setText(String.valueOf(error));**

**((EditText) findViewById(R.id.blurrinessNewton)).setText(String.valueOf(getBlurinessNewton()));**

**GraphView.getSeriesNewton(xt, ft);**

**GraphViewError.getErrorSeriesNewton();**

**}**

**}**

**} else {**

**Snackbar.make(view, getString(R.string.BeginEndError), Snackbar.LENGTH\_LONG).show();**

**}**

**} else {**

**Snackbar.make(view, getString(R.string.NotADot), Snackbar.LENGTH\_LONG).show();**

**}**

**} else {**

**Snackbar.make(view, getString(R.string.EmptyError), Snackbar.LENGTH\_LONG).show();**

**}**

**}**

**public static double getInterpolationNewton(double x, double[] xt, int r, double[] ft) {**

**result = ft[0];**

**double buf = 1;**

**for (int k = 1; k < r; k++) {**

**double tempSum = 0;**

**for (int i = 0; i <= k; i++) {**

**double temp = 1;**

**for (int j = 0; j < i; j++) {**

**temp = temp \* (xt[i] - xt[j]);**

**}**

**for (int j = i + 1; j <= k; j++) {**

**temp = temp \* (xt[i] - xt[j]);**

**}**

**temp = ft[i] / temp;**

**tempSum += temp;**

**}**

**buf = buf \* (x - xt[k - 1]);**

**result = result + tempSum \* buf;**

**}**

**return result;**

**}**

**public static double getErrorLagrange(double x) {**

**double[] xt = getXt(al, bl, r);**

**double[] ft = getFt(r, xt);**

**double result = getInterpolationLagrange(x, r, xt, ft);**

**double result1 = Math.exp(Math.sin(x));**

**return result1 - result;**

**}**

**public static double getErrorOfErrorLagrange(double x) {**

**double result2 = Math.exp(Math.sin(x));**

**double[] xt = getXt(al, bl, r + 1);**

**double[] ft = getFt(r + 1, xt);**

**double result1 = getInterpolationLagrange(x, r + 1, xt, ft);**

**xt = getXt(al, bl, r);**

**ft = getFt(r, xt);**

**double result = getInterpolationLagrange(x, r, xt, ft);**

**return (result2 - result1) / (result1 - result) \* Math.pow(10, -7);**

**}**

**public static double getBlurinessLagr() {**

**return getErrorOfErrorLagrange(x)/getErrorLagrange(x);**

**}**

**public static double getErrorNewton(double x) {**

**double[] xt = getXt(an, bn, r);**

**double[] ft = getFt(r, xt);**

**double result = getInterpolationNewton(x, xt, r, ft);**

**double result1 = Math.exp(Math.sin(x));**

**return result1 - result;**

**}**

**public static double getErrorOfErrorNewton() {**

**double result2 = Math.exp(Math.sin(x));**

**double[] xt = getXt(an, bn, r + 1);**

**double[] ft = getFt(r + 1, xt);**

**double result1 = getInterpolationNewton(x, xt, r + 1, ft);**

**xt = getXt(an, bn, r);**

**ft = getFt(r, xt);**

**double result = getInterpolationNewton(x, xt, r, ft);**

**return (result2 - result1) / (result1 - result) \* Math.pow(10, -7);**

**}**

**public static double getBlurinessNewton() {**

**return getErrorOfErrorNewton()/getErrorNewton(x);**

**}**

**public void loadFromMemory(View view) {**

**if (x != 0 && r != 0) {**

**if (al == 0 && an == 0 && bl == 0 && bn != 0 && an!=bn) {**

**((EditText) findViewById(R.id.a\_lagrange\_variable)).setText(String.valueOf(an));**

**((EditText) findViewById(R.id.b\_lagrange\_variable)).setText(String.valueOf(bn));**

**((EditText) findViewById(R.id.x\_variable)).setText(String.valueOf(x));**

**((EditText) findViewById(R.id.t\_lagrange\_variable)).setText(String.valueOf(r));**

**}**

**if (al == 0 && an == 0 && bl != 0 && bn == 0) {**

**((EditText) findViewById(R.id.a\_newton\_variable)).setText(String.valueOf(al));**

**((EditText) findViewById(R.id.b\_newton\_variable)).setText(String.valueOf(bl));**

**((EditText) findViewById(R.id.x\_newton\_variable)).setText(String.valueOf(x));**

**((EditText) findViewById(R.id.t\_newton\_variable)).setText(String.valueOf(r));**

**}**

**}**

**else Snackbar.make(view, R.string.EmptyMemory, Snackbar.LENGTH\_LONG).show();**

**}**

**}**

**ІІІ. Результати**

|  |  |
| --- | --- |
|  |  |
|  |  |

**ІV. Висновок**

У ході виконання лабораторної роботи було закріплено знання з інтерполяції функцій методом Лагранжа та Ньютона. Була написана програма, що інтерполює задану функцію, будує графіки заданої функції, її інтерпольвану версію та похибки інтерполяції.