

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**   
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**   
**імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**   
**Кафедра системного проектування**  
   
   
 

**Звіт**

**про виконання лабораторної роботи №1**7  
**«Графіка»**

Виконав:   
студент I курсу, групи ДА-12   
Ніколаєв Роман Юрійович

Прийняв:

к.т.н., доцент Безносик О. Ю.

**Завдання**

Дослідити область визначення і побудувати графік функції

y = ln (sqrt (x))

**Аналіз умови задачі**

Область визначення ф-її – x>=0.

Для створення вікна, відмальовки осей та графіка була використана бібліотека sfml. Для зручності обрахування координат, функція була перетворена до вигляду x=exp(2\*y). Координати заносились у вектор.

**Код програми**

#include <SFML/Graphics.hpp>

#include <iostream>

#include <string>

#include <vector>

#include <cmath>

#include <unistd.h>

#define XMIN 0

#define XMAX 2

#define ACCURACY 1000

using namespace std;

using namespace sf;

vector <pair <float, float>> curve\_coord(float xl, float xr, float yb, float yt)

{

vector <pair <float, float>> coord;

if ((xl <= XMAX) && (xr >= XMIN)) {

float x, y;

for (y = yb; y <= yt; y += (yt - yb) / ACCURACY) {

x = exp(2 \* y);

coord.push\_back({ x, y });

}

}

return coord;

}

void draw\_graph(RenderWindow& a\_window, float xl, float xr, float yb, float yt, vector <pair <float, float> >& a\_curve, VertexArray& graph)

{

graph.clear();

int i;

for (i = 0; i < a\_curve.size(); i++) {

float xpic = (a\_curve[i].first - xl) / (xr - xl) \* a\_window.getSize().x;

float ypic = (yt - a\_curve[i].second) / (yt - yb) \* a\_window.getSize().y;

graph.append(Vertex(Vector2f(xpic, ypic), Color::Blue));

}

}

void draw\_x\_axis(RenderWindow& a\_window, float xl, float xr, float yb, float yt, Font& a\_font, VertexArray& axis)

{

float ypic, markb, markt; int i;

Text name("X", a\_font);

name.setFillColor(Color::Green);

if (yb >= 0) {

ypic = markb = a\_window.getSize().y - 1;

markt = markb - 7;

name.setPosition(a\_window.getSize().x - 15, ypic - 45);

}

else if (yt <= 0) {

ypic = markt = 1;

markb = markt + 7;

name.setPosition(a\_window.getSize().x - 15, ypic);

}

else {

ypic = (yt) / (yt - yb) \* a\_window.getSize().y;

markb = ypic + 10;

markt = ypic - 10;

name.setPosition(a\_window.getSize().x - 15, ypic);

}

a\_window.draw(name);

axis.clear();

axis.append(Vertex(Vector2f(a\_window.getSize().x, ypic), Color::Green));

axis.append(Vertex(Vector2f(a\_window.getSize().x - 10, ypic - 10), Color::Green));

axis.append(Vertex(Vector2f(a\_window.getSize().x, ypic), Color::Green));

axis.append(Vertex(Vector2f(a\_window.getSize().x - 10, ypic + 10), Color::Green));

axis.append(Vertex(Vector2f(0, ypic), Color::Green));

axis.append(Vertex(Vector2f(a\_window.getSize().x, ypic), Color::Green));

for (i = trunc(xl); i <= trunc(xr); i++) {

float mark = (i - xl) / (xr - xl) \* a\_window.getSize().x;

//Отрисовка пометки

axis.append(Vertex(Vector2f(mark, markt), Color::Green));

axis.append(Vertex(Vector2f(mark, markb), Color::Green));

Text txt(to\_string(i), a\_font, 20);

if (yb >= 0)

txt.setPosition(mark - 7, markt - 23);

else

txt.setPosition(mark - 7, markb - 5);

txt.setFillColor(Color::Green);

a\_window.draw(txt);

}

}

void draw\_y\_axis(RenderWindow& a\_window, float xl, float xr, float yb, float yt, Font& a\_font, VertexArray& axis)

{

float xpic, markl, markr; int i;

Text name("Y", a\_font);

name.setFillColor(Color::Red);

if (xl >= 0) {

xpic = markl = 1;

markr = markl + 7;

name.setPosition(xpic + 15, -5);

}

else if (xr <= 0) {

xpic = markr = a\_window.getSize().x - 1;

markl = markr - 7;

name.setPosition(xpic - 25, -5);

}

else {

xpic = (0 - xl) / (xr - xl) \* a\_window.getSize().x;

markl = xpic - 10;

markr = xpic + 10;

name.setPosition(xpic - 25, -5);

}

a\_window.draw(name);

axis.clear();

axis.append(Vertex(Vector2f(xpic, 0), Color::Red));

axis.append(Vertex(Vector2f(xpic - 10, 10), Color::Red));

axis.append(Vertex(Vector2f(xpic, 0), Color::Red));

axis.append(Vertex(Vector2f(xpic + 10, 10), Color::Red));

axis.append(Vertex(Vector2f(xpic, 0), Color::Red));

axis.append(Vertex(Vector2f(xpic, a\_window.getSize().y), Color::Red));

for (i = trunc(yb); i <= trunc(yt); i++) {

float mark = (yt - i) / (yt - yb) \* a\_window.getSize().y;

axis.append(Vertex(Vector2f(markl, mark), Color::Red));

axis.append(Vertex(Vector2f(markr, mark), Color::Red));

Text txt(to\_string(i), a\_font, 20);

if (xl >= 0)

txt.setPosition(markr + 2, mark - 16);

else

txt.setPosition(markl - 15, mark - 16);

txt.setFillColor(Color::Red);

a\_window.draw(txt);

}

}

int main()

{

float x\_left = -1, x\_right = 6, y\_bottom = -5, y\_top = 2, zoom = 1;

float x\_left\_starting = x\_left, x\_right\_starting = x\_right, y\_bottom\_starting = y\_bottom, y\_top\_starting = y\_top;

Font font;

font.loadFromFile("calibri.otf");

if ((x\_left > 0) && (x\_left < 2)) {

y\_bottom = -4 \* M\_PI;

y\_top = 4 \* M\_PI;

}

else if ((x\_right < 2) && (x\_right > 0)) {

y\_bottom = -2 \* M\_PI;

y\_top = 2 \* M\_PI;

}

RenderWindow window(VideoMode(600, 600), "Graph");

while (window.isOpen())

{

VertexArray curve(*LineStrip*, 0);

VertexArray y\_axis(*Lines*, 0);

VertexArray x\_axis(*Lines*, 0);

Event event;

while (window.pollEvent(event))

{

switch (event.type) {

case Event::*Closed*:

window.close();

case Event::*MouseWheelMoved*:

if ((event.mouseWheel.delta == 1) && (zoom < (abs(x\_left\_starting)+abs(x\_right\_starting))/2)) {

zoom++;

//cout << "1: " << zoom << endl;

x\_left++;

x\_right--;

y\_bottom++;

y\_top--;

}

else if ((event.mouseWheel.delta == -1) && (zoom > -((abs(x\_left\_starting)+abs(x\_right\_starting))/2)\*2) ) {

zoom--;

//cout << "2: " << zoom << endl;

x\_left--;

x\_right++;

y\_bottom--;

y\_top++;

}

}

}

float moving\_speed = 0.01;

if (Keyboard::isKeyPressed(Keyboard::*Down*)) {

y\_bottom -= moving\_speed;

y\_top -= moving\_speed;

}

if (Keyboard::isKeyPressed(Keyboard::*Up*)) {

y\_bottom += moving\_speed;

y\_top += moving\_speed;

}

if (Keyboard::isKeyPressed(Keyboard::*Left*)) {

x\_left -= moving\_speed;

x\_right -= moving\_speed;

}

if (Keyboard::isKeyPressed(Keyboard::*Right*)) {

x\_left += moving\_speed;

x\_right += moving\_speed;

}

sleep(0.3);

window.clear();

vector <pair <float, float>> c = curve\_coord(x\_left, x\_right, y\_bottom, y\_top);

draw\_graph(window, x\_left, x\_right, y\_bottom, y\_top, c, curve);

draw\_y\_axis(window, x\_left, x\_right, y\_bottom, y\_top, font, y\_axis);

draw\_x\_axis(window, x\_left, x\_right, y\_bottom, y\_top, font, x\_axis);

window.draw(curve);

window.draw(y\_axis);

window.draw(x\_axis);

window.display();

}

return 0;

}

Результат виконання програми зображено на рисунках 1,2.

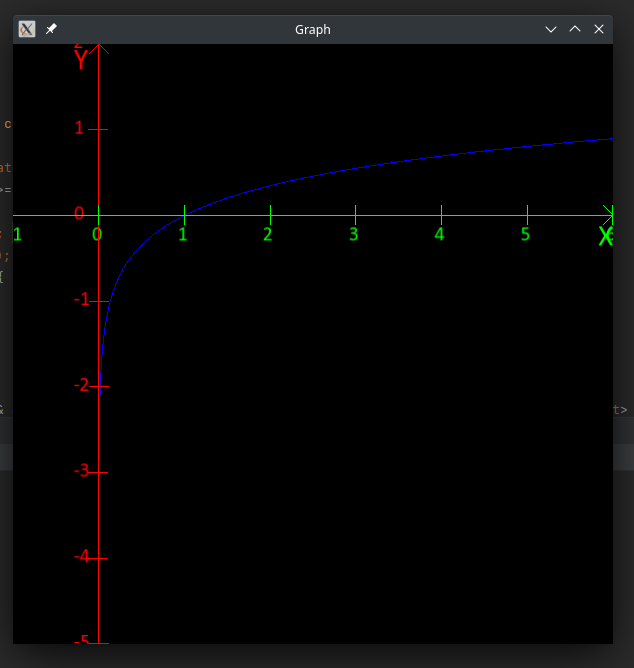


рисунок 1

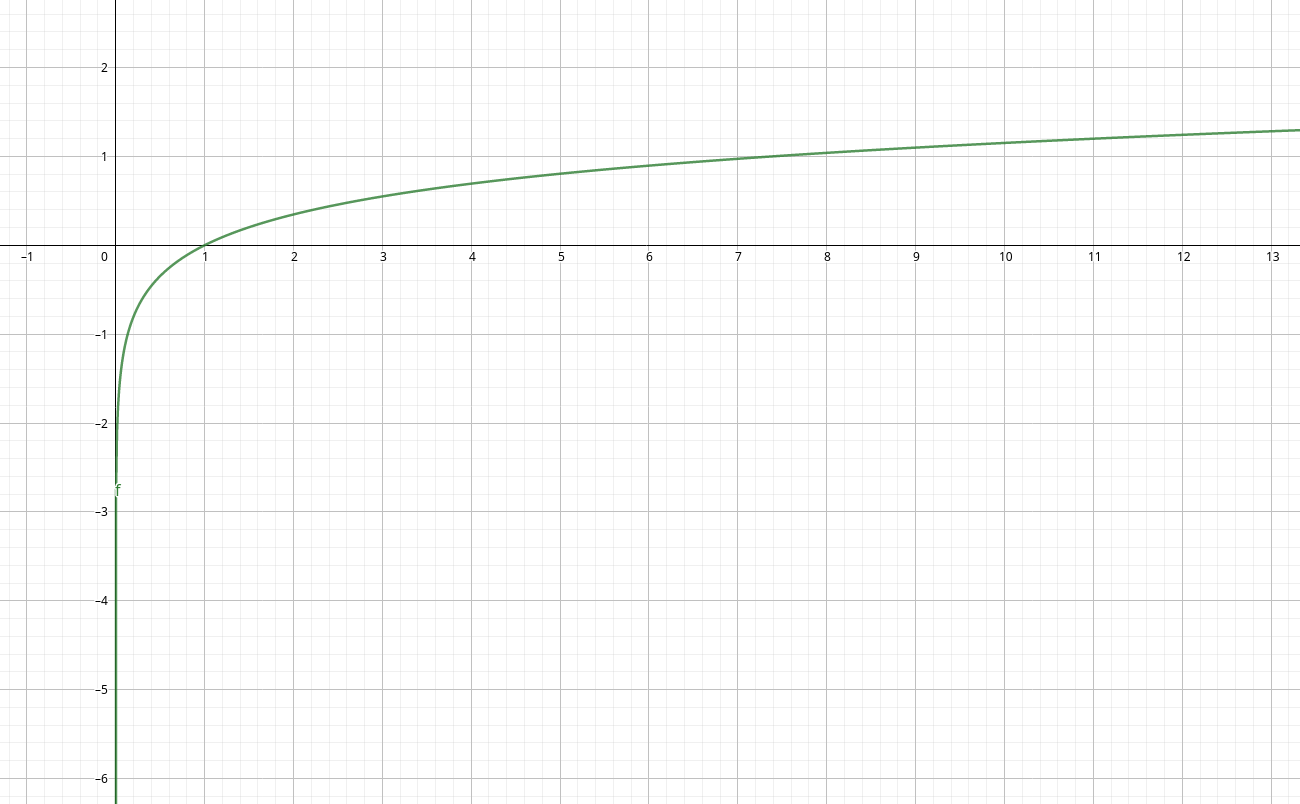


рисунок 1 – вигляд ф-її в онлайн-калькуляторі

**Виновки**

У ході виконання лабораторної роботи №17 я вивчив базовий функціонал бібліотеки sfml та навчився відтворювати графік в мові програмування С++.