**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа №1**

Тема: Построение изображений 2D-кривых

Студент: Алексеев Владислав Евгеньевич

Группа: М8О-306Б-19

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. **Постановка задачи**

Написать и отладить программу, строящую изображение заданной замечательной кривой.

**Вариант №2**: (x^2 + y^2)^2 = a^2(x^2 - y^2), где a > 0 - константа, значения которой выбираются пользователем (вводятся в окне программы).

Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна.

**2. Описание программы**

Программа состоит из окна для регулировки параметра a, из координатных осей с графиком и из навигационной панели для перемещения по графику.

**3. Набор тестов**

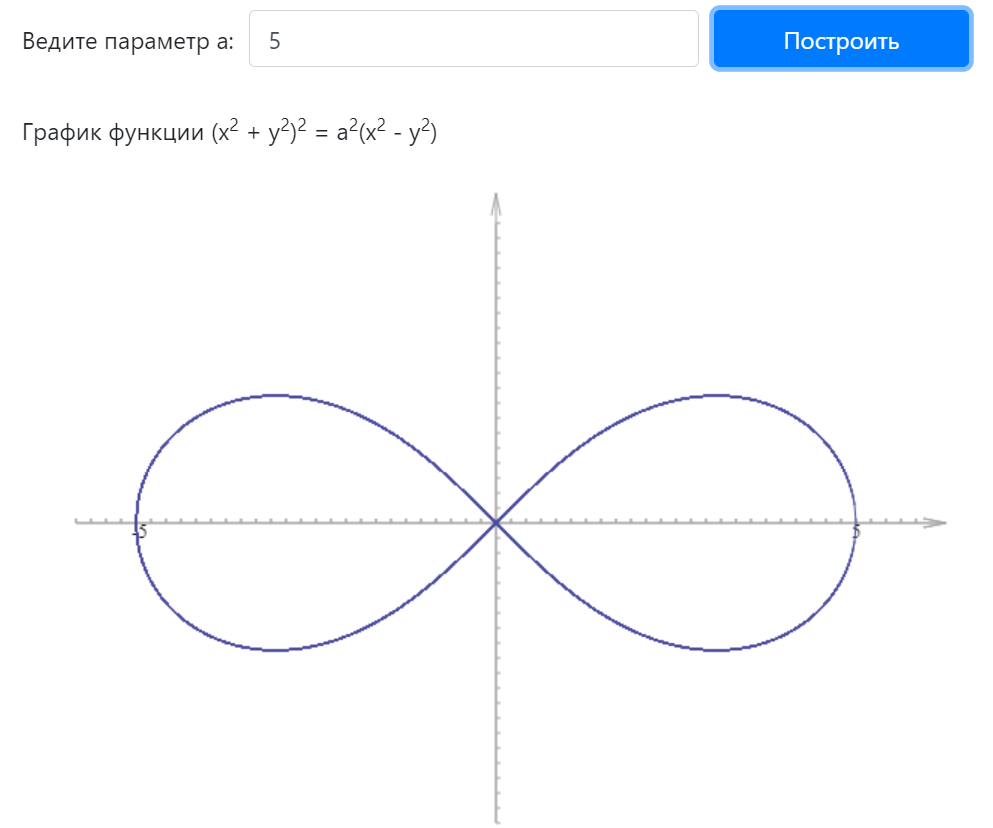
1. Начальная установка параметра а.

2. Изменение параметра a.

3. Изменение размера окна.

**4. Результаты выполнения тестов**

1. Начальная установка параметра a = 5:

****Рис. 4.1 Результат установки параметра a

2. Изменение параметра a = 10:

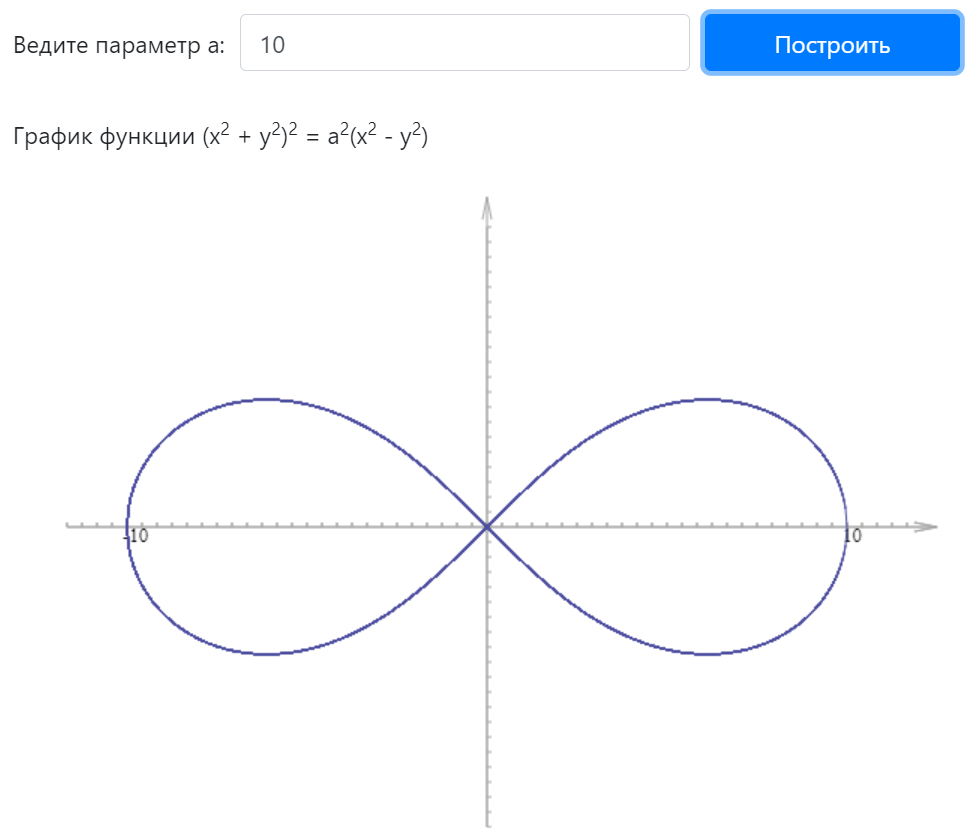


Рис. 4.2 Результат изменения параметра a

3. Изменение размера окна:

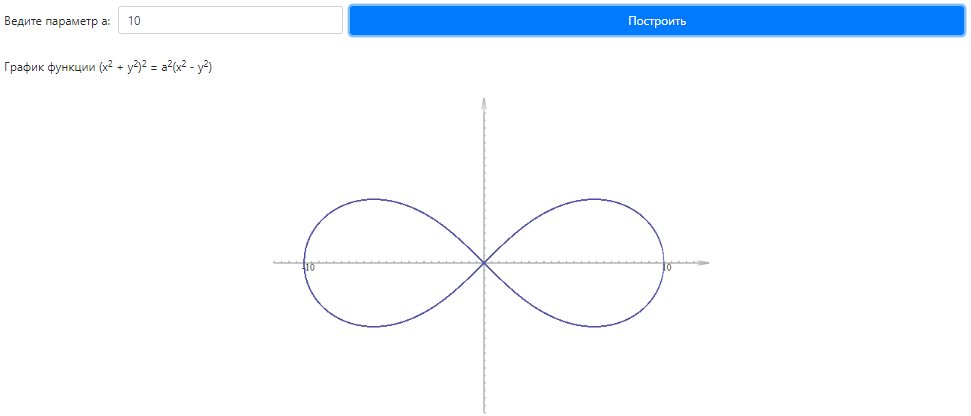


Рис. 4.3 Результат изменения размера окна

**5. Листинг программы**

**Lab1.html**

<!-- Алексеев В.Е. М80-306Б-19 -->

<!-- Условие: Вариант №2: (x^2 + y^2)^2 = a^2(x^2 - y^2) -->

<html>

<head>

<meta charset="utf8"/>

<meta name="viewport" content="width=device-width">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

</head>

<style>

.bg {

top: 0;

left: 0;

position: fixed;

width: 100%;

height: 100%;

background-image: url("https://coolbackgrounds.io/images/backgrounds/sea-edge-311c5cd5.png");

background-size: 100%;

opacity: 0.5;

z-index: -1;

}

#cs1 {

display: block;

margin: 0 auto;

}

.form {

width: 50%;

padding: 30px 0;

margin: 0 auto;

display: flex;

align-items: center;

}

#param {

width: 300px;

}

label {

margin-right: 10px;

margin-bottom: 0;

}

input {

margin-right: 10px;

}

#buildButton {

flex: 1;

}

.formula {

width: 50%;

margin: 0 auto;

margin-bottom: 10px;

}

</style>

<body>

<div class="bg"></div>

<form class="form">

<label>Введите параметр a:</label>

<input type="text" id="param" name="param" class="form-control">

<button type="button" id="buildButton" class="btn btn-primary">Построить</button>

</form>

<div class="formula">

<label>График функции (x<sup>2</sup> + y<sup>2</sup>)<sup>2</sup> = a<sup>2</sup>(x<sup>2</sup> - y<sup>2</sup>)</label>

</div>

<canvas id="cs1">???</canvas>

</body>

<script>

const canvasWidth = window.innerWidth \* 0.5;

const canvasHeight = canvasWidth \* 3 / 4;

const btn = document.getElementById('buildButton');

const canvas = document.getElementById("cs1");

canvas.width = canvasWidth;

canvas.height = canvasHeight;

const ctx = canvas.getContext("2d");

btn.addEventListener('click', start);

ctx.fillStyle = "white";

ctx.fillRect(0, 0, canvas.width, canvas.height); // Фон(белый прямоугольник)

ctx.strokeRect(0, 0, canvasWidth - 0.5, canvasHeight - 0.5); // Рамка(рисует прямоугольник)

let par = Math.ceil(canvasWidth / 2 - 90);

par = (10 - par % 10) + par;

function f1(x, par) {

y1 = -Math.sqrt((-2 \* x \* x - par \* par + Math.sqrt(8 \* par \* par \* x \* x + Math.pow(par, 4))) / 2);

return y1

}

function f2(x, par) {

y2 = Math.sqrt((-2 \* x \* x - par \* par + Math.sqrt(8 \* par \* par \* x \* x + Math.pow(par, 4))) / 2);

return y2

}

Line1 = [];

Line2 = [];

function start(event) {

const a = Number(document.getElementById('param').value);

if (!a) {

alert("Incorrect input!");

return;

}

if (a == 0) {

par = 0;

}

// Формируем массив точек

var i = 0;

for (var x = -par; x <= par; x += 1) {

p1 = [x, f1(x, par)];

p2 = [x, f2(x, par)];

Line1[i] = p1;

Line2[i] = p2;

++i;

}

ctx.fillStyle = '#fff';

ctx.fillRect(0, 0, canvasWidth, canvasHeight);

ctx.save();

ctx.strokeStyle = '#a6a6a6'; // Цвет осей

ctx.font = '13px serif';

ctx.fillStyle = '#000';

ctx.beginPath();

ctx.translate(canvasWidth / 2, canvasHeight / 2) // Начало координат

let widthHalf = Math.ceil(canvasWidth / 2);

let heightHalf = Math.ceil(canvasHeight / 2);

// Ось X

ctx.moveTo(0,0);

for (var i = 0; i > -widthHalf + 30; i -= 10) {

if (i == -par) {

ctx.fillText(-a, i - 3, 10);

}

ctx.lineTo(i, 0);

ctx.lineTo(i, -3);

ctx.moveTo(i, 0);

}

ctx.moveTo(0, 0);

for (var i = 0; i < widthHalf - 30; i += 10) {

if (i == par) {

ctx.fillText(a, i - 3, 10);

}

ctx.lineTo(i, 0);

ctx.lineTo(i, -3);

ctx.moveTo(i, 0);

}

// Стрелка для оси X

ctx.lineTo(widthHalf - 20, 0);

ctx.lineTo(widthHalf - 20 -15, -3);

ctx.moveTo(widthHalf - 20, 0);

ctx.lineTo(widthHalf - 20 -15, 3);

ctx.stroke();

// Ось Y

ctx.moveTo(0, 0);

for (var i = 0; i > -heightHalf + 30; i -= 10) {

ctx.lineTo(0, i);

ctx.lineTo(3, i);

ctx.moveTo(0, i);

}

ctx.moveTo(0, 0);

for (var i = 0; i < heightHalf - 30; i += 10) {

ctx.lineTo(0, i);

ctx.lineTo(3, i);

ctx.moveTo(0, i);

}

// Стрелка для оси Y

ctx.lineTo(0, -heightHalf + 20);

ctx.lineTo(-3, -heightHalf + 20 + 15);

ctx.moveTo(0, -heightHalf + 20);

ctx.lineTo(3, -heightHalf + 20 + 15);

ctx.stroke();

ctx.strokeStyle = '#4c4d9f';

// Линия выше оси Х

ctx.beginPath();

for (var i = 0; i < Line1.length; i++) {

ctx.lineTo(Line1[i][0], Line1[i][1]);

ctx.stroke();

}

// Линия ниже оси Х

ctx.beginPath();

for (var i = 0; i < Line2.length; i++) {

ctx.lineTo(Line2[i][0], Line2[i][1]);

ctx.stroke();

}

ctx.restore();

}

</script>

</html>

**6. Выводы**

Выполнив данную лабораторную работу, я научился пользоваться HTML, запускать графический интерфейс, перерисовывать его в зависимости от изменения окна и отрисовывать график по массиву точек.

**ЛИТЕРАТУРА**

1. Работа VS code с HTML[Электронный ресурс]URL: [https://blog.altuninvv.ru/программирование/visual-studio-code/98-настройка-vscode-для-работы-с-html](https://blog.altuninvv.ru/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/visual-studio-code/98-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-vscode-%D0%B4%D0%BB%D1%8F-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B-%D1%81-html) (Дата обращения: 20.09.2021).

2. Документация HTML[Электронный ресурс]URL: <https://htmlacademy.ru/blog/boost/tutorial/canvas-chart> (Дата обращения: 20.09.2021).