**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: Ватулин Валентин Михайлович

Группа: 80-306

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. Постановка задачи

Написать и отладить программу, строящую изображение заданной замечательной кривой.

Вариант 6:

(x2 + y2 + ax)2 = a2(x2 + y2)

1. Описание программы

Программа написана на JavaScript. Для отрисовки кривой используется Canvas API в браузере. Параметр *a* кривой можно регулировать с помощью слайдера. Отрисовка происходит при открытии страницы, изменении параметра и изменении размеров окна.

Способ отрисовки: вычислить точки кривой с помощью формулы кривой, выраженной через переменную *x*, отсортировать их по углу вектора от центра системы координат до точки, соединять точки прямыми в полученном массиве.

1. Набор тестов

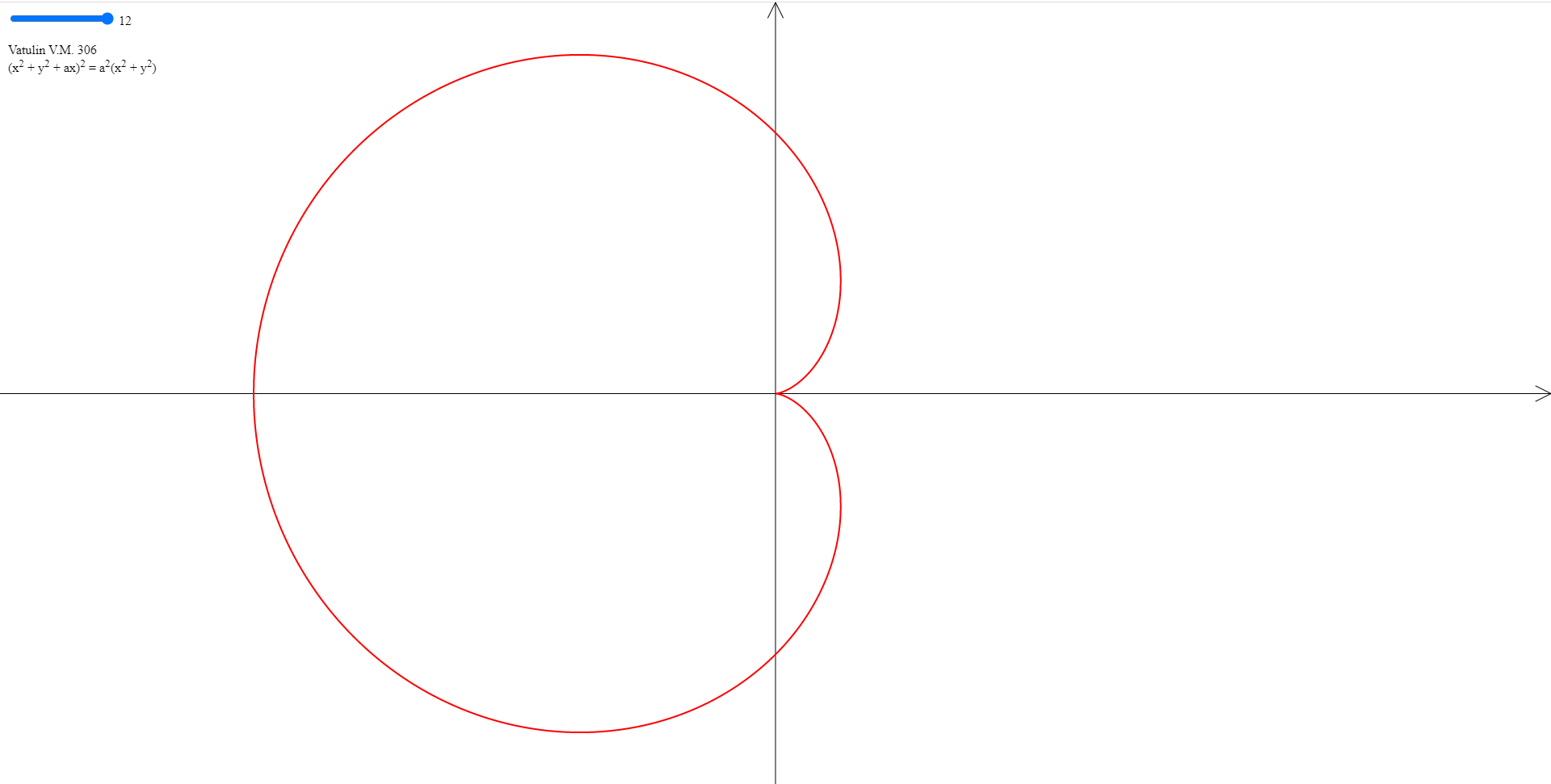


рис. 1 - Параметр равен 12

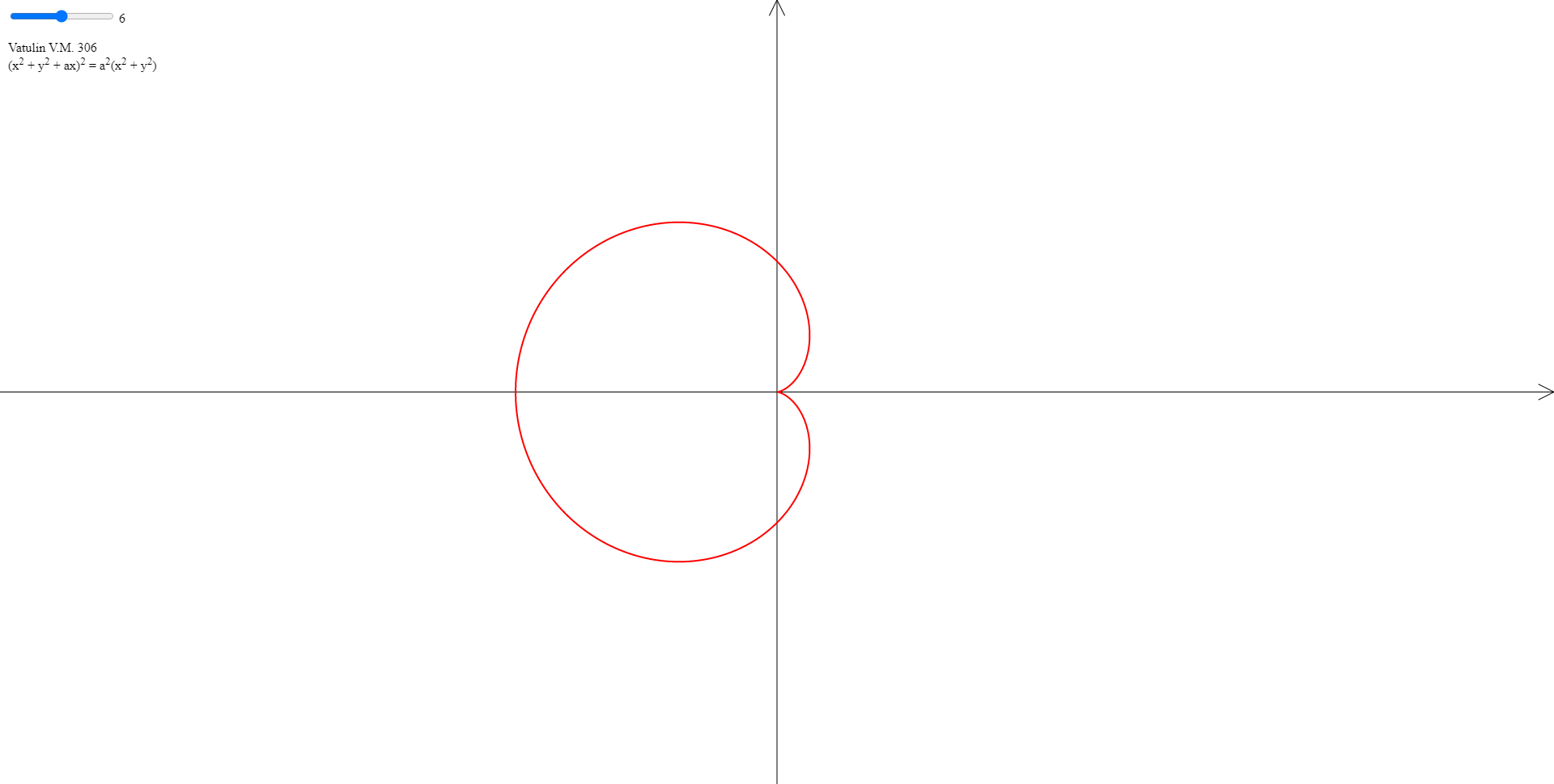


рис. 2 - Параметр равен 6

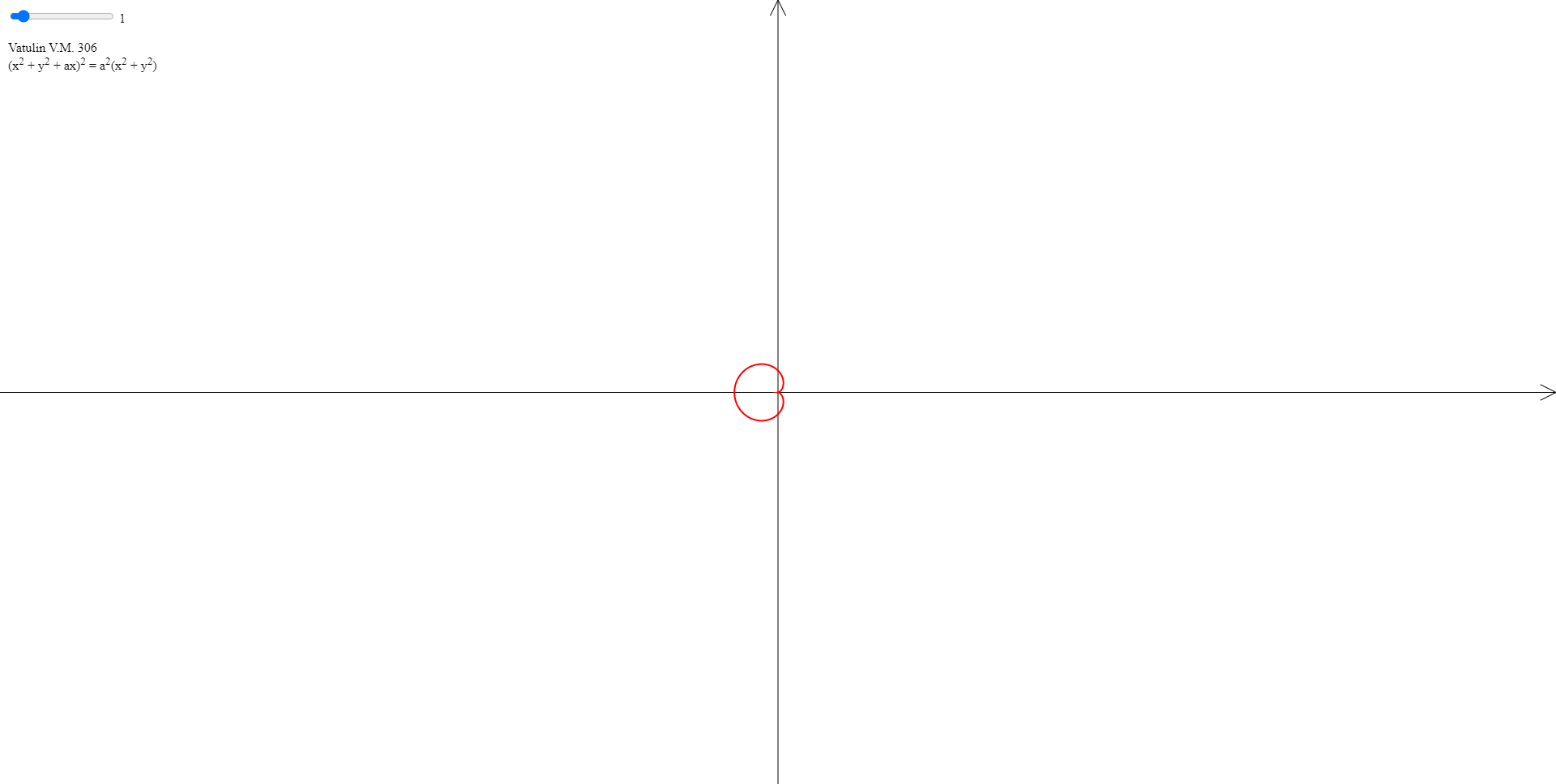


рис. 3 - Параметр равен 1

1. Листинг программы

index.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>1</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<canvas></canvas>

<div id="slider">

<input type="range" min="0.1" max="12" value="6" step="0.1" id="param">

<output id="param-out">6</output>

<br>

<br>

<span>Vatulin V.M. 306</span>

<br>

<span>(x<sup>2</sup> + y<sup>2</sup> + ax)<sup>2</sup> = a<sup>2</sup>(x<sup>2</sup> + y<sup>2</sup>)</span>

</div>

<script src="./main.js"></script>

</body>

</html>

style.css:

body {

margin: 0px;

width: 100vw;

height: 100vh;

}

#slider {

position: absolute;

top: 10px;

left: 10px;

}

main.js:

'use strict';

function getPoints(a) {

function f(temp1, temp2) {

return ((temp1 + temp2) / 2).toFixed(10) \*\* 0.5;

}

let points = [];

for (let x = -a\*2; x <= a / 2; x = Number((x + 0.01).toFixed(10))) {

const temp1 = a\*\*1.5 \* (a - 4\*x)\*\*0.5;

const temp2 = a\*\*2 - 2\*a\*x - 2\*x\*\*2;

let y = [f(temp1, temp2), -f(temp1, temp2), f(-temp1, temp2), -f(-temp1, temp2)];

if (!Number.isNaN(y[0])) {

points.push([x, y[0]]);

for (let i = 1; i < y.length; ++i) {

if (y[i] !== y[i-1] && !Number.isNaN(y[i])) {

points.push([x, y[i]]);

}

}

}

}

points.sort((a, b) => Math.atan2(a[1], a[0]) - Math.atan2(b[1], b[0]));

return points;

}

function drawAxes(ctx) {

const canvas = ctx.canvas;

ctx.strokeStyle = '#000';

ctx.lineWidth = 1;

ctx.beginPath();

ctx.moveTo(0, -canvas.height / 2);

ctx.lineTo(0, canvas.height / 2);

ctx.moveTo(-canvas.width / 2, 0);

ctx.lineTo(canvas.width / 2, 0);

ctx.moveTo(-canvas.width / 200, canvas.height \* 0.48);

ctx.lineTo(0, canvas.height / 2);

ctx.lineTo(canvas.width / 200, canvas.height \* 0.48);

ctx.moveTo(canvas.width \* 0.49, canvas.height / 100);

ctx.lineTo(canvas.width / 2, 0);

ctx.lineTo(canvas.width \* 0.49, -canvas.height / 100);

ctx.stroke();

}

function drawGraph(ctx, points) {

const canvas = ctx.canvas;

ctx.clearRect(-canvas.width / 2, -canvas.height / 2, canvas.width, canvas.height);

drawAxes(ctx);

ctx.strokeStyle = '#f00';

ctx.lineWidth = 2;

const scale = Math.min(canvas.width / 64, canvas.height / 36);

ctx.beginPath();

ctx.moveTo(points[0][0] \* scale, points[0][1] \* scale);

for (let i = 1; i < points.length; ++i) {

ctx.lineTo(points[i][0] \* scale, points[i][1] \* scale);

}

ctx.closePath();

ctx.stroke();

}

function main() {

const canvas = document.querySelector('canvas');

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

let a = document.querySelector('#param').value;

window.addEventListener('resize', function(event) {

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

ctx.setTransform(1, 0, 0, -1, canvas.width / 2, canvas.height / 2);

drawGraph(ctx, getPoints(a));

});

document.querySelector('#param').addEventListener('input', function() {

a = this.value;

document.querySelector('#slider>output').value = a;

drawGraph(ctx, getPoints(a));

});

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

ctx.setTransform(1, 0, 0, -1, canvas.width / 2, canvas.height / 2);

drawGraph(ctx, getPoints(a));

}

main();

1. Вывод

В ходе выполнения данной лабораторной работы я научился работать с Canvas API в браузере, а также попрактиковал обработку событий (таких как изменение размеров окна и положения слайдера).

ЛИТЕРАТУРА

# Canvas API [Электронный ресурс] URL: https://developer.mozilla.org/en-US/docs/Web/API/Canvas\_API

(дата обращения: 11.12.21)