**Цель курсового проектирования:**

Закрепить на практике приобретённые навыки работы по созданию одностраничных веб-приложений на базе фреймворка *Vue.js* при распараллеливании вычислений, проводимых в трёх различных разделах:

- разделе гипертекстовой разметки,

- разделе сценария выполнения расчётов по поставленной задаче,

- разделе стилевого оформления одностраничного приложения.

**Постановка задачи согласно варианту №20 на курсовой проект по дисциплине «Основы параллельного программирования»:**

«Создание форматируемой области, необходимой для последующего построения графика поверхности со специфическими, но фиксированными углами поворота осей абсцисс и ординат относительно горизонтали на 150 и 30 градусов, соответственно. Форматируемая область должна быть максимально чувствительной по управлению отображением входящих в её состав компонентов.»

**Содержательная часть проекта:**

Решалась задача автоматизации построения координатного пространства (три измерения) в одностраничном браузерном веб-приложении.

Данная задача распалась на следующие этапы:

- формирования и настройки области для размещения векторной графики;

- формирования объектов на области для размещения векторной графики:

- линии осей абсцисс, ординат, аппликат;

- подписи осей и начала координат;

- засечки, соответствующие шагу сетки области для построения графика функциональной зависимости;

- подписи к засечкам, соответствующим шагу сетки;

- линии сетки осей абсцисс, ординат, аппликат;

- формирование панели настроек для интерфейсного взаимодействия пользователя с разработанными функциями и методами.

Для некоторой прямоугольной области для размещения элементов векторной графики *SVG* (*Scalable Vector Graphics* – масштабируемая векторная графика) со значениями длины и высоты 500 и 300 пикселей, соответственно, заданными по умолчанию, выполнено изображение осей координатного пространства *X*, *Y*, *Z* (абсцисс, ординат и аппликат, соответственно). Именно эти величины (500 и 300 пикселей) позволяют обеспечить указанные согласно постановке углы поворота осей. Соотношение сторон при этом составляет 3:5. При необходимости рассматриваемое по умолчанию соотношение можно поменять.

Оси могут отображаться в двух режимах: а) от -∞ до +∞; б) от 0 до +∞. Для чего в панели настроек предусмотрена соответствующая опция.

Для удобства демонстрации получаемых результатов в одностраничном веб-приложении предусмотрена возможность отображения / сокрытия пунктов панели настроек по управлению отображением.

На Рисунке 1 показано исходное состояние интерфейса одностраничного веб-приложения.

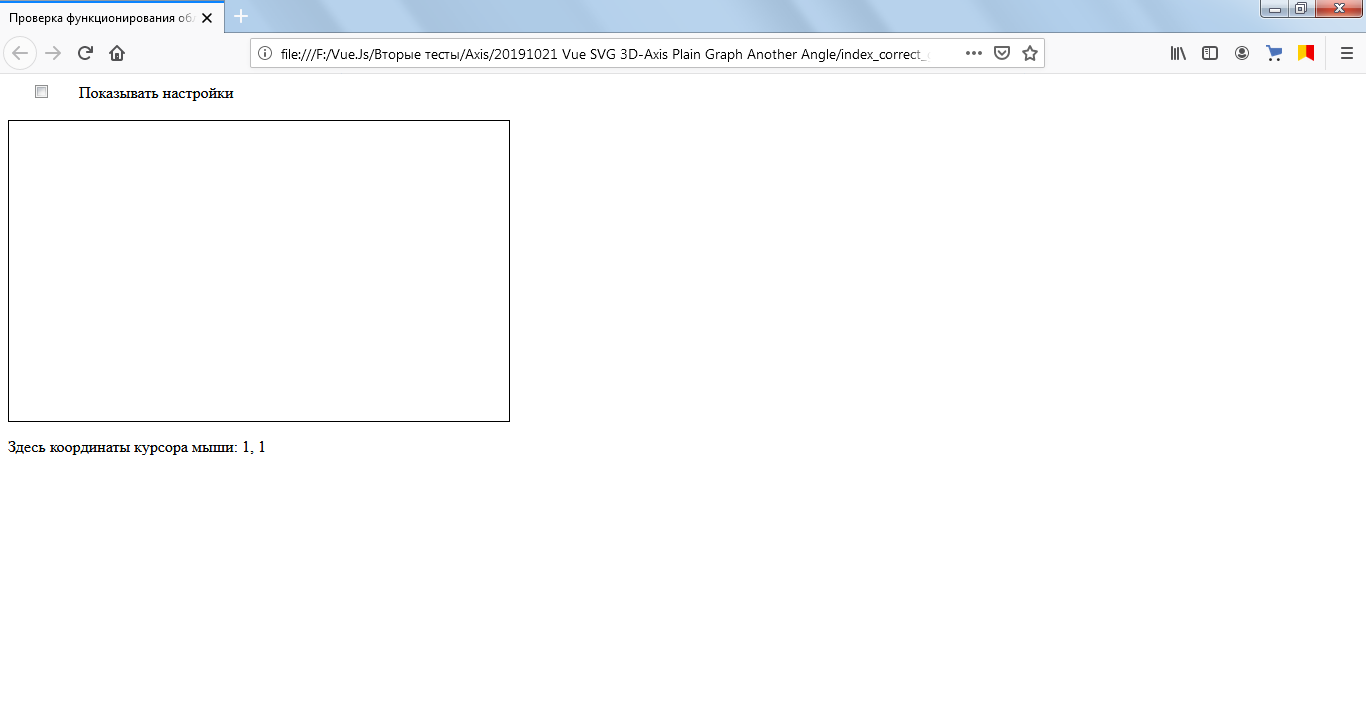


Рисунок 1

К этому состоянию интерфейс можно свести и вручную посредством снятия всех флажков, как показано на Рисунке 2.

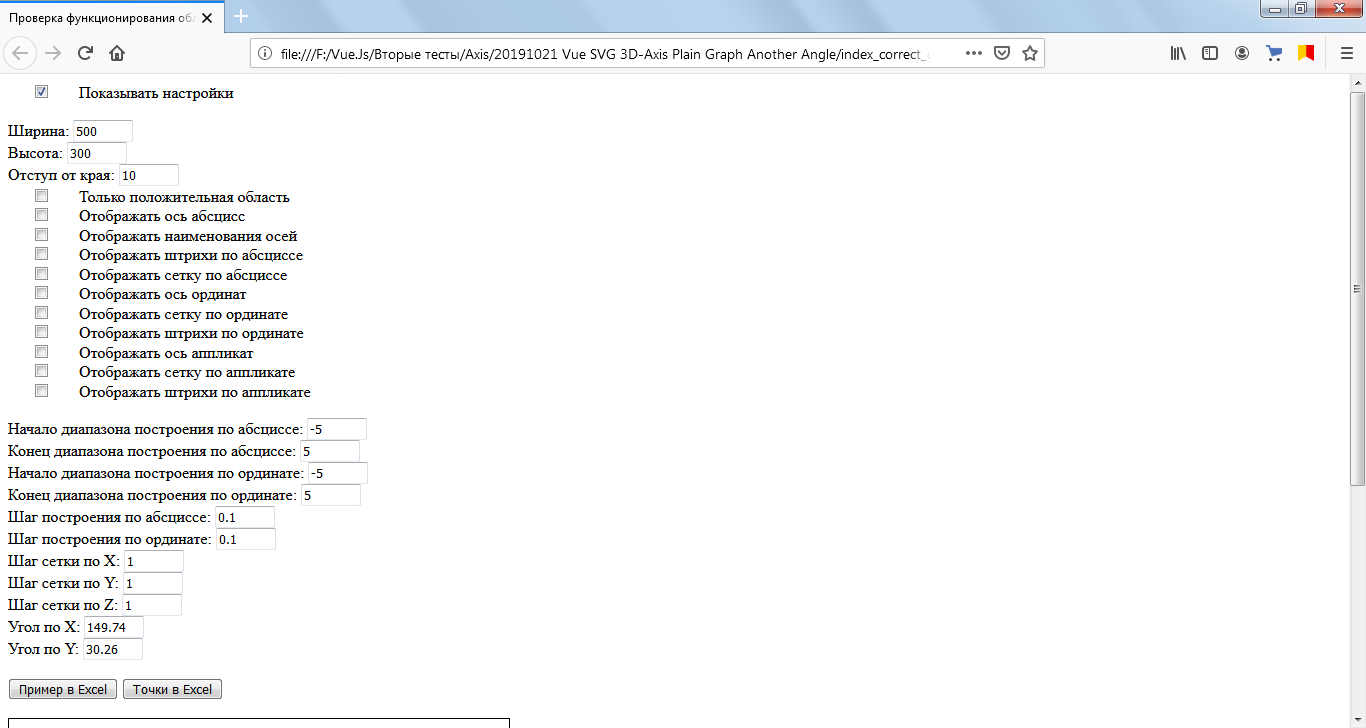


Рисунок 2

Далее на Рисунках 3-5 и 6-8, соответственно, показаны результаты отображения линий координатных осей и состояния панели настроек для каждого результата.

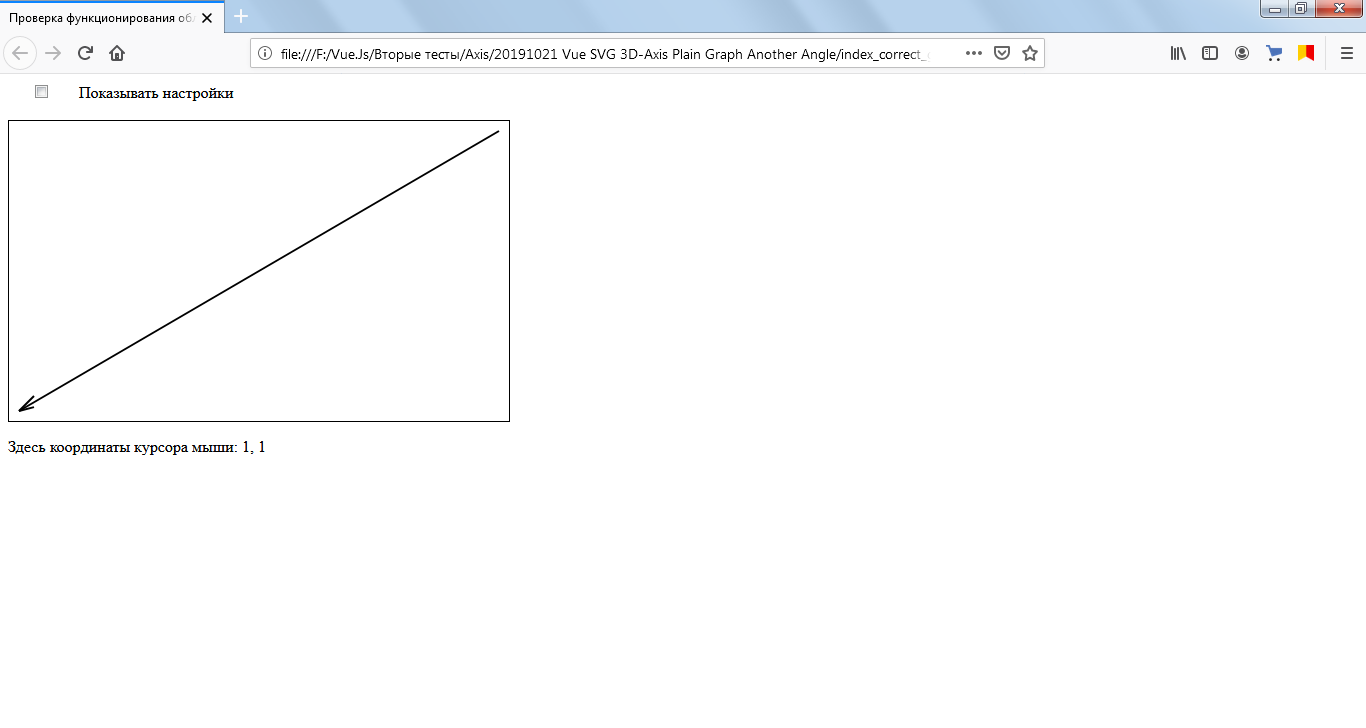


Рисунок 3

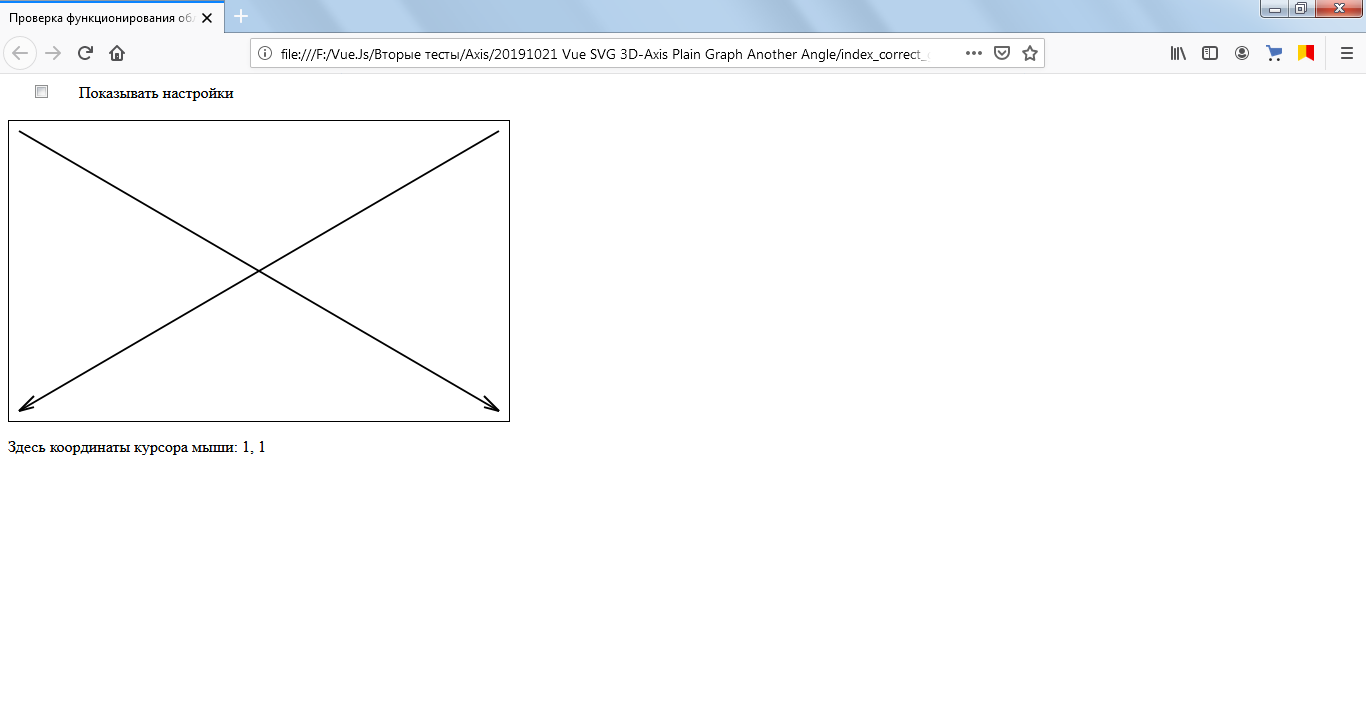


Рисунок 4

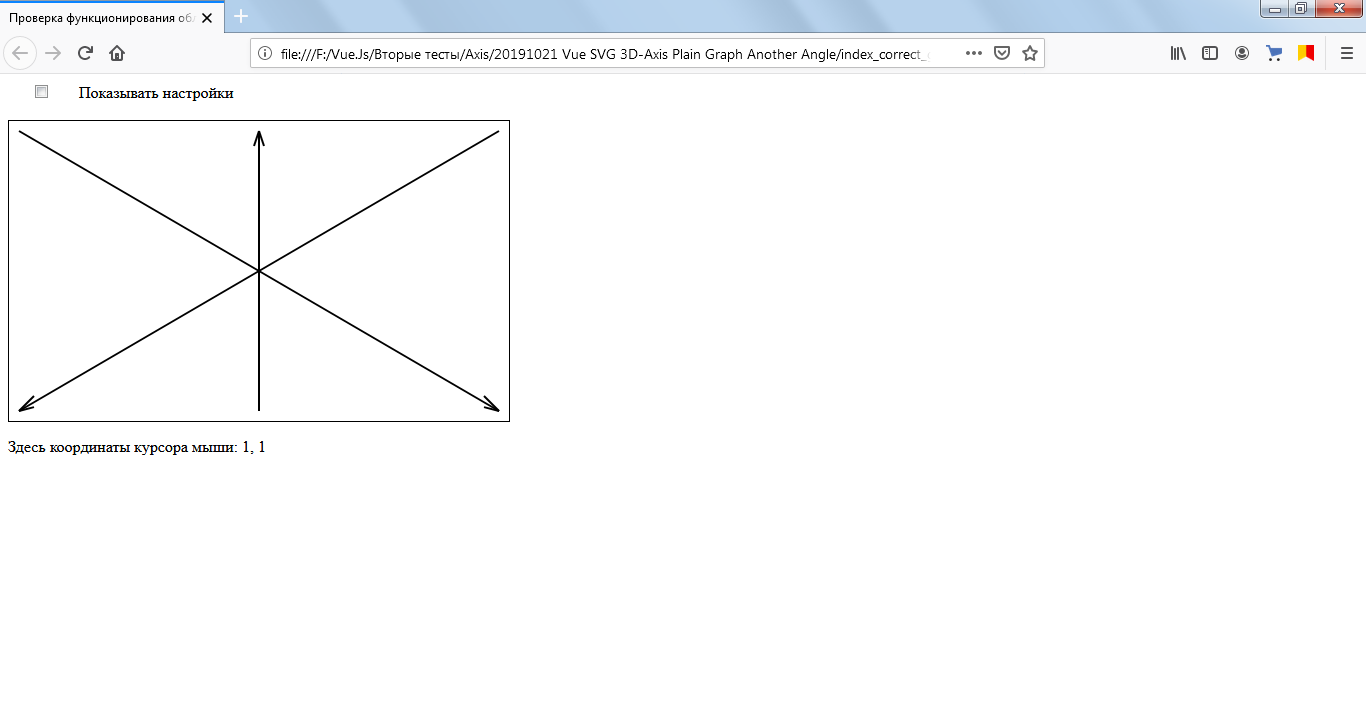


Рисунок 5

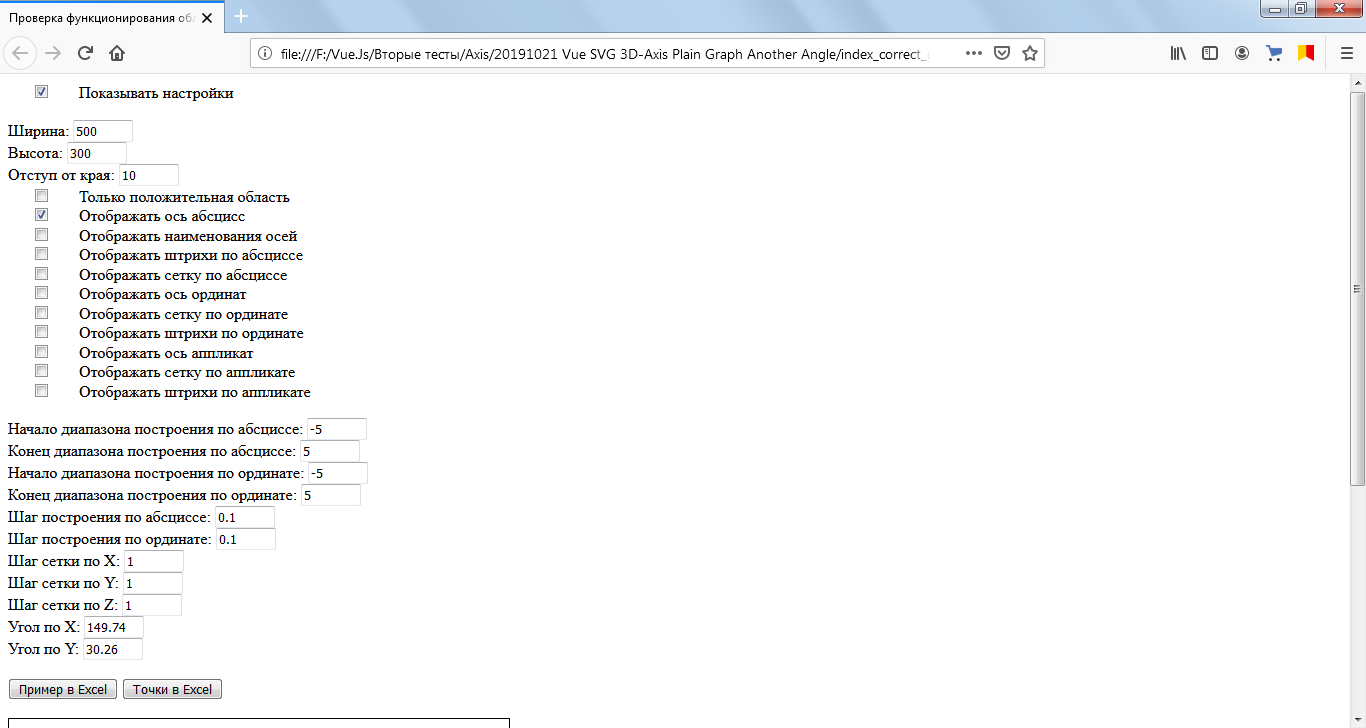


Рисунок 6

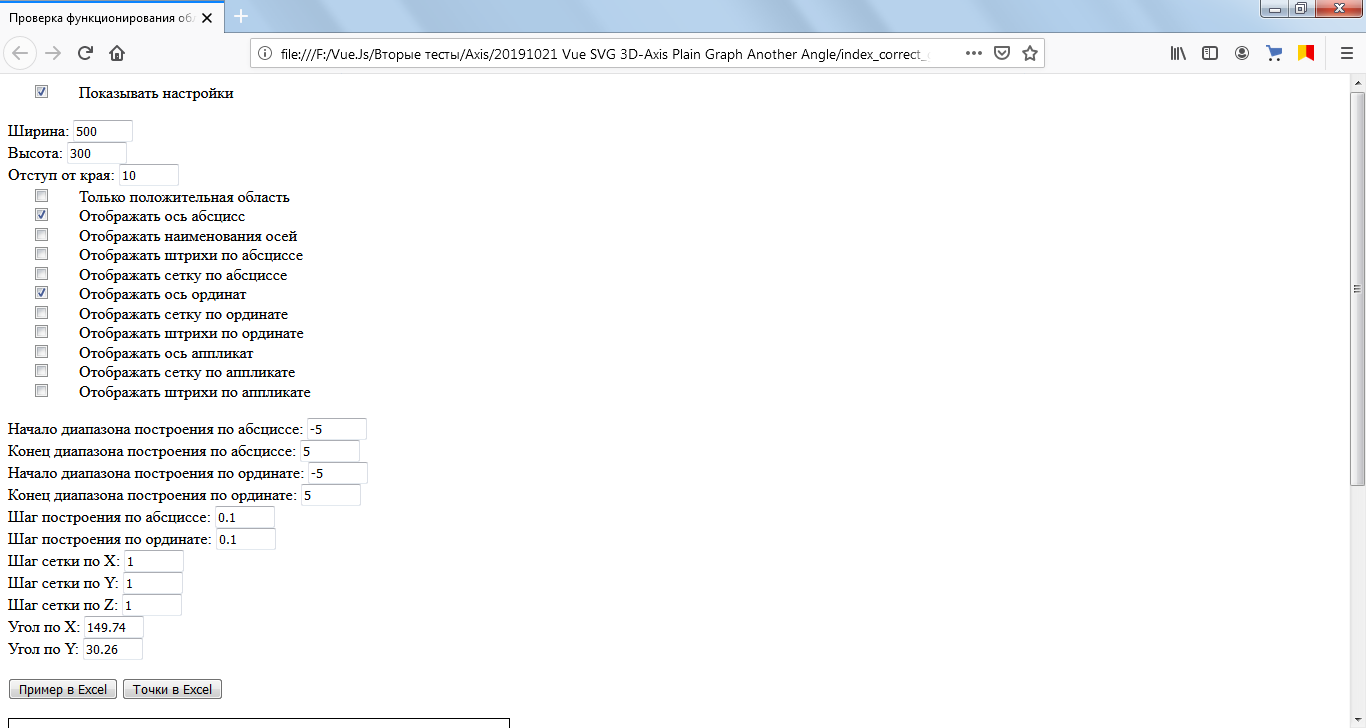


Рисунок 7

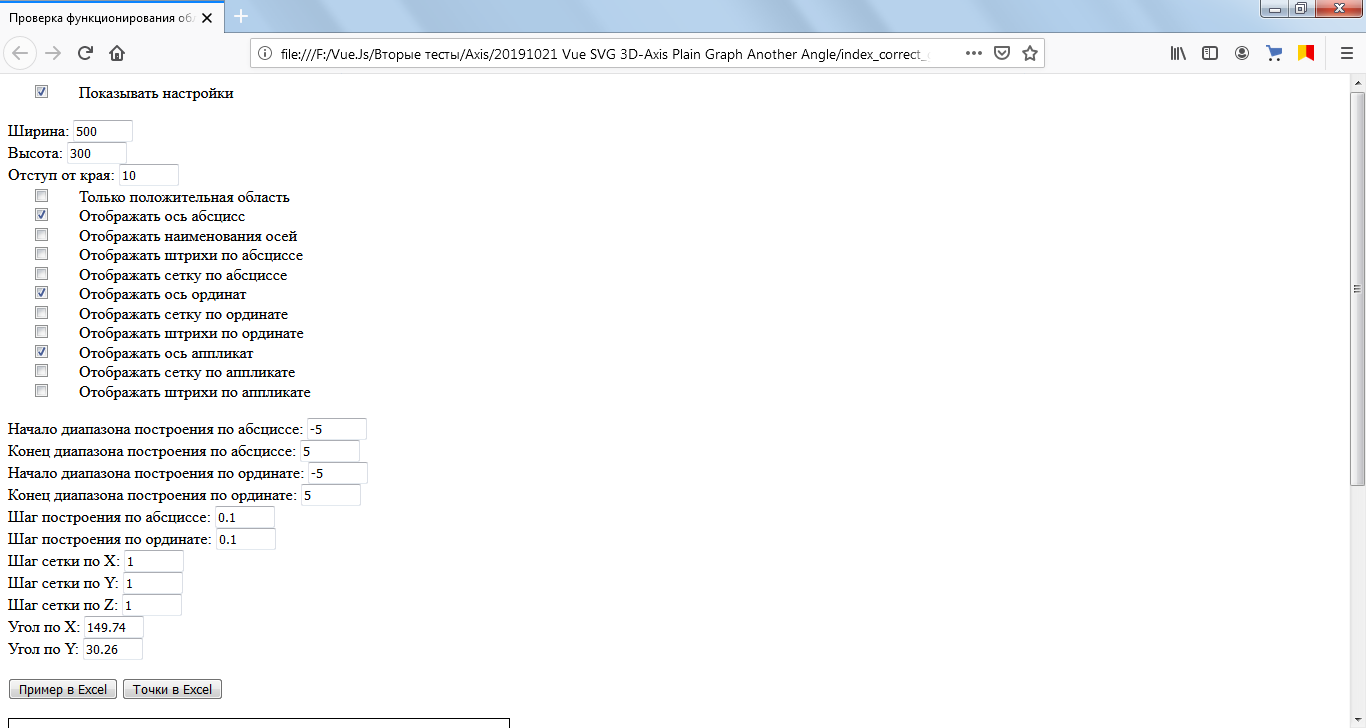


Рисунок 8

Отдельно организован вывод текстовой информации (Рисунки 9-10).

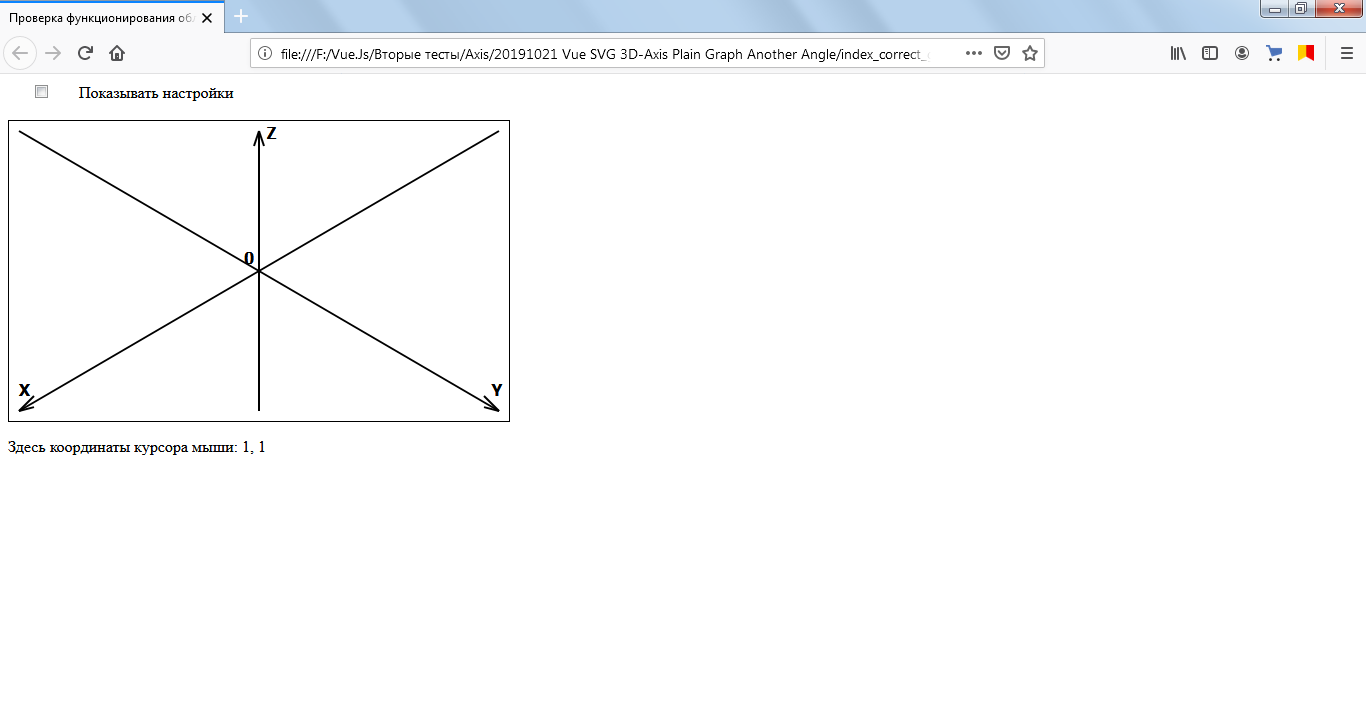


Рисунок 9

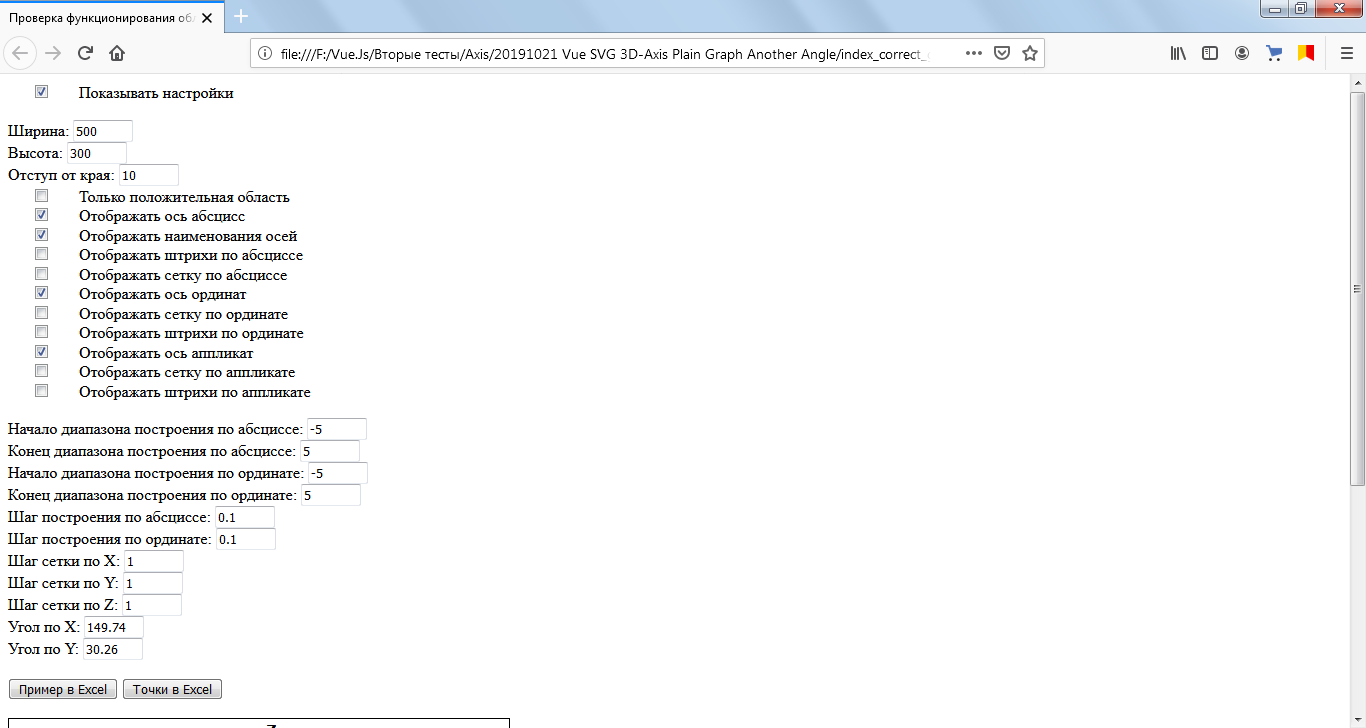


Рисунок 10

По каждой оси формируются свои засечки (Рисунки 11-16).

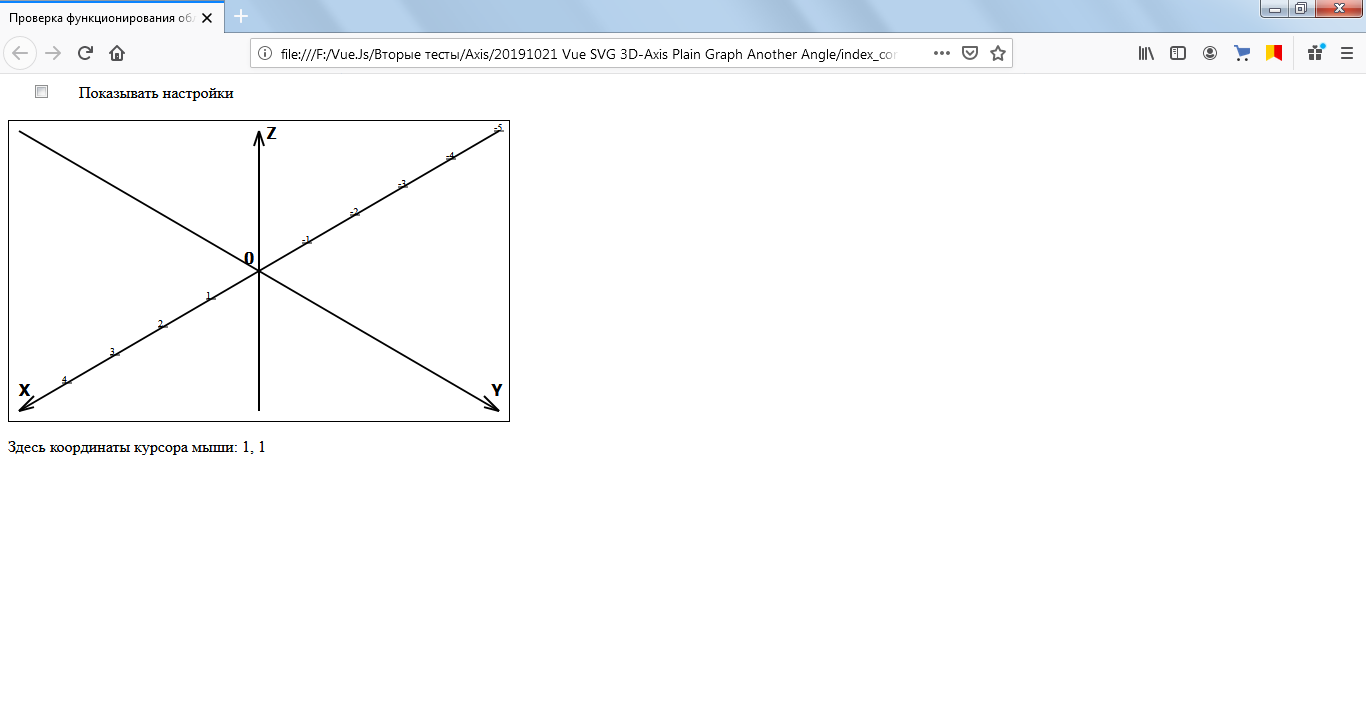


Рисунок 11

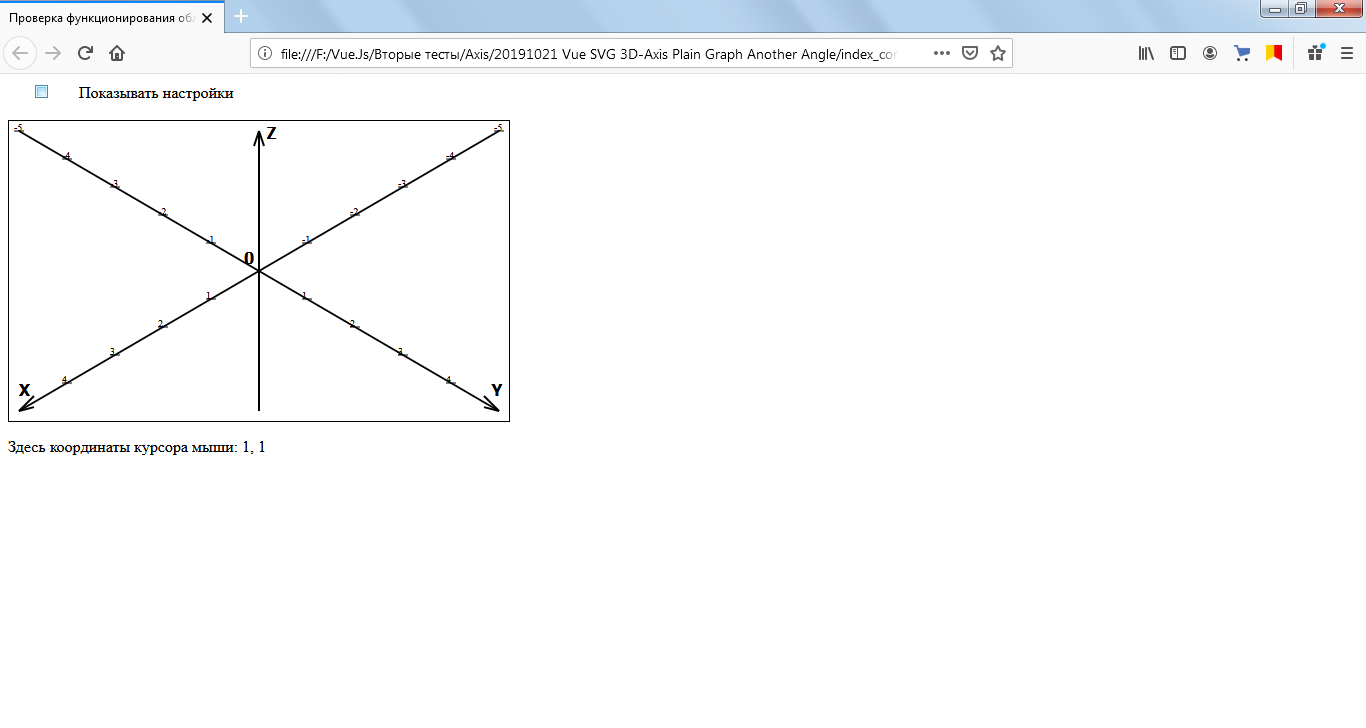


Рисунок 12

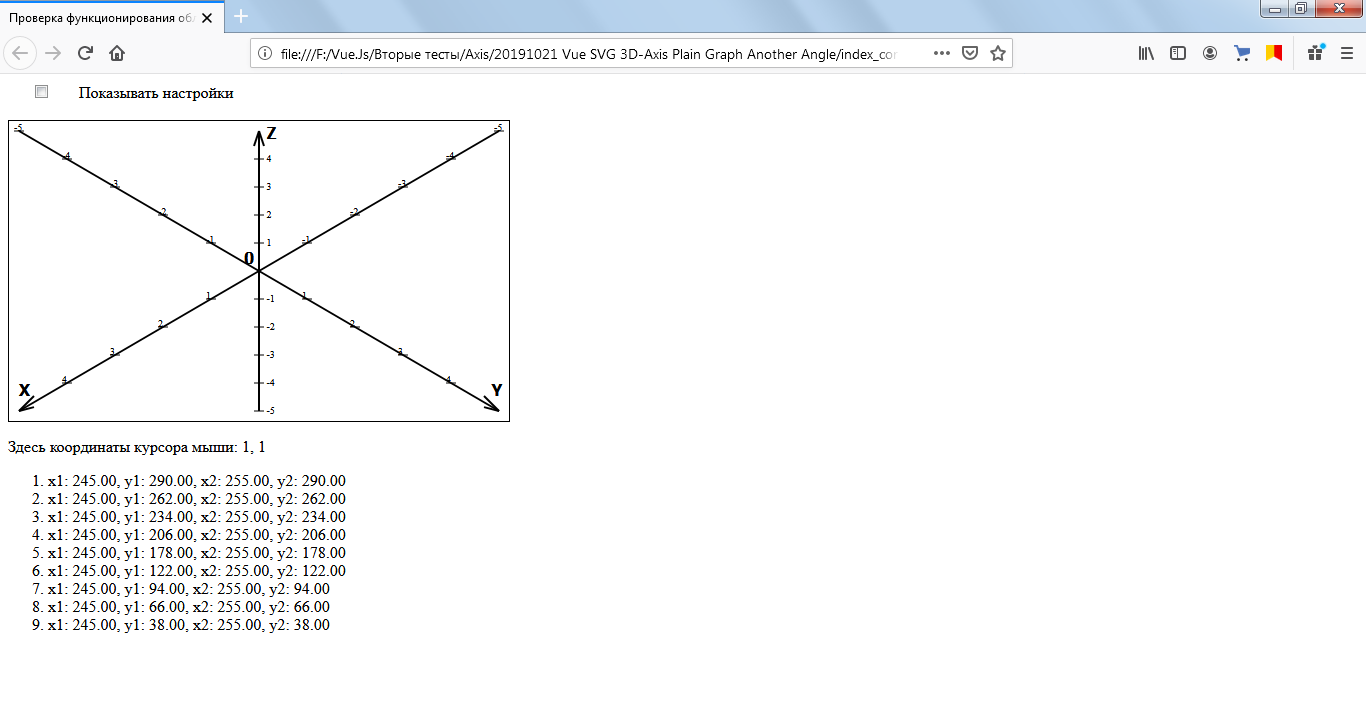


Рисунок 13

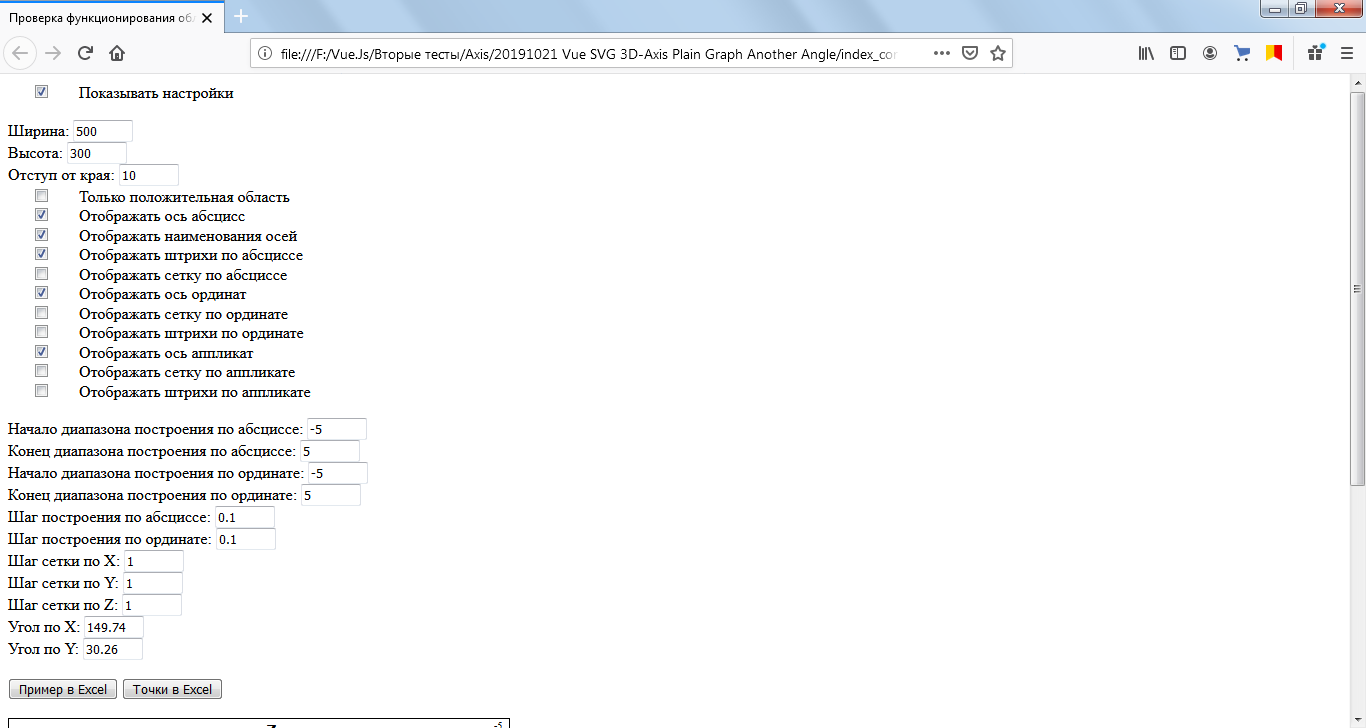


Рисунок 14

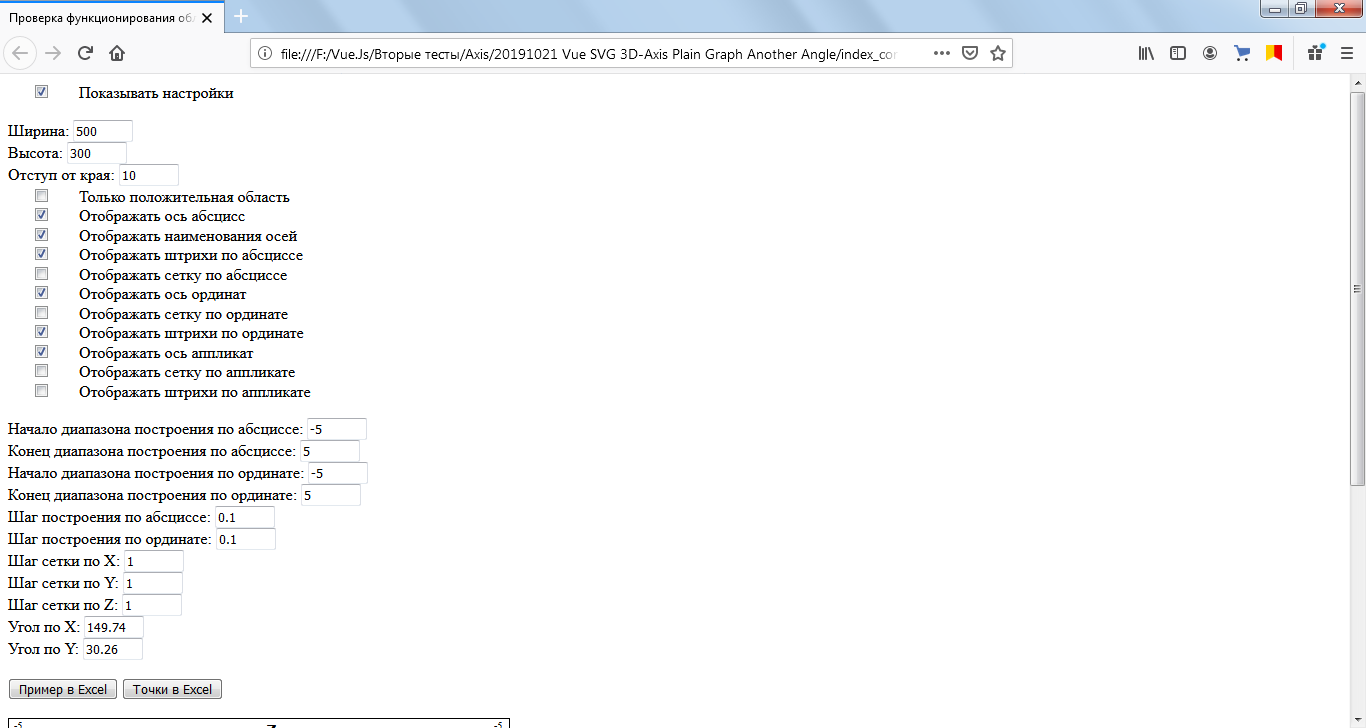


Рисунок 15

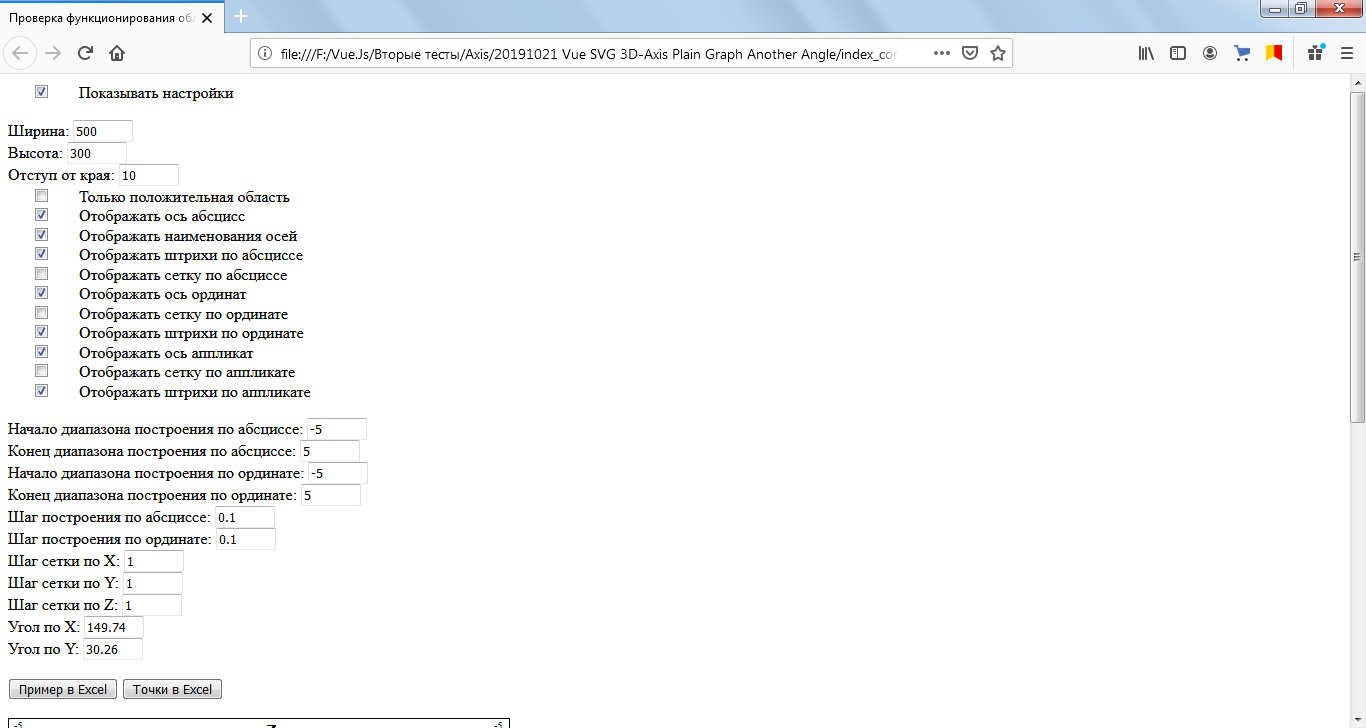


Рисунок 16

Аналогичным образом по каждой оси формируются свои линии сетки (Рисунки 17-22).

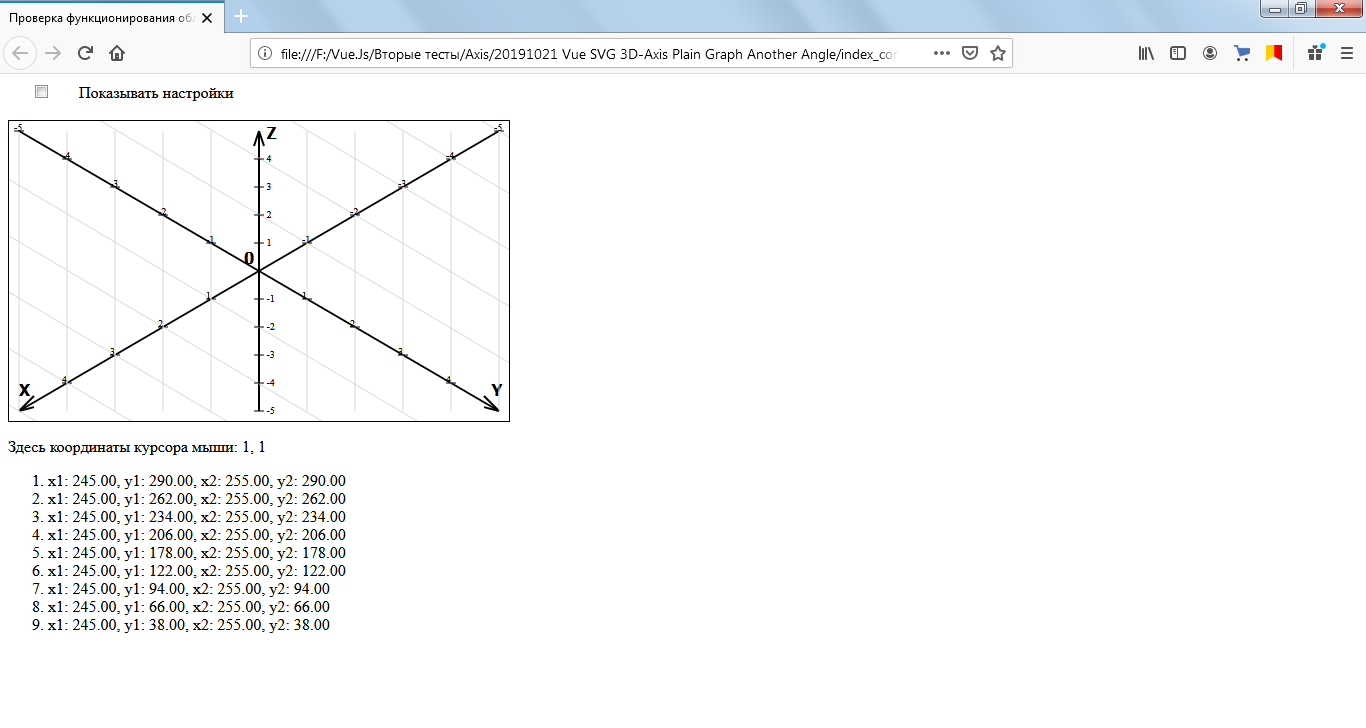


Рисунок 17

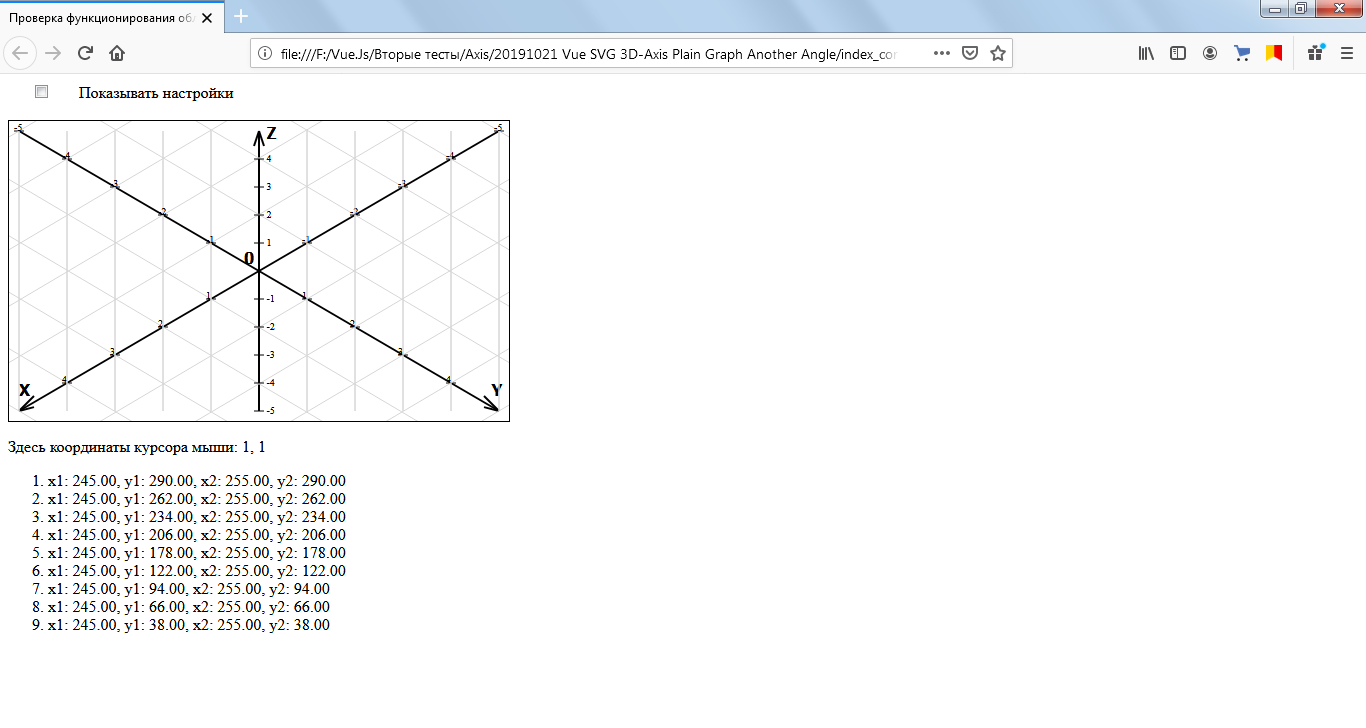


Рисунок 18

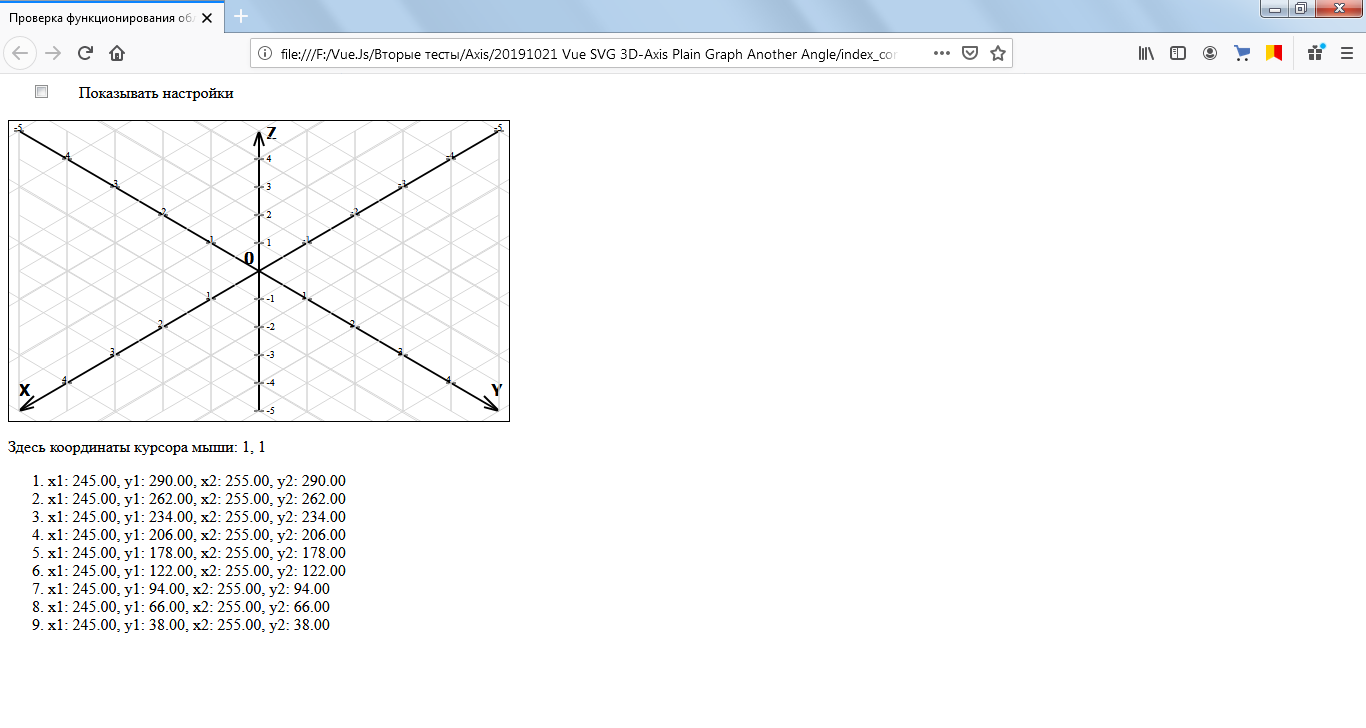


Рисунок 19

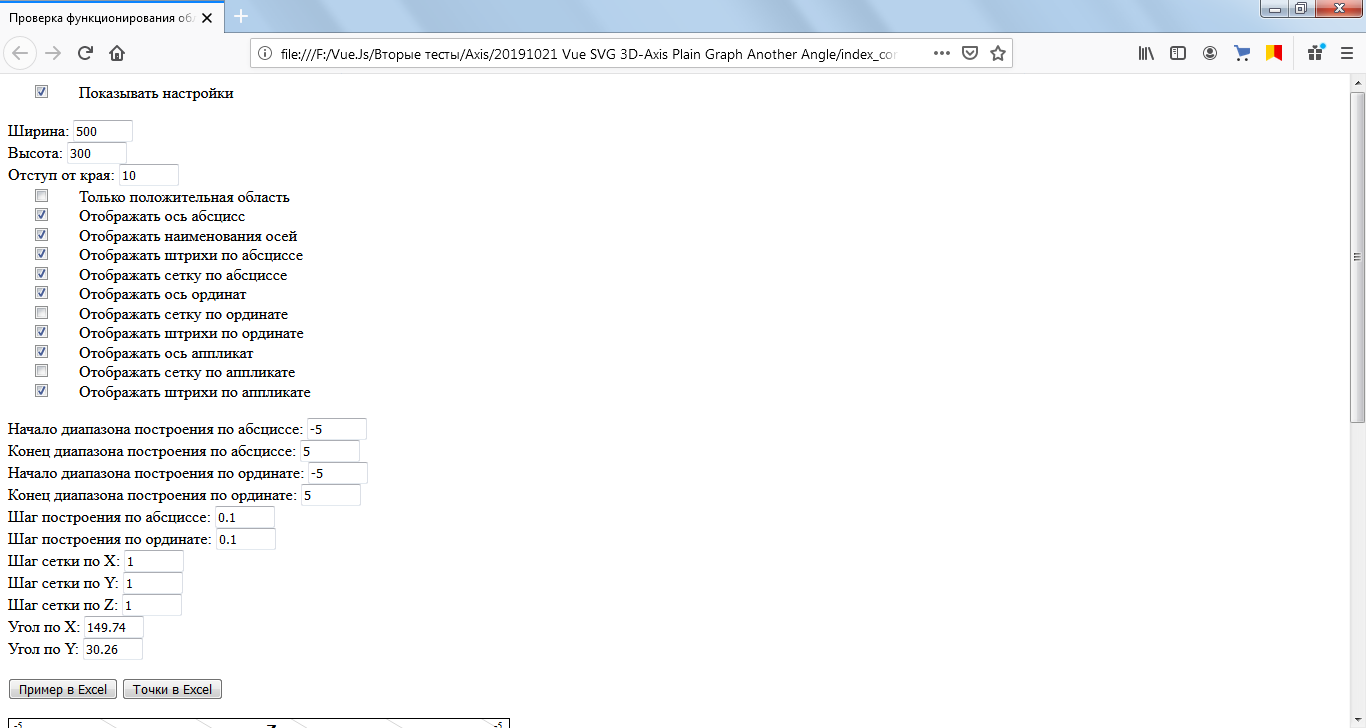


Рисунок 20

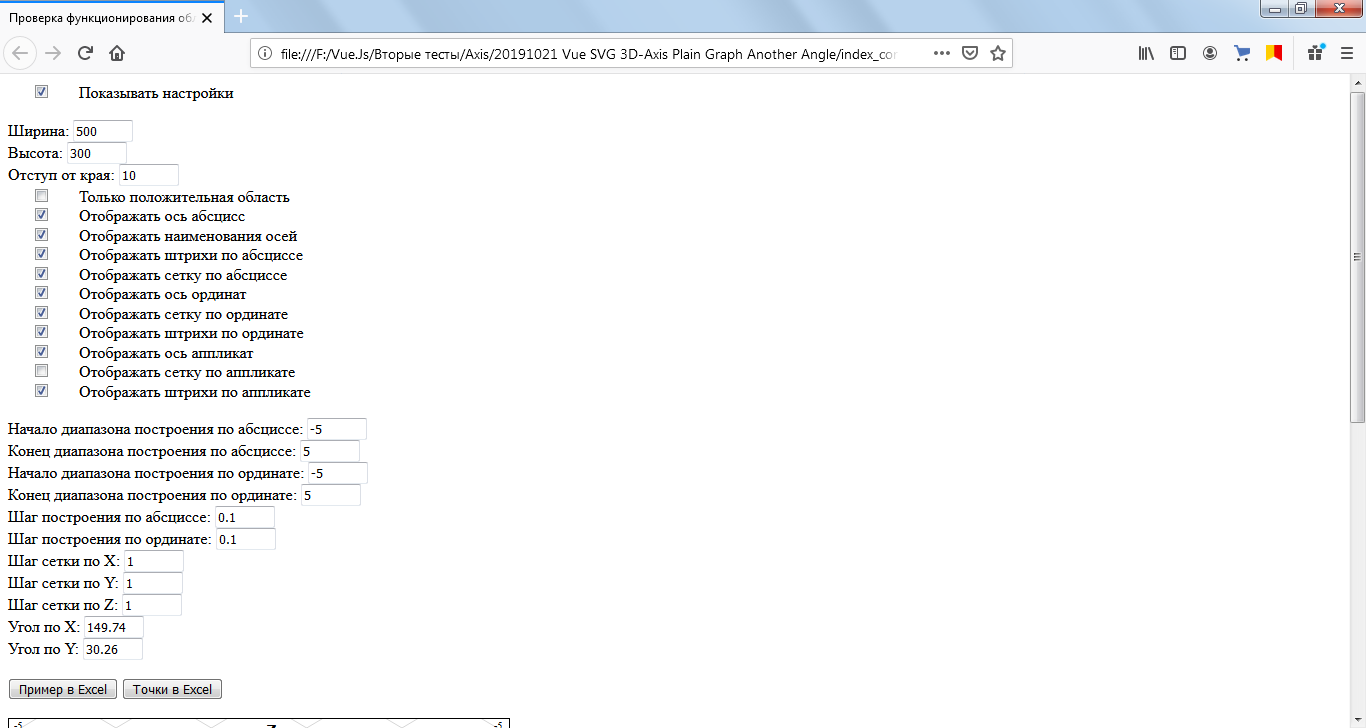


Рисунок 21

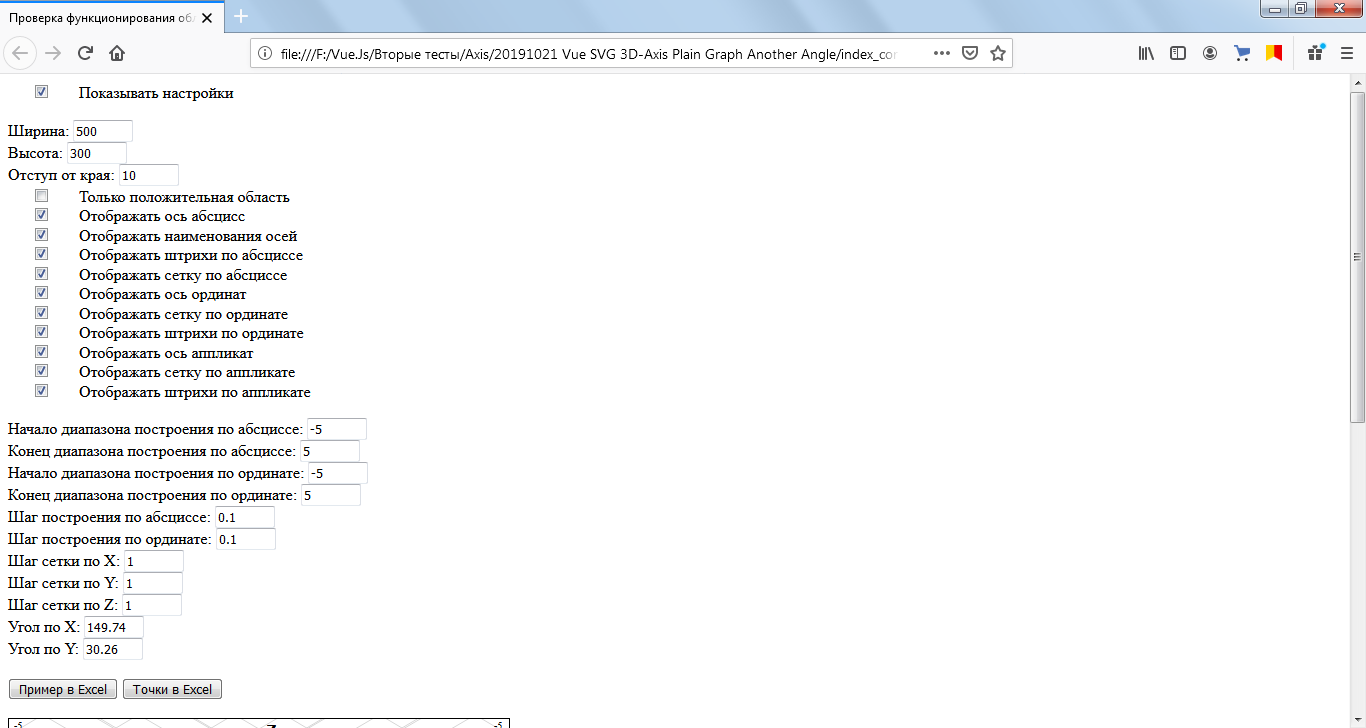


Рисунок 22

Для отображения графика поверхности только в положительной области предусмотрен особый режим, результат работы которого представлен на Рисунках 23-24.

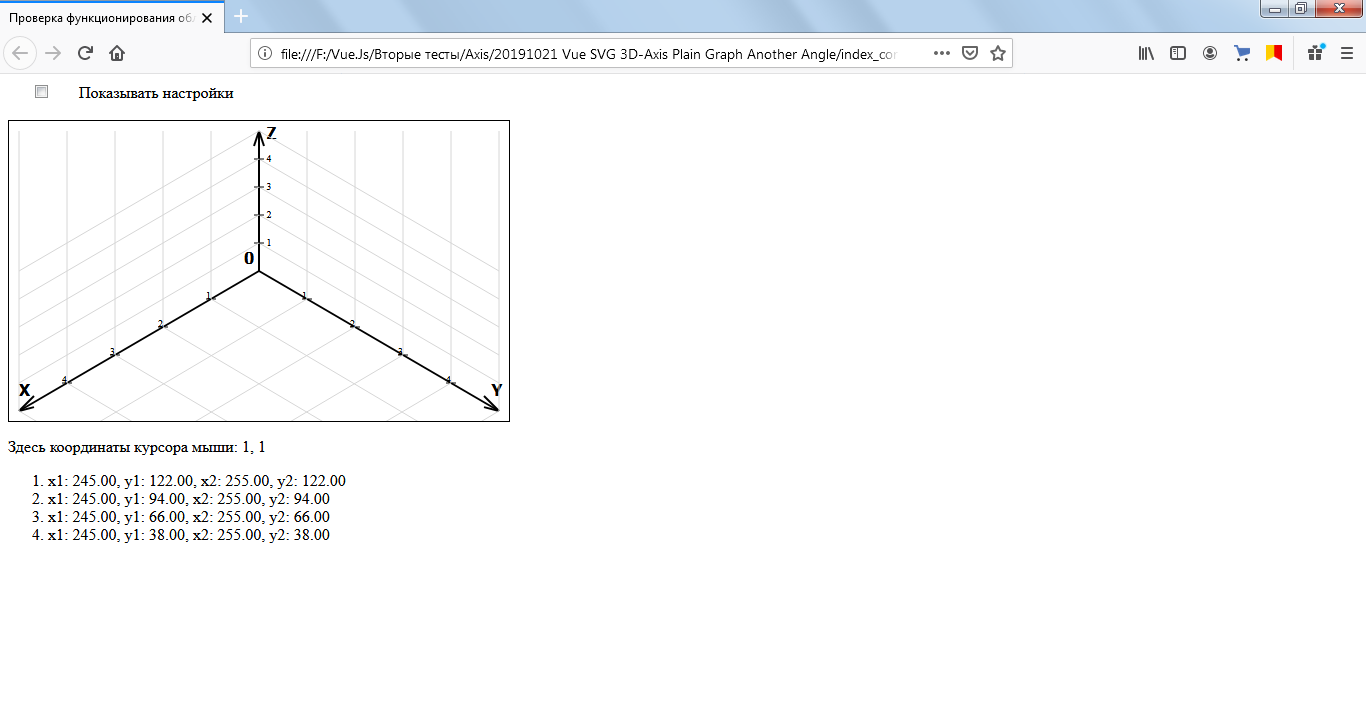


Рисунок 23

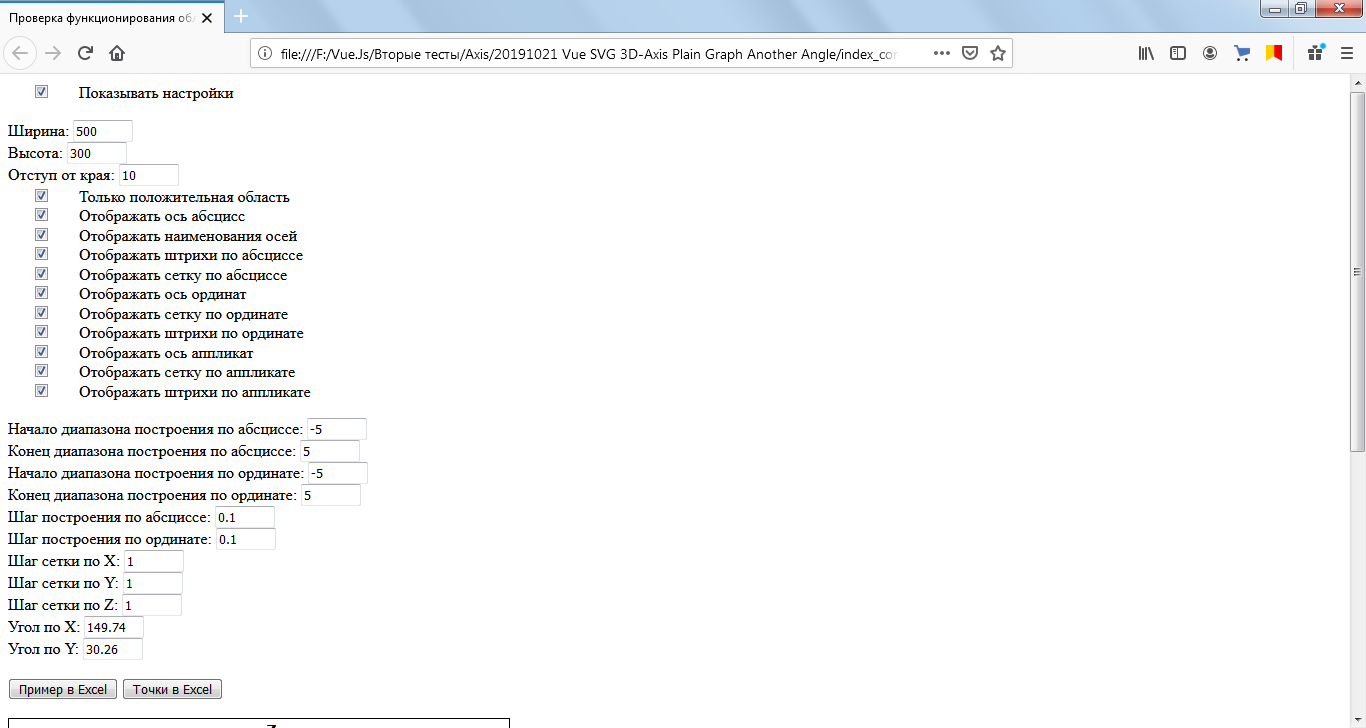


Рисунок 24

**Листинг гипертекстовой части одностраничного веб-приложения:**

<HTML>

<TITLE>Проверка функционирования области для рисования</TITLE>

<BODY>

<SCRIPT SRC="Vue.js"></SCRIPT>

<SCRIPT SRC="shim.min.js"></SCRIPT>

<SCRIPT SRC="xlsx.full.min.js"></SCRIPT>

<div id="draw">

<input id="fOptions" type="checkbox" v-model="flgOptions">

<label for="fOptions">Показывать настройки</label>

<br>

<template v-if="flgOptions">

<br>

Ширина: <input v-model="wi" @change="countPoints(false)">

<br>

Высота: <input v-model="hi" @change="countPoints(false)">

<br>

Отступ от края:

<input v-model="shift" @change="countPoints(false)">

<br>

<input id="fPositive"

type="checkbox"

v-model="flgPositive"

@change="countPoints(false)">

<label for="fPositive">Только положительная область</label> <br>

<input id="fXaxis"

type="checkbox"

v-model="flgXaxis"

@change="countPoints(false)">

<label for="fXaxis">Отображать ось абсцисс</label>

<br>

<input id="fTxts"

type="checkbox"

v-model="flgTxts"

@change="countPoints(false)">

<label for="fTxts">Отображать наименования осей</label> <br>

<input id="fXstrokes"

type="checkbox"

v-model="flgXstrokes"

@change="countPoints(false)">

<label for="fXstrokes">Отображать штрихи по абсциссе</label>

<br>

<input id="fXgrid"

type="checkbox"

v-model="flgXgrid"

@change="countPoints(false)">

<label for="fXgrid">Отображать сетку по абсциссе</label> <br>

<input id="fYaxis"

type="checkbox"

v-model="flgYaxis"

@change="countPoints(false)">

<label for="fYaxis">Отображать ось ординат</label>

<br>

<input id="fYgrid"

type="checkbox"

v-model="flgYgrid"

@change="countPoints(false)">

<label for="fYgrid">Отображать сетку по ординате</label> <br>

<input id="fYstrokes"

type="checkbox"

v-model="flgYstrokes"

@change="countPoints(false)">

<label for="fYstrokes">Отображать штрихи по ординате</label>

<br>

<input id="fZaxis"

type="checkbox"

v-model="flgZaxis"

@change="countPoints(false)">

<label for="fZaxis">Отображать ось аппликат</label> <br>

<input id="fZgrid"

type="checkbox"

v-model="flgZgrid"

@change="countPoints(false)">

<label for="fZgrid">Отображать сетку по аппликате</label> <br>

<input id="fZstrokes"

type="checkbox"

v-model="flgZstrokes"

@change="countPoints(false)">

<label for="fZstrokes">Отображать штрихи по аппликате</label> <br>

<br>

Начало диапазона построения по абсциссе:

<input v-model="startX" @change="countPoints(false)">

<br>

Конец диапазона построения по абсциссе:

<input v-model="endX" @change="countPoints(false)">

<br>

Начало диапазона построения по ординате:

<input v-model="startY" @change="countPoints(false)">

<br>

Конец диапазона построения по ординате:

<input v-model="endY" @change="countPoints(false)">

<input type="hidden"

v-model="startZ"

@change="countPoints(false)">

<input type="hidden"

v-model="endZ"

@change="countPoints(false)">

<br>

Шаг построения по абсциссе:

<input v-model="stepX" @change="countPoints(false)">

<br>

Шаг построения по ординате:

<input v-model="stepY" @change="countPoints(false)">

<br>

Шаг сетки по X:

<input v-model="axisStepX" @change="countPoints(false)">

<br>

Шаг сетки по Y:

<input v-model="axisStepY" @change="countPoints(false)">

<br>

Шаг сетки по Z:

<input v-model="axisStepZ" @change="countPoints(false)">

<br>

Угол по X: <input v-model="angleXShow">

<br>

Угол по Y: <input v-model="angleYShow">

<br>

<br>

<button id="btnExcelExample"

@click="writeExampleExcel">Пример в Excel

</button>

<button id="btnExcelPoints"

@click="writePointsExcel">Точки в Excel

</button>

<br>

</template>

<br>

<!--@mousedown.prevent="rotate" @mouseup.prevent="previous"-->

<svg

@click.prevent="incr"

@mousedown.prevent="printXY"

@mouseup.prevent="printLost"

:width="wi"

:height="hi">

<!-- линия оси абсцисс -->

<!-- цвет чёрный #000000 -->

<line v-for="x in axisX"

stroke="#000000"

stroke-width="2"

:x1="x.x1"

:y1="x.y1"

:x2="x.x2"

:y2="x.y2"/>

<!-- линия оси ординат -->

<!-- цвет чёрный #000000 -->

<line v-for="y in axisY"

stroke="#000000"

stroke-width="2"

:x1="y.x1"

:y1="y.y1"

:x2="y.x2"

:y2="y.y2"/>

<!-- линия оси аппликат -->

<!-- цвет чёрный #000000 -->

<line v-for="z in axisZ"

stroke="#000000"

stroke-width="2"

:x1="z.x1"

:y1="z.y1"

:x2="z.x2"

:y2="z.y2"/>

<!-- наименование осей -->

<!-- цвет чёрный #000000 -->

<text v-for="t in txts"

fill="#000000"

font-weight="bold"

font-family="Tahoma"

:x="t.x"

:y="t.y">{{ t.name }}</text>

<!-- нанесение штрихов по абсциссе -->

<!-- цвет чёрный #000000 -->

<line v-for="s in strokesX"

stroke="#000000"

:x1="s.x1"

:y1="s.y1"

:x2="s.x2"

:y2="s.y2"/>

<!-- нанесение сетки по абсциссе -->

<!-- как в MathCAD цвет #00ff00 - зелёный -->

<!-- пользовательское предпочтение #d6d6d6 - серый -->

<line v-for="g in gridX"

stroke="#d6d6d6"

:x1="g.x1"

:y1="g.y1"

:x2="g.x2"

:y2="g.y2"/>

<!-- нанесение текста по абсциссе -->

<text v-for="s in strokesX"

font-weight="normal"

font-size="10"

:x="s.txtX"

:y="s.txtY">{{s.txt}}</text>

<!-- нанесение штрихов по ординате -->

<!-- цвет чёрный #000000 -->

<line v-for="s in strokesY"

stroke="#000000"

:x1="s.x1"

:y1="s.y1"

:x2="s.x2"

:y2="s.y2"/>

<!-- нанесение сетки по ординате -->

<!-- как в MathCAD цвет #00ff00 - зелёный -->

<!-- пользовательское предпочтение #d6d6d6 - серый -->

<line v-for="g in gridY"

stroke="#d6d6d6"

:x1="g.x1"

:y1="g.y1"

:x2="g.x2"

:y2="g.y2"/>

<!-- нанесение текста по ординате -->

<text v-for="s in strokesY"

font-weight="normal"

font-size="10"

:x="s.txtX"

:y="s.txtY">{{s.txt}}</text>

<!-- нанесение штрихов по аппликате -->

<line v-for="s in strokesZ"

stroke="#000000"

:x1="s.x1"

:y1="s.y1"

:x2="s.x2"

:y2="s.y2"/>

<!-- нанесение сетки по аппликате -->

<!-- как в MathCAD цвет #00ff00 - зелёный -->

<!-- пользовательское предпочтение #d6d6d6 - серый -->

<line v-for="g in gridZ"

stroke="#d6d6d6"

:x1="g.x1"

:y1="g.y1"

:x2="g.x2"

:y2="g.y2"/>

<!-- нанесение текста по ординате -->

<text v-for="s in strokesZ"

font-weight="normal"

font-size="10"

:x="s.txtX"

:y="s.txtY">{{s.txt}}</text>

<!-- нанесение точек функциональной зависимости -->

<!-- как в MathCAD цвет #ff0000 - красный -->

<!-- пользовательское предпочтение #9f22ff - фиолетовый -->

<circle v-for="p in pts"

stroke="#9f22ff"

stroke-width="5"

:cx="p.x1"

:cy="p.y1"

r="2"/>

</svg>

<p>Здесь координаты курсора мыши: {{mouseX}}, {{mouseY}}</p>

<ol>

<li v-for="s in strokesZ">

x1: {{ s.x1.toFixed(2) }}, y1: {{ s.y1.toFixed(2) }},

x2: {{ s.x2.toFixed(2) }}, y2: {{ s.y2.toFixed(2) }}

</li>

</ol>

</div>

**[ЗДЕСЬ РАЗМЕЩЕНА СЦЕНАРНАЯ ЧАСТЬ]**

**[ЗДЕСЬ РАЗМЕЩЕНА СТИЛЕВАЯ ЧАСТЬ]**

</BODY>

</HTML>

**Листинг сценарной части одностраничного веб-приложения:**

<SCRIPT>

var myView = new Vue(

{

el: "#draw",

data:

{

startX: -5,

endX: 5,

startY: -5,

endY: 5,

startZ: -5,

endZ: 5,

axisStepX: 1,

axisStepY: 1,

axisStepZ: 1,

stepX: 0.1,

stepY: 0.1,

wi: 500,

hi: 300,

shift: 10,

arLen: 15,

arAng: 5,

flgOptions: true,

flgPositive: false,

flgTxts: true,

flgXaxis: true,

flgXstrokes: true,

flgXgrid: false,

flgYaxis: true,

flgYstrokes: true,

flgYgrid: false,

flgZaxis: true,

flgZstrokes: true,

flgZgrid: true,

angleXShow: 0,

angleYShow: 0,

mouseX: 1,

mouseY: 1,

vals: [],

lines: [],

pts:[],

axisX:[],

axisY:[],

axisZ:[],

txts:[],

strokesX:[],

strokesY:[],

strokesZ:[],

gridX:[],

gridY:[],

gridZ:[],

},

computed:

{

StartPosXforX:function()

{

if (!this.flgPositive)

return Number(this.wi) - Number(this.shift);

else

return Number(this.wi) / 2;

},

maxZ:

{

get: function()

{

return this.endZ;

},

set: function(newVal)

{

this.endZ = newVal;

}

},

minZ:

{

get: function()

{

return this.startZ;

},

set: function(newVal)

{

this.startZ = newVal;

}

},

StartPosYforX:function()

{

if (!this.flgPositive)

return Number(this.shift);

else

return Number(this.hi) / 2;

},

StartRangeForX:function()

{

if (!this.flgPositive)

return Number(this.startX);

else

return 0;

},

EndRangeForX:function()

{

return Number(this.endX);

},

EndPosXforX:function()

{

return Number(this.shift);

},

EndPosYforX:function()

{

return Number(this.hi) - Number(this.shift);

},

scaleHorizontalX:function()

{

var Xb = this.EndPosXforX;

var Xa = this.StartPosXforX;

return (Xb - Xa) / ( this.EndRangeForX - this.StartRangeForX );

},

angleXToHorizontal:function()

{

var ang = Math.acos(this.scaleHorizontalX / this.scaleX);

this.angleXShow = ((ang \* 180) / Math.PI).toFixed(2);

return ang;

},

scaleX:function()

{

//AB = SQRT( (Xb - Xa)^2 + (Yb - Ya)^2 )

var Xa = this.StartPosXforX;

var Ya = this.StartPosYforX;

var Xb = this.EndPosXforX;

var Yb = this.EndPosYforX;

return (Math.sqrt(Math.pow( Xb - Xa, 2) + Math.pow(Yb - Ya, 2)) /

( this.EndRangeForX - this.StartRangeForX ));

},

StartPosXforY:function()

{

if (!this.flgPositive)

return Number(this.shift);

else

return Number(this.wi) / 2;

},

StartPosYforY:function()

{

if (!this.flgPositive)

return Number(this.shift);

else

return Number(this.hi) / 2;

},

StartRangeForY:function()

{

if (!this.flgPositive)

return Number(this.startY);

else

return 0;

},

EndRangeForY:function()

{

return Number(this.endY);

},

StartRangeForZ:function()

{

return Number(this.startZ);

},

EndRangeForZ:function()

{

return Number(this.endZ);

},

EndPosXforY:function()

{

return Number(this.wi) - Number(this.shift);

},

EndPosYforY:function()

{

return Number(this.hi) - Number(this.shift);

},

StartPosXforZ:function()

{

return Number(this.wi) / 2;

},

StartPosYforZ:function()

{

if (!this.flgPositive)

return Number(this.hi) - Number(this.shift);

else

return Number(this.hi) / 2;

},

EndPosXforZ:function()

{

return Number(this.wi) / 2;

},

EndPosYforZ:function()

{

return Number(this.shift);

},

scaleHorizontalY:function()

{

var Xb = this.EndPosXforY;

var Xa = this.StartPosXforY

return (Xb - Xa) / ( this.EndRangeForY - this.StartRangeForY );

},

angleYToHorizontal:function()

{

var ang = Math.acos(this.scaleHorizontalY / this.scaleY);

this.angleYShow = ((ang \* 180) / Math.PI).toFixed(2);

return ang;

},

scaleY:function()

{

//AB = SQRT( (Xb - Xa)^2 + (Yb - Ya)^2 )

var Xb = this.EndPosXforY;

var Yb = this.EndPosYforY;

var Xa = this.StartPosXforY;

var Ya = this.StartPosYforY;

return (Math.sqrt(Math.pow( Xb - Xa, 2) + Math.pow( Yb - Ya, 2)) /

( this.EndRangeForY - this.StartRangeForY ));

},

scaleZ:function()

{

console.log(this.EndPosYforZ);

console.log(this.StartPosYforZ);

console.log(this.EndRangeForZ);

console.log(this.StartRangeForZ);

return (this.EndPosYforZ - this.StartPosYforZ) /

( this.EndRangeForZ - this.StartRangeForZ );

},

rotate:function()

{

this.axisStepX = 0.1;

this.axisStepY = 0.1;

this.axisStepZ = 0.1;

this.countPoints(false);

},

previous:function()

{

this.axisStepX = 1;

this.axisStepY = 1;

this.axisStepZ = 1;

this.countPoints(false);

},

},

methods:

{

incr:function()

{

if (Number(this.axisStepX) < 5)

{

this.axisStepX = Number(this.axisStepX) + 0.1;

this.axisStepY = Number(this.axisStepY) + 0.1;

this.axisStepZ = Number(this.axisStepZ) + 0.1;

}

else

{

this.axisStepX = 1;

this.axisStepY = 1;

this.axisStepZ = 1; }

this.countPoints(false);

},

printXY:function(e)

{

var Top = 0;

for (var i = 0;

i < e.target.parentElement.children.length ; i++)

{

if (e.target.parentElement.children[i] === e.target)

break;

Top =

Top + e.target.parentElement.children[i].

offsetHeight;

}

this.mouseX = e.x - e.target.parentElement.offsetLeft;

this.mouseY = e.y + e.target.parentElement.offsetTop - Top;

},

printLost:function(e)

{

this.mouseX = e.x - e.target.parentElement.offsetLeft;

this.mouseY = e.y;

},

countTxts:function()

{

this.txts = [];

var txt = new Object();

txt.name = "X";

txt.x = Number(this.shift);

txt.y = Number(this.hi) - Number(this.shift) –

Number(this.arLen);

this.txts.push(txt);

var txt = new Object();

txt.name = "Y";

txt.x = Number(this.wi) - Number(this.shift) –

Number(this.arLen) / 2;

txt.y = Number(this.hi) - Number(this.shift) –

Number(this.arLen);

this.txts.push(txt);

var txt = new Object();

txt.name = "Z";

txt.x = Number(this.wi) / 2 +

Number(this.arLen) / 2;

txt.y = Number(this.shift) +

Number(this.arLen) / 2;

this.txts.push(txt);

var txt = new Object();

txt.name = "0";

txt.x = Number(this.wi) / 2 –

Number(this.arLen);

txt.y = Number(this.hi) / 2 –

Number(this.arLen) / 2;

this.txts.push(txt);

},

countAxisX:function()

{

this.axisX = [];

var axis = new Object();

axis.x1 = this.StartPosXforX;

axis.y1 = this.StartPosYforX;

axis.x2 = this.EndPosXforX;

axis.y2 = this.EndPosYforX;

this.axisX.push(axis);

var arrow = new Object();

arrow.x1 = this.EndPosXforX;

arrow.y1 = this.EndPosYforX;

arrow.x2 = this.EndPosXforX + Number(this.arLen);

arrow.y2 = this.EndPosYforX - Number(this.arLen);

this.axisX.push(arrow);

var arrow = new Object();

arrow.x1 = this.EndPosXforX;

arrow.y1 = this.EndPosYforX;

arrow.x2 = this.EndPosXforX + Number(this.arLen);

arrow.y2 = this.EndPosYforX - Math.sqrt(this.arLen);

this.axisX.push(arrow);

},

countStrokesX:function()

{

var axisStep = Number(this.axisStepX);

var kolPoints = (this.EndRangeForX - this.StartRangeForX) /

axisStep;

this.strokesX = [];

for (var i = 0; i <= kolPoints ; i++)

{

if (this.StartRangeForX + i \* axisStep != 0)

{

if (kolPoints != i)

{

var str = new Object();

str.x1 = this.StartPosXforX - this.arAng +

i \* axisStep \* this.scaleX \*

Math.cos(

this.angleXToHorizontal

);

str.y1 = this.StartPosYforX + i \* axisStep

\* this.scaleX \*

Math.sin(

this.angleXToHorizontal

);

str.x2 = this.StartPosXforX + this.arAng +

i \* axisStep \* this.scaleX \*

Math.cos(

this.angleXToHorizontal

);

str.y2 = this.StartPosYforX + i \* axisStep

\* this.scaleX \*

Math.sin(

this.angleXToHorizontal

);

str.txtX = this.StartPosXforX - this.arAng

+ i \* axisStep \* this.scaleX

\* Math.cos( this.angleXToHorizontal);

str.txtY = this.StartPosYforX + i \*

axisStep \* this.scaleX \*

Math.sin(

this.angleXToHorizontal);

if (Number.isInteger(this.StartRangeForX +

i \* axisStep))

str.txt = (this.StartRangeForX +

i \* axisStep);

else

str.txt = (this.StartRangeForX + i \*

axisStep).toFixed(1);

this.strokesX.push(str);

}

this.vals.push(this.StartRangeForX + i \*

axisStep);

}

}

},

countGridX:function()

{

var axisStep = Number(this.axisStepX);

var kolPoints = (this.EndRangeForX - this.StartRangeForX) /

axisStep;

this.gridX = [];

for (var i = 0; i <= kolPoints ; i++)

{

if (this.StartRangeForX + i \* axisStep != 0)

{

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforX + i \* axisStep \* this.scaleX \* Math.cos(this.angleXToHorizontal);

gd.y1 = Number(this.shift);

gd.x2 = this.StartPosXforX + i \* axisStep \* this.scaleX \* Math.cos(this.angleXToHorizontal);

gd.y2 = ( this.flgPositive ? this.StartPosYforX + i \* axisStep \* this.scaleX \* Math.sin(this.angleXToHorizontal) : this.EndPosYforX);

this.gridX.push(gd);

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforX - ( !this.flgPositive ? (this.wi - this.shift) / 2 : 0 ) + i \* axisStep \* this.scaleX \* Math.cos(this.angleXToHorizontal);

this.gridX.push(gd);

}

}

},

countAxisY:function()

{

this.axisY = [];

var axis = new Object();

axis.x1 = this.StartPosXforY;

axis.y1 = this.StartPosYforY;

axis.x2 = this.EndPosXforY;

axis.y2 = this.EndPosYforY;

this.axisY.push(axis);

var arrow = new Object();

arrow.x1 = this.EndPosXforY;

arrow.y1 = this.EndPosYforY;

arrow.x2 = this.EndPosXforY - Number(this.arLen);

arrow.y2 = this.EndPosYforY - Math.sqrt(this.arLen);

this.axisY.push(arrow);

var arrow = new Object();

arrow.x1 = this.EndPosXforY;

arrow.y1 = this.EndPosYforY;

arrow.x2 = this.EndPosXforY - Number(this.arLen);

arrow.y2 = this.EndPosYforY - Number(this.arLen);

this.axisY.push(arrow);

},

countStrokesY:function()

{

var axisStep = Number(this.axisStepY);

var kolPoints = (this.EndRangeForY - this.StartRangeForY) / axisStep;

this.strokesY = [];

for (var i = 0; i <= kolPoints ; i++)

{

if (this.StartRangeForY + i \* axisStep != 0)

{

if (i != kolPoints)

{

var str = new Object();

str.x1 = this.StartPosXforY - this.arAng + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

str.y1 = this.StartPosYforY + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal);

str.x2 = this.StartPosXforY + this.arAng + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

str.y2 = this.StartPosYforY + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal);

str.txtX = this.StartPosXforY - this.arAng + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

str.txtY = this.StartPosYforY + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal);

if (Number.isInteger(this.StartRangeForY + i \* axisStep))

str.txt = (this.StartRangeForY + i \* axisStep);

else

str.txt = (this.StartRangeForY + i \* axisStep).toFixed(1);

this.strokesY.push(str);

}

}

}

},

countGridY:function()

{

var axisStep = Number(this.axisStepY);

var kolPoints = (this.EndRangeForY - this.StartRangeForY) / axisStep;

this.gridY = [];

for (var i = 0; i <= kolPoints ; i++)

{

if (this.StartRangeForY + i \* axisStep != 0)

{

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforY + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

gd.y1 = Number(this.shift);

gd.x2 = this.StartPosXforY + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

gd.y2 = ( this.flgPositive ? this.StartPosYforY + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal) : this.EndPosYforY);

this.gridY.push(gd);

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforY + ( !this.flgPositive ? (this.wi - this.shift) / 2 : 0 ) + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

gd.y1 = this.StartPosYforY - ( !this.flgPositive ? (this.hi - this.shift) / 2 : 0 ) + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal);

gd.x2 = this.StartPosXforY - ( (this.wi - this.shift) / 2 ) + i \* axisStep \* this.scaleY \* Math.cos(this.angleYToHorizontal);

gd.y2 = this.StartPosYforY + ( (this.hi - this.shift) / 2 ) + i \* axisStep \* this.scaleY \* Math.sin(this.angleYToHorizontal);

this.gridY.push(gd);

}

}

},

countAxisZ:function()

{

this.axisZ = [];

var axis = new Object();

axis.x1 = this.StartPosXforZ;

axis.y1 = this.StartPosYforZ;

axis.x2 = this.EndPosXforZ;

axis.y2 = this.EndPosYforZ;

this.axisZ.push(axis);

//Левая часть стрелки оси

var arrow = new Object();

arrow.x1 = Number(this.wi) / 2;

arrow.y1 = Number(this.shift);

arrow.x2 = (Number(this.wi) / 2) - Number(this.arAng);

arrow.y2 = Number(this.shift) + this.arLen;

this.axisZ.push(arrow);

//Правая часть стрелки оси

var arrow = new Object();

arrow.x1 = Number(this.wi) / 2;

arrow.y1 = Number(this.shift);

arrow.x2 = (Number(this.wi) / 2) + Number(this.arAng);

arrow.y2 = Number(this.shift) + Number(this.arLen);

this.axisZ.push(arrow);

},

countStrokesZ:function()

{

var axisStep = Number(this.axisStepZ);

var kolPoints = (this.EndRangeForZ - this.StartRangeForZ) / axisStep;

var Scale = ( !this.flgPositive ? (this.hi - 2 \* (this.shift)) : (this.hi / 2 - (this.shift)) ) / (!this.flgPositive ? 10 : 5);

this.strokesZ = [];

for (var i = 0; i <= kolPoints; i++)

{

if (this.StartRangeForZ + i \* axisStep != 0)

{

if (i != kolPoints)

{

var str = new Object();

str.x1 = this.StartPosXforZ - this.arAng;

str.y1 = this.StartPosYforZ - i \* axisStep \* Scale;

str.x2 = this.StartPosXforZ + this.arAng;

str.y2 = this.StartPosYforZ - i \* axisStep \* Scale;

str.txtX = this.StartPosXforZ + this.arLen / 2;

str.txtY = this.StartPosYforZ + (this.arAng / 2) - i \* axisStep \* Scale;

if (Number.isInteger(this.StartRangeForZ + i \* axisStep))

str.txt = (this.StartRangeForZ + i \* axisStep);

else

str.txt = (this.StartRangeForZ + i \* axisStep).toFixed(1);

this.strokesZ.push(str);

}

}

}

},

countGridZ:function()

{

var axisStep = Number(this.axisStepZ);

var kolPoints = (this.EndRangeForZ - this.StartRangeForZ) / axisStep;

var Scale = ( !this.flgPositive ? (this.hi - 2 \* (this.shift)) : (this.hi / 2 - (this.shift)) ) / (!this.flgPositive ? 10 : 5);

this.gridZ = [];

for (var i = 0; i <= kolPoints ; i++)

{

if (this.StartRangeForY + i \* axisStep != 0)

{

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforY;

gd.y1 = this.StartPosYforY + ( !this.flgPositive ? (this.hi - 2 \* this.shift) / 2 : 0 ) - i \* axisStep \* Scale;

gd.x2 = this.EndPosXforY;

gd.y2 = this.EndPosYforY + ( !this.flgPositive ? (this.hi - 2 \* this.shift) / 2 : 0 ) - i \* axisStep \* Scale;

this.gridY.push(gd);

var gd = new Object();

gd.i = i;

gd.x1 = this.StartPosXforX;

gd.y1 = this.StartPosYforX + ( !this.flgPositive ? (this.hi - 2 \* this.shift) / 2 : 0 ) - i \* axisStep \* Scale;

gd.x2 = this.EndPosXforX;

gd.y2 = this.EndPosYforX + ( !this.flgPositive ? (this.hi - 2 \* this.shift) / 2 : 0 ) - i \* axisStep \* Scale;

this.gridZ.push(gd);

}

}

},

countPoints:function(flgFirstTime)

{

this.startZ = !this.flgPositive ? -5 : 0;

this.endZ = 5;

if (this.flgXaxis)

this.countAxisX();

else

this.axisX = [];

if (this.flgYaxis)

this.countAxisY();

else

this.axisY = [];

if (this.flgZaxis)

this.countAxisZ();

else

this.axisZ = [];

if (this.flgTxts)

this.countTxts();

else

this.txts = [];

this.points = [];

this.pts = [];

for (var i = 0; i < this.points.length; i++)

{

var pt = new Object();

var x0 = Number(this.wi) / 2;

var y0 = Number(this.hi) / 2;

var dist = Math.sqrt(Math.pow(this.points[i].x, 2) + Math.pow(this.points[i].y, 2));

pt.x1 = x0 + dist \* this.angleXToHorizontal \* Math.cos(this.angleXToHorizontal);

pt.y1 = y0 + dist \* this.angleYToHorizontal \* Math.sin(this.angleYToHorizontal);

this.pts.push(pt);

}

if (this.flgXstrokes)

this.countStrokesX();

else

this.strokesX = [];

if (this.flgXgrid)

this.countGridX();

else

this.gridX = [];

if (this.flgYstrokes)

this.countStrokesY();

else

this.strokesY = [];

if (this.flgYgrid)

this.countGridY();

else

this.gridY = [];

if (this.flgZstrokes)

this.countStrokesZ();

else

this.strokesZ = [];

if (this.flgZgrid)

this.countGridZ();

else

this.gridZ = [];

},

writeExampleExcel:function()

{

var fileName = "example.xlsx";

var wb = XLSX.utils.book\_new();

var data =

[

[1, 2, 3],

[true, false, null, "sheetjs"],

["foo", "bar", new Date("2014-02-19T14:30Z"), "0.3"],

["baz", null, "qux"]

];

//Выдаём имя листу книги Excel

var ws\_name = "SheetJS";

var ws = XLSX.utils.aoa\_to\_sheet(data);

XLSX.utils.book\_append\_sheet(wb, ws, ws\_name);

XLSX.writeFile(wb, fileName);

},

writePointsExcel:function()

{

var filename = "points.xlsx";

var wb = XLSX.utils.book\_new();

var ws\_name = "Синусоида";

var ws = XLSX.utils.json\_to\_sheet(this.points);

XLSX.utils.book\_append\_sheet(wb, ws, ws\_name);

XLSX.writeFile(wb, filename);

}

},

created: function()

{ this.countPoints(true); }

});

</SCRIPT>

**Листинг стилевой части одностраничного веб-приложения:**

<STYLE TYPE="text/css">

svg

{ border: 1px solid #000000; }

input

{ width:60px; }

</STYLE>

**Вывод:**

В результате проделанной работы создано одностраничное веб-приложение, выполняющее все функции, отмеченные согласно постановке задачи, заданной по рассмотренному варианту курсового проекта (№ 20).

В одностраничном веб-приложении создан задел для развития его в полноценный браузерный построитель поверхностей для различных учебных и прикладных нужд.

Освоены навыки работы по настройке фреймворка *Vue.js* для учебных и прикладных нужд.

Закреплены навыки использования математического аппарата проецирования точек плоскости на координатные оси, расположенные под различными углами поворота.

**Список литературы:**

1. Сафронов А.И., Котова А.И. Проектирование типовой информационной системы управления с использованием технологии *web*-программирования на базе фреймворка *Vue.js*: Учебно-методическое пособие для проведения аудиторных занятий по дисциплине «Информационное обеспечение систем управления». – М.: РУТ (МИИТ), 2019. – 97 с.

2. *Vue.js* [Электронный ресурс] : Введение – *Vue.js*. *URL*: *https://ru.vuejs.org/v2/guide/* (дата обращения: 23.12.2019).

3. *Vue.js* [Электронный ресурс] : Редактор *Markdown – Vue.js*. URL: *https://ru.vuejs.org/v2/examples/* (дата обращения: 23.12.2019).