



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Фундаментальные науки»
КАФЕДРА «Вычислительная математика и математическая физика» (ФН-11)

ОТЧЕТ
по домашнему заданию №2

Дисциплина: Геометрическое моделирование

Студент группы ФН11-62Б

(Подпись, дата)

Л.В.Ладыгина
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

А.А.Захаров
(И.О.Фамилия)

Оценка: _____

Москва, 2022

Задание:

Используя построенный в домашнем задании №1 сплайн, осуществите расчёт его точек таким образом, чтобы аппроксимирующий этот сплайн полигон, отклонялся от него на величину не больше заданной. Проведите сравнение полученной визуализации с визуализацией с постоянным шагом.

Требуется найти параметр t_1 следующей точки полигона, так чтобы отклонение кривой от её полигона не превышало заданную величину δ .

Для этого используем формулы:

$$\rho = \frac{|\mathbf{r}'|^3}{|\mathbf{r}' \times \mathbf{r}''|}.$$
$$\Delta t \approx \frac{h}{|\mathbf{r}'|} = 2 \frac{\sqrt{\delta(2\rho - \delta)}}{|\mathbf{r}'|}.$$
$$t_1 = t_0 + \Delta t.$$

и формулы для производных:

$$\frac{d\mathbf{r}(t)}{dt} = n \sum_{i=0}^{n-1} B_i^{n-1}(t) \mathbf{a}_i.$$
$$\mathbf{a}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$$
$$\frac{d^2\mathbf{r}(t)}{dt^2} = n(n-1) \sum_{i=0}^{n-2} B_i^{n-2}(t) \mathbf{b}_i,$$
$$\mathbf{b}_i = \mathbf{a}_{i+1} - \mathbf{a}_i.$$

Расчет первой производной в `rt_diff`:

```
.....
for(i=0; i<n-1; i++){
  B_in=C(i,n-2)*Math.pow(t,i)*Math.pow(1-t, n-2-i);
  a_x[i]=this.pointsCtr[i+1].x-this.pointsCtr[i].x;
  a_y[i]=this.pointsCtr[i+1].y-this.pointsCtr[i].y;
  s_x0+=(n-1)*B_in*a_x[i];
  s_y0+=(n-1)*B_in*a_y[i];
}
const rt_diff=vec3.fromValues(s_x0,s_y0,0);
```

Расчет второй производной в `rt_2diff`:

```
for(i=0; i<n-2; i++){
  B_in=C(i,n-3)*Math.pow(t,i)*Math.pow(1-t, n-3-i);
  b_x[i]=a_x[i+1]-a_x[i];
  b_y[i]=a_y[i+1]-a_y[i];
  s_x1+=(n-1)*(n-2)*B_in*b_x[i];
  s_y1+=(n-1)*(n-2)*B_in*b_y[i];
}
const rt_2diff=vec3.fromValues(s_x1,s_y1,0);
```

Объявим

вектор *normal*- вектор векторного произведения $r' \times r''$

```
const normal = vec3.create();
vec3.cross(normal,rt_diff,rt_2diff);
```

и

переменные

$ro = \rho$,

rt_diff_norm – норма *rt_diff*,

normal_norm – норма вектора *normal*, посчитаем *rt_diff_norm*:

```
let ro,rt_diff_norm, normal_norm;
rt_diff_norm=Math.sqrt(Math.pow(s_x0,2)+Math.pow(s_y0,2));
```

Найдем норму вектора *normal*

```
normal_norm=Math.sqrt(Math.pow(normal[0],2)+Math.pow(normal[1],2)+Math.pow(normal[2],2));
```

и найдем ro по формуле:

$$\rho = \frac{|\mathbf{r}'|^3}{|\mathbf{r}' \times \mathbf{r}''|}.$$

```
ro=Math.pow(rt_diff_norm,3)/normal_norm;
```

Введем параметр δ и посчитаем Δt по формуле

$$\Delta t \approx \frac{h}{|\mathbf{r}'|} = 2 \frac{\sqrt{\delta(2\rho - \delta)}}{|\mathbf{r}'|}.$$

```
let delta=0.1;
delta_t=2*Math.sqrt(delta*(2*ro-delta))/rt_diff_norm;
```

текущее t будет изменяться на Δt :

```
t+=delta_t;
```

```

let B_in,s_x=0,s_y=0;
var a_x=[],a_y=[],b_x=[],b_y=[];
t=0; j=0;
while(t<1){

    for(i=0; i<n; i++){
        B_in=C(i,n-1)*Math.pow(t,i)*Math.pow(1-t, n-1-i);
        s_x+=B_in*this.pointsCtr[i].x;
        s_y+=B_in*this.pointsCtr[i].y;
    }
    rt=new Point(s_x,s_y);
    this.pointsSpline[j]=rt;
    s_x=0; s_y=0;

    let s_x0=0,s_x1=0,s_y0=0,s_y1=0;

    //производная
    for(i=0; i<n-1; i++){
        B_in=C(i,n-2)*Math.pow(t,i)*Math.pow(1-t, n-2-i);
        a_x[i]=this.pointsCtr[i+1].x-this.pointsCtr[i].x;
        a_y[i]=this.pointsCtr[i+1].y-this.pointsCtr[i].y;
        s_x0+=(n-1)*B_in*a_x[i];
        s_y0+=(n-1)*B_in*a_y[i];
    }
    const rt_diff=vec3.fromValues(s_x0,s_y0,0);

    console.log('rt_diff',rt_diff);

    //вторая производная
    for(i=0; i<n-2; i++){
        B_in=C(i,n-3)*Math.pow(t,i)*Math.pow(1-t, n-3-i);
        b_x[i]=a_x[i+1]-a_x[i];
        b_y[i]=a_y[i+1]-a_y[i];
        s_x1+=(n-1)*(n-2)*B_in*b_x[i];
        s_y1+=(n-1)*(n-2)*B_in*b_y[i];
    }
    const rt_2diff=vec3.fromValues(s_x1,s_y1,0);
    console.log('rt_2diff',rt_2diff);

    let ro,rt_diff_norm, normal_norm;
    rt_diff_norm=Math.sqrt(Math.pow(s_x0,2)+Math.pow(s_y0,2));

    const normal = vec3.create();
    vec3.cross(normal,rt_diff,rt_2diff);
    console.log(normal[0], normal[1],normal[2]);

    normal_norm=Math.sqrt(Math.pow(normal[0],2)+Math.pow(normal[1],2)+Math.pow(normal[2],2));
    console.log('normal',normal_norm);
    ro=Math.pow(rt_diff_norm,3)/normal_norm;

    let delta=0.1;
    delta_t=2*Math.sqrt(delta*(2*ro-delta))/rt_diff_norm;

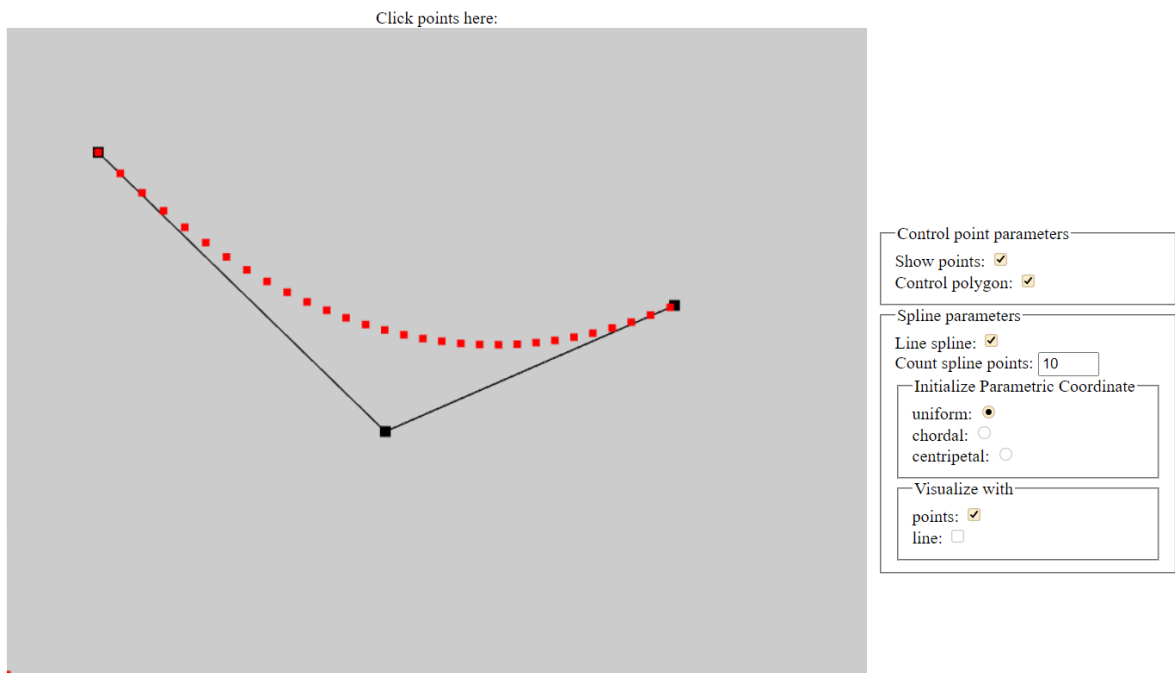
    console.log('delta_t',delta_t);

    j++;
    t+=delta_t;

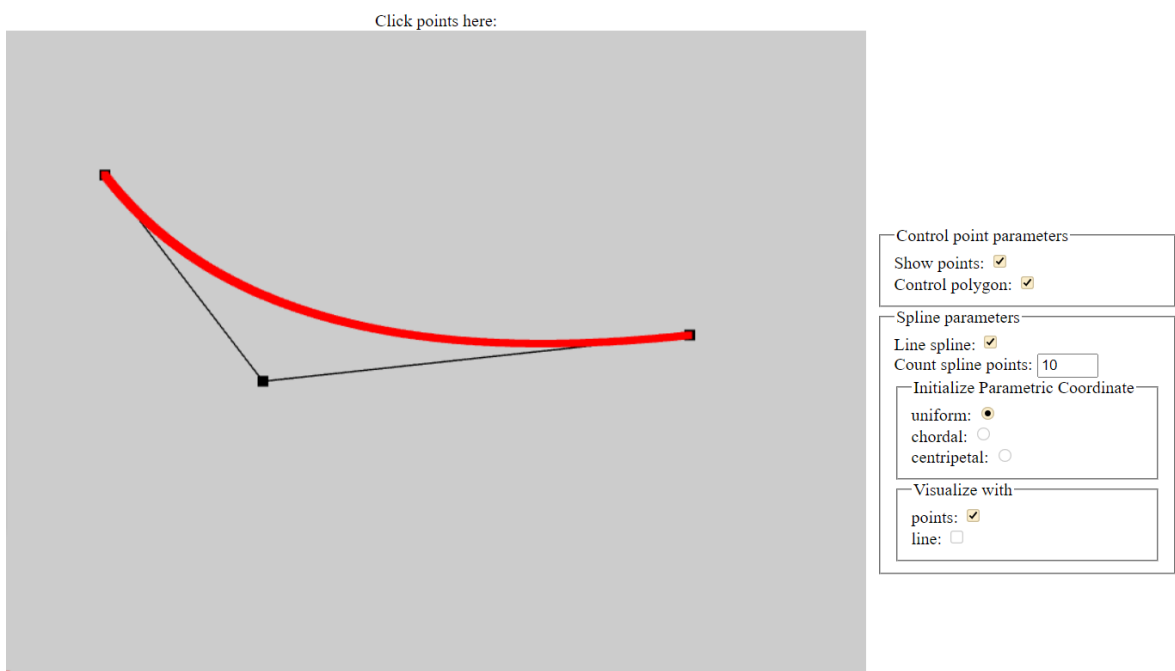
}

```

Результаты работы программы:



Для $\delta = 0.1$



Для $\delta = 0.001$