

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГАОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных**  **технологий** | **Кафедра**  **информационных технологий и вычислительных систем** |

ОТЧЕТ О ВЫПОЛНЕНИИ

ТРЕНИРОВКИ ЛАБОРАТОРНОЙ РАБОТЫ №3

ПО ДИСЦИПЛИНЕ

«Компьютерная графика»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТОВ | 2 | КУРСА | | *бакалавриата* | ГРУППЫ | *ИДБ-23-05* |
|  | | | *(уровень профессионального образования)* | |  | |

Кузнецова Вячеслава Дмитриевича

|  |
| --- |
|  |
| *(ФИО)* |

НА ТЕМУ

«Параметрические кривые»

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Направление: | 09.03.01 Информатика и вычислительная техника |
| Профиль подготовки: | **Разработка программных комплексов в рамках цифровой трансформации деятельности предприятий** |

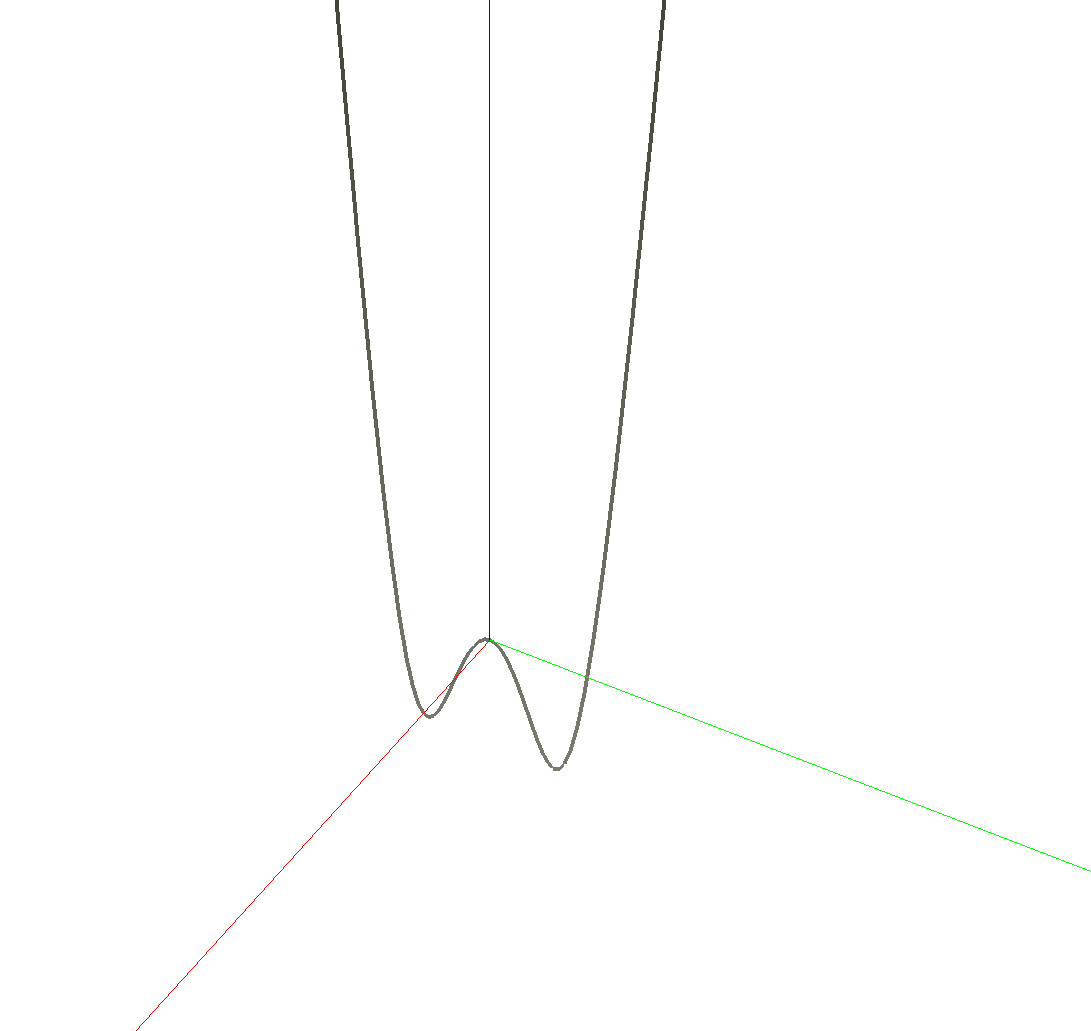
|  |  |  |  |
| --- | --- | --- | --- |
| Отчет сдан «\_\_13\_\_» \_\_\_июня\_\_\_\_\_\_\_2025\_г. | | | |
|  |  |  |  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
|  | | | |
| Преподаватель | Климаков М.А., преподаватель |  |  |
|  | *(Ф.И.О., должность, степень, звание.)* |  | *(подпись)* |

МОСКВА 2025

## Преамбула

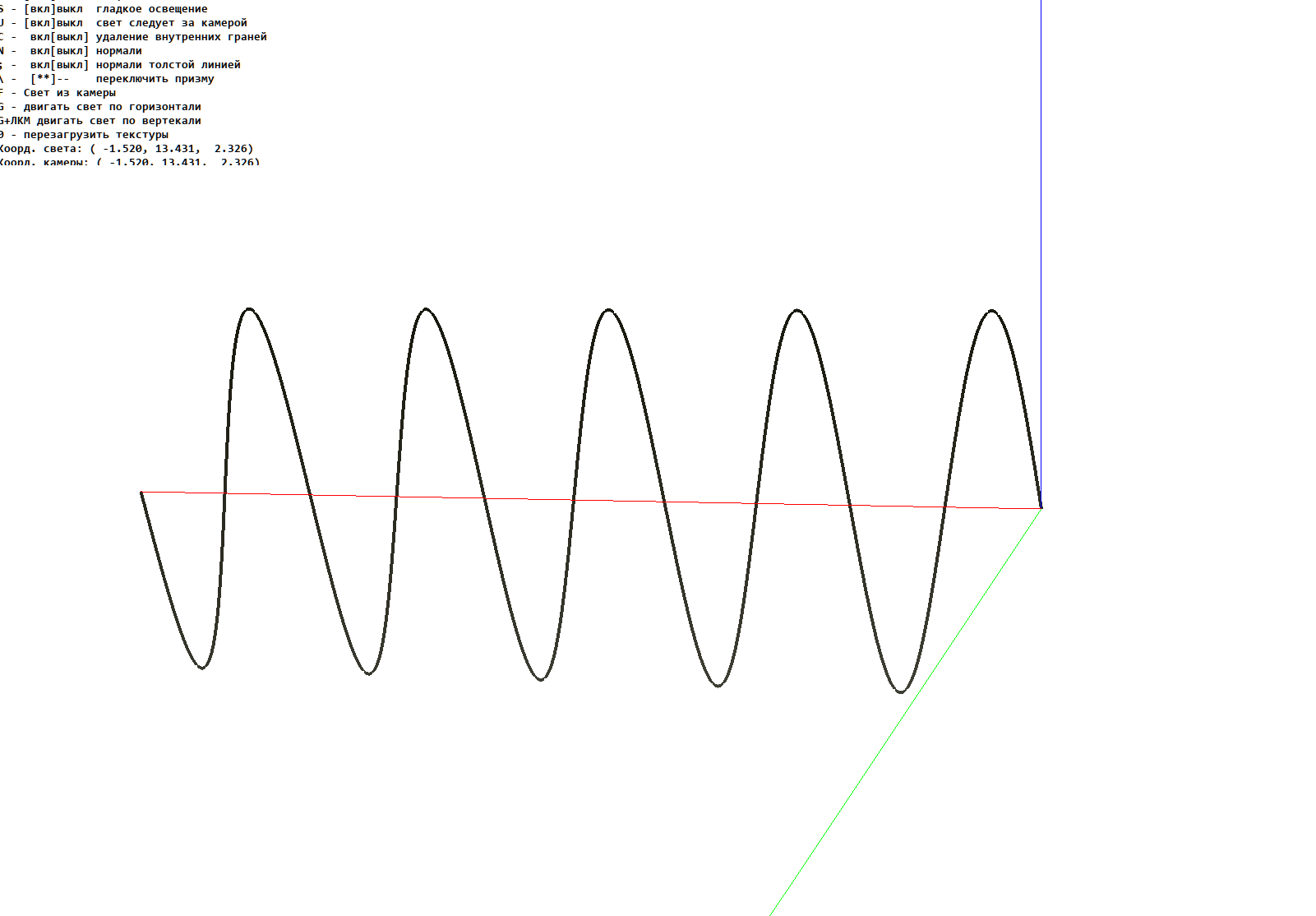
#include <windows.h>  
#include <GL/gl.h>  
#include <GL/glu.h>  
#include <cmath>  
#ifndef M\_PI  
#define M\_PI 3.14159265358979323846  
#endif  
struct vec3 {  
 double x, y, z;  
 /\* Тут другие важные вещи, но они не попали в релиз тренировки \*/  
};

## 1 а

void l3tr\_1a() {  
 // 1 a  
 double xmin = -5, xmax = 5;  
 size\_t samples = 100;  
 double scale = (xmax - xmin) / samples;  
 GLfloat linewidth;  
 glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double x = 0;  
 double y = xmin + i \* scale;  
 double z = pow(y, 4) - 3\*pow(y,2);  
 glVertex3d(x, y, z);  
 }  
 glEnd();  
 glLineWidth(linewidth);  
}

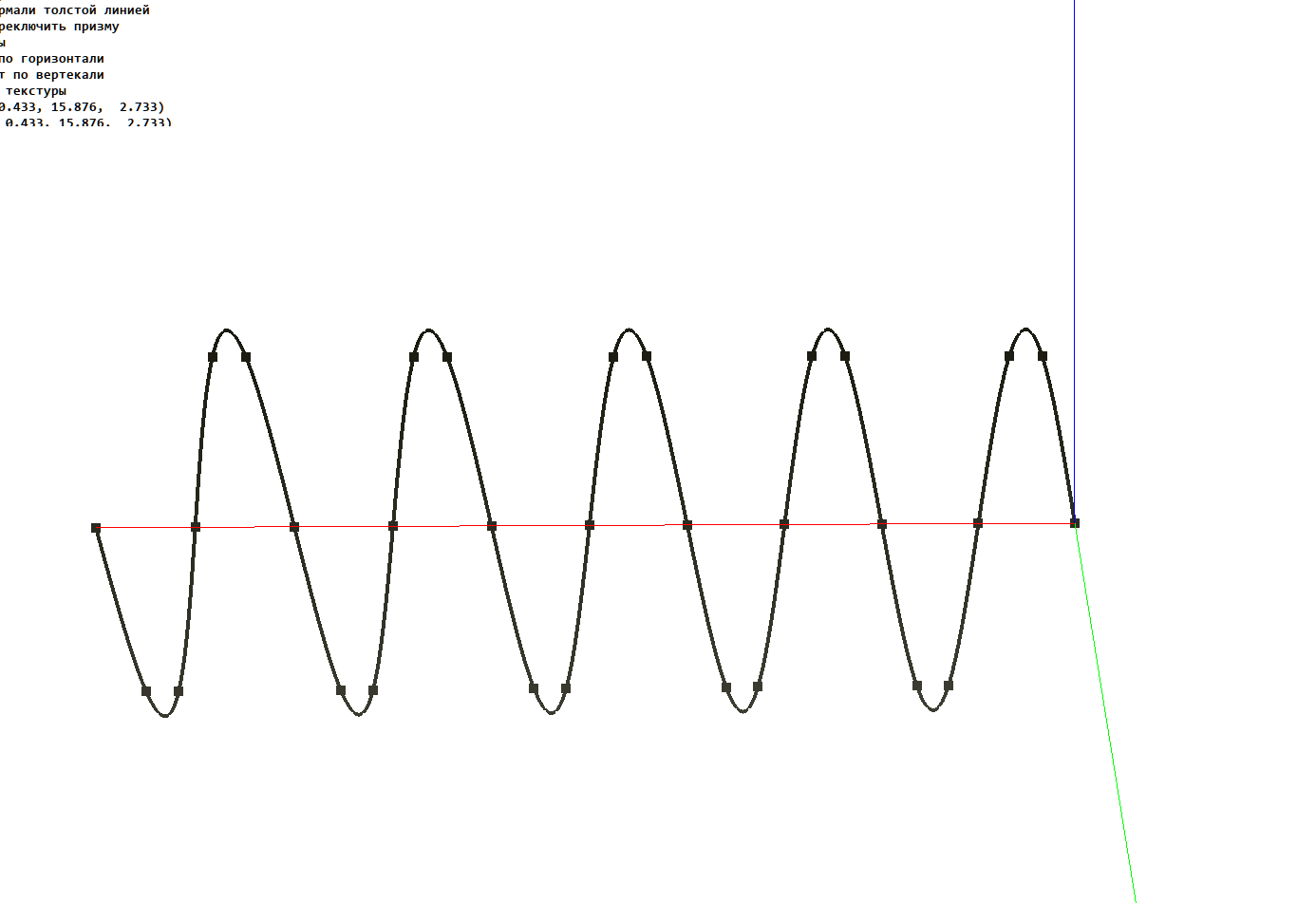
## 1 б

void l3tr\_1b() {  
 // 1 b  
 double xmin = 0, xmax = 10;  
 size\_t samples = 500;  
 double scale = (xmax - xmin) / samples;  
 GLfloat linewidth;  
 glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double x = xmin + i \* scale;  
 double y = 0;  
 double z = 2.0 \* sin(x \* M\_PI);  
 glVertex3d(x, y, z);  
 }  
 glEnd();  
 glLineWidth(linewidth);  
}



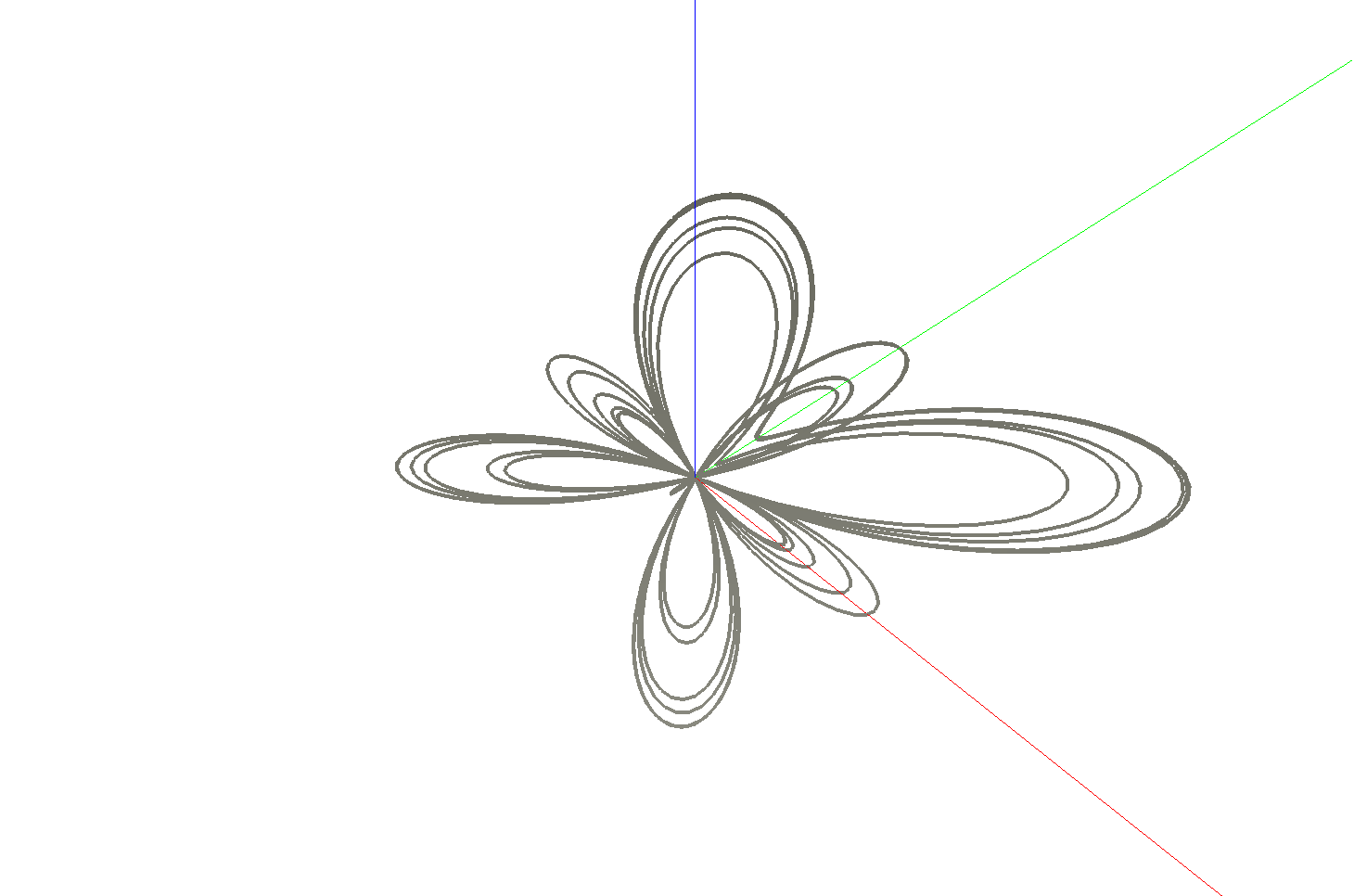
## 2

void l3tr\_2() {  
 // 2  
 size\_t samples = 500, samples2 = 31 - 1;  
 double scale = 10./samples, scale2 = 10./samples2;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double x = i \* scale;  
 glVertex3d(x, 0, 2.0 \* sin(x \* M\_PI));  
 }  
 glEnd();  
 GLfloat pointsize; glGetFloatv(GL\_POINT\_SIZE, &pointsize);  
 glPointSize(10);  
 glBegin(GL\_POINTS);  
 for (size\_t i = 0; i <= samples2; i++) {  
 double x = i \* scale2;  
 glVertex3d(x, 0, 2.0 \* sin(x \* M\_PI));  
 }  
 glEnd();  
 glPointSize(pointsize); glLineWidth(linewidth);  
}



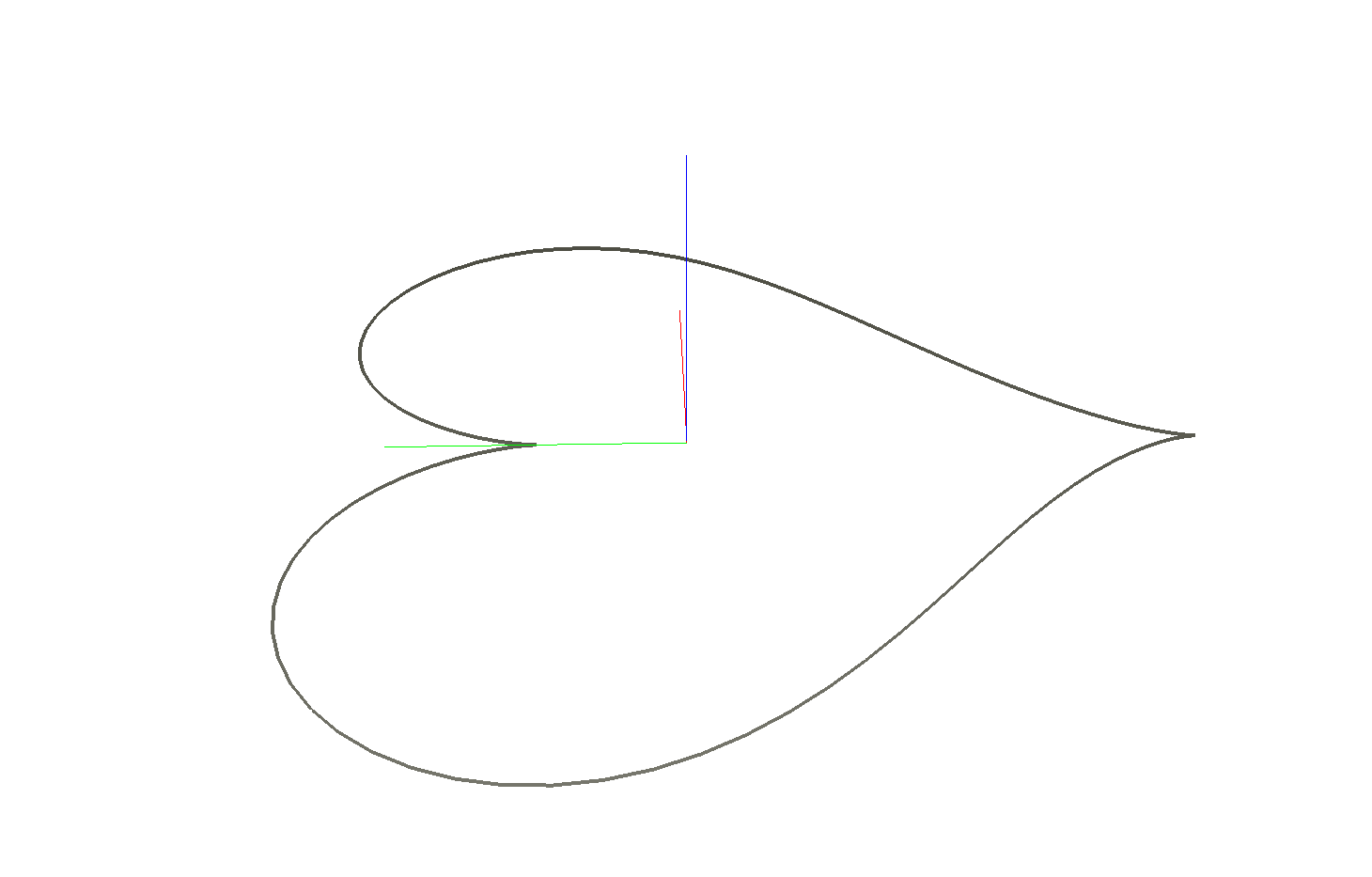
## 3 а

void l3tr\_3a() {  
 // 3 a  
 double tmin = 0.0, tmax = 12.0 \* M\_PI;  
 size\_t samples = 1000 - 1;  
 double scale = (tmax - tmin) / samples;  
 GLfloat linewidth;  
 glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double t = tmin + i \* scale;  
 double sint = sin(t);  
 double cost = cos(t);  
 double k = exp(cost) - 2 \* cos(4\*t) - pow(sin(t / 12), 5);  
 double x = sint \* k;  
 double y = cost \* k;  
 double z = 0.0;  
 glVertex3d(x, y, z);  
 }  
 glEnd();  
 glLineWidth(linewidth);  
}



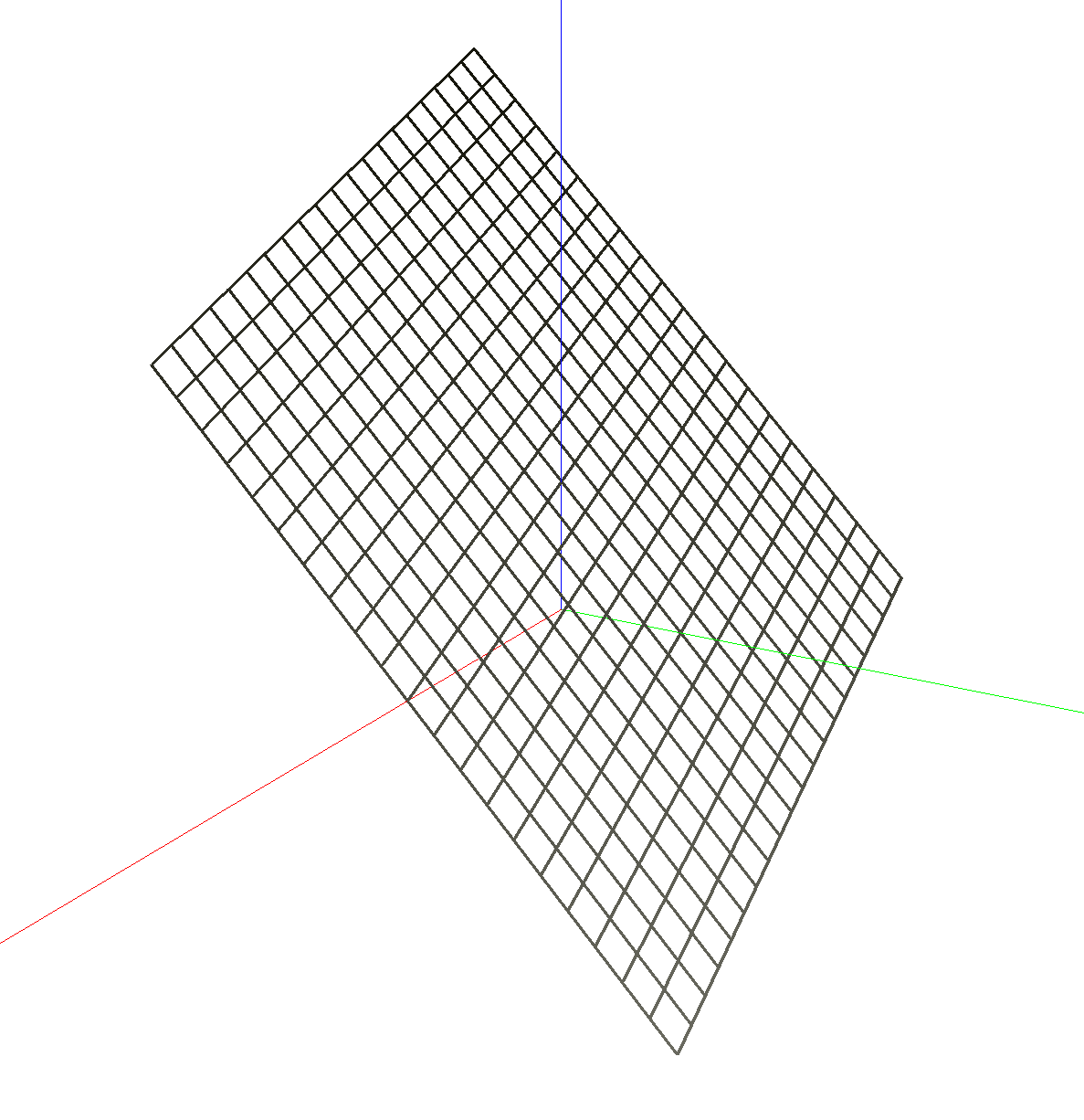
## 3 б

void l3tr\_3b() {  
 // 3 b  
 double tmin = 0.0, tmax = 2.0 \* M\_PI;  
 size\_t samples = 100 - 1;  
 double scale = (tmax - tmin) / samples;  
 GLfloat linewidth;  
 glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double t = tmin + i \* scale;  
 double x = 16\*pow(sin(t),3);  
 double y = 13\*cos(t)-5\*cos(2\*t)-2\*cos(3\*t)-cos(4\*t);  
 double z = 0.0;  
 glVertex3d(x, y, z);  
 }  
 glEnd();  
 glLineWidth(linewidth);  
}



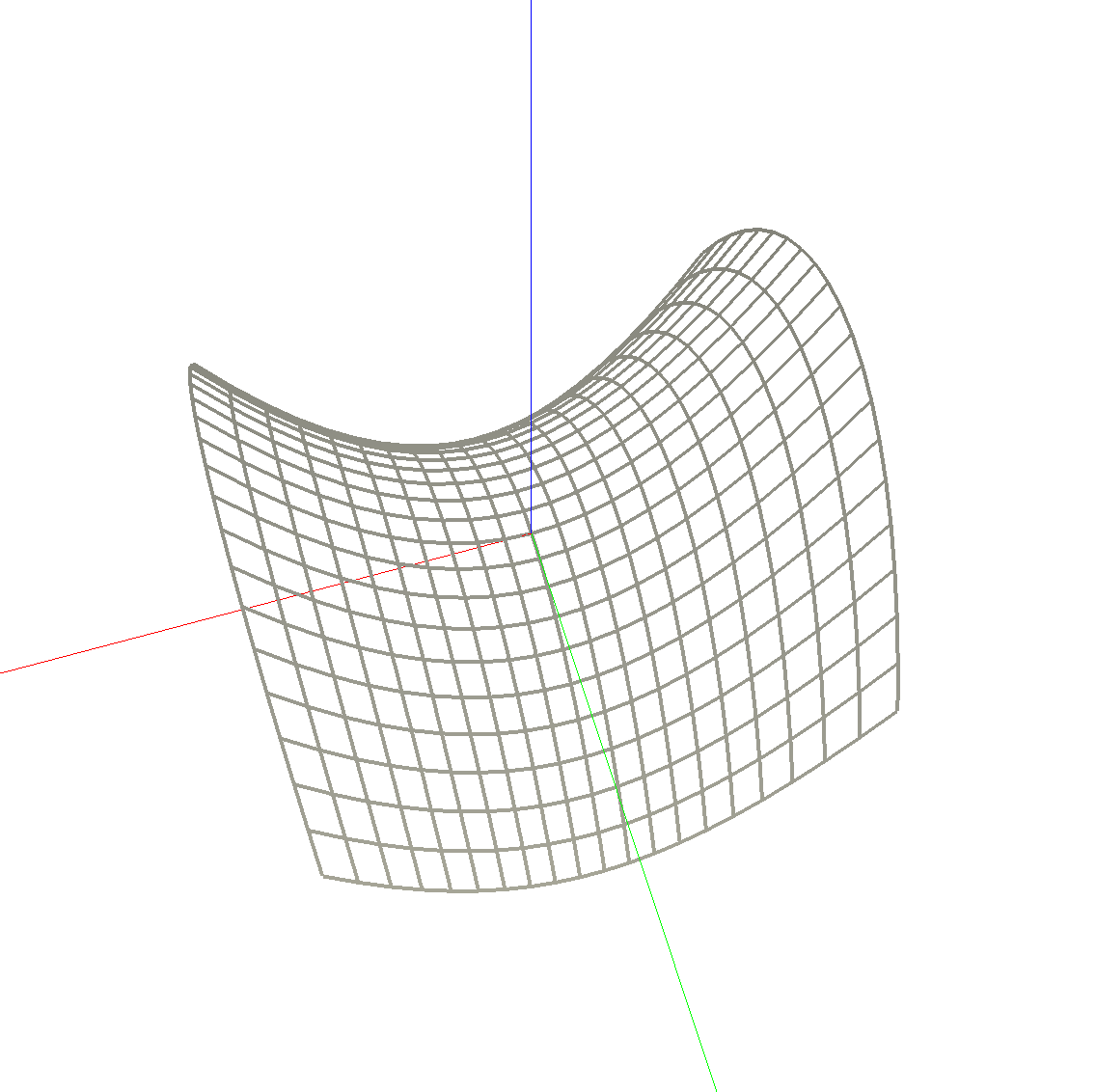
## 4 а

void l3tr\_4a() {  
 // 4 a  
 double xmin = -2, xmax = 2, ymin = -2, ymax = 2;  
 double xsamples = 20, ysamples = 20;  
 double xscale=(xmax-xmin)/xsamples, yscale=(ymax-ymin)/ysamples;  
 GLfloat linewidth;  
 glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f);  
 glBegin(GL\_LINES);  
 for (size\_t i = 0; i <= xsamples; i++) {  
 double x = xmin + i \* xscale;  
 double zmin = -(0.5\*x + ymin-1), zmax = -(0.5\*x + ymax-1);  
 glVertex3d(x, ymin, zmin);  
 glVertex3d(x, ymax, zmax);  
 }  
 for (size\_t j = 0; j <= ysamples; j++) {  
 double y = ymin + j \* yscale;  
 double zmin = -(0.5\*xmin + y-1), zmax = -(0.5\*xmax + y-1);  
 glVertex3d(xmin, y, zmin);  
 glVertex3d(xmax, y, zmax); }  
 glEnd();  
 glLineWidth(linewidth);  
}



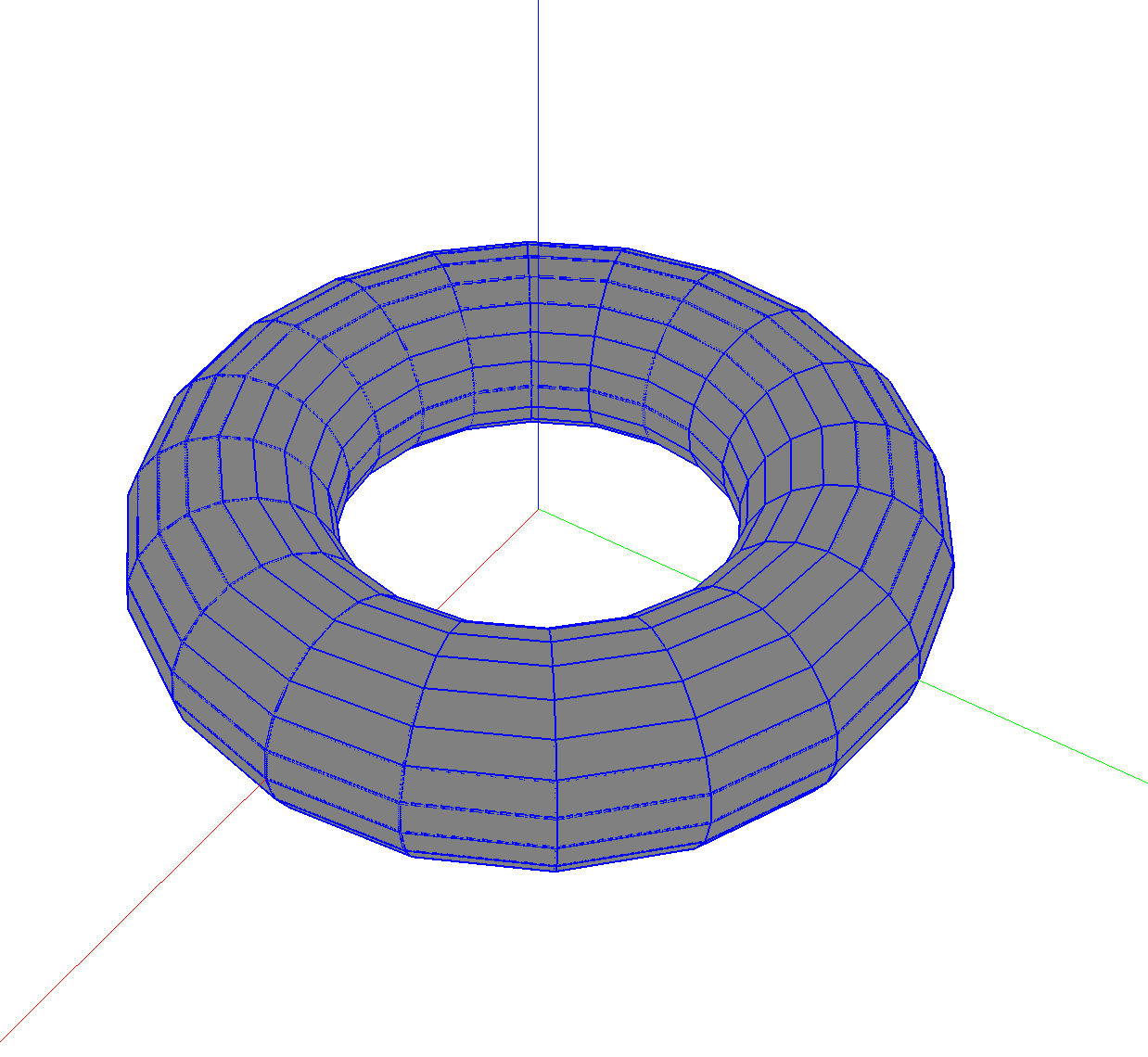
## 4 б

void l3tr\_4b() {  
 // 4 b  
 double xmin = -2, xmax = 2, ymin = -2, ymax = 2;  
 double xsamples = 20, ysamples = 20;  
 double xscale=(xmax-xmin)/xsamples, yscale=(ymax-ymin)/ysamples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glBegin(GL\_LINES);  
 for (size\_t i = 0; i <= xsamples; i++) {  
 double x = xmin + i \* xscale; double x1 = x + xscale;  
 for (size\_t j = 0; j <= ysamples; j++) {  
 double y = ymin + j \* yscale; double y1 = y + yscale;  
 double z = 0.5\*(x\*x-y\*y);  
 double zx = 0.5\*(x1\*x1-y\*y);  
 double zy = 0.5\*(x\*x-y1\*y1);  
 if (i!=xsamples) {glVertex3d(x,y,z);glVertex3d(x1,y,zx);}  
 if (j!=ysamples) {glVertex3d(x,y,z);glVertex3d(x,y1,zy);}  
 }  
 }  
 glEnd(); glLineWidth(linewidth);  
}

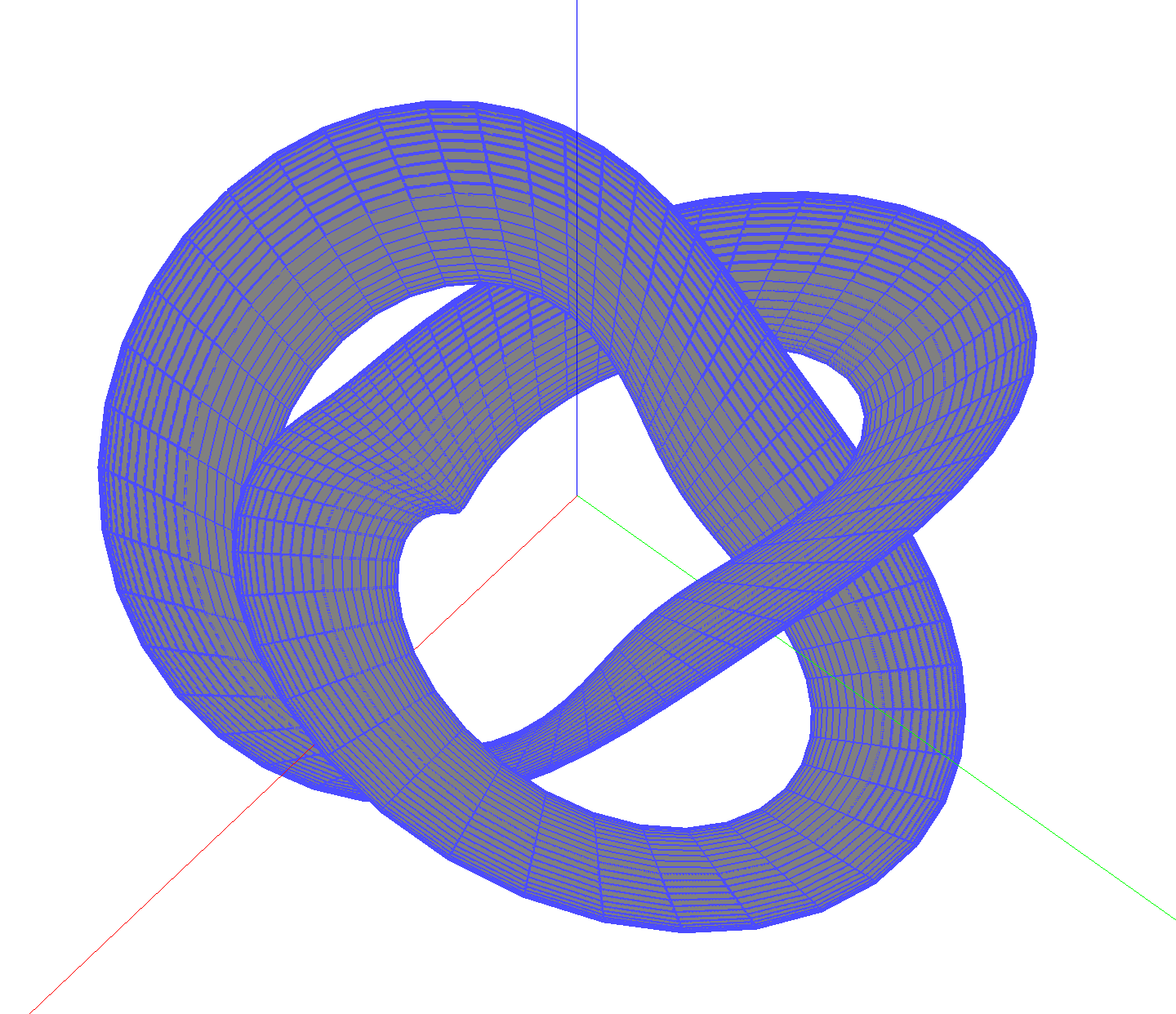


## 5 а

vec3 f5a(double u, double v) {  
 return vec3{ cos(u)\*(cos(v)+3), sin(u)\*(cos(v)+3), sin(v) };  
}  
void l3tr\_5a() {  
 // 5 a  
 double umin=-M\_PI,umax=+M\_PI,vmin=-M\_PI,vmax=+M\_PI;  
 size\_t usamples = 20, vsamples = 20;  
 double uscale=(umax-umin)/usamples, vscale=(vmax-vmin)/vsamples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glDisable(GL\_LIGHTING);  
 for (size\_t i = 0; i < usamples; i++) {  
 double u = umin + i \* uscale; double u1 = u + uscale;  
 for (size\_t j = 0; j < vsamples; j++) {  
 double v = vmin + j \* vscale; double v1 = v + vscale;  
 vec3 u0v0 = f5a(u, v), u1v0 = f5a(u1, v);  
 vec3 u0v1 = f5a(u, v1), u1v1 = f5a(u1, v1);  
 glColor3d(0.5, 0.5, 0.5); glBegin(GL\_TRIANGLES);  
 glVertex3dv(u0v0); glVertex3dv(u1v0); glVertex3dv(u1v1);  
 glVertex3dv(u1v1); glVertex3dv(u0v1); glVertex3dv(u0v0);  
 glEnd(); glColor3d(0.0, 0.0, 1.0); glBegin(GL\_LINE\_STRIP);  
 glVertex3dv(u0v0); glVertex3dv(u1v0); glVertex3dv(u1v1);  
 glVertex3dv(u0v1); glVertex3dv(u0v0); glEnd();  
 }  
 }  
 glLineWidth(linewidth);  
}



## 5 б

vec3 f5b(double u, double v) { double k = 1.5 + 0.5 \* sin(1.5 \* u);  
 return vec3{cos(u)\*sin(v)+3\*cos(u)\*k,sin(u)\*sin(v)+3\*sin(u)\*k,  
 sin(v)+2\*cos(1.5\*u) }; }  
void l3tr\_5b() {  
 // 5 b  
 double umin=-2.0\*M\_PI, umax=+2.0\*M\_PI, vmin=-M\_PI, vmax=+M\_PI;  
 size\_t usamples = 100, vsamples = 50;  
 double uscale=(umax-umin)/usamples, vscale=(vmax-vmin)/vsamples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glDisable(GL\_LIGHTING);  
 for (size\_t i = 0; i < usamples; i++) {  
 double u = umin + i \* uscale; double u1 = u + uscale;  
 for (size\_t j = 0; j < vsamples; j++) {  
 double v = vmin + j \* vscale; double v1 = v + vscale;  
 vec3 u0v0 = f5b(u, v), u1v0 = f5b(u1, v);  
 vec3 u0v1 = f5b(u, v1), u1v1 = f5b(u1, v1);  
 glColor3d(0.5, 0.5, 0.5); glBegin(GL\_TRIANGLES);  
 glVertex3dv(u0v0); glVertex3dv(u1v0); glVertex3dv(u1v1);  
 glVertex3dv(u1v1); glVertex3dv(u0v1); glVertex3dv(u0v0);  
 glEnd(); glColor3d(0.3, 0.3, 1.0); glBegin(GL\_LINE\_STRIP);  
 glVertex3dv(u0v0); glVertex3dv(u1v0); glVertex3dv(u1v1);  
 glVertex3dv(u0v1); glVertex3dv(u0v0); glEnd();  
 }  
 }  
 glLineWidth(linewidth);  
}

## 6

void l3tr\_6\_1() {  
 // Криваая Безье квадратичная  
 vec3 p[3] = {1,1,0,1.5,2,1,2.5,1.5,0};  
 vec3 A = 2[p], B = 2 \* (1[p] - 2[p]), C = 0[p] - 2 \* 1[p] + 2[p];  
 size\_t samples = 50;  
 double scale = 1. / samples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double t = scale \* i; glVertex3dv(A + B\*t + C\*t\*t);  
 }  
 glEnd(); glLineWidth(linewidth); glBegin(GL\_LINE\_STRIP);  
 glVertex3dv(0[p]); glVertex3dv(1[p]); glVertex3dv(2[p]);  
 glEnd();  
}  
void l3tr\_6\_2() {  
 // Герметичная кривая  
 vec3 p[2] = {0.34,-0.37,1,1.97,-1.22,-.5};  
 vec3 r[2] = {-.49,-.7,1,1.01,.17,.5};  
 vec3 D = 2\*0[p] + 0[r] - 2\*1[p] +1[r];  
 vec3 C = -3\*0[p] + 3\*1[p] - 2\*0[r] - 1[r], B = 0[r], A = 0[p];  
 size\_t samples = 50;  
 double scale = 1. / samples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double t = scale \* i; glVertex3dv(A + B\*t + C\*t\*t + D\*t\*t\*t);  
 }  
 glEnd(); glLineWidth(linewidth); glBegin(GL\_LINES);  
 glVertex3dv(0[p]); glVertex3dv(0[p] + 0[r]); glVertex3dv(1[p]);  
 glVertex3dv(1[p] + 1[r]); glEnd();  
}  
  
void l3tr\_6\_3() {  
 // Криваая Безье кубическая  
 vec3 p[4] = {.5,.5,0,.86,-1.04,2,1.88,-.96,1.25,1.6,-.04,.3};  
 vec3 A = 3[p], B = 3 \* (2[p] - 3[p]);  
 vec3 C = 3\*1[p] - 6\*2[p] + 3\*3[p], D = 0[p]-3\*1[p]+3\*2[p]-3[p];  
 size\_t samples = 50;  
 double scale = 1. / samples;  
 GLfloat linewidth; glGetFloatv(GL\_LINE\_WIDTH, &linewidth);  
 glLineWidth(4.f); glBegin(GL\_LINE\_STRIP);  
 for (size\_t i = 0; i <= samples; i++) {  
 double t = scale \* i; glVertex3dv(A + B\*t + C\*t\*t + D\*t\*t\*t);  
 }  
 glEnd(); glLineWidth(linewidth); glBegin(GL\_LINE\_STRIP);  
 glVertex3dv(0[p]); glVertex3dv(1[p]); glVertex3dv(2[p]);  
 glVertex3dv(3[p]); glEnd();  
}  
void l3tr\_6() { l3tr\_6\_1(); l3tr\_6\_2(); l3tr\_6\_3(); }

