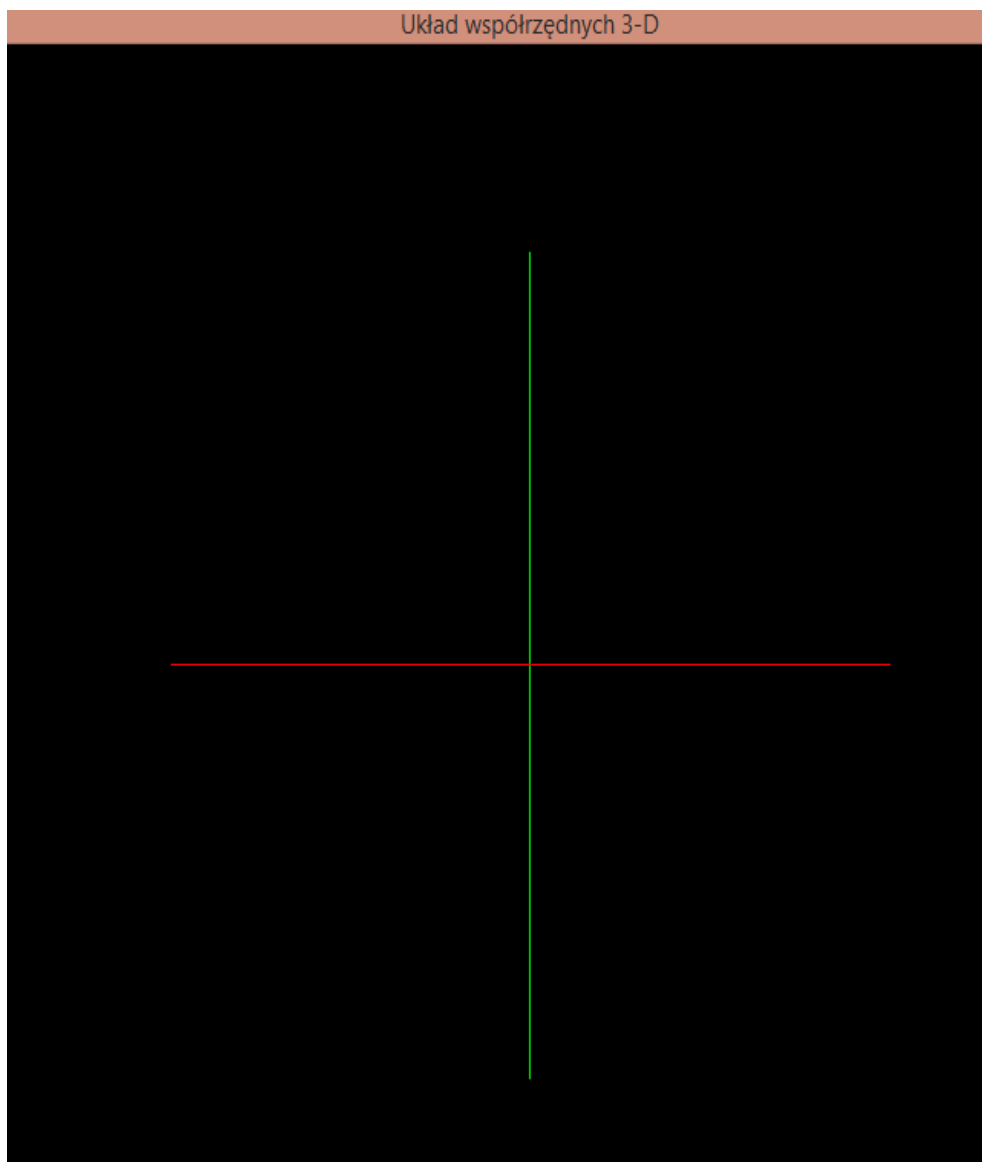


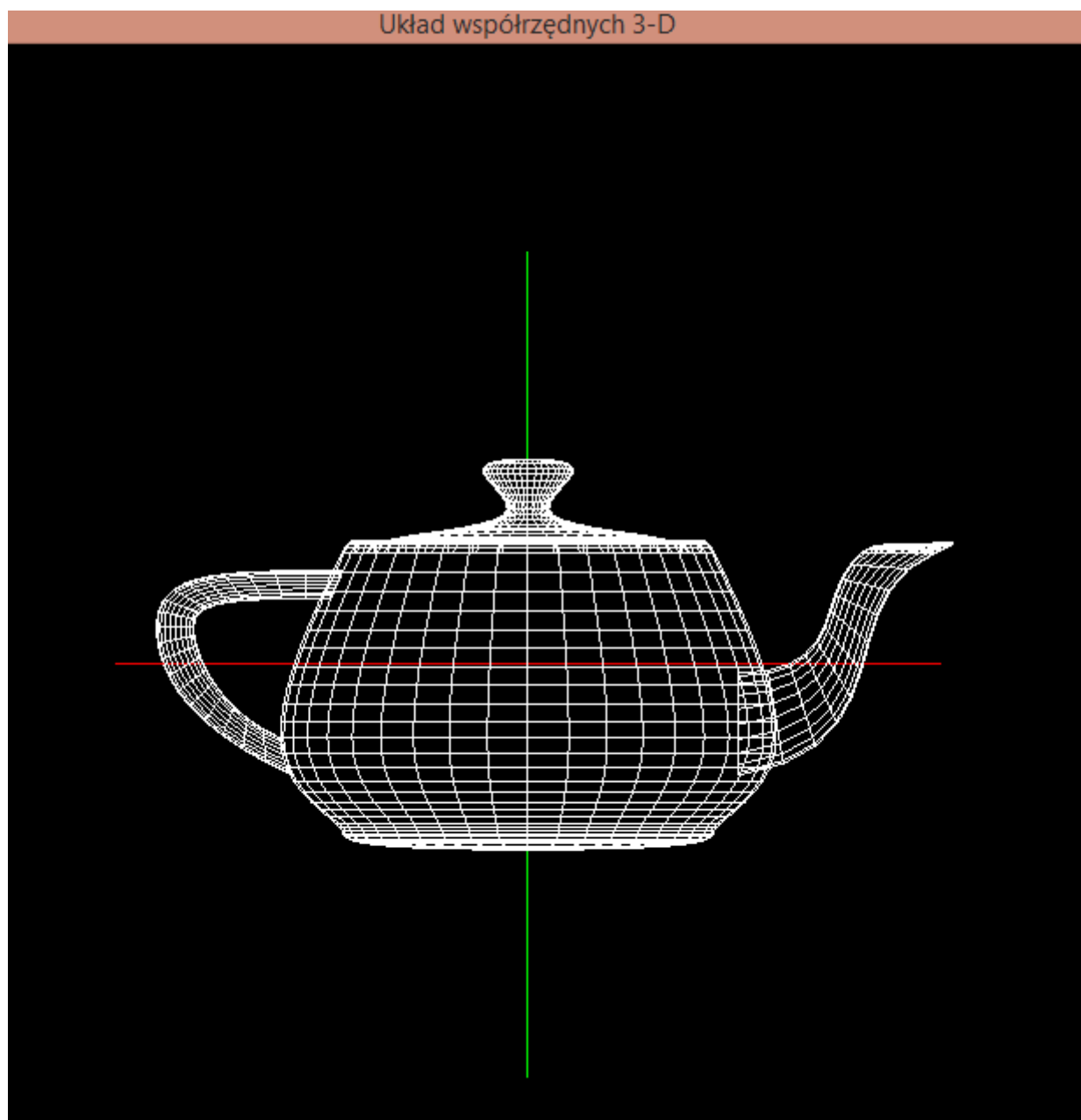
**Sprawozdanie z ćwiczenia 3**  
**OpenGL - modelowanie obiektów 3-D**  
**Maksymilian Iwanow**  
**209946**  
**Poniedziałek 10.15**

**Celem** ćwiczenia było wprowadzenie w zagadnienia modelowania i wizualizacji scen 3D z wykorzystaniem biblioteki OpenGL i rozszerzeniem GLUT. Początkowo zostały wykonane 3 proste zadania zgodnie ze wskazówkami ze strony internetowej kursu Grafika Komputerowa:

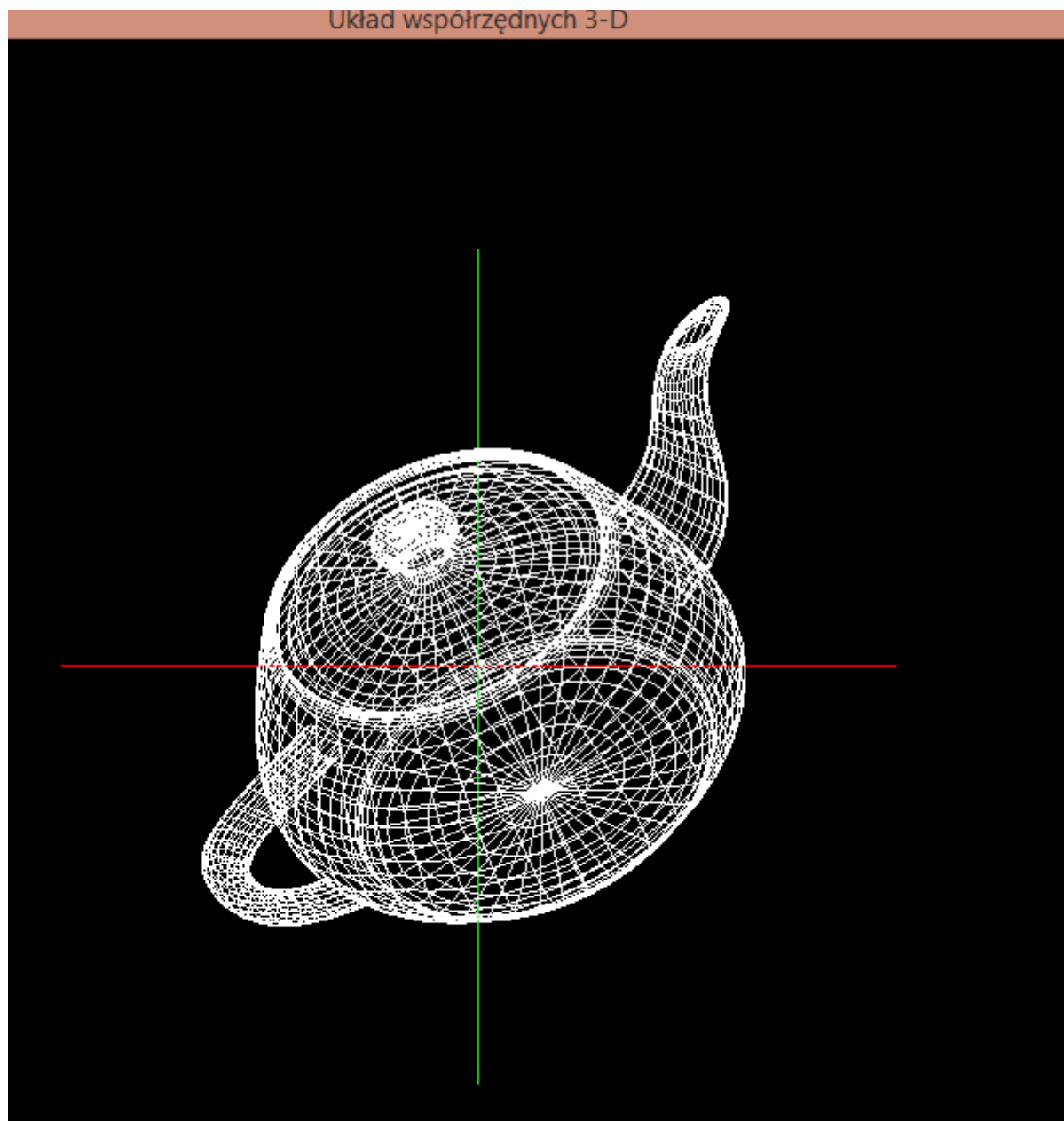
- narysowanie układu współrzędnych



- narysowanie czajnika do herbaty (z gotowego modelu)



- wykonanie transformacji na modelu czajnika (np. obrotu)



**Modelowanym przez nas obiektem było jajko, opisane następującymi równaniami parametrycznymi :**

$$x(u,v) = (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \cos(\pi v)$$

$$y(u,v) = 160u^4 - 320u^3 + 160u^2$$

$$z(u,v) = (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \sin(\pi v)$$

$$0 \leq u \leq 1$$

$$0 \leq v \leq 1$$

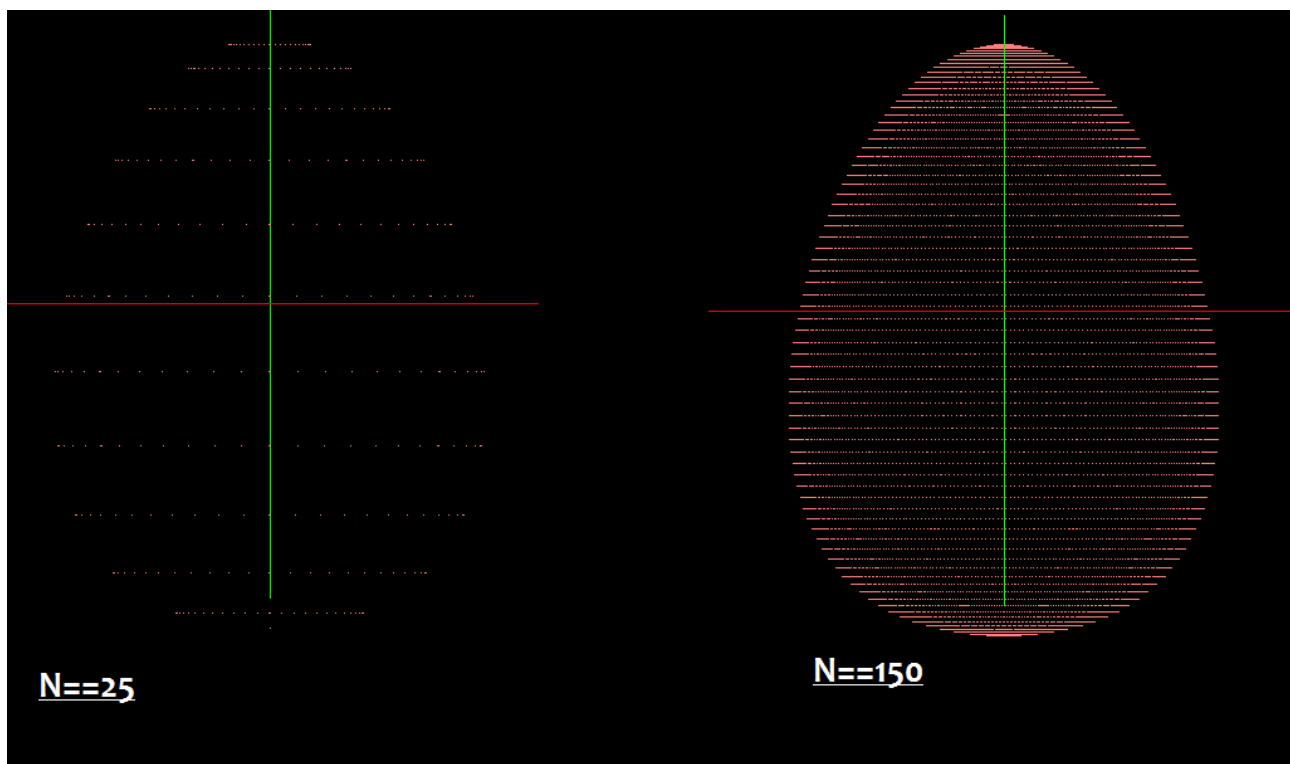
Wyliczanie punktów jajka w przestrzeni oraz wyświetlanie zbioru punktów.  
Za to odpowiedzialna była funkcja jajo1(), o następującym kodzie:

```
//tworzymy u i v
float u,v;
//liczba N - na ile zostanie podzielony
int N = 25;
float tabx[150][150]; // wsp. ptk. w przestrzeni dla x // NxN
float taby[150][150]; // wsp. ptk. w przestrzeni dla y
float tabz[150][150]; // wsp. ptk. w przestrzeni dla z
//podwójna pętla for - wypełniamy tablice współrzędnymi punktów, każda współrzędna (x,y,z) znajdzie się w innej tablicy na tej samej pozycji
//przechodzimy po wszystkich punktach wyznaczających dziedzinę parametryczną
for(int i=0; i< N; i++)
{
    u = (float)i/(N-1);
    for(int j=0; j<N; j++)
    {
        v = (float)j/(N-1);
        tabx[i][j] = (-90 * potg(u,5) + 225 * potg(u,4) - 270 * potg(u,3) + 180 * potg(u,2) - 45 * u) * cos(pi * v) ;

        taby[i][j] = 160 * potg(u,4) - 320 * potg(u,3) + 160 * potg(u,2);

        tabz[i][j] = (-90 * potg(u,5) + 225 * potg(u,4) - 270 * potg(u,3) + 180 * potg(u,2) - 45 * u) * sin(pi * v);
    }
}
glColor3f(1.0f, 0.5f, 0.5f);
glBegin(GL_POINTS);
for(int i=0; i<N; i++)
{
    for(int j=0; j<N; j++)
    {
        glVertex3f(tabx[i][j],taby[i][j]-5.5,tabz[i][j]+3); //przesuwamy aby całe jajko było widoczne
    }
}
glEnd();
```

*Kod jajo1()*

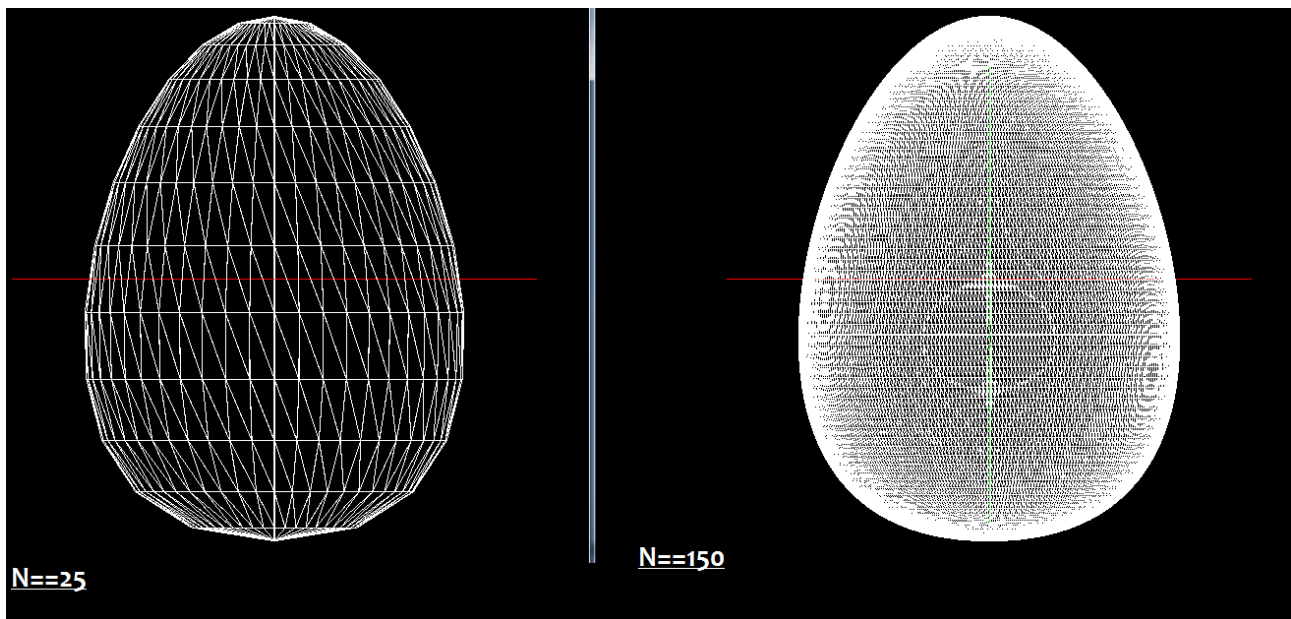


Wynik funkcji jajo1() dla  $N=25$  oraz  $N=150$ .

Jajko przedstawione jako siatka.  
Za to odpowiedzialna była funkcja jajo2(), o następującym kodzie:

```
float u = 0, v=0;
int N = 25;
float tab[150][150][3];
for(int i=0; i<N; ++i)
{
    u = (float)i/(N-1);
    for(int j = 0; j<N; ++j)
    {
        v = (float)j/(N-1);
        tab[i][j][0] = (-90*potg(u, 5)+225*potg(u, 4)-270*potg(u, 3)+180*potg(u, 2)-45*u)*cos(pi*v);
        tab[i][j][1] = 160*potg(u, 4)-320*potg(u, 3)+160*potg(u, 2)-5.0;
        tab[i][j][2] = (-90*potg(u, 5)+225*potg(u, 4)-270*potg(u, 3)+180*potg(u, 2)-45*u)*sin(pi*v);
    }
}
glColor3f(1.0f, 1.0f, 1.0f);
glBegin(GL_LINES);
for(int i=0; i<N-1; i++)
{
    for(int j=0; j<N-1; j++)
    {
        glVertex3fv(tab[i][j]);
        glVertex3fv(tab[i+1][j]);
        glVertex3fv(tab[i][j]);
        glVertex3fv(tab[i][j+1]);
        glVertex3fv(tab[i+1][j]);
        glVertex3fv(tab[i][j+1]);
    }
}
glEnd();
```

Kod jajo2()



Wynik funkcji `jajo2()` dla  $N=25$  oraz  $N=150$ .

Funkcja wprowadzająca jajko w ruch została napisana zgodnie ze wskazówkami ze strony <http://www.zsk.ict.pwr.wroc.pl> a więc następująco:

```

        theta[0] -= 0.5;
    if( theta[0] > 360.0 ) theta[0] -= 360.0;

        theta[1] -= 0.5;
    if( theta[1] > 360.0 ) theta[1] -= 360.0;

        theta[2] -= 0.5;
    if( theta[2] > 360.0 ) theta[2] -= 360.0;

```

```

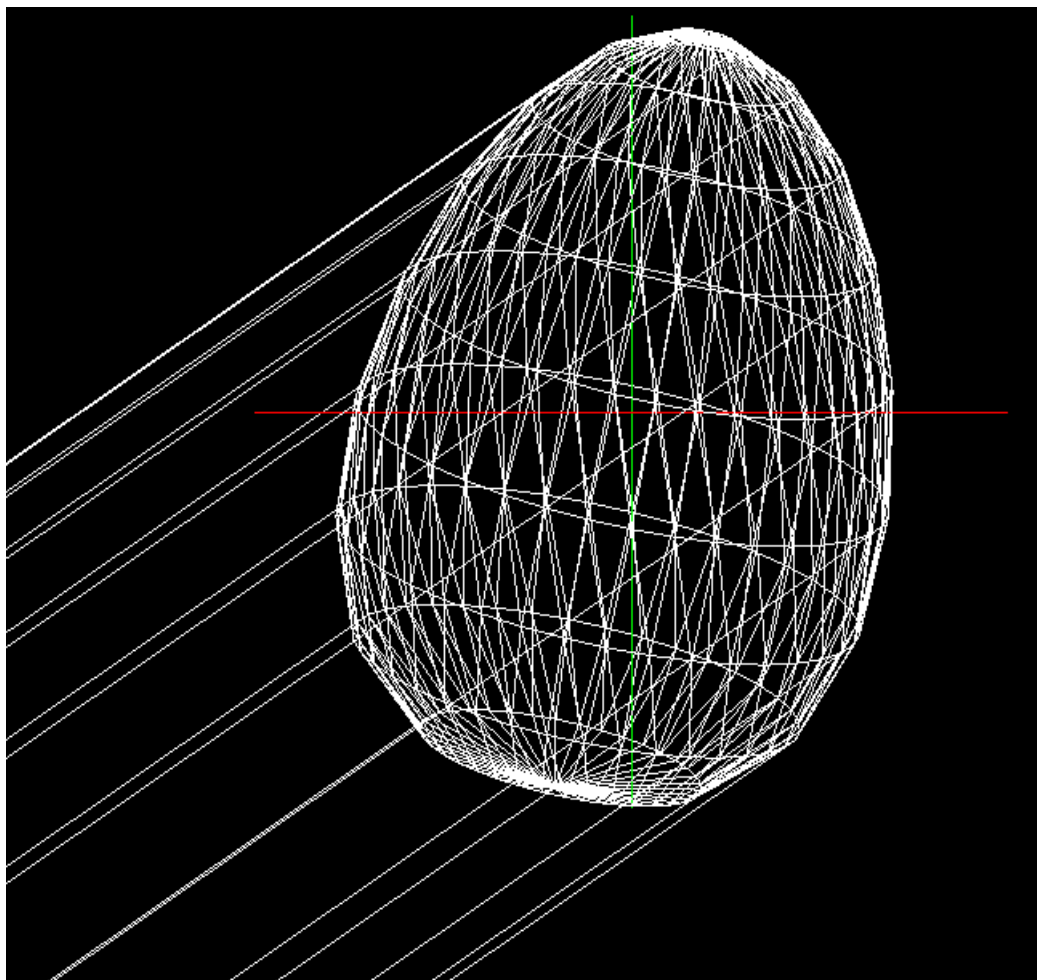
glutPostRedisplay(); //odświeżenie zawartości aktualnego okna

```

gdzie 'theta[]' to zmienna globalna przechowująca informacje o aktualnym kącie obrotu jajka.

### Podsumowanie, wnioski, trudności:

1. Największym problemem było znalezienie przyczyny złego wyświetlania w funkcji jajo2(). Przed znalezieniem błędu jajko prezentowało się jak na screenie poniżej. Okazało się, że wystarczyło zmniejszyć o ' 1 ' liczbę powtórzeń pętli ' for ' odpowiedzialnych za rysowanie ( zamiast  $[ i < N \text{ oraz } j < N ]$  –  $[ i < N-1 \text{ oraz } j < N-1 ]$  ).



2. Drugim największym problemem było zapisanie równań parametrycznych jajka. Początkowo ( dla funkcji jajo1() ) tworzone były 3 tablice dwuwymiarowe dla przechowywania współrzędnych, dla następnych funkcji zostały one zastąpione jedną tablicą trójwymiarową – zamiast 3 tablic  $N \times N$  jedna  $N \times N \times 3$ . Ułatwiło to zapisywanie funkcji wyświetlających.
3. Została napisana funkcja pomocnicza potg(float x, float y) podnosząca x do potęgi y – wykorzystywana była ona w zapisywaniu równań parametrycznych.