Magnetic Field Lines CMOD Builder

Various Field Definitions :

```
theta1a := 70 Pi / 180
phi1a := -80 Pi / 180
theta2a := 74 Pi / 180
phi2a := 160 Pi / 180

theta1b := 90 Pi / 180
phi1b := 180 Pi / 180
theta2b := 74 Pi / 180
phi2b := 150 Pi / 180

Mu1 := {
          Sin[theta1a] Cos[phi1a],
          Sin[theta1a] Sin[phi1a],
          Cos[theta1a]
      }

Mu2 := {
          Sin[theta2a] Cos[phi2a],
          Sin[theta2a] Sin[phi2a],
          Cos[theta2a]
      }

a1 := 0.88 {
           Sin[theta1b] Cos[phi1b],
           Sin[theta1b] Sin[phi1b],
           Cos[theta1b]
       }

a2 := 0.88 {
           Sin[theta2b] Cos[phi2b],
           Sin[theta2b] Sin[phi2b],
           Cos[theta2b]
       }

Pos1[x_, y_, z_] := {x, y, z} - a1
r1[x_, y_, z_] := Sqrt[Pos1[x, y, z].Pos1[x, y, z]]

Pos2[x_, y_, z_] := {x, y, z} - a2
r2[x_, y_, z_] := Sqrt[Pos2[x, y, z].Pos2[x, y, z]]

Bdip1[x_, y_, z_] := 3 (Mu1.Pos1[x, y, z]) Pos1[x, y, z] / r1[x, y, z]^5 - Mu1 / r1[x, y, z]^3

Bdip2[x_, y_, z_] := 3 (Mu2.Pos2[x, y, z]) Pos2[x, y, z] / r2[x, y, z]^5 - Mu2 / r2[x, y, z]^3

Bmono1[x_, y_, z_] := Pos1[x, y, z] / r1[x, y, z]^3

Bmono2[x_, y_, z_] := - Pos2[x, y, z] / r2[x, y, z]^3

Btot[x_, y_, z_] := 1 (Bdip1[x, y, z] + Bdip2[x, y, z]) + 0 (Bmono1[x, y, z] + Bmono2[x, y, z])

B[x_, y_, z_] := Sqrt[Btot[x, y, z].Btot[x, y, z]]

Bx[x_, y_, z_] := {1, 0, 0}.Btot[x, y, z]
By[x_, y_, z_] := {0, 1, 0}.Btot[x, y, z]
Bz[x_, y_, z_] := {0, 0, 1}.Btot[x, y, z]
```

Magnetic Field Lines :

```
theta01[k_] := (45 + 2 k) Pi / 180
theta02[k_] := (45 + 2 k) Pi / 180

NCurves := 24
phi[n_] := n 2 Pi / NCurves

StartRadius := 1
theta1c := 70 Pi / 180
phi1c := -80 Pi / 180
theta2c := 74 Pi / 180
phi2c := 160 Pi / 180

Ra[u_] := {{Cos[u], 0, Sin[u]}, {0, 1, 0}, {- Sin[u], 0, Cos[u]}}
Rb[v_] := {{Cos[v], - Sin[v], 0}, {Sin[v], Cos[v], 0}, {0, 0, 1}}

P1[n_, k_] := StartRadius Rb[phi1c].Ra[theta1c].{Sin[theta01[k]] Cos[phi[n]], Sin[theta01[k]] Sin[phi[n]], Cos[theta01[k]]}
P2[n_, k_] := StartRadius Rb[phi2c].Ra[theta2c].{Sin[theta02[k]] Cos[phi[n]], Sin[theta02[k]] Sin[phi[n]], Cos[theta02[k]]}

x10[n_, k_] := {1, 0, 0}.(P1[n, k] + a1)
y10[n_, k_] := {0, 1, 0}.(P1[n, k] + a1)
z10[n_, k_] := {0, 0, 1}.(P1[n, k] + a1)

x20[n_, k_] := {1, 0, 0}.(P2[n, k] + a2)
y20[n_, k_] := {0, 1, 0}.(P2[n, k] + a2)
z20[n_, k_] := {0, 0, 1}.(P2[n, k] + a2)

FieldCurves1[n_, k_] := NDSolve[
  {
   x'[s] == Bx[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],
   y'[s] == By[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],
   z'[s] == Bz[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],

   x[0] == x10[n, k],
   y[0] == y10[n, k],
   z[0] == z10[n, k]

  }, {x, y, z}, {s, -5000, 5000}, Method -> RungeKutta, MaxSteps → 10000000, StoppingTest → (

    r1[x[s], y[s], z[s]] < 0.05 || r2[x[s], y[s], z[s]] < 0.05
   )]

FieldCurves2[n_, k_] := NDSolve[
  {
   x'[s] == Bx[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],
   y'[s] == By[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],
   z'[s] == Bz[x[s], y[s], z[s]] / B[x[s], y[s], z[s]],

   x[0] == x20[n, k],
   y[0] == y20[n, k],
   z[0] == z20[n, k]

  }, {x, y, z}, {s, -5000, 5000}, Method -> RungeKutta, MaxSteps → 10000000, StoppingTest → (

    r2[x[s], y[s], z[s]] < 0.05 || r1[x[s], y[s], z[s]] < 0.05
   )]

Do[Table[{FieldCurves1[n, k], FieldCurves2[n, k]}, {n, 1, NCurves}], {k, -5, 5}]
```

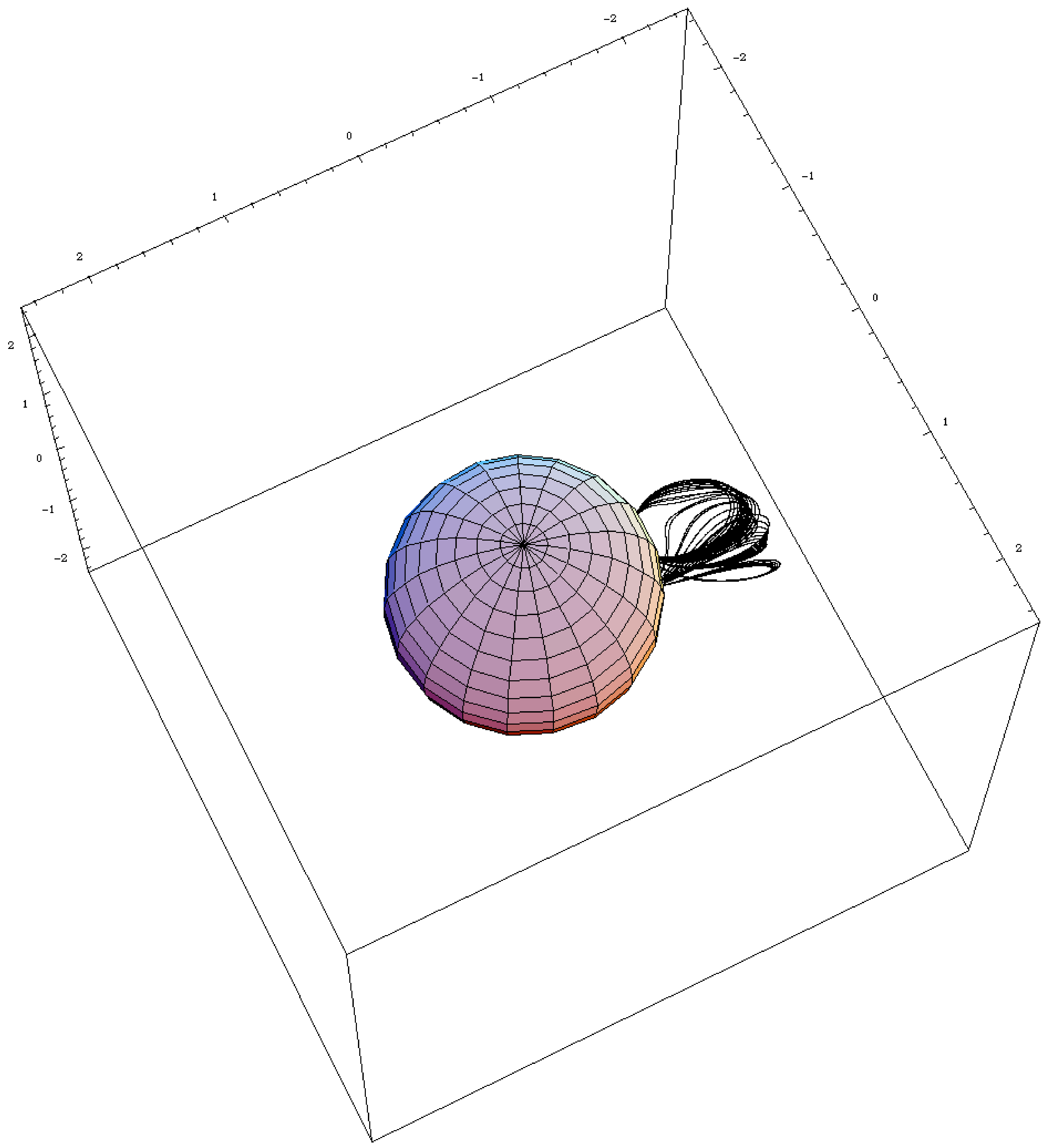
CMOD Builder of Field Lines :

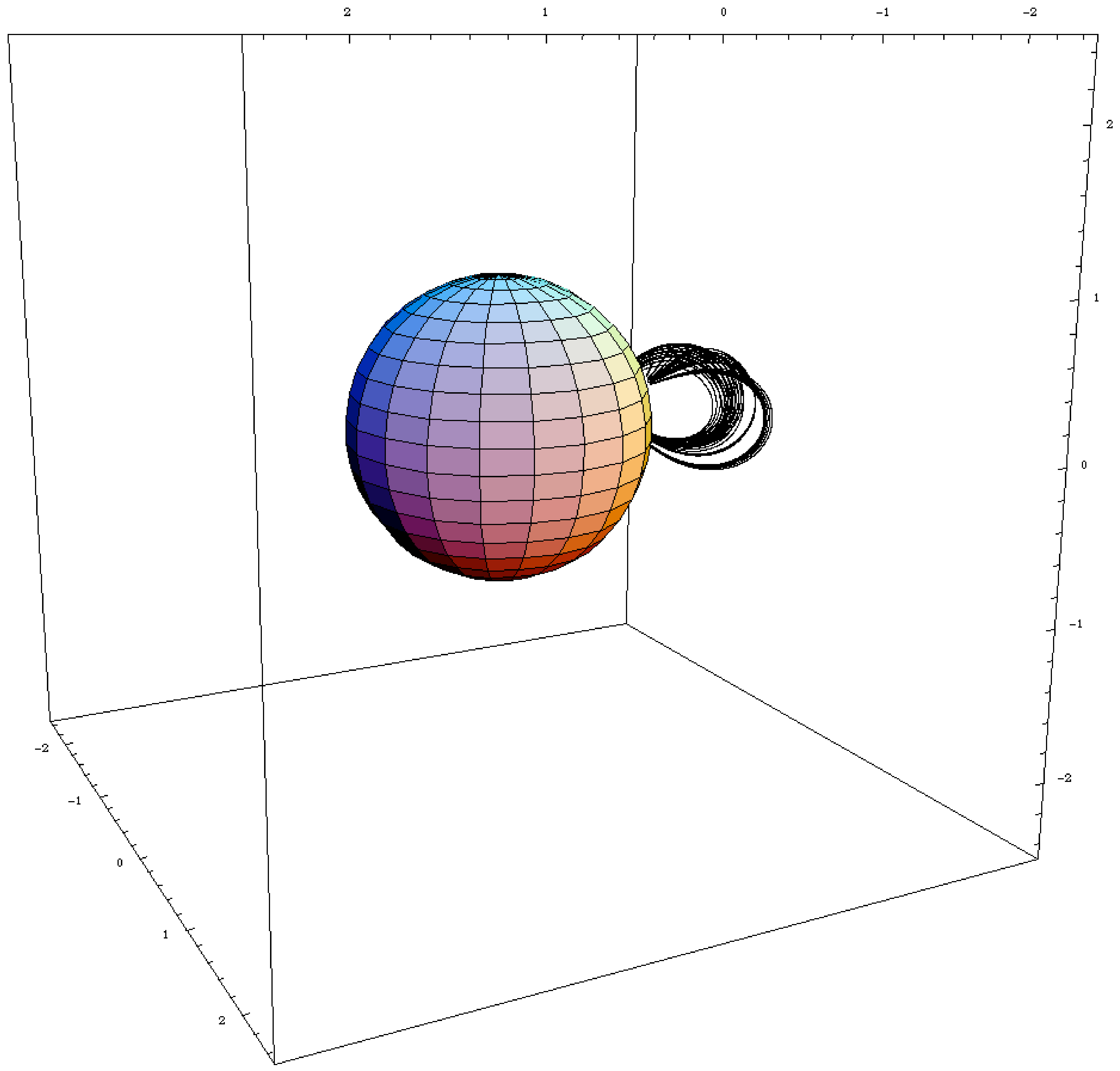
```
Smin1[n_, k_] := (x /. FieldCurves1[n, k])[[1]][[1]][[1]][[1]]
Smax1[n_, k_] := (x /. FieldCurves1[n, k])[[1]][[1]][[1]][[2]]

Smin2[n_, k_] := (x /. FieldCurves2[n, k])[[1]][[1]][[1]][[1]]
Smax2[n_, k_] := (x /. FieldCurves2[n, k])[[1]][[1]][[1]][[2]]

NPoints1[n_, k_] := IntegerPart[6]

CurvePrecision1[n_, k_] := (1 / 50) (Smax1[n, k] - Smin1[n, k]) / NPoints1[n, k]

Coords1[n_, k_, u_] := SetPrecision[Flatten[
                     Table[Evaluate[{x[s], y[s], z[s]} /. FieldCurves1[n, k],
     {s, Smin1[n, k] + ((u - 1) / 50) (Smax1[n, k] - Smin1[n, k]), Smin1[n, k] + (u / 50) (Smax1[n, k] - Smin1[n, k]), CurvePrecision1[n, k]}
    ]], 1], 10]

Joints1[n_, k_] := Table[x, {x, 0, NPoints1[n, k]}]

MeshHeader1[n_, k_] := StringForm[StringJoin[

       "mesh\r  vertexdesc position f3 end_vertexdesc\r\rvertices  ", ToString[NPoints1[n, k] + 1]

       ]]

LinesHeader1[n_, k_, u_] := StringForm[StringJoin["\rlinestrip ", ToString[If[u > 25, 51 - u, u]], "  ", ToString[NPoints1[n, k] + 1]]]

EndMesh := StringForm["\rend_mesh\r\r"]

OneFieldLineData1[n_, k_, u_] := Join[ Prepend[Coords1[n, k, u], MeshHeader1[n, k]], Append[Prepend[Joints1[n, k], LinesHeader1[n, k, u]], EndMesh]]

AllLinesData1[u_] := Join[Flatten[Table[OneFieldLineData1[n, k, u], {n, 1, 5}, {k, -3, 3}], 2]]

AllFieldLines := Do[Export[StringJoin["Mathematica/ALLFieldLines_", ToString[u], ".txt"], AllLinesData1[u], "Table"], {u, 1, 50}]
```

```
AllFieldLines          (* This will export all the field lines into 50 "curve cut" files *)
```

Graphical inspector of a complete familly of field lines :

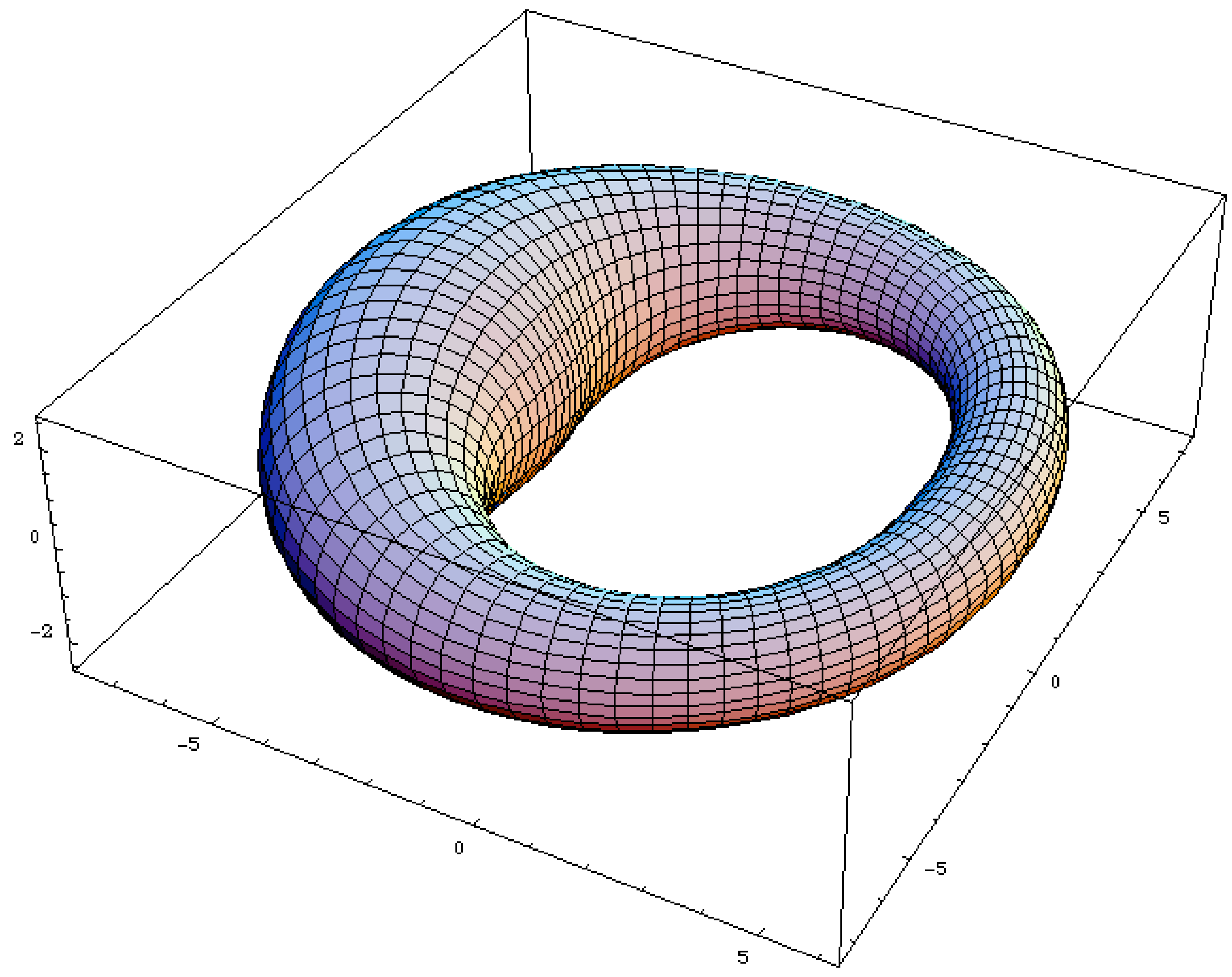
```
size := 1.8

Graph1[n_, k_] := ParametricPlot3D[
              Evaluate[{x[s], y[s], z[s]} /. FieldCurves1[n, k]],
  {s, Smin1[n, k], Smax1[n, k]}, PlotPoints → 1000, ViewPoint → {0, -1, 0.3}]

Graph2[n_, k_] := ParametricPlot3D[
              Evaluate[{x[s], y[s], z[s]} /. FieldCurves2[n, k]],
  {s, Smin2[n, k], Smax2[n, k]}, PlotPoints → 1000, ViewPoint → {0, -1, 0.3}]

Ball := ParametricPlot3D[{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]}, {u, 0, Pi}, {p, 0, 2 Pi}]

Ball1 := ParametricPlot3D[
      {
          {1, 0, 0}.(0.05 Rb[phi1a].Ra[theta1a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a1),
          {0, 1, 0}.(0.05 Rb[phi1a].Ra[theta1a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a1),
          {0, 0, 1}.(0.05 Rb[phi1a].Ra[theta1a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a1)
       },
      {u, 0, Pi}, {p, 0, 2 Pi}]

Ball2 := ParametricPlot3D[
      {
          {1, 0, 0}.(0.05 Rb[phi2a].Ra[theta2a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a2),
          {0, 1, 0}.(0.05 Rb[phi2a].Ra[theta2a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a2),
          {0, 0, 1}.(0.05 Rb[phi2a].Ra[theta2a].{Sin[u] Cos[p], Sin[u] Sin[p], Cos[u]} + a2)
       },
      {u, 0, Pi}, {p, 0, 2 Pi}]

AllGraphs = Show[Table[Table[Graph2[n, k], {n, 1, 5}], {k, -3, 3}], Ball,
  Ball1, Ball2, ViewPoint → {2, 0.7, 1}, PlotRange → {{-size, size}, {-size, size}, {-size, size}}]

Show[AllGraphs, ViewPoint → {1.0, 2.0, 6.0}, PlotRange → {{-size, size}, {-size, size}, {-size, size}}]
Show[AllGraphs, ViewPoint → {1.0, 2.0, 0.5}, PlotRange → {{-size, size}, {-size, size}, {-size, size}}]
Show[AllGraphs, ViewPoint → {-1.0, -2, 0.5}, PlotRange → {{-size, size}, {-size, size}, {-size, size}}]
```



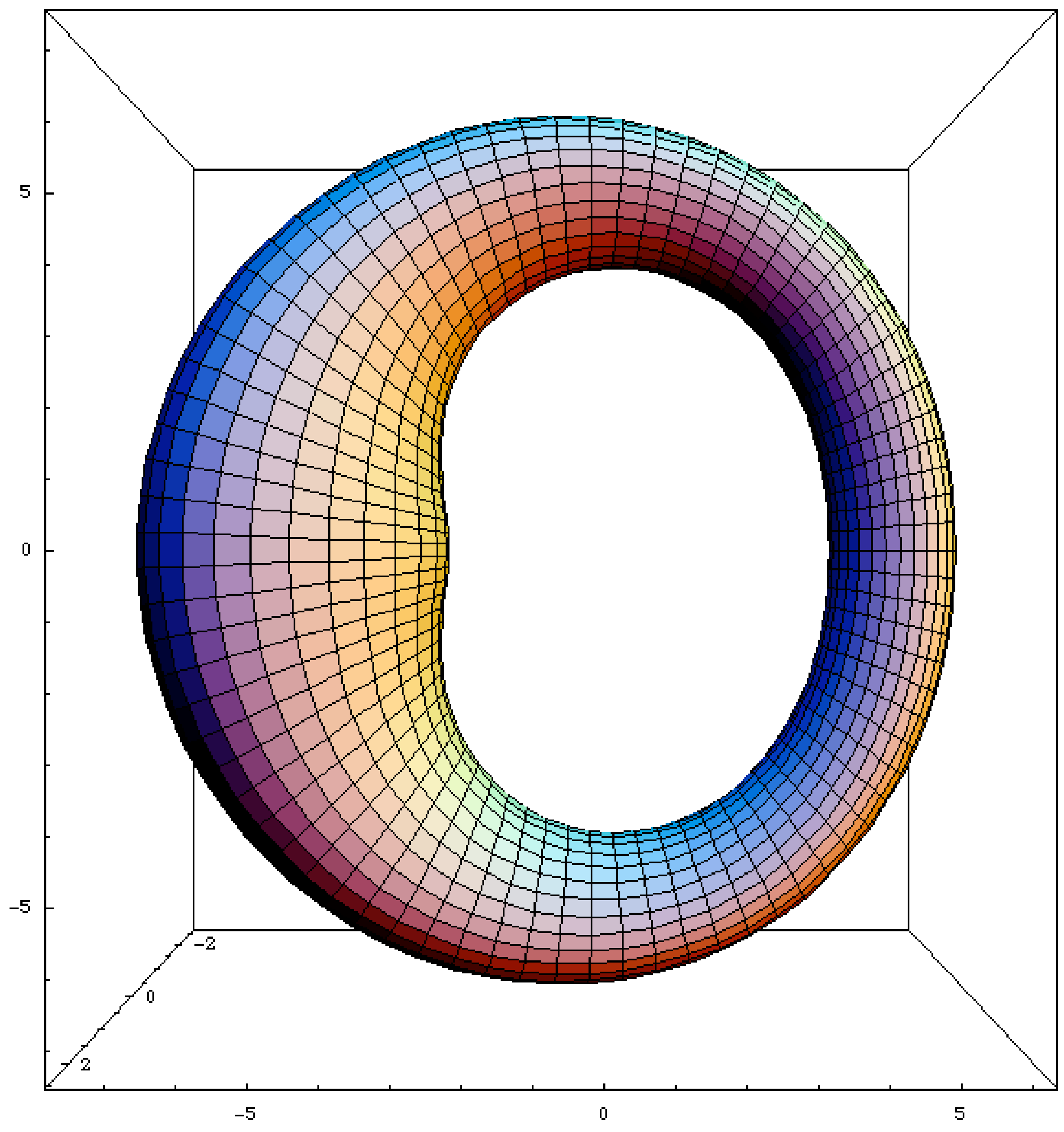


Various Curves Experimentations :

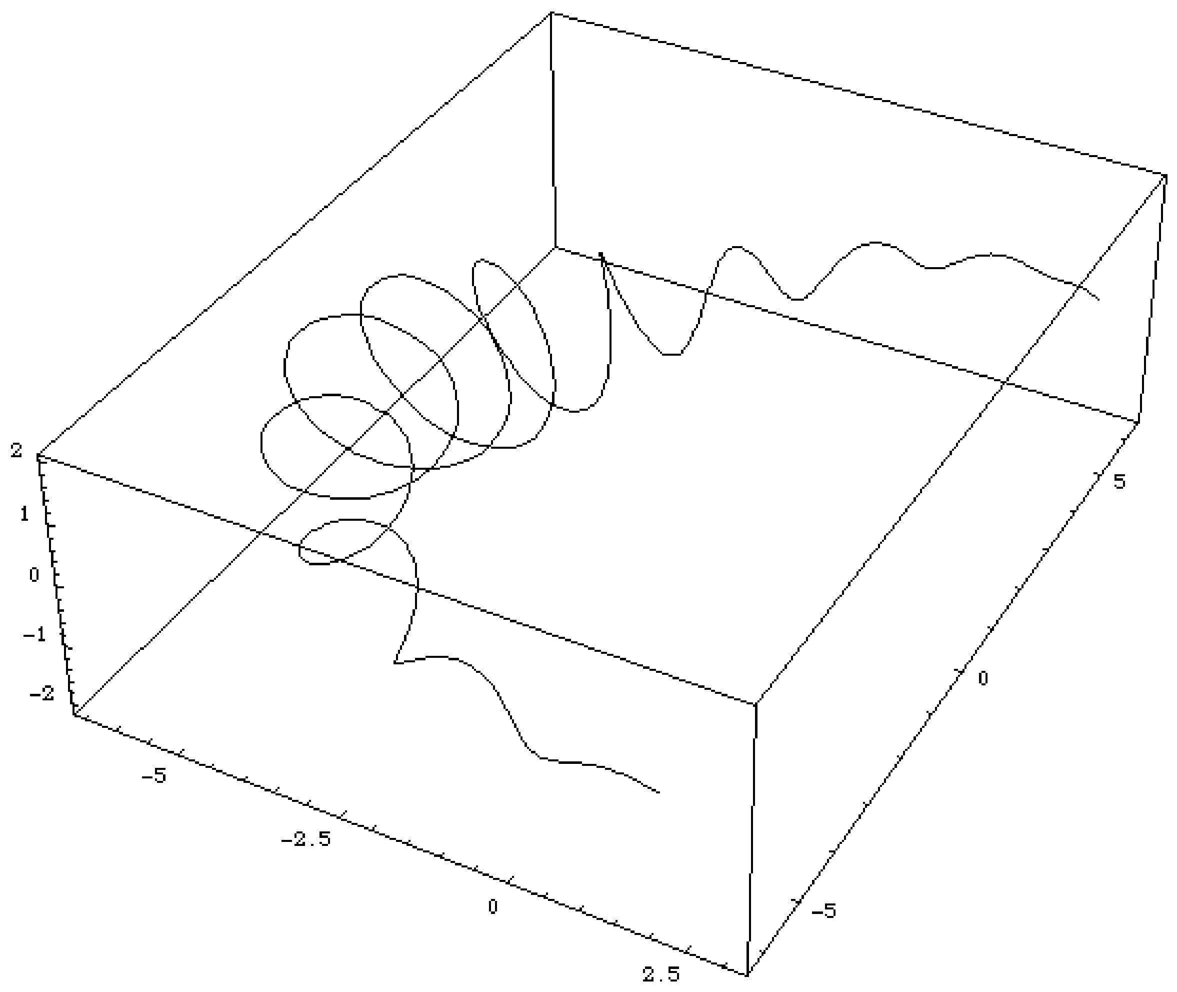
```
R[p_] := 5 + Sin[p]^2
r[p_] := 1 + 1.5 Sin[0.5 p]^6

Tube = ParametricPlot3D[{
                         (R[p] - r[p] Sin[t]) Cos[p],
                         (R[p] - r[p] Sin[t]) Sin[p],
                         r[p] Cos[t]

                    }, {t, 0, 2 Pi}, {p, 0, 2 Pi}, PlotPoints → {30, 80}]
```



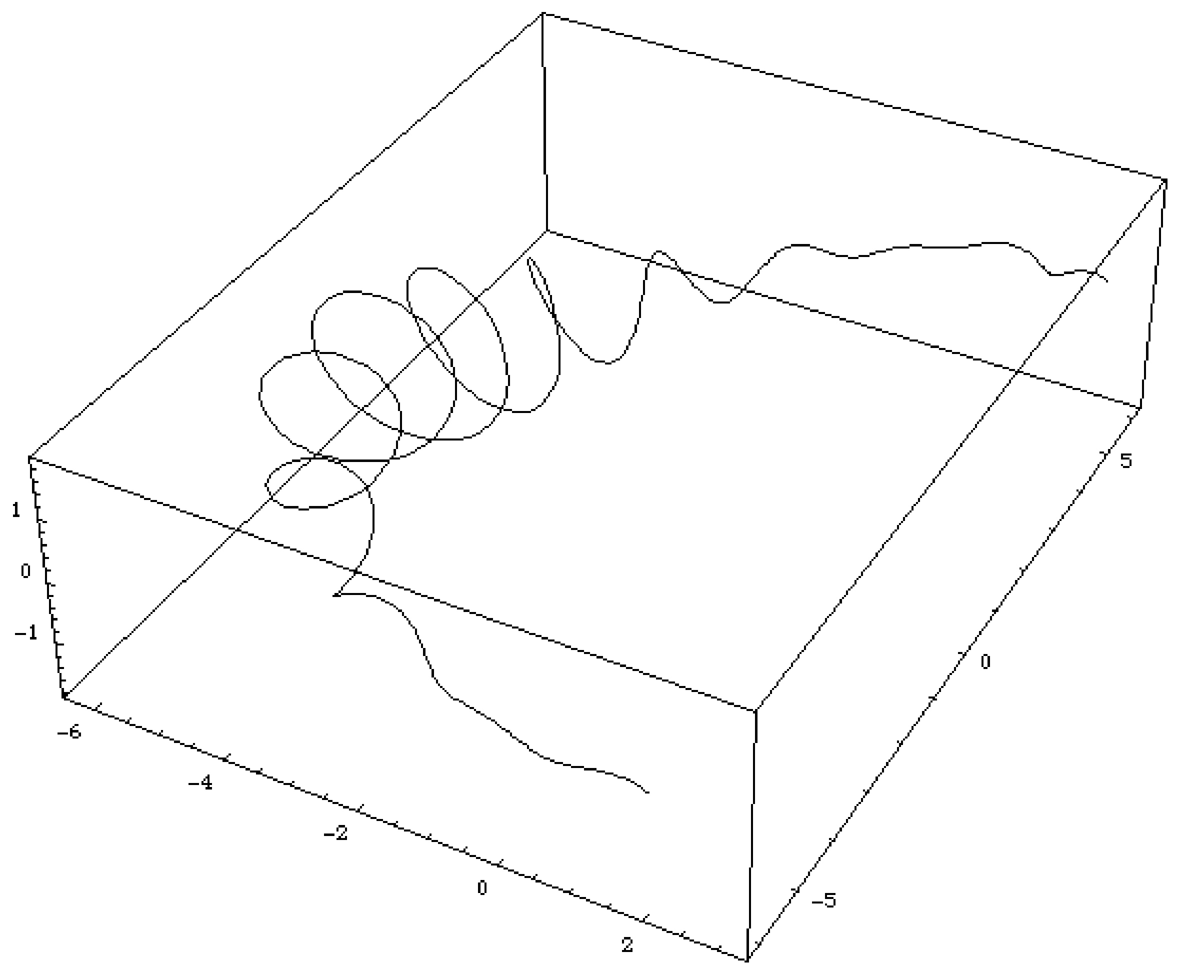

```
Show[Tube, ViewPoint → {0, 0, 1}]
```



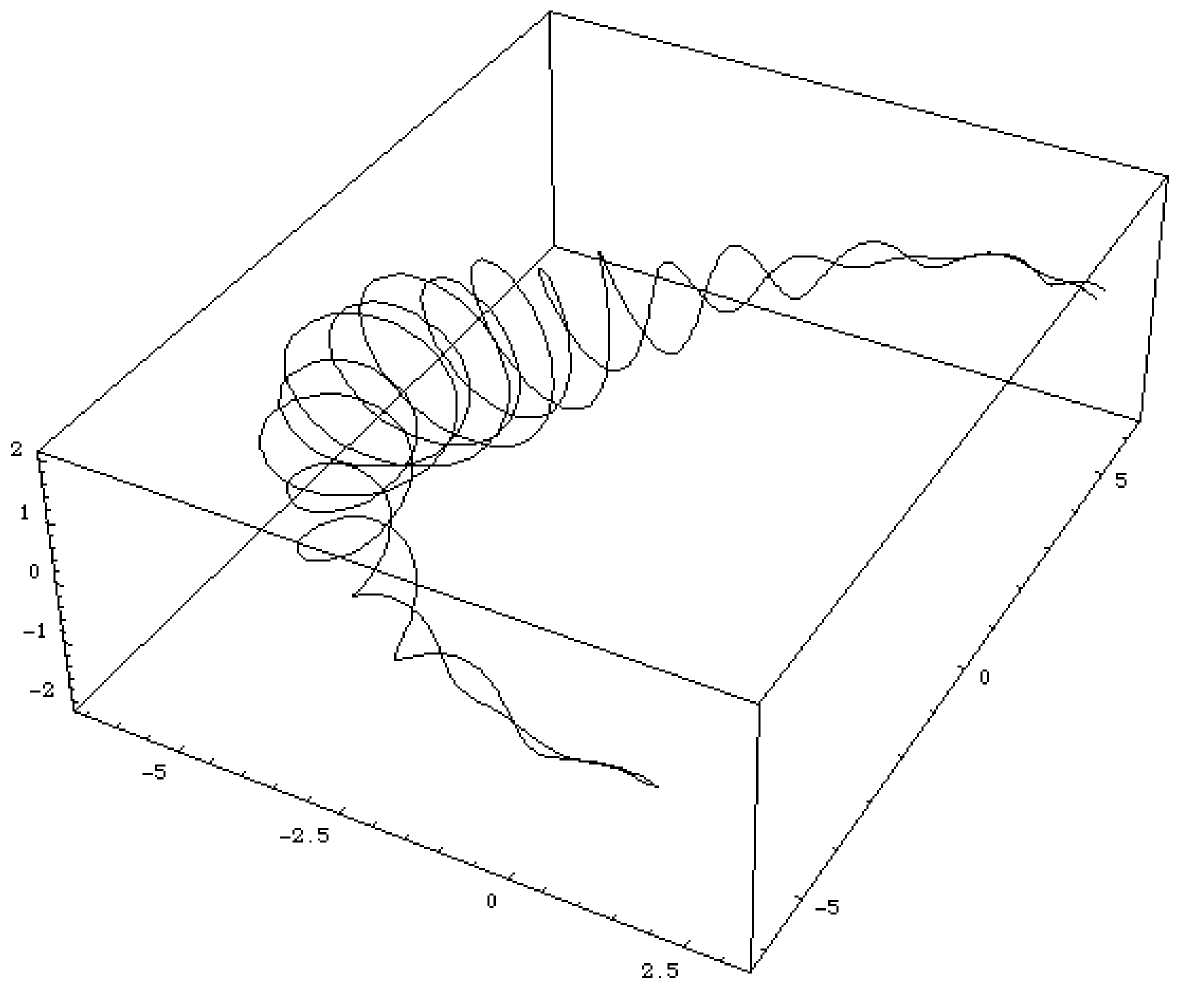
```
Ligne1 = ParametricPlot3D[{
                        (R[u] - (r[u] - 1) Sin[18 u]) Cos[1 u],
                        (R[u] - (r[u] - 1) Sin[18 u]) Sin[1 u],
                        (r[u] - 1) Cos[18 u] + Sin[2 u]

                }, {u, 1, 5}, PlotPoints → 500]
```



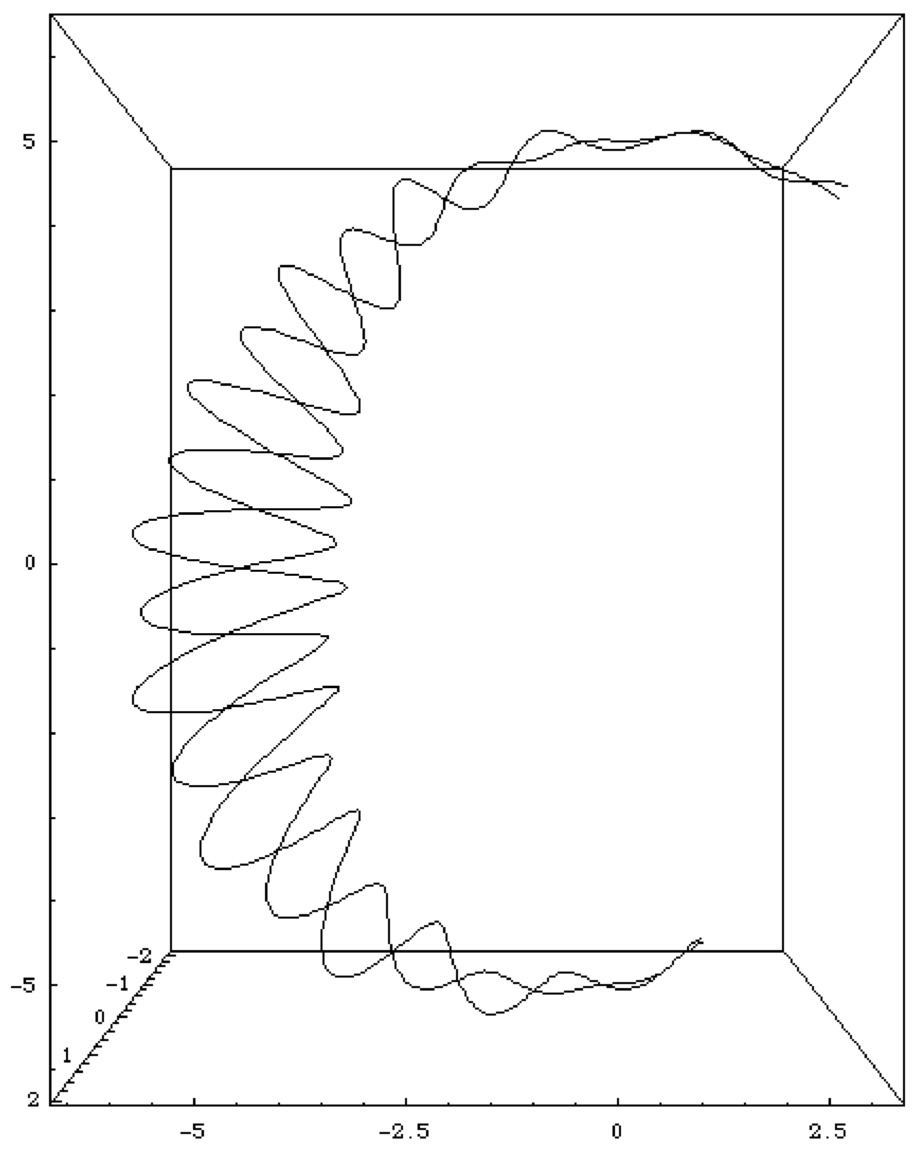

```
Ligne2 = ParametricPlot3D[{
                        (R[u] - (r[u] - 1.2) Sin[18 u + Pi]) Cos[u],
                        (R[u] - (r[u] - 1.2) Sin[18 u + Pi]) Sin[u],
                        (r[u] - 1.2) Cos[18 u + Pi] + Sin[2 u]

                }, {u, 1, 5}, PlotPoints → 500]
```




```
Show[Ligne1, Ligne2]
```




```
Show[Ligne1, Ligne2, ViewPoint → {0, 0, 1}]
```



CMOD Builder of a Curve :

```
R[t_] := 5 + Sin[t]^2
r[t_] := 1 + 1.5 Sin[0.5 t]^6

Curve1[t_] := {
              (R[t] - (r[t] - 1) Sin[18 t]) Cos[1 t] - 100,
              (R[t] - (r[t] - 1) Sin[18 t]) Sin[1 t],
              (r[t] - 1) Cos[18 t] + Sin[2 t]
              }

Curve2[t_] := {
              (R[t] - (r[t] - 1.2) Sin[18 t + Pi]) Cos[t] - 100,
              (R[t] - (r[t] - 1.2) Sin[18 t + Pi]) Sin[t],
              (r[t] - 1.2) Cos[18 t + Pi] + Sin[2 t]
              }

NPoints := 16
CurvePrecision := 0.005

CurveCoords[u_] := Flatten[Table[SetPrecision[Curve1[t], 9], {t, 1 + ((u - 1) / 50) 4, 1 + (u / 50) 4, CurvePrecision}], 0]

CurveJoints := Table[x, {x, 0, NPoints}]

MeshHeader := StringForm[StringJoin[

       "mesh\r  vertexdesc position f3 end_vertexdesc\r\rvertices  ", ToString[NPoints + 1], "\r"

       ]]

LinesHeader[u_] := StringForm[StringJoin["\rlinestrip ", ToString[If[u > 25, 51 - u, u]], "  ", ToString[NPoints + 1]]]

EndMesh := StringForm["\rend_mesh\r\r"]

CurveData[u_] := Join[ Prepend[CurveCoords[u], MeshHeader], Append[Prepend[CurveJoints, LinesHeader[u]], EndMesh]]

CurveExport := Do[Export[StringJoin["Mathematica/Curve_", ToString[u], ".txt"], CurveData[u], "Table"], {u, 1, 50}]
```

```
CurveExport      (* This will export the curve into 50 "curve cut" files *)
```