Curves and frames on a Torus

Consider a general torus:

In[182]:=
```
torus[a_: 1, b_: 2, c_: 1][u_, v_] :=
 {Cos[v] (b + a Cos[u]), Sin[v] (b + a Cos[u]), c * Sin[u]}
```

In[183]:=
```
torus[a, b, c][u, v]
```

Out[183]=
```
{(b + a Cos[u]) Cos[v], (b + a Cos[u]) Sin[v], c Sin[u]}
```

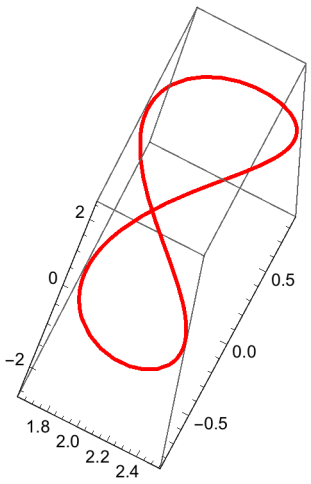
In[184]:=
```
torus[][u, v]
```

Out[184]=
```
{(2 + Cos[u]) Cos[v], (2 + Cos[u]) Sin[v], Sin[u]}
```

Let' s take any plane curve. I will use a circle {cos(*t*), sin(*t*)} but any parametrised curve can be used in the same way. We can plot this curve in space or on the torus by just substitution its parametirsation in this way:

In[185]:=
```
cv1 = ParametricPlot3D[
   torus[][Cos[t], Sin[t]], {t, 0, 2 Pi}, ColorFunction → (Red &)]
```

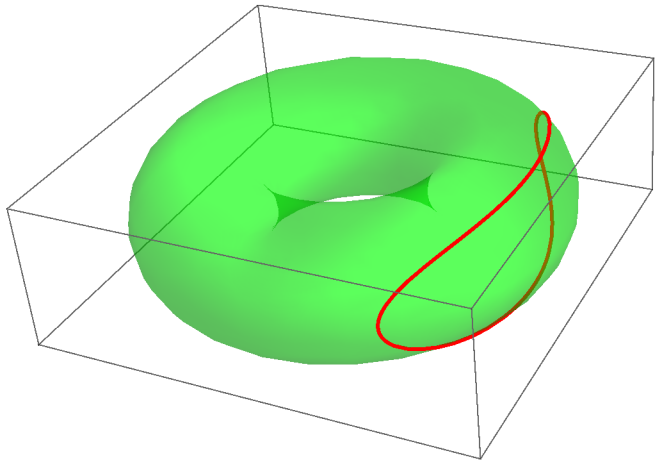
Out[185]=



In[186]:=
```
tr = ParametricPlot3D[torus[1, 2, 1][u, v], {u, 0, 2 Pi},
    {v, 0, 2 Pi}, ColorFunction → ({Opacity[0.4], Green} &), Mesh → False];
```

In[188]:=
```
Show[tr, cv1, Axes → False]
```

Out[188]=



Let' s define the equation of our curve:

In[189]:=
```
curve[t_] = torus[][Cos[t], Sin[t]]
```

Out[189]=
```
{(2 + Cos[Cos[t]]) Cos[Sin[t]], (2 + Cos[Cos[t]]) Sin[Sin[t]], Sin[Cos[t]]}
```

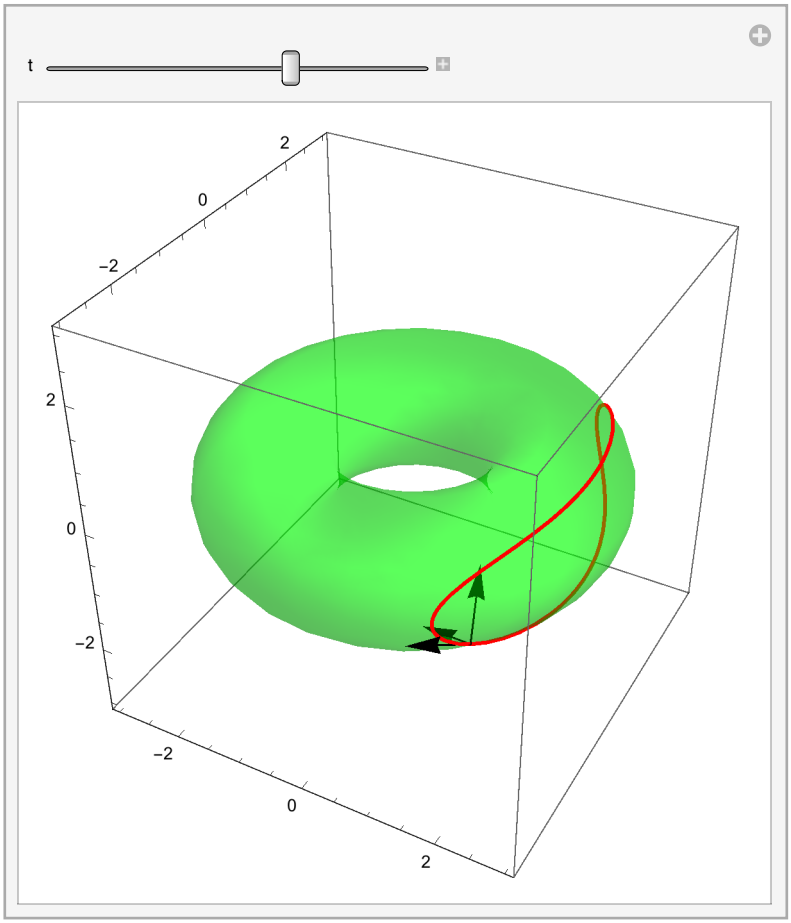
We can consider the curve as an ordinary curve in space and use the built-in function FrenetSerretSystem function:

In[190]:=
```
vecs[t_] = Last[FrenetSerretSystem[curve[t], t]];
```

Now we can see the Frenet-Serret frame moving on the curve on the torus:

In[191]:=
```
Manipulate[
 Show[tr, cv1, Graphics3D[Map[Arrow[{curve[t], curve[t] + #}] &, vecs[t]]],
  PlotRange → {{-3, 3}, {-3, 3}, {-3, 3}}], {t, 0, 2 Pi}, SaveDefinitions → True]
```

Out[191]=



Next we consider a Darboux frame. We must first compute the unit normal field to the torus.

In[192]:=
```
nr[u_, v_] = Simplify[Normalize[
    Apply[Cross, Transpose[D[torus[][u, v], {{u, v}}]]]], Element[{u, v}, Reals]];
```

A Darboux frame on a curve is given by three vectors: *t*,n×t,n

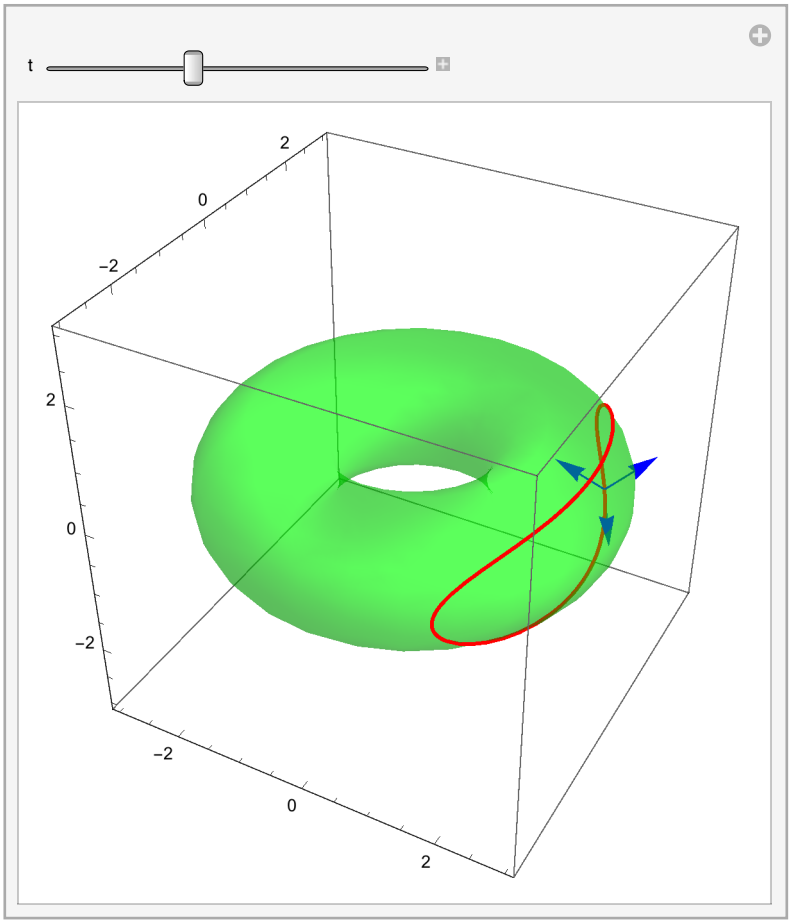
where t is a unit tangent vector to the curve, and *n* the unit normal to the surface.

In[193]:=
```
tg[t_] = Simplify[Normalize[D[curve[t], t]], Element[t, Reals]];
```

In[194]:=
```
vects1[t_] := {tg[t], nr[Cos[t], Sin[t]], Cross[nr[Cos[t], Sin[t]], tg[t]]}
```

In[196]:=
```
Manipulate[Show[tr, cv1,
  Graphics3D[{Blue, Map[Arrow[{curve[t], curve[t] + #}] &, vects1[t]]}],
  PlotRange → {{-3, 3}, {-3, 3}, {-3, 3}}], {t, 0, 2 Pi}, SaveDefinitions → True]
```

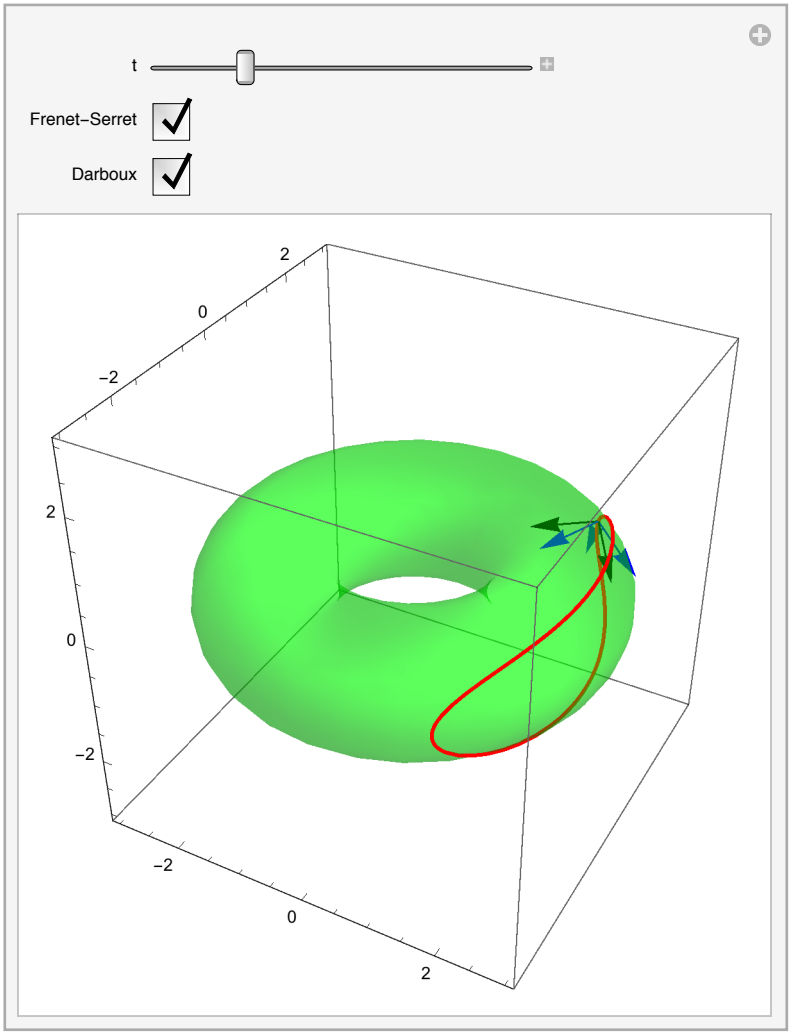
Out[196]=



We can put both frames together using the same Manipulate:

In[198]:=
```
Manipulate[Show[tr, cv1,
  Graphics3D[{If[fs, Map[Arrow[{curve[t], curve[t] + #}] &, vecs[t]], {}],
    If[dar, {Blue, Map[Arrow[{curve[t], curve[t] + #}] &, vects1[t]]}, {}]}],
  PlotRange → {{-3, 3}, {-3, 3}, {-3, 3}}], {t, 0, 2 Pi},
 {{fs, True, "Frenet-Serret"}, {True, False}},
 {{dar, True, "Darboux"}, {True, False}}]
```

Out[198]=



Geodesics

In this secotion we will use code based on Gray's book, notebook 17 and 18. I have modified it to make it more modern. Below Gray's code is given in "code cell" and the number 1 has been added to the name. If you want to use Gray's code, you have to convert it's cell to Input Form or copy the code into an input cell.

Here is Gray's code giving Christoffel symbols.

In[114]:=
```
christoffel1[x_][u_, v_] := Module[{A},
      si = Simplify[#] /. {U -> u, V -> v} &;
      xu = D[x[U, V], U];
      xv = D[x[U, V], V];
      ee = xu.xu;
      ff =  xu.xv;
      gg = xv.xv;
      A[1, 1, 1] =  D[ee, V] ff - 2 D[ff, U] ff + D[ee, U] gg;
      A[2, 2, 2] =  D[gg, V] ee - 2 D[ff, V] ff + D[gg, U] ff;
      A[2, 1, 1] = -D[ee, V] ee + 2 D[ff, U] ee - D[ee, U] ff;
      A[1, 2, 2] = -D[gg, V] ff + 2 D[ff, V] gg - D[gg, U] gg;
      A[1, 1, 2] =  D[ee, V] gg - D[gg, U] ff;
      A[2, 1, 2] = -D[ee, V] ff + D[gg, U] ee;
      A[i_, 2, 1] = A[i, 1, 2];
      G[i_, j_, k_] := A[i, j, k]/(ee gg - ff^2)/2 // si;
      Flatten[Table[G[i, j, k], {i, 2}, {j, 2}, {k, j, 2}]]
   ];
```

Here is our new code giving Chistoffel's symbols.

In[199]:=
```
christoffel[x_][u_, v_] := Module[{A = Array[0 &, {2, 2, 2}], ee, ff, gg, U, V},
ee = D[x[U, V], U].D[x[U, V], U];
ff =  D[x[U, V], U].D[x[U, V], V];
gg = D[x[U, V], V].D[x[U, V], V];
A[[1, 1, 1]] =  D[ee, V] ff - 2 D[ff, U] ff + D[ee, U] gg;
A[[2, 2, 2]] =  D[gg, V] ee - 2 D[ff, V] ff + D[gg, U] ff;
A[[2, 1, 1]] = -D[ee, V] ee + 2 D[ff, U] ee - D[ee, U] ff;
A[[1, 2, 2]] = -D[gg, V] ff + 2 D[ff, V] gg - D[gg, U] gg;
A[[1, 1, 2]] =  D[ee, V] gg - D[gg, U] ff;
A[[2, 1, 2]] = -D[ee, V] ff + D[gg, U] ee;
Do[A[[i, 2, 1]] = A[[i, 1, 2]], {i, 1, 2}];
(A/(ee gg - ff^2)/2) /. {U → u, V → v}]
```

To compute Chistoffel's symbols we use:

In[200]:=
```
G = Simplify[christoffel[torus[]][u, v]]
```

Out[200]=
$$\left\{\{\{0, 0\}, \{0, (2 + \text{Cos}[u])\ \text{Sin}[u]\}\}, \left\{\left\{0, -\frac{\text{Sin}[u]}{2 + \text{Cos}[u]}\right\}, \left\{-\frac{\text{Sin}[u]}{2 + \text{Cos}[u]}, 0\right\}\right\}\right\}$$

Individual symbols can then be given by

In[202]:=
```
G[[2, 2, 1]]
```

Out[202]=
$$-\frac{\text{Sin}[u]}{2 + \text{Cos}[u]}$$

Next, we define the unit mormal vector function.

In[203]:=
```
unitnormal[x_][u_, v_] :=
 Simplify[Normalize[Cross[Derivative[1, 0][x][u, v], Derivative[0, 1][x][u, v]]],
  Element[_, Reals]]
```

In[204]:=
```
unitnormal[torus[]][u, v]
```

Out[204]=
$$\{-\text{Cos}[u]\ \text{Cos}[v], -\text{Cos}[u]\ \text{Sin}[v], -\text{Sin}[u]\}$$

It gives the same answer as nr defined above:

In[205]:=
```
nr[u, v]
```

Out[205]=
```
{-Cos[u] Cos[v], -Cos[u] Sin[v], -Sin[u]}
```

Here is Gray's definition of the normal curvature function

In[57]:=
```
normalcurv1[x_,α_][t_]:=
Module[{A,B,Nx,Q},
   Nx= unitnormal[x]@@α[tt];
   xt= x@@α[tt];
   A= D[xt, tt];
   B= D[xt, tt, tt];
   Q= B.Nx/A.A;
   Simplify[Q,Element[_,Reals]]/.tt->t
];
```

Here is our new definition. As you can see it is simpler.

In[206]:=
```
normalcurv[x_, {α_, β_}][t_] := With[{ xt = x[α[#], β[#]] &},
 xt''[t].unitnormal[x][α[t], β[t]] / xt'[t].xt'[t]]
```

In[207]:=
```
Simplify[normalcurv[torus[], {Cos, Sin}][t]]
```

Out[207]=
```
(6 - 2 Cos[2 t] + Cos[2 t - 2 Cos[t]] + 4 Cos[2 t - Cos[t]] +
    8 Cos[Cos[t]] + 2 Cos[2 Cos[t]] + Cos[2 (t + Cos[t])] + 4 Cos[2 t + Cos[t]]) /
 (22 + 14 Cos[2 t] + Cos[2 t - 2 Cos[t]] + 8 Cos[2 t - Cos[t]] + 16 Cos[Cos[t]] +
    2 Cos[2 Cos[t]] + Cos[2 (t + Cos[t])] + 8 Cos[2 t + Cos[t]])
```

Here is Gray's definition of geodesic curvature.

In[60]:=
```
geodesiccurv1[x_, α_][t_] :=
  Module[{A, B, Nx, xt, Q},
      Nx = unitnormal[x] @@ α[tt];
      xt = x @@ α[tt];
      A = D[xt, tt];
      B = D[xt, tt, tt];
      Q = Det[{A, B, Nx}]/(A.A)^(3/2);
      Simplify[Q, Element[_, Reals]] /. tt -> t
   ];
```

Here is our replacement . Here x is the function parametrising the surface and $\{\alpha, \beta\}$ the parameters of the plane curve that defines the curve in the plane (whose image under x is a curve on the torus).

In[208]:=
```
geodesiccurv[x_, {α_, β_}][t_] := With[{ xt = x[α[#], β[#]] &},
Det[{xt'[t], xt''[t], unitnormal[x][α[t], β[t]]}] / (xt'[t].xt'[t])^(3 / 2)]
```

In[210]:=
```
geodesiccurv[torus[], {Cos, Sin}][0.5]
```

Out[210]=
```
-0.0976597
```

In[213]:=
```
Simplify[geodesiccurv[torus[], {Cos, Sin}][t]]
```

Out[213]=
$$\left(64 + 32\,\text{Cos}[\text{Cos}[t]] + 3\,\text{Sin}[t - 3\,\text{Cos}[t]] + \text{Sin}[3\,t - 3\,\text{Cos}[t]] + 24\,\text{Sin}[t - 2\,\text{Cos}[t]] + 8\,\text{Sin}[3\,t - 2\,\text{Cos}[t]] + 59\,\text{Sin}[t - \text{Cos}[t]] + 9\,\text{Sin}[3\,t - \text{Cos}[t]] - 59\,\text{Sin}[t + \text{Cos}[t]] - \text{Sin}[3\,(t + \text{Cos}[t])] - 9\,\text{Sin}[3\,t + \text{Cos}[t]] - 24\,\text{Sin}[t + 2\,\text{Cos}[t]] - 8\,\text{Sin}[3\,t + 2\,\text{Cos}[t]] - 3\,\text{Sin}[t + 3\,\text{Cos}[t]]\right) \Big/ \left(\sqrt{2}\,(22 + 14\,\text{Cos}[2\,t] + \text{Cos}[2\,t - 2\,\text{Cos}[t]] + 8\,\text{Cos}[2\,t - \text{Cos}[t]] + 16\,\text{Cos}[\text{Cos}[t]] + 2\,\text{Cos}[2\,\text{Cos}[t]] + \text{Cos}[2\,(t + \text{Cos}[t])] + 8\,\text{Cos}[2\,t + \text{Cos}[t]])^{3/2}\right)$$

Finally, here is Gray's function that constructs several pairs of parametric functions that have to be subsituted into the equation of the torus to obtain several geodesics starting from a given point. Here *x*is the equation of the surface, $\{u_0, v_0\}$ are the coordinates of the starting point, *b* is the length of flow and *m* the number of geodesics.

```
solvegeo1[x_,{u0_,v0_},b_,m_]:=
Module[{su,e,so},
   christoffel[x][u,v];
   su= {u->u[s], v->v[s], p->u'[s], q->v'[s]};
   e[j_]:= e[j]= G[j,1,1]p^2 + 2G[j,1,2]p q + G[j,2,2]q^2 /.su;
   eqic:= {u''[s]+e[1]==0, v''[s]+e[2]==0,
      u[0]==u0, v[0]==v0, u'[0]==Cos[θ], v'[0]==Sin[θ]};
   so:= NDSolve[eqic, {u,v}, {s,0,b}];
   Flatten[Table[so, {θ, 2π/m, 2π, 2π/m}],1]
];
```

Here is our replacement code (with the same meaning of parameters)

In[214]:=
```
solvegeo[x_, {u0_, v0_}, b_, m_] :=
Module[{e, eqic, so, G = christoffel[x][u0, v0]},
e = G[[All, 1, 1]] u'[s]^2 +
      2 G[[All, 1, 2]] u'[s] v'[s] + G[[All, 2, 2]] v'[s]^2 ;
eqic = {u''[s] + e[[1]] == 0, v''[s] + e[[2]] == 0, u[0] == u0,
      v[0] == v0, u'[0] == Cos[θ], v'[0] == Sin[θ]};
 Flatten[Table[NDSolve[eqic, {u, v}, {s, 0, b}], {θ, 2 π / m, 2 π, 2 π / m}],
    1]];
```

Let' s try to find one geodesic, of duration 6, starting from the point with parameters $\left\{\frac{\pi}{3}, \frac{\pi}{6}\right\}$.

```
solvegeo[torus[], {Pi/3, Pi/6}, 6, 1]
```

Out[215]=
$$\{\{u \to \text{InterpolatingFunction}[\langle\langle\rangle\rangle, \text{Domain: } \{\{0., 6.\}\} \text{ Output: scalar}]\,,$$
$$v \to \text{InterpolatingFunction}[\langle\langle\rangle\rangle, \text{Domain: } \{\{0., 6.\}\} \text{ Output: scalar}]\}\}$$

We got a pair of lists, each containing an interploating function. To get an actual pair of interpolating fucntions we need to do this:

In[217]:=
```
f = First[{u, v} /. solvegeo[torus[], {Pi/3, Pi/6}, 7, 1]]
```

Out[217]=
```
{InterpolatingFunction[ Domain: {{0., 7.}}
                        Output: scalar ],
 InterpolatingFunction[ Domain: {{0., 7.}}
                        Output: scalar ]}
```

Such a pair can now be applied to values as follows:

In[218]:=
```
Through[f[0.5]]
```

Out[218]=
```
{1.5472, 0.523599}
```

In order to get a point on the torus we need to use

In[219]:=
```
Apply[torus[], Through[f[0.5]]]
```

Out[219]=
```
{1.75249, 1.0118, 0.999722}
```

The same code can be written more efficiently as

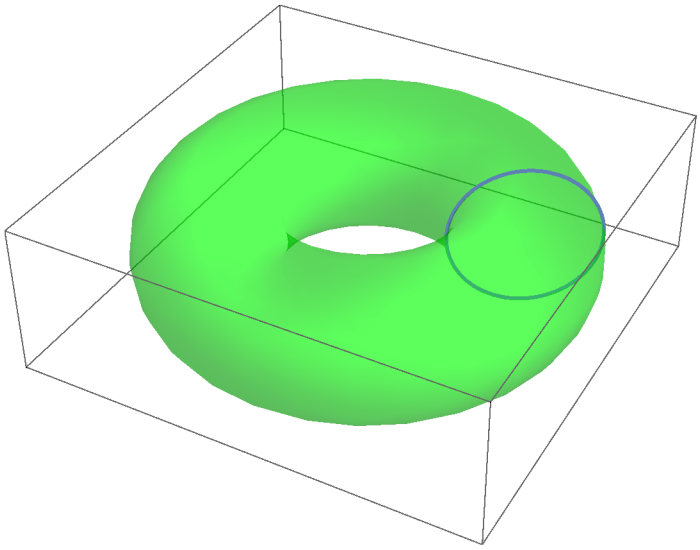
In[220]:=
```
torus[] @@ Through[f[0.5]]
```

Out[220]=
```
{1.75249, 1.0118, 0.999722}
```

We can now plot our geodesic on the torus:

In[222]:=
```
Show[tr, ParametricPlot3D[
  torus[] @@ Through[First[{u, v} /. solvegeo[torus[], {Pi/3, Pi/6}, 7, 1]][t]],
  {t, 0, 7}], Axes → False]
```

Out[222]=



We can obtain several geodesics at once. In this case we get 5:

In[223]:=
```
f = {u, v} /. solvegeo[torus[], {Pi/3, Pi/6}, 6, 5];
```

In[225]:=
```
Show[tr, ParametricPlot3D[
  torus[] @@@ Map[Through, Through[f[t]]], {t, 0, 7}], Axes → False]
```

Out[225]=