Plotting Surfaces

How to print special characters:

Pallettes → Special Characters → Letters → a (hover mouse over letter ℝ to find keyboard shortcut) Similarly in the α

→ Basic Math Assistant → Basic Commands → d∫Σ-> ∂ (hover over ∂ to find keyboard shortcut)

ℝ is [Esc]dsR[Esc]

ϕ is [Esc]f[Esc]

π is [Esc]p[Esc]

∂ is [Esc]pd[Esc]

ϕ^{T} is [Esc]f[Esc] [Esc]tr[Esc]   (transpose of matrix ϕ)

V^a is V [Ctrl]- a [Ctrl][Space]

Fraction: [Ctrl]/ , for example, for $\frac{x}{y}$ type x[Ctrl]/y

Torus T^2

Embedding Equations: x1 → x, x2 → y, x3 → z

The functions below give the parametric parametric representation:

In[ ]:=
```
x1[u_, v_] := R1 Cos[u] + R2 Cos[u] Cos[v];
x2[u_, v_] := R1 Sin[u] + R2 Sin[u] Cos[v];
x3[u_, v_] :=             R2       Sin[v];
```

In[ ]:=
```
{x1[u, v], x2[u, v], x3[u, v]}
```

Out[ ]=
```
{R1 Cos[u] + R2 Cos[u] Cos[v], R1 Sin[u] + R2 Cos[v] Sin[u], R2 Sin[v]}
```

Substitute anything for (u,v)

In[ ]:=
```
{x1[θ, ϕ], x2[θ, ϕ], x3[θ, ϕ]}
```

Out[ ]=
```
{R1 Cos[θ] + R2 Cos[θ] Cos[ϕ], R1 Sin[θ] + R2 Cos[ϕ] Sin[θ], R2 Sin[ϕ]}
```

See the result in a more familiar form:

In[ ]:=
```
{x1[θ, ϕ], x2[θ, ϕ], x3[θ, ϕ]} // TraditionalForm
```

Out[ ]//TraditionalForm=

$$\{\text{R1}\cos(\theta) + \text{R2}\cos(\theta)\cos(\phi),\ \text{R1}\sin(\theta) + \text{R2}\sin(\theta)\cos(\phi),\ \text{R2}\sin(\phi)\}$$

Towards obtaining numerical values: We pass numbers through the arguments of the functions x1,x2,x3.

Notice that R1 and R2 are still undefined symbols.

In[ ]:=
```
{x1[0.3, 0.1], x2[0.3, 0.1], x3[0.3, 0.1]}
```
Out[ ]=
```
{0.955336 R1 + 0.950564 R2, 0.29552 R1 + 0.294044 R2, 0.0998334 R2}
```

We obtain the same result by first evaluating the functions for (u,v), then substitute numerical values to the resulting expression.

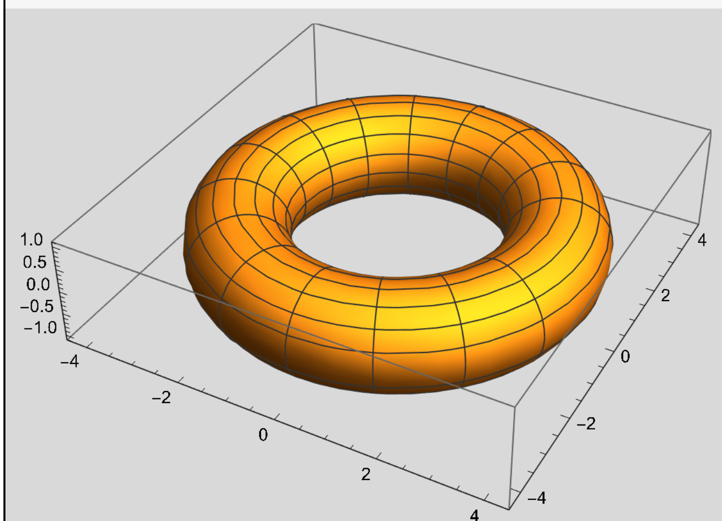
In[ ]:=
```
{x1[u, v], x2[u, v], x3[u, v]} /. {u → 0.3, v → 0.1}
```
Out[ ]=
```
{0.955336 R1 + 0.950564 R2, 0.29552 R1 + 0.294044 R2, 0.0998334 R2}
```

We obtain numerical results if we further substitute numerical values for R1, R2:

In[ ]:=
```
{x1[u, v], x2[u, v], x3[u, v]} /. {u → 0.3, v → 0.1, R1 → 3, R2 → 1}
```
Out[ ]=
```
{3.81657, 1.1806, 0.0998334}
```

This is the plot for the torus for R1=3 and R2=1. Notice that the plotting function needs to receive numerical results for the points, therefore we have to substitute the numerical values for R1 and R2. **ParametricPlot3D** plots a surface if it is a two parameter plot.

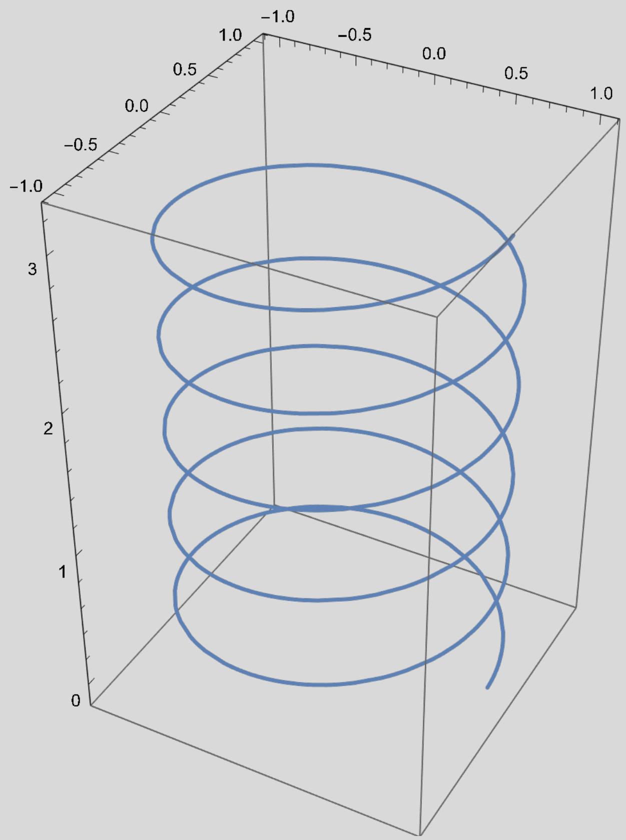
In[ ]:=
```
ParametricPlot3D[{x1[u, v], x2[u, v], x3[u, v]} /. {R1 → 3, R2 → 1}, {u, 0, 2 π}, {v, 0, 2 π}]
```
Out[ ]=



ParametricPlot3D plots a curve, when it is a one parameter plot:

In[∘]:=
```
ParametricPlot3D[{Cos[t], Sin[t], 0.1 t}, {t, 0, 10 π}]
```

Out[∘]=

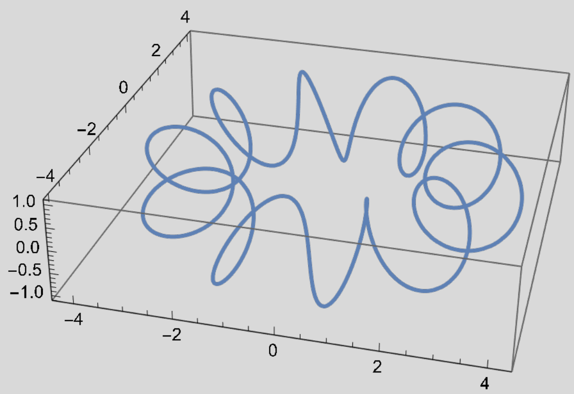


So, let’s plot a curve on the torus. A curve on the torus is a function $\gamma[t] = \{u[t], v[t]\}$, then its embedding in $\mathbb{R}^3$ is given by $\{x1[u[t], v[t]], x2[u[t], v[t]], x1[u[t], v[t]]\}$

In Mathematica, we obtain $u[t]$, $v[t]$ by taking the 1st and 2nd elements of the list {u[t],v[t]}, i.e. $u[t] = \gamma[t][[1]]$, $v[t] = \gamma[t][[2]]$

In[∘]:=
```
γ[t_] := {t, 10 t};
ParametricPlot3D[
 {x1[γ[t][[1]], γ[t][[2]]], x2[γ[t][[1]], γ[t][[2]]], x3[γ[t][[1]], γ[t][[2]]]} /. {R1 → 3, R2 → 1},
 {t, 0, 2 π}]
```

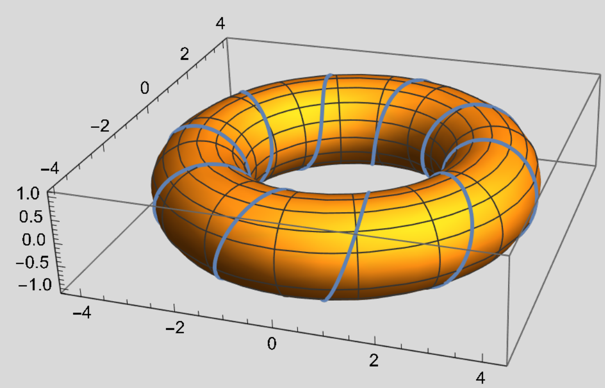
Out[∘]=



Let’s see the two plots together. We give names g1, g2 to the plots of the curve and the Torus respectively:

```
γ[t_] := {t, 10 t}; (*The curve*)
g1 = ParametricPlot3D[
    {x1[γ[t][[1]], γ[t][[2]]], x2[γ[t][[1]], γ[t][[2]]], x3[γ[t][[1]], γ[t][[2]]]} /. {R1 → 3, R2 → 1},
    {t, 0, 2 π}          ]; (* The plot of the curve *)
g2 = ParametricPlot3D[
    {x1[  u,       v    ], x2[  u,       v]    , x3[  u,       v    ]} /.
     {R1 → 3, R2 → 1}, {u, 0, 2 π}, {v, 0, 2 π}];(* The torus *)

(*Then we see the plots using the Show function,
which shows graphics together. *)
Show[g1, g2]
```

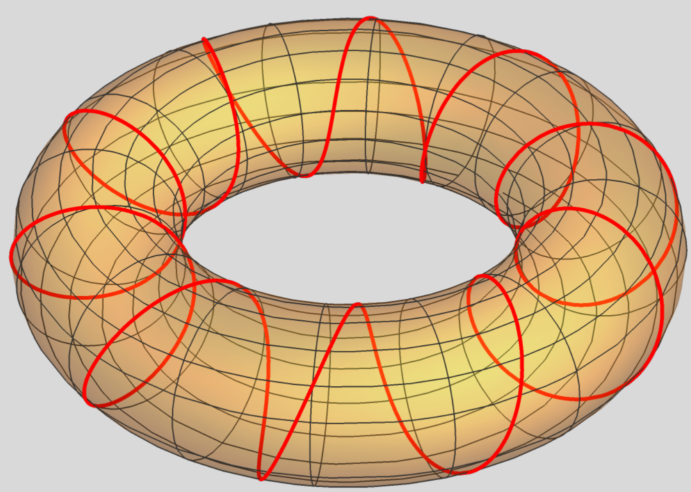
Out[∘]=



Add some style to the plots: Make the torus transparent and the curve Red+Thick. Add options in the Show function to show only the surface without the axes.

In[ ]:=
```
γ[t_] := {t, 10 t} ; (*The curve*)
g1 = ParametricPlot3D[
   {x1[ γ[t][[1]] , γ[t][[2]] ], x2[ γ[t][[1]] , γ[t][[2]] ], x3[ γ[t][[1]] , γ[t][[2]] ]} /. {R1 → 3, R2 → 1},
   {t, 0, 2 π}            , PlotStyle → {Red, Thick}]; (* The plot of the curve *)
g2 = ParametricPlot3D[{x1[   u     ,    v     ],
     x2[   u     ,    v     ], x3[   u     ,    v     ]} /. {R1 → 3, R2 → 1},
   {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];(* The torus *)

(*Then we see the plots using the Show function,
which shows graphics together. *)
Show[g1, g2, PlotRange → All, Axes → False, Boxed → False]
```

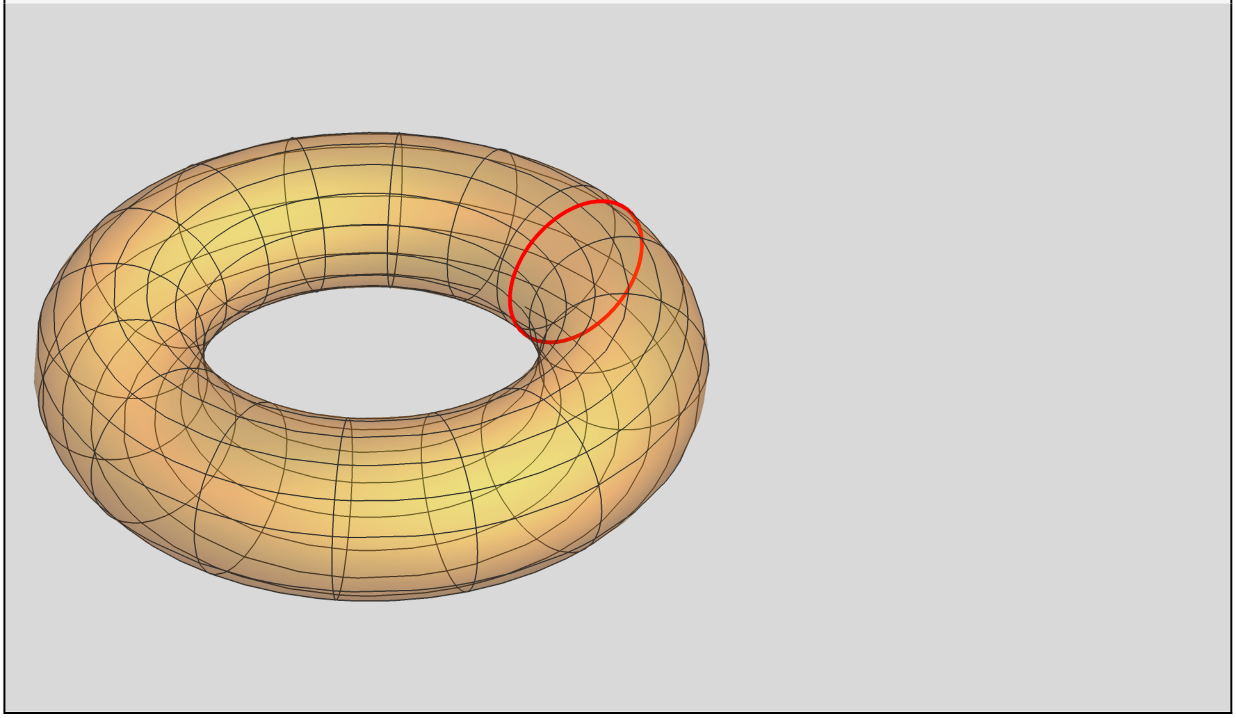
Out[ ]=



Follow coordinate curves on the torus: First a $u = \pi/3$ line

In[ ]:=
```
γ[t_] := {π / 3, t} ; (*The curve*)
g1 = ParametricPlot3D[
   {x1[ γ[t]〚1〛 , γ[t]〚2〛 ], x2[ γ[t]〚1〛 , γ[t]〚2〛 ], x3[ γ[t]〚1〛 , γ[t]〚2〛 ]} /. {R1 → 3, R2 → 1},
   {t, 0, 2 π}           , PlotStyle → {Red, Thick}]; (* The plot of the curve *)
g2 = ParametricPlot3D[{x1[  u    ,   v    ],
     x2[   u    ,   v    ], x3[   u    ,   v    ]} /. {R1 → 3, R2 → 1},
   {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];(* The torus *)

(*Then we see the plots using the Show function,
which shows graphics together. *)
Show[g1, g2, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=



See the u=const lines as u is varied. Use the manipulate function. It makes a series of plots, here depending on one parameter.

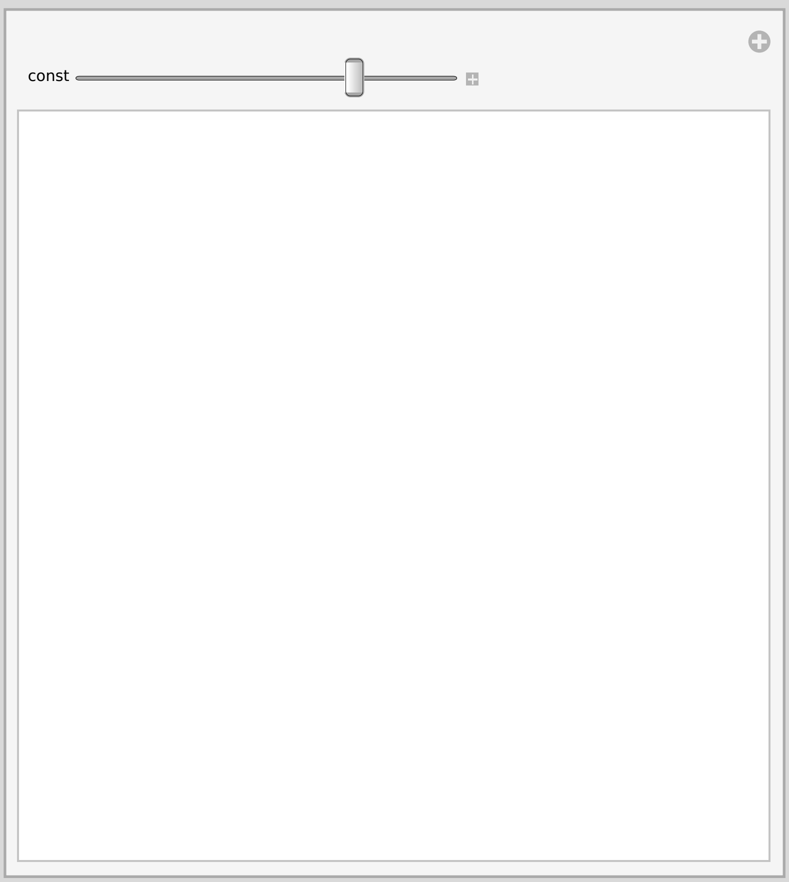
We simply wrap the contents of the previous cell around a

Manipulate[... ,{const,0,2π}]

function. Then the plots are shown by substitution of "const" by a value between the range {0,2π}

```
Manipulate[
 γ[t_] := {const, t} ; (*The curve*)
 g1 = ParametricPlot3D[
   {x1[ γ[t][[1]] , γ[t][[2]] ], x2[ γ[t][[1]] , γ[t][[2]] ], x3[ γ[t][[1]] , γ[t][[2]] ]} /. {R1 → 3, R2 → 1},
   {t, 0, 2 π}            , PlotStyle → {Red, Thick}]; (* The plot of the curve *)
 g2 = ParametricPlot3D[
   {x1[  u     ,   v    ], x2[   u     ,   v    ], x3[   u     ,   v    ]} /.
    {R1 → 3, R2 → 1}, {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];
 (* The torus *)

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[g1, g2, PlotRange → All, Axes → False, Boxed → False],
 {const, 0, 2 π}]
(*you may need to abort the evaluation for the Manipulate to stop
  eating your CPU cycles: Evaluation → Abort Evaluation – or [Alt]. *)
```

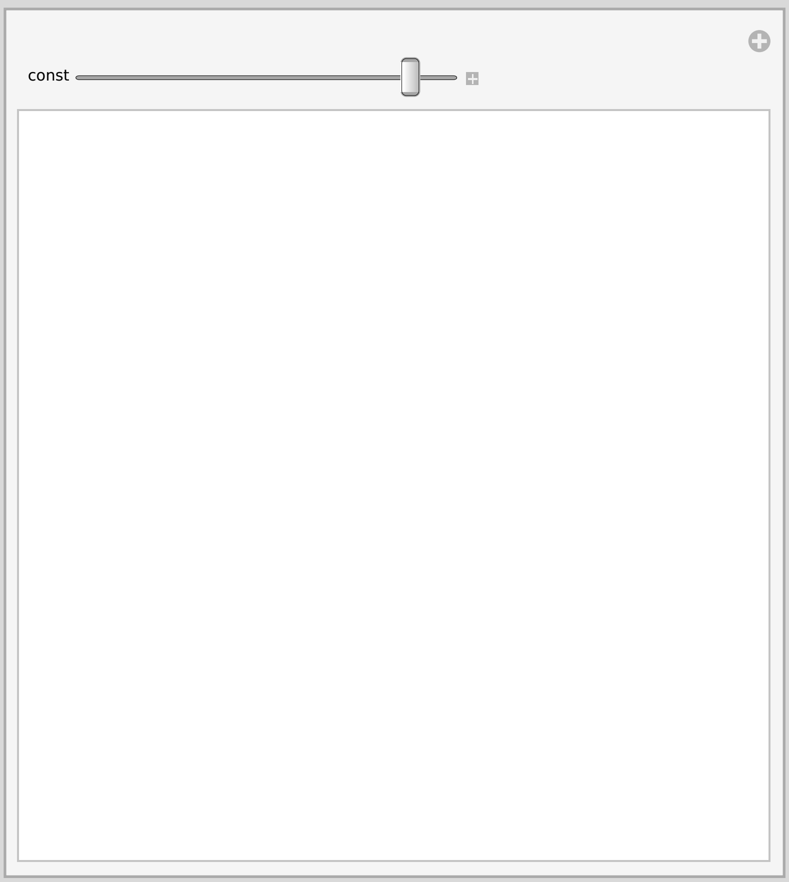
Out[ ]=



Let’s do the same for the v=const curves:

In[ ]:=
```
Manipulate[
 γ[t_] := {t, const} ; (*The curve*)
 g1 = ParametricPlot3D[
   {x1[ γ[t][[1]] , γ[t][[2]] ], x2[ γ[t][[1]] , γ[t][[2]] ], x3[ γ[t][[1]] , γ[t][[2]] ]} /. {R1 → 3, R2 → 1},
   {t, 0, 2 π}            , PlotStyle → {Red, Thick}]; (* The plot of the curve *)
 g2 = ParametricPlot3D[
   {x1[  u    ,   v    ], x2[  u    ,   v    ], x3[  u    ,   v    ]} /.
    {R1 → 3, R2 → 1}, {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];
 (* The torus *)

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[g1, g2, PlotRange → All, Axes → False, Boxed → False],
 {const, 0, 2 π}]
```

Out[ ]=

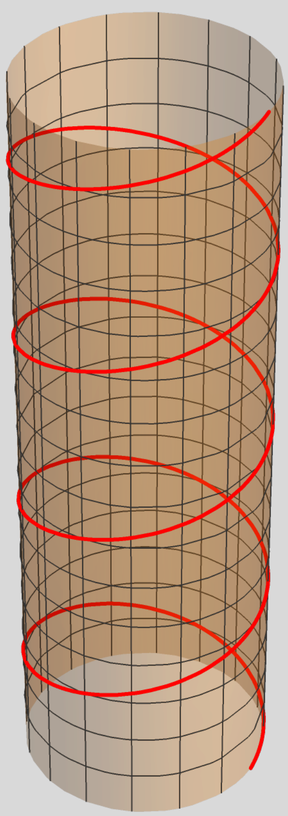


The Cylinder

In[ ]:=
```
x1[u_, v_] := Cos[u];
x2[u_, v_] := Sin[u];
x3[u_, v_] := v;
```

```
γ[t_] := {4 t, t} ; (*The curve*)
g1 = ParametricPlot3D[{x1[ γ[t][[1]] , γ[t][[2]] ], x2[ γ[t][[1]] , γ[t][[2]] ], x3[ γ[t][[1]] , γ[t][[2]] ]},
    {t, 0, 2 π}            , PlotStyle → {Red, Thick}];
g2 = ParametricPlot3D[{x1[   u     ,    v     ], x2[   u     ,    v     ],
     x3[   u     ,    v     ]}, {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

(*Then we see the plots using the Show function,
which shows graphics together. *)
Show[g1, g2, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=

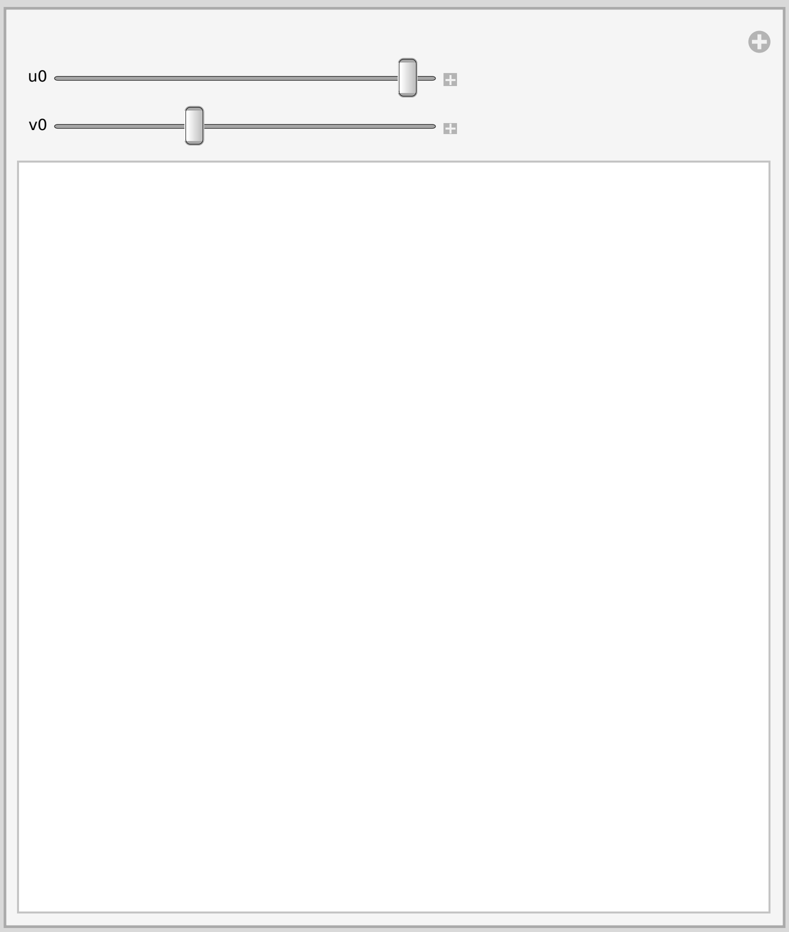


The u-v Coordinate curves. make a 2-parameter manipulate plot.

In[ ]:=

```
Manipulate[
 γu[t_] := {u0, t} ; (*The curves*)
 γv[t_] := {t, v0} ;
 gu = ParametricPlot3D[
   {x1[ γu[t][[1]] , γu[t][[2]] ], x2[ γu[t][[1]] , γu[t][[2]] ], x3[ γu[t][[1]] , γu[t][[2]] ]},
   {t, 0, 2 π}           , PlotStyle → {Red , Thick}]; (* The plot of the curve *)
 gv = ParametricPlot3D[
   {x1[ γv[t][[1]] , γv[t][[2]] ], x2[ γv[t][[1]] , γv[t][[2]] ], x3[ γv[t][[1]] , γv[t][[2]] ]},
   {t, 0, 2 π}           , PlotStyle → {Blue, Thick}]; (* The plot of the curve *)
 gs = ParametricPlot3D[{x1[   u    ,    v    ], x2[   u    ,    v    ],
    x3[   u    ,    v    ]}, {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];
 (* The torus *)

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[gu, gv, gs, PlotRange → All, Axes → False, Boxed → False],
 {u0, 0, 2 π}, {v0, 0, 2 π}]
```

Out[ ]=



The Klein Bottle

Immersion Equations: $x1 \to x,\ x2 \to y,\ x3 \to z$

We use the formulas in https://en.wikipedia.org/wiki/Klein_bottle

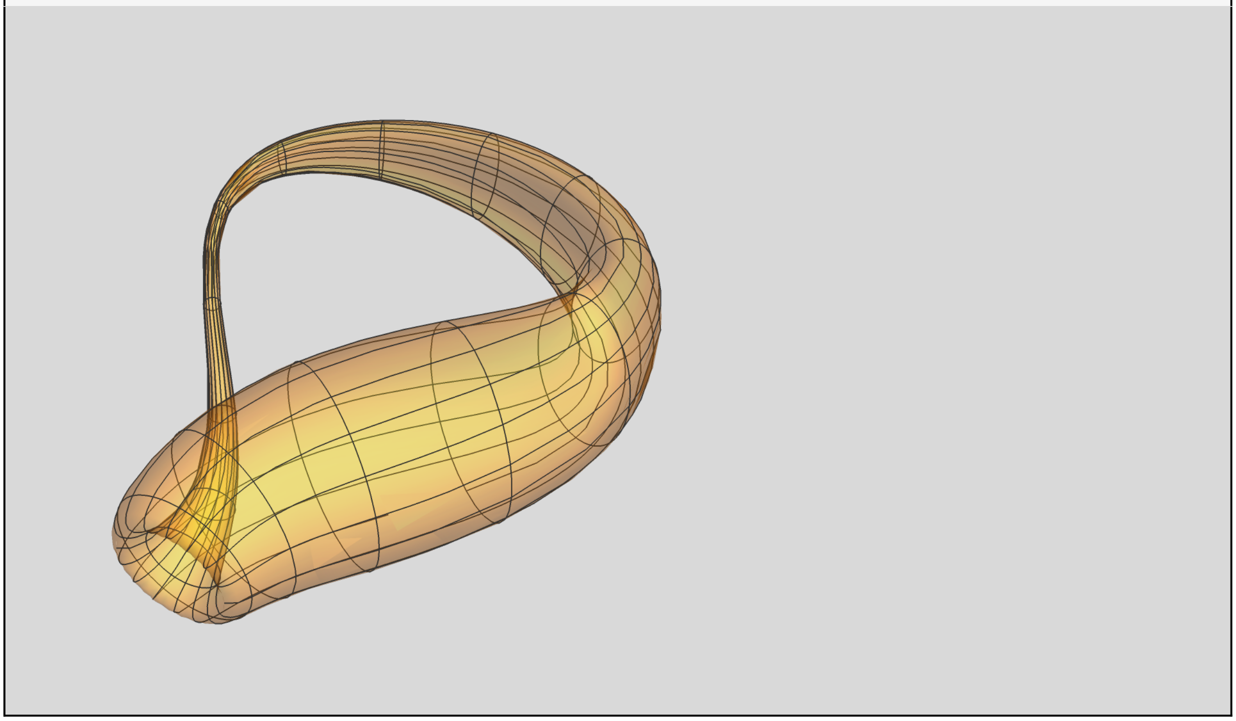
In[ ]:=
```
(* 0≤ u < π   0 ≤ v < 2 π *)
x1[u_, v_] := -2/15 Cos[u]
    (3 Cos[v] - 30 Sin[u] + 90 Cos[u]^4 Sin[u] - 60 Cos[u]^6 Sin[u] + 5 Cos[u] Cos[v] Sin[u]);
x2[u_, v_] := -1/15 Sin[u] (3 Cos[v] - 3 Cos[u]^2 Cos[v] -
      48 Cos[u]^4 Cos[v] + 48 Cos[u]^6 Cos[v] - 60 Sin[u] + 5 Cos[u] Cos[v] Sin[u] -
      5 Cos[u]^3 Cos[v] Sin[u] - 80 Cos[u]^5 Cos[v] Sin[u] + 80 Cos[u]^7 Cos[v] Sin[u]);
x3[u_, v_] := 2/15 Sin[v] (3 + 5 Cos[u] Sin[u]);
```

This is an immersion, not an embedding, which is not possible in $\mathbb{R}^3$.

It is possible to embed in $\mathbb{R}^4$, due to the Whitney's embedding theorem.

In[ ]:=
```
ParametricPlot3D[{x1[u, v], x2[u, v], x3[u, v]}, {u, 0, π}, {v, 0, 2 π},
 PlotStyle → Opacity[0.3], PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=

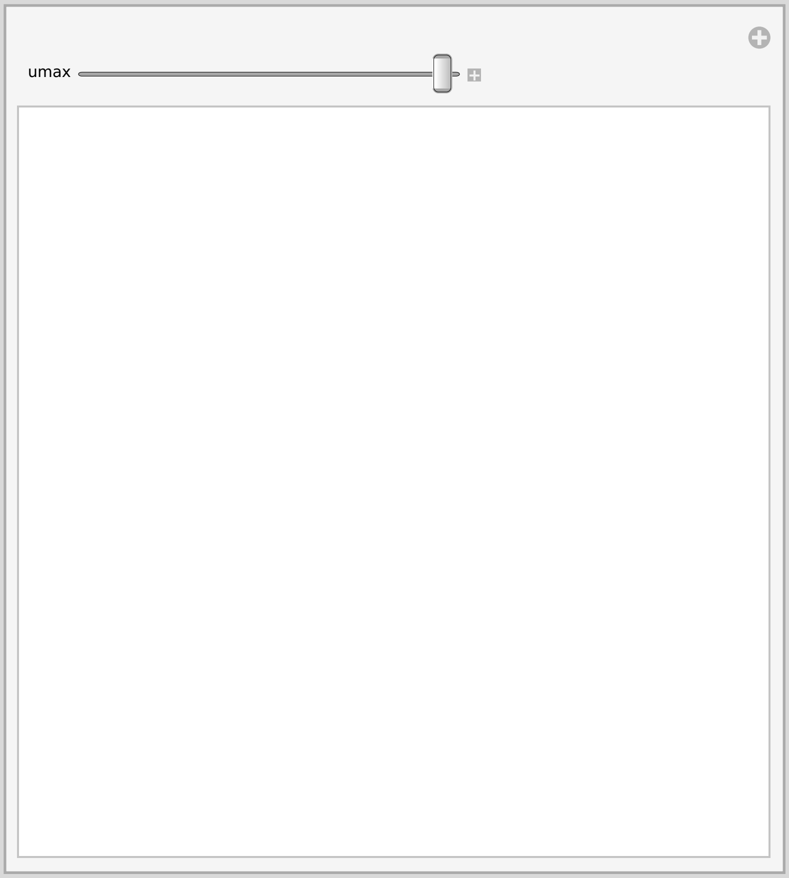


See how the surface is built as we increase u:

In[ ]:=

```
Manipulate[
 ParametricPlot3D[{x1[u, v], x2[u, v], x3[u, v]}, {u, 0, umax}, {v, 0, 2 π},
  PlotStyle → Opacity[0.3], PlotRange → All, Axes → False, Boxed → False],
 {umax, .4, π}
]
```

Out[ ]=



See how the surface is built as we increase v:

In[∘]:=

```
Manipulate[
 ParametricPlot3D[{x1[u, v], x2[u, v], x3[u, v]}, {u, 0, π}, {v, 0, vmax},
  PlotStyle → Opacity[0.3], PlotRange → All, Axes → False, Boxed → False],
 {vmax, π/3, π}
]
```

Out[∘]=

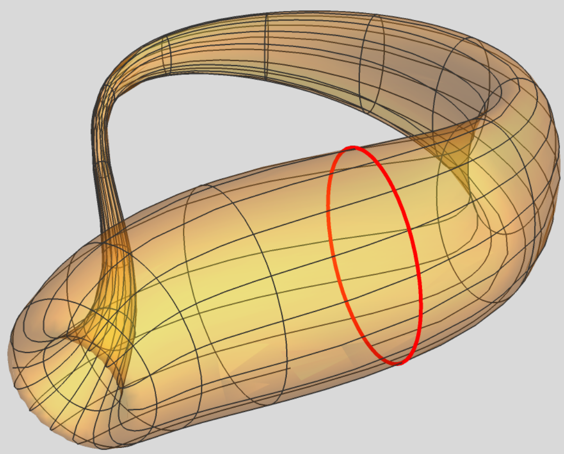
Now, also draw a curve on the bottle. Check the u and v-curves.

The u=u0 curve is a circle when $v \in (0, 2\pi)$.

Then vary $u0 \in (0, \pi)$ to visit all those circles.

In[ ]:=
```
u0 = π / 4; tmax = 2 π;
γu[t_] := {u0, t}; (*The curves*)
gu = ParametricPlot3D[{x1[γu[t][[1]], γu[t][[2]]], x2[γu[t][[1]], γu[t][[2]]],
    x3[γu[t][[1]], γu[t][[2]]]}, {t, 0, tmax}          , PlotStyle → {Red , Thick}];
gs = ParametricPlot3D[{x1[ u      ,   v    ], x2[   u    ,   v    ],
    x3[   u    ,   v    ]}, {u, 0,    π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

Show[gu, gs, PlotRange → All, Axes → False, Boxed → False]
```

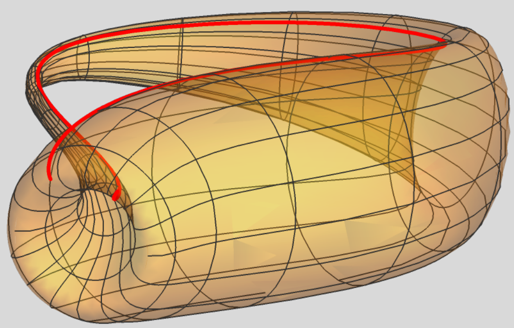
Out[ ]=



The v=v0 curve is trickier: as we move over the whole range $u \in (0, 2\pi)$, we end up at a different point!! The curve does not close, due to the opposite identification of the v=0 and v=2π lines

In[ ]:=
```
v0 = π / 3; tmax = π;
γv[t_] := {t, v0} ;
gv = ParametricPlot3D[{x1[ γv[t][[1]] , γv[t][[2]] ], x2[ γv[t][[1]] , γv[t][[2]] ],
     x3[ γv[t][[1]] , γv[t][[2]] ]}, {t, 0, tmax}           , PlotStyle → {Red , Thick}];
gs = ParametricPlot3D[{x1[  u       ,    v     ], x2[    u     ,    v     ],
     x3[    u     ,    v     ]}, {u, 0,    π }, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

Show[gv, gs, PlotRange → All, Axes → False, Boxed → False]
```

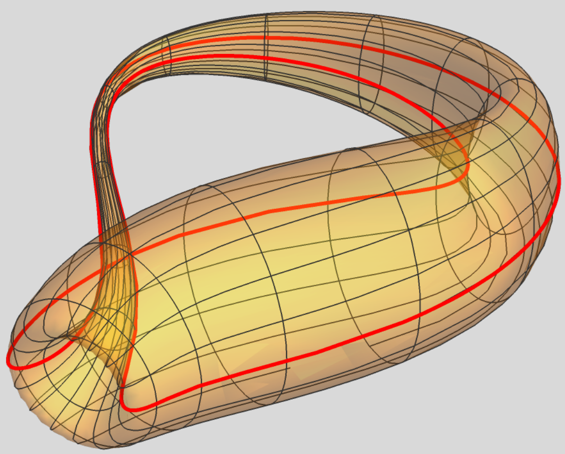
Out[ ]=



The v=v0 curve is trickier: as we move over the whole range $u \in (0, 2\pi)$, we end up at a different point!!
The curve does not close, due to the opposite identification of the v=0 and v=2π lines.
We have to go around one more time to meet the same point:

In[ ]:=
```
v0 = π; tmax = 2 π;
γv[t_] := {t, v0};
gv = ParametricPlot3D[{x1[γv[t][[1]], γv[t][[2]]], x2[γv[t][[1]], γv[t][[2]]],
     x3[γv[t][[1]], γv[t][[2]]]}, {t, 0, tmax}          , PlotStyle → {Red , Thick}];
gs = ParametricPlot3D[{x1[  u      ,    v     ], x2[    u     ,     v     ],
     x3[    u     ,    v     ]}, {u, 0,     π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

Show[gv, gs, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=



4d Embedding:
https://en.wikipedia.org/wiki/Klein_bottle

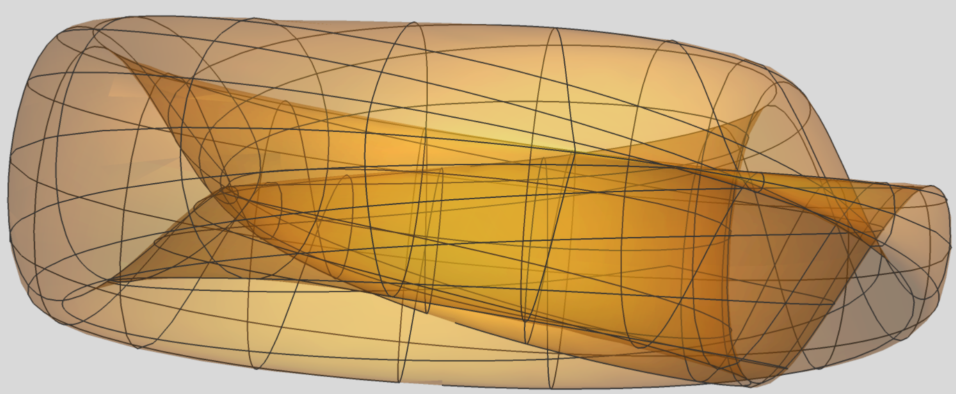
4-D non-intersecting [edit]

A non-intersecting 4-D parametrization can be modeled after that of the flat torus:

$$x = R\left(\cos\frac{\theta}{2}\cos v - \sin\frac{\theta}{2}\sin 2v\right)$$
$$y = R\left(\sin\frac{\theta}{2}\cos v + \cos\frac{\theta}{2}\sin 2v\right)$$
$$z = P\cos\theta\,(1 + \epsilon\sin v)$$
$$w = P\sin\theta\,(1 + \epsilon\sin v)$$

In[ ]:=
```
x1[u_, v_] := (Cos[u/2] Sin[v] - Sin[u/2] Sin[2 v]);
x2[u_, v_] := (Sin[u/2] Cos[v] + Cos[u/2] Sin[2 v]);
x3[u_, v_] := R Cos[u] (1 + ϵ Sin[v]);
x4[u_, v_] := R Sin[u] (1 + ϵ Sin[v]);
ρ[u_, v_] := Sqrt[x1[u, v]^2 + x2[u, v]^2]
r[u_, v_] := Sqrt[x1[u, v]^2 + x2[u, v]^2 + x3[u, v]^2]

rule = {R → 3, ϵ → .3};

g123 = ParametricPlot3D[
   {x1[ u , v ], x2[ u , v ], x3[ u , v ]} /. rule,
   {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];
g234 = ParametricPlot3D[
   {x2[ u , v ], x3[ u , v ], x4[ u , v ]} /. rule,
   {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];
gρ34 = ParametricPlot3D[
   {ρ [ u , v ], x3[ u , v ], x4[ u , v ]} /. rule,
   {u, 0, 2 π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

Show[g234, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=



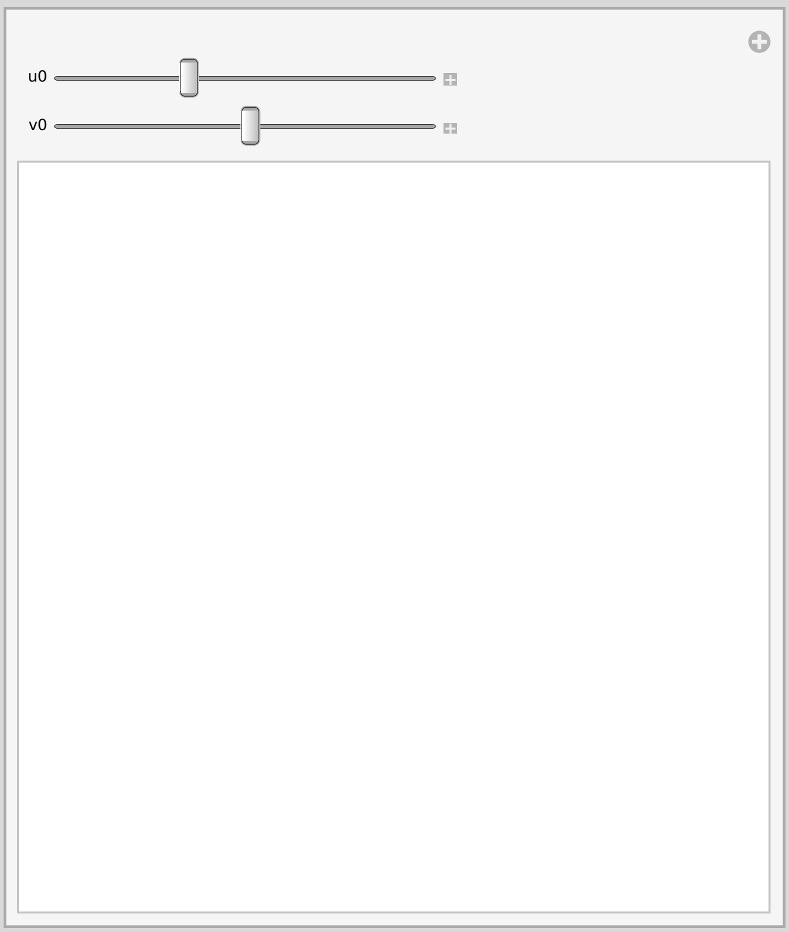
The Sphere S^2

$u \to \theta,\ v \to \phi$

In[ ]:=
```
x1[u_, v_] := Sin[u] Cos[v];
x2[u_, v_] := Sin[u] Sin[v];
x3[u_, v_] := Cos[u];
```

```
Manipulate[
 γu[t_] := {u0, t}; (*The curves*)
 γv[t_] := {t, v0};
 gu = ParametricPlot3D[{x1[γu[t][[1]], γu[t][[2]]], x2[γu[t][[1]], γu[t][[2]]],
    x3[γu[t][[1]], γu[t][[2]]]}, {t, 0, 2 π}, PlotStyle → {Red, Thick}];
 gv = ParametricPlot3D[{x1[γv[t][[1]], γv[t][[2]]], x2[γv[t][[1]], γv[t][[2]]],
    x3[γv[t][[1]], γv[t][[2]]]}, {t, 0, π}, PlotStyle → {Blue, Thick}];
 gs = ParametricPlot3D[{x1[u, v], x2[u, v],
    x3[u, v]}, {u, 0, π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]];

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[gu, gv, gs, PlotRange → All, Axes → False, Boxed → False],
 {u0, 0.1, π}, {v0, 0, 2 π}]
```

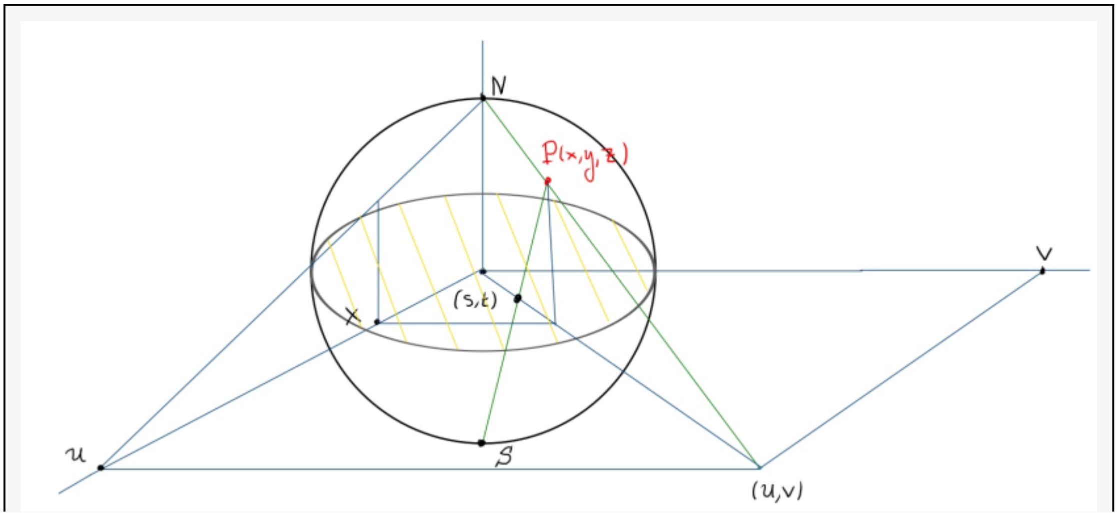
Out[ ]=



Now the stereographic projection w.r.t. to the North Pole:

$u = \frac{x}{1-z} = \frac{\sin(\theta)\cos(\phi)}{1-\cos(\theta)}$

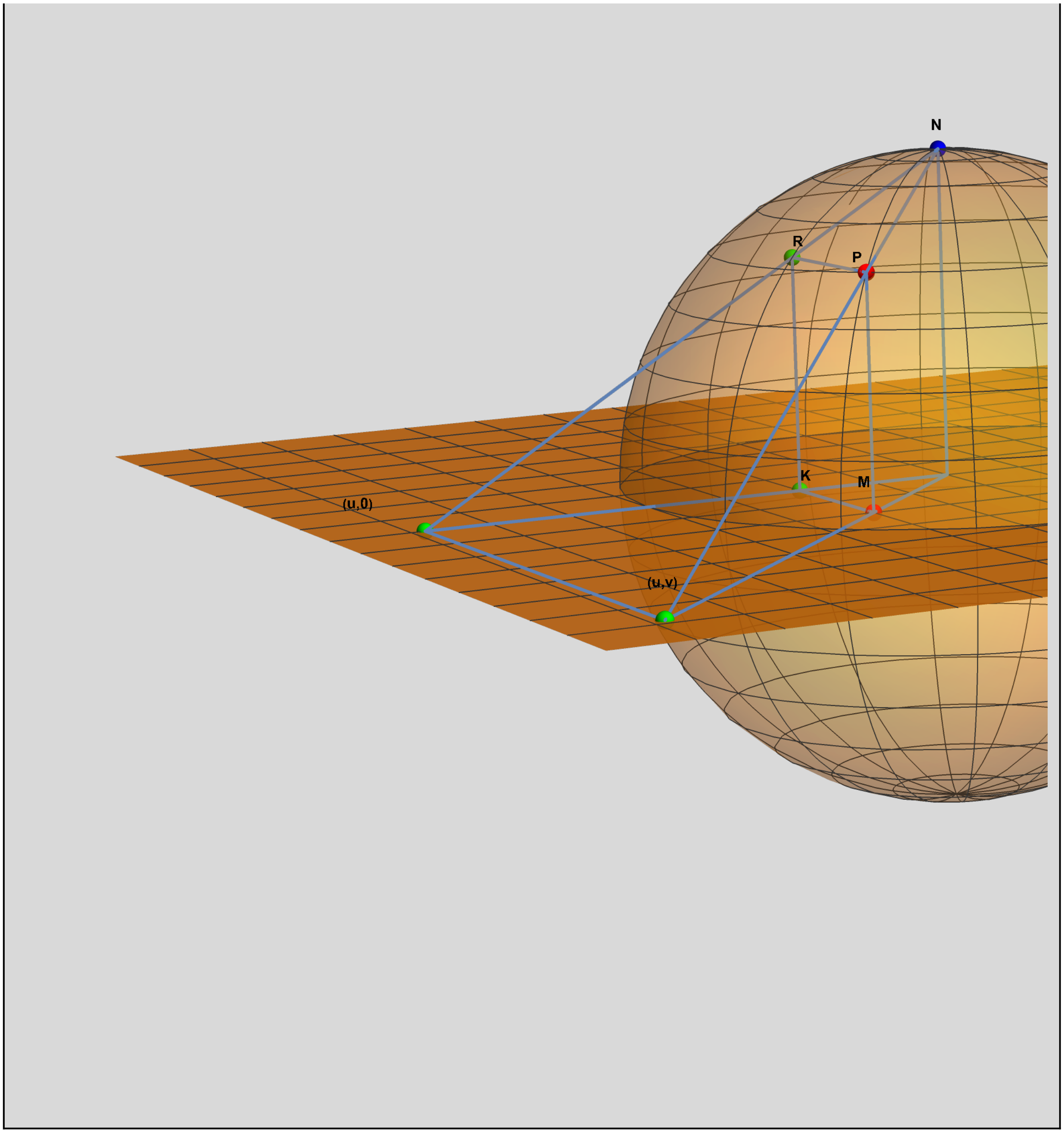
$v = \frac{y}{1-z} = \frac{\sin(\theta)\sin(\phi)}{1-\cos(\theta)}$




```
(*Set the (θ,ϕ) position of a point P: *)
θ  = π / 4          ; ϕ  = π / 4          ;
xP = Sin[θ] Cos[ϕ] ; yP = Sin[θ] Sin[ϕ]  ; zP = Cos[θ];
       xP
uP = ------          ;
     1 - zP
       yP
vP = ------          ; (* Projective coordinates *)
     1 - zP

gs = ParametricPlot3D[
   {x1[  u  ,  v  ], x2[  u  ,  v  ], x3[  u  ,  v  ]},
   {u, 0, π}, {v, 0, 2 π}, PlotStyle → Opacity[0.3]]; (* the sphere *)
gp = ParametricPlot3D[
   {x                , y                , 0                },
   {x, -2, 2}, {y, -2, 2}, PlotStyle → Opacity[0.9]]; (* the xy plane *)

(*The line passing through NP is γ⃗=t [(u,v,0)-(0,0,1)]+(0,0,1)= (t u,t v,1-t)*)
gNP = ParametricPlot3D[{t uP , t vP, 1 - t}, {t, 0, 1}];

(*Mark some points on the graph: *)
gP  = Graphics3D[{Red   , Sphere[{xP, yP, zP}, 0.025], Black,
     Text[Style["P"   , Bold], {xP  + 0.05, yP + 0.05, zP + 0.05}]}];
gN  = Graphics3D[{Blue   , Sphere[{0 , 0 , 1 }, 0.025],
     Black, Text[Style["N"   , Bold], {0      , 0      , 1  + 0.07}]}];
gUV = Graphics3D[{Green  , Sphere[{uP, vP, 0 }, 0.025],
     Black, Text[Style["(u,v)" , Bold], {uP      , vP      ,   0.10}]}];

gM  = Graphics3D[{Red   , Sphere[{xP, yP, 0 }, 0.025], Black,
     Text[Style["M"   , Bold], {xP  + 0.05, yP + 0.05,   0.10}]}];
gR  = Graphics3D[{Green  , Sphere[{xP, 0  , zP}, 0.025], Black,
```

```
    Text[Style["R"    , Bold], {xP       , 0.05    , zP + 0.05}]}];
gK  = Graphics3D[{Green  , Sphere[{xP, 0  , 0 }, 0.025], Black,
    Text[Style["K"    , Bold], {xP       , 0.05    ,    0.05}]}];
gV  = Graphics3D[{Green  , Sphere[{uP, 0 , 0 }, 0.025],
    Black, Text[Style["(u,0)", Bold], {uP + 0.2   , 0       ,    0.10}]}];

(*Draw some lines:*)
gON = ParametricPlot3D[{0   , 0   ,    t}, {t, 0, 1}];
gOP = ParametricPlot3D[{t xP , t yP , t zP}, {t, 0, 1}];
gOQ = ParametricPlot3D[{t uP , t vP ,    0}, {t, 0, 1}];
gPM = ParametricPlot3D[{  xP ,   yP , t zP}, {t, 0, 1}];
gRK = ParametricPlot3D[{  xP ,   0  , t zP}, {t, 0, 1}];
gPR  = ParametricPlot3D[{  xP , t yP ,   zP}, {t, 0, 1}];
gOV = ParametricPlot3D[{t uP ,   0 ,    0}, {t, 0, 1}];
gNV = ParametricPlot3D[{t uP ,   0 , 1 - t}, {t, 0, 1}];
gKM = ParametricPlot3D[{  xP, t yP ,    0}, {t, 0, 1}];
gUVV = ParametricPlot3D[{  uP, t vP ,    0}, {t, 0, 1}];

Show[gs, gp, gP, gM, gN, gR, gK, gUV, gV, gNP, gON, gOQ, gPM, gRK, gPR,
 gOV, gNV, gKM, gUVV, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=



Möbius strip

In[ ]:=

```
(*Position vector: *)
x[u_, v_] := {Cos[u] (1 + v/2 Cos[u/2]), Sin[u] (1 + v/2 Cos[u/2]), v/2 Sin[u/2]};

(*Tangent vectors: *)
dxu[u_, v_] = ∂_u x[u, v] // FullSimplify ;
dxv[u_, v_] = ∂_v x[u, v]  // FullSimplify ;
(*Normal vector: *)
xn [u_, v_] = Cross[dxu[u, v], dxv[u, v]] // FullSimplify;

Print["∂_ux= ", dxu[u, v] // TraditionalForm, "\n∂_vx= ",
 dxv[u, v] // TraditionalForm, "\nn= ", xn[u, v] // TraditionalForm]
```

$\partial_u x= \left\{-\frac{1}{4} \sin\left(\frac{u}{2}\right)\left(v\,(3\cos(u)+2)+8\cos\left(\frac{u}{2}\right)\right), \frac{1}{8}\,v\left(\cos\left(\frac{u}{2}\right)+3\cos\left(\frac{3\,u}{2}\right)\right)+\cos(u), \frac{1}{4}\,v\cos\left(\frac{u}{2}\right)\right\}$

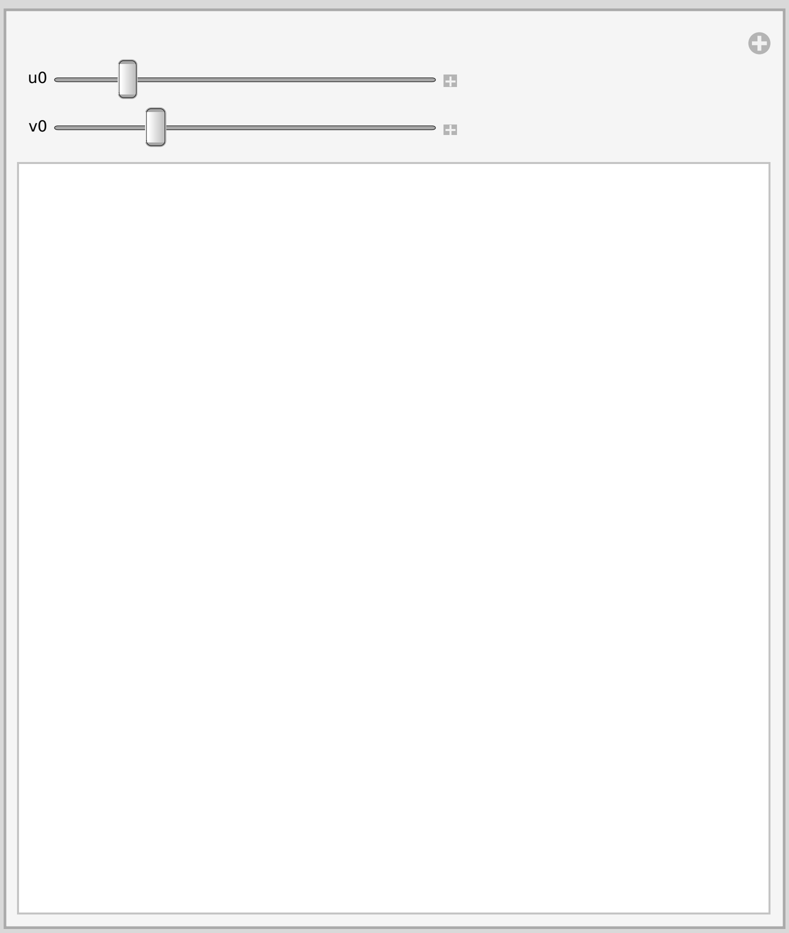
$\partial_v x= \left\{\frac{1}{2}\cos\left(\frac{u}{2}\right)\cos(u), \frac{1}{2}\sin(u)\cos\left(\frac{u}{2}\right), \frac{1}{2}\sin\left(\frac{u}{2}\right)\right\}$

$n= \left\{-\frac{1}{4}\sin\left(\frac{u}{2}\right)\left(v\sin\left(\frac{u}{2}\right)\sin(u)-2\cos(u)\right),\right.$
$\left.\frac{1}{8}\left(v\,(\sin^2(u)+\cos(u))+2\cos\left(\frac{u}{2}\right)-2\cos\left(\frac{3\,u}{2}\right)\right), -\frac{1}{4}\cos\left(\frac{u}{2}\right)\left(v\cos\left(\frac{u}{2}\right)+2\right)\right\}$

In[ ]:=

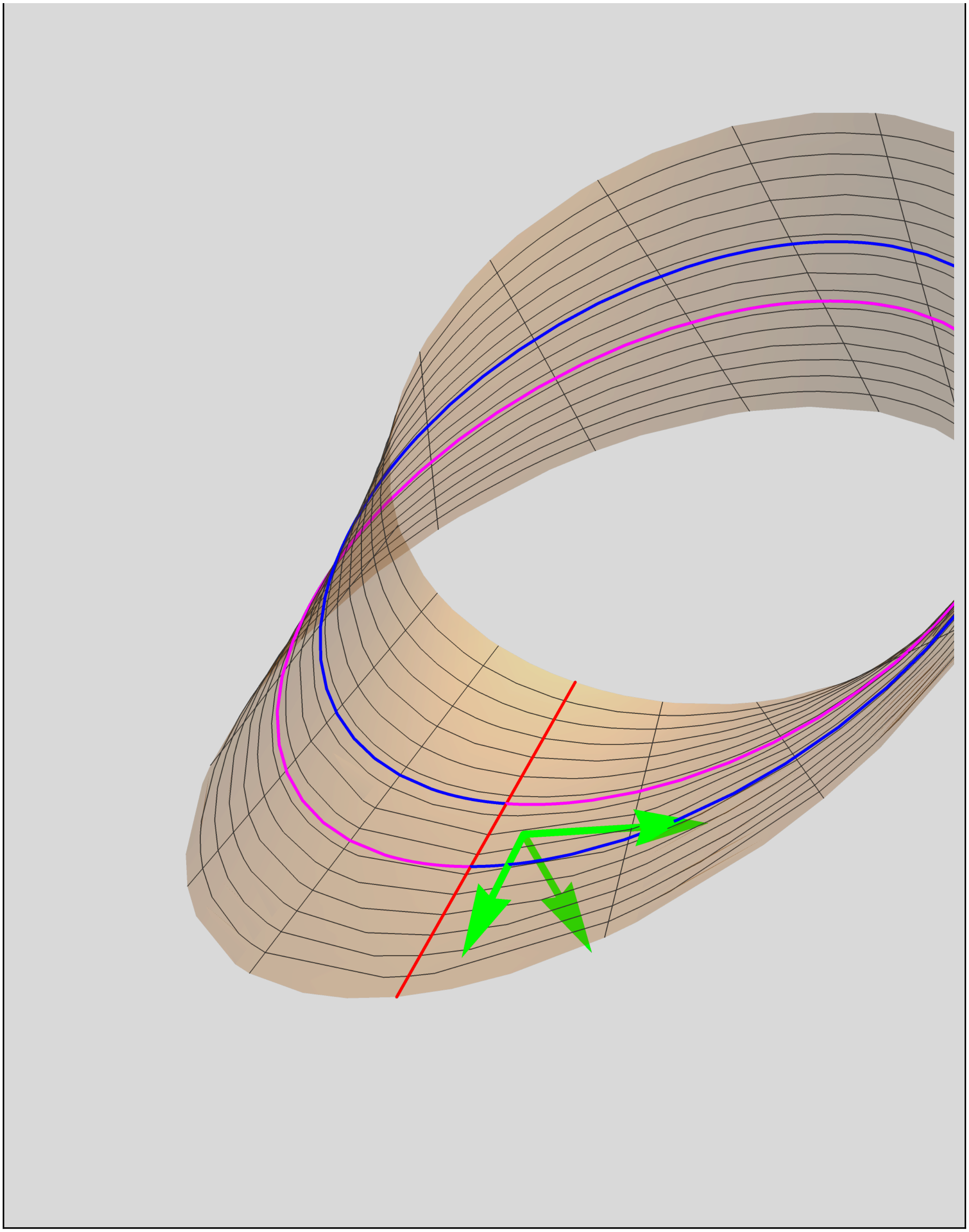
```
Manipulate[
 γu[t_] := {u0, t} ; (*The curves*)
 γv[t_] := {t, v0} ;
 gu  = ParametricPlot3D[x[ γu[t][[1]] , γu[t][[2]] ],
   {t, -1, 1  }            , PlotStyle → {Red     , Thick}];
 gv  = ParametricPlot3D[x[ γv[t][[1]] , γv[t][[2]] ],
   {t, 0  , 2 π}            , PlotStyle → {Blue    , Thick}];
 gs  = ParametricPlot3D[x[    u     ,    v     ],
   {u, 0  , 2 π}, {v, -1, 1}, PlotStyle → Opacity[0.3]    ];
 gv2 = ParametricPlot3D[x[ γv[t][[1]] , γv[t][[2]] ],
   {t, 2 π, 4 π}            , PlotStyle → {Magenta, Thick}];

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[gu, gv, gs, gv2, PlotRange → All, Axes → False, Boxed → False],
 {u0, 0, 2 π}, {v0, -1, 1}]
```

Out[ ]=



In[ ]:=
```
u0 = 0; v0 = 0.2;
γu[t_] := {u0, t}; (*The curves*)
γv[t_] := {t, v0};
gu  = ParametricPlot3D[x[γu[t][[1]], γu[t][[2]]],
   {t, -1, 1 }            , PlotStyle → {Red    , Thick}];
gv  = ParametricPlot3D[x[γv[t][[1]], γv[t][[2]]],
   {t, 0  , 2 π}            , PlotStyle → {Blue   , Thick}];
gs  = ParametricPlot3D[x[    u     ,    v    ],
   {u, 0  , 2 π}, {v, -1, 1}, PlotStyle → Opacity[0.3]   ];
gv2 = ParametricPlot3D[x[γv[t][[1]], γv[t][[2]] ],
   {t, 2 π, 4 π}            , PlotStyle → {Magenta, Thick}];

(* Plot tangent vectors at (u0,0) *)
gxu = Graphics3D[{Thickness[0.006], Green, Arrowheads[0.04],
     Arrow[{x[u0 + 0.1, 0], x[u0 + 0.1, 0] + 0.5 dxu[u0 + 0.1, 0]}]}];
gxv = Graphics3D[{Thickness[0.006], Green, Arrowheads[0.04],
     Arrow[{x[u0 + 0.1, 0], x[u0 + 0.1, 0] + 0.8 dxv[u0 + 0.1, 0]}]}];
gxn = Graphics3D[{Thickness[0.006], Green, Arrowheads[0.04],
     Arrow[{x[u0 + 0.1, 0], x[u0 + 0.1, 0] + 0.8 xn [u0 + 0.1, 0]}]}];

(*Then we see the plots using the Show function,
which shows graphics together. *)
Show[gu, gv, gs, gv2, gxu, gxv, gxn, PlotRange → All, Axes → False, Boxed → False]
```

Out[ ]=



In[ ]:=

```
u0 = 0;
gxu0 = Graphics3D[{Thickness[0.006], Red     ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.5 dxu[u0, 0]}]}];
gxv0 = Graphics3D[{Thickness[0.006], Red     ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.8 dxv[u0, 0]}]}];
gxn0 = Graphics3D[{Thickness[0.006], Magenta ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.8 xn [u0, 0]}]}];

Manipulate[

 gs  =
  ParametricPlot3D[x[u, v], {u, 0  , 2 π}, {v, -1, 1}, PlotStyle → Opacity[0.3]   ];

 (* Plot tangent vectors at (u0,0) *)
 gxu = Graphics3D[{Thickness[0.006], Green   ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.5 dxu[u0, 0]}]}];
 gxv = Graphics3D[{Thickness[0.006], Green   ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.8 dxv[u0, 0]}]}];
 gxn = Graphics3D[{Thickness[0.006], Blue    ,
     Arrowheads[0.04], Arrow[{x[u0, 0], x[u0, 0] + 0.8 xn [u0, 0]}]}];

 (*Then we see the plots using the Show function,
 which shows graphics together. *)
 Show[gs, gxu, gxv, gxn, gxu0, gxv0, gxn0,
  PlotRange → All, Axes → False, Boxed → False],

 {u0, 0, 4 π}
]
```

Out[ ]=