```
SetOptions[EvaluationNotebook[], CellContext → Notebook]
```

1. The tangent space and normal to a surface

Consider a trous given by parametric equations:

```
tor[θ_, ϕ_] := {(4 + Cos[θ]) Sin[ϕ], Sin[θ], (4 + Cos[θ]) Cos[ϕ]}
```

```
tor[{θ_, ϕ_}] := tor[θ, ϕ]
```

The Jacobi matrix gives two tangent vectors to the surface, which we can normalize to obtain two unit tangent vectors.

```
tg[f_][{x_, y_}] := Normalize /@ (D[f[u, v], {{u, v}}] /. {u → x, v → y} // Transpose)
```

```
tg[tor][{1, 2}] // N
```

```
{{-0.765147, 0.540302, 0.350175}, {-0.416147, 0., -0.909297}}
```

```
nr[f_][{x_, y_}] :=
 -Normalize[Cross @@ (Transpose[D[f[u, v], {{u, v}}]])] /. {u → x, v → y}
```

```
nr[tor][{Pi, -Pi}]
```

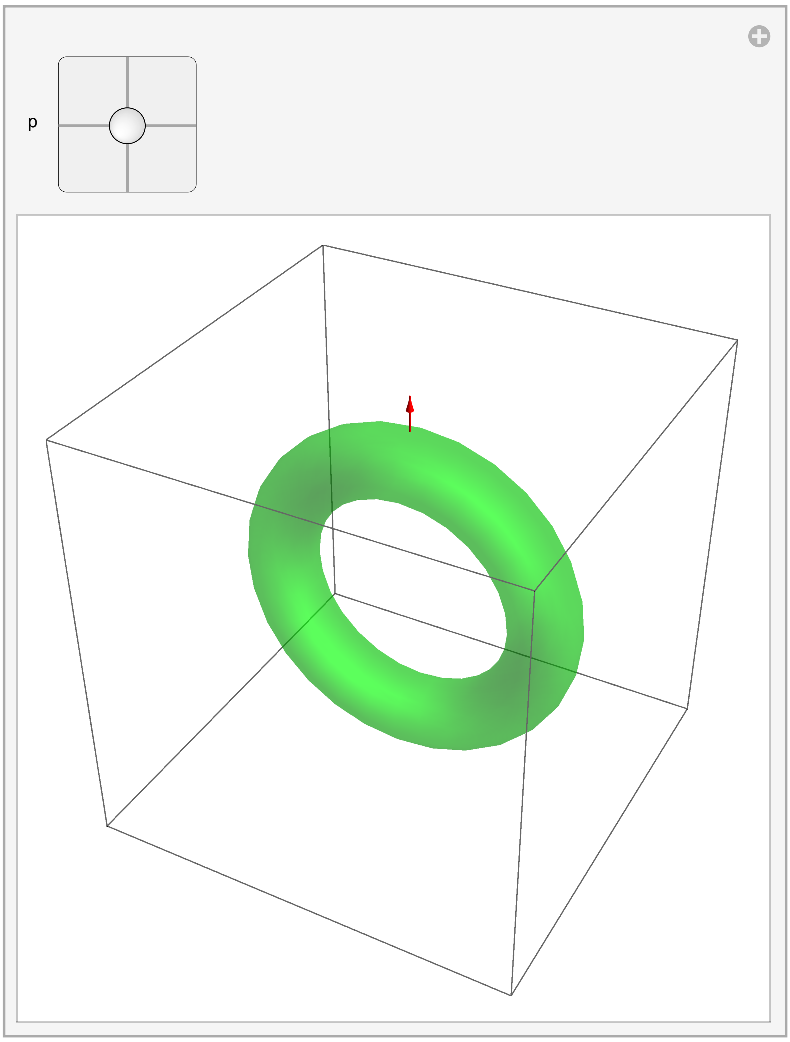
```
{0, 0, 1}
```

```
tgVects[f_][{u_, v_}] := Arrow[Tube[{f[u, v], f[u, v] + ##}, .02]] & /@ tg[f][{u, v}]
```

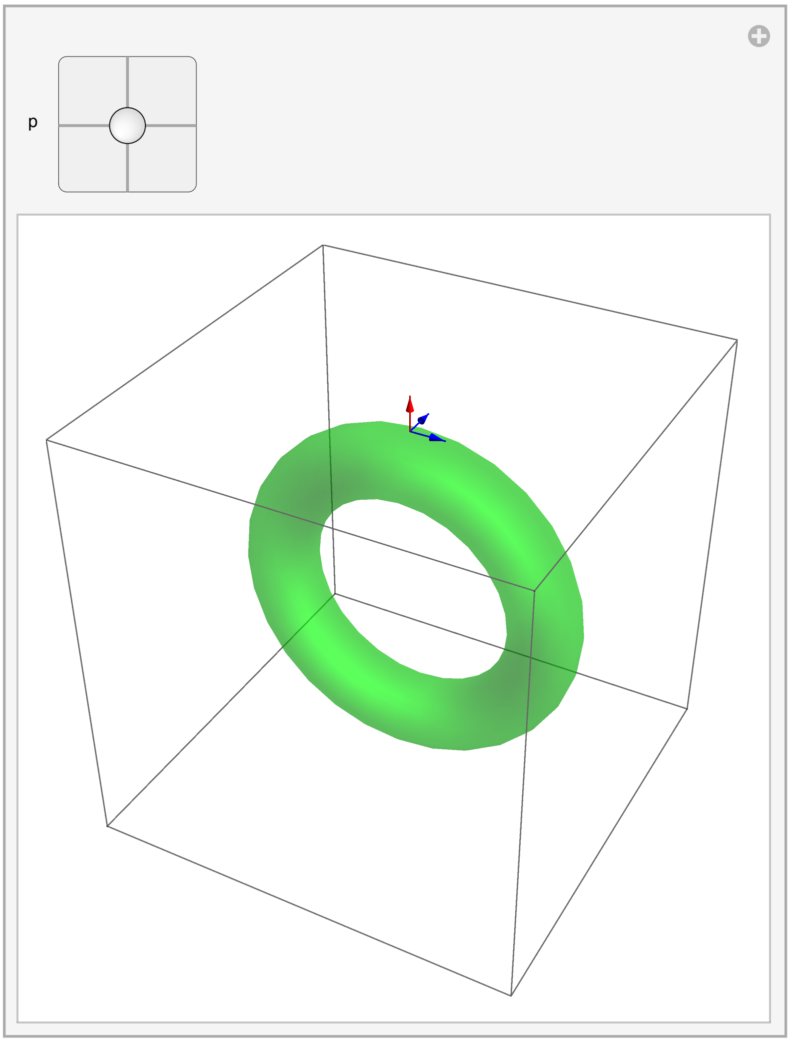
```
normVect[f_][{u_, v_}] := Arrow[Tube[{f[u, v], f[u, v] + nr[f][{u, v}]}, .02]]
```

```
ttr = ParametricPlot3D[tor[θ, ϕ], {θ, -Pi, Pi}, {ϕ, -Pi, Pi}, Axes → False,
   Mesh → None, ColorFunction → (Directive[Opacity[0.4], Green] &)];
```

```
Manipulate[
 Show[ttr, Graphics3D[{Thick, Red, Arrowheads[Small], normVect[tor][p]}],
  PlotRange → {{-6, 6}, {-6, 6}, {-6, 6}}],
 {{p, {0, 0}}, {-Pi, -Pi}, {Pi, Pi}, Slider2D}, SaveDefinitions → True]
```



```
Manipulate[Show[ttr, Graphics3D[
    {Thick, Red, Arrowheads[Small], normVect[tor][p], Blue, tgVects[tor][p]}],
   PlotRange → {{-6, 6}, {-6, 6}, {-6, 6}}],
  {{p, {0, 0}}, {-Pi, -Pi}, {Pi, Pi}}, SaveDefinitions → True]
```



2. Metrics on a surface

Let $f: D \to \mathbb{R}^3$ be a coordinate patch for a surface $S \subset \mathbb{R}^3$ and D is an open disk in $\mathbb{R}^2$. We define 4 functions on D.

```
ee[f_][u_, v_] := Simplify[f^(1,0)[u, v].f^(1,0)[u, v]]

ff[f_][u_, v_] := Simplify[f^(1,0)[u, v].f^(0,1)[u, v]]

gg[f_][u_, v_] := Simplify[f^(0,1)[u, v].f^(0,1)[u, v]]

hh[f_][u_, v_] := √(ee[f][u, v] gg[f][u, v] - ff[f][u, v]^2)
```

The Metric form

```
metricForm[f_][u_, v_] := ee[f][u, v] du^2 + 2 ff[f][u, v] du dv + gg[f][u, v] dv^2
```

```
metricForm[tor][u, v]
```

$du^2 + (4 + \text{Cos}[u])^2\, dv^2$

```
sph[r_][u_, v_] := r {Cos[u] Cos[v], Cos[v] Sin[u], Sin[v]}
```

```
metricForm[sph[r]][u, v]
```

$r^2\, \text{Cos}[v]^2\, du^2 + r^2\, dv^2$

```
parab[s_, t_] := {s, t, t^2 - s^2}
```

```
metricForm[parab][u, v]
```

$\left(1 + 4\, u^2\right) du^2 - 8\, u\, v\, du\, dv + \left(1 + 4\, v^2\right) dv^2$

```
hh[parab][u, v] // Simplify
```

$\sqrt{1 + 4\, u^2 + 4\, v^2}$

infinitesimal Area

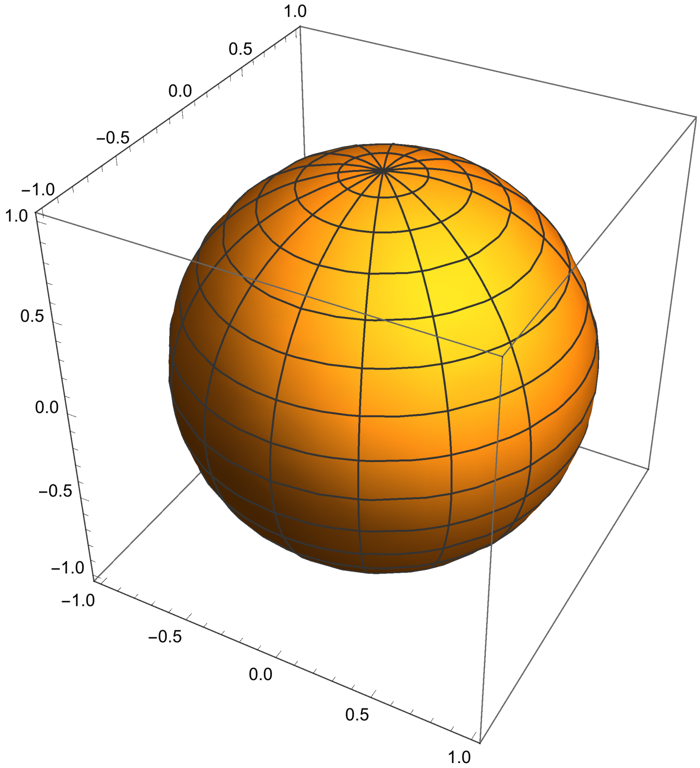
$\texttt{infArea[f\_][u\_, v\_] :=} \sqrt{\text{ee}(f)(u, v)\, \text{gg}(f)(u, v) - (\text{ff}(f)(u, v))^2}$

```
infArea[tor][u, v]
```

$\sqrt{(4 + \text{Cos}[u])^2}$

```
sph[r_][θ_, ϕ_] := r {Cos[θ] Cos[ϕ], Cos[ϕ] Sin[θ], Sin[ϕ]}
```

```
ParametricPlot3D[sph[1][θ, ϕ], {θ, 0, 2 Pi}, {ϕ, -Pi/2, Pi/2}]
```



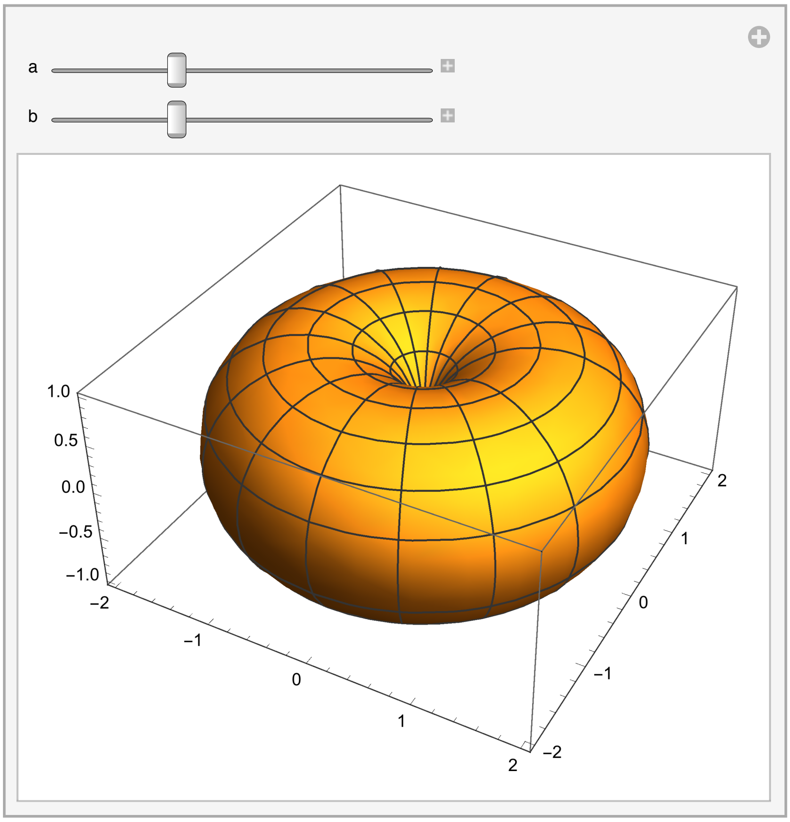

```
Assuming[r > 0,
 Simplify[Integrate[infArea[sph[r]][u, v], {u, 0, 2 Pi}, {v, -Pi/2, Pi/2}]]]
```

$4\pi r^2$

The torus

```
tt[a_, b_][s_, t_] := {Cos[t] (b + a Cos[s]), Sin[t] (b + a Cos[s]), a Sin[s]}
```

```
Manipulate[ParametricPlot3D[tt[a, b][s, t], {s, 0, 2 Pi}, {t, 0, 2 Pi}],
 {{a, 1, "a"}, 0.1, 3}, {{b, 1, "b"}, 0.1, 3}, SaveDefinitions → True]
```




```
Integrate[infArea[tt[a, b]][u, v], {u, 0, 2 Pi},
 {v, 0, 2 Pi}, Assumptions → {a > 0, b > 0, b > a, 0 < u < 2 Pi}]
```

$4\,a\,b\,\pi^2$

```
Simplify[infArea[tt[a, b]][u, v], {a > 0, b > a, 0 < u < 2 Pi}]
```

a (b + a Cos[u])

Second Fundamental Form

$$\text{eee[f\_][u\_, v\_] := Simplify}\left[\frac{\text{Det}\left[\left\{f^{(2,0)}[u,v],\ f^{(1,0)}[u,v],\ f^{(0,1)}[u,v]\right\}\right]}{\sqrt{\text{ee[f][u, v] gg[f][u, v]} - \text{ff[f][u, v]}^2}}\right]$$

$$\text{fff[f\_][u\_, v\_] := Simplify}\left[\frac{\text{Det}\left[\left\{f^{(1,1)}[u,v],\ f^{(1,0)}[u,v],\ f^{(0,1)}[u,v]\right\}\right]}{\sqrt{\text{ee[f][u, v] gg[f][u, v]} - \text{ff[f][u, v]}^2}}\right]$$

$$\text{ggg[f\_][u\_, v\_] := Simplify}\left[\frac{\text{Det}\left[\left\{f^{(0,2)}[u,v],\ f^{(1,0)}[u,v],\ f^{(0,1)}[u,v]\right\}\right]}{\sqrt{\text{ee[f][u, v] gg[f][u, v]} - \text{ff[f][u, v]}^2}}\right]$$

```
ggg[tt[1, 1]][u, v]
```

$\text{Cos}[u]\sqrt{(1 + \text{Cos}[u])^2}$

```
weingarten[f_][u_, v_] := {{eee[f][u, v] * gg[f][u, v] - fff[f][u, v] * ff[f][u, v],
    fff[f][u, v] * ee[f][u, v] - eee[f][u, v] * ff[f][u, v]},
   {fff[f][u, v] * gg[f][u, v] - ggg[f][u, v] * ff[f][u, v],
    ggg[f][u, v] * ee[f][u, v] - fff[f][u, v] * ff[f][u, v]}} /
  (ee[f][u, v] * gg[f][u, v] - ff[f][u, v] ^ 2)
```

```
gcurvature[x_][u_, v_] :=
 Simplify[(Det[{x^(2,0)[u, v], x^(1,0)[u, v], x^(0,1)[u, v]}] Det[{x^(0,2)[u, v],
       x^(1,0)[u, v], x^(0,1)[u, v]}] - Det[{x^(1,1)[u, v], x^(1,0)[u, v], x^(0,1)[u, v]}]^2) /
   (x^(1,0)[u, v].x^(1,0)[u, v] x^(0,1)[u, v].x^(0,1)[u, v] - x^(1,0)[u, v].x^(0,1)[u, v])^2]
```

```
gcurvature[sph[r]][u, v] // FullSimplify
```

$$\frac{1}{r^2}$$

```
mcurvature(x_)(u_, v_) := Simplify[
  (Det[{x^(2,0)[u, v], x^(1,0)[u, v], x^(0,1)[u, v]}] x^(0,1)[u, v].x^(0,1)[u, v] - 2 Det[{x^(1,1)[u, v], x^(1,0)[u, v], x^(0,1)[u, v]}]
      x^(1,0)[u, v].x^(0,1)[u, v] + Det[{x^(0,2)[u, v], x^(1,0)[u, v], x^(0,1)[u, v]}] x^(1,0)[u, v].x^(1,0)[u, v]) /
   (2 (-(x^(1,0)[u, v].x^(0,1)[u, v])^2 + x^(0,1)[u, v].x^(0,1)[u, v] x^(1,0)[u, v].x^(1,0)[u, v])^(3/2))]
```

```
FullSimplify[mcurvature[sph[r]][u, v], Assumptions → {-Pi/2 ≤ v ≤ Pi/2, r > 0}]
```

$$-\frac{1}{r}$$

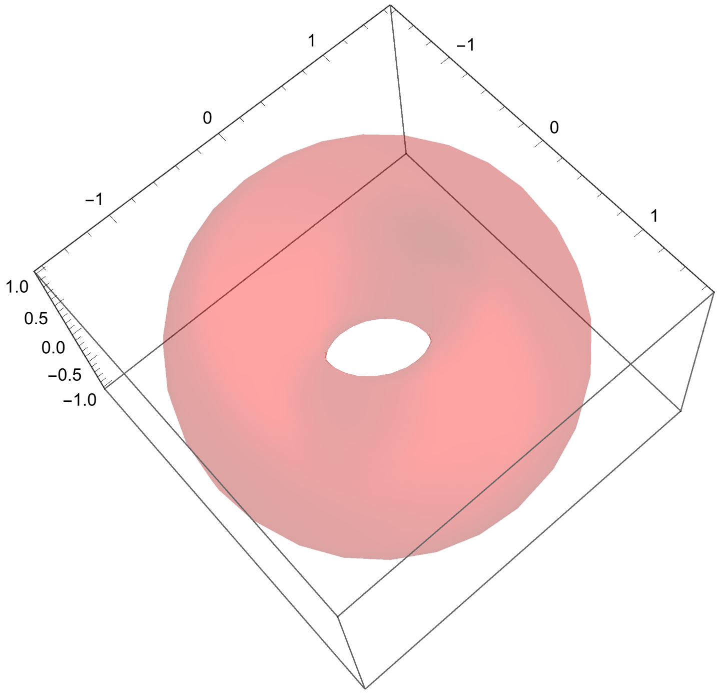
Slow

```
gaussiancurvature[x_][u_, v_] := Simplify[
      (eee[x][u, v]*ggg[x][u, v] - fff[x][u, v]^2)/
       (ee[x][u, v]*gg[x][u, v] -
            ff[x][u, v]^2)]
```

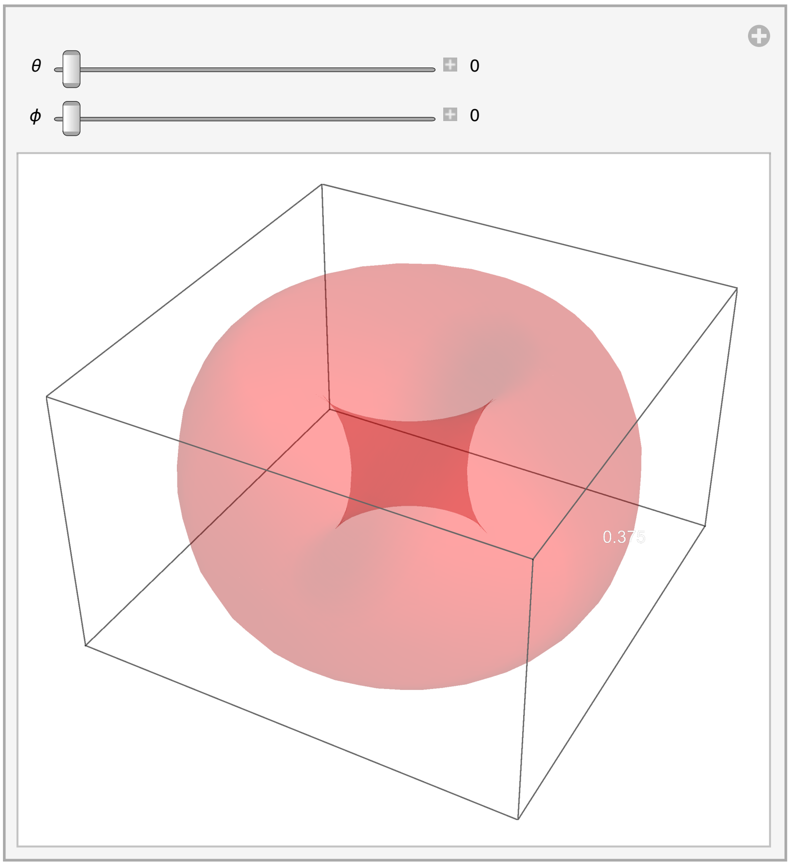
```
mcurvature[x_][u_, v_] := Simplify[
    (eee[x][u, v] * gg[x][u, v] -
         2 * fff[x][u, v] * ff[x][u, v] +
      ggg[x][u, v] * ee[x][u, v]) /
     (2 * (ee[x][u, v] * gg[x][u, v] -
           ff[x][u, v] ^ 2))]
```

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

```
tr = ParametricPlot3D[torus[1, 0.6, 1][u, v], {u, 0, 2 Pi}, {v, 0, 2 Pi}, Mesh → None,
  ColorFunction → Function[{x, y, z}, Directive[Opacity[0.2], Red]]]
```



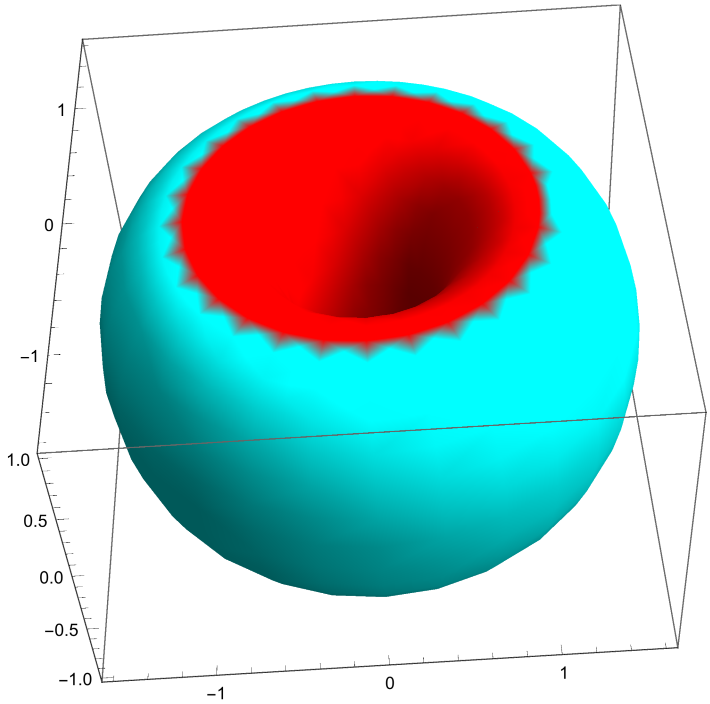

```
Manipulate[Show[Graphics3D[
    Style[Text[gcurvature[torus[1, 0.6, 1]][u, v], torus[1, 0.6, 1][u, v]],
     {White, Thick}]], tr], {{u, 0, "θ"}, 0, 2 Pi, Appearance → "Labeled"},
 {{v, 0, "ϕ"}, 0, 2 Pi, Appearance → "Labeled"}, SaveDefinitions → True]
```



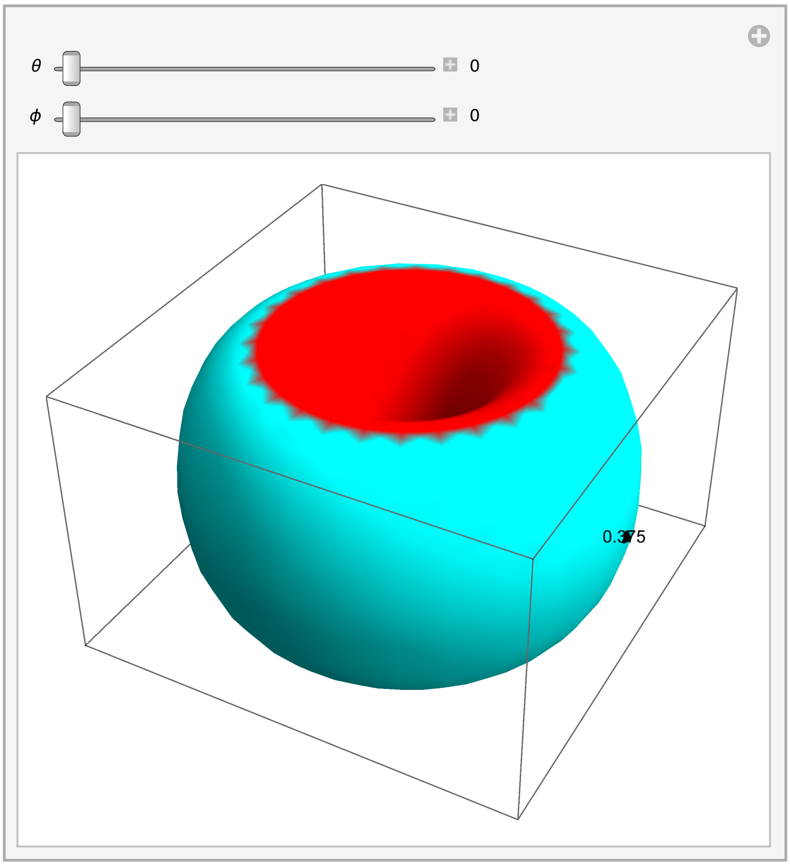
Torus colored by curvature (positive vs negative)

```
gcolorfunction1[u_, v_] := Which[gcurvature[torus[1, 0.6, 1]][u, v] > 0,
   0.5, gcurvature[torus[1, 0.6, 1]][u, v] < 0, 1, True, 0.1];
```

```
tr1 = ParametricPlot3D[torus[1, 0.6, 1][u, v], {u, 0, 2 Pi},
  {v, 0, 2 Pi}, Mesh → None, ColorFunctionScaling → False,
  ColorFunction → Function[{x, y, z, u, v}, Hue[gcolorfunction1[u, v]]]]
```



```
Manipulate[
 Show[{Graphics3D[{PointSize[0.02], Point[torus[1, 0.6, 1][u, v]], Style[
      Text[gcurvature[torus[1, 0.6, 1]][u, v], torus[1, 0.6, 1][u, v]], Black]}],
   tr1}], {{u, 0, "θ"}, 0, 2 Pi, Appearance → "Labeled"},
 {{v, 0, "ϕ"}, 0, 2 Pi, Appearance → "Labeled"},
 SaveDefinitions → True]
```



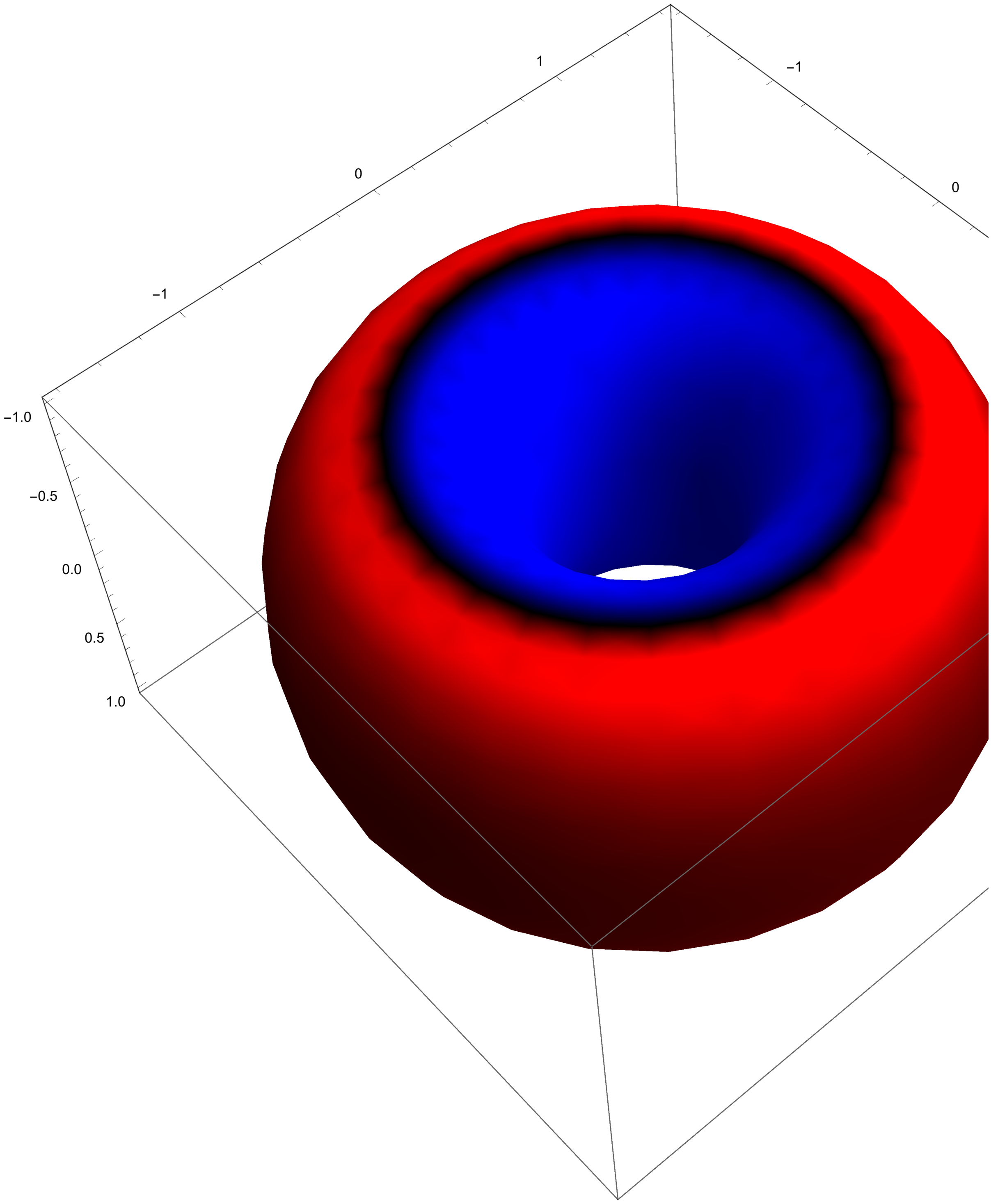
Torus colored by curvature

```
max =
 Max[Table[gcurvature[torus[1, 0.6, 1]][u, v], {u, 0, 2 Pi, 0.1}, {v, 0, 2 Pi, 0.1}]]
0.907106

min =
 Min[Table[gcurvature[torus[1, 0.6, 1]][u, v], {u, 0, 2 Pi, 0.1}, {v, 0, 2 Pi, 0.1}]]
-1.61337

gcolorfunction3[u_, v_] := Which[gcurvature[torus[1, 0.6, 1]][u, v] > 0,
  RGBColor[Rescale[gcurvature[torus[1, 0.6, 1]][u, v], {0, max}], 0, 0],
  gcurvature[torus[1, 0.6, 1]][u, v] < 0,
  RGBColor[0, 0, Rescale[-gcurvature[torus[1, 0.6, 1]][u, v], {0, -min}]],
  True, RGBColor[0, 0, 0]]
```

```
tr3 = ParametricPlot3D[torus[1, 0.6, 1][u, v], {u, 0, 2 Pi},
  {v, 0, 2 Pi}, Mesh → None, ColorFunctionScaling → False,
  ColorFunction → Function[{x, y, z, u, v}, gcolorfunction3[u, v]]]
```



```
Manipulate[Show[Graphics3D[Style[
    Text[gcurvature[torus[1, 0.6, 1]][u, v], torus[1, 0.6, 1][u, v]], White]],
  tr3], {{u, 0, "θ"}, 0, 2 Pi, Appearance → "Labeled"},
 {{v, 0, "ϕ"}, 0, 2 Pi, Appearance → "Labeled"}, SaveDefinitions → True]
```