Matrices, Geometry&*Mathematica*

Authors: Bruce Carpenter, Bill Davis and Jerry Uhl ©2001

Producer: Bruce Carpenter

Version 4 **Publisher: Math Everywhere, Inc.**

MGM.01 Perpendicular Frames
BASICS

Mathematica Initializations

B.5) 3D perpendicular frames. Hanging and aligning in 3D

□ B.5.a.i) 3D perpendicular frames and Euler angles

You dial up a 3D perpendicular frame by setting three angles r, s and t:

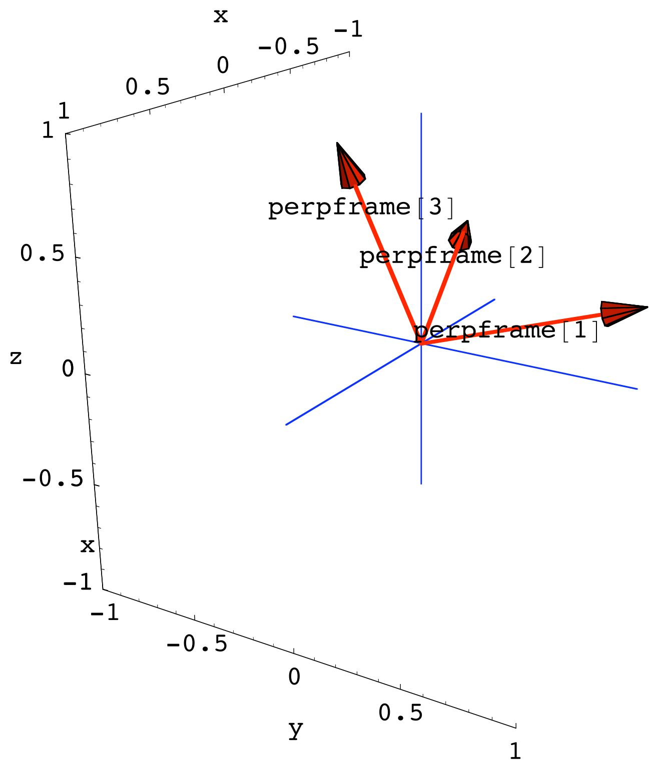
```
Clear[perpframe, r, s, t];
{perpframe[1], perpframe[2], perpframe[3]} =
 {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
   Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t],
   Cos[r] Cos[s] Cos[t] - Sin[r] Sin[t], Cos[r] Sin[s]},
  {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}}
```

```
{{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t],
  Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t], Sin[r] Sin[s]},
 {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] - Sin[r] Sin[t],
  Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}}
```

Here's one:

```
r = Pi / 4;
s = Pi / 8;
t = Pi / 3;
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};
ranger = 1.0;
frameplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[2, 0.1],
```

```
PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
Boxed -> False,
Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



Lots of folks call the angles r, s and t by the name "Euler angles."
What is the physical meaning meaning of the Euler angles r, s and t?

It's really hard to find a part of math that Euler didn't contribute to.

□ **Answer:**

As Euler himself once said, one example gives the idea.

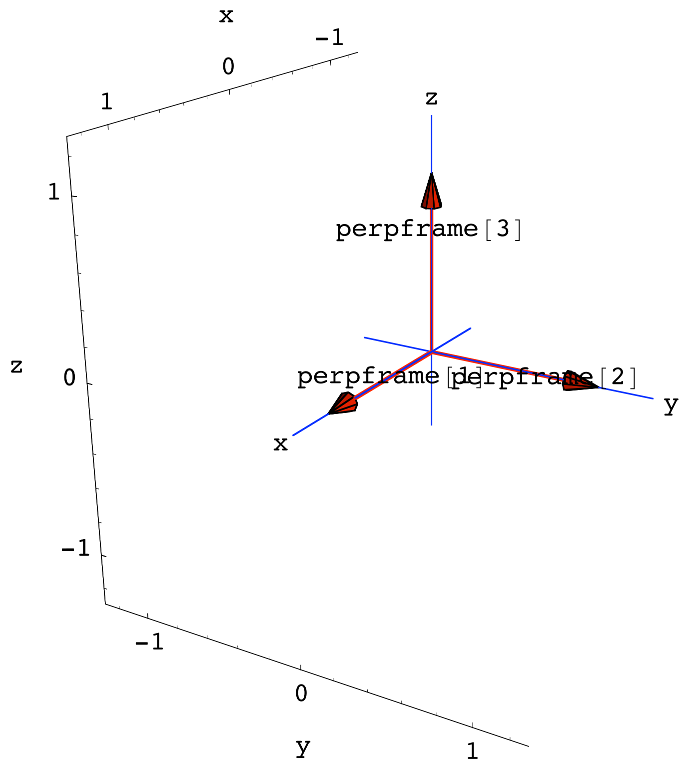
In the example above

$r = \frac{\pi}{4}$; $s = \frac{\pi}{8}$; $t = \frac{\pi}{3}$.

To see how this was built up, start with the usual perpendicular frame pointing out the positive x,y and z axes:

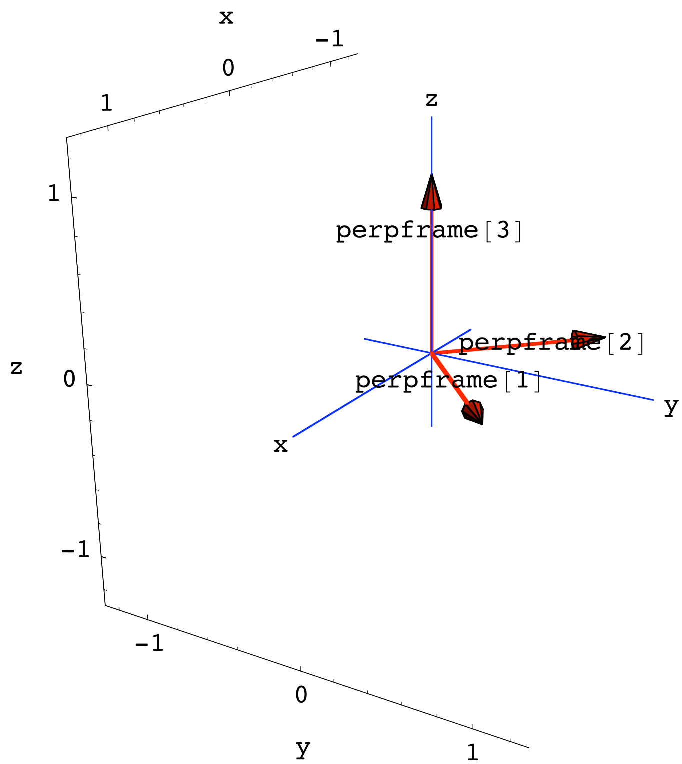
```
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
ranger = 1.3;
orginalframeplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[1.3, 0.1],
```

```
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed -> False,
  Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



To see the effect of r, go with Euler angles $r = \frac{\pi}{4}$, $s = 0$, $t = 0$:

```
r = Pi / 4;
s = 0;
t = 0;
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};


rframeplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[1.3, 0.1],
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed -> False,
  Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



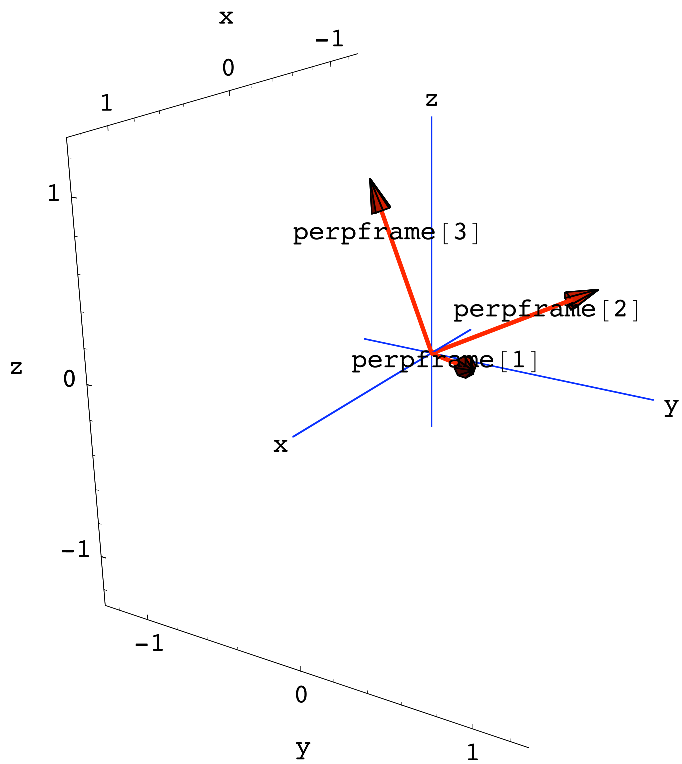
You get the perpendicular frame corresponding to Euler angles

$r = \frac{\pi}{4}$,s = 0 and t = 0 by rotating everything by r radians about the z axis.

To see the combined effect of r and s, go with Euler angles

$r = \frac{\pi}{4}$,$s = \frac{\pi}{8}$ and t = 0 :

```
r = Pi / 4;
s = Pi / 8;
t = 0;
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};
frameplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[1.3, 0.1],
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed -> False,
  Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



You get the perpendicular frame corresponding to r = $\frac{\pi}{4}$,s = $\frac{\pi}{8}$ and t = 0 by

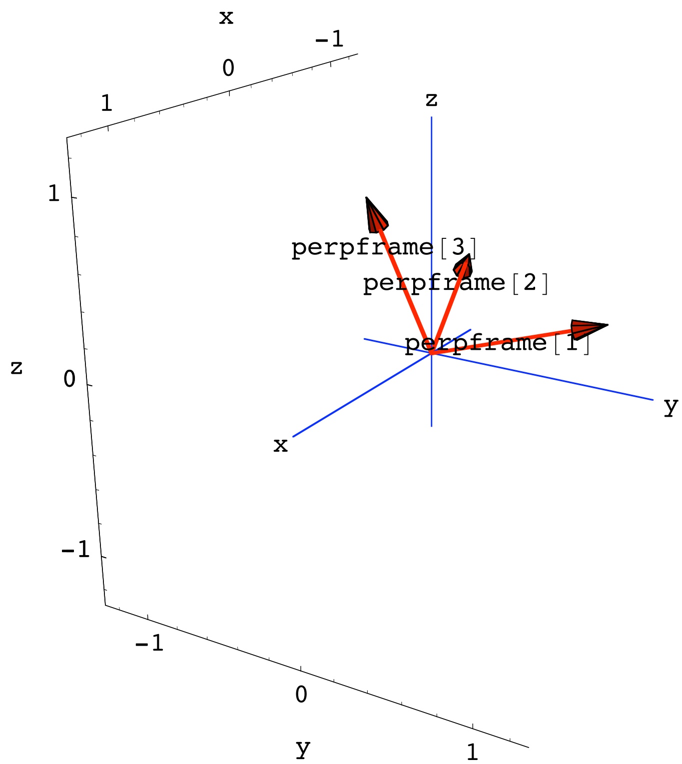
->first rotating everything by r radians about the z axis

-> and then rotating everything by s radians about the x-axis.

To see the combined effect of r, s and t go with Euler angles

r = $\frac{\pi}{4}$,s = $\frac{\pi}{8}$ and t = $\frac{\pi}{3}$:

```
r = Pi / 4;
s = Pi / 8;
t = Pi / 3;
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};
frameplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[1.3, 0.1],
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed -> False,
  Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



You get the perpendicular frame corresponding to r = $\frac{\pi}{4}$,s = $\frac{\pi}{8}$ and t = $\frac{\pi}{3}$ by

->first rotating everything by r radians about the z axis

-> and then rotating everything by s radians about the x-axis

and finally

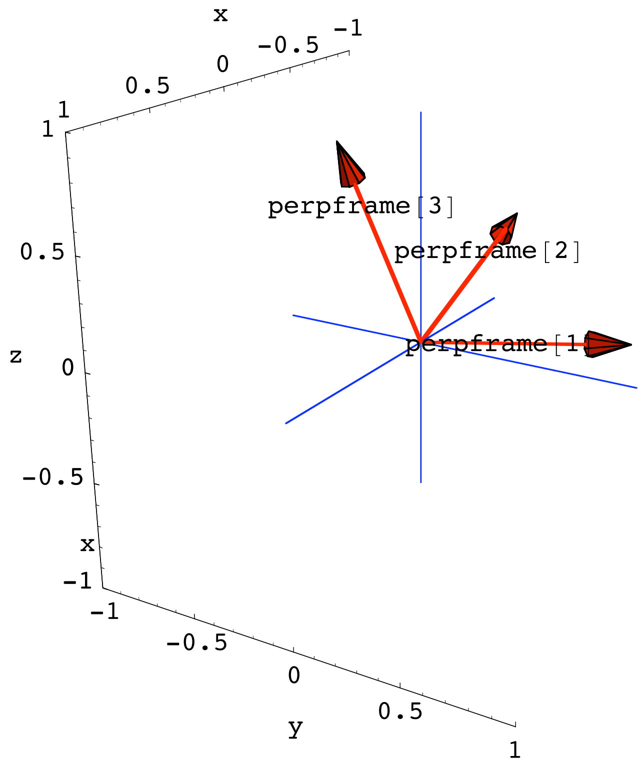
-> then rotating everything by t radians about the z-axis (again).

B.5.b.i) Hanging an ellipsoid on a 3D perpendicular frame

Here's a 3D perpendicular frame:

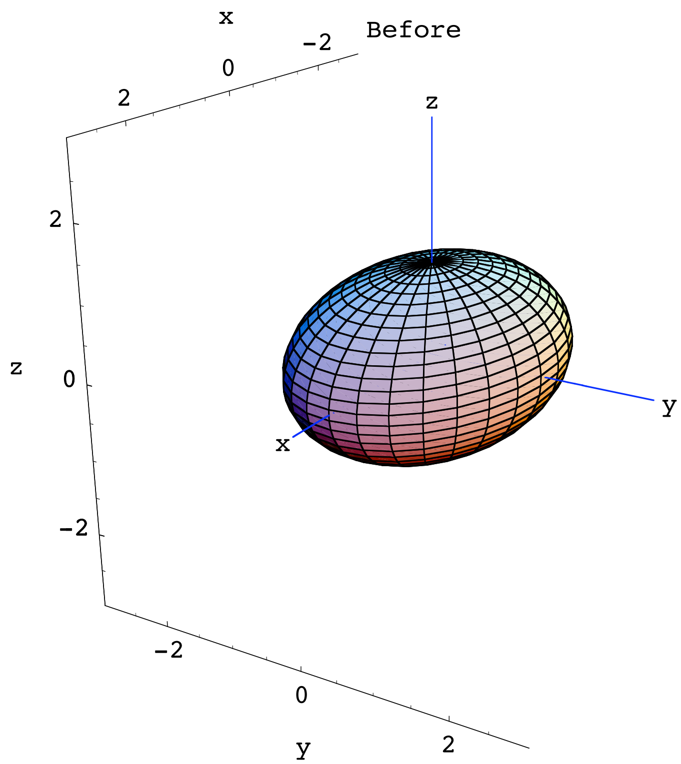
```
r = π/6;
s = π/8;
t = π/3;
Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};
ranger = 1.0;
frameplot =
```

```
Show[Table[Arrow[perpframe[k], Tail → {0, 0, 0}, VectorColor → Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]], Axes3D[2, 0.1],
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed → False, Axes → True, ViewPoint → CMView, AxesLabel → {"x", "y", "z"}];
```



Here's a fat ellipsoid (football) skewered on the the x,y,and z axes:

```
Clear[x, y, z, s, t, pointcolor];
{x[s_, t_], y[s_, t_], z[s_, t_]} = {2.3 Sin[s] Cos[t], 1.6 Sin[s] Sin[t], 1.2 Cos[s]};
{slow, shigh} = {0, π};
{tlow, thigh} = {0, 2 π};
ranger = 3.0;

football = ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, slow, shigh}, {t, tlow, thigh},
   PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
   Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False,
   ViewPoint → CMView, DisplayFunction → Identity];

Show[football, Axes3D[3, 0.2], PlotLabel → "Before", DisplayFunction → $DisplayFunction];
```



Hang this ellipsoid on the plotted perpendicular frame with
perpframe[1] playing the former role of the positive x-axis,
perpframe[2] playing the former role of the positive y-axis, and with
perpframe[3] playing the former role of the positive z-axis.

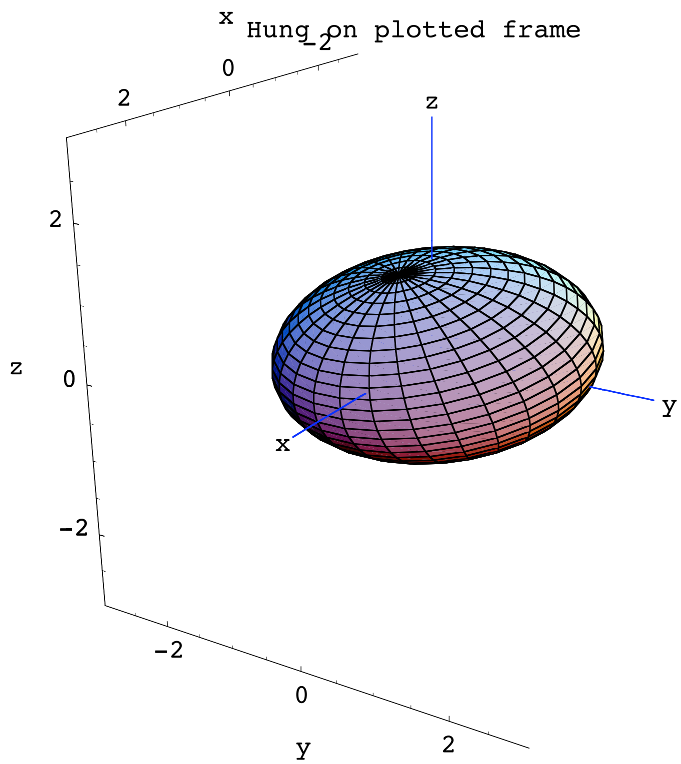
□ **Answer:**

You just take the xyz- parameterization {x[s,t], y[s,t], z[s,t]}, rip off x[s,t] , y[s,t] and z[s,t] and insert them like this

hangplotter[s, t] = .
x[s, t] perpframe[1] + y[s, t] perpframe[2] + z[s, t] perpframe[3]

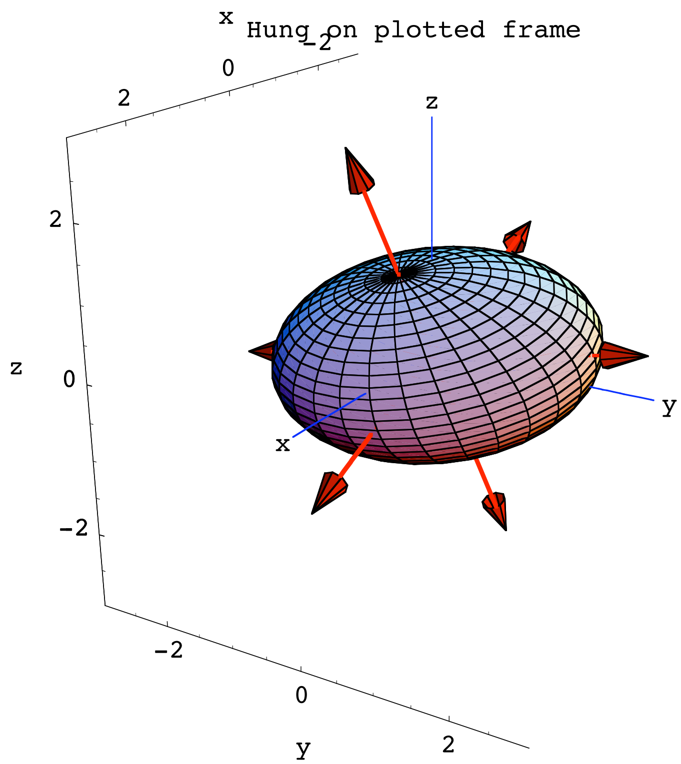
And plot:

```
Clear[hangplotter];
hangplotter[s_, t_] = x[s, t] perpframe[1] + y[s, t] perpframe[2] + z[s, t] perpframe[3];

hungfootball = ParametricPlot3D[hangplotter[s, t], {s, slow, shigh}, {t, tlow, thigh},
   PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
   Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False,
   ViewPoint → CMView, DisplayFunction → Identity];

hungplot = Show[hungfootball, Axes3D[3, 0.2],
   PlotLabel → "Hung on plotted frame", DisplayFunction → $DisplayFunction];
```



See this plot together with scaled versions of the perpendicular frame:

```
scaledframeplot = {Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, ScaleFactor -> 3, VectorColor -> Red],
    {k, 1, 3}], Table[Arrow[-perpframe[k], Tail -> {0, 0, 0},
     ScaleFactor -> 3, VectorColor -> Red], {k, 1, 3}]};
    Show[hungfootball, scaledframeplot, Axes3D[3, 0.2],
  PlotLabel -> "Hung on plotted frame", DisplayFunction -> $DisplayFunction];
```



Grab and animate the last three plots.

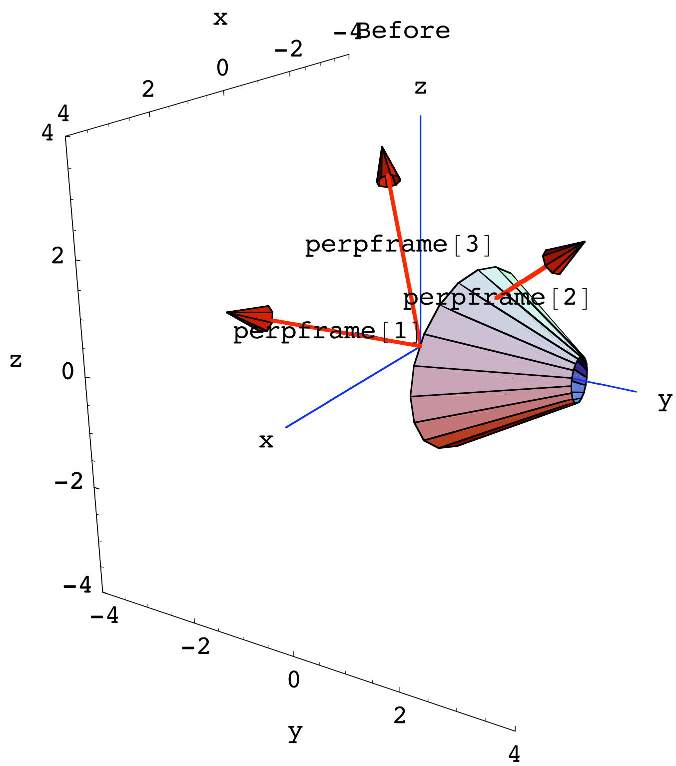
The new football is skewered on the plotted perpendicuar frame the same way that the orginal football is skewered on the x,y, and z axes.

□ B.5.b.ii) Hanging a surface on a 3D perpendicular frame

Here's a surface shown together with a scaled 3D perpendicular frame:

```
Clear[x, y, z, s, t];
{x[s_, t_], y[s_, t_], z[s_, t_]} =
  s {2 Cos[t], 1, 1.5 Sin[t]} + (1 - s) {0.3 Cos[t], 3, 0.4 Sin[t]};

{tlow, thigh} = {0, 2 π};
{slow, shigh} = {0, 1};
ranger = 4;

surfaceplot = ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, slow, shigh},
   {t, tlow, thigh}, PlotPoints → {2, Automatic}, DisplayFunction → Identity];

Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
     Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t],
     Cos[r] Cos[s] Cos[t] - Sin[r] Sin[t], Cos[r] Sin[s]},
    {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}} /. {r → 0.3, s → π/4, t → -π/4};

scaledframeplot =
```

```
 {Table[Arrow[ranger perpframe[k], Tail → {0, 0, 0}, VectorColor → Red], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.5 ranger perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.5 ranger perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.5 ranger perpframe[3]]]};

before = Show[scaledframeplot, surfaceplot, Axes3D[ranger],
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  AspectRatio → Automatic, Axes → True, AxesLabel → {"x", "y", "z"}, ViewPoint → CMView,
  PlotLabel → "Before", Boxed → False, DisplayFunction → $DisplayFunction];
```



Pick up this surface up and hang it on the plotted frame with
perpframe[1] playing the former role of {1,0,0} pointing along the positive x-axis
perpframe[2] playing the former role of {0,1,0} pointing along the positive y-axis
and with
perpframe[3] playing the former role of {0,0,1} pointing along the positive y-axis

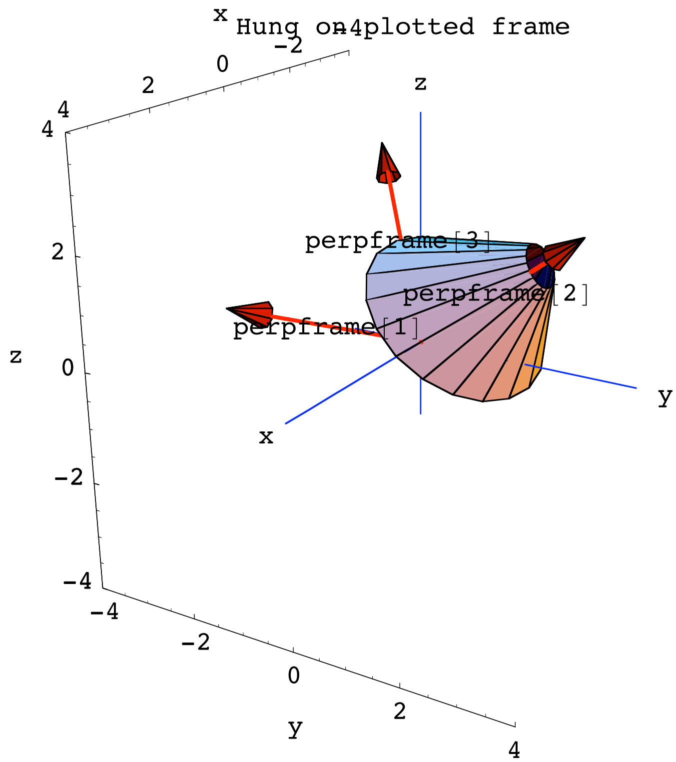
□ **Answer:**

All you do is:

You just take the xyz- parameterization {x[s,t], y[s,t], z[s,t]}, rip off x[s,t] , y[s,t] and z[s,t] and insert them like this

hangplotter[s, t] = .
x[s, t] perpframe[1] + y[s, t] perpframe[2] + z[s, t] perpframe[3]
And plot:

```
Clear[hangplotter];
hangplotter[s_, t_] = x[s, t] perpframe[1] + y[s, t] perpframe[2] + z[s, t] perpframe[3];

hungsurface = ParametricPlot3D[hangplotter[s, t],
   {s, slow, shigh}, {t, tlow, thigh}, PlotPoints → {2, Automatic},
   PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
   Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False,
   ViewPoint → CMView, DisplayFunction → Identity];

hungplot = Show[hungsurface, Axes3D[ranger], scaledframeplot,
   PlotLabel → "Hung on plotted frame", DisplayFunction → $DisplayFunction];
```



Grab both plots and animate.

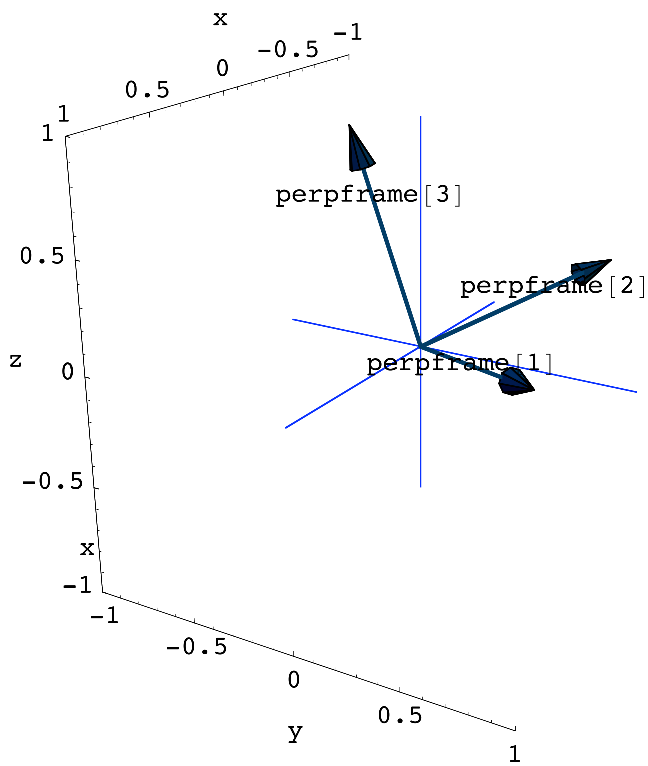
Done.

□ B.5.c.i) Aligning an ellipsoid on the x,y and z axes in 3D

Here's a 3D perpendicular frame:

```
r = 0.6;
s = 0.3;
t = 0.4;
Clear[perpframe];
```

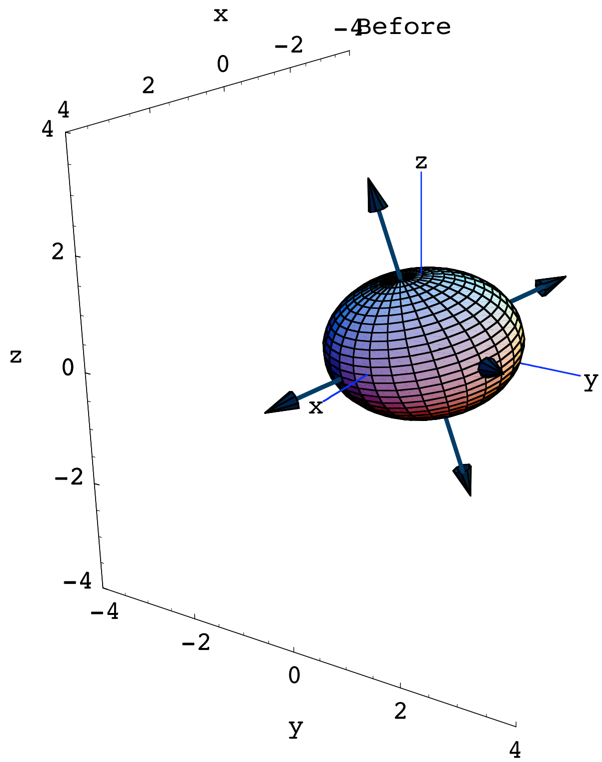
```
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};
ranger = 1.0;
frameplot = Show[Table[
  Arrow[perpframe[k], Tail -> {0, 0, 0}, VectorColor -> Indigo], {k, 1, 3}],
  Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
  Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
  Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]],
  Axes3D[2, 0.1],
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Boxed -> False,
  Axes -> True, ViewPoint -> CMView, AxesLabel -> {"x", "y", "z"}];
```



Here's a fat ellipsoid (football) skewered on the same perpendicular frame:

```
Clear[x, y, z, s, t, pointcolor];
{x[s_, t_], y[s_, t_], z[s_, t_]} =
  2.3 Sin[s] Cos[t] perpframe[1] + 1.6 Sin[s] Sin[t] perpframe[2] + 1.2 Cos[s] perpframe[3];

{slow, shigh} = {0, π};
{tlow, thigh} = {0, 2 π};
ranger = 4.0;

football = ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, slow, shigh}, {t, tlow, thigh},
   PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
   Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False,
   ViewPoint → CMView, DisplayFunction → Identity];

scaledframeplot =
```

```
  {Table[Arrow[perpframe[k], Tail → {0, 0, 0}, ScaleFactor → 3, VectorColor → Indigo],
    {k, 1, 3}], Table[Arrow[-perpframe[k], Tail → {0, 0, 0},
     ScaleFactor → 3, VectorColor → Indigo], {k, 1, 3}]};

Show[football, Axes3D[3, 0.2], scaledframeplot,
  PlotLabel → "Before", DisplayFunction → $DisplayFunction];
```



Align this ellipsoid on the x,y and z- axes with
{1,0,0} pointing along the positive x- axis playing the former role of perpframe[1],
{0,1,0} pointing along the positive y- axis playing the former role of perpframe[2],
{0,0,1} pointing along the positive z- axis playing the former role of perpframe[3].

□ **Answer:**

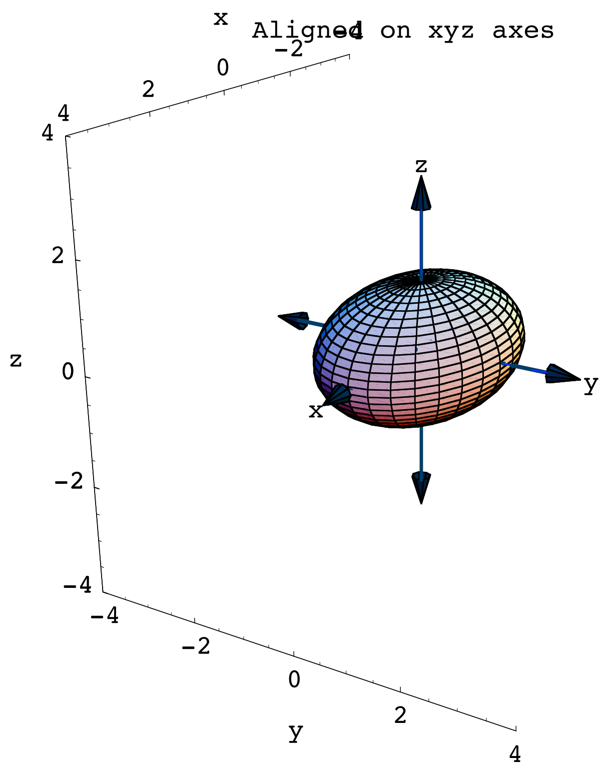
Just as in 2D, you take the given xyz- parameterization {x[s,t],y[s,t],z[s,t]} of the ellipsoid and go with the coordinates of {x[s,t],y[s,t],z[s,t]} and then plot

.

$$\{\{x[s, t], y[s, t], z[s, t]\}.\text{perpframe}[1], \{x[s, t], y[s, t], z[s, t]\}.\text{perpframe}[2], \{x[s, t], y[s, t], z[s, t]\}.\text{perpframe}[3]\} :$$

```
alignedfootball = ParametricPlot3D[
  {{x[s, t], y[s, t], z[s, t]}.perpframe[1], {x[s, t], y[s, t], z[s, t]}.perpframe[2],
   {x[s, t], y[s, t], z[s, t]}.perpframe[3]}, {s, slow, shigh}, {t, tlow, thigh},
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False,
  ViewPoint → CMView, DisplayFunction → Identity];
xyzperpframe[1] = {1, 0, 0};
xyzperpframe[2] = {0, 1, 0};
xyzperpframe[3] = {0, 0, 1};

scaledxyzunitvectors =
  {Table[Arrow[xyzperpframe[k], Tail → {0, 0, 0}, ScaleFactor → 3, VectorColor → Indigo],
    {k, 1, 3}], Table[Arrow[-xyzperpframe[k], Tail → {0, 0, 0},
     ScaleFactor → 3, VectorColor → Indigo], {k, 1, 3}]};


Show[alignedfootball, Axes3D[3, 0.2], scaledxyzunitvectors,
  PlotLabel → "Aligned on xyz axes", DisplayFunction → $DisplayFunction];
```

Aligned on xyz axes



Grab both plots and animate.

There you go.

This ellipsoid is now aligned on the x,y and z- axes with

{1,0,0} pointing along the positive x- axis playing the former role of perpframe[1],

{0,1,0} pointing along the positive y- axis playing the former role of perpframe[2],

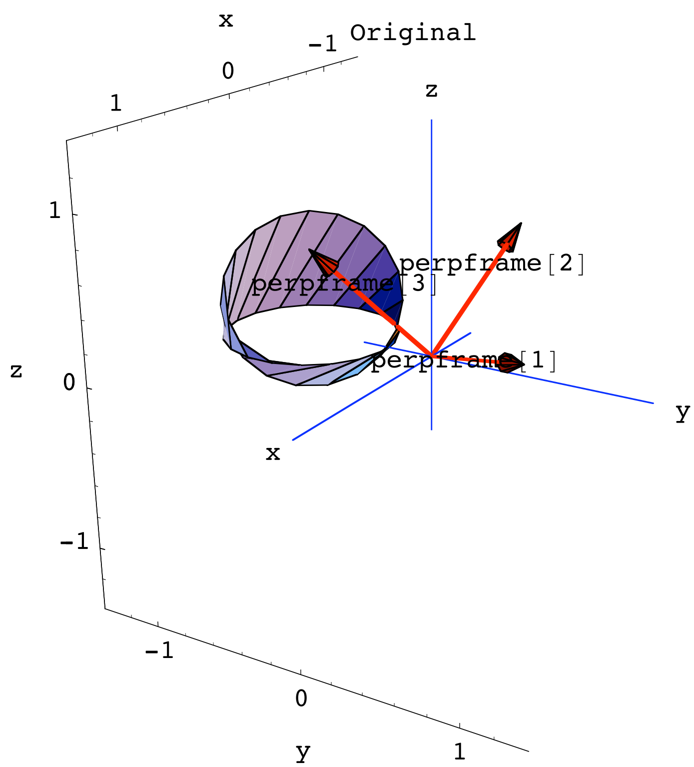
{0,0,1} pointing along the positive z- axis playing the former role of perpframe[3].

B.5.c.ii) Aligning a surface on the x,y and z axes in 3D

Here's a 3D surface shown with a plot of a 3D perpendicular frame::

```
r = Pi / 8;
s = Pi / 4;
t = Pi / 4

Clear[perpframe];
{perpframe[1], perpframe[2], perpframe[3]} =
  {{Cos[r] Cos[t] - Cos[s] Sin[r] Sin[t], Cos[s] Cos[t] Sin[r] + Cos[r] Sin[t],
    Sin[r] Sin[s]}, {-Cos[t] Sin[r] - Cos[r] Cos[s] Sin[t], Cos[r] Cos[s] Cos[t] -
     Sin[r] Sin[t], Cos[r] Sin[s]}, {Sin[s] Sin[t], -Cos[t] Sin[s], Cos[s]}};

ranger = 1.0;
scaledframeplot =
  {Table[Arrow[perpframe[k], Tail → {0, 0, 0}, VectorColor → Red], {k, 1, 3}],
   Graphics3D[Text["perpframe[1]", 0.4 perpframe[1]]],
   Graphics3D[Text["perpframe[2]", 0.7 perpframe[2]]],
   Graphics3D[Text["perpframe[3]", 0.7 perpframe[3]]]};

Clear[x, y, z, s, t];
{x[s_, t_], y[s_, t_], z[s_, t_]} = 0.5 s (Cos[t] perpframe[1] + Sin[t] perpframe[2]) +
   perpframe[3] - 0.5 {Cos[t], Sin[t], Cos[s]};

{slow, shigh} = {0, 1};
{tlow, thigh} = {0, 2 π};

ranger = 1.4;
surfaceplot = ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, slow, shigh},
   {t, tlow, thigh}, PlotPoints → {2, Automatic}, DisplayFunction → Identity];

before = Show[surfaceplot, Axes3D[ranger], scaledframeplot,
   PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
   Axes → True, AxesLabel → {"x", "y", "z"}, Boxed → False, PlotLabel → "Original",
   ViewPoint → CMView, DisplayFunction → $DisplayFunction];
```

$\frac{\pi}{4}$



Align this surface on the x,y and z- axes with
{1,0,0} pointing along the positive x- axis playing the former role of perpframe[1],
{0,1,0} pointing along the positive y- axis playing the former role of perpframe[2],
{0,0,1} pointing along the positive z- axis playing the former role of perpframe[3].

□ **Answer:**

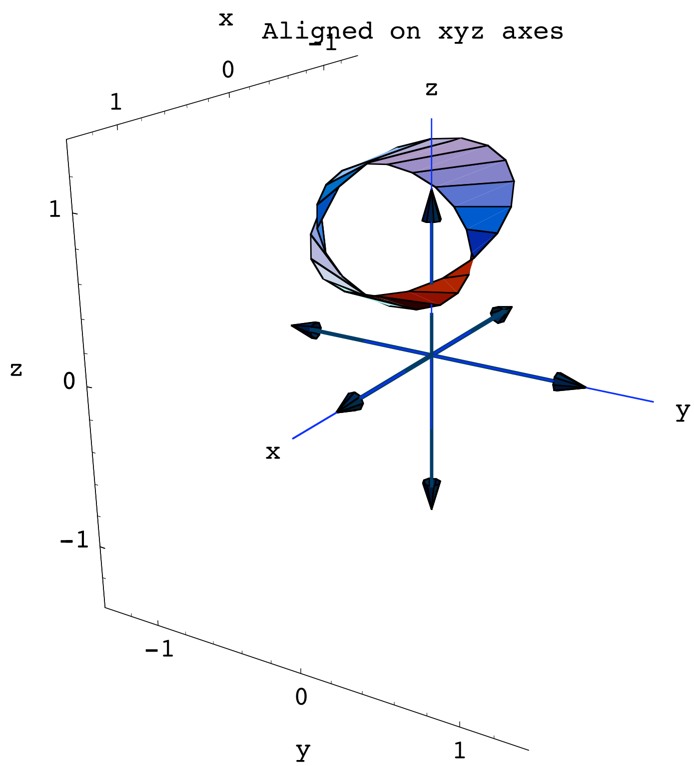
Just as in 2D, you take the given xyz- parameterization {x[s,t],y[s,t],z[s,t]} of the surface and go with the coordinates of {x[s,t],y[s,t],z[s,t]}realitive to the perpendicular frame

.

$$\{\{x[s,t], y[s,t], z[s,t]\}.\text{perpframe}[1],\\ \{x[s,t], y[s,t], z[s,t]\}.\text{perpframe}[2],\\ \{x[s,t], y[s,t], z[s,t]\}.\text{perpframe}[3]\}\ :$$

And then plot:

```
alignedsurface = ParametricPlot3D[
  {{x[s, t], y[s, t], z[s, t]}.perpframe[1], {x[s, t], y[s, t], z[s, t]}.perpframe[2],
   {x[s, t], y[s, t], z[s, t]}.perpframe[3]}, {s, slow, shigh}, {t, tlow, thigh},
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}, {-ranger, ranger}},
  Axes → True, AxesLabel → {"x", "y", "z"}, PlotPoints -> {2, Automatic},
  Boxed → False, ViewPoint → CMView, DisplayFunction → Identity];
xyzperpframe[1] = {1, 0, 0};
xyzperpframe[2] = {0, 1, 0};
xyzperpframe[3] = {0, 0, 1};

xyzunitvectors =
  {Table[Arrow[xyzperpframe[k], Tail → {0, 0, 0}, VectorColor → Indigo], {k, 1, 3}],
   Table[Arrow[-xyzperpframe[k], Tail → {0, 0, 0}, VectorColor → Indigo], {k, 1, 3}]};


after = Show[alignedsurface, xyzunitvectors, Axes3D[ranger],
   PlotLabel → "Aligned on xyz axes", DisplayFunction → $DisplayFunction];
```



```
Grab both plots,align amd animate.
```

Now the z-axis pierces the aligned surfrace just the way perpframe[3] pierces the orginal surface.
Done.