■ First define each of the axis-aligned rotation matrices as a function of some angle:

```
In[1]:= rx[θ_] := {{1, 0, 0}, {0, Cos[θ], -Sin[θ]}, {0, Sin[θ], Cos[θ]}};
```

$$rx[\theta\_] := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}[\theta] & -\text{Sin}[\theta] \\ 0 & \text{Sin}[\theta] & \text{Cos}[\theta] \end{pmatrix};$$

$$ry[\theta\_] := \begin{pmatrix} \text{Cos}[\theta] & 0 & \text{Sin}[\theta] \\ 0 & 1 & 0 \\ -\text{Sin}[\theta] & 0 & \text{Cos}[\theta] \end{pmatrix};$$

$$rz[\theta\_] := \begin{pmatrix} \text{Cos}[\theta] & -\text{Sin}[\theta] & 0 \\ \text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

■ Now, given Euler angles and assuming the rotation order X, Y, Z with fixed axes, we can convert from Euler angles to a matrix with:

```
In[4]:= eulertomatrix[θx_, θy_, θz_] := rz[θz].ry[θy].rx[θx]
```

■ We can now ask *Mathematica* to write out the matrix for an arbitrary set of input angles:

```
In[5]:= MatrixForm[eulertomatrix[θx, θy, θz]]
```

Out[5]//MatrixForm=

$$\begin{pmatrix} \text{Cos}[\theta y]\,\text{Cos}[\theta z] & \text{Cos}[\theta z]\,\text{Sin}[\theta x]\,\text{Sin}[\theta y] - \text{Cos}[\theta x]\,\text{Sin}[\theta z] & \text{Cos}[\theta x]\,\text{Cos}[\theta z]\,\text{Sin}[\theta y] + \text{Sin}[\theta x \\ \text{Cos}[\theta y]\,\text{Sin}[\theta z] & \text{Cos}[\theta x]\,\text{Cos}[\theta z] + \text{Sin}[\theta x]\,\text{Sin}[\theta y]\,\text{Sin}[\theta z] & -\text{Cos}[\theta z]\,\text{Sin}[\theta x] + \text{Cos}[\theta x]\,\text{Sin}[\theta \\ -\text{Sin}[\theta y] & \text{Cos}[\theta y]\,\text{Sin}[\theta x] & \text{Cos}[\theta x]\,\text{Cos}[\theta y] \end{pmatrix}$$

■ Using the notation Cx to mean Cos[*θ*x], Sz for Sin[*θ*z] and so forth, we can makethe above parameterized matrix look a little cleaner:

```
In[6]:= MatrixForm[eulertomatrix[θx, θy, θz] /.
          {Cos[θx] → Cx, Cos[θy] → Cy, Cos[θz] → Cz, Sin[θx] → Sx, Sin[θy] → Sy, Sin[θz] → Sz}]
```

Out[6]//MatrixForm=

$$\begin{pmatrix} \text{Cy Cz} & \text{Cz Sx Sy} - \text{Cx Sz} & \text{Cx Cz Sy} + \text{Sx Sz} \\ \text{Cy Sz} & \text{Cx Cz} + \text{Sx Sy Sz} & -\text{Cz Sx} + \text{Cx Sy Sz} \\ -\text{Sy} & \text{Cy Sx} & \text{Cx Cy} \end{pmatrix}$$

■ Also, show what the matrix looks like for *θ*y= ±90°

Note that this also shows how gimbal lock kills one degree of freedom.

```
In[7]:= MatrixForm[FullSimplify[eulertomatrix[θx, θy, θz] /. {Cos[θy] → 0, Sin[θy] → 1}]]
        MatrixForm[FullSimplify[eulertomatrix[θx, θy, θz] /. {Cos[θy] → 0, Sin[θy] → -1}]]
```

Out[7]//MatrixForm=

$$\begin{pmatrix} 0 & \mathrm{Sin}[\theta x - \theta z] & \mathrm{Cos}[\theta x - \theta z] \\ 0 & \mathrm{Cos}[\theta x - \theta z] & -\mathrm{Sin}[\theta x - \theta z] \\ -1 & 0 & 0 \end{pmatrix}$$

Out[8]//MatrixForm=

$$\begin{pmatrix} 0 & -\mathrm{Sin}[\theta x + \theta z] & -\mathrm{Cos}[\theta x + \theta z] \\ 0 & \mathrm{Cos}[\theta x + \theta z] & -\mathrm{Sin}[\theta x + \theta z] \\ 1 & 0 & 0 \end{pmatrix}$$

■ Now we can use the above to work out formulae for the Euler angles if we have the matrix:

Note that I think ArcTan takes arguments in reverse order from the C function atan2.

```
In[9]:= matrixtoeuler[m_] := Block[{x, y, z, cy, ϵ = 0.000001},
          y = ArcSin[-m⟦3, 1⟧];
          If[(Abs[m⟦3, 1⟧] - 1) < -ϵ,
           z = ArcTan[m⟦1, 1⟧, m⟦2, 1⟧];
           x = ArcTan[m⟦3, 3⟧, m⟦3, 2⟧];
           ,
           x = ArcTan[-m⟦1, 3⟧ / m⟦3, 1⟧, -m⟦1, 2⟧ / m⟦3, 1⟧];
           z = 0;
          ];
          {x, y, z}
         ]
```

■ Test for some random value

```
In[10]:= tm = eulertomatrix[.4, .9, -1.2];
         MatrixForm[tm]
         {tx, ty, tz} = matrixtoeuler[tm]
         MatrixForm[tmm = eulertomatrix[tx, ty, tz]]
         Norm[tm - tmm, ∞]
```

Out[11]//MatrixForm=

$$\begin{pmatrix} 0.225245 & 0.968999 & -0.101515 \\ -0.579365 & 0.0494427 & -0.813567 \\ -0.783327 & 0.242066 & 0.572541 \end{pmatrix}$$

Out[12]= {0.4, 0.9, -1.2}

Out[13]//MatrixForm=

$$\begin{pmatrix} 0.225245 & 0.968999 & -0.101515 \\ -0.579365 & 0.0494427 & -0.813567 \\ -0.783327 & 0.242066 & 0.572541 \end{pmatrix}$$

Out[14]= 0.

▪ Test for some random value where middle axis is at 90°

In[15]:=
```
tm = eulertomatrix[1.4, π / 2, -.9];
MatrixForm[tm]
{tx, ty, tz} = matrixtoeuler[tm]
MatrixForm[tmm = eulertomatrix[tx, ty, tz]]
Norm[tm - tmm, ∞]
```

Out[16]//MatrixForm=

$$\begin{pmatrix} 0. & 0.745705 & -0.666276 \\ 0. & -0.666276 & -0.745705 \\ -1. & 0. & 0. \end{pmatrix}$$

Out[17]= {2.3, 1.5708, 0}

Out[18]//MatrixForm=

$$\begin{pmatrix} 6.12303\times10^{-17} & 0.745705 & -0.666276 \\ 0. & -0.666276 & -0.745705 \\ -1. & 4.56598\times10^{-17} & -4.07963\times10^{-17} \end{pmatrix}$$

Out[19]= 1.72253×10^{-16}

In[20]:=
```
tm = eulertomatrix[-.6, -π / 2, -1.9];
MatrixForm[tm]
{tx, ty, tz} = matrixtoeuler[tm]
MatrixForm[tmm = eulertomatrix[tx, ty, tz]]
Norm[tm - tmm, ∞]
```

Out[21]//MatrixForm=

$$\begin{pmatrix} 0. & 0.598472 & 0.801144 \\ 0. & -0.801144 & 0.598472 \\ 1. & 0. & 0. \end{pmatrix}$$

Out[22]= {-2.5, -1.5708, 0}

Out[23]//MatrixForm=

$$\begin{pmatrix} 6.12303\times10^{-17} & 0.598472 & 0.801144 \\ 0. & -0.801144 & 0.598472 \\ 1. & -3.66446\times10^{-17} & -4.90543\times10^{-17} \end{pmatrix}$$

Out[24]= 8.56989×10^{-17}

■ Test for some more values using a set of loops

```
In[25]:= Do[
          Do[
           Do[

            tm = eulertomatrix[θx, θy, θz];
            {tx, ty, tz} = matrixtoeuler[tm];
            tmm = eulertomatrix[tx, ty, tz];

             err = Norm[tm - tmm, ∞];

            If[err > 0.0001, Print["Bad: (e=", err, ") ", θx, " ", θy, " ", θz]];

            , {θz, 0., 2 π, 2 π / 32}]
           , {θy, 0., 2 π, 2 π / 32}]
          , {θx, 0., 2 π, 2 π / 32}];
```