
Mechanics - Euler's Equations and Euler Angles | 11-08-18

N. T. Gladd

Initialization: Be sure the file *NTGUtilityFunctions.m* is in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

In[16]:=

```
SetDirectory[NotebookDirectory[]];  
(* set directory where source files are located *)  
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Background

This is the sixth in a series of Mathematica notebooks on classical mechanics. This series was motivated by a close reading and problem solving project I undertook in 2014. The focus of my attention was the text *Introduction to Classical Mechanics with Problems and Solutions*, by David Morin. This is a good book from which to learn and has a great collection of problems. I purchased it and recommend that those with interests in this topic acquire it for their library. I do note that an earlier version can be found on the web. This year, when I returned to this project, I decided to focus on generating Mathematica notebooks on material covered in *Chapter 9 Angular Momentum, Part II (General \vec{L})*, which deals the 3-D rigid body dynamics. This topic is notorious difficult/confusing for students and I felt I just skimmed by as a graduate student. I return in retirement after all these years to pay my dues and really understand how to solve problems in this area.

Although Morin's Chapter 9 guides these notebooks, I made frequent use of other sources such as textbooks available in libraries or on the web. I also found lots of video lectures available on YouTube. Confused about some physics topics? Google it and you'll be amazed what you find. Some relevant texts are

Classical Mechanics, Hebert Goldstein (my original text at University, late 60s). Newer versions exist.

Mechanics: Volume 1 A Course in Theoretical Physics, L. D. Landau and E. M. Lifshitz.

Classical Mechanics, John. R. Taylor

Classical Mechanics of Particles and Systems, Stephen T. Thornton, Jerry B. Marion

Analytical Mechanics, G. R. Fowles, G. L. Cassiday

Analytical Mechanics, Louis N. Hand, Janet D. Finch

I find Mathematica useful for this topic. It facilitates calculations, provides a vehicle for creating instructive visualizations and allows one to quickly generate numerical solutions. Mathematica is a favorite tool of mine but I think it is crucially important to also work with pen and paper. Our brains are closely linked to our hands and one thinks differently with a pen in hand than when sitting before a computer screen. For serious thoughts on this, read *The Craftsman*, by Richard Sennett.

Purpose

I develop the topics of Euler's equations and Euler Angles.

I Euler's Equations

The dynamical equations describing the motion of rigid bodies take a particularly simple form when expressed in principle coordinates

$$\vec{\tau} = \frac{d\vec{L}}{dt} = \frac{\partial \vec{L}}{\partial t} + \vec{\omega} \times \vec{L}$$

$$\vec{L} = \mathbb{I} \cdot \vec{\omega}$$

Express \vec{L} and $\vec{\omega}$ in terms of principle components. I use double stroke variable to denote vector quantities. There is no double stroke form of ω so I use w .

The above forms lead to an easy hand calculation but I illustrate how it could be systematically accomplished using Mathematica

$$\mathbf{w1[1]} = \mathbb{L}[t] = L_1[t] \mathbf{e}_1 + L_2[t] \mathbf{e}_2 + L_3[t] \mathbf{e}_3$$

$$\mathbb{L}[t] = \mathbf{e}_1 L_1[t] + \mathbf{e}_2 L_2[t] + \mathbf{e}_3 L_3[t]$$

$$\mathbf{w1[2]} = \mathbf{w}[t] = \omega_1[t] \mathbf{e}_1 + \omega_2[t] \mathbf{e}_2 + \omega_3[t] \mathbf{e}_3$$

$$\mathbf{w}[t] = \mathbf{e}_1 \omega_1[t] + \mathbf{e}_2 \omega_2[t] + \mathbf{e}_3 \omega_3[t]$$

$$\mathbf{w1[3]} = \mathbb{T} = \tau_1[t] \mathbf{e}_1 + \tau_2[t] \mathbf{e}_2 + \tau_3[t] \mathbf{e}_3$$

$$\mathbb{T} = \mathbf{e}_1 \tau_1[t] + \mathbf{e}_2 \tau_2[t] + \mathbf{e}_3 \tau_3[t]$$

I can transform to a list representation with

$$\mathbf{w1}[1] /. \mathbf{a_e1} + \mathbf{b_e2} + \mathbf{c_e3} \rightarrow \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$$

$$\mathbb{L}[\mathbf{t}] == \{\mathbf{L}_1[\mathbf{t}], \mathbf{L}_2[\mathbf{t}], \mathbf{L}_3[\mathbf{t}]\}$$

I represent the inertial tensor in the list form of vectors convenient for Mathematica operations with a rule. Alternatively, I could have used dyadic form by introducing rules for the manipulation of unit vectors. I think using a dyadic approach is not worth the effort in this simple context.

$$\mathbf{w1}[4] = \mathbf{II} == \{\{\mathbf{I}_1, \mathbf{0}, \mathbf{0}\}, \{\mathbf{0}, \mathbf{I}_2, \mathbf{0}\}, \{\mathbf{0}, \mathbf{0}, \mathbf{I}_3\}\}$$

$$\mathbf{II} == \{\{\mathbf{I}_1, \mathbf{0}, \mathbf{0}\}, \{\mathbf{0}, \mathbf{I}_2, \mathbf{0}\}, \{\mathbf{0}, \mathbf{0}, \mathbf{I}_3\}\}$$

I also have

$$\mathbf{w1}[5] = \mathbb{L}[\mathbf{t}] == \text{Dot}[\mathbf{II}, \mathbf{w}[\mathbf{t}]] /. (\mathbf{w1}[4] /. \text{Equal} \rightarrow \text{Rule}) /. (\mathbf{w1}[2] /. \mathbf{a_e1} + \mathbf{b_e2} + \mathbf{c_e3} \rightarrow \{\mathbf{a}, \mathbf{b}, \mathbf{c}\} /. \text{Equal} \rightarrow \text{Rule})$$

$$\mathbb{L}[\mathbf{t}] == \{\mathbf{I}_1 \omega_1[\mathbf{t}], \mathbf{I}_2 \omega_2[\mathbf{t}], \mathbf{I}_3 \omega_3[\mathbf{t}]\}$$

$$\mathbf{w1}[5] = \mathbf{D}[\mathbf{w1}[1], \mathbf{t}][[1]] ==$$

$$\mathbf{D}[\mathbf{w1}[5][[2]], \mathbf{t}] + \text{Cross}[\mathbf{w1}[2][[2]] /. \mathbf{a_e1} + \mathbf{b_e2} + \mathbf{c_e3} \rightarrow \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, \mathbf{w1}[5][[2]]]$$

$$\mathbf{L}'[\mathbf{t}] == \{-\mathbf{I}_2 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_3 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_1 \omega_1'[\mathbf{t}],$$

$$\mathbf{I}_1 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] - \mathbf{I}_3 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_2 \omega_2'[\mathbf{t}], -\mathbf{I}_1 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_2 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_3 \omega_3'[\mathbf{t}]\}$$

$$\mathbf{w1}[6] = \mathbf{w1}[5] /. \mathbf{w1}[5][[1]] \rightarrow \mathbf{w1}[3][[2]] /. \mathbf{a_e1} + \mathbf{b_e2} + \mathbf{c_e3} \rightarrow \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$$

$$\{\tau_1[\mathbf{t}], \tau_2[\mathbf{t}], \tau_3[\mathbf{t}]\} == \{-\mathbf{I}_2 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_3 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_1 \omega_1'[\mathbf{t}],$$

$$\mathbf{I}_1 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] - \mathbf{I}_3 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_2 \omega_2'[\mathbf{t}], -\mathbf{I}_1 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_2 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_3 \omega_3'[\mathbf{t}]\}$$

or

$$\mathbf{w1}[7] = \text{Table}[\mathbf{w1}[6][[1, \mathbf{i}]] == \mathbf{w1}[6][[2, \mathbf{i}]], \{\mathbf{i}, 1, 3\}]$$

$$\{\tau_1[\mathbf{t}] == -\mathbf{I}_2 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_3 \omega_2[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_1 \omega_1'[\mathbf{t}],$$

$$\tau_2[\mathbf{t}] == \mathbf{I}_1 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] - \mathbf{I}_3 \omega_1[\mathbf{t}] \omega_3[\mathbf{t}] + \mathbf{I}_2 \omega_2'[\mathbf{t}],$$

$$\tau_3[\mathbf{t}] == -\mathbf{I}_1 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_2 \omega_1[\mathbf{t}] \omega_2[\mathbf{t}] + \mathbf{I}_3 \omega_3'[\mathbf{t}]\}$$

$$(\text{TraditionalForm} /@ \mathbf{w1}[7]) // \text{ColumnForm}$$

$$\tau_1(\mathbf{t}) = \mathbf{I}_1 \omega_1'(\mathbf{t}) - \mathbf{I}_2 \omega_2(\mathbf{t}) \omega_3(\mathbf{t}) + \mathbf{I}_3 \omega_2(\mathbf{t}) \omega_3(\mathbf{t})$$

$$\tau_2(\mathbf{t}) = \mathbf{I}_2 \omega_2'(\mathbf{t}) + \mathbf{I}_1 \omega_1(\mathbf{t}) \omega_3(\mathbf{t}) - \mathbf{I}_3 \omega_1(\mathbf{t}) \omega_3(\mathbf{t})$$

$$\tau_3(\mathbf{t}) = \mathbf{I}_3 \omega_3'(\mathbf{t}) - \mathbf{I}_1 \omega_1(\mathbf{t}) \omega_2(\mathbf{t}) + \mathbf{I}_2 \omega_1(\mathbf{t}) \omega_2(\mathbf{t})$$

2 Euler Angles

Definition and visualization

Euler angles (ϕ, θ, ψ) are a way of representing the relationship between the body frame of reference or principal axes, typically denoted by coordinate names (x_1, x_2, x_3) and unit vectors $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$ and space/lab/inertial coordinates, typically denoted by coordinates (x, y, z) and unit vectors $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$. I start with a visualization of these angles.

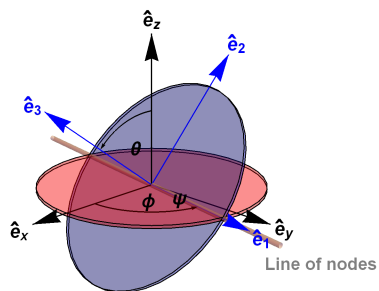
```
MManipulate [
  ShowEulerAngles[ $\phi$ ,  $\theta$ ,  $\psi$ ],
  {{ $\phi$ ,  $\pi / 6.$ , " $\phi$ "},  $\theta$ ,  $2 \pi$ ,  $0.1$ , Appearance  $\rightarrow$  "Labeled"},
  {{ $\theta$ ,  $\pi / 6.$ , " $\theta$ "},  $\theta$ ,  $\pi$ ,  $0.1$ , Appearance  $\rightarrow$  "Labeled"},
  {{ $\psi$ ,  $\theta$ , " $\psi$ "},  $\theta$ ,  $2 \pi$ ,  $0.1$ , Appearance  $\rightarrow$  "Labeled"}, ControlPlacement  $\rightarrow$  Left]
```

In[18]:=

```
DynamicModule[{ $\theta = 0.9^\circ$ ,  $\phi = 1.4000000000000001^\circ$ ,  $\psi = 0.^\circ$ }, ShowEulerAngles[ $\phi$ ,  $\theta$ ,  $\psi$ ]]
```

Euler Angles

$$\frac{d\vec{\phi}}{dt} = \dot{\phi} \hat{e}_z, \quad \frac{d\vec{\theta}}{dt} = \dot{\theta} \hat{e}_1, \quad \frac{d\vec{\psi}}{dt} = \dot{\psi} \hat{e}_3$$



Out[18]=

Show example of Euler angles in the context of an ellipsoid

```

MManipulate[
  ShowEllipsoid[ $\phi$ ,  $\theta$ ,  $\psi$ , a, b],
  {{a, 0.5, "a"}, 0.1, 3, 0.1, Appearance → "Labeled"},
  {{b, 1, "b"}, 0.1, 3, 0.1, Appearance → "Labeled"},
  {{ $\phi$ , 0, " $\phi$ "}, 0, 2  $\pi$ , 0.1, Appearance → "Labeled"},
  {{ $\theta$ , 0, " $\theta$ "}, 0,  $\pi$ , 0.1, Appearance → "Labeled"},
  {{ $\psi$ , 0, " $\psi$ "}, 0, 2  $\pi$ , 0.1, Appearance → "Labeled"}, ControlPlacement → Left]

```

In[19]=

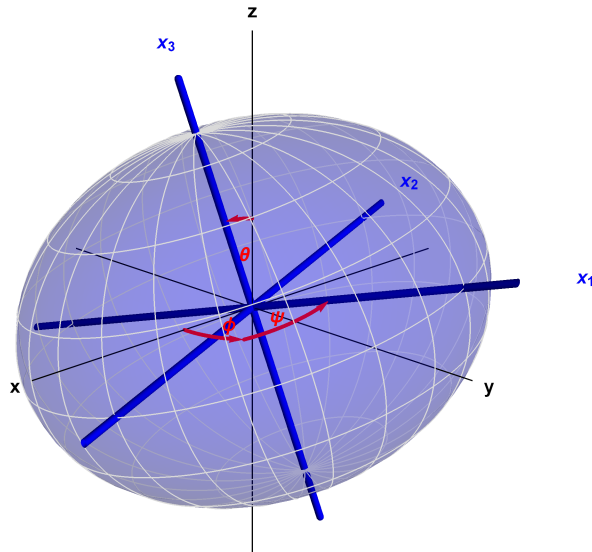
```

DynamicModule[{a = 1.3000000000000003`, b = 1,  $\theta$  = 0.3000000000000004`,
 $\phi$  = 0.7000000000000001`,  $\psi$  = 1.1`}, ShowEllipsoid[ $\phi$ ,  $\theta$ ,  $\psi$ , a, b]]

```

Body and space frames
Oblate ellipsoid

Out[19]=



3 Derivation of Euler angles

Three rotations transforms between space coordinates and body coordinates. In summary

space coordinates (x, y, z)

$$\{\hat{e}_x, \hat{e}_y, \hat{e}_z\} \rightarrow (\text{rot } \phi \text{ about } \hat{e}_z) \rightarrow \{\hat{a}_1, \hat{a}_2, \hat{a}_3\} \quad d\vec{\phi}/dt = \dot{\phi} \hat{e}_z$$

$$\{\hat{a}_1, \hat{a}_2, \hat{a}_3\} \rightarrow (\text{rot } \theta \text{ about } \hat{a}_1) \rightarrow \{\hat{b}_1, \hat{b}_2, \hat{b}_3\} \quad d\vec{\theta}/dt = \dot{\theta} \hat{a}_1$$

$$\{\hat{b}_1, \hat{b}_2, \hat{b}_3\} \rightarrow (\text{rot } \psi \text{ about } \hat{b}_3) \rightarrow \{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \quad d\vec{\psi}/dt = \dot{\psi} \hat{b}_3$$

body coordinates (x_1, x_2, x_3)

Calculation - Start with unit vectors in the inertial/space/lab frame

In[3]=

```
{ex, ey, ez} = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}
```

Out[3]=

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

Rotate counterclockwise by ϕ about the e_z axis.

```
R = RotationMatrix;
R1 = R[phi, ez];
R1 // MatrixForm
```

$$\begin{pmatrix} \cos[\phi] & -\sin[\phi] & 0 \\ \sin[\phi] & \cos[\phi] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
{a1, a2, a3} = Dot[R1, #] & /@ {ex, ey, ez}
```

```
{{Cos[phi], Sin[phi], 0}, {-Sin[phi], Cos[phi], 0}, {0, 0, 1}}
```

Second rotation is by θ around the a_1 axis.

```
R2 = Simplify[R[theta, a1], {phi ∈ Reals}] /. 1/Abs[Sec[phi]]^2 → 1/Sec[phi]^2;
R2 // MatrixForm
```

$$\begin{pmatrix} \cos[\phi]^2 + \cos[\theta] \sin[\phi]^2 & (\cos[\phi] - \cos[\theta] \cos[\phi]) \sin[\phi] & \sin[\theta] \sin[\phi] \\ \frac{1}{2} (1 - 2 \cos[\theta] \cos[\phi]^2 + \cos[2\phi]) \tan[\phi] & \cos[\theta] \cos[\phi]^2 + \sin[\phi]^2 & -\cos[\phi] \sin[\theta] \\ -\sin[\theta] \sin[\phi] & \cos[\phi] \sin[\theta] & \cos[\theta] \end{pmatrix}$$

```
{b1, b2, b3} = Dot[R2, #] & /@ {a1, a2, a3} // Simplify
```

```
{{Cos[phi], Sin[phi], 0}, {-Cos[theta] Sin[phi], Cos[theta] Cos[phi], Sin[theta]},
{Sin[theta] Sin[phi], -Cos[phi] Sin[theta], Cos[theta]}}
```

Third rotation is by ψ around the b_3 axis.

```
R3 = Simplify[R[ψ, b3], {φ ∈ Reals, θ ∈ Reals, ψ ∈ Reals}] /.
Abs[Csc[φ]]2 → Csc[φ]2 /. 1/Abs[Csc[φ]]2 → 1/Csc[φ]2
```

```
{ {Cos[φ]2 Cos[ψ] + Cos[θ]2 Cos[ψ] Sin[φ]2 + Sin[θ]2 Sin[φ]2,
Cos[φ] Cos[ψ] Sin[φ] - Cos[θ]2 Cos[φ] Cos[ψ] Sin[φ] - Cos[φ] Sin[θ]2 Sin[φ] -
Cos[θ] Sin[ψ], Sin[2θ] Sin[φ] Sin[ψ/2]2 - Cos[φ] Sin[θ] Sin[ψ] },
{Cos[φ] Cos[ψ] Sin[φ] - Cos[θ]2 Cos[φ] Cos[ψ] Sin[φ] - Cos[φ] Sin[θ]2 Sin[φ] +
Cos[θ] Sin[ψ], Cos[θ]2 Cos[φ]2 Cos[ψ] + Cos[φ]2 Sin[θ]2 + Cos[ψ] Sin[φ]2,
Sin[θ] (-2 Cos[θ] Cos[φ] Sin[ψ/2]2 - Sin[φ] Sin[ψ]) },
{Sin[2θ] Sin[φ] Sin[ψ/2]2 + Cos[φ] Sin[θ] Sin[ψ],
Sin[θ] (-2 Cos[θ] Cos[φ] Sin[ψ/2]2 + Sin[φ] Sin[ψ]) }, 1/2 (1 + Cos[2θ] + 2 Cos[ψ] Sin[θ]2) }
```

Finally {e1, e2, e3} represent units vectors in the body/principle coordinates

```
{e1, e2, e3} = Dot[R3, #] & /@ {b1, b2, b3} // Simplify;
{e1, e2, e3} // MatrixForm
```

```
( Cos[φ] Cos[ψ] - Cos[θ] Sin[φ] Sin[ψ]   Cos[ψ] Sin[φ] + Cos[θ] Cos[φ] Sin[ψ]   Sin[θ] Sin[ψ]
 -Cos[θ] Cos[ψ] Sin[φ] - Cos[φ] Sin[ψ]   Cos[θ] Cos[φ] Cos[ψ] - Sin[φ] Sin[ψ]   Cos[ψ] Sin[θ]
 Sin[θ] Sin[φ]                               -Cos[φ] Sin[θ]                               Cos[θ] )
```

Mathematica has a built-in function EulerMatrix

```
EM = EulerMatrix[{φ, θ, ψ}, {3, 1, 3}]
```

```
{ {Cos[φ] Cos[ψ] - Cos[θ] Sin[φ] Sin[ψ],
-Cos[θ] Cos[ψ] Sin[φ] - Cos[φ] Sin[ψ], Sin[θ] Sin[φ] },
{Cos[ψ] Sin[φ] + Cos[θ] Cos[φ] Sin[ψ], Cos[θ] Cos[φ] Cos[ψ] - Sin[φ] Sin[ψ],
-Cos[φ] Sin[θ] }, {Sin[θ] Sin[ψ], Cos[ψ] Sin[θ], Cos[θ] } }
```

which provides an alternate calculation of the body unit vectors. The second argument represents the sequence of axes about which the rotations are performed. The flexibility in the design of EulerMatrix accommodates the various conventions for Euler rotations that exist in the literature.

```
{e1Alt, e2Alt, e3Alt} = Dot[EM, #] & /@ {ex, ey, ez}
```

```
{ {Cos[φ] Cos[ψ] - Cos[θ] Sin[φ] Sin[ψ],
Cos[ψ] Sin[φ] + Cos[θ] Cos[φ] Sin[ψ], Sin[θ] Sin[ψ] },
{-Cos[θ] Cos[ψ] Sin[φ] - Cos[φ] Sin[ψ], Cos[θ] Cos[φ] Cos[ψ] - Sin[φ] Sin[ψ],
Cos[ψ] Sin[θ] }, {Sin[θ] Sin[φ], -Cos[φ] Sin[θ], Cos[θ] } }
```

I check that this calculation of the unit vectors agrees with the original calculation

```
MapThread[({#1 - #2}) &, {{e1, e2, e3}, {e1Alt, e2Alt, e3Alt}}]
{{{0, 0, 0}}, {{0, 0, 0}}, {{0, 0, 0}}}
```

A third way of performing this calculation is by forming a *Composition* of a list of *RotationTransform* representing the sequence of ϕ , θ and ψ rotations.

```
TF = Apply[Composition, {RotationTransform[phi, {0, 0, 1}],
  RotationTransform[theta, {1, 0, 0}], RotationTransform[psi, {0, 0, 1}]}]

TransformationFunction[


|                              |                                 |
|------------------------------|---------------------------------|
| Cos[phi] Cos[psi] -          | -Cos[theta] Cos[psi] Sin[phi] - |
| Cos[theta] Sin[phi] Sin[psi] | Cos[phi] Sin[psi]               |
| Cos[psi] Sin[phi] +          | Cos[theta] Cos[phi] Cos[psi] -  |
| Cos[theta] Cos[phi] Sin[psi] | Sin[phi] Sin[psi]               |
| Sin[theta] Sin[psi]          | Cos[psi] Sin[theta]             |
| 0                            | 0                               |


```

```
{e1Alt2, e2Alt2, e3Alt2} = TF /@ {ex, ey, ez}
```

```
{{Cos[phi] Cos[psi] - Cos[theta] Sin[phi] Sin[psi],
  Cos[psi] Sin[phi] + Cos[theta] Cos[phi] Sin[psi], Sin[theta] Sin[psi]},
 {-Cos[theta] Cos[psi] Sin[phi] - Cos[phi] Sin[psi], Cos[theta] Cos[phi] Cos[psi] - Sin[phi] Sin[psi],
  Cos[psi] Sin[theta]}, {Sin[theta] Sin[phi], -Cos[phi] Sin[theta], Cos[theta]}}
```

The result of this second alternative also agrees with the original calculation

```
MapThread[({#1 - #2}) &, {{e1, e2, e3}, {e1Alt2, e2Alt2, e3Alt2}}]
{{{0, 0, 0}}, {{0, 0, 0}}, {{0, 0, 0}}}
```

3 Angular frequencies and Euler Angles

Relations between the angular frequencies in the body frame and the rotations about the Euler axes are frequently needed for calculations

$$\vec{\omega} = \omega_1 e_1 + \omega_2 e_2 + \omega_3 e_3 = \dot{\phi} + \dot{\theta} + \dot{\psi} = \dot{\phi} e_z + \dot{\theta} a_1 + \dot{\psi} b_3$$

```
w3[1] = {phiDotVec, thetaDotVec, psiDotVec} = {phiDot ez, thetaDot a1, psiDot b3}
```

```
{{0, 0, phiDot}, {thetaDot Cos[phi], thetaDot Sin[phi], 0},
 {psiDot Sin[theta] Sin[phi], -psiDot Cos[phi] Sin[theta], psiDot Cos[theta]}}
```

The specific form of, say, $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$, in body coordinates can be easily calculated


```

φDotBody = Dot[φDotVec, #] & /@ {e1, e2, e3} // Simplify
{φDot Sin[θ] Sin[ψ], φDot Cos[ψ] Sin[θ], φDot Cos[θ]}

```

```

θDotBody = Dot[θDotVec, #] & /@ {e1, e2, e3} // Simplify
{θDot Cos[ψ], -θDot Sin[ψ], 0}

```

```

ψDotBody = Dot[ψDotVec, #] & /@ {e1, e2, e3} // Simplify
{0, 0, ψDot}

```

Then, in body coordinates,

```

ωVecBody = φDotBody + θDotBody + ψDotBody;
ωVecBody // MatrixForm

$$\begin{pmatrix} \theta \dot{\cos}[\psi] + \phi \dot{\sin}[\theta] \sin[\psi] \\ \phi \dot{\cos}[\psi] \sin[\theta] - \theta \dot{\sin}[\psi] \\ \psi \dot{\cos}[\theta] + \phi \dot{\cos}[\theta] \end{pmatrix}$$


```

I express this result in more familiar notation

```

ωBody = Thread[Equal[{ω1, ω2, ω3}, {ωVecBody[[1]], ωVecBody[[2]], ωVecBody[[3]]}]] /.
{ω1 → ω1, ω2 → ω2, ω3 → ω3, θDot → θ̇, φDot → φ̇, ψDot → ψ̇}
{ω1 = Cos[ψ] θ̇ + φ̇ Sin[θ] Sin[ψ], ω2 = Cos[ψ] φ̇ Sin[θ] - θ̇ Sin[ψ], ω3 = Cos[θ] φ̇ + ψ̇}

```

```

(TraditionalForm /@ ωBody) // ColumnForm
ω1 = θ̇ cos(ψ) + φ̇ sin(θ) sin(ψ)
ω2 = φ̇ sin(θ) cos(ψ) - θ̇ sin(ψ)
ω3 = φ̇ cos(θ) + ψ̇

```

Pixels are free, so I also calculate $\vec{\omega}$ in space coordinates

```

φDotSpace = Dot[φDotVec, #] & /@ {ex, ey, ez} // Simplify;
θDotSpace = Dot[θDotVec, #] & /@ {ex, ey, ez} // Simplify;
ψDotSpace = Dot[ψDotVec, #] & /@ {ex, ey, ez} // Simplify;

```

```

ωVecSpace = φDotSpace + θDotSpace + ψDotSpace;
ωVecSpace // MatrixForm

$$\begin{pmatrix} \theta \dot{\cos}[\phi] + \psi \dot{\sin}[\theta] \sin[\phi] \\ -\psi \dot{\cos}[\phi] \sin[\theta] + \theta \dot{\sin}[\phi] \\ \phi \dot{\cos}[\theta] + \psi \dot{\cos}[\theta] \end{pmatrix}$$


```

```

ωSpace =
Thread[Equal[{ω1, ω2, ω3}, {ωVecSpace[[1]], ωVecSpace[[2]], ωVecSpace[[3]]}]] /.
{ω1 → ω1, ω2 → ω2, ω3 → ω3, θDot → θ̇, φDot → φ̇, ψDot → ψ̇}
{ω1 == Cos[φ] θ̇ + ψ̇ Sin[θ] Sin[φ], ω2 == -Cos[φ] ψ̇ Sin[θ] + θ̇ Sin[φ], ω3 == φ̇ + Cos[θ] ψ̇}

```

```
(TraditionalForm /@ ωSpace) // ColumnForm
```

```

ω1 = ψ̇ sin(θ) sin(φ) + θ̇ cos(φ)
ω2 = θ̇ sin(φ) - ψ̇ sin(θ) cos(φ)
ω3 = ψ̇ cos(θ) + φ̇

```

4 Time dependence of body frame unit vectors

I quickly derive the time-dependence of the unit vectors in body coordinates in terms of time-independent space coordinates

```
w4[1] = ee1[t] == RotationMatrix[φ[t], ez].ex
```

```
ee1[t] == {Cos[φ[t]], Sin[φ[t]], 0}
```

```

w4[2] = ee3[t] == RotationMatrix[θ[t], ee1[t]].ez /. (w4[1] /. Equal → Rule);
w4[2] = Simplify[w4[2], {φ[t] ∈ Reals}] /. Abs[x_]² → x² /. 1/Abs[x_]² → 1/x²

```

```
ee3[t] == {Sin[θ[t]] Sin[φ[t]], -Cos[φ[t]] Sin[θ[t]], Cos[θ[t]}}
```

```

w4[3] = ee2[t] == Cross[ee3[t], ee1[t]] /. (w4[1] /. Equal → Rule) /.
(w4[2] /. Equal → Rule);

```

```
w4[3] = Simplify[w4[3], Trig → True]
```

```
ee2[t] == {-Cos[θ[t]] Sin[φ[t]], Cos[θ[t]] Cos[φ[t]], Sin[θ[t]}}
```

With Mathematica list operations finished, I switch to a more convenient notation

```

w4[4] =
{e1v[t] == Dot[ee1[t], {exv, eyv, ezv}], e2v[t] == Dot[ee2[t], {exv, eyv, ezv}],
e3v[t] == Dot[ee3[t], {exv, eyv, ezv}]} /. ({w4[1], w4[2], w4[3]} // ER) /.
{e1v → e1, e2v → e2, e3v → e3, exv → ex, eyv → ey, ezv → ez}

```

```

{e1[t] == Cos[φ[t]] ex + Sin[φ[t]] ey,
e2[t] == -Cos[θ[t]] Sin[φ[t]] ex + Cos[θ[t]] Cos[φ[t]] ey + Sin[θ[t]] ez,
e3[t] == Sin[θ[t]] Sin[φ[t]] ex - Cos[φ[t]] Sin[θ[t]] ey + Cos[θ[t]] ez}

```

For future convenience, I calculate some explicit expressions for the space unit vectors

```
w4[5] = Solve[w4[4], {e_x, e_y, e_z}][[1]];
w4[5] = Simplify[w4[5], Trig -> True]
```

$$\left\{ \begin{aligned} e_x &\rightarrow \cos[\phi[t]] e_1[t] + \sin[\phi[t]] (-\cos[\theta[t]] e_2[t] + \sin[\theta[t]] e_3[t]), \\ e_y &\rightarrow \sin[\phi[t]] (e_1[t] + \cot[\phi[t]] (\cos[\theta[t]] e_2[t] - \sin[\theta[t]] e_3[t])), \\ e_z &\rightarrow \sin[\theta[t]] e_2[t] + \cos[\theta[t]] e_3[t] \end{aligned} \right\}$$

I take the time derivative of the body unit vectors

```
w4[6] = MapEqn[D[#, t] &, w4[4]];
w3[6] = Collect[w4[6], {e_x, e_y, e_z, phi'[t], theta'[t], psi'[t]}]
```

$$\left\{ \begin{aligned} e_1'[t] &= -\sin[\phi[t]] e_x \phi'[t] + \cos[\phi[t]] e_y \phi'[t], \\ e_2'[t] &= \cos[\theta[t]] e_z \theta'[t] + e_x (\sin[\theta[t]] \sin[\phi[t]] \theta'[t] - \cos[\theta[t]] \cos[\phi[t]] \phi'[t]) + \\ &e_y (-\cos[\phi[t]] \sin[\theta[t]] \theta'[t] - \cos[\theta[t]] \sin[\phi[t]] \phi'[t]), e_3'[t] = \\ &-\sin[\theta[t]] e_z \theta'[t] + e_x (\cos[\theta[t]] \sin[\phi[t]] \theta'[t] + \cos[\phi[t]] \sin[\theta[t]] \phi'[t]) + \\ &e_y (-\cos[\theta[t]] \cos[\phi[t]] \theta'[t] + \sin[\theta[t]] \sin[\phi[t]] \phi'[t]) \end{aligned} \right\}$$

I used w3[5] to remove dependence on space unit vectors

```
w4[7] = w4[6] /. w4[5];
w4[7] = MapEqn[Simplify, w4[7]];
w4[7] = Collect[w4[7], {e1[t], e2[t], e3[t], phi'[t], theta'[t], psi'[t]}]
```

$$\left\{ \begin{aligned} e_1'[t] &= \cos[\theta[t]] e_2[t] \phi'[t] - \sin[\theta[t]] e_3[t] \phi'[t], \\ e_2'[t] &= e_3[t] \theta'[t] - \cos[\theta[t]] e_1[t] \phi'[t], e_3'[t] = -e_2[t] \theta'[t] + \sin[\theta[t]] e_1[t] \phi'[t] \end{aligned} \right\}$$

This last two expression can be written in more traditional form by suppressing the explicit time dependence.

```
w4[8] = w4[7] /. {e1[t] -> e1, e2[t] -> e2, e3[t] -> e3, e1'[t] -> e1dot, e2'[t] -> e2dot,
  e3'[t] -> e3dot, phi[t] -> phi, theta[t] -> theta, psi[t] -> psi, phi'[t] -> phidot, theta'[t] -> thetadot, psi'[t] -> psidot}
```

$$\left\{ \begin{aligned} \dot{e}_1 &= \cos[\theta] \dot{\phi} e_2 - \dot{\phi} \sin[\theta] e_3, \dot{e}_2 = -\cos[\theta] \dot{\phi} e_1 + \dot{\theta} e_3, \dot{e}_3 = \dot{\phi} \sin[\theta] e_1 - \dot{\theta} e_2 \end{aligned} \right\}$$

```
(TraditionalForm /@ w4[8]) // ColumnForm
```

$$\begin{aligned} \dot{e}_1 &= e_2 \dot{\phi} \cos(\theta) - e_3 \dot{\phi} \sin(\theta) \\ \dot{e}_2 &= e_3 \dot{\theta} - e_1 \dot{\phi} \cos(\theta) \\ \dot{e}_3 &= e_1 \dot{\phi} \sin(\theta) - e_2 \dot{\theta} \end{aligned}$$

This latter form is consistent with Morin (9.37).

Visualizations

Code for Section 2

In[4]:=

```

Clear[ShowEulerAngles];
ShowEulerAngles[φ_, θ_, ψ_] :=
Module[
  {0, ex, ey, ez, axesxyz, e1, e2, e3, axes123, refPointStartx, refPointFinx, φarc,
   θarc, ψarc, circleSpace, circleBody, lineOfNodes, RM = RotationMatrix, lab, G},
  {0, ex, ey, ez} = {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
  axesxyz = {Black, Arrowheads[Small], Arrow[{0, ex}], Tex["ê_x", 1.1 ex],
    Arrow[{0, ey}], Tex["ê_y", 1.1 ey], Arrow[{0, ez}], Tex["ê_z", 1.1 ez]};
  axes123 = {Blue, Arrowheads[Small], Arrow[{0, e1}], Tex["ê_1", 1.1 e1],
    Arrow[{0, e2}], Tex["ê_2", 1.1 e2], Arrow[{0, e3}], Tex["ê_3", 1.1 e3]};

  e1 = (RM[φ, ez].ex);
  φarc = {Arrowheads[0.01], Arrow@Table[0.5 RM[α, ez].ex, {α, 0, Max[φ, 0.01], 0.05}],
    Tex["φ", 0.3 RM[φ/2, ez].ex]};
  circleSpace = {Opacity[0.25, Red],
    Cylinder[{{0, 0, -0.01}, {0, 0, 0.01}}, 0.75]};
  e3 = (RM[θ, e1].ez);
  θarc = {Arrowheads[0.01], Arrow@Table[0.5 RM[α, e1].ez, {α, 0, Max[θ, 0.01], 0.05}],
    Tex["θ", 0.3 RM[θ/2, e1].ez]};
  e2 = Cross[e3, e1];

  ψarc = {Arrowheads[0.01], Arrow@Table[0.5 RM[α, e3].e1, {α, 0, Max[ψ, 0.01], 0.05}],
    Tex["ψ", 0.3 RM[ψ/2, e3].e1]};
  circleBody = {Opacity[0.25, Blue], Cylinder[{{0, 0, -0.01}, {0, 0, 0.01}}, 0.75]};
  circleBody = GeometricTransformation[circleBody, RM[θ, e1]];
  lineOfNodes = {LightGray, Opacity[0.5],
    Tube[1.5 {-e1, e1}], {Black, Text[Stl["Line of nodes"], 1.6 e1]}};

  lab = Stl@StringForm["Euler Angles\n dφ/dt = φ̇ ê_2, dθ/dt = θ̇ ê_1, dψ/dt = ψ̇ ê_3"];

  Graphics3D[
    {axesxyz, axes123, φarc, θarc, ψarc, circleSpace, circleBody, lineOfNodes},
    PlotRange → 1.25 {{-1, 1}, {-1, 1}, {-1, 1}}, ViewPoint → {2, 2, 1},
    ImageSize → {400}, SphericalRegion → True, Boxed → False, PlotLabel → lab]

```

In[6]:=

```

Clear[ShowEllipsoid];
ShowEllipsoid[φ_, θ_, ψ_, a_, b_] :=
Module[{scale = 1.5, δ = 0.1, vp = {2, 2, 1}, sz = 0.01, szArrow = 0.03,
  0, ex, ey, ez, e1, e2, e3, axesxyz, bodyAxes, EM, range, latlongcurves,
  ellipsoid, ℛ, θArcFcn, φArc, θArc, ψArc, RM = RotationMatrix, lab, G},

range = scale {{-1, 1}, {-1, 1}, {-1, 1}};
{0, ex, ey, ez} = {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
(* lab frame *)
axesxyz =
{Black, Line[scale {-ex, ex}], Line[scale {-ey, ey}], Line[scale {-ez, ez}],
  Tex["x", (scale + δ) ex], Tex["y", (scale + δ) ey], Tex["z", (scale + δ) ez]};
e1 = (RM[φ, ez].ex);
φArc = {Red, Thick, Arrowheads[0.015], Arrow@
  Table[0.5 RM[α, ez].ex, {α, 0, Max[φ, 0.01], 0.05}], Tex["φ", 0.3 RM[ $\frac{\phi}{2}$ , ez].ex]};
e3 = (RM[θ, e1].ez);
θArc = {Red, Thick, Arrowheads[0.015], Arrow@
  Table[0.5 RM[α, e1].ez, {α, 0, Max[θ, 0.01], 0.05}], Tex["θ", 0.3 RM[ $\frac{\theta}{2}$ , e1].ez]};
e2 = Cross[e3, e1];
ψArc = {Red, Thick, Arrowheads[0.015], Arrow@
  Table[0.5 RM[α, e3].e1, {α, 0, Max[ψ, 0.01], 0.05}], Tex["ψ", 0.3 RM[ $\frac{\psi}{2}$ , e3].e1]};

latlongcurves =
{LightGray, Table[Line@Table[{a Sin[θφ] Cos[φφ], a Sin[θφ] Sin[φφ], b Cos[θφ]},
  {θφ, 0, π, π/100}], {φφ, 0, 2π, π/12}],
  Table[Line@Table[{a Sin[θφ] Cos[φφ], a Sin[θφ] Sin[φφ], b Cos[θφ]},
  {φφ, 0, 2π, π/100}], {θφ, -π, π, π/8}];
bodyAxes = {Blue, Tube[1.3 a {{-1, 0, 0}, {1, 0, 0}}],
  Tube[1.3 a {{0, -1, 0}, {0, 1, 0}}], Tube[1.3 b {{0, 0, -1}, {0, 0, 1}}],
  Tex["x1", 1.5 a ex], Tex["x2", 1.5 a ey], Tex["x3", 1.5 b ez]};
ellipsoid = {{Opacity[0.25, Blue], Ellipsoid[0, {a, a, b}]},
  latlongcurves, bodyAxes};
(* for convenience, construct object in lab frame and
  transform into body frame using the EulerMatrix *)
EM = EulerMatrix[{φ, θ, ψ}, {3, 1, 3}];
ellipsoid = GeometricTransformation[ellipsoid, EM];

lab = Module[{type},
  type = Which[a == b, "Spheroid",
    a < b, "Prolate ellipsoid",
    a > b, "Oblate ellipsoid"];
  Stl@StringForm["Body and space frames\n`", type]];
Graphics3D[{axesxyz, ellipsoid, φArc, θArc, ψArc},
  ImageSize → 500, Axes → False, Boxed → False, PlotRange → range,
  SphericalRegion → True, ViewPoint → vp, PlotLabel → lab]

```

Graphics Utilities

In[8]:=

```

(* Helper Functions*)
Clear[StoC, Tex, Vec, VecLab, Mess, DimMarker3];
StoC[r_,  $\theta$ _,  $\phi$ _] := {r Sin[ $\theta$ ] Cos[ $\phi$ ], r Sin[ $\theta$ ] Sin[ $\phi$ ], r Cos[ $\theta$  ]};
Tex[text_, position_] := Text[Style[text, Bold, FontSize  $\rightarrow$  10], position];
Vec[vec_] := {Arrowheads[0.05], Arrow[Tube[vec, 0.02]]};
Vec[vec_, size_, sizeAH_] := {Arrowheads[sizeAH], Arrow[Tube[vec, size]]};
(* Draw vector with label place beyond the tip *)
VecLab[st_, fn_, sz_, szArrow_, txt_, txtScale_] :=
Module[{vec, vecLabel},
  vec = Vec[{st, fn}, sz, szArrow];
  vecLabel = Text[Stl[txt], st + txtScale (fn - st)];
  {vec, vecLabel}]
Mess[lab_, Pobj_, offset_] :=
Module[{Ptex = Pobj + offset, gText, pointer, dirVec, arrow},
  dirVec = Pobj - Ptex;
  arrow = {Arrowheads[Small], Arrow[{Ptex + 0.2 dirVec, Ptex + 0.8 dirVec}]};
  gText = {Black, Style[Text[lab, Ptex], 10, Italic], arrow};
DimMarker3[{tail_, head_}, lab_, frac_, offset_] :=
Module[{labPosn},
  labPosn = tail + frac (head - tail);
  {Arrowheads[{-0.01, 0.01}], Arrow[{tail, head}], Text[lab, labPosn + offset]}]

```