Rigid body rotational motion model using quaternions
Translational motion not treated

The conventions in this Mathcad worksheet follow "Quaternions and Rotation Sequences,"
Jack B. Kuipers, Princeton (1999) ISBN 0-691-05872-5
Occasional page numbers and equation numbers from this reference are cited.
This book has its own home page at
http://www.calvin.edu/~kprs/book/quaternions.html
Please note errata in this book from the HTML file reference on that page.
Reference is also made to "Engineering Applications of Quaternions," bound 2001 release.

1.0 Elementary Quaternion Arithmetic

1.1 Conversions, Elementary Operations and Isomorphisms, Quaternion Multiplication

$$\mathrm{r2q}(r) := \begin{pmatrix} r \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \mathrm{q2v}(q) := \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \qquad \mathrm{v2q}(v) := \begin{pmatrix} 0 \\ v_0 \\ v_1 \\ v_2 \end{pmatrix}$$

Real and vector conversions

$$\mathrm{qconj}(q) := \begin{pmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{pmatrix}$$

Quaternion conjugate

$$\mathrm{q2M}(q) := \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}$$

Matrix isomorphism, EAOQ Eq. 14

$$\mathrm{qprod}(q1,q2) := \mathrm{r2q}\left(q1_0 \cdot q2_0 - \mathrm{q2v}(q1) \cdot \mathrm{q2v}(q2)\right) + \mathrm{v2q}\begin{pmatrix} q1_0 \cdot \mathrm{q2v}(q2) + q2_0 \cdot \mathrm{q2v}(q1) \ldots \\ + \mathrm{q2v}(q1) \times \mathrm{q2v}(q2) \end{pmatrix}$$

Product q1*q2
EAOQ Eq. 3

$$\mathrm{qprod}(q1,q2) := \begin{pmatrix} q1_0 \cdot q2_0 - q1_1 \cdot q2_1 - q1_2 \cdot q2_2 - q1_3 \cdot q2_3 \\ q1_0 \cdot q2_1 + q1_1 \cdot q2_0 + q1_2 \cdot q2_3 - q1_3 \cdot q2_2 \\ q1_0 \cdot q2_2 - q1_1 \cdot q2_3 + q1_2 \cdot q2_0 + q1_3 \cdot q2_1 \\ q1_0 \cdot q2_3 + q1_1 \cdot q2_2 - q1_2 \cdot q2_1 + q1_3 \cdot q2_0 \end{pmatrix}$$

Fast inline form (Kuipers Eq. 7.1 p. 156)

$$\mathrm{qvprod}(q,v) := -\mathrm{r2q}(\mathrm{q2v}(q) \cdot v) + \mathrm{v2q}\left(q_0 \cdot v + \mathrm{q2v}(q) \times v\right)$$

Product q*v

$$\mathrm{qvprod}(q,v) := \begin{pmatrix} -q_1 \cdot v_0 - q_2 \cdot v_1 - q_3 \cdot v_2 \\ q_0 \cdot v_0 + q_2 \cdot v_2 - q_3 \cdot v_1 \\ q_0 \cdot v_1 - q_1 \cdot v_2 + q_3 \cdot v_0 \\ q_0 \cdot v_2 + q_1 \cdot v_1 - q_2 \cdot v_0 \end{pmatrix}$$

Fast inline form

1.2 Skew symmetric and subspace operators, symmetric and skew symmetric matrix components

$$\text{Skewsy}(v) := \begin{pmatrix} 0 & -v_2 & v_1 \\ v_2 & 0 & -v_0 \\ -v_1 & v_0 & 0 \end{pmatrix}$$ EAOQ Eq. 27

$$\text{Subsp}(v) := \text{identity}(\text{length}(v)) - \frac{v \cdot v^T}{v \cdot v}$$ EAOQ Eq. 38

$$\text{Symm}(M) := .5 \cdot \left(M + M^T\right) \qquad \text{Antisymm}(M) := .5 \cdot \left(M - M^T\right)$$ Symmetrical & anti symmetrical parts of matrices

$$\text{Skewsy2v}(S) := \begin{pmatrix} S_{2,1} \\ S_{0,2} \\ S_{1,0} \end{pmatrix}$$ Vector from skew symmetrical matrix, from EAOQ Eq. 27

$$\text{Skewsy2v}(S) := \frac{1}{2}\begin{pmatrix} S_{2,1} - S_{1,2} \\ S_{0,2} - S_{2,0} \\ S_{1,0} - S_{0,1} \end{pmatrix}$$ Vector from skew symmetrical part of matrix

$$\text{qn}(q) := \left(q_0\right)^2 + \left(q_1\right)^2 + \left(q_2\right)^2 + \left(q_3\right)^2$$ Norm of q, EAOQ Eq. 7

1.3 Quaternion Rotation, Rotation Matrix, and Conversions

1.3.1 The Aerospace Sequence

What are we rotating and why?

Unless otherwise noted, we are looking at a point or vector from the perspective of a reference coordinate system and finding its coordinates in our reference frame. This is done beginning with its coordinates in a rotated coordinate frame such as a ship coordinate system, and rotating it with the coordinate system.

We have its components as stated in the rotated rotated frame, and rotate the vector to our reference frame. Thus, we are rotating the vector
FROM its coordinates in the rotated frame
TO its coordinates in the reference frame.

When rotating a body or its coordinate frame from a reference or inertial frame to a rotating frame, the sequence is yaw and heading ψ, then pitch and elevation γ, then roll ϕ. All angles are positive about their respective axes using the right hand rule.

Note that Kuipers uses θ for pitch and α for rotation angle -- we use γ for pitch and θ for overall rotation angle. See EAOQ Section 3.2.2 and Kuipers pp. 84 - 85.

1.3.2 Rotations

$$\mathrm{qprot}(q,v) := \frac{\mathrm{q2v}(\mathrm{qprod}(\mathrm{qvprod}(q,v),\mathrm{qconj}(q)))}{\mathrm{qn}(q)}$$

Quaternion vector rotation q*v*(1/q)
EAOQ Eq. 77, Kuipers Eq. 5.5 p. 117
See "First Perspective" at bottom
of page 123. We use 1/q instead of q*.

Rotation matrix from quaternion (EAOQ Eq. 98, Kuipers Eq. 5.9 p. 125)

$$\mathrm{Arot}(q) := \frac{1}{\mathrm{qn}(q)}\cdot\left[\left[(q_0)^2-(q_1)^2-(q_2)^2-(q_3)^2\right]\cdot\mathrm{identity}(3)\ \ldots \atop +2\cdot\left(q_0\cdot\mathrm{Skewsy}(\mathrm{q2v}(q))+\mathrm{q2v}(q)\cdot\mathrm{q2v}(q)^T\right)\right]$$

When

A = cos(θ/2) + sin(θ/2)*u, A = cos(θ)*I + sin(θ)*S_u + 2*sin(θ/2)^2*u*u^T
= cos(θ)*I + sin(θ)*S_u + (1 - cos(θ))*u*u^T
= cos(θ)*(I - u*u^T) + sin(θ)*S_u + u*u^T
= u*u^T + (cos(θ) + sin(θ)*S_u)*B_u

When examined as an operator on a vector v, the first term extracts the component of v along u and the second term extracts the component of v normal to u (with the subspace operator B_u). The second term rotates the component of v normal to u by an angle θ counterclockwise looking out u -- i.e. by the right hand rule, with the thumb pointing out along u and θ positive in the direction that the fingers curl.

Fast closed form (EAOQ Eq. 98)
Kuipers Eq. 5.11 p. 126 has a form that requires |q|=1

$$\mathrm{Arot}(q) := \frac{2}{\mathrm{qn}(q)}\cdot\begin{bmatrix}\frac{(q_0)^2+(q_1)^2-(q_2)^2-(q_3)^2}{2} & q_1\cdot q_2-q_0\cdot q_3 & q_1\cdot q_3+q_0\cdot q_2\\ q_1\cdot q_2+q_0\cdot q_3 & \frac{(q_0)^2-(q_1)^2+(q_2)^2-(q_3)^2}{2} & q_2\cdot q_3-q_0\cdot q_1\\ q_1\cdot q_3-q_0\cdot q_2 & q_2\cdot q_3+q_0\cdot q_1 & \frac{(q_0)^2-(q_1)^2-(q_2)^2+(q_3)^2}{2}\end{bmatrix}$$

Fast closed form quaternion point rotation

$$\mathrm{qprot}(q,v) := \frac{2}{\mathrm{qn}(q)}\cdot\begin{bmatrix}\frac{(q_0)^2+(q_1)^2-(q_2)^2-(q_3)^2}{2}\cdot v_0+(q_1\cdot q_2-q_0\cdot q_3)\cdot v_1+(q_1\cdot q_3+q_0\cdot q_2)\cdot v_2\\ (q_1\cdot q_2+q_0\cdot q_3)\cdot v_0+\frac{(q_0)^2-(q_1)^2+(q_2)^2-(q_3)^2}{2}\cdot v_1+(q_2\cdot q_3-q_0\cdot q_1)\cdot v_2\\ (q_1\cdot q_3-q_0\cdot q_2)\cdot v_0+(q_2\cdot q_3+q_0\cdot q_1)\cdot v_1+\frac{(q_0)^2-(q_1)^2-(q_2)^2+(q_3)^2}{2}\cdot v_2\end{bmatrix}$$

$$\mathrm{qprotx}(q,v) := (q_0)^2\cdot v+\left[2\cdot q_0\cdot(\mathrm{q2v}(q)\times v)\right]+\mathrm{q2v}(q)\times(\mathrm{q2v}(q)\times v)+(v\cdot\mathrm{q2v}(q))\cdot\mathrm{q2v}(q)$$

$$\mathrm{A2uaxis}(A) := \frac{\mathrm{Skewsy2v}(A)}{|\mathrm{Skewsy2v}(A)|}$$

Rotation axis from rotation matrix, EAOQ Eq. 113

$$\theta\mathrm{rotcore}(A) := \mathrm{atan2}(\mathrm{tr}(A) - 1, 2\,|\mathrm{Skewsy2v}(A)|)$$

Rotation angle from rotation matrix, EAOQ Eq. 114, 2cos(θ) from Kuipers, 3.4 p. 57

Quaternion from rotation matrix

$$\mathrm{A2qrotcore}(A) := \mathrm{r2q}\left(\cos\left(\frac{\theta\mathrm{rotcore}(A)}{2}\right)\right) + \sin\left(\frac{\theta\mathrm{rotcore}(A)}{2}\right)\cdot \mathrm{v2q}(\mathrm{A2uaxis}(A))$$

EAOQ Eq. 90

NOTE that $\theta=\pi$ case is handled by A2qrotx(A) and θrotx(A) below

$$\mathrm{qtestA}(A) := |\mathrm{Skewsy2v}(A)|$$

Test quantity to check for $\theta=\pi$

$$\mathrm{max\_A}(A) := \max\left(A_{0,0}, A_{1,1}, A_{2,2}\right)$$

Finds largest diagonal element of A

$$\mathrm{index\_A}(A) := \mathrm{if}\left(A_{0,0} = \max(A), 0, \mathrm{if}\left(A_{1,1} = \max(A), 1, 2\right)\right)$$

Finds index of largest diagonal element of A

$$\mathrm{upiA}(A) := \begin{pmatrix} A_{0,\mathrm{index\_A}(A)} \\ A_{1,\mathrm{index\_A}(A)} \\ A_{2,\mathrm{index\_A}(A)} \end{pmatrix}$$

Column with largest diagonal element

$$\mathrm{qpi}(A) := \mathrm{v2q}\left(\frac{\mathrm{upiA}(A)}{|\mathrm{upiA}(A)|}\right)$$

Quaternion if $\theta=\pi$ is unit vector along axis of rotation

$$\mathrm{deltaA} := 10^{-5}$$

Lowest value of θ in radians before special case is taken

$$\mathrm{A2qrot}(A) := \mathrm{if}(\mathrm{qtestA}(A) > \mathrm{deltaA}, \mathrm{A2qrotcore}(A), \mathrm{qpi}(A))$$

$$\theta\mathrm{rotx}(A) := \mathrm{if}\left(\mathrm{qtestA}(A) > \mathrm{deltaA}, \theta\mathrm{rotcore}(A), \pi\right)$$

1.4 Euler Angles, Quaternion from Euler angles

Euler angles are rotation angles about axes of the current coordinate frame, taken one at a time.

1.4.1 The Aerospace Sequence (zyx)

The Aerospace Euler angle sequence is rotation from a reference coordinate system to a rotated frame by rotating about the axes in the order z, y, then x. If the reference frame is a North-East-down Cartesian coordinate frame, this is rotation in azimuth and yaw, positive North to East, then elevation and pitch, positive upward, and last roll, positve right wing or starboard side down.

First, we will find the components of the rotation quaternion from the Euler angles.
Rotation is from North-East-Down to bow-starboard-keel

Roll positive right side down   Pitch positive bow up   Yaw positive bow to right

$$\mathrm{qroll}(\phi) := \begin{pmatrix} \cos\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \end{pmatrix} \quad \mathrm{qpitch}(\gamma) := \begin{pmatrix} \cos\left(\frac{\gamma}{2}\right) \\ 0 \\ \sin\left(\frac{\gamma}{2}\right) \\ 0 \end{pmatrix} \quad \mathrm{qyaw}(\psi) := \begin{pmatrix} \cos\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \end{pmatrix}$$

EAOQ Eq. 103

$\text{eu2qn}(\phi,\gamma,\psi) := \text{qprod}(\text{qroll}(\phi),\text{qprod}(\text{qpitch}(\gamma),\text{qyaw}(\psi)))$ Quaternion from Euler angles

$$\text{eu2q}(\phi,\gamma,\psi) := \begin{pmatrix} \cos\left(\frac{\phi}{2}\right)\cdot\cos\left(\frac{\gamma}{2}\right)\cdot\cos\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\cdot\sin\left(\frac{\gamma}{2}\right)\cdot\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cdot\cos\left(\frac{\gamma}{2}\right)\cdot\cos\left(\frac{\psi}{2}\right) + \cos\left(\frac{\phi}{2}\right)\cdot\sin\left(\frac{\gamma}{2}\right)\cdot\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cdot\sin\left(\frac{\gamma}{2}\right)\cdot\cos\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\cdot\cos\left(\frac{\gamma}{2}\right)\cdot\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cdot\sin\left(\frac{\gamma}{2}\right)\cdot\cos\left(\frac{\psi}{2}\right) + \cos\left(\frac{\phi}{2}\right)\cdot\cos\left(\frac{\gamma}{2}\right)\cdot\sin\left(\frac{\psi}{2}\right) \end{pmatrix}$$

Closed form,
Point rotation
EAOQ Ex. 104

Rotation matrix NOTE: EAOQ Eq. 96, Kuipers Eq. 4.4 p. 86 or Eq. 7.17 p. 167 use DIFFERENT SIGNS for the Euler angles

Three dimensional matrices

Coordinate systems: N-E-D to bow, starboard (right), keel

$$\text{Aroll}(\phi) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix}$$

Roll, positive starboard down

$$\text{Apitch}(\gamma) := \begin{pmatrix} \cos(\gamma) & 0 & \sin(\gamma) \\ 0 & 1 & 0 \\ -\sin(\gamma) & 0 & \cos(\gamma) \end{pmatrix}$$

Pitch, positive bow up

$$\text{Ayaw}(\psi) := \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Yaw, positive bow to starboard

$$\text{eaA}(\phi,\gamma,\psi) := \begin{pmatrix} \cos(\gamma)\cdot\cos(\psi) & -\cos(\gamma)\cdot\sin(\psi) & \sin(\gamma) \\ \sin(\phi)\cdot\sin(\gamma)\cdot\cos(\psi) + \cos(\phi)\cdot\sin(\psi) & -\sin(\phi)\cdot\sin(\gamma)\cdot\sin(\psi) + \cos(\phi)\cdot\cos(\psi) & -\sin(\phi)\cdot\cos(\gamma) \\ -\cos(\phi)\cdot\sin(\gamma)\cdot\cos(\psi) + \sin(\phi)\cdot\sin(\psi) & \cos(\phi)\cdot\sin(\gamma)\cdot\sin(\psi) + \sin(\phi)\cdot\cos(\psi) & \cos(\phi)\cdot\cos(\gamma) \end{pmatrix}$$

1.4.2 The Orbital Element Sequence (zxz)

The orbital element Euler angle sequence is rotation from a reference coordinate system to a rotated frame by rotating about the axes in the order z, x, then again z. If the reference frame is an Earth centered inertial right handed Cartesian coordinate frame with the z axis through the North pole and the x axis toward the vernal equinox (Aries), this is rotation in latitude, positive East to the line of nodes (the latitude of the ascending node, or the point above which the satellite passes through the equatorial plane Northbound), then inclination of the orbital plane, positive Eastward half plane upward, and last true anomaly or angle from that point to the new x axis positve Northward. These euler angles are denoted by Ω, i, and ν, respectively.

References for orbital element geometries:
Fundamentals of Astrodynamics, by Bate, Mueller and White, Dover, 1971, ISBN 0-486-60061-0 pp. 58-59..
Fundamentals of Astrodynamics and Applications, by David A. Vallado, McGraw-Hill, 1997, ISBN 0-07-066834-5 pp. 130-131. Also available in hardcover ISBN 0-07-066829-9.

This orbital angle sequence can also be used to represnt body rotation. It is less natural for this purpose because incremental roll, pitch and yaw are complex in this Euler angle sequence but it is equivalent algebraically.

First, we will find the components of the rotation quaternion from the Euler angles.

Rotation from equatorial plane to orbital position — Rotation to inclination of orbital plane — Rotation to line of nodes toward ascending node

$$q\_tan(\nu) := \begin{pmatrix} \cos\left(\frac{\nu}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\nu}{2}\right) \end{pmatrix} \qquad q\_incl(i) := \begin{pmatrix} \cos\left(\frac{i}{2}\right) \\ \sin\left(\frac{i}{2}\right) \\ 0 \\ 0 \end{pmatrix} \qquad q\_lan(\Omega) := \begin{pmatrix} \cos\left(\frac{\Omega}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\Omega}{2}\right) \end{pmatrix}$$

$$eu2qn(\nu, i, \Omega) := qprod(q\_tan(\nu), qprod(q\_incl(i), q\_lan(\Omega)))$$ Quaternion from Euler angles

$$euzxz2qpc(\nu, i, \Omega) := \begin{pmatrix} \cos\left(\frac{\nu}{2}\right)\cdot\cos\left(\frac{i}{2}\right)\cdot\cos\left(\frac{\Omega}{2}\right) - \sin\left(\frac{\nu}{2}\right)\cdot\cos\left(\frac{i}{2}\right)\cdot\sin\left(\frac{\Omega}{2}\right) \\ \cos\left(\frac{\nu}{2}\right)\cdot\sin\left(\frac{i}{2}\right)\cdot\cos\left(\frac{\Omega}{2}\right) + \sin\left(\frac{\nu}{2}\right)\cdot\sin\left(\frac{i}{2}\right)\cdot\sin\left(\frac{\Omega}{2}\right) \\ \sin\left(\frac{\nu}{2}\right)\cdot\sin\left(\frac{i}{2}\right)\cdot\cos\left(\frac{\Omega}{2}\right) - \cos\left(\frac{\nu}{2}\right)\cdot\sin\left(\frac{i}{2}\right)\cdot\sin\left(\frac{\Omega}{2}\right) \\ \sin\left(\frac{\nu}{2}\right)\cdot\cos\left(\frac{i}{2}\right)\cdot\cos\left(\frac{\Omega}{2}\right) + \cos\left(\frac{\nu}{2}\right)\cdot\cos\left(\frac{i}{2}\right)\cdot\sin\left(\frac{\Omega}{2}\right) \end{pmatrix}$$

Rotation matrix

Three dimensional matrices — Coordinate systems: N-E-D to bow, starboard (right), keel

$$A\_lan(\Omega) := \begin{pmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ Rotation to line of nodes toward ascending node

$$A\_incl(i) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & -\sin(i) \\ 0 & \sin(i) & \cos(i) \end{pmatrix}$$ Rotation to inclination of orbital plane

$$A\_tan(\nu) := \begin{pmatrix} \cos(\nu) & -\sin(\nu) & 0 \\ \sin(\nu) & \cos(\nu) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ Rotation from equatorial plane to orbital position

$$Az(\nu, i, \Omega) := \begin{pmatrix} \cos(\nu)\cdot\cos(\Omega) - \sin(\nu)\cdot\cos(i)\cdot\sin(\Omega) & -\cos(\nu)\cdot\sin(\Omega) - \sin(\nu)\cdot\cos(i)\cdot\cos(\Omega) & \sin(\nu)\cdot\sin(i) \\ \sin(\nu)\cdot\cos(\Omega) + \cos(\nu)\cdot\cos(i)\cdot\sin(\Omega) & -\sin(\nu)\cdot\sin(\Omega) + \cos(\nu)\cdot\cos(i)\,\cos(\Omega) & -\cos(\nu)\,\sin(i) \\ \sin(i)\cdot\sin(\Omega) & \sin(i)\cdot\cos(\Omega) & \cos(i) \end{pmatrix}$$

1.5 Euler Angles from the Quaternion and Direction Cosines

$$q2\phi(q) := \operatorname{atan2}\left[\left(q_0\right)^2 - \left(q_1\right)^2 - \left(q_2\right)^2 + \left(q_3\right)^2, 2 \cdot \left(q_0 \cdot q_1 - q_2 \cdot q_3\right)\right]$$ Roll

$$q2\gamma(q) := \operatorname{asin}\left[2 \cdot \left(q_1 \cdot q_3 + q_0 \cdot q_2\right)\right]$$ Pitch

EAOQ Eq. 106, Eq. 109, Eq. 111; Kuipers, Sec. 7.8, p. 168, errors corrected

$$q2\psi(q) := \operatorname{atan2}\left[\left(q_0\right)^2 + \left(q_1\right)^2 - \left(q_2\right)^2 - \left(q_3\right)^2, 2 \cdot \left(q_0 \cdot q_3 - q_1 \cdot q_2\right)\right]$$ Yaw

$$A2\phi(A) := \operatorname{atan2}\left(A_{2,2}, -A_{1,2}\right)$$

$$A2\gamma(A) := \operatorname{asin}\left(A_{0,2}\right)$$

$$A2\psi(A) := \operatorname{atan2}\left(A_{0,0}, -A_{0,1}\right)$$

$$q2\Omega(q) := \operatorname{atan2}\left(q_2 \cdot q_3 + q_0 \cdot q_1, q_1 \cdot q_3 - q_0 \cdot q_2\right)$$ Argument of ascending node

$$q2i(q) := \operatorname{acos}\left[\left(q_0\right)^2 - \left(q_1\right)^2 - \left(q_2\right)^2 + \left(q_3\right)^2\right]$$ Incliniation of orbital plane

$$q2\nu(q) := \operatorname{atan2}\left(q_0 \cdot q_1 - q_2 \cdot q_3, q_1 \cdot q_3 + q_0 \cdot q_2\right)$$ True anomaly

$$A2\Omega(A) := \operatorname{atan2}\left(A_{2,1}, A_{2,0}\right)$$

$$A2i(A) := \operatorname{acos}\left(A_{2,2}\right)$$

$$A2\nu(A) := \operatorname{atan2}\left[(-A)_{1,2}, A_{0,2}\right]$$

$$\operatorname{atan2}(1, 0.01754) = 1.004865\,\text{deg}$$

1.6 Translating Between Aerospace and Orbital Euler Angles

$$as2\nu(\phi,\gamma) := \operatorname{atan2}(\sin(\phi) \cdot \cos(\gamma), \sin(\gamma))$$

$$as2i(\phi,\gamma) := \operatorname{acos}(\cos(\phi) \cdot \cos(\gamma))$$

$$as2\Omega(\phi,\gamma,\psi) := \operatorname{atan2}(\cos(\phi) \cdot \sin(\gamma) \cdot \sin(\psi) + \sin(\phi) \cdot \cos(\psi), -\cos(\phi) \cdot \sin(\gamma) \cdot \cos(\psi) + \sin(\phi) \cdot \sin(\psi))$$

$$zxz2\phi(\nu, i) := \operatorname{atan2}(\cos(i), \cos(\nu) \cdot \sin(i))$$

$$zxz2\gamma(\nu, i) := \operatorname{asin}(\sin(\nu) \cdot \sin(i))$$

$$zxz2\psi(\nu, i, \Omega) := \operatorname{atan2}(\cos(\nu) \cdot \cos(\Omega) - \sin(\nu) \cdot \cos(i) \cdot \sin(\Omega), \cos(\nu) \cdot \sin(\Omega) + \sin(\nu) \cdot \cos(i) \cdot \cos(\Omega))$$

$$\mathrm{Jacoe}(\phi,\gamma) := \begin{pmatrix} \dfrac{-\cos(\phi)\cdot\cos(\gamma)\cdot\sin(\gamma)}{1-\cos(\phi)^2\cdot\cos(\gamma)^2} & \dfrac{\sin(\phi)}{1-\cos(\phi)^2\cdot\cos(\gamma)^2} & 0 \\ \dfrac{\sin(\phi)\cdot\cos(\gamma)}{\sqrt{1-\cos(\phi)^2\cdot\cos(\gamma)^2}} & \dfrac{\cos(\phi)\cdot\sin(\gamma)}{\sqrt{1-\cos(\phi)^2\cdot\cos(\gamma)^2}} & 0 \\ \dfrac{\sin(\gamma)}{1-\cos(\phi)^2\cdot\cos(\gamma)^2} & \dfrac{-\cos(\phi)\cdot\sin(\phi)\cdot\cos(\gamma)}{1-\cos(\phi)^2\cdot\cos(\gamma)^2} & 1 \end{pmatrix}$$

Jacobian @[ν,i,Ω]/@[φ,γ,ψ]
Determinant is -cos(γ)/sin(i)

$$\mathrm{Jaceo}(\nu,i) := \begin{pmatrix} \dfrac{-\sin(\nu)\cdot\cos(i)\cdot\sin(i)}{1-\sin(\nu)^2\cdot\sin(i)^2} & \dfrac{\cos(\nu)}{1-\sin(\nu)^2\cdot\sin(i)^2} & 0 \\ \dfrac{\cos(\nu)\cdot\sin(i)}{\sqrt{1-\sin(\nu)^2\cdot\sin(i)^2}} & \dfrac{\sin(\nu)\cdot\cos(i)}{\sqrt{1-\sin(\nu)^2\cdot\sin(i)^2}} & 0 \\ \dfrac{\cos(i)}{1-\sin(\nu)^2\cdot\sin(i)^2} & \dfrac{-\cos(\nu)\cdot\sin(\nu)\cdot\sin(i)}{1-\sin(\nu)^2\cdot\sin(i)^2} & 1 \end{pmatrix}$$

Jacobian @[φ,γ,ψ]/@[ν,i,Ω]
Determinant is -sin(i)/cos(γ)

1.7 NUMERICAL EXAMPLES ====================================

$\mathrm{phi} := 33\cdot\mathrm{deg}$ $\quad \mathrm{gam} := -10\cdot\mathrm{deg}$ $\quad \mathrm{psi} := 42\cdot\mathrm{deg}$ Euler angles for examples

$\Omega_{lan} := 21\cdot\mathrm{deg}$ $\quad i_{incl} := 22\cdot\mathrm{deg}$ $\quad \nu_{tanom} := -240\cdot\mathrm{deg}$

$\mathrm{vx} := 1 \quad \mathrm{vy} := 1 \quad \mathrm{vz} := 1$ Vector components for examples

$$v := \begin{pmatrix} \mathrm{vx} \\ \mathrm{vy} \\ \mathrm{vz} \end{pmatrix}$$ Vector to be rotated

$\mathrm{qu} := \mathrm{eu2q}(\mathrm{phi},\mathrm{gam},\mathrm{psi})$ Rotation quaternion from closed form

$\mathrm{Arq} := \mathrm{Arot}(\mathrm{qu})$ Direction cosine rotation matrix from quaternion

$\mathrm{Ar} := \mathrm{eaA}(\mathrm{phi},\mathrm{gam},\mathrm{psi})$ Direction cosine rotation matrix from Euler angles

$\mathrm{qr} := \mathrm{A2qrot}(\mathrm{Ar})$ Quaternion from direction cosines

Factors of the direction cosine matrix

$$\mathrm{Aroll}(\mathrm{phi}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.838671 & -0.544639 \\ 0 & 0.544639 & 0.838671 \end{pmatrix}$$

$$\mathrm{Apitch}(\mathrm{gam}) = \begin{pmatrix} 0.984808 & 0 & -0.173648 \\ 0 & 1 & 0 \\ 0.173648 & 0 & 0.984808 \end{pmatrix}$$

$$\text{Ayaw(psi)} = \begin{pmatrix} 0.743145 & -0.669131 & 0 \\ 0.669131 & 0.743145 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{qroll(phi)} = \begin{pmatrix} 0.95882 \\ 0.284015 \\ 0 \\ 0 \end{pmatrix} \quad \text{qpitch(gam)} = \begin{pmatrix} 0.996195 \\ 0 \\ -0.087156 \\ 0 \end{pmatrix} \quad \text{qyaw(psi)} = \begin{pmatrix} 0.93358 \\ 0 \\ 0 \\ 0.358368 \end{pmatrix}$$

Factors of the quaternion

$$\text{qu} = \begin{pmatrix} 0.9006 \\ 0.234195 \\ -0.179411 \\ 0.319193 \end{pmatrix} \quad \text{qr} = \begin{pmatrix} 0.9006 \\ 0.234195 \\ -0.179411 \\ 0.319193 \end{pmatrix}$$

Quaternion

$$\text{qprod(qprod(qroll(phi), qpitch(gam)), qyaw(psi))} = \begin{pmatrix} 0.9006 \\ 0.234195 \\ -0.179411 \\ 0.319193 \end{pmatrix}$$

Direction cosine rotation matrix

$$\text{Ar} = \begin{pmatrix} 0.731855 & -0.658965 & -0.173648 \\ 0.490897 & 0.686537 & -0.536365 \\ 0.472662 & 0.307298 & 0.825929 \end{pmatrix}$$

$$\text{Arq} = \begin{pmatrix} 0.731855 & -0.658965 & -0.173648 \\ 0.490897 & 0.686537 & -0.536365 \\ 0.472662 & 0.307298 & 0.825929 \end{pmatrix}$$

$$\text{Arot(qr)} = \begin{pmatrix} 0.731855 & -0.658965 & -0.173648 \\ 0.490897 & 0.686537 & -0.536365 \\ 0.472662 & 0.307298 & 0.825929 \end{pmatrix}$$

$$\text{Aroll(phi)} \cdot \text{Apitch(gam)} \cdot \text{Ayaw(psi)} = \begin{pmatrix} 0.731855 & -0.658965 & -0.173648 \\ 0.490897 & 0.686537 & -0.536365 \\ 0.472662 & 0.307298 & 0.825929 \end{pmatrix}$$

$$\text{Ar} \cdot \text{v} = \begin{pmatrix} -0.100758 \\ 0.641069 \\ 1.605889 \end{pmatrix} \quad \text{qprot(qu, v)} = \begin{pmatrix} -0.100758 \\ 0.641069 \\ 1.605889 \end{pmatrix} \quad \text{qprotx(qu, v)} = \begin{pmatrix} -0.100758 \\ 0.641069 \\ 1.605889 \end{pmatrix}$$

Rotated vector

$\text{q2}\phi(\text{qr}) = 33\,\text{deg}$ $\quad \text{A2}\phi(\text{Ar}) = 33\,\text{deg}$

$\text{q2}\gamma(\text{qr}) = -10\,\text{deg}$ $\quad \text{A2}\gamma(\text{Ar}) = -10\,\text{deg}$

Euler angles from quaternion, direction cosines

$\text{q2}\psi(\text{qr}) = 42\,\text{deg}$ $\quad \text{A2}\psi(\text{Ar}) = 42\,\text{deg}$

Orbital Euler angle sequence

$qzxz := euzxz2qpc(\nu_{tanom}, i_{incl}, \Omega_{lan})$ $Azxz := Az(\nu_{tanom}, i_{incl}, \Omega_{lan})$

$$qzxz = \begin{pmatrix} -0.327674 \\ -0.123921 \\ -0.145092 \\ -0.925323 \end{pmatrix} \quad Azxz = \begin{pmatrix} -0.754547 & -0.570448 & 0.324419 \\ 0.642368 & -0.743156 & 0.187303 \\ 0.134247 & 0.349725 & 0.927184 \end{pmatrix} \quad Azxz \cdot v = \begin{pmatrix} -1.000576 \\ 0.086515 \\ 1.411156 \end{pmatrix}$$

$q2\nu(qzxz) = 120\,deg$ $q2i(qzxz) = 22\,deg$ $q2\Omega(qzxz) = 21\,deg$ Checks

$A2\nu(Azxz) = 120\,deg$ $A2i(Azxz) = 22\,deg$ $A2\Omega(Azxz) = 21\,deg$

$\phi_{zxz} := zxz2\phi(\nu_{tanom}, i_{incl})$ $\gamma_{zxz} := zxz2\gamma(\nu_{tanom}, i_{incl})$ $\psi_{zxz} := zxz2\psi(\nu_{tanom}, i_{incl}, \Omega_{lan})$

$\phi_{zxz} = -11.420796\,deg$ $\gamma_{zxz} = 18.930368\,deg$ $\psi_{zxz} = 142.910205\,deg$

$as2\nu(\phi_{zxz}, \gamma_{zxz}) = 120\,deg$ $as2i(\phi_{zxz}, \gamma_{zxz}) = 22\,deg$ $as2\Omega(\phi_{zxz}, \gamma_{zxz}, \psi_{zxz}) = 21\,deg$

$q2\phi(qzxz) = -11.420796\,deg$ $q2\gamma(qzxz) = 18.930368\,deg$ $q2\psi(qzxz) = 142.910205\,deg$

$A2\phi(Azxz) = -11.420796\,deg$ $A2\gamma(Azxz) = 18.930368\,deg$ $A2\psi(Azxz) = 142.910205\,deg$

$Aczxz := eaA(\phi_{zxz}, \gamma_{zxz}, \psi_{zxz})$ $qczxz := eu2q(\phi_{zxz}, \gamma_{zxz}, \psi_{zxz})$

$$qczxz = \begin{pmatrix} 0.327674 \\ 0.123921 \\ 0.145092 \\ 0.925323 \end{pmatrix} \quad Aczxz = \begin{pmatrix} -0.754547 & -0.570448 & 0.324419 \\ 0.642368 & -0.743156 & 0.187303 \\ 0.134247 & 0.349725 & 0.927184 \end{pmatrix} \quad Aczxz \cdot v = \begin{pmatrix} -1.000576 \\ 0.086515 \\ 1.411156 \end{pmatrix}$$

2.0 Dynamics of Rotating Bodies

2.1 Elementary Dynamics

2.1.1 Rotation of Vectors with the Quaternion

The rotation of a vector rb given in a rotated frame to a vector in the reference frame is

rr = q*rb*conj(q)

where q is a quaternion of norm 1. The time derivative of both sides is

rr_dot = qdot*rb*conj(q) + q*rb*conj(qdot)

where rb_dot is zero because the body is rigid and rr_dot is the relative velocity vr in the reference frame. We now rotate vr to the body frame with

vb = conj(q)*vr*q

This gives us

vb = conj(q)*qdot*rb + rb*conj(qdot)*q

We know from the time derivative of

conj(q)*q = 1

conj(qdot)*q + conj(q)*qdot = 0

that

conj(qdot)*q = -conj(q)*qdot

We also see that the two terms in the time derivative are the quaternion conjugate of each other. This means that the quantity conj(q)*qdot is a pure vector. So, we have

vb = conj(q)*qdot*rb - rb*conj(q)*qdot

We know from the fundamental definition of the cross product (EAOQ Eq. 23)

(1/2)*(v1*v2 - v2*v1) = v1 X v2

so that

vb = 2*[conj(q)*qdot] X rb

This means that the relative velocity between two points in a rotating coordinate system is given by a cross product between the vector 2*[conj(q)*qdot] and the vector between the two points.

2.1.2 Rotation of Vectors with the Direction Cosine Matrix

The rotation of a vector rb from the rotating frame to the reference frame is

rr = A*rb

The time derivative, which gives us the relative velocity of the two points on the rigid body where the vector rb is the vector from one to the other in the body frame, is

rr_dot = Adot*rb

where rb_dot is zero because the body is rigid and rr_dot is the relative velocity vr in the reference frame. We now rotate vr to the body frame with

vb = A^T*vr

This gives us

vb = A^T*Adot*rb

We can show that A^T*Adot is a skew symmetric matrix from the time derivative of

A^T*A = I

Adot^T*A + A^T*Adot = 0

Since these terms are the negative of each other and the transpose of each other, each term must be skew symmetric. Also, we know from classical dynamics that the relative velocity between the two points due to rotational motion is

vb = ωb X rb

This means that the skew-symmetric form for ωb and the time derivative of the direction cosine matrix are related by

S_ωb = A^T*Adot = -Adot^T*A

2.1.3 Differential Equation for the Quaternion

Setting the two equations for vb from equal to each other gives us

qdot = (1/2)*q*ωb

or, taking the quaternion conjugate of both sides,

conj(qdot) = -(1/2)*ωb*conj(q)

2.2 Euler's Equations for Rotating Rigid Bodies

Euler's equations are the differential equations of motion of rotating rigid bodies. They are derived from the principles of elementary rotational dynamics -- angular momentum is constant unless torque is applied to the body. The angular momentum vector h is the product of the moment of inertia matrix M and the rotational rate vector ω. The time derivative of the angular momentum vector h is the applied torque.

We dervie Euler's equations twice. The first time is the conventional method using the direction cosine matrix. The second time we use quaternions.

2.2.1 Euler's Equations from Direction Cosines

Angular momentum is constant. In the reference coordinate system,

hr = Mr*ωr

The moment of inertia matrix Mr of a rotating body is not constant in an inertial frame. In body coordinates, the moment of inertia matrix Mb is constant. If we freeze (arrest the rotation) of body coordinates so that we can simply view vectors of motion in that coordinate frame, we can see that the rotated angular momentum vector hb is, in terms of the rotated ωr vector ωb,

hb = Mb*ωb

The angular momentum vector in the body frame, rotated back to the inertial frame, is

hr = A^T*hb = A^T*Mb*ωb

Here we have an equation in which both the angular momentum vector and the moment of inertia matrix are constant. Taking the time derivative of both sides gives us

tr = 0 = (Adot)^T*Mb*ωb + A^T*Mb*ωb_dot

where we have used dh/dt = torque, which we have stated as zero. This gives us

Mb*ωb_dot = tb - A* (Adot)^T*Mb*ωb

From classical dynamics as expressed above, we have

Mb*ωb_dot = tb - S_ωb*Mb*ωb

This is Euler's equation for motion of a rotating rigid body. The equation for integration is ωb_dot = Mb^(-1)*(tb - S_ωb*Mb*ωb)

2.2.2 Euler's Equations from Quaternions

Angular momentum is constant. In the reference coordinate system,

hr = Mr*ωr

The moment of inertia matrix Mr of a rotating body is not constant in an inertial frame. In body coordinates, the moment of inertia matrix Mb is constant. If we freeze (arrest the rotation) of body coordinates so that we can simply view vectors of motion in that coordinate frame, we can see that the rotated angular momentum vector hb is, in terms of the rotated ωr vector ωb,

hb = Mb*ωb

The angular momentum vector in the body frame, rotated back to the inertial frame, is

hr = q*hb*qconj = q*[Mb*ωb]*qconj

Here we have an equation in which both the angular momentum vector and the moment of inertia matrix are constant. Taking the time derivative of both sides gives us

tr = 0 = qdot*[Mb*ωb]*qconj + q*[Mb*ωb]*qconj_dot + q*[Mb*ωb_dot]*qconj

where we have used dh/dt = torque, which we have stated as zero. This gives us

Mb*ωb_dot = tb - qdot*qconj*[Mb*ωb] - [Mb*ωb]*qconj_dot*q

From classical dynamics as expressed above, we have

Mb*ωb_dot = tb - S_ωb*Mb*ωb

This is Euler's equation for motion of a rotating rigid body. The equation for integration is

ωb_dot = Mb^(-1)*(tb - S_ωb*Mb*ωb)

2.3 Euler Angle Rates, the Rotation Rate Vector, and Quaternion Time Derivatives

2.3.1 Aerospace Sequence

Derivatives of components of the rotation quaternion with respect to the Euler angles

$$\mathrm{qdroll}(\phi) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \end{pmatrix} \qquad \mathrm{qdpitch}(\gamma) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{\gamma}{2}\right) \\ 0 \\ \cos\left(\frac{\gamma}{2}\right) \\ 0 \end{pmatrix} \qquad \mathrm{qdyaw}(\psi) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \cos\left(\frac{\psi}{2}\right) \end{pmatrix}$$

Quaternion derivative with respect to time from Euler angles & their derivatives with respect to time

$$\begin{aligned} \mathrm{eu2qd}(\phi,\gamma,\psi,\phi d,\gamma d,\psi d) := {} & \mathrm{qprod}(\mathrm{qdroll}(\phi),\mathrm{qprod}(\mathrm{qpitch}(\gamma),\mathrm{qyaw}(\psi)))\cdot\phi d\ \ldots \\ & + \mathrm{qprod}(\mathrm{qroll}(\phi),\mathrm{qprod}(\mathrm{qdpitch}(\gamma),\mathrm{qyaw}(\psi)))\cdot\gamma d\ \ldots \\ & + \mathrm{qprod}(\mathrm{qroll}(\phi),\mathrm{qprod}(\mathrm{qpitch}(\gamma),\mathrm{qdyaw}(\psi)))\cdot\psi d \end{aligned}$$

Two ways to get the angular rate vector ω from the quaternion

$$qqd2\omega(q,qd) := q2v\left(\frac{2\cdot qprod(qconj(q),qd)}{qn(q)}\right)$$

Angular rate vector ω from quaternion and its time derivative (Kuipers Eq. 11.9 p. 263)

$$ed2\omega(\phi,\gamma,\psi) := \begin{pmatrix} \cos(\gamma)\cdot\cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi)\cdot\cos(\gamma) & \cos(\psi) & 0 \\ \sin(\gamma) & 0 & 1 \end{pmatrix}$$

Convert from Euler angle rates vector to angular rate vector ω (a closed form) Determinant is cos(γ); singularity at γ=π/2

$$\omega 2ed(\phi,\gamma,\psi) := \frac{1}{\cos(\gamma)}\begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \cos(\gamma)\cdot\sin(\psi) & \cos(\gamma)\cdot\cos(\psi) & 0 \\ -\sin(\gamma)\cdot\cos(\psi) & \sin(\gamma)\cdot\sin(\psi) & \cos(\gamma) \end{pmatrix}$$

Euler angle rates from the rotation rate vector Determinant is 1/cos(γ)

2.3.2 Orbital Sequence zxz

Derivatives of components of the rotation quaternion with respect to the Euler angles

$$qdtan(\nu) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{\nu}{2}\right) \\ 0 \\ 0 \\ \cos\left(\frac{\nu}{2}\right) \end{pmatrix} \qquad qdincl(i) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{i}{2}\right) \\ \cos\left(\frac{i}{2}\right) \\ 0 \\ 0 \end{pmatrix} \qquad qdlan(\Omega) := \frac{1}{2}\begin{pmatrix} -\sin\left(\frac{\Omega}{2}\right) \\ 0 \\ 0 \\ \cos\left(\frac{\Omega}{2}\right) \end{pmatrix}$$

Quaternion derivative with respect to time from Euler angles & their derivatives with respect to time

$$\begin{aligned} euzxz2qd(\nu,i,\Omega,\nu d,id,\Omega d) := {} & qprod(qdtan(\nu),qprod(q\_incl(i),q\_lan(\Omega)))\cdot\nu d \ldots \\ & + qprod(q\_tan(\nu),qprod(qdincl(i),q\_lan(\Omega)))\cdot id \ldots \\ & + qprod(q\_tan(\nu),qprod(q\_incl(i),qdlan(\Omega)))\cdot\Omega d \end{aligned}$$

Two ways to get the angular rate vector ω from the quaternion

$$qqd2\omega(q,qd) := q2v\left(\frac{2\cdot qprod(qconj(q),qd)}{qn(q)}\right)$$

Angular rate vector ω from quaternion and its time derivative (Kuipers Eq. 11.9 p. 263)

$$ezxz2\omega(\nu,i,\Omega) := \begin{pmatrix} \sin(i)\cdot\sin(\Omega) & \cos(\Omega) & 0 \\ \sin(i)\cdot\cos(\Omega) & -\sin(\Omega) & 0 \\ \cos(i) & 0 & 1 \end{pmatrix}$$

Convert from Euler angle rates vector to angular rate vector ω (a closed form) Determinant is -sin(i); singularity at i=0

$$\omega 2ezxz(\nu,i,\Omega) := \frac{1}{\sin(i)}\cdot\begin{pmatrix} \sin(\Omega) & \cos(\Omega) & 0 \\ \sin(i)\cdot\cos(\Omega) & -\sin(i)\cdot\sin(\Omega) & 0 \\ -\sin(\Omega)\cdot\cos(i) & -\cos(\Omega)\cdot\cos(i) & \sin(i) \end{pmatrix}$$

Euler angle rates from the rotation rate vector Determinant is -1/sin(i)

2.4 Numerical Examples of Rotation Rate Conversions

$\mathrm{phi} = 33\,\mathrm{deg}$ $\quad \mathrm{gam} = -10\,\mathrm{deg}$ $\quad \mathrm{psi} = 42\,\mathrm{deg}$ Angles (from previous examples)

$\mathrm{phid} := -7 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ $\quad \mathrm{gamd} := 4 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ $\quad \mathrm{psid} := 3 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ Angle rates

$\Omega_{lan} = 21\,\mathrm{deg}$ $\quad i_{incl} = 22\,\mathrm{deg}$ $\quad \nu_{tanom} = -240\,\mathrm{deg}$ From previous examples

$\Omega d_{lan} := 4 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ $\quad id_{incl} := 7 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ $\quad \nu d_{tanom} := -3 \cdot \frac{\mathrm{deg}}{\mathrm{sec}}$ Angle rates

$$\mathrm{eurv} := \begin{pmatrix} \mathrm{phid} \\ \mathrm{gamd} \\ \mathrm{psid} \end{pmatrix}$$ Euler angle rate vector

$\omega\mathrm{bv} := \mathrm{ed2}\omega(\mathrm{phi}, \mathrm{gam}, \mathrm{psi}) \cdot \mathrm{eurv}$ Matrix mapping Euler angle rates to ωb

$$\omega\mathrm{bv} = \begin{pmatrix} -2.446461 \\ 7.585334 \\ 4.215537 \end{pmatrix} \frac{\mathrm{deg}}{\mathrm{sec}} \qquad \omega\mathrm{2ed}(\mathrm{phi}, \mathrm{gam}, \mathrm{psi}) \cdot \omega\mathrm{bv} = \begin{pmatrix} -7 \\ 4 \\ 3 \end{pmatrix} \frac{\mathrm{deg}}{\mathrm{sec}}$$

$\mathrm{qdex} := \mathrm{eu2qd}(\mathrm{phi}, \mathrm{gam}, \mathrm{psi}, \mathrm{phid}, \mathrm{gamd}, \mathrm{psid})$ Quaternion equation for ωb in terms of the quaternion and its time derivative

$\mathrm{qex} := \mathrm{eu2qn}(\mathrm{phi}, \mathrm{gam}, \mathrm{psi})$

$$\mathrm{qex} = \begin{pmatrix} 0.790334 \\ -0.086887 \\ 6.838161 \times 10^{-3} \\ 0.606445 \end{pmatrix}$$

$$\mathrm{qdex} = \begin{pmatrix} 0.294134 \\ -2.690392 \\ 2.531601 \\ 2.567018 \end{pmatrix} \frac{\mathrm{deg}}{\mathrm{sec}}$$

$$\mathrm{qqd2}\omega(\mathrm{qex}, \mathrm{qdex}) = \begin{pmatrix} -1.16606 \\ 6.814668 \\ 4.103984 \end{pmatrix} \frac{\mathrm{deg}}{\mathrm{sec}}$$ Numerical check with brute force numerical computations

$$\omega\mathrm{2ed}(\mathrm{phi}, \mathrm{gam}, \mathrm{psi}) \cdot \mathrm{qqd2}\omega(\mathrm{qex}, \mathrm{qdex}) = \begin{pmatrix} -5.510166 \\ 4.284039 \\ 3.147154 \end{pmatrix} \frac{\mathrm{deg}}{\mathrm{sec}}$$

$$\mathrm{qdzxz} := \mathrm{euzxz2qd}\left(\nu_{tanom}, i_{incl}, \Omega_{lan}, \nu d_{tanom}, id_{incl}, \Omega d_{lan}\right) \qquad \mathrm{qzxz} = \begin{pmatrix} -0.327674 \\ -0.123921 \\ -0.145092 \\ -0.925323 \end{pmatrix}$$

$\omega zxz := qqd2\omega(qzxz, qdzxz)$

$$\omega 1zxz := ezxz2\omega\left(\nu_{tanom}, i_{incl}, \Omega_{lan}\right)\cdot\begin{pmatrix}\nu d_{tanom}\\ id_{incl}\\ \Omega d_{lan}\end{pmatrix}$$

$$\omega zxz = \begin{pmatrix}6.132322\\ -3.557752\\ 1.218448\end{pmatrix}\frac{deg}{sec} \qquad \omega 1zxz = \begin{pmatrix}6.132322\\ -3.557752\\ 1.218448\end{pmatrix}\frac{deg}{sec}$$

$$\omega 2ezxz\left(\nu_{tanom}, i_{incl}, \Omega_{lan}\right)\cdot \omega zxz = \begin{pmatrix}-3\\ 7\\ 4\end{pmatrix}\frac{deg}{sec}$$

$\phi_{zxz} = -11.420796\,deg$ $\quad\gamma_{zxz} = 18.930368\,deg$ $\quad\psi_{zxz} = 142.910205\,deg$ From previous example

$$ed2\omega\left(\phi_{zxz}, \gamma_{zxz}, \psi_{zxz}\right)\cdot Jaceo\left(\nu_{tanom}, i_{incl}\right)\cdot\begin{pmatrix}\nu d_{tanom}\\ id_{incl}\\ \Omega d_{lan}\end{pmatrix} = \begin{pmatrix}6.132322\\ -3.557752\\ 1.218448\end{pmatrix}\frac{deg}{sec}$$

Chain rule ω vector

$$Jaceo\left(\nu_{tanom}, i_{incl}\right)\cdot Jacoe\left(\phi_{zxz}, \gamma_{zxz}\right) = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

Jacobians are inverse of each other

3.0 A Numerical Example of a Rotating Body Model

3.1 Specific parameters that model the body and the points on it

We will model a spinning hollow cone. The vertex is along the body X axis at distance .5*height from the orign.

Model Inputs

$height := 1\cdot m \qquad diam := .3\cdot m$ **Height and diameter of cone**

$thick := 0.05\cdot m$ **Half-thickness of cone**

Euler angles used here are aerospace sequence (zyx). See below for orbital element sequence (zxz) method.

$\phi_0 := 0.0 \qquad \gamma_0 := 0.05 \qquad \psi_0 := -0.05$ **Initial Euler angles**

$\nu_0 := 0.0 \qquad i_0 := 0.05 \qquad \Omega_0 := -0.05$ Alternative -- orbital element sequence

$\phi d_0 := 1.0 \qquad \gamma d_0 := -0.0 \qquad \psi d_0 := 0.0$ **Initial Euler angle rates**

$\nu d_0 := 1.0 \qquad id_0 := 0.05 \qquad \Omega d_0 := -.05$ Alternative -- orbital element rates

$\rho cone0 := 2500\cdot\frac{kg}{m^3}$ **Density of material of cone**

$\rho conea := \rho cone0\cdot thick\cdot 2$ **Mass per unit area**

$tanhc := \frac{diam}{2\cdot height}$ **Tangent of the half-cone angle**

$npm := 2$ **Number of point masses**

$npm := 2$

$sechalfcone := \sqrt{1 + tanhc^2}$

$sinhalfcone := \frac{tanhc}{sechalfcone}$

$mpm_0 := -\rho conea \cdot (.01 \cdot m)^2$ **Subtract 10 cm square fuze window**

$mpm_1 := 50 \cdot kg$ **Add 50 kg warhead**

$ppm_1 := \begin{pmatrix} \frac{height}{6} \\ \frac{height}{3} \cdot tanhc \\ 0 \cdot m \end{pmatrix}$ **Fuze window is 1/3 the height away from nose**

$ppm_2 := \begin{pmatrix} \frac{-height}{6} \\ 0 \cdot m \\ 0 \cdot m \end{pmatrix}$ **Warhead is 2/3 the height away from nose**

$height2 := .5 \cdot height$

$$mass_{cone} := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \, d\phi \, dx + if\left(npm > 0, \sum_{k=1}^{npm} mpm_{k-1}, 0 \cdot kg\right)$$ Mass

$$x_{cg} := \frac{1}{mass_{cone}} \cdot \left[\rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot (height2 - x) \, d\phi \, dx \ldots + if\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left(ppm_k\right)_0\right], 0 \cdot kg \cdot m\right]\right]$$ Center of gravity

$$y_{cg} := \frac{1}{mass_{cone}} \cdot \left[\rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot (x \cdot \cos(\phi)) \, d\phi \, dx \ldots + if\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left(ppm_k\right)_1\right], 0 \cdot kg \cdot m\right]\right]$$

$$z_{cg} := \frac{1}{mass_{cone}} \cdot \left[\rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot (x \cdot \sin(\phi)) \, d\phi \, dx \ldots + if\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left(ppm_k\right)_2\right], 0 \cdot kg \cdot m\right]\right]$$

$$mass_{cone} := \pi \cdot \rho conea \cdot tanhc \cdot sechalfcone \cdot height^2$$

Mass of the cone without point masses

The equations here assume thin skin hollow cone with no base. Add other masses or subtract missing mass to model irregularities such as fuze windows, mechanisms, warheads, etc. The moment of inertia matrix is given by (EAOQ Eq. 124)

$$M = \int\int\int \rho cone(x,y,z) \cdot \begin{bmatrix} (y-y_{cg})^2+(z-z_{cg})^2 & -(x-x_{cg})\cdot(y-y_{cg}) & -(x-x_{cg})\cdot(z-z_{cg}) \\ -(x-x_{cg})\cdot(y-y_{cg}) & (x-x_{cg})^2+(z-z_{cg})^2 & -(y-y_{cg})\cdot(z-z_{cg}) \\ -(x-x_{cg})\cdot(z-z_{cg}) & -(y-y_{cg})\cdot(z-z_{cg}) & (x-x_{cg})^2+(y-y_{cg})^2 \end{bmatrix} dz\,dy\,dx$$

where

$$mass_{cone} = \int\int\int \rho cone(x,y,z)\,dz\,dy\,dx \qquad \begin{pmatrix} x_{cg} \\ y_{cg} \\ z_{cg} \end{pmatrix} = \frac{1}{mass_{cone}} \int\int\int \rho cone(x,y,z) \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} dz\,dy\,dz$$

$$x_{cg} := \frac{-height}{6}$$

Center of gravity is 2/3 of height toward base

$$y_{cg} := 0$$

$$z_{cg} := 0$$

Moment of inertia matrix and its inverse

$$M0 := mass_{cone} \cdot \begin{bmatrix} \frac{height^2}{2}\cdot tanhc^2 & 0 & 0 \\ 0 & \frac{height^2}{2}\cdot\left(\frac{1}{9}+\frac{tanhc^2}{2}\right) & 0 \\ 0 & 0 & \frac{height^2}{2}\cdot\left(\frac{1}{9}+\frac{tanhc^2}{2}\right) \end{bmatrix}$$

No point masses

$mass_{cone} = 169.102711\ kg$ $\quad x_{cg} = -0.166716\ m$ $\quad y_{cg} = -7.391957 \times 10^{-6}\ m$

$z_{cg} = 0\ m$

$$Ix2 := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0\cdot m}^{height}\int_{-\pi}^{\pi} x \cdot \left(height2 - x - x_{cg}\right)^2 d\phi\,dx\ ...$$
$$+ if\left[npm > 0, \sum_{k=1}^{npm}\left[mpm_{k-1}\cdot\left[\left(ppm_k\right)_0 - x_{cg}\right]^2\right], 0\cdot kg\cdot m^2\right]$$

$$Iy2 := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot \left(x \cdot tanhc\cos(\phi) - y_{cg}\right)^2 d\phi\, dx \;...$$
$$+ \text{if}\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left[\left(ppm_k\right)_1 - y_{cg}\right]^2\right], 0 \cdot kg \cdot m^2\right]$$

$$Iz2 := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot \left(x \cdot tanhc\sin(\phi) - z_{cg}\right)^2 d\phi\, dx \;...$$
$$+ \text{if}\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left[\left(ppm_k\right)_2 - z_{cg}\right]^2\right], 0 \cdot kg \cdot m^2\right]$$

$$Ixy := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot \left(height2 - x - x_{cg}\right) \cdot \left(x \cdot tanhc \cdot \cos(\phi) - y_{cg}\right) d\phi\, dx \;...$$
$$+ \text{if}\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left[\left(ppm_k\right)_0 - x_{cg}\right] \cdot \left[\left(ppm_k\right)_1 - y_{cg}\right]\right], 0 \cdot kg \cdot m^2\right]$$

$$Ixz := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot \left(height2 - x - x_{cg}\right) \cdot \left(x \cdot tanhc \cdot \sin(\phi) - z_{cg}\right) d\phi\, dx \;...$$
$$+ \text{if}\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left[\left(ppm_k\right)_0 - x_{cg}\right] \cdot \left[\left(ppm_k\right)_2 - z_{cg}\right]\right], 0 \cdot kg \cdot m^2\right]$$

$$Iyz := \rho conea \cdot tanhc \cdot sechalfcone \cdot \int_{0 \cdot m}^{height} \int_{-\pi}^{\pi} x \cdot \left(x \cdot tanhc \cdot \cos(\phi) - y_{cg}\right) \cdot \left(x \cdot tanhc \cdot \sin(\phi) - z_{cg}\right) d\phi\, dx \;...$$
$$+ \text{if}\left[npm > 0, \sum_{k=1}^{npm} \left[mpm_{k-1} \cdot \left[\left(ppm_k\right)_1 - y_{cg}\right] \cdot \left[\left(ppm_k\right)_2 - z_{cg}\right]\right], 0 \cdot kg \cdot m^2\right]$$

$$M := \begin{pmatrix} Iy2 + Iz2 & -Ixy & -Ixz \\ -Ixy & Ix2 + Iz2 & -Iyz \\ -Ixz & -Iyz & Ix2 + Iy2 \end{pmatrix}$$

$$M_{inv} := M^{-1}$$

3.2 Numerical Evaluation of Moment of Inertia Matrix from Density Versus Position

The following treatment is a valid method, but is very slow, particularly in Mathcad or other interpretive general purpose languages because the method involves a lot of multiple nested numerical integrals of discontinuous functions. When possible, use closed forms determined by analysis as above. The equations below are disabled for computation because they would otherwise slow down the evaluation of the Mathcad spreadsheet.

Rationale for Distance from a Point to a Cone:
The equation for the surface of a cone can be written as

|(p - p0)Xu| = |p - p0|*sin(θ)

or

((p - p0),u) = |p - p0|*cos(θ)

where p0 is the vertex, θ is the half-cone angle, and u is a unit vector along the axis, postive from the vertex toward the base. The additional conditions

0 < ((p - p0),u) < height

apply to define the height of the cone. For an arbitrary point x, the geometry can be projected to two dimensions by noting that the point x is related to a ray of the cone by the X and Y coordinates defined by

$$X = \left[\left(x - p_0\right), u\right] = \left|x - p_0\right| \cdot \cos(\theta x)$$

$$Y = \left|\left(x - p_0\right) \times u\right| = \left|x - p_0\right| \cdot \sin(\theta x)$$

where θx is the angle between the axis of the cone and the line from the vertex to the point x. We now have reduced the three dimensional distance problem to the two dimensional problem of finding the distance from a point to a line, the line being defined by the two points (0,0), the vertex, and (height, diam/2). That distance is given by the absolute value of the cross product of the vectors from the vertex to the point and the normalized vector between the two points on the line.

$$\mathrm{lxyz}(x,y,z) := \frac{2 \cdot |x|}{\mathrm{height}} + \frac{2 \cdot |y|}{\mathrm{diam}} + \frac{2 \cdot |z|}{\mathrm{diam}}$$ Near the cone?

$$\mathrm{Xdist}(x,y,z) := x - .5 \cdot \mathrm{height}$$ X coordinate in 2D projection

$$\mathrm{Ydist}(x,y,z) := \sqrt{y^2 + z^2}$$ Y coordinate in 2D projection

$$\mathrm{TwoDcp}(x,y,z) := .5 \cdot (x - .5 \cdot \mathrm{height}) \cdot \mathrm{diam} + \mathrm{height} \cdot \sqrt{y^2 + z^2}$$ Two dimensional cross product

$$\mathrm{constl} := \sqrt{\mathrm{height}^2 + .25 \cdot \mathrm{diam}^2}$$ height of ray of cone

$$\mathrm{dist}(x,y,z) := \mathrm{if}\left(\mathrm{lxyz}(x,y,z) \le 1, \frac{|\mathrm{TwoDcp}(x,y,z)|}{\mathrm{constl}}, 0\right)$$

$$\mathrm{dist}(x,y,z) := \mathrm{if}\left[\mathrm{lxyz}(x,y,z) \le 3, \frac{\left|.5 \cdot (x - .5 \cdot \mathrm{height}) \cdot \mathrm{diam} + \mathrm{height} \cdot \sqrt{y^2 + z^2}\right|}{\mathrm{constl}}, 100\right]$$

$$\rho\mathrm{cone}(x,y,z) := \mathrm{if}\left(\mathrm{dist}(x,y,z) \le \mathrm{thick}, \rho\mathrm{cone0}, 0\right)$$ Density function for integrals

$$\mathrm{xlim} := .5 \cdot \mathrm{height}$$ Limit to parallepiped containing cone

$$\mathrm{yzlim} := .5 \cdot \mathrm{diam}$$

$$\mathrm{mass}_{\mathrm{cone}} := \int_{-\mathrm{xlim}}^{\mathrm{xlim}} \int_{-\mathrm{yzlim}}^{\mathrm{yzlim}} \int_{-\mathrm{yzlim}}^{\mathrm{yzlim}} \rho\mathrm{cone}(x,y,z)\, dz\, dy\, dx$$ Mass of cone

$$\mathrm{mass}_{\mathrm{cone}} = 169.102711\ \mathrm{kg}$$

-

$$x_{cg} := \frac{1}{mass_{cone}}\left(\int_{-xlim}^{xlim}\int_{-yzlim}^{yzlim}\int_{-yzlim}^{yzlim} x\cdot \rho cone(x,y,z)\,dz\,dy\,dx\right)$$

Center of gravity

$$y_{cg} := \frac{1}{mass_{cone}}\left(\int_{-xlim}^{xlim}\int_{-yzlim}^{yzlim}\int_{-yzlim}^{yzlim} y\cdot \rho cone(x,y,z)\,dz\,dy\,dx\right)$$

$$z_{cg} := \frac{1}{mass_{cone}}\left(\int_{-xlim}^{xlim}\int_{-yzlim}^{yzlim}\int_{-yzlim}^{yzlim} z\cdot \rho cone(x,y,z)\,dz\,dy\,dx\right)$$

$$x_{cg} = -0.166716\ \text{m}$$

$$y_{cg} = -7.391957\times 10^{-6}\ \text{m}$$

$$z_{cg} = 0\ \text{m}$$

$$magI(x,y,z) := \begin{pmatrix} x^2+y^2+z^2 & 0 & 0 \\ 0 & x^2+y^2+z^2 & 0 \\ 0 & 0 & x^2+y^2+z^2 \end{pmatrix}$$

Magnitude squared times I

$$outer(x,y,z) := \begin{pmatrix} x\cdot x & x\cdot y & x\cdot z \\ x\cdot y & y\cdot y & y\cdot z \\ x\cdot z & y\cdot z & z\cdot z \end{pmatrix}$$

Outer product of vector

$$moifun(x,y,z) := magI\left(x-x_{cg}, y-y_{cg}, z-z_{cg}\right) - outer\left(x-x_{cg}, y-y_{cg}, z-z_{cg}\right)$$

Kernel of moment of inertia integral

$ii := 0..2 \qquad ij := 0..2$

$$Mbx_{ii,ij} := \int_{-xlim}^{xlim}\int_{-yzlim}^{yzlim}\int_{-yzlim}^{yzlim} \rho cone(x,y,z)\cdot moifun(x,y,z)_{ii,ij}\,dz\,dy\,dx$$

$$\frac{M}{mass_{cone}} = \begin{pmatrix} 7.924913\times 10^{-3} & 2.46435\times 10^{-6} & 0 \\ 2.46435\times 10^{-6} & 0.043083 & 0 \\ 0 & 0 & 0.043083 \end{pmatrix} \text{m}^2$$

3.3 Computed Model Parameters

Model Inputs -- Comment out unused option
Default option: Aerospace sequence Euler angles

$$\omega_0 := \text{ed2}\omega(\phi_0, \gamma_0, \psi_0) \cdot \begin{pmatrix} \phi d_0 \\ \gamma d_0 \\ \psi d_0 \end{pmatrix}$$ Initial rotation rate vector

$$q_0 := \text{eu2q}(\phi_0, \gamma_0, \psi_0)$$ Initial rotation quaternion

$$qd_0 := \text{eu2qd}(\phi_0, \gamma_0, \psi_0, \phi d_0, \gamma d_0, \psi d_0)$$ Quaternion time derivative

Second option: Orbital element Euler angles

$$\omega_0 := \text{ezxz2}\omega(\nu_0, i_0, \Omega_0) \cdot \begin{pmatrix} \nu d_0 \\ id_0 \\ \Omega d_0 \end{pmatrix}$$ Initial rotation rate vector

$$q_0 := \text{euzxz2qpc}(\nu_0, i_0, \Omega_0)$$ Initial rotation quaternion

$$qd_0 := \text{euzxz2qd}(\nu_0, i_0, \Omega_0, \nu d_0, id_0, \Omega d_0)$$ Quaternion time derivative

$$\text{qqd2}\omega(q_0, qd_0) = \begin{pmatrix} 0.997502 \\ 0.049917 \\ 0.049979 \end{pmatrix} \qquad \omega_0 = \begin{pmatrix} 0.997502 \\ 0.049917 \\ 0.049979 \end{pmatrix}$$ Numerical checks

$$\text{yinit} := \text{stack}(q_0, \omega_0)$$ Initial state vector

3.4 Set up equations of motion

The differential equation must be stated in the form dy/dt = f(t,y) where y is a state vector.
The state vector is the quaternion for the first four states and the rotation rate vector for the last three states.
We state these differential equations separately and augment them for the Mathcad numerical differential equation software.

$$\text{qdot}(q, \omega) := .5 \cdot \text{qvprod}(q, \omega)$$ Base quaternion differential equation (not used)

$$\text{torque} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$ Homogeneous form -- no torque applied

$$\omega\text{dot}(\text{torque}, \omega) := M_{inv} \cdot (\text{torque} - \text{Skewsy}(\omega) \cdot M \cdot \omega)$$ **Euler's equations**

$$\text{qstab}(q) := -2\text{qstabconstant} \cdot (|q| - 1)$$

Stabilization of quaternion amplitude to unit length
Constant qstabconstant set below

$$\omega\text{aug}(q, \omega) := \begin{pmatrix} \text{qstab}(q) \\ \omega_0 \\ \omega_1 \\ \omega_2 \end{pmatrix}$$

Augment real part of rotation quaternion to stabilize quaternion magnitude

$$\text{qdot}(q, \omega) := .5 \cdot \text{qprod}(q, \omega\text{aug}(q, \omega))$$ **Quaternion differential equation**

$$\mathrm{qdot}(q_0,\omega_0) = \begin{pmatrix} 3.124349\times10^{-4} \\ 0.499688 \\ 0.012495 \\ 0.012495 \end{pmatrix}$$

Initial quaternion time derivative

3.5 Functions to Extract the Quaternion and the Rotation Rate Vector from the State Vector

$$\mathrm{yq}(y) := \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \qquad \mathrm{y\omega}(y) := \begin{pmatrix} y_4 \\ y_5 \\ y_6 \end{pmatrix}$$

First 4 states are the quaternion,
next 3 states are the rotation rate vector.

3.6 Differential equation as stated for numerical integration

$$\mathrm{Derivx}(t,y) := \begin{pmatrix} \mathrm{qdot}(\mathrm{yq}(y),\mathrm{y\omega}(y)) \\ \mathrm{\omega dot}(\mathrm{torque},\mathrm{y\omega}(y)) \end{pmatrix}$$

Derivative of state vector f(t,y)
Format incompatible with Mathcad

$$\mathrm{Deriv}(t,y) := \begin{pmatrix} \mathrm{qdot}(\mathrm{yq}(y),\mathrm{y\omega}(y))_0 \\ \mathrm{qdot}(\mathrm{yq}(y),\mathrm{y\omega}(y))_1 \\ \mathrm{qdot}(\mathrm{yq}(y),\mathrm{y\omega}(y))_2 \\ \mathrm{qdot}(\mathrm{yq}(y),\mathrm{y\omega}(y))_3 \\ \mathrm{\omega dot}(\mathrm{torque},\mathrm{y\omega}(y))_0 \\ \mathrm{\omega dot}(\mathrm{torque},\mathrm{y\omega}(y))_1 \\ \mathrm{\omega dot}(\mathrm{torque},\mathrm{y\omega}(y))_2 \end{pmatrix}$$

Derivative of state vector f(t,y)
Format compatible with Mathcad

3.7 Jacobians

Mathcad numerical integration routines require the Jacobian matrix. This is a matrix whose first column is the partial derivative of the f(t,y) vector with respect to time, and whose next 7 columns are partial derivatives of f(t,y) with respect to the 7 elements of y. We give equations for the gradients separately, then build the Jacobian through augmentation using the gradient matrices as submatrices.

$$\mathrm{q\omega qgradq}(\omega,\mathrm{qs}) := \begin{pmatrix} \mathrm{qs} & -\omega_0 & -\omega_1 & -\omega_2 \\ \omega_0 & \mathrm{qs} & \omega_2 & -\omega_1 \\ \omega_1 & -\omega_2 & \mathrm{qs} & \omega_0 \\ \omega_2 & \omega_1 & -\omega_0 & \mathrm{qs} \end{pmatrix}$$

Gradient of q*ω with respect to q as a vector
Note that this is NOT the matrix isomorphism
for ω as a quaternion. "qs" is qstab(q).

$$\mathrm{q\omega sgradq}(q) := -2\mathrm{qstabconstant}\cdot q\cdot q^T$$

Second term is gradient of qstab term with respect to q as a vector

$$\mathrm{q\omega gradq}(q,\omega,\mathrm{qs}) := \mathrm{q\omega qgradq}(\omega,\mathrm{qs}) + \mathrm{q\omega sgradq}(q)$$

Total is sum (chain rule for differentiation)

$$q\omega grad\omega(q) := \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}$$

Gradient of q*ω with respect to ω as a vector
Note that this IS the last 3 columns of the matrix isomorphism for q

$$Jacobq(q,\omega) := .5\, augment\left[\begin{pmatrix}0\\0\\0\\0\end{pmatrix}, q\omega gradq(q,\omega,qstab(q)), q\omega grad\omega(q)\right]$$

4 X 8

$$Jacob\omega(\omega) := augment\left[\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix}, -M_{inv}\cdot(Skewsy(\omega)\cdot M - Skewsy(M\cdot\omega))\right]$$

3 X 8

$$Jacoby(t,y) := stack(Jacobq(yq(y), y\omega(y)), Jacob\omega(y\omega(y)))$$

7 X 8

3.8 Numerical Solution of Nonlinear Ordinary Differential Equation

Now we are ready to solve the differential equation using the Rosenbrock method for stiff differential equations. A differential equation is stiff if the Jacobian is nearly singular. Most methods of numerical integration are unstable for stiff equations.

Numerical simulation inputs

$t_0 := 0$ $t_{max} := 100$ **Time of solution**

$npoints := 250$ **Number of data points**

$SOLN := Stiffr(yinit, t_0, t_{max}, npoints, Deriv, Jacoby)$ **Solution of differential equation**

3.9 Extract Quaternion and Rotation Rate Vector from Output

$n := 0 .. npoints$ Range variable for data output

$qout(n) := submatrix(SOLN, n, n, 1, 4)^T$ Extract the quaternion from the output

$\omega out(n) := submatrix(SOLN, n, n, 5, 7)^T$ Extract the angular rate vector from the output

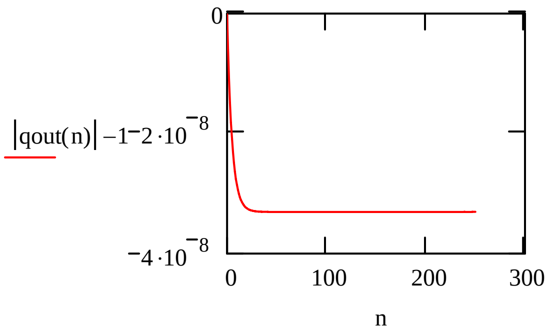
$tout(n) := submatrix(SOLN, n, n, 0, 0)$ Extract the effective time (epoch) of each output point

3.10 Monitor The Quaternion Damping for Keeping It to Unit Length

Set qstabconstant so that maximum peak quaternion error is minimized (try to keep below 10^(-6)).
The plot of the quaternion amplitude error is helpful but use the maximum peak error. Start with a value of about 0.5 (the stability limits are zero and two). Increase npoints if necessary.

Numerical simulation input

$qstabconstant \equiv .5$ **Set quaternion damping constant**



Quaternion amplitude damping coefficient

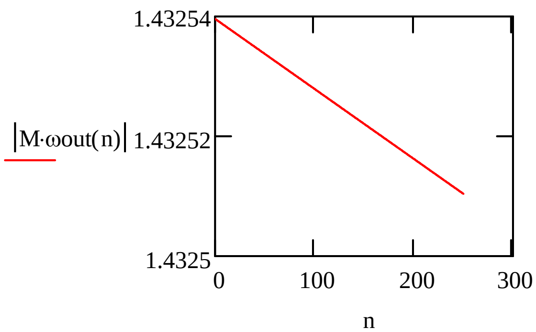
Monitor the maximum peak error

$vq_n := |qout(n)| - 1$

$\max(vq) = 0$

$\min(vq) = -3.310043 \times 10^{-8}$

$\max(vq, -vq) = 3.310043 \times 10^{-8}$



Angular momentum (should be nearly constant)

$$M \cdot \omega_0 = \begin{pmatrix} 1.336798 \\ 0.364085 \\ 0.364121 \end{pmatrix} m^2 \cdot kg$$

4.0 Watch a Point on the Rotating Body

4.1 Draw the cone

Assume the body is a cone with its axis in the X direction. We want to draw a line from the vertex to the base, then a circle around the base.

$Nbase := 6 \qquad Nheight := 4$

Number of points to plot the base and height

$ibs := 0 .. Nbase \qquad iln := 0 .. Nheight$

Range variables for drawing

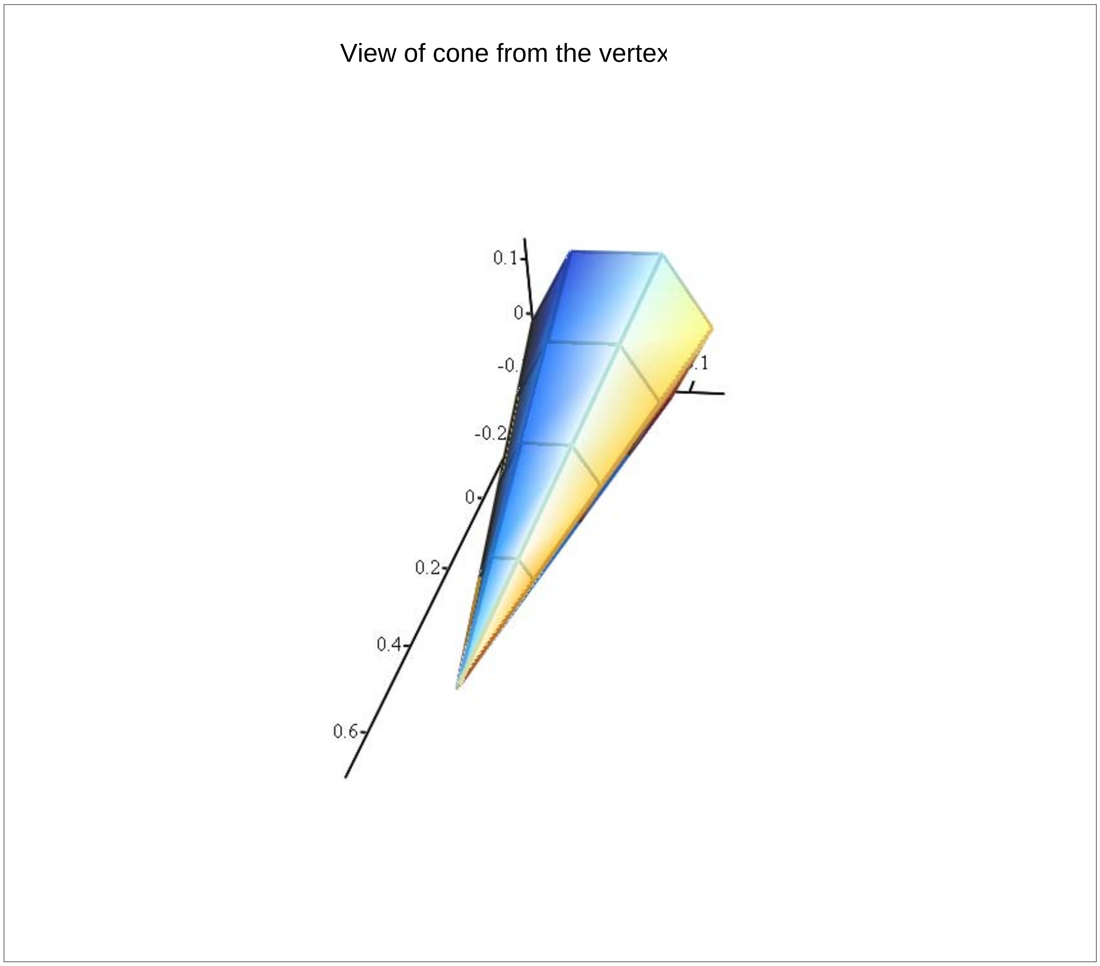
$$X_{ibs, iln} := \frac{height}{2} - \frac{height \cdot iln}{Nheight} - x_{cg}$$

Coordinates of points on the cone

$$Y_{ibs, iln} := \frac{iln \cdot diam}{2 \cdot Nheight} \cdot \cos\left(\frac{2 \cdot \pi \cdot ibs}{Nbase}\right) - y_{cg}$$

$$Z_{ibs, iln} := \frac{iln \cdot diam}{2 \cdot Nheight} \cdot \sin\left(\frac{2 \cdot \pi \cdot ibs}{Nbase}\right) - z_{cg}$$

View of cone from the vertex



(X, Y, Z)

4.2 Animate the rotation

$$pvect_{ibs,\,iln} := \begin{pmatrix} X_{ibs,\,iln} \\ Y_{ibs,\,iln} \\ Z_{ibs,\,iln} \end{pmatrix}$$

Store points in vectors for rotation

$$rpvect_{ibs,\,iln} := qprot\left(qout(FRAME),\, pvect_{ibs,\,iln}\right)$$

Rotate according to solution of differential equation

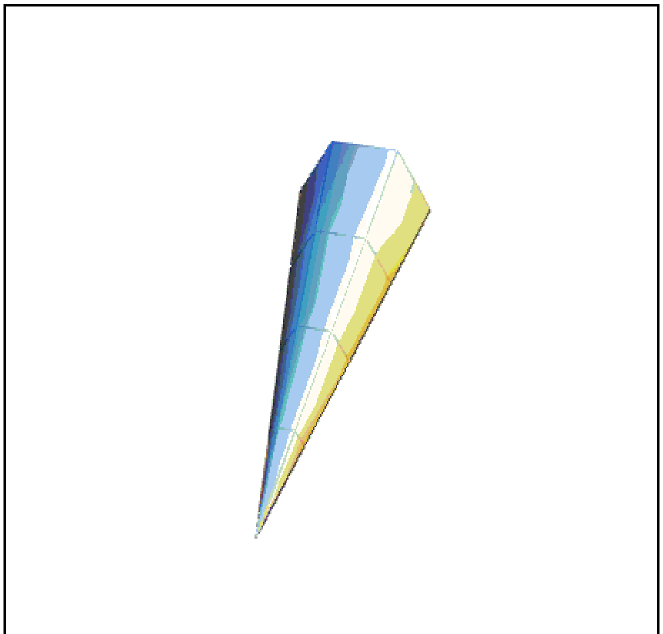
$$Xr_{ibs,\,iln} := \left(rpvect_{ibs,\,iln}\right)_0$$

Store rotated points back in array for plotting

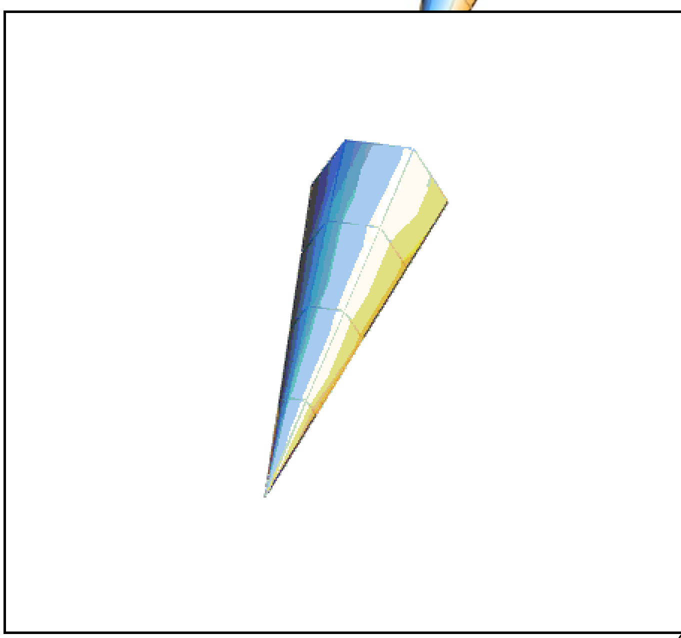
$$Yr_{ibs,\,iln} := \left(rpvect_{ibs,\,iln}\right)_1$$

$$Zr_{ibs,\,iln} := \left(rpvect_{ibs,\,iln}\right)_2$$

Plot for animation below



With_fuze&warhead.avi



Double-click the icons
to view animation

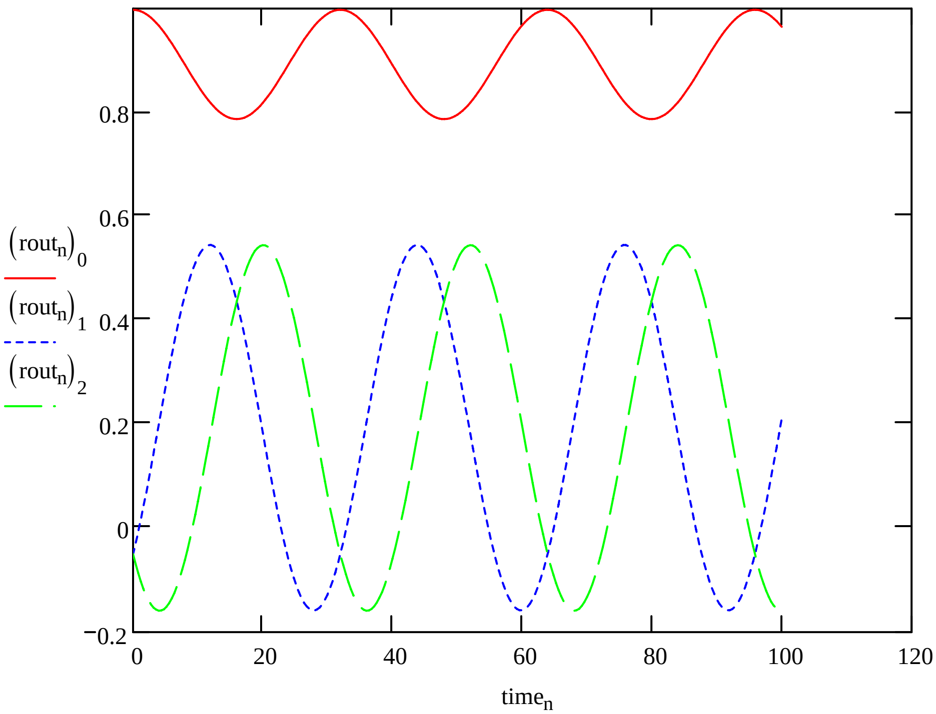
4.3 Show Plots of Rotated Positions

$rp1 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad rp2 := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad rp3 := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ Positions of points on the rotating body

$rout_n := qprot(qout(n), rp1)$

$time_n := SOLN_{n,0}$

Three components of position



$rout_0 = \begin{pmatrix} 0.997502 \\ -0.049979 \\ -0.049917 \end{pmatrix} \quad rout_1 = \begin{pmatrix} 0.997177 \\ -0.029286 \\ -0.069139 \end{pmatrix}$ Examples to identify starting points of plots

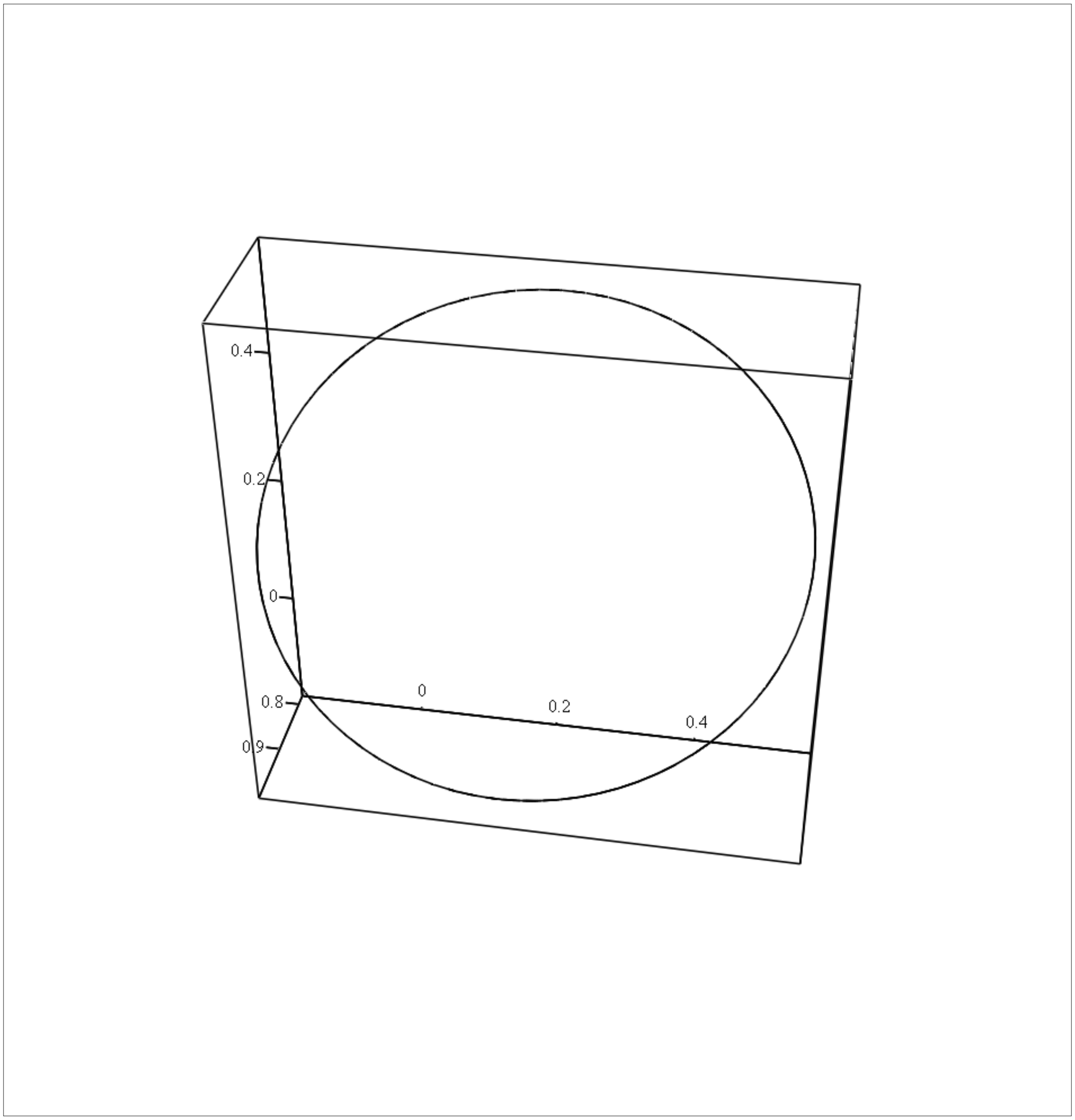
$rx_n := (rout_n)_0 \quad ry_n := (rout_n)_1 \quad rz_n := (rout_n)_2$ Vector components of output points

$sout_n := qprot(qout(n), rp2) \quad tout_n := qprot(qout(n), rp3)$ Other points for plotting

$sx_n := (sout_n)_0 \quad sy_n := (sout_n)_1 \quad sz_n := (sout_n)_2$ Vector components

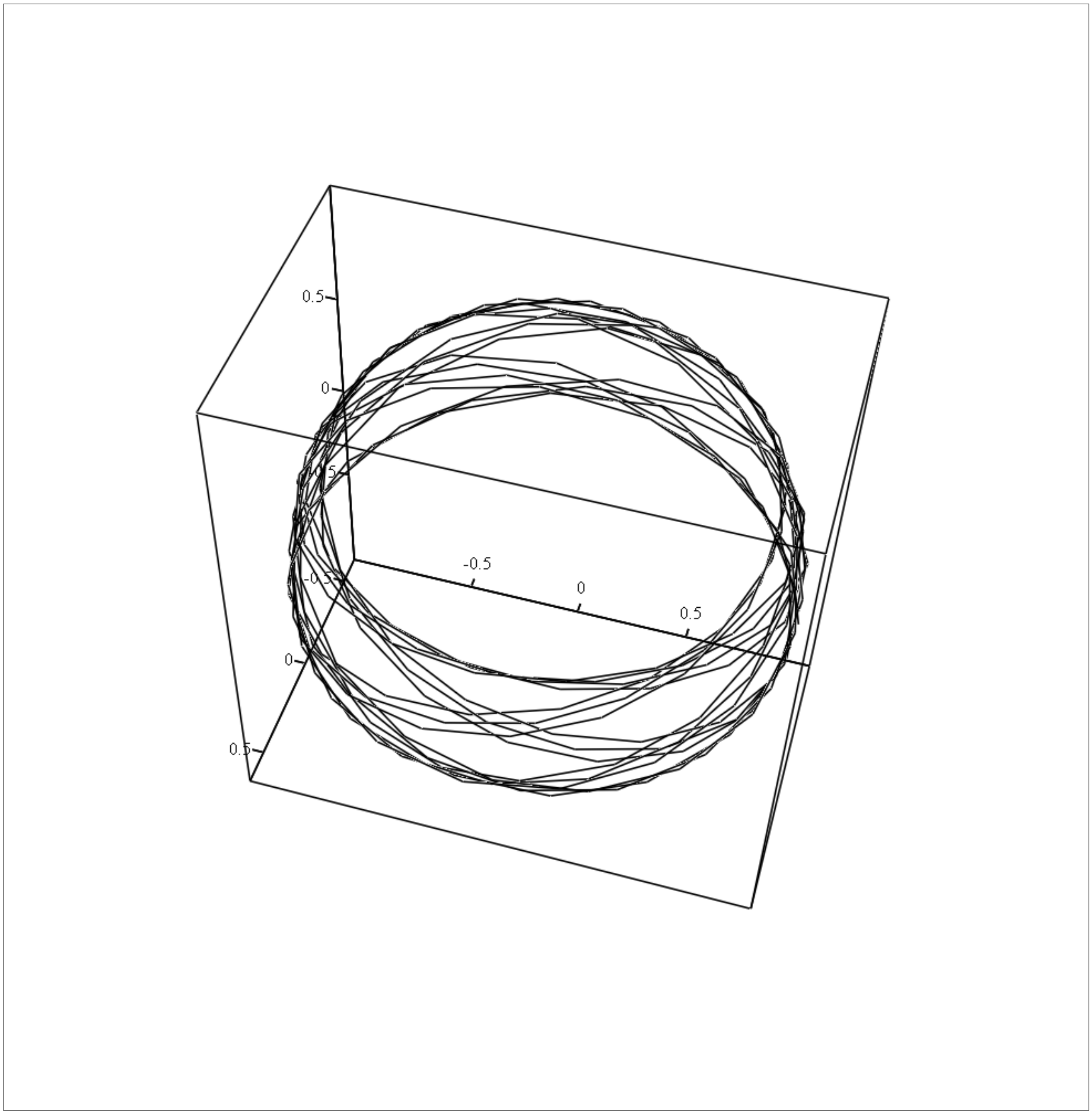
$tx_n := (tout_n)_0 \quad ty_n := (tout_n)_1 \quad tz_n := (tout_n)_2$

Point (1,0,0) versus time



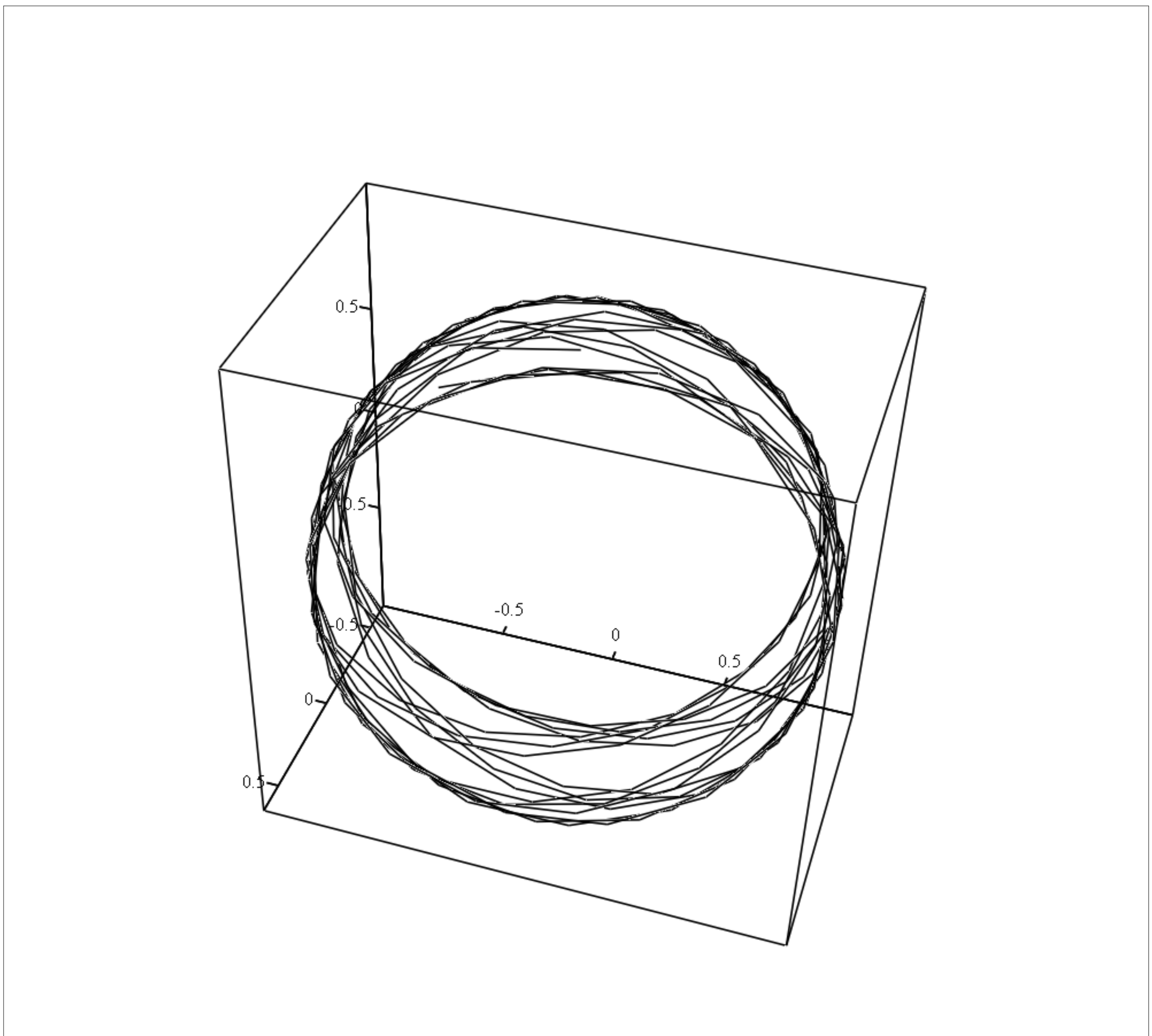
(rx, ry, rz)

Point (0,1,0) versus time



(sx, sy, sz)

Point (0,0,1) versus time



(tx, ty, tz)

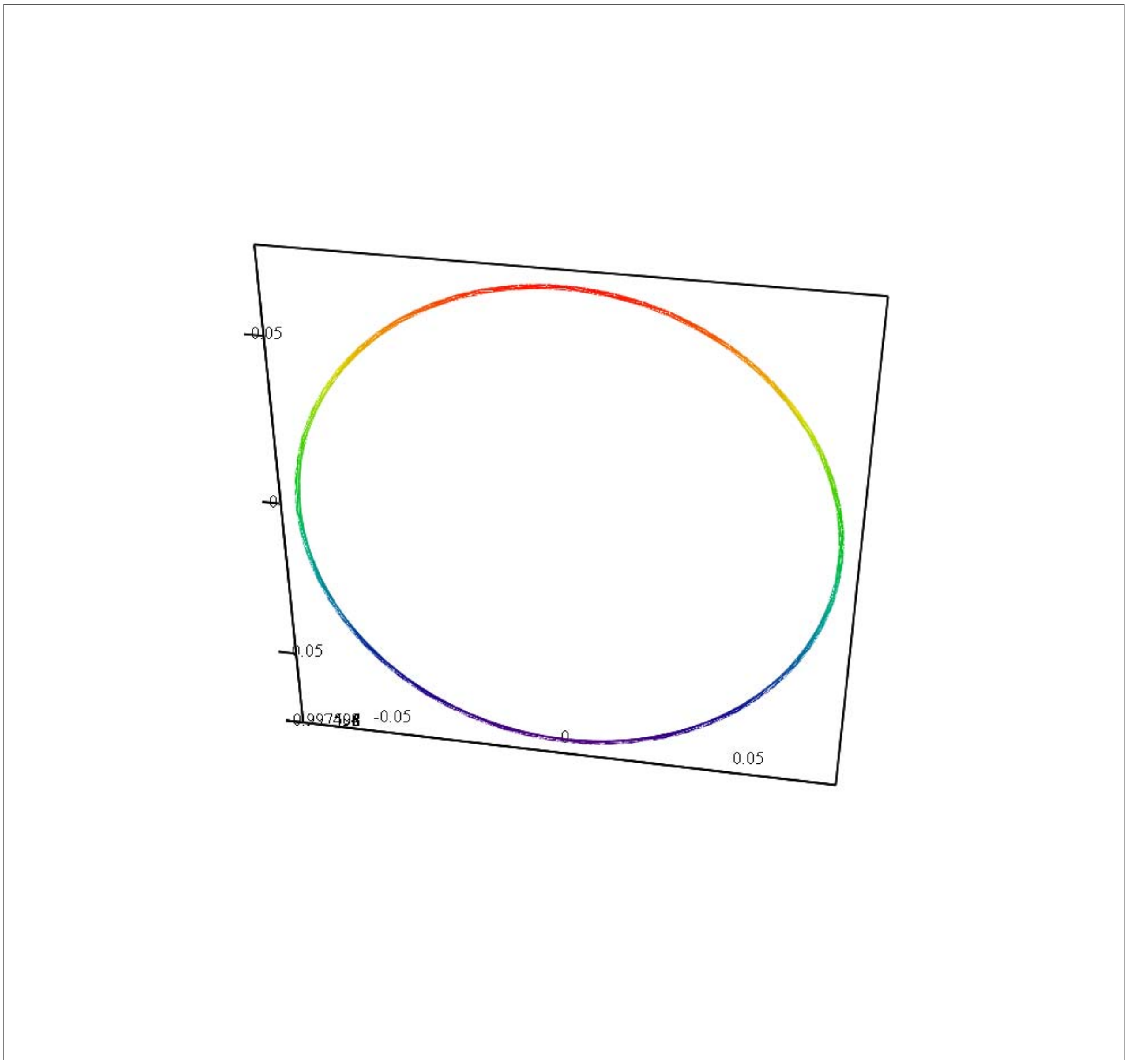
4.4 Watch Precession Through Rotation of the ω Vector

$\omega x_n := \omega out(n)_0$

$\omega y_n := \omega out(n)_1$ Components of rotation rate vector

$\omega z_n := \omega out(n)_2$

$\omega out(0) = \begin{pmatrix} 0.997502 \\ 0.049917 \\ 0.049979 \end{pmatrix}$ $\omega out(1) = \begin{pmatrix} 0.997503 \\ 0.063278 \\ 0.031363 \end{pmatrix}$ Initial points to identify starting point



$(\omega x, \omega y, \omega z)$

Components of the Rotation Rate Vector