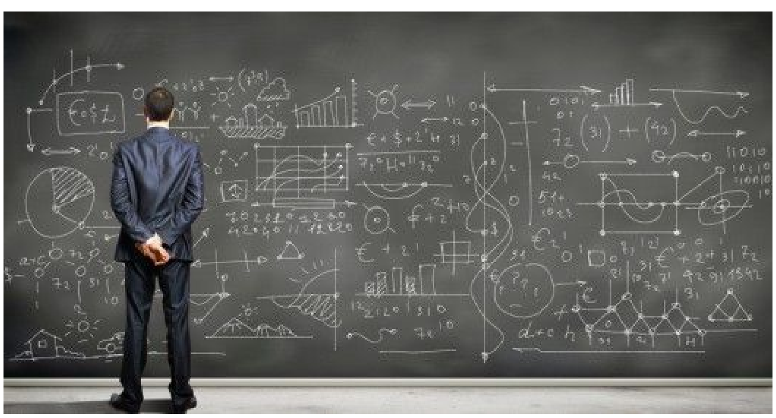
COMPUTATIONAL SCIENCES INTERNATIONAL

THE HYDRA TOOLKIT

RIGID BODY DYNAMICS THEORY MANUAL



CSI-2017-2, DECEMBER 2017
COMPUTATIONAL SCIENCES INTERNATIONAL

Copyright © 2017 Computational Sciences International

PUBLISHED BY COMPUTATIONAL SCIENCES INTERNATIONAL

All rights reserved. No claim to copyright is made for original U.S. Government Works.

First printing, December 2017

DOI: 10.13140/RG.2.2.27725.54248

Contents

List of Figures

I can calculate the motion of heavenly bodies,

but not the madness of people.

Isaac Newton.

1 *Introduction*

The rigid-body dynamics theory manual outlines the formulation for the six degree-of-freedom (DOF) rigid-body dynamics physics in the Hydra Toolkit. The rigid-body dynamics solver is intended to be used both as stand-alone solver, and optionally coupled to other physics through the multiphysics manager.

The formulation follows the development presented by Barraff [1997a,b]. In this document, the following notation convention is used to describe the rigid-body kinematics and solution methodology. All scalar quantities are presented as lower-case symbols, e.g., the mass is m. Vectors, second-rank tensors, and matrices are presented in **bold** and defined in the context they are used.

The theory manual first presents an overview of kinematics for rigid body dynamics in Chapter 2. The use of quaternions for rotations is presented in Chapter 3, followed by the time-integration algorithms in Chapter 4. Finally, the calculation of mass properties, i.e., the volume, mass, the body mass center, and inertia tensor calculations are presented in Chapter 5.

2 Kinematics and the Equations of Motion

Rigid bodies occupy space and may undergo both translation and rotation in a dynamic sense. In order to describe the translation of a rigid body in an inertial reference frame, we will use the location of the center-of-mass, $\mathbf{x}_{cm}(t)$ at time t. In order to describe the rotation, a rotation tensor, $\mathbf{R}(t)$, is used. Here, the components of the rotation tensor are given as

$$\mathbf{R}(t) = \begin{bmatrix} R_{xx} & R_{yx} & R_{zx} \\ R_{xy} & R_{yy} & R_{zy} \\ R_{xz} & R_{yz} & R_{zz} \end{bmatrix} \tag{2.1}$$

Together, $\mathbf{x}_{cm}(t)$ and $\mathbf{R}(t)$ are the spatial variables required to describe the position and orientation of a rigid body in our inertial reference frame.

In order to describe the shape of the rigid body, it is convenient to use a body-coordinate system that is attached to the center-of-mass. The body coordinate system is fixed to the rigid body and rotates and translates with the rigid body. The inertial and body coordinate systems are shown in Figure 2.1.

The initial location of the body is $\mathbf{x}_{cm}(0)$ and the rotation tensor is the identity, i.e., $\mathbf{R} = \mathbf{I}$. At time t, the rigid body has translated to $\mathbf{x}_{cm}(t)$ and experienced a finite rotation. The new body coordinates may be described in terms of the rotation tensor as

$$\begin{aligned} \mathbf{x}' &= \mathbf{R}(t)\,\mathbf{x} \\ \mathbf{y}' &= \mathbf{R}(t)\,\mathbf{y} \\ \mathbf{z}' &= \mathbf{R}(t)\,\mathbf{z} \end{aligned} \tag{2.2}$$

where

$$\mathbf{x} = \left\{ \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right\} \mathbf{y} = \left\{ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right\} \mathbf{z} = \left\{ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right\} \tag{2.3}$$

Performing the matrix operations in Eq. (2.2) will show that the columns of the rotation tensor, $\mathbf{R}(t)$, represent the orientation of the body coordinate axes at time t.

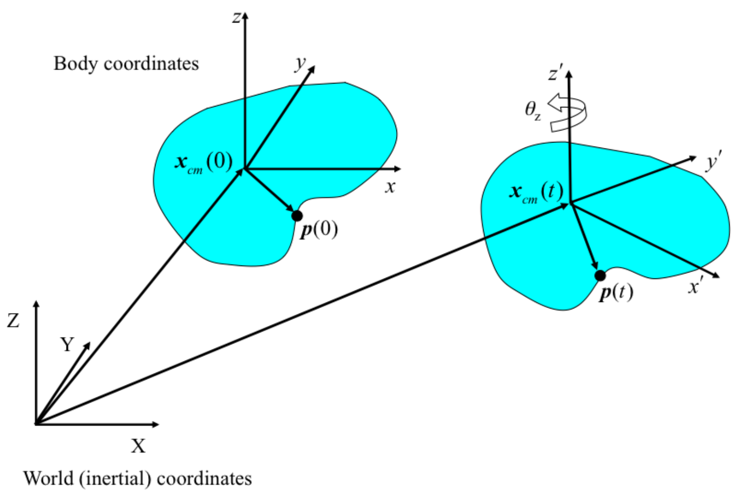


Figure 2.1: World and body coordinate systems.

The location of a point on the body, $\mathbf{p}$ may be calculated by rotating the point $\mathbf{p}(0)$ into the proper coordinate system at time t, and translating the point with the rigid body center-of-mass, i.e.,

$$\mathbf{p}(t) = \mathbf{R}(t)\mathbf{p}(0) + \mathbf{x}_{cm}(t) \tag{2.4}$$

Linear and Angular Velocity

The velocity of the center-of-mass is

$$\mathbf{v}_{cm} = \frac{d\mathbf{x}_{cm}}{dt} \tag{2.5}$$

For simplicity, we will refer to this as the linear velocity and drop the "cm" subscript.

The angular velocity, $\boldsymbol{\omega}$, describes the rate at which the rigid body spins about the center-of-mass. The direction of $\boldsymbol{\omega}$ defines the axis that the body is spinning about, and the magnitude, $\|\boldsymbol{\omega}\|$, indicates how fast the body is spinning.

The velocity of a given point on the rigid body is written in terms of the velocity of the center-of-mass and the angular velocity as

$$\begin{aligned} \mathbf{v}_p(t) &= \boldsymbol{\omega} \times \{\mathbf{p}(t) - \mathbf{x}_{cm}(t)\} + \mathbf{v}(t) \\ &= \boldsymbol{\omega} \times \{\mathbf{R}(t)\mathbf{p}(0)\} + \mathbf{v}(t) \end{aligned} \tag{2.6}$$

This relationship is shown graphically in Figure 2.2.

There is a linear relationship between the velocity and time-derivative of the position vector, i.e., $\mathbf{v} = \dot{\mathbf{x}}_{cm}$. There is a similar relationship for

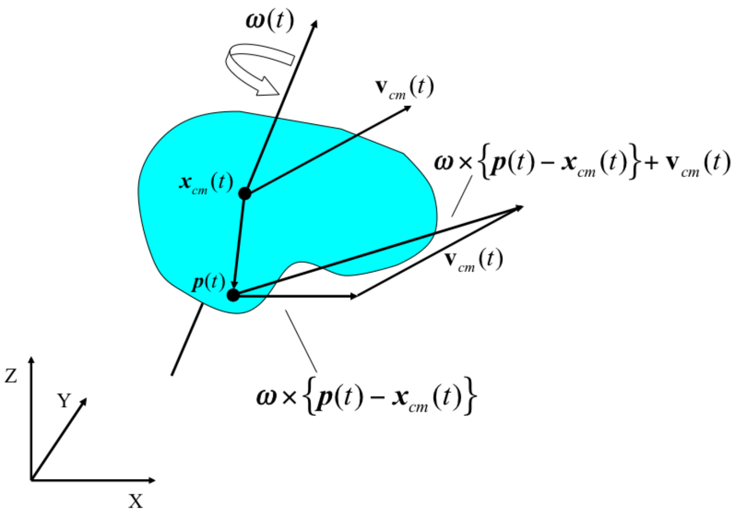


Figure 2.2: Linear and angular velocity components used in the computation of the velocity at a point.

the rotation,

$$\dot{\mathbf{R}}(t) = \boldsymbol{\omega}(t)^{*}\mathbf{R}(t) \tag{2.7}$$

The '*' notation indicates the following operation

$$\begin{aligned}
\boldsymbol{\omega}(t)^{*}\mathbf{R}(t) &= \left[\; \boldsymbol{\omega}(t) \times \left\{ \begin{array}{c} r_{xx} \\ r_{xy} \\ r_{xz} \end{array} \right\} \; \boldsymbol{\omega}(t) \times \left\{ \begin{array}{c} r_{yx} \\ r_{yy} \\ r_{yz} \end{array} \right\} \; \boldsymbol{\omega}(t) \times \left\{ \begin{array}{c} r_{zx} \\ r_{zy} \\ r_{zz} \end{array} \right\} \;\right] \\
&= \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} R_{xx} & R_{yx} & R_{zx} \\ R_{xy} & R_{yy} & R_{zy} \\ R_{xz} & R_{yz} & R_{zz} \end{bmatrix}
\end{aligned} \tag{2.8}$$

Thus, $\boldsymbol{\omega}(t)^{*}$ is a skew-symmetric operator defined as

$$\boldsymbol{\omega}(t) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.9}$$

As an example, if **a** and **b** are vectors, then

$$\mathbf{a} \times \mathbf{b} = \left\{ \begin{array}{c} a_y b_z - b_y a_z \\ -a_x b_z + b_x a_z \\ a_x b_y - b_x a_y \end{array} \right\} \tag{2.10}$$

If we define the matrix $\mathbf{a}^{*}$ to be

$$\mathbf{a}^{*} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{2.11}$$

then

$$\mathbf{a}^*\mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} = \mathbf{a} \times \mathbf{b} \tag{2.12}$$

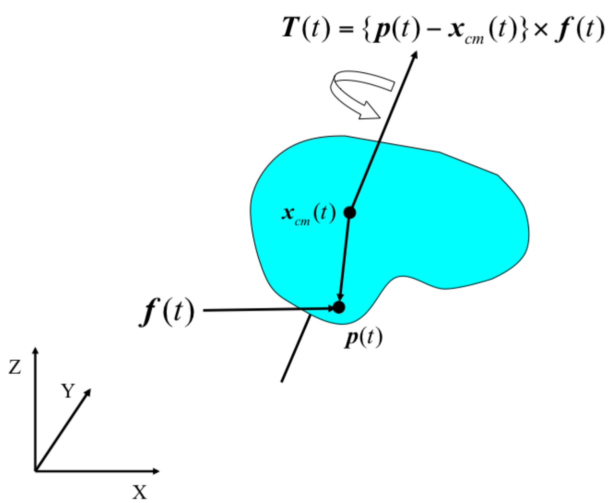
Forces and Moments

The net force on a rigid body, $\mathbf{f}(t)$, is simply the sum of all the external forces applied. The applied moments or torque, $\mathbf{T}(t)$, may be computed in terms of the applied forces at a given point $\mathbf{p}$ as

$$\mathbf{T}(t) = \{\mathbf{p}(t) - \mathbf{x}_{cm}(t)\} \times \mathbf{f}(t) \tag{2.13}$$

which is shown in Figure 2.3.

Figure 2.3: Forces and moments on the rigid body.



Linear and Angular Momentum

The linear momentum of a rigid body is given as

$$\mathbf{P}(t) \quad = \quad m\mathbf{v}(t) \tag{2.14}$$

where m is the mass of the body and $\mathbf{v}(t)$ is the velocity of the body at time t.

The conservation of linear momentum is given by

$$\frac{d\mathbf{P}}{dt} = m\frac{d\mathbf{v}}{dt} = \mathbf{f}(t) \tag{2.15}$$

where $\mathbf{f}$ represents the net external forces acting on the body.

Similarly, the angular momentum of a rigid body is defined as

$$\mathbf{L} = \mathbf{I}\boldsymbol{\omega} \tag{2.16}$$

where $\mathbf{I}$ is the inertia tensor and $\boldsymbol{\omega}$ is the angular velocity of the body. Conservation of angular momentum may be written as

$$\frac{d\mathbf{L}}{dt} = \mathbf{T}(t) \tag{2.17}$$

where $\mathbf{T}$ represents the net external moments (or torque) acting on the body.

Rigid-Body Equations of Motion

The state of a rigid body at any given time, t, is completely defined by it's position $\mathbf{x}(t)$, it's orientation described by $\mathbf{R}(t)$, it's linear momentum $\mathbf{P}(t)$, and it's angular momentum $\mathbf{L}(t)$. These quantities can be written in terms of a state vector, $\mathbf{Y}$, for the body as

$$\mathbf{Y} = \left\{ \begin{array}{c} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{array} \right\} \tag{2.18}$$

Thus, $\mathbf{Y}$ is a vector with eighteen components (three for position $\mathbf{x}$, nine for the rotation tensor $\mathbf{R}$, three for the linear momentum $\mathbf{P}$, and three for the angular momentum $\mathbf{L}$).

The motion of the rigid body is completely described by the evolution of this state vector over time. Thus, the rigid-body equations of motion can be represented by a first-order system in time as follows,

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\left\{ \begin{array}{c} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{v}(t) \\ \boldsymbol{\omega}(t) * \mathbf{R}(t) \\ \mathbf{f}(t) \\ \mathbf{T}(t) \end{array} \right\}. \tag{2.19}$$

Equation (2.19) can be integrated term-by-term to obtain the state of the rigid body at each time step. The time integration methods for this are presented in Chapter 4.

Kinetic Energy

The kinetic energy for general rigid body is given by

$$E_k = \frac{1}{2}m\|\mathbf{v}\|^2 + \sum \frac{1}{2}m_i r_i^2 \tag{2.20}$$

where m is the total mass of the body, $\mathbf{v}$ is the velocity of the center-of-mass, and r_i is the magnitude of the position vector of an element m_i with respect to the center-of-mass. These quantities are illustrated in Figure 2.4. The linear and angular momentum of the body are included in the figure as well.

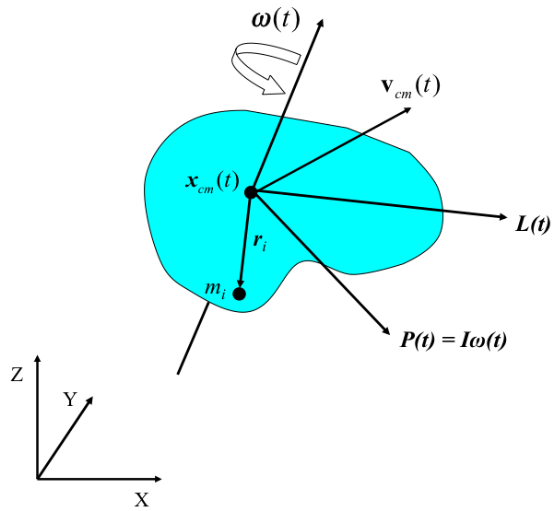


Figure 2.4: General rigid body with total mass m, velocity $\mathbf{v}$, linear momentum $\mathbf{P}$, and angular momentum $\mathbf{L}$.

The kinetic energy can be re-written in terms of the linear and angular momentum as follows,

$$E_k = \frac{1}{2}\mathbf{v} \cdot \mathbf{P} + \frac{1}{2}\boldsymbol{\omega} \cdot \mathbf{L} \tag{2.21}$$

Expanding the relation for angular momentum gives

$$\begin{aligned} \mathbf{L} \;&=\; (\;\; I_{xx}\omega_x - I_{xy}\omega_y - I_{xz}\omega_z)\,\mathbf{i} \\ &+\; (-I_{yx}\omega_x + I_{yy}\omega_y - I_{yz}\omega_z)\,\mathbf{j} \\ &+\; (-I_{zx}\omega_x - I_{zy}\omega_y + I_{zz}\omega_z)\,\mathbf{k} \end{aligned} \tag{2.22}$$

where $\mathbf{I}$ is the moment of inertia tensor about the center-of-mass.

Combining Eq. (2.21) and (2.22), the kinetic energy for a three-dimensional rigid body is

$$\begin{aligned} E_k = \frac{1}{2}m\mathbf{v}^2 \;&+\; \frac{1}{2}(I_{xx}\omega_x^2 + I_{yy}\omega_x^2 + I_{zz}\omega_z^2) \\ &-\; (I_{xy}\omega_x\omega_y + I_{xz}\omega_x\omega_z + I_{yz}\omega_y\omega_z) \end{aligned} \tag{2.23}$$

3 Quaternions and Robust Rotations

The 6-DOF formulation given by equation 2.19 involves a rotation tensor to describe the orientation of the body. However, use of a rotation tensor can present problems. Numerical errors will accumulate in the terms of the rotation tensor as the equations of motion are integrated. As this happens, the rotation tensor is loses its orthogonal properties. This manifests itself as skewing in the body rotations. Another problem with rotation matrices is that they can suffer from gimbal lock. That is, rotations in multiples of 90-degrees yield divide-by-zero errors due to the cosine terms in the rotation tensor.

As an alternative, a unit quaternion can be used instead of a rotation tensor. A unit quaternion is a four element vector normalized to unit length. To gain a physical understanding of quaternions, consider how a rotation tensor is formed. A rotation tensor is basically made up of three successive rotations, one about each coordinate axis. The same reorientation of a body can be accomplished by a single rotation, θ, about a single axis through the body. The unit vector, $\boldsymbol{\lambda}$, describes the orientation of this axis. This is illustrated in Figure 3.1.

This type of description may be cast in terms of a quaternion which is a four element vector represented as

$$\mathbf{q}(t) = s(t) + v_x(t)\mathbf{i} + v_y(t)\mathbf{j} + v_z(t)\mathbf{k} \tag{3.1}$$

Here s, v_x, v_y, and v_z are commonly referred to as Euler parameters. The quaternion can also be represented as the pair {s,$\mathbf{v}$}.

The terms of the quaternion, s and $\mathbf{v}$, are related to θ and $\boldsymbol{\lambda}$ as follows

$$\begin{aligned} s &= cos\frac{\theta}{2} \\ v_x &= \lambda_x sin\frac{\theta}{2} \\ v_y &= \lambda_y sin\frac{\theta}{2} \\ v_z &= \lambda_z sin\frac{\theta}{2} \end{aligned} \tag{3.2}$$

The terms of the quaternion are not independent and are related by

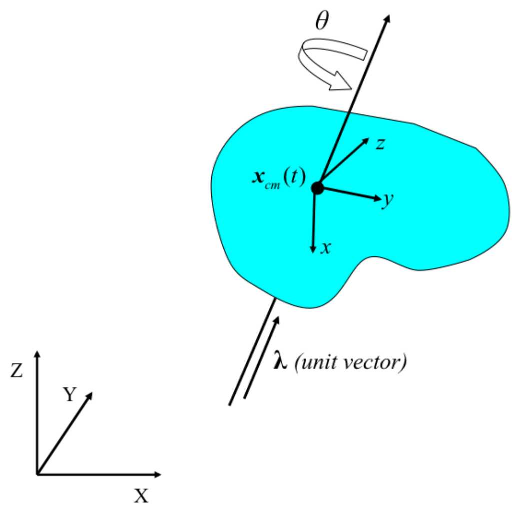


Figure 3.1: Body reorientation described by a rotation, θ, about an axis defined by the unit vector $\boldsymbol{\lambda}$.

the following equations

$$\begin{aligned} s^2 + v_x^2 + v_y^2 + v_z^2 &= 1 \\ {\lambda_x}^2 + {\lambda_x}^2 + {\lambda_x}^2 &= 1 \end{aligned} \tag{3.3}$$

As can be seen from Eq. (3.2), quaternions will not suffer from gimbal lock like a rotation tensor, because the terms will have finite value for any rotation θ. In addition, the quaternion can be re-normalized to unity every time-step preventing numerical errors from accumulating and skewing the rigid body during rotation. The third advantage of quaternions is that they require only four terms instead of the nine values required for a rotation tensor. For these reasons, quaternions will be used instead of rotation tensors in the equations of motions presented here.

Quaternion Multiplication

To use quaternions, a formula for multiplication with a four element vector must be established. This relation is

$$\{s_1, \mathbf{v}_1\}\{s_2, \mathbf{v}_2\} = \{s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2\} \tag{3.4}$$

This equation can also be used to multiply a quaternion by a vector simply by representing the vector as the pair {0,**v**}.

Recovering a Rotation Tensor from a Quaternion

A rotation tensor can be derived from a quaternion at any time, t. This is necessary to evolve the moment of inertia tensor, $\mathbf{I}$, calculate angular velocities, and calculate nodal positions. The relation between the terms of a quaternion and a rotation tensor is

$$[\mathbf{R}(t)] = \begin{bmatrix} 1-2v_y^2-2v_z^2 & 2v_xv_y-2sv_z & 2v_xv_z+2sv_y \\ 2v_xv_y+2sv_z & 1-2v_x^2-2v_z^2 & 2v_yv_z-2sv_x \\ 2v_xv_z-2sv_y & 2v_yv_z+2sv_x & 1-2v_x^2-2v_y^2 \end{bmatrix} \quad (3.5)$$

Equations of Motion Using Quaternions

The rotation tensor in the state vector, $\mathbf{Y}(t)$, can be replaced by a quaternion, $\mathbf{q}(t)$. However, a relation for $\dot{q}(t)$ is required. This relation is

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\boldsymbol{\omega}(t)\mathbf{q}(t). \quad (3.6)$$

The state vector, $\mathbf{Y}(t)$, and the equations of motion can now be written as

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{Bmatrix} \mathbf{x}(t) \\ \mathbf{q}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{Bmatrix} = \begin{Bmatrix} \mathbf{v}(t) \\ \frac{1}{2}\boldsymbol{\omega}(t)\mathbf{q}(t) \\ \mathbf{f}(t) \\ \mathbf{T}(t) \end{Bmatrix} \quad (3.7)$$

Equation (3.7) can be integrated term-by-term to obtain the state of the rigid body at each time step. Integration routines for this process are presented in Section §4.

4 Time Integration Methods

The rigid body dynamics solver is intended to be a light weight computational component for coupled fluid-solid interaction problems. This chapter presents the Runge-Kutta time integrators used in the rigid body dynamics solver, and a summary of the solution procedure and output processing steps.

The rigid body equations of motion are a first-order system of ordinary differential equations (ODEs). Using a quaternion to represent the rotation degrees-of-freedom, the system may be written as

$$\dot{\mathbf{Y}}(t) = \boldsymbol{\mathcal{F}}(\mathbf{Y}, t) \tag{4.1}$$

where the state vector $\mathbf{Y}$ is written in terms of the position $\mathbf{x}$, quaternion (rotation) $\mathbf{q}$, linear momentum $\mathbf{P}$, and angular momentum $\mathbf{L}$

$$\mathbf{Y}(t) = \left\{ \begin{array}{c} \mathbf{x}(t) \\ \mathbf{q}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{array} \right\} \tag{4.2}$$

and $\boldsymbol{\mathcal{F}}(\mathbf{Y}, t)$ is

$$\boldsymbol{\mathcal{F}}(\mathbf{Y}, t) = \left\{ \begin{array}{c} \mathbf{v}(t) \\ \frac{1}{2}\boldsymbol{\omega}(t)\mathbf{q}(t) \\ \mathbf{f}(t) \\ \mathbf{T}(t) \end{array} \right\} \tag{4.3}$$

Runge-Kutta Methods

The first-order system of ODEs is integrated using either a second-order or a fourth-order Runge-Kutta method. The second-order Runge-Kutta algorithm proceeds as follows.

Algorithm 1 *Second-Order Runge-Kutta (RK2)*

1. *Evaluate $\dot{\mathbf{Y}}$ at time t^n*

$$\dot{\mathbf{Y}}^n = \boldsymbol{\mathcal{F}}(\mathbf{Y}^n, t^n) \tag{4.4}$$

2. *Update the intermediate state vector at $t^{n+1/2}$*

$$\mathbf{Y}_1 = \mathbf{Y}^n + \frac{\Delta t}{2}\boldsymbol{\mathcal{F}}(\mathbf{Y}^n, t^n) \tag{4.5}$$

3. *Using $\mathbf{Y}_1$, evaluate $\dot{\mathbf{Y}}$ at $t^{n+1/2}$*

$$\dot{\mathbf{Y}}^{n+1/2} = \boldsymbol{\mathcal{F}}(\mathbf{Y}_1, t^{n+1/2}) \tag{4.6}$$

4. *Update the final state vector at t^{n+1}*

$$\mathbf{Y}^{n+1} = \mathbf{Y}^n + \Delta t\dot{\mathbf{Y}}^{n+1/2} \tag{4.7}$$

For problems that exhibit large rotations with each time-step, the fourth-order time integrator has been found to provide significantly more accurate results than the RK2 integrator. The fourth-order time integration algorithm proceeds as follows.

Algorithm 2 *Fourth-Order Runge-Kutta (RK4)*

1. *Evaluate $\dot{\mathbf{Y}}$ at time t^n*

$$\dot{\mathbf{Y}}^n = \boldsymbol{\mathcal{F}}_1 = \boldsymbol{\mathcal{F}}(\mathbf{Y}^n, t^n) \tag{4.8}$$

2. *Update the state vector at $t^{n+1/2}$*

$$\mathbf{Y}_1 = \mathbf{Y}^n + \frac{\Delta t}{2}\boldsymbol{\mathcal{F}}_1 \tag{4.9}$$

3. *Using $\mathbf{Y}_1$, evaluate $\dot{\mathbf{Y}} = \boldsymbol{\mathcal{F}}_2$ at $t^{n+1/2}$*

$$\boldsymbol{\mathcal{F}}_2 = \boldsymbol{\mathcal{F}}(\mathbf{Y}_1, t^{n+1/2}) \tag{4.10}$$

4. *Update the intermediate state vector at $t^{n+1/2}$*

$$\mathbf{Y}_2 = \mathbf{Y}^n + \frac{\Delta t}{2}\boldsymbol{\mathcal{F}}_2 \tag{4.11}$$

5. *Using $\mathbf{Y}_2$, evaluate $\dot{\mathbf{Y}} = \boldsymbol{\mathcal{F}}_3$ at $t^{n+1/2}$*

$$\boldsymbol{\mathcal{F}}_3 = \boldsymbol{\mathcal{F}}(\mathbf{Y}_2, t^{n+1/2}) \tag{4.12}$$

6. *Update the intermediate state vector at $t^n + 1$*

$$\mathbf{Y}_3 = \mathbf{Y}^n + \Delta t\boldsymbol{\mathcal{F}}_3 \tag{4.13}$$

7. *Using $\mathbf{Y}_3$, evaluate $\dot{\mathbf{Y}} = \boldsymbol{\mathcal{F}}_4$ at t^n*

$$\boldsymbol{\mathcal{F}}_4 = \boldsymbol{\mathcal{F}}(\mathbf{Y}_3, t^n) \tag{4.14}$$

8. *Update the final state vector at t^{n+1}*

$$\mathbf{Y}^{n+1} = \mathbf{Y}^n + \Delta t\left(\frac{1}{6}\boldsymbol{\mathcal{F}}_1 + \frac{1}{3}\boldsymbol{\mathcal{F}}_2 + \frac{1}{6}\boldsymbol{\mathcal{F}}_3 + \frac{1}{3}\boldsymbol{\mathcal{F}}_4\right) \tag{4.15}$$

Time-Integration Procedure

This section presents an overview of the initialization and algorithmic steps for the overall time-integration process. The initialization includes several pre-processing steps required for the RK2/RK4 time-integrators and for output processing.

Algorithm 3 *Initialization*

Precompute the mass m, center-of-mass $\mathbf{x}_{cm}$*, inertia tensor* $\mathbf{I}_{cm}$ *and* $\mathbf{I}_{cm}^{-1}$

Setup the initial conditions for $\mathbf{Y}$

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_{cm}(0) \\ \mathbf{q}(0) &= [1, 0, 0, 0]^T \\ \mathbf{P}(0) &= m\mathbf{v}_{cm}(0) \\ \mathbf{L}(0) &= \mathbf{I}(0)\boldsymbol{\omega}(0) \end{aligned} \tag{4.16}$$

Here, the quaternion initialization corresponds to setting the rotations to

$$\mathbf{R}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

During the time-integration process, the evaluation of $\mathcal{F}$ requires the following steps

1. Velocity update using the linear momentum and t mass

$$\mathbf{v}_{cm} = \frac{\mathbf{P}}{m} \tag{4.18}$$

2. Normalize the quaternion

$$\|\mathbf{q}\| = 1 \tag{4.19}$$

3. Recover the rotation tensor from the current quaternion

$$[\mathbf{R}] = \begin{bmatrix} 1 - 2v_y^2 - 2v_z^2 & 2v_xv_y - 2sv_z & 2v_xv_z + 2sv_y \\ 2v_xv_y + 2sv_z & 1 - 2v_x^2 - 2v_z^2 & 2v_yv_z - 2sv_x \\ 2v_xv_z - 2sv_y & 2v_yv_z + 2sv_x & 1 - 2v_x^2 - 2v_y^2 \end{bmatrix} \tag{4.20}$$

4. Update the intertia tensors

$$\begin{aligned} \mathbf{I}_{cm} &= \mathbf{R}\ \mathbf{I}_{cm}\ \mathbf{R}^T \\ \mathbf{I}_{cm}^{-1} &= \mathbf{R}\ \mathbf{I}_{cm}^{-1}\ \mathbf{R}^T \end{aligned} \tag{4.21}$$

$$\tag{4.22}$$

5. Compute the angular velocity

$$\boldsymbol{\omega} = \mathbf{I}_{cm}^{-1}\mathbf{L} \tag{4.23}$$

6. Update the nodal position $\mathbf{p}_i$, displacements $\mathbf{u}_i$, and linear velocity $\mathbf{v}_i$ at node points for output processing

$$\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm} \quad 1 \le i \le Nnp \tag{4.24}$$

   where Nnp is the number of nodal points in the rigid-body mesh

$$\mathbf{u}_i = \mathbf{R}\mathbf{p}_i + (\mathbf{x}_{cm}(t) - \mathbf{x}_{cm}(0)) \quad 1 \le i \le Nnp \tag{4.25}$$

$$\mathbf{v}_i = \boldsymbol{\omega} \times \mathbf{p}_i + \mathbf{v}_{cm} \quad 1 \le i \le Nnp \tag{4.26}$$

7. update the global kinetic energy

$$E_k = \frac{1}{2}m\mathbf{v}_{cm} \cdot \mathbf{v}_{cm} + \frac{1}{2}\mathbf{I}_{cm}\boldsymbol{\omega} \cdot \boldsymbol{\omega} \tag{4.27}$$

5 Mass Properties

Mass or inertial properties for a rigid body are required for the rigid body dynamics solver. Given a mass density $\rho(x, y, z)$, the mass is computed as

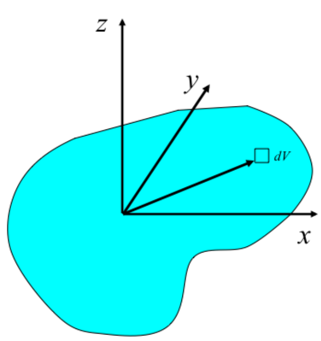
$$m = \int_{\Omega} \rho(x, y, z) \; d\Omega \tag{5.1}$$

where Ω the volume occupied by the rigid body.

The center-of-mass coordinates are

$$\begin{aligned} x_{cm} &= \frac{1}{m} \int_{\Omega} \rho(x, y, z) \; x \; d\Omega \\ y_{cm} &= \frac{1}{m} \int_{\Omega} \rho(x, y, z) \; y \; d\Omega \qquad (5.2) \\ z_{cm} &= \frac{1}{m} \int_{\Omega} \rho(x, y, z) \; z \; d\Omega \qquad (5.3) \end{aligned}$$

Figure 5.1: Location of a differential volume used in the computation of the mass and center-of-mass coordinates.



The inertia tensor is

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{5.4}$$

where

$$\begin{aligned} I_{xx} &= \int_\Omega \rho(x,y,z)[y^2+z^2]\; d\Omega \\ I_{yy} &= \int_\Omega \rho(x,y,z)[x^2+z^2]\; d\Omega \\ I_{zz} &= \int_\Omega \rho(x,y,z)[x^2+y^2]\; d\Omega \qquad (5.5) \\ I_{xy} &= I_{yx} = -\int_\Omega \rho(x,y,z)xy\; d\Omega \\ I_{xz} &= I_{zx} = -\int_\Omega \rho(x,y,z)xz\; d\Omega \\ I_{yz} &= I_{zy} = -\int_\Omega \rho(x,y,z)yz\; d\Omega \end{aligned}$$

(5.6)

6 Bibliography

David Barraff. An introduction to physically based modeling: Rigid body dynamics I - unconstrained rigid body dynamics. Technical report, Carnegie Mellon University, Pittsburgh, PA, 1997a. 1997 Siggraph Course Notes.

David Barraff. An introduction to physically based modeling: Rigid body dynamics II - nonpenetration constraints. Technical report, Carnegie Mellon University, Pittsburgh, PA, 1997b. 1997 Siggraph Course Notes.

7 Index