```
public class QuaternionApp extends AbstractCalculation {
   Display3DFrame frame = new Display3DFrame("Quaternion rotations");
   Quaternion transformation = new Quaternion(1, 0, 0, 0);
   BoxWithArrows box = new BoxWithArrows();

   public QuaternionApp() {
      frame.setDecorationType(VisualizationHints.DECORATION_AXES);
      // scene is simple, so draw it properly when rotating
      frame.setAllowQuickRedraw(false);
      frame.setPreferredMinMax(-6, 6, -6, 6, -6, 6);
      box.setTransformation(transformation);
      frame.addElement(box);
   }

   public void calculate() {
      double q0 = control.getDouble("q0");
      double q1 = control.getDouble("q1");
      double q2 = control.getDouble("q2");
      double q3 = control.getDouble("q3");
      transformation.setCoordinates(q0, q1, q2, q3);
      box.setTransformation(transformation);
   }

   public void reset() {
      control.clearMessages();
      control.setValue("q0", 1);
      control.setValue("q1", 0); // initial orientation is along x-axis
      control.setValue("q2", 0);
      control.setValue("q3", 0);
      calculate();
   }

   public static void main(String[] args) {
      CalculationControl.createApp(new QuaternionApp());
   }
}
```

Exercise 17.12 Quaternion representation of rotations

(a) Compile and run the `QuaternionApp` target class. Find and then test a quaternion that orients the long side of the box at 45° in the xy-plane. Repeat for the yz-plane.

(b) Use a quaternion to orient the long axis of the box along an axis in the $(1, 2, 1)$ direction.

(c) What happens if the quaternion does not have unit norm? ■

17.6 ■ QUATERNION EQUATIONS OF MOTION

Because we will often transform torque and other vectors to and from the rotating object's body frame, we need the transformation matrix from the space frame to the body frame using quaternions. We represent a rotation using a unit quaternion (q_0, q_1, q_2, q_3), carry out

(17.33) using (17.31), and express the result in matrix form. The resulting rotation matrix is

$$\mathcal{R} = 2\begin{bmatrix} \frac{1}{2} - q_2^2 - q_3^2 & q_1q_2 + q_0q_3 & q_1q_3 - q_0q_2 \\ q_1q_2 - q_0q_3 & \frac{1}{2} - q_1^2 - q_3^2 & q_2q_3 + q_0q_1 \\ q_1q_3 + q_0q_2 & q_2q_3 - q_0q_1 & \frac{1}{2} - q_1^2 - q_2^2 \end{bmatrix}. \tag{17.34}$$

This matrix is equal to the rotation matrix derived from the Rodrigues formula (17.15).

The angular velocity in the body frame can be written as $\dot{\hat{q}}(t) = \frac{1}{2}\hat{\omega}(t)\hat{q}(t)$, where $\hat{\omega}$ is a pure quaternion $(0, \omega_1, \omega_2, \omega_3)$. Our discussion follows the derivation in Rapaport. The time dependence of a vector $\mathbf{r}$ can be expressed as a transformation of its initial value $\mathbf{r}_0$ as

$$\hat{r}(t) = \hat{q}(t)\hat{r}_0\hat{q}^*(t). \tag{17.35}$$

If we differentiate $\hat{r}(t)$ with respect to time, we have

$$\dot{\hat{r}} = \dot{\hat{q}}\hat{r}_0\hat{q}^* + \hat{q}\hat{r}_0\dot{\hat{q}}^*, \tag{17.36}$$

where we have dropped the explicit time dependence of $\hat{q}$. We substitute $\hat{r}_0 = \hat{q}^*\hat{r}\hat{q}$ and obtain

$$\dot{\hat{r}} = \dot{\hat{q}}\hat{q}^*\hat{r} + \hat{r}\hat{q}\dot{\hat{q}}^* = \dot{\hat{q}}\hat{q}^*\hat{r} - \hat{r}\dot{\hat{q}}\hat{q}^*, \tag{17.37}$$

where we have used the fact $\hat{q}\hat{q}^* = 1$. The only part of the $\hat{r}$ and $\dot{\hat{q}}\hat{q}^*$ product that does not commute is the vector cross product. The scalar part commutes and is zero. If we denote the pure (vector) part of the quaternion $\dot{\hat{q}}\hat{q}^*$ by $\mathbf{u}$, we find

$$\dot{\mathbf{r}} = \mathbf{u} \times \mathbf{r} - \mathbf{r} \times \mathbf{u} = 2\mathbf{u} \times \mathbf{r}. \tag{17.38}$$

Because $\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}$ for rotational motion, we obtain

$$\boldsymbol{\omega} = 2\mathbf{u}. \tag{17.39}$$

Equation (17.39) can be expressed using components by writing $\boldsymbol{\omega}$ as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = 2\mathcal{W}\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \tag{17.40}$$

where

$$\mathcal{W} = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \\ q_0 & q_1 & q_2 & q_3 \end{bmatrix}. \tag{17.41}$$

Because $\mathcal{W}$ is orthogonal, the transpose of (17.41) is its inverse $\mathcal{W}^T\mathcal{W} = \mathbf{1}$, and

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\mathcal{W}^T\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix}. \tag{17.42}$$