Research Methods in Mathematics
Extended assignment 2: Rotations in 3 dimensions

T. PERUTZ

Due: beginning of class, 4 November. The extra-credit opportunity will remain open until the last day of classes.

If you're writing a 3D computer graphics package, an object (a triangle, say) might be specified by the coordinates (x, y, z) of various points (e.g. the vertices of the triangle). We represent this object on our 2-dimensional screen by a *projection*: the simplest possibility is to plot (x, y) and ignore z.

Suppose we want to view this object from a chosen camera-angle. Equivalently, we would like to be able to rotate the object by a chosen angle, about a chosen axis. In this assignment, we work out a formula to do this. An extra-credit opportunity invites you to write a computer program to implement the formula.

So: *we want to write down a* 3×3 *matrix* $R_{\mathbf{a},\theta}$ *whose associated matrix transformation* $rot_{\mathbf{a},\theta}$ *is rotation by* θ *about an axis pointing in the direction of vector* $\mathbf{a}$*. Here* $\mathbf{a}$ *is a non-zero vector which we normalize to have unit length:* $\|\mathbf{a}\| = 1$.

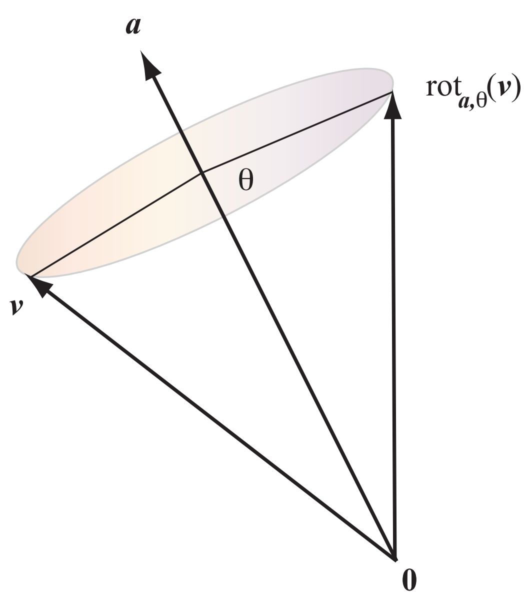
Such a rotation is depicted in the figure overleaf.

It is possible to do this by a direct but rather tricky calculation; see
`http://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula`

In this assignment, we'll explore a different method to obtain the matrix $R_{\mathbf{a},\theta}$. The virtue of this method is that it is *extremely* adaptable: it applies to all sorts of other mathematical problems (ranging from change-of-basis in linear algebra to solving differential equations using Laplace transforms).

The method is to apply a change of coordinate that makes the problem easy; solve it in those coordinates; and then apply the reverse change of coordinates. We will get a formula for $R_{\mathbf{a},\theta}$ in the form XRX^{-1}, where R is a particularly simple rotation matrix, and X is a matrix expressing a change of coordinates.

(1) Let's first set up convenient coordinates. Our axis of rotation points in the direction of the unit-vector $\mathbf{a}$ (a unit vector is one with length 1, so $\mathbf{a}\cdot\mathbf{a} = 1$). Let $\mathbf{b}$ be a unit-vector orthogonal to $\mathbf{a}$ (that is, $\mathbf{a}\cdot\mathbf{b} = 0$ and $\mathbf{b}\cdot\mathbf{b} = 1$). Let



$\mathbf{c} = \mathbf{a} \times \mathbf{b}$. We will refer our coordinates to the $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ axes. That is, we will represent a typical vector as $x\mathbf{a} + y\mathbf{b} + z\mathbf{c}$.
To start things off, write down $\mathbf{c} \cdot \mathbf{c}$, $\mathbf{c} \cdot \mathbf{a}$ and $\mathbf{c} \cdot \mathbf{a}$. Draw a diagram representing $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. Is this a right- or a left-handed coordinate system?

(2) Explain, with diagrams, why our sought rotation matrix $R_{\mathbf{a},\theta}$ should satisfy

$$R_{\mathbf{a},\theta}\mathbf{a} = \mathbf{a}$$

and

$$R_{\mathbf{a},\theta}\mathbf{b} = \cos(\theta)\mathbf{b} + \sin(\theta)\mathbf{c}.$$

What should $R_{\mathbf{a},\theta}\mathbf{c}$ be?

(3) Let $X = [\mathbf{a}\,\mathbf{b}\,\mathbf{c}]$ (3×3 matrix). Explain why X is an invertible matrix with $X^{-1} = X^T$.

(4) Show that $X^{-1}\mathbf{a} = \mathbf{e}_1$, $X^{-1}\mathbf{b} = \mathbf{e}_2$, $X^{-1}\mathbf{c} = \mathbf{e}_3$.

(5) Consider rotation $rot_{\mathbf{e}_3,\theta}$ by θ about the z-axis. What is $rot_{\mathbf{e}_3,\theta}(\mathbf{e}_i)$ for $i = 1, 2, 3$? Write down the 3×3 matrix $R_{\mathbf{e}_3,\theta}(\mathbf{e}_i)$ representing this rotation.

(6) Let $Y = XR_{\mathbf{a},\theta}X^{-1}$. Show that $Y\mathbf{a} = \mathbf{a}$ and $Y\mathbf{b} = \cos(\theta)\mathbf{b} + \sin(\theta)\mathbf{c}$. Calculate $Y\mathbf{c}$. Use your result to deduce that $Y = R_{\mathbf{a},\theta}$.

(7) Use the fact that $X^{-1} = X^T$, and the formula $R_{\mathbf{a},\theta} = XR_{\mathbf{a},\theta}X^{-1}$, to show that

$$\begin{aligned} R_{\mathbf{a},\theta} = \cos\theta & \begin{bmatrix} b_1^2 + c_1^2 & b_2b_1 + c_2c_1 & b_3b_1 + c_3c_1 \\ b_1b_2 + c_1c_2 & b_2^2 + c_2^2 & b_3b_2 + c_3c_2 \\ b_1b_3 + c_1c_3 & b_2b_3 + c_2c_3 & b_3^2 + c_3^2 \end{bmatrix} \\ + \sin\theta & \begin{bmatrix} 0 & b_2c_1 - c_1b_2 & b_3c_1 - c_1b_3 \\ b_1c_2 - c_2b_1 & 0 & b_3c_2 - c_2b_3 \\ b_1c_3 - b_3c_1 & b_2c_3 - b_3c_2 & 0 \end{bmatrix} \\ + & \begin{bmatrix} a_1^2 & a_2a_1 & a_3a_1 \\ a_1a_2 & a_2^2 & a_3a_2 \\ a_1a_3 & a_2a_3 & a_3^2 \end{bmatrix}. \end{aligned}$$

(8) The trouble with the formula above (besides its length) is that it involves $\mathbf{b}$ and $\mathbf{c}$ which are auxiliary choices and not part of the given data $\mathbf{a}$, θ. Use what we know about the dot products of $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, and about the cross product $\mathbf{a} \times \mathbf{c}$, to simplify this to

$$\begin{aligned} R_{\mathbf{a},\theta} = \cos\theta & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ + \sin\theta & \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \\ +(1 - \cos\theta) & \begin{bmatrix} a_1^2 & a_2a_1 & a_3a_1 \\ a_1a_2 & a_2^2 & a_3a_2 \\ a_1a_3 & a_2a_3 & a_3^2 \end{bmatrix}. \end{aligned}$$

(9) Use the formula from the previous part to show that

$$R_{\mathbf{a},\theta}\mathbf{v} = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{a} \times \mathbf{v} + (1 - \cos\theta)(\mathbf{a} \cdot \mathbf{v})\mathbf{a}.$$

(10) *Extra credit.* Write a computer program to implement this formula. You could use a programming language such as C++, Java or Python, or a specialized mathematics language such as Maple, Matlab, Mathematica or Sage (the last one is freely available.)
We can represent a point (x, y, z) on the screen by simply plotting (x, y).[1] Write a program that starts by plotting some simple shape (say, a square in 3-dimensional

[1] Of course, one can devise much more realistic projections than this one.

space) and show the result of rotating it about a chosen axis by angles that gradually increase from 0 to 2π. (You should submit the code and some annotated screen-shots.)