Building a rotation matrix directly

Building a rotation matrix from Euler angles, or even angle-axis, can be tricky. One frequently used trick is to build a matrix from two 3D coordinate frames.

Assume you want to rotate the standard coordinate frame vectors i, j and k to or from a new coordinate frame u, v, n so i -> u, j -> v and k -> n. You can build the orthonormal rotation matrix directly by putting the u, v and n vectors in as rows:

\[
M_r = \begin{bmatrix}
u_x & u_y & u_z \\
v_x & v_y & v_z \\
x_n & y_n & z_n
\end{bmatrix} = \begin{bmatrix}
u_x \\
v_y \\
x_z
\end{bmatrix}
\]

Now when we put in the vectors u, v and n, we get the standard vectors as output. This uses the fact that u, v and n are orthonormal – dot(u, u) = 1 but dot(u, v) = dot(u, n) = 0.

\[
M_r \bar{u} = \begin{bmatrix}
u_x & u_y & u_z \\
v_x & v_y & v_z \\
x_n & y_n & z_n
\end{bmatrix} \begin{bmatrix}
u_x \\
v_y \\
x_z
\end{bmatrix} = \begin{bmatrix}
u \cdot \bar{u} \\
v \cdot \bar{u} \\
x \cdot \bar{u}
\end{bmatrix} = \begin{bmatrix}1 \\
0 \\
0
\end{bmatrix}
\]

This trick works in reverse. If we want to rotate the i, j and k vectors to become u, v and n, we can build the matrix with u, v and n for columns.

\[
M_r^{-1} = M_r^T = \begin{bmatrix}
u_x & u_y & u_z \\
v_x & v_y & v_z \\
x_n & y_n & z_n
\end{bmatrix}^T = \begin{bmatrix}
u_x & vx & nx \\
v_y & vy & ny \\
x_z & vz & nz
\end{bmatrix}
\]

Now we have i -> u as in

\[
M_r^{-1} i = \begin{bmatrix}
u_x & vx & nx \\
v_y & vy & ny \\
x_z & vz & nz
\end{bmatrix} \begin{bmatrix}1 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}u_x \\
u_y \\
u_z
\end{bmatrix}
\]

Finally, we can observe that the product of M and M inverse is the identity matrix.

\[
M_r M_r^{-1} = \begin{bmatrix}1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]