1 Matrix representations of canonical matrices

2-d rotation around the origin:

$$R_0 = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

3-d rotation around the x-axis:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

3-d rotation around the y-axis:

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

3-d rotation around the z-axis:

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4-d rotation around the x-y plane:

$$R_{xy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{pmatrix}$$

4-d rotation around the y-z plane:

$$R_{yz} = \begin{pmatrix} \cos\theta & 0 & 0 & \sin\theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta & 0 & 0 & \cos\theta \end{pmatrix}$$

4-d rotation around the z-w plane:

$$R_{zw} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4-d rotation around the x-w plane:

$$R_{xw} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4-d rotation around the y-w plane:

$$R_{yw} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4-d rotation around the x-z plane:

$$R_{xz} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

In general, the diagonals alternate signs, with the main diagonal and sub-diagonal positive.

2 Constructing arbitrary rotation matrices

To construct an arbitrary 4D rotation matrix around the u-v plane, where $\mathbf{u}$ and $\mathbf{v}$ are orthonormal, let M be a rotation that transforms the x-y plane into the u-v plane. Then $R_{uv} = MR_{xy}M^{-1}$. We are left to construct M.

The choice of the first two columns of M is clear: they should be the images of $\hat{x}$ and $\hat{y}$ under M, namely, $\mathbf{u}$ and $\mathbf{v}$ themselves. The remaining two columns, $\mathbf{z}'$ and $\mathbf{w}'$, must be chosen such that $\{\mathbf{u}, \mathbf{v}, \mathbf{z}', \mathbf{w}'\}$ is pairwise orthonormal.

To solve for $\mathbf{z}'$ and $\mathbf{w}'$, start from the fact that

$$\mathbf{u} \cdot \mathbf{r} = 0 \tag{1}$$

and

$$\mathbf{v} \cdot \mathbf{r} = 0 \tag{2}$$

for every $\mathbf{r}$ on the plane perpendicular to both $\mathbf{u}$ and $\mathbf{v}$. We want two such vectors $\mathbf{r}$, and we want them to be orthonormal. So we have two equations in four unknowns, which means the solutions fill a plane (that's good). Let

$$\mathbf{u} = \begin{pmatrix} a_1 & b_1 & c_1 & d_1 \end{pmatrix}$$

and

$$\mathbf{v} = \begin{pmatrix} a_2 & b_2 & c_2 & d_2 \end{pmatrix}.$$

After some algebra (there must be a better way...) we can find at least one solution:

$$\mathbf{r}_1 = \left(0, 1, -\frac{b_1 d_2 - d_1 b_2}{c_1 d_2 - d_1 c_2}, -\frac{1}{d_2}(b_2 + c_2 z)\right).$$

This vector was arrived at by first solving for w in Equation 2 and plugging the result into Equation 1. The resulting equation was solved for z. Then I picked 0 and 1 for x and y to get $\mathbf{r}_1$. If we let $\mathbf{z}' = \mathbf{r}_1/|\mathbf{r}_1|$, then $\mathbf{z}'$ will be of unit length and will be orthogonal to both $\mathbf{u}$ and $\mathbf{v}$.

We still need $\mathbf{w}'$. But this is simple: it will just be $\mathbf{w}' = -\mathbf{u} \% \mathbf{v} \% \mathbf{z}'$. Note that the cross-product is negated because of the claim (still more or less unjustified) that in a right-handed system, we should generate each new dimension by taking the cross-product of the previous three, negated whenever going to an even number of dimensions.

Thus

$$M = \begin{pmatrix} | & | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{z}' & \mathbf{w}' \\ | & | & | & | \end{pmatrix} \tag{3}$$

3 Rotation by means of normal vector

Here I propose a method for the rotation of a quadric surface via its normal vector. If $\mathbf{n}(\mathbf{x})$ the normal vector to a given surface at point $\mathbf{x}$, and R is a rotation matrix, then at point $R\mathbf{x}$ (which is on the rotated surface), it is reasonable to guess that the normal vector will be $R\mathbf{n}(\mathbf{x})$. Since the normal vector is aquired via differentiation, I am led to believe that integration should provide a means for finding the equation of the rotated surface. At the very least, *some* surface corresponds to the rotated normal field, since the rotated normal field is stil a gradient field. That is, $\exists g$ s.t. $\nabla g = R\mathbf{n}$.

There are two main obstacles here. The first is the computation of g, which is unique up to a constant term. The second is the choice of that constant term.

The following notation will be used:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(Ax_i + \sum_{j=i+1}^{n} B_{ij} x_j + C_i \right) x_i + D. \tag{4}$$

That is, the A_i are the coefficients of quadratic terms; B_{ij} are coefficients of cross-terms; and C_i are coefficients of linear terms.

3.1 Computation of g

Suppose $f : \mathbf{R}^n \to \mathbf{R}$. Then the solutions of $f(\mathbf{x}) = 0$ lie on some n-dimensional hypersurface, and $\mathbf{n}(\mathbf{x}) = \nabla f(\mathbf{x})$ gives a vector normal to the hypersurface at point $\mathbf{x}$.

Let $R : \mathbf{R}^n \to \mathbf{R}^n$ be a rotation. Specifically, let

$$R = \begin{pmatrix} — & \mathbf{r}_1 & — \\ — & \mathbf{r}_2 & — \\ & \vdots & \\ — & \mathbf{r}_n & — \end{pmatrix}.$$

Then

$$R\mathbf{n} = R\nabla f = \begin{pmatrix} \mathbf{r}_1 \cdot \nabla f \\ \mathbf{r}_2 \cdot \nabla f \\ \vdots \\ \mathbf{r}_n \cdot \nabla f \end{pmatrix}.$$

Next, we integrate the result, the ith component with respect to the ith coordinate. That is,

$$\begin{aligned}
\int \mathbf{r}_1 \cdot \nabla f \; dx_1 &= g_1 + h_1(x_2, x_3, x_4, \ldots, x_n) \\
\int \mathbf{r}_2 \cdot \nabla f \; dx_2 &= g_2 + h_2(x_1, x_3, x_4, \ldots, x_n) \\
&\vdots \\
\int \mathbf{r}_k \cdot \nabla f \; dx_k &= g_k + h_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{n-1}) \\
&\vdots \\
\int \mathbf{r}_n \cdot \nabla f \; dx_n &= g_n + h_n(x_1, x_2, x_3, \ldots, x_{n-1})
\end{aligned}$$

Also worth noting is that, since g is assumed to be well-defined, all the RHS's in the above system of n equations in n unknowns happen to be equal. Thus the derivative of any term $g_k + h_k(\ldots)$ w.r.t x_m is known: it is nothing other than $r_m \cdot \nabla f$:

$$\frac{\partial g}{\partial x_m} = r_m \cdot \nabla f.$$

We can derive the following formula:

$$\frac{\partial g_k}{\partial x_m} = r_{km} \left(2A_k x_k + \sum_{\substack{i=1 \\ i \neq k}}^{n} B_{ki} x_i + C_k \right) + \sum_{\substack{i=1 \\ i \neq k}}^{n} r_{ki} B_{ki} x_i. \tag{5}$$

Refer to Eqn 4 for the meanings of the coefficients.

The next step is to find a similar expression for $\partial / \partial x_m \; (g_k + h_k)$ (which must be equal to either side of Eqn 5), cancel out terms, and arrive at an equation for h_k, which hopefully we can solve to find a unique (up to a constant term) expression for g.

3.2 Choice of the constant term

4 Implementation of translation

4.1 Alternate representation for quadric hypersurfaces that might make translations easier

First, I'll derive another representation for $f(\mathbf{x}) = 0$. Group the terms of f in the following way (using the same notation as in Eqn 4):

$$f(\mathbf{x}) = A_1 x_1^2 + C_1 x_1 + B_{12} x_1 x_2 + B_{13} x_1 x_3 + \cdots B_{1n} x_1 x_n$$

$$+ A_2 x_2^2 + C_2 x_2 + B_{23} x_2 x_3 + B_{24} x_2 x_4 + \cdots B_{2n} x_2 x_n$$

$$+ \cdots + A_n x_n^2 + C_n x_n + D.$$

Now we can complete the square involving the A and C terms:

$$A_i x_i^2 + C_i x_i = A_i \left(x_i^2 + \frac{C_i}{A_i} x_i \right)$$

$$= A_i \left[\left(x_i + \frac{C_i}{2A_i} \right)^2 - \frac{C_i^2}{4A_i^2} \right] = A_i \left(x_i + \frac{C_i}{2A_i} \right)^2 - \frac{C_i^2}{4A_i}.$$

Finally we can write f as a sum:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left[\underbrace{A_i \left(x_i + \frac{C_i}{2A_i} \right)^2}_{\text{square and linear terms}} + \underbrace{\sum_{j=i+1}^{n} B_{ij} x_i x_j}_{\text{cross terms}} \underbrace{- \frac{C_i^2}{4A_i}}_{\text{constant terms}} \right]. \tag{6}$$

Note how this is *almost* a description of a quadric hypersurface that has been translated by the vector $\mathbf{c} = -(C_1/2A_1, C_2/2A_2, \ldots, C_n/2A_n)$. The way in which this isn't the case is in the cross terms; we'd like to see something like

$$(x_i - c_i) \sum_{j=i+1}^{n} B_{ij} (x_j - c_j);$$

sadly this is not the case.

Can we force it to be the case? Let's try:

$$(x_i - c_i)(x_j - c_j) = x_i x_j - c_i x_i - c_j x_j + c_i c_j.$$

Then the "cross terms" section of Eqn 6 would be equivalent to the following:

$$\sum_{j=i+1}^{n} B_{ij} x_i x_j \underbrace{- \sum_{j=i+1}^{n} B_{ij} (c_i x_i + c_j x_j + c_i c_j)}_{\text{error terms}}. \tag{7}$$

These “error terms” are effectively changes that need to be made to the linear and constant terms. At the moment, however, I can’t seem to wrap my mind around this. Does it just mean we change C_i’s and D? But then we’d be using different values in the “square and linear terms” sections than we originally planned, which means we’re effectively doing a different translation... hmm. I see no way to reconcile this.