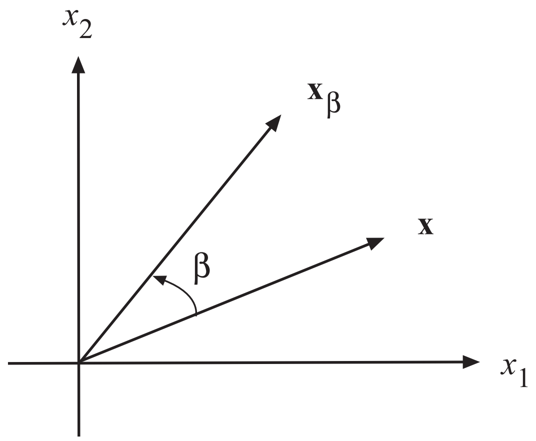
Holger Broman, August 19, 1999

Rotations

Applications of rotations are extensive in computer graphics. The car designer who wants to look at a new design from different angles, the chemist interested in the 3D structure of molecules are both applying rotations. We will start by considering rotations in a plane:



Let us make use of the results obtained in "Norm-preserving linear mappings from $\mathbb{R}^N$ to $\mathbb{R}^N$". First, convince yourself that rotation is a linear mapping. The preservation of the norm is obvious. Thus, there exists a rotation matrix, dependent on the rotation angle, β. Let us call this matrix R_β, and we know that

$$R_\beta^T R_\beta = I,$$

as it preserves the norm. Some further consideration tells us that the inverse of R_β must be the matrix $R_{-\beta}$, as this latter matrix will take us back:

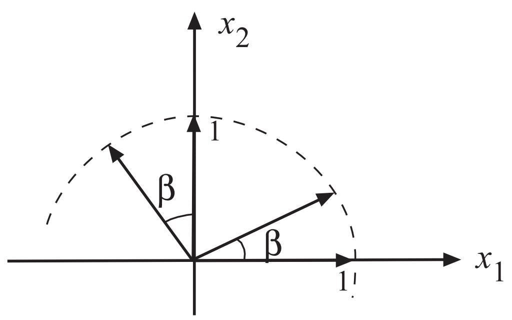
$$R_{-\beta} R_\beta \mathbf{x} = \mathbf{x}$$

We thus know

$$R_\beta^{-1} = R_\beta^T = R_{-\beta},$$

and that R_β most reasonably will contain $\cos(\beta)$ and $\sin(\beta)$ as elements.

Let us set up two special cases:



Obviously,

$$\begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix} = R_\beta \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} -\sin\beta \\ \cos\beta \end{pmatrix} = R_\beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We find

$$R_\beta = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix}$$

Exercise 1: Check

$$R_\beta^{-1} = R_\beta^T = R_{-\beta}$$

Exercise 2: Use

$$R_\beta \cdot R_\gamma = R_{\beta+\gamma}$$

to derive some trigonometric formulas.

Note 1 The rotation in a plane in N dimensions is performed by the matrix

$$R = \begin{bmatrix}
1 &&&&&&&&&& \\
& \ddots &&&&&&&&& \\
&& 1 &&&&&&&& \\
&&& \cos\beta &&&& -\sin\beta &&& \\
&&&& 1 &&&&&& \\
&&&&& \ddots &&&&& \\
&&&&&& 1 &&&& \\
&&& \sin\beta &&&& \cos\beta &&& \\
&&&&&&&& 1 && \\
&&&&&&&&& \ddots & \\
&&&&&&&&&& 1
\end{bmatrix},$$

zeroes in unmarked positions. Please convince yourself that the proposition in Note 1 is true.

Note 2 When you want to rotate "in several dimensions", you cascade rotation matrices as above.

Note 3 These matrices are named Givens rotations.

Please run the m-file rotations in Matlab.

```
% rotations.m
% run this program several times, random data
%
% The idea is as follows. You can rotate a point around an arbitrary axis
% by moving the axis of rotation and the point to the vertical direction, say,
% rotate around the 'z-axis', and then move the package back to where it came from.
clear

% Generate an axis of rotation, and a point to rotate.
rotax=rand(3,1);
point=rand(3,1);

% move the axis to the vertical in two steps, rotate around the y-axis to the y-z p
% then around the x-axis to coincide with the z-axis. These are two Givens rotation
gamma=atan(rotax(1)/rotax(3));
cg=cos(gamma); sg=sin(gamma);
roty=[cg 0 -sg ; 0 1 0 ; sg 0 cg];
rotax1=roty*rotax;
point1=roty*point;
beta=atan(rotax1(2)/rotax1(3));
cb=cos(beta); sb=sin(beta);
rotx=[1 0 0 ; 0 cb -sb ; 0 sb cb];
rotax2=rotx*rotax1;
point2=rotx*point1;
% check that rotax2 is parallell with the z-axis.

% now, rotate around the 'z-axis'
alpha=2*pi/1000;
ca=cos(alpha); sa=sin(alpha);
rotz=[ ca -sa 0 ; sa ca 0 ; 0 0 1 ];
dum(:,1)=point2;
for l=2:1000
dum(:,l)=rotz*dum(:,l-1);
end

% plot the circle of rotation
figure(1), clf, axis equal, hold on, view(3)
plot3(dum(1,:), dum(2,:), dum(3,:), 'r')
plot3(point2(1), point2(2), point2(3), 'rp')
plot3([0 0],[0 0],[0 1],'k')

% now go back to the original axis of rotation. Note, ' equals inverse.
dum1=rotx'*dum;
dum2=roty'*dum1; % done!
plot3(dum2(1,:), dum2(2,:), dum2(3,:), 'b')
plot3([0 rotax(1)],[0 rotax(2)],[0 rotax(3)],'y')
plot3(point(1), point(2), point(3), 'bp')
```

```
title(' original rotation - yellow/blue')

% In summary, you rotate around the desired axis of rotation by myltiplying the
% vector to the point to be rotated by the matrix
% rottot=roty'*rotx'*rotz*rotx*roty
% with alpha in rotz equal to the desired angle of rotation.
```