Common Transformations in Linear Algebra

(A self-contained series of lectures describing the transformation matrices most commonly encountered in engineering)

Table of Contents

1. Creating an Orthogonal Basis Set

1.A. Introduction to Basis Sets

A basis set is just a set of linearly independent vectors that spans a given space or subspace of interest to us. For example, in three-dimensional space in cartesian coordinates, a basis set for all of 3-D space is

$$\underline{b}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ \underline{b}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } \underline{b}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{1.1}$$

which can also be written as

$$\underline{\underline{B}} = \begin{bmatrix} \underline{b}_1 & \underline{b}_2 & \underline{b}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \underline{i} & \underline{j} & \underline{k} \end{bmatrix} \tag{1.2}$$

This set of vectors spans 3-D space. By spanning we mean that any point in 3-D space can be described as a linear combination of our basis set vectors. However, this basis set is certainly not unique. Other basis sets that span all of 3-D space are

$$\underline{\underline{B}} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \underline{\underline{B}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

because these vectors are all independent. The rank of these basis matrices is 3.

Examples of sets of vectors which are not basis sets are as follows:

$$\underline{\underline{B}} = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \underline{\underline{B}} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } \underline{\underline{B}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The first example is not a basis set because the first two columns are not linearly independent. The rank of this matrix is 2. The second example is not a basis set because this set of vectors does not span all of 3-D space. The rank of this matrix is 2. The third example is not a basis set even though it spans all of 3-D space. It an additional dependent vector. A basis set has the minimum number of basis sets necessary to span the space.

Sometimes we are not interested in all of 3-D space. Perhaps we are only interested in the y=x plane through 3-D space. Then we have a basis set as follows

$$\underline{\underline{B}} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The rank is 2, because we are in a 2-D subspace of 3-D space.

In order to get started talking about orthogonal or orthonormal (meaning both orthogonal and normalized to unity) basis sets, we should start with one such basis set that we already know. As shown above, a basis set need not be orthogonal, normalized, or orthonormal. However, as we shall see, orthogonal and orthonormal basis sets have many pleasing properties. An orthonormal basis set for three-dimensional space in cartesian coordinates is which can also be written as

$$\underline{\underline{B}} = \begin{bmatrix} \underline{b}_1 & \underline{b}_2 & \underline{b}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \underline{i} & \underline{j} & \underline{k} \end{bmatrix} \tag{1.2}$$

To test this or any other basis set for orthonormality the following conditions must be satisfied:

$$\underline{b}_i \cdot \underline{b}_j = \delta_{ij} \tag{1.3}$$

In words, (1) the dot product of any pair of distinct basis vectors is zero and (2) the dot product of a basis vector with itself is unity. If only the first condition is met the basis set is orthogonal. If both conditions are met, the basis set is orthonormal. If neither condition is met, the basis set is neither orthogonal nor normalized.

1.B. Use of a basis set

All points in space can be described as a linear combination of a sample-spanning basis set. For example, the point $\underline{a} = \begin{bmatrix} 2 & 0.5 & -3 \end{bmatrix}^T$ in three-dimensional Cartesian space can be written as

$$\underline{a} = 2 \cdot \underline{b}_1 + 0.5 \cdot \underline{b}_2 - 3 \cdot \underline{b}_3 = \underline{\underline{B}}\underline{a} \tag{1.4}$$

However frequently, we don't want to use this simple basis set for three-dimensional Cartesian space.

example:

Frequently when you obtain x-ray crystallographic data of the atomic coordinates of a crystalline material you are given the cell symmetry, the axes lengths, and the atomic coordinates in the (non-cartesian) reference frame of the crystal (defined by its axes and a (0,0,0) origin.

Consider an atom in a crystal cell with axes defined by a = 14Å, b=14Å, c=8Å. $\theta_{AB} = 120°$, $\theta_{AC} = 90°$, $\theta_{BC} = 90°$, and scaled (from 0.0 to 1.0) atomic position (in the crystal cell frame of reference) (0.25,0.5,0.25). Find the atomic position in cartesian coordinates.

Here we have a vector $\underline{x} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \end{bmatrix}^T$ but it is not based on the ordinary 3D Cartesian space basis set of equation (1.1). Rather, it is an expression a linear combination of the crystal cell axes. Given the angles above and a little experience dealing with crystal cells, any crystal cell axes can be written as (see Appendix)

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} a & b\cos\theta_{AB} & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix}$$

Is this basis set orthogonal or normalized? Check it.

$$\underline{b}_1 \cdot \underline{b}_2 = ab\cos\theta_{AB} \neq 0$$

so it is not orthogonal.

$$\underline{b}_1 \cdot \underline{b}_1 = a^2 \neq 1$$

so it is not normalized. Sometimes, the problem does not lend itself to an orthonormal basis set.

Regardless, we can first transform the unit cell axes to standard cartesian coordinates, using elementary trigonometry and algebra.

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} a & b\cos\theta_{AB} & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix}$$

See Appendix I for the derivation of the generalized method for obtaining this transformation matrix as well as for the calculation of this particular example. We superscript this matrix because it is our starting point in the transformation.

The atomic position in the crystal cell frame of reference was

$$\underline{x}^{(1)} = \begin{bmatrix} 0.25 \\ 0.5 \\ 0.25 \end{bmatrix}$$

The expression of this vector in standard 3D Cartesian space is given by equation (1.4)

$$\underline{x}^{(2)} = \underline{\underline{B}}\underline{x}^{(1)} = \begin{bmatrix} 0.25a + 0.5b\cos\theta_{AB} \\ 0.5b\sin\theta_{AB} \\ 0.25c \end{bmatrix}$$

This provides one example of how transformation matrices work. In this example, the fact that the basis set was not orthonormal didn't matter. In other transformations, particularly in rotational transformations, we see that it can be very useful.

1.C. Creating an orthonormal basis set

One process of creating an orthonormal basis set from some other set of sample-spanning basis vectors is called Gram-Schmidt Orthogonalization. There are other methods: modified Gram-Schmidt and Householder processes which are more amenable to computer applications. However, this one is nice and simple. We will stick to it.

In Gram-Schmidt Orthogonalization, we start with vectors $\underline{b}_1^{(1)}, \underline{b}_2^{(1)}, \underline{b}_3^{(1)}$ for a 3-D vector space. The superscript (1), indicates our starting point. These vector are not orthogonal. We want to create an orthogonal basis set that spans the same 3D space or subspace, $\underline{b}_1^{(2)}, \underline{b}_2^{(2)}, \underline{b}_3^{(2)}$. Let

$$\underline{b}_1^{(2)} = \underline{b}_1^{(1)} \tag{1.5}$$

$$\underline{b}_2^{(2)} = \underline{b}_2^{(1)} - \alpha_{1,2}\underline{b}_1^{(2)} \tag{1.6}$$

$$\underline{b}_3^{(2)} = \underline{b}_3^{(1)} - \alpha_{1,3}\underline{b}_1^{(2)} - \alpha_{2,3}\underline{b}_2^{(2)} \tag{1.7}$$

and in general

$$\underline{b}_i^{(2)} = \underline{b}_i^{(1)} - \alpha_{1,i}\underline{b}_1^{(2)} - \alpha_{2,i}\underline{b}_2^{(2)} - \ldots - \alpha_{i-1,i}\underline{b}_{i-1}^{(2)} \tag{1.8}$$

where

$$\alpha_{j,i} = \frac{\underline{b}_j^{(2)} \cdot \underline{b}_i^{(1)}}{\underline{b}_j^{(2)} \cdot \underline{b}_j^{(2)}} \tag{1.9}$$

These vectors are orthogonal. They are not normalized. If we want them normalized, then we have to normalize them ourselves by

$$\underline{b}_1^{(2)} = \frac{1}{\left|\underline{b}_1^{(2)}\right|}\underline{b}_1^{(2)} \tag{1.10}$$

where the magnitude is determined by

$$\left|\underline{a}\right| = \sqrt{\sum_{i=1}^{n}\left(a_i\right)^2} \tag{1.11}$$

example:

Create an orthogonal basis set from the basis vectors:

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} a & b\cos\theta_{AB} & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix}$$

$$\underline{b}_1^{(2)} = \underline{b}_1^{(1)} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \tag{1.5}$$

$$\alpha_{1,2} = \frac{\underline{b}_1^{(2)} \cdot \underline{b}_2^{(1)}}{\underline{b}_1^{(2)} \cdot \underline{b}_1^{(2)}} = \frac{ab\cos\theta_{AB}}{a^2} = \frac{b\cos\theta_{AB}}{a}$$

$$\alpha_{1,3} = \frac{\underline{b}_1^{(2)} \cdot \underline{b}_3^{(1)}}{\underline{b}_1^{(2)} \cdot \underline{b}_1^{(2)}} = \frac{0}{a^2} = 0$$

$$\alpha_{2,3} = \frac{\underline{b}_2^{(2)} \cdot \underline{b}_3^{(1)}}{\underline{b}_2^{(2)} \cdot \underline{b}_2^{(2)}} = \frac{0}{a^2} = 0$$

$$\underline{b}_2^{(2)} = \underline{b}_2^{(1)} - \alpha_{1,2}\underline{b}_1^{(2)} = \begin{bmatrix} b\cos\theta_{AB} - \dfrac{b\cos\theta_{AB}}{a}a \\ b\sin\theta_{AB} - 0 \\ 0 - 0 \end{bmatrix} = \begin{bmatrix} 0 \\ b\sin\theta_{AB} \\ 0 \end{bmatrix} \tag{1.6}$$

$$\underline{b}_3^{(2)} = \underline{b}_3^{(1)} - \alpha_{1,3}\underline{b}_1^{(2)} - \alpha_{2,3}\underline{b}_2^{(2)} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} \tag{1.7}$$

So that our new orthogonal basis set is

$$\underline{\underline{B}}^{(2)} = \begin{bmatrix} a & 0 & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix}$$

If we normalize these, we have

$$\underline{\underline{B}}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1.D. Properties of an orthogonal matrix

The most important property of an orthogonal basis set is that it's inverse is the same as its transpose.

$$\underline{\underline{B}}^{T} = \underline{\underline{B}}^{-1} \tag{1.12}$$

This is a very nice feature because calculating the inverse is a pain in the ass but calculating the transpose is trivial.

The adjoint of a matrix is defined to be the complex conjugate of the transpose.

$$\underline{\underline{\tilde{B}}} = \left(\underline{\underline{B}}^{T}\right)^{*} \tag{1.13}$$

Since most of the time, we have real matrices, the complex conjugate of a matrix is the matrix itself and the adjoint is the transpose

$$\underline{\underline{\tilde{B}}} = \left(\underline{\underline{B}}^{T}\right)^{*} = \underline{\underline{B}}^{T} \qquad \text{for real matrices} \tag{1.14}$$

In this case (a real orthogonal matrix), when $\underline{\underline{\tilde{B}}} = \underline{\underline{B}}^{T} = \underline{\underline{B}}^{-1}$, the matrix is called unitary.

2. Transforming coordinates systems

A point in space, P, is meaningful only if the relevant coordinate system is known. For whatever, sometimes one coordinate system is preferable over another. Therefore, we need to know how to change coordinate systems.

An N-dimensional coordinate system, $C^{(1)}$, is defined by an origin, $\underline{o}^{(1)}$, and by N independent N-dimensional vectors (a basis set, $\underline{\underline{B}}$).

The default three-dimensional cartesian coordinate system, $C^{(o)}$, has a basis set:

$$\underline{\underline{B}}^{(o)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and an origin

$$\underline{o}^{(o)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$\underline{\alpha}$ is a vector expressed in the default three-dimensional cartesian coordinate system, $C^{(o)}$. In general $\underline{\alpha}$ is related to any other arbitrary coordinate system as

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)}$$

This equation assumes that both origins are given with respect to $C^{(o)}$. In the case where the coordinate system is $C^{(o)}$, then

$$\underline{\alpha} = \underline{\underline{B}}^{(o)}\underline{x}^{(o)} + \underline{o}^{(o)} = \underline{x}^{(o)}$$

If we want to change from $C^{(1)}$ to $C^{(2)}$, where there is no reference to $C^{(o)}$, then the second origin, $\underline{o}^{(2)}$, is naturally expressed in the coordinates of the first coordinate system, $C^{(1)}$.

In this case, we must first transform $\underline{o}^{(2)}$ to $C^{(o)}$, in order to use the formula given above. This transformation is obtained the same way as any other point is transformed

$$\underline{o}^{(2)'} = \underline{\underline{B}}^{(1)}\underline{o}^{(2)} + \underline{o}^{(1)}$$

so that our generalized equation for transformation becomes

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)'} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{\underline{B}}^{(1)}\underline{o}^{(2)} + \underline{o}^{(1)}$$

in the case where $\underline{o}^{(2)}$, is expressed in the coordinates of the first coordinate system, $C^{(1)}$. Let us look at some specific examples below.

2.A. Changing origin only (same basis set)

If we have a point in space $\underline{x}^{(1)}$, with respect to a coordinate System $C^{(1)}$, with origin at $\underline{\underline{B}}$ and basis set $\underline{\underline{x}}^{(1)}$ and we want to change the coordinate system to System $C^{(2)}$, which has the same basis set, $\underline{\underline{B}}$, but a different origin, $\underline{o}^{(2)}$, the transformation from $C^{(1)}$ to $C^{(2)}$ is accomplished by the following:

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)} \tag{2.1a}$$

So that the representation of the point in the new coordinate system is

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{o}^{(2)}\right] = \underline{x}^{(1)} + \underline{\underline{B}}^{(2)^{-1}}\left[\underline{o}^{(1)} - \underline{o}^{(2)}\right] \tag{2.1b}$$

Again, this assumes that the $\underline{o}^{(2)}$ is given according to the same coordinate system as $\underline{o}^{(1)}$. If it is not, we must first transform $\underline{o}^{(2)}$.

example 1:

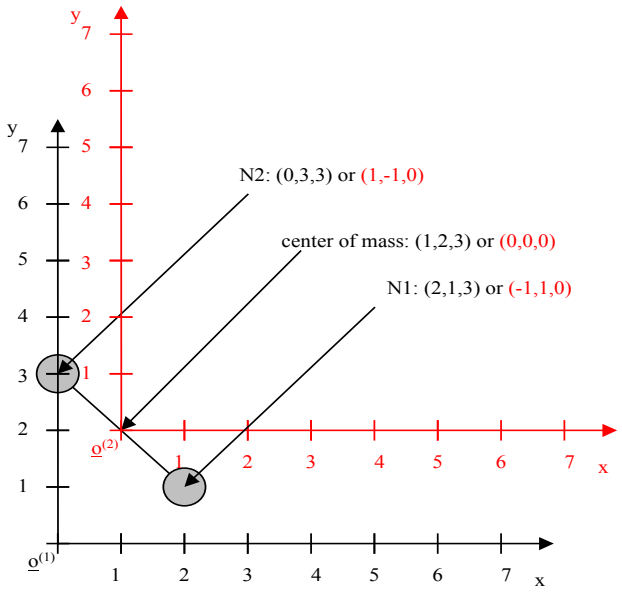
The center of a mass of a nitrogen molecule is at a point (1,2,3) in the laboratory frame of reference, which has an origin of (0,0,0) and the basis vectors given by the 3x3 identity matrix, $\underline{\underline{I}}_3$. In the center-of-mass frame of reference, the origin is at the center of mass, namely (1,2,3). The center of mass coordinate system has the same basis vectors as the lab. The nitrogen atoms are at points (1.06, -1.06, 0) (-1.06, 1.06, 0) with respect to the center of mass. Find the position of the atoms in the laboratory frame of reference. In this case $C^{(1)}$ is the molecular frame of reference and $C^{(2)}$ is the lab frame. Here the second origin is given with respect to the same coordinate system as the first origin, namely $C^{(o)}$.

$$\underline{x}_{N1}{}^{(2)} = \begin{bmatrix} 1.06 \\ -1.06 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left\{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right\} = \begin{bmatrix} 2.06 \\ 0.94 \\ 3 \end{bmatrix}$$

and

$$\underline{x}_{N2}{}^{(2)} = \begin{bmatrix} -1.06 \\ 1.06 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left\{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right\} = \begin{bmatrix} -0.06 \\ 3.06 \\ 3 \end{bmatrix}$$

We can see the graphical interpretation of this transformation in Figure 1.



x-y plane of N_2 problem

Figure 1.

example 2.

Consider a two-dimensional coordinate system with an origin at $\underline{o}^{(1)}$=(1,1) in the laboratory frame, $C^{(o)}$ and a basis set a $\underline{\underline{B}}^{(1)} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$. Consider a second coordinate system with an origin at $\underline{o}^{(2)}$=(1,1) in $C^{(1)}$ and the same basis set. We want to express the point $\underline{x}^{(1)}$=(2,2) in terms of $C^{(2)}$. In other words, find $\underline{x}^{(2)}$.

Since the second origin is given with respect to the first coordinate system, we need to account for it

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)'} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{\underline{B}}^{(1)}\underline{o}^{(2)} + \underline{o}^{(1)}$$

Rearranging for $\underline{x}^{(2)}$, we have

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)} - \underline{o}^{(1)}\right] = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)}\right]$$

Since the basis sets for $C^{(1)}$ and $C^{(2)}$ are the same, we have

$$\underline{x}^{(2)} = \underline{x}^{(1)} - \underline{o}^{(2)} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

We can see the graphical interpretation of this transformation in Figure 2. In Figure Two, $C^{(o)}$ is shown for reference purposes only. It was never used in the calculation.

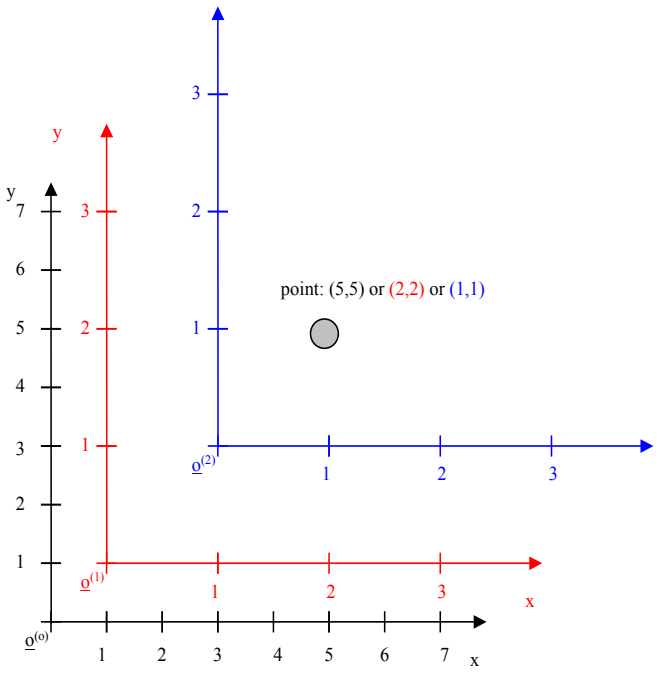


Figure 2.

2.B. Changing basis set only (same origin)

If we have a point in space $\underline{x}^{(1)}$, with respect to a coordinate System $C^{(1)}$, with origin at $\underline{o}$ and basis set $\underline{\underline{B}}^{(1)}$ and we want to change the coordinate system to System $C^{(2)}$, which has a different basis set, $\underline{\underline{B}}^{(2)}$, but the same origin, $\underline{o}$, the transformation from $C^{(1)}$ to $C^{(2)}$ is accomplished by the following:

2.B.1. express $\underline{x}^{(1)}$ as a linear combination of the column vectors (basis set) in $\underline{\underline{B}}^{(1)}$

$$\underline{\alpha} = \underline{\underline{B}}^{(1)} \underline{x}^{(1)} \tag{2.2}$$

or equivalently in scalar form:

$$\underline{\alpha} = x_1^{(1)} \underline{b}_1^{(1)} + x_2^{(1)} \underline{b}_2^{(1)} + \ldots + x_{n-1}^{(1)} \underline{b}_{n-1}^{(1)} + x_n^{(1)} \underline{b}_n^{(1)} \tag{2.3}$$

Our vector of unknowns, $\underline{\alpha}$, is just a representation of the point $\underline{x}^{(1)}$ with respect to the standard n-dimensional cartesian basis set, i.e.

$$\underline{\underline{B}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.B.2. express $\underline{x}^{(2)}$ as a linear combination of the column vectors (basis set) in $\underline{\underline{B}}^{(2)}$

$$\underline{\alpha} = \underline{\underline{B}}^{(2)} \underline{x}^{(2)} \tag{2.4}$$

equate $\underline{\alpha}$ in equation (2.4) and (2.3)

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} \tag{2.5}$$

Solve for $\underline{x}^{(2)}$

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\underline{\underline{B}}^{(1)}\underline{x}^{(1)} = \underline{\underline{M}}\underline{x}^{(1)}$$

We can see the physical meaning of the $\underline{\underline{M}}$ change-of-basis matrix if we attempt to express one basis set as a linear combination of the other basis set. Express each column vector of $\underline{\underline{B}}^{(1)}$ as a linear combination of the column vectors (basis set) in $\underline{\underline{B}}^{(2)}$

$$\underline{b}_i^{(1)} = m_{i,1}\underline{b}_1^{(2)} + m_{i,2}\underline{b}_2^{(2)} + \ldots + m_{i,n-1}\underline{b}_{n-1}^{(2)} + m_{i,n}\underline{b}_n^{(2)} \qquad \text{for i = 1 to N} \tag{2.5}$$

or in matrix notation

$$\underline{b}_i^{(1)} = \underline{\underline{B}}^{(2)}\underline{m}_i \tag{2.6}$$

We can obtain our vector of unknowns, $\underline{m}_i$

$$\underline{m}_i = \underline{\underline{B}}^{(2)^{-1}}\underline{b}_i^{(1)} \tag{2.7}$$

We have N of these equations which we can write as

$$\underline{\underline{B}}^{(1)} = \underline{\underline{B}}^{(2)}\underline{\underline{M}} \tag{2.8}$$

or as

$$\underline{\underline{M}} = \underline{\underline{B}}^{(2)^{-1}}\underline{\underline{B}}^{(1)} \tag{2.9}$$

The $\underline{\underline{M}}$ change-of-basis matrix is just a way of expressing one basis set as a linear combination of another basis set.

Before we look at an example we should see that the reverse transformation can also be acccomplished by

$$\underline{x}^{(1)} = \underline{\underline{B}}^{(1)^{-1}}\underline{\underline{B}}^{(2)}\underline{x}^{(2)}$$

Now, in equation (2.9) we have a definition for the transformation matrix $\underline{\underline{M}}$ for a change of basis from 1→2. If we take the inverse of $\underline{\underline{M}}$, we find that

$$\underline{\underline{M}}^{-1} = \left(\underline{\underline{B}}^{(2)^{-1}}\underline{\underline{B}}^{(1)}\right)^{-1} = \underline{\underline{B}}^{(1)^{-1}}\underline{\underline{B}}^{(2)}$$

so that the reverse transformation, from 2 → 1, can be carried out with the inverse transformation matrix.

$$\underline{x}^{(1)} = \underline{\underline{B}}^{(1)^{-1}} \underline{\underline{B}}^{(2)} \underline{x}^{(2)} = \underline{\underline{M}}^{-1} \underline{x}^{(2)}$$

example 1:

We have a vector with the value x =(1,0,1) in a coordinate system with basis vectors (1,0,0), (0,1,0), and (0,1,1). Express this vector in a coordinate system with basis vectors (2,0,0), (2,2,0), and (0,0,2).

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \underline{\underline{B}}^{(2)} = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \qquad \underline{\underline{B}}^{(2)^{-1}} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

$$\underline{\underline{M}} = \underline{\underline{B}}^{(2)^{-1}} \underline{\underline{B}}^{(1)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.5 & -0.5 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix}$$

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}} \underline{\underline{B}}^{(1)} \underline{x}^{(1)} = \underline{\underline{M}} \underline{x}^{(1)} = \begin{bmatrix} 0.5 & -0.5 & -0.5 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}$$

Now change back:

$$\underline{\underline{M}}^{-1} = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 2 & -2 \\ 0 & 0 & 2 \end{bmatrix}$$

$$\underline{x}^{(1)} = \underline{\underline{B}}^{(1)^{-1}} \underline{\underline{B}}^{(2)} \underline{x}^{(2)} = \underline{\underline{M}}^{-1} \underline{x}^{(2)} = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 2 & -2 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

example 2:

Consider a two-dimensional coordinate system with an origin at $\underline{o}^{(1)}$ in the laboratory frame, $C^{(o)}$ and a basis set a $\underline{\underline{B}}^{(1)} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$. Consider a second coordinate system with an origin at $\underline{o}^{(2)} = \underline{o}^{(1)}$ in $C^{(1)}$ and the basis set $\underline{\underline{B}}^{(2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We want to express the point $\underline{x}^{(1)}$=(1,1) in terms of $C^{(2)}$. In other words, find $\underline{x}^{(2)}$.

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}} \underline{\underline{B}}^{(1)} \underline{x}^{(1)} = \underline{\underline{M}} \underline{x}^{(1)}$$

$$\underline{x}^{(2)} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\underline{x}^{(1)} = \begin{bmatrix} 2 & -2 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

We can see the graphical interpretation of this transformation in Figure 3. In Figure 3, $C^{(o)}$ is shown for reference purposes only. It was never used in the calculation.

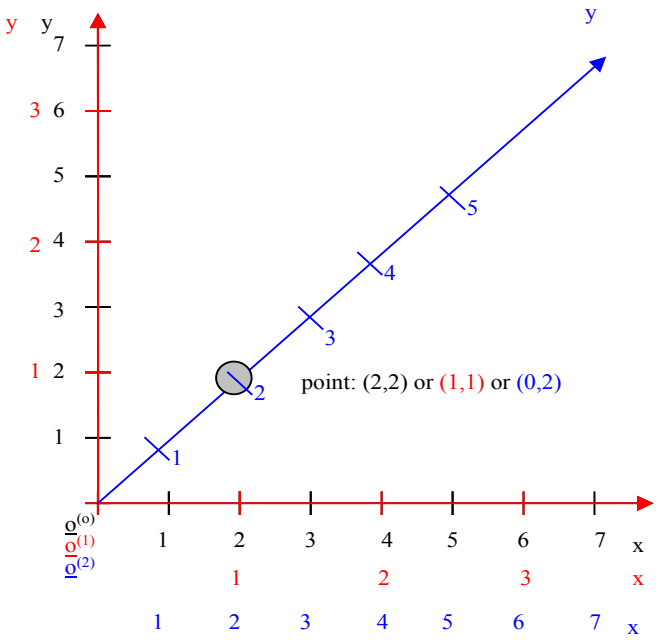


Figure 3.

2.C. Changing basis set and origin simultaneously

If we have a point in space $\underline{x}^{(1)}$, with respect to a coordinate System $C^{(1)}$, with origin at $\underline{o}^{(1)}$ and basis set $\underline{\underline{B}}^{(1)}$ and we want to change the coordinate system to System $C^{(2)}$, which has a different basis set, $\underline{\underline{B}}^{(2)}$ and a different origin, $\underline{o}^{(2)}$, the transformation from $C^{(1)}$ to $C^{(2)}$ is accomplished by the following:

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)} \tag{2.10}$$

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{o}^{(2)}\right] \tag{2.11}$$

Once again, this assumes that the $\underline{o}^{(2)}$ is given according to the same coordinate system as $\underline{o}^{(1)}$. If it is not, (if $\underline{o}^{(2)}$ is given in terms of $C^{(1)}$) we must first transform $\underline{o}^{(2)}$.

$$\underline{\alpha} = \underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{o}^{(2)'} = \underline{\underline{B}}^{(2)}\underline{x}^{(2)} + \underline{\underline{B}}^{(1)}\underline{o}^{(2)} + \underline{o}^{(1)}$$

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)} - \underline{o}^{(1)}\right] = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)}\right]$$

example 1:

Consider a two-dimensional coordinate system with an origin at $\underline{o}^{(1)} = (1,1)$ in the laboratory frame, $C^{(o)}$ and a basis set a $\underline{\underline{B}}^{(1)} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$. Consider a second coordinate system with an origin at $\underline{o}^{(2)} = (2,2)$ in $C^{(1)}$ and the basis set $\underline{\underline{B}}^{(2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We want to express the point $\underline{x}^{(1)}$=(1,1) in terms of $C^{(2)}$. In other words, find $\underline{x}^{(2)}$.

Since $\underline{o}^{(2)}$ is given in terms of $C^{(1)}$, we must take that into account.

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)} - \underline{o}^{(1)}\right] = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} - \underline{\underline{B}}^{(1)}\underline{o}^{(2)}\right]$$

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\underline{\underline{B}}^{(1)}\left[\underline{x}^{(1)} - \underline{o}^{(2)}\right] = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right\} = \begin{bmatrix} 2 & -2 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

We can see the graphical interpretation of this transformation in Figure 4. In Figure 4, $C^{(o)}$ is shown for reference purposes only. It was never used in the calculation.

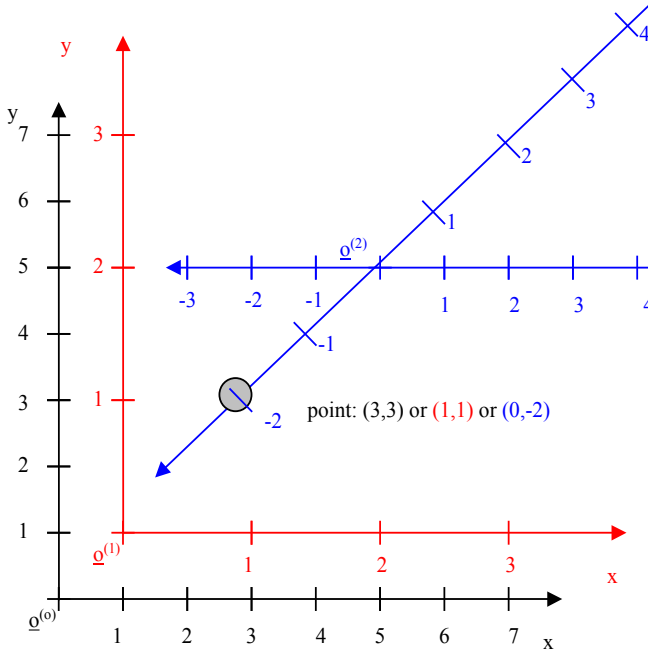


Figure 4.

example 2:

Consider example 1 again, but this time, assume that $\underline{o}^{(2)}$ is given in terms of the same coordinate system as $\underline{o}^{(1)}$, namely $C^{(0)}$. In this case,

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{o}^{(2)}\right]$$

$$\underline{x}^{(2)} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We can see the graphical interpretation of this transformation in Figure 5. In Figure 5, $C^{(o)}$ is shown for reference purposes only. It was never used in the calculation.

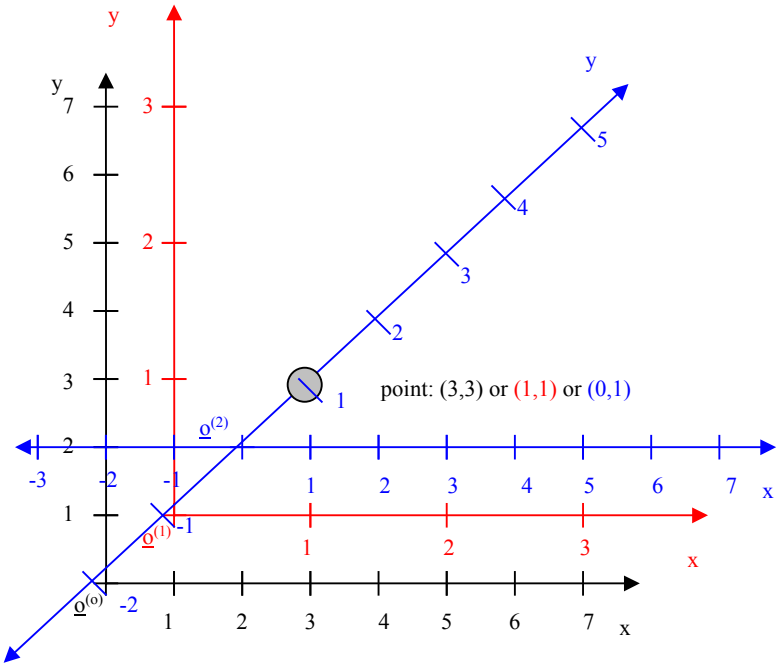


Figure 5.

example 3:

Frequently when you obtain x-ray crystallographic data of the atomic coordinates of a crystalline material you are given the cell symmetry, the axes lengths, and the atomic coordinates in the (non-cartesian) reference frame of the crystal (defined by its axes and a (0,0,0) origin.

Consider an atom in a crystal cell with axes defined by a = 14Å, b=14Å, c=8Å. $\theta_{AB} = 120°$, $\theta_{AC} = 90°$, $\theta_{BC} = 90°$, and scaled (from 0.0 to 1.0) atomic position (in the crystal cell frame of reference) (0.25,0.5,0.25). Find the position in cartesian coordinates with an origin at the center of the unit cell.

We did this already but now we are going to do it again, within the general formalism of transformations. We can first transform the unit cell axes to standard cartesian coordinates, using elementary trigonometry and geometry.

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} a & b\cos\theta_{AB} & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix}, \quad \underline{\underline{B}}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \underline{\underline{B}}^{(2)^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{x}^{(1)} = \begin{bmatrix} 0.25 \\ 0.5 \\ 0.25 \end{bmatrix}, \quad \underline{o}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{o}^{(2)} = ? \ , \ \underline{x}^{(2)} = ?$$

In order to obtain $\underline{o}^{(2)}$, the origin of our final reference frame, we can realize that the center of the cell is easiest to find in the crystal cell coordinates, where is it (0.5, 0.5, 0.5). This representation of $\underline{o}^{(2)}$ is with respect to $C^{(1)}$. We can either use the formula given above for this case or we can transform $\underline{o}^{(2)}$ to the same coordinates system in which $\underline{o}^{(1)}$ is defined, using the same transformation given in equation (2.11) where the origin of this transformation is (0,0,0)

$$\underline{x}_{center} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}_{center}{}^{(1)} + \underline{o}^{(1)} - \underline{o}\right] = \begin{bmatrix} 0.5a + 0.5b\cos\theta_{AB} \\ 0.5b\sin\theta_{AB} \\ 0.5c \end{bmatrix}$$

$$\underline{o}^{(2)} = \underline{x}_{center} = \begin{bmatrix} 0.5a + 0.5b\cos\theta_{AB} \\ 0.5b\sin\theta_{AB} \\ 0.5c \end{bmatrix}$$

Now that we have the center, $\underline{o}^{(2)}$, in the same coordinate system as $\underline{o}^{(1)}$, we can transform the point of interest using equation (2.11)

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{o}^{(2)}\right] \tag{2.11}$$

$$\underline{x}^{(2)} = \begin{bmatrix} 0.25a + 0.5b\cos\theta_{AB} \\ 0.5b\sin\theta_{AB} \\ 0.25c \end{bmatrix} - \begin{bmatrix} 0.5a + 0.5b\cos\theta_{AB} \\ 0.5b\sin\theta_{AB} \\ 0.5c \end{bmatrix} = \begin{bmatrix} -0.25a \\ 0 \\ -0.25c \end{bmatrix} = \begin{bmatrix} -3.5 \\ 0 \\ -6 \end{bmatrix}$$

3. Rotational Transformations

3.A. General introduction to rotational matrices

Any rotation in space can be accomplished with a rotational matrix. The trouble is coming up with the form of the rotational matrix.

In 3-D space, rotations about the x, y, and z-axes are given by, respectively,

$$\underline{\underline{A}}_{X-ROT} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.1}$$

$$\underline{\underline{A}}_{Y-ROT} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \tag{3.2}$$

$$\underline{\underline{A}}_{Z-ROT} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The problem is that frequently the rotation is more complicated than the a rotation about one of these axes. One methodical approach for arbitrary rotation in three dimensions is called the Euler angles. I am going to give a quick and dirty summary of the Euler Angles. For an elegant discussion and derivation of the rotational transformation matrix and the Euler Angles, refer to "Classical Mechanics" by Herbert Goldstein, Second Ed., Addison Wesley, Reading, Massachusetts, 1980, pp. 137-148, especially pp 146-147.

The Euler Angles arise from

(1) a rotation of ϕ about the z-axis, as given by equation (3.3)
(2) a rotation of θ about the newly rotated x-axis, as given by equation (3.2)
(3) a rotation of ψ about the newly rotated z-axis, as given by equation (3.3)

The net transformation is

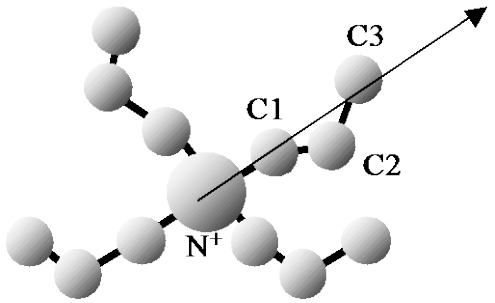
$$\underline{\underline{A}} = \underline{\underline{A}}_{Z-ROT}(\psi)\underline{\underline{A}}_{X-ROT}(\theta)\underline{\underline{A}}_{Z-ROT}(\phi) \tag{3.4}$$

which explicitly is (where s=sin and c=cos)

$$\underline{\underline{A}} = \begin{bmatrix} c\psi c\phi - c\theta s\phi s\psi & c\psi s\phi + c\theta s\phi s\psi & s\theta s\psi \\ -s\psi c\phi - c\theta s\phi c\psi & -s\psi s\phi + c\theta c\phi c\psi & s\theta c\psi \\ s\theta s\phi & -s\theta c\phi & c\theta \end{bmatrix} \tag{3.5}$$

One important property of all rotational matrices is that they are orthogonal matrices. The transpose is the inverse.

3.B. Rotation of atoms about bonds in polyatomic molecules



tetrapropyl ammonium cation

example:

Let's say that you want to find the minimum-energy configuration of a molecule, for example, the tetrapropyl ammonium ion. You are going to locate this minimum by making different sorts of translations and rotations of individual atoms and groups of atoms. One such move is the rotation of a propyl arm.

How would we accomplish this task, given that we know (i) the positions of all of the atoms in the laboratory frame of reference and (ii) the angle of rotation, θ?

First, we must realize that the axis of rotation is the bond that connects N and C1. So C2 and C3 and all of the hydrogen atoms attached to C1, C2, and C3 will change positions with this rotation.

STEP 1. Move Origin of system to N atom

For all atoms of interest, N, C1, C2, C3 and relevant hydgrogen, change coordinate systems from a system with the origin at $\underline{o}^{(1)} = (0,0,0)^T$ to $\underline{o}^{(2)}$ equal to the position of the nitrogen.

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}}\left[\underline{\underline{B}}^{(1)}\underline{x}^{(1)} + \underline{o}^{(1)} - \underline{o}^{(2)}\right] = \underline{x}^{(1)} + \underline{\underline{B}}^{(2)^{-1}}\left[\underline{o}^{(1)} - \underline{o}^{(2)}\right] \tag{2.1b}$$

Since $\underline{\underline{B}}^{(2)}$ is the lab frame of reference, which is the default 3-D cartesian coordinate system, it is the identity matrix so that

$$\underline{x}^{(2)} = \underline{x}^{(1)} + \left[\underline{o}^{(1)} - \underline{o}^{(2)}\right]$$

STEP 2. Change the basis set from cartesian lab frame to appropriate basis set

This appropriate basis set corresponds to one vector parallel to the N-C1 bond and two other vectors which form a plane perpendicular to the N-C1 bond. Of the nine elements in this transformation matrix, only six are independent.

To obtain the basis vector parallel to the N-C1 bond, just normalize the distance vector between N and C1.

$$\left|\underline{r}_{C1-N}\right| = \sqrt{(x_{C1} - x_N)^2 + (y_{C1} - y_N)^2 + (z_{C1} - z_N)^2} \tag{3.?}$$

$$\underline{b}_z{}^{(2)} = \left[\frac{x_{C1} - x_N}{\left|\underline{r}_{C1-N}\right|} \quad \frac{y_{C1} - y_N}{\left|\underline{r}_{C1-N}\right|} \quad \frac{z_{C1} - z_N}{\left|\underline{r}_{C1-N}\right|}\right]^T \tag{3.?}$$

To obtain the second new basis vector, realize that it must be perpendicular to the basis vector $\underline{b}_z^{(2)}$. This is the only constraint on this vector. Otherwise, the particular values of $\underline{b}_x^{(2)}$ are arbitrary. In fact you can select any two of them arbitrarily.

$$\underline{b}_x^{(2)} \cdot \underline{b}_z^{(2)} = b_{x,1}^{(2)} \cdot b_{z,1}^{(2)} + b_{x,2}^{(2)} \cdot b_{z,2}^{(2)} + b_{x,3}^{(2)} \cdot b_{z,3}^{(2)} = 0 \tag{3.?}$$

One choice is to set $b_{x,3}^{(2)} = 0$ and

$$|\underline{s}| = \sqrt{b_{z,1}^{(2)^2} + b_{z,2}^{(2)^2}} \tag{3.?}$$

$$b_{x,1}^{(2)} = -\frac{b_{z,2}^{(2)}}{|\underline{s}|} \tag{3.?}$$

which forces

$$b_{x,2}^{(2)} = \frac{b_{z,1}^{(2)}}{|\underline{s}|} \tag{3.?}$$

$$b_x^{(2)} = \begin{bmatrix} -\frac{b_{z,2}^{(2)}}{|\underline{s}|} & \frac{b_{z,1}^{(2)}}{|\underline{s}|} & 0 \end{bmatrix}^T \tag{3.?}$$

We can see that this was a very nice choice because $\underline{b}_x^{(2)}$ is now normalized and if you take the dot product of $\underline{b}_x^{(2)}$ and $\underline{b}_z^{(2)}$, you will see they are orthogonal. This is a bit of a trick, but the same old trick works every time. In the second vector, zero the element that corresponds to the first vector, then switch the other two elements from the first vector, normalize, and switch one of the signs.

To obtain the final basis vector, we have one remaining degree of freedom, if we choose to make this basis set orthonormal. We have two constraints, namely that the dot product of the remaining basis vector be zero with each of the two already determined basis vectors (because they are perpendicular to each other).

$$\underline{b}_y^{(2)} \cdot \underline{b}_z^{(2)} = 0 \tag{3.?}$$

$$\underline{b}_y^{(2)} \cdot \underline{b}_x^{(2)} = 0 \tag{3.?}$$

We have two equations and three unknowns. We can pick one of the elements of $\underline{b}_y^{(2)}$ to specify the other two. One choice that guarantees a perpendicular solution is

$$\underline{b}_y^{(2)} = \underline{b}_z^{(2)} \times \underline{b}_x^{(2)} \tag{3.?}$$

$$\underline{b}_y^{(2)} = \begin{bmatrix} \underline{b}_{z,2}^{(2)} \underline{b}_{x,3}^{(2)} - \underline{b}_{z,3}^{(2)} \underline{b}_{x,2}^{(2)} \\ \underline{b}_{z,3}^{(2)} \underline{b}_{x,1}^{(2)} - \underline{b}_{z,1}^{(2)} \underline{b}_{x,3}^{(2)} \\ \underline{b}_{z,1}^{(2)} \underline{b}_{x,2}^{(2)} - \underline{b}_{z,2}^{(2)} \underline{b}_{x,1}^{(2)} \end{bmatrix} \tag{3.?}$$

This cross-product yields a vector of magnitude unity because $\underline{b}_x^{(2)}$ and $\underline{b}_z^{(2)}$ were already normalized.

STEP 3. Implement the change of basis transformation

At this point we have the new basis set and we can transform the atomic positions of interest to the new frame.

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^{-1}} \underline{\underline{B}}^{(1)} \underline{x}^{(1)} = \underline{\underline{M}} \underline{x}^{(1)} \tag{2.?}$$

Well, $\underline{\underline{B}}^{(1)} = \underline{\underline{I}}$ and, because $\underline{\underline{B}}^{(2)}$ is a matrix of orthonormal basis vectors, $\underline{\underline{B}}^{(2)^{-1}} = \underline{\underline{B}}^{(2)^T}$ so

$$\underline{x}^{(2)} = \underline{\underline{B}}^{(2)^T} \underline{x}^{(1)} \tag{3.?}$$

STEP 4. Create Rotation Matrix

$$\underline{\underline{A}}_{Z-ROT} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.?}$$

STEP 5. Apply Rotation Matrix

$$\underline{x}^{(2)} = \underline{\underline{A}}_{Z-ROT} \underline{x}^{(2)} \tag{3.?}$$

STEP 6. Unimplement the Change-of basis Transformation Matrix

$$\underline{x}^{(1)} = \underline{\underline{B}}^{(1)^{-1}} \underline{\underline{B}}^{(2)} \underline{x}^{(2)} = \underline{\underline{B}}^{(2)} \underline{x}^{(2)} \tag{3.?}$$

STEP 7. Now shift the origin back to the lab frame of reference

$$\underline{x}^{(1)} = \underline{x}^{(2)} + \underline{o}^{(2)} - \underline{o}^{(1)} \tag{3.?}$$

These seven steps comprise the complete algorithm for rotation of a portion of a molecule about an arbitrary bond. It is an example of how three transformations (change of origin, change of basis set, and rotation) are applied in series to effect an arbitrary rotation.

4. Similarity Transformations

An in-depth discussion of the similarity transformations is given in your text "Advanced Engineering Mathematics" by Kreyzig. In the 8th Edition, it appears in Section 7.5, starting on page 392. This material here is only a brief summary.

If the matrix $\underline{\underline{A}}$ can be expressed as

$$\underline{\underline{A}}^{(2)} = \underline{\underline{P}}^{-1}\underline{\underline{A}}^{(1)}\underline{\underline{P}} \tag{4.1}$$

then this transformation is a similarity transformation. $\underline{\underline{A}}^{(1)}$ and $\underline{\underline{A}}^{(2)}$ will have the same eigenvalues.

A special case of this transformation occurs when $\underline{\underline{P}} = \underline{\underline{W}}_c$, the matrix of normalized eigenvectors of $\underline{\underline{A}}^{(1)}$.

$$\underline{\underline{W}}_c = \left[\underline{w}_{c,1}, \cdots, \underline{w}_{c,n}\right] \tag{4.2}$$

The result is that $\underline{\underline{A}}^{(2)} = \underline{\underline{\Lambda}}$ where $\underline{\underline{\Lambda}}$ is the diagonal matrix of eigenvalues

$$\underline{\underline{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \tag{4.4}$$

so that one could write

$$\underline{\underline{\Lambda}} = \underline{\underline{W}}_c^{-1}\underline{\underline{A}}\underline{\underline{W}}_c \tag{4.5}$$

or

$$\underline{\underline{A}} = \underline{\underline{W}}_c\underline{\underline{\Lambda}}\underline{\underline{W}}_c^{-1} \tag{4.6}$$

When the similarity matrix produces a diagonal matrix of eigenvalues, this is called diagonalizing the matrix. Equation (4.6) is necessary in deriving the solution to a system of linear first-order ordinary differential equations.

A demonstration of this similarity transformation in Matlab is shown below. In this example, we take an arbitrary 3x3 real matrix A, compute its eigenvalues and eigenvectors. We also compute the inverse of the eigenvector matrix. We then show that the matrix B generated by similarity transformation has the same eigenvectors and eigenvalues as A.

```
>> A = [1 2 3; 4 5 6; 7 8 10]

A =
     1     2     3
     4     5     6
     7     8    10

>> [w,lambda] = eig(A)

w =
   -0.2235   -0.8658    0.2783
   -0.5039    0.0857   -0.8318
   -0.8343    0.4929    0.4802

lambda =
   16.7075         0         0
         0   -0.9057         0
         0         0    0.1982

>> winv = inv(w)

winv =
   -0.4697   -0.5757   -0.7250
   -0.9745   -0.1300    0.3396
    0.1842   -0.8668    0.4742

>> A_check = w*lambda*winv

A_check =
    1.0000    2.0000    3.0000
    4.0000    5.0000    6.0000
    7.0000    8.0000   10.0000

>> diff1 = norm((A-A_check),2)

diff1 =   3.7151e-15
```

A second demonstration of this similarity transformation in Matlab is shown below. In this example, we take an arbitrary 3x3 real matrix A and an arbitrary 3x3 matrix real P. We also compute the inverse of P. We then show that the matrix A can be reconstituted in terms of the similarity transformation.

```
>> A = [1 2 3; 4 5 6; 7 8 10]

A =
     1     2     3
     4     5     6
     7     8    10

>> P = [1 2 7; 6 5 4; 9 8 10]

P =
     1     2     7
     6     5     4
     9     8    10

>> Pinv = inv(P)

Pinv =
   -2.0000   -4.0000    3.0000
    2.6667    5.8889   -4.2222
   -0.3333   -1.1111    0.7778

>> B = Pinv*A*P

B =
    3.0000    6.0000   21.0000
   12.6667    7.2222   -8.2222
    1.6667    2.2222    5.7778
```

```
>> [wA,lambdaA] = eig(A)

wA =
   -0.2235   -0.8658    0.2783
   -0.5039    0.0857   -0.8318
   -0.8343    0.4929    0.4802

lambdaA =
   16.7075         0         0
         0   -0.9057         0
         0         0    0.1982

>> [wB,lambdaB] = eig(B)

wB =
   -0.6779   -0.5896    0.5557
   -0.6934    0.7989   -0.8160
   -0.2443   -0.1186    0.1590

lambdaB =
   16.7075         0         0
         0   -0.9057         0
         0         0    0.1982


>> difflambda =
norm((lambdaA-lambdaB),2)

difflambda =     8.5265e-14
```

5. Permutations

5.A. Introduction to Permutations

A permutation is an arrangement of all or part of a set of objects. A permutation is a grouping of elements arranged in a particular way. For example, how many ways can you order the letters A, B & C? The six permutations are ABC, ACB, BAC, BCA, CAB and CBA. All sequences contain the same letters but in different orders. The key concept in permutations is that 'order matters'.

The number of permutations of n distinct objects is $n!$ That is read as "n factorial".

$$n! = n(n-1)(n-2)(n-3)...3 \cdot 2 \cdot 1 \tag{5.1}$$

The factorial only applies to non-negative integers. By definition, the factorial of zero is 1,

$$0! \equiv 1 \tag{5.2}$$

The number of permutations of n distinct objects taken r at a time, where $r \leq n$, is

$${}_nP_r = \frac{n!}{(n-r)!} \tag{5.3}$$

For the example above, where we ordered three letters, n = 3 and r = 3, so the result is

$${}_3P_3 = \frac{3!}{(3-3)!} = \frac{3!}{0!} = \frac{3 \cdot 2 \cdot 1}{1} = 6$$

For the example above, where we ordered two of the letters, n = 3 and r = 2, so the result is

$${}_3P_2 = \frac{3!}{(3-2)!} = \frac{3!}{1!} = \frac{3 \cdot 2 \cdot 1}{1} = 6$$

The six permutations are AB, AC, BA, BC, CA and CB.

The number of permutations in which some of the objects are indistinguishable is given by the general formula for permutations of multisets, in which you have m_1 values of type 1, m_2 values of type 2, etc., where $\sum_{i=1}^{\ell} m_i = n$

$$\binom{n}{m_1, m_2, m_3, \ldots m_\ell} = \frac{n!}{\prod_{i=1}^{\ell} m_i!} \tag{5.4}$$

So for example the number of permutations of (A, A, A, B, B, and C) is

$$\binom{6}{3,2,1} = \frac{n!}{\prod_{i=1}^{\ell} m_i!} = \frac{6!}{3!2!1!} = 60$$

5.B. Permutations and Linear Algebra

Permutations are relevant in linear algebra. Consider the system of linear algebraic equations,

$$a_{11}x_1 + a_{12}x_2 = b_1$$
$$a_{21}x_1 + a_{22}x_2 = b_2$$

so that

$$\underline{\underline{A}} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

consider the permutation matrix,

$$\underline{\underline{P}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

then operation of the permutation matrix on $\underline{\underline{A}}$ as shown below

$$\underline{\underline{Q}} = \underline{\underline{P}}\,\underline{\underline{A}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{21} & a_{22} \\ a_{11} & a_{12} \end{bmatrix}$$

generates a matrix $\underline{\underline{Q}}$ in which only the order of the rows of $\underline{\underline{A}}$ has changed.

If we consider a vector

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

then operation of the permutation matrix on the $\underline{x}$ as shown below

$$\underline{y} = \underline{\underline{P}}\,\underline{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}$$

generates a vector $\underline{y}$ in which only the order of the rows of $\underline{x}$ has changed. Here we have presented a simple 2x2 example, in which there are only ${}_2P_2 = 2$ permutations, with permutation matrices given by $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

For the 3x3 case, there are ${}_3P_3 = 6$ permutation matrices:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

If we equate, $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$ to the set ABC, then the six permutations matrices given above result respectively in the following 6 permutations, ABC, ACB, BAC, BCA, CAB and CBA.

By induction, the permutation matrix can be generalized to any nxn system. The permutation matrix is an orthonormal matrix. It is orthogonal and normalized. Thus the inverse of a permutation matrix is its transpose.

The concept of a permutation matrix can also be extended to non-square matrices, nxm. Consider the 3x4 permutation matrix,

$$\underline{\underline{P}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

The operation of the permutation matrix on the $\underline{x}$ as shown below

$$\underline{y} = \underline{\underline{P}}\underline{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \\ x_3 \\ x_1 \end{bmatrix}$$

Here we have left a second slot empty.

For an nxm, the number of such permutations can be expressed in terms of the multiset where we have m distinct objects and ($n-m$) indistinct (empty) objects. So we have $k_1 = k_2 = k_3 = ... = k_m = 1$ and $k_{m+1} = n - m$

$$\binom{n}{k_1, k_2, k_3, \ldots k_{m+1}} = \frac{n!}{\prod_{i=1}^{m+1} k_i!} = \frac{n!}{k_{m+1}!} = \frac{n!}{(n-m)!} \tag{5.5}$$

For example, if $n = 4$ and $m = 3$, then there are $\binom{4}{1,1,1,1} = 24$ permutations. For example, if $n = 5$ and $m = 3$, then there are $\binom{5}{1,1,1,2} = 60$ permutations.

Dealing with permutations in this way are useful. Consider the classical "non-square assignment problem". (The "square assignment problem" is a subset of this problem in which $n = m$.) You have m workers to fill n jobs. The cost associated with each worker in each job is given by the cost matrix, $\underline{\underline{C}}$, an mxn. The element, c_{ij}, is the cost associated with putting worker *i* in job *j*. The cost per worker, $\underline{f}$, an $mx1$ vector, for one arrangement of workers, $\underline{y}$, is given by

$$\underline{f} = \underline{\underline{C}}\,\underline{y}$$

The total cost is the L^1 norm of this vector, where the L^p norm is defined as $L^p \equiv \left(\sum_{i=1}^{m} |f_i|^p \right)^{\frac{1}{p}}$.

The optimal distribution that results in the minimum cost is a permutation problem. If we make an initial guess $\underline{x}_1$, corresponding to $\underline{x}_1^{\,T} = [1,2,3..m]$, then we can write the cost per job as

$$\underline{f} = \underline{\underline{C}}\,\underline{\underline{P}}\,\underline{x}_1$$

where $\underline{\underline{P}}$ is the nxm permutation matrix.

The solution to this linear optimization problem is called the "Hungarian Algorithm" or sometimes the "Kuhn-Munkres" algorithm. The original reference is:

Kuhn, H.W., The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly, 1955. 2: p. 83–97.

There is a description of it here: https://en.wikipedia.org/wiki/Hungarian_algorithm.

A routine in MATLAB, munkres.m, implements this optimization. The file is available at https://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems--v2-3-

Using this code, on a 2x2 sample cost matrix is shown below.

```
>> costmat = [1 4; 2 1];
>>  [ASSIGN,COST] = munkres(costmat)

ASSIGN =     1     2
COST =     2
```

Using this code, on a 2x3 sample cost matrix is shown below.

```
>>  costmat23 = [1 4; 5 5; 2 1];
>> [ASSIGN,COST] = munkres(costmat23)

ASSIGN =     1     0     2
COST =     2
```

In this output, the permutation matrix is collapsed into a vector named assign. For non-zero values elements of the assign vector, the permutation matrix corresponding to element (i,assign(i)) is set to 1.

Thus the assign matrix of [1,2] corresponds to $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and the assign matrix of [1,0,2] corresponds to a permutation matrix $\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$ so that $\underline{y} = \underline{\underline{P}}\,\underline{x}_1$ where $\underline{x}_1^{\,T} = [1,2,3..m]$ and $\underline{y}$ is the assign vector.

6. Singular Value Decomposition

6.A. Introduction to Singular Value Decomposition

Singular Value Decomposition (SVD) is a factorization of a real or complex matrix.

Suppose $\underline{\underline{M}}$ is an mxn matrix, real or complex. The singular value decomposition of $\underline{\underline{M}}$ is of the form

$$\underline{\underline{M}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^* \tag{6.1}$$

where

$\underline{\underline{U}}$ is an mxm unitary matrix.

$\underline{\underline{\Sigma}}$ is a diagonal mxm matrix with non-negative real numbers on the diagonal, and

$\underline{\underline{V}}^*$ is an nxn unitary matrix.

If complex, a unitary matrix satisfies the relationship: the conjugate transpose is the inverse.
If real, a unitary matrix satisfies the relationship: the transpose is the inverse. This means that the matrix is orthonormal.

The diagonal entries, σ_i, $\underline{\underline{\Sigma}}$ are known as the singular values of $\underline{\underline{M}}$.

The singular value decomposition can be performed based on the following observations.

$\underline{\underline{U}}$ contains the set of orthonormal eigenvectors of $\underline{\underline{M}}\,\underline{\underline{M}}^*$.

$\underline{\underline{V}}^*$ contains the set of orthonormal eigenvectors of $\underline{\underline{M}}^*\,\underline{\underline{M}}$.

$\underline{\underline{\Sigma}}$ contains the square roots of the non-zero eigenvalues of both $\underline{\underline{M}}\,\underline{\underline{M}}^*$ and $\underline{\underline{M}}^*\,\underline{\underline{M}}$.

In the example below, we demonstrate the singular value decomposition and the various observations given above using Matlab. The script is provided first. Then the output from Matlab is given. We should note that the order of the singular values in $\underline{\underline{\Sigma}}$ obtained from the svd command may be different than the order of the eigenvalues obtained from the eig command. Therefore, we must resort the eigenvalues and the corresponding eigenvectors. This resorting uses the Hungarian algorithm for optimizing the cost as described in the previous section. The cost matrix is simply the square of the difference between the singular values and the eigenvalues.

script: svd_script.m

```
clear all;
close all;
format short

% create a matrix
A = [ 1 2 3; 4 5 6; 7 8 10]
% perform svd
[U,S,V] = svd(A)
% confirm validity of svd
A_check = U*S*V'
diff_A = norm((A-A_check),2)
% compute eigenanalyis of AAt
AAt = A*A'
[WAAt,LAAt] = eig(AAt)
% compute eigenanalyis of AtA
AtA = A'*A
[WAtA,LAtA] = eig(AtA)
S_check = sqrt(LAAt)
%
%  To compute the norms between matrices
%  requires that the eigenvalues and eigenvectors
%  be listed in the proper order.
%  To find this order, we will use the
%  Hungarian algorithm.
%
n = max(size(A));
indx = zeros(n,1);
costmatrix = zeros(n,n);
for i = 1:1:n
    for j = 1:1:n
        costmatrix(i,j) = (S(i,i) - S_check(j,j) )^2;
    end
end
costmatrix
[indx,cost] = munkres(costmatrix)
%
%  resort eigenvalue matrix to compare with S
%
for i = 1:1:n
    ii = indx(i);
    S_check_sorted(i,i)  = S_check(ii,ii);
end
S_check_sorted
diff_L = norm((LAAt-LAtA),2)
diff_S = norm((S-S_check_sorted),2)
%
%  resort eigenvectors of AAt to compare with U
%
for i = 1:1:n
    for j = 1:1:n
       ii = indx(i);
       U_check(j,i)  = WAAt(j,ii);
    end
end
```

```
%
%  also get correct sign of eigenvectors of AAt
%
eps = 1.0e-8;
for i = 1:1:n
    delta_U = U(1,i) - U_check(1,i);
    if ( abs(delta_U) > eps)
       U_check(1:n,i)  = -U_check(1:n,i);
    end
end
U_check
diff_U = norm((U-U_check),2)
%
%  resort eigenvectors of AtA to compare with V
%
for i = 1:1:n
    for j = 1:1:n
       ii = indx(i);
       V_check(j,i)  = WAtA(j,ii);
    end
end
%
%  also get correct sign of eigenvectors of AtA
%
eps = 1.0e-8;
for i = 1:1:n
    delta_V = V(1,i) - V_check(1,i);
    if ( abs(delta_V) > eps)
       V_check(1:n,i)  = -V_check(1:n,i);
    end
end
V_check
diff_V = norm((V-V_check),2)
```

output of svd_script.m

```
>> svd_script

A =
    19     2     3
     4     5     6
     7     8    10

U =
   -0.7743    0.6327   -0.0136
   -0.3223   -0.4127   -0.8519
   -0.5447   -0.6552    0.5235

S =
   23.3067         0         0
         0   10.9901         0
         0         0    0.1288

V =
   -0.8501    0.5263   -0.0195
   -0.3225   -0.5496   -0.7707
   -0.4163   -0.6488    0.6369

A_check =
   19.0000    2.0000    3.0000
    4.0000    5.0000    6.0000
    7.0000    8.0000   10.0000

diff_A =   6.2456e-15

AAt =
   374   104   179
   104    77   128
   179   128   213

WAAt =
    0.0136   -0.6327    0.7743
    0.8519    0.4127    0.3223
   -0.5235    0.6552    0.5447

LAAt =
    0.0166         0         0
         0  120.7821         0
         0         0  543.2013

AtA =
   426   114   151
   114    93   116
   151   116   145

WAtA =
    0.0195   -0.5263    0.8501
    0.7707    0.5496    0.3225
   -0.6369    0.6488    0.4163

LAtA =
    0.0166         0         0
         0  120.7821         0
         0         0  543.2013

S_check =
    0.1288         0         0
         0   10.9901         0
         0         0   23.3067

costmatrix =

  537.2125  151.6984    0.0000
  117.9669    0.0000  151.6984
    0.0000  117.9669  537.2125

indx =
     3     2     1

cost =   4.3153e-26

S_check_sorted =
   23.3067         0         0
         0   10.9901         0
         0         0    0.1288

diff_L =   1.1369e-13

diff_S =   2.0769e-13

U_check =
   -0.7743    0.6327   -0.0136
   -0.3223   -0.4127   -0.8519
   -0.5447   -0.6552    0.5235

diff_U =   6.3485e-16

V_check =
   -0.8501    0.5263   -0.0195
   -0.3225   -0.5496   -0.7707
   -0.4163   -0.6488    0.6369

diff_V =   4.1483e-16
```

6.B. Singular Value Decomposition: Application to Optimal Rotation Matrices

The Kabsch algorithm is a method for calculating the optimal rotation matrix that minimizes the root mean squared deviation between two paired sets of points. It is a kind of point set registration involving only rotation.

Consider a set of n points in d dimensions, $\underline{\underline{P}}$, which is a dxn matrix. Each row corresponds to a dimension, e.g. x, y and z in 3-D and each column corresponds to an individual point. Also consider a second set of points given by $\underline{\underline{Q}}$, which is also a dxn matrix.

The cross-covariance matrix $\underline{\underline{A}}$, a dxd matrix is given by

$$\underline{\underline{A}} = \underline{\underline{P}}\underline{\underline{Q}}^{'} \tag{6.2}$$

The SVD of A can be expressed as

$$\underline{\underline{A}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^{*} \tag{6.3}$$

The rotation matrix to rotate $\underline{\underline{P}}$ into $\underline{\underline{Q}}$ is given by

$$\underline{\underline{M}}_{P\rightarrow Q} = \underline{\underline{V}}\,\underline{\underline{U}}^{'} \tag{6.4}$$

The rotation matrix to rotate $\underline{\underline{Q}}$ into $\underline{\underline{P}}$ is given by the inverse rotation matrix (or the transpose) since this rotation matrix is real and orthonormal.

$$\underline{\underline{M}}_{Q\rightarrow P} = \underline{\underline{M}}_{Q\rightarrow P}^{'} = \underline{\underline{U}}\,\underline{\underline{V}}^{'} \tag{6.5}$$

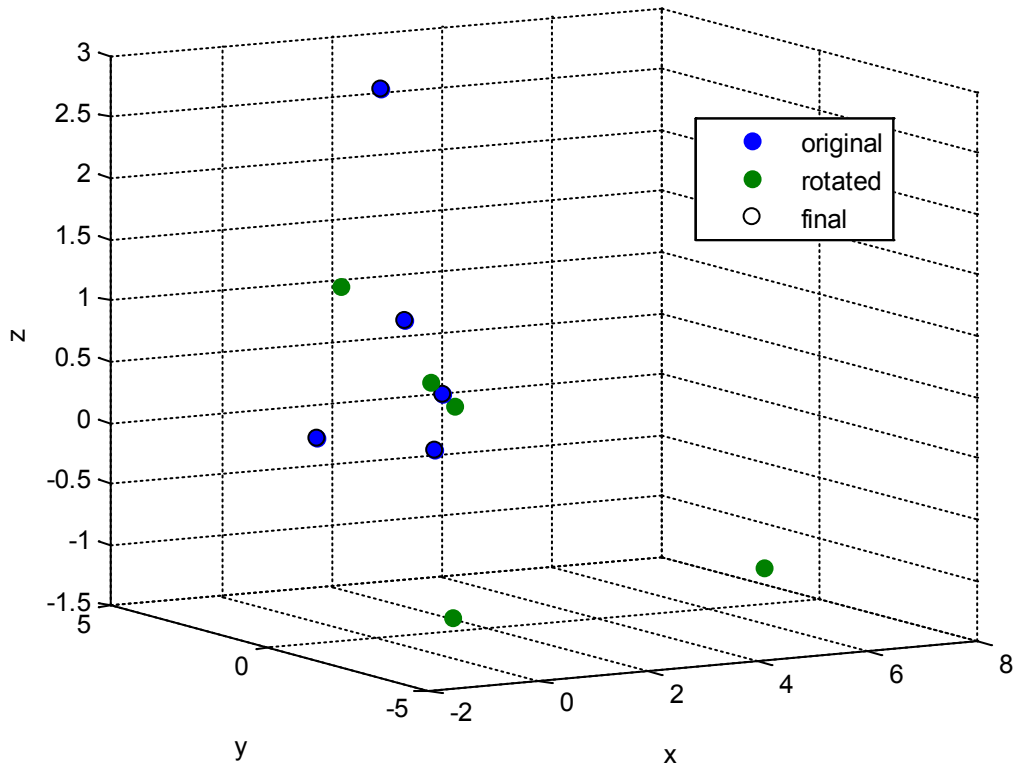
The application of this rotation matrix minimizes the root mean squared deviation between two paired sets of points, $\underline{\underline{P}}$ and $\underline{\underline{Q}}$.

In the example below, we provide a script kabsch.m that generates a set of 5 points in 3-D space. We use the Euler angles to create an arbitrary rotation matrix. We apply that rotation matrix and generate a new, rotated set of points. We then treat the original set and the rotated set as $\underline{\underline{P}}$ and $\underline{\underline{Q}}$. We perform the SVD operation, reconstitutes the rotation matrix and show that when the rotation matrix obtained from SVD is applied to $\underline{\underline{Q}}$, it perfectly reproduces the original $\underline{\underline{P}}$. After the script, we reproduce the output generated by the script including a plot of the original, rotated and final points.

```
clear all;
close all;
format short;
%
%  define a set of n points in d-dimensions
n = 5;
d = 3;
ptran = [1.0000000       0.0000000       0.0000000
0.0000000       2.0000000       0.0000000
0.0000000       0.0000000       3.0000000
4.0000000       5.0000000       0.0000000
1.0000000       1.0000000       1.0000000]
% transpose
p = transpose(ptran); % a dxn matrix

%
%  create random rotation matrix
%
pi = 2.0*asin(1.0);
d2r = pi/180.0;  % degrees to radians
alpha=17*d2r; % first Euler rotation angle
beta=67*d2r; % second Euler rotation angle
gamma=41*d2r; % third Euler rotation angle
%  first rotation matrix (about z-axis)
A1=[cos(alpha)  sin(alpha) 0
    -sin(alpha) cos(alpha) 0
    0 0 1];
%  second rotation matrix (about x-axis)
A2=[1 0 0
    0 cos(beta) sin(beta)
    0 -sin(beta) cos(beta)];
%  third rotation matrix (about z-axis)
A3=[cos(gamma) sin(gamma) 0
    -sin(gamma) cos(gamma) 0
    0        0 1];
%  generalized, arbitrary rotation matrix
Mrot=A1*A2*A3 % a dxd matrix

%
%  apply random rotation matrix
%
q = Mrot*p; % a dxn matrix
qtran = transpose(q) % a nxd matrix

%
%  use SVD to compute optimal rotation matrix
%  to map q back onto ptran
%
Amat = p*qtran;  % a dxd matrix

[U,S,V] = svd(Amat);
%[vmat,smat,wmat] = svd(Amat)
Ut = transpose(U);
rotmat = V*Ut % a dxd matrix
rott = transpose(rotmat) % a dxd matrix

%
```

```
%  apply inverse (transpose) of rotation matrix
%  obtained from SVD to q
%
qnew = rott*q; % a dxn matrix
qnewt = transpose(qnew)
% check
diff = norm((p - qnew),2)

%
%  create a plot
%
figure(1)
scatter3(p(1,1:n),p(2,1:n),p(3,1:n),'filled')
hold on;
scatter3(q(1,1:n),q(2,1:n),q(3,1:n),'filled')
hold on;
scatter3(qnew(1,1:n),qnew(2,1:n),qnew(3,1:n),'MarkerEdgeColor','k')
legend('original', 'rotated','final');
hold off;
view(-30,10)
xlabel('x');
ylabel('y');
zlabel('z');
```

```
>> kabsch

ptran =
     1     0     0
     0     2     0
     0     0     3
     4     5     0
     1     1     1

Mrot =
    0.6468    0.7136    0.2691
   -0.4658    0.0902    0.8803
    0.6039   -0.6947    0.3907

qtran =
    0.6468   -0.4658    0.6039
    1.4272    0.1804   -1.3894
    0.8074    2.6408    1.1722
    6.1552   -1.4122   -1.0579
    1.6295    0.5047    0.2999

rotmat =

    0.6468    0.7136    0.2691
   -0.4658    0.0902    0.8803
    0.6039   -0.6947    0.3907

rott =
    0.6468   -0.4658    0.6039
    0.7136    0.0902   -0.6947
    0.2691    0.8803    0.3907

qnewt =
    1.0000   -0.0000    0.0000
   -0.0000    2.0000   -0.0000
   -0.0000    0.0000    3.0000
    4.0000    5.0000   -0.0000
    1.0000    1.0000    1.0000

diff =   4.6534e-15
```



Visualization of the registration process via the Kabsch algorithm.

Appendix I. Derivation of the generalized crystal cell transformation matrix

Purpose:
Given axes magnitudes, a, b, and c, and axes angles θ_{AB}, θ_{AC}, and $\theta_{BC} = 90°$, derive the generalized transformation matrix, $\underline{\underline{B}}$, that will convert atomic position in the crystal-cell reference frame to atomic position in cartesian coordinates with the standard basis vectors ($\underline{i}, \underline{j}, \underline{k}$).

Derivation:
There is some arbitrariness to how one orientates the cartesian axes with respect to the crystal cell axes. In this case, we are going to keep the x-y face of the unit cell perpendicular to the Cartesian z-axis. Given this condition:

$$\underline{a} \cdot \underline{k} = 0 \tag{A1.1}$$

and

$$\underline{b} \cdot \underline{k} = 0 \tag{A1.2}$$

The consequence of these two statements immediately is that

$$B_{3,1} = 0 \qquad \text{and} \qquad B_{3,2} = 0 \tag{A1.3}$$

Explicity evaluate the dot product in equations (A1.1) and (A1.2) to verify this.

We have one more degree of freedom regarding the cartesian axes orientation. $\underline{i}$ and $\underline{j}$ must be perpendicular to k but still that leaves an arbitrary rotation of $\underline{i}$ and $\underline{j}$ within the a-b plane. Let's arbitrarily choose that $\underline{i}$ is parallel to $\underline{a}$. In this case

$$\underline{a} \cdot \underline{j} = 0 \tag{A1.4}$$

so that

$$B_{2,1} = 0 \tag{A1.5}$$

Because we know the magnitude of $\underline{a}$, and we have specified $\underline{\underline{B}}_{3,1} = 0$ and $\underline{\underline{B}}_{2,1} = 0$, we can determine the last remaining element of that column, $\underline{\underline{B}}_{1,1}$, by the 3-D Pythagorean theorem:

$$\sum_{m=1}^{3} \left(B_{m,1}\right)^2 = a^2 \tag{A1.6}$$

so that

$$B_{1,1} = \sqrt{a^2 - {B_{2,1}}^2 - {B_{3,1}}^2} = a \tag{A1.7}$$

At this point we have the first column and one element of the second column. In order to determine the other two elements of the second column, we can use our knowledge of the angles between $\underline{a}$ and $\underline{b}$, coupled with some

simple trigonometry. Because the i-j plane is the same as the a-b plane, we can project $\underline{b}$ onto $\underline{i}$ and $\underline{j}$ to get those components:

$$B_{1,2} = b \cdot \cos\theta_{AB} \tag{A1.8}$$

$$B_{2,2} = b \cdot \sin\theta_{AB} \tag{A1.9}$$

At this point we have the first two columns of the transformation matrix. Now, to obtain the final column, we use the invariance of the dot product with respect to coordinate systems to write two equations:

$$\underline{c} \cdot \underline{a} = \underline{b}_3 \cdot \underline{b}_1 = \cos\theta_{AC} \tag{A1.10}$$

and

$$\underline{c} \cdot \underline{b} = \underline{b}_3 \cdot \underline{b}_2 = \cos\theta_{BC} \tag{A1.11}$$

and we use the fact that we know the magnitude of $\underline{c}$

$$\sum_{m=1}^{3} \left(B_{m,3}\right)^2 = c^2 \tag{A1.12}$$

In equations (A1.10), (A1.11), and (A1.12), we have three equations, from which we can find three unknowns, which constitute the third column of the transformation matrix, $\underline{b}_3$. The rearrangement for the elements of $\underline{b}_3$ is just a little bit of ugly algebra, as follows. First we rearrange equations (A1.10) and (A1.11) to find the first two elements of the column vector as a function of the last element.

$$B_{1,3} = f_{x1} \cdot B_{3,3} + f_{x2} \tag{A1.13}$$

$$B_{2,3} = f_{y1} \cdot B_{3,3} + f_{y2} \tag{A1.14}$$

where the f constants are determined by the first two (already determined basis vectors)

$$f_{y1} = -\frac{\frac{B_{1,2} \cdot B_{3,1}}{B_{1,1}} + B_{32}}{\frac{B_{1,2} \cdot B_{2,1}}{B_{1,1}} + B_{22}} \tag{A1.15}$$

$$f_{y2} = \frac{\frac{B_{1,2}}{B_{1,1}}\cos\theta_{AC} + \cos\theta_{BC}}{\frac{B_{1,2} \cdot B_{2,1}}{B_{1,1}} + B_{22}} \tag{A1.16}$$

$$f_{x1} = B_{2,1} \cdot f_{y1} + \frac{B_{3,1}}{B_{1,1}} \tag{A1.17}$$

$$f_{x2} = B_{2,1} \cdot f_{y2} - \frac{\cos\theta_{AC}}{B_{1,1}} \tag{A1.18}$$

We then need only $B_{3,3}$, which we obtain by first substituting equations (A1.15)-(A1.18) into equations (A1.13) and (A1.14). Second, we substitute the resulting equations into equation (A1.12), which is then only a function of $B_{3,3}$. We rearrangeto find:

$$a_{qf} B_{3,3}^{\ 2} + b_{qf} B_{3,3} + c_{qf} = 0 \tag{A1.19}$$

where

$$a_{qf} = 1 + f_{y1}^{\ 2} + f_{x1}^{\ 2} \tag{A1.20}$$

$$b_{qf} = 2 \cdot (f_{y1} f_{y2} + f_{x1} f_{x2}) \tag{A1.21}$$

$$c_{qf} = f_{x2}^{\ 2} + f_{y2}^{\ 2} - c^2 \tag{A1.22}$$

We use the quadratic formula to solve

$$B_{3,3} = \frac{-b_{qf} \pm \sqrt{b_{qf}^{\ 2} - 4a_{qf}c_{qf}}}{2a_{qf}} \tag{A1.23}$$

We choose the positive root. Once we have $B_{3,3}$, we can substitute back into equations (A1.13) and (A1.14) to obtain the other two elements of the third basis vector.

Comments:

This derivation is not the most elegant creature but it has a couple virtues. The first virtue is that the derivation is easily understandable knowing only algebra and trigonometry. The second virtue is that this formulation as written above is easily amenable to computer usage. If you look carefully at the above equations, the unknowns always appear on the left-hand-side of the equation and the knowns appear on the right-hand-side which is as the computer must have it, in order to solve.

Example:

Consider a crystal cell with axes defined by a = 14Å, b=14Å, c=8Å. $\theta_{AB} = 120°$, $\theta_{AC} = 90°$, $\theta_{BC} = 90°$. Find the transformation matrix to the standard 3-D Cartesian coordinate system.

$$B_{3,1} = 0 \qquad \text{and} \qquad B_{3,2} = 0 \tag{A1.3}$$

$$B_{2,1} = 0 \tag{A1.5}$$

$$B_{1,1} = \sqrt{a^2 - B_{2,1}^{\ 2} - B_{3,1}^{\ 2}} = a = 14 \tag{A1.7}$$

$$B_{1,2} = b \cdot \cos\theta_{AB} = -7 \tag{A1.8}$$

$$B_{2,2} = b \cdot \sin\theta_{AB} = 12.12 \tag{A1.9}$$

$$f_{y1} = -\frac{\frac{B_{1,2} \cdot B_{3,1}}{B_{1,1}} + B_{32}}{\frac{B_{1,2} \cdot B_{2,1}}{B_{1,1}} + B_{22}} = 0 \tag{A1.15}$$

$$f_{y2} = \frac{\frac{B_{1,2}}{B_{1,1}}\cos\theta_{AC} + \cos\theta_{BC}}{\frac{B_{1,2} \cdot B_{2,1}}{B_{1,1}} + B_{22}} = 0 \tag{A1.16}$$

$$f_{x1} = B_{2,1} \cdot f_{y1} + \frac{B_{3,1}}{B_{1,1}} = 0 \tag{A1.17}$$

$$f_{x2} = B_{2,1} \cdot f_{y2} - \frac{\cos\theta_{AC}}{B_{1,1}} = 0 \tag{A1.18}$$

$$a_{qf} = 1 + {f_{y1}}^2 + {f_{x1}}^2 = 1 \tag{A1.20}$$

$$b_{qf} = 2 \cdot (f_{y1} f_{y2} + f_{x1} f_{x2}) = 0 \tag{A1.21}$$

$$c_{qf} = {f_{x2}}^2 + {f_{y2}}^2 - c^2 = -c^2 = -64 \tag{A1.22}$$

We use the quadratic formula to solve

$$B_{3,3} = \frac{-b_{qf} \pm \sqrt{{b_{qf}}^2 - 4a_{qf}c_{qf}}}{2a_{qf}} = \pm 8 \tag{A1.23}$$

$$B_{1,3} = f_{x1} \cdot B_{3,3} + f_{x2} = 0 \tag{A1.13}$$

$$B_{2,3} = f_{y1} \cdot B_{3,3} + f_{y2} = 0 \tag{A1.14}$$

So that the transformation matrix is

$$\underline{\underline{B}}^{(1)} = \begin{bmatrix} a & b\cos\theta_{AB} & 0 \\ 0 & b\sin\theta_{AB} & 0 \\ 0 & 0 & c \end{bmatrix} = \begin{bmatrix} 14 & -7 & 0 \\ 0 & 12.12 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

Appendix II. Transformations from Cartesian to Spherical and Cylindrical Coordinates

Another set of common transformations (nonlinear transformations) are given below.

AII.1 from Cartesian to Cylindrical

$$r = \sqrt{x^2 + y^2}$$

$$\theta = a\tan\left(\frac{y}{x}\right)$$

$$z = z$$

AII.2 from Cylindrical to Cartesian

$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$z = z$$

AII.3 from Cartesian to Spherical

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = a\tan\left(\frac{y}{x}\right)$$

$$\phi = a\tan\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$

AII.4 from Spherical to Cartesian

$$x = r\cos\theta\cos\phi$$

$$y = r\sin\theta\cos\phi$$

$$z = r\sin\phi$$

AII.5 from Cylindrical to Spherical

$$r = \sqrt{r^2 + z^2}$$

$$\theta = \theta$$

$$\phi = a\tan\left(\frac{z}{r}\right)$$

AII.6 from Spherical to Cylindrical

$$r = r\cos\phi$$

$$\theta = \theta$$

$$z = r\sin\phi$$