MATLAB Exercise • Level 1

CHARLES J. AMMON
DEPARTMENT OF GEOSCIENCES
PENN STATE UNIVERSITY

The Rotation of Vectors and Tensors

ROTATING SEISMOGRAMS

In many applications of mathematics involving direction quantities we must convert quantities from one coordinate system to another. One common rotation applied so often in seismology that we seldom even think of the details is the rotation from geographic coordinates used to align seismometers to the radial-tangential system more convenient for analyzing seismic waves. The coordinate systems are shown in Figure 1. The mathematical problem is to use the observed north and

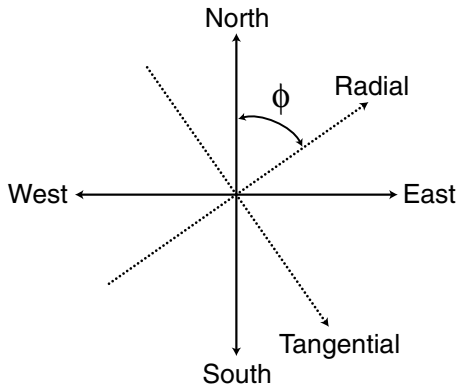

Figure 1. Seismograms are generally recorded with a vertical (not shown) and north and east components. Analysis is often easier when these observations are rotated into the radial-tangential coordinates, where P-SV-Rayleigh waves are separated from SH-Love waves. Formally the radial is the direction along the great circle connecting the epicenter and seismometer and is positive in a direction away from the source.

east components of ground motion to compute the radial and tangential components of motion. We can perform the rotation using a matrix multiplication of a vector consisting of the north and east components with a "rotation matrix" given by

$$A = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}. \tag{1}$$

Then we have

$$\begin{bmatrix} R \\ T \end{bmatrix} = A \cdot \begin{bmatrix} N \\ E \end{bmatrix}. \tag{2}$$

When we implement the computation in MATLAB we can perform the calculation as a matrix multiplication or we can perform the matrix computation analytically and relate the different signals using

$$\begin{aligned} R(t) &= \cos\phi \cdot N(t) + \sin\phi \cdot E(t) \\ T(t) &= -\sin\phi \cdot N(t) + \cos\phi \cdot E(t) \end{aligned} \quad . \tag{3}$$

Here's a script to rotate north & east into radial & tangential:

```
function [r, t] = rt_rotate(n,e,phi)
%
%    function to compute the radial and tangential
%       components of motion given the north and east
%       components and the azimuth from north of the radial
%       direction.
%
%   INPUT:
%           n   a vector of values in the north direction
%           e   a vector of values in the east direction
%           phi the azimuth of the radial direction of motion
%               input a value in degrees.
%
%   OUTPUT:
%           r radial vector
%           t tangential (or transverse) direction
%
%

deg_to_rad = pi/180; % changes to arguments in function do not
                     % affect the values in the calling function
cphi = cos(phi*deg_to_rad);
sphi = sin(phi*deg_to_rad);
%
r =  cphi * n + sphi * e;
t = -sphi * n + cphi * e;
```

Note that MATLAB does the vector operations with the same notation that you would use for scalar operations. That is, if *n* and *e* are vectors with 1000 points, the function will rotate all 1000 points, if *n* and *e* contain two points, the function will return *r* and *t* with the correct two points. Here's an example

```
>> n = [ 1, 2, 2, 2, 2];
>> e = [ 0, 0, 0, 0, 0];
>> [r t] = rt_rotate(n,e,0)
>> r =
>>      1     2     2     2     2
>> t =
>>      0     0     0     0     0
>>
>> [r t] = rt_rotate(n,e,90)
>> r =
>>      0     0     0     0     0
>> t =
>>     -1    -2    -2    -2    -2
```

```
>>
>> [r t] = rt_rotate(n,e,-45)
>> r =
>>     0.7071    1.4142    1.4142    1.4142    1.4142
>> t =
>>     0.7071    1.4142    1.4142    1.4142    1.4142
```

ROTATING A TENSOR

We can also rotate tensors, although the equation is a little more complicated. To rotate a tensor, S, through an angle of ϕ about a vertical axis, we use

$$B = A^T S A \tag{4}$$

where

$$A = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

This might come in handy if you want to specify a particular stress field using an azimuth but compute the stress tensor in geographic coordinates. For example suppose that you wanted a uniaxial compression oriented with an azimuth of 30°. The stress tensor is

$$\sigma = A^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} A \tag{6}$$

where A is computed using $\phi = 30°$. Here's a script to do the rotation

```
function [ B ] = tensor_rotate(A,phi)
%
%    function to rotate a three-by-three tensor through
%       an azimuth of phi
%
%   INPUT:
%           A   a tensor with
%                  nn ne nz
%                  en ee ez
%                  zn ze zz
%
%           phi the azimuth of the radial direction of motion
%               input a value in degrees.
%
%   OUTPUT:
%           B the rotated tensor
%
%
```

```
deg_to_rad = pi/180; % changes to arguments in function do not
                     % affect the values in the calling function
cphi = cos(phi*deg_to_rad);
sphi = sin(phi*deg_to_rad);
R = [cphi,sphi,0;-sphi,cphi,0;0,0,1];
%
B = R' * A * R;
```

and here's a sample application

```
>> S = [1,0,0;0,0,0;0,0,0]
>> S =
>>      1     0     0
>>      0     0     0
>>      0     0     0
>> sigma = tensor_rotate(S,30)
>> sigma =
>>     0.7500   -0.4330         0
>>    -0.4330    0.2500         0
>>          0         0         0
>> stensor_decomp(sigma)
>>
>> The eigenvalues are:
>>
>>        1.00   0.00   0.00
>>
>> The eigenvectors are:
>>
>>         V1     V2    V3
>>
>>   N: -0.866  0.000  0.500
>>
>>   E: -0.500  0.000 -0.866
>>
>>   D: -0.000  1.000  0.000
>>
>> ans =
>>    -0.8660         0    0.5000
>>    -0.5000         0   -0.8660
>>          0    1.0000         0
```

The script *stensor_decomp* is described in another MATLAB exercise. I used it to check the rotation. The rotated tensor has one non-zero principal value with a principal axis oriented in a southwest direction (uniaxial compression from an azimuth of 30N).

EXERCISES

Exercise 1: Rotate the vectors

$$n = [1, 1, 1, 1, 1]$$
$$e = [-1, -1, -1, -1]$$

Exercise 2: Rotate the vectors

$$n = [3, 2, 1]$$
$$e = [1, 2, 3]$$

Exercise 3: Construct a formula to perform the inverse transformation, from radial and transverse, to north and east.

Exercise 4: Compute a tensor in N•E•D (north-east-down) coordinates that corresponds to uniaxial compression from the southwest.