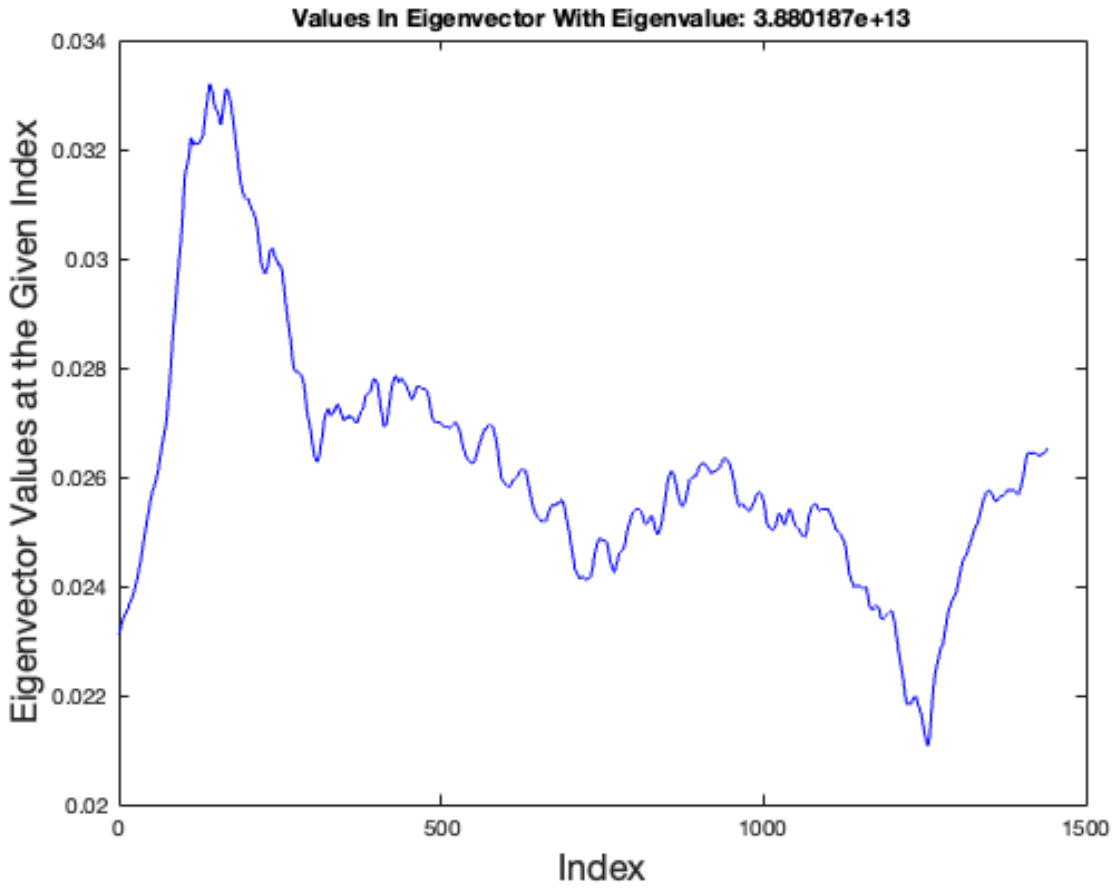
```
A = importdata('mariana_depth.csv');
lon = importdata('mariana_longitude.csv');
lat = importdata('mariana_latitude.csv');

%numRows = 1320;
%numCols = 1440;
%A = zeros(numRows,numCols);
%for i = 1:numRows
%   for j = 1:numCols
%      A(i,j)=randi(10,1);%first num in randi is random int generator
%   end
%end

fprintf('OUTPUT: \n');

%%2.2.1

A_t=transpose(A);
B= A_t*A;

[eVal1,v1] = findEigens(B);

figure(1);
plot(1:sizeColVect(B),v1,'blue');%Plots Values In Eigenvector
title(sprintf('Values In Eigenvector With Eigenvalue: %i', eVal1));
xlabel('Index','FontSize',16);
ylabel('Eigenvector Values at the Given Index','FontSize',16);


function [value, vector]=findEigens(matrix)%finds eigenvector and associated
 value by using the first method
    randUpBound = 10;

    u=randomUnitColVector1(sizeColVect(matrix),randUpBound);%generates a
 random vector to start with
    u1= matrix*u;
    u1=unitVect(u1);

    smallNum=1e-15;
    whileCounter =0;
    while(mag(u1-u)>smallNum)%does the same as above but does it until the
 unit vector becomes unchanging and therefore must be an eigenvector
        whileCounter=whileCounter+1;
        u=u1;
        u1= matrix*u;
        u1=unitVect(u1);
    end

    vector= u1;
    scaledV1 = matrix*vector;
    value = scaledV1(1,1)/vector(1,1);
    fprintf('whileCounter: %i \n',whileCounter);%
```

```
end

function sizeVect = sizeColVect(colVect)
    b=size(colVect);
    sizeVect=b(1);
end

function vect = randomUnitColVector1(size,randomUpperBound)
    vect = zeros(size,1);
    for k = 1:size
          vect(k,1)=randi(randomUpperBound,1);%first num in randi is random
 int generator
    end
    vect=unitVect(vect);
end

function vect = randomUnitColVector2(size)
    vect=randomUnitColVector1(size,10);
end

function unitVector = unitVect(array)
    unitVector = array/mag(array);
end

function magnitude = mag(array)
    magnitude = 0;
    for k = 1:sizeColVect(array)
      magnitude=magnitude+(array(k,1))^2;
    end
    magnitude= sqrt(magnitude);
end
```

OUTPUT:
whileCounter: 8



Published with MATLAB® R2021b