
```

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

%%2.1

figure(1)
surf(A)
colormap(cool)
shading interp
title("Mariana Trench Surface Model")
xlabel("X Position in Matrix")
ylabel("Y Position in Matrix")
zlabel("Depth in Relation to Ocean Floor")
[MinMat, MinInd] = min(A);%Gives a vector with the deepest point on
    every Y value.
[Min, Indey] = min(min(A));%Gives the min value in the vector of y min
    values and its Y coordinate.
Index = MinInd(1,Indey);%Finds the X value of the min value.
%Index is the X index of the smallest value
%Indey is the Y index of the smallest value
%Min is the actual value.
%For this code Lon is the X axis and Lat is the Y axis.
%disp(Min);
%disp(Index);
%disp(Indey);
deepestPoint = A(Index, Indey);
disp("Sanity Check: " + (deepestPoint == Min));%Checks to see if Min
    is the same as the value found using the X and Y indexes.
deepestLon = lon(Index, 1);
deepestLat = lat(Indey, 1);
disp("Deepest Point in m:" + deepestPoint);
disp("Longitude of deepest point: " + deepestLon);
disp("Latitude of deepest point: " + deepestLat);

meanDepthKM = mean(mean(A))/1000;
disp("Average depth below sea floor in km: " + meanDepthKM);

%%2.2
%%2.2.1
v = randn(1440, 1);
u = v/norm(v);

n = u;
n1 = zeros(1440, 1);
e = 0;
count = 0;

while norm(n1 - n) > 0.0001
    if (count > 0)
        n = n1;

```

```

    end
    e = norm((A'*A*n));
    n1 = (A'*A*n)/norm((A'*A*n));
    count = count + 1;
    %disp(norm(n1));
end
figure(2);
plot(n1);
title("Eigenvector Plot")
disp("2.2.1 Eigenvalue: " + e)

%%Warning this code takes a long time to execute on my powerful
laptop. 3+ minutes
%2.2.2
v2 = zeros(1440, 50);
e2 = zeros(1, 50);
for i = 1:50
    t = randn(1440, 1);
    m = t/norm(t);
    m1 = zeros(1440, 1);
    count = 0;
    while norm(m1 - m) > 0.0001
        if (count > 0)
            m = m1;
        end
        m1 = A'*A*m;
        temp = m1;
        store = zeros(1440, 1);
        %if (i - 1) > 1
        for j = 1:(i - 1)
            store = store + ((temp'*v2(:, j))*v2(:, j));
        end
        %end
        m1 = temp - store;
        e2(1, i) = norm(m1);
        m1 = m1/norm(m1);
        count = count + 1;
    end
    v2(:, i) = m1;
    %disp("Cycle: " + i + "/50 Norm Check: " + norm(m1));
end
figure(3)
semilogx(e2);
title("Eigenvalue semilog plot")

%%2.3
%2.3.1
E = diag(sqrt(e2));
U = zeros(1320, 50);
for i = 1:50
    U(:, i) = (A*v2(:, i))/e2(i);
end
%2.3.2

```

```

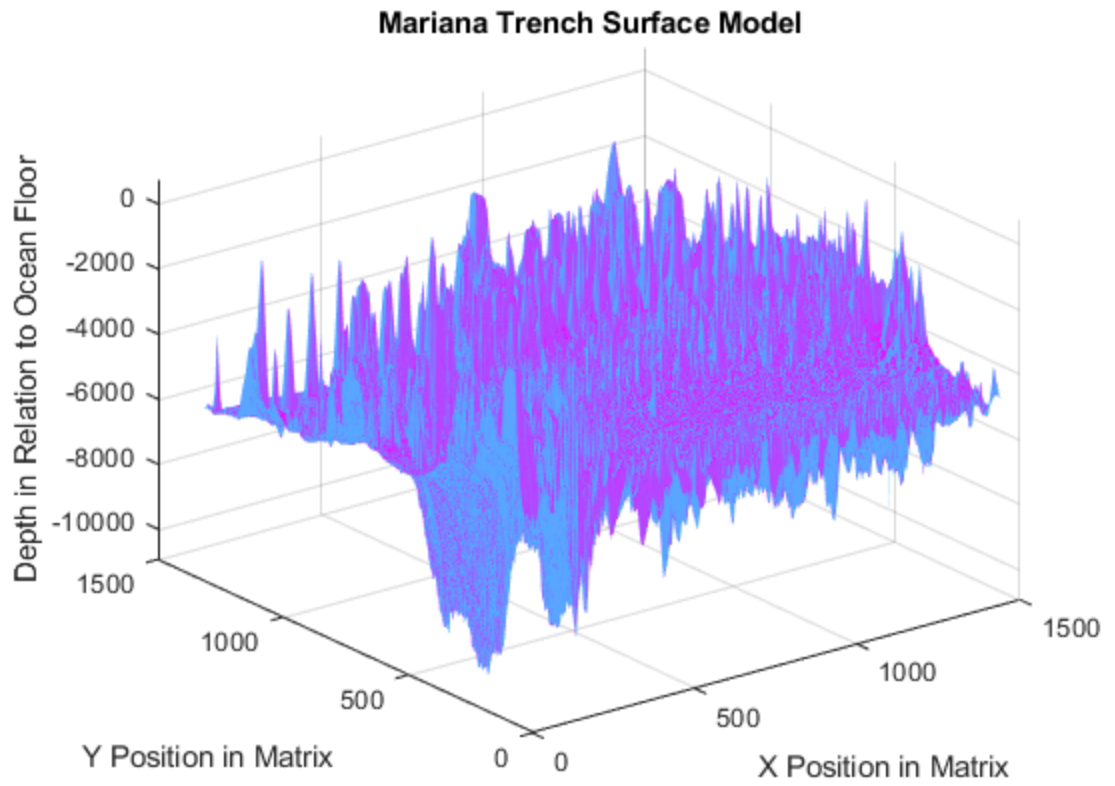
disp("Number of Elements in A: " + numel(A));
disp("Number of Elements in E + U + V: " + (numel(E) + numel(U) +
    numel(v2)));

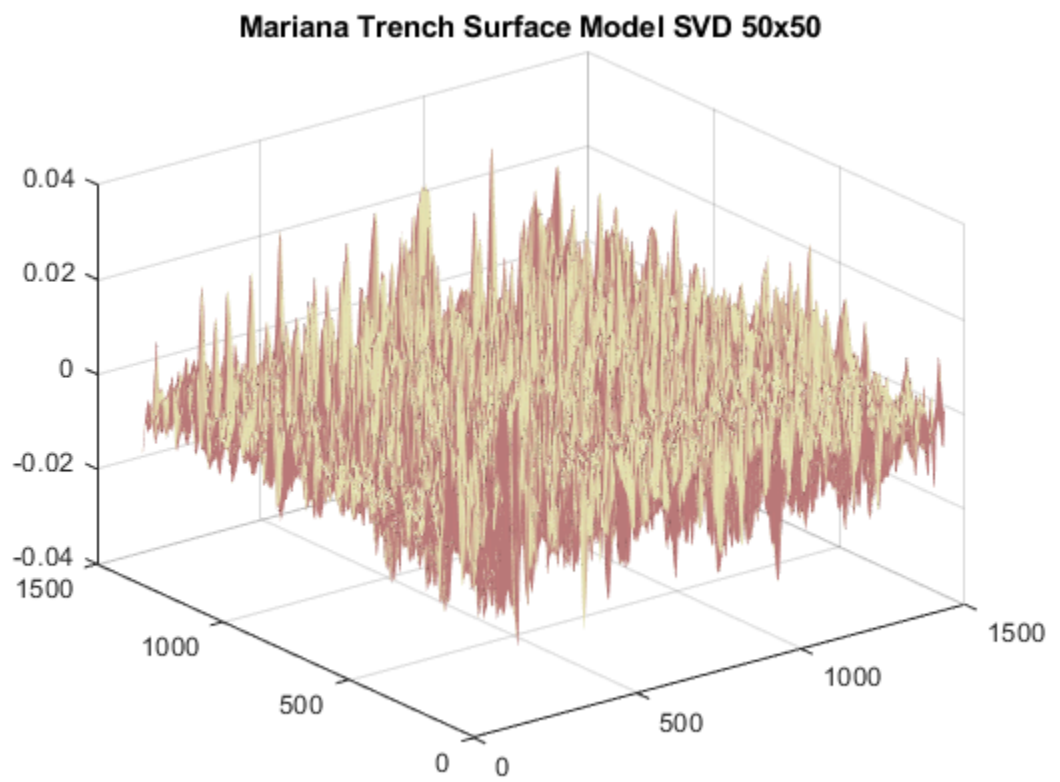
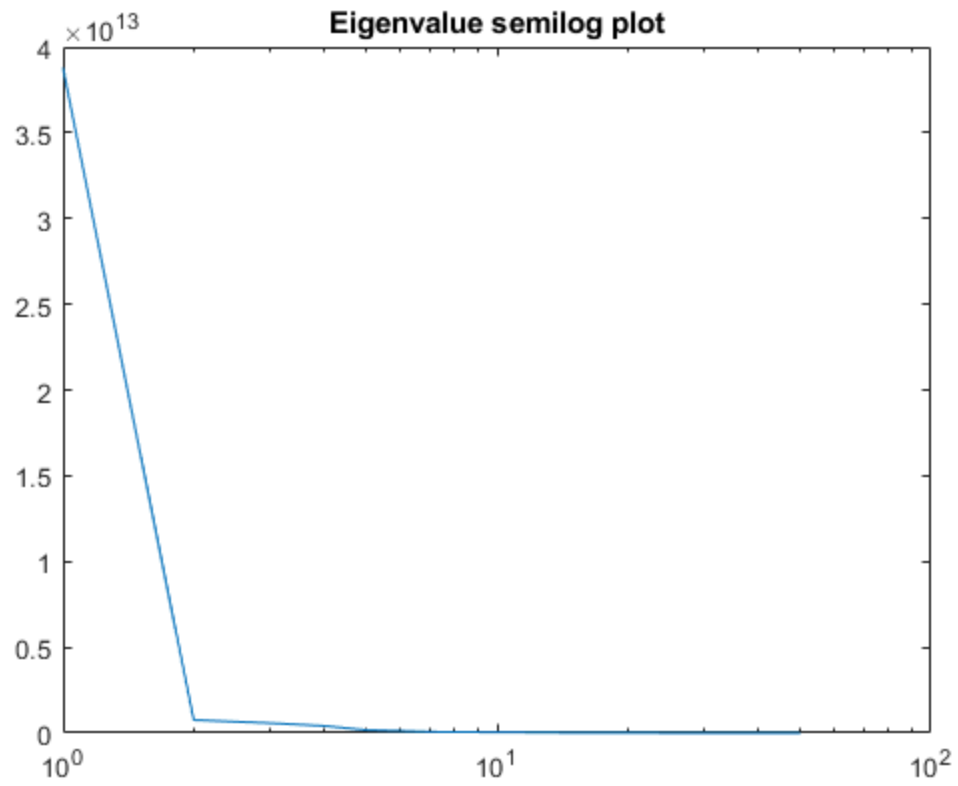
%2.3.3
figure(4)
surf(U*E*v2')
colormap(pink)
shading interp
title("Mariana Trench Surface Model SVD 50x50")

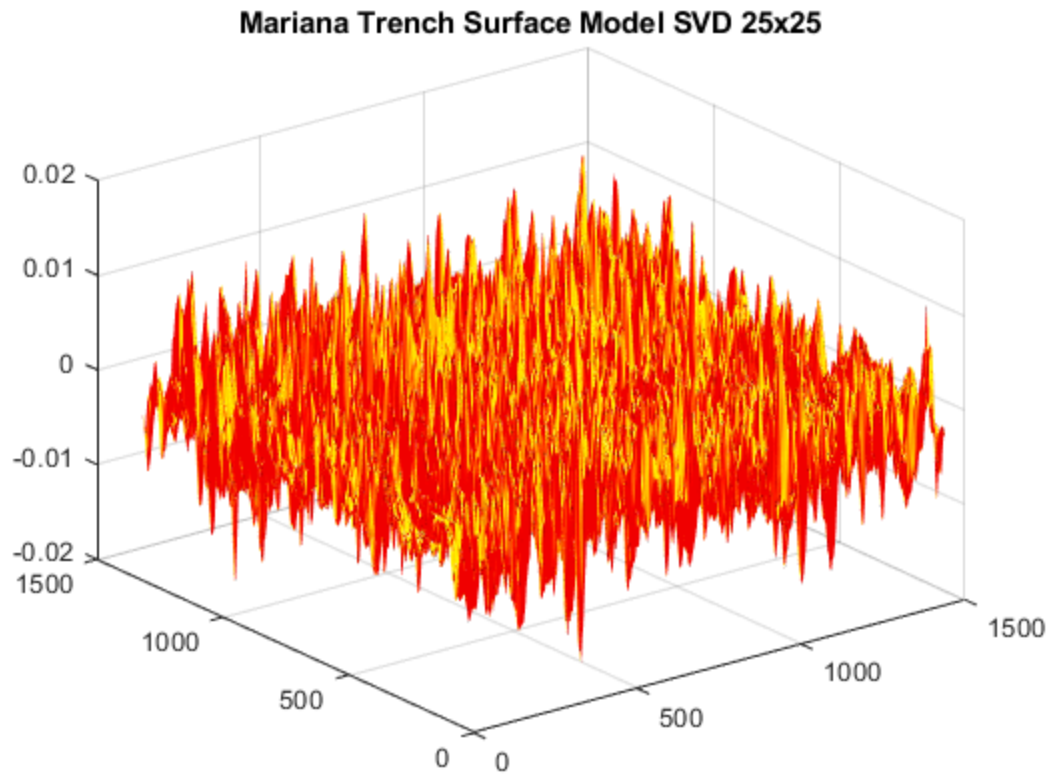
%2.3.4
E2 = diag(sqrt(e2(1:25)));
U2 = zeros(1320,25);
for i = 1:25
    U2(:, i) = (A*v2(:, i))/e2(i);
end
figure(5)
surf(U2*E2*v2(:, 1:2:50)')
colormap(hot)
shading interp
title("Mariana Trench Surface Model SVD 25x25")

Sanity Check: true
Deepest Point in m:-10930
Longitude of deepest point: 142.2
Latitude of deepest point: 11.333
Average depth below sea floor in km: -4.453
2.2.1 Eigenvalue: 38801868787862.16
Number of Elements in A: 1900800
Number of Elements in E + U + V: 140500

```







Published with MATLAB® R2019b