

APPM 2360 - INTRO DIFF EQ W/LIN ALG

FALL 2024

Project 2 - Analyzing The Mariana Trench

Daniel Alemayehu, Eli Grundberg, Samuel Meyn

November 12, 2024

Contents

1	Introduction	2
2	Initial Trench Investigations	2
3	Eigenvalue Computation	3
4	SVD Incomplete Decomposition	4
5	Conclusion	8
A	Figure 1 Code	9
B	Figure 2 and 3 Code	11
C	Figure ??, ??, ??, ?? Code	13

1 Introduction

We are interested in analyzing and understading the layout of the Mariana Trench. However, we have a large amount of data to work with so we'll need to work to create an approximation of the trench such that we can still have an accurate picture of the trench to better analyze while utilizing less data and memory to do so.

2 Inital Trench Investigations

In order to better understand reducing the trench data we'll take a subset of it and operate on it as normal. Utilizing Matlab's Contour and Surface Plots we can get a good picture of the trench below from our data subset.

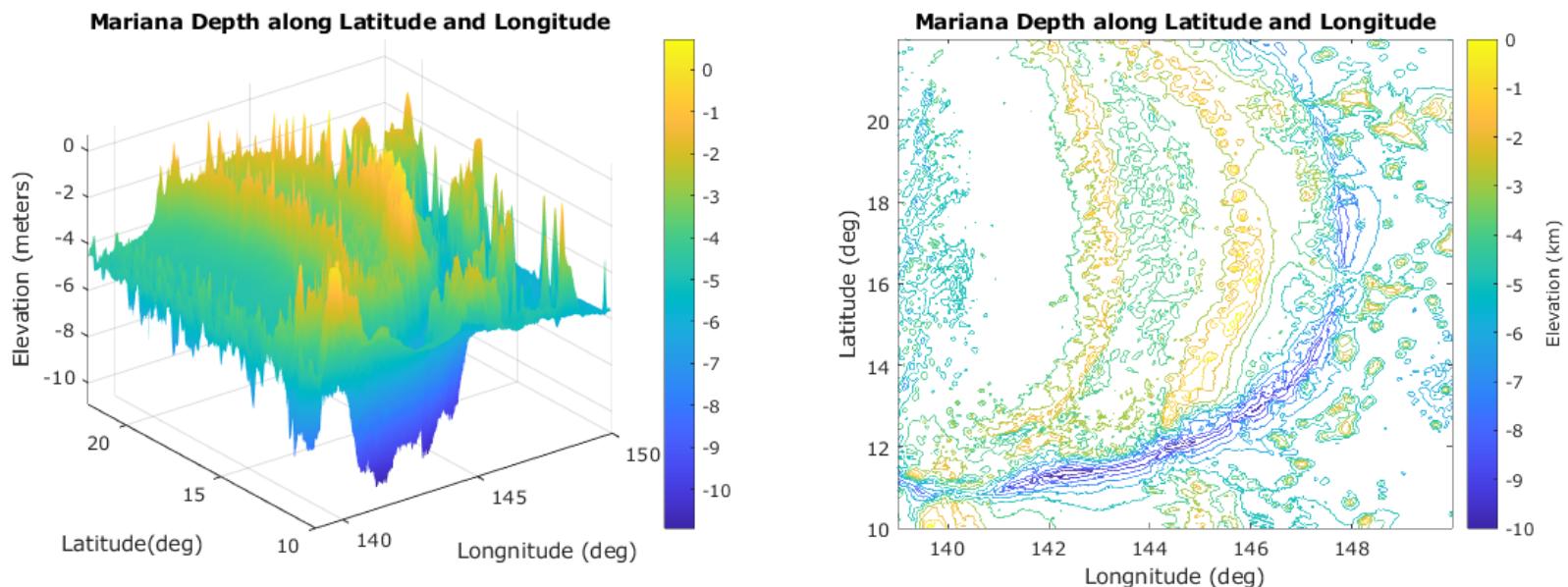


Figure 1: Code in Appendix [A](#)

We can also find that the maximum depth of the trench is -10.93 km at lattitude 13.2° longitude 140.3° (calculated in Appendix [A](#)). In addition, in our data subset can also find that the average depth of the trench below sea level is given to be -7.2048 km (calculated in

Appendix A).

3 Eigenvalue Computation

We can find an eigenvectors associated with our matrix using an algorithm outlined in Appendix ???. The reason this algorithm works can be explained as follows: We use this algorithm to find the eigenvector associated with the largest eigenvalue, and then plot the components of the eigenvector as a function of the n th component.

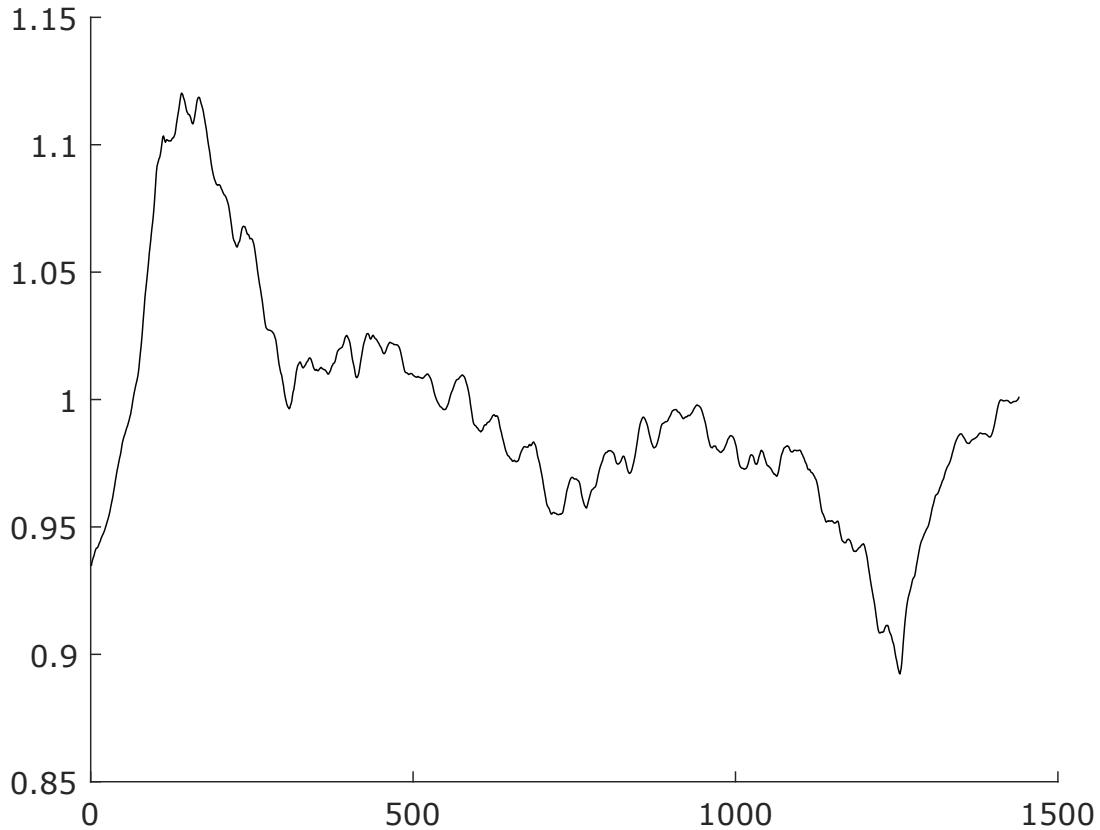


Figure 2: Eigenvector \mathbf{V}_1 plot

We can also apply *Gram-Schmidt Orthogonalization* to find eigenvectors as well which we'll use for the 50 largest eigenvectors. We plot the 50 largest eigenvalues associated with these eigenvectors:

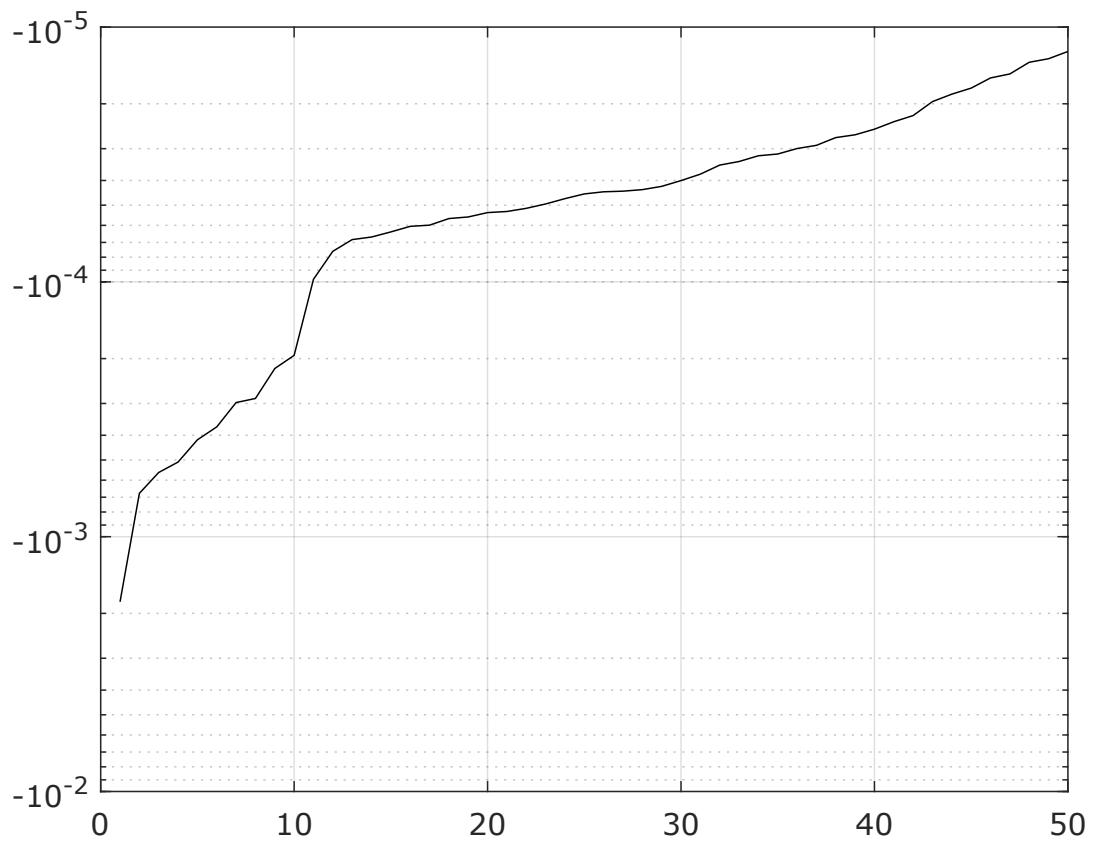
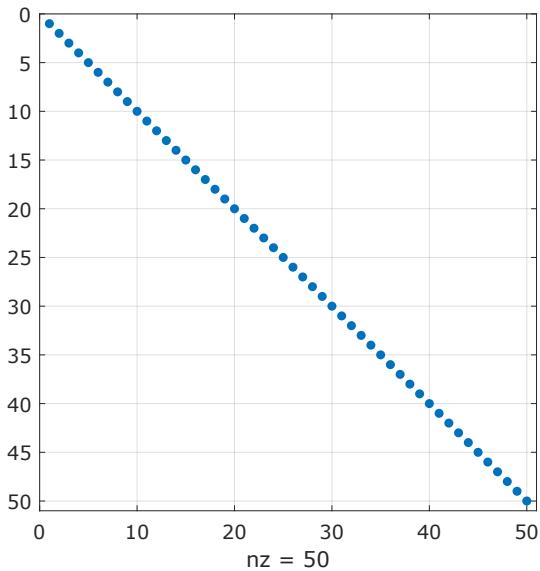


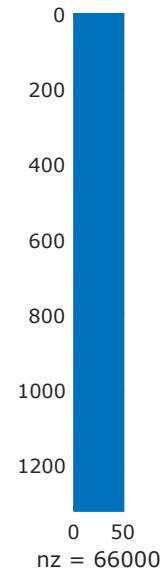
Figure 3: n th largest eigenvalue vs $\ln(E_n)$ plot

4 SVD Incomplete Decomposition

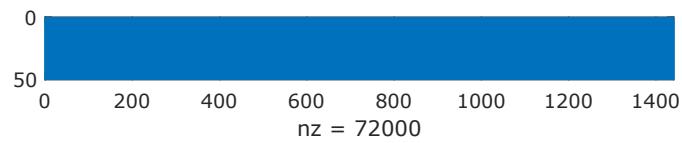
After completing our Incomplete SVD Decomposition (where \mathbf{U}, \mathbf{V}^T have 50 columns) our spy output looks like the following:



(a) Σ Matrix



(b) \mathbf{U} Matrix



(c) \mathbf{V}^T Matrix

As one method for comparing the efficiency of the Incomplete SVD vs the full matrix of trench data we can compare the total number of entries and the total number of nonzero entries. For our SVD up to the 50th eigenvalue we note the following:

- For the number of total entries, our matrix has 7.39% of the number of total matrix entries as matrix \mathbf{A} .

- For the number of nonzero entries, our matrix has 7.26% of the number of nonzero entries as matrix **A**.

We've made plots of several SVD decompositions with varying column counts (10, 50, 100) which appear below.

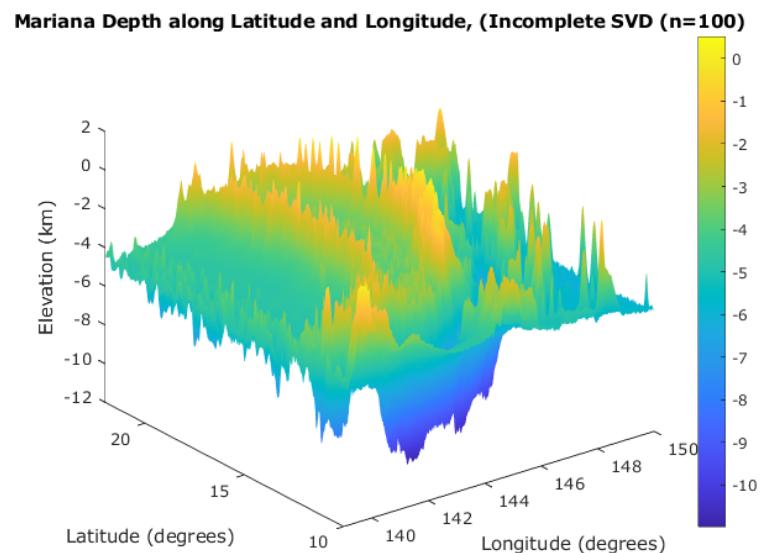
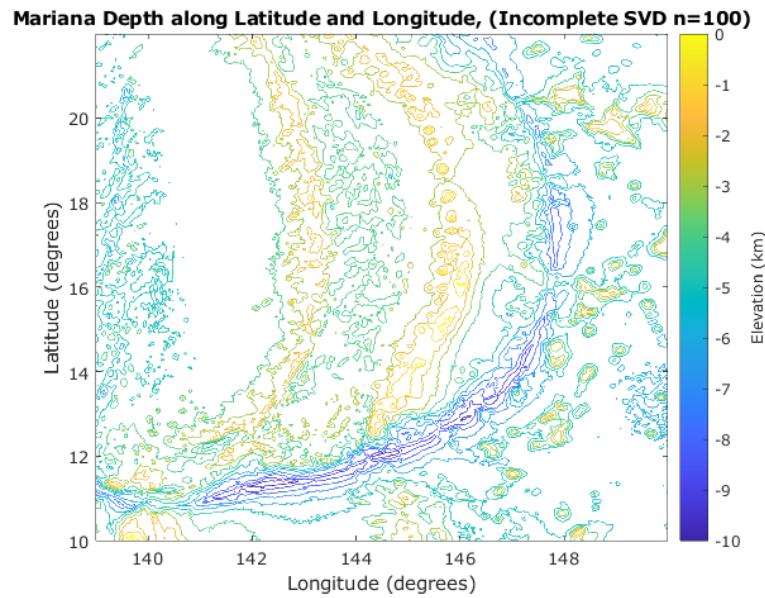
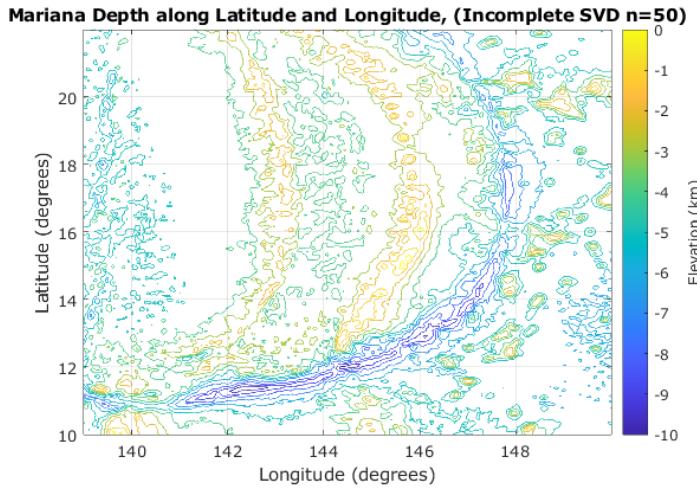
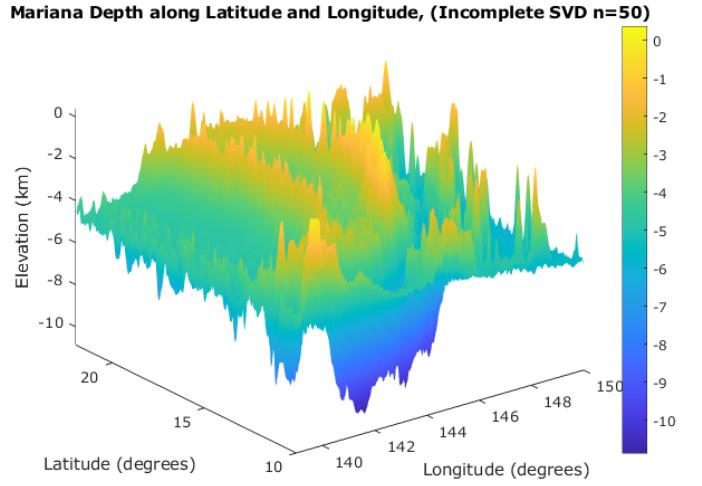


Figure 5: Code in Appendix C

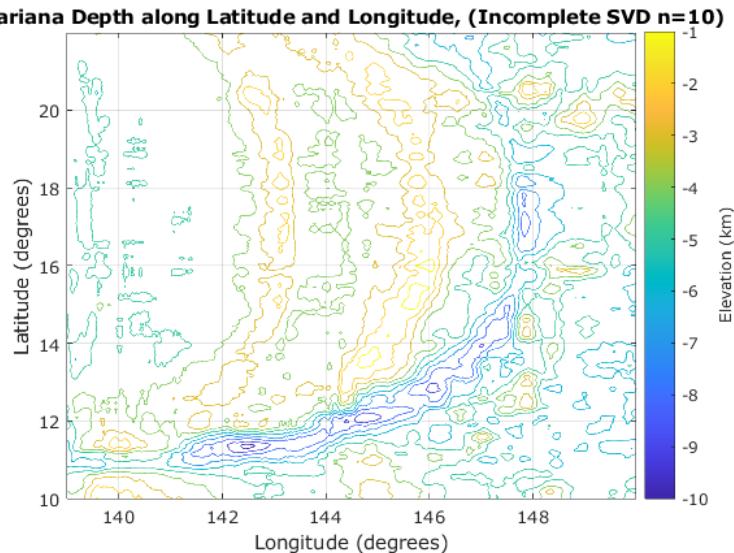


(a) Maximal Depth: km at Lattiude Longitude



(b) Average Depth Below Sea Level: km

Figure 6: Code in Appendix C



(a) Σ Matrix

Figure 7: Code in Appendix C

As we reduce the size of the Σ , \mathbf{U} , \mathbf{V} matrixies we notice that our graph appears to "smooth out". Commonly, this is refered to as a lower resolution version of our original data. Additionally, we'll note that the incomplete SVD up to the 10th largest eigenvalue is far more noticeable "low res" than the other two incomplete SVDs we provide which are far

closer/more similar to our original data.

5 Conclusion

Efficacy of incomplete SVD decomposition in data representation and memory management.

Appendix A Figure 1 Code

The following code is in MATLAB syntax:

```
clc
clear
close all

%% Part 2.1 - Samuel H. Meyn

%% 2.1.1
%
% Data Plots
%

%Import all data
Z = importdata('mariana_depth.csv');
Y = importdata('mariana_longitude.csv');
X = importdata('mariana_latitude.csv');
Z = transpose(Z);
Zkm = Z .* 1/1000;

figure(1)

Levels = [-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11];
contour(Y, X, Zkm, Levels)
title('Mariana Depth along Latitude and Longitude');
xlabel('Longitude (deg)');
ylabel('Latitude (deg)');
cb = contourbar("eastoutside");
cb.XLabel.String = "Elevation (km)";

figure(2)

surf(Y,X,Zkm, 'EdgeColor','none')
title('Mariana Depth along Latitude and Longitude');
xlabel('Longitude (deg)');
ylabel('Latitude(deg)');
zlabel('Elevation (meters)');
colorbar;

%% 2.1.2

% Finding Maximal Depth

%
% Incorrect Code (returns incorrect lat/long
%
```

```

% Min= min(Z, [], 'all');
% for i=1:length(Z)
%     for j=1:width(Z)
%         if Z(i,j) == Min
%             break
%         end
%     end
% end
%
% maxDepthLatitude = j;
% maxDepthLongitude = i;
% maxDepthLongitude;
% maxDepthLatitude;
% Min;
% Z(maxDepthLongitude, maxDepthLatitude);

[Min2, I] = min(Z, [], 'all');
[MinLong, MinLat] = ind2sub([1440 1320], I);
Min2
Z(MinLong, MinLat)
X(MinLat)
Y(MinLong)

%% 2.1.3

% Mean depth sub -6km
meanNum = 0;
N = 0;

for i=1:length(Z)
    for j=1:width(Z)
        if Z(i,j) < -6000
            meanNum = meanNum + Z(i,j);
            N = N + 1;
        end
    end
end

meanTrenchDepth = meanNum/N;
meanTrenchDepth/1000

```

Appendix B Figure 2 and 3 Code

The following code is in MATLAB syntax:

```
clear;
clc;
close all;

A = readmatrix('mariana_depth.csv');
%dataLat = readmatrix('mariana_latitude.csv');
%dataLong = readmatrix('mariana_longitude.csv');

%% Part 1
tic
%Transpose A
A_T = A';
%Create Unit vector
n = 1;
x = randn(n,1);
x = x / norm(x);

%Iterate until the unit vector for n+1 minus the previous unit vector
%doesn't change
for i = 1:10
    unit_x = A_T*A*x;
    unit_x = unit_x / (norm(unit_x));

    if norm(unit_x - x) <= 0
        break;
    end

    x = unit_x;

end
%Show final eigenvalue and eigenvector
eigen_final = x;
k = eig(A'*A);
x_n_vector = sqrt(sum(x^2));
N = length(x);
hold on;
plot(1:N, x_n_vector, 'k-');
toc
%% Part 2

v_50 = zeros(50,1);

%Initialize a unit vector
```

```

n = 1;
u = randn(n,1);
u = u / norm(u);
%
for i = 1:50
u_new_asterisk = A'*A*u;
u_new = u_new_asterisk - sum((u_new_asterisk'*v_50(i))*v_50(i)));
u_new = u_new / norm(u_new);
if norm(u_new - u) < 0.002
break;
end

u = u_new;
end

v_i = u_new;

figure(2)
semilogy(1:50, trimdata(k, [50 1]), 'k-');
grid

```

Appendix C Figure ??, ??, ??, ?? Code

The following code is in MATLAB syntax:

```
clear;
clc;
close all;
%
% PART 1: Incomplete SVD Decomposition
%
A = readmatrix('mariana_depth.csv');
dataLat = readmatrix('mariana_latitude.csv');
dataLong = readmatrix('mariana_longitude.csv');

A_T = A';

V = zeros(1440, 50);

n = 1440;
u = randn(n,1);
u = u / norm(u);

for i = 1:50
    while 1
        u_new = A_T*A*u;
        summation = 0;
        for j = 1:i
            summation = summation + (u_new'*V(:,j))*V(:,j);
        end
        u_new = u_new - summation;
        u_new = u_new/norm(u_new);
        if norm(u_new - u) < 0.001
            break;
        end
        u = u_new;
    end
    V(:,i) = u_new;
end

[sigma] = eig(A_T*A, "matrix");
sigma = sqrt(sigma);
sigma = trimdata(sigma, [50 50]);

U = [];
for i = 1:50
    U(:,i) = A*V(:,i)/sigma(i,i);
end
```

```

figure(11)
spy(sigma)
grid
figure(12)
spy(U)
grid
figure(13)
spy(V')
grid

%
% PART 2: SVD Efficacy for 50 columns of U, V
%
SVD_space = numel(sigma) + numel(U) + numel(V);
A_space = numel(A);
space_efficiency = SVD_space/A_space;
space_efficiency

SVD_nzs = nnz(sigma) + nnz(U) + nnz(V);
A_nzs = nnz(A);
nnz_efficiency = SVD_nzs/A_nzs;
nnz_efficiency

%
% PART 3: Image/Contour Map
%
figure(1)
surf(dataLong,dataLat,((U*sigma*V')/1000)', 'EdgeColor', 'none')
colorbar
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD n=50)');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');
zlabel('Elevation (km)');
grid

figure(2)
Levels = [-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11];
contour(dataLong,dataLat,(U*sigma*V'/1000)', Levels)
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD n=50)');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');
cb = contourbar("eastoutside");
cb.XLabel.String = "Elevation (km)";
grid

%
% Part 4: U V column counts?

```

```

%
%
% U, V with 10 columns
%
V10 = zeros(1440, 10);

n = 1440;
u = randn(n,1);
u = u / norm(u);

for i = 1:10
    while 1
        u_new = A_T*A*u;
        summation = 0;
        for j = 1:i
            summation = summation + (u_new'*V10(:,j))*V10(:,j);
        end
        u_new = u_new - summation;
        u_new = u_new/norm(u_new);
        if norm(u_new - u) < 0.001
            break;
        end
        u = u_new;
    end
    V10(:,i) = u_new';
end

[sigma10] = eig(A_T*A, "matrix");

sigma10 = sqrt(sigma10);
sigma10 = trimdata(sigma,[10,10]);

U10 = [];
for i = 1:10
    U10(:,i) = A*V10(:,i)/sigma10(i,i);
end

figure(3)
surf(dataLong,dataLat,((U10*sigma10*V10')/1000)', 'EdgeColor', 'none')
colorbar
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD n=10)');
xlabel('Longitude');
ylabel('Latitude');
zlabel('Elevation (km)');
grid

```

```

figure(4)
Levels = [-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11];
contour(dataLong,dataLat,(U10*sigma10*V10'/1000)', Levels)
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD n=10)');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');
cb = contourcbar("eastoutside");
cb.XLabel.String = "Elevation (km)";
grid

%
% U, V with 100 columns
%
V100 = zeros(1440, 100);

n = 1440;
u = randn(n,1);
u = u / norm(u);

for i = 1:100
    while 1
        u_new = A_T*A*u;
        summation = 0;
        for j = 1:i
            summation = summation + (u_new'*V100(:,j))*V100(:,j);
        end
        u_new = u_new - summation;
        u_new = u_new/norm(u_new);
        if norm(u_new - u) < 0.001
            break;
        end
        u = u_new;
    end
    V100(:,i) = u_new';
end

[ sigma100 ] = eig(A_T*A, "matrix");

sigma100 = sqrt(sigma100);
sigma100 = trimdata(sigma100,[100 100]);

U100 = [];
for i = 1:100
    U100(:,i) = A*V100(:,i)/sigma100(i,i);
end

figure(5)

```

```

surf(dataLong,dataLat,((U100*sigma100*V100')/1000)','EdgeColor','none')
colorbar
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD (n=100)');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');
zlabel('Elevation (km)');
grid

figure(6)
Levels = [-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11];
contour(dataLong,dataLat,(U100*sigma100*V100'/1000)', Levels)
title('Mariana Depth along Latitude and Longitude, (Incomplete SVD n=100)');
xlabel('Longitude (degrees)');
ylabel('Latitude (degrees)');
cb = contourcbar("eastoutside");
cb.XLabel.String = "Elevation (km)";
grid

```