APPM 2360 - Intro Diff Eq W/Lin Alg Fall 2024

Project 2 - Analyzing The Mariana Trench

Daniel Alemayehu, Eli Grundberg, Sam Meyn

November 12, 2024

Contents

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.

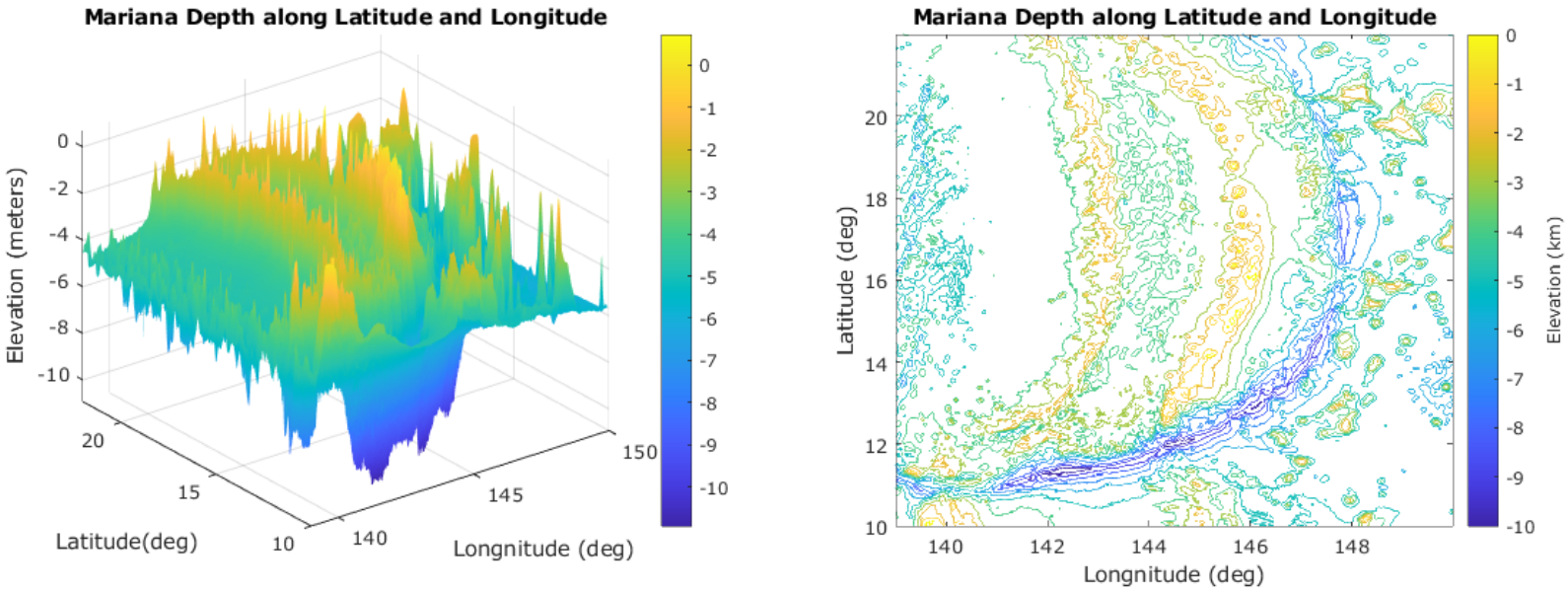


Figure 1: Code in Appendix A

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

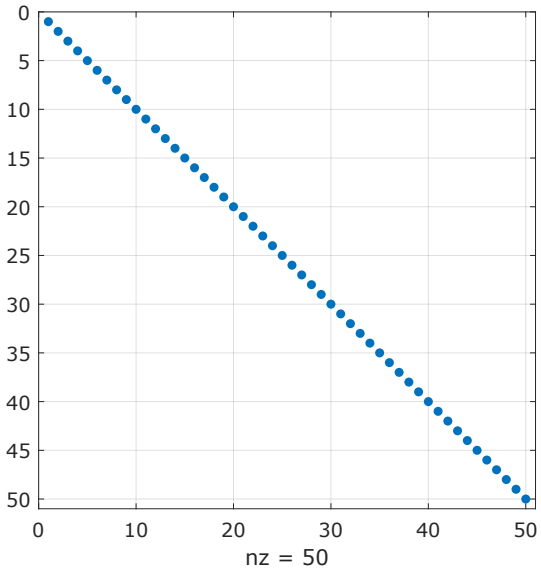
Appendix A).

3 Eigenvalue Computation

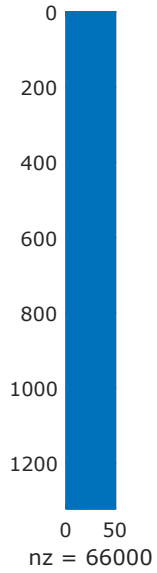
We can find an eigenvectors associated with our matrix using an algorithm outlined in (appendix w/ matlab code + algorithm overview). The reason this algorithm works can be explained as follows: We can also apply *Gram-Schmidt Orthogonalization* to find eigenvectors as well which we'll for for the 50 largest eigenvectors (code appendix).

4 SVD Incomplete Decomposition

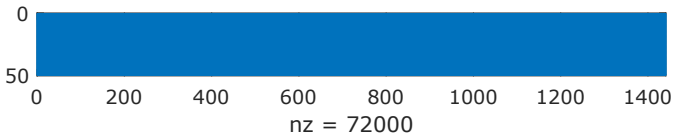
After completing our Incomplete SVD Decomposition (where U, V have 50 columns) our spy output looks like the following:



(a) $\mathbf{\Sigma}$ Matrix



(b) $\mathbf{U}$ Matrix



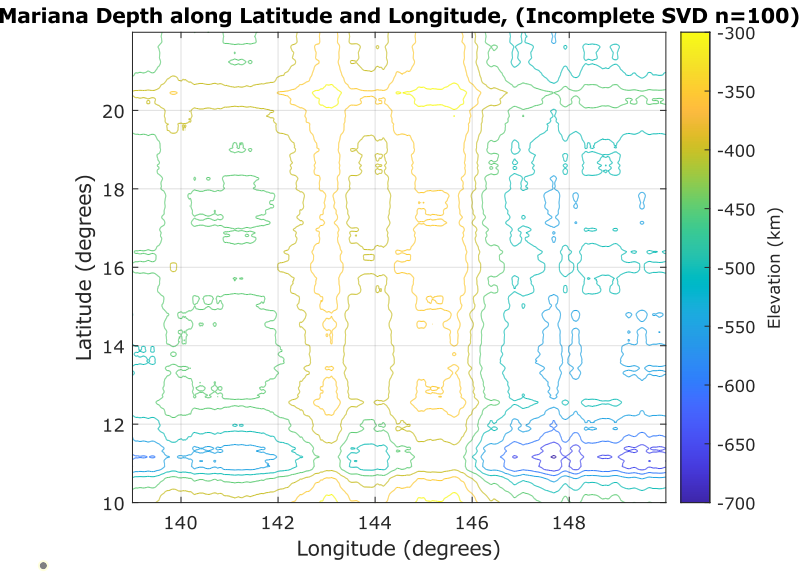
(c) $\mathbf{V^T}$ Matrix

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.

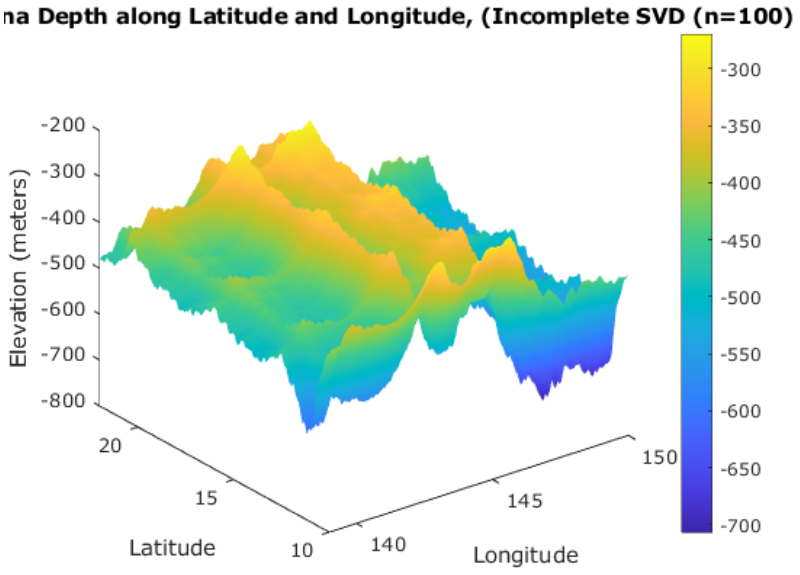
- For the number of total entries, our matrix has 7.39% of the number of total matrix entries as matrix $\mathbf{A}$.

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

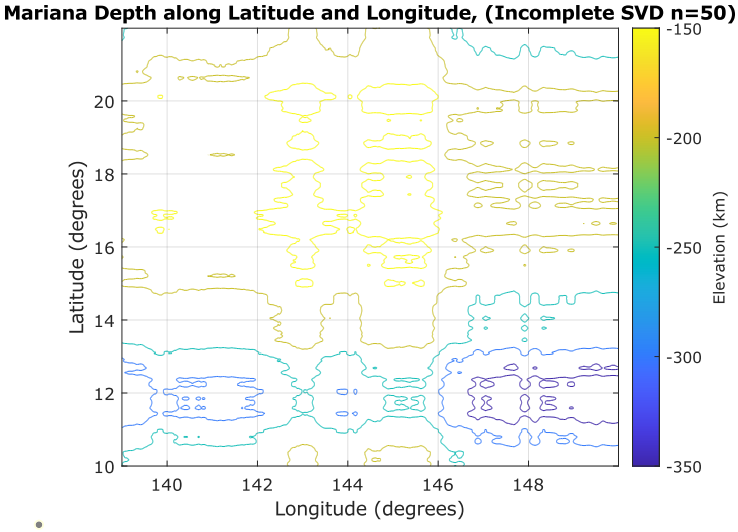
We've made plots of several SVD decompositions with varying column counts (10, 50, 100) which appear below.



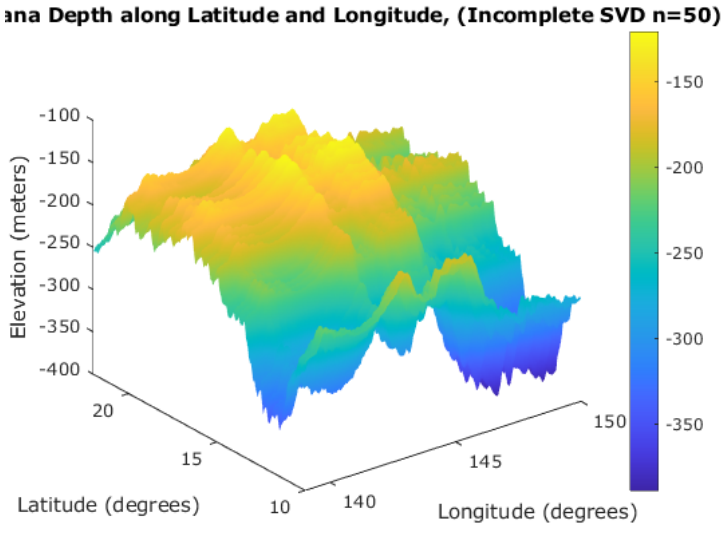
(a) $\mathbf{\Sigma}$ Matrix



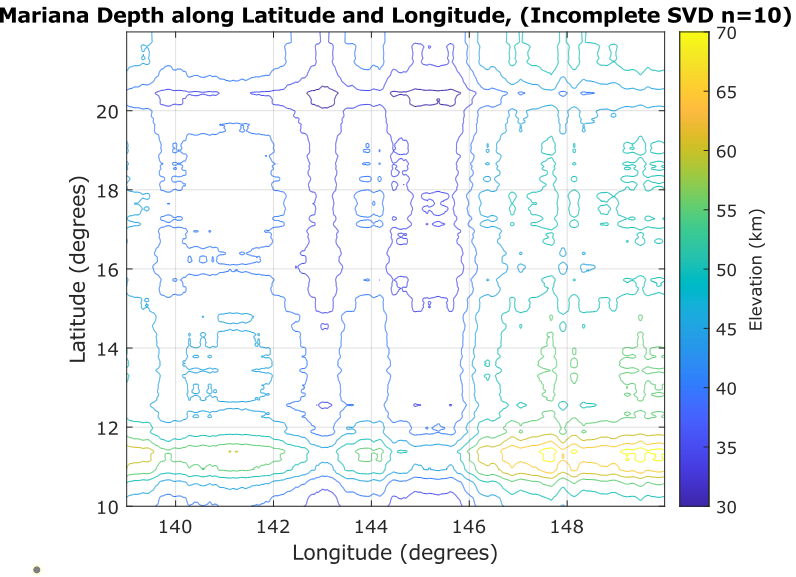
(b) $\mathbf{\Sigma}$ Matrix



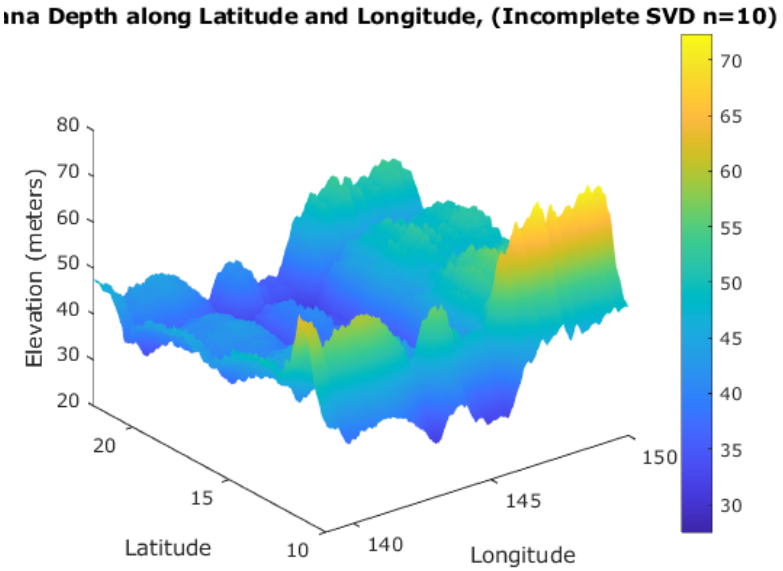
(a) $\mathbf{\Sigma}$ Matrix



(b) $\mathbf{\Sigma}$ Matrix



(a) $\mathbf{\Sigma}$ Matrix



(b) $\mathbf{\Sigma}$ Matrix

These look approximate to the original data? The maxxes and mins appear below.

- Lorem Ipsum

5 Conclusion

I have no idea what to put here...

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('Longnitude (deg)');
ylabel('Latitude (deg)');
cb = contourcbar("eastoutside");
cb.XLabel.String = "Elevation (km)";

figure(2)

surf(Y,X,Zkm, 'EdgeColor','none')
title('Mariana Depth along Latitude and Longitude');
xlabel('Longnitude (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
```