Contents


```
clc
close all
clear all
```

```
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('                 Geoid Modelling Project by Ravi Prakash Kumar and Abhishekk Anand')
disp('            Under the Supervision Of  Dr. Balaji Devaraju , Digvijay Singh and Arnab Laha')
disp('---------------------------------(Computation In Progress)-------------------------------')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                 Geoid Modelling Project by Ravi Prakash Kumar and Abhishekk Anand
            Under the Supervision Of  Dr. Balaji Devaraju , Digvijay Singh and Arnab Laha
---------------------------------(Computation In Progress)-------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Study area:

Gravity for the Redefinition of the American Vertical Datum : Grav-D Project Gravity data observed by the aircraft used during the project for Block_CN03

```
% Block Extent: 40 to 43 degree North latitude && 96 to 102 degree West longitude
% Region Name: Nebraska, USA
% Data collection period: From 7-2014 to 9-2014
% Data link: https://www.ngs.noaa.gov/GRAV-D/data_cn03.shtml
```

Reading the airborne gravity data of BLOCK CN03

Importing and reading data

```
Raw_Data = importdata('CN03.txt');

% Extracting latitudes from raw data
latitude_CN03 = Raw_Data.data(:,2);

% Extracting longitudes from raw data
longitude_CN03 = Raw_Data.data(:,3);

% Extracting Ellipsoidal heights of aeroplane from raw data
Ellht_cn03_NOAA = Raw_Data.data(:,4);

% Extracting observed gravity data from raw data
gravity_CN03_NOAA = Raw_Data.data(:,5);
```

Masking the airborne gravity data of BLOCK CN03 in the region of extent 40-41N and 97-98W

```
% Latitude of masked Airborne Gravity data
latnoaa = [];

% Longitude of masked Airborne Gravity data
lonnoaa = [];

% Observed gravity of masked Airborne Gravity data
g_NOAA = [];

% Orthometric Height of Aeroplane of masked Airborne Gravity data
h_aircraft1 = [];

for i = 1:length(latitude_CN03)
    a = latitude_CN03(i);
    b = longitude_CN03(i);
    if (a>=40 & a<=41) & (b>=-98 & b<=-97)
        latnoaa(end+1) = a;
        lonnoaa(end+1) = b;
        g_NOAA(end+1) = gravity_CN03_NOAA(i);
        h_aircraft1(end+1) = Ellht_cn03_NOAA(i);
    end
end
```

Creating equispaced latitude longitude points using linspace function

Creating the meshgrid of latitude and longitude of NOAA data of CN03 block

```
[X_lat Y_lon] = meshgrid(double(linspace(40,41,100)),double(linspace(-98,-97,100)));
```

Calculating DEM for NOAA data observation points

SRTM DEM data:

Resolution: 1 arc-second for global coverage (~30 meters) Extent: 40 to 43 degree North latitude && 96 to 102 degree West longitude Region name: Nebraska, USA Data collection period: 2014 Number of tiles available: 40 Data link: https://earthexplorer.usgs.gov/

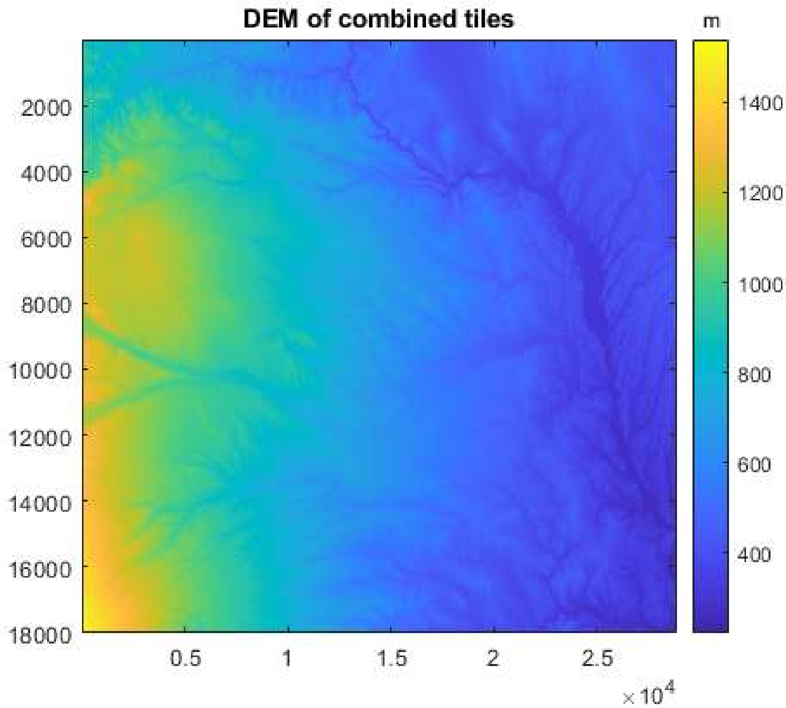
Reading the SRTM DEM data tiles

```
% Importing geotiff files for each tile for 1 degree difference
F1 = 'n39_w096_1arc_v3.tif';  % Extent: 39N-40N and 95W-96W
F2 = 'n39_w097_1arc_v3.tif';  % Extent: 39N-40N and 96W-97W
F3 = 'n39_w098_1arc_v3.tif';  % Extent: 39N-40N and 97W-98W
F4 = 'n39_w099_1arc_v3.tif';  % Extent: 39N-40N and 98W-99W
F5 = 'n39_w100_1arc_v3.tif';  % Extent: 39N-40N and 99W-100W
F6 = 'n39_w101_1arc_v3.tif';  % Extent: 39N-40N and 100W-101W
F7 = 'n39_w102_1arc_v3.tif';  % Extent: 39N-40N and 101W-102W
F8 = 'n39_w103_1arc_v3.tif';  % Extent: 39N-40N and 102W-103W
[t1,c1]= readgeoraster(F1);
[t2,c2] = readgeoraster(F2);
[t3,c3] = readgeoraster(F3);
[t4,c4] = readgeoraster(F4);
[t5,c5]= readgeoraster(F5);
[t6,c6] = readgeoraster(F6);
[t7,c7] = readgeoraster(F7);
[t8,c8] = readgeoraster(F8);

G1 = 'n40_w096_1arc_v3.tif';  % Extent: 40N-41N and 95W-96W
G2 = 'n40_w097_1arc_v3.tif';  % Extent: 40N-41N and 96W-97W
G3 = 'n40_w098_1arc_v3.tif';  % Extent: 40N-41N and 97W-98W
G4 = 'n40_w099_1arc_v3.tif';  % Extent: 40N-41N and 98W-99W
G5 = 'n40_w100_1arc_v3.tif';  % Extent: 40N-41N and 99W-100W
G6 = 'n40_w101_1arc_v3.tif';  % Extent: 40N-41N and 100W-101W
G7 = 'n40_w102_1arc_v3.tif';  % Extent: 40N-41N and 101W-102W
G8 = 'n40_w103_1arc_v3.tif';  % Extent: 40N-41N and 102W-103W
[u1,d1] = readgeoraster(G1);
[u2,d2] = readgeoraster(G2);
[u3,d3] = readgeoraster(G3);
[u4,d4] = readgeoraster(G4);
[u5,d5] = readgeoraster(G5);
[u6,d6] = readgeoraster(G6);
[u7,d7] = readgeoraster(G7);
[u8,d8] = readgeoraster(G8);

H1 = 'n41_w096_1arc_v3.tif';  % Extent: 41N-42N and 95W-96W
H2 = 'n41_w097_1arc_v3.tif';  % Extent: 41N-42N and 96W-97W
H3 = 'n41_w098_1arc_v3.tif';  % Extent: 41N-42N and 97W-98W
H4 = 'n41_w099_1arc_v3.tif';  % Extent: 41N-42N and 98W-99W
H5 = 'n41_w100_1arc_v3.tif';  % Extent: 41N-42N and 99W-100W
H6 = 'n41_w101_1arc_v3.tif';  % Extent: 41N-42N and 100W-101W
H7 = 'n41_w102_1arc_v3.tif';  % Extent: 41N-42N and 101W-102W
H8 = 'n41_w103_1arc_v3.tif';  % Extent: 41N-42N and 102W-103W
[v1,e1] = readgeoraster(H1);
[v2,e2] = readgeoraster(H2);
[v3,e3] = readgeoraster(H3);
[v4,e4] = readgeoraster(H4);
[v5,e5] = readgeoraster(H5);
[v6,e6] = readgeoraster(H6);
[v7,e7] = readgeoraster(H7);
[v8,e8] = readgeoraster(H8);

I1 = 'n42_w096_1arc_v3.tif';  % Extent: 42N-43N and 95W-96W
I2 = 'n42_w097_1arc_v3.tif';  % Extent: 42N-43N and 96W-97W
```

```
I3 = 'n42_w098_1arc_v3.tif';  % Extent: 42N-43N and 97W-98W
I4 = 'n42_w099_1arc_v3.tif';  % Extent: 42N-43N and 98W-99W
I5 = 'n42_w100_1arc_v3.tif';  % Extent: 42N-43N and 99W-100W
I6 = 'n42_w101_1arc_v3.tif';  % Extent: 42N-43N and 100W-101W
I7 = 'n42_w102_1arc_v3.tif';  % Extent: 42N-43N and 101W-102W
I8 = 'n42_w103_1arc_v3.tif';  % Extent: 42N-43N and 102W-103W
[u1_1,f1] = readgeoraster(I1);
[u1_2,f2] = readgeoraster(I2);
[u1_3,f3] = readgeoraster(I3);
[u1_4,f4] = readgeoraster(I4);
[u1_5,f5] = readgeoraster(I5);
[u1_6,f6] = readgeoraster(I6);
[u1_7,f7] = readgeoraster(I7);
[u1_8,f8] = readgeoraster(I8);

J1 = 'n43_w096_1arc_v3.tif';  % Extent: 43N-44N and 95W-96W
J2 = 'n43_w097_1arc_v3.tif';  % Extent: 43N-44N and 96W-97W
J3 = 'n43_w098_1arc_v3.tif';  % Extent: 43N-44N and 97W-98W
J4 = 'n43_w099_1arc_v3.tif';  % Extent: 43N-44N and 98W-99W
J5 = 'n43_w100_1arc_v3.tif';  % Extent: 43N-44N and 99W-100W
J6 = 'n43_w101_1arc_v3.tif';  % Extent: 43N-44N and 100W-101W
J7 = 'n43_w102_1arc_v3.tif';  % Extent: 43N-44N and 101W-102W
J8 = 'n43_w103_1arc_v3.tif';  % Extent: 43N-44N and 102W-103W
[v_1,g11] = readgeoraster(J1);
[v_2,g21] = readgeoraster(J2);
[v_3,g3] = readgeoraster(J3);
[v_4,g4] = readgeoraster(J4);
[v_5,g5] = readgeoraster(J5);
[v_6,g6] = readgeoraster(J6);
[v_7,g7] = readgeoraster(J7);
[v_8,g8] = readgeoraster(J8);

% Combining all the raw DEM tiles for getting DEM over throughout the extent of block CN03
DEM_raw_CN03 = [v_8 v_7 v_6 v_5 v_4 v_3 v_2 v_1;...
          u1_8 u1_7 u1_6 u1_5 u1_4 u1_3 u1_2 u1_1;...
          v8 v7 v6 v5 v4 v3 v2 v1;...
          u8 u7 u6 u5 u4 u3 u2 u1;...
          t8 t7 t6 t5 t4 t3 t2 t1];
```

Visualising DEM data compiled together

```
figure(1)
imagesc(DEM_raw_CN03), axis square
hcv = colorbar;
title(hcv,'m')
title('DEM of combined tiles')
```

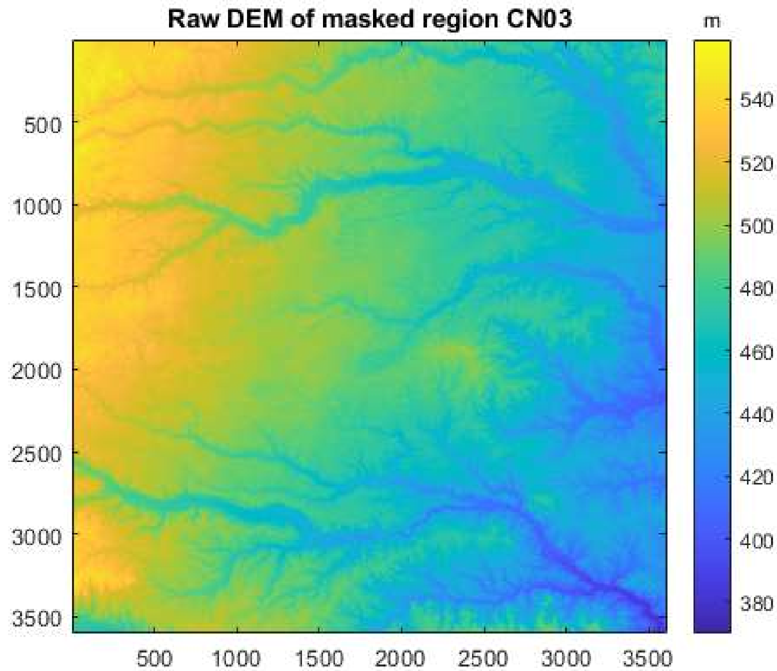


Used SRTM Data tile:

```
% Reading Geotiff file using geotiffread function
F11 = 'n40_w098_1arc_v3.tif';

t11 = geotiffread(F11);  %% It provides integer value of the DEM data
```

Visualising raw DEM data over masked region

```
figure(2)
imagesc(t11), axis square
hcv = colorbar;
title(hcv,'m')
title('Raw DEM of masked region CN03')

% **** Here we tried by readGeoraster function too **** It gives same integer values

% Since while interpolating the DEM data we need double values
% Converting DEM values into double
T11 = double(t11);

% Since DEM data is of size 3601x3601 so we need to create the grid of
% co-ordinates corresponding to those DEM points

% Creating meshgrid of SRTM DEM coordinates
a1 = double(linspace(40,41,3601));    % Since DEM Extent is 40-41 North Latitude
a2 = double(linspace(-98,-97,3601));  % Since DEM Extent is 97-98 West Longitude
[Lat_DEM,Lon_DEM] = ndgrid(a1,a2);    % Creating Meshgrid of Coordinates
```

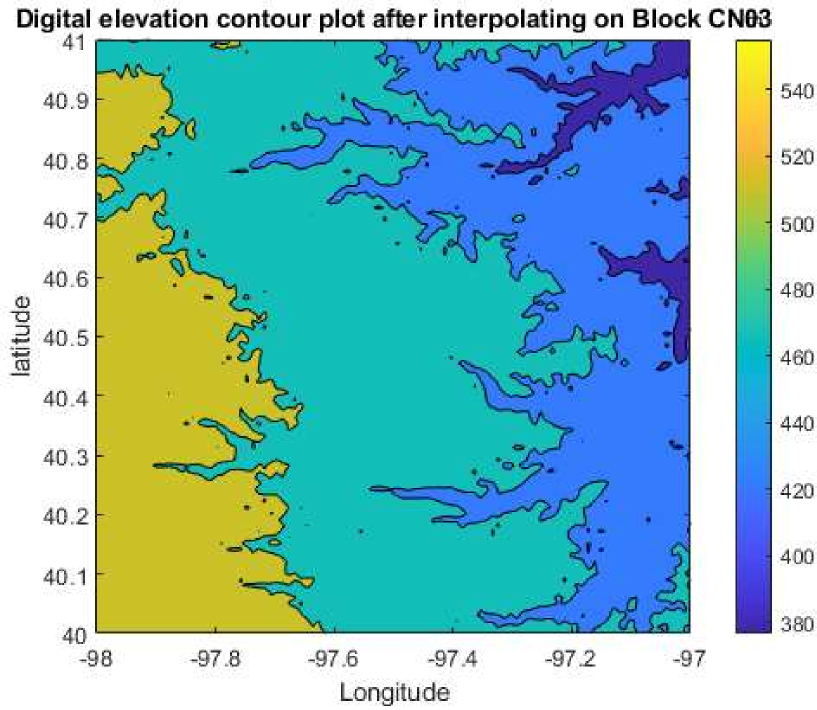


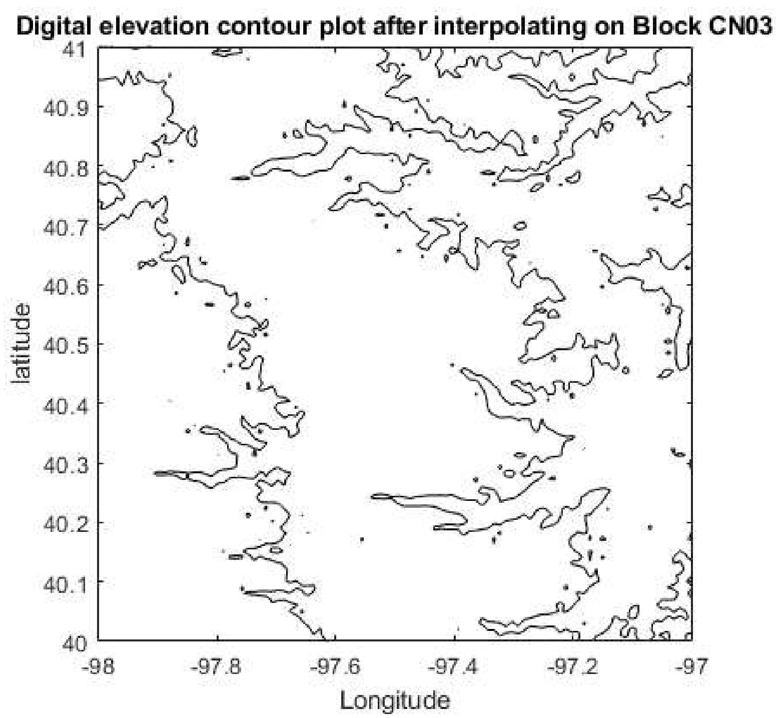
Interpolating DEM data over the latitude and longitude of the NOAA

```
% Since Observation points masked for the DEM extent comes inside the grid
% of DEM coordinates so griddata function will give best interpolation
% results on those observation points , For the best result we will use
% cubic method of interpolation as it is the best one in all other methods
% like 'nearest', 'natural' and 'v4'.

h_DEM_NOAA = double(griddata(Lon_DEM,Lat_DEM,T11,lonnoaa,latnoaa,'cubic'));

% Interpolating over created meshgrid of Latitude and Longitude of airborne gravity data
h_DEM_NOAA_mesh = double(griddata(Lat_DEM,Lon_DEM,T11,X_lat,Y_lon,'cubic'));
```

Visualising the DEM over our own block CN03

```
% Taking the colour levels as 3 we can visualize the contour plot of DEM
% corresponding to generated meshgrid points as:
figure(3)
contourf(Y_lon,X_lat,h_DEM_NOAA_mesh,3,'edgecolor','k')
axis square
title('Digital elevation contour plot after interpolating on Block CN03')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(h_DEM_NOAA_mesh)),max(max(h_DEM_NOAA_mesh))])

figure(4)

% Selected colour levels : 3 for separate visualization of data points

contour(Y_lon,X_lat,h_DEM_NOAA_mesh,3,'edgecolor','k')
axis square
title('Digital elevation contour plot after interpolating on Block CN03')
xlabel('Longitude')
ylabel('latitude')
```





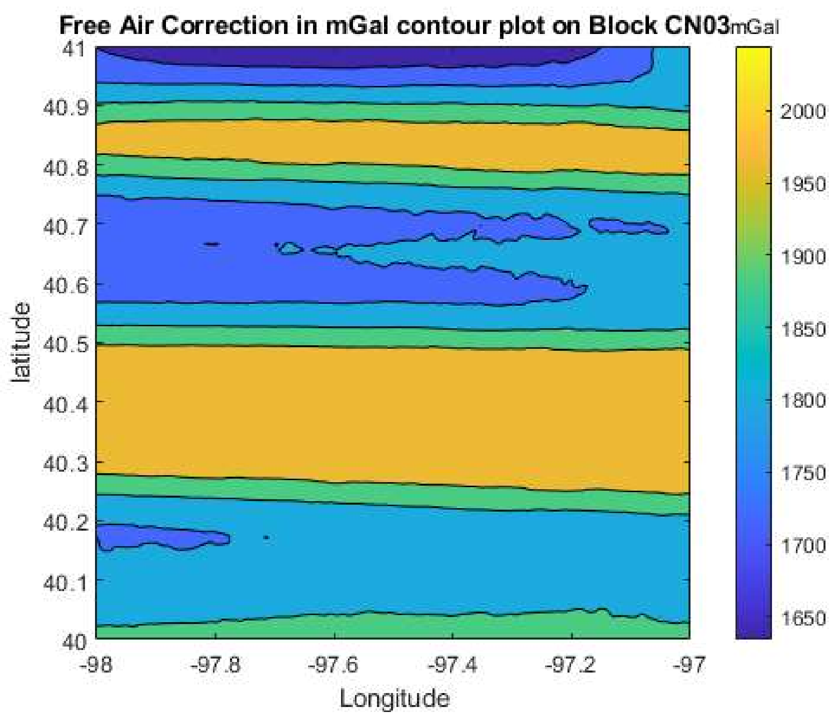
Calculation of Free Air Correstion(FAC) Over the latitude and the longitude of gravity Observation points of the NOAA

```
% Interpolating the ellipsoidal height of the observed gravity data over
% created latitude and longitude meshgrids of NOAA data sets

H_aircraft = double(griddata(latnoaa,lonnoaa,h_aircraft1,X_lat,Y_lon,'v4'));

% Finding the height of aircraft over topographical surface
% Since we need to provide correction for the air between the topography
% and the measurement points

% We can find this substracting the DEM from the ellipsoidal height
H_air = H_aircraft-h_DEM_NOAA_mesh;
```

```
% Free air correction can be computed as:
FAC_NOAA = 0.3086*H_air;
```

Plotting Free air correction

```
figure(5)

% Selected colour levels : 4 for separate visualization of data points

contourf(Y_lon,X_lat,FAC_NOAA,4,'edgecolor','k')
axis square
title('Free Air Correction in mGal contour plot on Block CN03')
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(FAC_NOAA)),max(max(FAC_NOAA))])
```



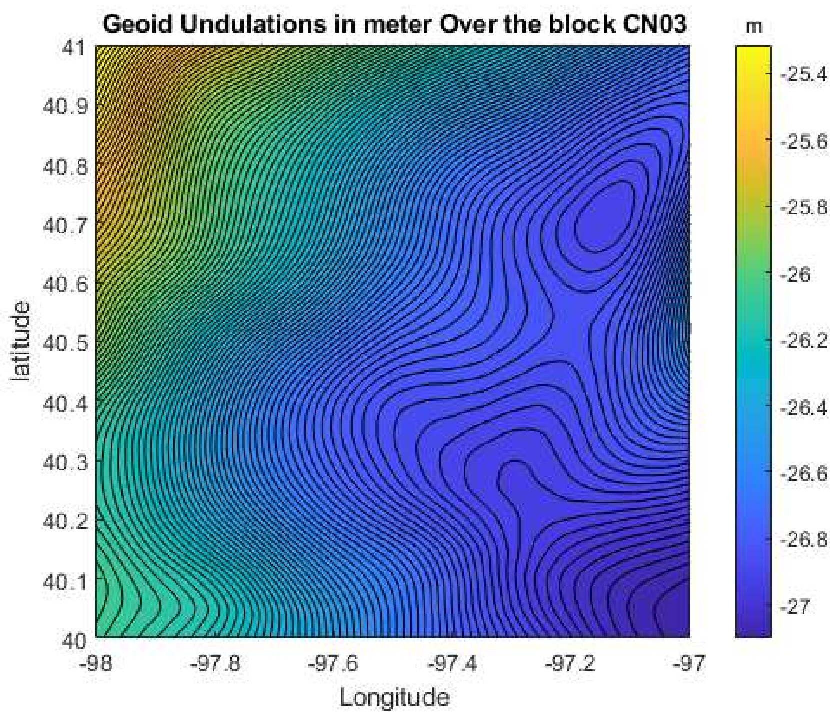
Calculation of Geoid Undulations Over the block CN03

```
% Geoid undulations of EGM2008
% Reading the EGM008 file
b = 'EGM2008_N.txt';
A1 = importdata(b);

% Extracting latitudes, longitudes and N from raw data
Lon_EGM08 = A1(:,1)-360; % Since Longitude is more than 180 degree
Lat_EGM08 = A1(:,2);
N_EGM08  = A1(:,3);

% Interpolating the Geoid undulation value of EGM2008 over meshgrid of Latitude and Longitude of airborne gravity data
N_NOAA_mesh = double(griddata(Lon_EGM08,Lat_EGM08,N_EGM08,Y_lon,X_lat,'cubic'));
```

Visualizing the Geoid undulation values interpolated over the CN03 block

```
figure(6)

% Selected colour levels : 100 for separate visualization of data points
```

```
contourf(Y_lon,X_lat,N_NOAA_mesh,100,'edgecolor','k')
axis square
title('Geoid Undulations in meter Over the block CN03')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_NOAA_mesh)),max(max(N_NOAA_mesh))])
```



Calculation of Bouguer Anomaly over the block CN03

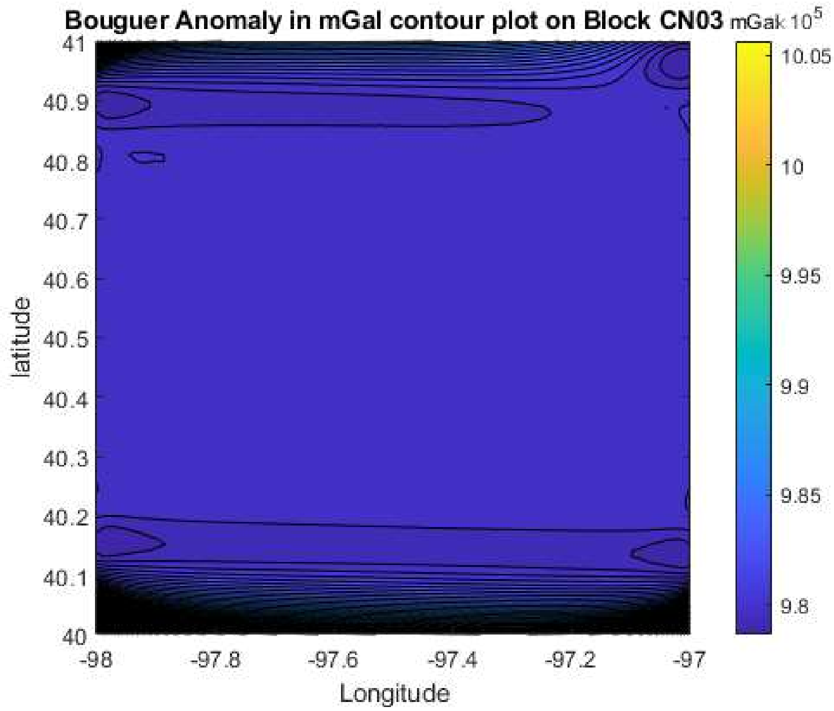
Height for calculating Bouguer anomaly

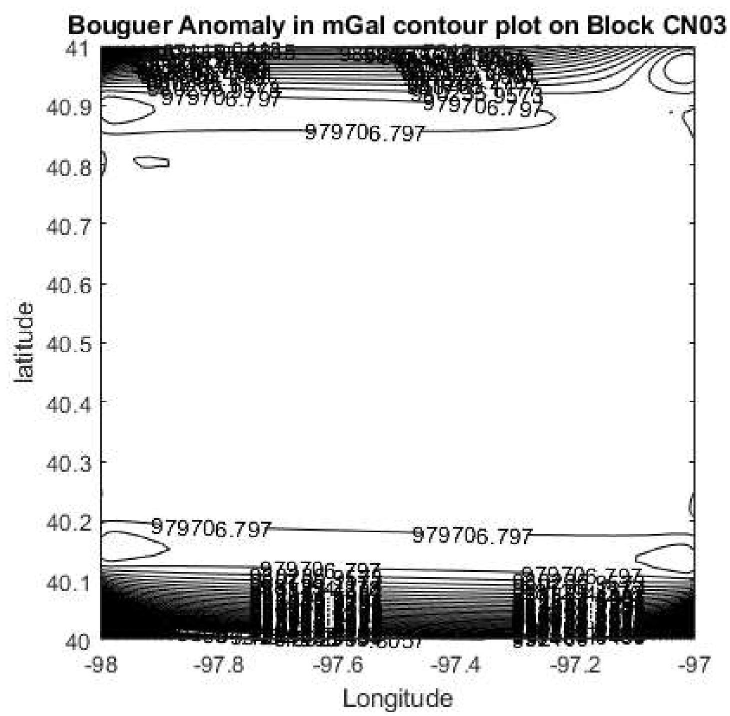
```
H_bg = h_DEM_NOAA_mesh - N_NOAA_mesh;

% Calculating Bouguer Anomaly in mGal using standard formula
G_NOAA_mesh = double(griddata(lonnoaa,latnoaa,g_NOAA,Y_lon,X_lat,'v4'));
Bg_NOAA_mesh = G_NOAA_mesh - 0.1119*H_bg + FAC_NOAA;
```

Visualizing the Bouguer Anomaly values interpolated over the CN03 block

```
figure(7)

% Selected colour levels : 50 for separate visualization of data points

contourf(Y_lon,X_lat,Bg_NOAA_mesh,50,'edgecolor','k')
axis square
title('Bouguer Anomaly in mGal contour plot on Block CN03')
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(Bg_NOAA_mesh)),max(max(Bg_NOAA_mesh))])

figure(8)

% Selected colour levels : 50 for separate visualization of data points

contour(Y_lon,X_lat,Bg_NOAA_mesh,50,'edgecolor','k','ShowText','on')
```

```
axis square
title('Bouguer Anomaly in mGal contour plot on Block CN03')
xlabel('Longitude')
ylabel('latitude')
```





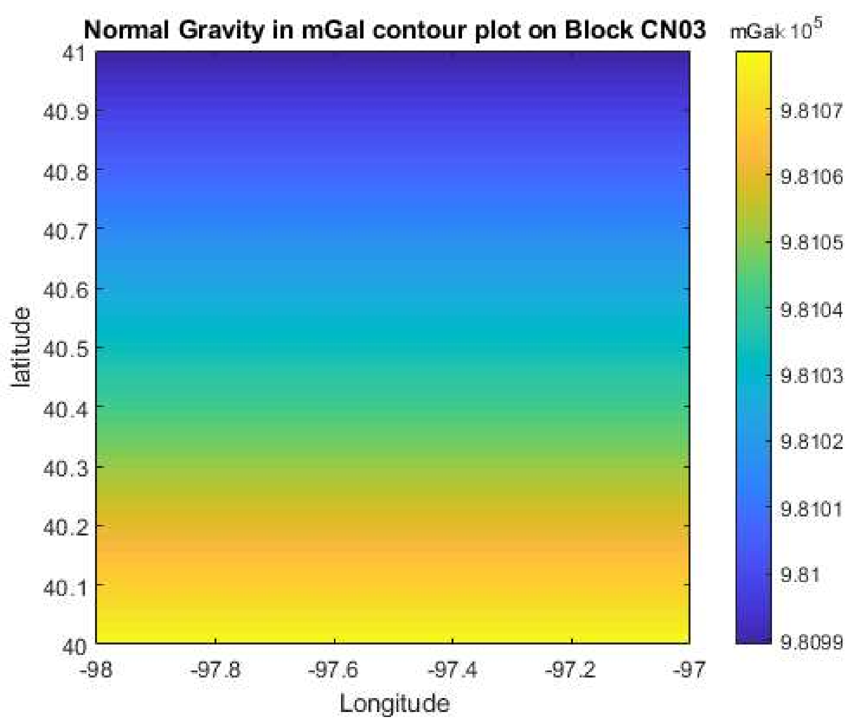
Calculating normal gravity value(gamma) in mGal on CN03 block

```
% WGS84 Parameters for calculating the normal gravity value(gamma)

a = 6378137 ;             % In meter
b = 6356752.3142 ;        % In meter
g_equator = 983218.49378; % Normal gravity value at the equatorial point
g_pole = 978032.7;        % Normal gravity value at the polar point
```

```
% Calculating the normal gravity value(gamma)
% Using the standard formula as:

g11 = (a*g_equator*(cosd(X_lat)).^2+b*g_pole*(sind(X_lat)).^2);
g21 = sqrt(a^2*(cosd(X_lat)).^2+b^2*(sind(X_lat)).^2);
gama = g11./g21;

g_11 = (a*g_equator*(cosd(latnoaa)).^2+b*g_pole*(sind(latnoaa)).^2);
g_21 = sqrt(a^2*(cosd(latnoaa)).^2+b^2*(sind(latnoaa)).^2);
gama1 = g_11./g_21;
R = a;
c = R./(2*gama1);
```

Visualizing the Normal Gravity values interpolated over the CN03 block

```
figure(9)

% Selected colour levels : 50 for separate visualization of data points

contourf(Y_lon,X_lat,gama,50,'edgecolor','none')
axis square
title('Normal Gravity in mGal contour plot on Block CN03')
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(gama)),max(max(gama))])
```



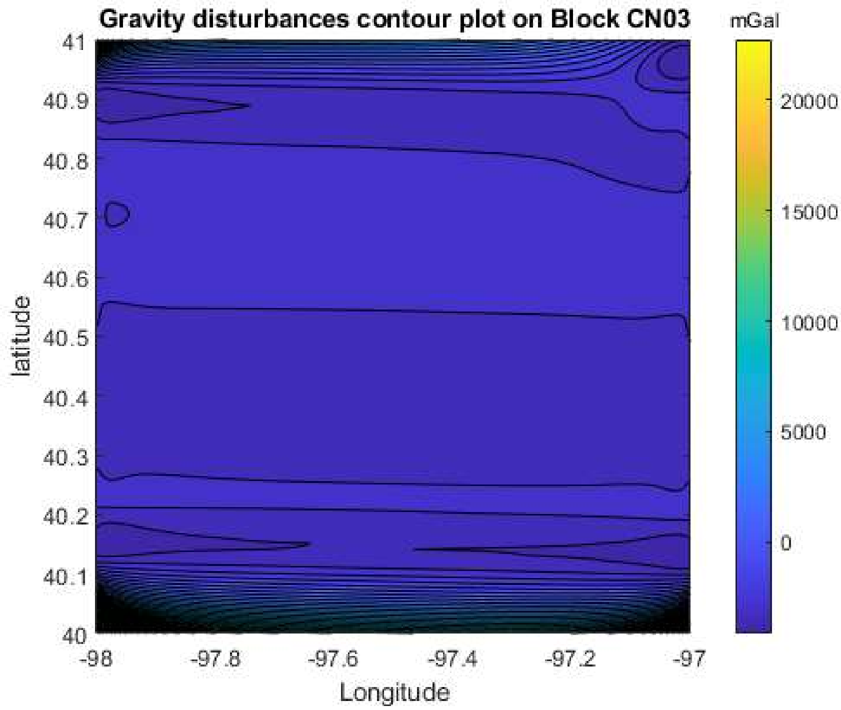
Gravity disturbances on CN03 block

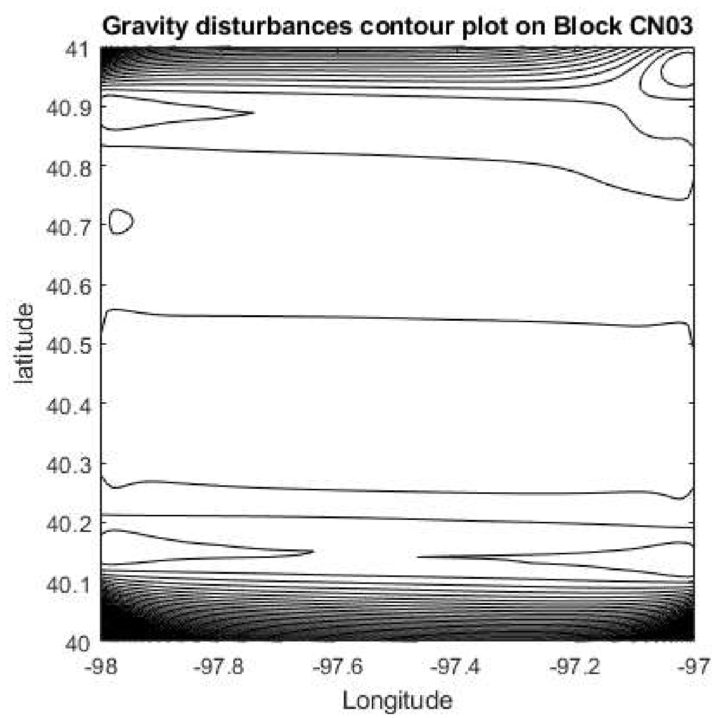
```
% Calculating the gravity disturbances
T_NOAA_mesh = G_NOAA_mesh - gama;
```

Visualizing the Gravity disturbances values interpolated over the CN03 block

```
figure(10)

% Selected colour levels : 40 for separate visualization of data points
```

```
contourf(Y_lon,X_lat,T_NOAA_mesh,40,'edgecolor','k')
axis square
title('Gravity disturbances contour plot on Block CN03')
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(T_NOAA_mesh)),max(max(T_NOAA_mesh))])

figure(11)

% Selected colour levels : 40 for separate visualization of data points

contour(Y_lon,X_lat,T_NOAA_mesh,40,'edgecolor','k')
axis square
title('Gravity disturbances contour plot on Block CN03')
xlabel('Longitude')
ylabel('latitude')
```





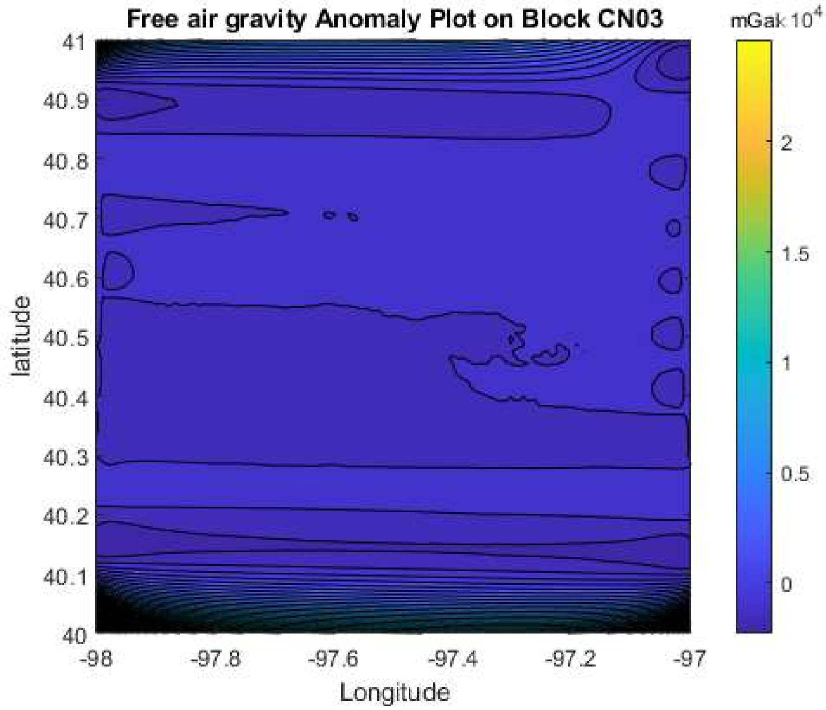
Calculating the free air gravity anomalies in mGal on block CN03

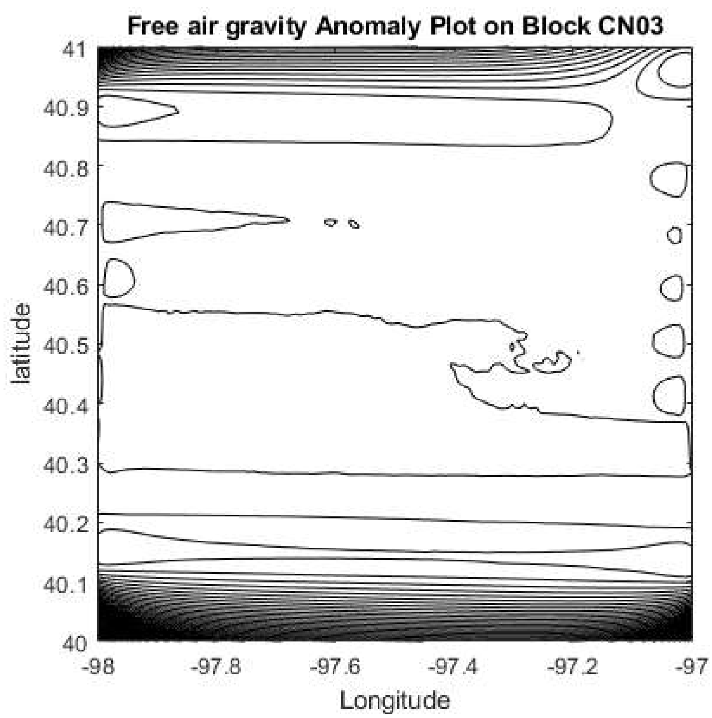
```
FDg_NOAA_mesh = G_NOAA_mesh + FAC_NOAA - gama;
```

Visualizing the free air gravity anomalies values interpolated over the CN03 block

```
figure(12)

% Selected colour levels : 40 for separate visualization of data points

contourf(Y_lon,X_lat,FDg_NOAA_mesh,40,'edgecolor','k')
axis square
title('Free air gravity Anomaly Plot on Block CN03')
```

```
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(FDg_NOAA_mesh)),max(max(FDg_NOAA_mesh))])

figure(13)

% Selected colour levels : 40 for separate visualization of data points

contour(Y_lon,X_lat,FDg_NOAA_mesh,40,'edgecolor','k')
axis square
title('Free air gravity Anomaly Plot on Block CN03')
xlabel('Longitude')
ylabel('latitude')
```





Adding bundles for the spherical harmonic coefficients Cnm and Ylm calculations

```
addpath(genpath('D:\Physical_Geodesy\Geoid_Modelling\Software_Ellman_2004\SHbundle-master'));
addpath(genpath('D:\Physical_Geodesy\Labs\Lab_05\uberall-master'));
addpath(genpath('D:\Physical_Geodesy\Labs\Lab_05\gracebundle-master'));
```

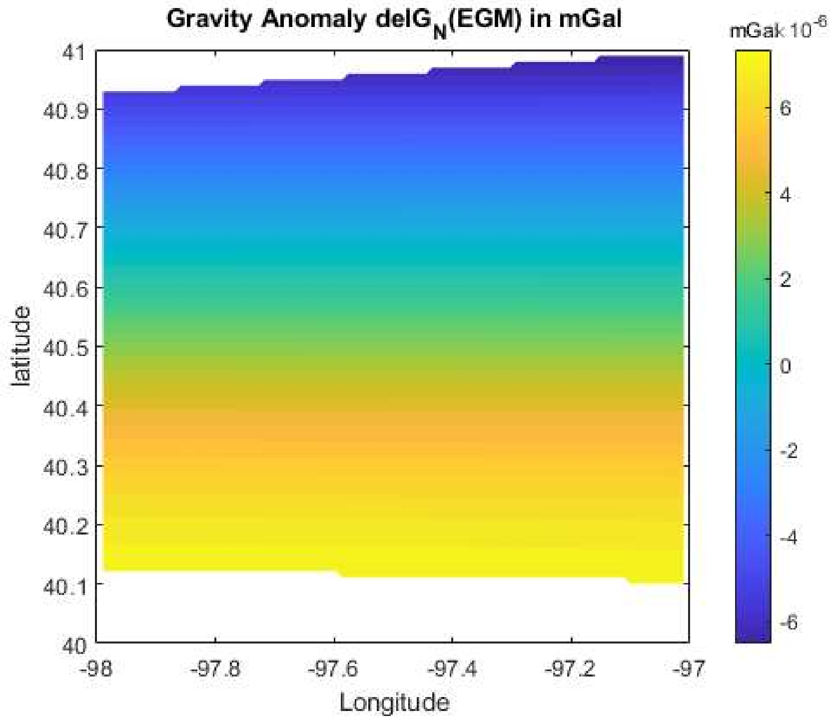
Finding the gravity anomaly values delG_N(EGM)

```
% Importing GGM data for degree and order and Cnm
load GGM01S.prn
mat = GGM01S(:,1:4);

% Using standard formula calculated as in the lab document we get:
```

```
% As we increase the degree gravity anomaly will decrease as we can see by
% changing the value of n

% Therefore we need to take the maximum degree to get the less gravity
% anomaly values:  n = 120

% Gravity Anomaly is the value which is difference between our observed
% gravity values and predicted by the model GGM01S

delgn_NOAA = [];
for i = 1:length(lonnoaa)                           % Data range
    f_nm = 0;
    n = 120;                                        % Maximum degree given in GGM data
    for m = 0:n                                   % Since order varies from 0 to n
        a  = 6378137;                               % Semimajor axis of the WGS84 reference ellipsoid
        gm = 3986005*10^8;                          % Geocentric gravitational constant of the earth
        r  = 6400000;                               % Geocentric radius of the earth
        C1 = (gm/(a.^2));
        C2 = (a/r).^(n+2);
        C3 = (n-1);
        Ct2  = C1*C2*C3;

        % Cnm : Fully normalised spherical harmonic coefficient of
        % disturbing potential from the normal gravity field refering to
        % the radius of a

        C_nm = get_coeff(mat,n,m);                  % Getting the Cnm coefficient from GGM files
        P_nm = plm(n,m,latnoaa(i)*pi/180);          % Calculating Associated legendre polynomials

        % Ynm : Fully normalised surface spherical harmonic

        Y_nm  = P_nm*cosd(m*lonnoaa(i)*pi/180);
        f_nm  = f_nm + Ct2*C_nm*Y_nm;
    end
    delgn_NOAA(end+1)= f_nm;
end

dgN = double(griddata(latnoaa,lonnoaa,delgn_NOAA,X_lat,Y_lon,'cubic'));
```

Visualizing the Gravity anomalies values interpolated over the CN03 block

```
figure(14)

% Selected colour levels : 40 for separate visualization of data points

contourf(Y_lon,X_lat,dgN,40,'edgecolor','none')
axis square
title('Gravity Anomaly delG_N(EGM) in mGal')
hcv = colorbar;
title(hcv,'mGal')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(dgN)),max(max(dgN))])
```



Finding Q_NLsi : Molodenskii's truncation coefficient

```
% Input data
% PSI, Integration radius for geoid determination, 0.2
% Max_exp,Maximum degree of series expansion, e.g. M_max=121
% L_max, Maximum modification degree, e.g. L_max = 70

% Output data
% ENK, Paul's coeff. are presented in the matrix with dimens. (Max_exp-1)x(L_max-1)
% QN, Molodenskii's trunc. coeff. are presented as a vector of (Max_exp-1) elements

% Passing 120 as maximum degree
[ENK,QN]=trunc_coeff_3(0.2,120,70);
```

```
--------Computation in progress .......     ---
```

Finding modified Stoke's Kernel(SLG_bias)

```
% Input data

% PSI, Integration radius for geoid determination
% L_max, Maximum modification degree, e.g. L_max = 70
% M_max, Upper limit of the used geopotential coefficients, e.g. M_max = 70
% Sn_bias, Vector of biased LS modification parameters
C = 6*c./(10^5); % mGal
% Sn_k_unb, Matrix of unbiased LS modification parameters (columnwise, solved from T-SVD)
% Sn_k1_unb, Matrix of unbiased LS modification parameters (columnwise, solved from T-TLS)
% Sn_k_opt, Matrix of optimum LS modification parameters (columnwise, solved from T-SVD)
% Sn_k1_opt, Matrix of optimum LS modification parameters (columnwise, solved from T-TLS)

% For Sn and Bn coefficients use software by Ellmann ''LS_coeff_v2.m''
Sn_bias = importdata('Sn_bias.prn');
Sn_k_unb = importdata('Sn_k_unb.prn');
Sn_k_opt = importdata('Sn_k_opt.prn');
Sn_k1_unb = importdata('Sn_k1_unb.prn');
Sn_k1_opt = importdata('Sn_k1_opt.prn');
Bn_opt = importdata('Bn_opt.prn');

PSI = 0.2;
```

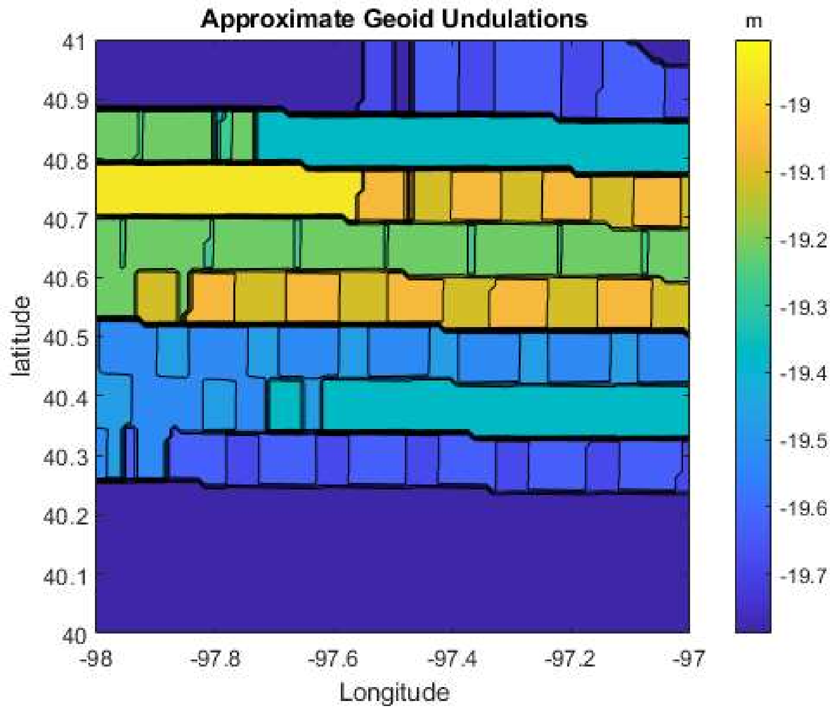
```
M_max =120;
L_max = M_max;
% [SL,SLG_bias] = modif_STOKES_kernel(PSI,M_max,L_max,Sn_bias,Sn_k_unb, Sn_k_opt,Sn_k1_unb,Sn_k1_opt);
```

Finding the approximate Geoid Undulation value N

```
delgn = [];
for i = 1:length(lonnoaa)
    sum = 0;
    for n = 2:120
        f_nm = 0;
        for m = 0:n
            a  = 6378137;
            gm = 3986005*10^8;
            r  = 6400000;
            C1 = (gm/(a.^2));
            C2 = (a/r).^(n+2);
            C3 = (n-1);
            Ct  = C1*C2*C3;
            C_nm = get_coeff(mat,n,m);
            P_nm = plm(n,m,latnoaa(i)*pi/180);
            Y_nm  = P_nm*cosd(m*lonnoaa(i)*pi/180);
            f_nm  = f_nm + Ct*C_nm*Y_nm;
        end
        sum = sum+f_nm;
    end
    delgn(end+1)= sum;
end
dg_N = double(griddata(latnoaa,lonnoaa,delgn,X_lat,Y_lon,'nearest'));

N_Bare = [];
for i = 1:length(lonnoaa)
    sum = 0;
    for n = 2:120
        N1 = C(i)*((2/(n-1))-QN(n-1)-Sn_k_opt(n-1)+Bn_opt(n-1))*dg_N(i);
        sum = sum+N1; %% Converting g_N in mGal
    end
    N_Bare(end+1) = sum;
end
N_Bar = griddata(latnoaa,lonnoaa,N_Bare,X_lat,Y_lon,'nearest');
```

Visualizing the approximate Geoid Undulation value N values interpolated over the CN03 block

```
% Selected colour levels : 16 for separate visualization of data points

figure(16)
contourf(Y_lon,X_lat,N_Bar,16,'edgecolor','k')
axis square
title('Approximate Geoid Undulations')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_Bar)),max(max(N_Bar))])
```



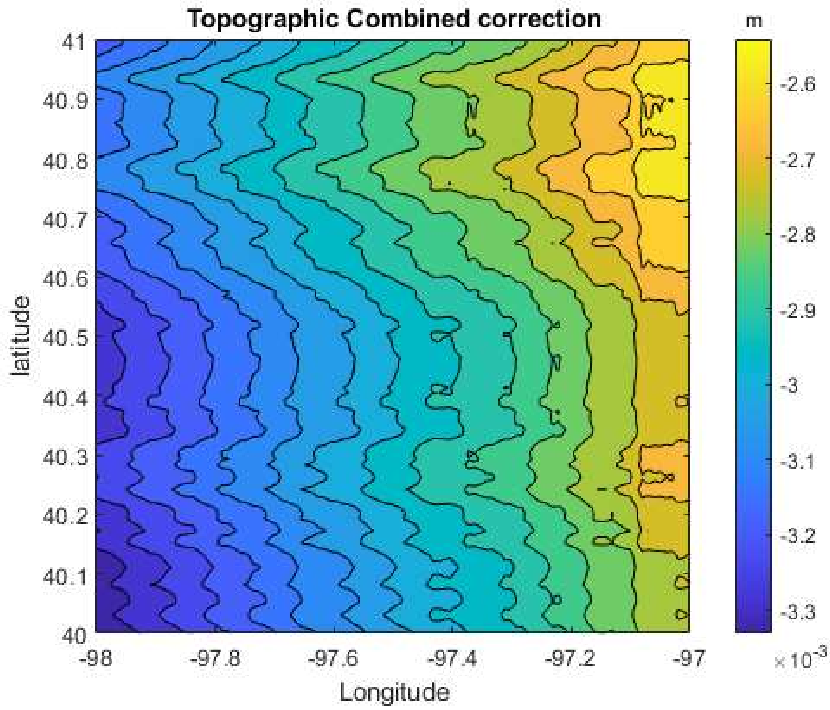
Providing Additive corrections

Topographic Combined correction

```
Rho = 2.65; % Crust density of earth in g/cc
% It is recommended to take Rho in g/cc
% gama in mGal
% H_bg : Orthometric height
dN_topo_Combined = ((-2*pi*(6.67*10^(-11))*Rho*H_bg^2)./gama)*10^5;
```

Visualizing the Topographic Combined correction values interpolated over the CN03 block

```
% Selected colour levels : 16 for separate visualization of data points

figure(17)
contourf(Y_lon,X_lat,dN_topo_Combined,16,'edgecolor','k')
axis square
title('Topographic Combined correction')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(dN_topo_Combined)),max(max(dN_topo_Combined))])
```

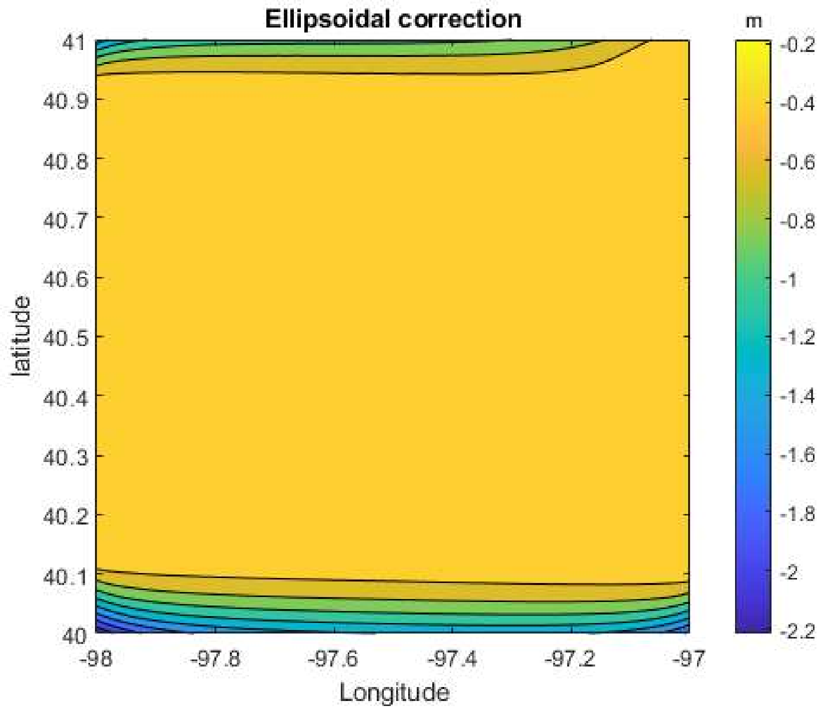


Ellipsoidal correction

```
% Integration cap PSI in radian
si_0 = 0.002;
% N_bar = Approximate geoid height
dN_Ellipsoidal = (si_0)*((0.12-0.38*(cosd(90-X_lat)).^2).*FDg_NOAA_mesh+0.17*N_Bar.*(sind(90-X_lat))^2);
```

Visualizing the Ellipsoidal correction values interpolated over the CN03 block

```
% Selected colour levels : 8 for separate visualization of data points

figure(18)
contourf(Y_lon,X_lat,dN_Ellipsoidal,8,'edgecolor','k')
axis square
title('Ellipsoidal correction')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(dN_Ellipsoidal)),max(max(dN_Ellipsoidal))])
```

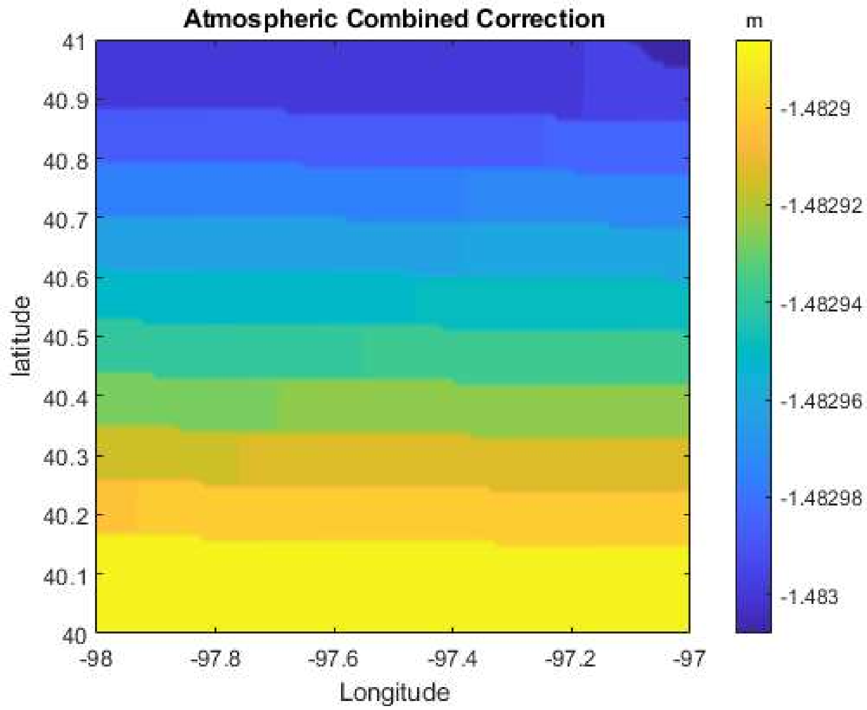


Atmospheric Combined Correction

```
dN_atm_Combined = [];
for i = 1:length(lonnoaa)
    sum = 0;
    for n = 2:120
        rho_0 = 1.23;
        Cg = (2*pi*r*rho_0/(gama1(i)));
        C2 = QN(n-1);
        CS = Sn_k1_opt(n-1);
        % As Refered to Abdalla 2011 paper to minimise the error they took
        % H_np = 0.00002 meter
        H_nP = 0.00002;
        C3 = 3*(n+1)/(2*n+1);
        C4 = 4/(n-1);
        Ceff = C3*C2+CS-C4;
        DN = Cg*Ceff*H_nP;
        sum = sum+DN;
    end
    dN_atm_Combined(end+1)= sum/10^5;
end
N_atm_Combined = double(griddata(latnoaa,lonnoaa,dN_atm_Combined,X_lat,Y_lon,'nearest'));
```

Visualizing the Atmospheric Combined correction values interpolated over the CN03 block

```
% Selected colour levels : 8 for separate visualization of data points
figure(19)
contourf(Y_lon,X_lat,N_atm_Combined,40,'edgecolor','none')
axis square
title('Atmospheric Combined Correction')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_atm_Combined)),max(max(N_atm_Combined))])
```

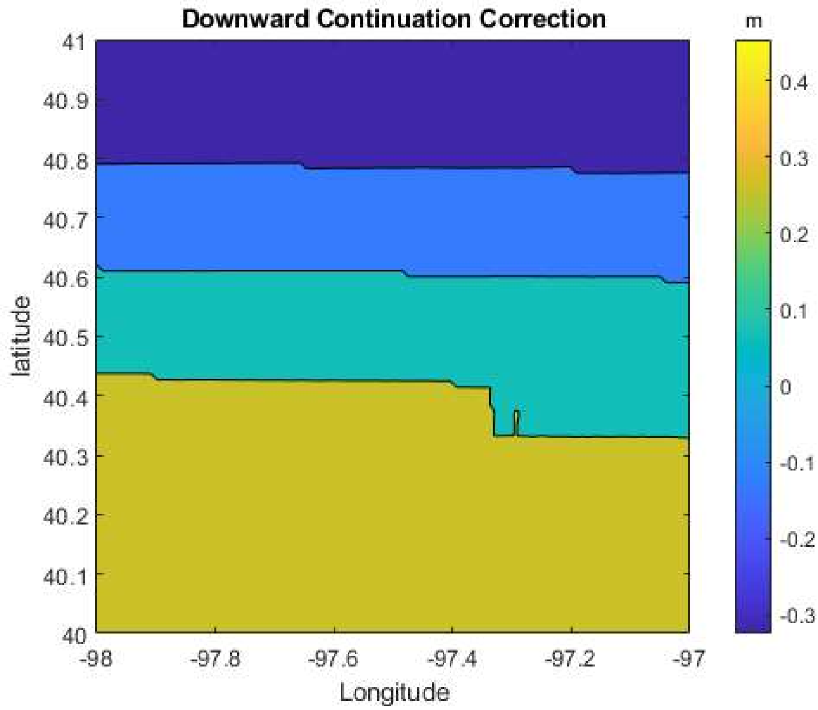


Downward Continuation Correction

```
DNW_Correct = [];
for i = 1:length(lonnoaa)
    sum1 = 0;
    for n = 2:120
        C1 = r/(2*gama1(i));
        C2 = QN(n-1);
        C3 = Sn_k1_opt(n-1);
        C4 = (r/(r+h_DEM_NOAA(i)))^(n+2)-1;
        C5 = delgn_NOAA(i);
        DN = C1*(C2+C3)*C4*C5;
        sum1 = sum1+DN;
    end
    DNW_Correct(end+1)= sum1;
end
DNW_Correction = double(griddata(latnoaa,lonnoaa,DNW_Correct,X_lat,Y_lon,'nearest'));
```

Visualizing the Downward Continuation correction values interpolated over the CN03 block

```
% Selected colour levels : 3 for separate visualization of data points
figure(20)
contourf(Y_lon,X_lat,DNW_Correction,3,'edgecolor','k')
axis square
title('Downward Continuation Correction')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(DNW_Correction)),max(max(DNW_Correction))])
```

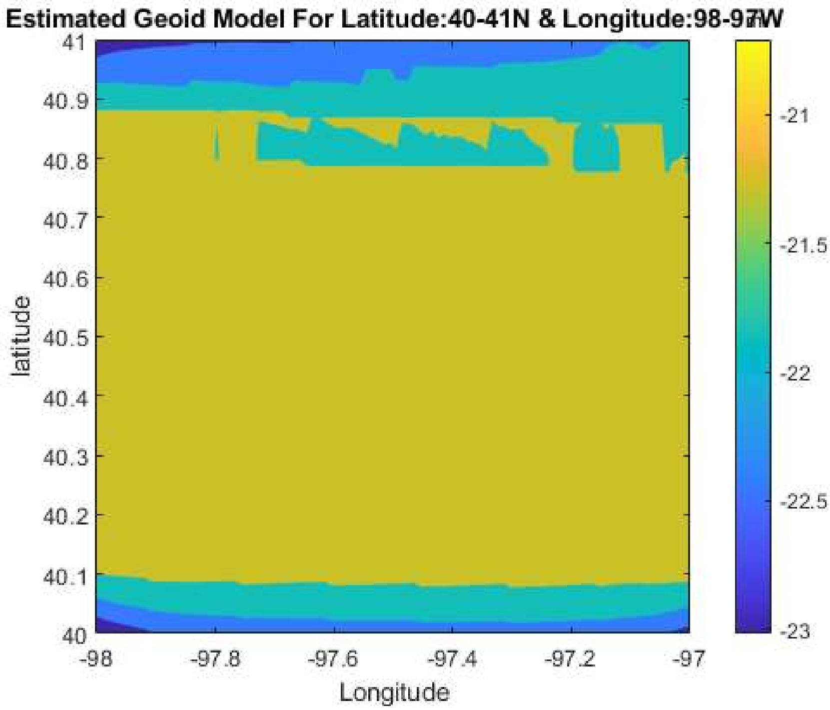


Estimated Geoid Model

```
N_cap1 = N_Bar + dN_topo_Combined + dN_Ellipsoidal + N_atm_Combined;
N_cap = N_cap1+DNW_Correction;
```

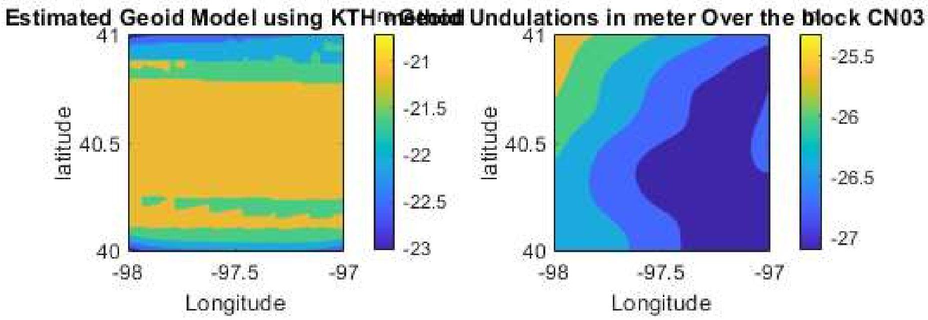
Visualizing the Downward Continuation correction values interpolated over the CN03 block

```
% Selected colour levels : 3 for separate visualization of data points
figure(21)
contourf(Y_lon,X_lat,N_cap,3,'edgecolor','none')
axis square
title('Estimated Geoid Model For Latitude:40-41N & Longitude:98-97W')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_cap)),max(max(N_cap))])
```



Visualizing the comparision over the CN03 block

```
% Selected colour levels : 3 for separate visualization of data points
figure(22)
subplot(1,2,1)
contourf(Y_lon,X_lat,N_cap,4,'edgecolor','none')
axis square
title('Estimated Geoid Model using KTH method')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_cap)),max(max(N_cap))])

subplot(1,2,2)
contourf(Y_lon,X_lat,N_NOAA_mesh,4,'edgecolor','none')
axis square
title('Geoid Undulations in meter Over the block CN03')
hcv = colorbar;
title(hcv,'m')
xlabel('Longitude')
ylabel('latitude')
caxis([min(min(N_NOAA_mesh)),max(max(N_NOAA_mesh))])
```


Estimated Geoid Model using KTH method
latitude
Longitude
Geoid Undulations in meter Over the block CN03
latitude
Longitude