Least Squares Collocation for Regional Gravimetric Quasigeoid Determination: A MATLAB Toolbox

Jack McCubbine, Neda Darbeheshti

June 26, 2024

1 Abstract

This paper presents an open-source tool for computing gravimetric geoids using gravity observations. Our objective is to establish a platform for analysis-ready gravity data, where algorithms and codes to work with gravity data are openly shared and enhanced. We introduce the initial release, which features a tilewise least-squares collocation method utilizing gravity anomaly observations. The analytical forms of spherical covariance function models for gravity anomaly and gravity gradient are presented, followed by a detailed description of the tilewise least squares collocation algorithm. Finally, we illustrate the functionality of the code with an example of gravity observations. The initial release comprises all MATLAB M-files and is publicly available at [insert link].

2 Introduction

Geoid determination stands as a fundamental task in geodesy, aiming to accurately model the Earth's gravitational field. Early initiatives, such as the introduction of GRAVSOFT (Forsberg and Tscherning, 2014), marked significant milestones in the discipline by providing open-source software for geoid determination. Building upon this legacy, our project follows a similar path, introducing a MATLAB-based code for regional geoid determination using least squares collocation.

This endeavor represents a modern iteration of GRAVSOFT, leveraging the capabilities of MATLAB and incorporating additional covariance models for gravity gradients. The use of least-squares collocation in this context gains significance, particularly with the growing emphasis on airborne gravity data in geodesy (McCubbine, 2016).

By sharing both the code and data, our objective is to facilitate collaboration and advancement towards achieving international geoid accuracy at the one-centimeter level. This paper outlines our approach and contributions towards this overarching goal.

3 Method

In summary, for computing a geoid from gravity anomalies, the process can be conceptualized as a sequence of "remove-predict-restore" operations, where a GGM and terrain effects are removed, a geoid is predicted, and then the effects are restored to obtain the final geoid model. This is achieved through the following steps:

1. **Remove:** Subtract the GGM and the terrain effects from the observed free-air gravity anomalies.

$$\Delta g_{res} = \Delta g - \Delta g_{GGM} - \delta g_{TE}$$

2. **Predict:** Perform a conversion to geoid using least squares collocation. This step involves predicting the geoid from the gravity anomalies.

$$\zeta_{res} = C_{\zeta_{res},\Delta g_{res}} \left(C_{\Delta g_{res},\Delta g_{res}}\right)^{-1} \Delta g_{res}$$

3. **Restore:** "Restore" the geoid effects of terrain and GGM. This step involves adding back the terrain and GGM effects to the computed geoid.

$$\zeta = \zeta_{res} + \zeta_{GGM} + \zeta_{TE}$$

4 Code Description

All the input files and parameters are defined in the main script `RunLSC.m`. Then `RunLSC.m` calls three respective functions:

- `importAndFormatData.m`: Imports and formats all input data, such as gravity anomalies.
- `computeTerrainEffect.m`: Prepares gravity anomalies for geoid computation, such as subtracting terrain corrections.
- `computeLSC.m`: Computes all the necessary steps for LSC, such as covariance computation.

All functions are in the `functions` folder. The hierarchy of the three main scripts and their related functions is listed in Table 1.

In addition to the main functions listed in Table 1, there are additional functions:

- Plot functions: These functions typically have a boolean parameter labeled as `true` or `false`, which enables or disables plotting.
- `haversine.m` function: Utilized in both `computeSphericalEmpiricalCovariance.m` and `interpolateCovarianceFunction.m` for computing covariance functions based on spherical distances.

<table>
<tr><th>Function Group</th><th>Function Name</th></tr>
<tr><td>importAndFormatData.m</td><td></td></tr>
<tr><td rowspan="2">computeTerrainEffect.m</td><td>computeTerrainCorrection.m
computePrismGravity.m
computeNagyFormula.m</td></tr>
<tr><td>filterDEM.m</td></tr>
<tr><td rowspan="6">computeGravimetryLSC.m
computeGradiometryLSC.m
computeGravimetryGradiometryLSC.m</td><td>computeCovarianceFunctionParameters.m
computeSphericalEmpiricalCovariance.m
fitEmpiricalCovariance.m</td></tr>
<tr><td>precomputeCovarianceFunction.m</td></tr>
<tr><td>interpolateCovarianceFunction.m</td></tr>
<tr><td>solveGravityLSCmatrix.m
solveGradientLSCmatrix.m
solveGravityGradientLSCmatrix.m</td></tr>
<tr><td>createGridWeights.m</td></tr>
<tr><td>comparetoLevellingData.m</td></tr>
</table>

Table 1: Function structure in `RunLSC.m`, the hierarchy is from left to right.

- `custom_grpstats.m`: A customized version of the MATLAB `grpstats` function used in `computeSphericalEmpiricalCovariance`.
- MATLAB `griddedInterpolant` function: Extensively used because all calculations are performed on a grid. Whenever a variable is created by `griddedInterpolant`, its name includes it in the variable name.

4.1 Import And Format Data Functions

`importAndFormatData.m` imports all the different datasets needed for LSC.

Here is a list of the datasets used in the computation of the quasigeoid:

- Gravity anomalies
- Gravity gradients
- Digital Elevation Model (DEM): A high-resolution DEM gridded at a resolution of 56 meters. It will be used to reconstruct the free-air anomalies from the gridded refined Bouguer anomalies, as described by Featherstone and Kirby (2000).
- The most contemporary and accurate global gravity model (GGM). It will be used to provide the reference long and medium-wavelength gravity and height anomalies in the remove-predict-restore stages of the processing.

- GPS/leveling data: These data will be used to assess the accuracy of the determined quasigeoid.

4.2 Topographic Effects Functions

The function `computeTerrainEffect.m` implements the "remove" part of the remove-predict-restore method. It calculates the topographic effects and subtracts them from the observed free-air gravity anomalies:

$$\Delta g_{res}^{obs} = \Delta g^{obs} - \Delta g_{GGM} - \delta g_{TE}$$

Topographic effects are divided into two components based on the DEMs used: long and full wavelengths. Initially, the visible topographic correction is computed using `computeTerrainCorrection.m` with the full DEM. Next, the smoothed long DEM topography is calculated using `filterDEM.m`, and it is subsequently added back.

$$\delta g_{TE} = \delta g_{TE}^{full} - \delta g_{TE}^{long}$$

4.2.1 `computeTerrainCorrection.m`

Function `computeTerrainCorrection.m` calculates the topographic corrections for gravity anomalies and gravity gradients. It utilizes the `computePrismGravity` and `computeNagyFormula.m` to compute the vertical component of the gravitational potential due to a rectangular prism at a point outside the prism. This calculation is based on Equation (8) in Nagy et al. (2000), which yields the full topographic effect for gravity:

$$\delta g_{TE}^{full} = |||x \log(y+r) + y \log(x+r) - z \arctan\left(\frac{xy}{zr}\right) |_{x_1}^{x_2}|_{y_1}^{y_2}|_{z_1}^{z_2}|$$

and Equation (24) for gravity gradient:

$$\delta gg_{TE}^{full} = ||| - \arctan\left(\frac{xy}{zr}\right) |_{x_1}^{x_2}|_{y_1}^{y_2}|_{z_1}^{z_2}|$$

where $r = \sqrt{x^2 + y^2 + z^2}$, except for a negative sign.

This function defines a half-minute differential element (`dy`) as required for the Nagy Prism formula computation.

4.2.2 `filterDEM.m`

`filterDEM` provides a filtered DEM by Fourier transform and low-pass filter mask. This filtered DEM is used to calculate long wavelength topographic effect:

$$\delta g_{TE}^{long} = 0.0419 \rho (DEM)^{long}$$

where $(DEM)^{long}$ is the long wavelength filtered DEM calculated by `filterDEM.m`, and ρ is the density.

4.3 Least Squares Collocation Functions

`computeGravimetryLSC.m` performs least squares prediction on observation data. It is a generic form that depends on the type of observation data available. There are three versions:

- `computeGravimetryLSC.m` - uses gravity observations.
- `computeGradiometryLSC.m` - uses gravity gradient observations.
- `computeGravimetryGradiometryLSC.m` - uses a combination of gravity and gravity gradient data.

LSC is performed on a grid of 0.5×0.5 degrees. For instance, in a 1×1-degree area, there are 9 blocks of LSC, each starting by computing covariances within the individual block. The calculation of covariances is a crucial aspect of LSC. The following section outlines the functions responsible for computing covariance matrices in the LSC.

4.3.1 `computeCovarianceFunctionParameters.m`

`computeCovarianceFunctionParameters.m` first calls `computeSphericalEmpiricalCovariance.m` to compute spherical empirical covariance functions of gravity anomalies and then calls `fitEmpiricalCovariance.m` to calculate parameters of covariance functions by fitting a spherical covariance function to the spherical empirical covariance function.

4.3.2 `computeSphericalEmpiricalCovariance.m`

This function computes an empirical covariance function of scalar quantities on a sphere by taking the mean of product-sums of samples of scalar values (here Δg) (Darbeheshti and Featherstone, 2009):

$$cov(\psi_j) = \frac{1}{N_j} \sum_{k,i}^{N_j} \Delta g(\phi_k, \lambda_k) \Delta g(\phi_i, \lambda_i)$$

where ψ_j is the Haversine distance and N_j is the number of pairs for each interval. The Haversine formula is used to calculate the great-circle distance between two points on a sphere given their longitudes and latitudes. The formula is as follows:

$$\psi = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

where:

- ψ is the distance between the two points along a great circle of the sphere.

- r is the radius of the sphere.
- ϕ_1 and ϕ_2 are the latitude of point 1 and latitude of point 2.
- λ_1 and λ_2 are the longitude of point 1 and longitude of point 2.

Please note that the formula assumes that the Earth is a perfect sphere, which is not entirely accurate. However, it provides a close approximation of the distance between two points on the Earth's surface. r is assumed to be 1 in our case.

4.3.3 `fitEmpiricalCovariance.m`

This function, `fitEmpiricalCovariance.m`, fits the spherical empirical covariances of gravity anomalies to the covariance function of gravity anomalies. We use the anomaly degree variance model given by equation (68) of Tscherning and Rapp (1974):

$\sigma_n(\Delta g, \Delta g) = \frac{A(n-1)}{(n-2)(n+B)}$

For the covariance function of gravity anomalies:

$$cov\left(\Delta g_{P,} \Delta g_Q\right) = \sum_{n=2}^{n_{\max}} \sigma_n(\Delta g, \Delta g) s^{(n+2)} P_n\left(\cos\psi\right)$$

where $s = \frac{R_B^2}{r_P r_Q}$, ψ is the spherical distance in radians, and P_n is the Legendre polynomial. r_P and r_Q are the geocentric radii to points P and Q, which are separated by ψ. R_B is the radius of the Bjerhammar sphere. The mathematical formulation of the radius of the Bjerhammar sphere is:

$$R_B = \frac{ab}{\sqrt{\left(a\sin\varphi\right)^2 + \left(b\cos\varphi\right)^2}}$$

where a is the major axis of the reference ellipsoid, b is the minor axis of the reference ellipsoid, and φ is the latitude in radians.

`fitEmpiricalCovariance.m` estimates the coefficients A and B by setting $s = 1$ and $n_{\max} = 250$.

4.3.4 `precomputeCovarianceFunction.m`

The function `precomputeCovarianceFunction.m` calculates the spherical covariance of six functionals of the gravity field.

Auto-covariance of gravity anomaly and potential, and cross-covariance of gravity anomaly and potential are calculated using equations from Tscherning and Rapp (1974):

$$cov\left(\Delta g_P, \Delta g_Q\right) = \sum_{n=2}^{n_{\max}} \sigma_n(\Delta g, \Delta g) s^{(n+2)} P_n\left(\cos\psi\right)$$

$$cov\left(T_P, T_Q\right) = R_B^2 \sum_{n=2}^{n_{\max}} \frac{\sigma_n(\Delta g, \Delta g)}{(n-1)^2} s^{(n+1)} P_n\left(\cos\psi\right)$$

$$cov\left(T_P, \Delta g_Q\right) = r_P \sum_{n=2}^{n_{\max}} \frac{\sigma_n(\Delta g, \Delta g)}{(n-1)} s^{(n+2)} P_n\left(\cos\psi\right)$$

$$cov\left(\Delta g_P, T_Q\right) = r_Q \sum_{n=2}^{n_{\max}} \frac{\sigma_n(\Delta g, \Delta g)}{(n-1)} s^{(n+2)} P_n\left(\cos\psi\right)$$

These equations are also implemented in the Fortran subroutine COVA of GRAVSOFT (Forsberg and Tscherning, 2014).

Auto-covariance of the vertical gradient of gravity anomaly (gravity gradient) is calculated using equation (2.69) of Jekeli (1978):

$$cov\left(dg_P, dg_Q\right) = \frac{\partial}{\partial r_P}\left(\frac{\partial}{\partial r_Q} cov\left(\Delta g_P, \Delta g_Q\right)\right) = \frac{1}{R_B^2} \sum_{n=2}^{n_{\max}} \sigma_n(\Delta g, \Delta g)(n+2)^2 s^{(n+3)} P_n\left(\cos\psi\right)$$

Cross-covariance of gravity gradient and gravity anomaly is calculated using equation (3.66) of Zhu (2007):

$$cov\left(dg_P, \Delta g_Q\right) = \frac{\partial}{\partial r_P}\left(\frac{\partial}{\partial r_P} cov\left(T_P, \Delta g_Q\right)\right) = \frac{1}{r_P} \sum_{n=2}^{n_{\max}} \frac{\sigma_n(\Delta g, \Delta g)(n+1)(n+2)}{(n-1)} s^{(n+2)} P_n\left(\cos\psi\right)$$

Cross-covariance of gravity gradient and potential is calculated using:

$$cov\left(dg_P, T_Q\right) = \frac{\partial}{\partial r_P}\left(\frac{\partial}{\partial r_P} cov\left(T_P, T_Q\right)\right) = \frac{R_B^2}{{r_P}^2} \sum_{n=2}^{n_{\max}} \frac{\sigma_n(\Delta g, \Delta g)(n+1)^2}{(n-1)^2} s^{(n+1)} P_n\left(\cos\psi\right)$$

4.3.5 `interpolateCovarianceFunction.m`

`interpolateCovarianceFunction.m` computes the covariance matrix between two points by using MATLAB griddedInterpolant function. This function builds auto and cross-covariance matrices of LSC for Q observation and P prediction points. Here, the observation points are gravity points and prediction points are DEM points, because DEM is provide in a regular grid, and the aim is to provide a regular grid from LSC. Therefore, we compute three covariance matrices, auto-covariance of gravity anomaly at gravity points:

$$C_{\Delta g, \Delta g}^{obs,obs} = \begin{bmatrix} cov\left(\Delta g_1, \Delta g_1\right) & \cdots & cov\left(\Delta g_1, \Delta g_Q\right) \\ \vdots & \ddots & \vdots \\ cov\left(\Delta g_Q, \Delta g_1\right) & \cdots & cov\left(\Delta g_Q, \Delta g_Q\right) \end{bmatrix}$$

, the auto-covariance of gravity anomaly at DEM points:

$$C^{DEM,DEM}_{\Delta g,\Delta g} = \begin{bmatrix} cov\left(\Delta g_1, \Delta g_1\right) & \cdots & cov\left(\Delta g_1, \Delta g_P\right) \\ \vdots & \ddots & \vdots \\ cov\left(\Delta g_P, \Delta g_1\right) & \cdots & cov\left(\Delta g_P, \Delta g_P\right) \end{bmatrix}$$

, the cross-covariance of gravity anomaly at DEM and gravity points:

$$C^{DEM,obs}_{\Delta g,\Delta g} = \begin{bmatrix} cov\left(\Delta g_1, \Delta g_1\right) & \cdots & cov\left(\Delta g_1, \Delta g_Q\right) \\ \vdots & \ddots & \vdots \\ cov\left(\Delta g_P, \Delta g_1\right) & \cdots & cov\left(\Delta g_P, \Delta g_Q\right) \end{bmatrix}$$

The cross-covariance of gravity anomaly and potential at gravity and DEM points:

$$C^{DEM,DEM}_{\Delta g_{Faye},\Delta g_{Faye}} = \begin{bmatrix} cov\left(\Delta g_1, \Delta g_1\right) & \cdots & cov\left(\Delta g_1, \Delta g_Q\right) \\ \vdots & \ddots & \vdots \\ cov\left(\Delta g_Q, \Delta g_1\right) & \cdots & cov\left(\Delta g_Q, \Delta g_Q\right) \end{bmatrix}$$

$$C^{DEM,DEM}_{T,\ \Delta g_{Faye}} = \begin{bmatrix} cov\left(T_1, \Delta g_1\right) & \cdots & cov\left(T_1, \Delta g_P\right) \\ \vdots & \ddots & \vdots \\ cov\left(T_P, \Delta g_1\right) & \cdots & cov\left(T_P, \Delta g_P\right) \end{bmatrix}$$

$$C^{DEM,obs}_{T,\ \Delta g_{RTM}} = \begin{bmatrix} cov\left(T_1, \Delta g_1\right) & \cdots & cov\left(T_1, \Delta g_Q\right) \\ \vdots & \ddots & \vdots \\ cov\left(T_P, \Delta g_1\right) & \cdots & cov\left(T_P, \Delta g_Q\right) \end{bmatrix}$$

$$C^{DEM,DEM}_{T,\ T} = \begin{bmatrix} cov\left(T_1, T_1\right) & \cdots & cov\left(T_1, T_P\right) \\ \vdots & \ddots & \vdots \\ cov\left(T_P, T_1\right) & \cdots & cov\left(T_P, T_P\right) \end{bmatrix}$$

4.3.6 `solveGravityLSCmatrix.m`

`solveGravityLSCmatrix.m` performs LSC twice using gravity data. First to predict gravity anomalies at DEM points using gravity anomalies at gravity points:

$$\Delta g^{DEM}_{res} = C^{DEM,obs}_{\Delta g,\Delta g}\left(C^{obs,obs}_{\Delta g,\Delta g} + E^{obs,obs}_{\Delta g,\Delta g}\right)^{-1}\Delta g^{obs}_{res}$$

and to compute error variance covariance matrix for predicted gravity anomalies:

$$\Sigma^{DEM,DEM}_{\Delta g,\Delta g} = C^{DEM,DEM}_{\Delta g,\Delta g} \ - \ C^{DEM,obs}_{\Delta g,\Delta g}\left(C^{obs,obs}_{\Delta g,\Delta g} + E^{obs,obs}_{\Delta g,\Delta g}\right)^{-1} C^{obs,DEM}_{\Delta g,\Delta g}$$

Then

$$\Delta g_{FreeAir}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{short}$$

$$\Delta g_{Bouguer}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{long}$$

$$\Delta g_{Faye}^{DEM} = \Delta g_{Bouguer}^{DEM} + 0.0419\rho h$$

where h is the DEM.

Finally, to predict quasigeoid at DEM points:

$$\zeta_{res}^{DEM} = C_{\zeta,\Delta g_{Faye}}^{DEM,DEM} \left(C_{\Delta g_{Faye},\Delta g_{Faye}}^{DEM,DEM} \right)^{-1} \Delta g_{Faye}^{DEM}$$

and to compute error variance covariance matrix for predicted quasigeoid:

$$\Sigma_{\zeta,\zeta}^{DEM} = C_{\zeta,\zeta}^{DEM,DEM} \; - \; C_{\zeta,\Delta g_{res}}^{DEM,obs} \left(C_{\Delta g_{res},\Delta g_{res}}^{obs,obs} \right)^{-1} C_{\Delta g_{res},\zeta}^{obs,DEM}$$

****

LSC was applied to grid residual gravity anomalies from gravity points into DEM points:

$$\Delta g_{res}^{DEM} = C_{\Delta g,\Delta g}^{PQ} \left(C_{\Delta g,\Delta g}^{QQ} + E_{\Delta g,\Delta g}^{QQ} \right)^{-1} \Delta g_{res}^{obs}$$

$$\Sigma_{\Delta g,\Delta g}^{PP} = C_{\Delta g,\Delta g}^{PP} \; - \; C_{\Delta g,\Delta g}^{PQ} \left(C_{\Delta g,\Delta g}^{QQ} + E_{\Delta g,\Delta g}^{QQ} \right)^{-1} C_{\Delta g,\Delta g}^{QP}$$

$E_{\Delta g,\Delta g}^{QQ}$ comes from formal error of preprocessing of gravity anomaly data, actual terrestrial data with std and positioning errors. This process overall returns gridded Faye anomalies; these are in effect terrain-corrected free air anomalies. The reverse Bouguer slab correction can be performed by extracting heights from a DEM. By gridding the Bouguer anomalies prior to the reverse Bouguer slab correction, the potential effect of topographic aliasing in the gridded signal is reduced (Featherstone and Kirby, 2000), since the topographic effect is high frequency and sparse spatial sampling will result in aliasing. Then Faye anomalies are computed by subtracting long wavelength topographic effect and Bouguer formula eq. (2.14) McCubbine (2016).

$$\Delta g_{resFreeAir}^{DEM} = \Delta g_{res}^{DEM} + \delta g_{TE}$$

$$\Delta g_{resBouguer}^{DEM} = \Delta g_{res}^{DEM} - 0.0419\rho (DEM)^{long}$$

$$\Delta g_{resFaye}^{DEM} = \Delta g_{res}^{DEM} + 0.0419\rho (DEM - (DEM)^{long})$$

Then LSC was used to calculate geoid heights from Faye anomalies at DEM points using:

$$\zeta_{res}^{DEM} = C_{\zeta,\Delta g}^{PP} \left(C_{\Delta g,\Delta g}^{PP}\right)^{-1} \Delta g_{resFaye}^{DEM}$$

$$\Sigma_{\Delta g,\Delta g}^{PP} = C_{\zeta,\zeta}^{PP} \; - \; C_{\zeta,\Delta g}^{PP} \left(C_{\Delta g,\Delta g}^{PP}\right)^{-1} C_{\Delta g,\zeta}^{PP}$$

****

There are two other versions of this function depending on which data is used:

`solveGradientLSCmatrix.m` performs LSC twice using gravity gradient data. First to predict gravity anomalies at DEM points using gravity gradients at gravity gradient points:

$$\Delta g_{res}^{DEM} = C_{\Delta g,dg}^{DEM,obs} \left(C_{dg,dg}^{obs,obs} + E_{dg,dg}^{obs,obs}\right)^{-1} dg_{res}^{obs}$$

and to compute error variance covariance matrix for predicted gravity anomalies:

$$\Sigma_{\Delta g,\Delta g}^{DEM,DEM} = C_{\Delta g,\Delta g}^{DEM,DEM} \; - \; C_{\Delta g,dg}^{DEM,obs} \left(C_{dg,dg}^{obs,obs} + E_{dg,dg}^{obs,obs}\right)^{-1} C_{\Delta g,dg}^{obs,DEM}$$

Then

$$\Delta g_{FreeAir}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{short}$$

$$\Delta g_{Bouguer}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{long}$$

$$\Delta g_{Faye}^{DEM} = \Delta g_{Bouguer}^{DEM} + 0.0419\rho h$$

where h is the DEM.
Finally, to predict the quasigeoid at DEM points:

$$\zeta_{res}^{DEM} = C_{\zeta,\Delta g_{Faye}}^{DEM,DEM} \left(C_{\Delta g_{Faye},\Delta g_{Faye}}^{DEM,DEM}\right)^{-1} \Delta g_{Faye}^{DEM,DEM}$$

and to compute error variance covariance matrix for predicted quasigeoid:

$$\Sigma_{\zeta,\zeta}^{DEM} = C_{\zeta,\zeta}^{DEM,DEM} \; - \; C_{\zeta,\Delta g_{res}}^{DEM,obs} \left(C_{\Delta g_{res},\Delta g_{res}}^{obs,obs}\right)^{-1} C_{\Delta g_{res},\zeta}^{obs,DEM}$$

`solveGravityGradientLSCmatrix.m` performs LSC twice using gravity and gravity gradient data. First, we predict gravity anomalies at DEM points using gravity anomalies and gravity gradients at observation points:

$$\Delta g_{res}^{DEM} = \begin{bmatrix} C_{\Delta g,dg}^{DEM,DEM} & C_{\Delta g,\Delta g}^{DEM,DEM} \end{bmatrix} \begin{bmatrix} C_{dg,dg}^{obs,obs} + E_{dg,dg}^{obs,obs} & C_{\Delta g,dg}^{obs,obs} \\ C_{dg,\Delta g}^{obs,obs} & C_{\Delta g,\Delta g}^{obs,obs} + E_{\Delta g,\Delta g}^{obs,obs} \end{bmatrix}^{-1} \begin{bmatrix} dg_{res}^{obs} \\ \Delta g_{res}^{obs} \end{bmatrix}$$

and to compute error variance covariance matrix for predicted gravity anomalies:

$$\Sigma_{\Delta g,\Delta g}^{DEM,DEM} = C_{\Delta g,\Delta g}^{DEM,DEM} - \begin{bmatrix} C_{\Delta g,dg}^{DEM,DEM} & C_{\Delta g,\Delta g}^{DEM,DEM} \end{bmatrix} \begin{bmatrix} C_{dg,dg}^{obs,obs} + E_{dg,dg}^{obs,obs} & C_{\Delta g,dg}^{obs,obs} \\ C_{dg,\Delta g}^{obs,obs} & C_{\Delta g,\Delta g}^{obs,obs} + E_{\Delta g,\Delta g}^{obs,obs} \end{bmatrix}^{-1} \begin{bmatrix} C_{\Delta g,dg}^{DEM,DEM} \\ C_{\Delta g,\Delta g}^{DEM,DEM} \end{bmatrix}$$

Then

$$\Delta g_{FreeAir}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{short}$$

$$\Delta g_{Bouguer}^{DEM} = \Delta g_{RTM}^{DEM} - \delta g_{TE}^{long}$$

$$\Delta g_{Faye}^{DEM} = \Delta g_{Bouguer}^{DEM} + 0.0419\rho h$$

where h is the DEM.

Finally, to predict the quasigeoid at DEM points:

$$\zeta^{DEM} = C_{\zeta,\Delta g_{Faye}}^{DEM} \left(C_{\Delta g_{Faye},\Delta g_{Faye}}^{DEM} \right)^{-1} \Delta g_{Faye}^{DEM}$$

and to compute error variance covariance matrix for predicted quasigeoid:

$$\Sigma_{\zeta,\zeta}^{DEM} = C_{\zeta,\zeta}^{DEM} \; - \; C_{\zeta,\Delta g_{RTM}}^{DEM,gra} \left(C_{\Delta g_{RTM},\Delta g_{RTM}}^{gra,gra} \right)^{-1} C_{\Delta g_{RTM},\zeta}^{gra,DEM}$$

4.3.7 `createGridWeights.m`

The primary numerical challenge in solving LSC is matrix inversion, as evident in Equation 1. To avoid inversion of large matrices in LSC, stepwise LSC is typically employed. However, in this context, we utilize tilewise LSC. This approach involves partitioning the computation area into tiles of maximum one degree by one degree. Geoid calculations are then performed within these tiles, followed by the utilization of a Gaussian filter to blend all the tiles together, resulting in the geoid grid. The `createGridWeights.m` function effectively generates a weighted filter that can be used to blend grids. The Gaussian filter, applied multiple times, acts as a moving average filter, providing a smoothing effect to the input grid data. This repetition of the Gaussian filter helps soften the edges from one tile to another during grid blending.

4.3.8 `comparetoLevellingData.m`

This function compares gravimetric geoid to geometric geoid which means quasigeoid can be compared to the leveling and GNSS derived quasigeoid height anomalies.

$$\zeta^{DEM}_{geometric} = C^{DEM,GPSlevelling}_{\zeta_{geometric},\zeta_{geometric}} \left(C^{GPSlevelling,GPSlevelling}_{\zeta_{geometric},\zeta_{geometric}} \right)^{-1} \zeta^{GPSlevelling}_{geometric}$$

residual and deterended, for LSC, we need to add back the trend.

5 Conclusions

In summary, we have introduced the initial version of our geoid determination code, implemented in MATLAB and utilizing least squares collocation. This code base serves as a foundation for further development, offering a platform for refining geoid determination through gravity anomaly observations.

The primary focus of this release has been on regional quasigeoid determination derived from gravity anomaly data. However, our ongoing efforts involve expanding the functionality of the code to incorporate additional observation types, such as gravity gradients, and diverse sources of gravity anomaly data, including airborne gravimetry and satellite altimetry.

By making our code openly accessible, we aim to foster collaboration and accelerate progress in this field. The transparency afforded by open-source code facilitates the exploration of unanswered questions and encourages more efficient solutions to challenges in geoid determination.

6 References

- Featherstone, W.E. and Kirby, J.F. The reduction of aliasing in gravity anomalies and geoid heights using digital terrain data. 2000. Geophysical Journal International, 141(1): 204-212, doi:10.1046/j.1365-246X.2000.00082.x.

- Forsberg, R. (1984) A Study of Terrain Reductions, Density Anomalies and Geophysical Inversion Methods in Gravity Field Modeling. Dept. of Geodetic Science and Surveying, the Ohio State University, Columbus, Report No. 355. https://doi.org/10.21236/ADA150788

- Forsberg, R. and C. C. Tscherning (2014). *An overview manual for the* GRAVSOFT Geodetic Gravity Field Modelling Programs. 3rd edition, Denmark.

- Darbeheshti, N., Featherstone, W.E. Non-stationary covariance function modelling in 2D least-squares collocation. *J Geod* **83**, 495–508 (2009). `https://doi.org/10.1007/s00190-008-0267-0`

- Darbeheshti, N., Featherstone, W.E. Tuning a gravimetric quasigeoid to GPS-levelling by non-stationary least-squares collocation. *J Geod* **84**, 419–431 (2010). `https://doi.org/10.1007/s00190-010-0377-3`

- McCubbine, Jack (2016). Airborne Gravity Across New Zealand - For An Improved Vertical Datum. Open Access Te Herenga Waka-Victoria University of Wellington. Thesis. `https://doi.org/10.26686/wgtn.17018918.v1`

- Nagy, D. The prism method for terrain corrections using digital computers. *PAGEOPH* **63**, 31–39 (1966). `https://doi.org/10.1007/BF00875156`

- Nagy, D., Papp, G. & Benedek, J. The gravitational potential and its derivatives for the prism. *Journal of Geodesy* 74, 552–560 (2000). `https://doi.org/10.1007/s001900000116`

- Tscherning, C. C. and R. H. Rapp (1974). Closed covariance expressions for gravity anomalies, geoid undulations, and deflections of the vertical implied by anomaly degree variance models. Report 208, Department of Geodetic Science, The Ohio State University, Columbus, USA.

- Tscherning, C.C. (1976): Covariance Expressions for Second and Lower Order Derivatives of the Anomalous Potential, Reports of the Department of Geodetic Science No. 225, The Ohio State University, Columbus, Ohio.

- Zhu L (2007) Gravity gradient modeling with gravity and DEM, Report 483. Geodetic Science and Surveying, The Ohio State University, Columbus.