UNIVERSITY OF KHARTOUM

GeiodApp: A Unified Framework for Geoid Computations

by

Mohamed Jaafar and Mohamed Yousif

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering
Surveying Engineering

November 2016

Declaration of Authorship

I, Mohamed Yousif and Mohamed Jaafar, declare that this thesis titled, 'GeoidApp: A Unified Framework for Geoid Computations' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

"Write a funny quote here."

If the quote is taken from someone, their name goes here

UNIVERSITY OF KHARTOUM

Abstract

Faculty of Engineering
Surveying Engineering

Doctor of Philosophy

by Mohamed Jaafar and Mohamed Yousif

Figure of the earth remains one of the most crucial aspects of geodesy. Modern studies of the figure of the earth have started with Gauss [**?**] who call it 'geoid'. The geoid according to Gauss is the surface that approximates the sea level. Another approach of describing the figure of the earth was proposed by Molodnskey [**?**]. His proposal was to treat the earth's figure as 'a boundary value problem'. We're interested in the later approach as we believe that it's economically feasible, yet accurate enough.

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

Abbreviations

GGM	**G**lobal **G**eopotentail **M**odel
BVP	**B**oundary **V**alue **P**roblem
GBVP	**G**eodetic **B**oundary **V**alue **P**roblem
GPS	**G**lobal **P**ositioning **S**ystem
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}
σ	standard deviation	meters

For/Dedicated to/To my...

Chapter 1

Introduction

Historically Sudan has two gravimetric geoid models [?]. The first one (Geoid91) was computed by [1] in 1991 using Geodetic reference System GRS80. The free-air co-geoid was computed from the combination of surface gravity data using a modified Stokes's kernel and GODDARD EARTH MODEL (GEM-T1) [1]. GEM-T1, a satellite-only model, is complete to degree and order of 36 [?]. The second model is known as 'KTH-SDG08' was computed in 2008 using optimum least-squares modification of Stoke's kernels, which is widely known as KTH method [?]. EIGEN-GRACE02S satellite-only model was adopted for KTHSDG08 final computation at spherical harmonic degree and order 120. Adam was the first to attempt to compute the geoid for Sudan in 1967 [?]. Due to lack of data from neighboring countries and large un-surveyed areas in North-west and Southwest parts of Sudan, Adam found that the information was insufficient to determine the accurate geoid in Sudan and recommended to fill the gabs over there [?].

1.1 Objectives of the thesis

The main objective of this thesis is to evaluate recent grace/goce models over Sudan using asterogeodetic data and GPS/Leveling data. As a result we also propose a general software framework for various geoid components computations. In particular, it can be used to compute "geoid-height", "geoid-undulation'', and "geoid-disturbance". To the best of our knowledge, we could not find a software for geoid computations that is

- Supported. Most of geoid computations libraries are no longer supported, their links are dead.

- Strong back-end, and simple front-end. It is common to use a low-level programming language in the calculation of geoid componenets e.g. geoid height. Low-level languages are very fast compared to high-level languages, but they are much harder. We built a framework on top of C/C++ libraries, with a very simple interface, in particular Java and Matlab.
- Customizable. Because we know that users need to explore different models with different degree. We offer them just that. GeoidApp is designed such that the user can easily modify any parameter.

In particular our contribution is

- Evaluate recent Grace/Goce money in Sudan
- GeoidApp: A unified software framework for geoid computations

1.2 Thesis structure

Following this introduction, the thesis is divided into five chapters

- Chapter 2
  Glances the theory behind GGMs, and summarizes dedicated satellites missions.
- Chapter 3
  Details about the input data
- Chapter 4
  Methodology

Chapter 2

GGMs

Chapter 3

Input Data

Chapter 4

Geoid Height Computation

In this chapter we will discuss the computation of the geoid height and the validation of our results based on local terrestrial data. High resolution models are required to convert GPS leveling data (ellipsoidal height) into orthometric height. As we mentioned in the previous chapter, we have two terrestrial data 1) Astrogeodetic data provided by [?], and 2) GPS leveling data by [?]. Our results show that high degrees will often result in a lower standard deviation. Combined models e.g., EGM2008, EIGEN-6C4, and Geco have a very similar trend–it is expected because both EIGEN-6C4 and Geco share some degrees with EGM2008. Interestingly, ITU GCC has shown a peak in higher degrees (above 150), we believe that is one of the problems of satellite-only models in high degrees.

4.1 Historical Background about Geoid Computations in Sudan

There are several attempts was done through the years to compute the geoid for Sudan. The first was committed by [?] in 1967. He was done it during his MSc in Cornell University. He used astrogeodetic data i.e., astrogeodetic geoid, it's common to use astrogeodetic observations when there is a lack of data. Even with that, Osman recommended to fill the gaps due to large un-surveyed areas (largely on Northwest and Southwest parts of Sudan), and lack of data from neighboring countries. He used Clarke 1880 as a reference ellipsoid which is reasonable because the well-known WGS wasn't established at that time. Another attempt was done by [1] in 1991. Unlike Osman's geoid, he used a gravimeteric data i.e., a gravimetric geoid. He covered a grid of $(5° \leq \phi \leq 22°, 22° \leq \lambda \leq 38°)$. Fashir introduced the use of GGMs in computing the long wavelength components of the Earth gravity. He used Goddard Earth Model

(GEM-T1) with a modified Stoke's kernel to compute the geoid height.
More recently [?] proposed a new gravimetric datum for Sudan KTH-SDG08. Ahmed's model was based on the new dedicated satellite missions (Goce/Grace, and CHAMP). New satellite mission have an improved results over the previous satellite missions, thanks to the gradiometer and the precise positioning system from GNSS. The new gravimetric geoid model (KTH-SDG08) has been determined over the whole country of Sudan at 5 x 5 grid for area ($4° \leq \phi \leq 23°, 22° \leq \lambda \leq 38°$). The optimum method provides the best agreement with GPS/levelling estimated to 29 cm while the agreement for the relative geoid heights to 0.493 ppm [?]. Ahmed used GRACE02S gravitational model (for the long wavelength part) and 30x 30 SRTM DEM (for the short wavelength part of Earth's gravity field), beside the terrestrial data for the medium wavelength part. For a fair comparsion, we compared only the results of [?] with our results. It clearly shows that the std has dropped from 0.576 (in KTH-SD08) to 0.349 in our case. We gained a 40% accuracy without any terrestrial observations. We wanted to compare our results with that of [1], but as indicated by [?] Doppler data of [1] is not available. From [?], Fashirs model looks smoother than KTH-SDG08. The drastic values of the geoidal height in the north-west corner and south-east corner are 14 m and -10 m for Fashirs model while 20 m and -14 m for KTH-SDG08, respectively. The fitting with the reference ellipsoid is similar from the northwest to south-east. Fashirs model covers the reference ellipsoid over a large area (approximately 75%). On the contrary KTH-SDG08 model apparently keeps the same fitting with reference ellipsoid as in the original area before resizing.

4.2 Computing the Geoid

We developed a software application called GeoidApp for this task. GeoidApp is a huge suite of applications to not only compute geoid height (in addition to gravity anomaly and gravity disturbance), but it also has analysis and interactive visualization features to help the users get the most out of it. For that reason we will discuss the core features about GeoidApp. To compute the geoid height you need to account for these details

- The data. GGMs are provided as *.gfc files from ICGEM. They contains in addition to the coefficients text contents about the authors, acknowledgments, etc. They also include model parameters, such as the maximum degree, the model name, the GM, and a few others. The idea is to store the coefficients in an array (for computations efficiency).

- Helper functions for Associated Legendre Polynomials. That could tricky, solving Legendre function is non-trivial. The recursive version of Legendre function is suitable for programming environment.
- The main function for geoid computations. You have to choose whether you want to work on point mode, grid mode.

4.2.1 Working with GGMs data

Typical GGMs provided by ICGEM are raw and need to be parsed to extract the model's coefficients and other useful informations about the model. When we say a 'patter', or 'template' we mean that there are certain keywords in the *.gfc files, and it is available in all models.

4.2.1.1 Parsing the data

A typical model from ICGEM would consist of the following

- General information about the mission.
- A header with structured details about the models, and their parameters
- The coefficients part. Which is the core part of the *.gfc files

Model's summary . It includes summary about the mission its release date, mission time, etc. There is also a link for detailed informations about the mission. You can safely ignore this part of of the *.gfc file. This part of *.gfc files does not follow any pattern, you cannot build a generic tool to extract informations from it.

The header . This summarizes the previous in a template way. It will begin with the keyword *original header*. Below it, there is the name of the model, and the release date, mission duration. It also has information about GM but you can also ignore it here as it will be provided later in more structured way. A good way to think of *.gfc files is xml format. xml files are constructed as `<begin_of_tag> some_data </end_of_tag>`. For our work, we treated *.gfc files as follow

```
<begin_of_head>
   <original header>
   "product_type": "gravity_field" /* In our case we need this */,
   "model_name": "model_name",
```

```
    "radius": "6378136.30" /* It varies over models */,
    "earth_gravity_constant": "398600.4415D+09" /* It varies over models */,
    "max_degree": "n",
    "errors": "formal", /* It could take other values */,
    "tide_system": "tide_free" /* other types are available */,
    </original header>
</end_of_head>
```

Note that we use those keywords to store these variables with their corresponding values in a data structure (a dictionary, or hash table is suitable for this purpose).

The last part of parsing global geopotential models is extracting the model coefficients C_{nm}, S_{nm}. We want to construct a $(n \times n)$ matrix, where n denotes maximum degree of our model. For each raw of our matrix will correspond to the degree of the model. Each will have no-zero elements as the index of of that raw. The first raw will always have value 1, and the rest of it are zeros. The second raw will depend on an argument that is specified by the user *subtract_normal_field*. In our evaluations we always set such that the "normal_field" will be subtracted from coefficients $c_{nm}(1:9, 1:9)$. It's important to note that indexing for global geopotential models starts from zero as oppose of one. You should account for that if your are using programming language that uses one as a base for indexing e.g., MATLAB/Octave or Julia.

$$C_{nm} = \begin{bmatrix} 1 & \dots & \dots & N_{max} \\ 0 & 0 & \dots & N_{max} \\ \vdots & \vdots & \ddots & N_{max} \\ -4.84e-04 & -3.98e-10 & 2.43e-06 & N_{max} \end{bmatrix}$$

For the S_{nm} part there the first column of it, i.e., the value of S_{nm} coefficients are always zero. That is $S_{nm}[:, 0] = 0$. Unlike C_{nm} the number of non-zero elements for each raw in S_{nm} equals the index of that raw minus 1.

$$S_{nm} = \begin{bmatrix} 0 & \dots & \dots & N_{max} \\ 0 & 0 & \dots & N_{max} \\ \vdots & \vdots & \ddots & N_{max} \\ 0 & 1.42e-09 & -1.40e-06 & N_{max} \end{bmatrix}$$

The same was applied to construct $e_{C_{nm}}, e_{S_{nm}}$ the std values for C_{nm}, S_{nm}, respectively.

4.3 Associated Legendre Function

4.3.1 Other Works

Solving Legendre function is not easy. It is always good to build on top of others projects, and reference them as needed. We found a few libraries that solve Legendre function. One of them is legendre function in provided by MATLAB. This function does not come with the standard version of MATLAB, so we avoided it. The implementation of it does not also meet with our use case. We have also found other implementation of Legendre that was written in either C++ or Java, both of them was not used during the development of GeoidApp. Another very recent implementation of legendre function was that of NumPy, a popular numerical library for Python. As the time of writing our code, NumPy's legendre was not yet implemented.
We had to either translate libraries from other languages, or modify that version of MATLAB. We decided to write our own implementation of Legendre. In both case we had modify something anyway, we also want to ensure that the implementation of our function should be compatible with other parts of our application.

Appendix A

An Appendix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus at pulvinar nisi. Phasellus hendrerit, diam placerat interdum iaculis, mauris justo cursus risus, in viverra purus eros at ligula. Ut metus justo, consequat a tristique posuere, laoreet nec nibh. Etiam et scelerisque mauris. Phasellus vel massa magna. Ut non neque id tortor pharetra bibendum vitae sit amet nisi. Duis nec quam quam, sed euismod justo. Pellentesque eu tellus vitae ante tempus malesuada. Nunc accumsan, quam in congue consequat, lectus lectus dapibus erat, id aliquet urna neque at massa. Nulla facilisi. Morbi ullamcorper eleifend posuere. Donec libero leo, faucibus nec bibendum at, mattis et urna. Proin consectetur, nunc ut imperdiet lobortis, magna neque tincidunt lectus, id iaculis nisi justo id nibh. Pellentesque vel sem in erat vulputate faucibus molestie ut lorem.

Quisque tristique urna in lorem laoreet at laoreet quam congue. Donec dolor turpis, blandit non imperdiet aliquet, blandit et felis. In lorem nisi, pretium sit amet vestibulum sed, tempus et sem. Proin non ante turpis. Nulla imperdiet fringilla convallis. Vivamus vel bibendum nisl. Pellentesque justo lectus, molestie vel luctus sed, lobortis in libero. Nulla facilisi. Aliquam erat volutpat. Suspendisse vitae nunc nunc. Sed aliquet est suscipit sapien rhoncus non adipiscing nibh consequat. Aliquam metus urna, faucibus eu vulputate non, luctus eu justo.

Donec urna leo, vulputate vitae porta eu, vehicula blandit libero. Phasellus eget massa et leo condimentum mollis. Nullam molestie, justo at pellentesque vulputate, sapien velit ornare diam, nec gravida lacus augue non diam. Integer mattis lacus id libero ultrices sit amet mollis neque molestie. Integer ut leo eget mi volutpat congue. Vivamus sodales, turpis id venenatis placerat, tellus purus adipiscing magna, eu aliquam nibh dolor id nibh. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed cursus convallis quam nec vehicula. Sed vulputate neque eget odio fringilla ac sodales urna feugiat.

Phasellus nisi quam, volutpat non ullamcorper eget, congue fringilla leo. Cras et erat et nibh placerat commodo id ornare est. Nulla facilisi. Aenean pulvinar scelerisque eros eget interdum. Nunc pulvinar magna ut felis varius in hendrerit dolor accumsan. Nunc pellentesque magna quis magna bibendum non laoreet erat tincidunt. Nulla facilisi.

Duis eget massa sem, gravida interdum ipsum. Nulla nunc nisl, hendrerit sit amet commodo vel, varius id tellus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac dolor est. Suspendisse ultrices tincidunt metus eget accumsan. Nullam facilisis, justo vitae convallis sollicitudin, eros augue malesuada metus, nec sagittis diam nibh ut sapien. Duis blandit lectus vitae lorem aliquam nec euismod nisi volutpat. Vestibulum ornare dictum tortor, at faucibus justo tempor non. Nulla facilisi. Cras non massa nunc, eget euismod purus. Nunc metus ipsum, euismod a consectetur vel, hendrerit nec nunc.

Bibliography

[1] H. H. Fashir. The gravimetric geoid of sudan. *Journal of Geodynamics*, 14:19–36, 1991. doi: 10.1016/0264-3707(91)90005-Y.