

# APG4001S Geoid Computation from spherical harmonic coefficients

Tim Marsh

10 August 2015

Assignment 1



**Contents**

**1 Introduction 2**

1.1 Subject of Report . . . . . 2

1.2 Background . . . . . 2

1.3 Objectives . . . . . 2

1.4 Scope and Limitations . . . . . 3

**2 Method 4**

**3 Results 6**

**List of Figures**

1 Relationship between ellipsoidal height, Geoidal height and  
orthometric height . . . . . 2

# 1 Introduction

## 1.1 Subject of Report

This is a report on Assignment 1 for APG4001S Geodesy. The aim of the report was to calculate orthometric heights for each of the 5 stations provided through the use of spherical harmonic coefficients.

## 1.2 Background

Using the GGM02S Grace derived geopotential model, to degree/order 160, and the coordinates of 5 Trignet stations. As well as an exert from the "Geodesists Handbook" which provides the constants required for the GRS80 ellipsoid.

A program is written to calculate the Geoid-ellipsoid (N) values at each station using the geopotential model provided. Using the N value calculate the orthometric height (H) at each station.

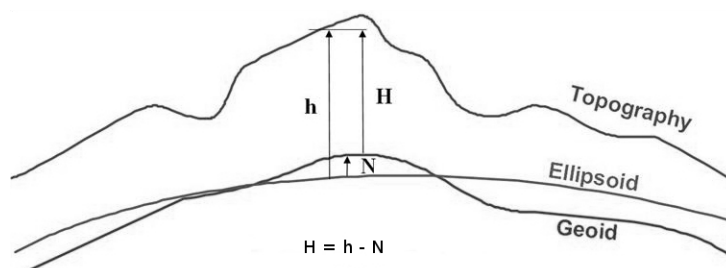


Figure 1: Relationship between ellipsoidal height, Geoidal height and orthometric height

The 5 Trignet stations used are:

Station Name	Latitude	Longitude	Ellipsoidal Height
HNUS	34 25 28.6671 S	19 13 23.0264 E	63.048
PRET	25 43 55.2935 S	28 16 57.4873 E	1387.339
RBAY	28 47 43.9616 S	32 04 42.1896 E	31.752
TDOU	23 04 47.6714 S	30 23 02.4297 E	630.217
ULDI	28 17 35.2196 S	31 25 15.3309 E	607.947

## 1.3 Objectives

The primary objective of this assignment was to create a program to calculate the geoid-ellipsoid values at each station using the provided global geopoten-

tial model, once these values are calculated they are used to calculate the orthometric height at each station.

## **1.4 Scope and Limitations**

This assignment was done using the GSM80 ellipsoid and the GGM02S Grace potential model to degree/order 160.

A substantial limitation to the program was that Python 3.4 has a size limit on floating point numbers and the code cannot be run for all 160 coefficients without getting an overflow error, so it is only run for 85.

## 2 Method

The program has a reasonably simple structure, the GGM02S file is read in and saved using a class structure in Python 3.4. and is accessible by using a simple query of what you want fed into a function. for example if you want S with  $n = 57$   $m = 3$  you feed in the query string "S 27 16" and it will return the value 0.39484545087973E-08.

Then for each station a Gamma value for the specific longitude is calculated using the formula:

$$\gamma = \gamma_e \frac{1 + k \sin^2(\Phi)}{\sqrt{1 - e^2 \sin^2(\Phi)}} \quad (1)$$

Where  $\gamma_e$  is the normal gravity at the equator

$$N(r, \theta, \lambda) = \frac{GM}{\gamma r} \sum_{n=2}^{\infty} \binom{a}{r} \sum_{m=0}^n [(\Delta C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) P_{nm}(\cos \theta)] \quad (2)$$

Then once the  $\gamma$  has been calculated, N is calculated in parts. by de-constructing equation (2) with the inner sum, outer sum and the outermost dividend calculated almost separately and brought together when returned from the function.

The inner sum which runs inside a loop that runs from 0 to  $n$

$$InnerSum = \sum_{m=0}^n [(\Delta C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) P_{nm}(\cos \theta)] \quad (3)$$

In this equation  $P_{nm}$  is equal to:

$$P_n(t) = \frac{1}{2^n n!} \cdot \frac{d^n}{dt^n} (t^n - 1)^n \quad (4)$$

Where the value  $t = \cos \theta$

Is then added to the outer sum value and run in a loop from 0 to a specified value, 160.

$$OuterSum = \sum_{n=2}^{\infty} \binom{a}{r} \times InnerSum \quad (5)$$

Then finally we add the values that are outside of the outer sum:

$$N(r, \theta, \lambda) = \frac{GM}{\gamma r} \times OuterSum \quad (6)$$

This gives the final value of N.

Then with this final value of N we calculate the orthometric height with the simple calculation of:

$$\text{Orthometric Height (H)} = \text{Ellipsoidal Height (h)} - N$$

### 3 Results

The results of calculating N after 85 iterations,  $n = 85$  are as follows:

Station Name	Orthometric Height	Geoidal Height
HNUS	29.638	33.410
PRET	1365.770	21.569
TDOU	611.833	18.384
ULDI	586.555	21.392
RBAY	10.520	21.232

These heights are displayed to 3 decimal places, however the program does calculate to around 10 decimal places so the results are rounded of to a usable and more readable size.

As for only doing 85 iterations the later harmonics are making such a small impact of the heights that when rounding off to 3 decimal places no difference will be noticed.