THE ALTIMETRY-GRAVIMETRY PROBLEM USING ORTHONORMAL BASE FUNCTIONS

by

André Mainville

Report No. 373
Department of Geodetic Science and Surveying
The Ohio State University
1958 Neil Avenue
Columbus, Ohio 43210-1247

December 1986

FOREWORD

This report was prepared by André Mainville, Graduate Student, Department of Geodetic Science and Surveying, The Ohio State University, under the appreciated supervision of Dr. Richard H. Rapp, adviser.

This report was also presented in partial fulfillment of the requirement for the Degree of Doctor of Philosophy in the Graduate School of The Ohio State University.

I am indebted to Geodetic Survey of Canada for having financially supported my education.

Most of the computer funds used in this study were generously supplied by the Instruction and Research Computer Center through the Department of Geodetic Science and Surveying.

The reproduction and distribution of this report was carried through funds supplied by the Department of Geodetic Science and Surveying.

ACKNOWLEDGEMENT

I am grateful to my adviser Professor Richard H. Rapp for suggesting this study, for his enthusiastic involvement, and for his expert guidance throughout this work.

I am especially grateful to Geodetic Survey of Canada for providing me with such an enjoyable research opportunity and providing financial support during my studies.

I cannot stress enough the degree to which I have relied on work from previous research projects; especially the work done by K. Arnold, M. Brillouin, D. Gleason, M.K. Paul, L. Pellinen and R.H. Rapp.

In addidtion to Dr. Rapp, thanks are also due to Drs. C. Goad and K. Kubik for their constructive comments as members of the reading committee.

Finally, this work would never have been completed without the comfort and relaxation brought by the presence of my wife Odette, and children Mathieu, Christine and Caroline.

ABSTRACT

This dissertation was undertaken in view of finding a numerical solution on a spherical earth of a mixed boundary value problem, the one of altimetry-gravimetry which is defined from gravity anomalies determined by gravimeter measurements mostly on continents and from geoid undulations known on the oceans from the methods of satellite altimetry.

The disturbing potential is represented by an expression of new orthonormal base functions over the sphere. These new base functions are formed using the Gram-Schmidt orthonormalization process applied to the spherical harmonics base functions. Also the Orthonormalization process needed to be applied to mixed domains. The new orthonormal base functions are related to the integration of two associated Legendre functions. This integration is computed using newly developed recursive relations similar to the ones integrating one associated Legendre function developed by Paul (1978). Then the fast Fourier transform is used in a similar way as the spherical harmonics analysis and synthesis.

The result of this solution to the "altimetry-gravimetry problem" is a set of coefficients of the new orthonormal base functions. These coefficients were retransformed into the ones of the usual spherical harmonics expansion. The spherical harmonic coefficients can then easily be analyzed and compared with existing earth's gravity field expansions.

This method is a "Least-Squares method" solution but it is different than a "Least-Squares adjustment". It is stressed that the Least-Squares method i.e. minimizing the integral and not the sum of the squares of the residuals is solved using orthonormal base functions. It is the solution that has been numerically applied here but it should be emphasized it is also the solution that permits the computation of the usual spherical harmonic geopotential coefficients in the classical single boundary value problem in physical geodesy. Numerical tests show that this Least-Squares method can solve the altimetry-gravimetry problem.

Errata Page for Report No. 373
Department of Geodetic Science and Surveying

The Altimetry-Gravimetry Problem Using Orthonormal Base Functions
by A. Mainville, December 1986

1. On page 38, before the 3rd line from the bottom, add:

The weight function used in this work is simply two values, one to scale all the $\overline{\Delta g}_{ij}$ values and a second one to scale all the $\overline{T}_{ij}$ values. Hence, the weight function of equation (4.5) defines 2 values which were only used initially i.e. before any iterations. The two values used as weight after each iterations k were the inverse of the mean of the squares of the residuals

$$\overline{W}_{ij}^{k} = \left| \begin{array}{ll} \left(\dfrac{1}{\mathrm{RMS}\,(T_{ij} - \frac{GM}{R} \Sigma\, T_n S_n)} \right)^2 & \text{if } i,j \in \sigma_1 \\ \left(\dfrac{1}{\mathrm{RMS}\,(\Delta g_{ij} - \frac{GM}{R^2} \Sigma\, (R_n - 1) T_n S_n)} \right)^2 & \text{if } i,j \in \sigma_2 \end{array} \right. . \qquad (4.5b)$$

2.	On page 104, 1st line, change	"value"	to	"residual".
3.	On page 104, 4th line, change	"values"	to	"residuals".
4.	On page 104, 10th line, change	"values"	to	"residuals".
5.	On page 120, 4th line, change	"values"	to	"residuals".
6.	On page 120, 10th line, change	"values"	to	"residuals".
7.	On page 120, 14th line, change	"values"	to	"residuals".
8.	On page 138, 16th line, change	"values"	to	"residuals".
9.	On page 138, 17th line, change	"values"	to	"residuals".
10.	On page 138, 18th line, change	"squares"	to	"square residuals".
11.	On page 104, 11th line, change	"4.5)"	to	"4.5 and 4.5b)".
12.	On page 120, 4th line, change	"4.5)"	to	"4.5 and 4.5b)".
13.	On page 119, 12th line, change	"RMS(Δg^0_{ij}), RMS(T^0_{ij})"		
	to	"RMS(Δg^1_{ij}), RMS(T^1_{ij})".		

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES (Continued)

Table Page

INTRODUCTION

This dissertation contains 9 chapters, 7 annexes including one for well documented FORTRAN routines. Chapter 1 contains the background explaining the altimetry-gravimetry problem, the needs for a solution to the altimetry-gravimetry problem (e.g. the direct use of altimetry data) and the needs for a knowledge of the higher frequencies of the gravity field.

Chapter 2 contains some definitions and is divided into 5 smaller sections of 2, 3 pages each. Section 2.1 introduces the expansion in spherical harmonics in a notation used throughout this work that permits an easy and clear way to write derivatives and write shortly long equations. Section 2.2 shows that the usual spherical harmonics expansion is a solution of the least-squares "method" (not "adjustment") without weights. Section 2.3 shows the notation used for the scalar product of functions. Section 2.4 and 2.5 generalize the scalar product for the case where weights are used and for the case where there is more then one domain as in the altimetry-gravimetry problem.

Chapter 3 describes Arnold's (1978) global solution to the altimetry-gravimetry problem using new orthonormal base functions derived from solid spherical harmonics. This is the solution to the altimetry-gravimetry problem that it is intended to numerically apply in this work. The solution is the disturbing potential in a series expansion that describes the potential of the earth globally at all latitudes and longitudes. It is a least-squares "method" (not "adjustment") solution, and it is the most natural kind of solution to the altimetry-gravimetry problem as the expansion of spherical harmonics is to the single boundary value problem.

While chapter 3 describes the proposed solution in general, chapters 4, 5, 6, 7 and 8 give the details of its computations that lead to the first numerical application in chapter 9.

Chapter 4 describes a first of three parts of the solution using fast Fourier transform applied to mixed data such as gravity anomalies and geoid undulations. For clarity this chapter refers to appendix F where the spherical harmonic analysis of Colombo (1981) using fast Fourier transform is introduced for easy reference. The De-smoothing operator is introduced there.

Chapter 5 introduces the Gram-Schmidt orthonormalization process. In 4 sections it describes successively the process itself, the organization of the computations, the process with mixed domains such as the altimetry-gravimetry problem, and the organization of the computations using spherical harmonics since these are used as starting base functions.

Chapter 6 shows that the solution requires the integration of two associated Legendre functions and that these are used in another fast Fourier transform application to compute the second of three steps of the solution.

Chapter 7 is used to derive the recurrence relations for integrals of two associated Legendre functions. First the recurrence relations for the integration of "one" associated Legendre function of Paul (1978) are introduced and there derivations are in three appendices for easy reference. The recurrences to compute the integral of the product of two associated Legendre functions are then fully derived. The validity of the newly derived relations is ensured by some properties of the spherical harmonics.

Chapter 8 gathers all the equations required in the last of three steps of the solution of the altimetry-gravimetry problem. It shows how the spherical harmonic coefficients are computed directly without having to compute new orthogonal base functions and coefficients. It shows how the orthonormalization process can be computed by a Cholesky factorization followed by a forward and backward solution (Freeden, 1983).

Chapter 9 shows a first numerical application of this solution to a model. geopotential coefficient set GEML2 (Lerch et al., 1982) is used to compute gravity anomalies and geoid undulations mixed on a sphere. The coefficients are computed back using our proposed solution and recovered exactly as they are in the single b.v.p. solution. Thus a numerical proof is made that this solution can solve the altimetry-gravimetry problem. The results are analyzed using tables of RMS differences, anomaly degree variances, storage required and cpu times.

Still in chapter 9, a difference is made between an "iterated" least-squares solution where the residuals are minimized after some iterations and a "deterministic" solution where there is no iteration and the residuals are not necessarily minimized. The solution demonstrated here can be used in both ways.

Chapter 9 makes also a return to the single boundary value problem to apply what we have learned from the mixed altimetry-gravimetry boundary value problem. It stresses that being a particular case of our solution, the usual spherical harmonic expansion with the coefficients being found so simply because of the orthogonality relationship can also be a least-squares solution found using iterations. It is also stressed that the difference between the least-squares adjustment techniques where the "sum" of the residuals (weighted and squared) are minimized while the least-squares method used here minimizes the "integral" of the residuals (weighted and squared) (Collatz, 1960). Due to approximations in computing the integrals involved, the numerical solution without iteration does not minimize the residuals even though the system of equations is linear. Again this is often called the deterministic solution. Being a least-squares method, the solution can be iterated until the sum of the squares of the residuals is minimized. It is shown that the iteration allows one to recover the coefficients exactly for a model and can allow one to improve the solution when using real world data. It is also suggested that this iteration process could be used to find a solution when the de-smoothing operator required can not be found such as solutions involving the ellipsoid and the topography. Numerical examples of iterative solutions of the usual spherical harmonic expansion are shown.

The conclusion summarizes the contribution and knowledge acquired by this work. One is the integration of 2 associated Legendre functions that might be necessary in other proposed solution to the mixed boundary value problem (Sacerdote and Sanso, 1985). Suggestions that might bring this solution to be more efficient are made which would make this solution ready to be used with real world data. Such practical solution would provide us with higher degrees of the spherical harmonic coefficients which can represent the disturbing potential on and outside the Earth.

CHAPTER I

BACKGROUND

In the classical boundary value problem (b.v.p.) of physical geodesy, the surface of the Earth is considered as a sphere and the gravity anomalies are the known boundary values. With these hypotheses the disturbing potential at the Earth's surface and in the external space of the Earth can be computed using Stokes' formulae which results from Stokes' theory. This solution is referred to as the "local solution" to the "single b.v.p. " since it is usually computed with a dense grid of gravity anomalies locally around the computation point.

The disturbing potential can also be represented by a spherical harmonic expansion. In the external space of the Earth we use a "solid" spherical harmonic expansion and on the Earth's surface a "surface" spherical harmonic expansion. The coefficients associated to the individual spherical harmonics, called also Stokes' constants, are obtained from a similar integration, as Stokes' formulae, of the gravity anomalies on the Earth's surface. This solution (given in appendix E) is referred to as the "global solution" to the "single b.v.p. " since when the Stokes' constants are found one can compute the disturbing potential and its components at any location on the globe.

This problem has been given a new form by the developments in the field of satellite altimetry. The altitude of the satellite above sea level can be determined directly point by point using an altimeter installed in a satellite. From an accurate determination of the satellite orbit, the values of the geoid undulations on the oceans are obtained point by point with an accuracy of about ± 2 metres and with a great abundance of details. From these geoid undulations, N, and Brun's formula, $T = N\gamma$, where γ is the normal gravity, the disturbing potential, T, is also known point by point over the oceans. However, since the method of satellite altimetry fails on the continents, in this case one has to resort to the gravity anomalies obtained from gravimeters. Thus the classical Stokes problem is given a new form. Now the values of the disturbing potential are given on the oceans, while gravity anomalies are given on the continents; an analytical expression for the disturbing potential on and external to the Earth's surface on the basis of these heterogeneous data is required.

One such analytical expression is again the spherical harmonic expansion. However the Stokes' constants have to be found by other means than a full coverage of gravity anomalies around the Earth. It is the global solution to this mixed b.v.p. that is sought in this dissertation and the spherical harmonic expansion is the form of solution expected.

Rapp (1978, 1981) has solved numerically the global solution to the single b.v.p.. In Rapp (1981) satellite altimetry data were used. However having no analytical expression to use directly with the geoid undulations, these undulations were transformed into gravity anomalies. Having bypassed the problem of using heterogeneous data the full coverage of gravity anomalies could be used to find a solution as a single b.v.p.. Many assumptions and approximations have been used in that enormous task of transforming undulations into anomalies (Rapp, 1979).

By all means an alternative solution would be welcome. Arnold (1981) has proposed an alternative and it is this global solution that has been numerically tested in this dissertation.

As in the classical b.v.p. of physical geodesy where the disturbing potential can be represented over the sphere by an expression of orthonormal base functions, the spherical harmonics, Arnold (1981) has proposed a similar expansion of the disturbing potential into another set of orthonormal base functions. These new base functions are formed using the Gram-Schmidt orthonormalization process applied to the spherical harmonics.

This kind of solution to the mixed b.v.p. was first given by Brillouin (1916) from which Arnold was inspired. This solution shows the Gram-Schmidt orthonormalization process using mixed integrals i.e. mixed scalar products. It then shows how the Least-Squares method i.e. minimizing the integral of the square of the weighted residuals can be solved using orthonormal base functions. It is different than Least-Squares adjustment where the sum and not the integral of the square of the weighted residuals is minimized. Here the solution sought computes integrals. It is an integral formulas solution and it will be shown that it can be reduced to the classical spherical harmonics orthogonality relationship which solves the single b.v.p..

One can easily imagine that the computations in this solution to the mixed altimetry-gravimetry b.v.p. are more demanding than for the simpler single gravimetric b.v.p. alone. Brillouin (1916) has given some directions on how to organize these computations. First one must relate the new orthonormal base functions to the integration of "two" associated Legendre functions. Then one must derive recursive relations between these integrals of "two" associated Legendre functions similar to the ones for "one" associated Legendre function developed by Paul (1978). These newly derived recursive relations are verified against the later and by other means. Other numerical problems had to be overcome, and fast Fourier applications have been used in view of having a practical solution that integrates as many gravity anomaly and geoid undulation information as possible.

The result of this solution to the altimetry-gravimetry problem is a set of coefficients of the new orthonormal base functions from which the components of the gravity field anywhere on and outside the Earth could be computed. However these coefficients were retransformed into the ones of the spherical harmonics which permits one to use existing efficient software to compute any component of the disturbing potential. It also permits the analysis and comparison with existing Earth's gravity field expansions. And most importantly it allows one to ultimately combine this solution with "satellite-derived potential coefficients". This ultimate combination would give the desired solution to the mixed altimetry-gravimetry b.v.p. i.e. the improvement of the knowledge of the geoid and the Earth's gravity field by deriving a better set of high degree spherical harmonic potential coefficients.

As it is well known now, the use of such potential coefficients is well appreciated for computing geoid undulations, gravity anomalies, etc., in both global and local gravity field applications.

Chapter 9 shows that it has been possible to solve numerically the altimetry-gravimetry problem. As it will be seen in chapter 8 and 9 the numerical tests show that a large computer or improvement in the efficiency of the computations is still required to get a practical solution for the high degree of the gravity field. Least-Squares collocation (Moritz, 1980) and (Colombo, 1981) or Least-Squares adjustment (Wenzel, 1985) could have been tried to solve the problem. Their usage in physical geodesy is often rejected because they require a large matrix inversion or the solution of a large system of equations. Still this same inconvenience has been encountered here. But the similarity with the spherical harmonic solution to the single b.v.p. pushed us to try Arnold and Brillouin's proposal to obtain a simpler and possible numerical solution. After reading other geodesists' works on the subject we became aware of the theoretical drawback of using a Least-Squares solution (Svensson, 1985). However it was thought that any parts of a numerical task like done in this dissertation could become a contribution to help future numerical and perhaps theoretical studies to solve in a practical manner the mixed altimetry-gravimetry b.v.p. in physical geodesy. Disregarding efficiency it is believed that the numerical results in chapter 9 proves that the least-squares solution proposed by Arnold (1981) for the "global" altimetry-gravimetry problem is successful.

CHAPTER II

SOME DEFINITIONS

2.1 The Expansion in Spherical Harmonics.

Through all this dissertation we will be on a spherical Earth and will use the polar spherical coordinate system (θ, λ, r) where θ is the colatitude, λ the longitude positive east and r the radius vector. A piecewise continuous (Colombo, 1981, p.2) function $f(\theta, \lambda)$ known on this sphere of unit radius can be expanded as an infinite series of fully normalized surface spherical harmonic functions, $\overline{R}_{nm}(\theta, \lambda)$, $\overline{S}_{nm}(\theta, \lambda)$, as in Heiskanen and Moritz (1967, eq.1-75) (herein abbrev. (HM,(1-75)) thus;

$$f(\theta,\lambda) = \sum_{n=0}^{\infty} \sum_{m=0}^{n} (\overline{a}_{nm} \cos m\lambda + \overline{b}_{nm} \sin m\lambda) \; \overline{P}_{nm}(\cos\theta) \quad (2.1)$$

$$= \sum_{n=0}^{\infty} \sum_{m=0}^{n} (\overline{a}_{nm} \overline{R}_{nm}(\theta,\lambda) + \overline{b}_{nm} \overline{S}_{nm}(\theta,\lambda) \quad (2.2)$$

$$= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \overline{c}_{nm} \overline{Y}_{nm}(\theta,\lambda) \quad (2.3)$$

$$f(\theta,\lambda) = \sum_{n=0}^{\infty} f_n \, g_n(\theta,\lambda) \quad . \quad (2.4)$$

These equivalent relations show different notations found in the literature. We will mostly use the last one where the surface spherical harmonics, $\overline{R}_{nm}$ and $\overline{S}_{nm}$ are arranged in vector form, g_n (without overbar) also fully normalized. This vector $g_n(\theta,\lambda)$ of orthogonal base functions and the vector f_n associated to it is related to (2.2) as follow

$$g_n(\theta,\lambda) = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{R}_{00} \\ \bar{R}_{10} \\ \bar{R}_{11} \\ \bar{S}_{11} \\ \bar{R}_{20} \\ \bar{R}_{21} \\ \vdots \end{bmatrix} \qquad f_n = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{a}_{00} \\ \bar{a}_{10} \\ \bar{a}_{11} \\ \bar{b}_{11} \\ \bar{a}_{20} \\ \bar{a}_{21} \\ \vdots \end{bmatrix} \quad . \tag{2.5}$$

In practice the infinite summation in (2.1) is always truncated to a maximum degree $n = N$ and the function f is thus approximated by the truncated series $\hat{f}$

$$\hat{f}(\theta,\lambda) \approx \sum_{n=0}^{N} \sum_{m=0}^{n} (\bar{a}_{nm} \cos m\lambda + \bar{b}_{nm} \sin m\lambda) \bar{P}_{nm}(\cos\theta) \, . \tag{2.6}$$

To get an equivalent truncated single subscripted series one has to truncate (2.4) at $\nu = (N+1)^2 - 1$, i.e. equation (2.6) is equivalent to

$$\hat{f}(\theta,\lambda) \approx \sum_{n=0}^{\nu} f_n \, g_n(\theta,\lambda) \tag{2.7}$$

where

$$\nu = (N+1)^2 - 1 \quad . \tag{2.8}$$

The "-1" term appears because the series (2.7) starts at $n = 0$.

Due to the orthogonality relations of the surface spherical harmonics (HM, (1-68), (1-69) and (1-74)) the coefficients in equation (2.1) are given in (HM,(1-76)) as

$$\begin{Bmatrix} \bar{a}_{nm} \\ \bar{b}_{nm} \end{Bmatrix} = \frac{1}{4\pi} \iint_{\sigma} f(\theta,\lambda) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \bar{P}_{nm}(\cos\theta)\, d\sigma \quad . \qquad (2.9)$$

Here $d\sigma$ is $\sin\theta\, d\theta\, d\lambda$, an element on the surface σ of the unit sphere. According to the notation in (2.7), (2.9) becomes simply

$$f_n = \frac{1}{4\pi} \iint_{\sigma} f(\theta,\lambda)\; g_n(\theta,\lambda)\; d\sigma \quad . \qquad (2.10)$$

With this notation, clarity in the expressions related to scalar products will follow throughout this work.

2.2 The Spherical Harmonics and the Least-Squares Method.

In the preceding section we have seen that the coefficients $\bar{a}_{nm}$ and $\bar{b}_{nm}$ in (2.6) are obtained using (2.9) because of the orthogonality relationship. In this section we want to recall that the $\bar{a}_{nm}$ and $\bar{b}_{nm}$ are more than due to the orthogonality relationship but that they are a solution of the least-squares method. The error in approximating a function $f(\theta,\lambda)$ by a truncated series like (2.6) is

$$f(\theta,\lambda) - \sum_{n=0}^{\nu} f_n\, g_n(\theta,\lambda) \quad . \qquad (2.11)$$

According to the least-squares method (Collatz, 1960, p.29, eq.4.5), one must minimize the integral of the square of the error i.e.

$$\Gamma = \iint_{\sigma} [f(\theta,\lambda) - \sum_{n=0}^{\nu} f_n\, g_n(\theta,\lambda)]^2 \, d\sigma \quad . \qquad (2.12)$$

The minimum of (2.12) is obtained by making equal to zero the differentiation of Γ with respect to the unknowns, the coefficients f_q i.e.

$$\frac{d\Gamma}{df_q} = \iint_{\sigma} [f - \sum_{n=0}^{\nu} f_n\, g_n]\, g_q \, d\sigma = 0. \qquad (2.13)$$

which can be written directly

$$\iint_{\sigma} f \; g_q \; d\sigma \;=\; \sum_{n=0}^{\nu} f_n \iint_{\sigma} g_n \; g_q \; d\sigma \quad . \qquad (2.14)$$

Now this equation reduces to (2.10) or (2.9) because of the orthogonality relationship. Thus (2.9) is a least-squares solution in the sense specified by equation (2.12). Not a solution of a least-squares adjustment but of a least-squares method. It is with this method that solves the single boundary value problem (b.v.p.) that we intend to solve the mixed b.v.p.. In the next section another notation will be presented that will be required extensively later on.

2.3 The Scalar Product of Functions.

It is widely known that the above concept of finding easily the coefficients (2.9) of a series approximating a function such as (2.1) is due to the scalar product and orthogonality relations of two functions. We shall denote the scalar product of two harmonics, $g_n(\theta, \lambda)$ as

$$(g_n, g_q) = \frac{1}{4\pi} \iint_\sigma g_n\, g_q\, d\sigma = 0 \ , \qquad n \neq q \quad . \tag{2.15}$$

This equation is equivalent to (HM,(1-68)) and shows that two harmonics of different degree or order are orthogonal on the sphere i.e. on the domain $0 \leq \theta \leq \pi$ and $0 \leq \lambda \leq 2\pi$.

The non-negative square root of (g_n, g_q) is called the norm of $g_n(\theta, \lambda)$ and is denoted by $\|g_n\|$; thus as (HM,(1-74))

$$\|g_n\| = [\,(g_n, g_n)\,]^{1/2} = \left(\frac{1}{4\pi} \iint_\sigma g_n^2(\theta, \lambda)\, d\sigma \right)^{1/2} = 1 \quad . \tag{2.16}$$

If one would apply the scalar product of g_q to $f(\theta,\lambda)$ of (2.4) he would get

$$(f, g_q) = \frac{1}{4\pi} \iint_\sigma f(\theta, \lambda)\, g_q(\theta, \lambda)\, d\sigma = \sum_{n=0}^{\infty} f_n\, (g_n, g_q) \ . \tag{2.17}$$

In view of the orthogonality (2.15) and (2.16) the only non-zero integral happens when $q = n$; thus (2.17) reduces to

$$(f, g_n) = f_n \, \|g_n\|^2 = f_n \quad . \tag{2.18}$$

This equation is the same as (2.10). This notation related to "scalar products of functions" is widely used in the literature (Kreyszig, 1972, pp.134-135), (Courant and Hilbert, 1953, p.56) and will be used throughout this work.

2.4 The Weighted Scalar Product of Functions.

Following the ideas of the previous section one could also generalize and find some sets of functions say $h_n(\theta, \lambda)$ which would be orthogonal only with respect to a weight function $W(\theta, \lambda)$ on the domain $0 \leq \theta \leq \pi$ and $0 \leq \lambda \leq 2\pi$ (Kreyszig, 1972, pp.137-138). We have

$$(h_n, h_q) = \frac{1}{4\pi} \iint_\sigma h_n(\theta, \lambda) \; h_q(\theta, \lambda) \; W(\theta, \lambda) \; d\sigma \;, \quad n \neq q \quad . \tag{2.19}$$

The norm of $h_n(\theta, \lambda)$ would now be defined as

$$\|h_n\| = \left(\frac{1}{4\pi} \iint_\sigma h_n^2(\theta, \lambda) \; W(\theta, \lambda) \; d\sigma \right)^{1/2} \quad . \tag{2.20}$$

It would be equal to 1 if h_n is orthonormal but in respect to the weight function $W(\theta, \lambda)$.

The weight function $W(\theta, \lambda)$ must be positive so one can take its square root and write say $g_n(\theta, \lambda) = W^{1/2} h_n(\theta, \lambda)$ and then (2.19) and (2.20) are equivalent to (2.15) and (2.16). Clearly if all these functions h_n, g_n are to be real, $W(\theta, \lambda)$ must be non-negative.

Like (2.17) and (2.18) if we want to expand a function as a Generalized Fourier Series (Kreyszig, 1972, p.136) of $h_n(\theta,\lambda)$ then

$$f(\theta,\lambda) = \sum_{n=0}^{\infty} c_n \, h_n(\theta,\lambda) \tag{2.21}$$

and

$$c_n = \frac{1}{4\pi \, \|h_n\|^2} \iint_{\sigma} f(\theta,\lambda) \, h_n(\theta,\lambda) \, W(\theta,\lambda) \, d\sigma \quad . \tag{2.22}$$

Equation (2.22) differs from (2.10) by the weight function which we have introduced in this section to keep the following theory as general as possible and to leave open the possibility of considering the Method of Least-Squares with weights. In fact it will be demonstrated in Chapter 9 that the weight function is required to solve the mixed altimetry-gravimetry b.v.p..

2.5 The Weighted Scalar Product of Functions on Mixed Domains.

Similarly to the previous two sections one can start with the following orthogonality relationship involving two functions $X_k(\theta,\lambda)$ and $Y(X_k(\theta,\lambda))$ of varying rank k (i.e. degree and order) and a weight function $W(\theta,\lambda)$

$$\frac{1}{4\pi}\iint_{\sigma_1} X_n\ X_q\ W\ d\sigma + \frac{1}{4\pi}\iint_{\sigma_2} Y(X_n)\ Y(X_q)\ W\ d\sigma = \left|\begin{array}{l} 0,\ n \neq q \\ 1,\ n = q \end{array}\right. \qquad (2.23)$$

Then one can think of expanding a function in a series

$$f(\theta,\lambda) = \sum_{n=0}^{\infty} E_n\ X_n(\theta,\lambda) \qquad (2.24)$$

Using the scalar product by X_q on the domain σ_1 of (2.24) gives

$$\iint_{\sigma_1} f\ X_q\ W\ d\sigma = \sum_{n=0}^{\infty} E_n \iint_{\sigma_1} X_n\ X_q\ W\ d\sigma \qquad (2.25)$$

One can derive another function from (2.24)

$$g(f(\theta,\lambda)) = \sum_{n=0}^{\infty} E_n\ \ Y(X_n) \qquad (2.26)$$

This time the scalar product by $Y(X_q)$ on the domain σ_2 of (2.26) gives

$$\iint_{\sigma_2} g\ Y(X_q)\ W\ d\sigma = \sum_{n=0}^{\infty} E_n \iint_{\sigma_2} Y(X_n)\ Y(X_q)\ W\ d\sigma \quad . \qquad (2.27)$$

One can now sum (2.25) and (2.27) to get

$$\iint_{\sigma_1} f\ X_q\ W\ d\sigma + \iint_{\sigma_2} g\ Y(X_q)\ W\ d\sigma =$$

$$= \sum_{n=0}^{\infty} E_n \left(\iint_{\sigma_1} X_n\ X_q\ W\ d\sigma + \iint_{\sigma_2} Y(X_n)\ Y(X_q)\ W\ d\sigma \right). \qquad (2.28)$$

Because of the assumption (2.23), (2.28) reduces to

$$E_n = \frac{1}{4\pi} \iint_{\sigma_1} f\ X_n\ W\ d\sigma + \frac{1}{4\pi} \iint_{\sigma_2} g\ Y(X_n)\ W\ d\sigma \ . \qquad (2.29)$$

One can verify that what we have done in this section is the same as what was done in sections 2.2 and 2.3. Chapter 3 which follows shows how this scalar product on mixed domains is used to solve the altimetry-gravimetry boundary value problem. Chapter 5 will show how one can form X_n and $Y(X_n)$ such that the assumption (2.23) is satisfied. This will permit us to use (2.29) in chapter 4 to find the coefficients E_n which defines $f(\theta, \lambda)$ and $g(f(\theta, \lambda))$ in (2.24) and (2.26).

If one sets $W = 1$ in (2.29) he gets

$$E_n = \frac{1}{4\pi} \iint_{\sigma_n} f \, X_n \, d\sigma + \frac{1}{4\pi} \iint_{\sigma_2} g \, Y(X_n) \, d\sigma \, . \quad (2.30)$$

This is quite similar to (2.10). In fact if $g = f$ then by comparing (2.24) to (2.26) we see that $Y(X_n) = X_n$. It follows that (2.30) reduces to (2.10) i.e. (2.9) which is what physical geodesists are familiar with (Colombo,1981).

CHAPTER III

THE DISTURBING POTENTIAL EXPANSION OF THE ALTIMETRY-GRAVIMETRY PROBLEM USING ORTHONORMAL BASE FUNCTIONS

This chapter follows the solution given by Arnold (1978) to solve the "altimetry-gravimetry problem" using orthonormal base functions. This chapter also shows that the use of orthonormal base functions, such as the expansion of spherical harmonics, is a solution of the Least-Squares method (Brillouin, 1916).

If one would subtract a normal or reference gravity potential U from the actual potential W to obtain a disturbing potential T that is harmonic, then T would satisfy the Laplace equation i.e.

$$\nabla^2 (W-U) = \nabla^2 T = 0 \quad . \tag{3.1}$$

Using the same polar spherical coordinates system of chapter 2, the solution of the Laplacian (3.1) is (HM, (2-152))

$$T(\theta,\lambda,r) = \frac{GM}{R} \sum_{n=2}^{\infty} \left(\frac{R}{r}\right)^{n+1} \sum_{m=0}^{n} (\bar{C}^*_{nm} \cos m\lambda + \bar{S}_{nm} \sin m\lambda) \bar{P}_{nm}(\cos\theta) \tag{3.2}$$

GM is the geocentric gravitational constant, R a constant near the earth radius, $\bar{C}^*_{nm}$ and $\bar{S}_{nm}$ are dimensionless coefficients called Stokes' constants. This solution on the surface $r = R$ is then

$$T(\theta,\lambda) = \frac{GM}{R} \sum_{n=2}^{\infty} \sum_{m=0}^{n} (\bar{C}^{*}_{nm}\cos m\lambda + \bar{S}_{nm} \sin m\lambda)\bar{P}_{nm}(\cos\theta) \; . \quad (3.3)$$

In the notation of section 2.1 equation (3.3) is written as

$$T(\theta,\lambda) = \frac{GM}{R} \sum_{n=0}^{\infty} T_n \, S_n(\theta,\lambda) \quad . \qquad (3.4)$$

In this notation the T_n coefficients are the Stokes constants and the $S_n(\theta,\lambda)$ functions without overbar are still fully normalized and are related to the surface spherical harmonics $\bar{R}_{nm}(\theta,\lambda)$ and $\bar{S}_{nm}(\theta,\lambda)$ (see equation (2.2)) as

$$[T_n] = \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \bar{C}^{*}_{20} \\ \bar{C}_{21} \\ \bar{S}_{21} \\ \bar{C}_{22} \\ \bar{S}_{22} \\ \bar{C}_{30} \\ \bar{C}_{31} \\ \bar{S}_{31} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} , \quad [S_n(\theta,\lambda)] = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \bar{R}_{20} \\ \bar{R}_{21} \\ \bar{S}_{21} \\ \bar{R}_{22} \\ \bar{S}_{22} \\ \bar{R}_{30} \\ \bar{R}_{31} \\ \bar{S}_{31} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad . \qquad (3.5)$$

The relation between the gravity anomalies Δg and T is (HM, (2-154))

$$\Delta g(\theta,\lambda,r) = -\frac{\partial T}{\partial r} - \frac{2\,T}{r} \quad . \qquad (3.6)$$

Inserting (3.2) in (3.6) gives the usual relation (HM, p.108)

$$\Delta g(\theta,\lambda,r) = \frac{GM}{R^2}\sum_{n=2}^{\infty}(n-1)\left(\frac{R}{r}\right)^{n+2}\sum_{m=0}^{n}(\bar{C}^{*}_{nm}\cos m\lambda+\bar{S}_{nm}\sin m\lambda)\bar{P}_{nm}(\cos\theta)$$

$$(3.7)$$

Again setting $r = R$ one gets

$$\Delta g(\theta,\lambda) = \frac{GM}{R^2}\sum_{n=2}^{\infty}(n-1)\sum_{m=0}^{n}(\bar{C}^{*}_{nm}\cos m\lambda+\bar{S}_{nm}\sin m\lambda)\bar{P}_{nm}(\cos\theta)$$

$$(3.8)$$

or simply

$$\Delta g(\theta,\lambda) = \frac{GM}{R^2}\sum_{n=0}^{\infty}(R_n-1)\;T_n\;S_n(\theta,\lambda) \qquad (3.9)$$

where (R_n-1) is equivalent to the term $(n-1)$ in (3.8). One can use the relation in (3.5) between vectors to verify that

$$R_n = \mathrm{INT}[(n+4)^{1/2}] \quad . \qquad (3.10)$$

This notation is necessary to shorten the equations that follow and allows a clearer presentation of the following development. In (3.8) one usually does not compute the coefficients to infinity but from degree 2 and order 0 up to degree and order N.

In (3.9) the equivalent is to compute the coefficients T_n from rank 0 up to rank $\nu-4=(N-1)^2-5$ as defined in (2.8). The number -4 reflects the absence of the coefficients $\bar{C}_{00}$, $\bar{C}_{10}$, $\bar{C}_{11}$ and $\bar{S}_{11}$ in (3.8) and (3.9). The error in finding truncated series up to degree and order N i.e. up to rank $\nu-4$ is

$$T(\theta,\lambda) - \frac{GM}{R} \sum_{n=0}^{\nu-4} T_n \; S_n(\theta,\lambda) \tag{3.11}$$

and

$$\Delta g(\theta,\lambda) - \frac{GM}{R^2} \sum_{n=0}^{\nu-4} (R_n-1) \; T_n \; S_n(\theta,\lambda) \quad . \tag{3.12}$$

If one assigns a weight $W(\theta,\lambda) = W(\sigma)$ to $T(\sigma_1)$ and $\Delta g(\sigma_2)$ where the domain $\sigma = \sigma_1 + \sigma_2$ then according to the Least-Squares method (Collatz, 1960, p.29, eq.4.5) one has the following conditions for the weighted sum of the errors

$$\Gamma = \iint_{\sigma_1} [T-\sum_{n=0}^{\nu-4} T_n S_n]^2 W(\theta,\lambda) d\sigma + \iint_{\sigma_2} [\Delta g-\sum_{n=0}^{\nu-4} (R_n-1) T_n S_n]^2 \; W(\theta,\lambda) d\sigma \quad , \tag{3.13}$$

which we want to minimize. Instead of carrying units and for simplicity T and Δg will have no units within these mixed integrals, i.e. T(no units)= T(with units)/(GM/R) and Δg(no units)=Δg(with units)/(GM/R^2). The use of GM/R and GM/R^2 will clearly show without confusion when T and Δg have units or not. This shortens the equations and clarifies the developments. In the same manner it will always be possible to transform the weight function W into a non-units function.

The condition to minimize (3.13) is

$$\frac{d\Gamma}{dT_q} = 0 \qquad . \tag{3.14}$$

Inserting (3.13) in (3.14) gives

$$\iint_{\sigma_1} [T-\Sigma T_n S_n] S_q W \, d\sigma + \iint_{\sigma_2} [\Delta g-\Sigma (R_n-1) T_n S_n] (R_q-1) S_q W \, d\sigma = 0 \tag{3.15}$$

The integral (2.15) is zero because the integration covers the complete sphere σ, but not here in (3.15) where the integration covers only the domain σ_1 or σ_2. Thus (3.15) cannot be simplified further and we are left with

$$\iint_{\sigma_1} T \; S_q \; W \; d\sigma \; + \iint_{\sigma_2} \Delta g \; (R_q-1) \; S_q \; W \; d\sigma \; =$$
$$= \sum_{n=0}^{\nu-4} T_n \left[\iint_{\sigma_1} S_n S_q W \; d\sigma + (R_q-1) \; (R_n-1) \iint_{\sigma_2} S_n S_q W \; d\sigma \right] \tag{3.16}$$

where $q = 0, 1, 2, ..., \nu-4$. Equation (3.16) in matrix notation is shown on the next page as equation (3.17) where $\mu = \nu-4 = (N+1)^2 - 5$. In (3.17) T_n (i.e. $\bar{C}^*_{nm}$ and $\bar{S}_{nm}$) are the unknowns.

Equation (3.17).

$$\begin{bmatrix} (S_0,S_0)_1+(S_0,S_0)_2 & \cdots & (S_5,S_0)_1+2(S_5,S_0)_2 & \cdots & (S_\mu,S_0)_1+(R_\mu-1)(S_\mu,S_0)_2 \\ (S_0,S_1)_1+(S_0,S_1)_2 & \cdots & (S_5,S_1)_1+2(S_5,S_1)_2 & \cdots & (S_\mu,S_1)_1+(R_\mu-1)(S_\mu,S_1)_2 \\ (S_0,S_2)_1+(S_0,S_2)_2 & \cdots & (S_5,S_2)_1+2(S_5,S_2)_2 & \cdots & (S_\mu,S_2)_1+(R_\mu-1)(S_\mu,S_2)_2 \\ (S_0,S_3)_1+(S_0,S_3)_2 & \cdots & (S_5,S_3)_1+2(S_5,S_3)_2 & \cdots & (S_\mu,S_3)_1+(R_\mu-1)(S_\mu,S_3)_2 \\ (S_0,S_4)_1+(S_0,S_4)_2 & \cdots & (S_5,S_4)_1+2(S_5,S_4)_2 & \cdots & (S_\mu,S_4)_1+(R_\mu-1)(S_\mu,S_4)_2 \\ (S_0,S_5)_1+2(S_0,S_5)_2 & \cdots & (S_5,S_5)_1+4(S_5,S_5)_2 & \cdots & (S_\mu,S_5)_1+2(R_\mu-1)(S_\mu,S_5)_2 \\ \vdots & & \vdots & & \vdots \\ (S_0,S_\mu)_1+(R_\mu-1)(S_0,S_\mu)_2 & \cdots & (S_5,S_\mu)_1+2(R_\mu-1)(S_5,S_\mu)_2 & \cdots & (S_\mu,S_\mu)_1+(R_\mu-1)(S_\mu,S_\mu)_2 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \vdots \\ T_\mu \end{bmatrix} = \begin{bmatrix} (T,S_0)_1+(\Delta g,S_0)_2 \\ (T,S_1)_1+(\Delta g,S_1)_2 \\ (T,S_2)_1+(\Delta g,S_2)_2 \\ (T,S_3)_1+(\Delta g,S_3)_2 \\ (T,S_4)_1+(\Delta g,S_4)_2 \\ (T,S_5)_1+2(\Delta g,S_5)_2 \\ \vdots \\ (T,S_\mu)_1+(R_\mu-1)(\Delta g,S_\mu)_2 \end{bmatrix}$$

If blocks of $1°\times1°$ mean Δg and T covering the complete earth were used in equation (3.17) to find the potential coefficients $\bar{C}_{nm}$ and $\bar{S}_{nm}$ up to degree N = 180 it would result in trying to solve $\mu+1 = (N+1)^2-4 = (180+1)^2-4 = 32757$ unknowns from a system of 32757 equations.

We can show that this system of equations (3.17) reduces to the solution of the single b.v.p. when only values of Δg (or T) are given. In this case, the scalar products (S_i, S_j) are zero if $i \neq j$ and unity if $i = j$. Thus the square matrix in (3.17) reduces to a diagonal matrix with terms equal to $(n-1)(n-1)$. The array to the right of the equal sign contains terms equal to $(n-1)(\Delta g, \bar{R}_{nm})$ or $(n-1)(\Delta g, \bar{S}_{nm})$. The inverse of the diagonal matrix is another diagonal matrix; thus the unknowns $[T_n]$ become simply $(\Delta g, \bar{R}_{nm})/(n-1)$ or $(\Delta g, \bar{S}_{nm})/(n-1)$ which is (E.1), the solution of the single b.v.p..

(Brillouin, 1916) showed that such a system involving the Least-Squares method applied to a set of base functions here S_n, can be solved by forming another set of orthonormal base functions say X_n. Thus instead of the solid spherical harmonics the functions $X_n(\theta, \lambda, r)$ are introduced in another series representation of the disturbing potential

$$\hat{T}(\theta, \lambda, r) = \frac{GM}{R} \sum_{n=0}^{\nu-4} E_n X_n(\theta, \lambda, r) \qquad . \qquad (3.18)$$

Using the well known Gram-Schmidt orthonormalization process one can form the orthonormal base functions X_n satisfying the orthonormality condition (3.33) with

$$X_n(\theta,\lambda,r) = u_n \left(\sum_{p=0}^{n-1} c_{np}\, X_p + L_n \right), \quad (n=0,1,2...) \qquad (3.19)$$

where u_n and c_{np} are some constants to determine. The functions L_n will herein be the solid spherical harmonics

$$L_n(\theta,\lambda,r) = S_n \left(\frac{R}{r} \right)^{R_n+1} \qquad (3.20)$$

and the functions S_n are given by (3.5). One can verify that the functions L_n are solid spherical harmonics i.e.

$$[L_n] = \begin{bmatrix} L_0 \\ L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{R}_{20} \left(\frac{R}{r}\right)^3 \\ \bar{R}_{21} \left(\frac{R}{r}\right)^3 \\ \bar{S}_{21} \left(\frac{R}{r}\right)^3 \\ \bar{R}_{22} \left(\frac{R}{r}\right)^3 \\ \bar{S}_{22} \left(\frac{R}{r}\right)^3 \\ \bar{R}_{30} \left(\frac{R}{r}\right)^4 \\ \vdots \end{bmatrix} . \qquad (3.21)$$

One can differentiate (3.18)

$$\frac{\partial T}{\partial r} = \frac{GM}{R} \sum_{n=0}^{\nu-4} E_n \frac{\partial X_n}{\partial r} \qquad (3.22)$$

which leads one to differentiate (3.19) where (3.20) is needed to obtain

$$\frac{\partial X_n}{\partial r} = u_n \left(\sum_{p=0}^{n-1} c_{np} \frac{\partial X_p}{\partial r} - \frac{(R_n+1)}{r} L_n \right) . \qquad (3.23)$$

Inserting (3.22) and (3.19) in (3.6) one obtains successively

$$\Delta g(\theta,\lambda,r) = - \frac{\partial T}{\partial r} - \frac{2\ T}{r}$$

$$= \frac{GM}{R} \left(- \sum_{n=0}^{\nu-4} E_n \frac{\partial X_n}{\partial r} - \frac{2}{r} \sum_{n=0}^{\nu-4} E_n X_n \right)$$

$$= \frac{GM}{R} \sum_{n=0}^{\nu-4} E_n \left(- \frac{\partial X_n}{\partial r} - \frac{2}{r} X_n \right)$$

$$\Delta g(\theta,\lambda,r) = \frac{GM}{R^2} \sum_{n=0}^{\nu-4} E_n\ Y(X_n) \qquad (3.24)$$

where $Y(X_n)$ was defined as

$$Y(X_n) = R \left(- \frac{\partial X_n}{\partial r} - \frac{2}{r} X_n \right) . \qquad (3.25)$$

The relations (3.22) to (3.25) were found to show how Δg is related to the new base functions $X_n(\theta,\lambda,r)$.

We can now get back at (3.13) and introduce $X_n(\theta, \lambda, r=R)$ instead of $S_n(\theta, \lambda)$; thus we now have

$$\Gamma = \iint_{\sigma_1} [T - \sum_{n=0}^{\nu-4} E_n \; X_n]^2 W \; d\sigma + \iint_{\sigma_2} [\Delta g - \sum_{n=0}^{\nu-4} E_n \; Y(X_n)]^2 W \; d\sigma \tag{3.26}$$

where again T and Δg have no units. In the preceding equation, the two functions X_n and $Y(X_n)$ are now evaluated at $r = R$; thus with $r = R$ (3.20) becomes

$$L_n(\theta, \lambda) = S_n(\theta, \lambda) \quad . \tag{3.27}$$

This also modifies (3.19) as

$$X_n(\theta, \lambda, r=R) = u_n \left(\sum_{p=0}^{n-1} c_{np} \; X_p + S_n \right), \; (n = 0, 1, 2...) \tag{3.28}$$

and (3.23) as

$$\left. \frac{\partial X_n}{\partial r} \right|_{r=R} = u_n \left(\sum_{p=0}^{n-1} c_{np} \left. \frac{\partial X_p}{\partial r} \right|_{r=R} - \frac{(R_n+1)}{R} \; S_n \right) . \tag{3.29}$$

Inserting (3.23) and (3.19) in (3.25) and evaluating at $r = R$ one gets $Y(X_n)$ at $r = R$

$$Y(X_n)\Big|_{r=R} = u_n \left(\sum_{p=0}^{n-1} c_{np}\, Y(X_p)\Big|_{r=R} + (R_n - 1)\, S_n \right) . \tag{3.30}$$

R_n was given by (3.10). The relations (3.27) to (3.30) will be required in chapter 5. We can now apply the same condition (3.14) to (3.26) which yields instead of (3.15) and (3.16) the following forms

$$\iint\limits_{\sigma_1} [T - \Sigma E_n X_n] X_q W \, d\sigma + \iint\limits_{\sigma_2} [\Delta g - \Sigma E_n Y(X_n)] Y(X_q) W \, d\sigma = 0 \tag{3.31}$$

and

$$\iint\limits_{\sigma_1} T\, X_q\, W\, d\sigma - \iint\limits_{\sigma_2} \Delta g\; Y(X_q)\; W\, d\sigma$$
$$- \sum_{n=0}^{\nu-4} E_n \left(\iint\limits_{\sigma_1} X_n\, X_q\, W\, d\sigma + \iint\limits_{\sigma_2} Y(X_n)\; Y(X_q)\; W\, d\sigma \right) = 0 \quad . \tag{3.32}$$

This time we can simplify this system of linear equations by finding X_n and $Y(X_n)$ ($n = 0, 1, 2, \ldots, (N+1)^2-5$) such that the above bracket becomes zero or one, i.e.

$$\frac{1}{4\pi}\iint_{\sigma_1} X_n\ X_q\ W\ d\sigma + \frac{1}{4\pi}\iint_{\sigma_2} Y(X_n)\ Y(X_q)\ W\ d\sigma = \left|\begin{matrix} 0 & n\neq q \\ 1 & n=q \end{matrix}\right. . \tag{3.33}$$

In chapter 5 it will be shown that the condition (3.33) can be realized using the Gram-Schmidt orthonormalization process. We will obtain X_n from (3.19), defined with the coefficients c_{np} and u_n such as

$$c_{np} = \frac{-1}{4\pi}\iint_{\sigma_1} L_n\ X_p\ W\ d\sigma - \frac{1}{4\pi}\iint_{\sigma_2} Y(L_n)\ Y(X_p)\ W\ d\sigma \tag{3.34}$$

and

$$\left(\frac{1}{u_n}\right)^2 = -\sum_{p=0}^{n-1} c_{np}^2 + \frac{1}{4\pi}\iint_{\sigma_1} L_n^2\ W\ d\sigma + \frac{1}{4\pi}\iint_{\sigma_2} Y^2(L_n)\ W\ d\sigma$$

$$p < n, \quad p = 0, 1, 2, \ldots, (n-1), \quad n = 0, 1, 2, \ldots \quad . \tag{3.35}$$

Then from (3.32) the desired coefficients are obtained as

$$E_n = \frac{1}{4\pi}\iint_{\sigma_1} T\ X_n\ W\ d\sigma + \frac{1}{4\pi}\iint_{\sigma_2} \Delta g\ Y(X_n)\ W\ d\sigma\ . \tag{3.36}$$

This relation solves the system of linear equations (3.16) and the coefficients E_n give us the disturbing potential T everywhere on and above the surface of the spherical Earth. This is Arnold's (1978) and Brillouin's (1916) proposed solution to the "mixed altimetry-gravimetry b.v.p.".

Since expansions in spherical harmonics are usually employed, one can imagine how useful it would be to get a retransformation of (3.18) into the spherical harmonics i.e. to determine the T_n coefficients from the following equality

$$\hat{T}(\theta,\lambda) = \frac{GM}{R} \sum_{n=0}^{\nu-4} E_n X_n(\theta,\lambda) = \frac{GM}{R} \sum_{n=0}^{\nu-4} T_n S_n(\theta,\lambda) \quad . \tag{3.37}$$

This will be done in chapter 8. Chapter 5 will explain the Gram-Schmidt orthonormalization process to find (3.34) and (3.35) required in (3.19) to define X_n and $Y(X_n)$. Meanwhile, the next chapter will show how to organize the computations to obtain the coefficients E_n.

One can follow the similarity between section (2.5) and this chapter which shows the applications of the mixed scalar product of functions and that we could generalize its application to more data sets then T and Δg known in more regions then σ_1 and σ_2.

One can also follow the similarity between section (2.2) and this chapter which shows that the proposed solution to the mixed b.v.p. is a solution of the least-squares method like (2.9) or (E.1) is for the single b.v.p.. According to these similarities if one considers (2.9) to have the simplest solution of the global single b.v.p. then the proposed solution might be the simplest solution to the global mixed b.v.p..

CHAPTER IV

COMPUTING THE ALTIMETRY-GRAVIMETRY COEFFICIENTS FROM MEAN GRAVITY ANOMALIES AND MEAN DISTURBING POTENTIAL VALUES

We will call the E_n coefficients of (3.18) and (3.36) the altimetry-gravimetry coefficients. We would like to numerically compute (3.36) which we rewrite here

$$E_n = F_n + G_n = \frac{1}{4\pi} \iint_{\sigma_1} T \; X_n \; W \; d\sigma + \frac{1}{4\pi} \iint_{\sigma_2} \Delta g \; Y(X_n) \; W \; d\sigma \quad . \qquad (4.1)$$

We have defined F_n and G_n as being each an integral. Also define I_n and J_n as the following two integrals

$$E_n' = I_n + J_n = \frac{1}{4\pi} \iint_{\sigma_1} T \; S_n \; W \; d\sigma + \frac{R_n}{4\pi} \iint_{\sigma_2} \Delta g \; S_n \; W \; d\sigma \quad . \qquad (4.2)$$

R_n is given by (3.10) and S_n are still the surface spherical harmonics as defined in (3.5). We know how to compute (4.2) and this is shown in this section. But first let us derive the relation between (4.1) and (4.2).

One should remember from chapter 3 and the comments around (3.26) that the integrals in (4.1) are evaluated at $r = R$. Thus we insert (3.28) and (3.30) in (4.1) to obtain

$$E_n = u_n\left(\sum_{p=0}^{n-1} c_{np}\, F_p + I_n\right) + u_n\left(\sum_{p=0}^{n-1} c_{np}\, G_p + J_n\right)$$

$$= u_n\left(\sum_{p=0}^{n-1} c_{np}\, (F_p + G_p) + I_n + J_n\right)$$

or

$$E_n = u_n\left(\sum_{p=0}^{n-1} c_{np}\, E_p + E_n'\right) \quad . \tag{4.3}$$

From the recursive relation (4.3) one sees that the integrals in (4.2) are the only computations needed to compute (4.1).

According to (3.5), (4.2) can be written at length as

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi} \iint_{\sigma_1} T(\theta,\lambda)\; W(\theta,\lambda)\; \bar{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \sin\theta \; d\lambda \; d\theta \; +$$

$$+ \frac{n-1}{4\pi} \iint_{\sigma_2} \Delta g(\theta,\lambda)\; W(\theta,\lambda)\; \bar{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \sin\theta \; d\lambda \, d\theta \tag{4.4}$$

where $T(\theta,\lambda)$, $\Delta g(\theta,\lambda)$ and $W(\theta,\lambda)$ have no units. We will use the Fast Fourier transform to compute this relation. Appendix E shows how to apply the Fast Fourier transform to the simpler case of the single boundary value problem. All the details are given there and one should refer to it to follow this discussion. First the sphere is partitioned into a finite number of discrete equiangular blocks of the size of the data available, here 1°X1° mean values for T and Δg (see figures 9.1 and 9.2). Thus we divide the spherical Earth into a regular grid as defined in appendix E, equation (E.3). The block mean values available for T, Δg and W will be defined as $\overline{T}_{ij}$, $\overline{\Delta g}_{ij}$ and $\overline{W}_{ij}$. Δ_{ij} will be the associated areas as defined in appendix E, equation (E.4).

Chapter 9 shows the numerical computations where we have verified that the weight function $W(\theta,\lambda)$ assigned to each $\overline{T}_{ij}$ and $\overline{\Delta g}_{ij}$ should be defined by

$$\overline{W}_{ij} = \begin{cases} \left(\dfrac{1}{\mathrm{RMS}\,(T_{ij})}\right)^2 & \text{if } i,j \in \sigma_1, \\ \left(\dfrac{1}{\mathrm{RMS}\,(\Delta g_{ij})}\right)^2 & \text{if } i,j \in \sigma_2 \;. \end{cases} \tag{4.5}$$

The RMS(.) is the root mean square computed as

$$\mathrm{RMS}\,(\overline{T}_{ij}) = \left(\frac{1}{4\pi} \sum_i \sum_j \overline{T}_{ij}^{\,2} \, \Delta_{ij}\right)^{1/2} \quad \text{where } i,j \in \sigma_1$$

and

$$\mathrm{RMS}\,(\overline{\Delta g}_{ij}) = \left(\frac{1}{4\pi} \sum_i \sum_j \overline{\Delta g}_{ij}^{\,2} \, \Delta_{ij}\right)^{1/2} \quad \text{where } i,j \in \sigma_2 \;.$$

If T (similarly for Δg) were constant within each block σ_{ij} then every point disturbing potential T inside the ijth block would equal its mean value $\overline{T}_{ij}$ and one could take T (and Δg) out of the integral (4.4) as follows

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{G}_{ij} \, \overline{W}_{ij} \, R_n^{ij} \iint_{\sigma_{ij}} \overline{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \sin\theta \, \Delta\lambda \, \Delta\theta \quad , \tag{4.6}$$

where we have set

$$\overline{G}_{ij} = \begin{vmatrix} \overline{T}_{ij} & \text{if } i,j \in \sigma_1 \\ \overline{\Delta g}_{ij} & \text{if } i,j \in \sigma_2 \end{vmatrix} \tag{4.7}$$

and

$$R_n^{ij} = \begin{vmatrix} 1 & \text{if } i,j \in \sigma_1 \\ n-1 & \text{if } i,j \in \sigma_2 \end{vmatrix} \quad . \tag{4.8}$$

The integral (4.6) might become applicable in the future when the block size used will be smaller then the 1 X 1 degree that we will use herein. However it is obvious that usually every point value in the ijth block is different than the mean value $\overline{T}_{ij}$ (or $\overline{\Delta g}_{ij}$) and thus this integral is not exact. Pellinen (1965) and Katsambalos (1978) have shown that for circular blocks of radius ψ_0 the Pellinen-Meissl smoothing operator β_n must be used to compute a better approximation. Colombo (1981, p.76) has shown that the de-smoothing operator η_n which is a function of the square of β_n is more appropriate and thus (4.6) becomes

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{G}_{ij} \, \overline{W}_{ij} \, R_n^{ij} \iint_{\sigma_{ij}} \overline{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \sin\theta d\lambda d\theta \quad . \tag{4.9}$$

Appendix E gives details about η_n and its computation. Equation (4.9) can be written as

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \overline{I}^{\,i}_{nm}(\theta) \sum_{j=0}^{2N-1} \overline{G}_{ij}\, \overline{W}_{ij}\, R_n^{ij} \begin{Bmatrix} J^{j}_{m}(\lambda) \\ K^{j}_{m}(\lambda) \end{Bmatrix} \tag{4.10}$$

where $\overline{I}_{nm}(\theta)$, $J^{j}_{m}(\lambda)$ and $K^{j}_{m}(\lambda)$ are defined by (E.11) and (E.13) in appendix E. Finally inserting (E.13) in (4.10) results in

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \overline{I}^{\,i}_{nm}(\theta) \left(\begin{Bmatrix} A(m) \\ -B(m) \end{Bmatrix} \sum_{j=0}^{2N-1} \overline{G}_{ij}\, \overline{W}_{ij}\, R_n^{ij} \cos mj\Delta\lambda + \begin{Bmatrix} B(m) \\ A(m) \end{Bmatrix} \sum_{j=0}^{2N-1} \overline{G}_{ij}\, \overline{W}_{ij}\, R_n^{ij} \sin mj\Delta\lambda \right) . \tag{4.11}$$

Due to the similarity of this relation with the case in appendix E where we describe the Fast Fourier "analysis" of Colombo (1981) one should be able to find a way to compute this equation also using the FFT. One can write (4.11) as

$$\begin{Bmatrix} E_{nm} \\ F_{nm} \end{Bmatrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \overline{I}^{\,i}_{nm}(\theta) \left(\begin{Bmatrix} A(m) \\ -B(m) \end{Bmatrix} RE[X^{i}_{n}(m)] + \begin{Bmatrix} B(m) \\ A(m) \end{Bmatrix} IM[X^{i}_{n}(m)] \right) \tag{4.12}$$

where

$$RE[X^{i}_{n}(m)] = \sum_{j=0}^{2N-1} y^{i}_{n}(j) \cos(mj\Delta\lambda) \tag{4.13}$$

$$IM[X_n^i(m)] = \sum_{j=0}^{2N-1} y_n^i(j)\ \sin(mj\Delta\lambda)$$

and

$$y_n^i(j) = \overline{G}_{ij}\ \overline{W}_{ij}\ R_n^{ij} \quad . \qquad (4.14)$$

$X_n^i(m)$ is still dependent on "n", while this was not the case in (E.21). Here a latitudinal row "i" of mixed values $\overline{T}_{ij}$ and $\overline{\Delta g}_{ij}$, without units, is entered in the IMSL FFTCC subroutine and the same row is reentered in FFTCC for each "n" value of R_n^{ij}. Thus each latitudinal row of data is Fourier transformed "n" times.

Equation (4.12) will be computed in chapter 9. This chapter has shown how (4.3) solves (3.36). The E_n' coefficients are obtained from (4.12) where the relation between E_n' and E_{nm} and F_{nm} is still the same as (3.5) i.e.

$$\begin{bmatrix} E_0' \\ E_1' \\ E_2' \\ E_3' \\ E_4' \\ E_5' \\ E_6' \\ E_7' \\ E_8' \\ \cdot \\ \cdot \\ \cdot \\ E_V' \end{bmatrix} = \begin{bmatrix} E_{20} \\ E_{21} \\ F_{21} \\ E_{22} \\ F_{22} \\ E_{30} \\ E_{31} \\ F_{31} \\ E_{32} \\ \cdot \\ \cdot \\ \cdot \\ F_{NN} \end{bmatrix} \quad . \qquad (4.15)$$

The coefficients c_{np} and u_n in (4.3) are given by (3.34) and (3.35). However these two relations have not been proven yet in this work and it is not clear from looking at them what computations they imply. Next chapters 5 and 6 will clarify these 2 equations. Chapter 7 will show how to compute $\overline{I}_{nm}(\theta)$ in (4.14).

CHAPTER V

THE GRAM-SCHMIDT ORTHONORMALIZATION PROCESS USING SPHERICAL HARMONICS

5.1 The Orthonormalization Process.

Here we want to find a function X_n such that it satisfies the following orthonormal relation

$$\frac{1}{4\pi}\iint X_n\, X_q\, W\, d\sigma = \begin{cases} 0 & n \neq q \\ 1 & n = q \end{cases} \tag{5.1}$$

or simply

$$(X_n, X_q) = 0, \quad n \neq q \quad ; \qquad (X_n, X_n) = \|X_n\|^2 = 1 \quad . \tag{5.2}$$

The Gram-Schmidt process can be used to form such orthonormal functions X_n $(n=0,1,2,\ldots,m)$ from a base of linearly independent functions, say L_n $(n=0,1,2,\ldots,m)$ is well known in the literature (Pearson, 1974, pp.958-963), (Courant and Hilbert, 1953, p.4 and p.50). A first notation often encountered, say method A, forms the X_n as

$$X_n = \sum_{p=0}^{n-1} c_{np}\, X_p + L_n \, , \quad n=0,1,2,3,\ldots, \tag{5.3}$$

Applying the scalar products (5.2) to (5.3), one can develop the following Table to find the coefficients c_{np}.

Table 1 A First Notation for the Gram-Schmidt Process.

$$X_0 = L_0$$

$$X_1 = L_1 + c_{10} X_0 \quad \Rightarrow c_{10} = \frac{-(L_1, X_0)}{\|X_0\|^2}$$

$$X_2 = L_2 + c_{20} X_0 + c_{21} X_1 \quad \Rightarrow c_{20} = \frac{-(L_2, X_0)}{\|X_0\|^2}$$

$$\Rightarrow c_{21} = \frac{-(L_2, X_1)}{\|X_1\|^2}$$

$$X_3 = L_3 + c_{30} X_0 + c_{31} X_1 + c_{32} X_2 \Rightarrow c_{30} = \frac{-(L_3, X_0)}{\|X_0\|^2}$$

$$\Rightarrow c_{31} = \frac{-(L_3, X_1)}{\|X_1\|^2}$$

$$\Rightarrow c_{32} = \frac{-(L_3, X_2)}{\|X_2\|^2}$$

From this Table one finds that the coefficients of method A are given by

$$c_{np} = \frac{-(L_n, X_p)}{\|X_p\|^2} , \qquad p < n, \quad n = 1, 2, 3, \ldots \quad . \tag{5.4}$$

A second notation, say method B, forms the X_n as

$$X_n = u_n \left(\sum_{p=0}^{n-1} c_{np} X_p + L_n \right) , \quad n = 0, 1, 2, \ldots \tag{5.5}$$

Compared to (5.3), the coefficients u_n are added to simplify the computations as shown below. Applying the scalar products (5.2) to (5.5), one can develop the following Table to find the coefficients c_{np} and u_n.

Table 2 A Second Notation for the Gram-Schmidt Process.

$$
\begin{aligned}
X_0 &= u_0\, L_0 && \Rightarrow (1/u_0)^2 = \|L_0\|^2 \\
X_1 &= u_1\, (L_1 + c_{10}\, X_0) && \Rightarrow c_{10} = -(L_1, X_0) \\
& && \Rightarrow (1/u_1)^2 = \|L_1\|^2 - c_{10}^2 \\
X_2 &= u_2\, (L_2 + c_{20}\, X_0 + c_{21}\, X_1) && \Rightarrow c_{20} = -(L_2, X_0) \\
& && \Rightarrow c_{21} = -(L_2, X_1) \\
& && \Rightarrow (1/u_2)^2 = \|L_2\|^2 - c_{20}^2 - c_{21}^2 \\
X_3 &= u_3\, (L_3 + c_{30}\, X_0 + c_{31}\, X_1 + c_{32}\, X_2) && \Rightarrow c_{30} = -(L_3, X_0) \\
& && \Rightarrow c_{31} = -(L_3, X_1) \\
& && \Rightarrow c_{32} = -(L_3, X_2) \\
& && \Rightarrow (1/u_3)^2 = \|L_3\|^2 - c_{30}^2 - c_{31}^2 - c_{32}^2
\end{aligned}
$$

From this Table the coefficients of method B are given by

$$c_{np} = -(L_n, X_p) \quad , \quad p < n, \quad n = 1, 2, \ldots, \tag{5.6}$$

and

$$\left(\frac{1}{u_n}\right)^2 = -\sum_{p=0}^{n-1} c_{np}^2 + \|L_n\|^2 \quad , \quad n = 0, 1, 2, \ldots \quad . \tag{5.7}$$

These coefficients inserted in (5.5) ensure us that the $X_n(\theta,\lambda)$ are orthogonal functions satisfying (5.2) or (5.1). From either method A or B one can expand a function f into a series of orthogonal functions X_n such that

$$f = \sum a_n X_n \tag{5.8}$$

where

$$a_n = (f, X_n) \tag{5.9}$$

because the X_n formed in that manner satisfy (5.2) and thus

$$(f, X_q) = \sum a_n (X_n, X_q) = a_n \|X_n\|^2 = a_n \quad . \tag{5.10}$$

It appears preferable to use method B because $\|L_n\|^2$ in (5.7) is simpler to compute than $\|X_n\|^2$ in (5.4). It will also be simpler to organize the computations such that instead of the (L_n, X_q)'s in (5.6), the simpler expression (L_n, L_q) will be required. This will be shown in the next section.

5.2 The Organization of the Computations.

It is possible to decrease the number of integrals involved in (5.6). To do this another notation is used, the one on the left side of Table 3. The left parts of Table 3 and 2 are compared to find the right part of Table 3.

Table 3 A Third Notation for the Gram-Schmidt Process

$$
\begin{array}{ll}
X_0 = g_{00}\, L_0 & \Rightarrow g_{00} = u_0 \\
X_1 = g_{10}\, L_0 + g_{11}\, L_1 & \Rightarrow g_{11} = u_1 \\
 & \Rightarrow g_{10} = u_1\, c_{10}\, g_{00} \\
X_2 = g_{20}\, L_0 + g_{21}\, L_1 + g_{22}\, L_2 & \Rightarrow g_{22} = u_2 \\
 & \Rightarrow g_{21} = u_2\, c_{21}\, g_{11} \\
 & \Rightarrow g_{20} = u_2\, [c_{21}\, g_{10} + c_{20}\, g_{00}] \\
X_3 = g_{30}\, L_0 + g_{31}\, L_1 + g_{32}\, L_2 + g_{33}\, L_3 & \Rightarrow g_{33} = u_3 \\
 & \Rightarrow g_{32} = u_3\, c_{32}\, g_{22} \\
 & \Rightarrow g_{31} = u_3\, [c_{32}\, g_{21} + c_{31}\, g_{11}] \\
 & \Rightarrow g_{30} = u_3\, [c_{32}\, g_{20} + c_{31}\, g_{10} + c_{30}\, g_{00}]
\end{array}
$$

From Table 3, the coefficients g_{pq} are given by

$$g_{pp} = u_p \quad , \quad p = 0, 1, 2, \ldots ,$$

and (5.11)

$$g_{pq} = u_p \sum_{i=q}^{p-1} c_{pi} \, g_{iq} \quad , \quad q \leq p, \quad p = 1, 2, \ldots \quad .$$

Then one inserts the left part of Table 3 in (5.6) to get

$$\begin{aligned}
c_{10} &= -(L_1, X_0) = -g_{00}(L_1, L_0) \\
c_{20} &= -(L_2, X_0) = -g_{00}(L_2, L_0) \\
c_{21} &= -(L_2, X_1) = -g_{10}(L_2, L_0) - g_{11}(L_2, L_1) \\
c_{30} &= -(L_3, X_0) = -g_{00}(L_3, L_0) \\
c_{31} &= -(L_3, X_1) = -g_{10}(L_3, L_0) - g_{11}(L_3, L_1) \\
c_{32} &= -(L_3, X_2) = -g_{20}(L_3, L_0) - g_{21}(L_3, L_1) - g_{22}(L_3, L_2)
\end{aligned} \tag{5.12}$$

From these last relations one gets

$$c_{np} = - \sum_{q=0}^{p} g_{pq}(L_n, L_q) \quad , \quad p < n \quad . \tag{5.13}$$

This is an easier way than (5.6) to compute the c_{np}. The only integrals to compute now are (L_n, L_q) in (5.13), and $||L_n||^2$ in (5.7). This is much simpler than trying to compute (L_n, X_p) and $||X_p||^2$ in (5.4) by the usual method A.

Equation (5.13) and (5.7) show that one can always simplify the computations involved in the Gram-Schmidt process to the integrations involving only the starting base functions L_n.

The order of the computations would here be 1st: $\|L_0\|^2$, u_0 (i.e. g_{00}) and X_0 then a_0, 2nd: (L_1,L_0) (i.e. c_{10}), $\|L_1\|^2$, u_1 (i.e. g_{11}) and g_{10}, X_1, a_1, 3rd: (L_2,L_0) (i.e. c_{20}), (L_2,L_1) (i.e. c_{21}), $\|L_2\|^2$, u_2 (i.e. g_{22}) and g_{20}, g_{21}, X_2 then a_2, etc..

These last two sections have shown the usual orthonormalization process and the organization of the computations for the usual single integral. However our application, equation (3.31), requires a more complicated orthonormalization involving two integrals. These sections were included to show clearly what may be less apparent in the next two sections.

5.3 The Orthonormalization Process on Mixed Domains.

We will here go on using the previous method B with (5.5)

$$X_n = u_n \left(\sum_{p=0}^{n-1} c_{np} X_p + L_n \right) . \qquad (5.14)$$

However the problem will not be to form orthonormal functions X_n that satisfy the conditions (5.1) or (5.2)

$$(X_n, X_q) = \frac{1}{4\pi} \iint_{\sigma} X_n(\sigma) \; X_q(\sigma) \; W \; d\sigma = \left| \begin{array}{ll} 0 & n \neq q \\ 1 & n = q \end{array} \right. \qquad (5.15)$$

but the following one

$$\frac{1}{4\pi} \iint_{\sigma_1} X_n \; X_q \; W \; d\sigma + \frac{1}{4\pi} \iint_{\sigma_2} Y(X_n) \; Y(X_q) \; W \; d\sigma = \left| \begin{array}{ll} 0 & n \neq q \\ 1 & n = q \end{array} \right. \qquad (5.16)$$

or simply

$$(X_n, X_q)_1 + (Y(X_n), Y(X_q))_2 = \left| \begin{array}{ll} 0 & n \neq q \\ 1 & n = q \end{array} \right. . \qquad (5.17)$$

This condition is the one previously met at (2.18) and in our solution to the "altimetry-gravimetry problem" at (3.31). $Y(X_n) = Y(X_n(\theta,\lambda))$ is a function of X_n as shown in section 2.4 and chapter 3, and accordingly one can use (5.14) to write

$$Y(X_n) = u_n \left(\sum_{p=0}^{n-1} c_{np} \, Y(X_p) + Y(L_n) \right) . \tag{5.18}$$

Equation (5.18) shows that the functions $Y(X_n)$ are formed with the same orthonormalization process as the X_n in (5.14). Equation (5.17) with the indexes 1 and 2 is the notation used in the following Gram-Schmidt orthonormalization process on mixed domains. The following also shows the details of what one must do to produce the previous Table 2 now on mixed domains.

From (5.14) one has for $n = 0$

$$X_0 = u_0 \, L_0 \quad \Rightarrow \quad \|X_0\|_1^2 = u_0^2 \, \|L_0\|_1^2 \tag{5.19}$$

and from (5.18)

$$Y(X_0) = u_0 \, Y(L_0) \quad \Rightarrow \quad \|Y(X_0)\|_2^2 = u_0^2 \, \|Y(L_0)\|_2^2 \quad . \tag{5.20}$$

One can sum these two equations to get

$$\|X_0\|_1^2 + \|Y(X_0)\|_2^2 = 1 = u_0^2 \, [\|L_0\|_1^2 + \|Y(L_0)\|_2^2] \tag{5.21}$$

where we have used (5.17) with $n = q = 0$. The right side of (5.21) can then be written as

$$(1/u_0)^2 = \|L_0\|_1^2 + \|Y(L_0)\|_2^2 \quad . \tag{5.22}$$

Equation (5.22) can be compared with is equivalent relation in Table 2. We can go on with the next functions, $n = 1$, in (5.14) and (5.18) to have

$$X_1 = u_1 \ (c_{10} \ X_0 + L_1)$$

and (5.23)

$$Y(X_1) = u_1 \ [c_{10} \ Y(X_0) + Y(L_1)] \quad .$$

Again applying the scalar product to (5.23) one gets

$$(X_1, X_0)_1 = u_1 \ c_{10} \ \|X_0\|_1^2 + u_1 \ (L_1, X_0)_1$$

and (5.24)

$$(Y(X_1), Y(X_0))_2 = u_1 \ c_{10} \ \|Y(X_0)\|_2^2 + u_1 (Y(L_1), Y(X_0))_2 \quad .$$

Summing the two equations in (5.24) results in

$$(X_1, X_0)_1 + (Y(X_1), Y(X_0))_2 = 0 =$$
$$u_1 \ c_{10} \ [\|X_0\|_1^2 + \|Y(X_0)\|_2^2] +$$
$$u_1 \ [(L_1, X_0)_1 + (Y(L_1), Y(X_0))_2] \quad . \tag{5.25}$$

According to (5.17) the first bracket [.] in (5.25) equals unity and (5.25) reduces to

$$c_{10} = -(L_1, X_0)_1 - (Y(L_1), Y(X_0))_2 \quad . \tag{5.26}$$

This relation can also be compared with its equivalent in Table 2. Applying another scalar product on (5.23) results in

$$\begin{aligned} \|X_1\|_1^2 + \|Y(X_1)\|_2^2 &= 1 = \\ & u_1^2 \{ c_{10}^2 [\|L_1\|_1^2 + \|X_0\|_1^2] + 2\, c_{10} (L_1, X_0)_1 \} + \\ & u_1^2 \{ c_{10}^2 [\|Y(L_1)\|_2^2 + \|Y(X_0)\|_2^2] + 2 c_{10} (Y(L_1), Y(X_0))_2 \} . \end{aligned} \tag{5.27}$$

Using (5.26), (5.27) reduces to

$$(1/u_1)^2 = -c_{10}^2 + \|L_1\|_2^2 + \|Y(L_1)\|_2^2 \quad . \tag{5.28}$$

This relation can also be compared to its equivalent in Table 2. Proceeding on for $n = 2, 3, \ldots$ and comparing to Table 2 one finds the coefficients to be given by

$$c_{np} = -(L_n, X_p)_1 - (Y(L_n), Y(X_p))_2 \quad . \tag{5.29}$$

and

$$\left(\frac{1}{u_n}\right)^2 = -\sum_{p=0}^{n-1} c_{np}^2 + \|L_n\|_1^2 + \|Y(L_n)\|_2^2 \quad . \tag{5.30}$$

One can compare these relations with (5.6) and (5.7). One should also compare the equivalence of these two relations with the previously mentioned equations (3.32) and (3.33) which were given without proof. The above notation will be used in the following discussion. However one should not forget its relation with the "altimetry-gravimetry problem" and that the integrals 1 and 2 implied in (5.29) and (5.30), each covers only a fraction of a sphere.

5.4 The Organization of the Computations with Spherical Harmonics.

The last section left us with equations (5.29) and (5.30). We have used the first three sections of this chapter to explain the reason for these two relations in our solution of the "altimetry-gravimetry problem" of chapter 3, equations (3.32) and (3.33). The computation of these two relations would be practically impossible without the following organization of the computations.

First the number of integrations required can be simplified as it was done in section 5.2. Following Table 3 one can write for the mixed case

$$X_0 = g_{00}\ L_0 \quad , \quad Y(X_0) = g_{00}\ Y(L_0) \quad . \tag{5.31}$$

Comparing (5.31) to Table 2 one gets

$$g_{00} = u_0 \quad . \tag{5.32}$$

Again from Table 3

$$X_1 = g_{10}\ L_0 + g_{11}\ L_1 \ , \quad Y(X_1) = g_{10}\ Y(L_0) + g_{11}\ Y(L_1)$$

and comparing it to Table 2 one gets

$$g_{11} = u_1 \ , \quad g_{10} = u_1\ c_{10}\ g_{00} \quad . \tag{5.33}$$

Going on, one finds out that (5.11) is still valid for the mixed case. Similarly to (5.12) one can write

$$c_{10}=-(L_1,X_0)_1+(Y(L_1),Y(X_0))_2=-g_{00}[(L_1,L_0)_1+(Y(L_1),Y(L_0))_2]$$

The relation between Table 3 and (5.13) is established and this for the mixed case, it is

$$c_{np} = - \sum_{q=0}^{p} g_{pq} \, [(L_n,L_q)_1 + (Y(L_n),Y(L_q)_2] \quad . \quad (5.34)$$

To find out about the functions L_n we now look back to chapter 3, the altimetry-gravimetry problem. The functions L_n in (5.34) comes from (5.14) which is (3.19) where L_n is defined by (3.20) as the solid spherical harmonics. It should be remembered from the comments surrounding equations (3.24) and (3.25) that all the integrals in (3.24), (3.32) and (3.33) are computed near the earth's surface, $r = R$. Because $r = R$ we found in (3.25) that

$$L_n = S_n \quad , \quad (5.35)$$

and thus in (5.34)

$$(L_n,L_p)_1 = (S_n,S_p)_1 \quad (5.36)$$

and in (5.30)

$$\|L_n\|_1^2 = \|S_n\|_1^2 \quad . \quad (5.37)$$

Thus the integrals in (5.34) involve simply the surface spherical harmonics.

To discover the functions $Y(L_n)$ in (5.34) one must go back to its definition (3.25), replace X_n by L_n and use the definition of L_n in (3.20) and differentiate, i.e.

$$Y(L_n) = R\left(- \frac{\partial L_n}{\partial r} - \frac{2}{r} L_n \right)$$

$$= - S_n (R_n+1) \left(\frac{R}{r}\right)^{R_n} \left(-\frac{R}{r}\right)^2 - \frac{2R}{r} S_n \left(\frac{R}{r}\right)^{R_n+1}$$

$$= (R_n-1) S_n \left(\frac{R}{r}\right)^{R_n+2} \qquad (5.38)$$

where R_n is given by (3.10). As a check we also had from (3.25) and (3.25)

$$Y(X_n) = R\left(- \frac{\partial X_n}{\partial r} - \frac{2}{r} X_n \right)$$

$$= -u_n R \left(\sum_{p=0}^{n-1} c_{np} \frac{\partial X_p}{\partial r} - \frac{(R_n+1)}{r} L_n \right) - \frac{2R}{r} u_n \left(\sum_{p=0}^{n-1} c_{np} X_p + L_n \right)$$

$$= u_n \left(\sum_{p=0}^{n-1} c_{np} Y(X_p) + (R_n-1) \frac{R}{r} L_n \right) . \qquad (5.39)$$

Comparing (5.39) with (5.18) and using (3.20) one finds that

$$Y(L_n) = L_n (R_n-1) \frac{R}{r} = S_n (R_n-1) \left(\frac{R}{r}\right)^{R_n+2} \qquad (5.40)$$

which equals (5.38) as it should. Since the integrals (5.34) and (5.30) must be computed at $r = R$, (5.38) or (5.40) becomes

$$Y(L_n)\Big|_{r=R} = (R_n-1)\; S_n \quad . \tag{5.41}$$

Thus we have just found out from (5.41) that the scalar products in (5.34) and (5.30) become simply

$$(Y(L_n), Y(L_p))_2 = (R_n-1)(R_p-1)\;(S_n, S_p)_2 \tag{5.42}$$

and

$$\|Y(L_n)\|_2^2 = (R_n-1)^2\;\|S_n\|_2^2 \quad . \tag{5.43}$$

where S_n are simply the surface spherical harmonics, R_n is given by (3.10) and the term (R_n-1) is equivalent to $(n-1)$ in equation (3.8). Finally (5.34) and (5.30) are simply

$$c_{np} = -\sum_{q=0}^{p} g_{pq}\left((S_n, S_q)_1 + (R_n-1)(R_q-1)(S_n, S_q)_2\right) \tag{5.44}$$

and

$$\left(\frac{1}{u_n}\right)^2 = -\sum_{p=0}^{n-1} c_{np}^2 + \|S_n\|_1^2 + (R_n-1)^2\;\|S_n\|_2^2 \quad . \tag{5.45}$$

The coefficients g_{pq} are still given by (5.11). While we had no idea of the computations involved by (3.32) and (3.33) now it is becoming much clearer with these integrals of surface spherical harmonics $S_n(\theta,\lambda)$. These integrals will be solved in the next chapter.

To close this chapter we have used (3.5) and (3.8) in (5.44) and (5.45) to show a table of the relations between the coefficients u_n, c_{np} and the harmonics S_n, $\bar{R}_{nm}$ and $\bar{S}_{nm}$.

Table 4 The Equivalence Between Different Notations Used With Harmonic Coefficients.

u_0	$\Leftrightarrow$	S_0^2	$\Leftrightarrow$	$\bar{R}_{20}^2$	$\Leftrightarrow$	A_{2020}
u_1	$\Leftrightarrow$	S_1^2	$\Leftrightarrow$	$\bar{R}_{21}^2$	$\Leftrightarrow$	A_{2121}
u_2	$\Leftrightarrow$	S_2^2	$\Leftrightarrow$	$\bar{S}_{21}^2$	$\Leftrightarrow$	B_{2121}
u_3	$\Leftrightarrow$	S_3^2	$\Leftrightarrow$	$\bar{R}_{22}^2$	$\Leftrightarrow$	A_{2222}
u_4	$\Leftrightarrow$	S_4^2	$\Leftrightarrow$	$\bar{S}_{22}^2$	$\Leftrightarrow$	B_{2222}
u_5	$\Leftrightarrow$	S_5^2	$\Leftrightarrow$	$\bar{R}_{30}^2$	$\Leftrightarrow$	A_{3030}
$\vdots$		$\vdots$		$\vdots$		$\vdots$

Table 4 The Equivalence Between Different Notations Used With Harmonic Coefficients (continued)

c_{10}	$\Leftrightarrow$	S_1S_0	$\Leftrightarrow$	$\overline{R}_{21}\overline{R}_{20}$	$\Leftrightarrow$	A_{2120}
c_{20}	$\Leftrightarrow$	S_2S_0	$\Leftrightarrow$	$\overline{S}_{21}\overline{R}_{20}$	$\Leftrightarrow$	D_{2120}
c_{21}	$\Leftrightarrow$	S_2S_1	$\Leftrightarrow$	$\overline{S}_{21}\overline{R}_{21}$	$\Leftrightarrow$	D_{2121}
c_{30}	$\Leftrightarrow$	S_3S_0	$\Leftrightarrow$	$\overline{R}_{22}\overline{R}_{20}$	$\Leftrightarrow$	A_{2220}
c_{31}	$\Leftrightarrow$	S_3S_1	$\Leftrightarrow$	$\overline{R}_{22}\overline{R}_{21}$	$\Leftrightarrow$	A_{2221}
c_{32}	$\Leftrightarrow$	S_3S_2	$\Leftrightarrow$	$\overline{R}_{22}\overline{S}_{21}$	$\Leftrightarrow$	C_{2221}
c_{40}	$\Leftrightarrow$	S_4S_0	$\Leftrightarrow$	$\overline{S}_{22}\overline{R}_{20}$	$\Leftrightarrow$	D_{2220}
c_{41}	$\Leftrightarrow$	S_4S_1	$\Leftrightarrow$	$\overline{S}_{22}\overline{R}_{21}$	$\Leftrightarrow$	D_{2221}
c_{42}	$\Leftrightarrow$	S_4S_2	$\Leftrightarrow$	$\overline{S}_{22}\overline{S}_{21}$	$\Leftrightarrow$	B_{2221}
c_{43}	$\Leftrightarrow$	S_4S_3	$\Leftrightarrow$	$\overline{S}_{22}\overline{R}_{22}$	$\Leftrightarrow$	D_{2222}
c_{50}	$\Leftrightarrow$	S_5S_0	$\Leftrightarrow$	$\overline{S}_{30}\overline{R}_{20}$	$\Leftrightarrow$	A_{3020}
$\vdots$		$\vdots$		$\vdots$		$\vdots$

CHAPTER VI

THE NEED TO INTEGRATE TWO ASSOCIATED LEGENDRE FUNCTIONS.

According to the relation in (3.5) between $S_n(\theta,\lambda)$ and the $\overline{R}_{nm}(\theta,\lambda)$ and $\overline{S}_{nm}(\theta,\lambda)$ the integrals in (5.44) and (5.45) can be written at length as

$$\begin{Bmatrix} A_{nmpq} \\ B_{nmpq} \\ \\ C_{nmpq} \\ D_{nmpq} \end{Bmatrix} = \frac{1}{4\pi} \iint_{\sigma_1} \overline{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \\ \\ \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \overline{P}_{pq}(\cos\theta) \begin{Bmatrix} \cos q\lambda \\ \sin q\lambda \\ \\ \sin q\lambda \\ \cos q\lambda \end{Bmatrix} W(\theta,\lambda) \sin\theta d\lambda d\theta +$$

$$+ \frac{R_{np}}{4\pi} \iint_{\sigma_2} \overline{P}_{nm}(\cos\theta) \begin{Bmatrix} \cos m\lambda \\ \sin m\lambda \\ \\ \cos m\lambda \\ \sin m\lambda \end{Bmatrix} \overline{P}_{pq}(\cos\theta) \begin{Bmatrix} \cos q\lambda \\ \sin q\lambda \\ \\ \sin q\lambda \\ \cos q\lambda \end{Bmatrix} W(\theta,\lambda) \sin\theta \, d\lambda \, d\theta \tag{6.1}$$

where

$$R_{np} = (n-1)(p-1) \quad . \tag{6.2}$$

The weight function $\overline{W}_{ij}$ is defined by (4.5). Again it has no units for the reason explained after equation (3.13). Also there is no relation between the indices n, p and q here and the ones in (5.44) and (5.45). Table 4 shows the relation between the c_{np} and u_n coefficients and the A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq}. Equation (6.1) will be computed using Fast Fourier transform i.e. with the same kind of development that was performed in appendix E and chapter 4.

In practice the area of the sphere is subdivided into a number of blocks (viz. 5°X5°, 1°X1°, 30'X30', see Figures 1 and 2 in Chapter 9) and the weight function $W(\theta,\lambda)$ can be assumed constant over any such block and thus, the above integrals (6.1) can be rewritten as

$$\begin{array}{l} A_{nmpq} \\ B_{nmpq} \\ C_{nmpq} \\ D_{nmpq} \end{array} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R^{ij}_{np} \, \overline{I}^{\,i}_{nmpq}(\theta) \begin{array}{l} J^{j}_{mq}(\lambda) \\ K^{j}_{mq}(\lambda) \\ L^{j}_{mq}(\lambda) \\ M^{j}_{mq}(\lambda) \end{array} . \qquad (6.3)$$

In (6.3) we have set

$$R^{ij}_{np} = \left| \begin{array}{ll} 1 & \text{if } i,j \in \sigma_1 \\ & \\ (n-1)(p-1) & \text{if } i,j \in \sigma_2 \ . \end{array} \right. \qquad (6.4)$$

In (6.3) we have also gathered together all the terms dependent on θ as

$$\overline{I}^{\,i}_{nmpq}(\theta) = \int_{\theta_i}^{\theta_{i+1}} \overline{P}_{nm}(\cos\theta) \; \overline{P}_{pq}(\cos\theta) \; \sin\theta \; d\theta \ . \qquad (6.5)$$

These integrals involve the integration of two associated Legendre functions and we will develop in next chapter 7 the recurrence relations that compute them efficiently. Also in (6.3) we have gathered together all the terms dependent on λ as

$$\begin{matrix} J^j_{mq} \\ K^j_{mq} \\ L^j_{mq} \\ M^j_{mq} \end{matrix} = \int_{\lambda_j}^{\lambda_{j+1}} \begin{matrix} \cos m\lambda \; \cos q\lambda \\ \sin m\lambda \; \sin q\lambda \\ \cos m\lambda \; \sin q\lambda \\ \sin m\lambda \; \cos q\lambda \end{matrix} \, d\lambda = \frac{1}{2} \int_{j\Delta\lambda}^{j\Delta\lambda + \Delta\lambda} \begin{matrix} \cos(m-q)\lambda + \cos(m+q)\lambda \\ \cos(m-q)\lambda - \cos(m+q)\lambda \\ \sin(m+q)\lambda - \sin(m-q)\lambda \\ \sin(m+q)\lambda + \sin(m-q)\lambda \end{matrix} \, d\lambda \quad . \tag{6.6}$$

Since in practice we are using blocks of size $\Delta\lambda$, we have replace the integration limit λ_j by $j\Delta\lambda$. The integration of (6.6) is straightforward and gives

$$\begin{aligned}
J^j_{mq} &= A(m-q) \; \cos(m-q)j\Delta\lambda + B(m-q) \; \sin(m-q)j\Delta\lambda + \\
& A(m+q) \; \cos(m+q)j\Delta\lambda + B(m+q) \; \sin(m+q)j\Delta\lambda \\
K^j_{mq} &= A(m-q) \; \cos(m-q)j\Delta\lambda + B(m-q) \; \sin(m-q)j\Delta\lambda + \\
& -A(m+q) \; \cos(m+q)j\Delta\lambda - B(m+q) \; \sin(m+q)j\Delta\lambda \\
L^j_{mq} &= B(m-q) \; \cos(m-q)j\Delta\lambda - A(m-q) \; \sin(m-q)j\Delta\lambda + \\
& -B(m+q) \; \cos(m+q)j\Delta\lambda + A(m+q) \; \sin(m+q)j\Delta\lambda \\
M^j_{mq} &= -B(m-q) \; \cos(m-q)j\Delta\lambda + A(m-q) \; \sin(m-q)j\Delta\lambda + \\
& -B(m+q) \; \cos(m+q)j\Delta\lambda + A(m+q) \; \sin(m+q)j\Delta\lambda
\end{aligned} \tag{6.7}$$

where A(k) and B(k) are defined as

$$A(k) = \begin{cases} \dfrac{\sin(k\Delta\lambda)}{2k} & \text{if } k \neq 0 \\[2ex] \dfrac{\Delta\lambda}{2} & \text{if } k = 0 \end{cases}$$

$$B(k) = \begin{cases} \dfrac{\cos(k\Delta\lambda) - 1}{2k} & \text{if } k \neq 0 \\[2ex] 0 & \text{if } k = 0 \end{cases} \quad . \qquad (6.8)$$

Inserting the first relation of (6.7) in (6.3) gives

$$A_{nmpq} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R_{np}^{ij} \, \overline{I}_{nmpq}^{i}(\theta) \, [A(m-q) \cos(m-q) j\Delta\lambda + B(m-q) \sin(m-q) j\Delta\lambda + A(m+q) \cos(m+q) j\Delta\lambda + B(m+q) \sin(m+q) j\Delta\lambda] \quad . \qquad (6.9)$$

We can write (6.9) as

$$\begin{aligned} A_{nmpq} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \overline{I}_{nmpq}^{i}(\theta) \; [A(m-q) & \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R_{np}^{ij} \cos(m-q) j\Delta\lambda \; + \\ + \; B(m-q) & \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R_{np}^{ij} \sin(m-q) j\Delta\lambda \; + \\ + \; A(m+q) & \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R_{np}^{ij} \cos(m+q) j\Delta\lambda \; + \\ + \; B(m+q) & \sum_{j=0}^{2N-1} \overline{W}_{ij} \, R_{np}^{ij} \sin(m+q) j\Delta\lambda \end{aligned} \quad . \qquad (6.10)$$

Similar to (6.10) one obtains the relations for the other coefficients B_{nmpq}, C_{nmpq} and D_{nmpq} by inserting (6.7) in (6.3) to get

$$
\begin{matrix} A_{nmpq} \\ B_{nmpq} \\ C_{nmpq} \\ D_{nmpq} \end{matrix} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \bar{I}^{i}_{nmpq}(\theta) \left(\begin{matrix} A(m-q) \\ A(m-q) \\ B(m-q) \\ -B(m-q) \end{matrix} \sum_{j=0}^{2N-1} \bar{W}_{ij} \, R^{ij}_{np} \cos(m-q) j\Delta\lambda \; + \right.
$$

$$
+ \begin{matrix} B(m-q) \\ B(m-q) \\ -A(m-q) \\ A(m-q) \end{matrix} \sum_{j=0}^{2N-1} \bar{W}_{ij} \, R^{ij}_{np} \sin(m-q) j\Delta\lambda \; +
$$

$$
+ \begin{matrix} A(m+q) \\ -A(m+q) \\ -B(m+q) \\ -B(m+q) \end{matrix} \sum_{j=0}^{2N-1} \bar{W}_{ij} \, R^{ij}_{np} \cos(m+q) j\Delta\lambda \; +
$$

$$
\left. + \begin{matrix} B(m+q) \\ -B(m+q) \\ A(m+q) \\ A(m+q) \end{matrix} \sum_{j=0}^{2N-1} \bar{W}_{ij} \, R^{ij}_{np} \sin(m+q) j\Delta\lambda \right). \tag{6.11}
$$

As in chapter 4 we can here again use the Fast Fourier transform to compute (6.11). Because of (E.22) we set

$$
\mathrm{REAL}[X^{i}_{np}(m+q)] = \sum_{j=0}^{2N-1} Y^{i}_{np}(j) \cos(m+q) j\Delta\lambda
$$

$$
\mathrm{IMAG}[X^{i}_{np}(m+q)] = \sum_{j=0}^{2N-1} Y^{i}_{np}(j) \sin(m+q) j\Delta\lambda
$$

$$
\mathrm{REAL}[X^{i}_{np}(m-q)] = \sum_{j=0}^{2N-1} Y^{i}_{np}(j) \cos(m-q) j\Delta\lambda
$$

$$\mathrm{IMAG}[X_{np}^{i}(m-q)] = \sum_{j=0}^{2N-1} y_{np}^{i}(j)\ \sin(m-q)j\Delta\lambda \tag{6.12}$$

where

$$y_{np}^{i}(j) = \overline{W}_{ij}\ R_{np}^{ij}\ . \tag{6.13}$$

Using the preceding relations one can write (6.11) as

$$\begin{matrix} A_{nmpq} \\ B_{nmpq} \\ \\ C_{nmpq} \\ D_{nmpq} \end{matrix} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \overline{I}_{nmpq}^{\,i}(\theta) \left(\begin{matrix} A(m-q) \\ A(m-q) \\ \\ B(m-q) \\ -B(m-q) \end{matrix} \ \mathrm{REAL}\ [\ X_{np}^{i}(m-q)\ ] + \begin{matrix} B(m-q) \\ B(m-q) \\ \\ -A(m-q) \\ A(m-q) \end{matrix} \ \mathrm{IMAG}\ [\ X_{np}^{i}(m-q)\ ] + \begin{matrix} A(m+q) \\ -A(m+q) \\ \\ -B(m+q) \\ -B(m+q) \end{matrix} \ \mathrm{REAL}\ [\ X_{np}^{i}(m+q)\ ] + \begin{matrix} B(m+q) \\ -B(m+q) \\ \\ A(m+q) \\ A(m+q) \end{matrix} \ \mathrm{IMAG}\ [\ X_{np}^{i}(m+q)\ ] \right) . \tag{6.14}$$

Here X_{np}^{i} is dependent on n and p. This implies that a latitudinal row "i" of $W(\sigma_1)$

and $(n-1)(p-1)W(\sigma_2)$ are entered in the IMSL FFTCC subroutine and the same row is reentered in FFTCC for each possible value of n and p. More details concerning the computation of (6.14) will be given in chapters 8 and 9.

This chapter has shown how to use Fast Fourier transform to compute the coefficients A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq} which are the elements forming the symmetric positive definite matrix in equation (3.17). It is also shown that to compute these coefficients it is required to compute $\bar{I}_{nmpq}(\theta)$, the integrals of the product of two associated Legendre functions. The following chapter 7 will show how to compute these integrals $\bar{I}_{nmpq}(\theta)$ which are defined in (6.5).

CHAPTER VII

INTEGRATING ASSOCIATED LEGENDRE FUNCTIONS.

7.1 Integrating One Associated Legendre Functions.

This section describes how integral (4.17) that we reproduce here

$$\overline{I}^{\,i}_{nm}(\theta) = \int_{\theta_i}^{\theta_{i+1}} \overline{P}_{nm}(\cos\theta)\ \sin\theta\ d\theta = \int_{t_S}^{t_N} \overline{P}_{nm}(t)\ dt = \overline{I}_{nm}(t_S, t_N) \tag{7.1}$$

is efficiently and stably computed using recurrence relations. These results are known (Paul, 1978) but they are a required preamble to new developments shown in the next section regarding the integration of two associated Legendre functions instead of the one shown here in (7.1). The development of the recurrence relations of this section is reproduced in appendices B, C and D for reference in developing the more complicated relations of next section. The notation used in this section is consistent with the next section and can easily be recognized in the Fortran routines PNMI and PNMI2 given in appendix H.

We remember having divided the spherical earth into σ_{ij} blocks for which θ_i and θ_{i+1} are respectively the northern and southern geocentric colatitude of each block (see chapters 4 and 6, and appendix E). Accordingly we have set in (7.1) $t_S=\cos\theta_{SOUTH}$ and $t_N=\cos\theta_{NORTH}$ and also $t=\cos\theta$ with $dt=-\sin\theta d\theta$.

From (Paul, 1978, eq.20a) or our appendix C the fully normalized recurrence relation solving (7.1) is

$$\bar{I}_{nm}(t_S,t_N) = \frac{n-2}{n+1}\,\frac{a(n,m)}{a(n-1,m)}\,\bar{I}_{n-2,m}(t_S,t_N) - $$

$$- a(n,m)\,\frac{1-t^2}{n+1}\,\bar{P}_{n-1,m}(t)\Bigg|_{t_S}^{t_N} \quad , \; m \neq n \, , \qquad (7.2)$$

where

$$a(n,m) = \left(\frac{(2n+1)\,(2n-1)}{(n+m)\;(n-m)}\right)^{1/2} \quad . \qquad (7.3)$$

For $m = n$, one finds from (Gleason, 1983, p.15) or our appendix C

$$\bar{I}_{nn}(t_S,t_N) = \frac{nb(n)b(n-1)}{n+1}\,\bar{I}_{n-2,n-2}(t_S,t_N) + \frac{t}{n+1}\,\bar{P}_{nn}(t)\Bigg|_{t_S}^{t_N} \qquad (7.4)$$

where

$$b(n) = \left(\frac{(2n+1)}{2n}\right)^{1/2} , \quad n > 1 \;\; ; \quad b(1) = 3^{1/2} \; . \qquad (7.5)$$

The required fully normalized associated Legendre functions $\bar{P}_{nm}(t)$ are also computed from recurrence relations. These can also be found in (Paul 1978, eq.13a and 21a)

$$\overline{P}_{nm}(t) = a(n,m)\ t\ \overline{P}_{n-1,m}(t) - \frac{a(n,m)}{a(n-1,m)}\ \overline{P}_{n-2,m}(t),\quad m \neq n, \tag{7.6}$$

and for $m = n$

$$\overline{P}_{nn}(t) = b(n)\ (1-t^2)^{1/2}\ \overline{P}_{n-1,n-1}(t) \quad . \tag{7.7}$$

The computation of all the above recurrences is simplified because they share the same coefficients $a(n,m)$ and $b(n)$. The starting values for all these recurrences are also given in (Paul, 1978, eq.26a) and are

$$\overline{P}_{00}(t) = 1,\ \overline{P}_{10}(t) = 3^{1/2}\ t,\ \overline{P}_{11}(t) = [3(1-t^2)]^{1/2}\ ,$$

$$\overline{I}_{00}(t_S,t_N) = t_N - t_S,\quad \overline{I}_{10}(t_S,t_N) = \frac{3^{1/2}}{2}(t_N^2 - t_S^2),$$

$$\overline{I}_{11}(t_S,t_N) = \frac{3^{1/2}}{2}\left[t(1-t^2)^{1/2} - \arccos(t)\right]\Bigg|_{t_S}^{t_N} ,$$

$$\overline{P}_{nm}(t) = 0 \text{ and } \overline{I}_{nm}(t) = 0 \text{ if } m > n \quad . \tag{7.8}$$

(Gerstl, 1980) and (Gleason, 1983) show that (7.4) quickly becomes unstable in polar regions but also that such instability arises at higher degrees and orders in mid-latitude regions. They show how to overcome this instability problem by using the backward version of (7.4)

$$\bar{I}_{nn}(t_S,t_N) = \frac{1}{(n+2)\,b(n+2)\,b(n+1)}\Big[(n+3)\,\bar{I}_{n+2,n+2}(t_S,t_N) - t\,\bar{P}_{n+2,n+2}(t)\Big|_{t_S}^{t_N}\Big] \qquad (7.9)$$

obtained by directly inverting (7.4). Wherever the forward recurrence (7.4) is unstable the backward recurrence (7.9) is stable and vice-versa. To use (7.9), the starting values required are $\bar{I}_{NMAX,NMAX}(t_S,t_N)$ and $\bar{I}_{NMAX-1,NMAX-1}(t_S,t_N)$ where NMAX reflects the maximum degrees implemented. These are obtained by integrating a McLaurin series and the result is given in appendix D as

$$\bar{I}_{nn}(t_S,t_N) = -\,b(n)\,b(n-1)\ldots b(1)\;y^{n+2}\Big[\frac{1}{n+2} + \frac{1}{2}\frac{y^2}{n+4} + \frac{1}{2}\frac{3}{4}\frac{y^4}{n+6} + \frac{1}{2}\frac{3}{4}\frac{5}{6}\frac{y^6}{n+8} + \ldots\Big]\Bigg|_{y_S}^{y_N} \qquad (7.10)$$

where $y^{1/2} = (1-t^2) = \sin\theta$ and $b(n)$ is defined at (7.5). To attain a desired accuracy of ε, e.g. $\varepsilon = 10^{-12}$, a sufficient number of terms "M" needed in the series expansion (7.10) is given by (Gerstl, 1980) as

$$M = 1 + \mathrm{INT}(M_0) \tag{7.11}$$

where

$$M_0 = \frac{1 + \ln(\varepsilon)}{\ln(x)} \quad , \quad x = \sin^2\left(\frac{\theta_N + \theta_S}{2}\right) \quad . \tag{7.12}$$

INT denotes the integer part of the argument M_0. (Gerstl, 1980) shows that the condition number

$$k = \frac{NMAX}{(NMAX+1)(\sin\theta_N)^2} \tag{7.13}$$

can determine whether the forward (7.4) or the backward (7.9) is most stable for any given θ_S, θ_N and NMAX situation. If $k < 1$ then the forward sectorial recurrence should be used and if $k > 1$ the backward recurrence should be used.

All the algorithms of this section were programmed in Fortran and can be found in the routine PNMI of appendix G. The first version of this efficient and stable routine was developed by Gleason (1983). Even if it looked rather complicated to compute the $I_{nn}(t_S, t_N)$'s, the integrals of the sectorials $P_{nn}(\cos\theta)$, one should be aware that this problem of instability is due to compute the rather simple integrals $I_{nn} = c(n) \int \sin^{n+1}\theta \, d\theta$, since the $P_{nn}(\cos\theta)$'s are simply equal to $c(n) \sin^n\theta$. Here $c(n)$ are integer values depending on "n".

7.2 Integrating the Product of Two Associated Legendre Functions.

We will now develop the recurrence relations that solves the integral (6.5) which we rewrite non-normalized and with $t = \cos\theta$

$$I_{nmpq}(t_S, t_N) = I_{pqnm}(t_S, t_N) = \int_{t_S}^{t_N} P_{nm}(t) P_{pq}(t)\, dt. \quad (7.14)$$

We will later need to remember the symmetry in the indices "nm" and "pq".

It might be the first time that these recurrence formulae are developed but only because it is the first time that they are required in an application. These derivations are simple and follow the integration of one associated Legendre function given in appendices B, C and D.

As in appendix B where we derived (7.2), we start with equation (A.4) from appendix A multiplied by $P_{pq}(t)$

$$A = \int (1-t^2) \frac{dP_{nm}}{dt} P_{pq} dt = \int (n+1) t P_{nm} P_{pq} dt - (n-m+1) \ I_{n+1,mpq} \quad . \quad (7.15)$$

In this section the two following abbreviations are used to simplify the writing

$$P_{nm} = P_{nm}(t) \quad \text{and} \quad I_{nmpq} = I_{nmpq}(t_S, t_N) \tag{7.16}$$

and the integration limits t_N and t_S are omitted as it was done in (7.15). Integrating by parts the left integral in (7.15) with

$$u=(1-t^2)P_{pq}, \quad du=-2tP_{pq}dt+(1-t^2)\frac{dP_{pq}}{dt}dt, \quad dv=\frac{dP_{nm}}{dt}dt, \quad v=P_{nm}, \tag{7.17}$$

one gets

$$A = (1-t^2)P_{nm}P_{pq}\Bigg|_{t_S}^{t_N} + \int 2tP_{nm}P_{pq}dt - \int (1-t^2)\frac{dP_{pq}}{dt} P_{nm}\, dt \quad . \tag{7.18}$$

For the right integral in (7.18) equation (A.5) is used;

$$\int (1-t^2)\frac{dP_{pq}}{dt}P_{nm}dt = \int -ptP_{nm}P_{pq}dt + (p+q)\; I_{nm,p-1,q} \quad . \tag{7.19}$$

Inserting (7.19) in (7.18) and then equating (7.18) to the right side of (7.15) one gets

$$(1-t^2)P_{nm}P_{pq}\Bigg|_{t_S}^{t_N} = \int (n-p-1)tP_{nm}P_{pq}dt + (p+q)I_{nm,p-1,q} - \\ - (n-m+1)I_{n+1,mpq} \quad . \tag{7.20}$$

The integral left in (7.20) is taken from equation (A.6)

$$(2n+1)\int tP_{nm}P_{pq}dt = (n-m+1)I_{n+1,mpq} + (n+m)I_{n-1,mpq} \quad . \tag{7.21}$$

By inserting (7.21) in (7.20) one gets the final result

$$I_{n+1,mpq} = \frac{(n-p-1)(n+m)}{(n+p+2)(n-m+1)} I_{n-1,mpq} + \frac{(p+q)(2n+1)}{(n+p+2)(n-m+1)} I_{nm,p-1,q} - \frac{(2n+1)(1-t^2)}{(n+p+2)(n-m+1)} P_{nm}P_{pq}\Bigg|_{t_S}^{t_N} . \tag{7.22}$$

However this equation must be normalized to not get large numbers unfitted for use on computers. From equation (B.7) and (7.14) we see that

$$\overline{I}_{nmpq}(t_S,t_N) = H_{nm}\ H_{pq}\ I_{nmpq}(t_S,t_N) \tag{7.23}$$

where H_{nm} is given by (B.8). Inserting (7.23) and (B.7) with (B.8) in (7.22) we get the final relation

$$\overline{I}_{nmpq}(t_S,t_N) = \frac{a(n,m)}{(n+p+1)}\left(\frac{(n-p-2)}{a(n-1,m)}\overline{I}_{n-2,mpq}(t_S,t_N) + \frac{(2p+1)}{a(p,q)}\overline{I}_{n-1,m,p-1,q}(t_S,t_N) - (1-t^2)\overline{P}_{n-1,m}(t)\overline{P}_{pq}(t)\Bigg|_{t_S}^{t_N}\right) \tag{7.24}$$

where $m \neq n$ and we have used $a(n,m)$ defined in (7.3).

From the definition of $a(n,m)$, (7.24) is undefined for $m = n$. This seems to restrict us to finding for example $\bar{I}_{2210}$. But because of the symmetry in (7.14), $\bar{I}_{nmpq} = \bar{I}_{pqnm}$ and $\bar{I}_{2210}$ can be computed with (7.24) by computing $\bar{I}_{1022}$. Following this finding, one can set $p = q$ in (7.24) and rename the indices to obtain the following result

$$\bar{I}_{nnpq}(t_S,t_N) = \frac{a(p,q)}{(p+n+1)}\left(\frac{(p-n-2)}{a(p-1,q)}\,\bar{I}_{nn,p-2,q}(t_S,t_N) + \right.$$

$$\left. + (1-t^2)\bar{P}_{p-1,q}(t)\bar{P}_{nn}(t)\Big|_{t_S}^{t_N}\right) \quad . \tag{7.25}$$

Thus one finds out that the only integrals which cannot be computed from (7.24) are the $\bar{I}_{nnpp}$ kind. One can see the similarity between the development in this section and the previous section.

Following the idea in appendix C where we derived (7.4), we will now find a recurrence relation that solves the following integral

$$I_{nnpp} = \int_{t_S}^{t_N} P_{nn}\, P_{pp}\, dt = \frac{(2n)!\,(2p)!}{2^n n!\; 2^p p!}\int_{t_S}^{t_N} (1-t^2)^{(n+p)/2}\, dt \tag{7.26}$$

where we have used equation (A.3). Lets integrate by parts the right side by setting

$$u=(1-t^2)^{z/2}\ , \quad du=-zt(1-t^2)^{z/2-1}\,dt, \quad dv=dt, \quad v=t, \tag{7.27}$$

where we have put $z=n+p$. One obtains

$$\int_{t_S}^{t_N} (1-t^2)^{z/2} dt = t(1-t^2)^{z/2}\Big|_{t_S}^{t_N} + n\int_{t_S}^{t_N} t^2(1-t^2)^{z/2-1} dt \quad . \tag{7.28}$$

When one has verified that the last term can be written as

$$t^2(1-t^2)^{z/2-1} = (1-t^2)^{z/2-1} - (1-t^2)^{z/2} \tag{7.29}$$

then (7.28) becomes

$$(z+1)\int_{t_S}^{t_N} (1-t^2)^{z/2}\, dt = t(1-t^2)^{z/2}\Big|_{t_S}^{t_N} + z\int_{t_S}^{t_N} (1-t^2)^{z/2-1}\, dt \quad . \tag{7.30}$$

Again $z = n+p$ thus one can insert (7.26) in (7.30) and use (A.3) to get

$$\frac{(z+1)2^n n!}{(2n)!} I_{nnpp} = \frac{t2^n n!}{(2n)!} P_{nn}P_{pp}\Big|_{t_S}^{t_N} + z2^{n-2}\frac{(n-2)!}{(2n-4)!} I_{n-2,n-2,pp} \quad . \tag{7.31}$$

This can be simplified to the final relation

$$I_{nnpp} = \frac{1}{n+p+1}\left[tP_{nn}P_{pp}\Big|_{t_S}^{t_N} + (n+p)(2n-1)(2n-3)\, I_{n-2,n-2,pp}\right] \quad . \tag{7.32}$$

In the computations we use the normalized relation of (7.32) which is obtained from (7.23) and (B.7) with (B.8). This results in

$$\overline{I}_{nnpp}(t_S,t_N) = \frac{1}{n+p+1} \left[t\,\overline{P}_{nn}(t)\,\overline{P}_{pp}(t) \Big|_{t_S}^{t_N} + \right.$$

$$\left. + (n+p)\, b(n)\, b(n-1)\, \overline{I}_{n-2,n-2,pp}(t_S,t_N) \right]$$

$$, \ n \neq 0, \ n \neq 1, \quad (7.33)$$

where again $b(n)$ is defined at (7.5). Since in (7.33), "n" cannot be equal to 0 (zero) in $b(n)$ and "n" cannot be equal to 1 (one) in $b(n-1)$, we have to find $\overline{I}_{0000}$, $\overline{I}_{1100}$ and $\overline{I}_{1111}$.

One will find out that these 3 numbers, $\overline{I}_{0000}$, $\overline{I}_{1100}$ and $\overline{I}_{1111}$, are the only required starting values for all three recurrence relations (7.33) and (7.24) and (7.25). These starting values are

$$\overline{I}_{0000}(t_S,t_N) = \int_{t_S}^{t_N} \overline{P}_{00}(t)\,\overline{P}_{00}(t)\,dt = \int_{t_S}^{t_N} dt = t_N - t_S,$$

$$\overline{I}_{1100}(t_S,t_N) = \int_{t_S}^{t_N} \overline{P}_{11}(t)\,\overline{P}_{00}(t)\,dt = 3^{1/2} \int_{t_S}^{t_N} (1-t^2)^{1/2}\,dt$$

i.e.

$$\overline{I}_{1100}(t_S,t_N) = \frac{3^{1/2}}{2} \left[t(1-t^2)^{1/2} - \arccos(t) \right] \Big|_{t_S}^{t_N},$$

and

$$\overline{I}_{1111}(t_S,t_N) = \int_{t_S}^{t_N} \overline{P}_{11}(t)\ \overline{P}_{11}(t)\,dt = (3t - t^3)\Big|_{t_S}^{t_N} . \tag{7.34}$$

As described in the previous section 7.1 the forward recurrence relation (7.4) is unstable and so will be (7.33). The great similarities between both relations is a sufficient proof and could be seen numerically.

As described in section 7.1 this problem of instability in (7.33) is solved by using a backward recurrence relation which is directly obtained from (7.33) itself as

$$\overline{I}_{nnpp}(t_S,t_N) = \frac{1}{(n+p+2)b(n+2)b(n+1)} [(n+p+3)\overline{I}_{n+2,n+2,pp} - - t\ \overline{P}_{n+2,n+2}(t)\ \overline{P}_{pp}(t)\Big|_{t_S}^{t_N}] . \tag{7.35}$$

To use (7.35) the starting values required are $\overline{I}_{NMAX,NMAX,NMAX,NMAX}$, $\overline{I}_{NMAX,NMAX,NMAX-1,NMAX-1}$ and $\overline{I}_{NMAX-1,NMAX-1,NMAX-1,NMAX-1}$. These starting values are obtained from the integration of a McLaurin series as it was done in appendix D to get (7.10). From (D.4) we have

$$I_{nnpp} = b(n)b(n-1)\ldots b(1)b(p)b(p-1)\ldots b(1) \int_{t_S}^{t_N} y^z\ dt . \tag{7.36}$$

where $z = n+p$ and y is defined at (D.2). The integration is performed in appendix D where one can compare (D.5) with (D.8) with (7.36) to find that

$$\overline{I}_{nnpp} = -b(n)b(n-1)\ldots b(1)b(p)b(p-1)\ldots b(1)y^{n+p+2}\left[\frac{1}{n+p+2} + \frac{1}{2}\frac{y^2}{(n+p+4)} + \frac{1}{2}\frac{3}{4}\frac{y^4}{(n+p+6)} + \ldots\right]\Bigg|_{t_S}^{t_N} . \tag{7.37}$$

where $y_S = \sin\theta_S$ and $y_N = \sin\theta_N$. This relation is used to find $\overline{I}_{NMAX,NMAX,NMAX,NMAX}$, $\overline{I}_{NMAX,NMAX,NMAX-1,NMAX-1}$ and $\overline{I}_{NMAX-1,NMAX-1,NMAX-1,NMAX-1}$. The procedure explained in section 7.1 was used to decide on the number of terms required in (7.37) and when to use the forward or the backward recurrence. It is appropriate because the $I_{nnpp}(t_S,t_N)$'s are basically the same functions as the $I_{nn}(t_S,t_N)$'s. By definition they are both related to the integrals of sine functions (see last paragraph of section 7.1). It was numerically verified, see below, that this procedure was appropriate.

Since the above relations are newly developed they must be checked in some way. While the relations of section 7.1 were checked against a Gaussian quadrature by Christodoulidius and Katsambalos (1977) this will not be required for the new relations. We have verified numerically the results of the new recurrence relation against the following analytical relations:

$$\overline{I}_{nm00}(t_S, t_N) = \overline{I}_{nm}(t_S, t_N)$$

$$\overline{I}_{nmnm}(\cos(\pi/2),\ \cos 0) = \int_{\cos(\pi/2)}^{\cos 0} \overline{P}_{nm}^{2}(t)\ dt = 2,\quad m \neq 0$$

$$\bar{I}_{n0n0}(\cos(\pi/2),\ \cos 0) = \int_{\cos(\pi/2)}^{\cos 0} \bar{P}^2_{n0}(t)\ dt = 1 \quad . \tag{7.38}$$

We have also verified numerically the following summation

$$\sum_{i=0}^{k-1} \bar{I}_{nmpq}(\cos(\theta_i+\Delta\theta),\ \cos\theta_i) = \bar{I}_{nmpq}(\cos(\pi/2),\ \cos 0) \tag{7.39}$$

which agreed to ten digits. "k" in (7.39) is the number of $\Delta\theta°$ in the northern hemisphere i.e. 90° divided by $\Delta\theta°$. During these tests we could also verify an important relation between the values in the northern and southern hemispheres. Similar to the relation between the associated Legendre functions computed in the northern and southern hemisphere (Colombo, 1981, p.15, last paragraph) where

$$\begin{aligned} \bar{P}_{nm}(-\theta) &= \bar{P}_{nm}(\theta) && \text{when n+m is even} \\ \bar{P}_{nm}(-\theta) &= -\bar{P}_{nm}(\theta) && \text{when n+m is odd} \end{aligned} \tag{7.40}$$

and between the integrals of one Legendre function where

$$\begin{aligned} \bar{I}^{NH}_{nm}(t_S,t_N) &= \bar{I}^{SH}_{nm}(t_S,t_N) && \text{when n+m is even} \\ \bar{I}^{NH}_{nm}(t_S,t_N) &= -\bar{I}^{SH}_{nm}(t_S,t_N) && \text{when n+m is odd} \end{aligned} \tag{7.41}$$

we have found that between the integrals of the product of two Legendre functions the following relations exist between the northern NH and southern SH hemispheres

$$\bar{I}^{NH}_{nmpq}(t_S,t_N) = \bar{I}^{SH}_{nmpq}(t_S,t_N) \quad \text{when } n+m+p+q \text{ is even}$$
$$\bar{I}^{NH}_{nmpq}(t_S,t_N) = -\bar{I}^{SH}_{nmpq}(t_S,t_N) \quad \text{when } n+m+p+q \text{ is odd} \quad . \tag{7.42}$$

These last relations (7.42) like (7.41) and (7.40) permit us to save computer time by requiring only the values in the northern hemisphere to be computed. The Fortran routines PNMI and PNMI2 in appendix H compute respectively the $\bar{I}nm(\theta)$ and $\bar{I}_{nmpq}(\theta)$ values required in (4.12) and (6.14). These routines PNMI and PNMI2 can also be incorporated as subroutines into routines that require them.

While the altimetry-gravimetry problem is the first application known to the author requiring $\bar{I}_{nmpq}(\theta)$ values, these could very possibly be required in the future for other applications such as the one of Sacerdote and Sanso (1985) regarding the "Overdetermined b.v.p. in Physical Geodesy" where the $\bar{I}_{nmpq}(\theta)$ are required to compute their equation (A2.A, p.207).

The recurrence relations developed in the first section of this chapter can be used for the integral computations required in equation (4.12) while the recurrence relations of the second section can be used for the computations of all the integrals required in (6.14). This second set of recurrence relations for the integration of two associated Legendre functions are developed for the first time. Their validity was obtained by comparing them with other analytical relations (7.38) and numerical summations (7.39). This chapter completes the relations needed to solve numerically the altimetry-gravimetry problem. The next chapter will collect all the final equations and will analyze this theory of the proposed solution to the altimetry-gravimetry problem. Then chapter 9 will describe the computations and tests done during this project.

CHAPTER VIII

COMPUTING THE ALTIMETRY-GRAVIMETRY SPHERICAL HARMONIC POTENTIAL COEFFICIENTS

8.1 Transforming Altimetry-Gravimetry Coefficients into Spherical Harmonic Ones.

We have seen at equation (3.37) that one can express the disturbing potential $\hat{T}$ at the earth's surface into the following two series expressions

$$\hat{T}(\theta,\lambda) = \frac{GM}{R} \sum_{n=0}^{\nu-4} E_n \, X_n(\theta,\lambda) = \frac{GM}{R} \sum_{n=0}^{\nu-4} T_n \, S_n(\theta,\lambda) \quad . \qquad (8.1)$$

In the previous chapters we have shown how the orthonormal base functions $X_n(\theta,\lambda)$ in (8.1) are used to compute the E_n coefficients, herein called the altimetry-gravimetry coefficients. These coefficients are the solution to the altimetry-gravimetry problem. In physical geodesy however, the second expansion in (8.1) is used where the $S_n(\theta,\lambda)$ are the spherical harmonics, see equations (3.2) to (3.5). Thus it is desired to retransform the altimetry-gravimetry solution E_n into the T_n harmonic coefficients.

From table 5.3 and using (3.27) one can write the matrix equation

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_\mu \end{bmatrix} = \begin{bmatrix} g_{00} & & & & 0 \\ g_{10} & g_{11} & & & \\ g_{20} & g_{21} & g_{22} & & \\ \vdots & & & \ddots & \\ g_{\mu 0} & \cdots & & & g_{\mu\mu} \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_\mu \end{bmatrix} \quad . \tag{8.2}$$

Equation (8.1) can also be written as a matrix equation

$$\hat{T} = [\, E_0 \; E_1 \; E_2 \; \ldots \; E_\mu \,] \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_\mu \end{bmatrix} = [\, T_0 \; T_1 \; T_2 \; \ldots \; T_\mu \,] \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_\mu \end{bmatrix} \quad . \tag{8.3}$$

where $\mu = \nu - 4$. Inserting (8.2) in (8.3) gives

$$\hat{T} = [\, E_0 \; E_1 \; E_2 \; \ldots \; E_\mu \,] \begin{bmatrix} g_{00} & & & & 0 \\ g_{10} & g_{11} & & & \\ g_{20} & g_{21} & g_{22} & & \\ \vdots & & & \ddots & \\ g_{\mu 0} & \cdots & & & g_{\mu\mu} \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_\mu \end{bmatrix} \quad . \tag{8.4}$$

Comparing (8.4) to (8.3) one gets

$$[\ T_0\ T_1\ T_2 \ldots T_\mu] = [\ E_0\ E_1\ E_2 \ldots E_\mu] \begin{bmatrix} g_{00} & & & 0 \\ g_{10} & g_{11} & & \\ g_{20} & g_{21} & g_{22} & \\ \vdots & & & \ddots \\ g_{\mu 0} & \cdots & & g_{\mu\mu} \end{bmatrix} \quad . \qquad (8.5)$$

We know that the above comparison is usually not a valid matrix operation. However since the $s_n(\theta,\lambda)$'s are linearly independent one can compare each line independently which makes (8.5) valid. More clearly, comparing (8.3) and (8.4) we have

$$T_0\, S_0 + T_1\, S_1 + \ldots = (E_0\, g_{00} + E_1\, g_{10} + \ldots)\, S_0 + (E_1\, g_{11} + E_2\, g_{21} + \ldots)\, S_1 + \ldots \quad . \qquad (8.6)$$

Since s_0 is linearly independent of s_1 and s_2, etc., we can write from (8.6)

$$T_0 = (E_0\, g_{00} + E_1\, g_{10} + \ldots + E_\mu\, g_{\mu 0}) \qquad (8.7)$$

which is the result expressed in (8.5). The matrix relation (8.5) can be simply written

$$T_n = \sum_{p=n}^{\nu-4} E_p\, g_{pn} \qquad (8.8)$$

where ν is given by (2.8) and the coefficients g_{pn} and E_n respectively by (5.11) and (4.3).

Relation (8.8) is the result sought. This transformation of the E_n coefficients into the spherical harmonic coefficients T_n will permit one to use the existing efficient software to compute the components of the Earth's gravity field, gravity anomalies, geoid undulations, deviations of the vertical, etc. Contrarily to the E_n, the T_n coefficients can be compared with existing earth's gravity field expansion and better analyzed coefficient by coefficient; this is another advantage of this transformation. But most importantly this transformation allows one to combine this solution with "satellite-derived potential coefficients". This combination is very important because it is well known that the low degree spherical harmonic potential coefficients are best determined from satellite solutions while the high degree coefficients are best obtained from one using terrestrial data. Both sets complement each other and their combination permits one to derive the optimum Earth's gravity field expansion. This sort of combination was performed by Rapp (1978). We will now focus our attention to the computation requirements and the numerical computations of our altimetry-gravimetry problem.

8.2 Gathering All Relations for Computations.

All the equations obtained so far that solve the altimetry-gravimetry problem will be gathered to clearly see the computations required.

From (4.12) one has

$$\begin{matrix} E_{nm} \\ F_{nm} \end{matrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \bar{I}^{\,i}_{nm}(\theta) \left(\begin{matrix} A(m) \\ -B(m) \end{matrix} RE[X^i_n(m)] + \begin{matrix} B(m) \\ A(m) \end{matrix} IM[X^i_n(m)] \right) \quad . \qquad (8.9)$$

One can write (8.9) as

$$\begin{matrix} E_{nm} \\ F_{nm} \end{matrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N-1} \bar{I}^{\,i}_{nm}(\theta) \begin{matrix} E^i_{nm} \\ F^i_{nm} \end{matrix} \qquad (8.10)$$

where E^i_{nm} and F^i_{nm} are the expressions within the brackets of (8.9) i.e.

$$\begin{matrix} E^k_{nm} \\ F^k_{nm} \end{matrix} = \begin{matrix} A(m) \\ -B(m) \end{matrix} RE[X^k_n(m)] + \begin{matrix} B(m) \\ A(m) \end{matrix} IM[X^k_n(m)] \quad . \qquad (8.11)$$

Because of (7.41) the computations involved in (8.10) can be reduced in half by writing it as follows:

$$\begin{matrix} E_{nm} \\ \\ F_{nm} \end{matrix} = \frac{1}{4\pi\eta_n} \sum_{i=0}^{N/2} \bar{I}^{\,i}_{nm}(\theta) \left(\begin{matrix} E^{i}_{nm}(NH) \\ \\ F^{i}_{nm}(NH) \end{matrix} + (1-2*MOD(n+m,2)) \begin{matrix} E^{i}_{nm}(SH) \\ \\ F^{i}_{nm}(SH) \end{matrix} \right) \quad . \qquad (8.12)$$

As in (7.41) HN and HS means North and South Hemispheres. And MOD(I,J) is the remainder of I divided by J.

The same reduction in the computations applies to (6.14), because of (7.42). We can write (6.14) as

$$\begin{matrix} A_{nmpq} \\ B_{nmpq} \\ C_{nmpq} \\ D_{nmpq} \end{matrix} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \bar{I}^{\,i}_{nmpq}(\theta) \begin{matrix} A^{i}_{nmpq} \\ B^{i}_{nmpq} \\ C^{i}_{nmpq} \\ D^{i}_{nmpq} \end{matrix} \qquad (8.13)$$

where

$$\begin{matrix} A^{i}_{nmpq} \\ B^{i}_{nmpq} \\ C^{i}_{nmpq} \\ D^{i}_{nmpq} \end{matrix} = \begin{matrix} A(m-q) \\ A(m-q) \\ B(m-q) \\ -B(m-q) \end{matrix} RE[X^{i}_{np}(m-q)] + \begin{matrix} B(m-q) \\ B(m-q) \\ -A(m-q) \\ A(m-q) \end{matrix} IM[X^{i}_{np}(m-q)] +$$

$$+ \begin{pmatrix} A(m+q) \\ -A(m+q) \\ -B(m+q) \\ -B(m+q) \end{pmatrix} RE[X^{i}_{np}(m+q)] + \begin{pmatrix} B(m+q) \\ -B(m+q) \\ A(m+q) \\ A(m+q) \end{pmatrix} IM[X^{i}_{np}(m+q)] \,. \quad (8.14)$$

Because of (7.42) the computations involved in (8.14) can be reduced by half by rewriting it as

$$\begin{matrix} A_{nmpq} \\ B_{nmpq} \\ C_{nmpq} \\ D_{nmpq} \end{matrix} = \frac{1}{4\pi} \sum_{i=0}^{N-1} \bar{I}^{i}_{nmpq}(\theta) \left(\begin{matrix} A^{i}_{nmpq}(NH) \\ B^{i}_{nmpq}(NH) \\ C^{i}_{nmpq}(NH) \\ D^{i}_{nmpq}(NH) \end{matrix} \right. +$$

$$\left. + (1 - 2\ MOD(n+m+p+q,2)) \begin{matrix} A^{i}_{nmpq}(SH) \\ B^{i}_{nmpq}(SH) \\ C^{i}_{nmpq}(SH) \\ D^{i}_{nmpq}(SH) \end{matrix} \right) . \quad (8.15)$$

These relations (8.12) and (8.15) and their notation can be recognized in the two FORTRAN routines that computes them, FFTENM for (8.12) and FFTABC for (8.15) in appendix G.

Going on in view of gathering all the equations required to compute a solution we have from (5.11)

$$g_{pp} = u_p \tag{8.16}$$

from which (4.3) becomes

$$E_p = g_{pp}\left(\sum_{n=0}^{p-1} c_{pn}\, E_n + E_p{}'\right), \quad p = 0,1,2,\ldots,\nu-4, \tag{8.17}$$

(5.11) becomes

$$g_{pk} = g_{pp} \sum_{n=k}^{p-1} c_{pn}\, g_{nk}\ , \quad k<p, \quad \begin{array}{l} p = 1,2,\ldots,\nu-4, \\ k = 0,1,2,\ldots,\nu-5, \end{array} \tag{8.18}$$

(5.44) remains unchanged

$$c_{pn} = - \sum_{q=0}^{n} g_{nq}\, c_{pq}{}', \quad n<p, \quad \begin{array}{l} p = 1,2,\ldots,\nu-4, \\ n = 0,1,2,\ldots,\nu-5, \end{array} \tag{8.19}$$

and (5.45) becomes

$$\left(\frac{1}{g_{pp}}\right)^2 = - \sum_{n=0}^{p-1} c_{pn}^{2} + u_p{}'\ , \quad p = 0,1,2,\ldots,\nu-4. \tag{8.20}$$

The $E_n{}'$ coefficients in (8.17) are given by (8.12). The $c_{nm}{}'$ and $u_n{}'$ coefficients in (8.19) and (8.20) are given by (8.15). The relation between the $E_n{}'$ and the E_{nm} and F_{nm} is as described in (3.5). The relations between the $c_{nq}{}'$, $u_n{}'$ and the

A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq} are shown in Table 5.4. Once all the above relations have been computed up to $\mu = \nu\text{-}4 = (N+1)^2 - 5$, (see (2.8) and the paragraph before (3.11)), allows one to compute the final harmonic coefficients with (8.8), i.e.

$$T_k = \sum_{p=k}^{\nu-4} E_p \, g_{pk} \,, \qquad k = 0, 1, 2, \ldots, \nu-4. \tag{8.21}$$

More precisely the computations of (8.17) to (8.20) start as shown in the following table.

Table 5 Storage Required by the Gram-Schmidt Orthonormalization Process.

We replace:	by:	where:
u_0'	g_{00}	$g_{00} = u_0 = 1/(u_0')^{1/2}$
E_0'	E_0	$E_0 = g_{00}\, E_0'$
c_{10}'	c_{10}	$c_{10} = -g_{00}\, c_{10}'$
u_1'	g_{11}	$g_{11} = u_1 = 1/(-c_{10}^2 + u_1')^{1/2}$
E_1'	E_1	$E_1 = g_{11}\, (c_{10}\, E_0 + E_1')$
c_{10}	g_{10}	$g_{10} = g_{11}\, c_{10}\, g_{00}$
c_{21}'	c_{21}	$c_{21} = -g_{10}\, c_{20}' - g_{11}\, c_{21}'$
c_{20}'	c_{20}	$c_{20} = -g_{00}\, c_{20}'$
u_2'	g_{22}	$g_{22} = u_2 = 1/(-c_{20}^2 - c_{21}^2 + u_2')^{1/2}$
E_2'	E_2	$E_2 = g_{22}\, (c_{20}\, E_0 + c_{21}\, E_1 + E_2')$
c_{20}	g_{20}	$g_{20} = g_{22}\, (c_{20}\, g_{00} + c_{21}\, g_{10})$

Table 5 Storage Required by the Gram-Schmidt Orthonormalization Process. (Continued).

We replace:	by:	where:
c_{21}	g_{21}	$g_{21} = g_{22}\ c_{21}\ g_{11}$
c_{32}'	c_{32}	$c_{32} = -g_{20}\ c_{30}'\ -g_{21}\ c_{31}'\ -g_{22}\ c_{32}'$
c_{31}'	c_{31}	$c_{31} = -g_{10}\ c_{30}'\ -g_{11}\ c_{31}'$
c_{30}'	c_{30}	$c_{30} = -g_{00}\ c_{30}'$
u_3'	g_{33}	$g_{33} = u_3 = 1/(-c_{30}^2\ -c_{31}^2\ -c_{32}^2\ +u_3')^{1/2}$
E_3'	E_3	$E_3 = g_{33}\ (c_{30}\ E_0+c_{31}\ E_1+c_{32}\ E_2\ +\ E_3')$
c_{30}	g_{30}	$g_{30} = g_{33}\ (c_{30}\ g_{00}\ +c_{31}\ g_{10}\ +c_{32}\ g_{20})$
c_{31}	g_{31}	$g_{31} = g_{33}\ (c_{31}\ g_{11}\ +c_{32}\ g_{21})$
c_{32}	g_{32}	$g_{32} = g_{33}\ c_{32}\ g_{22}$
ETC...		

From this table one finds out that the storage required is as follow. The c_{nm} replace the c_{nm}' and the g_{nm} replace the c_{nm}; also the E_n replace the E_n' and the T_n replace the E_n. So a lower triangular matrix g(n,n) and the vector E(n) are the storage required to compute the vector T_n as shown with the following matrices equivalence:

$$\begin{bmatrix} u_0' & & & \\ c_{10}' & u_1' & & \\ c_{20}' & c_{21}' & u_2' & \\ c_{30}' & c_{31}' & c_{32}' & u_3' \\ \vdots & \vdots & & \ddots \\ c_{\mu 0}' & c_{\mu 1}' & \cdots & u_\mu' \end{bmatrix} \Leftrightarrow \begin{bmatrix} g_{00} & & & \\ c_{10} & g_{11} & & \\ c_{20} & c_{21} & g_{22} & \\ c_{30} & c_{31} & c_{32} & g_{33} \\ \vdots & \vdots & & \ddots \\ c_{\mu 0} & c_{\mu 1} & \cdots & g_{\mu\mu} \end{bmatrix} \Leftrightarrow \begin{bmatrix} g_{00} & & & \\ g_{10} & g_{11} & & \\ g_{20} & g_{21} & g_{22} & \\ g_{30} & g_{31} & g_{32} & g_{33} \\ \vdots & \vdots & & \ddots \\ g_{\mu 0} & g_{\mu 1} & \cdots & g_{\mu\mu} \end{bmatrix}$$

and

$$\begin{bmatrix} E_0' \\ E_1' \\ E_2' \\ \vdots \\ E_\mu' \end{bmatrix} \Leftrightarrow \begin{bmatrix} E_0 \\ E_1 \\ E_2 \\ \vdots \\ E_\mu \end{bmatrix} \Leftrightarrow \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_\mu \end{bmatrix} .$$

The following table gives the size of the arrays required to store the above vector E(n) and lower triangular matrix g(n,n) when a maximum degree and order of surface harmonic coefficients is sought.

Table 6 Vector Sizes for the Altimetry-Gravimetry Solution.

N = Maximum degree and order of surface harmonic solution,		
μ = $\nu-3$ = $(N+1)^2-4$ = Vector size to store the above vector $E(n)$,		
$\mu(\mu+1)/2$ = $(\nu-3)(\nu-2)/2$ = Vector size to store the lower triangular matrix $g(n,n)$.		
	$E(\mu)$	$g(\mu,\mu)$ symmetric
N	$\mu = \nu-3$	$(\nu-3)(\nu-2)/2$
28	837	350,703
36	1365	932,295
180	32757	536,526,903

The coefficients E_n' of (8.12) are computed using FFTENM in appendix G. The c_{np}' and u_n' of (8.15) are computed with FFTABC in appendix G. The results of FFTENM and FFTABC are entered in ORTHO of appendix G which computes equations (8.17) to (8.21) as shown in Table 5. The result of ORTHO is T_n the desired spherical harmonic coefficients that solves the altimetry-gravimetry b.v.p.. Geopotential coefficients defined with no units are usually manipulated. Thus by having previously defined T, Δg and W without units (see around equation (3.13)) i.e. T(no units)=T(with units)/(GM/R) and Δg(no units)=Δg(with units)/ (GM/R^2), $W(\sigma_1)$(no units)=$W(\sigma_1)$(with units) $(GM/R)^2$ and $W(\sigma_2)$(no units)=$W(\sigma_2)$(with units) $(GM/R^2)^2$, all the coefficients c_{np}', u_n' (i.e. A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq}), E_n' (i.e. E_{nm} and F_{nm}), c_{np}, u_n, E_n and T_n (i.e. $\bar{C}_{nm}$ and $\bar{S}_{nm}$) have no units.

Numerical results are presented in the next chapter. The next section shows how to make the computations cheaper.

8.3 The Cholesky Factorization.

One might have recognized that equations (8.17) to (8.21) are the relations that involve the inversion of a matrix. This is proven in this section. The Cholesky factorization enables one to solve a system of equation without having to compute a matrix inverse. This is much cheaper than computing the matrix inverse. Freeden (1983) shows the relation between the Cholesky factorization and the Gram-Schmidt orthonormalization. As suggested by Freeden (1983) we have used the efficient routines provided by the mathematical package "LINPACK" (Dongarra et al, 1979). This section shows how the Cholesky factorization is applied to compute T_n when E_n', c_{np}' and u_n' are provided, i.e. to compute equations (8.17) to (8.21).

The coefficients T_n are the solution of the system of linear equations (3.17) where the matrix is the Gram matrix G which contains the c_{np}' and u_n' given by (8.15) or (6.1). The right hand side vector of (3.17) contains the E_n' values given by (8.9). Since G is symmetric and positive definite G can be decomposed uniquely in the form

$$G = C\,C^{T} \quad . \tag{8.22}$$

In (8.22) C is a lower triangular matrix with positive diagonal element. The splitting of G is known as the Cholesky factorization (Dongarra et al., 1979, p.10-1). Equation (3.17) can be written as

$$G\ T = E' \quad . \tag{8.23}$$

Inserting (8.22) in (8.23) gives

$$C\ C^T\ T = E' \tag{8.24}$$

or simply

$$C\ E = E' \tag{8.25}$$

where we have defined

$$C^T\ T = E \quad . \tag{8.26}$$

Following these last equations we have used the "LINPACK" subroutines; DPOFA which find C from G because of (8.22), DPOSL to solve for E from (8.25) and DPOSL again to solve T from (8.26) which is the desired solution. The "LINPACK" subroutine DCHDC also finds C from G but by pivoting. The use and cost of pivoting was found unnecessary due to the high stability of the Gram matrix G.

We can write (8.5) as

$$T = D^T\ E \tag{8.27}$$

where we have defined D as the lower triangular matrix of element g_{np}. Replacing (8.26) into (8.27) one gets

$$T = D^T C^T T \qquad \text{or} \qquad T^T = T^T C D \tag{8.28}$$

which is true only if

$$D = C^{-1} \quad . \tag{8.29}$$

This proves that the g_{np} in (8.5) are the elements of the inverse of C, thus computing (8.17) to (8.21) is computing the inverse of C. In other words the Gram-Schmidt orthonormalization does not compute the inverse of the Gram matrix G but of C, the triangular factorization of G. Since from a numerical point of view the inversion is often not very economical it is preferable to avoid any inversion, of G or C, and use the Cholesky factorization.

For comparison with equations (8.17) to (8.21) we here give the relations to compute the solution by the Cholesky factorization.

The triangular decomposition or Cholesky factorization (8.22) is computed with

$$C_{00} = (u_0')^{1/2}$$

$$C_{0p} = c_{0p}' \; / \; C_{00}$$

and

$$C_{pp} = \left(u_p' - \sum_{n=0}^{p-1} C_{np}^2 \right)^{1/2}$$

$$C_{pn} = \left(C_{pn}' - \sum_{q=0}^{p-1} C_{qp} C_{qn} \right) / C_{pp} \quad . \qquad (8.30)$$

The forward solution (8.25) is computed with

$$E_0 = E_0' / C_{00}$$

$$E_p = \left(E_p' - \sum_{n=0}^{p-1} C_{np} E_n' \right) / C_{pp} \quad . \qquad (8.31)$$

And the backward solution (8.26) is computed with

$$T_{v-4} = E_{v-4} / C_{v-4}$$

$$T_p = \left(E_p - \sum_{k=p+1}^{n} C_{pk} T_n \right) / C_{pp} \quad . \qquad (8.32)$$

The solution using the Cholesky factorization is computed by the routine ORTHOC in appendix G.

It was verified that both the Cholesky factorization using the ORTHOC and the Gram-Schmidt orthonormalization using the routine ORTHO give the same results. While it is much more efficient to use the Cholesky than the Gram-Schmidt solution the Gram-Schmidt equations can be of much more help when analyzing the solution than the Cholesky equations. With the Cholesky solution one does not see the base function $X_n(\theta,\lambda)$ and its associated set of coefficients E_n. With the Gram-

Schmidt solution one finds out that each coefficient $E_0, E_1, E_2, \ldots$ is "independent" since they are defined as a base vector (associated to a base function). As can be seen from Table 5 the coefficient E_2 is computed from the previous coefficients E_1 and E_0 and the coefficient E_{30} would be computed from the previous coefficients E_{29} to E_0. This shows that each coefficient $E_0, E_1, E_2, \ldots$ is "independent" of the degree and order of the solution sought in the same way the $\bar{C}_{nm}$ and $\bar{S}_{nm}$ coefficients are in the single b.v.p. solution; this is desirable.

Also the Gram-Schmidt solution shows that the last computed coefficients T_n are "dependent" of the degree and order of the solution, this is undesirable but unavoidable. As can be seen from equation (8.5) the coefficient T_0 is computed from the coefficients $E_0, E_1, E_2, \ldots$,up to $E_{\nu-4}$ where $\nu-4$ again is the rank of the solution. If ν is large the effect of the other coefficients $E_{\nu-3}$ up to the ones at infinity E_∞, will be small on T_0 and T_1, etc.. But where ever the solution is truncated, say to degree and order 180 where $\nu-4 = 32756$ then from (8.5) or (8.22)

$$T_{32756} = g_{32756,32756}\, E_{32756} \quad . \tag{8.33}$$

This coefficient would better be defined if computed with the other coefficients up to infinity such as

$$\begin{aligned} T_{32756} = {} & g_{32756,32756}\, E_{32756} + g_{32757,32756}\, E_{32757} + \\ & + g_{32758,32756}\, E_{32758} + \cdots \quad . \end{aligned} \tag{8.34}$$

So we should compute terms higher than E_n to compute good T_n coefficients. And this shows why we must expect the last coefficients to be less well defined and why it would be acceptable to reject the last coefficients of this least-squares solution. Again we emphasize that this could be found out only when one studies the Gram-Schmidt orthonormalization equations and not when one tries to analyze the equations of the Cholesky factorization.

While it is much faster to compute the Cholesky factorization than the Gram-Schmidt orthonormalization, the same amount of storage is required. Table 6 gives the size of the two main arrays required during the computations. The last column gives the number of different elements in the symmetric Gram matrix. Since we have not tried to use magnetic disk or tape storage to solve the problem due to high cost, we can see from Table 6 that at least 350K words (double precision values) is required for a solution up to degree and order 28,28. We can also see from Table 6 that the computer storage and the computation time required increases drastically with the number of coefficients we want to solve for. To overcome this main drawback of the solution we have tried to solve the problem by using only the diagonal of the Gram matrix. The diagonal elements are generally ten times larger than the other elements. These numerical results and others, with their analysis, will be given in next chapter.

Finally this chapter has shown that the least-squares solution to a mixed b.v.p. like the altimetry-gravimetry problem involves the computations of coefficients such as (8.12) and (8.15) and the solution of a system of linear equations by the Cholesky factorization (equations (8.30), (8.31) and (8.32)) which can be computed using the efficient Fortran routines of the "LINPACK" package or of the "IMSL" library.

CHAPTER IX

NUMERICAL RESULTS AND ANALYSIS

All calculations were carried out on the Ohio State University's AMDHAL 470 V/8 computer using the IBM's Multiple Virtual Storage (MVS) operating system and the VS FORTRAN Level 4.0 (Oct 1984) compiler.

The solution to the altimetry-gravimetry problem as proposed here from chapters 2 to 8 was tested using geopotential coefficients, those of GEML2 (Lerch et al., 1982). They are complete to degree 20 with additional terms to degree 30, order 28. Large matrices can be manipulated by direct access files and magnetic tapes. By not doing so, for financial reasons, we restricted ourselves to the size of matrices involved in the solution that would fit the memory available in the computer. This restricted us to test our solution on recovering the GEML2 coefficients only up to degree 28 and order 28.

Up to degree 28, the GEML2 fully normalized potential coefficients ($\bar{C}_{nm}$ and $\bar{S}_{nm}$) were used to compute mean gravity anomalies $\overline{\Delta g}_{ij}$ and mean disturbing potential values $\bar{T}_{ij}$ for equiangular blocks σ_{ij} of size equal to 1 degree of latitude by 1 degree of longitude. Such values were computed on a regular grid covering the Earth (spherical unit sphere) using the efficient FFT harmonic synthesis of (Colombo, 1981). A brief summary of the equations involved to compute such a grid of mean values is given in appendix F, while the FORTRAN routine used is FFTDGN in appendix G. All the computations in this volume were done on a sphere.

From the values of the two regular grids ($\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$) we have produced one regular grid of mixed values of $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$ to simulate the mixed boundary value problem. The situation is shown on Figure 1 where $\overline{\Delta g}_{ij}$ are given on continents and $\overline{T}_{ij}$ on oceans. That mean disturbing potential values $\overline{T}_{ij}$ be provided or mean geoid undulations values $\overline{N}_{ij}$ be provided is of no concern here since according to Bruns' formula, $N = T / \gamma$ (HM, 1967, eq.(2-144)) and $\overline{N}_{ij}$ can always be transformed into $\overline{T}_{ij}$ using normal gravity γ.

Of more concern is the fact that the set of $\overline{T}_{ij}$ obtained from satellite altimetry might not be consistent with the set of $\overline{\Delta g}_{ij}$ obtained from terrestrial gravimeter. In other words, if one computes the mean of the altimetry data $\overline{T}_{ij}$ given over the oceans σ_1, this mean value is directly related to the zero degree harmonic $T_0(\sigma_1)$ which zero degree term defines an ellipsoid different from the mean earth ellipsoid (HM, Section 2-19). Similarly the mean value of the gravimetry data $\overline{\Delta g}_{ij}$ alone is related to a zero degree harmonic $\Delta g_0(\sigma_2)$ which will most probably define another ellipsoid. To be consistent, the solution must shift or scale at least one of the data sets, the $\overline{T}_{ij}$'s or the $\overline{\Delta g}_{ij}$'s, in such a way that the solution defines only one ellipsoid. To overcome this inconsistency between $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$ some authors like Sacerdote and Sanso (1985) suggest a solution with overlapping areas between the two sets of $\overline{T}_{ij}$ and $\overline{\Delta g}_{ij}$. Svensson (1983, p.350) states

> *"in the spherical case it is shown that the problem has one and only one solution ..., and provided that the zero degree component is removed.".*

In response to this statement, Arnold (1984) suggests that the mean square value of $T(\sigma_1)$ (i.e. μ_T) and of $\Delta g(\sigma_2)$ (i.e. μ_g) should be the weight used in his least-squares solution (Arnold, 1981) which we are developing in this dissertation. According to Arnold this scaling by the mean square values would remove this inconsistency between the two sets $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$. Arnold (1984, p.350) states

> *"In the least-square solution of the mixed b.v.p. the relative residuals* T/μ_T *and* $\Delta g/\mu_g$ *come to be adjusted and not the heterogeneous residuals* T *and* Δg. μ_T *and* μ_g *are the associated mean square residuals. The mean square values of* T/μ_T *and* $\Delta g/\mu_g$ *are both equal to unity."*

As shown later we have used the scaling by the mean square values (see equation 4.5) and it proves to be exact in the sense that this scaling was required to solve the altimetry-gravimetry problem. However the precise reason for this weight procedure as suggested by Arnold (1984) is not clear in his paper.

For our numerical solution to be feasible we had to use an efficient way of performing the calculation. Without the FFT applications of Colombo (1981) it would not have been financially possible. The FFT application restricts one to use a regular grid where overlapping is not possible.

To find an FFT harmonic analysis solution with overlapping data (two values, one $\overline{\Delta g}_{ij}$ and one $\overline{T}_{ij}$, for the same block σ_{ij}), if at all possible, is a suggestion for future research.

Apart from this altimetry-gravimetry problem with different data on continents and oceans (Figure 1) we have also tried our solution on a mixed b.v.p. which has a random distribution of $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$ (Figure 2). Results obtained were similar to the continent/ocean case and are thus not shown.

To later analyze the solution of the mixed b.v.p. we first solved the single b.v.p. with the same apriori model. That is, the previously derived 1° X 1° mean values $\overline{\Delta g}_{ij}$ computed using the GEML2 coefficients, up to degree 28, were input in the harmonic analysis FFTCNM routine of appendix G to compute back potential coefficients. These new potential coefficients were input in the harmonic synthesis FFTDGN routine to compute another set of $\overline{\Delta g}_{ij}$. The agreement of the two sets of $\overline{\Delta g}_{ij}$, the maximum difference, the RMS difference and the mean of the difference between the two sets are given on the first line of Table 7.

As we have seen in section 2.2 this solution of the single b.v.p. is a least-squares solution. We are not performing a least-squares adjustment but as a solution of the least-squares method the residuals should be minimized. When computing (E.1) with (E.6) we make approximations. These approximations are due to the use of mean values and of an approximated de-smoothing operator η_n, sometimes referred to as a noise amplificator. Because of these approximations the residuals (3.11) and (3.12) are not minimized. When trying to recover a geopotential model known apriori, as done here, the residuals should be zero. Similar to the least-squares adjustment where one must iterate because the model has been linearized we can iterate the solution to minimize the residuals. Thus the set of differences $\delta\overline{\Delta g}_{ij}$ between the two sets of $\overline{\Delta g}_{ij}$ are the residuals and these were entered in the analysis FFTCNM routine to compute corrections to potential coefficients. These corrections are added to the last set of potential coefficients

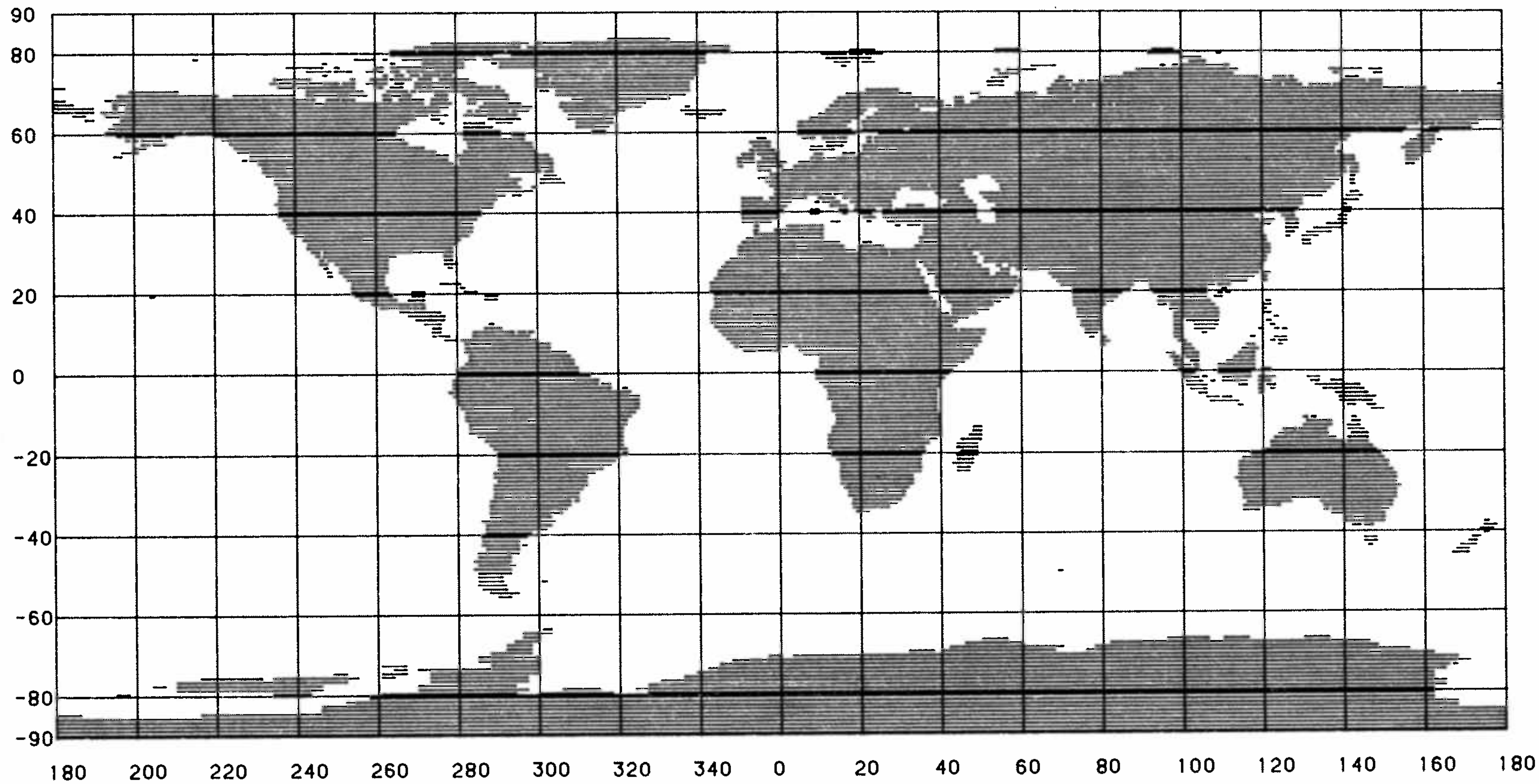

Figure 1 The Earth Covered with $1°\times1°$ Mean Gravity Anomaly Values on Continents 30% (-) and with $1°\times1°$ Mean Disturbing Potential Values on Oceans 70% .

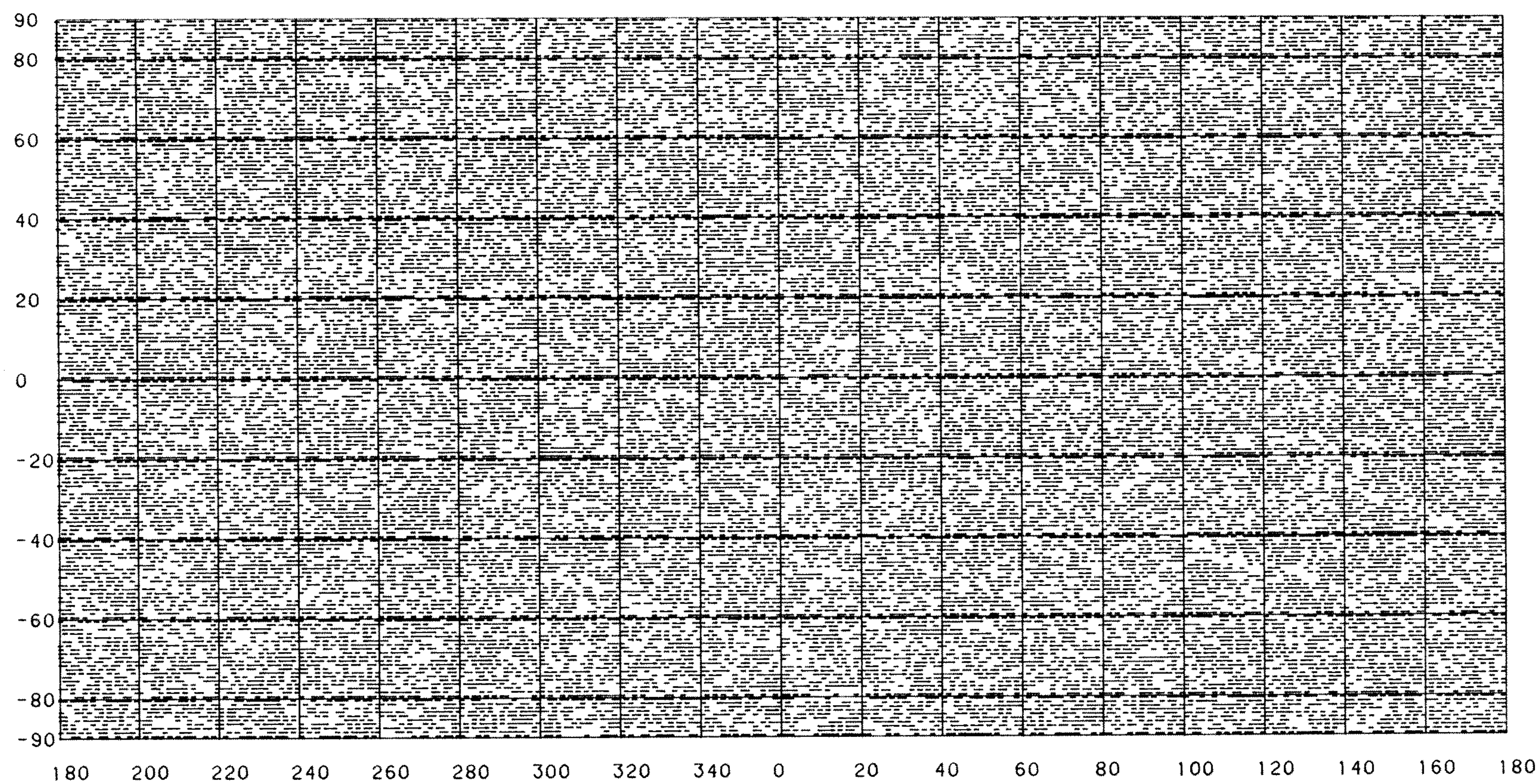

Figure 2 The Earth Covered Randomly with 50% of 1°X1° Mean Gravity Anomaly Values (-) and of 50% of 1°X1° Mean Disturbing Potential Values.

using ADDCNM routine, also in appendix G. These new potential coefficients are entered into the synthesis FFTDGN routine to compute another set of $\overline{\Delta g}_{ij}$. The maximum difference, the RMS difference and the mean of the difference between this set and the first set of $\overline{\Delta g}_{ij}$ derived from GEML2 are given in Table 7 as the 1 iteration case. The ADDDGN routine, also in appendix G, computes these statistics and creates the next set of gravity anomaly differences, $\delta\overline{\Delta g}_{ij}$ for the next iteration. As seen in Table 7 the RMS differences converge, and it is possible to recover all the GEML2 coefficients exactly to 7 digits after 5 iterations. The iteration process is shown on a flow chart in Figure 3.

One important remark should be given at this point. A least-squares solution where the residuals are minimized after some iterations shall be referred to as an "iterated" solution. When the mathematical model is not linear it takes some iterations to minimize the residuals. Here the mathematical model is linear but because of approximations during the computations it also requires some iterations to minimize the residuals. However, we will see that in practice, with actual observed data, an iterated solution may not be desired. Iterating causes all the frequency information up to infinity to enter into the finite number of coefficients, thus distorting the coefficients. On the other hand we will also see that one or two iterations might not yet distort the coefficients. There may be a problem here in deciding when to stop iterating. This problem might require further studies. In any case if no iteration is performed then the computation of equation (E.1) with (E.6) is called a "deterministic" solution. For the same reason as above, a deterministic solution instead of an iterated solution might be desired for the mixed altimetry-gravimetry b.v.p.. And we will show that this option is offered by the solution presented in this dissertation.

Table 7 Statistics on Single b.v.p. solution using GEML2

	$\delta\Delta g$ (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	.06	.07	-.06	0.0	.008	.06	-.06
1	0.0	.00	.00	.00	0.0	.000	.00	.00

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	14.06	44.0	-51.6	0.0	30.31	76.6	-104.2

		$\delta\Delta g$ (%)		δN (%)	
Degree n	# of coeff.	# of iter.: 0	1	0	1
2	5	.01	.00	.00	.00
3	7	.01	.00	.01	.00
4	9	.01	.00	.01	.00
5	11	.03	.00	.03	.00
6	13	.03	.00	.03	.00
7	15	.04	.00	.05	.00
8	17	.05	.00	.06	.00
9	19	.08	.00	.09	.00
10	21	.06	.00	.07	.00
11	23	.10	.00	.14	.00
12	25	.11	.00	.13	.00
13	27	.11	.00	.15	.00
14	29	.10	.00	.13	.00
15	31	.17	.00	.23	.00
16	33	.14	.00	.15	.00
17	35	.22	.00	.30	.00
18	37	.15	.00	.16	.00
19	39	.17	.00	.25	.00
20	41	.21	.00	.23	.00
21	31	.28	.00	.39	.00
22	31	.24	.00	.25	.00
23	11	.31	.00	.41	.00
24	11	.43	.00	.48	.00
25	11	.27	.00	.34	.00
26	5	.50	.00	.46	.00
27	9	.31	.00	.31	.00
28	11	.25	.00	.24	.00
29	6	n/a	n/a	n/a	n/a
30	2	n/a	n/a	n/a	n/a

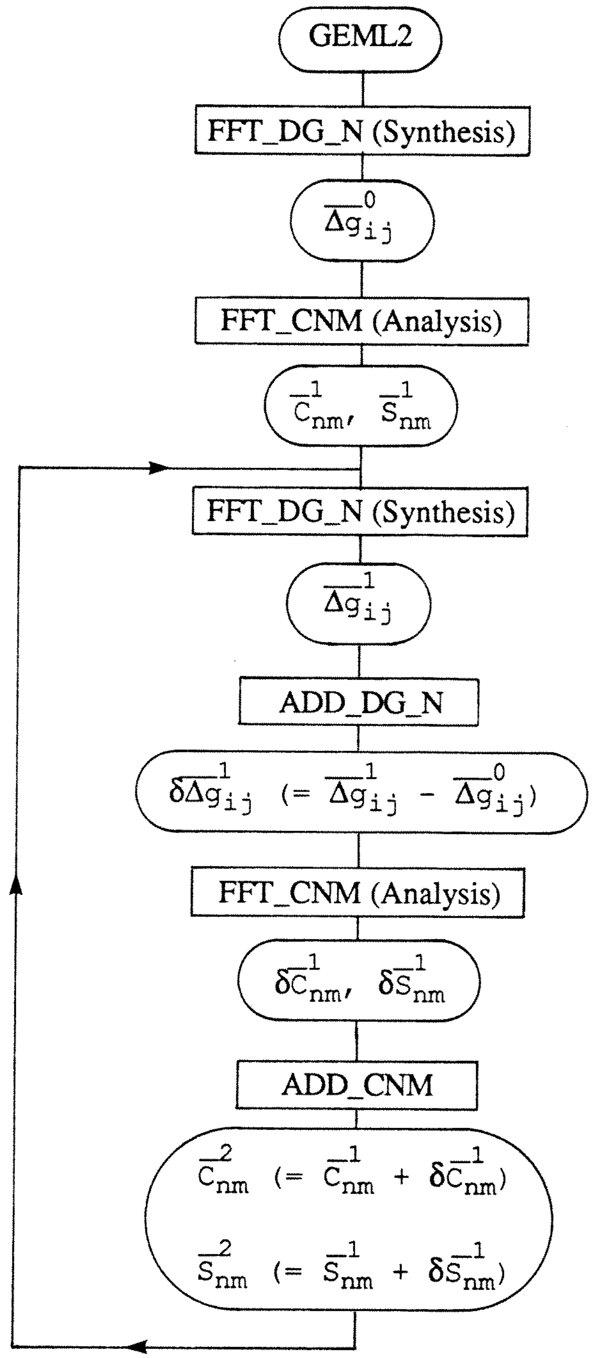

Figure 3 Flow Chart to Test the Single b.v.p. Solution

(Boxes are FORTRAN routines and curves are data sets).

The same process of iteration as above was applied using a set of mean geoid undulations $\bar{N}_{ij}$. The statistics are also given in Table 7. The deterministic solution is given as the 0 iteration case.

The same process of iteration can be applied to a larger set of geopotential coefficients. Those of (Rapp, 1981) known up to degree and order 180 were used and recovered to produce similar statistics (see Table 8). The iterations converge and Table 8 indicates that it would be possible to recover all the coefficients exactly. As in Table 7, in the lower part of Table 8 we give in percentage the disagreement between the recovered coefficients versus the original coefficients and this by degree and for each iteration. It is interesting to note how the coefficients converge, the lowest first and the higher last, as it will be the case for the mixed b.v.p. solution. Again the deterministic solution is given as the 0 iteration case.

The same process of iteration can be applied without using the de-smoothing operator η_n. In this case the convergence is slower but still converges as shown in Table 9. This is important since integrals for which the η_n function would not be known could still be computed accurately. Knowing the η_n function or its approximation is however useful for faster convergence. Our results in Table 8 and 9 show the correctness and effectiveness of the de-smoothing operator η_n. Without using η_n it took 4 iterations to recover the 64800 $\bar{N}_{ij}$ values with an RMS difference of 11 cm while using η_n it took 2 iterations to recover the 64800 $\bar{N}_{ij}$ values with an RMS difference of 10 cm. The deterministic solution is the case with no iteration. Comparing the zero iteration case of the last two tables one

Table 8 Statistics on Single b.v.p. Solution using RAPP81 and the De-smoothing Operator η_n.

	$\delta\Delta g$ (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	2.09	38.5	-26.9	0.0	.133		
1	0.0	.49	8.2	-6.1	0.0	.029		
2	0.0	.15	1.9	-1.6	0.0	.010	n/a	
3	0.0	.06	.7	-.7	0.0	.006		
4	0.0	.03	.6	-.6	0.0	.005		

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	22.5	255	-229	0.0	30.37	81.7	-106.8

		$\delta\Delta g$ (%)				
Degree n	# of coeff.	# of iter.: 0	1	2	3	4
2	5	.01	.00	.00	.00	.00
3	7	.01	.00	.00	.00	.00
4	9	.01	.00	.00	.00	.00
5	11	.03	.00	.00	.00	.00
6	13	.02	.00	.00	.00	.00
8	15	.05	.00	.00	.00	.00
10	21	.06	.00	.00	.00	.00
20	41	.28	.00	.00	.00	.00
30	61	.47	.01	.00	.00	.00
40	81	.88	.02	.00	.00	.00
50	101	1.21	.04	.00	.00	.00
60	121	1.86	.08	.01	.00	.00
70	141	4.89	.28	.02	.00	.00
80	161	6.89	.53	.04	.00	.00
90	181	8.72	.82	.08	.01	.00
100	201	10.71	1.21	.15	.02	.00
110	221	12.31	1.68	.24	.03	.01
120	241	14.13	2.29	.38	.07	.01
130	261	16.02	3.10	.60	.12	.03
140	281	18.41	4.04	.90	.21	.05
150	301	22.22	5.46	1.43	.37	.11
160	321	24.58	7.02	2.03	.59	.18
170	341	28.57	8.64	2.85	.94	.36
177	355	29.28	9.86	3.43	1.24	.53
178	357	29.76	11.00	5.47	3.98	3.54
179	359	28.51	10.01	4.24	2.31	1.49
180	361	30.56	15.75	12.11	10.77	9.97

Table 9 Statistics on Single b.v.p. Solution using RAPP81 and no De-smoothing operator η_n

	δΔg (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	n/a	3.83	n/a	n/a	n/a	.255	n/a	n/a
1		1.43				.088		
2		.56				.040		
3		.19				.020		
4		.06				.011		

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	22.5	255	-229	0.0	30.37	81.7	-106.8

		δΔg (%)				
Degree n	# of coeff.	# of iter.: 0	1	2	3	4
2	5	.01	.00	.00	.00	.00
3	7	.02	.00	.00	.00	.00
4	9	.04	.00	.00	.00	.00
5	11	.06	.00	.00	.00	.00
6	13	.09	.00	.00	.00	.00
8	15	.16	.00	.00	.00	.00
10	21	.24	.00	.00	.00	.00
20	41	1.00	.01	.00	.00	.00
30	61	2.06	.04	.00	.00	.00
40	81	3.48	.13	.01	.00	.00
50	101	5.59	.33	.02	.00	.00
60	121	7.81	.64	.05	.00	.00
70	141	10.18	1.10	.12	.01	.00
80	161	13.58	1.95	.29	.04	.01
90	181	17.12	3.04	.55	.10	.02
100	201	20.66	4.43	.97	.21	.05
110	221	24.03	6.05	1.55	.40	.11
120	241	27.61	8.02	2.38	.72	.22
130	261	31.21	10.36	3.53	1.22	.43
140	281	35.14	13.16	5.05	1.96	.77
150	301	39.94	16.90	7.30	3.20	1.41
160	321	43.88	20.51	9.80	4.74	2.30
170	341	48.60	24.89	12.99	6.85	3.80
177	355	51.14	27.59	15.22	8.50	4.78
178	357	51.31	28.22	16.21	9.84	6.51
179	359	51.76	27.98	15.61	9.01	5.43
180	361	50.91	29.42	19.29	14.55	12.31

verifies how useful the use of the de-smoothing operator is for the deterministic solution since it enables us to recover the coefficients two times more accurately, with an RMS difference of .133 metres (in Table 8) instead of .255 metres (in Table 9).

The same process of iteration can now be applied to our solution of the mixed b.v.p.. The same tables will be produced and compared with the preceding ones.

The two sets of $\overline{\Delta g}_{ij}$ and $\overline{T}_i$ produced earlier from GEML2 are entered into the FFTENM routine, with the continent/ocean distribution. This routine computes the E_n', i.e. E_{nm} and F_{nm} coefficients using equation (8.12) (which is also (4.4)). The FFTABC routine used the same distribution to compute the c_{np}' and u_n' i.e. the A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq} coefficients using (8.15) (which is also (6.1)). These two pieces of software (FFTENM and FFTABC) use FFT but not as efficiently as the FFTCNM and FFTDGN routines do. This is due to the $(n-1)$ and $(n-1)(p-1)$ factors in equations (4.4) and (6.1) that must be applied to the $\overline{\Delta g}_{ij}$ values while it is not required for the $\overline{T}_{ij}$ values. This causes the row of values along one latitude which is entered to FFTCC IMSL subroutine to be dependent on "n" or "n and p". To visualize the problem we can represent one row of 45° X 45° mean values as

$$\overline{T}_{11} \quad \overline{T}_{12} \quad \overline{T}_{13} \quad \overline{T}_{14} \quad \overline{T}_{15} \quad \overline{T}_{16} \quad \overline{T}_{17} \quad \overline{T}_{18}$$

This latitudinal row of values (without units) is entered into FFTCC and the frequencies $m = 0, 1, 2, 3, 4$ are returned. In the usual harmonic synthesis FFTCNM case, the task is then completed since we can compute all the coefficients from these frequency informations, $m = 0, 1, 2, 3, 4$. However in FFTENM and

FFTABC we have to enter the row again and again in FFTCC for each factor (n-1) or (n-1)(p-1) like this

1st time:	$\bar{T}_{11}$ $\bar{T}_{12}$ $\bar{T}_{13}$ $\bar{T}_{14}$ $\overline{\Delta g}_{15}$ $\overline{\Delta g}_{16}$ $\bar{T}_{17}$ $\bar{T}_{18}$
2nd time:	$\bar{T}_{11}$ $\bar{T}_{12}$ $\bar{T}_{13}$ $\bar{T}_{14}$ $2\overline{\Delta g}_{15}$ $2\overline{\Delta g}_{16}$ $\bar{T}_{17}$ $\bar{T}_{18}$
and so on:	$\bar{T}_{11}$ $\bar{T}_{12}$ $\bar{T}_{13}$ $\bar{T}_{14}$ $3\overline{\Delta g}_{15}$ $3\overline{\Delta g}_{16}$ $\bar{T}_{17}$ $\bar{T}_{18}$

To find a solution to this problem is another suggestion for future research that would improve the efficiency of our solution. Because of this problem FFTENM is at least 25 times slower than FFTCNM. The computer control processing unit (cpu) times are given in Table 10. FFTABC suffers from the same problem as FFTENM but in addition it has to compute many many more coefficients. As shown in Table 10 FFTCNM or FFTENM computes 2*NENM coefficients where NENM = (NMAX)(NMAX+1)/2 while FFTABC computes 4*NANMPQ coefficients where NANMPQ = (NENM)(NENM+1)/2, thus is much much more time consuming than FFTENM.

The two sets of coefficients E_{nm}, F_{nm} and A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq} representing E_n' and c_{np}' and u_n' are then entered into the ORTHO routine to perform the Gram-Schmid orthonormalization, equations (8-17) to (8- 21). This provides us with the final deterministic solution.

The same solution is obtained if instead of ORTHO we use the ORTHOC routine which performs the Cholesky Factorization, equations (8.30) to (8.32) (which is also (8.22), (8.25) and (8.26)).

Table 10 Computer Time and Storage.

Fortran routines:	PNMI	PNMI2	FFTCNM	FFTENM	FFTABC	ORTHOC	ORTHO
Output of routines	$\bar{I}^{i}_{nm}(\theta)$	$\bar{I}^{i}_{nmpq}(\theta)$	$\bar{C}_{nm}$ $\bar{S}_{nm}$	E_{nm} F_{nm}	A_{nmpq} B_{nmpq} C_{nmpq} D_{nmpq}	$\bar{C}_{nm}$ $\bar{S}_{nm}$	$\bar{C}_{nm}$ $\bar{S}_{nm}$
Up to degree and order ⇓	CPU TIMES:						
10 28 180	1 sec 2 sec 20 sec	6 sec 98 sec ≈1 hr	2 sec 5 sec 43 sec	40 sec 2 min 12 min	4 min 32 min ≈20 hres	2 sec 4 min ≈10 days	4 sec 10 min ≈30 days
	STORAGE (IN WORDS):						
28 180	90*435 90*16,471	90*94,830 90*135,655,156	2*435 2*16,471	2*435 2*16,471	4*94,830 4*135,655,156	841*841 32761*32761	837*837 32757*32757
	These values need not be stored since they can be computed with a subroutine.		These coefficients can always be computed by parts since each degrees are independent.			These are problematic since they require solving a very large system of equations.	

It was verified that both routines, ORTHO and ORTHOC, give the same solution but the second routine is, as expected, much faster. The cpu times are given in Table 10. The output of this ORTHOC routine is our final solution to the altimetry-gravimetry b.v.p. given as spherical harmonic potential coefficients.

From these coefficients a $1°\times 1°$ regular grid of 64800 mean $\overline{\Delta g}_{ij}$ values were computed. Another 64800 mean $\overline{T}_{ij}$ values covering the Earth were also computed. These two sets can be compared to the original two sets $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$ derived earlier from GEML2. The statistics are given in Table 11. The zero (0) iteration case is the deterministic solution just obtained. This table shows the result where the weight function was set to one (unity). We will later be able to appreciate the improvement brought by using the mean square values of $\overline{T}(\sigma_1)$ and $\overline{\Delta g}(\sigma_2)$.

As previously noted we can iterate the solution and look at the convergence. Figure 4 shows the flow chart of the computations involved and of the iteration process. The one (1) and two (2) iteration cases in Table 11 show that it is converging, which shows the correctness of the theory and of the numerical computations (routines) in this dissertation. However one can see in the lower part of the table how strange the first degrees 2, 3, 4 and 5 converge with the agreement decreasing after 1 iteration and then increasing. Also the zero (0) iteration case does not show an as good agreement as in the single b.v.p. solution. This is seen by comparing the first line of Table 11 with the first line of Table 7. A root mean square value of 1.02 mgals is obtained instead of 0.06 mgals. On the same line we can notice that the mean value of the 64800 $\overline{\Delta g}_{ij}$ recovered is no longer zero. The

Table 11 Statistics on Mixed b.v.p. Solution using GEML2 with Unity as Weight.

	δΔg (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	-.06	1.02	6.2	-5.5	0.40	2.08	5.0	-4.5
1	-.02	.39	2.7	-2.4	-.10	.80	2.3	-2.5
2	-.01	.11	.7	-.	-.06	.38	.9	-.9

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.00	14.06	44.0	-51.6	0.0	30.31	76.6	-104.2

		δΔg (%)		
Degree n	# of coeff.	# of iter.: 0	1	2
2	5	.17	1.63	.76
3	7	.43	1.35	.44
4	9	.81	1.83	.48
5	11	.87	2.20	.49
6	13	1.71	1.56	.47
7	15	2.20	.78	.35
8	17	3.35	1.82	.65
9	19	3.96	2.16	.66
10	21	3.04	2.79	.75
11	23	4.89	2.95	.87
12	25	5.80	3.53	.93
13	27	5.20	2.29	.70
14	29	6.92	3.67	1.05
15	31	9.77	4.08	1.21
16	33	10.26	5.11	1.47
17	35	12.49	5.44	1.73
18	37	9.66	4.51	1.25
19	39	12.82	5.19	1.40
20	41	11.39	4.64	1.44
21	31	19.91	5.43	1.64
22	31	15.86	4.79	1.84
23	11	15.31	7.76	2.19
24	11	14.35	4.77	1.48
25	11	15.69	5.22	1.66
26	5	24.74	5.82	1.37
27	9	19.18	4.24	1.17
28	11	19.00	3.61	1.10
29	6	n/a	n/a	n/a
30	2	n/a	n/a	n/a

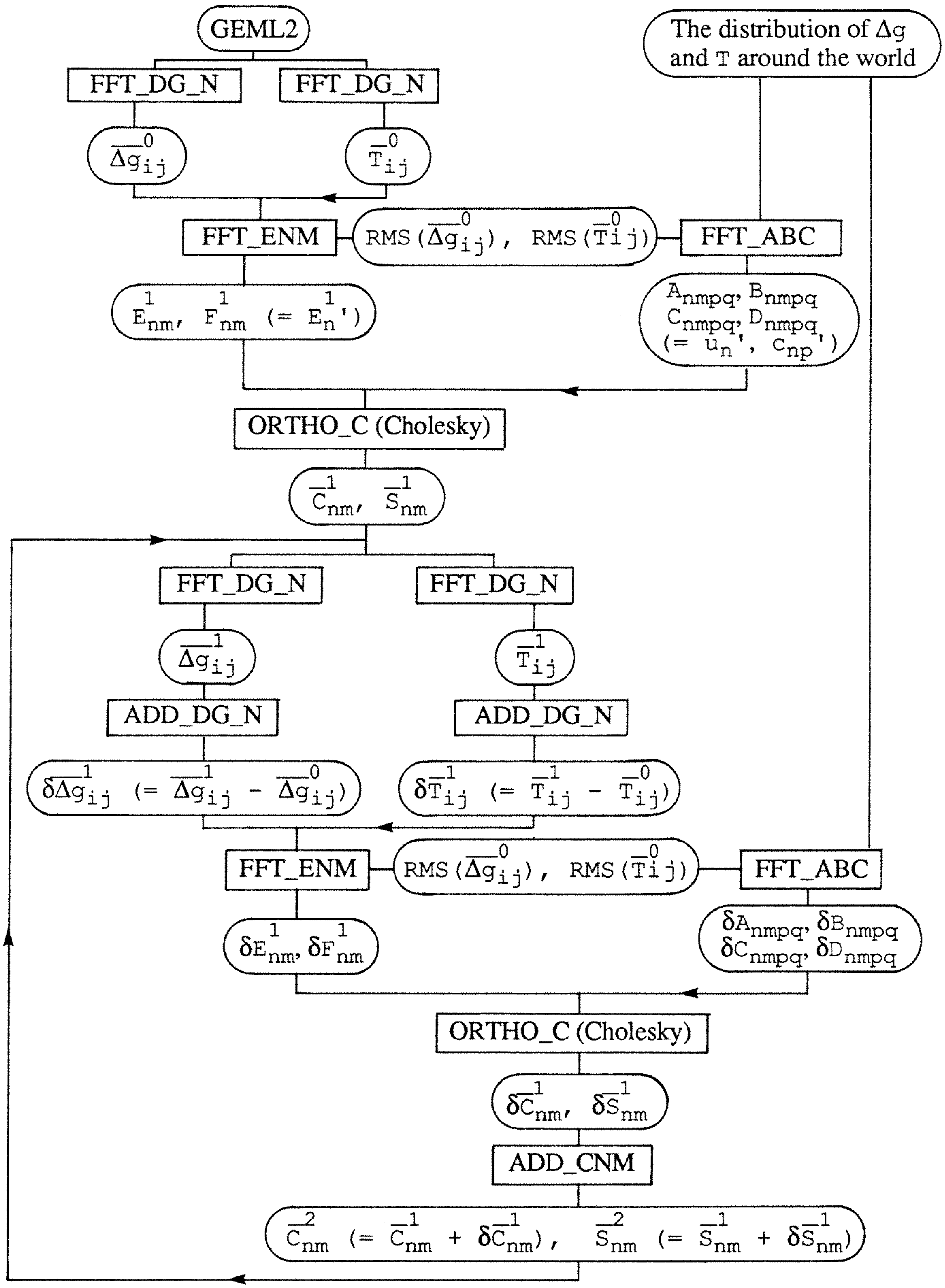

Figure 4 Flow Chart to Test the Mixed b.v.p. Solution. (Boxes are FORTRAN routines and curves are data sets).

solution is obviously wrong when no weight is used. The following test corrected the situation.

Table 12 shows the same computations with the $\overline{\Delta g}_{ij}$ and $\overline{T}_{ij}$ now scaled by their mean square values as computed by (4.5). Now the convergence is almost as fast as in the single b.v.p. solution (compare Table 12 with Table 7). And now the mean value of the 64800 $\overline{\Delta g}_{ij}$ computed from the recovered coefficients is zero. And the mean value of the 64800 mean $\overline{T}_{ij}$ values computed from the recovered coefficients is also zero. The improvement obtained here, with a convergence in only 3 iterations, proves undoubtedly that it is mandatory to use some kind of weight in the solution. The use of the mean square values is one possibility. However it is not understood how this scaling affects the mean value (and the zero degree harmonic). As mentionned earlier it is not clear why Arnold (1984, p.350) suggested this weight . Further research would be required to find out if the mean square values are the only weights that would solve the problem. The mean value in Table 12 was computed from the 64800 mean $\overline{\Delta g}_{ij}$ values derived from the coefficients of the solution. The mean value being zero should indicate that the $\overline{C}_{00}$ harmonic of the solution is zero. However the $\overline{C}_{00}$ term was not directly computed here since the software was developed to compute the coefficients starting at $\overline{C}_{20}$. In the future it would be more appropriate, specially with real data, to compute the $\overline{C}_{00}$ coefficient through the Cholesky solution and verify how close to zero $\overline{C}_{00}$ really is. In any case the results in Table 12 prove that we have a viable solution to the mixed b.v.p.. For the deterministic solution i.e. without iteration the 64800 $\overline{\Delta g}_{ij}$ are recovered with a RMS difference of 0.48 mgals and the $\overline{N}_{ij}$ with a RMS difference of 0.18 metre (Table 12).

Table 12 Statistics on Mixed b.v.p. Solution using GEML2 with the Mean Square Values as Weight.

	$\delta\Delta g$ (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	.48	2.21	-2.63	0.0	.182	.91	-1.17
1	0.0	.01	.05	-.04	0.0	.002	.02	-.02
2	0.0	.00	.00	.00	0.0	.000	.00	.00

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.00	14.06	44.0	-51.6	0.0	30.31	76.6	-104.2

		$\delta\Delta g$ (%)		
Degree n	# of coeff.	# of iter.: 0	1	2
2	5	.07	.00	.00
3	7	.14	.00	.00
4	9	.32	.00	.00
5	11	.37	.00	.00
6	13	.64	.01	.00
7	15	1.00	.01	.00
8	17	1.42	.01	.00
9	19	1.65	.01	.00
10	21	1.50	.01	.00
11	23	2.32	.03	.00
12	25	2.53	.03	.00
13	27	2.42	.04	.00
14	29	2.80	.04	.00
15	31	3.83	.05	.00
16	33	4.48	.07	.00
17	35	4.86	.05	.00
18	37	4.03	.06	.00
19	39	4.63	.06	.00
20	41	4.94	.07	.00
21	31	8.09	.10	.00
22	31	5.97	.10	.00
23	11	5.06	.09	.00
24	11	5.29	.09	.00
25	11	6.24	.12	.00
26	5	9.83	.12	.00
27	9	10.82	.16	.00
28	11	12.37	.13	.00
29	6	n/a	n/a	n/a
30	2	n/a	n/a	n/a

As a check, the software developed here that solve the mixed b.v.p. must also solve the single b.v.p.. First we entered only the $\overline{\Delta g}_{ij}$ in FFTENM to compute the E_{nm} and F_{nm} coefficients. As seen in (4.4) these coefficients are multiplied by $(n-1)$ instead of being divided by $(n-1)$ as in the familiar equation (E.1). But FFTABC computes for this single b.v.p. case a Gram matrix (left matrix of equation 3.17) which is diagonal and with terms in the diagonal which are equal to $(n-1)(n-1)$ coming from the factor $(n-1)(p-1)$. After having solved the system of equations using ORTHOC the results were exactly the same as the output from FFTCNM i.e. of the single b.v.p.. This was demonstrated analytically in chapter 3 just before equation (3.17).

All the software mentioned here can be found in appendix G; they are: PNMI, FFTDGN, FFTCNM, ADDDGN, ADDCNM, PNMI2, FFTENM, FFTABC, ORTHO and ORTHOC.

As was mentioned at the end of chapter 8 and as also seen from the cpu times in Table 10 the main drawback of the solution is the large system of equations to solve. To overcome this problem we have tried to solve the system by inverting only the diagonal. This possibility is very attractive since it would make the solution applicable with the existing mainframe computers and for high degree solution. Inverting the diagonal is no cost compared to solving a system of equations. Inverting the diagonal also means that only the integrals $\bar{I}_{nmnm}(\theta)$ and the coefficients A_{nmnm} and B_{nmnm} (instead of $\bar{I}_{nmpq}(\theta)$, A_{nmpq}, B_{nmpq}, C_{nmpq} and D_{nmpq}) are required to be computed which reduces drastically the number of coefficients to be computed. Table 13 shows the results of 5 iterations

Table 13 Statistics on Mixed b.v.p. Solution using GEML2 with the Mean Square Values as Weight and using only the Diagonal of the Gram Matrix

	$\delta\Delta g$ (mgals)				δN (metres)			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	17.0	90.3	-105	0.0	12.2	56.6	-69.1
1	0.0	7.0	47.8	-34	0.0	5.9	36.6	-27.0
2	0.0	4.2	27.9	-26	0.0	3.8	22.2	-17.8
3	0.0	3.1	17.9	-17	0.0	3.0	19.3	-13.9
4	0.0	2.7	15.6	-19	0.0	2.6	17.0	-13.7
5	0.0	2.5	14.9	-20	0.0	2.3	15.7	-13.5

Δg (mgals)				N (metres)			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.00	14.06	44.0	-51.6	0.0	30.31	76.6	-104.2

		$\delta\Delta g$ (%)					
Degree n	# of coeff.	# of iter.: 0	1	2	3	4	5
2	5	11	6	3	3	2	2
3	7	18	10	7	5	4	4
4	9	34	20	16	12	11	10
5	11	62	33	23	18	15	14
6	13	60	43	26	21	17	15
7	15	79	36	20	16	13	12
8	17	81	44	26	22	19	18
9	19	147	64	38	32	27	25
10	21	150	62	42	35	31	28
11	23	156	49	33	26	24	22
12	25	204	67	38	30	25	22
13	27	177	55	35	30	27	25
14	29	222	52	35	30	27	25
15	31	160	54	30	26	22	20
16	33	233	47	31	30	27	27
17	35	261	63	36	35	33	32
18	37	179	54	28	22	20	19
19	39	201	71	38	30	27	25
20	41	210	94	42	28	25	24
21	31	260	112	53	28	26	26
22	31	193	86	47	26	26	26
23	11	89	96	48	25	23	22
24	11	154	110	57	19	20	21
25	11	149	90	61	14	14	13
26	5	259	138	108	8	17	22
27	9	139	82	53	13	14	13
28	11	132	82	44	11	10	10
29	6	n/a	n/a	n/a	n/a	n/a	n/a
30	2	n/a	n/a	n/a	n/a	n/a	n/a

where we are trying to recover the GEML2 coefficients. It is converging but if we compare the RMS with the ones of Table 12 the convergence is rather slow. In fact it does not dispose one to try solving for higher degree solution. Some way to accelerate the convergence should be assessed. This is another suggestion for further research. The use of a banded matrix (Wenzel, 1985) could be a possible compromise between the use of the diagonal and a full matrix which could improve the convergence and reduce the cpu time. Only for an iterated solution is that option possible. For a deterministic solution no iteration is permitted and one must solve the system of equations with the full Gram matrix; unless one accepts that few iterations do not distort the coefficients. Only further testing will answer this question.

Meanwhile, it is easy to be convinced that this solution is to the mixed b.v.p. what the harmonic orthogonal relationship is to the classical single b.v.p. in physical geodesy. It is certainly the most equivalent solution. Thus we can do all the testing with the single b.v.p. and the results and understanding will apply to the mixed b.v.p. as well. This is why we did other testing with the single b.v.p. which can be done with a set of coefficients of higher degree and at lower cost than the mixed solution.

As shown by Colombo (1981) the computation of the spherical harmonic coefficients is contaminated by the sampling error. The "sampling error" (Colombo, 1981, p.13) includes two errors, the "aliasing error" and the "quadrature error". The aliasing error is commonly encountered in Fourier analysis. It is the error resulting from frequencies in the data to analyse, being mixed with other frequencies. To have mixed frequencies is to have frequencies added or subtracted with other frequencies. It is then said to have "aliased frequencies". The quadrature error is the error resulting from the numerical integration. Some integrals satisfy the mean-value theorem for integrals and their numerical integration is performed exactly without quadrature error. This is the case in Fourier analysis where the Fourier integral formulas can be computed exactly using finite discrete Fourier formulas dependent on some regular grid (Colombo, 1981, p.10). In spherical harmonic analysis the second theorem of the mean for integrals (Gerald and Wheatley, 1984, p.A.3) could be satisfied if a special grid was used where the parallels are situated at the same latitudes as the zeros of $P_{N+1}(\cos\theta)$ (Colombo, 1981, p.12) (Payne, 1971). Because an equal angular grid is used, for practical reasons, the second theorem of the mean for integrals is not satisfied and the numerical integration gives rise to a numerical integration error, the quadrature error. Note that even if the quadrature error was eliminated using the special grid we would still be left with the aliasing error. The de-smoothing operator is used to attenuate the sampling error, it does not eliminate it completely.

During numerical simulations the aliasing error can be made totally absent and the quadrature error can then be eliminated completely. These simulations are very instructive and the results will be shown below. In practice however the quadrature error and the aliasing error can no longer be separated and we refer to them by the sampling error.

The fact that the iterative process permits one to recover the coefficients exactly is true only when there is no aliased frequency in the data to analyse. In this case the recovered coefficients have no aliasing error and the quadrature error can be eliminated by iterations. Such regular grid of data (point or mean gravity anomalies) which does not contain aliased frequencies can always be computed using the harmonic synthesis (appendix F). When N is the maximum degree at which an apriori geopotential model is used, the rule to compute a regular grid of values without aliasing the frequencies is to compute point values at every $\Delta\theta° \leq 180°/N$ or mean values of size $\Delta\theta° \leq 180°/N$. If this rule is not respected the grid values will contain aliased frequencies. This will be numerically demonstrated below. This rule is in accordance with the Nyquist frequency in Fourier analysis. It can be shown that "k" values regularly spaced on a circle where $k = 180/\Delta\theta°$ can contain only N non-aliased frequencies where $N \leq k$ i.e. $N \leq 180°/\Delta\theta°$. This rule has important implications on real data. A grid of "observed" gravity anomalies or geoid undulations of size $\Delta\theta°$ contains an infinite number of frequencies, and thus $N > 180°/\Delta\theta°$, and the grid contains aliased frequencies. In that case the coefficients obtained from the harmonic analysis will be tainted with the aliasing error, in addition to the quadrature error, and the iterative process under the effect of the aliasing error will produce distorted coefficients. This will be simulated below. However the simulation will show that it takes more than one iteration to distort the solution and that one iteration can improve the solution.

There is also another very important advantage of the iteration process. We know that if the Pellinen-Meissl smoothing operator β_n or the de-smoothing operator η_n would not be known we would not be able to recover as well the coefficients. This is true for the case of the single b.v.p. as well as for the case of the mixed b.v.p.. The de-smoothing operator is known for these two cases on the sphere only. With the iteration process it is not required to know the smoothing operator. Hence in cases where we are unable to derive the smoothing operator as for the complicated cases involving the ellipsoid or the topography the iteration process would enable us to recover acceptably well the coefficients.

The following simulations were done to sustain these facts. We have seen that using the iteration process we can recover all the coefficients of the RAPP81 set, up to degree 180, from 1° X 1° mean anomalies (Table 8). This time only the first coefficients up to degree 36 will be recovered from the same set of 1° X 1° mean anomalies computed from the coefficients up to degree 180 of RAPP81. The calculations are, however, difficult to follow when one iterates the solution. One should follow the iterative computations using the flow chart in Figure 3. From RAPP81 to degree 180, 1° X 1° mean anomalies are computed. From these mean anomalies, a first set of coefficients up to degree 36 are recovered . From these coefficients, a new set of 1° X 1° mean anomalies are computed. This set of mean anomalies is subtracted from the initial set of 1° X 1° mean anomalies computed from RAPP81 coefficients to degree 180. Here the residuals of equation (3.12) have just been computed and these are being minimized in the iteration process. They are large as expected with a RMS of 16.76 mgals (Table 14) compared to the RMS of 22.54 mgals (Table 14) for the true set of 1° X 1° mean anomalies computed from the RAPP81 coefficients to degree 180 and compared to the RMS of 15.07 mgals (Table 14) of the true set of 1° X 1° mean anomalies computed from the coefficients up to degree 36 of RAPP81. The set of 1° X 1° mean residuals computed above is entered as shown in Figure 3 in the harmonic "analysis" software to compute corrections to the previously recovered coefficients up to degree 36. From these improved coefficients up to degree 36 another set of 1° X 1° mean anomalies are computed and their subtraction with the true set of 1° X 1° mean anomalies computed from RAPP81 to degree 180 gives the new

Table 14 Statistics on Single b.v.p. Solution using RAPP81 and Recovering up to Degree 36 From 1 Degree Mean Anomalies.

	$\delta\Delta g$ (mgals) (36-36)*				$\delta\Delta g$ (mgals) (36-180)**			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	.03	.2	-.2	0.0	16.76	235.7	-203.1
1	0.0	.01	.1	-.1	0.0	16.76	235.7	-203.1
true residuals					0.0	16.77	235.8	-203.7

Δg (mgals) (36)***				Δg (mgals) (180)****			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	15.07	64.4	-59.3	0.0	22.54	255	-229

Table 15 Statistics on Single b.v.p. Solution using RAPP81 and Recovering up to Degree 36 From 2 Degree Mean Anomalies.

	$\delta\Delta g$ (mgals) (36-36)*				$\delta\Delta g$ (mgals) (36-180)**			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	.42	4.2	-3.4	0.0	11.99	89.7	-116.2
1	0.0	.37	4.1	-3.3	0.0	11.99	89.4	-116.2
2	0.0	.37	4.1	3.4	0.0	11.99	89.4	-116.2
true residuals					0.0	12.00	89.4	-117.6

Δg (mgals) (36)***				Δg (mgals) (180)****			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	14.68	64.0	-58.2	0.0	18.98	128.5	-130.7

* Statistics between the Δg_{ij} computed from the recovered coefficients up to degree 36 and the apriori Δg_{ij} computed from RAPP81 up to degree 36.

** Statistics between the Δg_{ij} computed from the recovered coefficients up to degree 36 and the apriori Δg_{ij} computed from RAPP81 up to degree 180.

*** Statistics of the apriori Δg_{ij} computed from RAPP81 up to degree 36.

**** Statistics of the apriori Δg_{ij} computed from RAPP81 up to degree 180.

residuals. The RMS of these residuals is shown as the 1 iteration case in Table 14 and they are being minimized at 16.76 mgals, as they should since the true residual should be 16.77 mgals. The agreement between the coefficients up to degree 36 recovered after each iteration with the RAPP81 coefficients up to degree 36 were computed. The agreement is given on the left side of Table 14 in terms of RMS difference values which were derived using the 1° X 1° mean anomalies computed from the recovered coefficients and the RAPP81 coefficients to degree 36. These RMS difference values on the left side of Table 14 show that a set of coefficients up to degree 36 can be recovered exactly by iterations from a set of 64800 mean $\overline{\Delta g}_{ij}$ containing RAPP81 information up to degree 180. One must realize that this result was not evident because the iterations are processed with large residuals of 16.76 mgals. How come the frequencies information from degree 37 to 180 did not contaminate the solution during the iterations? These residuals must contain clean information, i.e. very specific frequencies from degree 37 to degree 180 which were not aliased with the lower frequencies. And it is so because the 1° X 1° grid used could contained all the frequencies up to degree 180 according to the above rule of the Nyquist frequency. We must assume this is why the gravity information above degree 36 was not mixed with the first 36 degree. In conclusion, with the iterations the sampling error was reduced to zero. However it is really just the quadrature error which has been reduced to zero since there was no aliased frequencies in the data to produce the aliasing error. In the same way we have been able to recover exactly different set of coefficients to degree 45, 90, 120 and 160 with few iterations and large residuals. In addition we have been able to recover exactly the coefficients up to degree 90 from a grid of 2° X 2° mean anomalies computed with RAPP81 up to degree 90. This shows the validity of the above rule to compute a regular grid without aliasing the frequencies. As long as the grid values do not contain aliased frequencies the coefficients can be recovered exactly after few iterations. But we will see that it is not the case when we use a grid of data which contains aliased frequencies.

We have computed a set of 2° X 2° mean anomalies from the same RAPP81 coefficients to degree 180. When computing these 2° X 2° mean anomalies the frequencies are being aliased. When the harmonic synthesis is performed with the FFT algorithm it is easy to see that the aliasing of the frequencies occurs when computing equation (F.21) of appendix F (or Colombo, 1981, p.10 and 106). With other algorithms such as the efficient trigonometric algorithm in (Rizos, 1979) or the usual computation on a point-by-point basis, the aliasing of the frequencies is not as apparent but certainely present since all these algorithms give the same numerical result. Following the flow chart in Figure 3 we tried to recover a set of coefficients to degree 36 from this set of 2° X 2° mean anomalies. Table 15 shows that with the iteration process the residuals converge (right column) to an RMS of 11.99 mgals. But we are unable to recover the coefficients up to degree 36 exactly. By computing a 2° X 2° grid of mean anomalies from the recovered coefficients up to degree 36 and subtracting these values from a 2° X 2° grid of mean anomalies computed using RAPP81 to degree 36 the RMS differences in the left column of Table 15 were obtained. These RMS differences of .42, .37 and .37 mgals do not converge to zero. This is at first glance surprising since we would have thought that according to the rule of the Nyquist frequency we can recover exactly the coefficients up to the frequency $N = 180/\Delta\theta°$ from a grid of $\Delta\theta°$ X $\Delta\theta°$. We should have been able to recover exactly the coefficients up to degree 90 from a set of 2° X 2° mean anomalies. However as it is shown here even the frequency information up to degree 36 is no longer contained in the 2° X 2° mean anomalies and this is because the frequencies in the set of 2° X 2° mean anomalies were mixed i.e. aliased by forcing into the grid all the RAPP81 frequency information up to degree 180. Here we can no longer differentiate between the aliasing error and the quadrature error, and the RMS differences of .42 and .37 mgal must be referred to as the sampling error.

Table 16 shows the results when trying to recover a set of coefficients to degree 90 from the same grid of 2° X 2° mean anomalies generated from the RAPP81 coefficients to degree 180. Here again the iterations converge (right column), but we are unable to recover the anomalies from the coefficients up to degree 90 (left column). We see two possible explanations. The grid of 2° X 2° mean anomalies has "lost" some information about the frequencies lower than 90 degrees, possibly when they were aliased with higher frequencies. Or the solutions in Table 15 and 16 have "added" information from the higher frequencies into the recovered coefficients. If this is true, then even the deterministic solution has already distorted the solution with the higher frequencies. This is supported by the RMS value of the residuals of the deterministic solution (the 0 iteration case) in Table 16 which is 6.21 mgals, smaller than the true residuals of 6.67 mgals (Table 16) between the 2° X 2° mean anomalies computed with RAPP81 to degree 90 and to degree 180.

Table 17 shows the same test but without using the de-smoothing operator η_n. Without the de-smoothing operator we knew the deterministic solution would not recover the coefficients as well; compare the RMS value 3.41 mgals (Table 17) with 2.80 mgals (Table 16). But with one iteration the RMS value (the sampling

Table 16 Statistics on Single b.v.p. Solution using RAPP81 and Recovering up to Degree 90 From 2° Mean Anomalies with the De-smoothing Operator η_n.

	$\delta\Delta g$ (mgals) (90-90)*				$\delta\Delta g$ (mgals) (90-180)**			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	2.80	22.0	-35.7	0.0	6.21	55.5	-54.5
1	0.0	2.85	23.7	-40.5	0.0	5.96	54.6	-54.2
2	0.0	2.97	27.7	-42.0	0.0	5.93	54.7	-53.9
true residuals					0.0	6.67	57.3	-54.0

Δg (mgals) (90)***				Δg (mgals) (180)****			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	17.75	131	-111	0.0	18.99	128.5	-130.7

* Statistics between the Δg_{ij} computed from the recovered coefficients up to degree 90 and the apriori Δg_{ij} computed from RAPP81 up to degree 90.

** Statistics between the Δg_{ij} computed from the recovered coefficients up to degree 90 and the apriori Δg_{ij} computed from RAPP81 up to degree 180.

*** Statistics of the apriori Δg_{ij} computed from RAPP81 up to degree 90.

**** Statistics of the apriori Δg_{ij} computed from RAPP81 up to degree 180.

Table 17 Statistics on Single b.v.p. Solution using RAPP81 and Recovering up to Degree 90 From 2° Mean Anomalies without the De-smoothing Operator η_n.

	$\delta\Delta g$ (mgals) (90-90)*				$\delta\Delta g$ (mgals) (90-180)**			
ITERATION	MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0	0.0	3.41	35.4	-31.3	0.0	6.78	58.7	-66.4
1	0.0	2.69	19.8	-38.0	0.0	6.09	54.8	-57.0
2	0.0	2.79	22.0	-39.8	0.0	5.97	54.7	-54.8
3	0.0	2.89	25.8	-41.2	0.0	5.94	54.8	-54.2
true residuals					0.0	6.67	54.0	-57.3

Δg (mgals) (90)***				Δg (mgals) (180)****			
MEAN	RMS	MAX	MIN	MEAN	RMS	MAX	MIN
0.0	17.75	131	-111	0.0	18.99	128.5	-130.7

		$\delta\Delta g$ (%)			
Degree n	# of coeff.	# of iter.: 0	1	2	3
2	5	.06	.04	.04	.04
3	7	.11	.04	.04	.04
4	9	.20	.12	.12	.12
5	11	.27	.07	.07	.07
6	13	.44	.22	.22	.22
7	15	.43	.21	.21	.21
8	17	.77	.31	.30	.30
10	21	1.18	.49	.49	.49
20	41	4.95	3.18	3.19	3.19
30	61	11.81	7.26	7.07	7.05
40	81	16.67	11.28	11.29	11.31
50	101	24.29	15.29	15.06	15.11
60	121	31.33	22.29	22.91	23.43
70	141	43.21	35.89	37.01	38.00
80	161	50.68	41.84	43.15	44.97
87	175	53.57	44.51	46.77	49.81
88	177	60.53	51.55	52.71	55.24
89	179	57.81	50.41	53.24	56.68
90	181	54.28	49.05	53.74	58.12

error) in Table 17 is now 2.69 mgals, compared to 2.80 mgals, with the deterministic solution in Table 16. At this point we should agree that the solution with one iteration without the de-smoothing operator is as acceptable as the deterministic solution with the de-smoothing operator. It is however dangerous to iterate too many times as it can distort the solution as shown in Table 17. There are two important messages one should remember from the above discussion. A set of $1° \times 1°$ "observed" mean gravity anomalies like those of Rapp (1983) is a set in which the frequencies are "aliased". This is so because a set of "observed" $1° \times 1°$ mean anomalies contains an infinite number of frequencies of the earth gravity field but not all the frequencies can be recovered exactly, like the simulated data set with aliased frequencies. This is very important to remember because if we try to compute a set of coefficients up to degree 180 from this set of "observed" $1° \times 1°$ mean anomalies the iteration process will not converge to zero but it will converge in the same manner as given in Tables 16 and 17. And because of this the second important message of all this is that Table 17 shows that if the de-smoothing operator η_n or β_n was not known, one iteration would enable us to obtain as good or better a solution than using the de-smoothing operator η_n. And thus, other single or mixed b.v.p. involving the ellipsoid or the topography for which the smoothing operator is not known could be solved. Similar tests but involving few degrees set of coefficients showed that the solution to the mixed b.v.p. obtained in this dissertation reacts in the same way as the above tests for the single b.v.p..

In conclusion, it was seen that the coefficients cannot be recovered exactly after some iterations when using observed gravity anomalies because the frequencies in it are aliased. We have seen the rule to create a simulated set of anomalies containing aliased frequencies. And the conclusion is that any test to look at how well coefficients can be recovered using any method such as least-squares collocation or least-squares adjustment or integral formula should be done using such sets of values containing aliased frequencies.

This chapter has shown numerically a solution to the altimetry-gravimetry problem. It showed the equivalence between the solution to this mixed b.v.p. and the classical solution to the single b.v.p. in physical geodesy. It also showed how an iterative process can improve existing solution and it suggests that this iterative process could help solve other single and mixed b.v.p. involving ellipsoid or topography.

CONCLUSION

A solution to the altimetry-gravimetry problem, a mixed boundary value problem, has been developed and tested numerically. The solution is a set of spherical harmonic coefficients. These coefficients are here denoted $[\bar{C}_{nm}, \bar{S}_{nm}]$. The test was made to recover a set of 837 geopotential coefficients known apriori up to degree and order 28. The solution established and solved the matrix equation (3.17) in which the array $[T_n] = [\bar{C}_{nm}, \bar{S}_{nm}]$ are the unknown coefficients to estimate.

The elements of the right hand side array of (3.17) are given by equation (4.4). These elements are denoted $[E_{nm}, F_{nm}]$. Already one can see in (4.4) the similarity between this solution and the usual solution of the single b.v.p. given by the integration over the sphere of the gravity anomalies to compute the spherical harmonic coefficients $[\bar{C}_{nm}, \bar{S}_{nm}]$ (see equation (E.1) in appendix E).

The left hand side matrix in (3.17) is called the Gram matrix, and its elements are given by equation (6.1). These elements are denoted $[A_{nmpq}, B_{nmpq}, C_{nmpq}, D_{nmpq}]$. The Gram matrix is positive definite and thus the Cholesky factorization can be used to solve this system of equation (3.17). This we have done and we could recover the coefficients as accurately as we could do it to solve the single b.v.p. using the usual orthogonality relationship.

This solution to the altimetry-gravimetry problem was given by Brillouin (1916) and Arnold (1978). It is the solution of a least-squares method but again the usual integration to compute the spherical harmonic coefficients that solves the single b.v.p. is based on the same least-squares method (Brillouin, 1916). In the single b.v.p. case the Gram matrix in (3.17) is a diagonal matrix for which the inverse is simply obtained by taking the inverse of each element in the diagonal. The solution (3.17) is to the altimetry-gravimetry problem what the usual solution (E.1) is to the single b.v.p.. It is the most natural solution and perhaps the simplest solution to the mixed b.v.p. if one considers (E.1) to be the simplest solution for the single b.v.p..

The problematic part of this solution is the Cholesky factorization since the system of equation to solve is terribly large when the high degree coefficient are sought for. This is why we were restricted to test numerically the solution with a smaller set of coefficients than we usually carry out for the single b.v.p.. One good aspect during the factorization is the high stability of the Gram matrix due to the structure we gave it, by ordering the unknown coefficients by degree, from the largest coefficients to the smallest coefficients $\bar{C}_{20}, \bar{C}_{21}, \bar{S}_{21}, \bar{C}_{22}, \bar{S}_{22}, \bar{C}_{30}$, etc.

The most efficient way we found to compute the elements of the Gram matrix and of the right hand side array in (3.17) was the fast Fourier transform. The fast Fourier applications used are mostly based on the spherical harmonic "analysis" and "synthesis" of Colombo (1981). These are described in here respectively in appendix E and F. By using the fast Fourier technique we restricted us to use a regular equiangular grid covering globally the earth. On such a regular grid no overlapping of data (2 values in the same block) is permissible. Nevertheless the mixed coverage of mean gravity anomaly and mean disturbing potential values can be distributed very randomly like figure 9.2, as well as by oceans and continents like figure 9.1. Because of the mixture of gravity anomaly and disturbing potential data on the same row of latitude, the efficiency of the fast Fourier technique is partially destroyed making the computations for the elements in (3.17) more time consuming then the usual spherical harmonic analysis or synthesis. However the computation of the coefficients $[E_{nm}, F_{nm}]$ which are the elements of the right hand side array of (3.17) and the computation of the coefficients $[A_{nmpq}, B_{nmpq}, C_{nmpq}, D_{nmpq}]$ which are the coefficients of the Gram matrix in (3.17) are independent by degree like the $[\bar{C}_{nm}, \bar{S}_{nm}]$ coefficients computed from the usual integration (E.1) is. Thus the computation of these coefficients can always be performed independently, thus by small jobs or on a super computer having parallel processors.

In this solution to the altimetry-gravimetry problem the input observations differ from the input in the single b.v.p. in that they must be scaled. The input must be T/μ_T and $\Delta g/\mu_G$ where μ_T and μ_G are respectively the mean square values of T on the domain σ_1 and Δg on the domain σ_2. The mean square values of T/μ_T and $\Delta g/\mu_G$ are both equal to unity. The use of these mean squares as weight in this least-squares solution is a necessary weighting scheme which takes out the inconsistency between the zero order harmonic of both group of data, the T and Δg.

When solving the single b.v.p. by the usual integration over the sphere of "point" gravity anomalies, recurrence relations are required to compute "associated Legendre functions". When only "mean" gravity anomalies over some block size are available then the integration requires recurrence relations to compute the "integral over some block size of one associated Legendre function" denoted $[\int\overline{P}_{nm}d\sigma]$. To solve the altimetry-gravimetry problem with "mean" gravity anomalies and "mean" geoid undulations it requires recurrence relations which computes the "integral of two associated Legendre functions", i.e. $[\int\overline{P}_{nm}\overline{P}_{pq}d\sigma]$. These recurrence relations were here derived for the first time in chapter 7. While it is the first time that an application requires these new recurrences other solutions to the mixed b.v.p. might require them in the future. It would appear that already the proposed solution by (Sacerdote and Sanso, 1985) could use these newly developed recurrence relations.

Because it is a least-squares solution we have shown an iterative process which can be used to improve the mixed b.v.p. solution as well as the single b.v.p. solution. This iteration process permits us to recover the coefficients even in situations where the smoothing operator like the Pellinen/Meissl operator β_n would be unknown. Such situation could occur when trying to solve the single b.v.p. on the ellipsoid or when taking into account the topography and the sea-surface topography.

What is left is to compute this solution to a higher degree, to use real altimetry data and to compare such solution with the existing single b.v.p. solutions; not before would we have solved the real problem of altimetry-gravimetry.

LIST OF REFERENCES

Arnold, K., 1978. "The Spherical-Harmonics Expansion of the Gravitational Potential of the Earth in the External Space and its Convergence", Gerlands Beitr. Geophysik, Leipzig 87, pp.81-90, 1978.

Arnold, K., 1981. "Complex Evaluation of Gravity Anomalies and of Data Obtained from Satellite Altimetry", Gerlands Beitr. Geophysik, Leipzig 90, pp.38-42, 1981.

Arnold, K., 1984. "The Compatibility Conditions, the Uniqueness and the Solution of the Mixed Boundary Value Problem of Geodesy", Gerlands Beitr. Geophysik, Leipzig 93, pp.339-355, 1984.

Bjerhammar, A., and L. Svensson, 1978. "The Convergence of Harmonic Reduction to an Internal Sphere", Division of Geodesy, The Royal Institute of Technology, Stockholm, 12 pp., 1978.

Bjerhammar, A., 1983. "A Stochastic Approach to the Mixed Boundary Value Problems in Physical Geodesy", Geodesy in Transition, pp.25- 42, The University of Calgary, Alberta, Canada. July 1983.

Brillouin, M., 1916. "La méthode des moindres carrés et les équations aux dérivées partielles de la physique mathématique", Annales de Physique, 9e Série, Tome VI, pp.137-175, Septembre-Octobre 1916.

Christodoulidius, D.C., and Katsambalos, K.E., "An Analysis on the Precision in the Computation of the Integrals of the Fully Normalized Associated Legendre Functions", Internal Report of the Dept. of Geodetic Science, The Ohio State University, 1977.

Collatz, L., 1960. "The Numerical Treatment of Differential Equations", 3rd Edition, pp.28-35, Springer-Verlag, Berlin, 1960.

Colombo, O., 1978. "Optimal Estimation from Data Regularly Sampled on a Sphere with Applications in Geodesy", Reports of the Dept. of Geodetic Science, No.291, The Ohio State University, Columbus, Ohio, 29 pp., September, 1979.

Colombo, O., 1981. "Numerical Methods for Harmonic Analysis on the Sphere", Reports of the Dept. of Geodetic Science, No.310, The Ohio State University, Columbus, Ohio, 140 pp., March 1981.

Courant, R. and D., Hilbert, 1953. "Methods of Mathematical Physics", First English Edition, Translated and Revised from German Original, 2 Volumes, Interscience Publisher, New York, 1953.

Dongarra, J.J., J.R. Bunch, C.B. Moler and G.W. Stewart, 1979. "LINPACK Users' Guide", Society for Industrial and Applied Mathematics, SIAM, 365 pp., Philadelphia, 1979.

Freeden, W., 1983. "Least Square Approximation by Linear Combinations of (Multi)Poles", Reports of the Dept. of Geodetic Science, No.344, The Ohio State University, Columbus, Ohio, 86 pp., April 1983.

Gerald, C.F. and P.O. Wheatley, 1984. "Applied Numerical Analysis",Addison-Wesley Publishing Company Inc., 3rd Edition, 621 pp., 1984.

Gerstl, M., 1980. "On the Recursive Computation of the Integrals of the Associated Legendre Functions", Manuscripta Geodaetica, Vol.5, pp.181-199, 1980.

Lerch, F.J., S.M. Klosko and G.B. Patel, 1982. "A Refined Gravity Model from Lageos (GEM-L2)", Geophysical Research Letters, Vol.9, No.11, pp.1263-1266, November 1982.

Gleason, D., 1983. "On Solving the Stability Problem in Computing the Integrals of the Legendre Functions", Technical Paper in the Dept. of Geodetic Science, The Ohio State University, Columbus, Ohio, 105 pp., August 1983.

Gleason, D., 1985. "Using the Fast Fourier Transform and Fortran 77 in Gravimetric Spherical Harmonic Expansions", Term Paper in the Dept. of Geodetic Science, The Ohio State University, Columbus, Ohio, 58 pp., Winter Quarter 1985.

Gradshtein, I.S., and I.M. Ryzhik, 1965. "Table of Integrals Series and Products", Academic Press, New York, 1086 pp., 1965.

Heiskanen, W., and H. Moritz, 1967. "Physical Geodesy", W.H. Freeman and Co., San Francisco, 1967.

Hobson, E.W., 1965. "The Theory of Spherical and Ellipsoidal Harmonics", Chelsea Publishing Company, New York, Second Reprint, 1965.

Hofmann-Wellenhof, B., 1985 . "The Use of Multipoles for the Altimetry-Gravimetry Problem", Institute of Theoretical Geodesy, Technical University of Graz, pp.75-85, 1985.

Holota, P., 1980a. "Direct Methods for Geodetic Boundary Problems", Proceedings of 4th International Symposium on Geodesy and Physics of the Earth, GDR, Karl-Marx-Stadt, May 1980.

Holota, P., 1980b. "Direct Methods for Geodetic Boundary Problems", Proceedings of the 4th International Symposium on Geodesy and Physics of the Earth, GDR, Karl-Marx-Stadt, 35 pp., May 1980.

Holota, P., 1980c. "The Altimetry-Gravimetry Boundary Value Problem", Int. Scientific Conference of Section 6 Intercosmos, Albena, Varna, Sept. 1980.

Katsambalos, K.E., 1979. "The Effect of the Smoothing Operator on Potential Coefficient Determinations", Reports of the Dept. of Geodetic Science, No.287, The Ohio State University, Columbus, Ohio, 73 pp., March 1979.

Kaula, W.M., 1966. "Tests and Combination of Satellite Determinations of the Gravity field with Gravimetry", Journal of Geophysical Research, Vol.71, No.22, pp.5303-5314, November 1966.

Moritz, H., 1980a. "Geodetic Reference System 1980", The Geodesist Handbook 1980, BulletinGeodesique, Vol.54, No.3, pp.395-405, 1980.

Moritz, H., 1980b. "Advanced Physical Geodesy", Abacus Press, Tunbridge Wells Kent, 500 pp, U.K., 1980.

Paul, M.K. 1978. "Recurrence Relations for Integrals of Associated Legendre Functions", Bulletin Geodesique, Vol.52, pp.177-190, 1978.

Payne, M.H., 1971. "Truncation Effects in Geopotential Modelling", Analytical Mechanics Associates, Seabrook, Maryland, 1971.

Pearson, C.E., 1974. "Handbook of Applied Mathematics", Van Nostrand, Reinhold cie, New York, 1974.

Pellinen, L.P., 1966. "A Method for Expanding the Gravity Potential of the Earth in Spherical Functions", Transactions of the Central Scientific Research Institute of Geodesy, Aerial Survey and Cartography, Issue 171, 92p. published by Nedra, Moscow, 1966; Translation ACIC-TC-1282, available: NTIS, Springfield, VA, AD661810.

Rapp, R.H., 1969. "Analytical and Numerical Differences Between Two Methods for the Combination of Gravimetric and Satellite Data", Bollettino di Geofisica Teorica ed Applicata, Vol.XI, N.41-42, pp.108-118, 1969.

Rapp, R.H., 1977. "Potential Coefficient Determinations from 5 Degrees Terrestrial Gravity Data", Reports of the Dept. of Geodetic Science, No.251, The Ohio State University, Columbus, Ohio, 65 pp., Jan. 1977.

Rapp, R.H., 1978. "A Global 1 Degree X 1 Degree Anomaly Field Combining Satellite, Geos-3 Altimeter and Terrestrial Anomaly Data", Report of the Dept. of Geodetic Science, No.278, The Ohio State University, Columbus, Ohio, 19 pp., September 1978.

Rapp, R.H. 1979. "Geos 3 Data Processing for the Recovery of Geoid Undulations and Gravity Anomalies", Journal of Geophysical Research, Vol.84, No.8, pp.3784-3792, July 1979.

Rapp, R.H., 1981. "The Earth's Gravity Field to Degree and Order 180 Using Seasat Altimeter Data, Terrestrial Gravity Data, and Other Data", Report of the Dept. of Geodetic Science, No.322, The Ohio State University, Columbus, Ohio, 42 pp., December 1981.

Rapp, R.H. 1983. "The Development of the January 1983 1 Degree X 1 Degree Mean Free-Air Anomaly Data Tape", Internal Report of the Dept. of Geodetic Science, The Ohio State University, Columbus, Ohio, July 1983.

Rapp, R.H. 1984. "The Determination of High Degree Potential Coefficient Expansions from the Combination of Satellite and Terrestrial Gravity Information", Report of the Dept. of Geodetic Science, The Ohio State University, Columbus, Ohio, No.361, 33 pp., September 1984.

Rizos, C., 1979. "An Efficient Computer Technique for the Evaluation of Geopotential From Spherical Harmonic Models", Australian Journal of Geodesy, Photogrammetry and Surveying, No.31, pp.161-170, Dec.1979.

Sacerdote, F., and F. Sanso, 1985. "Overdetermined Boundary Value Problems in Physical Geodesy", Manuscripta Geodaetica, Vol.10, No.3, pp.195-207, Springer-Verlag, New York, 1985.

Sanso, F., 1981. "Recent Advances in the Theory of the Geodetic Boundary Value Problem", Advances in Geodesy, pp.13-25, 1984 and Reviews of Geophysics and Space Physics, Vol.19, No.3, pp.437-449, August 1981.

Sanso, F., 1983. "A Discussion on the Altimetry-Gravimetry Problems", Geodesy in Transition, pp.71-107, The University of Calgary, Alberta, July 1980.

Sanso, F., and B. Stock, 1985. "A Numerical Experiment in the Altimetry-Gravimetry", Manuscripta Geodaetica, Vol.10, No.1, pp.23-31, Springer-Verlag, New York, 1985.

Sjoberg, L., 1980. "A Recurrence Relation for the Beta-n Function", Bulletin Geodesique, Vol.54, No.1, pp.69-72, 1980.

Sjoberg, L., 1982. "On the Altimetry-Gravimetry Boundary Problem", Bollettino di Geodesia E Scienze Affini, No.4, p.377-392, 1982.

Svensson, S.L., 1985. "Some Remarks on the Altimetry-Gravimetry Problem", 1st Hotine-Marussi Symposium on Mathematical Geodesy, Roma, 26 pp., June 1985.

Wenzel, H-G., 1985. "Hochaufloesende Kugelkenktionsmodelle fuer das Gravitationspotential der erde", Wiss. Arb. Fachrichtung Vermessungswesen der Universitaet Hannover, 1985.

APPENDIX A. Recurrence Relations for Associated Legendre Functions.

The purpose of this appendix is to have an easy and consistent reference list of recurrence relations for the functions $P_{nm}(t)$. From (Gradshteyn and Ryzhik, 1965, eq.8.810) here abbreviated as (GR-8.810) we have

$$P_{nm}(t) = \frac{(-1)^m}{2^n\ n!} (1-t^2)^{m/2} \frac{d^m P_n(t)}{dt^m} \quad . \qquad (A.1)$$

In (HM,(1-60)), in (Paul, 1978, eq.10) and in most geodetic references the definition of $P_{nm}(t)$ used does not include the $(-1)^m$ term like in (A.1). We shall also follow the geodetic definition and use the following definition

$$P_{nm}(t) = \frac{(1-t^2)^{m/2}}{2^n\ n!} \frac{d^m P_n(t)}{dt^m} \quad . \qquad (A.2)$$

All our recurrence relations are taken from (Gradshteyn and Ryzhik, 1965). From (GR-8.812) with $m = n$ we have

$$P_{nn}(t) = \frac{(2n)!}{2^n\ n!} (1-t^2)^{n/2} \qquad (A.3)$$

where the $(-1)^m$ term was taken out to satisfy the definition in (A.2).

From (GR-8.733.1a) and (GR- 8.733.1b) we have

$$(1-t^2)\frac{dP_{kl}(t)}{dt} = (k+l)tP_{kl}(t) - (k-l+1)P_{k+1,l}(t), \qquad (A.4)$$

$$= -\ ktP_{kl}(t) + (k+l)P_{kl}(t) \qquad . \qquad (A.5)$$

From (GR-8.733.2) we have

$$(2k+1)tP_{kl}(t) = (k-l+1)P_{k+1,l}(t) + (k+l)P_{k-1,l}(t). \qquad (A.6)$$

From (GR-8.733.4) we have

$$P_{k-1,l}(t) - P_{k+1,l}(t) = -(2k+1)(1-t^2)^{1/2}\ P_{k,l-1}(t) \qquad (A.7)$$

to which we have added a sign correction to satisfy the definition (A.2). And from (GR-8.753) we have

$$P_{kl}(t) = 0 \qquad \text{for } l > k \qquad . \qquad (A.8)$$

APPENDIX B. Derivation of Equation (7.2).

The purpose of this appendix is not as much to show the derivation of equation (7.2) which solves (7.1) as to show a model to follow in deriving the more complicated equation (7.24).

The derivation of (7.2) is shown in (Paul, 1978), (Gerstl,1980) and (Gleason, 1983) and the idea is as followed.

Similar to (7.1), let define the non-normalized integral I_{nm}

$$I_{nm} = I_{nm}(t_S, t_N) = \int_{t_S}^{t_N} P_{nm}(t)\ dt \quad . \qquad \text{(B.1)}$$

From equation (A.4) of appendix A we have

$$\int_{t_S}^{t_N} (1-t^2)\frac{dP_{nm}(t)}{dt}dt = \int_{t_S}^{t_N} (n+1)tP_{nm}(t)\,dt - \int_{t_S}^{t_N} (n-m+1)P_{n+1,m}(t)\,dt \quad . \qquad \text{(B.2)}$$

Let integrate by parts the left side by setting

$$u=1-t^2,\ du=-2tdt,\ dv=\frac{dP_{nm}(t)}{dt}dt,\ v=P_{nm}(t)\ . \qquad \text{(B.3)}$$

The result is

$$(1-t^2)P_{nm}(t)\Big|_{t_S}^{t_N} = (n-1)\ I\ tP_{nm}(t)\ dt - (n-m+1)\ I_{n+1,m} \quad . \tag{B.4}$$

From (A.6) we have

$$(2n+1)\int_{t_S}^{t_N} tP_{nm}(t)\ dt = (n-m+1)\ I_{n+1,m} + (n+m)\ I_{n-1,m} \tag{B.5}$$

where we have again used the definition (B.1). Replacing the integral in (B.4) by (B.5) gives

$$I_{n+1,m} = \frac{(n-1)(n+m)}{(n+2)(n-m+1)}\ I_{n-1,m} - \frac{(2n+1)(1-t^2)}{(n+2)(n-m+1)}P_{nm}(t)\Big|_{t_S}^{t_N} . \tag{B.6}$$

This is the final result which must however be normalized to not get large numbers unfitted for use in computers. Following (Heiskanen and Moritz, 1967, equation (1-73)) the fully normalized associated Legendre functions and their integrals are defined by

$$\overline{P}_{nm}(t) = H_{nm}\ P_{nm}(t),\quad \overline{I}_{nm}(t_S,t_N) = H_{nm}\ I_{nm}(t_S,t_N) \tag{B.7}$$

where

$$H_{nm} = \left(\frac{2(2n+1)(n-m)!}{(n+m)!}\right)^{1/2},\quad m\neq 0;\quad H_{n0} = (2n+1)^{1/2}. \tag{B.8}$$

Finally (7.2) is obtained by introducing (B.7) in (B.6).

APPENDIX C. Derivation of Equation (7.4).

The purpose of this appendix is not as much to show the derivation of equation (7.4) which solves (7.1) as to show a model to follow in deriving the more complicated equation (7.32).

Similar to (7.1), let define the non-normalized integral Inn

$$I_{nn} = I_{nn}(t_S, t_N) = \int_{t_S}^{t_N} P_{nn}(t)\ dt \quad . \tag{C.1}$$

From (A.3) we have

$$P_{nn}(t) = \frac{(2n)!}{2^n\ n!} (1-t^2)^{n/2} \quad . \tag{C.2}$$

As required in (C.1) let integrate by parts the right side of (B.2) by setting

$$u=(1-t^2)^{n/2},\ du=-nt(1-t^2)^{n/2-1}\ dt,\ dv=dt,\ v=t. \tag{C.3}$$

One obtains

$$\int_{t_S}^{t_N} (1-t^2)^{n/2}dt = t(1-t^2)^{n/2}\Big|_{t_S}^{t_N} + n\int_{t_S}^{t_N} t^2(1-t^2)^{n/2-1}\ dt \quad . \tag{C.4}$$

When one verifies that the last term can be written as

$$t^2(1-t^2)^{n/2-1} = (1-t^2)^{n/2-1} - (1-t^2)^{n/2} \qquad (C.5)$$

then (C.4) becomes

$$(n+1)\int_{t_S}^{t_N}(1-t^2)^{n/2}dt = t(1-t^2)^{n/2}\Big|_{t_S}^{t_N} + n\int_{t_S}^{t_N}(1-t^2)^{n/2-1}\,dt\ . \qquad (C.6)$$

Inserting (C.2) in (C.6) gives

$$\frac{(n+1)2^n n!}{(2n)!}\int_{t_S}^{t_N}P_{nn}dt = \frac{t2^n n!}{(2n)!}P_{nn}\Big|_{t_S}^{t_N} + n2^{n-2}\frac{(n-2)!}{(2n-4)!}\int_{t_S}^{t_N}P_{n-2n-2}dt \qquad (C.7)$$

which with (C.1) simplifies to the final relation

$$I_{nn} = \frac{1}{n+1}\left[\ tP_{nn}(t)\Big|_{t_S}^{t_N} + n(2n-1)(2n-3)\ I_{n-2,n-2}\right]\ . \qquad (C.8)$$

The desired normalized equation (7.4) is obtained after the insertion of (B.7) with (B.8) in (C.8).

APPENDIX D. Derivation of Equation (7.10).

The purpose of this appendix is not as much to show the derivation of equation (7.10) which solves (7.1) as to show a model to follow in deriving the more complicated equation (7.36).

The derivation of (7.10) is shown in (Paul, 1978), (Gerstl,1980) and (Gleason, 1983) and the idea is as followed.

From equation (A.7) with $u = n$ and $v = n-1$ and using (A.8) we have

$$P_{nn}(t) = (2n-1)\ y\ P_{nn-1,n-1}(t) \qquad (D.1)$$

where

$$y = (1-t^2)^{1/2}, \quad dy = -t/y\ dt \quad . \qquad (D.2)$$

We will already normalized (D.1) by inserting (B.7) with (B.8) in (D.1) to get

$$\overline{P}_{nn}(t) = b(n)\ y\ \overline{P}_{n-1,n-1}(t) \qquad (D.3)$$

where $b(n)$ is defined at (7.5). Inserting many times (D.3) into itself we get

$$P_{nn}(t) = b(n)b(n-1)b(n-2)\ldots b(1)\ y^n \quad . \qquad (D.4)$$

Following (C.1) the integration of (D.4) gives

$$I_{nn}(t_S,t_N) = b(n)b(n-1)\ldots b(1)\int_{t_S}^{t_N} y^n\, dt \tag{D.5}$$

and

$$I_{nn}(t_S,t_N) = -b(n)b(n-1)\ldots b(1)\int_{y_S}^{y_N} y^{n+1}(1-y^2)^{-1/2}dy \tag{D.6}$$

where we have used (D.2). The McLaurin's series of the last term is

$$(1-y^2)^{-1/2} = 1 + \frac{y^2}{2} + \frac{9y^4}{4!} + \ldots \quad . \tag{D.7}$$

Inserting (D.7) in (D.6) and integrating term by term one gets the final result

$$I_{nn}(t_S,t_N) = -b(n)b(n-1)\ldots b(1)\; y^{n+2}\Big[\frac{1}{n+2} + \frac{1}{2}\frac{y^2}{n+4} +$$

$$+ \frac{1}{2}\frac{3}{4}\frac{y^4}{n+6} + \ldots\Big]\Bigg|_{y_S}^{y_N} \quad . \tag{D.8}$$

APPENDIX E. The Spherical Harmonics Analysis using Fast Fourier Transform.

The purpose of this appendix is to describe a known application of the Fast Fourier Transform (FFT) technique to Spherical Harmonics Analysis (Colombo, 1981), (Gleason, 1985) to which we are required to refer in chapters 4 and 6. Here we apply the FFT to the single boundary value problem (b.v.p.) which can help understand the application of the FFT to the mixed b.v.p. in chapters 4 and 6.

In the single b.v.p. where a continuous set of point gravity anomalies is given everywhere on the surface of a unit sphere σ the corresponding set of fully normalized geopotential coefficients are given by

$$\begin{matrix}\overline{C}_{nm} \\ \overline{S}_{nm}\end{matrix} = \frac{1}{4\pi\,\gamma\,(n-1)} \iint_{\sigma} \Delta g(\theta,\lambda) \begin{matrix}\cos m\lambda \\ \sin m\lambda\end{matrix}\, \overline{P}_{nm}(\cos\theta)\; d\sigma. \quad (E.1)$$

We are always using the same polar spherical coordinates of chapter 2 and $\gamma = GM/R^2$ is the mean value of normal gravity associated with the reference ellipsoid employed.

In practice we are provided with a set of discrete mean gravity anomalies covering all the surface of the earth. Thus to compute (E.1) the sphere is partitioned into a finite number of discrete equiangular blocks of the size of the data available, here 1°X1° mean values of Δg. Thus we divide the spherical Earth into a regular grid of meridians and parallels which defines blocks σ_{ij}

$$\sigma_{ij} = \left| \begin{array}{l} \theta_i < \theta \leq \theta_{i+1} \\ \lambda_j < \lambda \leq \lambda_{j+1} \end{array} \right. \tag{E.2}$$

where

$$\begin{aligned} \theta_i &= i\Delta\theta \quad , \qquad & i = 0,1,2,\ldots,N-1 \\ \lambda_j &= j\Delta\lambda \quad , \qquad & j = 0,1,2,\ldots,2N-1 \ . \end{aligned} \tag{E.3}$$

N is equal to $\pi/\Delta\theta = \pi/\Delta\lambda$ and, as in (2.6), N is also the highest degree at which one can wish to compute a complete set of potential coefficients. The harmonic coefficients up to infinity, are all independent when one is dealing with the continuous case. It is said that the set of harmonic coefficients is complete. But in the discrete case like here we are dealing with N independent latitudes and thus the harmonic coefficients are said to be complete (i.e. linearly independent) only up to degree and order N. Trying to solve for more coefficients then up to degree N would result in getting a sampling error (Colombo, 1981, pp.11-13). Also in (E.3), $\Delta\theta$ and $\Delta\lambda$ define the dimension of the blocks in latitude and longitude respectively, and we will herein use blocks of size $\Delta\theta \times \Delta\lambda = 1°\mathrm{x}1°$. The area of a block on the unit sphere is

$$\Delta_{ij} = \iint_{\sigma_{ij}} d\sigma = \Delta\lambda \ [\ \cos(\theta_i) - \cos(\theta_{i+1})\ ] \quad . \tag{E.4}$$

For blocks of area Δ_{ij} we can define block mean values of gravity anomalies as $\overline{\Delta g}_{ij}(\theta,\lambda)$. If $\Delta g(\theta,\lambda)$ was constant within each block σ_{ij} then every point gravity anomaly $\Delta g(\theta,\lambda)$ inside the ijth block would equal its mean value $\overline{\Delta g}_{ij}(\theta,\lambda)$ and one could take $\overline{\Delta g}_{ij}(\theta,\lambda)$ out of the integral (E.1) as follows

$$\begin{matrix}\overline{C}_{nm}\\ \overline{S}_{nm}\end{matrix} = \frac{1}{4\pi\,\gamma\,(n-1)} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{\Delta g}_{ij} \iint_{\sigma_{ij}} \overline{P}_{nm}(\cos\theta) \begin{matrix}\cos m\lambda\\ \sin m\lambda\end{matrix} \sin\theta\, d\lambda\, d\theta \tag{E.5}$$

The integral (E.5) might become applicable in the future when the block size used will be smaller then we will use herein. However it is obvious that usually every point value $\Delta g(\theta,\lambda)$ in the ijth block is different then the mean value $\overline{\Delta g}_{ij}(\theta,\lambda)$ and thus this integral is not exact. Pellinen (1966) and Katsambalos (1979) has shown that for circular blocks of radius ψ_0 a smoothing operator β_n must be used to get a better approximation. Colombo (1981, p.76, eq.(3.9)) has shown that the de-smoothing operator η_n was more appropriated. Thus a better approximation is obtained by using instead of (E.5) the following

$$\begin{matrix}\overline{C}_{nm}\\ \overline{S}_{nm}\end{matrix} = \frac{1}{4\pi\gamma(n-1)\eta_n} \sum_{i=0}^{N-1} \sum_{j=0}^{2N-1} \overline{\Delta g}_{ij} \iint_{\sigma_{ij}} \overline{P}_{nm}(\cos\theta) \begin{matrix}\cos m\lambda\\ \sin m\lambda\end{matrix} \sin\theta\, d\lambda\, d\theta \tag{E.6}$$

where

$$\eta_n = \begin{cases} \beta_n^2 & \text{if } 0 \le n \le N/3 , \\ \beta_n & \text{if } N/3 < n \le N , \\ 1 & \text{if } n > N \end{cases}$$

and

$$\beta_n = \frac{1}{1-\cos\psi_0} \; \frac{1}{2n+1} \; [P_{n-1}(\cos Y \,) - P_{n+1}(\cos\psi_0)] \quad (E.7)$$

The Legendre polynomials in (E.7) are computed from the recurrence relation

$$P_n(\cos\psi_0) = \frac{2n-1}{n}\cos\psi_0 \; P_{n-1}(\cos\psi_0) - \frac{n-1}{n} \; P_{n-2}(\cos\psi_0) \quad (E.8)$$

The starting values for (E.8) are $P_0(\cos\psi_0) = 1$ and $P_1(\cos\psi_0) = \cos\psi_0$.

As we have said, equiangular square blocks are employed in practice and not circular blocks. Thus one must find the ψ_0 radius of the circular cap on the sphere whose area is approximately equal to the area of the equiangular square block σ_{ij} at the latitude θ_i.

Although the areas of the blocks will vary with latitude Katsambalos (1979) has shown that if $\Delta\theta = \Delta\lambda$ is in radians then one can use

$$\psi_0 = 2 \text{ ARCSIN } \left[\left(\frac{\Delta\theta \sin \Delta\theta}{4\pi} \right)^{1/2} \right] \quad (E.9)$$

on a global basis in (E.7) and (E.8) for most applications.

Equation (E.6) can be written as

$$\left.\begin{array}{c}\overline{C}_{nm}\\ \overline{S}_{nm}\end{array}\right\} = \frac{1}{4\pi\gamma(n-1)\eta_n} \sum_{i=0}^{N-1} \overline{I}^{\,i}_{nm}(\theta) \sum_{j=0}^{2N-1} \overline{\Delta g}_{ij} \left\{\begin{array}{c} J^{\,j}_m(\lambda)\\ K^{\,j}_m(\lambda)\end{array}\right. \tag{E.10}$$

where we have set

$$\overline{I}^{\,i}_{nm}(\theta) = \int_{\theta_i}^{\theta_{i+1}} \overline{P}_{nm}(\cos\theta)\ \sin\theta\ d\theta \quad . \tag{E.11}$$

The solution of (E.11) is given in chapter 7. In (E.10) we have also set

$$J^{\,j}_m(\lambda) = \int_{\lambda_j}^{\lambda_{j+1}} \cos m\lambda\ d\lambda = \frac{1}{m}\ (\sin m\lambda)\ \Bigg|_{j\Delta\lambda}^{j\Delta\lambda\ +\ \Delta\lambda}$$

$$K^{\,j}_m(\lambda) = \int_{\lambda_j}^{\lambda_{j+1}} \sin m\lambda\ d\lambda = \frac{1}{m}\ (-\cos m\lambda)\ \Bigg|_{j\Delta\lambda}^{j\Delta\lambda\ +\ \Delta\lambda} \quad . \tag{E.12}$$

Since we are using a regular grid of blocks of size $\Delta\lambda$, we have replace the integration limit λ_j by $j\Delta\lambda$ as defined in (E.3). The integration of (E.12) gives

$$J_m^j(\lambda) = \int_{\lambda_j}^{\lambda_{j+1}} \cos m\lambda \, d\lambda = A(m)\ \cos(mj\Delta\lambda) + B(m)\ \sin(mj\Delta\lambda)$$

$$K_m^j(\lambda) = \int_{\lambda_j}^{\lambda_{j+1}} \sin m\lambda \, d\lambda = -B(m)\ \cos(mj\Delta\lambda) + A(m)\ \sin(mj\Delta\lambda) \qquad \text{(E.13)}$$

where

$$A(m) = \begin{cases} \dfrac{\sin(m\Delta\lambda)}{m} & \text{if } m \neq 0 \\ \Delta\lambda & \text{if } m = 0 \end{cases}$$

$$B(m) = \begin{cases} \dfrac{\cos(m\Delta\lambda) - 1}{m} & \text{if } m \neq 0 \\ 0 & \text{if } m = 0 \end{cases} \qquad \text{(E.14)}$$

Inserting (E.13) in (E.10) results in

$$\begin{matrix} \overline{C}_{nm} \\ \overline{S}_{nm} \end{matrix} = \frac{1}{4\pi\gamma(n-1)\eta_n} \sum_{i=0}^{N-1} \overline{I}_{nm}^{\,i}(\theta) \left(\begin{matrix} A(m) \\ -B(m) \end{matrix} \sum_{j=0}^{2N-1} \overline{\Delta g_{ij}} \cos mj\Delta\lambda + \right.$$

$$\left. + \begin{matrix} B(m) \\ A(m) \end{matrix} \sum_{j=0}^{2N-1} \overline{\Delta g_{ij}} \sin mj\Delta\lambda \right) \qquad \text{(E.15)}$$

To evaluate this last equation the Fast Fourier transform has been proven to be very efficient (Colombo, 1981), (Goad and al., 1984).

By definition, the discrete complex Fourier transform sequence X of an input sequence y of P complex numbers is given (Gleason, 1985) and (I.M.S.L., routine FFTCC)

$$X(k) = \sum_{l=0}^{P-1} y(l)\; e^{i'\left(\frac{2\pi}{P}\, k\, l\right)} \qquad (k=0,1,2...,P-1)\quad . \quad (E.16)$$

Here $i' = (-1)^{1/2}$ and P is the number of given complex numbers in the sequence y to be transformed. In Fourier Analysis textbooks the index k is referred to as a Frequency Domain Counter. The value of $k = P/2$ is called the Nyquist Frequency. The Nyquist Frequency $P/2$ is the highest frequency counter that can be properly recovered from a given input sequence y of P complex number to be transformed.

From elementary complex variable theory it follows that for any complex number $z = x+i'y$

$$e^z = e^{x+i'y} = e^x\,(\cos y + i' \sin y) \quad . \qquad (E.17)$$

Substituting (E.17) in (E.16) with $x = 0$ yields

$$X(k) = \sum_{l=0}^{P-1} y(l)\; [\cos(\frac{2\pi}{P} k\, l) + i' \sin(\frac{2\pi}{P} k\, l)]\,.(E.18)$$

This can also be written

$$X(k) = \mathrm{REAL}[X(k)] + i' \, \mathrm{IMAG}[X(k)]$$

where

$$\mathrm{REAL}[X(k)] = \sum_{l=0}^{P-1} y(l) \cos\left(\frac{2\pi}{P} k \, l\right)$$

and (E.19)

$$\mathrm{IMAG}[X(k)] = \sum_{l=0}^{P-1} y(l) \sin\left(\frac{2\pi}{P} k \, l\right) \quad .$$

Comparing the elements in (E.19) and (E.15) we find the following equivalence

$$\begin{array}{lcl} P & \Leftrightarrow & 2N \\ y(l) & \Leftrightarrow & \Delta g_{ij} + 0 \, i' = \overline{\Delta g}_{ij}(j) + 0 \, i' \\ l & \Leftrightarrow & j \\ k & \Leftrightarrow & m \\ \frac{2\pi}{P} & \Leftrightarrow & \Delta\lambda \quad . \end{array}$$

We can thus write (E.15) as

$$\begin{array}{c} \overline{C}_{nm} \\ \overline{S}_{nm} \end{array} = \frac{1}{(n-1)\eta_n} \sum_{i=0}^{N-1} \overline{I}^{\,i}_{nm}(\theta) \left(\begin{array}{c} A(m) \\ -B(m) \end{array} \mathrm{REAL}[X^i(m)] + \begin{array}{c} B(m) \\ A(m) \end{array} \mathrm{IMAG}[X^i(m)] \right)$$

(E.20)

where

$$\mathrm{REAL}[X^i(m)] = \sum_{j=0}^{2N-1} y^i(j) \cos(mj\Delta\lambda)$$

(E.21)

$$\mathrm{IMAG}[X^i(m)] = \sum_{j=0}^{2N-1} y^i(j) \sin(mj\Delta\lambda)$$

in which

$$y^{i}(j) = \frac{1}{4\pi\gamma}\,\overline{\Delta g}_{ij} \quad . \tag{E.22}$$

The $REAL[.]$ and $IMAG[.]$ parts are obtained from the complex Fourier transform sequence $X^{i}(m)$, output of the routine FFTCC (IMSL Library of FORTRAN 77) after the input of the complex sequence of gravity anomalies $y^{i}(j)$ along a parallel (i). Since $2N$ Δg_{ij} are input, the output recovers $X^{i}(m)$ up to the Nyquist frequency here $m = N$. The subroutine FFTCNM in appendix G computes the algorithms contained in this appendix.

APPENDIX F. The Spherical Harmonics Synthesis Using Fast Fourier Transform.

To numerically test the solution of the mixed b.v.p. gravity anomaly and disturbing potential values covering globally the Earth are required. Such values can be computed from geopotential models given by spherical harmonic series. When such values are computed on a regular grid it is called synthesis and fast Fourier transform (FFT) is well suited to perform efficiently this task.

The purpose of this appendix is to describe this known application of the FFT technique to "spherical harmonics synthesis" (Colombo, 1981) and (Gleason, 1985) which we are required to refer in chapter 9.

Let us divide the spherical Earth into the same regular grid of equiangular blocks σ_{ij} as described in appendix E. From equation (3.7) in chapter 3 the fully normalized spherical harmonic representation of the "point" gravity anomaly for the ijth block is given by

$$\Delta g_{ij} = \frac{GM}{a^2} \sum_{n=2}^{\infty} (n-1) \left(\frac{a}{R}\right)^{n+2} \sum_{m=0}^{n} (\bar{C}^{*}_{nm} \cos m\lambda_j + \bar{S}_{nm} \sin m\lambda_j) \bar{P}_{nm}(\cos\theta_i) \tag{F.1}$$

where R, θ_i, λ_j are the polar spherical coordinates of the southwest corner of the block σ_{ij} and the gravity anomaly is given at the surface of the sphere i.e. $r = R$.

In (F.1) "a" is the semi-major axis of the reference ellipsoid employed. For the ijth "mean" gravity anomaly (F.1) becomes

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\,\Delta_{ij}} \sum_{n=2}^{\infty} (n-1)\left(\frac{a}{R}\right)^{n+2} \sum_{m=0}^{n} \int_{\theta_i}^{\theta_{i+1}} \int_{\lambda_j}^{\lambda_{j+1}} (\bar{C}^{*}_{nm}\cos m\lambda + \bar{S}_{nm}\sin m\lambda) \bullet$$

$$\bullet\ \bar{P}_{nm}(\cos\theta)\ \sin\theta\ d\theta\ d\lambda\ . \quad \text{(F.2)}$$

In (F.2) Δ_{ij} is the surface area of the ijth equiangular block as given by (E.4) and the term a/R is constant at the surface of the sphere. Making use of (E.11) and (E.12) one can write (F.2) as

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\,\Delta_{ij}} \sum_{n=2}^{\infty} (n-1)\left(\frac{a}{R}\right)^{n+2} \sum_{m=0}^{n} \bar{I}^{\,i}_{nm}(\theta)\,[\bar{C}^{*}_{nm}\,J^{j}_{m}(\lambda) + \bar{S}_{nm}\,K^{j}_{m}(\lambda)]\ . \quad \text{(F.3)}$$

One can verify that we can interchange the order of summation in (F.1) and (F.3). This and denoting as before the maximum degree and order attained as N yields

$$\Delta g_{ij} = \frac{GM}{a^2} \sum_{m=0}^{N} \left(\cos m\lambda_j \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \bar{C}^{*}_{nm}\,\bar{P}_{nm}(\cos\theta_i) + \right.$$

$$\left. + \sin m\lambda_j \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \bar{S}_{nm}\,\bar{P}_{nm}(\cos\theta_i) \right) \quad \text{(F.4)}$$

and

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\,\Delta_{ij}} \sum_{m=0}^{N} \left(J^{j}_{m}(\lambda) \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \bar{C}^{*}_{nm}\,\bar{I}^{\,i}_{nm}(\theta) + \right.$$

$$\left. + K^{j}_{m}(\lambda) \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \bar{S}_{nm}\,\bar{I}^{\,i}_{nm}(\theta) \right). \quad \text{(F.5)}$$

We can simplify the writing and the computations by defining the coefficients alpha

$$\overset{i}{\alpha}_m = \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \overline{C}^{*}_{nm}\ \overline{P}_{nm}(\cos\theta_i)$$

and beta (F.6)

$$\overset{i}{\beta}_m = \sum_{n=m}^{N} (n-1)\left(\frac{a}{R}\right)^{n+2} \overline{S}_{nm}\ \overline{P}_{nm}(\cos\theta_i) \quad .$$

These coefficients allow one to write (F.4) as

$$\Delta g_{ij} = \frac{GM}{a^2} \sum_{m=0}^{N} \left[\ \overset{i}{\alpha}_m \cos m\lambda_j + \overset{i}{\beta}_m \sin m\lambda_j\ \right] \quad . \qquad \text{(F.7)}$$

Similarly by letting

$$\overset{i}{\overline{\alpha}}_m = \sum_{n=m}^{N} (n-1) \left(\frac{a}{R}\right)^{n+2} \overline{C}^{*}_{nm}\ \overset{i}{\overline{I}}_{nm}(\theta)$$

and (F.8)

$$\overset{i}{\overline{\beta}}_m = \sum_{n=m}^{N} (n-1) \left(\frac{a}{R}\right)^{n+2} \overline{S}_{nm}\ \overset{i}{\overline{I}}_{nm}(\theta)$$

allows one to write (F.5) as

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\ \Delta_{ij}} \sum_{m=0}^{N} \left[\ \overset{i}{\overline{\alpha}}_m\ \overset{j}{J}_m(\lambda) + \overset{i}{\overline{\beta}}_m\ \overset{j}{K}_m(\lambda)\ \right] . \qquad \text{(F.9)}$$

Substituting equations (E.13) for $J_m^j(\lambda)$ and $K_m^j(\lambda)$ in (F.9) gives

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\ \Delta_{ij}} \sum_{m=0}^{N} \Bigg([\bar{\alpha}_m^i\ A(m) - \bar{\beta}_m^i\ B(m)]\ \cos mj\Delta\lambda\ + $$

$$+ \ [\bar{\alpha}_m^i\ B(m) - \bar{\beta}_m^i\ A(m)]\ \sin mj\Delta\lambda \Bigg). \qquad (F.10)$$

If we want to compute a set of NLON equally spaced gravity anomaly values going completely around a constant colatitude band "i" starting at the zero meridian, then it follows from elementary trigonometry that

$$\cos(m\lambda_j) = \cos(mj\Delta\lambda) = \cos\left(mj\frac{2\pi}{NLON}\right)$$

$$\sin(m\lambda_j) = \sin(mj\Delta\lambda) = \sin\left(mj\frac{2\pi}{NLON}\right) \qquad (F.11)$$

where

$$\Delta\lambda = \lambda_{j+1} - \lambda_j\ , \qquad j = 0, 1, 2, \ldots, NLON-1,$$

thus (F.12)

$$\Delta\lambda = \frac{2\pi}{NLON} \quad .$$

Inserting (F.11) in (F.7) and (F.10) gives

$$\Delta g_{ij} = \frac{GM}{a^2} \sum_{m=0}^{N} \left(\alpha_m^i \cos\left(mj\frac{2\pi}{NLON}\right) + \beta_m^i \sin\left(mj\frac{2\pi}{NLON}\right) \right) \qquad (F.13)$$

and

$$\overline{\Delta g}_{ij} = \frac{GM}{a^2\ \Delta_{ij}} \sum_{m=0}^{N} \left([\overline{\alpha}_m^{\,i}\ A(m) - \overline{\beta}_m^{\,i}\ B(m)]\ \cos\left(mj\frac{2\pi}{NLON}\right) + \right.$$

$$\left. + \ [\overline{\alpha}_m^{\,i}\ B(m) - \overline{\beta}_m^{\,i}\ A(m)]\ \sin\left(mj\frac{2\pi}{NLON}\right) \right) \quad . \qquad (F.14)$$

(F.13) can be written using complex numbers with $i' = (-1)^{1/2}$ as

$$\Delta g_{ij} = \frac{GM}{a^2}\ \mathrm{REAL}\left(\sum_{m=0}^{N} (\alpha_m^{\,i} - i'\ \beta_m^{\,i})\ [\cos\left(mj\frac{2\pi}{NLON}\right) + i'\ \sin\left(mj\frac{2\pi}{NLON}\right)\ ] \right) \qquad (F.15)$$

or simply

$$\Delta g_{ij} = \frac{GM}{a^2}\ \mathrm{REAL}\left(\sum_{m=0}^{N} (\alpha_m^{\,i} - i'\ \beta_m^{\,i})\ e^{\,i'mj\frac{2\pi}{NLON}} \right) \quad . \qquad (F.16)$$

To use the fast Fourier tansform we can compare (F.16) with (E.16) and one finds out that instead of a summation up to N we must have a summation up to NLON−1 i.e.

$$\Delta g_{ij} = \frac{GM}{a^2}\ \mathrm{REAL}\left(\sum_{m=0}^{NLON-1} C^{\,i}(m)\ e^{\,i'mj\frac{2\pi}{NLON}} \right) \quad . \qquad (F.17)$$

For (F.17) to be equivalent to (F.16) one can verify that when $2N = NLON$ the coefficients $C^i(m)$ are related to the coefficients $\alpha_m{}^i$ and $\beta_m{}^i$ by

$$
\begin{aligned}
C^i(0) &= \alpha_0^i && \text{for } m = 0, \\
C^i(m) &= \frac{1}{2}(\alpha_m^i - i'\,\beta_m^i) && \text{for } m = 1, 2, \ldots, N-1, \\
C^i(N) &= \alpha_N^i - i'\,\beta_N^i && \text{for } m = N.
\end{aligned}
\tag{F.18}
$$

and also

$$
C^i(m) = \frac{1}{2}(\alpha_m^i + i'\,\beta_m^i) \qquad \text{for } m = N+1, N+2, \ldots, NLON-1. \tag{F.19}
$$

This is the case where we compute a grid of values say at $1°X1°$ spacing from an harmonic expansion up to degree 180, then $N = 180$ and $2N = NLON = 360$.

When $2N < NLON$, this is the case where we compute a grid say at $1°X1°$ spacing, $NLON = 360$, from an expansion up to degree 36, $N = 36$, then (F.18) is still valid but we must also have

$$
C^i(m) = 0 + i'\,0 = 0 \qquad \text{for } m = N+1, N+2, \ldots, NLON-N
$$

and (F.20)

$$
C^i(m) = \frac{1}{2}(\alpha_{NLON-m}^i + i'\,\beta_{NLON-m}^i)
$$

for $m = NLON-(N-1),\ NLON-(N-2), \ldots,\ NLON-2,\ NLON-1$.

When $2N > NLON$, this is the case where we compute a grid say at 5°X5° spacing, $NLON = 72$, from an expansion up to degree 180, $N = 180$, then the coefficients alpha and beta must be aliased (Colombo, 1981, p.10 and p.106). For this example where we want to compute a grid at 5°X5° from a set of coefficient up to degree 180, the 180 coefficients alpha and beta must be aliased i.e. reduced in quantity and merged into 36 coefficients ($180°/\Delta\theta° = 180°/5°$ according to the rule of the Nyquist frequency). When the coefficients have been aliased to degree N, where now $2N = NLON$, then the relations (F.18) and (F.19) can be used to find the coefficients $C^i(m)$.

It was numerically verified that aliasing the coefficients alpha and beta (same as aliasing the frequencies) from degree 180 to degree 36 and computing a grid of 5°X5° mean anomalies and on the other hand, computing a grid of 1°X1° mean anomalies from the set of degree 180 and then taking the average of the 1°X1° mean anomalies to obtain 5°X5° mean anomalies, we obtained the same mean values. The "aliased coefficients" alpha hat and beta hat are computed as

$$\hat{\alpha}^i_m = \alpha^i_m + \sum_{i=0}^{M} \alpha^i_{m+iNLON} + \alpha^i_{iNLON-m}$$

$$\hat{\beta}^i_m = \beta^i_m + \sum_{i=0}^{M} \beta^i_{m+iNLON} + \beta^i_{iNLON-m} \qquad (F.21)$$

where $m = 0, 1, \ldots, NLON/2$ and M is a large enough integer like "N". Here the coefficients alpha and beta without hat are defined by (F.6) i.e. to compute point values.

All what has been said from (F.15) to here is also applicable to compute "mean" gravity anomalies. Comparing (F.14) to (F.13), (F.13) would provide us with mean values if its alpha and beta coefficients would be defined as

$$\alpha_m^i = \frac{1}{\Delta_{ij}} \left[\bar{\alpha}_m^i \, A(m) - \bar{\beta}_m^i \, B(m) \right]$$
$$\beta_m^i = \frac{1}{\Delta_{ij}} \left[\bar{\alpha}_m^i \, B(m) - \bar{\beta}_m^i \, A(m) \right] \quad . \tag{F.22}$$

The coefficients alpha and beta with bar are defined by (F.8). Then all the relations (F.15) to (F.21) are still valid but to compute mean values. Hence (F.14) for mean values and (F.13) for point values are computed in a very similar way using fast Fourier transform.

Having computed the sequence $C^i(m)$ containing complex numbers for point or mean values this sequence for a colatitude band "i" is entered in the IMSL routine FFTCC which according to equation (E.16) returns the discrete Fourier transform $X^i(j)$, a vector of complex numbers. The real part of it, REAL[.], contains according to (F.17) the NLON gridded gravity anomaly values desired along the colatitude band "i". (F.17) is computed for each colatitude band i, i = 1,2,...,NLON/2.

Because of the relation between the associated Legendre functions in the northern and southern hemispheres, equations (7.40), or (7.41) for their integrals, the computations are carried out with both hemispheres at the same time, for efficiency. The subroutine FFTDGN in appendix G computes the theory contained in this appendix.

This subroutine computes mean and point values. By convention, the grid employed when the input geopotential coefficients were generated, starts at the zero meridian. To compute point gravity anomalies at the "center" of each square, instead of the southwest corner as was derived in this appendix, the reference grid must be rotated by $\Delta\lambda/2$ eastward from the zero meridian. Colombo (1981, p.106) shows that this can be accomplished by modifying the input coefficients as follow

$$\begin{aligned} \hat{C}^{*}_{nm} &= \bar{C}^{*}_{nm} \cos\frac{m\Delta\lambda}{2} + \bar{S}_{nm} \sin\frac{m\Delta\lambda}{2} \\ \hat{S}_{nm} &= \bar{S}_{nm} \cos\frac{m\Delta\lambda}{2} - \bar{C}^{*}_{nm} \sin\frac{m\Delta\lambda}{2} \quad . \end{aligned} \tag{F.23}$$

This rotation is accomplished in this routine FFTDGN of appendix G. Should point anomalies be desired at the grid intersections instead of the center of the squares then the input coefficients should remain unchanged.

APPENDIX G. Listing of Computer Routines.

```
      PROGRAM PNMI
C
C
C     PPPPPPPPP   NN      NN   MM      MM  IIIIIIIIII
C     PPPPPPPPPP  NNN     NN   MMMM  MMMM  IIIIIIIIII
C     PP      PP  NN N    NN   MM MMMM MM      II
C     PP      PP  NN NN   NN   MM  MM  MM      II
C     PPPPPPPPPP  NN  NN  NN   MM  MM  MM      II
C     PPPPPPPPP   NN  NN  NN   MM      MM      II
C     PP          NN   NN NN   MM      MM      II
C     PP          NN    N NN   MM      MM      II
C     PP          NN     NNN   MM      MM  IIIIIIIIII
C     PP          NN      NN   MM      MM  IIIIIIIIII
C
C
C******* THIS IS A DRIVER TO RUN PNMIS SUBROUTINE.
C******* ITS COMPUTES EFFICIENTLY THE ASSOCIATED LEGENDRE
C******* FUNCTIONS (PNM) OR THEIR INTEGRALS (PINM) OVER THE
C******* NORTHERN HEMISPHERE (AND SOUTHERN IF DESIRED).
C*******
C******* A BACKWARD RECURRENCE IS USED TO COMPUTE THE INTEGRALS
C******* OF THE SECTORIALS EXCEPT NEAR THE EQUATOR WHERE THE FORWARD
C******* RECURRENCE IS USED.
C
C******* IPNM   = 1.OR.0.  TO COMPUTE  PNM.OR.PINM
C******* IGEOD  = 1.OR.0.  TO COMPUTE AT GEODETIC.OR.GEOCENTRIC LAT.
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED FOR PNM OR PNMI
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. WANTED
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NBAND2 = NBR. OF BANDS FROM NORTH POLE TO SOUTH POLE
C
      PARAMETER(IPNM=0,IGEOD=0,NMAX=28,BLKSIZ=1.0E0)
      PARAMETER (NBANDS=90/BLKSIZ+1.E-7,NBAND2=2*NBANDS)
C
      OPEN(6,FILE='OUTPUTPNMI')
      OPEN(11,FILE='OUTPNMI')
C
C******* ASSIGN PRESCRIBED ACCURACY
C******* HOWEVER THE USE OF REAL*8 PERMITS AN ACCURACY OF 1.E-7 ONLY
C
      EPSLON=1.E-20
C
C         THS AND THN ARE THE SOUTHERN AND NORTHERN COLATITUDE.
C         WHEN THS=90 DEG. THE FORWARD RECURRENCE IS ALWAYS USED
C         WHEN THN=0 DEG. THE BACKWARD RECURRENCE IS ALWAYS USED
C
      INIT=0
      PI1=ACOS(-1.E0)
      RDDG=PI1/180.E0
C
C******* DETERMINE THE MAX NBR OF TERMS REQUIRED FOR THE MOST
C******* SOUTHERN BAND THAT NEEDS THE BACKWARD SECTORIAL RECURRENCE
C******* USING EQUATIONS (7.11) AND (7.12).
C******* HERE IT ONLY CHECKS IF SOME DIMENSIONS HAVE TO BE INCREASED.
C
      IF(IPNM.EQ.0)THEN
      THS=90.E0-BLKSIZ
      THN=THS-BLKSIZ
C
      DO 5 I=2,NBANDS
      S2=SIN(THN*RDDG)**2
      IF(ABS(S2).LT.1.E-5)GOTO 3
      RK=FLOAT(NMAX)/(FLOAT(NMAX+1)*S2)
      IF(RK.LE.1.E0)GOTO 4
    3 THM=.5E0*(THN+THS)*RDDG
      RN2=1.E0+ALOG(EPSLON)
      RD2=ALOG(SIN(THM)**2)
      ARG=RN2/RD2
      NTERMS=1+INT(ARG)
      WRITE(6,12)NTERMS
C     IF(NTERMS.GT.2870)STOP
      IF(NTERMS.GT.7296)STOP
      GOTO 6
    4 CONTINUE
      THS=THN
      THN=THN-BLKSIZ
    5 CONTINUE
C
    6 THN=0.E0
C
      ELSE IF(IPNM.EQ.1)THEN
      THN=-BLKSIZ/2
      ENDIF
C
C******* SET ** DO 10 I=1,NBAND2 ** TO SEE THE RESULTS
C******* IN THE SOUTHERN HEMISPHERE.  ONE FINDS OUT THAT
C******* PNM(-LAT)=-PNM(LAT)  IF N+M IS ODD
C******* PNM(-LAT)= PNM(LAT)  IF N+M IS EVEN
C******* AND A SIMILAR RELATION FOR THE INTEGRALS OF PNM:  I(N,M)
C******* INM(LAT1,LAT2)=-INM(LAT2,LAT1)  IF N+M IS ODD
C******* INM(LAT1,LAT2)= INM(LAT2,LAT1)  IF N+M IS EVEN
C
      THS=THN+BLKSIZ
C
C
C     DO 10 I=1,NBAND2
      DO 10 I=1,NBANDS
      WRITE(6,13)SECOND()
      WRITE(6,14)THN,THS
C
      CALL PNMIS (IPNM,IGEOD,NMAX,THN,THS,RN2,NTERMS,INIT)
C
C******* THE RESULTS ARE PRINTED OUT IN SUBROUTINE PNMIS
C
      THN=THS
      THS=THS+BLKSIZ
   10 CONTINUE
C
      WRITE(6,13)SECOND()
   12 FORMAT('    NBR OF TERMS SHOULD NOT EXCEED 7296.  NTERMS =',I10)
```

```
 13    FORMAT(5X,'CPU TIME =',F10.5)
 14    FORMAT(30X,'THN =',F10.5,'      THS =',F10.5)
C
       STOP
       END
       SUBROUTINE PNMIS(IPNM,IGEOD,NMAX,THDEGN,THDEGS,RN2,NTMAX,INIT)
C
C******* ITS COMPUTES EFFICIENTLY THE ASSOCIATED LEGENDRE
C******* FUNCTIONS (PNM) OR THEIR INTEGRALS (PINM) OVER THE
C******* NORTHERN HEMISPHERE.
C*******
C******* A BACKWARD RECURRENCE IS USED TO COMPUTE THE INTEGRALS
C******* OF THE SECTORIALS EXCEPT NEAR THE EQUATOR WHERE THE FORWARD
C******* RECURRENCE IS USED.
C
C      IPNM   : 1.OR.0, TO COMPUTE  PNM.OR.PINM
C      IGEOD  : 1.OR.0, TO COMPUTE AT GEODETIC.OR.GEOCENTRIC LAT.
C      NMAX   : MAXIMUM DEGREE AND ORDER TO COMPUTE PNM OR INM
C      THDEGS : GEOD. POLAR ANGLE IN DEG. OF LOWER LIMIT OF INTEGRAL
C      THDEGN : GEOD. POLAR ANGLE IN DEG. OF UPPER LIMIT OF INTEGRAL
C      P      : THE MATRIX THAT WILL CONTAIN THE PNM VALUES
C      PI     : THE MATRIX THAT WILL CONTAIN THE INM VALUES
C      RN2    : 1.E0 + ALOG(EPSLON)
C      NTMAX  : NUMBER OF TERMS REQUIRED IN MOST SOUTHERN BANO THAT
C               NEEDS BACKWARD SECTORIAL SCHEME
C      INIT   : MUST BE INITIALIZED AT ZERO (=0) IN MAIN PROGRAM
C
C      --- IN COMMENT STATEMENTS, ALL EQUATION NUMBERS ARE REFERENCED
C      FROM DISSERTATION ---
C
       PARAMETER(NNN=28,NNN2=NNN*2+1,NCOEFF=(NNN+1)*(NNN+2)/2)
C
C******* THE ARRAYS SER3 AND SER4 ARE DIMENSIONED TO ACCOMODATE
C******* A PRESCRIBED ACCURACY OF 1.E-20, IF MODIFIED CHANGE ALSO
C******* THE DIMENSIONS IN SUBROUTINE SERIES.
C******* HOWEVER THE USE OF REAL*8 PERMITS AN ACCURACY OF 1.E-7 ONLY.
C
C******* A(N,M) IS STORED IN A(N,M)   I.E. LOWER TRIANG. PART OF A=PI
C******* I(N,M) IS STORED IN PI(M,N)  I.E. UPPER TRIANG. PART OF PI=A
C******* P(N,M) OF TS IS IN P(N,M)    I.E. LOWER TRIANG. PART OF P
C******* P(N,N) OF TS IS IN P(N,N)    I.E. DIAGONAL PART OF P
C******* P(N,M) OF TN IS IN P(M,N)    I.E. UPPER TRIANG. PART OF P
C******* P(N,N) OF TN IS IN PN(N)
C
C      DIMENSION SER3(2870),SER4(2870),DIV(NNN),SQROOT(NNN2)
       DIMENSION SER3(7296),SER4(7296),DIV(NNN),SQROOT(NNN2)
C      DIMENSION SER3(12066),SER4(12066),DIV(NNN),SQROOT(NNN2)
       DIMENSION  A(0:NNN,0:NNN),B(NNN)
       DIMENSION PI(0:NNN,0:NNN)
       DIMENSION  P(0:NNN,0:NNN),PN(0:NNN)
       DIMENSION OUT(NCOEFF),IDIAG(0:NNN)
       EQUIVALENCE (A(0,0),PI(0,0))
C
C      --- E2 OF GRS80 ---
C
       DATA E2 /.0066943800229E0/
C
       IF(NMAX.NE.NNN)THEN
       WRITE(6,1000)
1000   FORMAT('    CHECK INPUT PARAMETER')
       STOP
       ENDIF
C
       NMAXM1=NMAX-1
       NMAXM2=NMAX-2
       NMAXP1=NMAX+1
       NMAXP2=NMAX+2
       IF(INIT.NE.0)GOTO 50
       INIT=1
C
C******* LOAD ARRAYS THAT WILL BE USED REPEATEDLY
C
       SQROOT(1)=1.E0
       DO 1 I=2,NNN2
       D1=I
 1     SQROOT(I)=SQRT(D1)
C
C      --- COMPUTE B(N) FROM EQUATION (7.5) ---
C      --- SER1 CONTAINS B(NMAX)B(NMAX-1)...B(1) OF (7.10) ---
C      --- SER2 CONTAINS B(NMAX-1)B(NMAX-2)...B(1) OF (7.10) ---
C
       B(1)=SQROOT(3)
       SER1=B(1)
       DO 2 N=2,NMAX
       N2=2*N
       B(N)=SQROOT(N2+1)/SQROOT(N2)
 2     SER1=SER1*B(N)
       SER2=SER1/B(NMAX)
C
C      --- COMPUTE A(N,M) FROM EQUATION (7.3) ---
C
       DO 3 N=1,NMAX
       N2=2*N
       DUM=SQROOT(N2+1)*SQROOT(N2-1)
       DO 3 M=0,(N-1)
 3     A(N,M)=DUM/(SQROOT(N+M)*SQROOT(N-M))
C
C      --- DIV AND IDIAG WILL BE USED TO SIMPLIFY COMPUTATIONS ---
C
       DO 4 N=0,NMAX
 4     IDIAG(N)=(N*N+N)/2+1
C
       DO 5 N=1,NMAX
       N1=N+1
 5     DIV(N)=1.E0/N1
C
C      --- SER3 AND SER4 HELP BUILD THE 3RD FACTOR OF EQ. (7.10)
C      I(NMAX,NMAX) AND I(NMAX-1,NMAX-1) ARE THE INITIAL CONDITIONS
C      FOR THE BACKWARD  RECURRENCES ---
C
```

```
      D1=NMAX+2
      D2=NMAXM1+2
      SER3(1)=1.E0/D1
      SER4(1)=1.E0/D2
      FRACT=1.E0
      D1=1.E0
      D2=2.E0
      D3=NMAX+4
      D4=NMAXM1+4
      DO 6 J=2,NTMAX
      FRACT=FRACT*D1/D2
      SER3(J)=FRACT/D3
      SER4(J)=FRACT/D4
      D1=D1+2.E0
      D2=D2+2.E0
      D3=D3+2.E0
      D4=D4+2.E0
    6 CONTINUE
C
      PI1=ACOS(-1.E0)
      RDDG=PI1/180.E0
      PI2=PI1/2.E0
C
C******* CONVERT GEODETIC COLATITUDES TO GEOCENTRIC COLATITUDES
C******* PHI = GEODETIC LATITUDE, PSI = GEOCENTRIC LATITUDE
C
   50 IF(IGEOD.EQ.1)THEN
      PHIS=(90.E0-THDEGS)*RDDG
      PHIN=(90.E0-THDEGN)*RDDG
      IF(ABS(PHIS+PI2).LT.1.E-5)THEN
      PSIS=-PI2
      ELSE
      PSIS=ATAN((1.E0-E2)*TAN(PHIS))
      ENDIF
      IF(ABS(PHIN-PI2).LT.1.E-5)THEN
      PSIN=PI2
      ELSE
      PSIN=ATAN((1.E0-E2)*TAN(PHIN))
      ENDIF
      THS=PI2-PSIS
      THN=PI2-PSIN
C
      ELSE IF(IGEOD.EQ.0)THEN
      THS=THDEGS*RDDG
      THN=THDEGN*RDDG
      ENDIF
C
      TS=COS(THS)
      YS=SIN(THS)
      TN=COS(THN)
      YN=SIN(THN)
      YS2=YS*YS
      YN2=YN*YN
C
      IF(IPNM.EQ.1)GOTO 70
C
C******* COMPUTE THE CURRENT CONDITION NUMBER RK FROM (7.13)
C
      IF(ABS(YS-1.E0).LT.1.E-5)THEN
      RK=.5E0
      GOTO 70
      ENDIF
      IF(ABS(YN).LT.1.E-5)THEN
      RK=1.5E0
      GOTO 60
      ENDIF
      RK=FLOAT(NMAX)/(FLOAT(NMAXP1)*YN2)
      IF(RK.LE.1.E0)GOTO 70
C
C******* COMPUTE NUMBER OF TERMS NEEDED FOR CURRENT TASK
C
   60 THETAM=.5E0*(THDEGS+THDEGN)*RDDG
      RD2=ALOG(SIN(THETAM)**2)
      ARG=RN2/RD2
      NTERMS=1+INT(ARG)
      WRITE(6,101)NTERMS
  101 FORMAT(40X,'BY BACKWARD RECURRENCE, NTERMS =',I10)
C
C     --- USE EQUATION (7.10) TO DETERMINE I(NMAX,NMAX) AND
C     I(NMAX-1,NMAX-1) ---
C
      CALL SERIES (YS2,SER5,SER3,NTERMS)
      CALL SERIES (YN2,SER6,SER3,NTERMS)
      CALL SERIES (YS2,SER7,SER4,NTERMS)
      CALL SERIES (YN2,SER8,SER4,NTERMS)
C
      PI(NMAX,NMAX)=SER1*(YS**NMAXP2*SER5-YN**NMAXP2*SER6)
      PI(NMAXM1,NMAXM1)=SER2*(YS**NMAXP1*SER7-YN**NMAXP1*SER8)
C
      IF(THDEGN.GT.90.E0)THEN
      PI(NMAX,NMAX)=-PI(NMAX,NMAX)
      PI(NMAXM1,NMAXM1)=-PI(NMAXM1,NMAXM1)
      ENDIF
C
C
C     --- INITIALIZE P(N,M) AND I(N,M) FROM (0,0) THRU (1,1)
C     EQUATION (7.8) ---
C
   70 P(0,0)=1.E0
      P(1,0)=B(1)*TS
      P(1,1)=B(1)*YS
      PN(0)=1.E0
      PN(1)=B(1)*YN
      P(0,1)=B(1)*TN
      PI(0,0)=TN-TS
      PI(0,1)=B(1)*(TN*TN-TS*TS)*.5E0
      PI(1,1)=B(1)*(TN*YN+THS-(TS*YS+THN))*.5E0
C
C
C     --- COMPUTE FORWARD OR BACKWARD ---
```

```
C
      DO 100 N=2,NMAX
      N1=N-1
C
C     --- COMPUTE (7.4) OR (7.9) USING (7.7) ---
C
      P(N,N)=B(N)*YS*P(N1,N1)
  100 PN(N)=B(N)*YN*PN(N1)
C
      IF(IPNM.EQ.1)GOTO 350
C
      IF(RK.LE.1.E0)GOTO 250
      DO 200 N=NMAXM2,0,-1
      N2=N+2
  200 PI(N,N)=(PI(N2,N2)*(N+3)+TS*P(N2,N2)-TN*PN(N2))/(N2*B(N2)*
     1        B(N+1))
      GOTO 350
C
  250 DO 300 N=2,NMAX
  300 PI(N,N)=(N*B(N)*B(N-1)*PI(N-2,N-2)+TN*PN(N)-TS*P(N,N))*DIV(N)
C
  350 DO 500 N=2,NMAX
      N1=N-1
      N2=N-2
      DO 400 M=0,(N-3)
C
C     --- COMPUTE (7.2) USING (7.6) ---
C
      P(N,M)=(TS*P(N1,M)-P(N2,M)/A(N1,M))*A(N,M)
      P(M,N)=(TN*P(M,N1)-P(M,N2)/A(N1,M))*A(N,M)
  400 PI(M,N)=((N2)*PI(M,N2)/A(N1,M)+YS2*P(N1,M)-
     1        YN2*P(M,N1))*A(N,M)*DIV(N)
      P(N,N2)=(TS*P(N1,N2)-P(N2,N2)/A(N1,N2))*A(N,N2)
      P(N2,N)=(TN*P(N2,N1)-PN(N2)/A(N1,N2))*A(N,N2)
      PI(N2,N)=((N2)*PI(N2,N2)/A(N1,N2)+YS2*P(N1,N2)-
     1          YN2*P(N2,N1))*A(N,N2)*DIV(N)
      P(N,N1)=TS*P(N1,N1)*A(N,N1)
      P(N1,N)=TN*PN(N1)*A(N,N1)
  500 PI(N1,N)=(YS2*P(N1,N1)-YN2*PN(N1))*A(N,N1)*DIV(N)
C
      IF(IPNM.EQ.1)THEN
      DO 510 N=0,NMAX
      DO 510 M=0,N
  510 OUT(IDIAG(N)+M)=P(N,M)
      WRITE(6,15)NMAX,NMAX,P(NMAX,NMAX)
C
      ELSE IF(IPNM.EQ.0)THEN
      DO 520 N=0,NMAX
      DO 520 M=0,N
  520 OUT(IDIAG(N)+M)=PI(M,N)
      WRITE(6,15)NMAX,NMAX,PI(NMAX,NMAX)
      ENDIF
C
      WRITE (11) OUT
C
C     I=NMAX
C     IF(NMAX.GT.4)I=4
C     WRITE(6,15)((N,M,P(N,M),M=0,N),N=0,I)
C     WRITE(6,102)
C     WRITE(6,15)((N,M,P(M,N),M=0,N),N=0,I)
C     WRITE(6,102)
C     WRITE(6,15)(N,N,PN(N),N=0,I)
C     WRITE(6,102)
C     WRITE(6,15)((N,M,PI(M,N),M=0,N),N=0,I)
C     WRITE(6,102)
C     IF(NMAX.GT.4)WRITE(6,15)(NMAX,M,PI(M,NMAX),M=0,NMAX)
C
   15 FORMAT(2I5,E27.16)
  102 FORMAT(//)
C
      RETURN
      END
      SUBROUTINE SERIES (Y2,SER,SERI,NTERMS)
C
C     --- THIS SUBROUTINE DETERMINES THE 3RD FACTOR OF EQ. (7.10)
C     OR (7.37) FOR I(NMAX,NMAX) AND I(NMAX-1,NMAX-1) AT YS AND YN
C     OR I(NMAX,NMAX,NMAX,NMAX), I(NMAX-1,NMAX-1,NMAX-1,NMAX-1) AND
C     I(NMAX,NMAX,NMAX-1,NMAX-1) AT YS AND YN ---
C
      DIMENSION SERI(7296)
      SER=SERI(1)
      IF(ABS(Y2).LT.1.E-5)RETURN
      YY=Y2
      DO 10 I=2,NTERMS
      SER=SER+SERI(I)*YY
      YY=YY*Y2
   10 CONTINUE
      RETURN
      END
      PROGRAM FFTDGN
C
C
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT DDDDDDDDD   GGGGGGGG  NN      NN
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT DDDDDDDDDD GGGGGGGGGG NNN     NN
C     FF         FF             TT     DD      DD GG        G NN N    NN
C     FF         FF             TT     DD      DD GG          NN NN   NN
C     FFFFFF     FFFFFF         TT     DD      DD GG  GGGGG  NN  NN  NN
C     FFFFFF     FFFFFF         TT     DD      DD GG  GGGGGG NN  NN  NN
C     FF         FF             TT     DD      DD GG      GG NN   NN NN
C     FF         FF             TT     DD      DD GG      GG NN    N NN
C     FF         FF             TT     DDDDDDDDDD GGGGGGGGGG NN     NNN
C     FF         FF             TT     DDDDDDDDD   GGGGGGGG  NN      NN
C
C
C     THIS PROGRAM COMPUTES A GLOBAL SET OF CENTER POINT OR MEAN
C     BLKSIZ BY BLKSIZ (=1 DEG X 1 DEG) GRAVITY ANOMALIES OR
C     GEOID UNDULATIONS FROM A GIVEN SET OF CNM & SNM GEOPOTENTIAL
C     COEFFICIENTS:
C
C     - FOR MEAN VALUES:
```

```
C
C                                     N+2
C            GM        NMAX       A        N        INTEGRAL OF
C     DG = --------- SUM (N-1) -      SUM        (CNM COS+SNM SIN) PNM
C          A**2 AREA N=2          R       M=0
C
C
C                                     N+1
C            GM            NMAX   A        N        INTEGRAL OF
C     N  = ------------ SUM   -      SUM        (CNM COS+SNM SIN) PNM
C            A  G  AREA  N=2    R       M=0
C
C
C     IF IELL.EQ.0,  THEN COMPUTE ON THE SPHERE WITH R=A.
C     IF IHARM.EQ.1, THEN HARMONIC COEFFICIENTS ARE INPUT, AND:
C
C                   1    NMAX  N
C     OUT = -------- SUM  SUM  INTEGRAL OF  (CNM COS+SNM SIN) PNM
C               AREA  N=0  M=0
C
C
C     - FOR POINT VALUES:  AREA = 1 AND THERE IS NO INTEGRATION
C
C     - FAST FOURIER TRANSFORM IS USED THROUGH THE IMSL SUBR. FFTCC
C
C     - ASS LEG FUNCT OR THEIR INTEGRATION COME FROM PROGRAM PNMI
C
C     COMPUTATIONS WITH 8 BYTES IS EQUIVALENT TO RETAINING THE FIRST
C     SEVEN (7) SIGNIFICANT FIGURES IN ALL ARITHMETIC OPERATIONS.
C
C******* NMAX   = MAXIMUM DEGREE AND ORDER OF INPUT COEFFICIENTS
C******* NCOEFF = NUMBER OF CNM OR SNM COEFFICIENTS
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. WANTED
C******* NLON   = NBR. OF BLKSIZ SQUARES AROUND EACH BAND WANTED
C******* NLAT   = 180/BLKSIZ = NYQUIST FREQUENCY
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NSQ    = GLOBAL NUMBER OF BLKSIZ X BLKSIZ SQUARES
C******* NWK    = NEEDED DIMENSION FOR IMSL FFT ARRAYS
C******* IHARM  = 1.OR.0, FOR HARMONIC.OR.GEOPOTENTIAL COEFF INPUT
C******* IMEAN  = 1.OR.0, TO COMPUTE MEAN.OR.POINT VALUES
C******* IANOM  = 1.OR.0, TO COMPUTE ANOMALIES.OR.UNDULATIONS
C******* IJ2J4  = 1.OR.0, TO SUBSTRACT OR NOT J2, J4 & J6.
C******* IELL   = 1.OR.0, TO COMPUTE ON THE ELLIPSOID OR THE SPHERE
C
      PARAMETER (NMAX=28,BLKSIZ=1.0E0)
      PARAMETER (NLON=360/BLKSIZ+1.E-7,NLAT=NLON/2)
      PARAMETER (NMAXP1=NMAX+1,NMAXP2=NMAX+2,NCOEFF=NMAXP1*NMAXP2/2)
      PARAMETER (NBANDS=NLAT/2,NSQ=2*NLAT*NLAT,NWK=6*NLON+150)
      PARAMETER (IHARM=0,IJ2J4=1,IMEAN=1,IELL=0,IANOM=0)
C
C******* OUT     = STORES THE OUTPUT OF ANOMALIES OR UNDULATIONS
C******* CNM,SNM = STORE THE INPUT SET OF COEFFICIENTS
C******* PNMI    = STORES THE ASS. LEG. FUNCT. OR THEIR INTEGRATION
C******* IDIAG   = IS A LOCATING ARRAY FOR CNM, SNM AND PNMI
C******* IWK,WK  = ARE REQUIRED BY THE IMSL FFT SUBROUTINE
C******* AM,BM   = STORE SINES AND COSINES REQUIRED
C******* A1,A2   = STORE INPUT.AND.OUTPUT OF THE IMSL FFT SUBR. FOR
C*******           THE NORTHERN AND SOUTHERN HEMISPHERE RESPECTIVELY
C******* R1,C1 & R2,C2 = STORE THE REAL AND COMPLEX PARTS OF A1 & A2
C
      DIMENSION  CNM(NCOEFF),SNM(NCOEFF),PNMI(NCOEFF),OUT(NSQ)
      DIMENSION  IDIAG(0:NMAX),IWK(NWK),WK(NWK)
      DIMENSION  AM(0:NMAX),BM(0:NMAX)
      DIMENSION  R1(0:NMAX),C1(0:NMAX),R2(0:NMAX),C2(0:NMAX)
      COMPLEX A1(NLON),A2(NLON)
C
C******* GRS80 GRAVITY MODEL VALUES ARE USED
C
      DATA A/6378137.E0/,E2/6.6943800229E-3/,F/298.257222101E0/
      DATA GM/3.986005E14/,OMEGA/7.292115E-5/,XK/1.931851353E-3/
      DATA GE/9.7803267715E0/,XJ2/1.08263E-3/
      DATA GN/9.797644656E0/,RMEAN/6371008.7714/
      F=1.E0/F
C
      PI=ACOS(-1.E0)
      PI2=PI/2.E0
      DGRD=PI/180.E0
      BLKSZ=BLKSIZ*DGRD
C
      TIME0=SECOND()
C
      DO 10 N=0,NMAX
 10   IDIAG(N)=(N*N+N)/2+1
C
C******* READ IN COEFFICIENTS (NO UNITS AND OF THE ORDER OF E-6)
C******* THE S(N,0) ARE INCLUDED THOUGH THEY ARE ALL ZERO.
C******* FOR GEOPOTENTIAL COEFF: C(0,0),C(1,0),C(1,1),S(0,0),S(1,0)
C******* AND S(1,1) ARE INCLUDED BUT ARE ZERO.
C
      OPEN(14,FILE='GEML2')
      OPEN(12,FILE='OUTPNMI')
      OPEN(6,FILE='OUTPUTFFTDGN')
      OPEN(11,FILE='OUTNO')
C
      READ(14)CNM
      READ(14)SNM
C
      TIME1=SECOND()
      WRITE(6,1)'NMAX =',NMAX,'CPU TIME =',(TIME1-TIME0)
      WRITE(6,1)'NCOEFF =',NCOEFF,'C00 TO S11',CNM(1),SNM(1),CNM(2),
     1                                          SNM(2),CNM(3),SNM(3)
C
C******* MODIFY CNM(2,0), CNM(4,0) AND CNM(6,0) COEFFS
C
      IF(IHARM.EQ.0)THEN
      XJN2=XJ2/SQRT(5.E0)
      XJN4=-(E2**2/35.E0)*(10.E0*XJ2/E2-1.E0)
      XJN6=(E2**3/21.E0)*(15.E0*XJ2/E2-2.E0)/SQRT(13.E0)
C
      XM=OMEGA**2*A**3*(1.E0-F)/GM
```

```
      XJNN2=(2.E0/3.E0*(F*(1.E0-F/2.E0)-XM/2.E0*(1.E0-2.E0/7.E0*F
     1      +11.E0*F*F/49.E0)))/SQRT(5.E0)
      XJNN4=(-4.E0/35.E0*F*(1.E0-F/2.E0)*(7.E0*F*(1.E0-F/2.E0)
     1      -5.E0*XM*(1.E0-2.E0*F/7.E0)))/3.E0
      XJNN6=(4.E0*F**2*(6.E0*F-5.E0*XM)/21.E0)/SQRT(13.E0)
C
      IF(IJ2J4.EQ.0)THEN
      XJN2=0.E0
      XJN4=0.E0
      XJN6=0.E0
      ENDIF
C
      CNM(4)=CNM(4)+XJN2
      CNM(11)=CNM(11)+XJN4
      CNM(22)=CNM(22)+XJN6
C
      TIME2=SECOND()
      WRITE(6,1)'NLAT =',NLAT,'CPU TIME =',(TIME2-TIME1)
      WRITE(6,1)'NLON =',NLON,'J2 J4 J6=',XJN2,XJN4,XJN6,
     1                                    XJNN2,XJNN4,XJNN6
      ENDIF
C
C******* COMPUTE THE REQUIRED SINES AND COSINES ARRAYS AM & BM
C
      IF(IMEAN.EQ.1)THEN
      CALL TRIGO (NMAX,BLKSZ,AM,BM)
C
      DO 20 M=1,NMAX
      AM(M)=AM(M)/M
   20 BM(M)=(BM(M)-1.E0)/M
      AM(0)=BLKSZ
      BM(0)=0.E0
C
C******* FOR POINT VALUES: AM & BM CONTAINS SINE & COSINE TO ROTATE
C******* THE GRID BY HALF BLOCK SIZE IN LONGITUDE.
C
      ELSE IF(IMEAN.EQ.0)THEN
      CALL TRIGO (NMAX,BLKSZ*.5E0,BM,AM)
C
      DO 30 M=0,NMAX
   30 BM(M)=-BM(M)
      ENDIF
      TIME3=SECOND()
      WRITE(6,1)'NSQ =',NSQ,'CPU TIME =',(TIME3-TIME2)
C
C******* SET FACTORS FOR HARMONIC, ANOMALIES(MGAL) OR UNDULATIONS(M)
C
      IF(IHARM.EQ.1)THEN
      FACTOR=1.E6
C
      ELSE IF(IANOM.EQ.1)THEN
      IEXPON=2
      FACTOR=1.E5*GM/A**2
C
      ELSE IF(IANOM.EQ.0)THEN
      IEXPON=1
      FACTOR=GM/A
      ENDIF
C
C******* MAIN OUTER LOOP WHERE "OUT" IS COMPUTED WITH 2 EQUATORIALLY
C******* SYMMETRICALLY BANDS AT A TIME TAKING ADVANTAGE OF THE FACT
C******* THAT ALL ASSOCIATED LEGENDRE FUNCTIONS (AND THEIR
C******* INTEGRALS) ARE EITHER EVEN OR ODD WITH RESPECT TO EQUATOR.
C******* THIS IS WHY THE SIZE OF THE BLOCKS MUST BE AN EXACT
C******* DIVIDER OF 90 DEGREES SO THERE IS NO EQUATORIAL BAND
C******* (I.E. LATITUDE = 0), AND "NBANDS" IN ONE HEMISPHERE IS
C******* EVEN AND EQUAL TO 90/"BLKSIZ".
C*******
C******* (PHI: GEODETIC LATITUDE, PSI: GEOCENTRIC LATITUDE)
C******* (PHI: ELLIPSOID OR SPHERICAL LATITUDE DEPENDING ON IELL)
C
      PHI1=PI2
      PSI1=PHI1
      PHI2=PHI1-BLKSZ
      SPARSE=0.E0
      SUMCOS=0.E0
      AREA=1.E0
      GAMMA=1.E0
C
      DO 100 I=1,NBANDS
C
C******* COMPUTE GEOCENTRIC DISTANCE FOR CURRENT CENTER POINT
C******* RAPP GEOMETRY I, EQ.(3.56)
C
      PHIM=(PHI1+PHI2)*.5E0
      SIN2=SIN(PHIM)**2
      W=SQRT(1.E0-E2*SIN2)
      RD=SQRT(1.E0+E2*(E2-2.E0)*SIN2)*A/W
      ARD=A/RD
C
C******* COMPUTE NORMAL GRAVITY FOR CURRENT CENTER POINT
C******* GRS80, P.403
C
      IF(IANOM.EQ.0)THEN
      IF(IELL.EQ.1)GAMMA=(1.E0+XK*SIN2)*GE/W
      IF(IELL.EQ.0)GAMMA=9.79E0
      ENDIF
C
C******* CONVERT TO GEOCENTRIC LATITUDE FOR MEAN CASE
C******* RAPP GEOMETRY I, EQ.(3.62)
C
      IF(IMEAN.EQ.1)THEN
      IF(PHI1.EQ.PI2)GOTO 35
      PSI1=PHI1
      IF(IELL.EQ.1)PSI1=ATAN((1.E0-E2)*TAN(PHI1))
   35 PSI2=PHI2
      IF(IELL.EQ.1)PSI2=ATAN((1.E0-E2)*TAN(PHI2))
C
C******* COMPUTE THE CURRENT BLOCK'S AREA
C
```

```
      AREA=(SIN(PSI1)-SIN(PSI2))*BLKSZ
      ENDIF
C
C******* THE FACTOR .5E0 IS INCLUDED HERE BECAUSE LATER THE
C******* ALPHA & BETA ARE TWICE THERE VALUES.
C
      FACT=.5E0*FACTOR/(AREA*GAMMA)
C
C******* READ IN APPLICABLE LEGENDRE OR INTEGRATED LEGENDRE VALUES
C******* NORMALIZED AND STORED BY BAND FROM NORTH POLE TO EQUATOR.
C******* THE PROGRAM PNMI PROVIDES THESE VALUES.
C*******
C******* FOR POINT VALUES: PNM MUST BE GIVEN AT LAT. OF CENTER BLK
C******* FOR HARM COEFF: PNM AND PNMI MUST BE COMPUTED WITH GEOC LAT
C******* FOR GEOP COEFF: PNM AND PNMI MUST BE COMPUTED WITH GEOD LAT
C*******                     OR GEOC LAT DEPENDING ON IELL
C
      READ(12)PNMI
C
C******* COMPUTE THE ALPHA (R1,R2) AND BETA ARRAYS (C1,C2)
C******* AND ORGANIZED THEM INTO COMPLEX SEQUENCES
C*******
C******* (1) FOR NORTH AND (2) FOR SOUTH PARTS
C******* (R) FOR REAL AND (C) FOR IMAGINARY PARTS
C
      DO 50 M=0,NMAX
      SUMR1=0.E0
      SUMC1=0.E0
      SUMR2=0.E0
      SUMC2=0.E0
C
C******* IF IELL.EQ.1 COMPUTE THE FACTOR FACTN = (A/R)**(N-1)
C
      FACTN=1.E0
      IF(IELL.EQ.1)FACTN=ARD**(M+IEXPON)
C
      DO 40 N=M,NMAX
      NM=IDIAG(N)+M
C
      R=PNMI(NM)
      IF(IANOM.EQ.1)R=R*(N-1)
      IF(IELL.EQ.1)R=R*FACTN
      FACTN=FACTN*ARD
      C=R*CNM(NM)
      S=R*SNM(NM)
      SUMR1=SUMR1+C
      SUMC1=SUMC1+S
      IF(MOD(N+M,2).EQ.1)GOTO 45
C
C******* WHEN N+M IS EVEN, PNM(-LAT)=PNM(LAT)
C******* AND        PINM(-LAT1,-LAT2)=PINM(LAT2,LAT1)
C
      SUMR2=SUMR2+C
      SUMC2=SUMC2+S
      GOTO 40
C
C******* WHEN N+M IS ODD, PNM(-LAT)=-PNM(LAT)
C******* AND       PINM(-LAT1,-LAT2)=-PNM(LAT2,LAT1)
C
   45 SUMR2=SUMR2-C
      SUMC2=SUMC2-S
   40 CONTINUE
C
      R1(M)=SUMR1*AM(M)-SUMC1*BM(M)
      C1(M)=SUMC1*AM(M)+SUMR1*BM(M)
      R2(M)=SUMR2*AM(M)-SUMC2*BM(M)
   50 C2(M)=SUMC2*AM(M)+SUMR2*BM(M)
C
C******* DOUBLE R1(0),R2(0),R1(NMAX),C1(NMAX),R2(NMAX) & C2(NMAX)
C
      R1(0)=2.E0*R1(0)
      R2(0)=2.E0*R2(0)
      C1(0)=0.E0
      C2(0)=0.E0
      IF(NLAT.GE.NMAX)THEN
      R1(NMAX)=2.E0*R1(NMAX)
      C1(NMAX)=2.E0*C1(NMAX)
      R2(NMAX)=2.E0*R2(NMAX)
      C2(NMAX)=2.E0*C2(NMAX)
      ELSE IF(NLAT.LT.NMAX)THEN
C
C******* ALIASE THE COEFFICIENTS WHEN  NLAT.LT.NMAX,
C******* COLOMBO, 1981, OSU REPORT NO.310, P.106 & P.10
C
      DO 65 M=0,NLAT
      M1=M
      M2=-M
      DO 60 K=0,NMAX
      M1=M1+NLON
      M2=M2+NLON
      IF(M2.GT.NMAX)GOTO 65
      IF(M1.GT.NMAX)GOTO 55
      R1(M)=R1(M)+R1(M1)+R1(M2)
      C1(M)=C1(M)+C1(M1)-C1(M2)
      R2(M)=R2(M)+R2(M1)+R2(M2)
      C2(M)=C2(M)+C2(M1)-C2(M2)
      GOTO 60
   55 R1(M)=R1(M)+R1(M2)
      C1(M)=C1(M)-C1(M2)
      R2(M)=R2(M)+R2(M2)
      C2(M)=C2(M)-C2(M2)
   60 CONTINUE
   65 CONTINUE
      ENDIF
C
C******* FORM THE COMPLEX SEQUENCES TO BE FFT
C
      NNN=NMAX
      IF(NLAT.LT.NMAX)NNN=NLAT
      DO 70 M=0,NNN
```

```
      M1=M+1
      A1(M1)=CMPLX(R1(M),-C1(M))
   70 A2(M1)=CMPLX(R2(M),-C2(M))
C
      J=NLON-NNN
      DO 75 M=(NNN+1),J
      M1=M+1
      A1(M1)=CMPLX(0.E0,0.E0)
   75 A2(M1)=CMPLX(0.E0,0.E0)
C
      DO 80 M=(NLON-(NNN-1)),(NLON-1)
      M1=M+1
      M2=NLON-M
      A1(M1)=CMPLX(R1(M2),C1(M2))
   80 A2(M1)=CMPLX(R2(M2),C2(M2))
C
C******* TRANSFORM THE NORTHERN (1) & SOUTHERN (2) COMPLEX SEQUENCES
C
      CALL FFTCC (A1,NLON,IWK,WK)
      CALL FFTCC (A2,NLON,IWK,WK)
C
C******* COMPUTE VALUES FOR CURRENT NORTH AND SOUTH LATITUDES
C
      JN=(I-1)*NLON
      JS=NSQ-(JN+NLON)
      DO 90 J=1,NLON
      JN=JN+1
      JS=JS+1
      OUT(JN)=REAL(A1(J))*FACT
   90 OUT(JS)=REAL(A2(J))*FACT
C
      PHI1=PHI1-BLKSZ
  100 PHI2=PHI2-BLKSZ
C
      WRITE(6,1)'NMAX =',NMAX
C
      TIME4=SECOND()
      WRITE(6,1)'IHARM =',IHARM,'CPU TIME =',(TIME4-TIME3)
C
C****** PRINT OUT THE RESULTING SET OF ANOMALIES OR UNDULATIONS
C
      J=NSQ
      IF(NSQ.GT.600)J=600
      WRITE(6,2)(I,OUT(I),I=1,J)
C
      TIME5=SECOND()
      WRITE(6,1)'IMEAN =',IMEAN,'CPU TIME =',(TIME5-TIME4)
C
      DO 130 I=1,NLAT
      L=(I-1)*NLON+1
      M=L+NLON-1
  130 WRITE(11)(OUT(J),J=L,M)
C
      TIME6=SECOND()
      WRITE(6,1)'IANOM =',IANOM,'CPU TIME =',(TIME6-TIME0)
C
    1 FORMAT(1X,A10,I7,3X,A10,6F15.10)
    2 FORMAT(10(I4,F9.2))
      STOP
      END
      SUBROUTINE TRIGO (NMAX,TETA,SINE,COSINE)
C
C     THIS SUBROUTINE COMPUTES EFFICIENTLY SINE(M)=SIN(M*TETA) AND
C     COSINE(M)=COS(M*TETA),  FOR M=0,NMAX
C
      DIMENSION SINE(0:NMAX),COSINE(0:NMAX)
C
      SINE  (0)=0.E0
      COSINE(0)=1.E0
      SINLON=SIN(TETA)
      COSLON=COS(TETA)
      DO 20 I=1,NMAX
      I1=I-1
      COSINE(I)=COSINE(I1)*COSLON-SINE  (I1)*SINLON
   20 SINE  (I)=SINE  (I1)*COSLON+COSINE(I1)*SINLON
      RETURN
      END
      PROGRAM FFTCNM
C
C
C     FFFFFFFFF FFFFFFFFF TTTTTTTTT  CCCCCCC  NN      NN MM       MM
C     FFFFFFFFF FFFFFFFFF TTTTTTTTT CCCCCCCCC NNN     NN MMMM   MMMM
C     FF        FF           TT     CC      C NN N    NN MM MMMM MM
C     FF        FF           TT     CC        NN NN   NN MM  MM  MM
C     FFFFFF    FFFFFF       TT     CC        NN  NN  NN MM  MM  MM
C     FFFFFF    FFFFFF       TT     CC        NN   NN NN MM      MM
C     FF        FF           TT     CC        NN    NN NN MM     MM
C     FF        FF           TT     CC      C NN     N NN MM     MM
C     FF        FF           TT     CCCCCCCCC NN     NNN MM      MM
C     FF        FF           TT      CCCCCCC  NN      NN MM      MM
C
C
C     THIS PROGRAM COMPUTES GEOPOTENTIAL COEFFICIENTS CNM & SNM
C     UP TO NMAX (=180) FROM A GLOBAL SET OF BLKSIZ BY BLKSIZ
C     (= 1 DEG X 1 DEG) MEAN GRAVITY ANOMALIES OR GEOID UNDULATIONS.
C
C     N.B.: BECAUSE FFT IS HEREIN USED, NMAX & BLKSIZ ARE DEPENDANT
C           ON EACH OTHER BY THE RELATION NMAX=180/BLKSIZ.
C
C     FOR GRAVITY ANOMALIES:
C
C     CNM          1                                  COS
C        = --------------------- INTEGRAL OF ( DG         PNM )
C     SNM   4 PI G (N-1) B(N)**2                      SIN
C
C     FOR GEOID UNDULATIONS:
C
C     CNM          1                               COS
C        = ------------------ INTEGRAL OF ( UND        PNM )
C     SNM   4 PI RMEAN B(N)**2                     SIN
```

```
C
C      WHERE   G = GM/A**2 AND RMEAN = GM/(GAMMA(I)*A)
C
C      IF IHARM.EQ.1, THEN HARMONIC COEFFICIENTS ARE COMPUTED
C
C      CNM            1                                COS
C           = ------------------ INTEGRAL OF ( VALUE       PNM )
C      SNM         4 PI B(N)**2                        SIN
C
C      - FAST FOURIER TRANSFORM IS USED THROUGH THE IMSL SUBR. FFTCC
C
C      - INTEGRALS OF ASS. LEG. FUNCT. COME FROM THE PROGRAM PNMI
C
C      - PELLINEN/MEISSL SMOOTHING OPERATOR BETA(N) IS USED HEREIN
C
C
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. INPUT
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED
C******* NLAT   = 180/BLKSIZ = NYQUIST FREQUENCY
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NSQ    = GLOBAL NUMBER OF BLKSIZ X BLKSIZ SQUARES
C******* NLON   = NUMBER OF BLKSIZ SQUARES AROUND EACH BAND
C******* NCOEFF = NUMBER OF CNM OR SNM COEFFICIENTS
C******* NWK    = NEEDED DIMENSION FOR IMSL FFT ARRAYS
C******* IHARM  = 1.OR.0, TO COMPUTE HARMONIC.OR.GEOPOTENTIAL COEFFS
C******* IBETA  = 1.OR.0, TO USE OR NOT THE PELLINEN/MEISSL
C*******                               SMOOTHING OPERATOR BETA(N)
C******* IANOM  = 1.OR.0, WHEN INPUT IS ANOMALIES.OR.UNDULATIONS
C
       PARAMETER (NMAX=28,BLKSIZ=1.E0,NLAT=180/BLKSIZ+1.E-7)
       PARAMETER (NMAXP1=NMAX+1,NMAXP2=NMAX+2,NCOEFF=NMAXP1*NMAXP2/2)
       PARAMETER (NBANDS=NLAT/2,NLON=2*NLAT,NSQ=NLAT*NLON)
       PARAMETER (NWK=6*NLON+150)
       PARAMETER (IHARM=0,IBETA=1,IANOM=0)
C
C******* CNM,SNM = STORE THE OUTPUT SET OF COEFFICIENTS
C******* PINM    = STORES THE INTEGRALS OF THE ASS. LEG. FUNCT.
C******* UNDG    = STORES THE INPUT SET OF MEAN GRAVITY ANOMALIES
C******* AREA    = STORE AREAS OF BLOCKS
C******* GAMMA   = STORE NORMAL GRAVITY AT CENTER OF BLOCKS
C******* IDIAG   = IS A LOCATING ARRAY FOR CNM, SNM AND PINM
C******* IWK,WK  = ARE REQUIRED BY THE IMSL FFT SUBROUTINE
C******* AM,BM   = STORE SINES AND COSINES REQUIRED
C******* BETA    = STORES THE SMOOTHING OPERATOR BETA(N)
C******* A1,A2   = STORE INPUT.AND.OUTPUT OF THE IMSL FFT SUBR. FOR
C*******             THE NORTHERN AND SOUTHERN HEMISPHERE RESPECTIVELY
C
       DIMENSION  CNM(NCOEFF),SNM(NCOEFF),PINM(NCOEFF)
       DIMENSION  UNDG(NSQ),AREA(NBANDS),GAMMA(NBANDS)
       DIMENSION  IDIAG(0:NMAX),IWK(NWK),WK(NWK)
       DIMENSION  AM(0:NLAT),BM(0:NLAT),BETA(0:NMAX)
       COMPLEX A1(NLON),A2(NLON)
C
C******* GRS80 GRAVITY MODEL VALUES ARE USED
C
       DATA A/6378137.E0/,GM/3.986005E14/
       DATA E2/6.6943800229E-3/,XK/1.931851353E-3/
       DATA GE/9.7803267715E0/
C
       IF(NMAX.GT.NLAT)THEN
       WRITE(6,4)
 4     FORMAT(//,' *** STOP, BECAUSE NMAX MUST BE .LE. TO NLAT')
       ENDIF
C
       DO 5 I=1,NCOEFF
       CNM(I)=0.E0
 5     SNM(I)=0.E0
C
       TIME0=SECOND()
C
       PI=ACOS(-1.E0)
       PI2=PI/2.E0
       DGRD=PI/180.E0
       BLKSZ=BLKSIZ*DGRD
C
       DO 10 N=0,NMAX
 10    IDIAG(N)=N*(N+1)/2+1
C
C******* COMPUTE THE REQUIRED NORMAL GRAVITY
C
       CALL ELAREA (0,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
C
C******* READ IN GRAVITY ANOMALIES (MGAL) OR GEOID UNDULATIONS (M).
C
       OPEN(11,FILE='OUTNO')
       OPEN(12,FILE='OUTPNMI')
       OPEN(6,FILE='OUTPUTFFTCNM')
       OPEN(14,FILE='OUTCNMO')
C
C      READ(11)UNDG
C
       DO 20 I=1,NLAT
       L=(I-1)*NLON+1
       M=L+NLON-1
 20    READ(11) (UNDG(J),J=L,M)
C
C
       TIME1=SECOND()
       WRITE(6,1)'NSQ =',NSQ,'CPU TIME =',TIME1,
      1           'UNDG(1)=',UNDG(1),'UNDG(NSQ)=',UNDG(NSQ)
C
C******* COMPUTE THE REQUIRED SINES AND COSINES ARRAYS AM & BM
C
       CALL TRIGO (NMAX,BLKSZ,AM,BM)
C
       DO 30 M=1,NMAX
       AM(M)=AM(M)/M
 30    BM(M)=(BM(M)-1.E0)/M
       AM(0)=BLKSZ
       BM(0)=0.E0
```

```
C
      TIME2=SECOND()
      WRITE(6,1)'NMAX =',NMAX,'CPU TIME =',TIME2
C
C******* COMPUTE THE SMOOTHING OPERATOR BETA(N)
C
      IF(IBETA.EQ.0)THEN
      DO 40 N=0,NMAX
   40 BETA(N)=1.E0
C
      ELSE IF(IBETA.EQ.1)THEN
      CALL BETAN (NMAX,NLAT,BLKSZ,DGRD,BETA)
      ENDIF
C
      TIME3=SECOND()
      WRITE(6,1)'IBETA =',IBETA,'CPU TIME =',TIME3
C
      DO 45 N=0,NMAX
   45 WRITE(6,1)'N =',N,'BETA(N)  =',BETA(N)
C
C******* PREPARE THE FACTORS FOR GEOPOTENTIAL OR HARMONIC COEFF.
C
      IF(IHARM.EQ.0)THEN
      IF(IANOM.EQ.0)THEN
      DO 50 N=2,NMAX
   50 BETA(N)=1.E0/BETA(N)
C
      F=1.E0/(4.E0*PI*(GM/A))
      DO 55 I=1,NBANDS
      WRITE(6,1)'I =',I,'AREA(I)  =',AREA(I),'GAMMA(I)  =',GAMMA(I)
   55 GAMMA(I)=GAMMA(I)*F
C
      ELSE IF(IANOM.EQ.1)THEN
      DO 60 N=2,NMAX
   60 BETA(N)=1.E0/((N-1)*BETA(N))
C
      F=1.E-5/(4.E0*PI*(GM/A**2))
      DO 65 I=1,NBANDS
      WRITE(6,1)'I =',I,'AREA(I)  =',AREA(I),'GAMMA(I)  =',GAMMA(I)
   65 GAMMA(I)=F
      ENDIF
C
C******* BY SETTING HERE BETA(0)=BETA(1)=0, WILL ASSURE US TO
C******* OBTAIN GEOPOTENTIAL COEFFICIENTS C00=C10=C11=S11=0
C
C     BETA(0)=0.E0
C     BETA(1)=0.E0
C
      ELSE IF  (IHARM.EQ.1)  THEN
      F=1.E0/(4.E0*PI)
      DO 70 N=0,NMAX
   70 BETA(N)=F/BETA(N)
      ENDIF
C
C******* MAIN OUTER LOOP
C
      DO 100 I=1,NBANDS
C
C******* READ IN THE INTEGRATED LEGENDRE VALUES
C
      READ(12)PINM
C
C******* ORGANIZED THE UNDG INTO COMPLEX SEQUENCES TO BE FFT
C
      JN=(I-1)*NLON
      JS=NSQ-(JN+NLON)
      DO 80 J=1,NLON
      JN=JN+1
      JS=JS+1
      A1(J)=CMPLX((UNDG(JN)*GAMMA(I)),0.E0)
   80 A2(J)=CMPLX((UNDG(JS)*GAMMA(I)),0.E0)
C
      TIME3=SECOND()
      WRITE(6,1)'BAND NO =',I,'CPU TIME =',TIME3
      TIME2=TIME3
C
C******* TRANSFORM THE NORTHERN (1) & SOUTHERN (2) COMPLEX SEQUENCES
C
      CALL FFTCC  (A1,NLON,IWK,WK)
      CALL FFTCC  (A2,NLON,IWK,WK)
C
C
C******* COMPUTE CURRENT CONTRIBUTION TO POTENTIAL COEFFICIENTS
C******* MAIN INNER LOOP
C
      DO 100 M=0,NMAX
      M1=M+1
      R1=REAL(A1(M1))
      C1=AIMAG(A1(M1))
      R2=REAL(A2(M1))
      C2=AIMAG(A2(M1))
      CC1=AM(M)*R1+BM(M)*C1
      SS1=AM(M)*C1-BM(M)*R1
      CC2=AM(M)*R2+BM(M)*C2
      SS2=AM(M)*C2-BM(M)*R2
C
C******* WHEN N+M IS EVEN, PINM(-LAT1,-LAT2)=PINM(LAT2,LAT1)
C
      EVENC=CC1+CC2
      EVENS=SS1+SS2
C
C******* WHEN N+M IS ODD, PINM(-LAT1,-LAT2)=-PINM(LAT2,LAT1)
C
      ODDC=CC1-CC2
      ODDS=SS1-SS2
C
      DO 100 N=M,NMAX
      NM=IDIAG(N)+M
      F=BETA(N)*PINM(NM)
      IF(MOD(N+M,2).EQ.1) GOTO 90
```

```
      CNM(NM)=CNM(NM)+F*EVENC
      SNM(NM)=SNM(NM)+F*EVENS
      GOTO 100
 90   CNM(NM)=CNM(NM)+F*ODDC
      SNM(NM)=SNM(NM)+F*ODDS
 100  CONTINUE
C
      TIME4=SECOND()
      WRITE(6,1)'NCOEFF =',NCOEFF,'CPU TIME =',TIME4
C
C****** PRINT OUT THE RESULTING COEFFICIENTS
C
      J=NCOEFF
      IF(NCOEFF.GT.600)J=300
      DO 110 I=1,J
 110  WRITE(6,3)'COEFF NO =',I,'CNM =',CNM(I),'SNM =',SNM(I)
C
      IF(NCOEFF.GT.600)THEN
      DO 120 I=300,0,-1
      J=NCOEFF-I
 120  WRITE(6,3)'COEFF NO =',J,'CNM =',CNM(J),'SNM =',SNM(J)
      ENDIF
C
 1    FORMAT(1X,A10,I7,3(3X,A10,F20.10))
 3    FORMAT(1X,A10,I7,2(3X,A10,E20.10))
C
      TIME5=SECOND()
      WRITE(6,1)'NWK =',NWK,'CPU TIME =',TIME5
C
      WRITE(14)CNM
      WRITE(14)SNM
C
      TIME6=SECOND()
      WRITE(6,1)'IHARM =',IHARM,'CPU TIME =',TIME6
C
      TIME7=SECOND()
      WRITE(6,1)'IANOM =',IANOM,'CPU TIME =',TIME7
C
      STOP
      END
      SUBROUTINE BETAN (NMAX,NLAT,TETA,DGRD,B)
C
C     THIS SUBROUTINE COMPUTES THE VECTOR B(N),
C     WHICH IS THE PELLINEN/MEISSL'S SMOOTHING COEFFICIENTS.
C
C     HOWEVER SOME MODIFICATIONS SHOWN BELOW ARE DONE TO GET THE
C     DE-SMOOTHING OPERATOR OF COLOMBO, 1981, OSU REPORT NO.310
C
      DIMENSION B(0:NMAX),P(0:361)
C
      PI2=DGRD*360.E0
      COSPSI=1.E0-(TETA*SIN(TETA)/PI2)
      NMAXP1=NMAX+1
      NLAT3=NLAT/3
      CALL LEGPOL (NMAXP1,COSPSI,P)
      F=1.E0/(1.E0-COSPSI)
      B(0)=1.E0
      DO 10 N=1,NMAX
      B(N)=F/(2*N+1)*(P(N-1)-P(N+1))
C
C     SQUARE THE LOWEST DEGREE, COLOMBO, 1981, P.76
C
      IF(N.LE.NLAT3)B(N)=B(N)**2
 10   CONTINUE
C
C     PUT B(1)=1, COLOMBO, 1981, P.76
C
      B(1)=1.E0
      RETURN
      END
      SUBROUTINE LEGPOL (NMAX,T,P)
C
C     THIS SUBROUTINE COMPUTES THE LEGENDRE POLYNOMIALS USING THE
C     RECURSION FORMULAE (HM, 1967, EQ.(1-59))
C
      DIMENSION P(0:NMAX)
C
      P(0)=1.E0
      P(1)=T
      DO 10 N=2,NMAX
 10   P(N)=(-(N-1)*P(N-2)+(2*N-1)*T*P(N-1))/N
      RETURN
      END
      SUBROUTINE ELAREA (IELL,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
C
C******* THIS SUBROUTINE COMPUTES THE AREA OF EQUIANGULAR BLOCKS
C******* OF SIZE "BLKSZ" (IN RADIANS) FOR "NBANDS" BLOCKS FROM
C******* POLE TO EQUATOR.  IT ALSO COMPUTES THE NORMAL GRAVITY AT
C******* THE CENTER OF THESE BLOCK WHICH ARE ON AN ELLIPSOID WITH
C******* PARAMETERS E2,GE,XK
C
C******* IF IELL.EQ.0, THE COMPUTATIONS ARE DONE ON THE SPHERE.
C
      DIMENSION AREA(NBANDS),GAMMA(NBANDS)
C
C******* (PHI: GEODETIC LATITUDE, PSI: GEOCENTRIC LATITUDE)
C
      PHI1=PI2
      PSI1=PHI1
      PHI2=PHI1-BLKSZ
C
      DO 40 I=1,NBANDS
      PHIM=(PHI1+PHI2)*.5E0
      SIN2=SIN(PHIM)**2
      W=SQRT(1.E0-E2*SIN2)
C
C******* COMPUTE NORMAL GRAVITY FOR CURRENT CENTER POINT
C******* GRS80, P.403
C
      GAMMA(I)=(1.E0+XK*SIN2)*GE/W
```

```
      IF(IELL.EQ.0)GAMMA(I)=9.79E0
C
C******* CONVERT TO GEOCENTRIC LATITUDE
C******* RAPP GEOMETRY I, EQ. (3.62)
C
      IF(PHI1.EQ.PI2)GOTO 10
      PSI1=PHI1
   10 PSI2=PHI2
      IF(IELL.EQ.0)GOTO 30
      IF(PHI1.EQ.PI2)GOTO 20
      PSI1=ATAN((1.E0-E2)*TAN(PHI1))
   20 PSI2=ATAN((1.E0-E2)*TAN(PHI2))
C
C******* COMPUTE THE CURRENT BLOCK'S AREA
C
   30 AREA(I)=(SIN(PSI1)-SIN(PSI2))*BLKSZ
C
      PHI1=PHI1-BLKSZ
   40 PHI2=PHI2-BLKSZ
C
      RETURN
      END
      SUBROUTINE TRIGO (NMAX,TETA,SINE,COSINE)
C
C     THIS SUBROUTINE COMPUTES EFFICIENTLY SINE(M)=SIN(M*TETA) AND
C     COSINE(M)=COS(M*TETA),  FOR M=0,NMAX
C
      DIMENSION SINE(0:NMAX),COSINE(0:NMAX)
C
      SINE  (0)=0.E0
      COSINE(0)=1.E0
      SINLON=SIN(TETA)
      COSLON=COS(TETA)
      DO 20 I=1,NMAX
      I1=I-1
      COSINE(I)=COSINE(I1)*COSLON-SINE  (I1)*SINLON
   20 SINE  (I)=SINE  (I1)*COSLON+COSINE(I1)*SINLON
      RETURN
      END
      PROGRAM ADDDGN
C
C
C      AAAAAAAA  DDDDDDDDD  DDDDDDDDD  DDDDDDDDD   GGGGGGGG  NN      NN
C     AAAAAAAAAA DDDDDDDDDD DDDDDDDDDD DDDDDDDDDD GGGGGGGGGG NNN     NN
C     AA      AA DD      DD DD      DD DD      DD GG       G NN N    NN
C     AA      AA DD      DD DD      DD DD      DD GG         NN NN   NN
C     AA      AA DD      DD DD      DD DD      DD GG   GGGGG NN  NN  NN
C     AAAAAAAAAA DD      DD DD      DD DD      DD GG  GGGGGG NN   NN NN
C     AAAAAAAAAA DD      DD DD      DD DD      DD GG      GG NN    NN NN
C     AA      AA DD      DD DD      DD DD      DD GG      GG NN     N NN
C     AA      AA DDDDDDDDDD DDDDDDDDDD DDDDDDDDDD GGGGGGGGGG NN     NNN
C     AA      AA DDDDDDDDD  DDDDDDDDD  DDDDDDDDD   GGGGGGGG  NN      NN
C
C
C     THIS PROGRAM SUBSTRACS TWO REGULAR GRIDS OF N OR DG AND COMPUTES
C     THEIR ROOT MEAN SQUARE (RMS) AND RMS DIFFERENCE (AGREEMENT),
C     AND THIS ON THE SPHERE OR ON THE ELLIPSOID.
C
C
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. INPUT
C******* NLAT   = 180/BLKSIZ = NYQUIST FREQUENCY
C******* NLON   = NUMBER OF BLKSIZ SQUARES AROUND EACH BAND
C******* IELL   = 0.OR.1, ON THE SPHERE OR ON THE ELLIPSOID.
C
      PARAMETER (BLKSIZ=1.E0,IELL=0)
      PARAMETER (NLAT=180/BLKSIZ+1.E-7,NLON=2*NLAT,NSQ=NLAT*NLON)
      PARAMETER (NBANDS=NLAT/2)
C
      DIMENSION  UNDG1(NLON),UNDG2(NLON),AREA(NBANDS),GAMMA(NBANDS)
C
      DATA S1,S2,SD,S12,S22,SD2,XMIN1,XMAX1,XMIN2,XMAX2,XMIND,
     1     XMAXD,XMEAN1,XMEAN2,XMEAND,RMS1,RMS2,RMSD/18*0.E0/
      DATA UNDG2/NLON*0.E0/
C
C     GRS80 ELLIPSOID IS USED.
C
      DATA E2/6.6943800229E-3/,XK/1.931851353E-3/
      DATA GE/9.780326771 5E0/
C
      PI=ACOS(-1.E0)
      DGRD=PI/180.E0
      PI2=PI/2.E0
      PI4=4.E0*PI
      BLKSZ=BLKSIZ*DGRD
C
C     COMPUTE THE REQUIRED AREAS.
C
      CALL ELAREA (IELL,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
C
      OPEN(11,FILE='OUTDGO')
      OPEN(12,FILE='OUTDG1')
      OPEN(6,FILE='OUTPUTADDDGN')
      OPEN(14,FILE='OUTDGR1')
C
      DO 20 I=1,NLAT
      READ(11) (UNDG1(J),J=1,NLON)
      READ(12) (UNDG2(J),J=1,NLON)
C
      S1=0.E0
      S2=0.E0
      SD=0.E0
      S12=0.E0
      S22=0.E0
      SD2=0.E0
C
      DO 10 J=1,NLON
      S1=S1+UNDG1(J)
      S2=S2+UNDG2(J)
      S12=S12+UNDG1(J)**2
      S22=S22+UNDG2(J)**2
```

```
      XMAX1=AMAX1(XMAX1,UNDG1(J))
      XMIN1=AMIN1(XMIN1,UNDG1(J))
      XMAX2=AMAX1(XMAX2,UNDG2(J))
      XMIN2=AMIN1(XMIN2,UNDG2(J))
      UNDG1(J)=UNDG1(J)-UNDG2(J)
      SD=SD+UNDG1(J)
      SD2=SD2+UNDG1(J)**2
      XMAXD=AMAX1(XMAXD,UNDG1(J))
      XMIND=AMIN1(XMIND,UNDG1(J))
   10 CONTINUE
      WRITE(14) (UNDG1(J),J=1,NLON)
C
      II=I
      IF(II.GT.NBANDS) II=(NLAT-I)+1
      XMEAN1=XMEAN1+S1*AREA(II)
      XMEAN2=XMEAN2+S2*AREA(II)
      XMEAND=XMEAND+SD*AREA(II)
      RMS1=RMS1+S12*AREA(II)
      RMS2=RMS2+S22*AREA(II)
      RMSD=RMSD+SD2*AREA(II)
   20 CONTINUE
C
      S1=XMEAN1/PI4
      S2=XMEAN2/PI4
      SD=XMEAND/PI4
      S12=SQRT(RMS1/PI4)
      S22=SQRT(RMS2/PI4)
      SD2=SQRT(RMSD/PI4)
C
      WRITE(6,6)' NLAT  =',NLAT,' NLON  =',NLON,' NSQ  =',NSQ
      WRITE(6,7)' MEAN1 =',S1,' RMS1 =',S12,' MAX1 =',XMAX1,
     1                                       ' MIN1 =',XMIN1
      WRITE(6,7)' MEAN2 =',S2,' RMS2 =',S22,' MAX2 =',XMAX2,
     1                                       ' MIN2 =',XMIN2
      WRITE(6,7)' MEAND =',SD,' RMSD =',SD2,' MAXD =',XMAXD,
     1                                       ' MIND =',XMIND
    6 FORMAT(1X,4(A7,I9))
    7 FORMAT(1X,4(A7,F9.3))
C
      STOP
      END
      SUBROUTINE ELAREA (IELL,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
C
C******* THIS SUBROUTINE COMPUTES THE AREA OF EQUIANGULAR BLOCKS
C******* OF SIZE "BLKSZ" (IN RADIANS) FOR "NBANDS" BLOCKS FROM
C******* POLE TO EQUATOR.  IT ALSO COMPUTES THE NORMAL GRAVITY AT
C******* THE CENTER OF THESE BLOCK WHICH ARE ON AN ELLIPSOID WITH
C******* PARAMETERS E2, GE, XK
C
C******* IF IELL.EQ.0, THE COMPUTATIONS ARE DONE ON THE SPHERE.
C
      DIMENSION AREA(NBANDS),GAMMA(NBANDS)
C
C******* (PHI: GEODETIC LATITUDE, PSI: GEOCENTRIC LATITUDE)
C
      PHI1=PI2
      PSI1=PHI1
      PHI2=PHI1-BLKSZ
C
      DO 40 I=1,NBANDS
      PHIM=(PHI1+PHI2)*.5E0
      SIN2=SIN(PHIM)**2
      W=SQRT(1.E0-E2*SIN2)
C
C******* COMPUTE NORMAL GRAVITY FOR CURRENT CENTER POINT
C******* GRS80, P.403
C
      GAMMA(I)=(1.E0+XK*SIN2)*GE/W
      IF(IELL.EQ.0) GAMMA(I)=9.79E0
C
C******* CONVERT TO GEOCENTRIC LATITUDE
C******* RAPP GEOMETRY I, EQ. (3.62)
C
      IF(PHI1.EQ.PI2) GOTO 10
      PSI1=PHI1
   10 PSI2=PHI2
      IF(IELL.EQ.0) GOTO 30
      IF(PHI1.EQ.PI2) GOTO 20
      PSI1=ATAN((1.E0-E2)*TAN(PHI1))
   20 PSI2=ATAN((1.E0-E2)*TAN(PHI2))
C
C******* COMPUTE THE CURRENT BLOCK'S AREA
C
   30 AREA(I)=(SIN(PSI1)-SIN(PSI2))*BLKSZ
C
      PHI1=PHI1-BLKSZ
   40 PHI2=PHI2-BLKSZ
C
      RETURN
      END
      PROGRAM ADDCNM
C
C
C      AAAAAAAA  DDDDDDDDD  DDDDDDDDD   CCCCCCCC  NN      NN MM      MM
C     AAAAAAAAAA DDDDDDDDDD DDDDDDDDDD CCCCCCCCCC NNN     NN MMMM  MMMM
C     AA      AA DD      DD DD      DD CC       C NN N    NN MM MMMM MM
C     AA      AA DD      DD DD      DD CC         NN NN   NN MM  MM  MM
C     AA      AA DD      DD DD      DD CC         NN  NN  NN MM  MM  MM
C     AAAAAAAAAA DD      DD DD      DD CC         NN   NN NN MM      MM
C     AAAAAAAAAA DD      DD DD      DD CC         NN    NN NN MM      MM
C     AA      AA DD      DD DD      DD CC       C NN     N NN MM      MM
C     AA      AA DDDDDDDDDD DDDDDDDDDD CCCCCCCCCC NN      NNN MM      MM
C     AA      AA DDDDDDDDD  DDDDDDDDD   CCCCCCCC  NN      NN MM      MM
C
C
C     THIS PROGRAM ADDS TWO SETS OF COEFFICIENTS AND COMPUTES BY
C     DEGREE THE AGREEMENT WITH A THIRD SET OF COEFFICIENTS.
C
C
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED
```

```
C******* NCOEFF = NUMBER OF CNM OR SNM COEFFICIENTS
C******* IAGREE = 1 OR 0, TO COMPUTE OR NOT THE AGREEMENT.
C
      PARAMETER (NMAX=28, IAGREE=1)
      PARAMETER (NMAXP1=NMAX+1, NMAXP2=NMAX+2, NCOEFF=NMAXP1*NMAXP2/2)
C
      DIMENSION  CNMA(NCOEFF), SNMA(NCOEFF)
      DIMENSION  CNMB(NCOEFF), SNMB(NCOEFF)
C
      DIMENSION  SIGSQ(NMAXP1), SIGSQP(NMAXP1), SIGMA(NMAXP1), R(NMAXP1)
      DIMENSION  DEL(NMAXP1), PN(NMAXP1), NUM(NMAXP1), DG(NMAXP1), Q(NMAXP1)
      DIMENSION  UNDMAG(NMAXP1), ANOMAG(NMAXP1)
C
      OPEN(11, FILE='OUTCNMO')
      OPEN(12, FILE='OUTCNMC1')
      OPEN(13, FILE='GEML2')
      OPEN(6, FILE='OUTPUTADDCNM')
      OPEN(14, FILE='OUTCNM1')
C
      READ(11) CNMA
      READ(11) SNMA
      READ(12) CNMB
      READ(12) SNMB
C
      DO 1 I=1, NCOEFF
      CNMA(I)=CNMA(I)+CNMB(I)
      SNMA(I)=SNMA(I)+SNMB(I)
    1 CONTINUE
C
      II=NCOEFF
      IF(NCOEFF.GT.600) II=300
      DO 2 I=1, II
      WRITE(6,3) I, CNMA(I), SNMA(I)
    2 CONTINUE
    3 FORMAT(I6, 2E16.8)
C
      IF(NCOEFF.GT.600) II=NCOEFF-300
      DO 4 I=II, NCOEFF
      WRITE(6,3) I, CNMA(I), SNMA(I)
    4 CONTINUE
C
      WRITE(14) CNMA
      WRITE(14) SNMA
C
C-----------------------------------------------------------------------
      IF(IAGREE.EQ.0) STOP
C-----------------------------------------------------------------------
C
      READ(13) CNMB
      READ(13) SNMB
C
C****** COMPUTE THE DEGREE VARIANCES OF 2ND SET (SIGSQ)
C
      DO 8 N=2, NMAX
      SIGSQ(N)=0.0E0
      N1=N+1
      DO 8 M1=1, N1
      M=M1-1
      K=N*(N+1)/2+M+1
      IF(CNMB(K).EQ.0.0E0.AND.SNMB(K).EQ.0.0E0) GO TO 8
      SIGSQ(N)=SIGSQ(N)+(CNMB(K)**2+SNMB(K)**2)*1.E12
      TEM=SQRT(SIGSQ(N))
      UNDMAG(N)=TEM*6.371E0
      ANOMAG(N)=TEM*(N-1)*.9798E0
    8 CONTINUE
C
      SUMSIG=0.0E0
      DO 9 I=2, NMAX
    9 SUMSIG=SUMSIG+SIGSQ(I)
C
C******* PRINT OUT THE VALUES OF SIGSQ(I) AND SUMSIG
C
      DO 10 I=2, NMAX
   10 WRITE(6,11) I, SIGSQ(I), UNDMAG(I), ANOMAG(I)
   11 FORMAT(10X,'N=', I3, 5X,'SIGSQ=', F10.4, 5X,'UNDMAG = ', F10.2, 5X,
     1        'ANOMAG = ', F10.2)
      WRITE(6,12) SUMSIG
   12 FORMAT(10X,'SUMSIG=', F10.4)
C
      WRITE(6,13)
   13 FORMAT(1H1, 10X,'COMPARISIONS MADE USING ONLY COMMON COEFFICIENTS')
C
C******* COMPUTE THE DEGREE VARIANCES OF 1ST SET (SIGSQP)
C
      DO 14 N=2, NMAX
      SIGSQP(N)=0.0E0
      N1=N+1
      DO 14 M1=1, N1
      M=M1-1
      K=N*(N+1)/2+M+1
      IF(CNMB(K).EQ.0.0E0.AND.SNMB(K).EQ.0.0E0) GO TO 14
      IF(CNMA(K).EQ.0.0E0.AND.SNMA(K).EQ.0.0E0) GO TO 14
      SIGSQP(N)=SIGSQP(N)+(CNMA(K)**2+SNMA(K)**2)*1.E12
   14 CONTINUE
C
      SMSGP=0.0E0
      DO 15 I=2, NMAX
   15 SMSGP=SMSGP+SIGSQP(I)
C
C******* PRINT OUT THE VALUES OF SIGSQP(I) AND SMSGP
C
      DO 16 I=2, NMAX
   16 WRITE(6,17) I, SIGSQP(I)
   17 FORMAT(10X,'N=', I3, 5X,'SIGSQP=', F10.4)
      WRITE(6,18) SMSGP
   18 FORMAT(10X,'SMSGP=', F10.4)
C
C******* COMPUTE THE CORRELATION COEFFICIENTS "R"
C
      DO 19 N=2, NMAX
```

```
      Q(N)=0.0E0
      N1=N+1
      DO 19 M1=1,N1
      M=M1-1
      K=N*(N+1)/2+M+1
      IF(CNMB(K).EQ.0.0E0.AND.SNMB(K).EQ.0.0E0)GO TO 19
      IF(CNMA(K).EQ.0.0E0.AND.SNMA(K).EQ.0.0E0)GO TO 19
      Q(N)=Q(N)+(CNMB(K)*CNMA(K))*1.E12+(SNMB(K)*SNMA(K))*1.E12
 19   CONTINUE
      DO 20 I=2,NMAX
 20   WRITE(6,21)I,Q(I)
 21   FORMAT(10X,'N=',I3,5X,'Q=',F20.14)
      DO 22 I=2,NMAX
 22   R(I)=Q(I)/((SIGSQ(I)*SIGSQP(I))**0.5E0)
      DO 23 I=2,NMAX
 23   WRITE(6,24)I,R(I)
 24   FORMAT(10X,'N=',I3,5X,'R=',F20.14)
      SUM=0.0E0
      DO 25 I=2,NMAX
 25   SUM=SUM+Q(I)
      R1=SUM/((SUMSIG*SMSGP)**0.5E0)
      WRITE(6,26)R1
 26   FORMAT(10X,'R1=',F10.4)
C
C******* COMPUTE THE PERCENTAGE DIFFERENCE PER DEGREE
C
      DO 27 N=2,NMAX
      KS=N*(N+1)/2+1
      NUM(N)=-1
      IF(CNMB(KS).EQ.0.0E0.OR.CNMA(KS).EQ.0.0E0)NUM(N)=0
      SIGMA(N)=0.0E0
      DG(N)=0.0E0
      N1=N+1
      DO 27 M1=1,N1
      M=M1-1
      K=KS+M
      IF(CNMB(K).EQ.0.0E0.AND.SNMB(K).EQ.0.0E0)GO TO 27
      IF(CNMA(K).EQ.0.0E0.AND.SNMA(K).EQ.0.0E0)GO TO 27
      NUM(N)=NUM(N)+2
      TEM=(((CNMB(K)-CNMA(K))**2)*1.E12)+((SNMB(K)-SNMA(K))**2)*1.E12
      SIGMA(N)=SIGMA(N)+TEM
      DG(N)=DG(N)+.96E0*(N-1)**2*TEM
 27   CONTINUE
      PS=0.0E0
      DO 28 I=2,NMAX
      PN(I)=SQRT(SIGMA(I)/SIGSQ(I))*100.E0
      PS=PS+PN(I)
 28   DEL(I)=(SIGMA(I)/NUM(I))**0.5E0
      WRITE(6,29)
 29   FORMAT(//,20X,'RMS DIFF PER DEGREE',40X,'MAGNITUDE (METRES)',5X,
     1                                       'MAGNITUDE (MGALS)')
      DO 30 I=2,NMAX
      AAC=SQRT(DG(I))
      UNA=6.371E0*SQRT(SIGMA(I))
 30   WRITE(6,31)I,DEL(I),PN(I),NUM(I),UNA,AAC
 31   FORMAT(10X,' N=',I3,5X,'  DEL = ',F10.7,'  PERCENTAGE DIFF',
     1     ' = ',F10.2,' NBR OF COEFF = ',I4,' UND D =',F10.6,' AND D =',
     2        F10.6)
C
C******* COMPUTE THE CUMULATIVE PERCENTAGE
C
      SS=0.0E0
      DO 32 N=2,NMAX
 32   SS=SS+SIGMA (N)
      T=0.0E0
      TG=0.0E0
      DO 33 N=2,NMAX
      T=T+SIGMA(N)
      TG=TG+DG(N)
      A=SQRT(T)
 33   WRITE(6,34)N,A
 34   FORMAT(10X,'N =',I3,5X,'CUMULATIVE DIFFERENCE TO THIS DEGREE = ',
     1        F14.8)
      NTOT=0
      A=A*6.371E0
      DO 35 I=2,NMAX
 35   NTOT=NTOT+NUM(I)
      DEL2=(SS/NTOT)**0.5E0
      PS=PS/(NMAX-1)
      WRITE(6,36)DEL2,PS
 36   FORMAT(10X,'DEL2=',F14.8,'  AVERAGE PERCENTAGE DIFFERENCE = ',
     1        F14.6)
      TG=SQRT(TG)
      WRITE(6,37)A,TG
 37   FORMAT(10X,'RMS UNDULATION DIFF = ',F9.6,/,
     1         10X,'RMS ANOMALY DIFF =',F9.6)
      STOP
      END
      PROGRAM PNMI2
C
C
C     PPPPPPPPP   NN      NN  MM      MM  IIIIIIIII   22222222
C     PPPPPPPPPP  NNN     NN  MMMM  MMMM  IIIIIIIII  2222222222
C     PP      PP  NN N    NN  MM MMMM MM     II      2        22
C     PP      PP  NN NN   NN  MM  MM  MM     II               22
C     PPPPPPPPPP  NN  NN  NN  MM  MM  MM     II              22
C     PPPPPPPPP   NN  NN  NN  MM      MM     II            22
C     PP          NN   NN NN  MM      MM     II          22
C     PP          NN    N NN  MM      MM     II        22
C     PP          NN     NNN  MM      MM  IIIIIIIII  2222222222
C     PP          NN      NN  MM      MM  IIIIIIIII  2222222222
C
C
C******* THIS IS A DRIVER TO RUN PNMI2S SUBROUTINE.
C******* IT COMPUTES EFFICIENTLY THE ASSOCIATED LEGENDRE
C******* FUNCTIONS (PNM) AND THE INTEGRAL OF TWO PNM I.E.  I(N,M,P,Q)
C******* OVER THE NORTHERN HEMISPHERE (AND SOUTHERN IF DESIRED).
C*******
C******* A BACKWARD RECURRENCE IS USED TO COMPUTE THE INTEGRALS
C******* OF THE SECTORIALS EXCEPT NEAR THE EQUATOR WHERE THE FORWARD
```

```
C******* RECURRENCE IS USED.
C
C******* IPNM   = 1.OR.0.  TO COMPUTE  PNM.OR.INMPQ
C******* IGEOD  = 1.OR.0,  TO COMPUTE AT GEODETIC.OR.GEOCENTRIC LAT.
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED FOR PNM OR INMPQ
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. WANTED
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NBAND2 = NBR. OF BANDS FROM NORTH POLE TO SOUTH POLE
C
      PARAMETER (IPNM=0, IGEOD=0, NMAX=28, BLKSIZ=1.E0)
      PARAMETER (NBANDS=90/BLKSIZ+1.E-7, NBAND2=2*NBANDS)
C
      OPEN(6,FILE='OUTPUTPNMI2')
      OPEN(11,FILE='OUTPNMI2')
C
C******* ASSIGN PRESCRIBED ACCURACY
C******* HOWEVER THE USE OF REAL*8 PERMITS AN ACCURACY OF 1.E-7 ONLY
C
      EPSLON=1.E-20
C
C     BY TAKING OUT THE NEXT ERRSET ONE FINDS OUT WHERE FLOATING-
C     POINT UNDERFLOWS OCCUR IN THE SUBROUTINE PNMI2S
C     CALL ERRSET(208,256,-1,1,1)
C
C        THS AND THN ARE THE SOUTHERN AND NORTHERN COLATITUDE.
C        WHEN THS=90 DEG, THE FORWARD RECURRENCE IS ALWAYS USED
C        WHEN THN=0 DEG, THE BACKWARD RECURRENCE IS ALWAYS USED
C
      INIT=0
      PI1=ACOS(-1.E0)
      RDDG=PI1/180.E0
C
C******* DETERMINE THE MAX NBR OF TERMS REQUIRED FOR THE MOST
C******* SOUTHERN BAND THAT NEED THE BACKWARD SECTORIAL RECURRENCE.
C******* HERE ONLY TO CHECK IF SOME DIMENSIONS HAVE TO BE INCREASED
C******* IT USES EQUATIONS (7.13) AND (7.11).
C
      IF(IPNM.EQ.0)THEN
      THS=90.E0-BLKSIZ
      THN=THS-BLKSIZ
C
      II=2
      IF(NBANDS.EQ.1)II=1
      DO 5 I=II,NBANDS
      S2=SIN(THN*RDDG)**2
      IF(ABS(S2).LT.1.E-5)GOTO 3
      RK=FLOAT(NMAX)/(FLOAT(NMAX+1)*S2)
      IF(RK.LE.1.E0)GOTO 4
    3 THM=.5E0*(THN+THS)*RDDG
      RN2=1.E0+ALOG(EPSLON)
      RD2=ALOG(SIN(THM)**2)
      ARG=RN2/RD2
      NTERMS=1+INT(ARG)
      WRITE(6,12)NTERMS
      IF(NTERMS.GT.2870)STOP
      GOTO 6
    4 CONTINUE
      THS=THN
      THN=THN-BLKSIZ
    5 CONTINUE
C
    6 THN=0.E0
C
      ELSE IF(IPNM.EQ.1)THEN
      THN=-BLKSIZ/2
      ENDIF
C
C******* PUT ** DO 10 I=1,NBAND2 ** TO SEE THE RESULTS
C******* IN THE SOUTHERN HEMISPHERE.  ONE FINDS OUT THAT
C******* PNM(-LAT)=-PNM(LAT)  IF N+M IS ODD
C******* PNM(-LAT)= PNM(LAT)  IF N+M IS EVEN
C******* AND A SIMILAR RELATION FOR THE INTEGRALS OF TWO PNM: INMPQ
C******* INMPQ(LAT1,LAT2)=-INMPQ(LAT2,LAT1)  IF N+M+P+Q IS ODD
C******* INMPQ(LAT1,LAT2)= INMPQ(LAT2,LAT1)  IF N+M+P+Q IS EVEN
C
      THS=THN+BLKSIZ
C
C
C     DO 10 I=1,NBAND2
      DO 10 I=1,NBANDS
      WRITE(6,13)SECOND()
      WRITE(6,14)THN,THS
C
      CALL PNMI2S (IPNM,IGEOD,NMAX,THN,THS,RN2,NTERMS,INIT)
C
C******* THE RESULTS ARE PRINT OUT IN THE SUBROUTINE PNMI2S
C
      THN=THS
      THS=THS+BLKSIZ
   10 CONTINUE
C
      WRITE(6,13)SECOND()
   12 FORMAT('   NBR OF TERMS SHOULD NOT EXCEED 2870,  NTERMS =',I10)
   13 FORMAT(5X,'CPU TIME =',F10.5)
   14 FORMAT(30X,'THN =',F10.5,'      THS =',F10.5)
C
      STOP
      END
      SUBROUTINE PNMI2S(IPNM,IGEOD,NMAX,THDEGN,THDEGS,RN2,NTMAX,INIT)
C
C
C******* IT COMPUTES EFFICIENTLY THE ASSOCIATED LEGENDRE
C******* FUNCTIONS (PNM) AND THE INTEGRAL OF TWO PNM I.E.  I(N,M,P,Q)
C******* OVER THE NORTHERN HEMISPHERE (AND SOUTHERN IF DESIRED).
C*******
C******* A BACKWARD RECURRENCE IS USED TO COMPUTE THE INTEGRALS
C******* OF THE SECTORIALS EXCEPT NEAR THE EQUATOR WHERE THE FORWARD
C******* RECURRENCE IS USED.
C
C
```

```
C     IPNM   : 1.OR.0,  TO COMPUTE  PNM.OR.INMPQ
C     IGEOD  : 1.OR.0,  TO COMPUTE AT GEODETIC.OR.GEOCENTRIC LAT.
C     NMAX   : MAXIMUM DEGREE AND ORDER TO COMPUTE PNM OR INMPQ
C     THDEGS : GEOD. POLAR ANGLE IN DEG. OF LOWER LIMIT OF INTEGRAL
C     THDEGN : GEOD. POLAR ANGLE IN DEG. OF UPPER LIMIT OF INTEGRAL
C     PI(P,N): THE MATRIX THAT WILL CONTAIN THE I(N,N,P,P) VALUES
C     OUT    : THE ARRAY  THAT WILL CONTAIN THE I(N,M,P,Q) VALUES
C     RN2    : 1.E0 + ALOG(EPSLON)
C     NTMAX  : NUMBER OF TERMS REQUIRED IN MOST SOUTHERN BAND THAT
C              NEEDS BACKWARD SECTORIAL SCHEME
C     INIT   : MUST BE INITIALIZED AT ZERO (=0) IN MAIN PROGRAM
C
C     --- IN COMMENT STATEMENTS, ALL EQUATION NUMBERS ARE REFERENCED
C     FROM DISSERTATION ---
C
      PARAMETER (NNN=28,NMAX21=NNN*2+1,NENM=(NNN+1)*(NNN+2)/2)
      PARAMETER (NANMPQ=NENM*(NENM+1)/2)
C
C******* ARRAYS SER3, SER4 AND SER5 ARE DIMENSIONED TO ACCOMODATE
C******* A PRESCRIBED ACCURACY OF 1.E-20.
C******* HOWEVER THE USE OF REAL*8 PERMITS AN ACCURACY OF 1.E-7 ONLY
C
C******* A(N,M) IS STORED IN A(N,M)   I.E. LOWER TRIANG. PART OF A=PI
C******* I(N,N,P,P)STORED IN PI(P,N) I.E. UPPER TRIANG. PART OF PI=A
C******* P(N,M) OF TS IS IN P(N,M)    I.E. LOWER TRIANG. PART OF P
C******* P(N,N) OF TS IS IN P(N,N)    I.E. DIAGONAL PART OF P
C******* P(N,M) OF TN IS IN P(M,N)    I.E. UPPER TRIANG. PART OF P
C******* P(N,N) OF TN IS IN PN(N)
C
      DIMENSION SER3(2870),SER4(2870),SER5(2870)
      DIMENSION DIV(NMAX21),SQROOT(NMAX21)
      DIMENSION  A(0:NNN,0:NNN),B(NNN)
      DIMENSION PI(0:NNN,0:NNN)
      DIMENSION  P(0:NNN,0:NNN),PN(0:NNN)
      DIMENSION OUT(NANMPQ),IDIAGO(0:NNN),IDIAG1(0:NNN)
      EQUIVALENCE (A(0,0),PI(0,0))
      HI(N,M,K,L)=OUT(IDIAG1(N)+IDIAGO(N)*M+IDIAGO(M)+IDIAGO(K)+L)
C
C     --- E2 OF GRS80 ---
C
      DATA E2 /.0066943800229E0/
C
      IF(NNN.NE.NMAX)THEN
      WRITE(6,999)
  999 FORMAT('    CHECK INPUT PARAMETER: NMAX')
      STOP
      ENDIF
      NMAXM1=NMAX-1
      NMAXM2=NMAX-2
      NMAXP1=NMAX+1
      NMAXP2=NMAX+2
      IF(INIT.NE.0)GOTO 50
      INIT=1
C
C******* LOAD ARRAYS THAT WILL BE USED REPEATEDLY
C
      SQROOT(1)=1.E0
      DO 1 I=2,NMAX21
      D1=I
    1 SQROOT(I)=SQRT(D1)
C
C     --- COMPUTE B(N) FROM EQUATION (7.5) ---
C     --- SER1 CONTAINS B(NMAX)B(NMAX-1)...B(1) OF (7.37) ---
C     --- SER2 CONTAINS B(NMAX-1)B(NMAX-2)...B(1) OF (7.37) ---
C
      B(1)=SQROOT(3)
      SER1=B(1)
      DO 2 N=2,NMAX
      N2=2*N
      B(N)=SQROOT(N2+1)/SQROOT(N2)
    2 SER1=SER1*B(N)
      SER2=SER1/B(NMAX)
C
C     --- COMPUTE A(N,M) FROM EQUATION (7.3) ---
C
      DO 3 N=1,NMAX
      N2=2*N
      DUM=SQROOT(N2+1)*SQROOT(N2-1)
      DO 3 M=0,(N-1)
    3 A(N,M)=DUM/(SQROOT(N+M)*SQROOT(N-M))
C
C     --- DIV, IDIAGO AND IDIAG1 WILL SIMPLIFY COMPUTATIONS
C
      DO 4 N=0,NMAX
      I=N*(N+1)/2
      IDIAGO(N)=I
    4 IDIAG1(N)=I*(I+1)/2+1
C
      DO 5 N=1,NMAX21
      N1=N+1
    5 DIV(N)=1.E0/N1
C
C     --- SER3 ,SER4 AND SER5 HELP BUILD THE 3RD FACTOR OF EQ. (7.37)
C     I(NMAX,NMAX,NMAX,NMAX),  I(NMAX-1,NMAX-1,NMAX-1,NMAX-1) AND
C     I(NMAX,NMAX,NMAX-1,NMAX-1) ARE THE INITIAL CONDITIONS
C     FOR THE BACKWARD RECURRENCES ---
C
      D1=NMAX+NMAX+2
      D2=NMAXM1+NMAXM1+2
      D3=NMAX+NMAXM1+2
      SER3(1)=1.E0/D1
      SER4(1)=1.E0/D2
      SER5(1)=1.E0/D3
      FRACT=1.E0
      D1=1.E0
      D2=2.E0
      D3=NMAX+NMAX+4
      D4=NMAXM1+NMAXM1+4
      D5=NMAX+NMAXM1+4
      DO 6 J=2,NTMAX
```

```
      FRACT=FRACT*D1/D2
      SER3(J)=FRACT/D3
      SER4(J)=FRACT/D4
      SER5(J)=FRACT/D5
      D1=D1+2.E0
      D2=D2+2.E0
      D3=D3+2.E0
      D4=D4+2.E0
      D5=D5+2.E0
 6    CONTINUE
C
      PI1=ACOS(-1.E0)
      RDDG=PI1/180.E0
      PI2=PI1/2.E0
C
C******* CONVERT GEODETIC COLATITUDES TO GEOCENTRIC COLATITUDES
C******* PHI = GEODETIC LATITUDE, PSI = GEOCENTRIC LATITUDE
C
 50   IF(IGEOD.EQ.1)THEN
      PHIS=(90.E0-THDEGS)*RDDG
      PHIN=(90.E0-THDEGN)*RDDG
      IF(ABS(PHIS+PI2).LT.1.E-5)THEN
      PSIS=-PI2
      ELSE
      PSIS=ATAN((1.E0-E2)*TAN(PHIS))
      ENDIF
      IF(ABS(PHIN-PI2).LT.1.E-5)THEN
      PSIN=PI2
      ELSE
      PSIN=ATAN((1.E0-E2)*TAN(PHIN))
      ENDIF
      THS=PI2-PSIS
      THN=PI2-PSIN
C
      ELSE IF(IGEOD.EQ.0)THEN
      THS=THDEGS*RDDG
      THN=THDEGN*RDDG
      ENDIF
C
      TS=COS(THS)
      YS=SIN(THS)
      TN=COS(THN)
      YN=SIN(THN)
      YS2=YS*YS
      YN2=YN*YN
C
      IF(IPNM.EQ.1)GOTO 70
C
C******* COMPUTE THE CURRENT CONDITION NUMBER RK FROM (7.13)
C
      IF(ABS(YS-1.E0).LT.1.E-5)THEN
      RK=.5E0
      GOTO 70
      ENDIF
      IF(ABS(YN).LT.1.E-5)THEN
      RK=1.5E0
      GOTO 60
      ENDIF
      RK=FLOAT(NMAX)/(FLOAT(NMAXP1)*YN2)
      IF(RK.LE.1.E0)GOTO 70
C
C******* COMPUTE NUMBER OF TERMS NEEDED FOR CURRENT TASK
C
 60   THETAM=.5E0*(THDEGS+THDEGN)*RDDG
      RD2=ALOG(SIN(THETAM)**2)
      ARG=RN2/RD2
      NTERMS=1+INT(ARG)
      WRITE(6,101)NTERMS
 101  FORMAT(40X,'BY BACKWARD RECURRENCE,  NTERMS =',I10)
C
C     --- USE EQUATION (7.37) TO DETERMINE I(NMAX,NMAX,NMAX,NMAX),
C     I(NMAX-1,NMAX-1,NMAX-1,NMAX-1) AND I(NMAX,NMAX,NMAX-1,NMAX-1)
C
      CALL SERIES (YS2,SER6,SER3,NTERMS)
      CALL SERIES (YN2,SER7,SER3,NTERMS)
      CALL SERIES (YS2,SER8,SER4,NTERMS)
      CALL SERIES (YN2,SER9,SER4,NTERMS)
      CALL SERIES (YS2,SER10,SER5,NTERMS)
      CALL SERIES (YN2,SER11,SER5,NTERMS)
C
      N1=NMAX+NMAX+2
      N2=NMAXM1+NMAXM1+2
      N3=NMAX+NMAXM1+2
      PI(NMAX,NMAX)     =-SER1*SER1*(YN**N1*SER7-YS**N1*SER6)
      PI(NMAXM1,NMAXM1)=-SER2*SER2*(YN**N2*SER9-YS**N2*SER8)
      PI(NMAXM1,NMAX)   =-SER1*SER2*(YN**N3*SER11-YS**N3*SER10)
C
      IF(THDEGN.GT.90.E0)THEN
      PI(NMAX,NMAX)=-PI(NMAX,NMAX)
      PI(NMAXM1,NMAXM1)=-PI(NMAXM1,NMAXM1)
      PI(NMAXM1,NMAX)=-PI(NMAXM1,NMAX)
      ENDIF
C
C     --- INITIALIZE P(N,M) AND I(N,N,P,P) FROM (0,0) THRU (1,1)
C     EQUATION (7.8) AND (7.34) ---
C
 70   P(0,0)=1.E0
      P(1,0)=B(1)*TS
      P(1,1)=B(1)*YS
      PN(0)=1.E0
      PN(1)=B(1)*YN
      P(0,1)=B(1)*TN
      PI(0,0)=TN-TS
      PI(0,1)=B(1)*(TN*YN+THS-(TS*YS+THN))*.5E0
      PI(1,1)=TN*(2.E0+YN2)-TS*(2.E0+YS2)
C
C
C     --- COMPUTE ALL P(N,M) WITH (7.7) AND (7.6) ---
C
      DO 100 N=2,NMAX
```

```
      N1=N-1
      P(N,N)=B(N)*YS*P(N1,N1)
  100 PN(N)=B(N)*YN*PN(N1)
C
      DO 120 N=2,NMAX
      N1=N-1
      N2=N-2
      DO 110 M=0,(N-3)
      P(N,M)=(TS*P(N1,M)-P(N2,M)/A(N1,M))*A(N,M)
  110 P(M,N)=(TN*P(M,N1)-P(M,N2)/A(N1,M))*A(N,M)
      P(N,N2)=(TS*P(N1,N2)-P(N2,N2)/A(N1,N2))*A(N,N2)
      P(N2,N)=(TN*P(N2,N1)-PN(N2)/A(N1,N2))*A(N,N2)
      P(N,N1)=TS*P(N1,N1)*A(N,N1)
  120 P(N1,N)=TN*PN(N1)*A(N,N1)
      IF(IPNM.EQ.1)GOTO 1000
C
C     --- COMPUTE ALL I(N,N,M,M)=I(M,M,N,N) AND STORE IN PI(M,N) ---
C
      IF(RK.LE.1.E0)GOTO 160
      DO 150 M=NMAX,1,-1
C
C     --- COMPUTE THE BACKWARD RECURRENCE (7.35) ---
C
      M1=M+1
      M2=M+2
      IF(M.EQ.NMAX)GOTO 140
      IF(M.EQ.NMAXM1)GOTO 130
      PI(M,M)=((M2+M1)*PI(M,M2)+TS*P(M2,M2)*P(M,M)-TN*PN(M2)*PN(M))/
     1          ((M2+M)*B(M2)*B(M1))
  130 PI(M-1,M)=((M1+M1)*PI(M,M1)+TS*P(M1,M1)*P(M,M)-TN*PN(M1)*PN(M))/
     1          ((M1+M)*B(M1)*B(M))
  140 DO 150 N=(M-2),0,-1
      N2=N+2
  150 PI(N,M)=((N2+M1)*PI(N2,M)+TS*P(N2,N2)*P(M,M)-TN*PN(N2)*PN(M))/
     1          ((N2+M)*B(N2)*B(N+1))
      GOTO 190
C
  160 DO 180 N=2,NMAX
C
C     --- COMPUTE THE FORWARD RECURRENCE (7.33) ---
C
      N1=N-1
      N2=N-2
      DO 170 M=0,N2
      NM=N+M
  170 PI(M,N)=(NM*B(N)*B(N1)*PI(M,N2)+TN*PN(N)*PN(M)-
     1                                  TS*P(N,N)*P(M,M))*DIV(NM)
      NM=2*N-1
      PI(N1,N)=(NM*B(N)*B(N1)*PI(N2,N1)+TN*PN(N)*PN(N1)-
     1                                  TS*P(N,N)*P(N1,N1))*DIV(NM)
  180 PI(N,N)=((NM+1)*B(N)*B(N1)*PI(N2,N)+TN*PN(N)*PN(N)-
     1                                  TS*P(N,N)*P(N,N))*DIV(NM+1)
C
C     --- COMPUTE ALL I(N,M,K,L) WITH (7.24) AND (7.25) AND STORE IN OUT
C
C     --- THE FOLLOWING DO LOOPS ARE ORGANIZED SUCH AS:
C           DO 900 N=0,NMAX
C           DO 900 M=0,N
C           DO 300 K=0,N-1
C           DO 300 L=0,K
C   300     OUT(NMKL)=INMKL
C           K=N
C           DO 900 L=0,M
C   900     OUT(NMKL)=INMKL
C
  190 OUT(1)=PI(0,0)
      IN=1
C
      DO 900 N=1,NMAX
      N1=N-1
      N2=N1-1
      N3=N2-1
      NN=2*N
      NN1=NN-1
      NN2=NN1-1
      NN3=NN2-1
C
      DO 400 M=0,N2
      M1=M-1
      DO 300 K=0,N3
      K1=K-1
      K21=2*K+1
      NK=N+K
      N2K=N2-K
C
      DO 200 L=0,K1
      IN=IN+1
  200 OUT(IN)=(N2K*HI(N2,M,K,L)/A(N1,M)+K21*HI(N1,M,K1,L)/A(K,L)+
     1          YS2*P(N1,M)*P(K,L)-YN2*P(M,N1)*P(L,K))*A(N,M)*DIV(NK)
C
C     --- L=K ---
      IN=IN+1
  300 OUT(IN)=(N2K*HI(N2,M,K,K)/A(N1,M)+
     1          YS2*P(N1,M)*P(K,K)-YN2*P(M,N1)*PN(K))*A(N,M)*DIV(NK)
C
C     --- K=N-2 ---
      DO 320 L=0,N3
      IN=IN+1
  320 OUT(IN)=(                          NN3*HI(N1,M,N3,L)/A(N2,L)+
     1          YS2*P(N1,M)*P(N2,L)-YN2*P(M,N1)*P(L,N2))*A(N,M)*DIV(NN2)
C
C     --- L=K=N-2 ---
      IN=IN+1
      OUT(IN)=(YS2*P(N1,M)*P(N2,N2)-YN2*P(M,N1)*PN(N2))*A(N,M)*DIV(NN2)
C
C     --- K=N-1 ---
      DO 360 L=0,N2
      IN=IN+1
  360 OUT(IN)=(-HI(N1,L,N2,M)/A(N1,M)+NN1*HI(N1,M,N2,L)/A(N1,L)+
     1          YS2*P(N1,M)*P(N1,L)-YN2*P(M,N1)*P(L,N1))*A(N,M)*DIV(NN1)
```

```
C
C      --- L=K=N-1 ---
       IN=IN+1
       OUT(IN)=(-HI(N1,N1,N2,M)/A(N1,M)+
     1         YS2*P(N1,M)*P(N1,N1)-YN2*P(M,N1)*PN(N1))*A(N,M)*DIV(NN1)
C
C      --- K=N ---
       DO 400 L=0,M
       IN=IN+1
  400  OUT(IN)=(-2.E0*HI(N,L,N2,M)/A(N1,M)+(NN+1)*HI(N1,M,N1,L)/A(N,L)+
     1         YS2*P(N1,M)*P(N,L)-YN2*P(M,N1)*P(L,N))*A(N,M)*DIV(NN)
C
C      --- M=N-1 ---
       DO 600 K=0,N1
       K1=K-1
       K21=2*K+1
       NK=N+K
       N2K=N2-K
       DO 500 L=0,K1
       IN=IN+1
  500  OUT(IN)=(                             K21*HI(N1,N1,K1,L)/A(K,L)+
     1         YS2*P(N1,N1)*P(K,L)-YN2*PN(N1)*P(L,K))*A(N,N1)*DIV(NK)
C
C      --- L=K,  M=N-1 ---
       IN=IN+1
  600  OUT(IN)=(YS2*P(N1,N1)*P(K,K)-YN2*PN(N1)*PN(K))*A(N,N1)*DIV(NK)
C
C      --- K=N,  M=N-1 ---
       DO 700 L=0,N1
       IN=IN+1
  700  OUT(IN)=(                             (NN+1)*HI(N1,N1,N1,L)/A(N,L)+
     1         YS2*P(N1,N1)*P(N,L)-YN2*PN(N1)*P(L,N))*A(N,N1)*DIV(NN)
C
C      --- M=N ---
       IN=IN+1
       OUT(IN)=PI(0,N)
C
       DO 900 K=1,N
       K1=K-1
       K2=K-2
       NK=N+K
       N2K=K2-N
       DO 800 L=0,K2
       IN=IN+1
  800  OUT(IN)=(N2K*HI(N,N,K2,L)/A(K1,L)+
     1         YS2*P(K1,L)*P(N,N)-YN2*P(L,K1)*PN(N))*A(K,L)*DIV(NK)
C
C      --- L=K-1,  M=N ---
       IN=IN+1
       OUT(IN)=(YS2*P(K1,K1)*P(N,N)-YN2*PN(K1)*PN(N))*A(K,K1)*DIV(NK)
C
C      --- L=K,  M=N ---
       IN=IN+1
  900  OUT(IN)=PI(K,N)
C
C*******    PRINT OUT THE RESULTS
C
C
       WRITE (11) OUT
       WRITE(6,16)NMAX,NMAX,0,0,HI(NMAX,NMAX,0,0)
       WRITE(6,16)NMAX,NMAX,NMAX,NMAX,HI(NMAX,NMAX,NMAX,NMAX)
C
       IPRINT=0
       IF(IPRINT.EQ.0)RETURN
       IF(IPRINT.EQ.1)WRITE(6,*)OUT
       IF(IPRINT.EQ.1)RETURN
       DO 990 N=0,NMAX
       DO 990 M=0,N
       DO 950 K=0,(N-1)
  950  WRITE(6,16)(N,M,K,L,HI(N,M,K,L),L=0,K)
       K=N
  990  WRITE(6,16)(N,M,K,L,HI(N,M,K,L),L=0,M)
       IF(IPRINT.EQ.2)RETURN
C
 1000  I=NMAX
       IF(NMAX.GT.4)I=4
       WRITE(6,15)((N,M,PI(M,N),M=0,N),N=0,I)
       WRITE(6,14)
       WRITE(6,15)((N,M,P(N,M),M=0,N),N=0,I)
       WRITE(6,14)
       WRITE(6,15)((N,M,P(M,N),M=0,N),N=0,I)
       WRITE(6,14)
       WRITE(6,15)(N,N,PN(N),N=0,I)
       WRITE(6,14)
       IF(NMAX.GT.4)WRITE(6,15)(NMAX,M,PI(M,NMAX),M=0,NMAX)
C
   14  FORMAT(//)
   15  FORMAT(2I5,E27.16)
   16  FORMAT(4I5,E27.16)
       RETURN
       END
       SUBROUTINE SERIES (Y2,SER,SERI,NTERMS)
C
C      --- THIS SUBROUTINE DETERMINES THE 3RD FACTOR OF EQ. (7.10)
C      OR (7.37) FOR I(NMAX,NMAX) AND I(NMAX-1,NMAX-1) AT YS AND YN
C      OR I(NMAX,NMAX,NMAX,NMAX), I(NMAX-1,NMAX-1,NMAX-1,NMAX-1) AND
C      I(NMAX,NMAX,NMAX-1,NMAX-1) AT YS AND YN ---
C
       DIMENSION SERI(7296)
       SER=SERI(1)
       IF(ABS(Y2).LT.1.E-5)RETURN
       YY=Y2
       DO 10 I=2,NTERMS
       SER=SER+SERI(I)*YY
       YY=YY*Y2
   10  CONTINUE
       RETURN
       END
       PROGRAM FFTENM
C
```

```
C
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT EEEEEEEEEE NN      NN MM      MM
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT EEEEEEEEEE NNN     NN MMMM  MMMM
C     FF         FF             TT     EE         NN N    NN MM MMMM MM
C     FF         FF             TT     EE         NN NN   NN MM  MM  MM
C     FFFFFF     FFFFFF         TT     EEEEEE     NN  NN  NN MM  MM  MM
C     FFFFFF     FFFFFF         TT     EEEEEE     NN   NN NN MM      MM
C     FF         FF             TT     EE         NN    NN NN MM      MM
C     FF         FF             TT     EE         NN     N NN MM      MM
C     FF         FF             TT     EEEEEEEEEE NN     NNN MM      MM
C     FF         FF             TT     EEEEEEEEEE NN      NN MM      MM
C
C
C     THIS PROGRAM COMPUTES THE ENM & FNM COEFFICIENTS OF THE
C     ALTIMETRY-GRAVIMETRY BOUNDARY-VALUE PROBLEM
C     UP TO NMAX (=180) FROM A GLOBAL SET OF BLKSIZ BY BLKSIZ
C     (= 1 DEG X 1 DEG) MEAN GRAVITY ANOMALIES AND GEOID UNDULATIONS
C
C******* IT IS EQUATION (4.4) (SAME AS (4.11) OR (4.12) OR (8.12)) IN
C******* THE DISSERTATION.
C
C     ENM      1     N-1          2N-1            A(M)            B(M)
C         = ------- SUM I(N,M) SUM R(I,J,N) (       COS(M,J) +      SIN(M,J))
C     FNM   B(N)**2 I=0          J=0            -B(M)            A(M)
C
C                                     IJ
C     WHERE R(I,J,N) = G    * W    * R    /(4 * PI)
C                       IJ     IJ     N
C
C     - THE WEIGHT FUNCTION W    = 1/RMS(N   )**2 OR 1/RMS(DG   )**2.
C                            IJ          IJ                 IJ
C
C     - FAST FOURIER TRANSFORM IS USED THROUGH THE IMSL SUBR. FFTCC
C
C     - INTEGRALS OF ASS. LEG. FUNCT. COME FROM THE PROGRAM PNMI
C
C     - PELLINEN/MEISSL SMOOTHING OPERATOR BETA(N) IS USED HEREIN
C
C
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. INPUT
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED
C******* NLAT   = 180/BLKSIZ = NYQUIST FREQUENCY
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NSQ    = GLOBAL NUMBER OF BLKSIZ X BLKSIZ SQUARES
C******* NLON   = NUMBER OF BLKSIZ SQUARES AROUND EACH BAND
C******* NENM   = NUMBER OF ENM OR FNM COEFFICIENTS
C******* NWK    = NEEDED DIMENSION FOR IMSL FFT ARRAYS
C******* IBETA  = 1.OR.0, TO USE OR NOT THE PELLINEN/MEISSL
C*******                                SMOOTHING OPERATOR BETA(N)
C******* IRMS   = 1.OR.0, TO SCALE OR NOT THE DG AND N BY THEIR RMS.
C
      PARAMETER (NMAX=28,BLKSIZ=1.E0,NLAT=180/BLKSIZ+1.E-7)
      PARAMETER (NMAXP1=NMAX+1,NMAXP2=NMAX+2,NENM=NMAXP1*NMAXP2/2)
      PARAMETER (NBANDS=NLAT/2,NLON=2*NLAT,NSQ=NLAT*NLON)
      PARAMETER (NWK=6*NLON+150,IBETA=1,IRMS=1)
C
C******* ENM,FNM = STORE THE OUTPUT SET OF COEFFICIENTS
C******* PINM    = STORES INTEGRALS OF ASS. LEG. FUNCT.
C******* UNDG    = STORES THE INPUT SET OF MEAN GRAVITY ANOMALIES
C*******             AND MEAN GEOID UNDULATIONS
C******* AREA    = STORES AREAS OF BLOCKS ON THE ELLIPSOID
C******* GAMMA   = STORES NORMAL GRAVITY AT CENTER OF BLOCKS
C******* IDIAG   = IS A LOCATING ARRAY FOR ENM, FNM AND PINM
C******* IWK,WK  = ARE REQUIRED BY THE IMSL FFT SUBROUTINE
C******* AM,BM   = STORE SINES AND COSINES REQUIRED
C******* BETA    = STORES THE SMOOTHING OPERATOR BETA(N)
C******* A1,A2   = STORE INPUT.AND.OUTPUT OF THE IMSL FFT SUBR. FOR
C*******             THE NORTHERN AND SOUTHERN HEMISPHERE RESPECTIVELY
C******* UNDGC   = INDICATOR OF GRAVITY ANOMALIES OR UNDULATIONS
C
      DIMENSION  ENM(NENM),FNM(NENM),PINM(NENM)
      DIMENSION  UND(NSQ),UNDG(NSQ),AREA(NBANDS),GAMMA(NBANDS)
      DIMENSION  IDIAG(0:NMAX),IWK(NWK),WK(NWK)
      DIMENSION  AM(0:NMAX),BM(0:NMAX),BETA(0:NMAX)
      COMPLEX A1(NLON),A2(NLON)
      CHARACTER*1 UNDGC(NSQ),DG
C
      DATA DG/'1'/
C
C******* GRS80 GRAVITY MODEL VALUES ARE USED
C
      DATA A/6378137.E0/,GM/3.986005E14/
      DATA E2/6.6943800229E-3/,XK/1.931851353E-3/
      DATA GE/9.7803267715E0/
C
      OPEN(15,FILE='OUTMAPDG',RECL=64800)
      OPEN(12,FILE='OUTPNMI')
      OPEN(10,FILE='OUTDGO')
      OPEN(11,FILE='OUTNO')
      OPEN(6,FILE='OUTPUTFFTENM')
      OPEN(14,FILE='OUTENMO')
C
      DO 5 I=1,NENM
      ENM(I)=0.E0
    5 FNM(I)=0.E0
C
      TIME0=SECOND()
C
      PI=ACOS(-1.E0)
      PI2=PI/2.E0
      PI4=4.E0*PI
      DGRD=PI/180.E0
      BLKSZ=BLKSIZ*DGRD
C
      DO 10 N=0,NMAX
   10 IDIAG(N)=(N*N+N)/2+1
C
C******* COMPUTE AND PRINT THE REQUIRED AREAS.
C
      CALL ELAREA (0,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
```

```
C
      DO 20 I=1,NBANDS
      WRITE(6,1)'I =',I,'AREA(I)  =',AREA(I),'GAMMA(I)  =',GAMMA(I)
   20 CONTINUE
C
C******* READ IN THE DISTRIBUTION OF ANOMALIES AND UNDULATIONS.
C
      READ(15,'(64800A1)')UNDGC
C
C******* NEXT TO CREATE A RANDOM DISTRIBUTION.
C
C     DO 70 I=1,NSQ
C     J=GGUBFS(123456.E0)+.5E0
C     UNDGC(I)='0'
C     IF(J.EQ.1)UNDGC(I)='1'
C70   CONTINUE
C
C******* PRINT THE DISTRIBUTION OF ANOMALIES AND UNDULATIONS.
C
C     CALL MAPDGN (NLAT,NLON,NSQ,UNDGC)
C
C******* READ IN THE GRAVITY ANOMALIES (MGALS) AND UNDULATIONS (M),
C******* AND COMPUTE THEIR RMS VALUES.
C
      NDG=0
      NUND=0
      RMSDG=0.E0
      RMSN=0.E0
C
      DO 15 I=1,NLAT
      L=(I-1)*NLON+1
      M=L+NLON-1
      READ(10) (UNDG(J),J=L,M)
      READ(11) (UND(J),J=L,M)
      SDG2=0.E0
      SUND2=0.E0
      DO 14 J=L,M
      IF(UNDGC(J).EQ.DG)THEN
      NDG=NDG+1
      SDG2=SDG2+UNDG(J)**2
      ELSE
      NUND=NUND+1
      SUND2=SUND2+UND(J)**2
      UNDG(J)=UND(J)
      ENDIF
   14 CONTINUE
      II=I
      IF(II.GT.NBANDS)II=(NLAT-I)+1
      RMSN=RMSN+SUND2*AREA(II)
      RMSDG=RMSDG+SDG2*AREA(II)
   15 CONTINUE
C
      RMSN=SQRT(RMSN/PI4)
      RMSDG=SQRT(RMSDG/PI4)
      RMSN2=((GM/A)/9.79E0)/RMSN**2
      RMSDG2=((GM/A**2)/1.0E-5)/RMSDG**2
C
      IF(IRMS.EQ.0)THEN
      RMSDG2=1.0E-5/(GM/A**2)
      RMSN2=9.79E0/(GM/A)
      ENDIF
C
      WRITE(6,1)'NBR OF DG=',NDG,'RMS OF DG=',RMSDG,'1/RMSDG2 =',RMSDG2
      WRITE(6,1)'NBR OF N =',NUND,'RMS OF N =',RMSN,'1/RMSN2  =',RMSN2
C
      TIME1=SECOND()
      WRITE(6,1)'NSQ =',NSQ,'CPU TIME =',(TIME1-TIME0),
     1          'UNDG(1)=',UNDG(1),'UNDG(NSQ)=',UNDG(NSQ)
C
C******* SCALE THE INPUT ANOMALIES AND UNDULATIONS BY THEIR RMS**2
C******* AND TRANSFORM TO NON-UNIT VALUES.
C
      JN=0
C
      DO 30 I=1,NBANDS
      DO 30 J=1,NLON
      JN=JN+1
      IF(UNDGC(JN).EQ.DG)GOTO 25
      UNDG(JN)=UNDG(JN)*RMSN2
      GOTO 30
   25 UNDG(JN)=UNDG(JN)*RMSDG2
   30 CONTINUE
C
      DO 40 I=1,NBANDS
      II=(NBANDS-I)+1
      DO 40 J=1,NLON
      JN=JN+1
      IF(UNDGC(JN).EQ.DG)GOTO 35
      UNDG(JN)=UNDG(JN)*RMSN2
      GOTO 40
   35 UNDG(JN)=UNDG(JN)*RMSDG2
   40 CONTINUE
C
C******* COMPUTE THE REQUIRED SINES AND COSINES ARRAYS AM & BM
C
      CALL TRIGO (NMAX,BLKSZ,AM,BM)
C
      DO 45 M=1,NMAX
      AM(M)=AM(M)/M
   45 BM(M)=(BM(M)-1.E0)/M
      AM(0)=BLKSZ
      BM(0)=0.E0
C
      TIME2=SECOND()
      WRITE(6,1)'NMAX =',NMAX,'CPU TIME =',(TIME2-TIME1)
C
C******* COMPUTE THE SMOOTHING OPERATOR BETA(N)
C
      IF(IBETA.EQ.0)THEN
      DO 50 N=0,NMAX
```

```
   50 BETA(N)=1.E0
C
      ELSE IF(IBETA.EQ.1)THEN
      CALL BETAN  (NMAX,NLAT,BLKSZ,DGRD,BETA)
      ENDIF
C
      TIME3=SECOND()
      WRITE(6,1)'IBETA =',IBETA,'CPU TIME =',(TIME3-TIME2)
C
      DO 55 N=0,NMAX
   55 WRITE(6,1)'N =',N,'BETA(N)  =',BETA(N)
C
C******* INVERT BETA(N)
C
      DO 60 N=1,NMAX
   60 BETA(N)=1.E0/(PI4*BETA(N))
C
C******* MAIN OUTER LOOP
C
      DO 100 I=1,NBANDS
C
C******* READ IN THE INTEGRATED LEGENDRE VALUES
C
      READ(12)PINM
C
      TIME4=SECOND()
      WRITE(6,1)'BAND NO =',I,'CPU TIME =',(TIME4-TIME3),UNDGC(I*360)
      TIME3=TIME4
C
      DO 100 N=2,NMAX
C
C******* ORGANIZED THE UNDG INTO COMPLEX SEQUENCES TO BE FFT
C
      JN=(I-1)*NLON
      JS=NSQ-(JN+NLON)
C
      DO 80 J=1,NLON
      JN=JN+1
      JS=JS+1
C
      IF(UNDGC(JN).EQ.DG)GOTO 73
      UNDG1=UNDG(JN)
      GOTO 75
   73 UNDG1=UNDG(JN)*(N-1)
   75 IF(UNDGC(JS).EQ.DG)GOTO 77
      UNDG2=UNOG(JS)
      GOTO 79
   77 UNDG2=UNDG(JS)*(N-1)
   79 A1(J)=CMPLX(UNDG1,0.E0)
   80 A2(J)=CMPLX(UNDG2,0.E0)
C
C******* TRANSFORM THE NORTHERN (1) & SOUTHERN (2) COMPLEX SEQUENCES
C
C
      CALL FFTCC  (A1,NLON,IWK,WK)
      CALL FFTCC  (A2,NLON,IWK,WK)
C
C
C******* COMPUTE CURRENT CONTRIBUTION TO POTENTIAL COEFFICIENTS
C******* MAIN INNER LOOP
C
      DO 100 M=0,N
      M1=M+1
      R1=REAL(A1(M1))
      C1=AIMAG(A1(M1))
      R2=REAL(A2(M1))
      C2=AIMAG(A2(M1))
      EE1=AM(M)*R1+BM(M)*C1
      FF1=AM(M)*C1-BM(M)*R1
      EE2=AM(M)*R2+BM(M)*C2
      FF2=AM(M)*C2-BM(M)*R2
C
      NM=IDIAG(N)+M
      F=BETA(N)*PINM(NM)
      IF(MOD(N+M,2).EQ.1)  GOTO 95
C
C******* WHEN N+M IS EVEN,  PINM(-LAT1,-LAT2)=PINM(LAT2,LAT1)
C
      ENM(NM)=ENM(NM)+F*(EE1+EE2)
      FNM(NM)=FNM(NM)+F*(FF1+FF2)
      GOTO 100
C
C******* WHEN N+M IS ODD,  PINM(-LAT1,-LAT2)=-PINM(LAT2,LAT1)
C
   95 ENM(NM)=ENM(NM)+F*(EE1-EE2)
      FNM(NM)=FNM(NM)+F*(FF1-FF2)
  100 CONTINUE
C
      TIME4=SECOND()
      WRITE(6,1)'NENM =',NENM,'CPU TIME =',(TIME4-TIME3)
C
C****** PRINT OUT THE RESULTING COEFFICIENTS
C
      J=NENM
      IF(NENM.GT.600)J=600
      DO 110 I=1,J
  110 WRITE(6,3)'COEFF NO =',I,'ENM =',ENM(I),'FNM =',FNM(I)
C
      TIME5=SECOND()
      WRITE(6,1)'NLAT =',NLAT,'CPU TIME =',(TIME5-TIME4)
C
      WRITE(14)ENM
      WRITE(14)FNM
C
      TIME6=SECOND()
      WRITE(6,1)'NWK =',NWK,'CPU TIME =',(TIME6-TIME0)
C
    1 FORMAT(1X,A10,I7,4(3X,A10,F15.5))
    3 FORMAT(1X,A10,I7,2(3X,A10,E20.10))
      STOP
```

```
      END
      SUBROUTINE BETAN (NMAX, NLAT, TETA, DGRD, B)
C
C     THIS SUBROUTINE COMPUTES THE VECTOR B(N),
C     WHICH IS THE PELLINEN/MEISSL'S SMOOTHING COEFFICIENTS.
C
C     HOWEVER SOME MODIFICATIONS SHOWN BELOW ARE DONE TO GET THE
C     DE-SMOOTHING OPERATOR OF COLOMBO, 1981, OSU REPORT NO. 310
C
      DIMENSION B(0:NMAX),P(0:361)
C
      PI2=DGRD*360.E0
      COSPSI=1.E0-(TETA*SIN(TETA)/PI2)
      NMAXP1=NMAX+1
      NLAT3=NLAT/3
      CALL LEGPOL (NMAXP1,COSPSI,P)
      F=1.E0/(1.E0-COSPSI)
      B(0)=1.E0
      DO 10 N=1,NMAX
      B(N)=F/(2*N+1)*(P(N-1)-P(N+1))
C
C     SQUARE THE LOWEST DEGREE, COLOMBO, 1981, P.76
C
      IF(N.LE.NLAT3)B(N)=B(N)**2
   10 CONTINUE
C
C     PUT B(1)=1, COLOMBO, 1981, P.76
C
      B(1)=1.E0
      RETURN
      END
      SUBROUTINE LEGPOL (NMAX,T,P)
C
C     THIS SUBROUTINE COMPUTES THE LEGENDRE POLYNOMIALS USING THE
C     RECURSION FORMULAE (HM, 1967, EQ. (1-59))
C
      DIMENSION P(0:NMAX)
C
      P(0)=1.E0
      P(1)=T
      DO 10 N=2,NMAX
   10 P(N)=(-(N-1)*P(N-2)+(2*N-1)*T*P(N-1))/N
      RETURN
      END
      SUBROUTINE TRIGO (NMAX,TETA,SINE,COSINE)
C
C     THIS SUBROUTINE COMPUTES EFFICIENTLY SINE(M)=SIN(M*TETA) AND
C     COSINE(M)=COS(M*TETA), FOR M=0,NMAX
C
      DIMENSION SINE(0:NMAX),COSINE(0:NMAX)
C
      SINE  (0)=0.E0
      COSINE(0)=1.E0
      SINLON=SIN(TETA)
      COSLON=COS(TETA)
      DO 20 I=1,NMAX
      I1=I-1
      COSINE(I)=COSINE(I1)*COSLON-SINE  (I1)*SINLON
   20 SINE  (I)=SINE  (I1)*COSLON+COSINE(I1)*SINLON
      RETURN
      END
      SUBROUTINE ELAREA (IELL,NBANDS,BLKSZ,PI2,E2,GE,XK,AREA,GAMMA)
C
C******* THIS SUBROUTINE COMPUTES THE AREA OF EQUIANGULAR BLOCKS
C******* OF SIZE "BLKSZ" (IN RADIANS) FOR "NBANDS" BLOCKS FROM
C******* POLE TO EQUATOR.  IT ALSO COMPUTES THE NORMAL GRAVITY AT
C******* THE CENTER OF THESE BLOCK WHICH ARE ON AN ELLIPSOID WITH
C******* PARAMETERS E2,GE,XK
C
C******* IF IELL.EQ.0, THE COMPUTATIONS ARE DONE ON THE SPHERE.
C
      DIMENSION AREA(NBANDS),GAMMA(NBANDS)
C
C******* (PHI: GEODETIC LATITUDE. PSI: GEOCENTRIC LATITUDE)
C
      PHI1=PI2
      PSI1=PHI1
      PHI2=PHI1-BLKSZ
C
      DO 40 I=1,NBANDS
      PHIM=(PHI1+PHI2)*.5E0
      SIN2=SIN(PHIM)**2
      W=SQRT(1.E0-E2*SIN2)
C
C******* COMPUTE NORMAL GRAVITY FOR CURRENT CENTER POINT
C******* GRS80, P.403
C
      GAMMA(I)=(1.E0+XK*SIN2)*GE/W
      IF(IELL.EQ.0)GAMMA(I)=9.79E0
C
C******* CONVERT TO GEOCENTRIC LATITUDE
C******* RAPP GEOMETRY I, EQ.(3.62)
C
      IF(PHI1.EQ.PI2)GOTO 10
      PSI1=PHI1
   10 PSI2=PHI2
      IF(IELL.EQ.0)GOTO 30
      IF(PHI1.EQ.PI2)GOTO 20
      PSI1=ATAN((1.E0-E2)*TAN(PHI1))
   20 PSI2=ATAN((1.E0-E2)*TAN(PHI2))
C
C******* COMPUTE THE CURRENT BLOCK'S AREA
C
   30 AREA(I)=(SIN(PSI1)-SIN(PSI2))*BLKSZ
C
      PHI1=PHI1-BLKSZ
   40 PHI2=PHI2-BLKSZ
C
      RETURN
      END
```

```
      SUBROUTINE MAPDGN  (NLAT, NLON, NSQ, UNDGC)
C
C     THIS SUBROUTINE PRINTS THE DISTRIBUTION OF DG(1) AND UND(0)
C     AS A MAP.
C
      CHARACTER*1 UNDGC(NSQ)
C
      NLON3=NLON/3
      N=-NLON3+1
      M=0
      DO 10 K=1,3
      WRITE(6,1)
      N=N+NLON3
      M=M+NLON3
      DO 10 I=1,NLAT
      L=(I-1)*NLON
   10 WRITE(6,2) (UNDGC(L+J),J=N,M)
C
    1 FORMAT('1')
C2    FORMAT(5X,NLON3(A1))
    2 FORMAT(5X,120A1)
      RETURN
      END
      PROGRAM FFTABC
C
C
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT  AAAAAAAA  BBBBBBBBB   CCCCCCCC
C     FFFFFFFFFF FFFFFFFFFF TTTTTTTTTT AAAAAAAAAA BBBBBBBBBB CCCCCCCCCC
C     FF         FF             TT     AA      AA BB      BB CC        C
C     FF         FF             TT     AA      AA BB      BB CC
C     FFFFFF     FFFFFF         TT     AA      AA BBBBBBBBB  CC
C     FFFFFF     FFFFFF         TT     AAAAAAAAAA BBBBBBBBB  CC
C     FF         FF             TT     AAAAAAAAAA BB      BB CC
C     FF         FF             TT     AA      AA BB      BB CC        C
C     FF         FF             TT     AA      AA BBBBBBBBBB CCCCCCCCCC
C     FF         FF             TT     AA      AA BBBBBBBBB   CCCCCCCC
C
C
C     THIS PROGRAM COMPUTES THE ANMPQ, BNMPQ, CNMPQ & DNMPQ COEFF.
C     OF THE ALTIMETRY-GRAVIMETRY BOUNDARY VALUE PROBLEM
C     UP TO NMAX (=180) FROM A GLOBAL SET OF BLKSIZ BY BLKSIZ
C     (= 1 DEG X 1 DEG) WEIGHT FUNCTIONS OF MEAN GRAVITY ANOMALIES
C     AND GEOID UNDULATIONS.
C
C******* IT IS EQUATION (6.1) (SAME AS (6.11) OR (6.14) OR (8.15)) IN
C******* THE DISSERTATION.
C
C
C     ANMPQ                          A(M-Q)               B(M-Q)
C     BNMPQ   N-1        2N-1        A(M-Q)               B(M-Q)
C          = SUM INMPQ SUM RIJNP(         COS(M-Q,J)+          SIN(M-Q,J)+
C     CNMPQ   I=0        J=0         B(M-Q)              -A(M-Q)
C     DNMPQ                         -B(M-Q)               A(M-Q)
C
C                                    A(M+Q)               B(M+Q)
C                               -A(M+Q)               -B(M+Q)
C                             +          COS(M+Q,J)+          SIN(M+Q,J))
C                               -B(M+Q)                A(M+Q)
C                               -B(M+Q)                A(M+Q)
C
C                            IJ
C     WHERE RIJNP = W    * R     /(4 * PI)
C                    IJ     NP
C
C     - THE WEIGHT FUNCTION W   = 1/RMS(N  )**2 OR 1/RMS(DG  )**2.
C                            IJ         IJ                IJ
C
C     - THE RMS VALUES MUST HAVE NO UNITS.
C
C     - FAST FOURIER TRANSFORM IS USED THROUGH THE IMSL SUBR. FFTCC
C
C     - INTEGRALS OF TWO ASS. LEG. FUNCT. COME FROM PROGRAM PNMI2
C
C******* BLKSIZ = BLOCK SIZE (LATITUDINAL BANDWITH) IN DEG. INPUT
C******* NMAX   = MAXIMUM DEGREE AND ORDER WANTED
C******* NLAT   = 180/BLKSIZ = NYQUIST FREQUENCY
C******* NBANDS = NBR. OF BANDS IN NORTHERN HEMISPHERE (=90/BLKSIZ)
C******* NSQ    = GLOBAL NUMBER OF BLKSIZ X BLKSIZ SQUARES
C******* NLON   = NBR. OF BLKSIZ SQUARES AROUND EACH BAND
C******* NANMPQ = NBR. OF INMPQ OR ANMPQ OR B OR C OR DNMPQ COEFF.
C******* NWK    = NEEDED DIMENSION FOR IMSL FFT ARRAYS
C******* IBETA  = 1.OR.0, TO USE OR NOT THE PELLINEN/MEISSL
C*******                             SMOOTHING OPERATOR BETA(N)
C******* RMSDG  = THE RMS VALUE OF THE GRAVITY ANOMALIES IN MGALS.
C******* RMSN   = THE RMS VALUE OF THE GEOID UNDULATIONS IN METRES.
C******* NDG    = NBR. OF GRAVITY ANOMALIES THAT COMPUTED RMSDG.
C******* NUND   = NBR. OF GEOID UNDULATIONS THAT COMPUTED RMSN.
C*******          'NSQ' MUST BE EQUAL TO 'NDG+NUND'.
C******* IRMS   = 1.OR.0, TO USE OR NOT THE RMS VALUES OF DG AND N.
C
      PARAMETER (NMAX=28,BLKSIZ=1.E0,NLAT=180/BLKSIZ+1.E-7)
      PARAMETER (NMAXP1=NMAX+1,NMAXP2=NMAX+2,NENM=NMAXP1*NMAXP2/2)
      PARAMETER (NANMPQ=NENM*(NENM+1)/2,IBETA=0)
      PARAMETER (NBANDS=NLAT/2,NLON=2*NLAT,NSQ=NLAT*NLON)
      PARAMETER (NWK=6*NLON+150,NMAX2=2*NMAX,NMAX1=-(NMAX-1))
      PARAMETER (NOG=19270,RMSDG=7.43853E0,NUND=45530,RMSN=26.84784E0)
      PARAMETER (IRMS=1)
C
C******* ANMPQ, BNMPQ, CNMPQ, DNMPQ = STORE THE OUTPUT SET OF COEFF.
C******* PINMPQ  = STORES THE INTEGRALS OF THE TWO ASS. LEG. FUNCT.
C******* IDIAGO &1 = ARE LOCATING ARRAYS FOR INMPQ,ANMPQ,BNMPQ,CNMPQ
C******* IWK,WK  = ARE REQUIRED BY THE IMSL FFT SUBROUTINE
C******* AM,BM   = STORE SINES AND COSINES REQUIRED
C******* BETA    = STORES THE SMOOTHING OPERATOR BETA(N)
C******* A1,A2   = STORE INPUT.AND.OUTPUT OF THE IMSL FFT SUBR. FOR
C*******           THE NORTHERN AND SOUTHERN HEMISPHERE RESPECTIVELY
C******* R1,C1,R2,C2 = STORE REAL & IMAGINARY PARTS OF A1 & A2
C******* UNDGC   = INDICATOR OF ANOMALIES (1) OR UNDULATIONS (0)
C
      DIMENSION  ANMPQ(NANMPQ),BNMPQ(NANMPQ)
```

```
      DIMENSION   CNMPQ (NANMPQ), DNMPQ (NANMPQ), PINMPQ (NANMPQ)
      DIMENSION   BETA (0: NMAX)
      DIMENSION   IDIAGO (0: NMAX), IDIAG1 (0: NMAX), IWK (NWK), WK (NWK)
      DIMENSION   SINLON (0: NMAX2), COSLON (0: NMAX2)
      DIMENSION   AM (NMAX1: NMAX2), BM (NMAX1: NMAX2)
      DIMENSION   R1 (NMAX1: NMAX2), C1 (NMAX1: NMAX2)
      DIMENSION   R2 (NMAX1: NMAX2), C2 (NMAX1: NMAX2)
      COMPLEX A1 (NLON), A2 (NLON)
      CHARACTER*1 UNDGC (NSQ), DG
      INTEGER P, Q
      EQUIVALENCE  (AM (0), SINLON (0)), (BM (0), COSLON (0))
      INDEX (N, M, P, Q) =IDIAG1 (N) +IDIAGO (N) *M+IDIAGO (M) +IDIAGO (P) +Q
C
      IF (NMAX. GT. NLAT) THEN
      WRITE (6, 6)
 6    FORMAT (//,'  *** STOP,  BECAUSE NMAX MUST BE . LE.  TO NLAT')
      ENDIF
C
      DATA DG/'1'/
C
C******* GRS80 GRAVITY MODEL VALUES ARE USED
C
      DATA A/6378137. E0/, GM/3. 986005E14/
C
      OPEN (15, FILE='OUTMAPDG', RECL=64800)
      OPEN (12, FILE='OUTPNMI2')
      OPEN (6, FILE='OUTPUTFFTABC')
      OPEN (14, FILE='OUTABCD')
C
      DO 5 I=1, NANMPQ
      ANMPQ (I) =0. E0
      BNMPQ (I) =0. E0
      CNMPQ (I) =0. E0
 5    DNMPQ (I) =0. E0
C
      TIME1=SECOND ()
C
      PI=ACOS (-1. E0)
      PI4=4. E0*PI
      DGRD=PI/180. E0
      BLKSZ=BLKSIZ*DGRD
C
      DO 10 N=0, NMAX
      I=N* (N+1) /2
      IDIAGO (N) =I
 10   IDIAG1 (N) =I* (I+1) /2+1
C
C******** THE RMS VALUES ARE TRANSFORMED TO NON-UNIT VALUES AND SQUARED
C
      RMSDG2= (RMSDG*1. E-5/ (GM/A**2)) **2
      RMSN2= (RMSN*9. 79E0/ (GM/A)) **2
      RMSDG2=1. E0/RMSDG2
      RMSN2=1. E0/RMSN2
C
      IF (IRMS. EQ. 0) THEN
      RMSN2=1. E0
      RMSDG2=1. E0
      ENDIF
C
      WRITE (6, 1)'NBR OF DG=', NDG, 'RMS OF DG=', RMSDG, '1/RMSDG2 =', RMSDG2
      WRITE (6, 1)'NBR OF N =', NUND, 'RMS OF N =', RMSN, '1/RMSN2  =', RMSN2
C
C******* COMPUTE THE REQUIRED SINES AND COSINES ARRAYS AM & BM
C
      CALL TRIGO  (NMAX2, BLKSZ, SINLON, COSLON)
C
      DO 45 M=1, NMAX2
      M2=2*M
      AM (M) =AM (M) /M2
 45   BM (M) = (BM (M) -1. E0) /M2
      AM (0) =BLKSZ/2
      BM (0) =0. E0
C
      DO 50 M=NMAX1, -1
      AM (M) =AM (-M)
 50   BM (M) =-BM (-M)
C
      TIME2=SECOND ()
      WRITE (6, 1)'NMAX =', NMAX, 'CPU TIME =', (TIME2-TIME1)
C
C******* COMPUTE THE SMOOTHING OPERATOR BETA (N)
C
      IF (IBETA. EQ. 0) THEN
      DO 55 N=0, NMAX
 55   BETA (N) =1. E0
C
      ELSE IF (IBETA. EQ. 1) THEN
      CALL BETAN  (NMAX, NLAT, BLKSZ, DGRD, BETA)
      ENDIF
C
      TIME3=SECOND ()
      WRITE (6, 1)'IBETA =', IBETA, 'CPU TIME =', (TIME3-TIME2)
C
      DO 60 N=0, NMAX
 60   WRITE (6, 1)'N =', N, 'BETA (N)  =', BETA (N)
C
C******* INVERT BETA (N)
C
      DO 65 N=1, NMAX
 65   BETA (N) =1. E0/ (PI4*BETA (N))
C
C******* READ THE DISTRIBUTION OF ANOMALIES AND UNDULATIONS.
C
      READ (15, '(64800A1)') UNDGC
C
C******* NEXT TO CREATE A RANDOM DISTRIBUTION.
C
C     DO 70 I=1, NSQ
C     J=GGUBFS (123456. E0) +. 5E0
C     UNDGC (I) ='0'
```

```
C      IF (J. EQ. 1) UNDGC (I) =' 1'
C70    CONTINUE
C
C******* PRINT THE DISTRIBUTION OF ANOMALIES AND UNDULATIONS.
C
C      CALL MAPDGN  (NLAT, NLON, NSQ, UNDGC)
C
C******* MAIN OUTER LOOP
C
       DO 100 I=1, NBANDS
C
C******* READ IN THE INTEGRATED LEGENDRE VALUES
C
       READ (12) PINMPQ
C
       TIME4=SECOND ()
       WRITE (6, 1)'BAND NO =', I, 'CPU TIME =', (TIME4-TIME3), UNDGC (I*360)
       TIME3=TIME4
C
       DO 100 N=0, NMAX
       DO 100 P=0, N
       NP= (N-1) * (P-1)
C
C******* ORGANIZED THE ANOM INTO COMPLEX SEQUENCES TO BE FFT
C
       JN= (I-1) *NLON
       JS=NSQ- (JN+NLON)
C
       DO 80 J=1, NLON
       JN=JN+1
       JS=JS+1
C
       IF (UNDGC (JN) . EQ. DG) GOTO 73
       UNDG1=RMSN2
       GOTO 75
  73   UNDG1=NP*RMSDG2
  75   IF (UNDGC (JS) . EQ. DG) GOTO 77
       UNDG2=RMSN2
       GOTO 79
  77   UNDG2=NP*RMSDG2
  79   A1 (J) =CMPLX (UNDG1, 0. E0)
  80   A2 (J) =CMPLX (UNDG2, 0. E0)
C
C******* TRANSFORM THE NORTHERN (1) & SOUTHERN (2) COMPLEX SEQUENCES
C
C
       CALL  FFTCC  (A1, NLON, IWK, WK)
       CALL  FFTCC  (A2, NLON, IWK, WK)
C
C
C******* STORE COMPLEX ARRAYS INTO USEFUL REAL ARRAYS
C
       DO 85 J=0, NMAX2-1
       J1=J+1
       R1 (J) =REAL (A1 (J1) )
       C1 (J) =AIMAG (A1 (J1) )
       R2 (J) =REAL (A2 (J1) )
  85   C2 (J) =AIMAG (A2 (J1) )
C
       DO 90 J=NMAX1, -1
       R1 (J) =R1 (-J)
       C1 (J) =-C1 (-J)
       R2 (J) =R2 (-J)
  90   C2 (J) =-C2 (-J)
C
       NMAX21=NMAX2+1
       IF (NMAX2. EQ. NLON) NMAX21=1
       R1 (NMAX2) =REAL (A1 (NMAX21) )
       C1 (NMAX2) =AIMAG (A1 (NMAX21) )
       R2 (NMAX2) =REAL (A2 (NMAX21) )
       C2 (NMAX2) =AIMAG (A2 (NMAX21) )
C
C******* COMPUTE CURRENT CONTRIBUTION TO POTENTIAL COEFFICIENTS
C******* MAIN INNER LOOP
C
       DO 100 M=0, N
       MP=P
       IF (P. EQ. N) MP=M
       DO 100 Q=0, MP
       MMQ=M-Q
       MPQ=M+Q
       AAA=AM (MMQ) *R1 (MMQ) +BM (MMQ) *C1 (MMQ)
       BBB=AM (MPQ) *R1 (MPQ) +BM (MPQ) *C1 (MPQ)
       CCC=BM (MMQ) *R1 (MMQ) -AM (MMQ) *C1 (MMQ)
       DDD=AM (MPQ) *C1 (MPQ) -BM (MPQ) *R1 (MPQ)
       AA1=AAA+BBB
       BB1=AAA-BBB
       CC1=CCC+DDD
       DD1=DDD-CCC
       AAA=AM (MMQ) *R2 (MMQ) +BM (MMQ) *C2 (MMQ)
       BBB=AM (MPQ) *R2 (MPQ) +BM (MPQ) *C2 (MPQ)
       CCC=BM (MMQ) *R2 (MMQ) -AM (MMQ) *C2 (MMQ)
       DDD=AM (MPQ) *C2 (MPQ) -BM (MPQ) *R2 (MPQ)
       AA2=AAA+BBB
       BB2=AAA-BBB
       CC2=CCC+DDD
       DD2=DDD-CCC
C
       NMPQ=INDEX (N, M, P, Q)
       F=BETA (N) *PINMPQ (NMPQ)
       IF (MOD (N+M+P+Q, 2) . EQ. 1)  GOTO 95
C
C******* WHEN N+M+P+Q IS EVEN,  PINMPQ (-LAT1, -LAT2) =PINMPQ (LAT2, LAT1)
C
       ANMPQ (NMPQ) =ANMPQ (NMPQ) +F* (AA1+AA2)
       BNMPQ (NMPQ) =BNMPQ (NMPQ) +F* (BB1+BB2)
       CNMPQ (NMPQ) =CNMPQ (NMPQ) +F* (CC1+CC2)
       DNMPQ (NMPQ) =DNMPQ (NMPQ) +F* (DD1+DD2)
       GOTO 100
C
```

```
C******* WHEN N+M+P+Q IS ODD, PINMPQ(-LAT1,-LAT2)=-PINMPQ(LAT2,LAT1)
C
 95   ANMPQ(NMPQ)=ANMPQ(NMPQ)+F*(AA1-AA2)
      BNMPQ(NMPQ)=BNMPQ(NMPQ)+F*(BB1-BB2)
      CNMPQ(NMPQ)=CNMPQ(NMPQ)+F*(CC1-CC2)
      DNMPQ(NMPQ)=DNMPQ(NMPQ)+F*(DD1-DD2)
100   CONTINUE
C
      TIME4=SECOND()
      WRITE(6,1)'NANMPQ =',NANMPQ,'CPU TIME =',(TIME4-TIME3)
C
C****** PRINT OUT THE RESULTING COEFFICIENTS
C
      J=NANMPQ
      IF(NANMPQ.GT.120) J=120
      DO 110 I=1,J
110   WRITE(6,3)'I =',I,'ANMPQ =',ANMPQ(I),'BNMPQ =',BNMPQ(I),
     1                    'CNMPQ =',CNMPQ(I),'DNMPQ =',DNMPQ(I)
      IF(NANMPQ.GT.120) THEN
      DO 120 J=59,0,-1
      I=NANMPQ-J
120   WRITE(6,3)'I =',I,'ANMPQ =',ANMPQ(I),'BNMPQ =',BNMPQ(I),
     1                    'CNMPQ =',CNMPQ(I),'DNMPQ =',DNMPQ(I)
      ENDIF
C
C
      TIME5=SECOND()
      WRITE(6,1)'NLAT =',NLAT,'CPU TIME =',(TIME5-TIME4)
C
      WRITE(14) ANMPQ
      WRITE(14) BNMPQ
      WRITE(14) CNMPQ
      WRITE(14) DNMPQ
C
      TIME6=SECOND()
      WRITE(6,1)'NWK =',NWK,'CPU TIME =',(TIME6-TIME1)
C
  1   FORMAT(1X,A10,I7,4(3X,A10,F15.5))
  3   FORMAT(1X,A3,I7,4(3X,A8,E19.10))
      STOP
      END
      SUBROUTINE BETAN (NMAX,NLAT,TETA,DGRD,B)
C
C     THIS SUBROUTINE COMPUTES THE VECTOR B(N),
C     WHICH IS THE PELLINEN/MEISSL'S SMOOTHING COEFFICIENTS.
C
C     HOWEVER SOME MODIFICATIONS SHOWN BELOW ARE DONE TO GET THE
C     DE-SMOOTHING OPERATOR OF COLOMBO, 1981, OSU REPORT NO.310
C
      DIMENSION B(0:NMAX),P(0:361)
C
      PI2=DGRD*360.E0
      COSPSI=1.E0-(TETA*SIN(TETA)/PI2)
      NMAXP1=NMAX+1
      NLAT3=NLAT/3
      CALL LEGPOL (NMAXP1,COSPSI,P)
      F=1.E0/(1.E0-COSPSI)
      B(0)=1.E0
      DO 10 N=1,NMAX
      B(N)=F/(2*N+1)*(P(N-1)-P(N+1))
C
C     SQUARE THE LOWEST DEGREE, COLOMBO, 1981, P.76
C
      IF(N.LE.NLAT3) B(N)=B(N)**2
 10   CONTINUE
C
C     PUT B(1)=1, COLOMBO, 1981, P.76
C
      B(1)=1.E0
      RETURN
      END
      SUBROUTINE LEGPOL (NMAX,T,P)
C
C     THIS SUBROUTINE COMPUTES THE LEGENDRE POLYNOMIALS USING THE
C     RECURSION FORMULAE (HM, 1967, EQ.(1-59))
C
      DIMENSION P(0:NMAX)
C
      P(0)=1.E0
      P(1)=T
      DO 10 N=2,NMAX
 10   P(N)=(-(N-1)*P(N-2)+(2*N-1)*T*P(N-1))/N
      RETURN
      END
      SUBROUTINE TRIGO (NMAX,TETA,SINE,COSINE)
C
C     THIS SUBROUTINE COMPUTES EFFICIENTLY SINE(M)=SIN(M*TETA) AND
C     COSINE(M)=COS(M*TETA), FOR M=0,NMAX
C
      DIMENSION SINE(0:NMAX),COSINE(0:NMAX)
C
      SINE  (0)=0.E0
      COSINE(0)=1.E0
      SINLON=SIN(TETA)
      COSLON=COS(TETA)
      DO 20 I=1,NMAX
      I1=I-1
      COSINE(I)=COSINE(I1)*COSLON-SINE  (I1)*SINLON
 20   SINE  (I)=SINE  (I1)*COSLON+COSINE(I1)*SINLON
      RETURN
      END
      SUBROUTINE MAPDGN (NLAT,NLON,NSQ,UNDGC)
C
C     THIS SUBROUTINE PRINTS THE DISTRIBUTION OF DG(I) AND UND(O)
C     AS A MAP.
C
      CHARACTER*1 UNDGC(NSQ)
C
      NLON3=NLON/3
      N=-NLON3+1
```

```
      M=0
      DO 10 K=1,3
      WRITE(6,1)
      N=N+NLON3
      M=M+NLON3
      DO 10 I=1,NLAT
      L=(I-1)*NLON
   10 WRITE(6,2) (UNDGC(L+J),J=N,M)
C
    1 FORMAT('1')
C2    FORMAT(5X,NLON3(A1))
    2 FORMAT(5X,120A1)
      RETURN
      END
      PROGRAM ORTHO
C
C
C      00000000   RRRRRRRRR   TTTTTTTTT  HH       HH   00000000
C     0000000000  RRRRRRRRRR  TTTTTTTTT  HH       HH  0000000000
C     00      00  RR      RR     TT      HH       HH  00      00
C     00      00  RR      RR     TT      HH       HH  00      00
C     00      00  RRRRRRRRRR     TT      HHHHHHHHHHH  00      00
C     00      00  RRRRRRRRR      TT      HHHHHHHHHHH  00      00
C     00      00  RR    RR       TT      HH       HH  00      00
C     00      00  RR     RR      TT      HH       HH  00      00
C     0000000000  RR      RR     TT      HH       HH  0000000000
C      00000000   RR       RR    TT      HH       HH   00000000
C
C
C     THIS PROGRAM COMPUTES THE HARMONIC COEFFICIENTS TK
C     UP TO DEGREE AND ORDER NMAX
C     SOLVING THE ALTIMETRY-GRAVIMETRY BOUNDARY VALUE PROBLEM
C     BY THE GRAM-SCHMIDT ORTHONORMALIZATION PROCESS.
C
C     IT COMPUTES EQUATIONS (8.17) TO (8.21) OF THE DISSERTATION.
C
C
C     NMAX   = MAXIMUM DEGREE AND ORDER WANTED
C     NENM   = NBR. OF ENM OR FNM COEFF.
C     NEF    = (NMAX+1)**2 - 4 = NBR. OF ENM & FNM WITHOUT THE
C              COEFF. FN0=0 & WITHOUT E00, E10, E11, F11.
C     NEF1   = V IN DISSERTATION = NEF-1
C     NEF1   = ALSO NBR. OF ELEMENTS IN ARRAY EK OR TK
C     NANMPQ = NBR. OF ANMPQ OR BNMPQ OR CNMPQ OR DNMPQ COEFF.
C     NABC   = NBR. OF ELEMENTS IN MATRIX CNM OR GNM
C     T      : THE ARRAY THAT WILL CONTAINS THE ALT-GRAV. COEFF.
C
      INTEGER P,Q,P1,PP
      PARAMETER (NMAX=28)
      PARAMETER (NENM=(NMAX+1)*(NMAX+2)/2,NANMPQ=NENM*(NENM+1)/2)
      PARAMETER (NEF=(NMAX+1)**2-4,NEF1=NEF-1)
      PARAMETER (NABC=NEF*(NEF+1)/2)
C
      DIMENSION C(NABC),G(NABC),E(0:NEF1),T(0:NEF1),IDIAG(0:NEF)
      DIMENSION ENM(NENM),ANMPQ(NANMPQ),BNMPQ(NANMPQ)
      DIMENSION FNM(NENM),CNMPQ(NANMPQ),DNMPQ(NANMPQ)
      DIMENSION CMATRX(NEF,NEF)
      EQUIVALENCE (CMATRX(1,1),C(1),G(1)),(E(0),T(0))
      EQUIVALENCE (ENM(1),ANMPQ(1)),(FNM(1),BNMPQ(1))
C
      OPEN(11,FILE='OUTENMO')
      OPEN(12,FILE='OUTABCD')
      OPEN(6,FILE='OUTPUTORTHO')
      OPEN(14,FILE='OUTORTHO')
C
      WRITE(6,52)'NMAX =',NMAX,'NEF =',NEF,'NABC =',NABC
      WRITE(6,52)'NENM =',NENM,'NANMPQ =',NANMPQ
C
C
      WRITE(6,53)SECOND()
C
C     --- IDIAG IS USED TO STORE MATRICES C AND G IN ARRAYS C AND G
C         AS:        G(N,M)=G(IDIAG(N)+M)
C                    C(N,M)=C(IDIAG(N)+M)
C
      DO 10 I=0,NEF
   10 IDIAG(I)=(I*I+I)/2+1
C
C     --- READ IN ANMPQ, BNMPQ, CNMPQ & DNMPQ AND STORE IN CMATRX
C
      READ(12)ANMPQ
      READ(12)BNMPQ
      READ(12)CNMPQ
      READ(12)DNMPQ
C
      CALL ABCDOR (NMAX,NEF,NANMPQ,ANMPQ,BNMPQ,CNMPQ,DNMPQ,CMATRX)
C
      K=0
      DO 11 I=1,NEF
      DO 11 J=1,I
      K=K+1
   11 C(K)=CMATRX(J,I)
      WRITE(6,*)K
C
C     --- READ IN ENM & FNM AND STORE IN ARRAY E ---
C
      READ(11)ENM
      READ(11)FNM
C
      IENM=4
      K=0
      DO 12 N=2,NMAX
C..M=0
      E(K)=ENM(IENM)
      K=K+1
      IENM=IENM+1
      DO 12 M=1,N
      E(K)=ENM(IENM)
      K=K+1
      E(K)=FNM(IENM)
```

```
      K=K+1
   12 IENM=IENM+1
C
    6 WRITE(6,54)E
      WRITE(6,51)
C     WRITE(6,54)C
      WRITE(6,53)SECOND()
C
C     --- START THE ORTHOGONALIZATION PROCESS ---
C
      G(1)=1.E0/SQRT(C(1))
      E(0)=G(1)*E(0)
      WRITE(6,54)E(0)
      WRITE(6,54)G(1)
      WRITE(6,55)1,NEF
      DO 30 P=1,NEF1
      P1=P-1
C
C     --- COMPUTE C(P,P-1),C(P,P-2),....,C(P,0) USING EQ. (8.14) ---
C
      DO 22 N=P1,0,-1
      SUM=0.E0
      DO 21 Q=0,N
   21 SUM=SUM+G(IDIAG(N)+Q)*C(IDIAG(P)+Q)
   22 C(IDIAG(P)+N)=-SUM
C
C     --- COMPUTE G(P,P) USING EQ. (8.15) ---
C
      SUM=0.E0
      DO 23 N=0,P1
   23 SUM=SUM+C(IDIAG(P)+N)**2
      PP=IDIAG(P)+P
      G(PP)=1.E0/SQRT(G(PP)-SUM)
C
C     --- COMPUTE E(P) USING EQ. (8.12) ---
C
      SUM=0.E0
      DO 24 N=0,P1
   24 SUM=SUM+C(IDIAG(P)+N)*E(N)
      E(P)=G(PP)*(SUM+E(P))
C
C     --- COMPUTE G(P,P-1),G(P,P-2),....,G(P,0) USING EQ. (8.13) ---
C
      DO 26 K=0,P1
      SUM=0.E0
      DO 25 N=K,P1
   25 SUM=SUM+C(IDIAG(P)+N)*G(IDIAG(N)+K)
   26 G(IDIAG(P)+K)=G(PP)*SUM
C
      WRITE(6,54)E(P)
      WRITE(6,54) (G(I),I=IDIAG(P),PP)
      WRITE(6,55) (P+1),NEF
   30 CONTINUE
C
C     --- COMPUTE T(K) USING EQ. (8.16) ---
C
      DO 40 K=0,NEF1
      SUM=0.E0
      DO 35 P=K,NEF1
   35 SUM=SUM+E(P)*G(IDIAG(P)+K)
   40 T(K)=SUM
C
C     --- PRINT THE RESULTING COEFFICIENTS ---
C
      WRITE(6,51)
C
      I=0
      J=4
      DO 45 N=2,NMAX
      WRITE(6,56)N,0,T(I)
      ENM(J)=T(I)
      FNM(J)=0.E0
      I=I+1
      J=J+1
      DO 45 M=1,N
      WRITE(6,56)N,M,T(I),T(I+1)
      ENM(J)=T(I)
      FNM(J)=T(I+1)
      J=J+1
   45 I=I+2
C
      WRITE(14)ENM
      WRITE(14)FNM
C
      WRITE(6,53)SECOND()
   51 FORMAT(//)
   52 FORMAT(3(5X,A10,3X,I10))
   53 FORMAT(/,'       CPU TIME =',F10.5)
   54 FORMAT(10E13.5)
   55 FORMAT('     LINE',I10,' OF',I10,' OF MATRIX G WAS COMPUTED ',
     1       'AND WRITTEN OUT.')
   56 FORMAT(2I5,2E26.16)
      STOP
      END
      SUBROUTINE ABCDOR (NMAX,NEF,NANMPQ,ANMPQ,BNMPQ,CNMPQ,DNMPQ,C)
C
C     THIS SUBROUTINE REORGANIZES ANMPQ, BNMPQ, CNMPQ & DNMPQ INTO
C     THE UPPER TRIANGULAR PART OF MATRIX C.
C
      DIMENSION ANMPQ(NANMPQ),BNMPQ(NANMPQ)
      DIMENSION CNMPQ(NANMPQ),DNMPQ(NANMPQ),C(NEF,NEF)
C
      IA=10
      I=1
      J=1
      DO 4 N=2,NMAX
C..M=0
      DO 1 K=2,N-1
C..L=0
      C(I,J)=ANMPQ(IA)
```

```
      I=I+1
      IA=IA+1
      DO 1 L=1,K
      C(I,J)=ANMPQ(IA)
      C(I+1,J)=CNMPQ(IA)
      I=I+2
 1    IA=IA+1
C..K=N
C..L=0
      C(I,J)=ANMPQ(IA)
      IA=IA+4
      J=J+1
      I=1
      DO 4 M=1,N
      DO 2 K=2,N-1
C..L=0
      C(I,J)=ANMPQ(IA)
      C(I,J+1)=DNMPQ(IA)
      I=I+1
      IA=IA+1
      DO 2 L=1,K
      C(I,J)=ANMPQ(IA)
      C(I,J+1)=DNMPQ(IA)
      C(I+1,J)=CNMPQ(IA)
      C(I+1,J+1)=BNMPQ(IA)
      I=I+2
 2    IA=IA+1
C..K=N
C..L=0
      C(I,J)=ANMPQ(IA)
      C(I,J+1)=DNMPQ(IA)
      I=I+1
      IA=IA+1
      DO 3 L=1,M-1
      C(I,J)=ANMPQ(IA)
      C(I,J+1)=DNMPQ(IA)
      C(I+1,J)=CNMPQ(IA)
      C(I+1,J+1)=BNMPQ(IA)
      I=I+2
 3    IA=IA+1
C..L=M
      C(I,J)=ANMPQ(IA)
      C(I,J+1)=DNMPQ(IA)
      I=I+1
      IA=IA+1
      J=J+1
      C(I,J)=BNMPQ(IA-1)
      I=1
      J=J+1
 4    IA=IA+3
      RETURN
      END
      PROGRAM ORTHOC
C
C
C      00000000  RRRRRRRRR  TTTTTTTTT HH       HH  00000000    CCCCCCCC
C     0000000000 RRRRRRRRRR TTTTTTTTT HH       HH 0000000000 CCCCCCCCCC
C     00      00 RR       RR     TT    HH       HH 00      00 CC        C
C     00      00 RR       RR     TT    HH       HH 00      00 CC
C     00      00 RRRRRRRRRR      TT    HHHHHHHHHHH 00      00 CC
C     00      00 RRRRRRRRR       TT    HHHHHHHHHHH 00      00 CC
C     00      00 RR    RR        TT    HH       HH 00      00 CC
C     00      00 RR     RR       TT    HH       HH 00      00 CC        C
C     0000000000 RR      RR      TT    HH       HH 0000000000 CCCCCCCCCC
C      00000000  RR       RR     TT    HH       HH  00000000    CCCCCCCC
C
C
C     THIS PROGRAM COMPUTES THE HARMONIC COEFFICIENTS TK
C     UP TO DEGREE AND ORDER NMAX
C     SOLVING THE ALTIMETRY-GRAVIMETRY BOUNDARY VALUE PROBLEM
C     BY THE ORTHOGONALIZATION PROCESS USING CHOLESKY FACTORIZATION
C
C     - DPPFA, DPPSL & DPPCO ARE EFFICIENT ROUTINES FROM
C       LINPACK PACKAGE.
C
C     IT COMPUTES EQUATIONS (8.17) TO (8.21) BUT BY CHOLESKY
C     FACTORIZATION, I.E. IT HERE SOLVES (8.22), (8.25) AND (8.26)
C     USING (8.30), (8.31) AND (8.32).
C
C     NMAX   = MAXIMUM DEGREE AND ORDER WANTED
C     NENM   = NBR. OF ENM OR FNM COEFF.
C     NEF    = (NMAX+1)**2 - 4 = NBR. OF ENM & FNM WITHOUT THE
C              COEFF. FN0=0 & WITHOUT E00, E10, E11, F11.
C     NEF1   = V IN DISSERTATION = NEF-1
C     NEF1   = ALSO NBR. OF ELEMENTS IN ARRAY EK OR TK
C     NANMPQ = NBR. OF ANMPQ OR BNMPQ OR CNMPQ OR DNMPQ COEFF.
C     NABCD  = NBR. OF ELEMENTS IN SYMMETRIC MATRIX C.
C     T      : THE ARRAY THAT WILL CONTAINS THE ALT-GRAV. COEFF.
C
      PARAMETER (NMAX=28,ICOND=1)
      PARAMETER (NENM=(NMAX+1)*(NMAX+2)/2,NANMPQ=NENM*(NENM+1)/2)
      PARAMETER (NEF=(NMAX+1)**2-4,NEF1=NEF-1)
      PARAMETER (NABCD=NEF*(NEF+1)/2)
C
      DIMENSION C(NABCD),E(0:NEF1),T(NEF),IDIAG(NEF)
      DIMENSION ENM(NENM),ANMPQ(NANMPQ),BNMPQ(NANMPQ)
      DIMENSION FNM(NENM),CNMPQ(NANMPQ),DNMPQ(NANMPQ)
      EQUIVALENCE (E(0),T(1))
      EQUIVALENCE (ENM(1),ANMPQ(1)),(FNM(1),BNMPQ(1))
C
      OPEN(11,FILE='OUTENMO')
      OPEN(12,FILE='OUTABCD')
      OPEN(6,FILE='OUTPUTORTHOC')
      OPEN(14,FILE='OUTORTHOC')
C
      WRITE(6,52)'NMAX =',NMAX,'NEF =',NEF,'NABCD =',NABCD
      WRITE(6,52)'NENM =',NENM,'NANMPQ =',NANMPQ
C
      WRITE(6,53)SECOND()
C
```

```
C     --- READ IN ANMPQ, BNMPQ, CNMPQ & DNMPQ AND STORE IN MATRIX C
C
      REWIND 12
      READ(12)ANMPQ
      READ(12)BNMPQ
      READ(12)CNMPQ
      READ(12)DNMPQ
C
      CALL ABCDV (NMAX,NEF,NABCD,NANMPQ,ANMPQ,BNMPQ,CNMPQ,
     1            DNMPQ,IDIAG,C)
      WRITE(6,51)
C     WRITE(6,54)C
      WRITE(6,54) (C(IDIAG(I)+I),I=1,NEF)
      WRITE(6,53)SECOND()
C
C    --- CHOLESKY FACTORIZATION OF THE SYM. POS. DEFINITE C MAT. ---
C
      IF(ICOND.EQ.0)CALL DPPFA (C,NEF,INFO)
      IF(ICOND.EQ.1)CALL DPPCO (C,NEF,RCOND,T,INFO)
      WRITE(6,*)INFO
      IF(ICOND.EQ.1) WRITE(6,*)'      THE CONDITION NUMBER MUST BE',
     1                       ' CLOSE TO 1, IT IS =',1.E0/RCOND
      WRITE(6,51)
C     WRITE(6,54)C
      WRITE(6,54) (C(IDIAG(I)+I),I=1,NEF)
      WRITE(6,53)SECOND()
C
C     --- READ IN ENM & FNM AND STORE IN ARRAY E ---
C
      READ(11)ENM
      READ(11)FNM
C
      IENM=4
      IE=0
      DO 10 N=2,NMAX
C..M=0
      E(IE)=ENM(IENM)
      IE=IE+1
      IENM=IENM+1
      DO 10 M=1,N
      E(IE)=ENM(IENM)
      IE=IE+1
      E(IE)=FNM(IENM)
      IE=IE+1
   10 IENM=IENM+1
C
      WRITE(6,51)
      WRITE(6,54)E
      WRITE(6,53)SECOND()
C
C     --- SOLVE THE TWO SYSTEMS, FORWARD & BACKWARD ---
C
      CALL DPPSL (C,NEF,T)
C
C     --- PRINT THE RESULTING COEFFICIENTS ---
C
      WRITE(6,51)
C
      I=1
      J=4
      DO 40 N=2,NMAX
      WRITE(6,55)N,0,T(I)
      ENM(J)=T(I)
      FNM(J)=0.E0
      I=I+1
      J=J+1
      DO 40 M=1,N
      WRITE(6,55)N,M,T(I),T(I+1)
      ENM(J)=T(I)
      FNM(J)=T(I+1)
      J=J+1
   40 I=I+2
C
      WRITE(14)ENM
      WRITE(14)FNM
C
      WRITE(6,53)SECOND()
   51 FORMAT(//)
   52 FORMAT(3(5X,A10,3X,I10))
   53 FORMAT(/,'       CPU TIME =',F10.5)
   54 FORMAT(10E13.5)
   55 FORMAT(2I5,2E26.16)
      STOP
      END
      SUBROUTINE ABCDV (NMAX,NEF,NABCD,NANMPQ,ANMPQ,BNMPQ,CNMPQ,
     1                  DNMPQ,IDIAG,C)
C
C     THIS SUBROUTINE REORGANIZES ANMPQ, BNMPQ, CNMPQ & DNMPQ INTO
C     A SYMMETRIC MATRIX WHICH IS STORED IN AN ARRAY C.
C
      DIMENSION ANMPQ(NANMPQ),BNMPQ(NANMPQ),IDIAG(NEF)
      DIMENSION CNMPQ(NANMPQ),DNMPQ(NANMPQ),C(NABCD)
C
      DO 5 I=1,NEF
    5 IDIAG(I)=I*(I-1)/2
C
      IA=10
      I=1
      J=1
      DO 4 N=2,NMAX
C..M=0
      DO 1 K=2,N-1
C..L=0
      C(IDIAG(J)+I)=ANMPQ(IA)
      I=I+1
      IA=IA+1
      DO 1 L=1,K
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J)+I+1)=CNMPQ(IA)
      I=I+2
```

```
 1    IA=IA+1
C..K=N
C..L=0
      C(IDIAG(J)+I)=ANMPQ(IA)
      IA=IA+4
      J=J+1
      I=1
      DO 4 M=1,N
      DO 2 K=2,N-1
C..L=0
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J+1)+I)=DNMPQ(IA)
      I=I+1
      IA=IA+1
      DO 2 L=1,K
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J+1)+I)=DNMPQ(IA)
      C(IDIAG(J)+I+1)=CNMPQ(IA)
      C(IDIAG(J+1)+I+1)=BNMPQ(IA)
      I=I+2
 2    IA=IA+1
C..K=N
C..L=0
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J+1)+I)=DNMPQ(IA)
      I=I+1
      IA=IA+1
      DO 3 L=1,M-1
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J+1)+I)=DNMPQ(IA)
      C(IDIAG(J)+I+1)=CNMPQ(IA)
      C(IDIAG(J+1)+I+1)=BNMPQ(IA)
      I=I+2
 3    IA=IA+1
C..L=M
      C(IDIAG(J)+I)=ANMPQ(IA)
      C(IDIAG(J+1)+I)=DNMPQ(IA)
      I=I+1
      J=J+1
      C(IDIAG(J)+I)=BNMPQ(IA)
      I=1
      J=J+1
 4    IA=IA+4
      RETURN
      END
```