An overview manual for the

GRAVSOFT

Geodetic Gravity Field Modelling Programs

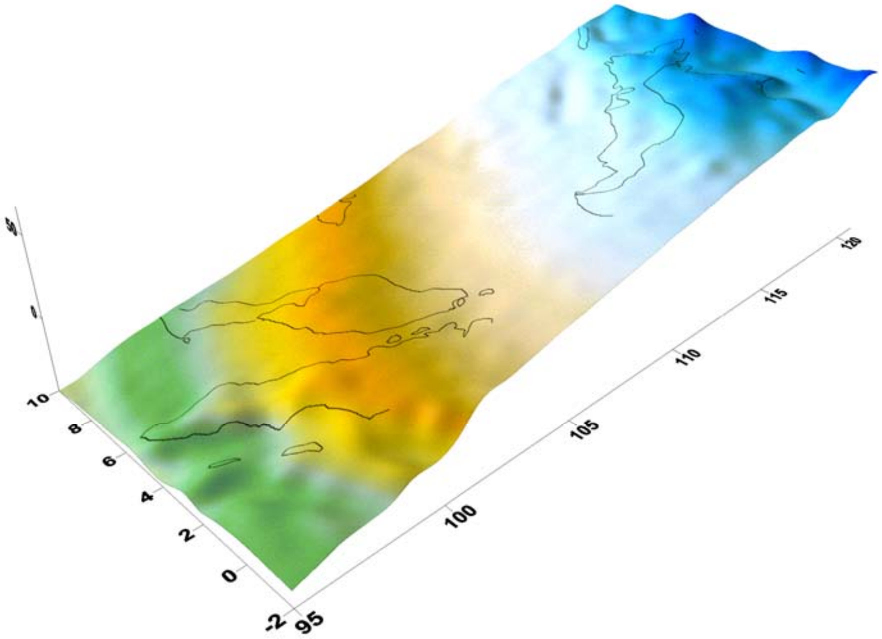
by

Rene Forsberg
National Space Institute (DTU-Space), Denmark
rf@space.dtu.dk

and

C. C. Tscherning
Niels Bohr Institute, University of Copenhagen
cct@gfy.ku.dk

2.edition
August 2008



CONTENTS

1. Background

Software for regional and local gravity field modelling – e.g. geoid determination, determination of deflections of the vertical and recovery of gravity anomalies from satellite altimetry - has been developed since the early 1970's first at the Geodetic Institute, and later at it’s successor Kort og Matrikelstyrelsen (KMS; National Survey and Cadastre of Denmark, now DTU-Space) and the Geophysical Institute (now Geophysics Dept. of the Niels Bohr Institute), University of Copenhagen.

GRAVSOFT consists of a rather large suite of Fortran programs, which have evolved over many years to tackle many different problems of physical geodesy. The roots of the oldest program – the general collocation program GEOCOL – dates back to 1973. Some other modules – such as the terrain effect program TC – dates to the early 1980’s, with Fourier analysis and altimetry programs added mainly in the 1980’s. The Python graphical Interface was added in 2008 on contract to JUPEM, Malaysia.

In 1988 it was decided to streamline and collect the software in a package called GRAVSOFT, mainly by defining some preferred formats (see below) which allowed data to be easily exchanged between the various programs. Since then GRAVSOFT have evolved with addition of a few extra programs (e.g., for spherical FFT), and otherwise errors have been corrected and input changed to a minor degree. As of today distributed versions of GRAVSOFT includes both newer and older program versions. Some programs, such as e.g. satellite altimetry support, have not been upgraded in the general release versions in recent years.

The GRAVSOFT programs are not freeware, and users of the programs sign an agreement that the programs are not to be redistributed. A commercial license is available for parts or all of the programs. For scientific use the programs are usually made available for free.

The different GRAVSOFT programs have been programmed by C. C. Tscherning (Cph. Univ.) – mainly the ”large” collocation program GEOCOL; Rene Forsberg (KMS) – most other programs (terrain, grids, FFT, planar collocation etc.); and P. Knudsen (KMS) – covariance estimation for GEOCOL and altimetry cross-over adjustment. In addition D. Arabelos (University of Thessaloniki) has contributed the basics for a simpler-to-use program for evaluation of spherical harmonics. For an earlier published description of the programs see (Tscherning, Forsberg and Knudsen, 1992).

The explanatory manual pages in the sequel are mainly focussing on the RF-authored programs, which in effect make a complete subset of GRAVSOFT for applications such as geoid determination, downward continuation of airborne data, and the prediction of gravity anomalies from altimetry.

2. Overview of the GRAVSOFT programs

The GRAVSOFT programs do basic operations of physical geodesy, and do basic arithmetic and handling of data files, either in point formats or grids. Table 1 below shows the main programs. A number of minor support programs are not included.

Table 1. The basic GRAVSOFT programmes

Program Group	*Program name*	*Functions*	*Authors*
Selection and reformatting of data	SELECT	Select, thin and/or average data in either point or grid format. May also add noise to data for simulation purposes and produce crude line-printer plots	RF
	FCOMP	Add/subtract, linearly modify or merge point data files with statistics	CCT RF
	GCOMB	Add/subtract/overwrite and recombine grids, include specialized corrections such as geoid-quasigeoid separation	RF
	GBIN	Conversion between ascii and binary grid formats	RF
	G2SUR	Conversion of GRAVSOFT grids to SURFER format	RF
Interpolation and gridding	GEOIP	Interpolation (linear or splines) from grid(s) to points or grid to grid. Also subtract/add or 3-D “sandwich” grid interpolation.	RF
	GEOID	Special binary linear interpolation for geoid use. Includes coordinate transformations and transformation of heights	RF
	GEOGRID	Least squares collocation (Kriging) or weighted means prediction from points to a grid. May also interpolate unknown values in grids, or predict points or profiles.	RF
Spherical harmonics	HARMEXP	Evaluates geoid, gravity and deflections in grids or points (may also be done by GEOCOL, use same subroutine)	DA CCT
Terrain prism integration	TC	Prism integration of terrain effects (topographic, topographic-isostatic, RTM and classical terrain corrections)	RF
	TCGRID	Support program for making average grids and mean terrain surfaces for RTM method in TC	RF
Stokes' integral	STOKES	Evaluate Stokes or Vening-Meinesz integrals by space-domain integration	RF
Fourier methods	GEOFOUR	Planar FFT program with many modes: upward continuation, gravity geoid, geoid to gravity, Molodenskys boundary value problems a.o.	RF
	SPFOUR	Spherical multi-band FFT for geoid determination, upward continuation and terrain effects	RF
	SP1D	1-D spherical FFT for geoid determination. Allows very large grids to be transformed	RF
	TCFOUR	Computation of terrain effects by FFT convolutions in planar approximation	RF
	COVFFT	Estimate 2-D covariance functions by FFT	RF
Least-squares collocation (spherical)	GEOCOL	General least squares collocation on the sphere. Evaluation of spherical harmonic fields.	CCT
	EMPCOV	Finds empirical covariance function from point data sets	CCT
	COVFIT	Fits covariance parameters for use in GEOCOL with using EMPCOV output	PK
Least-squares collocation (planar)	GPCOL	Least-squares collocation with the planar logarithmic covariance model	RF
	GPCOL1	Like GPCOL, computing in moving blocks with overlaps	RF
	GPFIT	Fits planar logarithmic covariance functions to point data	RF
Altimetry support	CRSADJ	Cross-over adjustment (bias and/or tilt) of altimeter data	PK
	ASTACK	Stacking of co-linear altimeter data	RF/PK
Other	POINTMASS	Makes pointmass effects in grid or point formats	RF
	CONVOLVE	Filtering of along-track data, e.g. for airborne gravity tc's	RF

The listed GRAVSOFT programs are usually coming together with a number of smaller programs for special tasks, such as e.g. specialized format conversions (CONV80) or computation of Bouguer anomalies (BA). Using the text bodies of such programs it is relatively easy to program similar tasks in a minimal amount of time.

The GRAVSOFT programs should be used together with plotting programs, such as e.g. GMT or Surfer (Windows), to check data and output. The GRAVSOFT programs usually also go hand in hand with a number of other program groups (GRADJ for static gravimetry; EOTVOS and XING for ship gravimetry; and the AG-suite of programs for airborne gravimetry; and specialized programs for satellite altimetry), none of which are included here.

3. Compilation of GRAVSOFT programs

All the GRAVSOFT programs of Table 1 are in principle self-contained Fortran 77-programs. The programs have been compiled on many different platforms over the years, and should compile with little or no problems on most Unix, Linux or Windows-based systems. Basic program development is currently mainly done under Unix (Sun and Silicon Graphics).

For Windows/DOS both GNU-fortran and Lahey Fortran compilers have been used with success. Note, though, that DOS-extenders needed to run older versions of Lahey Fortran do *not* run properly under Windows XP.

For some of the major programs (GEOCOL, GEOGRID) some compilers may give warning messages (e.g., on poor alignment of common blocks), these messages are usually harmless and just expressions pf some of the strangeness of Fortran and/or old code which never was updated (but probably ran without problems on other compilers).

All program texts contain in the beginning detailed instructions for the different parameters of the program, and in some cases also references to methods.

Ability to read Fortran code is essential for spotting problems and error identification when running GRAVSOFT. Also, since Fortran suffers from the ability to declare array sizes dynamically, the user will often find himself out of array memory, and must increase size of arrays in the program text and recompile. In some programs this can be done just one place, in other programs it must be done throughout the text in all subroutines. *Caution is always advised when doing this, especially for blind search-and-replace editing.*

4. Running GRAVSOFT jobs under Unix and Windows – the JOBB program

The philosophy of the GRAVSOFT programs is – with the exception of GEOCOL – to be modular, and perform computations as a sequence of use of different programs. The output of one program is used as input to another program, and so on (this put quite a strain on the user, to remember all the different file names!). For this the Unix job features are most convenient.

A Unix or Linux job file is made executable by the command “chmod +x jobfile”.

Example 4.1 – a simple GRAVSOFT job:

Running a Unix job to compute spherical harmonics “eigen-2s.bin” and subtract values from a file with gravity “borneo.faa” (the !-lines are comments – such comments can only be used in lines where numbers are read):

```
# first run spherical harmonic
harmexp <<!
eigen-2s.bin
2 1                      ! isys (1: egm,2:champ), mode
eigen2n.gri
eigen2g.gri
0
40 t                     ! the maximal degree
0 10 95 120 .125 .125 0  ! grid specification and height
!
# then subtract values by linear interpolation, output in borneo.dif
geoip <<!
eigen2g.gri
borneo.dif
11 0 0 f
borneo.faa
1
!
```

For Windows users complications arise because of a lack of a similar feature under the “command prompt” which must be used to run GRAVSOFT.

A program “jobb” has been made to make possible the running of Unix-like jobs under the Windows command prompt, accepting e.g. the <<!-option and #-comments. “JOBB” generates a new bat-file “runjob.bat”, which contains all program names and make a set of temporary input files, one for each program. To run type

```
jobb < filename | runjob
```

On Windows2000 (but apparently not on Windows XP) the following will do as well:

```
job filename
```

The Unix-job should then be in file “filename”. Output can be redirected to an outputfile for closer inspection by

```
job filename > outfile
```

5. Running GRAVSOFT from the Python GUI

Most GRAVSOFT programmes have been implemented as a Python application. “Python” is a shareware system providing a more user-friendly interface to the programs rather than the command prompt in Windows. Python is originally developed in Unix (Linux) but ported to Windows, see www.python.org.

The Python program all presents a window, in which default parameters are listed. The Python window is set up with default parameters, which can be changed. When the “save” buttom is pressed an input to the relevant program is saved, and when “run” is pressed the program is executed. The output will be the same as the command prompt version, i.e. it is important to maintain a good understanding of every single program function, with using the GRAVSOFT Python interface (“pyGRAVSOFT”).

The Python interface do not include all program options, and it is not possible to run a sequence of programs (like a command prompt job file). Therefore car should be taken not to close the Python window, until the program run has been completed.

Some new Python modules:

GEOEGM.PY is a module using GEOCOL to compute a spherical harmonic expansion. This includes the use a ultrahigh spherical harmonic expansions (like EGM08, complete to degree and order 2190).

FITGEIOD.PY is a module which does the complete set of runs for fitting a gravimetric geoid to GPS control, cf. Example A7. “Fitgeoid” runs the sequence of “GEOIP” (difference GPS/levelling minus geoid), “GEOGRID” (grid geoid corrections) and “GEOIP” (add gridded geoid corrections to obtain final geoid). The input parameters for “fitgeoid” corresponds to GEOGRID. The GPS levelling data must be in a file of format

```
number, lat, lon, height, geoid value
```

System setup for running py-GRAVSOFT smoothly

All Python applications are of type “.py”. The application will run in any directory, provided the “PATH” parameter is properly set in the windows, so both the PyGRAVSOFT executables and the Python code can be found. In this case just the program name must be typed, i.e. “geoid.py”.

To set the correct path, do the following (administrator rights required):

Windows – Control Panel – System – Advanced – Environment variables
Extend PATH= with the directory for pyGRAVSOFT:
PATH = ...; c:\python25; c: \pygravsoft; c:\pygravsoft\bin

Examples of sample Python runs are shown in Appendix C.

6. GRAVSOFT basic data formats

A recommended standard data format makes the use of the program suite easier:

6.1 Spherical harmonic coefficients

Coefficients are stored in a simple file with the coefficients stored either in txt format (free format in “harmexp”) or in a direct binary format as

n, m, Cnm, Snm (optional more data on same line)
….

```
Example:
2  0 -4.8416902576341E-04  0.0000000000000E+00
2  1 -2.6064871543930E-10  1.4150696890028E-09
2  2  2.4393057664421E-06 -1.4002827713200E-06
3  0  9.5709933625445E-07  0.0000000000000E+00
…
```

The binary format of the potential coefficients speeds up the reading. To convert a file with spherical harmonic coefficient from or to binary format use the support program “potbin”.

6.2 Point data format

Data on individual points, e.g. a free-air anomaly value and its standard deviation, is stored in list format as free format, with lines

id, φ, λ (degrees), h, data1, data2, ...

```
Example:
10180   4.21974   113.77524  1578.14   16.61  2.0
10200   4.21127   113.76625  1578.17   17.61  2.0
10220   4.20248   113.75748  1578.18   17.51  2.0
10240   4.19376   113.74859  1578.20   16.71  2.0
```

The first column must always be integer identifier, the follows lat, lon (degrees) and height and one or more data. The height is sometimes understood as data number zero. Some programs may also operate with latitude and longitude replaced by UTM Northing and Easting. Unknown data values may be signalled by “9999” (or higher) – but not all programs are treating the unknown data option.

6.3 Grid data (text format)

GRAVSOFT grid data are stored rowwise from north to south, like you would read the values if they were printed in an atlas. The grid values are initiated with label of latitude (ϕ) and longitude (λ) limits and spacing, the follows the data in free format:

φ_1 , φ_2 , λ_1 , λ_2 , $\Delta\varphi$, $\Delta\lambda$

d_{n1} d_{n2} . . . . d_{nm}
. . . . . .
. . . . . .
d_{11} d_{12} . . . . d_{1m}

```
Example (a grid in Borneo with 1’ spacing in lat and lon:
 0.000000    9.000000  106.000000  121.000000  0.01666667  0.01666667

-11.250   -11.060   -10.662   -10.053    -9.219    -8.151    -6.546
```

```
 -5.083    -3.253    -1.366     0.450     2.060     3.403     4.041
  4.368     4.405     4.255     4.016     4.051     3.850     3.770
  3.694     3.576     3.358     3.169     2.371     1.417     0.373
 -0.671    -1.641    -2.321    -3.065    -3.582    -3.927    -4.125
....
```

Each east-west row must be starting on a new line. Unknown data may be signalled by "9999". The grid label defines the exact latitude and longitude of the grid points, irrespectively whether the grids point values or average values over grid cells. The first data value in a grid file is thus the NW-corner (ϕ_2, λ_1) and the last the SE-corner (ϕ_1, λ_2). The number of points in a grid file is thus

$$nn = (\varphi_2-\varphi_1)/\Delta\varphi + 1$$
$$ne = (\lambda_2-\lambda_1)/\Delta\lambda + 1$$

Note that in fortran integers are truncated, not rounded, so statements in programs typically will have a half-unit added (+ 1.5) to secure correct rounding of integers nn and ne.

Generally it is recommended to use longitudes in range –180 to 180, but 0 to 360 will also be ok in most cases. λ_2 should always be greater than λ_1.

For some applications (e.g. 3-D “sandwich” interpolation or grids of vertical deflections) more than one grid may be in a grid file.

Grids may be in UTM projection. In this case latitude and longitude must be replaced by Northing and Easting (in meter), followed by an ellipsoid number defining the semi-major axis and flattening (1: WGS84; 2: Hayford-ED50, 3: Clarke-NAD27, 4: Bessel ellipsoid ..) and UTM zone number. Note that the programs accepting UTM grids will check by the magnitude of the latitude and longitude label information, if values larger than 360 is read, an UTM grid is assumed to be in the file.

```
Example of a UTM grid in zone 32, wgs84 ellipsoid used

6305000 6345000 505000 545000 10000 10000
1 32

 60 200 110 200 200
101 201  40  30  10
 40  20  10  20  10
 30  10   0   0   0
 15   0   0   0   0
```

6.4 Grid data (binary)

Binary grid data are stored in exactly the same way as the grid text files, but in real*4 binary format, thus saving roughly half the space. The binary grid formats allows the direct-access to specific rows in the grid data, and are thus much faster to use as well.

The binary grid data are stored in internal records of 16 data values, with the first record contaning the grid header and a special code (777) allowing programs to check if grids are in ascii og binary formats (some programs like GEOIP and GCOMB will work with either binary or text grids). Binary grids can generally not be moved between different operating systems or even compilers.

To convert between text or binary grids, use the program “GBIN”:

```
Example of grid conversion(interactive)

gbin
borneo.gri
borneo.bin
1
```

This will convert the grid file “borneo.gri” to the binary format “borneo.bin”. The opposite operation will be (use 2 for real format, 3 for integer format of data):

```
gbin
borneo.bin
borneo.gri
2
```

We generally recommend that text grid files are named with extension “ .gri”, since many plotting programs such as surfer use the default ending “.grd”. To e.g. convert a GRAVSOFT grid for plotting in surfer use the G2SUR (or G2SURB) program

```
Example of conversion of gravsoft grid to surfer format

g2sur
borneo.gri
borneo.grd
```

The G2SUR program reorders sequence of rows to match surfer format, transforms unknown values, and adds appropriate header. All data need for this purpose to be stored in RAM memory, so for large grids changes in the dimension statements in the program may be needed.

7. Some notes on the use of FFT programs

The GRAVSOFT Fourier domain programs in either the plane or the sphere use the discrete Fast Fourier Transform algoritm to approximate the various continous integrals of physical geodesy. The continous two-dimensional Fourier transform has the form

$$F(g) = F(k_x, k_y) = \iint g(x,y) e^{-i(k_x x + k_y y)} dxdy$$

$$F^{-1}(G) = g(x,y) = \frac{1}{4\pi^2} \iint G(k_x, k_y) e^{i(k_x x + k_y y)} dk_x dk_y$$

where g is the data, G the spectrum and k the wavenumbers. By the nature of the FFT the data must be assumed to be periodic and given on a finite grid interval, and the continous transform integral is each direction approximated by the fundamental discrete Fourier transform

$$G(n) = \sum_{k=0}^{N-1} g(k) e^{-2\pi i \frac{kn}{N}} \qquad for\ n = 0,1,...,N-1$$

$$g(k) = \frac{1}{N} \sum_{n=0}^{N-1} G(n) e^{2\pi i \frac{kn}{N}} \qquad for\ k = 0,1,...,N-1$$

The approximation of the continous with the discrete Fourier transform gives rise to a number of problems, the most serious of which is the *periodicity effect*. This can traditionally be avoided by windowing of data (e.g., applying a cosine taper to the data, where data close to the edge of the grid is multiplied by a function w decaying from 1 to 0 as a cosinus curve). Alternatively – and usually better in physical geodesy – is the use of *zero padding*.

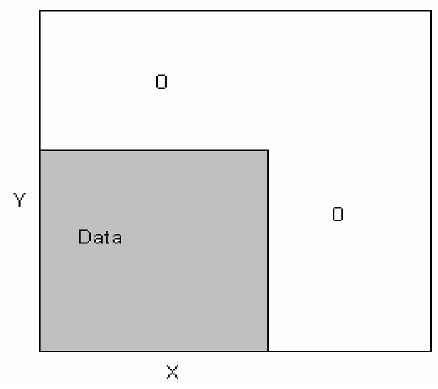


Fig. 6.1 Zero padding of grid

Zero padding is implemented for the main Fourier programs (GEOFOUR, SPFOUR) but not for the older programs like TCFOUR (padding can be done by repeated calls of GCOMB instead). The zero padding is typically controlled by defining the number of points along the grid margin where data should smoothly go to zero.

The FFT algoritm "FOURT" used in all programs are a "mixed-radix" algorithm originating from NORSAR (Norwegian Seismological Array), which is again based on old MIT software.

The FFT algorithm is based on the prime factorisation of the number of points. The number of points in a grid direction (either north or east) is internally in the algorithm factored into primes

$$N = n_1 \cdot n_2 \cdot ... \cdot n_q$$

The algorithm works quickest if the primes n_i are small numbers, ideally quickest if N is a power of 2 (i.e., 128, 256, 512 etc.). Since geographic grids tends to have number of grid points like 60, 120, 240 .. etc., these numbers are "good" for FFT with the mixed-radix algorithm as well (e.g. $60 = 2^2 \cdot 3 \cdot 5$), where as the odd numbers like 61, 121, 241 .. are less well suited.

Because of the dependence of the prime factorisation in program performance, all the GRAVSOFT programs require the user to specify the number of points to be transformed in the north and east directions. The user must thus manually compute the number of points in a given grid. In the main programs (GEOFOUR and SPFOUR) a subgrid will be analysed if the number of wanted points transformed is *less* than the actual number of points; if the wanted number is *larger*, zero-padding will automatically be performed on the extended grid. The output grid will, however, never be larger than the original grid (zero-padded border zones never output), but the internal memory dimensions required must match the full-zero padded grid.

Although the FFT algorithm should work for all number of grid points, only *even* numbers have been tested, and should be the only values used.

The FFT programs have generally been optimised to restrict the size of arrays needed. A basic complex array containing the complete data grid plus the zero-padding is always needed; this memory space is reused in some programs, and other tricks like doing two (real) Fourier transforms in one complex transform is also utilized. The logical structure in some of the FFT programs is thus rather complex.

Adding to this complexity is also the inherent *periodic nature* of the 2-D Fourier transform. The FFT spectrum is a periodic function, which is computed from the wavenumber 0 to 2k$_N$ where k$_N$ = π/X is the Nyquist frequency, i.e. the highest wavenumber recoverable from a function sampled at the (grid) spacing Δx. Due to the periodicity the wavenumbers above the Nyquist wavenumber k$_N$ correspond to the negative wavenumbers, and in two dimensions wavenumbers thus "map" into the resulting array as shown below in Fig. 6.2.

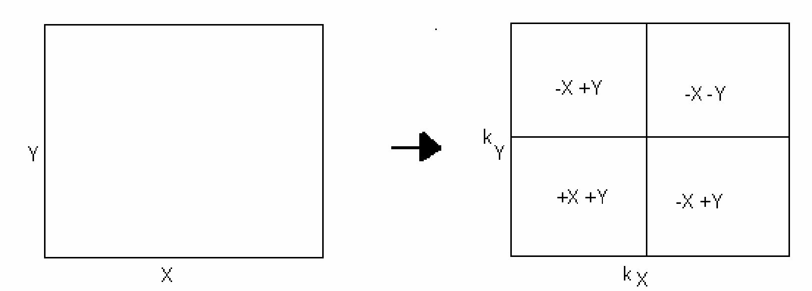


Fig 6.2 Periodic nature of 2-D Fourier transforms: An X-Y grid is transformed into a similar grid in the wavenumber domain, with the positive and negative wavenumbers located as shown on the left.

In the covariance program COVFFT the output is shifted so that the 2-D covariance and power spectrum output is centered with (0,0) in the center of the grid, otherwise no shift or centering of the Fourier arrays is performed (rather, the frequency filters are applied directly on the FFT grids, taking the symmetries of Fig. 6.2 into account). This add to the efficiency of the Gravsoft FFT software.

In most of the main applications of FFT in physical geodesy, the main use of FFT is to provide a fast convolution, utilizing the convolution theorem

$$f*g(x,y)=\iint f(x-x',y-y')g(x',y')dx'dy'$$
$$\Rightarrow$$
$$F(f*g)=F(f)\cdot F(g)$$

An example of such a formula is Stokes' formula for the (quasi)geoid ζ in planar approximation

$$\varsigma(x,y) \;=\; \frac{1}{2\pi}\iint\frac{\Delta g(x',y')}{\sqrt{(x-x')^2+(y-y')^2}}\,dx'\,dy'$$

In these cases the mapping of wavenumbers play little role, since only space-domain results are of relevance. However, periodicity effects inherent to FFT still occur in full force.

The FFT evaluation of many physical convolution integrals such as the planar Stokes' integral above involves a *singular* kernel at distance zero. In most of the FFT programs an "inner zone effect" are computed separately from the FFT operations, taking into account the integral a small circular zone of radius half the grid spacing. However, it is not implemented fully, and some caution are advised for special problems, such as upward contination of data to heights much smaller than the grid spacing.

8. Some notes on remove-restore terrain reductions

The utilization of digital terrain models is essential for obtaining good gravity field modelling results in mountainous areas. The GRAVSOFT programs TC, TCFOUR and SPFOUR will produce various kinds of terrain effects by Newtonian integration of density effects in either space og wavenumber domain.

Digital Elevation Models (DEM's) must be given in one or more grids, and by convention heights above sealevel signal land (default density 2.67 g/cm^3) and negative heights signal ocean (default density –1.64 g/cm^3). This means that land areas below zero will not be computed correctly.

A special variant of terrain reduction is the *residual terrain model (RTM)* where only the topographic irregularities relative to a smooth mean height surface is taken into account. This reduction has the advantage that the effects, e.g. on geoid or airborne computations are quite small, in principle corresponding to consistent use of a spherical harmonic reference field when the right resolution of the mean height surface is selected.

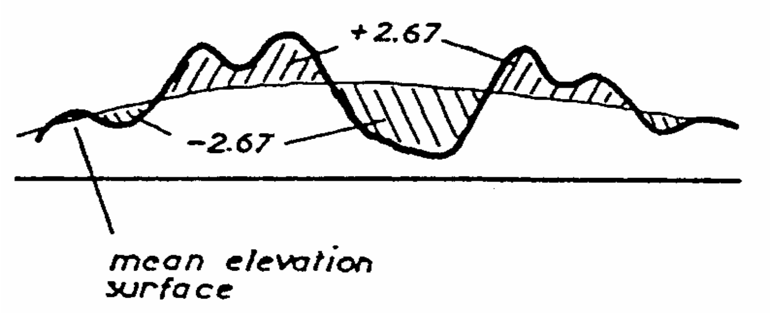


Fig.7.1. Principle of RTM mean height surface and residual topography density anomaly.

The details of the RTM method is outlined in Forsberg (1984). It should be pointed out that the GRAVSOFT programs contain approximations ("harmonic correction" in TC for points below reference level, and "Bouguer plate" corrections in GEOIP) so that rigorously correct results may not be obtained, especially for high-resolution reference surfaces. Caution should in other words be exercised; the same is true for use of RTM reductions over the ocean (should work in TC after recent corrections by A. Arabelos).

The generation of the mean height grid surface can be done with either TCGRID (making block averages of a DEM; by SELECT in mode 3; or by a special moving-average program GFILT, otherwise not mentioned here). A suitable resolution for a spherical harmonic model to degree 360 would be somewhere between 50 and 100 km, depending on the actual accuracy of the spherical harmonic model in the region of interest. Moving averages should in this case be generated from a DEM model no more coarse than 10-20 km spacing to have a reasonably "smooth" representation of the reference grid.

In general we recommend the terrain reductions (and "EGM" spherical harmonic effects) to be applied with GRAVSOFT in a remove-restore fashion, i.e. formally splitting the anomalous potential into three parts

$$T = T_{EGM} + T_{DEM} + T_{residual}$$

In the case of e.g. computing a geoid from gravity anomalies, this corresponds to subtracting EGM and terrain effects from the observed free-air gravity anomalies; performing a conversion to geoid (e.g. by FFT); and then “restoring” the geoid effects of terrain and EGM, in priciple doing the *“remove-predict-restore”* operations as follows

$$\Delta g_{res} = \Delta g - \Delta g_{EGM} - \Delta g_{RTM}$$

$$\zeta_{res} = \frac{1}{\gamma} F^{-1}[\, F(S) F(\Delta g_{res})]$$

$$\zeta = \zeta_{res} + \zeta_{EGM} + \zeta_{RTM}$$

The subtraction/addition and associated statistics is built into some central programs (mainly GEOCOL and TC), otherwise use must be made of reformatting and data manipulation programs such as GCOMB (addition/subtraction of grids), GEOIP (points plus/minus grids) or FCOMP (add/subtract point datafiles).

The use of other types of methods than “remove-restore” are perfectly possible with GRAVSOFT; it is just a matter for the users to set up the adequate sequence of jobs. GCOMB contains e.g. special modes to make first-corrections between geoid and quasigeoid, or to compute Helmert condensation effects.

9. References

The theory behind the GRAVSOFT programs is contained in a large number of journal papers. For basic references see the personal homepages of the authors at http://research.kms.dk or http://www.gfy.ku.dk/~cct. The following publications contains some selected papers with details related to implementation in the GRAVSOFT programs.

Forsberg, R.: A Study of Terrain Reductions, Density Anomalies and Geophysical Inversion Methods in Gravity Field Modelling. Reports of the Department of Geodetic Science and Surveying, No. 355, The Ohio State University, Columbus, Ohio, 1984.

Forsberg, R.: Gravity Field Terrain Effect Computations by FFT. Bulletin Geodesique, Vol. 59, pp. 342-360, 1985.

Forsberg, R. and C.C. Tscherning: The use of Height Data in Gravity Field Approximation by Collocation. J. Geophys. Res., Vol. 86, No. B9, pp. 7843-7854, 1981.

Forsberg, R.: A new covariance model for inertial gravimetry and Gradiometry. Journ. Geoph. Res., vol. 92, B2, pp. 1305- 1310, 1987.

Forsberg, R. and M. G. Sideris: Geoid computations by the multi-band spherical FFT approach. Manuscripta Geodaetica, 18, 82-90, 1993.

Forsberg, R.: Terrain effects in geoid computations. Lecture Notes, International School for the Determination and Use of the Geoid, Milano, Oct. 10-15. International Geoid Service, pp. 101-134, 1994.

Knudsen, P.: Estimation and Modelling of the Local Empirical Covariance Function using gravity and satellite altimeter data. Bulletin Geodesique, Vol. 61, pp. 145-160, 1987.

Schwarz, K.-P., M. G. Sideris, R. Forsberg: Use of FFT methods in Physical Geodesy. Geophysical Journal International, vol. 100, pp. 485-514, 1990.

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

Tscherning, C.C.: Computation of covariances of derivatives of the anomalous gravity potential in a rotated reference frame. Manuscripta Geodaetica, Vol. 18, no. 3, pp. 115-123, 1993.

Tscherning, C C: Gravity field modelling with GRAVSOFT least-squares collocation. Lecture Notes, International School for the Determination and Use of the Geoid, Milano, Oct. 10-15. International Geoid Service, pp. 101-134, 1994.

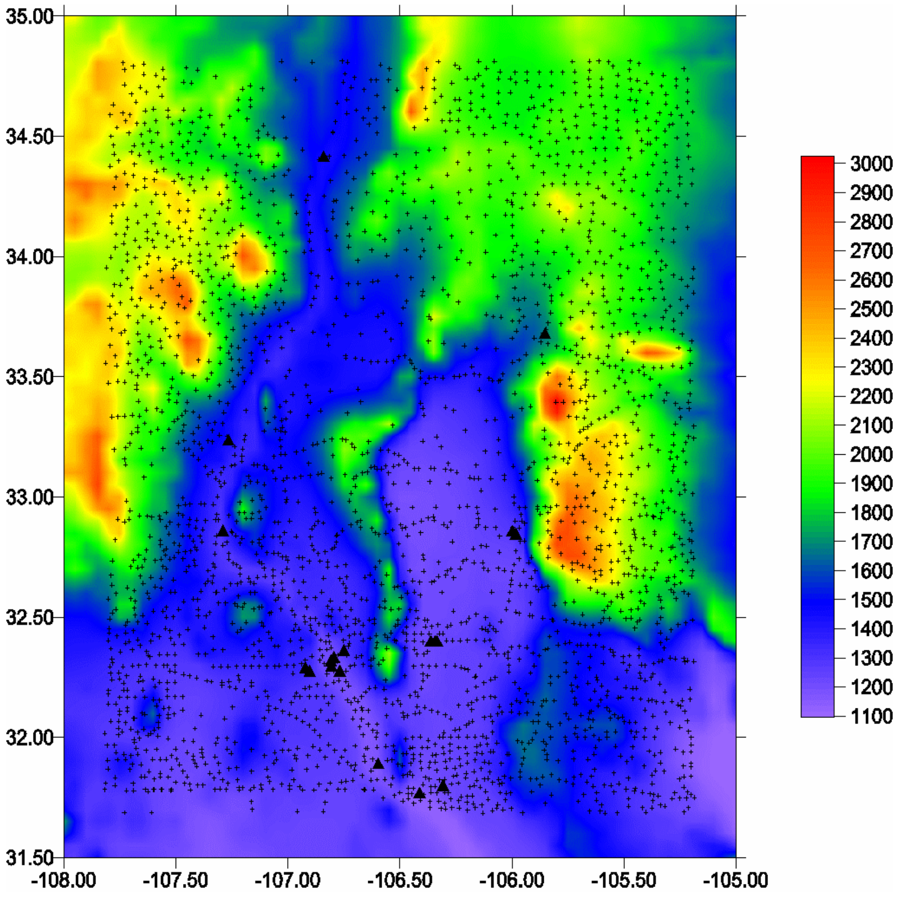
Appendix A

Some cookbook examples – from the IAG International Geoid School

The practical cookbook examples below are based on a data set from an area in New Mexico, USA. The purpose of the present examples are to illustrate the computation of RTM terrain effects, and to compute a preliminary geoid and quasigeoid using the GRAVSOFT programs. You should make sure you try to read the documentation in the program texts – sample input files are given below for some of the cases.

The New Mexico basic data consist of various data sets:

nmdtm	*Basic 30" x 30" height data set, covers area 31.5-35 N, 108-105 W with some rows along the border missing.*
nmfa	*Free-air anomalies in format (statno, lat, lon, height, fa)*
nmba	*Bouguer anomalies (statno, lat, lon, height, ba)*
nmdfv	*Deflections of the vertical (statno, lat, lon, height, ksi, eta)*
nmgpslev.ha	*A set of 20 GPS levelling point height anomalies (.n is geoid heights)*
nmegm96.n	*EGM96 geoid heights in grid form (computed by "geocol" or "harmexp")*
nmegm96.g	*EGM96 gravity anomalies in grid form*
nmegm96.dfv	*EGM96 deflections*



New Mexico height, gravity and GPS levelling data (triangles)

Example A1 – Prepare terrain grids.

Prepare terrain grids for use in TC and TCFOUR.

1) Run SELECT to make a basic 0.05 deg average height grid. New grid label: 31.5 35 -108 -105 .05 .05. Call result *nmdtm5*. Note: SELECT is a very general reformatting and selection/thinning/averaging program, and may be used to read many DEM data formats into GRAVSOFT format.

```
Sample input file
select <<!
nmdtm                 ! name of input file
nmdtm5                ! name of output file
3 7 1                 ! 3=make grid, 7=input grid, 1=no of data
31.5 35 -108 -105 .05 .05
!
```

2) Run TCGRID to average nmdtm5 into a reference height grid. To obtain optimal smoothing we would like to use a reference height grid resolution around 100 km. TCGRID does two operations: It first makes an average grid (like select), and then filters the averaged grid with a moving average operator. Specify the area, mean factors 2 2 (will give a 0.1 x 0.1 deg grid), and average factors 9 9 (will give a resolution of 0.9 deg or nearly 100 km in latitude). Call output grid *nmhref*. NB: Check this grid for 9999-values (may occur at edges), rerun with a slightly smaller area if such values are found.

```
tcgrid <<!
nmdtm5
nmdtmref
0 0 0 0 0      ! dummy values, may be used to select smaller area
2 2 9 9        ! average 2 x 2 cells, then do 9 x 9 moving window
!
```

Check that the “nmdtmref” reference height grid are smaller, and plot.

Example A2 - Make terrain effects by TC prism integration

Because the gravity data set is very big, a subset of gravity data is selected for a test run with TC. Gravity RTM effects for all points are computed with TCFOUR in exersize 3. Alternatively just let TC run for a little while – “rd.job” will do both as terrain reduction and a reduction for EGM96.

1) Make subset of nmfa by running SELECT with pixel select (mode 1) in area 33 34 -107 -106 and pixel gridspacing .1 .1. Pixel select means that the point closest to the nodes in the grid is selected (if the grid also contains standard deviations it is possible to select the most accurate point instead). Call output *nmfasel*. Sample input file to select in this case:

```
select <<!
nmfa            (name of input file)
nmfasel         (name of output file)
1 1 1          ! 1=select points, 1=input point format, 1=no of data
33 34 -107 -106 .05 .05
!
```

2) Run TC to compute RTM-reduced gravity data. Compute gravity disturbance, and let the innerzone be modified to fit the topography. Note smoothing of gravity field before and after terrain reduction – statistics is output by TC. Call output file *nmfasel.rtm.*

```
tc <<!
nmfasel
nmdtm
nmdtm5
nmdtmref
nmfasel.rtm
1 4 1 3 2.67      !1=dg, 4=rtm, 1=station on topo, 3=subtract results
31.5 35 -108 -105 !limits in lat/lon of complete area
12 999            !  r1, r2 in km
!
```

Optional - 3) Terrain-reduce deflections of vertical. Produce terrain-reduced set by TC. Deflections should be computed without modifying topography (IZCODE=0). Note smoothing. Put reduced deflections of the vertical in file *nmdfv.rtm.* Note the different use of "izcode": because gravity terrain effects depend strongly on height (through the Bouguer term)

```
nmdfv
nmdtm
nmdtm5
nmdtmref
nmdfv.rtm
2 4 1 3 2.67          ! control,1=dg,4=rtm,1=station on topo,3=subtr
31.5 35 -108 -105     ! limits in lat/lon of complete area
12 999                ! r1, r2 in km
```

Example A3: Terrain effects by Fourier methods

RTM terrain effects can be directly computed by TCFOUR in mode 4. A sample input file is given in "tcfour.inp":

```
tcfour <<!
nmdtm
nmdtmref
nm_rtm.gri
4 t 999.9 0       ! 4=rtm terrain effects, 0 to 999 km
0 0 420 360       ! 0 0 use SW corner of grid, 420 360 no of points
!
```

This give directly RTM effects in file "nm_rtm.grd", which will contain the 1-km RTM effects. These need subsequently to be subtracted from the observations. This is done in the following way:

1) First reduce for EGM96:

```
-> geoip
   nmegm96.g          (file name for EGM96 gravity grid)
   pip                (output file - use temporary name)
   11 0 0 f           (code for difference file minus grid)
   nmfa               (file name of gravity data)
   1                  (location of data in file - first column)
```

2) Now subtract terrain effects just computed:

```
-> geoip
   nm_rtm.gri         (gridded RTM effects just computed)
   nmfa.rd            (output file - fully reduced gravity data)
   11 0 0 f
   pip                (temporary file from before holding fa-ref)
   1
```

The reduced data are now ready for gridding, FFT and final geoid!

Example A4: Geoid restore effects by FFT

For use in the restore process we also need the geoid restore effects. We do this before starting the gridding process. It is here done by TCFOUR mode 5, sample job is "tcfour_z.inp".

```
      tcfour <<!
      nmdtm5
      nmdtmref
      nm_z_rtm.gri
      5 t 999.9 0       ! 5=geoid rtm terrain effects, 0 to 999 km
      0 0 70 60         ! 0 0 = use SW corner of grid, 70 60 = no of points
      !
```

This will generate a grid file "nm_z_rtm.grd" containing the effects. Geoid restore effects might also be done by TC prism integration directly at wanted points. A linear or 3rd order expansion has recently been implemented in spfour as well, presenting a spherical FFT alternative for larger areas.

Excersize A5: Now use the computed terrain effects to make a gravimetric geoid

Now we do a complete New Mexico geoid on a 5 km grid, and compare to GPS levelling after EGM96 and terrain effects from excersize 4 are restored. Here follows the steps - names changed slightly relative to notes:

1) Gridding by GEOGRID - sample input in "geogrid.inp"

```
     geogrid <<!
     nmfa.rd
     nmfa_rd.gri
     nmfa_rd.err
     1 1 5 0 1      ! control: 1 1 = std format, 5 pts/quadrant, 0 1 = std
     25 1.0         ! correlation length (km), noise (mgal)
     1 50.0         ! 1=make normal grid, use data within 50 km
     31.5 35 -108 -105 .05 .05
     !
```

-> results in grid file "nmfa_rd.grd" (reduced data in grid) and "nmfa_rd.err" (collocation error estimates).

2) Run spherical FFT with 100pct zero padding - sample job "spfour.inp". Be aware of the version of spfour: Newest version has an extra integer for kernel modification.

```
spfour <<!
nmfa_rd.gri
nm_z_rd.gri
1 t 1 0 999 0     ! 1=stokes,t=subtract mean,1 band,kernel 0 to 999 km
0                 ! wong-gore kernel modification
0 0 140 120 0     ! no of points (140 120 gives 100 pct zero padding)
!
```

-> results in grid file "nm_z_rd.grd" contains now reduced geoid values.

3) Now restore terrain effects by adding output grid from Example A4 and
the output from SPFOUR:

```
gcomb <<!
nm_z_rtm.gri      (first input file)
nm_z_rd.gri       (second input file)
pip               (output file - temporary)
2 2               (code for add grid files)
!
```

4) Restore EGM96 by GEOIP mode 16 which can add grids with different spacing:

```
geoip <<!
nmegm96.n         (EGM96 geoid grid)
nm_z.gri          (the final New Mexico geoid in grid format)
16 0 0 f          (code for add+interpolate grids)
pip
!
```

Now you have made the final geoid - it's in the form which can be used directly by GPS users .. for a check of the whole thing try and compare to the GPS-levelling points in New Mexico in file "nmgps". This you do by:

```
geoip <<!
nm_z.gri          (name of geoid grid you want to interpolate)
pip               (output file = difference GPS-grav geoid)
11 0 0 f          (11: code for make difference point-grid)
nmgpslev
1
!
```

What you have now is a *gravimetric* quasigeoid – it is in a global datum, and there will be a mean offset to a local datum – especially due to the use of a tide gauge to define the height system. Two more steps are therefore often used in practice – conversion from geoid to quasigeoid, and a geoid fitting to GPS.

Example A6 – Converting a quasigeoid to a geoid

The unix-like job below will do the trick – it grids the Bouguer anomalies, compute the N-ζ difference by the simple formula (12) in a grid, and finally add this difference to the quasigeoid grid, to get a final geoid grid “nm_n.gri ”. The job file below is in “zeta2n.job”.

```
# grid bouguer anomalies
geogrid <<!
nmba
nmba.gri
nmba.err
1 1 5 0 1      ! 1 1 = std format, 5 pts/quadrant, 0 1 = std
25 1.0         ! correlation length (km), noise (mgal)
1 50.0         ! 1=make normal grid, use data within 50 km
31.5 35 -108 -105 .05 .05
!
pause
# compute grid of n-zeta differences
gcomb <<!
nmba.gri
nmdtm5
pipnz.gri
9 2
!
pause
# add to quasigeoid to get geoid grid
gcomb <<!
nm_z.gri
pipnz.gri
nm_n.gri
2 2
!
```

Example A7 – Fitting of a geoid to GPS control … with some general remarks

The outcome of the different geoid computation methods, based on gravity, using methods such as FFT, collocation or Stokes ring integration, all give the *gravimetric geoid* - which in principle refers to a global reference system, i.e. global center of mass, average zero-potential surface etc. Such a geoid may be offset by up to 1-2 m from the apparent geoid heights determined from GPS and levelling by

$$N^{GPS} = h^{GPS} - H^{levelling}$$

The reason for the difference is mainly the assumption of zero-level: Levelling zero refers to local or regional mean sea-level, which is different from the global zero vertical datum due to the sea-surface topography. Also, error in long-wavelength geopotential models, underlying a local geoid estimation, may yield offsets of up to 1 m or so.

Since most countries are interested in using GPS to determine heights in a local vertical datum, to be consistent with existing levelling, there is a need to taylor the gravimetric geoid to the local level. This might be easily done in GRAVSOFT using the collocation gridding module GEOGRID. In principle the difference between gravimetric and GPS geoid

$$\varepsilon = N^{GPS} - N^{gravimetric}$$

is gridded "softly" using collocation (possibly with parameters, e.g., taking out a constant mean value and modelling the residuals as a stochastic process), and then the final geoid is obtained by

$$N^{draped} = N^{gravimetric} + \varepsilon^{grid}$$

The draped geoid will be consistent with the GPS data to the degree of the assumed standard deviations for the observed ε-values. A typical used error value would be a few cm, dependent on the assumed levelling and GPS errors. In the GRAVSOFT case the covariance function of ε is modelled by a second-order Gauss-Markov model

$$C_{\varepsilon\varepsilon} \;=\; C_0(1+\alpha s)e^{-\alpha s}$$

where s is the distance and α a parameter which determines the *correlation length. T*he variance C_0 is determined automatically, whereas the correlation length must be specified by the user. A useful correlation length depends on the spacing of the GPS data, with values around 50-100 km being typical. This means that at distances significantly further away from any GPS point than the collocation correlation length, the draped geoid would be identical to the gravimetric geoid (except for as possible estimated constant offset or trend parameter).

If the geoid differences show a strong trend, a combination of both trend estimation and collocation signal might be useful. In this case a "4-parameter" model of form

$$\varepsilon = \cos\varphi\cos\lambda\, a_1 + \cos\varphi\sin\lambda\, a_2 + \sin\varphi\, a_3 + a_4 + \varepsilon'$$
$$\hat{\varepsilon} = C_{xx}^{-1}C_{sx}\{\varepsilon'\}$$

has often proven itself useful, where the parameters a_1 to a_4 corresponds to the geoid effects of a Helmert transformation (change of origin and scale; rotations has no first-order effects on the geoid). The combined model is built into GEOGRID.

Great care should be taken in using the draping method, as a good gravimetric geoid may be badly distorted by poor GPS-levelling data. This is especially true if the region of draping has levelling based on different tide gauges, where the sea-level has been constrained. This yields effectively a systematic error in ε due to differences in the permanent sea-surface topography at the tide gauges. Another typical source of errors in ε is movement of points, as difference in the epochs of levelling and GPS - often decades apart - will yield a direct error in the estimated GPS geoid heights.

The fitting of a geoid may be done by the following GRAVSOFT setup:

```
# job to fit geoid
# first find differences to gravimetric geoid
geoip <<!
nm_z.gri
pip.dif
11 0 0 f
nmgpslev.ha
1
!
# then grid difference
geogrid <<!
pip.dif
pip.gri
pip.err
1 1 8 5 1      ! geogrid control: 1 1 = std format, 8 pts/quadrant,
trend 5
50 0.02        ! correlation length (km), noise (m)
```

```
1 50.0         ! 1=make normal grid, use data within 50 km of area
wanted
31.5 35.0 -108 -105 .05 .05
!
# add back gridded differences to geoid
gcomb <<!
nm_z.gri
pip.gri
nm_z_fit.gri
2 2
!
```

The final geoid fitted to GPS is in the file “nm_z_fit.gri”.

Appendix B

Alphabetic listing of GRAVSOFT program instructions

In the sequel the different listed GRAVSOFT programs are documented to a various degree. Some programs are very complex (e.g., GEOCOL) and the user will need to go through the code to try and attempt to give correct input.

For some of the less frequently used programs, some format descriptions and checks need to be resolved in the original program tests as well.

The desciptions below include time history of the program development. Future versions of this manual will hopefully provide a little better documentation!

ASTACK	Stacking of co-linear altimeter data (version not used recently but should be ok) Altimeter data format is special, see program text for details (also used in "select")

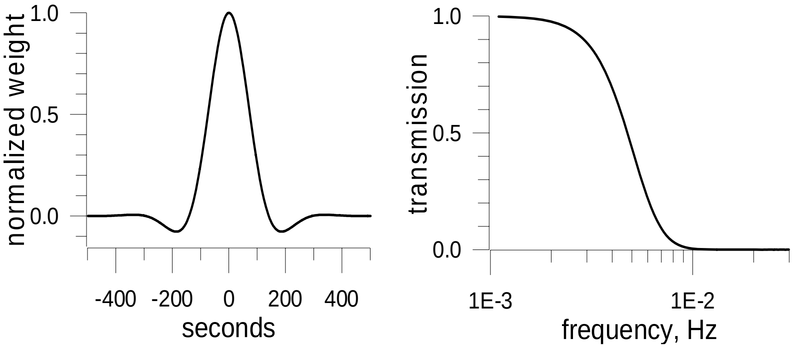
```
                          A S T A C K

  Program to stack colinear tracks of altimeter data (e.g. Geosat).
  the observations are interpolated on a time grid defined by
  the best track and minimum and maximum times relative to a reference
  time associated with each track. The tracks are merged in a free
  adjustment using 1 cy/rev cosine and sine minimizing the differences.
  this version of the program uses a direct access file for data sorting.
  the maximum id-no of the track must be adjusted in 'maxno'

  Input:

  ifile
  ofile
  mode, ntrack, lcov,
  rfi1, rfi2, rla1, rla2

  with

  mode = 1: GEOSAT 17 day repeat
         2: ERS-1 3 day
         3: ERS-1 34 day
  ntrack    max number of tracks to be stacked (i.e., max no of repeats)
  lcov      true is statistical analysis wanted

  rfi1, rfi2, rla1, rla2 area of stacking


  (c) programmed by rene forsberg, national survey and cadastre (denmark)
  march 1989, university of new south wales vax
  modified by Per Knudsen and Maria Brovelli, Nov. 1990.
  further modified by rf, univ of calgary, march 1993
  updates, rf feb 94 kms
```

CONVOLVE	Filtering of along-track data, e.g. for airborne gravity terrain effects. The filtering function must be given in a separate files as the impulse response of the filter.

```
                         C O N V O L V E

  This program convolves a file with track data and time(decimal julian
  date), with a kernel (impulse response) given in a separate file.
  Maximal kernel value is assumed to represent t=0, and should
  be approximately in the center of the file.

  Input:

  datafile name  (format: trackno*1000+no,lat,lon,h,dat,time)
  impulseresponse file  (format: equidistant values, 1 pr line)
  outputfile
  dtfilt, dtdat  (time spacing of filter coef and data, seconds)
                 (if dtdat = 0 the data time will be read from the
                  data file)

  RF/KMS, Greenland aug 92
  last updated sep 93
  update to Malaysia 2002 ... change trackno to trackno*10000
```

Input example:

```
convolve <<!
grfa4km.dat      (a file with airborne gravity at 4 km elevation)
ir.norm          (a filter function)
grfa4km.flt      (the filtered output)
1
!
```



Example of a typical transfer function (left) and corresponding transfer function.
The file "ir.norm" above must contain unnormalized, equidistant values of the left curve

COVFFT	Estimate 2-D covariance functions, and the isotropically averaged 1-D covariance functions and power spectral density by FFT

```
                    C O V F F T

  Program for computing covariance functions, power spectra,
  and potential degree variances for gridded gravity data
  by fft. Power spectrum will be given in units of
  mgal**2*degree**2, and expressed in db (10log10).
  The program also outputs anisotropy index and slope of
  of power spectrum

  input:

  rfi1, rla1, inn, ine, iwndow

  where

  unit20    name of grid file containing data grid in tc-format.
            the data must be stored in free format as scan lines
            from w to e, starting from n. The grid must be initia-
            ted by a label (la1,lat2,lon1,lon2,dlat,dlon) defining
            grid boundaries.
            only a subgrid on the data grid may be analyzed.
  unit30    outputpfile
  unit32    outputfile for 2-d covariance function grid

  rfi1, rla1  sw corner of wanted subgrid (degrees)
  inn, ine    number of points of subgrid (should be even numbers)
  iwndow      width of cosine-tapered window zone in grid points

  rf, may 1986. Based on osu program cov, december 1983.
```

Note: The program has not been updated recently, but should work. The input file name is "hardwired" into program as "caly.grd", output files as "cov30.out" and "cov32.out".

COVFIT	Fits covariance parameters for use in GEOCOL with using EMPCOV output

```
                              C O V F I T 1 1

Test and fitting of covariance functions. Originally programmed by
C c.c.tscherning, dep. Of geodetic science, the ohio state university,
And geodaetisk institut, danmark, june 1975.
Modified sept 1985 by c.c.tscherning (adapted to rc fortran).
Additions 1986 by p.knudsen for covariance function fitting.
Latest modification: nov., 7 ,1996 by cct.
(c) copyright by c.c.tscherning and p.knudsen, 1975, 1985, 1987, 1988,
     1989, 1990, 1991, 1996.

References:
(a) Tscherning,c.c.: covariance expressions for second and lower order
    derivatives of the anomalous potential, reports of the dep. Of
    geodetic science no. 225,1975.
(c) Krarup, T. And C C Tscherning: evaluation of isotropic covariance
    functions of torsion balance observations. Bulletin geodesique,
    vol. 58, no. 2, pp. 180-192, 1984.
    covariance function using gravity and satellite altimeter data.
    bulletin geodesique, vol. 61, pp. 145-160, 1987.
(e) Sanso, f. And w.-d. Schuh: finite covariance functions. Bulletin
    geodesique, vol. 61, pp. 331-347, 1987.
```

Note: Detailed instructions are to be found throughout the program or by using interactive input. The philosophy of the program is explained in more details in the geoid school notes. The general covariance model fitted is of the "Tscherning-Rapp" type (A: scaling, R_B Bjerhammar sphere depth) with "fudge" factor k applied to a set of spherical harmonic error degree-variances κ

$$\sigma_i^{TT} = \begin{cases} k \cdot \kappa_i & i = 2,...,360 \\ \dfrac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1} & i = 361,.... \end{cases}$$

Input example: Job to fit covariance functions (use output from EMPCOV example)

```
covfit<<!
f
4 f f t f
2
4
-6.150 200.0 360 f t t
0 2 0.27 F
osu91.edg
10 0.5 0.2 0.1
1 1 000 000 T
18 3 3 1700 1700 1 1.0 t
177.8 177.8 40.0 42.0 0.2 100.0 103.0 0.2
nm91.covt
!
```

CRSADJ	Cross-over adjustment (bias and/or tilt) of altimeter data

```
                          C R S A D J

Cross-over program for altimeter data, computing
cross-over differences, estimating bias/tilt parameters, and
correction of the altimeter data.

Input:
ifile
ofile
lat1,lat2,lon1,lon2,ltilt,mode

Input altimeter data format:
mode 1:
rev.no, lat, lon, dummy, ssh, std.dev.
mode 2:
rev.no, t, lat, lon, ssh, std.dev.

Output files:
crsdif.dat, crspar.dat

The ssh should preferably be minus an EGM model

Programmed by Per Knudsen, KMS - Denmark, Oct 5, 1990.
Minor corrections by RF, University of Calgary, March 1993
```

EMPCOV	Finds empirical covariance function from point data sets

```
                        E M P C O V

Program empirical covariance function, programmed by c.c.tscherning,
dep.geod.sc.,osu, 30 oct. 73, version 27 jan 1974.
updated june 13, 1996 by cct.

The program computes an empirical covariance function of scalar
or vector quantities on a spherical surface by taking the mean of
product-sums of samples of scalar values or of the longitudional and
transversal components of vector quantities. Covariances between
maximally 2 sets of scalar or vector quantities may be computed.
each quantity must be identified by an integer between 0 and 16. If
a quantity is a scalar, it is identified by for example (1,0).

The following codes may be used:
geoid undulations, height anomalies, sea surface heights       1
gravity disturbances                                           2
gravity anomalies                                              3
radial derivatives of gravity anomalies                        4
radial derivatives of gravity disturbances                     5
meridian component of deflection of the vertical               6
prime vertical component of same                               7
gravity anomaly gradient, meridian component                   8
  -       -        -    , prime vertical component             9
  -     disturbance gradient, meridian component              10
  -          -          -   , prime vertical                  11
2*mixed second order derivative                               12
difference between horizontal 2'order derivatives             13
```

Detailed descriptions of input must be found in program text, or program run interactively.

Input example: Make an empirical covariance function from a gravity file “nmfa.rd”. The outfile may be used to fit a covariance function, see example in COVFIT.

```
empcov<<!
MN GRAVITY DATA - OSU91A -TC .
5.0 96 3 F T F
nm91.covt
4800 9 T 3 2 0 5.0 f
3 3 0
nmfa.rd
T
!
```

FCOMP	Add/subtract, linearly modify or merge point data files with statistics

```
                              F C O M P

program reads data from two files and produce sum
or difference and writes statistics. the program may
also add bias to data, or merge two data files.

input: inputfile1,
       inputfile2 (evt. dummy name),
       outputfile (or '0'),
       kind, ndata, hist.spacing.

where kind:  0: just statistics for ifile1
             1: difference ifile1 minus ifile2
             2: sum
             3: factor and bias to ifile1 (one for each data)
             4: merge file1 and file2 (ndata pts in file1, 1 in file2)
             5: subtract file1-file2 only when 2nd field in file1
                less than 'trsh' (input), otherwise reject
             6: as 5, except 2nd field is in file2
      10,11,..: as 0,1,.. but with long data lines in file 1,
                additional input: total number of data and wanted
                data positions (file 2 standard format)
                wanted data pos = 99 implies BA land, FA at sea at first
                and second field, respectively
            -1: differences between successive points in ifile1,
                profile statistics for both raw differences and ppm,
                useful for gps traverses.
            -2: output first data as cumulative profile dist in km

ndata is the number of data following id,lat,lon,elev
hist.spacing is histogram spacing, if 0 no histogram is output.
if outputfile = '0' no output is written (saves disc space!)
programmer: rene forsberg, jan 85
last updated: jul 93, rf, may 95, nov97, rf
```

G2SUR **G2SURB**	Conversion of GRAVSOFT grids to SURFER format (G2SURB works for binary grid files)

```
                          G 2 S U R

program to convert gravsoft grids to surfer ascii format (.grd)

input:

gravsoft grid file (usually .gri)
surfer file name (should be .grd)

(c) Rene Forsberg KMS, Denmark
Rio sep 11, 1997
updated for UTM july 1999
```

GBIN	Conversion between ascii and binary grid formats

```
                         G B I N

Program for converting a text grid into binary or the reverse.
The grid file may only contain one grid.

A direct access file format is used for binary files.
One record consist of 16 floating point (real*4) numbers = 64 bytes.
Due to the use of real*4, binary files will not work on machines
with a single precision word length different from 4 bytes, in
this case the 'recl' must be changed in the open statements below.

Input:

    ifile
    ofile
    mode (1: txt-> bin, 2: bin->txt, 3: bin->txt integers)

If ofile = '0' is specified no output is produced (useful for a check).

(c) rene forsberg oct 89, updated dec 89
```

GCOMB	Subtract (1), add (2), overwrite (5) or combine grids in various select models. Also for one-grid scale+bias modification (0), or some specialized physical geodesy corrections (9,10)

```
                         G C O M B

Program for combining two grids into one by various manipulations.
The grids must have the same grid spacings and relative positions,
but not neccessarily the same coverage area.
The output grid will cover the same area as the first grid
9999-values (unknown) are not added/subtracted.
Input/output may be either binary or text format

Input:

gridfile1               or          gridfile
gridfile2                           *
outfile                             (lat1,lat2,lon1,lon2 - interactive)
mode, iout

where mode determines function of program (except *-mode)

mode = 0:  one-grid modification. Additional input: factor, bias
           (new  = old*factor + bias)
mode = 1:  subtract 'grid1' minus 'grid2'
mode = 2:  add 'grid1' plus 'grid2'
mode = 3:  general grid convert: output = a*grid1 + b*grid2
           additional input: a, b
mode = 4:  treshold values. set values in grid1 to 9999 when
           lgt=true: value in grid2 > trsh (input lgt,trsh after iout).
           lgt=false: value in 'grid2' < 'trsh'
mode = 5:  grid overwrite
           values in 'grid2' is written on top of 'grid1', except when
           9999 is encountered, then the grid1-values are kept.
           (special: mode = -5 allows many text grids in 'grid2' file,
           output must be binary)
mode = 6:  grid select
           values in 'grid1' are written only when there is no data in
           'grid2'.
mode = 7:  Treshold select. Keep values in grid1 when grid1 > treshold,
           keep values in grid2 when both grid1 and grid2 < trsh
mode = 8:  Variable multiplication. Multiply grid 1 by "fak1" when
           grid 2 .lt. 'treshold', else multiply by "fak2"
mode = 9:  Grid 1 is a Bouguer anomaly grid, grid 2 a height grid.
           Output N - zeta (difference geoid minus quasigeoid) in
           linear approximation. If grid 1 is a free-air grid,
           the difference zeta* - zeta is obtained.
mode =10:  Grid 1 is a geoid model with Helmert, grid2 a height grid.
           Output is a geoid corrected for indirect effect.
iout:      output format:
           1  binary
           2  txt, reals
           3  txt, integers

special options:

if 'grid1' = 0 or 9999 then a label must be input, and grid1 will be
```

GCOMB (continued)

```
all zeros or 9999's.

if 'outfile' = 0 the output is on the screen without statistics.
if both 'grid1' = '0' and 'outfile' = '0' this gives interactive listing

if 'grid1' and 'outfile' are identical only the relevant rows are
updated for binary files (gives fast overwrite)

  © rene forsberg, KMS; updated thessaloniki/unsw nov 1988
updated and changed dec 89, feb 92, rf
updated jan 94, chung cheng inst of technology, taiwan, rf
update with binary grids, march 1996, rf
update with indirect helmert geoid effect, may 21, 2000
update with treshold less than, jan 23, 2002
```

Input example #1 – subtract two grid files with identical spacing

```
gcomb <<!
grid1.gri
grid2.gri
dif.gri
1 2
!
```

Input example #2 - use overwrite mode fast extract of a subgrid from a large binary file:

```
gcomb <<!
0
binfile
subgrid
5 3
55.0 55.1  10.0 10.1  .01  .01  (grid label for wanted area)
!
```

Notes:
"gcomb" is frequently used interactively.
For subtraction/addition of grids with different grid spacings GEOIP may be used.

GEOCOL	General least squares collocation on the sphere. Evaluation of spherical harmonic fields and many other things …

Note: GEOCOL is by the largest program in the GRAVSOFT package and maybe used for many applications without other software. Input for the program is *very* complicated, and further hampered by changes between different version. Below is shown a single example of input. For a comprehensive review see the Geoid School notes (Tscherning, 1994). The text of the program contains a detailed list of the background, some logical control variables and some general information on the types (e.g., gravity anomalies, 2nd order gradients etc.)

Input example #1: Job to predict geoid from residual gravity and compare predicted and observed deflections of the vertical .. *from New Mexico excersizes*

```
geocol11<<!
t
f t t f f f t
t f f f f
nmrestart
nmneq1
f t f
t
5
osu91a
3.98600500e+14 6378137.0 -484.1655 36 f f t f
(2i3,2d19.12)
osu91a1f    (egm model, use local file name !!)
2
4
-6.061 196.20 360 f f t f
0 4 0.207227
osu91.edg
t
1 2 3 3 4 7 0 -13 -1 1700.00 f f f t f f t t f t
nmfa.rd
25
5.0
32.5 34.5 -107.5 -105.5
0.2
t
t f
f f f
f t f
1 2 3 3 5 0 0 11 -1 1700.0 t f f f f f f t f t
nm.h2
27
nm.geoid
33.0 34.0 -107.0 -106.0
t
f
f t t
1 2 3 3 4 5 6 -5 -1 1700.0 f f f f f f t t f t
nmdfv.rd
20
0.2
33.0 34.0 -107.0 -106.0
0.6
-1.0
t
t
```

GEOFOUR	Planar FFT program with many modes: upward continuation, gravity geoid, geoid to gravity, Molodenskys boundary value problems a.o.

```
                         G E O F O U R

Program for gravity field modelling by fft. a input data grid
e.g. generated by 'geogrid' is transformed, using kernels given
in the frequency domain. In the simple mode (<10) a gravity or geoid
grid is transformed assuming data to be given on a common level.
If data is given on the surface of the topography a first-order
harmonic continuation scheme may be used, and in this case a
height grid must also be input. For dual grid input, the program
uses conjugate symmetry, so only one forward FFT is required.

Input:
------
  <gfile>
<hfile> (or possibly dummy name)
<ofile>
mode, attkm,
(hkm - for mode 6 only)
rfi1, rla1, inn, ine, iwndow

where

<gfile>   is the grid file containing data grid in standard
(unit 20) format, i.e. scanned in e-w lines from n to s,
          initiated by label (lat1,lat2,lon1,lon2,dlat,dlon)
          describing grid limits. utm grids may be used, in
          this case lat/lon should be northing/easting in
          meter, followed by ellipsoid number (1:wgs84,
          2:ed50, 3:nad27) and utm zone.
          a subgrid of the data grid is analyzed, see below.

<hfile>   is a height grid file on text format corresponding
(unit 21) to <gfile>. not needed in the simple mode (dummy name
          must be specified). the height grid must have the
          same spacings and relative position as the <gfile> data
          grid, and must have at least the wanted area in common.

<ofile>   outputfile (output in grid format). deflections are
(unit 30) written as two grids within the same file.
(unit 31) temporary file for storing intermediate results
(unit 32) do, for harmonic continuation modes only.

mode      1  conversion gravity to geoid (stokes formula).
          2  gravity to deflections (vening meinesz).
          3  conversion geoid to gravity.
          4  gravity to tzz (eotvos)
          5  gravity to plumbline curvatures tyz,txz (eotvos)
          6  upward continuation to h (km) (NB! point innerzone)
          7  deflections (ksi,eta) to gravity
          8  ksi (arcsec) to gravity
          9  eta (arcsec) to gravity
         10  downward continuation of gravity data to sea level.
         11  gravity to geoid, using harmonic continuation to a
             mean height reference level, followed by stokes and
```

GEOFOUR (continued)

```
                continuation of computed geoid to surface of topography.
            12  gravity to deflections using harmonic continuation.
                continuation of deflections from reference level done
                by using plumbline curvature parameters txz and tyz.
             0  nothing, gravity wiener filtering only

attkm           wiener filtering resolution for noisy data (kaula rule).
                this option should be used only for high-pass filtering
                operations (modes.ge.3), 'attkm' specifies the resolu-
                tion (km) where the wiener attenuation filter is 0.5.
                attkm = 0.0 means no attenuation used.

rfi1, rla1      (degrees (or m for utm)) sw corner of wanted subgrid
                if rfi1=rla1=0 then the sw-corner of the grid is used.
inn, ine        number of points of subgrid (must be even numbers)
                if the wanted subgrid is bigger than the actual grid
                the transform grid is padded with zeros. on output only
                the non-padded part of the grid is written.
iwndow          width of cosine-tapered window zone in grid points

(c) rf, danish geodetic institute/national survey and cadastre
original version, june 1986
modified and updated aug 87.
frequency domain filters changed jan 89, rf, unsw vax
last update jan 91, rf (removal of inner zone bug)
            dec 93     (upw cont)
            jan 96     (deflections to g)
```

Input example: Prediction of T_{zz} second-order gradient from gravity data file, using a Wiener smoothing of 20 km. A grid of 200 x 256 points is transformed, with the 5 closest points to border cosine-tapered. The basic data (gap91.rd1grd) has been reduced for EGM effects.

```
geofour <<!
gap91.rd1grd
dummy
gap91.rd1tzz
4 20.0
0 0 200 256 5
!
```

Note: The GEOFOUR program applies filters on Fourier transformed data in the frequency domain, e.g. for upward continuation

$$F(\Delta g^h) = e^{-kh} F(\Delta g), \quad k = \sqrt{k_x^2 + k_y^2}$$

$$F(\Delta g) = \iint \Delta g(x, y) e^{-i(k_x x + k_y y)} dxdy$$

Different modes are supported, see below. For gravity to geoid modes (e.g., for converting satellite altimetry to gravity) stabilized Wiener filtering is needed, taking into account noise and the Kaula rule for the estimate og signal p.s.d., i.e.

$$\Delta g^{wiener} = \Delta g \frac{e^{kh}}{1 + ck^4}$$

The Wiener filtering parameters are defined in terms of effective filter resolution.

GEOGRID	Least squares collocation (Kriging) or weighted means prediction from points to a grid. May also interpolate unknown values in grids, or predict points or profiles.

```
                            G E O G R I D

Program for gridding of irregular distributed data into a regular
rectangular grid, for interpolationg data in profiles, or for
interpolating individual points using enhanced, fast weighted means
interpolation or collocation/kriging. the program may detrend data prior
to gridding, or just fit a trend surface to data.

A quadrant search method is used internally in the program to speed up
computations. this means that at each prediction point only the 'nqmax'
nearest points is used in each of the four quadrants around the point. to
speed up the quadrant search an internal data organization is used, where
an internal data grid ensuring 'rdat' (p.t. 3) points per compartment in
average is used. this grid is not related to the possible wanted
prediction grid.

Program input:

  <ifile>
  <ofile>
  <efile>
  nd, idno, nqmax, itrend, ipred,
  <predpar>,
  mode, rkm,
  <pred points>

 where
  nd     ...  data values in line pr. point in <ifile>
              not used when <ifile> contain a grid (mode 7, see below)
  idno   ...  data value used out of the 'nd' possibilities.
              if 'idno' is zero the height data are gridded.
              if 'idno' is negative then the data number abs(idno)
              is assumed to be followed by a standard deviation.
              the standard deviation is used in the collocation
              prediction only if it is greater than the specified
              sigma, see predpar below.
              Standard deviations on heights is specified as idno = -99
              Special bouguer/free-air option: if idno = 99,
              free-air is used at sea, bouguer on land, with free-air
              anomaly position first on line. sea points have negative h.
  nqmax  ...  number of closest points per quadrant used in prediction,
              if nqmax<=0 all points are used.
  itrend ...  trend surface removal: 0  none (just grid data)
                                     1  remove mean
                                     2  remove linear function in x and y
                                     3  remove 2nd order polynomium
                                     4  remove 3rd order polynomial
                                     5  remove 4-par (geographic only),
                                        corresponds to 7-par datum shift
             if 'itrend' is negative only the residuals from the trending
             are output (i.e. data trend is not restored after rediction)
 ipred  ...  prediction type:  0  none (detrend only)
                               1  collocation (kriging)
                               2  weighted means
```

GEOGRID (continued)

```
<predpar>   depends on prediction type:
             ipred = 1: xhalf(km), rms(noise)
               -     2: prediction power (e.g. 2)
             note: for collocation rms(noise) is the smallest standard
             deviation used when data is given with sigmas.
 mode   ...  specifies data to be predicted:
             mode = 1: grid in geographical coordinates
                    2, 3: grid in utm projection
                    4: profile interpolation
                    5: point interpolation, data points given in
                       <efile> in standard gi format, i.e. as
                       gi-id, lat, lon, height, ...
                    6: do, with input data assumed to be bouguer
                       anomalies (GRS80/2.67). output data will be
                       gravity values, derived from interpolated
                       anomalies and prediction point heights.
                    7: fill-in of unknown (9999) grid values. In this case
                       <ifile> must contain a grid rather than a pt list.
                        Only the 9999-values are actually predicted.
              negative: Same as positive value, except only positive
                        prediction values allowed (all negative values
                        set to zero). Use for DTM interpolation only.
  rkm    ...  margin (km) for data selection area surrounding the
              wanted grid (or rectangular envelope of the profile
              or wanted points for mode > 3).

  <pred points>  specifies prediction points depending on 'mode':
  mode 1: fi1, fi2, la1, la2, dfi, dla        (geographic grid)
  -    2: n1, n2, e1, e2, dn, de / iell, zone (utm grid)
  -    3: fi1, la1, dn, de, nn, ne / iell, zone (utm grid with
                                               sw-corner lat/lon,
                                               spacing, and number of
                                               points)
  -    4: fi1, la1, fi2, la2, n               (profile with 'n' points)
  -    5, 6, 7: (nothing)

  the utm projections are defined by their zone no ('uzone') and
  ellipsoid no ('ie'): ie = 1: wgs84/nad83, ie = 2: hayf/ed50,
  ie = 3: clarke/nad27.

  the <ifile> contains data points in standard gi-format, i.e. given
  as lines 'stat-id, lat, lon, height, data(1), .. , data(nd)', where
  'idno' specifies which data to be used.
  data values <= -9999 or >= 9999 signals missing data and not used.

  output on grid format is written in 'ofile'. for collocation errors
  are written on 'ofile'. for weighted mean interpolation 'efile'
  contains distances in km to the closest data point.
  for profile and point interpolation all output is on 'ofile'.

  for collocation a second order markov model is used, with c0
  assigned from the data variance.
```

Programmer: Rene Forsberg, danish geodetic institute, nov 85. Updated oct 29, 1991, rf/se. Modified for vax, rf unsw, dec 88. Inclusion of individual variances and detrending, rf jan 1990. Grid fill- included aug 1990, rf. Negative modes dec 16, 1991, rf. Sorting arrays expanded and lambert/polar projections jan2002,rf.

GEOGRID (continued)

Input example: Grid value and error prediction from point data using Kriging, correlation length 25 km.

```
geogrid <<!
grav.dat                  (input data file, grid data #1)
grav.gri                  (output grid file)
grav.err                  (output collocation error file)
1 1 5 0 1                 (use 5 nearest neighbours, no detrending, coll)
25.0 1.0                  (25 km correlation length, 1 mgal noise)
1 0.0
55 56 10 11 0.05 0.1      (grid label for prediction grid)
!
```

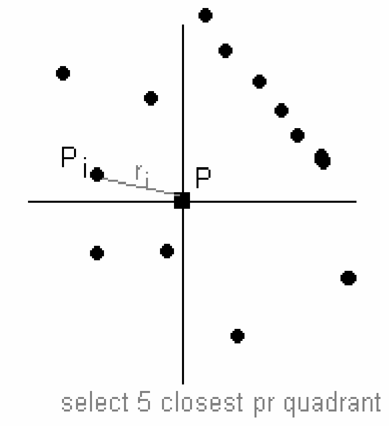
Note: GEOGRID uses a quadrant-based nearest neighbour search combined. The program typically searches for e.g. the 5 closest neighbours in each quadrant around a prediction point. It then at each prediction point applies either weighted means prediction

$$\hat{s} = \frac{\sum_i \frac{x_i}{r_i^2}}{\sum_i \frac{1}{r_i^2}}$$

or interpolation by least-squares collocation (Kriging), using a 2nd order Markov covariance model.

$$\hat{s} = C_{sx} C_{xx}^{-1} \underline{x}$$

$$c(r) = C_0 (1 + \frac{r}{\alpha}) e^{(-r/\alpha)}$$



The covariance scale C_0 is found automatically from data, and the parameter α determined from the correlation length specified by the user.

GEOID	Linear binary grid interpolation for geoid use. Includes coordinate transformations and transformation of heights. Should also allow reading of alphanumeric station names

```
                         G E O I D

  Linear interpolation and transformation program for binary
  geoid gridfiles. This program is an alternative to 'geoip', and
  differs a.o. in the way that subgrid are are not stored in memory,
  but accessed everytime from disc.

  The program may also be used for transformation between UTM, geographic
  and cartesian coordinate systems, may convert ellipsoidal to orthometric
  heights and vice versa. An option to fit geoid residuals to known
  control may be activated (not in this version),
  Input to the program is interactive.

  The program runs in a number of basic modes:

  0: just coordinate conversion (i.e., geoid height assumed to be zero)
  1: interpolate geoid heights in same system as in file
  2: convert ellipsoidal to "normal" heights
  3: convert "normal" heights to ellipsoidal heights
  4: interpolate geoid height and convert to other datum (e.g. ED50 geoid)
  5: deflections of the vertical (simple linear differentiation of geoid)
  6: deflections of the vertical in local system (e.g. ED50 deflections)

  NOTE: The geoid height given is always assumed to be geocentric  WGS84).
  The coordinate transformations in mode 0-3 only relate to the
  transformation for obtaining latitude and longitude, not for
  interpolating of grid values.

  Station numbers may contain letters, up to 11 characters
  For latitude and longitude N,E,S,W are allowed as prefix, ok like:
   N 45 02 21.1  E 00 02 21
     45 02 21.3    -0 02 21
     45 02 21.5     0 -2 21
  Output is always is gravsoft format (like last line)

  (c) Rene Forsberg, Kort- og Matrikelstyrelsen, dec 1990.
  Last update feb 1992.
  Special version for Dubai Municipality Nov 2001
  Modified for OSGM02 project jan 2002
```

Notes:

GEOID takes advantage of the special structure of the GRAVSOFT binary grid file to avoid reading a complete geoid grid file, when only few points need to be interpolated.

GEOID is usually given in specific versions for actual geoid projects (e.g., implementing special map projections), and may have name with a subscript.

A “nice” windows-based menu geoid interpolator is also available (author: Arne V. Olesen)

GEOIP	Interpolation (linear or splines) from grid(s) to points or grid to grid. Also implements subtract/add or 3-D “sandwich” grid interpolation. UTM grids are fully handled, included UTM to UTM interpolation.

GEOIP

program for interpolating values from a grid using bilinear or spline interpolation. the grid or the prediction points may be in either geographical or utm coordinates. the spline prediction is performed in a window of size 'nsp' x 'nsp' points around the wanted points, with typical value of nsp being 8 for a good interpolation.

the grid file must be in standard format, i.e. scanned in e-w bands from n to s, initiated by a label (lat1,lat2,lon1,lon2,dlat,dlon). for utm grid northing and easting replaces lat and lon, and additionally ellipsoid number (1: wgs84, 2:ed50, 3:nad27) and utm zone must be given in label. if utm zone 99 is specified, this signals the swedish national projection rt39 (only an approximative transformation is currently implemented, only for geoid use etc.) the program may interpolate from one utm zone to another.

the grid file may be in direct access binary format, as produced by program 'gbin'. the use of direct access format speeds up access time. the program recognizes binary files by a special code (777) written in the first record.

the program attempts to interpolate all points with a distance 'rmin' or more from the margins (for fft applications, e.g., points near the margin are often useless). the program may interpolate in very big grid files, but assumes the prediction points to be reasonably close, reading in the smallest necessary subgrid to perform the wanted interpolations.

special options:

- the program may interpolate in two grids in the same file. this option is especially designed for deflections of the vertical. the two grids must have identical labels. two grid interpolation is signalled by negative mode or mode > 100, see below. two-grid interpolation can only be done for grid files in txt format.
- two grids representing data in different elevations may be used to interpolate values at some elevation between the two grids. in this case mode = mode+100 must be specified.
- the program may also be used to convert a list of terrain corrections into rtm-effects through a bouguer reduction to the interpolated reference level. this approximation is only valid for long-wavelength reference grids. stations at sea (negative heights) will have height set to zero.
- in bilinear interpolation mode the grid file may contain 9999-values (signals unknown), interpolated values are assigned to 9999 if any unknown values are encountered in the closest 4 points.

For grid interpolation the wanted grid may be to large (9999's will be written at unknown values)

GEOIP (continued)

```
program input:
  gridfile,
  outputfile,
  mode, nsp, rmin, lsel
  (lat1,lat2,lon1,lon2 - for lsel true)
  (pointfile - for mode 1 - 3 and modes > 10)
  (idno - for modes 11-14,19-22 etc.)
  (lint,lat1,lat2,lon1,lon2,dlat,dlon - for mode 4 only)
  (iell,izone - for utm only)
  (h1,h2 - for modes > 100 only)

  where

  mode ...  1: prediction point list in geographic coordinates (degrees)
            2: do, with lat and lon given in degrees, minutes, seconds
            3: prediction point list in utm coordinates
            4: predictions wanted in grid (geographic or utm)
            5: individual prediction points in lat, lon (degrees)
            6: do, with lat, lon in degrees, minutes, seconds
            7: individual prediction points in utm
           10: like 1, with a data value in file written after predictions
           11: like 1, predictions  s u b t r a c t e d  from values
               given in file (data value must follow after the height)
           12: do, but predictions  a d d e d  to values in file
           13: like 11, with a second data value for each point not hanged
           14: like 12, doc
           15: 'pointfile' contains a grid, from which the
               interpolated values from 'gridfile' are subtracted
               i.e. 'outfile' = 'pointfile' - 'gridfile'
           16: 'pointfile' contains a grid, to which the
               interpolated values from gridfile are added.
           17: 'pointfile' contains a grid which defines the interpolation
               points. the given grid values are only used in two-height
               interpolation mode (117), see below.
           18: 'pointfile' contains a grid with unknown values (9999).
               the unknown values are interpolated from the gridfile,
               other grid values left untouched.
           19: list of terrain corrections converted to rtm anomalies
               through bouguer reduction to reference level 'gfile'
           20: list of free-air anomaly data converted to rtm-reduced
               data using a bouguer plate approximation only
               additional input: density
           21: conversion of free-air data to Bouguer using grid
               additional input: density
           22: migration of ERS-1 data over ice caps. grid is slope
               grid in degrees
           31, 32, ..: like 11, 12, .. for utm coordinates in 'pointfile'

           - if mode is negative, mode = abs mode and  t w o  grid
           interpolation is performed (the two grids must be in
           the same file, e.g. deflections of the vertical from
           program "geofour")
           - if mode > 100 then point interpolation is done between
           two grids in different heights using mode = mode-100.
           the levels to which the two grids refer must be input.
           if mode=117 then 'pointfile' is assumed
           to be a height grid defining the interpolation level.
```

GEOIP (continued)

```
 nsp .... spline window size. nsp=0 means bilinear interpolation,
           nsp=1 is equivalent to a 8x8 spline (nsp=8)
 rkm .... minimum required distance to closest edge (km) for
           interpolation to take place
 lsel  .. select only data wihin a subregion (lat1-lat2, lon1-lon2)
          this option only works with point lists
 idno  .. data number in line (statno,lat,lon,height,d(1),d(2)..)
 lint ... a logical variable specifying that output should be in
          integer grid format (only needed for mode 3 and 4)
Files:

 unit10  gridfile   grid file in standard format
 unit20  pointfile  station list file (no, lat, lon, height, ...)
 unit30  outfile    output file
 unit31,32          scratch files for intermediate storage of all
                    prediction points in binary format

programmer: rene forsberg, danish geodetic institute.
university of calgary, mar 87. modified october 87, honefoss.
minor changes, rf thessaloniki nov 88
revised for swedish map projection and more points, rf june 89
selection added, greenland aug 12, 1992, rf
new modes for ice etc., nov 1992, rf
allowing grid interpolation outside (with 9999's) in linear mode, dec95
```

Input example #1: Simple interpolation of grid file (“geoid.gri”), to obtain linearly interpolated values at point locations in file “points.dat”. Note that the output file name always must be the 2nd file name in the output stream (not very logical, but necessary due to the many options in GEOIP!)

```
geoip <<!
geoid.gri
output.dat
1 0 0
points.dat
!
```

Input example #2: Interpolation of grid file with two grids, representing a spherical harmonic gravity grid at heights 0 and 3 km, interpolated at the actual station height in the data file “borneo.rd”, and then subtracted from an observed gravity file “borneo.faa”.

```
geoip <<!
egm96gh.gr
..\data\borneo.rd
113 0 0 f                ! mode 113 is the sandwich mode
..\data\borneo.faa
1                        ! datano in file
0 3000                   ! heights of reference grids
!
```

GPCOL	Least-squares collocation with the planar logarithmic covariance model

```
                          G P C O L

program for testing covariance function and least-squares
collocation. a data file (max 'maxobs' pts) is read in and
predictions carried out in a grid

input: sqrtC0, D, T  (covariance parameters, mgal**2 and km)
       ki1, sig1
       <ifile1>
       ki2, sig2
       <ifile2>
       -1 -1         (code for end of observations)
       ko, rlat1, rlat2, rlon1, rlon2, dlat, dlon, h
       <ofile>
       <efile>

ki, ko: input/output type (1:geoid, 3:dg, 6:ksi, 7:eta)
        deflections given in mgal
sig:    input data std.dev. If sig = -1 then the std.dev. must
        be given in the datafile.

rf, feb 86; f77 version sep 1993
```

Input example: Prediction of gravity at geoid from airborne gravity data in”pip.dat”. Data r.m.s. ($\sqrt{C_o}$) is set to 10, D=10 km and T=50 km.

```
gpcol <<!
10 10 50
3 2.0
gravair.dat
-1 -1
3 55.3 55.7 10.6 11.4 .1 .2 0
grav0.gri
grav0.err
!
```

Note: GPCOL is a general collocation program, solving collocation linear equations by the Cholesky method. Opposed to GEOCOL the equations are not blocked, and should all fit in memory. The number of possible observations in practice is therefore limited (more than 3-4000 would just take too long time!). The program implements the planar logarithmic covariance model, a self-consistent, spatial covariance model, for details see Forsberg (1987). For gravity the model gives an expression of form:

$$C(\Delta g^{h_1}, \Delta g^{h_2}) = -\sum_i \alpha_i \log(D_i + \sqrt{s^2 + (D_i + h_1 + h_2)^2}$$

which may be expressed by three free parameters: C_0, D and T, where C_o is a variance, D a depth parameter corresponding the Bjerhammar sphere depth, and T a long-wavelength attenuation factor (corresponding to a kind of “compensation depth”). The parameter may be estimated by GPFIT.

GPCOL1	Least-squares collocation with the planar logarithmic covariance model in blocks (corresponds to GPCOL, except computation are done in blocks with overlap)

```
                              G P C O L 1

Program for flat-earth logarithmic least-squares collocation. One or more
data files are read in and predictions carried out in a grid. The program
may also be used just for interactive listing of the analytical covariance
functions Cgg, CgN, CNN (use nifiles=0). An empirical cov.fkt. will be
output for the first data kind.

input:   nifiles            (number of input files, 0=list cov)
         sqrtC0, D, T       (covariance parameters, mgal and km)
         ki1, sig1          (input kind and noise)
         <ifile1>           (first input file)
         (dlim  -  for ki1 = 5 only)
         ki2, sig2          (.. second input file ..)
         <ifile2>
         ...
         ko, mode, lblock
         (rlat1, rlat2, rlon1, rlon2, dlat, dlon, h  - for mode 1)
         (dfile - for mode 2 or 3)
         (rlat1, rlat2, rlon1, rlon2 - for lblock=true and mode=3)
         (dlat, dlon, blat, blon, nmin  - for lblock=true)
         ofile
         (efile - for lblock=f)


ki, ko: input/output type (1:geoid, 2: defl.pair, 3:dg, 6:ksi, 7:eta)
        deflections given in mgal
        specials:
        ki = 5: input geoid heights, but use slopes (along-track dfv)
                additional input: dlim (km) .. in-track max dist
        ko = 0: just check blocks and emp.cov. of first data
sig1:   input data std.dev. If sig = -1 then the std.dev. must
        be given in the datafile.
mode    1  predict grid at constant height
        2  predict grid at heights given in dfile
        3  predict at list of points (id,lat,lon,h)
        note that point list sequence is changed in blocked mode
        whole grid predicted in mode 1-2, except in blocked mode

lblock  make blocked computations in (dlat x dlon) blocks with blat,blon
        borders (degrees). Output is in this case in text line format.
rlat1, rlat2, rlon1, rlon2, dlat, dlon, h: wanted grid and height
dlat,dlon,blat,blon   block size and border in degrees
nmin                  minimum number of obs in central block for a
                      solution (the central block expanded with
                      blat/3,blon/3)

(c) Rene Forsberg, KMS-Denmark
f77 version sep 1993; last updated dec 1, 1994 (sign error correction)
jan 29, 1996 (lhgrid/blocks/slopes - careful, slopes not tested)
nov 29, 2002 (errors corrected + point lists)jun 10, 2003 (expansion of
central block with 1/3-border)
```

GPFIT	Fits planar logarithmic covariance functions to point data

```
                              G P F I T

Program for fitting flat-earth logarithmic covariance function to
gravity data in a file. The program is the planar equivalent of “empcov”
and “covfit”.

input: ifile  (gravity data)
       ds,smax,d1,d2,t1,t2  (all in km)

ds, smax defines the interval and spacing of summing up an empirical
covariance  funtion between all points.

d1,d2 and t1,t2 defines the search ranges of “best” D and T parameters.

Reference: A new covariance model for inertial gravimetry and
gradiometry. Journal of Geophysical Research vol. 92 pp. 1305-10, 1987.
```

(c) Rene Forsberg, KMS-Denmark, sep 1998

HARMEXP	Evaluates geoid, gravity and deflections in grids or points (alternative to GEOCOL)

```
                                H A R M E X P

  program for computing grid of geoid, gravity and deflections from
  a high degree and order spherical harmonic expansion, using 'gpotdr'.
  output will be in three files (geoid, gravity and ksi/eta) in units
  of m, mgal, and arcsec, respectively, in standard grid format.

  input:

  coefficient file name,
  irefsys, mode

  irefsys determines a and gm coefficients:
       0: use grs80 constants (i.e., only sum from J2)
       1: egm96 with grs80 normal field
       2: champ eigen-2 with grs80

  * additional input, mode = 1 (grid computation):
  geoidfile,
  gravityfile,
  deflectionfile,
  nmax, lbin,
  fi1, fi2, la1, la2, dfi, dla (degrees), h (m)

  'lbin' is true for coefficients on binary form.
  file name 'dummy' or '0' ensures no data of that kind is written.

  * mode = 2 (point computation):
  pointfile
  outputfile
  kind, omode

  where kind =  1: geoid
                2: gravity
                3: deflections
        omode = 0: list computed values
                1: list difference (pointfile - ref.field.)
                2: list sum (pointfile + ref.field.)

  modification of program from d. arabelos, university of thessaloniki,
  nov 88, rf. original subroutines from tscherning and goad.
  modified and updated, university of new south wales vax, feb 89
  modified nov 2002,rf  (gnu fortran, last binary coeff must be nmax,nmax)
                        (normal gravity field update)
```

Input example: Prediction of geoid and gravity effects in a grid.

```
harmexp<<!
0 1
egm96geoid.gri
egm96grav.gri
0
360 t
33 37 22 28 .1 .1 0
!
```

POINTMASS	Makes pointmass geoid, gravity or deflection effects in grid or point formats. Useful for checking if transformation (FFT, collocation ..) programs run properly!

```
                        P O I N T M A S S

  program for generating a grid of point mass gravitational
  effects.

  input:

  <outputfile>,
  np, kind, iform,
  rfi(1), rla(1), d(1), g(1),
  rfi(2), rla(2), d(2), g(2),
  ........
  rfi1, rfi2, rla1, rla2, dfi, dla (grader)

  where np is the number of point masses, each specified by
  rfi, rla: latitude and longitude in degrees,
  d: depth in km, g: peak anomaly in mgal at surface from mass.
  kind: 1 geoid, 2 gravity, 3 ksi, 4 eta
  iformat: 1 grid format, 2 dataline format (on output)
  last input line specifies wanted grid.

  rf, june 1986, modified for norsk data oct 87
```

Input example: Deflections of the vertical and geoid grid from a pointmass anomaly of 20 mgal max amplitude at 10 km depth, at 55.5 N, 11E.

```
pointmass <<!
pmksi
1 3 1
55.5 11 10 20
55 56 10 12 .1 .2
!
pointmass <<!
pmeta
1 4 1
55.5 11 10 20
55 56 10 12 .1 .2
!
pointmass <<!
pmgeoid
1 1 1
55.5 11 10 20
55 56 10 12 .01 .02
!
```

SELECT	Select, thin and/or average data in either point or grid format. May also add noise to data for simulation purposes and produce crude line-printer plots

```
                    SELECT  (on unix: SELEC)

program for selection, averaging or reformatting data.
the program may select one data point per pixel, closest to the
knots of a given grid, or produce a straight average of all values
in the grid cell. the program may also be used only to reformat data
into standard format (id, lat, lon, height, data ..). it may also
produce a primitive coverage plot.

input:

inputfile,
outputfile,
mode, iang, n,
   (dataline code spec plus format - only for iang<=0)
(rfi1, rfi2, rla1, rla2, dfi, dla (degrees or m) - only for mode>0)
   (iell, izone - only when limits are northing and easting in meter)
   (rejection level - for mode 5 and 7 only)
   (window min., max. latitude and longitude - for mode 6 and 7 only)

'mode' determines the function of the program:
 0: reformat all data
 1: select data closest to knots (if dfi=0 all data in area selected)
 2: average all data in grid cell and output in list format
 3: do, output in grid format with 9999.0 in cells without data
 4: primitive screen plot
 5: as 1, but data with estimated err larger than 'rejlev' are rejected
 6: as 1, but a window inside the area is excluded
 7: as 1, but select minimal value in cell
 8: as 1, but with random, normal distributed noise added. noise
    standard deviation must be input after grid specification

'iang' is a code for input coordinates and format:
    1: no, lat/lon in degrees, height, data1, data2  ..
    2: do, with lat/lon in degrees and minutes
    3: do, with lat/lon in deg, min, sec
    4: altimetry format (n=1 gives ssh only, 2 ssh+error, 3 sh+error+t),
    5: binary format,
    6: gravity data in kms 80-char format
       (n=1 fa, n=2 fa+err, n=3 g,fa,ba, n=-3: fa,err,tc)
    7: grid format (if n=0 a height grid of integers assumed)=
    if 'iang' = 0 a data line specification must be input (see below),
    if 'iang' is negative a standard data line format must be input (do)

'n' is the number of data following no, lat, lon, elev (max 3) in stan-
    dard format, or number of wanted data in data line format.
    for averaging only data number 'n' is used and output.

'rfi1, rfi2, rla1, rla2' (degrees) specifies boundaries of wanted grid

'dfi, dla' (degrees) specifies wanted grid spacing, if zero all data
    within area is selected (mode 1), or a default plot is made (mode 3)

utm option: if rfi1 or rfi2 > 100 then the wanted grid is a utm grid,
and ellipsoid number (1: wgs 84, 2: ed50, 3: nad27, ..) and utm zone
```

SELECT (continued)

```
must be input.
if coordinates in file are > 90 or 180 the file is assumed to contain
northings and eastings in same system.

input data is normally assumed to free format.
if 'iang' is negative iang = abs(iang) and a data format in brackets
without quotation marks must be specified, e.g. (i4,2f10.6,f8.2,2f7.2)

dataline option:
if 'iang' = 0 then 'n' is the number of data wanted in output.
data sequence in a line (free format) is specified by
codes 1: station number, 2: latitude (21: lat min, 22: lat sec),
3: longitude (31: lon min, 32: lon sec), 4: height, 5, 6, .. data.
the data codes must be specified in i3 format, followed by lfree and
wanted data, e.g.

  1  2 21  3 31  4  5  6
  t  6  5
  (plus format if lfree.eq.false)

which will read a line with station number, coordinates in degrees
and minutes, height followed by two data values, but output data in
reverse order. if a station number is not specified (code 1) a
sequential number will be given. if heights are not given 0 will be used
'lfree' is true if free (*) format may be used. otherwise the
format (all reals) must be given on the next line.

map option: a primitive contour map of
data values at pixels will be output for check of station coverage
rather than the output list of coordinates. if dfi = 0 a grid will
be selected with default options in this case.
(the map option should only be attempted for reasonably small grids)

rejection option:
in mode 5 or 7 the rejection level must be given. if a negative
rejection level is given, priority is given in the pixel search to
smallet standard deviation (i.e., a low std.dev. is selected rather
than the closest point). in standard format the std.dev. must be the
second data (n=2).

programmer: rene forsberg, januar 1985
updated jan 89, university of new south wales vax, rf
updated june 22, 1989 by cct.
last updated jan 16, 1992, rf, mv, hd; jan 20, 2001, rf (utm input);
mar 19, 2002, rf (minimal pixel values)
```

<u>Note:</u> Select is a quite powerful reformatting and search program, and may do tasks such as reversing order or columns in a file. The select input is obsually run interactively.

SP1D	Implements the 1-D spherical FFT of Haagmans for geoid determination. Allows very large grids to be transformed, but is slower than SPFOUR

```
                              S P 1 D

 program for spherical fft transformation using
 Haagmans 1D FFT method. All data are for speed and simplicity
 stored in main memory.

 input to program:

     <gridfile>
     <ofile>
     lmean, dist1, dist2, nmax, ine

 'lmean'  if true the mean of the data is removed prior to FFT and
          windowing.

 'dist1, dist2' is the inner and outer range of the kernels in degrees

 'nmax'   is the maximum spherical harmonic wong-gore modification
          of stokes' integral. Only used for mode=1.

 'ine'    is the number of e-w points actually transformed (may be a
          subgrid).
          if the implied subgrid is greater than the
          actual grid zero padding will be done.
          ine must be even, and should be a number with a good prime
          factorization to ensure a good fft speed.

 Overwiev of files:

 file 20   data grid (free format, initiated
           with label lat1, lat2, lon1, lon2, dlat, dlon
           specifying grid boundaries)
 file 33   output file (note that results close to the
           edges are unreliable due to fft-periodicity)

 (c) Rene Forsberg, Kort- og Matrikelstyrelsen,
     Rentemestervej 8, DK-2400 Copenhagen NV, Denmark.
     First version dec 1999 - based on 'spfour'
```

Input example: Make geoid from gravity grid, remove mean prior to FFT, don’t use Wong-gore modification (see spfour) or cap size.

```
sp1d <<!
grav.gri
geoid_1d.gri
t 0 999 0 160
!
```

SPFOUR	Spherical multi-band FFT for geoid determination, upward continuation and some terrain effects

```
                    S P F O U R

program for spherical fft transformation using a sequence
of reference latitude values, with linear interpolation of
spherical formulas between latitude parallels.

the method is based on an expansion of the spherical fft
method of strang van hess (manuscripta geodaetica, 1990).
the program uses analytical transforms of the fft kernel,
this allows additionally control over the effective integration
radius.

a subgrid of the grid in 'gridfile' is analyzed. the subgrid
is specified by its sw corner (rfic, rlac) and number of points
(inn, ine). The numbers should have a good prime factorization
with many small factors for the FFT to obtain maximum speed.

input to program:

    <gridfile>
    <ofile>
    mode, lmean, nref, dist1, dist2
    (nmod1, nmod2 - only mode 1)
    (height - only mode 3 or 11)
    (<reffile> - only mode 13)
    fic, lac, inn, ine, iwn

'mode' determines the function of the program:

      1    geoid prediction from gravity data
      2    (deflections - not implemented yet)
      3    harmonic upward continuation of gravity data
     10    gravity effect of isostasy - 2nd order expansion
           (sign: positive on land. add to Bouguer anomalies)
     11    bouguer effect in airborne gravimetry at "height"
     12    geoid terrain effect (mass layer approximation)
     13    geoid RTM terrain effect (3rd order approximation)
           reffile must be exactly same grid params as the basic grid
'lmean'  if true the mean of the data is removed prior to FFT and
        windowing.
'nref'   is the number of reference parallels used. if nref = 1 just
         one fft is done, nref = 2 yields two ffts with reference on
         north and south boundary of subgrid, npar = 3 yields three
         FFT's with reference latitudes on north, central and south
         parallel, npar = 4 ...

'dist1, dist2' is the inner and outer range of the kernels in degrees

'nmod1, nmod2'  is the maximum spherical harmonic modification
         of stokes' integral. Low harmonics are removed from stokes'
         function up to degree 'nmod1' ("wong-gore"), and then linearly
         tapered to 'nmod2'. Only used for mode=1.

'height' is the height in m
```

SPFOUR (continued)

```
  'fic, lac' is geographical coordinates of the wanted SW-corner

  'inn, ine' is the number of points transformed (may be aubgrid).
           if the implied subgrid by fic,lac,inn,ine is greater than the
           actual grid zero padding will be done. The reference latitude
           will not be affected by the zero padding, and the padded values
           will not be output (the output grid might thus be smaller than
           the wanted number of points)

  'iwn'    if iwn.gt.0 the 'iwn' points closed to the edge of the internal
           selected grid will be windowed by a cosine-tpaer window.
           note: the windowing is done after taking the data mean!

  Overwiev of files:

  file 20   data grid (free format, initiated
            with label lat1, lat2, lon1, lon2, dlat, dlon
            specifying grid boundaries)
  file 33   output file (note that results close to the
            edges are unreliable due to fft-periodicity)
  file 30,31,32  scratch files, used for intermediate results
```

(c) Rene Forsberg, Kort og Matrikelstyrelsen (KMS),
Rentemestervej 8, DK-2400 Copenhagen NV, Denmark.
programmed at u of c, september 1990, based on 'tcfour' and 'geofour'
last update feb 1992, july 1992, nov 1992, may 1994
wong-gore aug 97
mode 13 and updates feb 2002

Notes:

The SPFOUR geoid mode implements Stokes function with the Wong-Gore modification, tapered over an interval. In the method the original Stokes function convolution kernel is modified by setting the lowest harmonics to zero, thus eliminating the influence of the local data at long wavelengths

$$S_{\text{mod}}(\psi) = S(\psi) - \sum_{n=2}^{N_2} \alpha(n)\frac{2n+1}{n-1} P_n \cos\psi$$

$$\alpha(n) = \begin{cases} 1 & for \quad 2 \le n \le N_1 \\ \dfrac{N_2 - n}{N_2 - N_1} & for \quad N_1 \le n \le N_2, \qquad n = 2,\dots,N \\ 0 & for \quad N \ge n > N_2 \end{cases}$$

Input example: Transformation of pointmass gravity to geoid in a 40 x 40 grid using a single reference band.

```
geoip <<!
pmdg
geoid.gri
1 f 1 0 999
0 0
55 10 40 40 0
!
```

STOKES	Evaluate Stokes or Vening-Meinesz integrals by space-domain integration, using spline densification in inner zone analogously to TC

```
                         S T O K E S

program for integration of gravity data to geoid or deflectios
of the vertical. gravity data must be given in gridded format,
e.g. produced by 'geogrid'. the integrals are evaluated at given
station locations. in a 3 x 3 innerzone around each computation point
a local spline densification of the given data is made, analogous
to the terrain effect integration programme 'tc'.
gravity grid values 9999 or larger signals unknown values. such
cells are not taking part in the integration, but 9999's must not be
found in the innerzone (7 x 7 subgrid around computation points).

input:

gridfile,
stationfile,
outfile,
mode, lmean, psi1, psi2 (degrees)

where

mode = 1: stokes integration
       2: vening-meinesz
lmean: true if mean of gravity data to be removed, false otherwise

the stationfile must contain the prediction point coordinates
in standard format (no, lat, lon, height, ..)

programmer: rene forsberg, danish geodetic institute,
vax version, university of new south wales, jan 1989
```

Input example: Stokes' integration of a gravity grid "grav.gri" at locations "wanted.pts" (a standard data list file), remove mean and integrate with cap size range 0 to 9 degrees.

```
stokes <<!
grav.gri
wanted.pts
stokes.out
1 t 0 9.0
!
```

TC	Prism integration of terrain effects (topographic, topographic-isostatic, RTM and classical terrain corrections)

TC

Program to compute terrain effects on gravimetric quantities. various modes of terrain effects may be computed:

1) Direct topographic effect of all masses above sea-level, assuming the density to be constant.

2) Topographic/isostatic reduction using an airy-model.

3) Gravimetric terrain correction (effect of topographic irregularities with respect to a spherical bouguer plate) - in this case with conventional inversion of sign so that the computed quantities should be added to observations.

4) RTM (residual terrain model) effects, i.e. the effects of the topographic irregularities with respect to a mean surface. Pre-computed terrain corrections (e.g., from very dense DTM's) can be used, but only for points on the topography.

The program computes gravity disturbance, deflections of the the vertical and/or geoid undulations (height anomalies) in units of mgal, arcsec and m respectively, or second-order derivatives in eotvos units (1 E = 0.1 mgal/km)

The computation is based on two digital elevation models, a detailed and a coarse, which is used in the inner and outer zones respectively. the two grids are assumed to have 'common' boundaries, which is the case if the coarse grid have been constructed from the detailed grid by averaging.

The integration of the terrain effects is performed using the formulas for the gravitational effects of a homogeneous rectangular prism. depending on the geometry and accuracy various formulas are used: exact formulas, spherical harmonic expansion (mc millan) or centered point mass approximation. parameters to determine formula choice and accuracy may be set by the user in subroutine 'prism1'.

The computation may be done out to a fixed distance from the computation point or for all masses in a given area. the detailed elevation grid is used out to a specified radius, which should be at least 2 times the gridspacing in the outer grid. in the local neighbourhood around the point the terrain information may be densified using a bicubic spline interpolation. if the computation point is known to be at the surface of the topography, the terrain model is modified smoothly in an innerzone to give the correct elevation at the computation point. (a circular area around the computation point may be skipped for for special applications. in this case all prisms with a center within the specified skip circle is not evaluated. for rtm a harmonic correction is not done in this case.) the curvature of the earth is taken into account to the first order.

TC (continued)

If residual terrain effects is wanted, an additional mean elevation grid (e.g. 30' x 30' mean heights) must be specified. This mean elevation grid should be smooth, e.g. produced by the program 'tcgrid'

The program may use grid files either in utm or geographic coordinates. If utm grids are used, all grids must be in the same system (system is specified in gridlabel, see 'rdelev'). computation points may be given in geographical coordinates, but are always output in utm when utm grids are used.

The grids must be in standard format, i.e. stored rowwise from N to S with label definining grid, see 'rdelev'

Unknown heights in the height data files may be signalled by the value 9999. The program will not output the results for points where the computations have encountered unknown height values. A warning will be written on unit 6, however.
NB: a reference grid must not contain unknown height values.

input

```
statfile,           (station list file)
dtmfile1,           (detailed elevation grid)
dtmfile2,           (coarse elevation grid)
dtmfile3,           (reference elevation grid)
outfile,            (output file)
itype, ikind, izcode, istyp, rho
fi1, fi2, la1, la2
r1, r2,
  (rskip - only for r1 < 0),
  (rtc - only for ikind = 5)
  (latmin, latmax, lonmin, lonmax, dlat, dlon, elev - for istyp = 0)
```

dummy file names must be specified if file is not used.
the following codes determine the quantities to be calculated, the computation kind and the innerzone options:

```
itype    1  gravity disturbance (dg) - mgal
         2  deflections (ksi, eta) - arcsec
         3  height anomaly (ha) - meter
         4  dg, ksi, eta
         5  gravity anomaly (dg - ind.eff.)
         6  anomaly, ksi, eta, ha
         7  tzz - vertical gravity gradient (z positive up)
         8  txx, tyy, tzz
         9  all gradients

ikind    1  topographic effect
         2  isostatic effect
         3  terrain corrections
         4  residual terrain effetcs
         5  do, using precomputed terrain corrections to dist 'rtc' km
            (if tc-values are missing, ordinary rtm effects computed)

izcode   0  station on terrain, change station elevation
         1          do       , change terrain model
         2          do       , change terrain model in land points oy
```

TC (continued)

```
         3  station free
         4  station free, no spline densification
            (dma special - if topography is above computation
            level, the terrain effects will be computed at the
            topography level in both mode 3 and 4)

istyp    0  no statfile, compute in grid
         1  compute effects in statfile
         2  add effects to value in statfile
         3  subtract effects
         4  statfile with 80-char KMS gravity recs (80-char output
            for ikind=3, otherwise normal output format)
       neg  as 1,2 or 3 but UTM data in statfile

fi1, fi2, la1, la2   fixed maximum area for which ter-
            rain effect is computed, irrespectively of the
            specified computational radii. unit: deg (utm: m)

r1          minimum computation distance of inner grid (km).
            the inner grid is actually used in the smallest
            'subsquare' of the coarse grid, covering a circle
            of radius r1. (special option: negative r1 signals
            computation where topography within a radius rskip
            is  not  computed. rskip must be input after 'r2')

r2          maximal radius of computation (km). if r2 = 0
            no outer grid is used. if a fixed-area computation
            is wanted 'r2' must be sufficiently large.
```

The station list consists of a
listing of no, latitude, longitude (in degrees) and elevation.

Marine convention: computations at sea (elevation < 0) are done at sea level. The depths at a point are shown as a negative height. Caution: Only use izcode=1 for marine data if depths are proper! Use izcode=2 for terrain reductions where sea depths are not used.

Grid option (istyp=0):
Computations are performed in the points of a grid, defined by latitudes 'latmin' to 'latmax' and longitudes 'lonmin' to 'lonmax', with spacing 'dlat' and 'dlon' specified in degrees. Elevation of points: 'elev', unless izcode=1 or 2. Output file is in grid format as well if only one value computed

Note: If the terrain grids does not cover the 'computation area' as defined by the maximum zone and the radii, elevation 9999 (signalling missing heights) are assumed by the program. Stations encountering missing heights during computations are not computed to end and not output by program.
Since the inner grid is always used out to a boundary in the outer grid which is at least 'r1' km away, the 'missing heights' condition may be met even if the point is more than 'r1' away from the inner grid outer boundaries.

Note: Especially for fixed area reductions, be very careful to avoid extension of the grids with 9999's. This may be avoided by specifying a slightly smaller max area, and verified from the grid specification output of this program.

TC (continued)

```
Original version described in ohio state university, dept. of geodetic science and surveying,
report 355, 1984. note that the grid specification have been  changed  from this report:
limits of grids must allways refer to the  c e n t e r  of the cells, as natural for point
heights.

Programmer: Rene Forsberg, Ohio State University / Danish Geodetic Institute, july 1983.
modified for cdc november 1984. grid specification changedto specify boundaries of grid cell
centers. Splineinterpolation have been changed from 5x5 to 7x7 innerzone
modified for rc8000-fortran, d. arabelos june 86. changes and inclusion of second order
gradients, rf jan 87.modified for cdc fortran, rf february 87.modified for double precision,
9999-unknown options implemented, rf november 88.
modified for swedish national projection, kms unix april 89, signalled by utm zone = 99
last updated: march 94, rf (80-char input), dec 94, rf (tcnew-dmaac option),jan 96
(add/subtract, inner zone change, exact prism pot)
```

Note: Detailed background of this program may be found in Appendix in Department of Geodetic Science, Ohio State University, Report 355.

Input example: Computation of RTM gravity effects in a file “fa.rtm” for gravity points at locations “fard.osu”. The computations are using a detailed height grid “sbc.dtm” with 1-km heights out to a minimal distance of 20 km to fill out the voids in an outer, more coarse 5-km grid (“sbc.dtm_5”) which is used out to an outer limit of 47-56N, 126-112 W (southern British Columbia). Density 2.67 is assumed, relative to a mean height surface defined by the grid in “sbc_dtm.ref”, produced by TCGRID.

```
   tc <<!
   fa.redosu
   sbc.dtm
   sbc.dtm_5
   sbc.dtm_ref
   fa.rtm
   5 4 1 1 2.67
   47 56 -126 -112
   20.0 999
   -1
   !
```

The innerzode code in the above example was 1, requiring the DEM heights in “ sbc.dtm” be forced to match with the heights given in the data file “fa.redosu”. Note that the “izcode” option should be used with great care, as aliasing errors may be generated, especially harmful for geoid computations!

TCFOUR — Computation of terrain effects by FFT convolutions in planar approximation

```
                     T C F O U R

program for fft analysis of digital terrain models.
space domain expressions of the integral kernels are
transformed rather than using the fourier domain analytical
kernels, in order to allow control over integration
radii.

a subgrid of the grid in 'dtmfile1' is analyzed, specified
by its sw-corner (fic,lac) and number of
points. both 'in' and 'ie' must be  e v e n .
a reference height grid may be subtracted (for rtm effects).

input to program:

    <dtmfile1>
    <dtmfile2>
    <ofile>
    mode, lref, dist1, dist2 or height,
    fic1, lac1, in1, ie1
   (fic2, lac2, in2, ie2 - for mode 2 only)

'lref' is reference grid specification: f none, t subtract ref.
    the reference grid name must be specified as 'dtmfile2', this file
    is only opened if lref is true, and in mode 2.

'mode' determines the function of the program:

mode = 0: simple filtering. 'dist1' gives the wavelength
          for cut-off in units of km.
          'dist2' > 0: low pass, < 0: high pass.
       1: covariance function and power spectra,
          (2-d cov.fct. output as dtm with c(0,0)=1000)
         'dist1, dist2' is not used.
       2: terrain corrections, using grid1 to distance 'dist1',
          grid2 to distance 'dist2' (both in km).
          (the detailed grid is used out to a square of half side-
          length 'dist1', border points in both inner and outer
          grid being attenuated to compensate edge effects.)
       3: terrain corrections from 'dist1' to 'dist2' (km).
       4: rtm gravity effect, computed as terrain correction
          to distance 'dist1' and bouguer reduction to reference
          level.
       5: geoid effect for residual topography, condensation
          approximation. only one grid, second grid specification
          is the reference grid (lref = false means no ref used).
          effects will be computed in elevation 'dist2' (km) above
          the topography (i.e., above the condensation level).
          computed to distance 'dist1'.
       6: isostatic geoid effects, second order expansion, in
          elevation 'dist2' above the geoid. (mixed continental and
          oceanic area using equivalent sea/rock root conversion).
          computed to distance 'dist1' (km).
       7: gravity effect of the isostatic compensation,
          computed to distance 'dist1' (km).
```

TCFOUR (continued)

```
       8: rtm deflections of the vertical (condensation approximation)
          (written on outfile as two consecutive grids in unit arcsec)
          effect computed between distances 'dist1' to 'dist2' (km).

The program recognizes grids in utm by the label on 'dtmfile1'.
if utm grids are used (fic, lac) must be northing and easting in km.
if (fic, lac) = (0, 0) the sw-corner of the grid is used.

overwiev of files:

file 20   elevation data grid (free format, initiated
          with label lat1, lat2, lon1, lon2, dlat, dlon
          specifying grid boundaries)
file 21   possible elevation grid for outer zones
file 22   possible reference elevation grid
file 32   output file (note that results close to the
          edges are unreliable due to fft-periodicity)
file 30,31  scratch files, used for intermediate results

- note: results for mode 1 are listed with basic distance
unit degree.

© rf, danish geodetic institute, october 1984
modified for cdc, december 1984, rf
changed and updated for unsw vax, rf mar 1989
mode 4 errors updated - dec 95
```

Note: TCFOUR do not implement zero padding. Zero-padding may be done “manually” by GCOMB.

Input example: Computing RTM terrain effects approximately – first computing terrain corrections, then making Bouguer reduction to mean height level (note this is an approximate procedure; TC might be better).

```
tcfour <<!
dem.gri
dem.ref
pip.out
4 t 0 100
55 10 120 120
!
```

TCGRID	Support program for making average DEM grids and mean terrain surfaces for RTM method in TC

TCGRID

Program to produce a mean elevation grid of a file containing a digital terrain model (standard format or ngs modified). The mean grid is obtained by simple averaging. areas with no elevations are given value 9999. if an individual grid element is covered partly by elevations, only these will be averaged. If no data is available the unknown flag 9999 will be written.

If wanted, the mean elevation grid may be low-pass filtered using a moving-average window of 'iffi' x 'ifla' cells. If 'ifla' = -1 the longitude resolution is changed to match the latitude variation.

<u>Program input:</u>

inputfile,
outputfile,
itype, fimin, fimax, lamin, lamax, idfi, idla, iffi, ifla

where

'itype' specifies dtm-format type (0, 1: std, 2: ngs),
'fimin, fimax, lamin, lamax' specifies wanted area (degrees)
(if all are zero the entire grid is averaged),
'idfi' and 'idla' specifies mean grid average size (in units
of given grid - integers only), and
'iffi' and 'ifla' (odd numbers) specify averaging window
(= 0 or 1 means no low pass filtering).

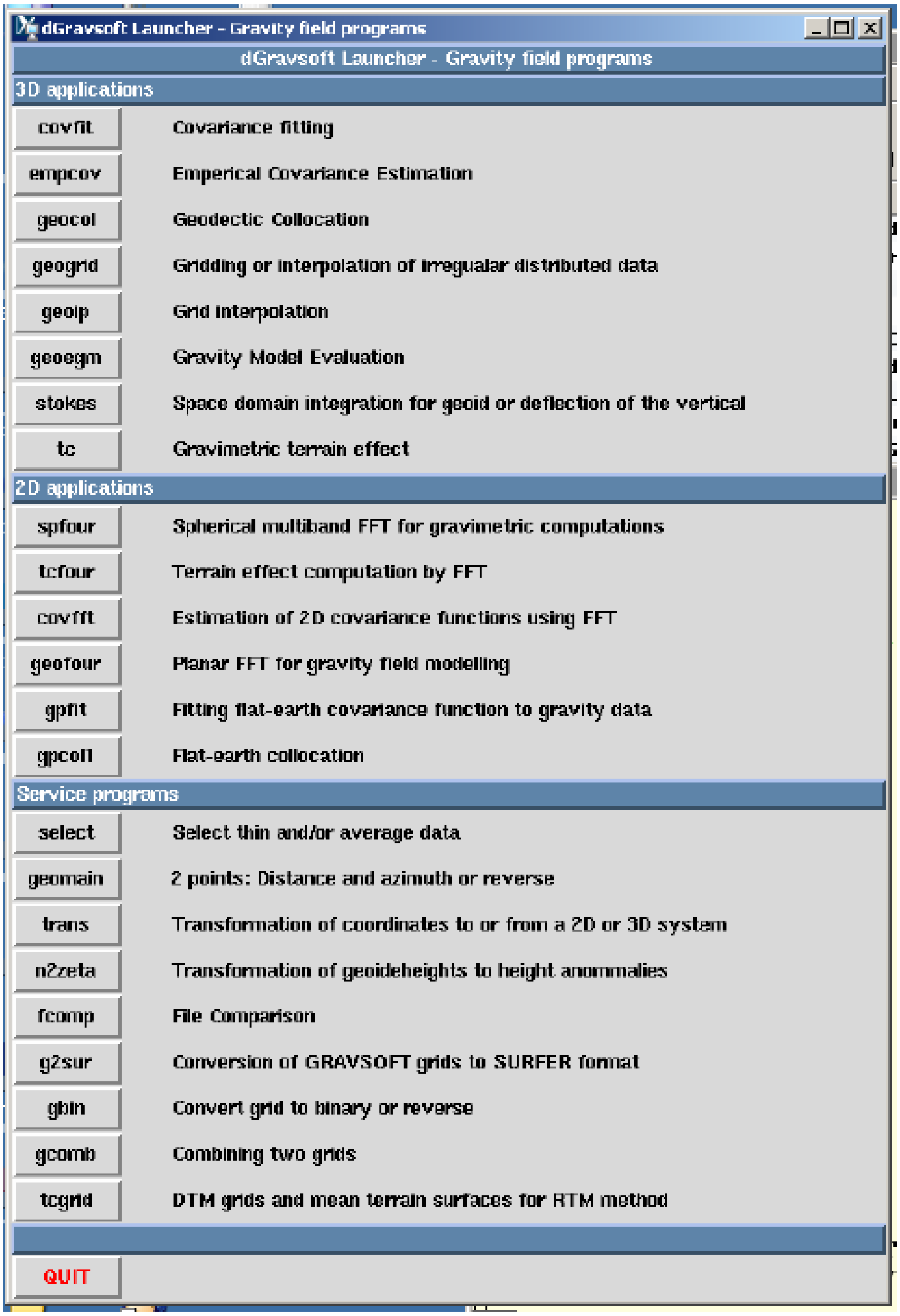
note: grids are specified by a label (fi1, fi2, la1, la2, dfi, dla - degrees), where the coordinate limits refer to the center of the outer blocks. this implies that mean elevation grids will typically have somewhat "ackward" labels.

values with integer part 9999 signals unknown values.
if more than half of the values for an average is unknown,
the average value itself is considered unknown and assigned 9999.
it is not allowed with 9999-values in filtering operations.

© *rene forsberg/KMS, june 1983. updated dec 7, 1983. modified for cdc, with inclusion of filtering, jan 85. Modified for real input/output and 9999-codes (unknown), November 1988.*

Appendix C. Example of Python interface (New Mexicio examples from International Geoid School)

We here show step-by-step how the GRAVSOFT programs are used to predict reduced height-anomalies from free-air gravity and height-anomalies derived from GPS-levelling in New Mexico. First an overview of the software:

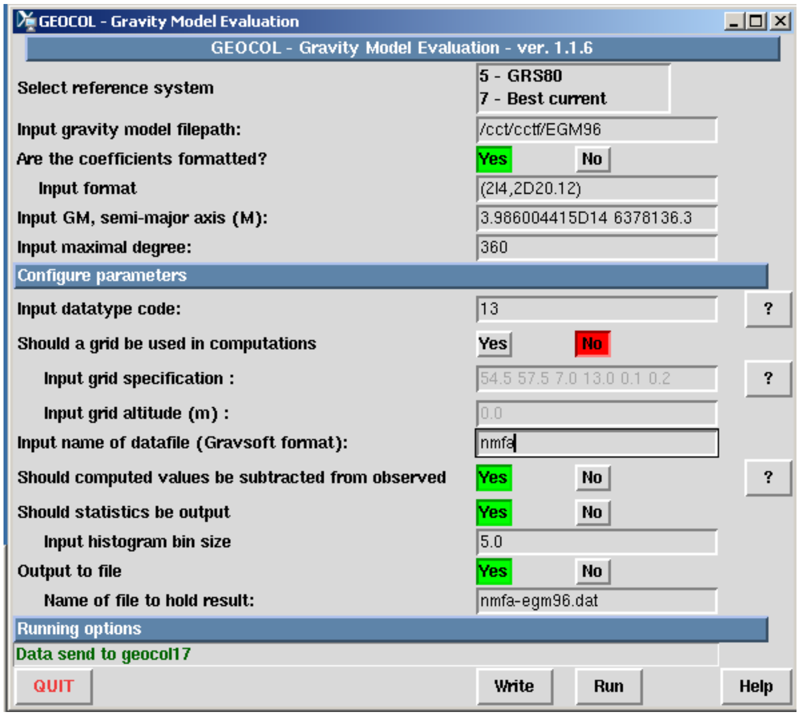


dGravsoft Launcher - Gravity field programs

dGravsoft Launcher - Gravity field programs

3D applications	
covfit	Covariance fitting
empcov	Emperical Covariance Estimation
geocol	Geodectic Collocation
geogrid	Gridding or interpolation of irregualar distributed data
geoip	Grid interpolation
geoegm	Gravity Model Evaluation
stokes	Space domain integration for geoid or deflection of the vertical
tc	Gravimetric terrain effect
2D applications	
spfour	Spherical multiband FFT for gravimetric computations
tcfour	Terrain effect computation by FFT
covfft	Estimation of 2D covariance functions using FFT
geofour	Planar FFT for gravity field modelling
gpfit	Fitting flat-earth covariance function to gravity data
gpcoll	Flat-earth collocation
Service programs	
select	Select thin and/or average data
geomain	2 points: Distance and azimuth or reverse
trans	Transformation of coordinates to or from a 2D or 3D system
n2zeta	Transformation of geoideheights to height anommalies
fcomp	File Comparison
g2sur	Conversion of GRAVSOFT grids to SURFER format
gbin	Convert grid to binary or reverse
gcomb	Combining two grids
tcgrid	DTM grids and mean terrain surfaces for RTM method

The first steps are to subtract the contribution from a global model (here EGM96) from the free-air anomalies (file nmfa) and the height anomalies (nmzeta.dat). This is followed by steps where residual terrain effects are subtracted

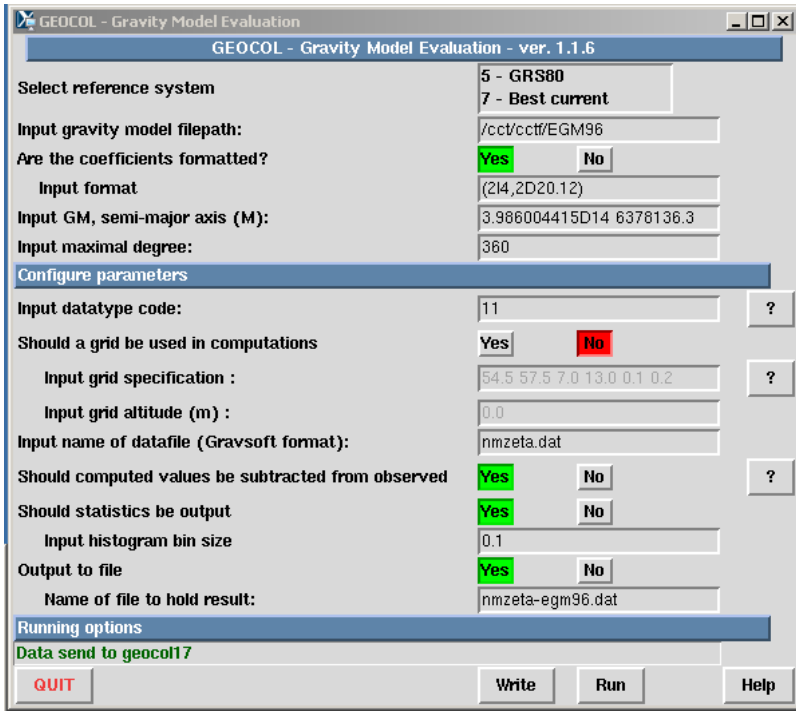


GEOCOL - Gravity Model Evaluation

GEOCOL - Gravity Model Evaluation - ver. 1.1.6

Select reference system	5 - GRS80 7 - Best current
Input gravity model filepath:	/cct/cctf/EGM96
Are the coefficients formatted?	[✓ Yes] No
Input format	(2I4,2D20.12)
Input GM, semi-major axis (M):	3.986004415D14 6378136.3
Input maximal degree:	360
Configure parameters	
Input datatype code:	13
Should a grid be used in computations	Yes [✓ No]
Input grid specification :	54.5 57.5 7.0 13.0 0.1 0.2
Input grid altitude (m) :	0.0
Input name of datafile (Gravsoft format):	nmfa
Should computed values be subtracted from observed	[✓ Yes] No
Should statistics be output	[✓ Yes] No
Input histogram bin size	5.0
Output to file	[✓ Yes] No
Name of file to hold result:	nmfa-egm96.dat
Running options	

Data send to geocol17



GEOCOL - Gravity Model Evaluation - ver. 1.1.6

Select reference system: 5 - GRS80 / 7 - Best current

Input gravity model filepath: /cct/cctf/EGM96

Are the coefficients formatted? [✓ Yes] No

Input format: (2I4,2D20.12)

Input GM, semi-major axis (M): 3.986004415D14 6378136.3

Input maximal degree: 360

Configure parameters

Input datatype code: 11 ?

Should a grid be used in computations: Yes [✓ No]

Input grid specification : 54.5 57.5 7.0 13.0 0.1 0.2 ?

Input grid altitude (m) : 0.0

Input name of datafile (Gravsoft format): nmzeta.dat

Should computed values be subtracted from observed: [✓ Yes] No ?

Should statistics be output: [✓ Yes] No

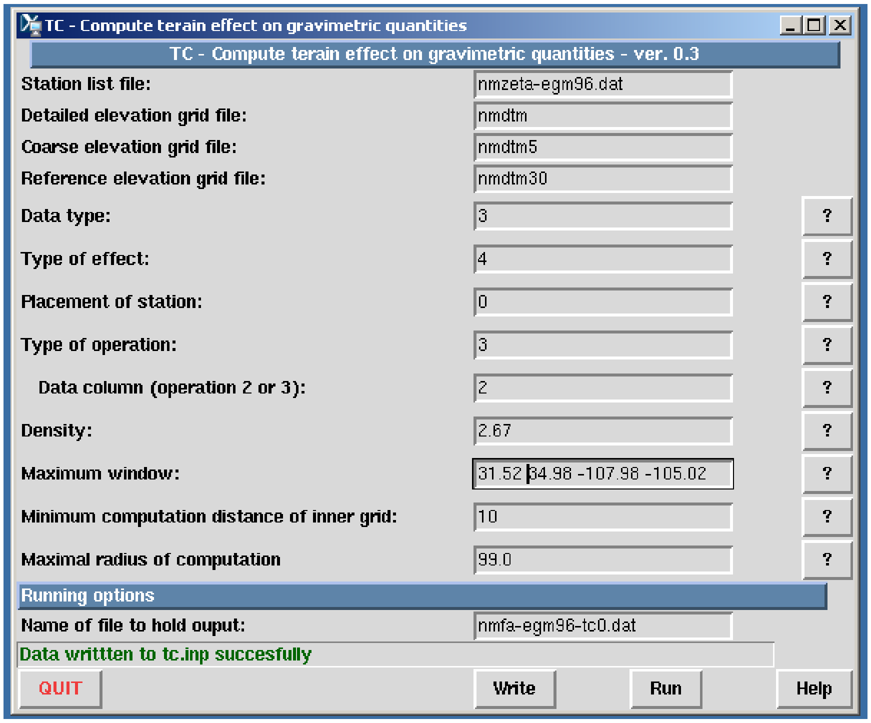
Input histogram bin size: 0.1

Output to file: [✓ Yes] No

Name of file to hold result: nmzeta-egm96.dat

Running options

Data send to geocol17



TC - Compute terain effect on gravimetric quantities - ver. 0.3

Station list file: nmzeta-egm96.dat

Detailed elevation grid file: nmdtm

Coarse elevation grid file: nmdtm5

Reference elevation grid file: nmdtm30

Data type: 3 ?

Type of effect: 4 ?

Placement of station: 0 ?

Type of operation: 3 ?

Data column (operation 2 or 3): 2 ?

Density: 2.67 ?

Maximum window: 31.52 34.98 -107.98 -105.02 ?

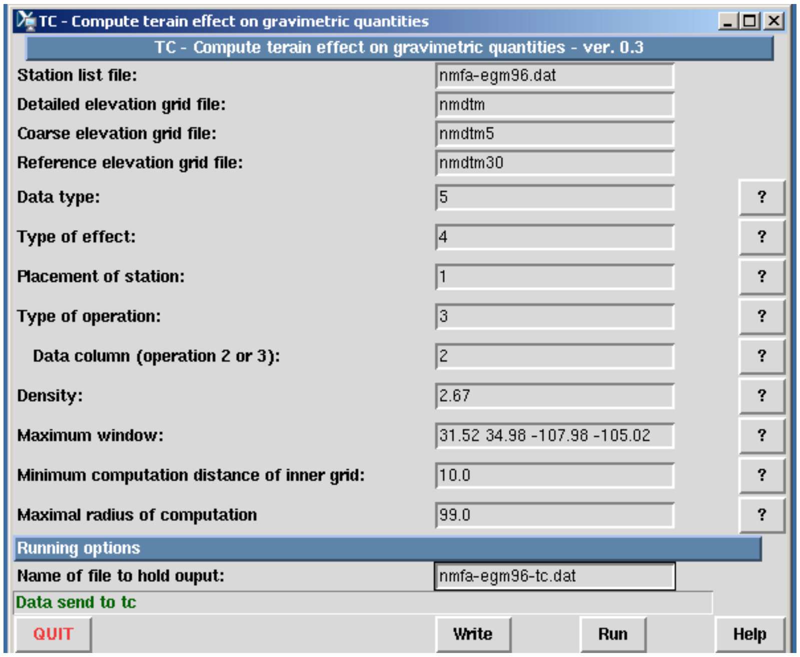
Minimum computation distance of inner grid: 10 ?

Maximal radius of computation: 99.0 ?

Running options

Name of file to hold ouput: nmfa-egm96-tc0.dat

Data writtten to tc.inp succesfully



TC - Compute terain effect on gravimetric quantities - ver. 0.3

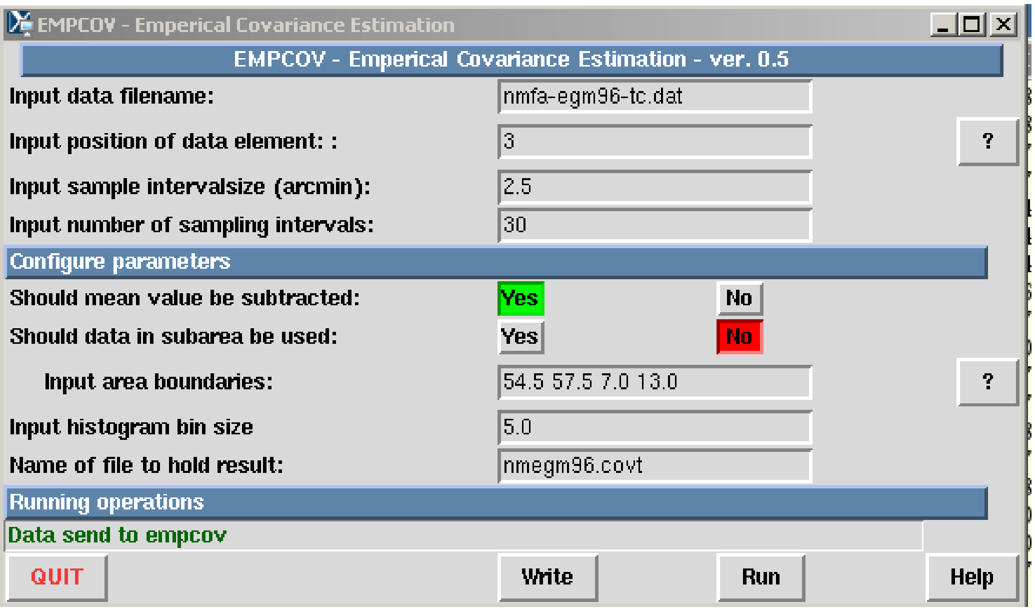
Station list file: nmfa-egm96.dat
Detailed elevation grid file: nmdtm
Coarse elevation grid file: nmdtm5
Reference elevation grid file: nmdtm30
Data type: 5
Type of effect: 4
Placement of station: 1
Type of operation: 3
Data column (operation 2 or 3): 2
Density: 2.67
Maximum window: 31.52 34.98 -107.98 -105.02
Minimum computation distance of inner grid: 10.0
Maximal radius of computation 99.0

Running options

Name of file to hold ouput: nmfa-egm96-tc.dat

Data send to tc

Then the empirical covariance function is estimated and fitted using COVFIT.



EMPCOV - Emperical Covariance Estimation - ver. 0.5

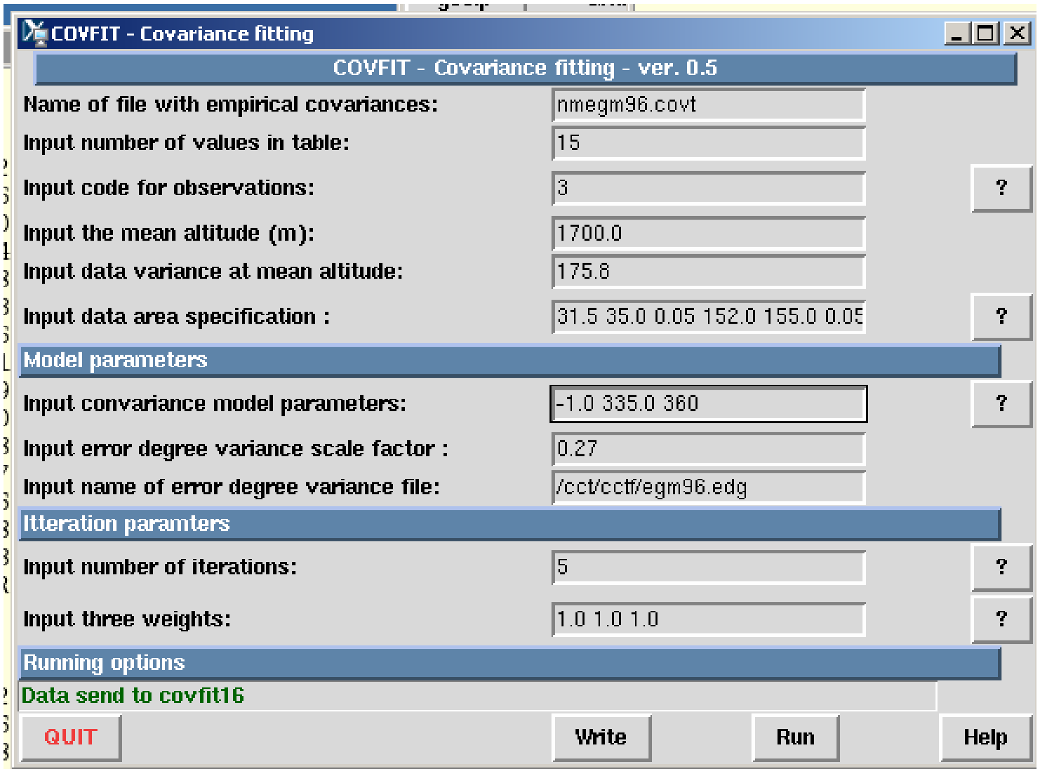
Input data filename: nmfa-egm96-tc.dat
Input position of data element: : 3
Input sample intervalsize (arcmin): 2.5
Input number of sampling intervals: 30

Configure parameters

Should mean value be subtracted: [✓ Yes] No
Should data in subarea be used: Yes [✓ No]
Input area boundaries: 54.5 57.5 7.0 13.0
Input histogram bin size 5.0
Name of file to hold result: nmegm96.covt

Running operations

Data send to empcov



Name of file with empirical covariances: nmegm96.covt

Input number of values in table: 15

Input code for observations: 3

Input the mean altitude (m): 1700.0

Input data variance at mean altitude: 175.8

Input data area specification : 31.5 35.0 0.05 152.0 155.0 0.05

Input convariance model parameters: -1.0 335.0 360

Input error degree variance scale factor : 0.27

Input name of error degree variance file: /cct/cctf/egm96.edg

Input number of iterations: 5

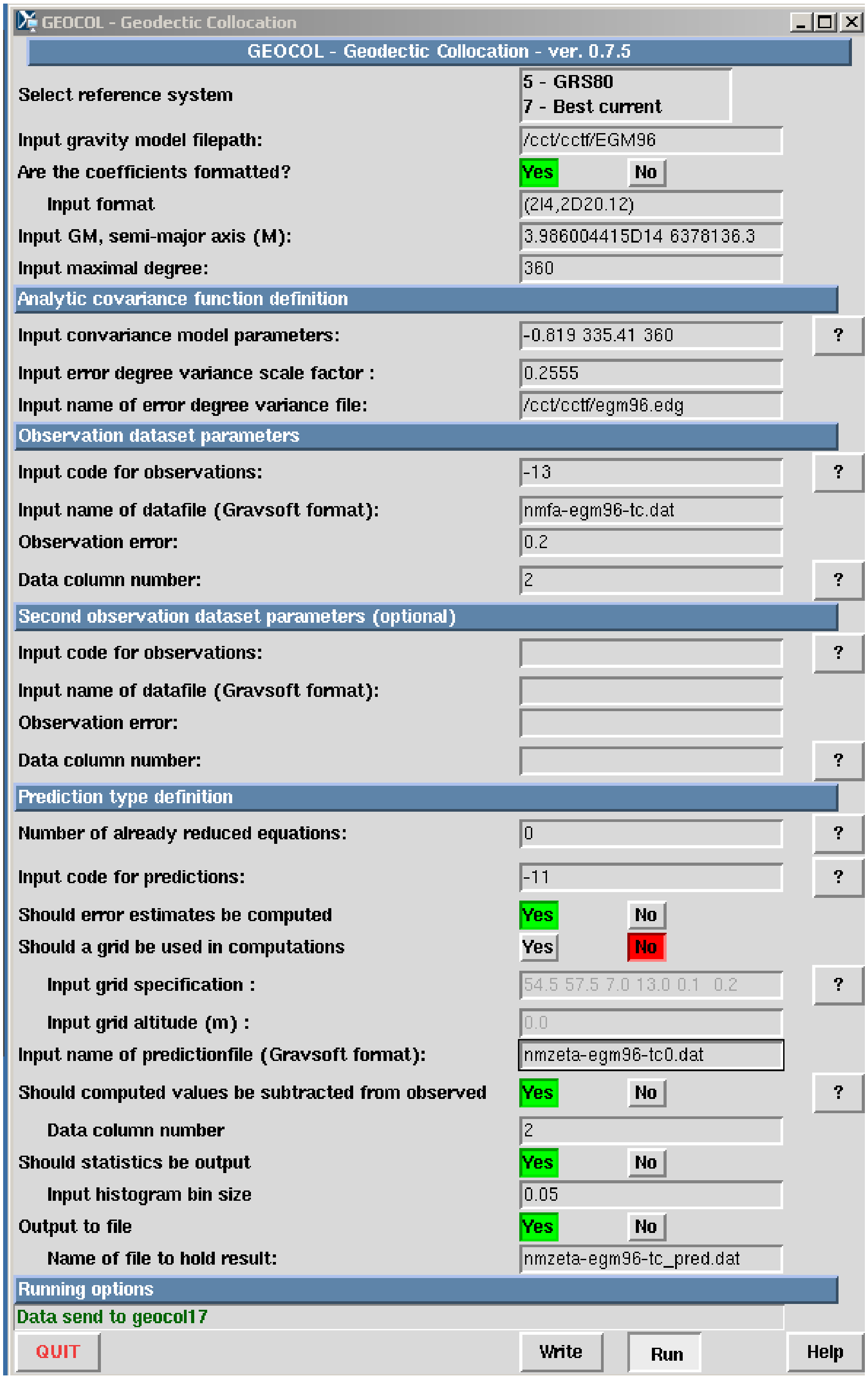
Input three weights: 1.0 1.0 1.0

Data send to covfit16

Final part of covfit.log with the results of the fitting.
The values VARDG=335.41, AA=0.2555 and RE-RB = -819.7 are used as parameters for GEOCOL. The column with the heading OBS are the values estimated by EMPCOV.

```
 RESULT OF ITERATION NO.  5:
 --------------------------
                                 AA           A          RB-RE
 NEW VALUE:                   0.255454     682639.     -819.274
 LAST ADJ.:                   0.000109       -345.        1.570

     KP  KQ     PSI      HP       HQ       OBS      MODEL       ERR   DIF/ERR

   1   3   3    0.000   1700.0   1700.0  175.8490  174.1635    3.4091  0.4944
   2   3   3    0.042   1700.0   1700.0  136.9017  147.2436    6.5481 -1.5794
   3   3   3    0.083   1700.0   1700.0  103.9408  105.6302    6.7251 -0.2512
   4   3   3    0.125   1700.0   1700.0   73.8018   70.2822    7.0395  0.5000
   5   3   3    0.167   1700.0   1700.0   47.9292   43.5676    7.3255  0.5954
   6   3   3    0.208   1700.0   1700.0   27.3777   24.3020    7.6195  0.4037
   7   3   3    0.250   1700.0   1700.0   13.9125   10.0831    7.9262  0.4831
   8   3   3    0.292   1700.0   1700.0    1.0594    0.2856    8.2305  0.0940
   9   3   3    0.333   1700.0   1700.0   -4.8086   -5.8998    8.5167  0.1281
  10   3   3    0.375   1700.0   1700.0  -10.9165   -9.5143    8.8516 -0.1584
  11   3   3    0.417   1700.0   1700.0  -13.9891  -11.0021    9.0615 -0.3296
  12   3   3    0.458   1700.0   1700.0  -17.2667  -10.9209    9.3502 -0.6787
  13  3  3   0.500   1700.0   1700.0  -18.5129   -9.7227    9.7607 -0.9006
  14  3  3   0.542   1700.0   1700.0  -19.3898   -7.7919    9.9716 -1.1631
  15  3  3   0.583   1700.0   1700.0  -18.5653   -5.5235   10.2708 -1.2698
 RMS VALUE OF DIFFERENCES/ERRORS:      0.739308
 (RELATIVE CHANGE:                     0.000002).

  TAU(J) USED IN THE CX MATRIX      0.10E+01      0.10E+01      0.10E+01
RESULTS IN VARIANCE OF GRAVITY ANOMALIES:
  1'TH ROW OF INVERSE MATRIX      0.5870E-01   -0.1459E-01   -0.3330E-01
  2'TH ROW OF INVERSE MATRIX     -0.1459E-01    0.1312E-01    0.4660E-01
  3'TH ROW OF INVERSE MATRIX     -0.3330E-01    0.4660E-01    0.1796E+00
  STD.DEV.       0.618907E-01  0.781821E+05  0.347233E+03
  STD.DEV.*RMS   0.457563E-01  0.578007E+05  0.256712E+03
 RESULTS IN VARIANCE OF GRAVITY ANOMALIES:   335.41 MGAL**2.

 N       RATIO      AA      A         RE-RB    VARG IT
 360  0.7393D+00  0.2555  0.6826D+06  -819.27   335.41    5
```

In the following step we predict the reduced height-anomalies from the reduced gravity anomalies. This is then followed by a step where both the reduced gravity data and the reduced height anomalies are used to predict heigt anomalies at terrain level. Note, tat in the final step we use that 2920 columns of the normal-equations already have been reduced.



GEOCOL - Geodectic Collocation - ver. 0.7.5

Select reference system: 5 - GRS80
7 - Best current

Input gravity model filepath: /cct/cctf/EGM96

Are the coefficients formatted? [✓ Yes] No

Input format: (2I4,2D20.12)

Input GM, semi-major axis (M): 3.986004415D14 6378136.3

Input maximal degree: 360

Analytic covariance function definition

Input convariance model parameters: -0.819 335.41 360

Input error degree variance scale factor : 0.2555

Input name of error degree variance file: /cct/cctf/egm96.edg

Observation dataset parameters

Input code for observations: -13

Input name of datafile (Gravsoft format): nmfa-egm96-tc.dat

Observation error: 0.2

Data column number: 2

Second observation dataset parameters (optional)

Input code for observations:

Input name of datafile (Gravsoft format):

Observation error:

Data column number:

Prediction type definition

Number of already reduced equations: 0

Input code for predictions: -11

Should error estimates be computed [✓ Yes] No

Should a grid be used in computations Yes [✓ No]

Input grid specification : 54.5 57.5 7.0 13.0 0.1 0.2

Input grid altitude (m) : 0.0

Input name of predictionfile (Gravsoft format): nmzeta-egm96-tc0.dat

Should computed values be subtracted from observed [✓ Yes] No

Data column number: 2

Should statistics be output [✓ Yes] No

Input histogram bin size: 0.05

Output to file [✓ Yes] No

Name of file to hold result: nmzeta-egm96-tc_pred.dat

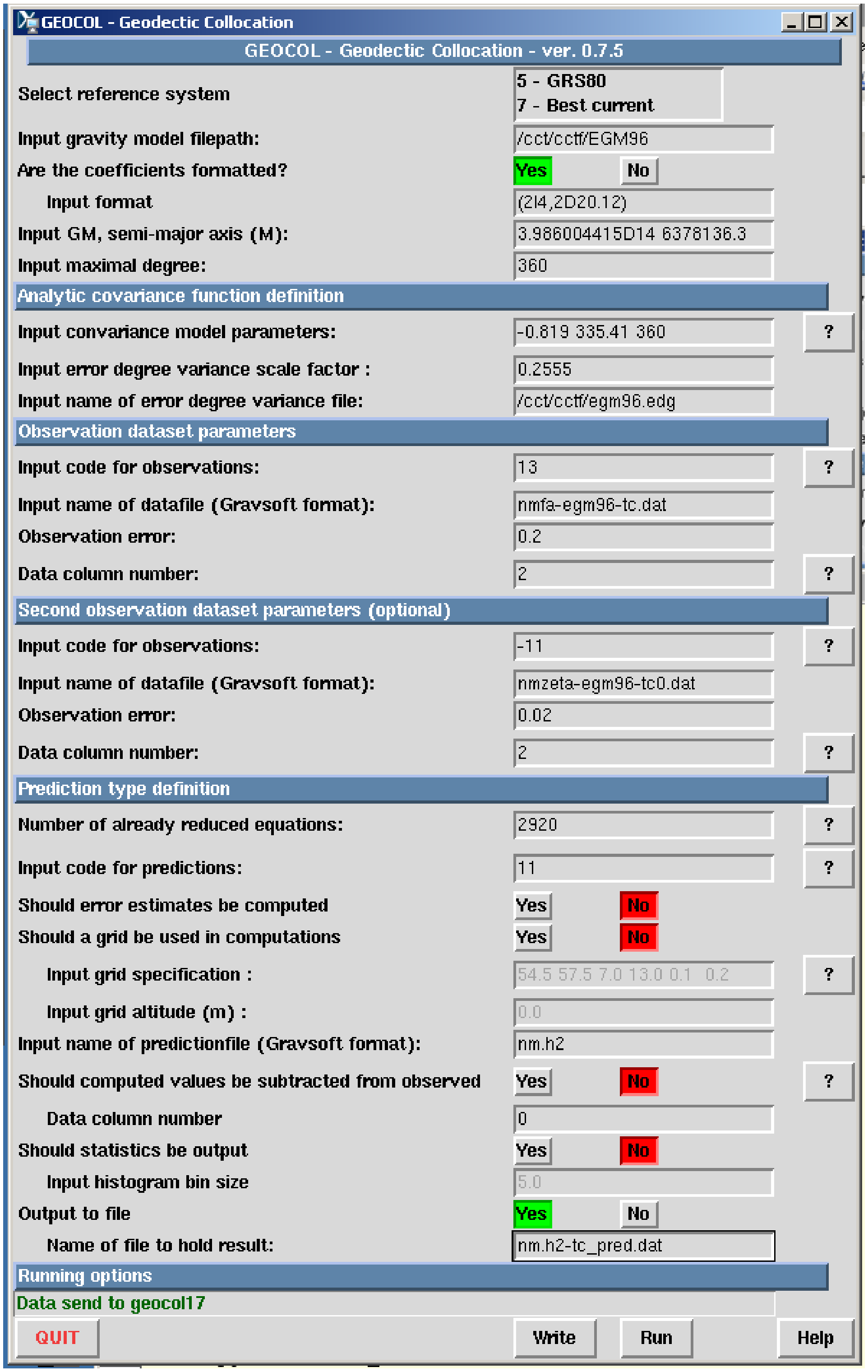
Running options

Data send to geocol17

The comparison between predicted anf observed height anomalies are found in the file geocol.log, of which here is shown the last part:

```
 COMPARISON OF PREDICTIONS AND OBSERVATIONS
 DATA TYPE = 11
 NUMBER:      20
          OBSERVATIONS     PREDICTIONS     DIFFERENCE
 MEAND            -0.897275       -0.000249       -0.897026        0.057190
 ST.DEVI.          0.159149        0.145572        0.051812        0.014752
 MAX              -0.632800        0.210118       -0.789369        0.090756
 MIN              -1.267500       -0.320443       -0.987109        0.045835
 DISTRIBUTION OF DIFFERENCES, UNITS:  0.050000
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0      20
 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1  0  1  2  3  4  5  6  7  8  9 10 OUTSIDE
```

The result shows that we are able to predict the reduced height anomalies with an error standard deviation of 0.05 m. Due to the bias, the histogram values can not be used.



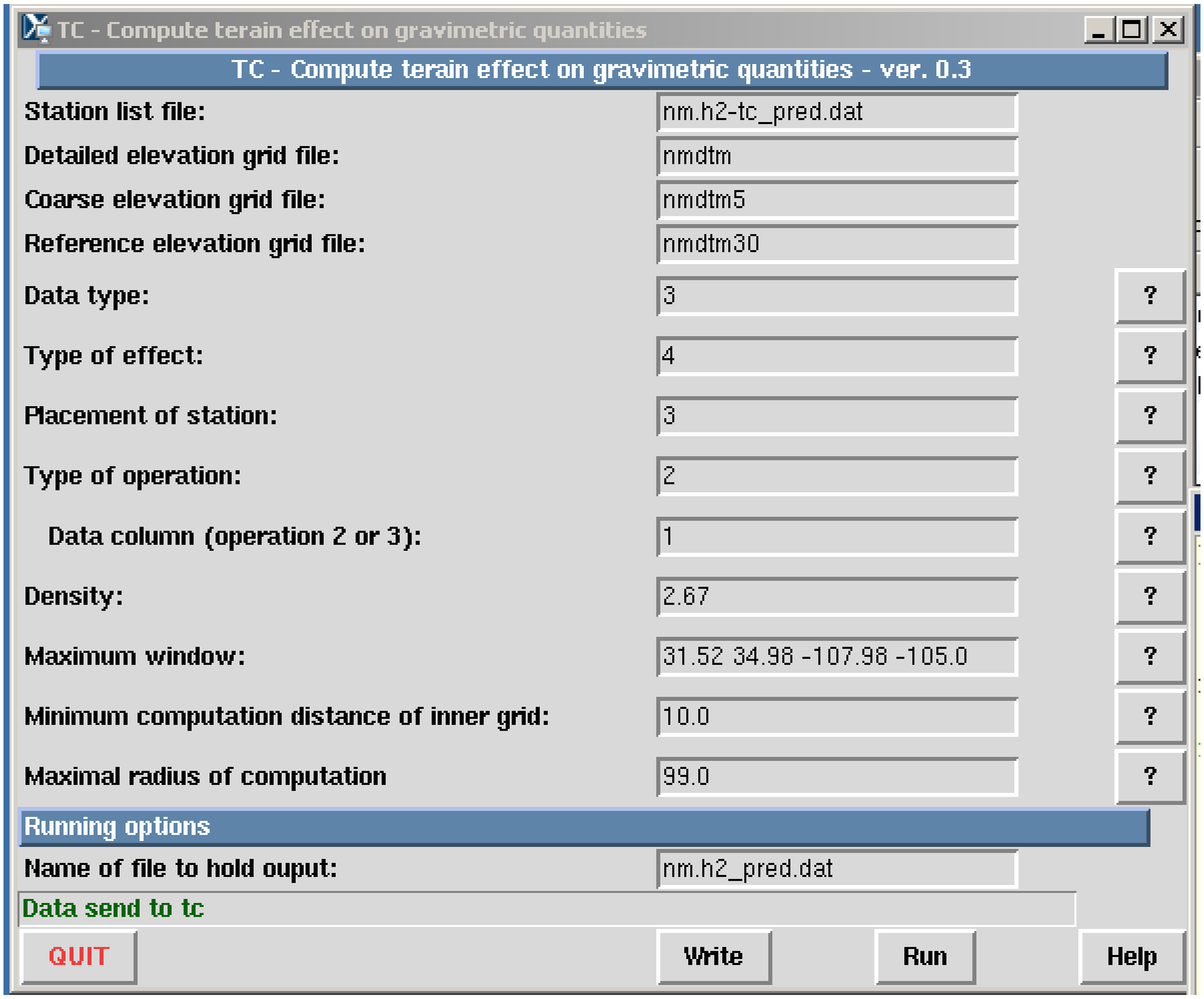
GEOCOL - Geodectic Collocation

GEOCOL - Geodectic Collocation - ver. 0.7.5

Select reference system	5 - GRS80 7 - Best current
Input gravity model filepath:	/cct/cctf/EGM96
Are the coefficients formatted?	[✓ Yes] No
Input format	(2I4,2D20.12)
Input GM, semi-major axis (M):	3.986004415D14 6378136.3
Input maximal degree:	360
Analytic covariance function definition	
Input convariance model parameters:	-0.819 335.41 360
Input error degree variance scale factor :	0.2555
Input name of error degree variance file:	/cct/cctf/egm96.edg
Observation dataset parameters	
Input code for observations:	13
Input name of datafile (Gravsoft format):	nmfa-egm96-tc.dat
Observation error:	0.2
Data column number:	2
Second observation dataset parameters (optional)	
Input code for observations:	-11
Input name of datafile (Gravsoft format):	nmzeta-egm96-tc0.dat
Observation error:	0.02
Data column number:	2
Prediction type definition	
Number of already reduced equations:	2920
Input code for predictions:	11
Should error estimates be computed	Yes [✓ No]
Should a grid be used in computations	Yes [✓ No]
Input grid specification :	54.5 57.5 7.0 13.0 0.1 0.2
Input grid altitude (m) :	0.0
Input name of predictionfile (Gravsoft format):	nm.h2
Should computed values be subtracted from observed	Yes [✓ No]
Data column number	0
Should statistics be output	Yes [✓ No]
Input histogram bin size	5.0
Output to file	[✓ Yes] No
Name of file to hold result:	nm.h2-tc_pred.dat
Running options	

Data send to geocol17

Remaining step is that TC is used to add back the effect of the residual topography.



TC - Compute terain effect on gravimetric quantities - ver. 0.3

Station list file:	nm.h2-tc_pred.dat
Detailed elevation grid file:	nmdtm
Coarse elevation grid file:	nmdtm5
Reference elevation grid file:	nmdtm30
Data type:	3
Type of effect:	4
Placement of station:	3
Type of operation:	2
Data column (operation 2 or 3):	1
Density:	2.67
Maximum window:	31.52 34.98 -107.98 -105.0
Minimum computation distance of inner grid:	10.0
Maximal radius of computation	99.0

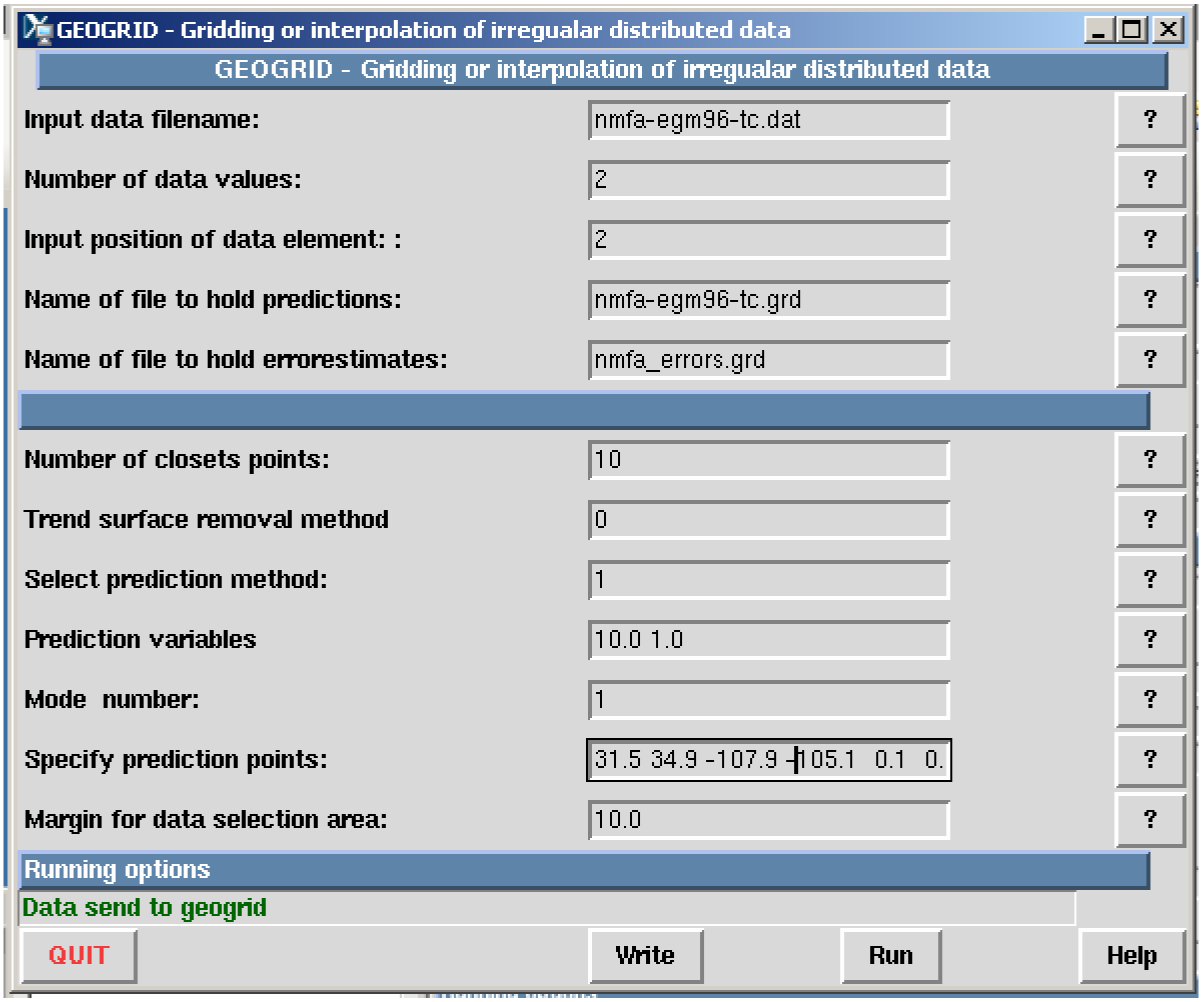
Running options

Name of file to hold ouput:	nm.h2_pred.dat

Data send to tc

We now prepare for 2D solutions using Stokes or Collocation.

First a gridding is needed using geogrid. The result is compared with the original data using geoip.


GEOGRID - Gridding or interpolation of iregualar distributed data
GEOGRID - Gridding or interpolation of irregualar distributed data
?
Running options
QUIT
Write
Run
Help

Input data filename: nmfa-egm96-tc.dat

Number of data values: 2

Input position of data element: : 2

Name of file to hold predictions: nmfa-egm96-tc.grd

Name of file to hold errorestimates: nmfa_errors.grd

Number of closets points: 10

Trend surface removal method 0

Select prediction method: 1

Prediction variables 10.0 1.0

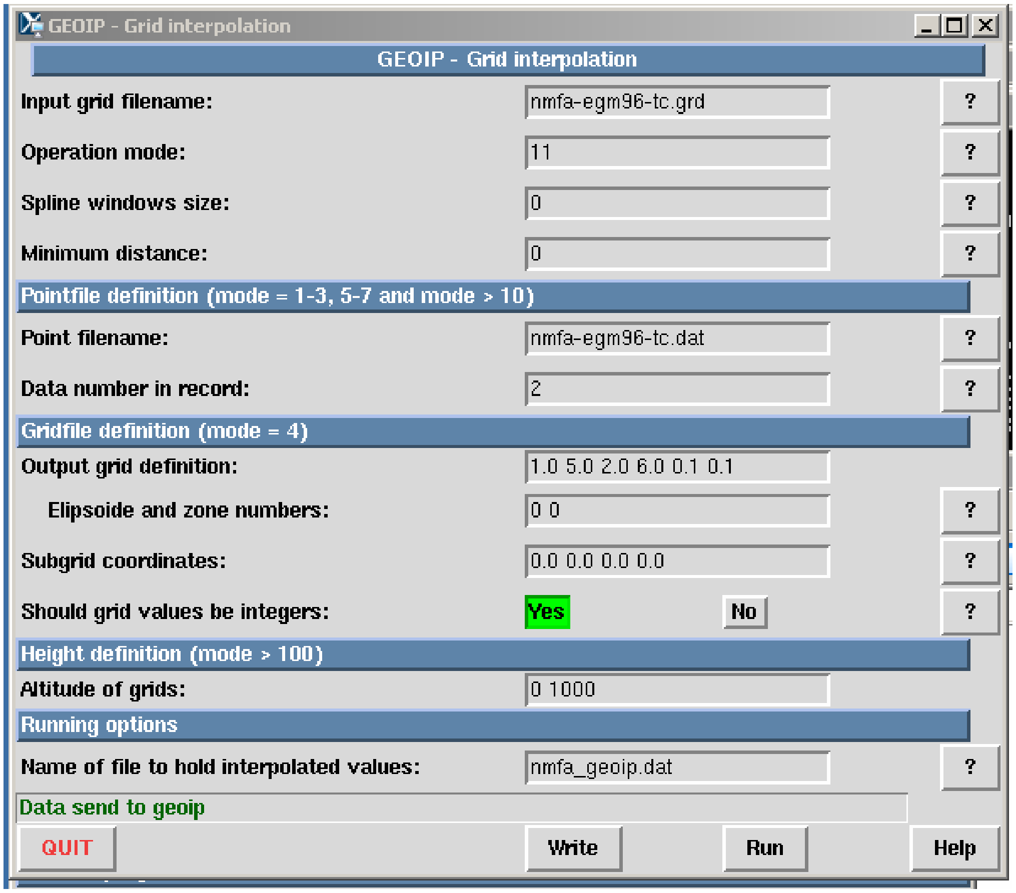
Mode number: 1

Specify prediction points: 31.5 34.9 -107.9 -105.1 0.1 0.

Margin for data selection area: 10.0

Data send to geogrid

```
*********************************************************************
 *    GEOGRID - GRAVSOFT data gridding - vers. JAN 06 - (c) RF/DNSC  *
 *********************************************************************
 internal lambert map projection center:    33.200 -106.500
 rkm =   10.0, data selection area:   31.4100  34.9900-108.0075-104.9925

 data values per point:  2, used no.:  2
 total points in file:   2920, selected:   2920
 min max mean stddev:    -41.031    45.899     0.307    13.173

 collocation prediction - sqrc0,xhalf(km),rmsn =   13.17  10.00   1.00
 selection: 10 closest points per quadrant

 data organization limits in lambert proj: -168534.  179793. -124480.  124377.
 subrectangles (n,e,total):    37   26  962
 size (km):      9.41     9.57, average pts per rect (rdat):    3.035
 max points in subrects:   11, percentage with no points:   7.5

 predicted:    1015 points
 prediction pts mean std.dev. min max:      0.55     12.35    -37.59    47.66
 prediction error values min max:                              0.94    13.14
```



GEOIP - Grid interpolation

Input grid filename: nmfa-egm96-tc.grd

Operation mode: 11

Spline windows size: 0

Minimum distance: 0

Pointfile definition (mode = 1-3, 5-7 and mode > 10)

Point filename: nmfa-egm96-tc.dat

Data number in record: 2

Gridfile definition (mode = 4)

Output grid definition: 1.0 5.0 2.0 6.0 0.1 0.1

Elipsoide and zone numbers: 0 0

Subgrid coordinates: 0.0 0.0 0.0 0.0

Should grid values be integers: [✓ Yes] No

Height definition (mode > 100)

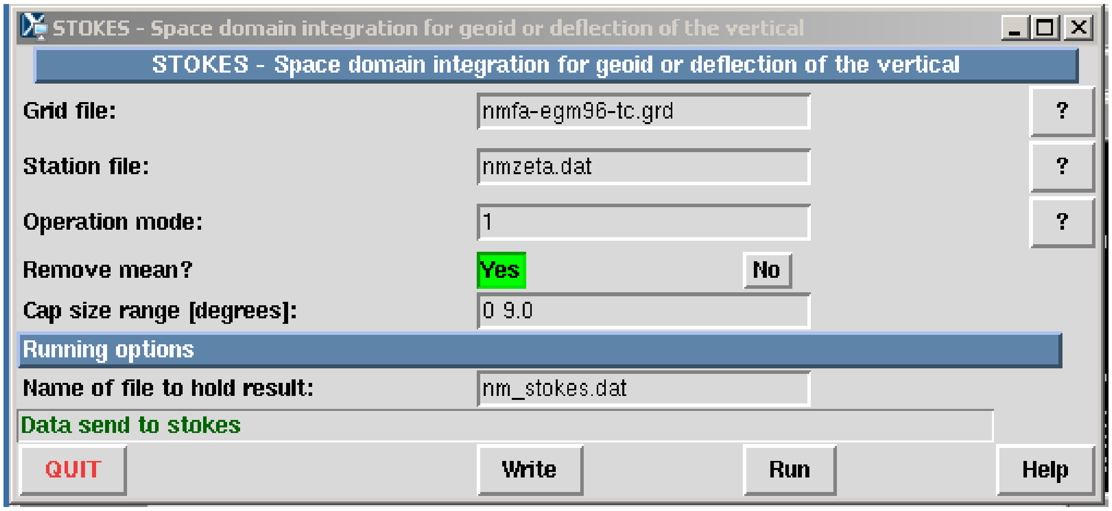
Altitude of grids: 0 1000

Running options

Name of file to hold interpolated values: nmfa_geoip.dat

Data send to geoip

```
---  G E O I P  ---
 grid file name: nmfa-egm96-tc.grd
 output file name: nmfa_geoip.dat
 mode = 1, nsp =  0, minimum edge dist    0.0 km
 point file name: nmfa-egm96-tc.dat
 - bilinear interpolation -
 - subtraction of interpolated values from pointfile -

 number of prediction points:    2920
 within area    31.6836   34.8134 -107.8166 -105.1850

 grid file information:
 gridlab:    31.5000  34.9000-107.9000-105.1000   0.1000   0.1000   35  29
 selected subgrid:   31.6000  34.9000-107.9000-105.1000
 points:  34 x  29 =     986, zero values:       0, unknown (9999):       0
 min  max  mean  std.dev.:    -37.59     47.66      0.57     12.53

 points predicted:   2920,  skipped points:      0
 minimum distance to grid edges for predictions:    7.8 km
 statistics:                     mean   std.dev.   min      max    unknown
 original data (pointfile) :    0.307   13.173  -41.031   45.899       0
 grid interpolation results:    0.291   11.606  -34.880   43.166       0
 predicted values output   :    0.016    3.699  -17.386   21.422       0
```


STOKES - Space domain integration for geoid or deflection of the vertical

STOKES - Space domain integration for geoid or deflection of the vertical

Grid file: nmfa-egm96-tc.grd

Station file: nmzeta.dat

Operation mode: 1

Remove mean? [✓ Yes] No

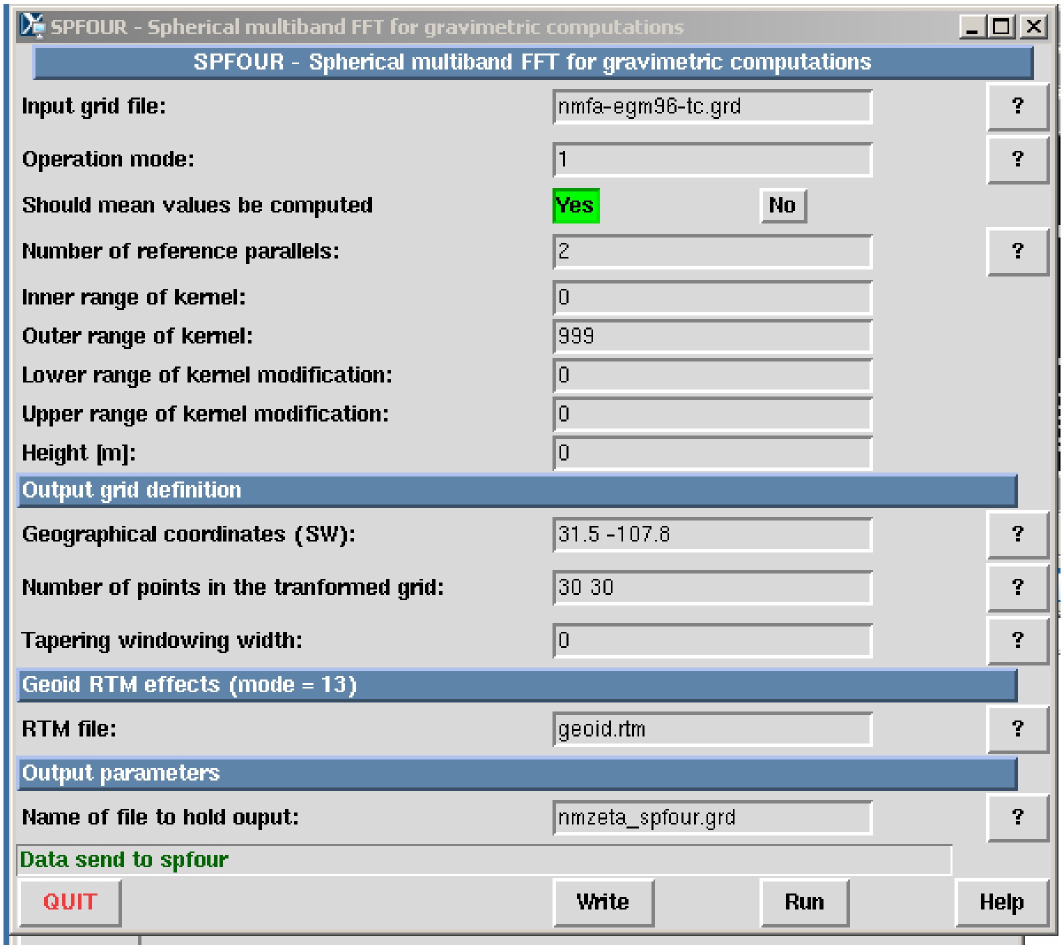
Cap size range [degrees]: 0 9.0

Running options

Name of file to hold result: nm_stokes.dat

Data send to stokes

Using the gridded data we use spfour.


SPFOUR - Spherical multiband FFT for gravimetric computations

SPFOUR - Spherical multiband FFT for gravimetric computations

Input grid file: nmfa-egm96-tc.grd

Operation mode: 1

Should mean values be computed [✓ Yes] No

Number of reference parallels: 2

Inner range of kernel: 0

Outer range of kernel: 999

Lower range of kernel modification: 0

Upper range of kernel modification: 0

Height [m]: 0

Output grid definition

Geographical coordinates (SW): 31.5 -107.8

Number of points in the tranformed grid: 30 30

Tapering windowing width: 0

Geoid RTM effects (mode = 13)

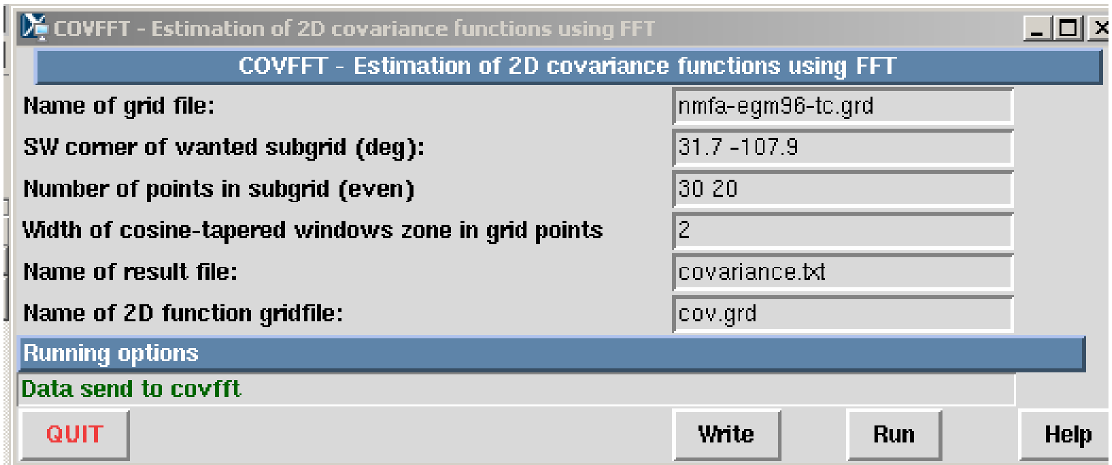
RTM file: geoid.rtm

Output parameters

Name of file to hold ouput: nmzeta_spfour.grd

Data send to spfour

Again, using the gridded data, we may compute the covariance function using covfft.


COVFFT - Estimation of 2D covariance functions using FFT
COVFFT - Estimation of 2D covariance functions using FFT
Running options
QUIT
Write
Run
Help

Name of grid file: nmfa-egm96-tc.grd

SW corner of wanted subgrid (deg): 31.7 -107.9

Number of points in subgrid (even) 30 20

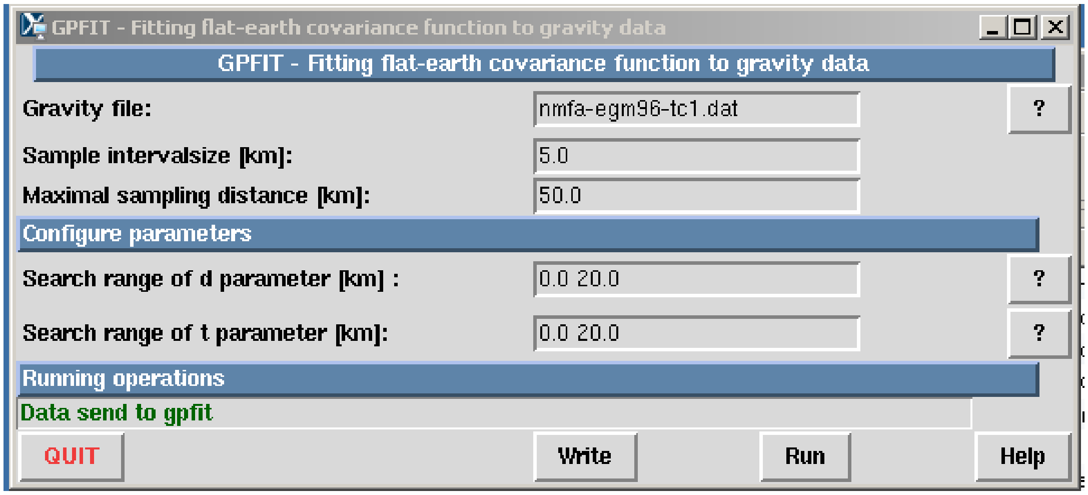
Width of cosine-tapered windows zone in grid points 2

Name of result file: covariance.txt

Name of 2D function gridfile: cov.grd

Data send to covfft

Or, we may directly from the point data estimate geoid heights using planar collocation with gpcol1.


GPFIT - Fitting flat-earth covariance function to gravity data
GPFIT - Fitting flat-earth covariance function to gravity data
?
Configure parameters
?
?
Running operations
QUIT
Write
Run
Help

Gravity file: nmfa-egm96-tc1.dat

Sample intervalsize [km]: 5.0

Maximal sampling distance [km]: 50.0

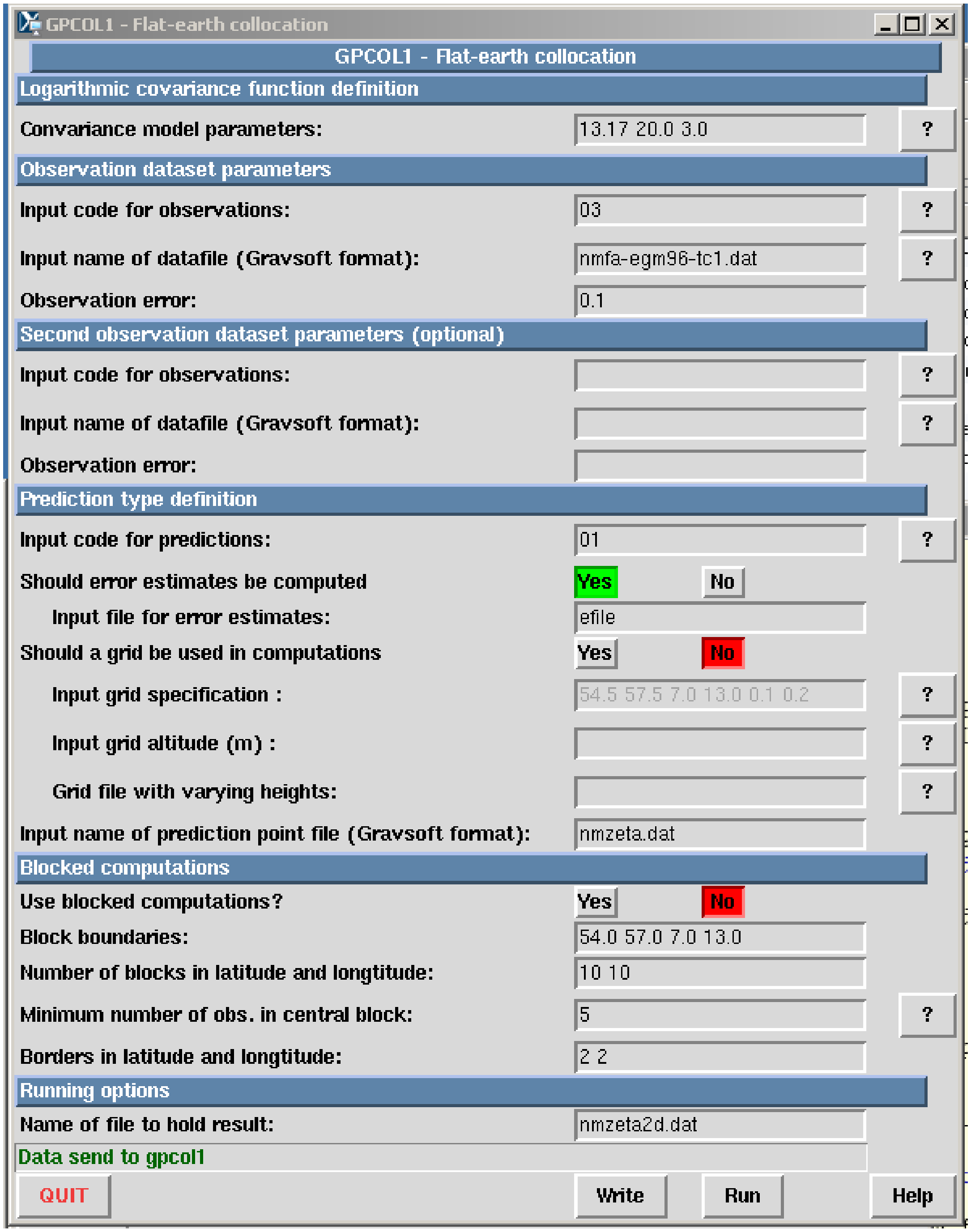
Search range of d parameter [km] : 0.0 20.0

Search range of t parameter [km]: 0.0 20.0

Data send to gpfit

```
--- G P F I T ---
 input: gravityfile
 input: ds,smax,d1,d2,t1,t2 (km)
 n = 2920, mean =   0.31, C0 =   173.57, sqrtC0 =  13.17
 - searching for best fit in interval:  0 -  20,  0 -  20 km
 Best values of fit: D =  20 km, T =   3 km

 Empirical and fitted covariances:
    Dist (km)     No of products      Cov     Fitted cov
       0.0            3435          174.34     173.57
       5.0            8926          136.32     152.84
      10.0           17078           98.54     106.77
      15.0           24469           66.30      62.12
      20.0           31357           41.09      31.24
      25.0           37212           21.27      13.40
      30.0           43213            7.50       4.14
      35.0           47965           -2.66      -0.31
      40.0           53321           -8.60      -2.24
```

45.0	57901	-13.33	-2.94
50.0	62148	-16.58	-3.04



GPCOL1 - Flat-earth collocation

Logarithmic covariance function definition

Convariance model parameters: 13.17 20.0 3.0

Observation dataset parameters

Input code for observations: 03

Input name of datafile (Gravsoft format): nmfa-egm96-tc1.dat

Observation error: 0.1

Second observation dataset parameters (optional)

Input code for observations:

Input name of datafile (Gravsoft format):

Observation error:

Prediction type definition

Input code for predictions: 01

Should error estimates be computed [✓ Yes] No

Input file for error estimates: efile

Should a grid be used in computations Yes [✓ No]

Input grid specification : 54.5 57.5 7.0 13.0 0.1 0.2

Input grid altitude (m) :

Grid file with varying heights:

Input name of prediction point file (Gravsoft format): nmzeta.dat

Blocked computations

Use blocked computations? Yes [✓ No]

Block boundaries: 54.0 57.0 7.0 13.0

Number of blocks in latitude and longtitude: 10 10

Minimum number of obs. in central block: 5

Borders in latitude and longtitude: 2 2

Running options

Name of file to hold result: nmzeta2d.dat

Data send to gpcol1