Programs for computing mass changes on the Earth's surface From GRACE/GRACE-FO Level 2 data

Jun-Yi Guo
Division of Geodetic Sciences
School of Earth Sciences
The Ohio State University

July 30, 2025

Preface

This is a suit of programs for computing mass changes on the Earth's surface using the GRACE/GRACE-FO Level 2 data. The user should first compile the programs to executables and then copy the executables to the data folder for computation. The usage is demonstrated by two examples, one is for the CSR GRACE data during 2014-2016 and the other is for the GRACE-FO data during 2018-2021.

Swarztrauber's Fortran FFTPACK translated to pure ISO C/C++ by Fernandes[1] is used to compute the discrete Fourier and Cosine transforms.

The readers can use the programs for their purpose with an acknowledgement and list in the reference the following book: Guo, J.Y. (2025). Physical Geodesy: Global Elastic Deformation of the Earth. Springer.

[1]https://people.math.sc.edu/Burkardt/c_src/fftpack4/fftpack4.html

Contents

Chapter 1

Preliminary approach

In this chapter, we present the most essential programs to compute Gaussian smoothed mass changes on the Earth's surface, assuming changes in the Stokes coefficients of the geopotential determined by GRACE are purely caused by mass changes on the Earth's surface.

1.1 Rename GRACE L2 data files

The GRACE/GRACE-FO L2 monthly GSM- data files include the data span information of the starting and end dates. We take the average of the starting and end dates as the date of the solution for each month. For example, the data span information of the CSR solution (from CSR L2 Release 6)

```
GSM-2_2013335-2013365_GRAC_UTCSR_BA01_0600
GSM-2_2013335-2013365_GRAC_UTCSR_BB01_0600
```

are

- Starting date: 335st day of 2013,
- End date: 365st day of 2013.

Other information in the file names include the maximum degree and order of Stokes coefficients (BA01 for 60 and BB01 for 96, which can be found in the files; BC01 for 180 for only AOD models) and the release number (0600 for 6 and 0603 for 6.03).

We rename the files for our convenience of usage. The program for renaming the L2 data files is

```
Rename_L2_Files.cpp
```

This is a single file program, and thus, the compilation and execution are straightforward.

It copies the files (the original files are kept) to the names in the format of `YYYY_DDD_XXX.txt`, where `YYYY` is the year, `DDD` is the median date of the data span, and `XXX` (could be a string of any length) is a user assigned suffix (see input parameters below).

The input/output information is given in the file `Rename_L2_Files.txt`. The following are the descriptions of the data in this file.

- The first line is a string. It is the command for copying files of the operating system. For example, windows: `copy`, Unix/Linux: `cp`.

- The second line has two integers. The first one is the location of the first digit of year in the original data file name (counted from zero). The second one is the number of digits of year.
- The third line has two integers. The first one is the location of the first digit of the first day in the original data file name (counted from zero). The second one is the number of digits for the first day.
- The fourth line has two integers. The first one is the location of the first digit of the last day in the original data file name (counted from zero). The second one is the number of digits for the last day. If the last day is in the next year, the program will adjust the year and the median date in the output file name.
- The fifth line is a string to be included as suffix in output file names, i.e., `_00`
- The sixth line is an integer. It is the number of data files.
- The following lines are the data files. One file name per line.

An example of the content of `Rename_L2_Files.txt` is

```
copy
6 4
10 3
18 3
_00
28
GSM-2_2013335-2013365_GRAC_UTCSR_BB01_0600
GSM-2_2014001-2014016_GRAC_UTCSR_BB01_0600
......
```

The location of year, first day and last day in all file names should be the same. For location of them, refer to the following:

```
 GSM-2_2013335-2013365_GRAC_UTCSR_BB01_0600
 GSM-2_2014001-2014016_GRAC_UTCSR_BB01_0600
       ----===     ===
       P:6 P:10    P:18
       N:4 N:3     N:3

 P: Position of first digit, N: Number of digits
```

1.2 Extract spherical harmonic coefficients (SHCs)

The L2 data files from different centers are arranged in different ways. However, we have written a single program to handle the data from all centers (CSR, GFZ, JPL). Here we extract only the GRACE solutions of C_{lm} and S_{lm}.

The program file is

```
Extract_SHCs.cpp
```

This is a single file program. However, it requires the `.h` files `rshc.h` and `check_error.h` for to be compiled.

The original data file must have the following format for GRACE results:

```
GRCOF2  l  m  c  s ...
```

where `GRCOF2` means that this line is GRACE solution, `l` and `m` are degree and order, respectively, and `c` and `s` are coefficients. Data after `s` in each line are discarded. Lines that do not begin with `GRCOF2` are discarded.

The input/output information is assigned in the file `Extract_SHCs.txt`, where

- The first line is the maximum degree (and order).
- The second line is the total number of data files.
- The following lines are input and output names of each data file.

The following is an example `Extract_SHCs.txt` file:

```
96
28
2013_350_00.txt 2013_350_01.txt
2014_008_00.txt 2014_008_01.txt
......
```

The output format is

```
l  m  c  s
```

in each line. Data in output files are in the order `m = 0, 1, 2, ...`, and then `l = m, m+1, ....`

If the coefficients of degrees 0 and 1 are not included in the original data files, we set $C_{0,0} = 1$ and $C_{1,0} = C_{1,1} = S_{1,1} = 0$.

1.3 Change of the scale length

The semi-major axis of a reference Earth ellipsoid a is used as the length scale in the spherical harmonic coefficients of the GRACE L2 data. The value of a can be found in the GRACE L2 data files. The coefficients are multiplied with $(a/r)^l$, where l is the degree. This program changes the scale length from a to R. This is to multiply the coefficients by $(a/R)^l$.

The program file is

```
Rescale_SHCs.cpp
```

This is a single file program. However, it requires the `.h` files `rshc.h` and `vec.h` for to be compiled.

The input/output information is specified in the file `Rescale_SHCs.txt`, where

- The first line is the maximum degree (and order).
- The second line is the value of a in the L2 data.

- The third line is the value of the new scale length R.
- The fourth line is the number of data files
- The following lines are input and output names of each data file.

The following is an example `Rescale_SHCs.txt` file:

```
96
6378136.3
6371000.0
28
2013_350_01.txt   2013_350_02.txt
2014_008_01.txt   2014_008_02.txt
......
```

1.4 Interpolation of missing months

There are some months of missing GRACE solution. For certain applications it is preferable to fill these gaps by interpolation. Here we use a quadratic polynomial to fit the solutions of several months—preferably 2 months before and 2 months after—to interpolate the data for the missing month or months.

The program file is

```
Interpolate_SHCs.cpp
```

This program requires the `.cpp` file `linalg.cpp` and the `.h` files `linalg.h`, `rshc.h`, and `vec.h` for to be compiled.

The input/output information is specified in the file `Interpolate_SHCs.txt`, where

- The first line is the maximum degree and order of the GRACE solution.
- The second line is the number of known months to be used for interpolation.
- The following 'number of known months' lines are file names of the GRACE L2 monthly solution to be used for interpolation.
- The next line is the number of unknown months to be interpolated.
- The following 'number of unknown months' lines are file names of the GRACE L2 monthly solution to be interpolated.

Our file name convention including year and median day is used for time. The following is an example `Interpolate_SHCs.txt` file:

```
96
5
2015_075_02.txt
2015_105_02.txt
2015_116_02.txt
2015_196_02.txt
```

```
2015_228_02.txt
2
2015_135_i2.txt
2015_165_i2.txt
......
```

This is to use 5 solutions to interpolate 2 solutions. `2015_116_02.txt` is to be replaced by `2015_135_i2.txt` when computing the annual average, as `2015_116_02.txt` is too close to `2015_105_02.txt`. When computing the annual average, we expect that the 12 solutions are homogeneously distributed through the year.

1.5 Truncation of SHCs up to a suitable degree and order

Higher degree and order coefficients include larger errors, and may be preferred to be omitted in certain applications. So there is a need to truncate the SHCs up to a suitable degree and order. This program will also be used to truncate the glacial isostatic adjustment (GIA) geopotential model to the GRACE maximum degree and order.

The program file is `Truncate_SHCs.cpp`. It needs the `.h` file `rshc.h` for to be compiled.

The input/output information is specified in the file `Truncate_SHCs.txt`, where

- The first line is the maximum degree of the input SHCs.
- The second line is the maximum degree of the output SHCs.
- The third line is the number of files of the SHCs.
- The following are the input and output files.

The following is an example `Truncate_SHCs.txt` file:

```
96
63
37
2014_008_02.txt   2014_008_63_02.txt
2014_042_i2.txt   2014_042_63_i2.txt
......
```

This is to truncate the original data of degree and order up to 96 to 63. If the maximum degree of the output SHCs is larger than that of the input SHCs, 0s are appended. For example, if the maximum degree of the output SHCs is assigned to be 127, the coefficients beyond degree and order 96 are set to 0. We truncate to 63 and 95. The reason will be seen in Sect. 1.8.1 and the rest of this chapter.

1.6 Averaging the GRACE L2 SHCs

GRACE L2 data are normally used for studying mass changes. So a reference field is normally subtracted from all the monthly solutions. In most cases, the reference field is the average in a certain time span, or simply a monthly solution.

Here we write a program to average the SHCs of a series of monthly L2 data with the flexibility to compute multiple averages.

The program is in the file `Average_SHCs.cpp`. This is a single file program. It requires the `.h` files `rshc.h` and `check_error.h` for to be compiled.

The input/output information is specified in the file `Average_SHCs.txt`, where

- The first line is a suffix `XXX` for the output file. The output file name follows our `YYYY` and `DDD` convention, where `YYYY` and `DDD` are computed by averaging the median time of all the input files specified bellow.
- The second line is the maximum degree of the SHCs.
- The third line is the number of averages to be computed.
- The third line is the number of files to be averaged in each average.
- The following lines are input data file names in our `YYYY` and `DDD` convention sequentially for each average.

The following is an example `Truncate_SHCs.txt` file:

```
_63_REFERENCE
63
1
36
2014_008_63_02.txt
2014_042_63_i2.txt
......
```

The output file is `2015_183_63_REFERENCE.txt`, where the `2015` and `183` are computed.

This program can be used more generally, e.g., computing the yearly averages of multiple years. This requires to change 1 to 3 and 36 to 12. In this case, the 36 sets of SHCs are averaged 12 by 12.

1.7 Subtraction of SHCs of a reference model

The program is in the file `Subtract_Reference_SHCs.cpp`. This is a single file program. It requires the `.h` files `rshc.h` and `check_error.h` for to be compiled.

The input/output information is specified in the file `Subtract_Reference_SHCs.txt`, where

- The first line is the maximum degree of the SHCs.
- The second line is file name of the reference model to be removed.
- The third line is the number of SHC files to be handled.
- The following lines are input and output data file names.

The following is an example `Subtract_Reference_SHCs.txt` file:

```
63
2015_183_63_REFERENCE.txt
37
2014_008_63_02.txt   2014_008_63_03.txt
2014_042_63_i2.txt   2014_042_63_i3.txt
......
```

All file names should be in our YYYY and DDD convention. The reference model removed should be better the average of a whole number of years, so that the result is not affected by a particular season.

We refer to the monthly SHCs after removing the reference model as anomalous SHCs of the gravitational potential, which are to be used to represent the changes of the gravitational field.

1.8 Transformations between SHCs and geoid

1.8.1 From SHCs to geoid

The program for computing the geoid height over a grid using the spherical harmonic coefficients is in the file SHCs2Geoid.cpp. This program requires the .h rsht.h, fftpack++.h, and fftpack.h and the .cpp file fftpack.c for to be compiled.

The results are given over the grid

$$B_j = 90 - (j + 1/2) * 180/(2 * bw) \qquad \text{for} \qquad j = 0, ..., j = 2 * bw - 1$$
$$\lambda_k = k * 360/(2 * bw) \qquad \text{for} \qquad k = 0, ..., k = 2 * bw - 1$$

where bw is called bandwidth. The corresponding spherical harmonic series is up to degree and order $bw-1$.

The input/output information are assigned in the file SHCs2Geoid.txt, where

- The first line is the radius of the Earth.
- The second line is the bandwidth bw.
- The third line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example SHCs2Geoid.txt file:

```
6.371E+6
64
37
2014_008_63_03.txt   2014_008_63_04_Geoid.txt
2014_042_63_i3.txt   2014_042_63_i4_Geoid.txt
......
```

Remark: Remark: If the SHCs are those of the disturbing potential, the output is the geoidal height. If the SHCs are those of the change of the gravitational potential, the output is the change of the geoidal height.

1.8.2 From geoid to SHCs

The program for computing the spherical harmonic coefficients from the geoidal height over a grid as defined in last subsection is in the file `Geoid2SHCs.cpp`. This program requires the `.h` `rsht.h`, `fftpack++.h`, and `fftpack.h` and the `.cpp` file `fftpack.c` for to be compiled.

The input/output information are assigned in the file `SHCs2Geoid.txt`, where

- The first line is the radius of the Earth.
- The second line is the bandwidth bw.
- The third line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example `Geoid2SHCs.txt` file:

```
6.371E+6
64
37
2014_008_63_04_Geoid.txt     2014_008_63_04_Geoid_SHCs.txt
2014_042_63_i4_Geoid.txt     2014_042_63_i4_Geoid_SHCs.txt
......
```

The geoidal height must be provided over the same grid as in Sect. 1.8.1.

Remark: If the input is the geoidal height, the output is the SHCs of the disturbing potential; if the input is the change of the geoidal height, the output is the change of SHCs of the gravitational potential.

1.9 Transformation between SHCs and mass changes

1.9.1 From SHCs to mass

The mass changes are computed from the changes of SHCs based on loading theory. The same as in Sect. 1.8.1, mass data are given over the grid

$$\begin{array}{lll} B = 90 - (j + 1/2) * 180/(2 * bw) & \text{for} & j = 0, ..., j = 2 * bw - 1 \\ \lambda = k * 360/(2 * bw) & \text{for} & k = 0, ..., k = 2 * bw - 1 \end{array}$$

where bw is called bandwidth. This grid corresponds to a spherical harmonic series up to degree and order $bw - 1$.

The program is in the file `SHCs2Mass.cpp`. This program requires the `.h` `rsht.h`, `fftpack++.h`, and `fftpack.h` and the `.cpp` file `fftpack.c` for to be compiled.

The input/output information is assigned in the file `SHCs2Mass.txt`, where

- The first line is the radius of the Earth.
- The second line is the average density of the Earth.
- The third line is the bandwidth bw.

- The fourth line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example `SHCs2Mass.txt` file:

```
6.371E+6
5.5151E+3
64
37
2014_008_63_03.txt   2014_008_63_04_Mass.txt
2014_042_63_i3.txt   2014_042_63_i4_Mass.txt
......
```

Remark: The change of SHCs is assumed to represent the change of the gravitation potential caused by the change of a thin layer of mass over the Earth's surface assumed to be spherical. The elastic loading effect is included in the change of SHCs. The results of the density within the thin layer are referred to as mass change or anomalous mass.

1.9.2 From mass to SHCs

The program for computing the change of SHCs from mass changes based loading theory is in the file `mass2SHCs.cpp`. This program requires the `.h` `rsht.h`, `fftpack++.h`, and `fftpack.h` and the `.cpp` file `fftpack.c` for to be compiled.

It uses the same grid as `SHCs2Mass.cpp`. The input/output information is assigned in the file `Mass2SHCs.txt`, where

- The first line is the radius of the Earth.
- The second line is the average density of the Earth.
- The third line is the bandwidth bw.
- The fourth line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example `SHCs2Mass.txt` file:

```
6.371E+6
5.5151E+3
64
37
2014_008_63_04_Mass.txt    2014_008_63_04_Mass_SHCs.txt
2014_042_63_i4_Mass.txt    2014_042_63_i4_Mass_SHCs.txt
......
```

Remark: The change of SHCs is assumed to represent the change of the gravitation potential caused by the change of a thin layer of mass over the Earth's surface assumed to be spherical. The elastic loading effect is included in the change of SHCs.

1.10 Transformations between geoid and mass changes

1.10.1 From geoid to mass

The mass change is computed from the change of the geoidal height based loading theory. Grided data are used. The same as in Section 1.8.1, data must be given over the grid

$$
\begin{aligned}
&B = 90 - (j + 1/2) * 180/(2 * bw) \qquad \text{for} \qquad j = 0, 1, \cdots, 2 * bw - 1\,,\\
&\lambda = k * 360/(2 * bw) \qquad \text{for} \qquad k = 0, 1, \cdots, 2 * bw - 1\,,
\end{aligned}
$$

where bw is called bandwidth. When transformed to spherical harmonic series, the degree and order are up to $bw - 1$.

The program is in the file `Geoid2Mass.cpp`. This program requires the `.h` `rsht.h`, `fftpack++.h`, and `fftpack.h` and the `.cpp` file `fftpack.c` for to be compiled.

The input/output information is assigned in the file `Geoid2Mass.txt`, where

- The first line is the average density of the Earth.
- The second line is the bandwidth bw.
- The third line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example `Geoid2Mass.txt` file:

```
5.5151e+3
64
37
2014_008_63_04_Geoid.txt   2014_008_63_04_Geoid_Mass.txt
2014_042_63_i4_Geoid.txt   2014_042_63_i4_Geoid_Mass.txt
......
```

Remark: The change of the geoidal height is assumed to be caused by the change of mass in a thin layer over the Earth's surface, including the elastic loading effect. The Earth's surface is assumed to be spherical.

1.10.2 From mass to geoid

The program for computing geoid changes from mass changes using loading theory is in the file `Mass2Geoid.cpp`. This program requires the `.h` `rsht.h`, `fftpack++.h`, and `fftpack.h` and the `.cpp` file `fftpack.c` for to be compiled.

It uses the same grid as `Geoid2Mass.cpp`. The input/output information is assigned in the file `Mass2Geoid.txt`, where

- The first line is the average density of the Earth.
- The second line is the bandwidth bw.
- The third line is the number of data file to be transformed.
- The following are input and output data files.

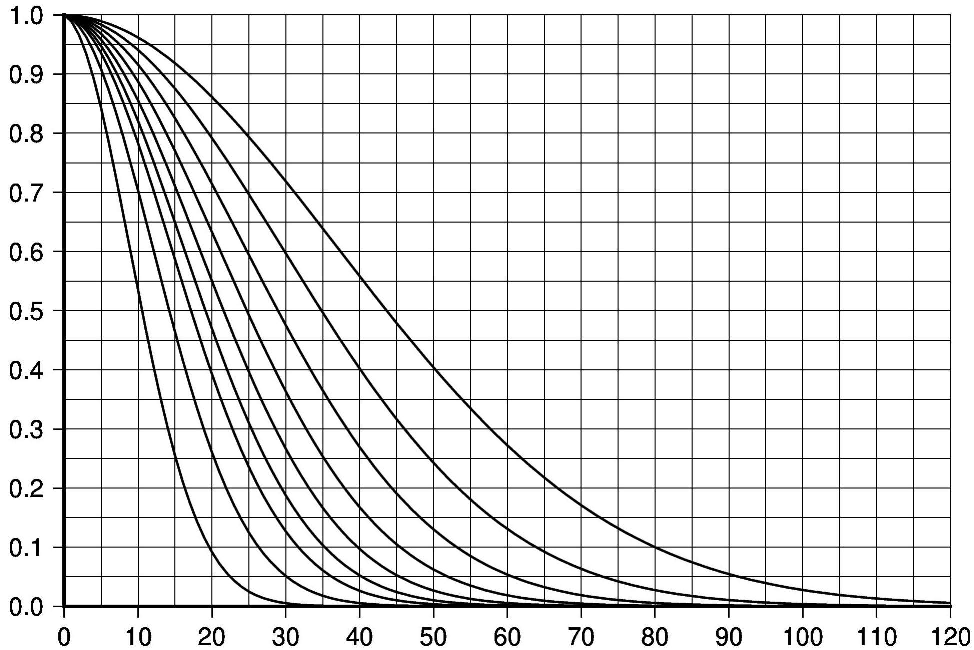

Figure 1.1: The isotropic Gaussian smoothing weight W_l as a function of degree. Smoothing radius from top to bottom: 200, 250, 300, 350, 400, 450, 500, 600, and 800 km.

The following is an example `Mass2Geoid.txt` file:

```
5.5151e+3
64
37
2014_008_63_04_Mass.txt   2014_008_63_04_Mass_Geoid.txt
2014_042_63_i4_Mass.txt   2014_042_63_i4_Mass_Geoid.txt
......
```

Remark: The change of the geoidal height is assumed to be caused by the change of mass in a thin layer over the Earth's surface, including the elastic loading effect. The Earth's surface is assumed to be spherical.

1.11 Isotropic Gaussian smoothing

Due to systematic errors in the L2 data, the geoid and mass changes computed have very large errors, mostly in the form of south-north stripes. Customarily, these errors are reduced by performing an isotropic Gaussian smoothing in the spatial domain.

Even though the smoothing is done in the spatial domain, its computation in the spectral domain, i.e., using the SHCs, is much simpler. This consists of multiplying the l-th degree SHCs by a degree-dependent-only weight function W_l depending on the smoothing radius. Some examples of W_l are shown in Fig. 1.1.

We write the programs to apply the isotropic Gaussian smoothing in two ways. One way is to use the SHCs, which consists of multiplying W_l to the l-th degree SHCs. The other way is to use the grid values, which consists to first transform the grid values to SHCs, multiply W_l to the l-th degree SHCs, and then transform the SHCs to grid values.

The smoothing eliminates the effect of higher degree SHCs. Customarily, the smoothing radius is set to about $R * (\pi/l_{\max})$, e.g., 334 km for $l_{\max} = 60$ and 208km for $l_{\max} = 96$. After the smoothing, the effect of the SHCs beyond the degree 60 or 96 in the result is practically negligible. This implies that the smoothed result would be indifferent from that obtained using SHCs up to whatever higher degree and order. We will use 300 for 60 ($W_{60} \sim 0.05$) and 63, and 200 for 95 ($W_{95} \sim 0.05$). Strictly speaking, if no other information is available, only the smoothed result is meaningful, since the magnitude of the effect of the $l > l_{\max}$ SHCs is unknown if not smoothed to a negligible level.

1.11.1 Isotropic Gaussian smoothing using SHCs

The program is in the file `Isotropic_Gaussian_SHCs.cpp`. This is a single `.cpp` file program which requires the `.h` files `rshc.h` and `vec.h` for to be compiled.

The input/output information is assigned in the file `Isotropic_Gaussian_SHCs.txt`, where

- The first line is the radius of the earth.
- The second line is the smoothing radius.
- The third line is the maximum degree of the SHCs,
- The fourth line is the number of data file to be transformed.
- The following are input and output data files.

The following is an example `Isotropic_Gaussian_SHCs.txt` file:

```
6371E+3
300E+3
63
37
2014_008_63_03.txt   2014_008_63_03_300s.txt
2014_042_63_i3.txt   2014_042_63_i3_300s.txt
......
```

Note that the input and output are the original SHCs and that with the smoothing applied, respectively.

1.11.2 Isotropic Gaussian smoothing using grid values

The program is in the file `Isotropic_Gaussian_Grid.cpp`. It requires the `.cpp` file `fftpack.c` and the `.h` files `rshc.h`, `rsht.h`, `vec.h`, `mat.h`, `fftpack.h`, and `fftpack++.h` for to be compiled.

The input/output information is assigned in the file `Isotropic_Gaussian_Grid.txt`, where

- The first line is the radius of the earth.
- The second line is the smoothing radius.
- The third line is the bandwidth bw defining the grid (see section 1.10.1).
- The fourth line is the number of data file to be transformed.

- The following are input and output data files.

The following is an example Isotropic_Gaussian_Grid.txt file:

```
6371E3
300E3
64
37
2014_008_63_04_Mass.txt   2014_008_63_04_Mass_300s.txt
2014_042_63_i4_Mass.txt   2014_042_63_i4_Mass_300s.txt
......
```

The grid must be the same as defined in section 1.10.1.

1.12 Summary and graphical examination of the result

We now obtained the monthly mass anomaly with respect to a reference model (computations are done in the folders of the data files GRACE-2014-2016 and GRACE-FO-2019-2021). Taking for example the case of GRACE-2014-2016, the procedure is as follows:

1. Rename files, generate files ending with 00.txt.
2. Extract SHCs, generate files ending with 01.txt.
3. Rescale the SHCs (from a to R), generate files ending with 02.txt.
4. Interpolate missing/to be replaced months, generate files ending with i2.txt.
5. Truncate SHCs from 96 to both 63 and 95, generate files ending with 63_x2.txt and 63_x2.txt (x=0 or i).
6. Compute averaged reference models (for both 63 and 95 SHCs) to be removed from all monthly models (models of a whole number of years with homogeneous time interval).
7. Remove the 63 and 95 reference models from all the monthly models 63_x2.txt and 63_x2.txt, respectively, generate files ending with x3.txt (x=0 or i).
8. Smoothing using the SHCs (300 km radius for 63, and 200 km for 95), generate files ending with 03_300s.txt and 03_200s.txt.
9. Compute mass change using the smoothed SHCs, generate files ending with 04_300s_Mass.txt and 04_200s_Mass.txt. These are the results!
10. Plot the monthly anomalous mass ending with 04_300s_Mass.txt and 04_200s_Mass.txt.

In the data folder, we have also run Mass2SHCs.exe, Geoid2Mass.exe, SHCs2Mass.exe, Mass2Geoid.exe, and Isotropic_Gaussian_Grid.exe for confirming the correctness of these programs by intercomparing the results. The last step, i.e., plotting the results, is to be detailed below.

At this stage, it is necessary to check the smoothed results of the monthly anomalous mass graphically. There may be months when the solution is corrupted by extremely heavy stripes. For these months, one may

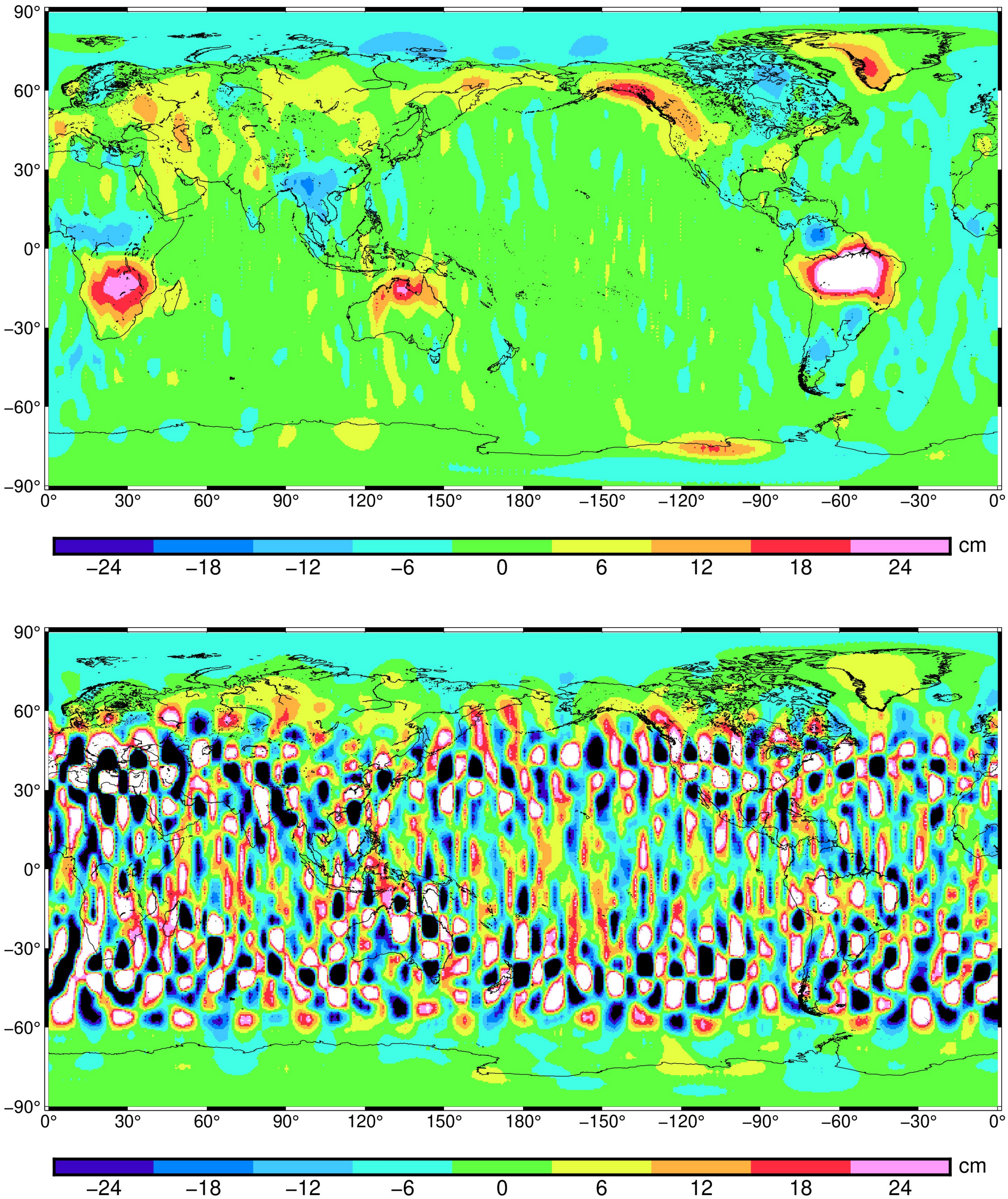

Figure 1.2: The smoothed mass anomaly in the month centered on the 76th day in 2014 (top) and that in the month centered on the 45th day in 2015 (bottom). Evidently, the latter is too bad to be used in further studies.

want to replace the solutions by interpolations using the solutions of the neighbouring months, and then redo the computation starting by averaging for the reference model.

We use Generic Mapping Tools (GMT) by running the program `00_plot_xxx.cpp`, where `xxx` is the smoothing radius. It should be compiled and run under Cygwin/Windows or Linux/Unix with GMT6 installed. The program generates a GMT `.sh` file and then run it for each month. The program is in the folders of the example data.

The input/output information is assigned in the file `00_plot_xxx.txt`, where

- The first line is the number of figures to be drawn.
- The following are input data files and output figure `.ps` files.

The following is an example `00_plot_xxx.txt` file:

```
37
2014_008_63_04_300s_Mass.txt  300\2014_008_63_04_300s_Mass.ps
2014_042_63_i4_300s_Mass.txt  300\2014_042_63_i4_300s_Mass.ps
......
```

This example is for 300km smoothing. Note that the smoothing radius in terms of angle with respect to the origin is 300/6371. The figure is written in the subfolder 300. We used Cygwin/Windows. Under Linux/Unix, the '\' should be replaced by '/'.

Shown in 1.2 are two figures of the smoothed monthly mass anomaly (the difference with respect to the reference model subtracted).

Chapter 2

Refinements

The procedure presented in the last chapter could be improved in various aspects, which are topics of this chapter.

2.1 Decorrelation filtering

Removing correlated errors in the data of SHCs before smoothing may help to get better results, i.e., with reduced stripes in the result. Here we follow the procedure of Duan et al. (2009).[1]

The portion kept unchanged, i.e., the SHCs not to be affected by the decorrelation, is defined by the formula

$$l = l_0 + \beta m^r, \tag{2.1}$$

where l and m are the degree and order od the SHCs. The window width, i.e., the number of SHCs to be fitted by a quadratic polynomial, is defined by

$$w = \max\Big\{Ae^{-\frac{[(1-\gamma)m^p+\gamma l^p]^{1/p}}{K}} + 1, 5\Big\}, \tag{2.2}$$

The parameters in both formulae are empirically chosen based on the pattern of the standard deviation of the monthly SHCs. The detail is referred to Duan et al. (2009).

2.1.1 Extract the standard deviations of the SHCs

As the decorrelation parameters are empirically chosen based on the pattern of the standard deviation, we need to first extract the standard deviation from the monthly data file of the SHCs. Here we chose to use the standard deviation of C_{lm} of three months with the average level of stripes as to be shown in Fig. 2.1.

The program is `Extract_Error_C_lm.cpp`, which is a single `.cpp` file program located in the folder `Decorrelation_Parameters`. It extracts the standard deviation of C_{lm}. It needs the `.h` files `rshc.h` and `check_error.h` for to be computed.

The input/output information is assigned in the file `Extract_Error_C_lm.txt`, where

- The first line is the maximum degree (or order) of the original model.

[1] *Duan, X.J., Guo, J.Y., Shum, C.K. and van der Wal, W., 2009. On the postprocessing removal of correlated errors in GRACE temporal gravity field solutions, J. Geodesy, 83, 1095-1106, DOI 10.1007/s00190-009-0327-0*

- The second line is the maximum degree (or order) to be extracted.
- The third line is the number of months to be computed.
- The following lines are the input and output file names.

The following is an example `Extract_Error_C_lm.txt` file:

```
96
63
3
2014_008_00.txt  2014_008_00_STD_Clm.txt
2014_105_00.txt  2014_105_00_STD_Clm.txt
2014_228_00.txt  2014_228_00_STD_Clm.txt
```

Note that the input files are those just after renaming.

2.1.2 Computation of the portion kept unchanged

Computation

The program is in the file `Decorrelation_Preserve.cpp` located in the folder `Decorrelation_Parameters`. This is a single file program. The parameters l_0, β, and r are chosen based on the pattern of the standard deviation of the monthly SHCs as shown in Fig. 2.1.

The input is in the file `Decorrelation_Preserve.txt`.

There are 3 data as input: the value of r, the value of l_0, and the value of the maximum order of the preserved coefficients (order or degree of the upper point of the curve of unchanged portion in Fig. 2.1). The value of r is chosen so that the curve approximately follows the isoline of the pattern of the standard deviation. This is done by doing the computation with various values of r, and then choose the one that agrees reasonable well with our criteria. The output file names are generated using the values of r, l_0, and maximum order of the preserved coefficients.

There are two output files for each set of parameters. The one without `_figure` in the name is for using in the decorrelation computation. The one with `_figure` in the name is for plotting the figure, which is provides for 0.1 increments of m computed using (2.1).

The output file name is self explanative. In the result, the first value is the maximum order of coefficients preserved unchanged, and the following are degree-order pairs of the curve defining, below it in Fig. 2.1, the preserved portion.

Plotting

The program for plotting the preserved portion in contrast of the pattern of the standard deviation is the single file program `Decorrelation_Preserve_Plot.cpp` located in the folder `Decorrelation_Parameters`. It should be compiled and run under the Windows/Cygwin or Linux/Unix operation system.

The input/output information is in the file `Decorrelation_Preserve_Plot.txt`. An example of it used to draw the Fig. 2.1 is

```
63
3
2014_008_00_STD_Clm.txt   2014_008_00_STD_Clm_Preserve.ps
```

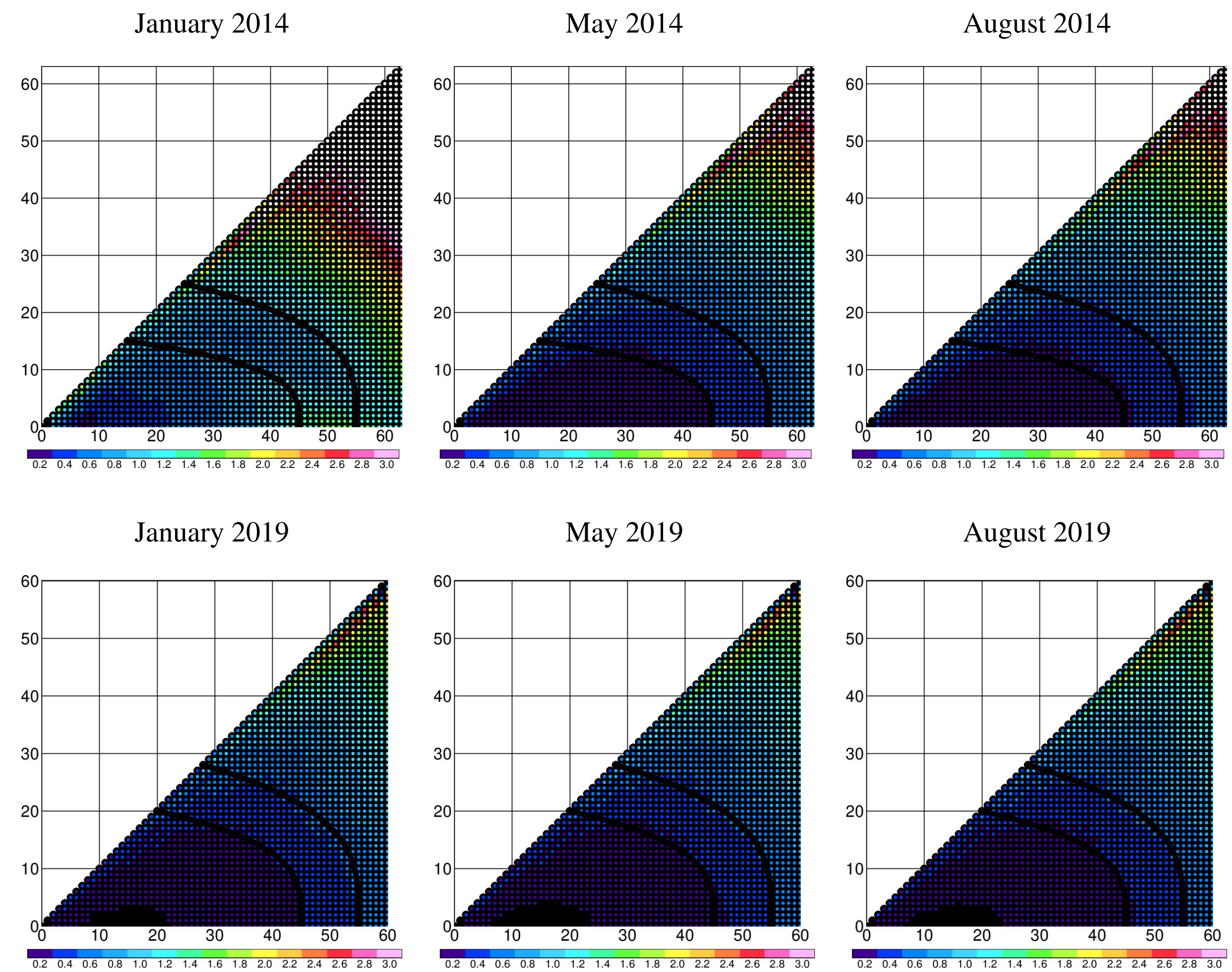

Figure 2.1: Error pattern scaled by 10^{12} with the portion kept unchanged in the decorrelation. Empirical parameter: $r = 3.5$. Predefined parameters: End point (l, m) pair of unchanged portion. GRACE, i.e., top panel: $(45, 0)$ and $(15, 15)$; $(55, 0)$ and $(25, 25)$. GRACE-FO, i.e., bottom panel: $(45, 0)$ and $(20, 20)$; $(55, 0)$ and $(28, 28)$

```
2014_105_00_STD_Clm.txt   2014_105_00_STD_Clm_Preserve.ps
2014_228_00_STD_Clm.txt   2014_228_00_STD_Clm_Preserve.ps
2
preserved_r3.5_l045_lm15_figure.txt
preserved_r3.5_l055_lm25_figure.txt
```

where

- The first line, 63, is the maximum degree and order of the SHCs.
- The second line, 3, is the number of figures to be drawn.
- The following 3 lines: input file of the standard deviation of the SHCs and the output figure file for each figure.
- The line that follows, 2, is the number of curves of the preserved portion to be put on the figures.
- The 2 lines that follow are the file names of the curves for the preserved portion. These are the files with `_figure` in the name.

2.1.3 Computation of window width

Computation

The program is in the file `Decorrelation_Window.cpp` located in the folder `Decorrelation_Window`. This is a single file program.

The input information is in the file `Decorrelation_Window.txt`. An example is

```
63
2
20 2.1 0.17 30
20 2.1 0.17 40
```

where

- The first line, 63, is the maximum degree and order of the SHCs.
- The second line, 2, is the number of cases to be computed.
- Each of the following 3 lines are the values of A, p, γ, and K.

There are two output files for each set of parameters. The one without `_figure` in the name is for using in the decorrelation computation. The one with `_figure` in the name is for plotting the figure.

The output file name is self-explanative. In the result, the first line is the maximum degree of the L2 data to be decorrelated (not output to the file for figure), and each of the following lines includes degree, order and window width.

Plotting

The program for plotting the preserved portion in contrast of the pattern of the standard deviation is the single file program `Decorrelation_Window_Plot.cpp` located in the folder `Decorrelation_Parameters`. It should be compiled and run under the Windows/Cygwin or Linux/Unix operation system.

The input/output information is in the file `Decorrelation_Window_Plot.txt`. An example of it used to draw the Fig. 2.2 is

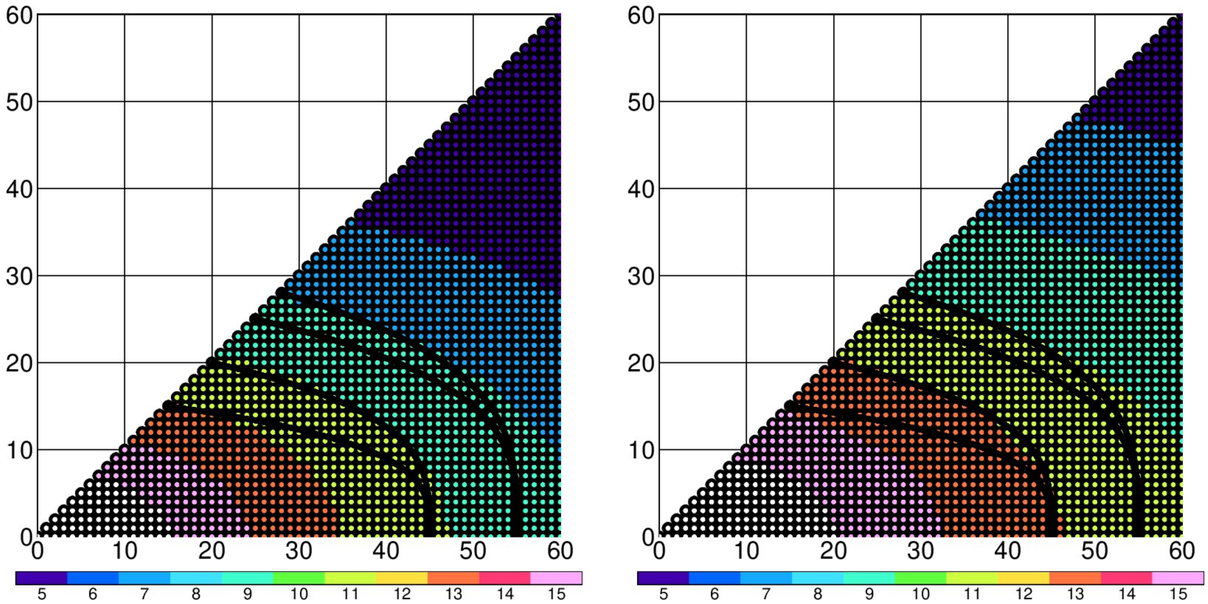

Figure 2.2: Window width. Empirical parameters: $A = 20$, $p = 2.1$, $\gamma = 0.18$, $K = 30$ (left) and $k = 40$ (right).

```
60
4
preserved_r3.5_l045_lm15_figure.txt
preserved_r3.5_l055_lm25_figure.txt
preserved_r3.5_l045_lm20_figure.txt
preserved_r3.5_l055_lm28_figure.txt
2
window_A20_p2.1_gamma0.18_K30_Nmax60_figure.txt  A20p2.1gamma0.18K30N60.ps
window_A20_p2.1_gamma0.18_K40_Nmax60_figure.txt  A20p2.1gamma0.18K40N60.ps
```

where

- The first line, 60, is the maximum degree and order of the SHCs.
- The second line, 4, is the number of curves of the preserved portion to be put on the figures.
- The 4 lines that follow are the file names of the curves for the preserved portion. These are the files with `_figure` in the name.
- The line that follows, 2, is the number of figures to be drawn.
- The following 2 lines: input file of the window size and the output figure file for each figure.

We have put the lines defining the reserved portion of both the GRACE and GRACE-FO (two for each, see Fig. 2.1) on the figures. The window width is then understood as a compromise for both the GRACE and GRACE-FO data.

2.1.4 Decorrelation filtering

The program is in the file `Decorrelation_Filter.cpp`. It requires the `.cpp` files `linalg.cpp` and `polyfit_sub.cpp` and the `.h` files `linalg.h`, `polyfit_sub.h`, `vec.h`, `mat.h`, and `rshc.h` for to be compiled.

The input/output information is assigned in the file `Decorrelation_Filter.txt`, where

- The first line is the file name of the preserved portion (output of `Decorrelation_Preserve.cpp`).
- The second line is the file name of the window width (output of `Decorrelation_Window.cpp`).
- The third line is the order of polynomial used to fit the correlated part (fixed to be 2 here).
- The fourth line is the maximum degree of the SHCs.
- The fifth line is the number of models of SHCs to be decorrelated.
- The following lines are input and output files.

An example of `Decorrelation_Filter.txt` is

```
preserved_r3.5_l055_lm28.txt
window_A20_p2.1_gamma0.18_K30_Nmax63.txt
2
63
3
2014_008_63_03.txt   2014_008_63_d4.txt
2014_105_63_03.txt   2014_105_63_d4.txt
2014_228_63_03.txt   2014_228_63_d4.txt
```

The effect of this decorrelation is shown in Fig. 2.3. We see the alleviation of the stripes.

Note that the decorrelation is empirical. The signal is also distort while removing the stripes. Therefore, a compromise should be made between the levels of stripe removal and signal distortion (Duan et al., 2009). In practical applications, the decorrelation should be applied at the step of the procedure best for the quantity studied. For example, when the linear trend in a period of time is studied, we should first fit the linear trend of the SHCs, and then apply the decorrelation.

2.2 Fitting polynomial and sinusoidal signals

The largest constituents in mass changes are the annual changes and trend. Hence, if the primary interest is trend or annual sinusoidal changes, one can first fit the change of SHCs or geoidal height to get the trend or annual changes before decorrelation-smoothing. As the fit removes random errors, this permits to use a weaker decorrelation filter, or even without the need of decorrelation.

The fit is done using the program `Polynomial_Sinusoids_fit.cpp`. However, there is a need of a program to prepare data file for the program `Polynomial_Sinusoids_fit.cpp`, and a program to extract results from the output.

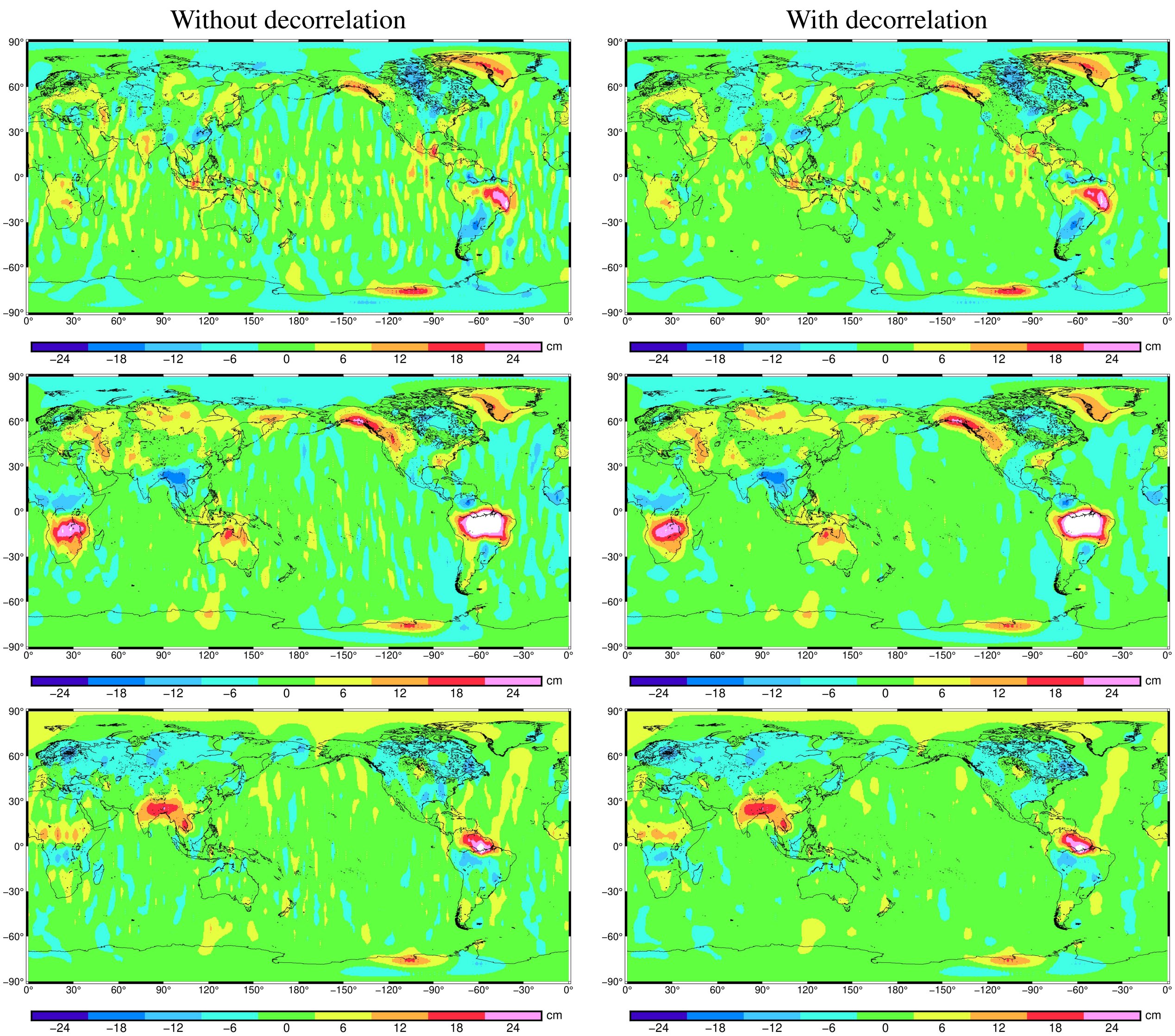

Figure 2.3: A comparison between the results without and with decorrelation. From top to bottom: January 2014, May 2014, and August 2014

2.2.1 Least squares fitting with a polynomial and sinusoids

We fit each data sequence with

$$\sum_{i=1}^{N_p} a_i(t-t_0)^{r_i} + \sum_{i=1}^{N_s} \left[b_i \cos\frac{2\pi(t-t_0)}{T_i} + c_i \sin\frac{2\pi(t-t_0)}{T_i} \right], \tag{2.3}$$

where the parameters t_0, N_p, $r_1, \cdots, r_{N_p}$ and N_s, $T_1, \cdots, T_{N_s}$ are preassigned.

The main program is `Polynomial_Sinusoids_fit.cpp`. It requires the following the `.cpp` file files `linalg.cpp` and the `.h` file `linalg.h` for to be compiled.

The program is written to fit multiple data sequences with the same sampling epochs.

The input/output information is specified in the file `Polynomial_Sinusoids_fit.txt`, where

- The first line is the reference epoch t_0.
- The second line is N_p, i.e., the number of monomials in the polynomial.
- The third line lists successively $r_1, \cdots, r_{N_p}$, noting that the program permits these to be any real values.
- The fourth line is N_s, i.e., the number of sinusoids.
- The fifth line lists successively $T_1, \cdots, T_{N_s}$.
- The sixth line is the file name of input data sequences, `data_sequences.txt`.
- The last line is the output file name.

An example of `Polynomial_Sinusoids_fit.txt` for the SHCs is

```
2015.5
2
0 1
2
0.5 1.0
Degree_2_SHCs_time_series.txt
Degree_2_SHCs_time_series_fit.txt
```

where the input `data_sequences.txt` is `Degree_2_SHCs_time_series.txt`.

An example of `Polynomial_Sinusoids_fit.txt` for the geoidal height is

```
2015.5
2
0 1
2
0.5 1.0
Geoid_63_merged.txt
Geoid_63_merged_fit.txt
```

where the input `data_sequences.txt` is `Geoid_63_merged.txt`.

The time unit is year in both the above examples.

The format of the file of the input data sequences is as follows:

- The first line is the number of data of the data sequences (the same for all the data sequences).
- Each of the following 'number of data' lines has 2 strings specifying `YYYY` and `DDD` (the same for all the data sequences). These are to be converted to year, e.g., $2002 + 289/365.25$ in the example file below.
- The next line includes two numbers: the first is the number of sequences; the second is the number of attributes describing each data sequence of length 'number of data'. The attributes are strings for keeping track of the name of each data sequence.
- In the next 'number of sequences' lines, the 'number of attributes' strings are followed by 'number of data' values of the sequence.

An example of `Degree_2_SHCs_time_series.txt` for the SHCs is

```
25
2014   008
2014   076
......
5  3
c  2   0   -1.081924160721906e-10   -6.487253209662772e-11  ......
s  2   0    0                        0                      ......
c  2   1    3.103183797920844e-11    6.774487134721438e-12  ......
s  2   1   -2.290238247993819e-11   -5.753257243708003e-12  ......
c  2   2   -1.097086951424352e-11   -3.340096412427336e-11  ......
s  2   2    1.72880754340331e-11    -1.070888939612101e-11  ......
```

This is for only the degree 2 SHCs. The attributes of the sequence 'c', '2', and '0', implies $C_{2,0}$.

An example of `Geoid_63_merged.txt` for the geoidal height is

```
25
2014  008
2014  076
......
16384   2
89.296875   0          -0.002850482963385581   -0.002605387694872563  ......
89.296875   2.8125     -0.002857665433543956   -0.002608664011569384  ......
89.296875   5.625      -0.002865035300725278   -0.002612052502029967  ......
......
```

This is for the $128 \times 128 = 16384$ grid points.

The format of the output data file is specified in the file itself.

2.2.2 Preparation of input data file

SHCs

This program is `Merge_SHCs_to_fit.cpp`. It requires the `.h` files `rshc.h` and `check_error.h` for to be compiled. It combines all the data files of the SHCs into one for using `Polynomial_Sinusoids_fit.cpp`, i.e, in the format of `data_sequences.txt` mentioned above.

The input/output information is specified in the file `Merge_SHCs_to_fit.txt`, where

- The first line is the output file name
- The second line is the degree of the SHCs.
- The third line is the number of files of SHCs to be used in the fit (file name must conform with the `YYYY_DDD_XXXX` convention.
- The following lines are list of file names of SHCs.

An example of `Merge_SHCs_to_fit.txt` is

```
SHCs_merged.txt
60
36
2019_016_03.txt
2019_046_03.txt
......
```

In the output file, the sequences of the coefficients are kept tracked using 3 attributes: 'c' or 's', degree, and order. The format is the same as `Degree_2_SHCs_time_series.txt` in the last subsection with all the SHCs included (for the maximum degree 60, the total number including $S_l, 0$ is $(60+1) \times (60+2) =$ 3782.

Geoid

The program is `Merge_Geoid_to_fit.cpp`. This is a single fie program.

The input/output information is specified in the file `Merge_Geoid_to_fit.txt`, where

- The first line is the output file name
- The second line includes successively the number of nodes along latitude and that along longitude.
- The third line is the number of data files to be used in the fit (file name must conform with the `YYYY_DDD_XXXX` convention.
- The following lines are list of file names of geoid data.

An example of `Merge_Geoid_to_fit.txt` is

```
Geoid_merged.txt
122 122
36
2019_016_03_Geoid.txt
2019_046_03_Geoid.txt
......
```

In the output file, the values of the geoidal height are kept tracked using 2 attributes: latitude and longitude. The format is the same as `Geoid_63_merged.txt` in the last subsection.

We used the geoidal height as an example of grided data. The program applies to any kinds of grided data.

2.2.3 Least squares fitting

This is straightforward using the program `Polynomial_Sinusoids_fit.cpp`. The format of the output file is specified in the file itself.

2.2.4 Extraction of results

SHCs

The program is in the file `Extract_SHCs_After_Fit.cpp`. It requires the `.h` files `rshc.h` and `check_error.h` for to be to be compiled.

The input/output information is specified in the file `Extract_SHCs_After_Fit.txt`, where

- The first line is the file name of the output of the program `Polynomial_Sinusoids_fit.cpp`.
- The second line is the number of polynomial terms used in the fit.
- The third line is a list of the exponentials (not used, the total number is equal to the value of the previous line).
- The fourth line is the number of sinusoids used in the fit.
- The fifth line is a list of the periods of the sinusoid (not used, the total number is the same as the value of the previous line).
- The sixth line should either `polynomial` or `sinusoid` indicating the type of constituent to be extracted.
- The seventh line gives the order of the constituent (polynomial of sinusoid) to be extracted.
- If the sixth line is `polynomial`, the eighth line gives the file name of the output of the polynomial coefficients.
- If the sixth line is `polynomial`, the ninth line gives the file name of the output of the polynomial coefficients in an alternative format (the format of a line is [‘c or s’ ‘degree’ ‘order’ ‘coefficient’], for use by the program `Remove_Polynomial_Sinusoids.txt`).
- If the sixth line is `sinusoid`, the eighth line gives the file name of the outputs of the cos coefficients and that of the sin coefficients.
- If the sixth line is `sinusoid`, the eighth line gives the file name of the outputs of the cos coefficients and that of the sin coefficients in an alternative format (the format of a line is [‘c or s’ ‘degree’ ‘order’ ‘coefficient’], for use by the program `Remove_Polynomial_Sinusoids.txt`).

An example of `Extract_SHCs_After_Fit.txt` is

```
SHCs_merged_fit.txt
2
0 1
2
0.5 1
polynomial
```

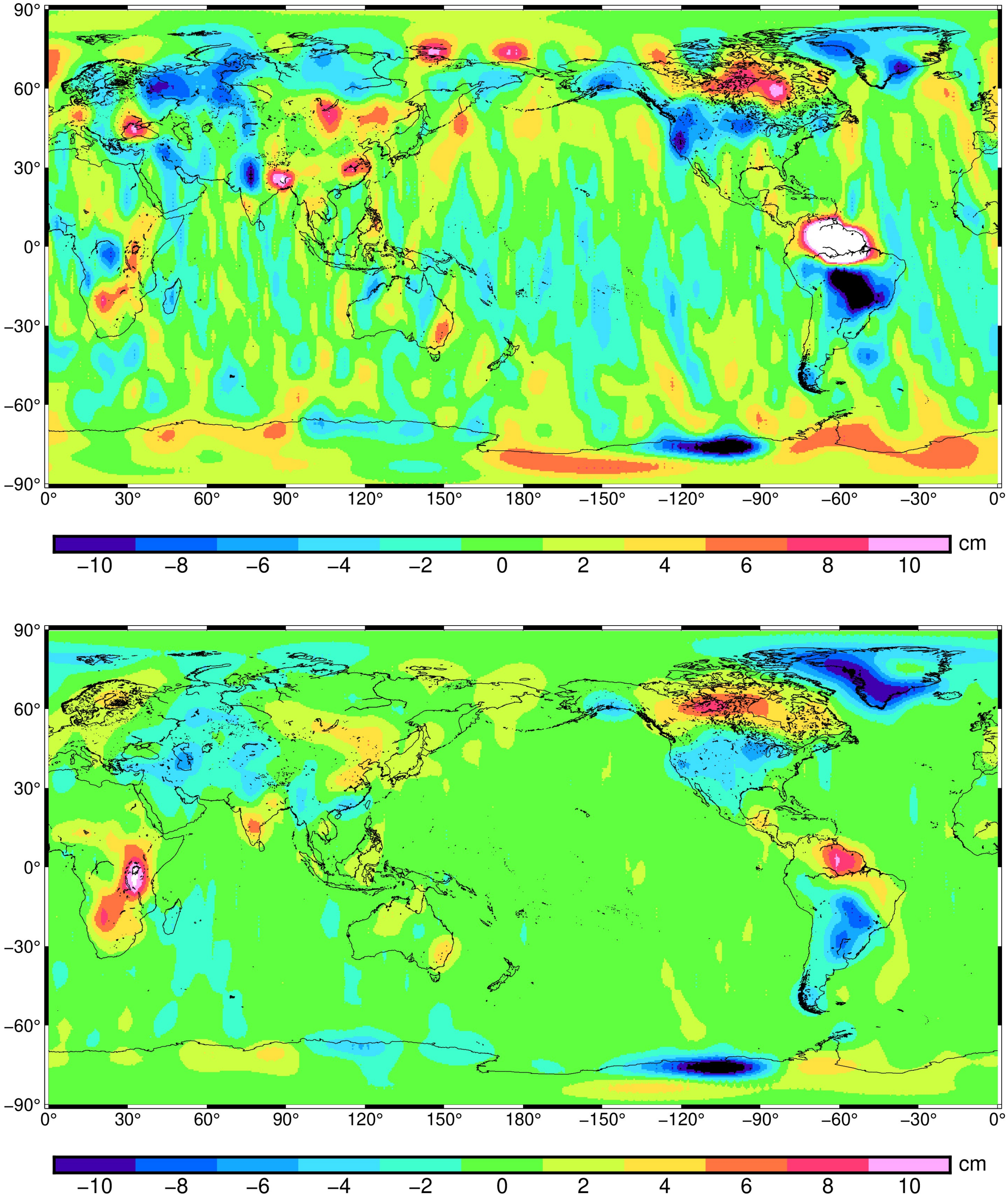

Figure 2.4: A comparison between the anomalous mass of a month (July 2021, Top) and the trend (January 2019 to December 2021, Bottom), both smoothed with a smoothing radius of 300 km. These are computed using the GRACE-FO models up to degree and order 60. The stripes are much weaker in the trend.

```
2
SHCs_trend.txt
SHCs_trend_ADO_Format.txt
```

This is to extract the trend, i.e., the coefficient of the second monomial.

Geoid

The program is in the file `Extract_Geoid_After_Fit.cpp`. This is a single file program.

The input/output information is specified in the file `Extract_Geoid_After_Fit.txt`. Data are similar to the file Extract_SHCs_After_Fit.txt without the output in an alternative format.

An example of `Extract_Geoid_After_Fit.txt` is

```
Geoid_merged_fit.txt
2
0 1
2
0.5 1
sinusoid
2
Geoid_cos.txt  Geoid_sin.txt
```

This is to extract the cosine and sine coefficients of the second sinusoid, i.e., the annual signal. The output format for of the geoid conforms with the requirement of the program `Remove_Polynomial_Sinusoids.txt`.

Again, we used the geoidal height as an example of grided data. The program applies to any kinds of grided data.

2.2.5 Comments

While fitting, the bad monthly solutions can be excluded, since the epoches of the solutions are handled explicitly. Particularly, the monthly solutions can be used without subtracting a reference model. In this case, the order zero term in the result of the fit can be used as the reference model, where the initial epoch can be chosen as the middle of the first and last month.

2.3 Extract and replace a group of SHCs

We may want to study a particular group of the GRACE SHCs, or replace them by those determined using other technologies such as satellite laser ranging. The information of the group of SHCs includes the coefficient attributes (C/S, degree, order), epochs (year and median date as provided by the file names), and value at each epoch.

2.3.1 Extract

The program for extracting a group of SHCs is in the file `Extract_Group_SHCs.cpp`. This is a single `.cpp` file program that requires the `.h` files `rshc.h`, `vec.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Extract_Group_SHCs.txt`, where

- The first line is the 'number of coefficients' to be extracted.
- The following 'number of coefficients' lines specify the coefficients. There are 3 data in each line. The first data is either 'c' or 's', meaning the coefficient is 'C' or 'S'. The other two data are degree and order.
- The (one plus number of coefficients)th line is the output file name.
- The (two plus number of coefficients)th line is the maximum degree and order of the SHCs.
- The (three plus number of coefficients)th line is the number of SHCs files.
- The next 'number of SHCs files' lines are the names of the SHCs files (the file names should conform with the format `YYYY_DDD_XXX.txt`).

An example of `Extract_Group_SHCs.txt` is

```
6
c 2 0
s 2 0
c 2 1
s 2 1
c 2 2
s 2 2
Degree_2_SHCs_time_series.txt
63
25
2014_008_63_03.txt
2014_076_63_03.txt
  ......
```

This is to extract only the degree 2 SHCs. The output file `Degree_2_SHCs_time_series.txt` is the same as that of the same name in Sect. 2.2.1.

2.3.2 Replace

The program for replacing a group of SHCs is `Replace_Group_SHCs.cpp`, which is single `.cpp` file and requires the `.h` files `rshc.h`, `vec.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Replace_Group_SHCs.txt`, where

- The first line is the maximum degree and order of the SHCs data.
- The second line is the file including the new SHCs to replace the SHCs in the data files. The format is the same as the output file of `Extract_Group_SHCs.cpp` (or the output of `Merge_SHCs_to_fit.cpp` described in Sect. 2.2.4 if all the SHCs are concerned).
- The third line is the number of data files.
- The following 'number of data files' lines are the input and output of the data files of SHCs.

The number of data files and the names of the data files must agree with the number and the YYYY and DDD in the data file specifying the new SHCs.

An example of Replace_Group_SHCs.txt is

```
63
Degree_2_SHCs_time_series.txt
25
2014_008_63_03.txt  2014_008_63_03_D2_Replaced.txt
2014_076_63_03.txt  2014_076_63_03_D2_Replaced.txt
......
```

This is to replace the SHCs in 2014_008_63_03.txt, 2014_076_63_03.txt, ... by those specified in Degree_2_SHCs_time_series.txt, and save as 2014_008_63_03_D2_Replaced.txt, 2014_008_63_03_D2_Replaced.txt, ...

Replacing all the SHCs implies to create a sequence of new files of the SHCs. In this case, the files of the old SHCs can be any whose YYYY and DDD agree with those specified in the data file specifying the new SHCs.

2.4 Subtraction of polynomial-sinusoids from the SHCs

This program is for removing a combination of a polynomial and several sinusoids as expressed in eqn (2.3) from a group of SHCs extracted using Extract_Group_SHCs.cpp described in Section 2.3.1 (or the output of Merge_SHCs_to_fit.cpp described in Sect. 2.2.4 if all the SHCs are concerned).

The program file is Remove_Polynomial_Sinusoids.cpp, which requires the .cpp file linalg.cpp and the .h file linalg.h for to be compiled.

- The input/otput information is specified in the file Remove_Polynomial_Sinusoids.txt, where
- The first line specifies the reference time t_0 in year like 2015.5.
- The second line specifies the number of monomials in the polynomial N_p.
- The following N_p lines specify r_i and the file name of a_i.
- The next line specify the number of sinusoids N_s.
- The following N_s lines specify T_i and the file names of b_i and c_i.
- The last two lines are input and output files of the SHCs in the format of the output of the program Extract_Group_SHCs.txt (or the output of Merge_SHCs_to_fit.cpp described in Sect. 2.2.4 if all the SHCs are concerned).

An example of Remove_Polynomial_Sinusoids.txt for the SHCs is

```
2015.5
1
1.0 SHCs_63_trend_attr.txt
1
1.0 SHCs_63_cos_attr.txt  SHCs_63_sin_attr.txt
SHCs_63_merged.txt
SHCs_63_merged_removed_trend_annual.txt
```

This removes the linear trend and the 1-year period sinusoid.

An example of `Remove_Polynomial_Sinusoids.txt` for the geoidal height is

```
2015.5
1
1.0 Geoid_63_trend.txt
1
1.0 Geoid_63_cos.txt  Geoid_63_sin.txt
Geoid_63_merged.txt
Geoid_63_merged_removed_trend_annual.txt
```

This also removes the linear trend and the 1-year period sinusoid.

This program can be used to remove periodical errors such as the 161 day error. Certainly, to find its coefficients of the sine and cosine terms, this period (161/365.25=0.4414 year) should have been included in the polynomial-sinusoidal fitting using the program `Polynomial_Sinusoids_fit.cpp`.

After having removed the polynomial-sinusoids in the format of the output of `Extract_Group_SHCs.txt`, the SHCs can be used to replace the original data using `Replace_Group_SHCs.cpp`.

2.5 Remove the GIA effect from the SHCs

Here we assume to remove the glacial isostatic adjustment (GIA) effect from the GRACE SHCs before studying present day mass changes. Contemporary GIA models provide a gravitational potential model specifically for use to interpret GRACE data. Therefore, we just have to truncate the model to the degree and order of the GRACE SHCs. We take as example a model provided by Peltier. The first few lines of the original model file are edited for use by the program `Truncate_GIA_SHCs.cpp`, which reads the SHCs in the GIA model format and then truncate and output in the GRACE SHC data format of our convention. The original model and the reduction to the formats for our applications are all in the folder `GIA`.

The program for removing the GIA effect from the GRACE data of SHCs is in the single `.cpp` file `Remove_GIA.cpp`, which requires the ḣ file `rshc.h` for to be compiled.

The input/output information is specified in the file `Remove_GIA.txt`, where

- The first line is the maximum degree of the SHCs of both the GIA rate and GRACE data.
- The second line is the file name of the reference model removed from all GRACE monthly data. We don't use data in this file. We only need the file name in our `YYYY_DDD_XXX.txt` convention to retrieve the reference epoch from which the GIA effect (changing part) is computed.
- The third line is the file name of the SHCs of the GIA model.
- The fourth line is the number of GRACE data files of SHCs.
- The following are the inputs and outputs of the data files.

An example of `Remove_GIA.txt` is

```
63
2015_183_63_REFERENCE.txt
GIA_63.txt
```

```
25
2014_008_63_03.txt  2014_008_63_03_Removed_GIA.txt
2014_076_63_03.txt  2014_076_63_03_Removed_GIA.txt
......
```

The GIA effect should be removed before decorrelation and smoothing if it is not to be studied using the data of SHCs.

2.6 Smoothing and leakage reduction

Gaussian smoothing merges data of neighboring locations, causing leakage. If the signal outside of the region of interest is of much smaller magnitude, or could be reduced to much smaller magnitude using an model, leakage could be very much alleviated. The reference is Guo et al. (2010).[2] In our programs, we assume the signal over the oceans is far smaller than that over the land, with land and ocean defined by the ocean function (0 over the land, 1 over the oceans). If the region of interest is a particular basin, the ocean function could be replaced by the basin function (0 within the basin, 1 outside).

Here we work on a regular latitude-longitude grid of grid intervals $\Delta B = 180/M$ and $\Delta\lambda = 360/N$, where M and N are numbers of nodes along meridians and parallels. The grid centers are defined by

$$\begin{array}{lll} B_j = 90 - (j + 1/2) * \Delta B & \text{for} & j = 0, 1, \cdots, M - 1\,, \\ \lambda_k = k * \Delta\lambda & \text{for} & k = 0, 1, \cdots, N - 1\,. \end{array}$$

This grid is more general than that used in the discrete spherical harmonic transform in the transformation between the SHCs and the grid values. Particularly, it is of interest to use a grid with the same grid interval along both B and λ, e.g., $M = 180$ and $N = 360$ for a $1^\circ \times 1^\circ$ grid.

The programs are composed of Gaussian smoothing, leakage reduction for the land, leakage reduction for the oceans, and merging for three different smoothing filters: isotropic Guassian, non-isotropic Gaussian and their mix (non-isotropic Gaussian at lower latitudes and isotropic Guassian at polar region).

2.6.1 Smoothing

Isotropic Gaussian

The program is in the file `Filter_Global_Gauss_Iso.cpp`. It requires the `.c` file `fftpack.c` and the `.h` files `Filter_Global_Gauss_Iso.h`, `grid_conv.h`, `rsht.h`, `fftpack++.h`, `fftpack.h`, `rshc.h`, and `mat.h` for to be compiled.

The input/output information is assigned in the file `Filter_Global_Gauss_Iso.txt`, where

- The first line includes two integers, the first one is the number of nodes in latitude, and the second that in longitude.
- The second line is the smoothing radius in degree.
- The third line is the number of data files to apply the smoothing.
- The following lines are input and output files.

[2]Guo, J.Y., Duan, X.J. and Shum, C.K., 2010. Non-isotropic Gaussian smoothing and leakage reduction for determining mass changes over land and ocean using GRACE data, Geophys. J. Int., 181, 290-302, 10.1111/j.1365-246X.2010.04534.x

An example of `Filter_Global_Gauss_Iso.txt` is

```
180 360
3.0
2
Mass_1dx1d_trend.txt        Mass_1dx1d_trend_ISO_3d.txt
Mass_1dx1d_2014_105.txt     Mass_1dx1d_2014_105_ISO_3d.txt
```

This program does the same as that of `Isotropic_Gaussian_Grid.cpp`. The difference is that this program allows different numbers of grid points along the latitude and longitude. Anyway, the smoothing is done in the same way as in `Isotropic_Gaussian_Grid.cpp` with the input grid values transformed to the $M = N = 2bw$ grid adopted in the discrete spherical harmonic transform with $bw = N/2$ ($bw = (N-1)/2$ if N is odd). The result is then transformed back to the original grid of M points of B and N points of λ. If the grid type of the input data is of different type, the data should be first transformed to the format adopted. See Appendix A.2 for the transformation of data from one grid to another.

Non-isotropic Gaussian

The program is `Filter_Global_Gauss_Non_Iso.cpp`, which requires the `.c` file `fftpack.c` and the `.h` files `Filter_Global_Gauss_Non_Iso.h`, `rfss.h`, `rfss.h`, `fftpack++.h`, `fftpack.h`, and `mat.h` for to be compiled.

The input/output information is assigned in the file `Filter_Global_Gauss_Non_Iso.txt`, where

- The first line is the number of nodes in latitude and longitude, respectively.
- The second line should be either "wW" or "nW", indicating if the weight (for plane or spherical Earth" is applied.
- The third line is the smoothing radii alone latitude and longitude, respectively, in degree.
- The fourth line is the number of data files to apply the program.
- The following lines are input and output files.

An example of `Filter_Global_Gauss_Non_Iso.txt` is

```
180 360
wW
3.0 6.0
2
Mass_1dx1d_trend.txt        Mass_1dx1d_trend_NonISO_3x6.txt
Mass_1dx1d_2014_105.txt     Mass_1dx1d_2014_105_NonISO_3x6.txt
```

Mixed Gaussian

The program is `Filter_Global_Gauss_Mix.cpp`, which requires the `.c` file `fftpack.c` and the `.h` files `Filter_Global_Gauss_Mix.h`, `Filter_Global_Gauss_Iso.h`, `grid_conv.h`, `rdsht.h`, `rfss.h`, `fftpack++.h`, `fftpack.h`, and `mat.h` for to be compiled.

The input/ouput information is assigned in the file `Filter_Global_Gauss_Mix.txt`, where

- The first line includes the number of nodes in latitude and that in longitude.

- The second line should be either ”wW” or ”nW”, indicating if the weight (for plane or spherical Earth) is applied.
- The third line includes the smoothing radii alone the latitude and longitude in degree.
- The fourth line is the number of data files to apply the program.
- The following lines are input and output files.

An example of `Filter_Global_Gauss_Mix.txt` is

```
180 360
wW
3.0 6.0
2
Mass_1dx1d_trend.txt        Mass_1dx1d_trend_Mix_3x6.txt
Mass_1dx1d_2014_105.txt     Mass_1dx1d_2014_105_Mix_3x6.txt
```

2.6.2 Leakage reduction for land

Leakage reduction depends on the type of smoothing filter. The inputs to leakage reduction programs are smoothed mass data. The outputs of the programs are leakage reduced (scaled) data over land, and zero over ocean.

Isotropic

The program is `Leakage_Land_Gauss_Iso.cpp`, which requires the `.h` files `Leakage_Land_Gauss_Iso.h`, `grid.h`, `vec.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Leakage_Land_Gauss_Iso.txt`, where

- The first line is the file name of ocean function data.
- The second line includes number of nodes along latitude and that along longitude.
- The third line is the smoothing radius in degree.
- The fourth line includes the number of small grids in each of the original grid along latitude and longitude for more precisely evaluation. The two values should be odd.
- The fifth line is the number of data files to apply leakage reduction.
- The following lines are input and output file pairs.

An example of `Leakage_Land_Gauss_Iso.txt` is

```
OF_1dx1d_gauss.txt
180 360
3.0
3 3
2
Mass_1dx1d_trend_ISO_3d.txt       Mass_1dx1d_trend_ISO_3d_Land.txt
Mass_1dx1d_2014_105_ISO_3d.txt    Mass_1dx1d_2014_105_ISO_3d_Land.txt
```

Non-isotropic

The program file is `Leakage_Land_Gauss_Non_Iso.cpp`, which requires the `.h` files `Leakage_Land_Gauss_Non_Iso.h`, `grid.h`, `vec.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Leakage_Land_Gauss_Non_Iso.txt`, where

- The first line is the file name of ocean function data.
- The second line includes number of nodes along latitude and longitude.
- The third line should either ”wW” or ”nW”, indicating if the $\sin\theta$ factor is to be included (consider Earth’s surface as flat or spherical).
- The fourth line includes the smoothing radius along latitude and longitude in degree of angle.
- The fifth line includes the number of small grids in each of the original grid along latitude and longitude for more precisely evaluation. The two values should be odd.
- The sixth line is the number of data files to apply leakage reduction.
- The following lines are input and output file pairs.

An example of `Leakage_Land_Gauss_Non_Iso.txt` is

```
OF_1dx1d_gauss.txt
180 360
wW
3.0 6.0
3 3
2
Mass_1dx1d_trend_NonISO_3x6.txt       Mass_1dx1d_trend_NonISO_3x6_Land.txt
Mass_1dx1d_2014_105_NonISO_3x6.txt    Mass_1dx1d_2014_105_NonISO_3x6_Land.txt
```

Mixed Gaussian

The program file is `Leakage_Land_Gauss_Mix.cpp`, which requires the `.h` files `Leakage_Land_Gauss_Mix.h`, `grid.h`, `vec.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Leakage_Land_Gauss_Mix.txt`, where

- The first line is the file name of ocean function data.
- The second line includes number of nodes along latitude and that along longitude.
- The third line should either ”wW” or ”nW”, indicating if the $\sin\theta$ factor is to be included (consider Earth’s surface as flat or spherical).
- The fourth line includes the smoothing radius along latitude and longitude in degree of angle.
- The fifth line includes the number of small grids in each of the original grid along latitude and longitude for more precisely evaluation. The two values should be odd.
- The sixth line is the number of data files to apply leakage reduction.

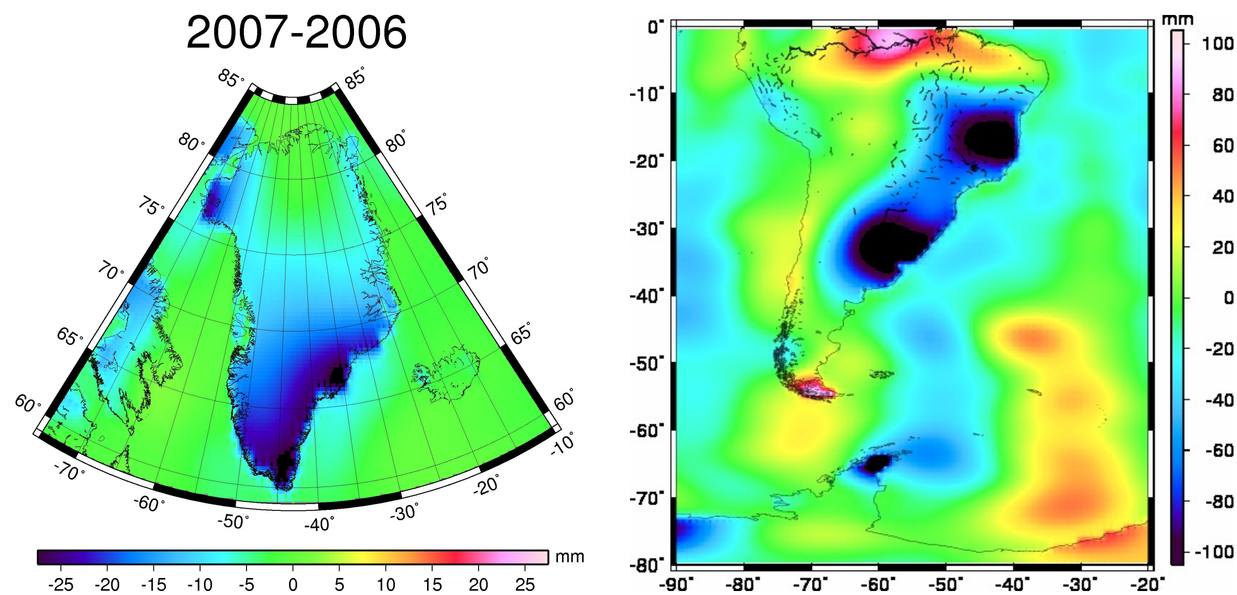

Figure 2.5: Demonstration of the effect of leakage reduction.

- The following lines are input and output file pairs.

An example of `Leakage_Land_Gauss_Mix.txt` is

```
OF_1dx1d_gauss.txt
180 360
wW
3.0 6.0
3 3
2
Mass_1dx1d_trend_Mix_3x6.txt        Mass_1dx1d_trend_Mix_3x6_Land.txt
Mass_1dx1d_2014_105_Mix_3x6.txt     Mass_1dx1d_2014_105_Mix_3x6_Land.txt
```

2.6.3 Leakage reduction for ocean and merging with results over land

The program file is `Leakage_Land_Ocean_Merge.cpp`, which requires the `.h` files `grid.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is assigned in the file `Leakage_Land_Ocean_Merge.txt`, where

- The first line is the file name of ocean function data.
- The second line provides the latitudinal and longitudinal grid numbers.
- The third line is the number of mass change files computed.
- The following lines are inputs and output of a group of four files. There are 3 input files in each group: (1) the smoothed mass changes, (2) leakage reduced land mass changes (zero over ocean), (3) the re-smoothed leakage reduced land mass changes. The last one of the group is the file name for the result. The same smoothing filter should be used throughout.

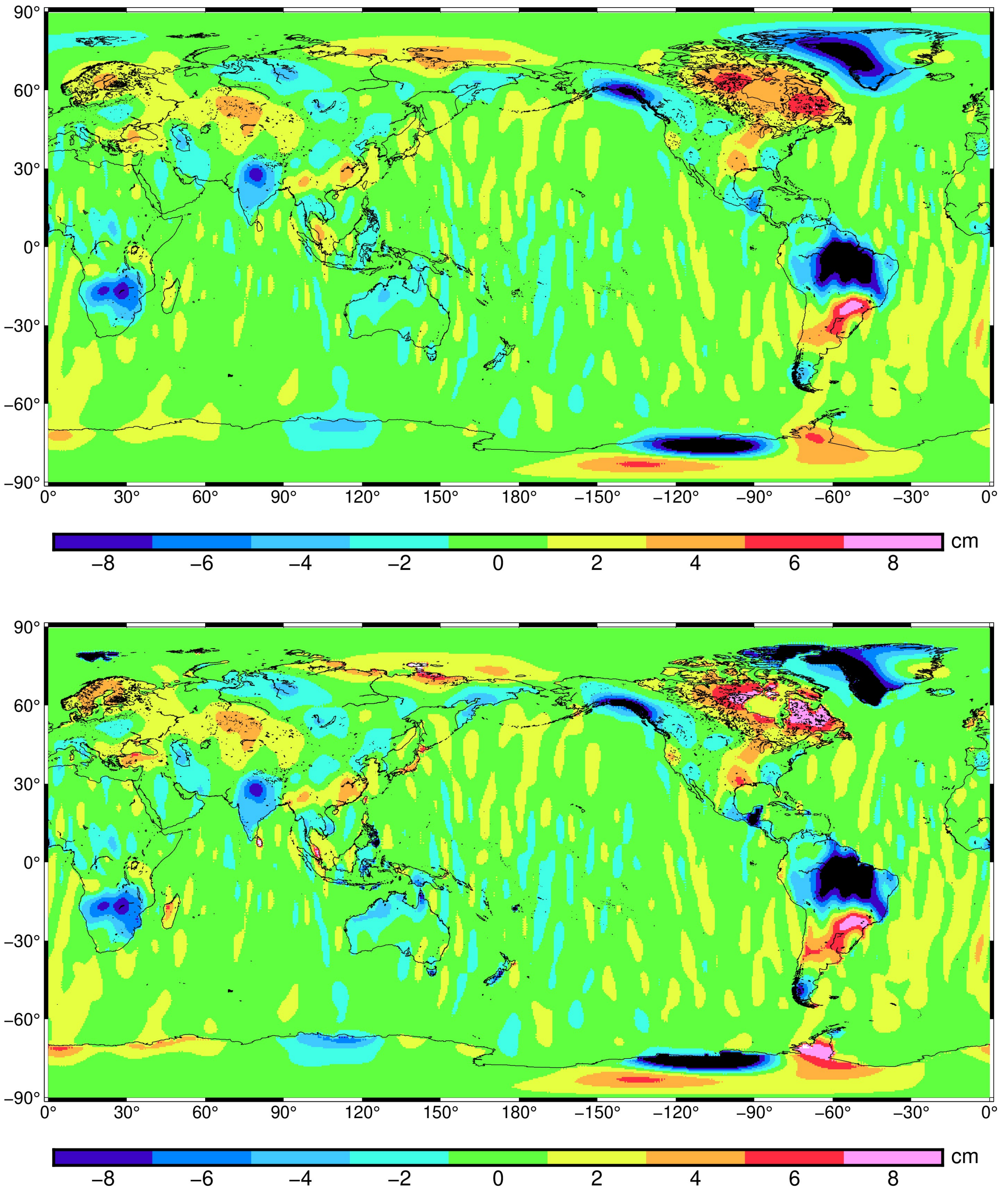

Figure 2.6: A comparison between the mass change without (top) and with (bottom) leakage reduction: The isotropic Gaussian smoothing with a smoothing radius of 3°.

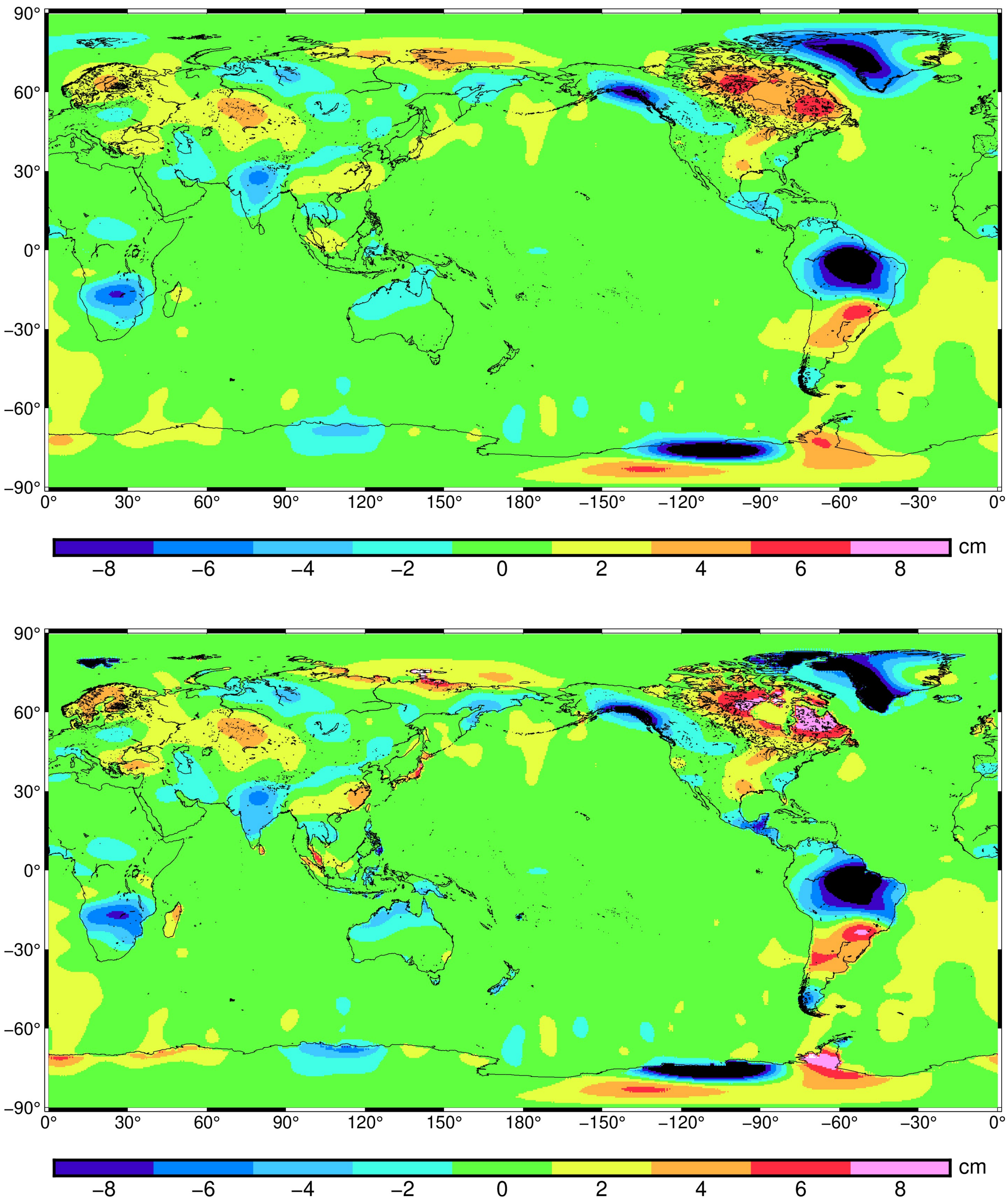

Figure 2.7: A comparison between the mass change without (top) and with (bottom) leakage reduction: The mixed isotropic Gaussian smoothing with a 3° smoothing radius along the latitude and 6° along the longitude in $B < 60^\circ$, and an isotropic Gaussian smoothing with a smoothing radius of 3° in $B \geq 60^\circ$

This program requires only the same smoothing is done for each of the lines after the third line. Certainly, the data in the first three lines are the same for each group of the four files.

An example of `Leakage_Land_Ocean_Merge.txt` is

```
OF_1dx1d_gauss.txt
180 360
3
Mass_ISO_s.txt  Mass_ISO_s_L.txt  Mass_ISO_s_L_s.txt  Mass_ISO_Merged.txt
Mass_nISO_s.txt Mass_nISO_s_L.txt Mass_nISO_s_L_s.txt Mass_nISO_Merged.txt
Mass_Mix_s.txt  Mass_Mix_s_L.txt  Mass_Mix_s_L_s.txt  Mass_Mix_Merged.txt
```

The first column is the first smoothed result. The second column is the result of leakage reduction over land. The third column is the result obtained by smoothing the second column. The forth column is the result of leakage reduction over both the ocean s and the land.

Fig. 2.5 demonstrates the effect of leakage reduction with the zoomed views of two examples. We show in Fig. 2.6 and 2.7 the result of leakage reduction for the isotropic and mixed Gausian smoothing. We see that the larger smoothing radius along the longitude removes more stripes.

2.7 Transformation between SHCs and Mass for an Aspherical Earth surface

The formulation is included in the book manuscript Guo (2025).[3] The Earth's surface is assumed to be aspherical with a topography superimposed on an ellipsoid.

As the topographic height above sphere of radius $R = 6371$ km, we add the CRUST1.0 topographic height to the hight of the ellipsoid of flattening $\alpha = 1/298.257$ above the sphere.

The height of the ellipsoid above the sphere is computed using

$$r - R = R\left[\frac{2}{3}\alpha P_2(\cos\theta)\right]$$

It is a degree 2 and order 0 spherical harmonics.

Two programs are written for the transformation between the mass density and the SHCs of the potential based on the aspherical Earth surface: one for computing the Stokes coefficients (the SHCs of the gravitational potential) of a layer of mass on the aspherical Earth surface, one for computing the mass density of the layer on the aspherical Earth surface from the Stokes coefficients.

We have created topographic surface of two resolutions (height of the ellipsoid above the sphere plus the topography above the ellipsoid). We used CRUST1.0 and etopo5m, both being understood as the height above the ellipsoid.

The computation of the mass density from the SHCs is an iterative procedure. We have fix the number of iteration to 5. The relative error is on the order of 1.4×10^{-6} percent. This is absolutely sufficient for GRACE application. Note that this iteration takes only a small fraction of the computation time.

[3]Guo, J.Y. (2025) Global Elastic Deformation of the Earth. Division of Geodetic Science, School of Earth Sciences, The Ohio State University.

2.7.1 Mass2SHCs_AsphericalEarth

The program is `Mass2SHCs_AsphericalEarth.cpp`. It requires the `.c` program `fftpack.c` and the `.h` files `.h` files `rsht.h`, `legendre.h`, `rshana.h`, `rshsyn.h`, `rshc.h`, `grid.h`, `mat.h`, `vec.h`, `trig.h`, , , `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is described in the file `Mass2SHCs_AsphericalEarth.txt`, where

- The first line is the value of R, i.e., the average radius of the Earth.
- The second line is the value of bw of the topography grid data.
- The third line is the file name of the topography grid data.
- The fourth line is the value of bw of the GRACE data.
- The fifth line is the number of GRACE data file to be processed.
- Each of the following line includes the input file name and output file name.

An example of `Mass2SHCs_AsphericalEarth.txt` is

```
6371.0e+3
128
surface_flat_etopo_bw128_meter.txt
64
1
Mass_63_trend.txt SHCs_63_trend.txt
```

2.7.2 SHCs2Mass_AsphericalEarth

The program is `SHCs2Mass_AsphericalEarth.cpp`. It requires the `.c` program `fftpack.c` and the `.h` files `.h` files `rsht.h`, `legendre.h`, `rshana.h`, `rshsyn.h`, `rshc.h`, `grid.h`, `mat.h`, `vec.h`, `trig.h`, , , `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is described in the file `SHCs2Mass_AsphericalEarth.txt`, where

- The first line is the value of R, i.e., the average radius of the Earth.
- The second line is the value of bw of the topography grid data.
- The third line is the file name of the topography grid data.
- The fourth line is the value of bw of the GRACE data.
- The fifth line is the number of GRACE data file to be processed.
- Each of the following line includes the input file name and output file name.

An example of `SHCs2Mass_AsphericalEarth.txt` is

```
6371.0e+3
128
surface_flat_etopo_bw128_meter.txt
64
1
SHCs_63_trend.txt Mass_63_trend_1.txt
```

An iteration is required. We verified the accuracy by first transforming the mass density to SHCs. We then compute the mass from the SHCs under different approximations. The relative accuracy are as follows: Spherical approximation: 5.0%, The approximate formula with an aspherical correction: 0.33%, One more iteration: 0.023%. We have fix the number of iteration to 5. The relative error is on the order of $1.4 \times 10^{-6}\%$ (run `Mass2SHCs_AsphericalEarth` and `SHCs2Mass_AsphericalEarth` successively, and then compare). These error levels vary with different data of input, but their orders of magnitude remain approximately the same.

2.7.3 Mass_Corrected_Asphericity

The program is `Mass_Corrected_Asphericity.cpp`. It corrects the mass computed based on spherical approximation by adding the correction due to topography. It requires the `.c` program `fftpack.c` and the `.h` files `.h` files `rsht.h`, `legendre.h`, `rshana.h`, `rshsyn.h`, `rshc.h`, `grid.h`, `mat.h`, `vec.h`, `trig.h`, , , `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is described in the file `Mass_Corrected_Asphericity.txt`, where

- The first line is the value of R, i.e., the average radius of the Earth.
- The second line is the value of bw of the topography grid data.
- The third line is the file name of the topography grid data.
- The fourth line is the value of bw of the GRACE data.
- The fifth line is the number of GRACE data file to be processed.
- Each of the following line includes the input file name and output file name.

An example of `Mass_Corrected_Asphericity.txt` is

```
6371e+3
128
surface_flat_etopo_bw128_meter.txt
64
1
Mass_63_trend_s.txt Mass_63_trend_2.txt
```

An iteration is required. The accuracy of our example is on the order of 0.03% (successively run `Mass2SHCs_AsphericalEarth`, `SHCs2Mass`, and `Mass_Corrected_Asphericity`, and then compare). The difference from the that of `SHCs2Mass_AsphericalEarth` could be due to the placement of the load Love numbers: the former to the SHCs of the result obtained under spherical approximation, and the later to the SHCs of the result with topographic effect (the same is in `Mass2SHCs_AsphericalEarth`.

As the load Love numbers are formulated based on spherical symmetry, both cased are considered to be correct. Hence, 0.03% should be understood as the accuracy of the formulation. Furthermore, the different Earth parameters with limited number of digits used in `SHCs2Mass` and the other programs can also bring some difference. GRACE/GRACE-FO is not supposed to achieve an accuracy of 0.03%.

2.7.4 SHCs_Corrected_Asphericity

The program is `SHCs_Corrected_Asphericity.cpp`. It corrects the SHCs computed based on spherical approximation by adding the correction due to topography. It requires the `.c` program `fftpack.c` and the `.h` files `.h` files `rsht.h`, `legendre.h`, `rshana.h`, `rshsyn.h`, `rshc.h`, `grid.h`, `mat.h`, `vec.h`, `trig.h`, , , `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is described in the file `SHCs_Corrected_Asphericity.txt`, where

- The first line is the value of R, i.e., the average radius of the Earth.
- The second line is the value of bw of the topography grid data.
- The third line is the file name of the topography grid data.
- The fourth line is the value of bw of the GRACE data.
- The fifth line is the number of GRACE data file to be processed.
- Each of the following line includes the input file name and output file name.

An example of `SHCs_Corrected_Asphericity.txt` is

```
6371.0e+3
128
surface_flat_etopo_bw128_meter.txt
63
1
SHCs_63_trend_s.txt SHCs_63_trend_1.txt
```

The accuracy of our example is on the order of 0.03% (successively run `Mass2SHCs`, `SHCs_Corrected_Asphericity`, and `SHCs2Mass_AsphericalEarth`, and then compare). The explanation is the same as that of `Mass_Corrected_Asphericity`.

2.8 Solution of the sea level equation for the elastic Earth model

Water storage exchanges between land and ocean influences sea level, that in turn influences estimates of land water storages, especially over Greenland and Antarctica.

The process to alleviate this kind of leakage is to obtain a preliminary estimate of land water storage changes, compute sea level changes by solving the sea level equation with an elastic Earth model, and remove the sea level change from GRACE mass changes before smoothing. The procedure adopted is included in the book manuscript Guo (2025).[4]

[4] Guo, J.Y. (2025) Global Elastic Deformation of the Earth. Division of Geodetic Science, School of Earth Sciences, The Ohio State University.

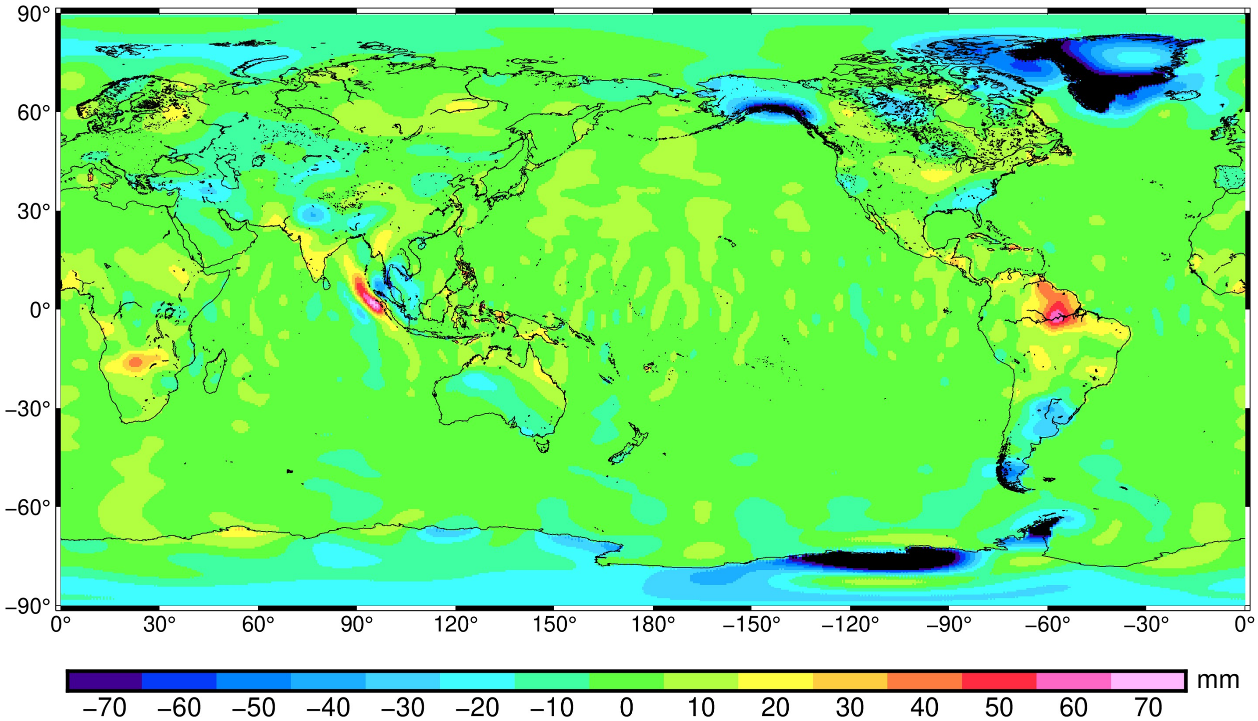

Figure 2.8: An example of mass redistribution used as input of the sea level equation solver.

The program file for solving the sea level equation is `Sea_Level_Equation_Elastic.cpp`. It requires the `.c` file `fftpack.c` and the `.h` files `Sea_Level_Equation_Elastic.h`, `rsht.h`, `legendre.h`, `rshana.h`, `rshsyn.h`, `rshc.h`, `grid.h`, `mat.h`, `vec.h`, `trig.h`, , , `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is escribed in the file `Sea_Level_Equation_Elastic.txt`, where

- The first line is a string to describe the problem, and will be used as first letters of output files.
- The second line is the band width bw related to grid nodes as given bellow:

$$B = 90 - (j + 1/2) * 180/(2 * bw) \quad \text{for} \quad j = 0, 1, \cdots, 2 * bw - 1$$
$$\lambda = k * 360/(2 * bw) \quad \text{for} \quad k = 0, 1, \cdots, 2 * bw - 1$$

- The third line is a threshold of precision used to stop the iteration.
- The fourth line is the radius of the Earth.
- The fifth line is the average density of the Earth.
- The sixth line is the value of J_2.
- The seventh line is the value of

$$\frac{\Omega^2 R^3}{GM} = 3.4498 \times 10^{-3}.$$

- The eighth line is the degree 2 body tides Love numbers in the order of k2,h2 and l2;
- The ninth line is the density of ocean water.

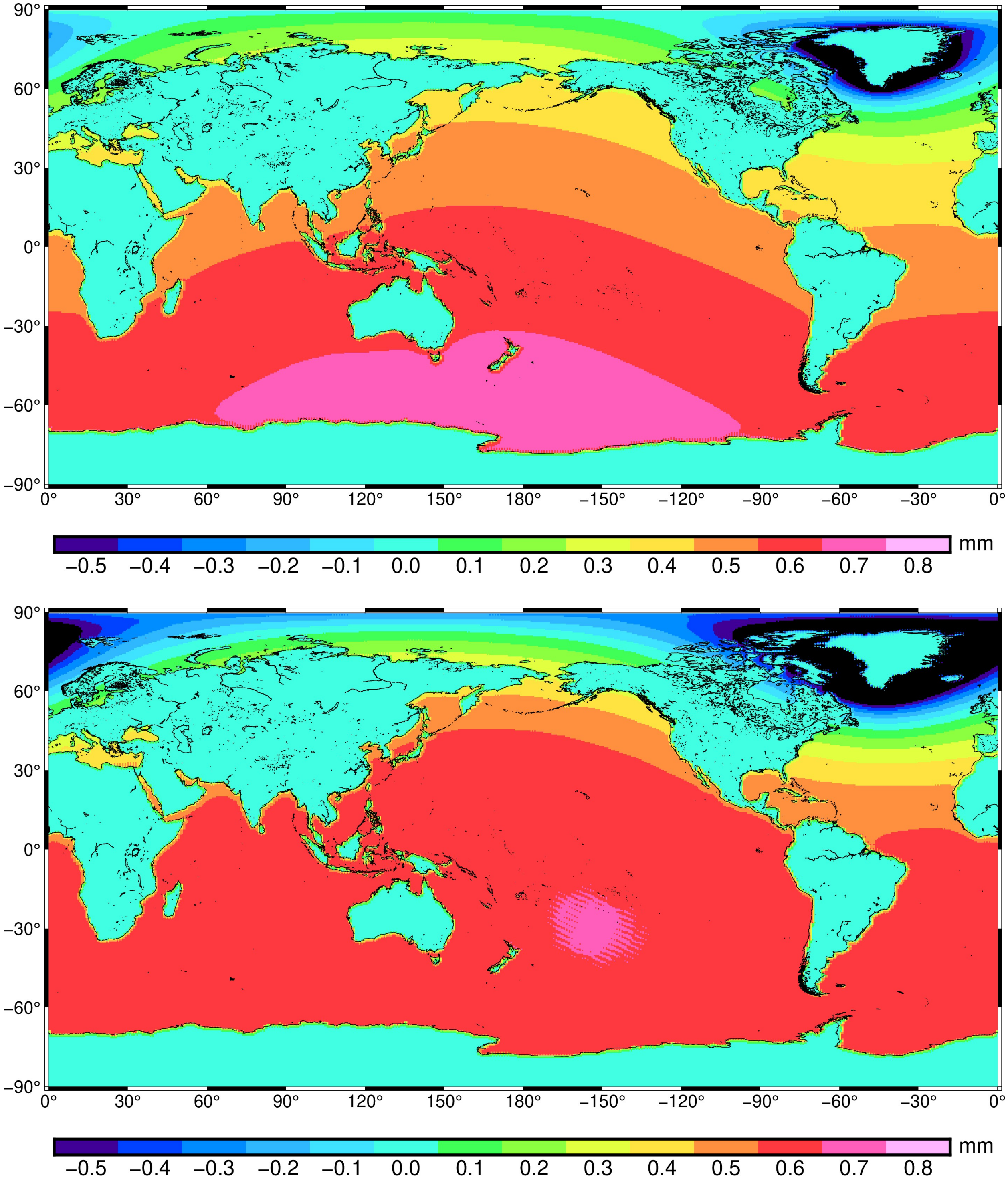

Figure 2.9: Sea level pattern (top: altimetry; bottom: tide gauge) due to water exchange between Greenland and the oceans

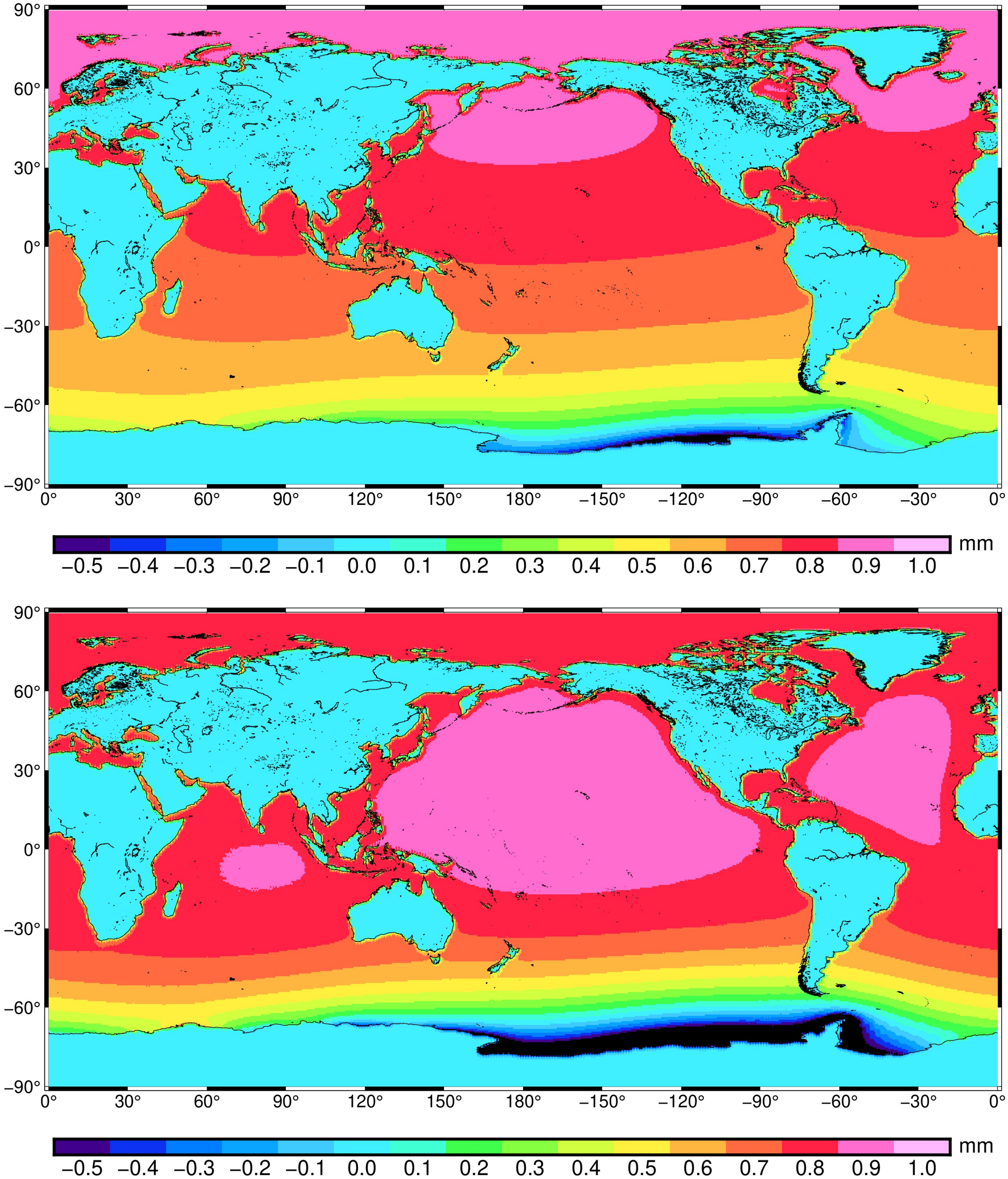

Figure 2.10: Sea level pattern (top: altimetry; bottom: tide gauge) due to water exchange between Antarctica and the oceans

- The tenth line is the file name of load Love numbers, one degree each line starting from degree 0, and in the order of h, l and k.
- The eleventh line is the ocean function.
- The twelfth line is the function of the land source region (1 in the region, 0 outside) considered (so that the input mass could be global).
- The thirteenth line is the file of input data of mass changes. Only the region specified by the 'land source region' is taken as the source.
- The last line is either "Rotational_Feedback_On" or "Rotational_Feedback_Off".

This program outputs a bunch of files with self explanatory names.

An example of `Sea_Level_Equation_Elastic.txt` is

```
GLD
180
1.0e-5
6.371e+6
5.5151e+3
1.0826e-3
3.4498e-3
0.30 0.60 0.084
1.00e+3
lln.txt
OF_360.txt
GLD_0d5_1.txt
Mass_360x360.txt
Rotational_Feedback_On
```

This is to compute the sea level pattern due to a mass change over Greenland. The file `GLD_0d5_1.txt` is the 'Greenland function'.

We show as example the sea level pattern as water exchange between Greenland/Antarctica and the oceans. Fig. 2.8 is the global mass change, where only that over Greenland or Antarctica is taken as the input. The altimetry and tide gauge measured sea levels are shown in Fig. 2.9 and 2.10.

2.9 Difference or sum of a pair of data sets

When using the leakage reduction method, the signal outside the region of interest should be made mush smaller than inside. One way to do so is to remove the signal outside based on a model. Thus there is a need to compute the difference between two data sets. We use the same program to compute both the differences or the sums.

2.9.1 Grid data

The program is `Diff_Sum_Pairs_Grid.cpp`. It requires the `.h` file `check_error.h` for to be compiled.

The input is specified in the file `Diff_Sum_Pairs_Grid.txt`, where

- The first line must be ‘DIFF’ or ‘SUM’. The meaning is evident.
- The second line includes the number of grid points along the latitude and longitude.
- The third line is the number of computation to be done.
- The rest lines includes three file names: two inputs and one output. The difference is the first takes away the second.

An example of the file `Diff_Sum_Pairs_Grid.txt` is

```
SUM
128 128
1
2014_008_63_04_Mass.txt 2014_076_63_04_Mass.txt SUM_Grid.txt
```

2.9.2 SHCs

The program is `Diff_Sum_Pairs_SHCs.cpp`. It requires the `.h` files `check_error.h` and `rshc.h` for to be compiled..

The input is specified in the file `Diff_Sum_Pairs_SHCs.txt`, where

- The first line must be ‘DIFF’ or ‘SUM’. The meaning is evident.
- The second line is the maximum degree and order of the data files.
- The third line is the number of computation to be done.
- The rest lines includes three file names: two inputs and one output. The difference is the first takes away the second.

An example of the file `Diff_Sum_Pairs_Grid.txt` is

```
DIFF
63
1
2014_008_63_03.txt 2014_076_63_03.txt DIFF_SHCs.txt
```

2.9.3 Using year-to-year changes

There are several ways to study mass changes. One particular way is to look how water/ice/snow storage changes from one year to another. We use the program `Average_SHCs.cpp` presented in Sect. 1.6 to compute yearly averages. We can then use this program to compute the year-by-year changes between successive years.

2.10 Extraction and sum of regional data

2.10.1 Extraction

This is to set the grid values outside the region of interest to zero.

The program is `Extract_Region.cpp`. This is a single file program.

The input/output information is specified in the file `Extract_Region.txt`, where

- The first line includes the number of nodes along latitude and longitude.
- The second is the number of computations to be performed.
- Each of the following lines includes three file names. The first is the input data file, the second is the mask (1 within the region, 0 outside), and the third is the result.

An example of `Extract_Region.txt` is

```
128 128
2
2014_008_63_04_Mass.txt  OF_128_gauss.txt  2014_008_63_04_Ocean.txt
2014_076_63_04_Mass.txt  OF_128_gauss.txt  2014_076_63_04_Ocean.txt
```

2.10.2 Sum

Very often, the reginal sum, e.g., the total amount of water storage changes within a water shed, needs to be computed. This program computes the total of one region from a series of data files.

The program is `Sum_Region.cpp`. This is a single file program.

The input/output information is specified in the file `Sum_Region.txt`, where

- The first line gives the radius of the earth.
- The second line gives the numbers of nodes along latitude and longitude.
- The third line gives the file name of region mask (1 within the region, 0 outside).
- The fourth line gives the output file name (all results are outputted to the same file)
- The fifth line gives the number of sums to be performed.
- Each of the following lines includes a file name of input grid data.

An example of `Sum_Region.txt` is

```
6371000.0
128 128
OF_128_gauss.txt
SUM_OCEAM.txt
2
2014_008_63_04_Mass.txt
2014_076_63_04_Mass.txt
```

The output includes three columns: The input file name, the total amount, and the average amount.

The region is defined by the region mask, and it does not have to be extracted in the input data files.

2.11 Posteriori iteration correction – scales regional data by a factor

A Posteriori iteration may be used for further reducing errors caused by the truncation of the spherical harmonic series of the L2 data and the decorrelation. This process may be applied to refine very large mass change estimates within a region.

The process is the following:

1. Compute the leakage reduced mass changes following the decorrelation-smoothing approach.
2. Re-express the mass changes in terms of the Stokes coefficients like GRACE L2 data truncated to the same degree.
3. Re-compute the leakage reduced mass changes using the Stokes coefficients of the last step following the same decorrelation-smoothing approach.
4. The difference between the results of steps (1) and (3) represents the error caused by truncation and decorrelation. However, the result of the step (3) is a smoothed version of that of the step (1). Hence we cannot use the difference to do the correction point by point differently. Let r_1 and r_3 represent total amount of steps (1) and (3) within a small region of interest. The corrected result is then to multiply the results of step (1) by the factor r_1/r_3; this ratio approximate the effect of the procedure on the result.

There is only one additional program for this, i.e., multiply the results within a region by a factor.

The program is `Scale_Region.cpp`. This is a stand alone program.

The input/output information is in the file `Scale_Region.txt`, where

- The first line contains the numbers of nodes along latitude and longitude.
- The second line is the scale factor.
- The third line is the regional mask (1 within, 0 outside).
- The forth line is the number of data files to be rescaled.
- The following are input and output file pairs.

An example of `Scale_Region.txt` is

```
128 128
1.2
OF_128_gauss.txt
2
2014_008_63_04_Mass.txt  2014_008_63_04_Mass_Scaled_Ocean.txt
2014_076_63_04_Mass.txt  2014_076_63_04_Mass_Scaled_Ocean.txt
```

Appendix A

Some utility programs

A.1 Spherical harmonic transform

The relation between the spatial and spectral data is

$$f(\theta, \lambda) = a \sum_{l=0}^{bw-1} \sum_{m=0}^{bw-1} [C_{lm} \cos(m\lambda) + S_{lm} \sin(m\lambda)] \bar{P}_{lm}(\cos\theta)$$

where $\bar{P}_{lm}(\cos\theta)$ is the fully normalized Legendre function. The grid used is ($\theta = 90^\circ - B$)

$$\begin{array}{ll} B = 90 - (j + 1/2) * 180/(2 * bw) \quad \text{for} \quad j = 0, 1, \cdots, 2 * bw - 1 \\ L = k * 360/(2 * bw) \quad \text{for} \quad k = 0, 1, \cdots, k = 2 * bw - 1 \end{array}$$

where bw is called bandwidth.

See the folder `CRUST1.0` for examples of application of these programs.

Forward transform

The forward transform program is `Grid2SHCs.cpp`, which requires the `.cpp` file `rsht.cpp`, the `.c` file `fftpack.c`, and the `.h` files `rsht.h`, `fftpack++.h`, and `fftpack.h` for to be compiled.

The input/output information is specified in the file `Grid2SHCs.txt`, where

- The first line is the value of a.
- The second line is the value of bw.
- The third line is the number of files to be transformed.
- The following lines are file names on input and output in pair.

AN example of `Grid2SHCs.txt` is

```
1.0
180
1
crust1_topo_3.txt  crust1_topo_SHCs_bw180.txt
```

Inverse transform

The inverse transform program is `SHCs2Grid.cpp`, which requires the same files as those by `Grid2SHCs.cpp` for to be compiled.

The input/output information is specified in the file `SHCs2Grid.txt`, which has the same format as that of `Grid2SHCs.txt`. An example is

```
1.0
128
1
crust1_topo_SHCs_bw128.txt crust1_topo_Grid_bw128.txt
```

A.2 Transformation of data between grids

The reference is Guo and SHum (2009).[1] In the discrete spherical harmonic transform used in the transformations among the GRACE SHCs data, the geoidal height, and the surface mass density, we use the grid type

$$B_j = 90 - (j + 1/2) * 180/(2 * bw) \qquad \text{for} \qquad j = 0, 1, \cdots, 2 * bw - 1$$
$$\lambda_k = k * 360/(2 * bw) \qquad \text{for} \qquad k = 0, 1, \cdots, 2 * bw - 1$$

where bw is called bandwidth. The corresponding highest degree and order of the SHCs is $bw - 1$. We have written three programs to transform data between the above grid and different types.

See the folder `CRUST1.0` for examples of application of these programs.

Transformation between two regular latitude-longitude grids

The grid points along the longitude must be in the form of

$$\lambda_k = k * \Delta\lambda \qquad \text{for} \qquad k = 0, ..., k = N - 1,$$

where $\Delta\lambda = 360/N$, and N is the number of grid points. The grid points along the latitude must be one of the following:

- Lobbato type:

  $$B_j = 90 - j * \Delta B \qquad \text{for} \qquad j = 0, 1, \cdots, M - 1;$$

  where $\Delta B = 180/(M - 1)$, and M is the number of grid points.

- Gauss type:

  $$B = 90 - (j + 1/2) * \Delta B \qquad \text{for} \qquad j = 0, 1, \cdots, M - 1.$$

  where $\Delta B = 180/M$, and M is the number of grid points.

The program is `TransformGrids.cpp`, which requires the `.c` file `fftpack.c` and the `.h` files `TransformGrids.h`, `fftpack.h`, `fftpack++.h`, and `rfss.h` for to be compiled.

The input/output information is specified in the file `TransformGrids.txt`, where

[1] Guo, J.Y., Shum, C.K., 2009, Application of the cos-Fourier expansion to data transformation between different latitude-longitude grids, Computers & Geosciences, 35, 1439-1444, doi:10.1016/j.cageo.2008.09.010

- The first line should be ”floating” or ”integral”, meaning the data are floating type ot integral type. If the data are integral type, the computation is done as if they were floating values, but truncated to closest integral values when outputing.
- The second line should be ”Gauss” or ”Lobbato”, meaning the input data are over a Gauss or Lobbato type grid.
- The third line contains two integer values, which are, respectively, the numbers of nodes in latitude and longitue of the grid of input data.
- The fourth line should be ”Gauss” or ”Lobbato”, meaning the output data should be over a Gauss or Lobbato type grid.
- The fifth line contains two integer values, which are, respectively, the numbers of nodes in latitude and longitue of the grid of output data.
- The sixth line is an integer. It is the number of files to be converted.
- In each of the following lines, the first is the file of input data, and the second is the file of output data. Total number of these lines should be equal to the value given in the sixth line.

An example of `TransformGrids.txt` is

```
floating
Gauss
180 360
Gauss
360 360
1
crust1_topo_2.txt crust1_topo_3.txt
```

The data format of input and output files should be in the form

```
 Lat(0) Lon(0) Data(0,0)
 Lat(0) Lon(1) Data(0,1)
 ....
 Lat(1) Lon(0) Data(1,0)
 Lat(1) Lon(1) Data(1,1)
 ....
```

Lat(0),Lat(1),.... should be in descending order, from 90 to -90, and Lon(0),Lon(1),.... in ascending order from 0 to 360. Input data of grid are checked for correctness.

Transformation between two different types of grids over the longitude

A second program, `longitudeShift.cpp`, transforms data between two different types of longitudinal grids,

$$\lambda_k = k * \Delta\lambda \quad \text{for} \quad k = 0, ..., k = N - 1$$

and

$$\lambda_k = (k + 1/2) * \Delta\lambda \quad \text{for} \quad k = 0, ..., k = N - 1,$$

where $\Delta\lambda = 360/N$, and N is the number of grid points. The grid over the latitude is not altered.

The input/output information is specified in the file `longitudeShift.txt`, where

- The first line should be ”rmShift” or ”addShift”. ”rmShift” means ”remove the shifted half interval”, or transform the data from longitude grid of
$$\lambda_k = (k + 1/2) * \Delta\lambda \qquad \text{for} \qquad k = 0, ..., k = N - 1$$
to
$$\lambda_k = k * \Delta\lambda \qquad \text{for} \qquad k = 0, ..., k = N - 1\,.$$
”addShift” does the reverse.
- The second line contains two integers, which are the numbers of nodes in latitude and that in longitude of the grid of input data.
- The third line is an integer. It is the number of files to be converted.
- In each of the following lines, the first is the file of input data, and the second is the file of output data. Total number of these lines should be equal to the value given in the third line.

An example of `longitudeShift.txt` is

```
rmShift
180 360
1
crust1_topo_1.txt  crust1_topo_2.txt
END
```

The format of input and output grid data should be the same as that of the program `TransformGrids.cpp`. Here the values of `Lat(j)` and `Lon(k)` are those for the present data files.

Transformation between two formats of longitude

A third program, `longitudeFormat.cpp`, convert longitudinal formats between $[0, 360)$ and $[-180, 180)$. The program does not depend on any other files.

The input/output information is specified in the file `longitudeFormat.txt`, where

- The first line should be ”from180to360” or ”from360to180”. Their meaning are self-explanative.
- The second line contains two integer values, which are, respectively, the numbers of nodes in latitude and longitude of the grid of input data. The number of longitudinal nodes must be even.
- The third line is an integer. It is the number of files to be converted.
- In each of the following lines, the first is the file of input data, and the second is the file of output data. Total number of these lines should be equal to the value given in the third line.

An example of `longitudeFormat.txt` is

```
from180to360
180 360
1
crust1_topo.txt crust1_topo_1.txt
```

Data format of input and output grid data should be the same as for the program `TransformGrids.cpp`. Here the values of `Lat(j)` and `Lon(k)` are those for the present data files.

Swapping the columns of latitude and longitude

A fourth program, `swapLatLon.cpp`, swaps the first two columns of a data file. This is to swap the columns of latitude and longitude. This is a single file program.

The input/output information is specified in the file `swapLatLon.txt`, where

- The first line is the number of grid points in latitude (the second column of the input file) and that in longitude (the first column of the input file).
- The second line is the number of files to be handled.
- Each of the rest lines are the input and output file names

An example of `swapLatLon.txt` is

```
180 360
1
crust1_topo.xyz crust1_topo.txt
```

A.3 Ocean and land functions

The data files and programs of this group are in the folder `Ocean_Land_Function`.

A.3.1 Generation of ocean function

The reference is Guo and Shum (2009).[2] This program generates the ocean function over any regular grid of constant latitude and longitude intervals using the $5' \times 5'$ resolution ocean function provided by Hans-Peter Plag. The conversion of the original data format to that for our usage is explained in the file `00_RADME.txt` in the folder.

The program is `OF_generate.cpp`, which requires the `.h` file `check_error.h` for to be compiled. The two executables `TransformGrids.exe` and `longitudeShift.exe` should be in the same folder or in the search path.

The input/output information is specified in the file `OF_generate.txt`, where

- The first line should be either ‘Gauss’ or ‘Lobbato’ describing the type of grid over latitude of the finer OF as input.
- The second line gives the numbers of nodes along latitude and longitude of the finer OF used as input.
- The third line is the file name of the finer OF data.
- The fourth line is either ‘Gauss’ or ‘Lobbato’ describing the type of grid over latitude of the output OF.

[2] Guo, J.Y., Shum, C.K., 2009, Application of the cos-Fourier expansion to data transformation between different latitude-longitude grids, Computers & Geosciences, 35, 1439-1444, doi:10.1016/j.cageo.2008.09.010

- The fifth line should be either ‘wShift’ or ‘nShift’, indicating if the longitudinal grid of the output OF should be shifted half the grid interval — ‘wShift’ indicates the longitudinal grid points are $(1/2)\Delta\lambda, (3/2)\Delta\lambda, \cdots$; ‘nShift’ indicates the longitudinal grid points are $0, \Delta\lambda, 2\Delta\lambda, \cdots$.
- The sixth line gives the numbers of nodes along latitude and longitude of the output OF.
- The last line is the file name of output OF.

An example of the file `OF_generate.txt` is

```
Gauss
2160 4320
OF_5m5m_my_format_no_shift.dat
Gauss
nShift
180 360
OF_180_360.txt
```

The input finer OF is not to be modified, which is soecified by the first three lines.

A.3.2 Mask of coastal oceans

The solution of sea level changes from GRACE data is most likely not valid over coastal oceans. Hence, costal oceans need to be masked sometimes.

The program is `Mask_Coastal_Ocean.cpp`, which requires the `.h` files `grid.h`, `mat.h`, `vec.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `Mask_Coastal_Ocean.txt`, where

- The first line must be the string “Gauss ”, meaning that the grid over latitude must be of Gauss type.
- The second line includes the number of nodes along the latitude and that along the longitude.
- The third line is the file name of ocean function data.
- The fourth line is the radius of the earth.
- The fifth line is a value that specifies the distance to coasts to be masked. An ocean grid is masked if the distance between its center and a land grid center is smaller than the specified value.
- The sixth line is a real value to be assigned to the masked grids. If the masked grids are to be considered identical to the land grids, this value should be 0. If the masked grids are to be considered half land half ocean, this value could be assigned as 0.5.
- The last line is the output file name.

An example of `Mask_Coastal_Ocean.txt` is

```
Gauss
360 720
OF_360X720_gauss.txt
6371
400
0.5
OF_360X720_gauss_400.txt
```

A.3.3 Generation of land function from ocean function

The program is in the file `OF_LF_Switch.cpp`. This program actually switches between land and ocean functions. It is a stand alone program.

The input/output information is specified in the file `OF_LF_Switch.txt`, where

- The first line includes the number of nodes along latitude and that along the longitude.
- The second line includes the input and output file names.

An example of `OF_LF_Switch.txt` is

```
360 720
OF_360X720_gauss.txt  LF_360X720_gauss.txt
```

A.3.4 Divide pieces of land separated by oceans

This program uses as input the ocean function (1 over ocean, 0 over land), and output the land function with different values assigned to different piece of land separated by the oceans: 0 over the oceans, and the pieces of land are numbered 1, 2, 3, $\cdots$.

The program is `land_divider.cpp`, which requires the `.h` files `grid.h`, `mat.h`, and `check_error.h` for to be compiled.

The input/output information is specified in the file `land_divider.txt`, where

- The first line is the a criteria to separate land: 0 if two land grids with a common edge are considered to belong to the same piece of land, a positive integer if two land grids with a common edge or a common vertex are considered to belong to the same piece of land.
- The second line includes the numbers of grid points along latitude and longitude.
- The third line is the grid type over latitude, either 'Gauss' or 'Lobbato'.
- The last line includes the input file name of ocean function and the output file name of land function.

An example of `land_divider.txt` is

```
1
360 720
Gauss
OF_360X720_gauss.txt LF_DIV_360X720_gauss.txt
```

A.3.5 Extraction of pieces of land separated by ocean

The program uses as input the output of the program `land_divider.cpp`, and generates land functions of selected pieces of lands, i.e., 1 over selected pieces of land, and 0 elsewhere.

The program is `land_pieces.cpp`, which requires the `.h` files `grid.h`, `mat.h`, `vec.h`, and `check_error.h` for to be compiled.

The the input/output information is specified in the file `land_pieces.txt`, where

- The first line include numbers of nodes along latitude and longitude.
- The second line includes the type of grid along latitude: ‘Gauss’ or ‘Lobbato’.
- The third line is the input data file name.
- The fourth line is the number of pieces of land to be included in the output.
- The following ‘number of pieces of land to be included in the output’ are the numbers in the output of the program `land_divider.cpp`, i.e., the input of this program, to represent these pieces of land.
- The last line is the output file name.

An example of `land_pieces.txt` is

```
360 720
Gauss
LF_DIV_360X720_gauss.txt
1
1
Antarctica_360X720.txt
```

A.4 Smoothing over land and ocean separately

Although this kind of smoothing filter is not to be applied to GRACE L2 data, it may be useful for other kind of data like land hydrology or satellite altimetry determined sea level. Hence we put these programs in the appendix. The reference is Guo et al. (2010)[3]

Isotropic Gaussian

The program is `Filter_Land_Ocean_Gauss_Iso.cpp`. It requires `grid.h`, `grid_conv.h`, `rfss.h`, `fftpack.h`, `fftpack++.h` and `fftpack.c` for to be compiled.

The input/output information is assigned in the file `Filter_Land_Ocean_Gauss_Iso.txt`, where

- The first line is the file name of land or ocean function. It could be the land function with different values assigned to different continent, so that the filter is applied to different land separately.
- The second line gives the numbers of nodes along latitude and longitude.
- The third line is the smoothing radius in terms of geocentric angle.
- The fourth line is the number of interpolated nodes (1 for no additional nodes). The values must be odd (3 or 5 normally).
- The fifth line is the number of data files the filter is supposed to apply.
- The following lines are input and output data files.

An example of `Filter_Land_Ocean_Gauss_Iso.txt` is

[3] Guo, J.Y., Duan, X.J. and Shum, C.K., 2010. Non-isotropic Gaussian smoothing and leakage reduction for determining mass changes over land and ocean using GRACE data, Geophys. J. Int., 181, 290-302, doi:10.1111/j.1365-246X.2010.04534.x

```
land_360_360.txt
360 360
3.5
3
1
grd1f.txt  grd1f_flt.txt
```

Non-isotropic Gaussian

The program is `Filter_Land_Ocean_Gauss_Non_Iso.cpp`. It requires `grid.h`, `grid_conv.h`, `rfss.h`, `fftpack.h`, `fftpack++.h` and `fftpack.c` for to be compiled.

The input/output information is assigned in the file `Filter_Land_Ocean_Gauss_Non_Iso.txt`, where

- The first line is the file name of land or ocean function. It could be the land function with different values assigned to different continent, so that the filter is applied to different land separately.
- The second line gives the numbers of nodes along latitude and longitude.
- The third line should be "nW" or "wW" to indicate if the factor $\sin\theta$ is included.
- The fourth line gives the smoothing radius along latitude and longitude.
- The fifth line is the number of interpolated nodes (1 for no additional nodes). The values must be odd (3 or 5 normally).
- The sixth line is the number of data files the filter is supposed to apply.
- The following lines are input and output data files.

An example of `Filter_Land_Ocean_Gauss_Non_Iso.txt` is

```
LAND_121_242.txt
121 242
nW
3 6
3
2
trend_121_242_mass_1.txt   trend_121_242_mass_1_logg.txt
trend_121_242_mass_2.txt   trend_121_242_mass_2_logg.txt
```

Mixed Gaussian

The program `Filter_Land_Ocean_Gauss_Mix.cpp`, and the input data are specified in the file `Filter_Land_Ocean_Gauss_Mix.txt`. Every other thing is the same as the case of non-isotropic smoothing.