

Conversion and use of Spherical Harmonic Coefficients Generated with Different Normalization Conventions

Gareth G. Roberts
gareth.roberts@imperial.ac.uk

1 Introduction

I am interested in comparing observations and predictions of surface deflections that arise from mantle convection in the spherical harmonic domain. This document makes use of work presented in Wieczorek and Meschede [2018], Ghelichkhan et al. [2021], Forte and Rowley [2022], O’Malley et al. [2024], Davies et al. [2025].

Deflections of the surface of a sphere (e.g. the Earth) can be described as a function, f , of longitude, θ , and latitude, ϕ , using a linear combination of spherical harmonics of degree l and order m ,

$$f(\theta, \phi) = \sum_{l=1}^L \sum_{m=-l}^l f_{lm} Y_{lm}(\theta, \phi), \quad (1)$$

where f_{lm} are the spherical harmonic coefficients and Y_{lm} are the spherical harmonic basis functions, and L is the maximum degree being considered. A variety of normalization conventions are used to define Y_{lm} in the geosciences (e.g. 4π -normalized, fully complex harmonics, with and without phase conventions). Consequently, the values of the coefficients, calculated from spherical harmonic transformation or by fitting spherical harmonics to observations of $f(\theta, \phi)$ for instance, depend on the normalization convention used [see e.g. Equation 30 in Wieczorek and Meschede, 2018, Hoggard et al., 2016]. Wieczorek and Meschede [2018] and their `pyshtools` package provide useful introductions to spherical harmonics and some of the normalizations used in the geosciences, as well as their conversions.

This document first describes the extraction of spherical harmonic coefficients from the mantle convection code `TERRA` and their use in the propagator matrix code `propmat` [see e.g. Ghelichkhan et al., 2021, Davies et al., 2025]. Once numerical simulations of mantle convection have been run using `TERRA`, various parameters (e.g. stress, density) at specific radii are available for the entire sphere (e.g. at the surface, core-mantle boundary, or at other radii). They can be expanded into the spherical harmonic domain using the `de_sph_write()` and `plmbar` subroutines within `TERRA`. `TERRA` can then be used to output the spherical harmonic coefficients, f_{lm} , which can be used to generate the function, $f(\theta, \phi)$, within `propmat`. The `propmat` code is designed to use spherical harmonic expansions of density, $\rho_{lm}(r)$, at the given radii, r , to calculate surface deflections [see e.g. Ghelichkhan et al., 2021, O’Malley et al., 2024, and references therein for details]. Secondly, conversion of spherical harmonic coefficients in the complex form used by Forte and Rowley [2022] for use in `propmat` is described.

This document is principally concerned with conversion of spherical harmonic coefficients to ensure that the different spherical harmonic normalizations and coefficients used by `TERRA`, `propmat` and by Forte and

Rowley [2022] yield consistent results for a specific estimate of global dynamic topography at Earth’s surface, $f(\theta, \phi)$.

2 Spherical harmonic functions in TERRA

The subroutines `plmbar` calculates the spherical harmonic normalization such that

$$Y_{lm} = \sqrt{(2 - \delta_{0,m})(2l + 1) \frac{(l - m)!}{(l + m)!}} P_{lm} \text{trig}(m\phi), \quad (2)$$

where $\delta_{0,m}$ is the Kronecker delta, i.e. $\delta_{0,m} = 1$ when $m = 0$, and $\delta_{0,m} = 0$ when $m \neq 0$. P_{lm} are the associated Legendre functions, and `trig` is either `sin` or `cos`. Note that this normalization does not include the Condon-Shortley phase: $(-1)^m$.

3 Spherical harmonic functions in propmat

The spherical harmonic coefficients (f_{lm}) exported from TERRA are first read into `propmat` by `util.f90` where they are multiplied by $\sqrt{4\pi}$ (see Appendix A). Note that in `calc.f`, when $m \neq 0$ the calculated coefficients contain `smpi/sqr(2)`, where `smpi`= $1/\sqrt{\pi}$, thus when multiplied by $\sqrt{4\pi}$ (in `util.f90`), these constants combine to give $\sqrt{2}$. When $m = 0$, the $0.5 * \text{smpi}$ term within the normalization simplifies to 1 when multiplied by $\sqrt{4\pi}$. The Condon-Shortley phase is not included. Consequently, the spherical harmonic functions are 4π -normalized [see e.g. Wieczorek and Meschede, 2018],

$$Y_{lm} = \begin{cases} \sqrt{2(2l + 1) \frac{(1+m)!}{(1-m)!}} P_{lm} \cos(m\phi) & \text{if } m < 0, \\ \sqrt{(2l + 1)} P_{lm} & \text{if } m = 0, \\ \sqrt{2(2l + 1) \frac{(1-m)!}{(1+m)!}} P_{lm} \sin(m\phi) & \text{if } m > 0. \end{cases} \quad (3)$$

Thus the spherical harmonic functions used by TERRA and `propmat` are consistent.

As an example consider Figure 1 in which panel (a) shows the `./degree_1_to_40.grd` from Holdt et al. [2022]’s Supporting Information, which is the target here. Their grid was first converted into (evenly sampled) latitude, longitude, deflection (x, y, z) coordinates using `gmt grd2xyz`. The spherical harmonic coefficients were then estimated by inverting that suite of x, y, z values using the (unweighted) least squares approach in `pyshtools v4.13` [Wieczorek and Meschede, 2018]:

```
holdt = pygmt.load_dataarray('degree_1_to_40.grd')
holdt_xyz = pygmt.grd2xyz(grid=holdt, output_type='pandas')
fname=holdt_xyz
lon, lat, val = fname["lon"], fname["lat"], fname["z"]
holdt_coeffs=psh.SHCoeffs.from_least_squares(val, lat, lon, lmax)
! the spherical harmonics are expanded in the usual way
holdt_grid = holdt_coeffs.expand(grid='DH2')
```

Panel (b) shows the resultant spherical harmonics generated from the coefficients calculated by inversion of those regularly sampled points (black dots). Panel (c) shows the spherical harmonics calculated by inserting the calculated coefficients into `gmt sph2grd`. Panel (d) shows the resultant grid once the coefficients used in plotting of (b) and (c) are put throughout the `propmat` code (with the additional multiplicative factor of $\sqrt{4\pi}$ removed). In this example the spherical harmonic coefficients are used to define the reference dynamic topography within `propmat`: `./CODE/SRC/ref flds.f90`, and the outputted coefficients, stored in `SPH_REF_dyntopography`, are used in the plotting (additional details, for other ‘reference’ dynamic topographies, follow below). Panel (e) shows the results from an alternative approach to generating the spherical harmonic coefficients (courtesy S. Ghelichkhan). In this case, Holdt et al. [2022]’s scattered data are first interpolated onto an evenly spaced global grid and then expanded into the spherical harmonic domain using `psh tools`. Panels a–d demonstrate the consistency in grids generated by the various (`propmat`, `gmt`, `psh tools`) methodologies.

4 Complex spherical harmonics used by Forte & Rowley (2022)

The spherical harmonic coefficients for the dynamic topography in Forte and Rowley [2022] provided by Alessandro Forte are complex and the file containing them (`dyn-surf-FR2021-L100.1m`) is formatted such that each row contains:

$$l \ m \ \Re \ \Im$$

where \Re and \Im are the real and imaginary parts of the harmonic coefficients, in order of increasing l and m , $L = 100$ and $m \geq 0$. The spherical harmonic basis function used by Forte and Rowley [2022] is

$$Y_l^m(\theta, \phi) = (-1)^m \sqrt{(2l+1) \frac{(l-m)!}{(l+m)!}} P_l^m e^{im\phi}. \quad (4)$$

Note that in this instance the normalization includes the Condon-Shortley phase. I am using mixed (upper and lower) subscripts to differentiate real from complex coefficients and functions [*sensu* Wieczorek and Meschede, 2018].

The following code snippets summarize the conversion of the real and imaginary coefficients into coefficients for use in `propmat` (i.e. conversion to real 4π -normalization), they make use of simple bash scripts and `psh tools` libraries (in Python):

Conversion of complex coefficients

The complex spherical harmonic coefficients are first converted into real form. Following Wieczorek and Meschede [2018]

$$f_l^m = \begin{cases} (-1)^m f_l^{-m*} & \text{if } m < 0, \\ f_{l0} & \text{if } m = 0, \\ (f_{lm} - i f_{l-m})/\sqrt{2} & \text{if } m > 0, \end{cases} \quad (5)$$

where $*$ indicate complex conjugation. The conversion is performed with a simple awk script (note that $m \geq 0$ in `dyn-surf-FR2021-L100.1m`):

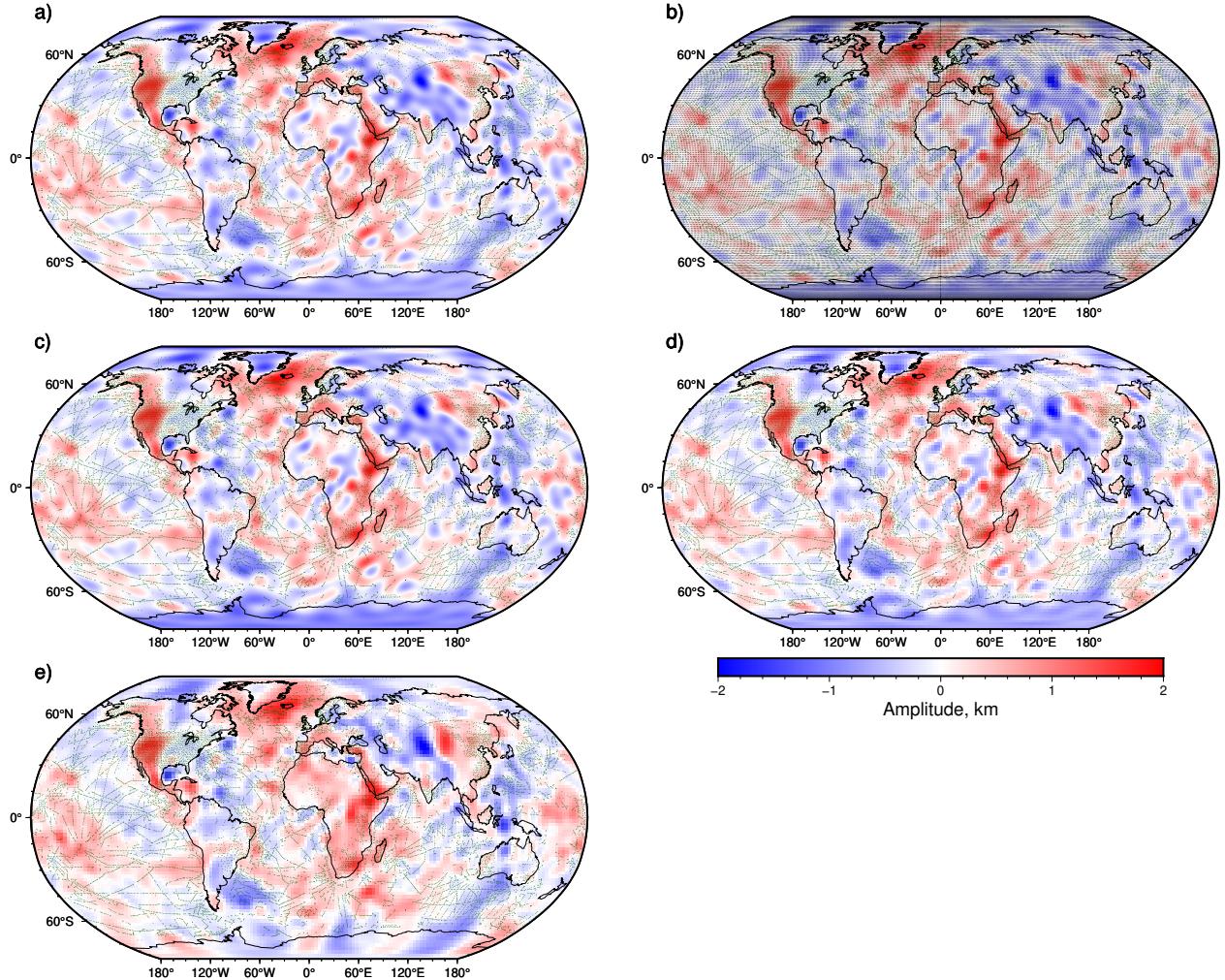


Figure 1: Comparisons of dynamic topography grids generated from Holdt et al. [2022]'s scattered points (green dots) and even sampling of their `./degree_1_to_40.grd` (black dots). (a) Holdt et al. [2022]'s grid (the target). (b) Results from inversion of regularly sampled points. (c) Output from gmt `sph2grd` using the spherical harmonic coefficients used to produce the grid shown in panel b. (d) Resultant grid once the coefficients used in plotting of (b) and (c) are put throughout the `propmat` code (with the additional multiplicative factor of $\sqrt{4\pi}$ removed). (e) Grid that results from the interpolation and expansion of Holdt et al. [2022]'s scattered data (courtesy S. Ghelichkhan).

```

forte=dyn-surf-FR2021-L100.lm
awk '{print NR, $0}' < $forte > temp
awk '{if ($3==0) print $0}' < temp > temp1
awk '{if ($3!=0) printf "%d %d %d %.14f %.14f\n ", $1, $2, $3, sqrt(2) *
    ↪ $4, sqrt(2) * -1 * $5}' < temp > temp2
cat temp1 temp2 | sort -n -k1,1 | awk '{print $2, $3, $4, $5}' > recast.
    ↪ sph

```

Note the inclusion of the formatting statement to ensure that the coefficients are output with 14 decimal places (consistent with `dyn-surf-FR2021-L100.lm`; otherwise awk can truncate them to a few d.p., reducing the accuracy of the calculated fields). Thus the file `recast.sph` is formatted such that rows with $m = 0$ contain: $l \ m \ \Re \Im$, and rows with $m > 0$ contain: $l \ m \ \Re\sqrt{2} \ -\Im\sqrt{2}$. These adjusted coefficients are then inserted into `pyshtools` with the following command:

```

coeffs_global = pysh.SHCoeffs.from_file('recast.sph', normalization='4pi',
    ↪ cphase = -1, lmax=100, format='shtools')

```

Note that the Condon-Shortley phase is included.

Spherical harmonic expansion

It is straightforward to use `pyshtools` to perform the spherical harmonic expansion and construct grids from the spherical harmonic coefficients:

```
grid_global = coeffs_global.expand(grid='DH2')
```

Here, the resultant grid conforms with the sampling theorem of Driscoll and Healy [1994].

Reducing L and conversion to `propmat` format

It is straightforward to use `pyshtools` to only include the spherical harmonics of the first L degrees:

```

coeffs_global_lmax50 = pysh.SHCoeffs.from_file('recast.sph', normalization
    ↪ ='4pi', cphase = -1, lmax=50, format='shtools')

```

and to convert the spherical harmonic coefficients into the form used by `propmat`:

```

new_coeffs = coeffs_global_lmax50.convert(cphase=1, lmax=50,
    ↪ normalization='4pi') * np.sqrt(4*np.pi)

```

note the removal of the Condon-Shortley phases, and the multiplicative $\sqrt{4\pi}$ term, which is included because the `ref flds.f90` in `propmat` does not include this factor when reading in coefficients for reference dynamic topography (at Earth's surface) and hence the `normalization` (and resultant grids and spherical harmonic coefficients output from `propmat`) would be a factor of $\sqrt{4\pi}$ too low with out it.

Writing output

The coefficients can then be written to a file for insertion into `propmat`:

```

new_coeffs_arr = pysh.SHCoeffs.to_array(new_coeffs)
pysh.shio.shwrite(filename='for_sia_code.sph', coeffs=new_coeffs_arr, lmax
    ↪ =50)
cmd = " awk -F',' '{print $3, $4}' < for_sia_code.sph > dyn-surf-FR2021-
    ↪ L100_real_ggr.lm "
output = sb.check_output(cmd, stderr=sb.STDOUT, shell=True)
sys.stdout.write('{}\n'.format(output))

```

At this stage, the grids can also be converted into an ascii file containing latitude, longitude and amplitude of surface deflections using gmt:

```
new_grid = pygmt.grdsample(grid=grid_global.to_xarray(), spacing=[0.5, 0.5],
    ↪ interpolation='n')
pygmt.grd2xyz(grid=new_grid, output_type='file', outfile='./fortegrd_FR21.
    ↪ xyz')
coeffs_global = pysh.SHCoeffs.from_file('SPH_REF_dyntopography_cut',
    ↪ normalization='4pi', cphase = 1, lmax=50, format='shtools') / np.
    ↪ sqrt(4*np.pi)
coeffs_global.info()
grid_global = coeffs_global.expand(grid='DH2')
```

which can be straightforwardly compared to the original dynamic topography grid of Forte and Rowley [2022] for instance. This step can be useful for comparing grids generated with different spherical harmonic content (e.g. different maximum degrees, L). However, note that the interpolation and gridding can result in the grids not being true representations of the spherical harmonic coefficients. A safer option for generating the grids, making use of `xarray`, is:

```
coeffs = pysh.SHCoeffs.from_file('input_coeffs_file', normalization=
    ↪ normalization, cphase = cphase, lmax=lmax, format='shtools')
grid = coeffs.expand(grid='DH2')
grid_xarr = grid.to_xarray()
grid_xarr.to_netcdf(path='./grid_xarr.nc')
pygmt.grd2xyz(grid='./grid_xarr.nc', output_type='file', outfile='./
    ↪ grid_xarr.xyz')
```

examples of doing so follow below (see Figure 2).

Use of converted coefficients in propmat

The converted coefficients can be imported into `propmat`, as, for instance, a new reference field in the directory `RefGeo`. Note that this new reference field must be declared in `./CODE/SRC/ref flds.f90`, and the code recompiled, which is the default behavior in my run scripts.

Use of coefficients generated by propmat in pyshtools

Once `propmat` has been run, the resultant spherical harmonic coefficients for the reference field are stored as `SPH_REF_dyntopography`, which can then be re-normalized to e.g. 4π normalization, and used to produce a x, y, z file for comparison with the original data provided by Forte and Rowley [2022]:

```
cmd = " awk '{if (NR>4) print $2, $3, $4, $5}' < SPH_REF_dyntopography >
    ↪ SPH_REF_dyntopography_cut " ! remove header
output = sb.check_output(cmd, stderr=sb.STDOUT, shell=True)
sys.stdout.write('{}\n'.format(output))

coeffs_global = pysh.SHCoeffs.from_file('SPH_REF_dyntopography_cut',
    ↪ normalization='4pi', cphase = 1, lmax=50, format='shtools') / np.
    ↪ sqrt(4*np.pi)
coeffs_global.info()
grid_global = coeffs_global.expand(grid='DH2')
sia = grid_global.to_xarray()
sia.to_netcdf(path='./sia.nc')
```

```
pygmt.grd2xyz(grid='./sia.nc', output_type='file', outfile='./
    ↪ SPH_REF_dyntopography.xyz')
```

Note the division by $\sqrt{4\pi}$, undoing the earlier multiplication for insertion into `propmat` (which does not include this factor in its normalization for the reference dynamic topography field).

Comparison of dynamic topographies

Figure 2 shows maps and histograms of surface deflections from grids (`forte.nc` & `sia.nc`) of dynamic topography produced following conversion of the complex spherical harmonic coefficients provided by Forte & Rowley (2022: `dyn-surf-FR2021-L100.1m` → `recast.sph`) into real-valued coefficients (still with the Condon-Shortley phase and 4π -normalization):

```
coeffs_global = pysh.SHCoeffs.from_file('recast.sph', normalization='4
    ↪ pi', csphase = -1, lmax=100, format='shtools')
coeffs_global.info()
grid_global = coeffs_global.expand(grid='DH2')
forte = grid_global.to_xarray()
forte.to_netcdf(path='./forte.nc')
pygmt.grd2xyz(grid='./forte.nc', output_type='file', outfile='./
    ↪ fortegrd_FR21.xyz')
```

and using output from the `propmat` code:

```
coeffs_global = pysh.SHCoeffs.from_file('SPH_REF_dyntopography_cut',
    ↪ normalization='4pi', csphase = 1, lmax=100, format='shtools') / np.
    ↪ sqrt(4*np.pi)
coeffs_global.info()
grid_global = coeffs_global.expand(grid='DH2')
sia = grid_global.to_xarray()
sia.to_netcdf(path='./sia.nc')
pygmt.grd2xyz(grid='./sia.nc', output_type='file', outfile='./
    ↪ SPH_REF_dyntopography.xyz')
```

note the lack of Condon-Shortley phase and 4π -normalization. It is encouraging that the maps and histograms of elevations are very similar, and that the map of differences show no obvious systematic patterns (suggested that we are not missing information at specific lm). The maximum difference elevations is < 0.18 m, which is attributed to the truncation of decimal places in output from `propmat`. Figure 3 shows the associated spherical harmonic coefficient values, their residuals and a 1:1 comparison of gridded dynamic topographies.

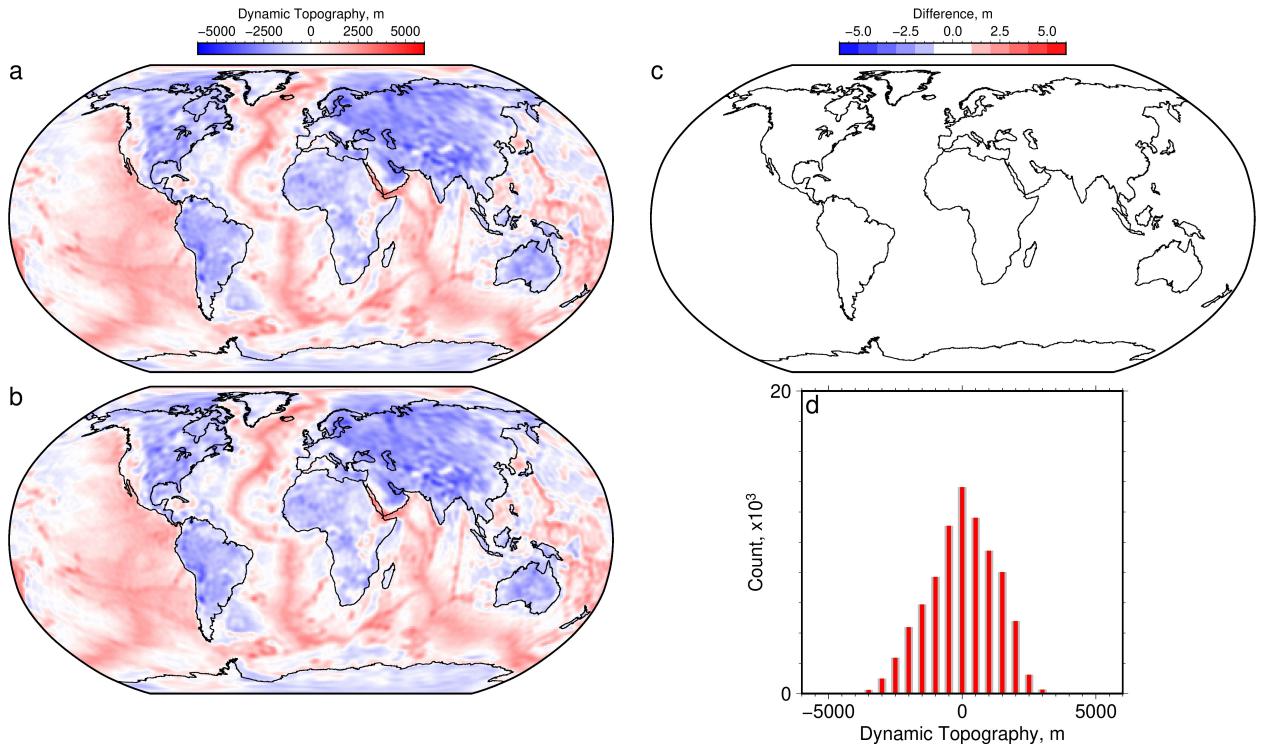


Figure 2: Comparison of surface deflections in dynamic topography grids from Forte and Rowley [2022] (a) and those generated following conversion of their complex spherical harmonic coefficients ($L = 100$) to real 4π -normalization using `pyshtools` and insertion into `propmat` (b), and their difference (c; a-b). (d) Grey = count of deflections produced by first converting the complex coefficients provided by Forte & Rowley (2022: `dyn-surf-FR2021-L100.1m`; panel a) into real form. Red = results from `propmat` (`SPH_REF_dyntopography.xyz`; panel b; see body text for details).

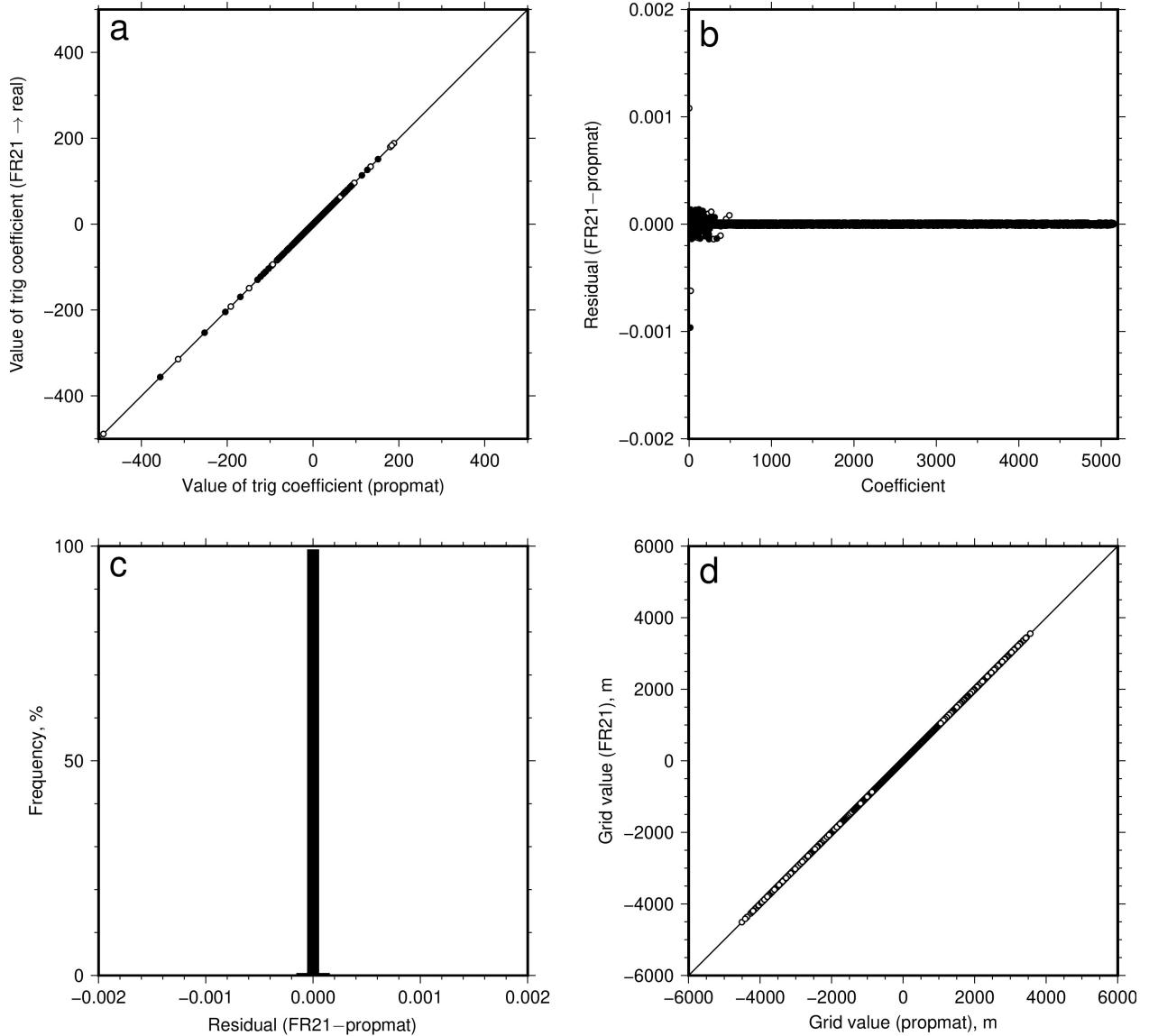


Figure 3: Comparison of coefficients and grid values from Forte and Rowley [2022] once they have been converted into real form with those exported from `propmat`. (a) Comparison of coefficient values: white/black filled circles = sine/cosine ('trig') components. (b-c) Coefficient component residuals: subtracting trig values exported from `propmat` from the real formatted values of Forte and Rowley [2022]. (d) Comparison of dynamic topography values (expanded spherical harmonics) on the grids generated with the real valued Forte and Rowley [2022] coefficients and those exported from `propmat`. Solid lines = 1:1.

References

- M. A. Wieczorek and M. Meschede. SHTools: Tools for Working with Spherical Harmonics. *Geochemistry, Geophysics, Geosystems*, 19:1–19, 2018. doi: 10.1029/2018GC007529.
- S. Ghelichkhan, H.-P. Bunge, and J. Oeser. Global mantle flow retrodictions for the early Cenozoic using an adjoint method: Evolving dynamic topographies, deep mantle structures, flow trajectories and sub-lithospheric stresses. *Geophysical Journal International*, 226(2):1432–1460, 2021. ISSN 1365246X. doi: 10.1093/gji/ggab108.
- A. M. Forte and D. B. Rowley. Earth’s Isostatic and Dynamic Topography—A Critical Perspective. *Geochemistry, Geophysics, Geosystems*, 23(e2021GC009740):1–48, 2022. ISSN 1525-2027. doi: 10.1029/2021gc009740.
- C. P. B. O’Malley, G. G. Roberts, J. Panton, F. D. Richards, J. H. Davies, V. M. Fernandes, and S. Ghelichkhan. Reconciling surface deflections from simulations of global mantle convection. *Geoscientific Model Development*, 17(24):9023–9049, 2024. doi: 10.5194/gmd-17-9023-2024. URL <https://gmd.copernicus.org/articles/17/9023/2024/>.
- J. H. Davies, J. Panton, I. Altoe, M. Andersen, P. Béguelin, A. Biggin, C. Davies, T. Elliott, Y. A. Engbers, V. M. Fernandes, A. M. G. Ferreira, S. Fowler, S. Ghelichkhan, B. J. Heinen, P. Koelmanijer, F. Latallerie, W. Li, G. Morgan, S. J. Mason, R. Myhill, A. Nowacki, C. P. O’Malley, A. Plummer, D. Porcelli, N. Récalde, G. G. Roberts, J. B. Rodney, J. Shea, O. Shorttle, W. Sturgeon, A. M. Walker, J. Ward, and J. Wookey. How to assess similarities and differences between mantle circulation models and earth using disparate independent observations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2315):20240827, 2025. doi: 10.1098/rspa.2024.0827. URL <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2024.0827>.
- M. J. Hoggard, N. White, and D. Al-Attar. Global dynamic topography observations reveal limited influence of large-scale mantle flow. *Nature Geoscience*, 9(May):1–8, 2016. ISSN 1752-0894. doi: 10.1038/ngeo2709.
- M. C. Holdt, N. J. White, S. N. Stephenson, and B. W. Conway-Jones. Densely Sampled Global Dynamic Topographic Observations and Their Significance. *Journal of Geophysical Research: Solid Earth*, 127: 1–32, 2022.
- J. R. Driscoll and D. M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2):202–250, 1994. ISSN 0196-8858. doi: <https://doi.org/10.1006/aama.1994.1008>. URL <https://www.sciencedirect.com/science/article/pii/S0196885884710086>.

Appendix A: propmat code snippets

The code snippet (lines 17–52) from `util.f90` in which the spherical harmonic coefficients are first read into `propmat` as `anom` follows. Note `*sqrt` at the bottom, where $\text{sqrt} = \sqrt{4\pi}$ (see line 14 of `constants.h`):

```

open(001, file= trim(input), status='old',
.
.
.
! initialize all coeffs with zero
anom(:,:,:) = 0.0
do ir = nr+1,1,-1
    !raed in header
    read(001,'(A)') line_temp
    do l=0,l_max
        do m=0,l
            read(001,*) test1, test2
            if (l <= l_max_calc) then
                test_anom(l+1,l_max_calc+1+m,ir) = test2
                test_anom(l+1,l_max_calc+1-m,ir) = test1
            endif
        enddo
    enddo
    if ((r(nr+1)-r(ir))<(top_depth*1000) .or. &
&      ( r(ir)-r( 1))<(bot_depth*1000) )&
&      test_anom(:,:,:ir)=0.0
enddo
anom = test_anom
close(001)
anom(:,:,:,:) = anom(:,:,:,:)*sqrt

```

The code snippet from `calc.f` in which `anom` is used and the normalizations are calculated on lines 79–118 follows. Note that $\text{sqpi} = 1/\sqrt{\pi}$ (see line 10 in `constants.h`) and `facto(:)=sqrt(facto(:))` (see line 52 of `sph_tools.f90`):

```

do l=l_min, l_max_calc
    do m= -1, 1
        if (mod(proc_caller,nproc) == myproc) then
            do j=nr+1,1,-1
                geoidCO(l,m) =
&                  geoidCO(l,m) + anom(l,m,j)*GeoK(l,j)*laysize
                gravCO (l,m) =
&                  gravCO (l,m) + anom(l,m,j)*GravK(l,j)*laysize
                veloCO(l,m) =
&                  veloCO(l,m) + anom(l,m,j)*VelK(l,j)*laysize
                SurTopCO(l,m)=
&                  SurTopCO(l,m)+ anom(l,m,j)*SurK(l,j)*laysize
                CMBTopCO(l,m)=
&                  CMBTopCO(l,m)+ anom(l,m,j)*CMBK(l,j)*laysize
            enddo
            do i=1,grid
                do j=1,grid2
                    if(m==0) then

```

```

        sphharm = sqr(2*l+1)*0.5*sqpi*leg(l,m,i)
    elseif (m<0) then
        sphharm = sqr(2*l+1)*sqpi/sqr(2)*facto(l+m)/facto(l-m)
&                                *leg(l,-m,i)*cos(m*phi(j))
    else
        sphharm = sqr(2*l+1)*sqpi/sqr(2)*facto(l-m)/facto(l+m)
&                                *leg(l,m,i)*sin(m*phi(j))
    endif
    geo_grd (i,j) = geo_grd (i,j) + geoidCO (l,m)*sphharm
    grv_grd (i,j) = grv_grd (i,j) + gravCO (l,m)*sphharm
    srfvl_grd(i,j) = srfvl_grd(i,j) + veloCO (l,m)*sphharm
    srfdt_grd(i,j) = srfdt_grd(i,j) + SurTopCO(l,m)*sphharm
    cmbdt_grd(i,j) = cmbdt_grd(i,j) + CMBTopCO(l,m)*sphharm
    georf_grd(i,j) = georf_grd(i,j) + georfCO (l,m)*sphharm
    grvrf_grd(i,j) = grvrf_grd(i,j) + grvrfCO (l,m)*sphharm
    dynrf_grd(i,j) = dynrf_grd(i,j) + dynrfCO (l,m)*sphharm
    cmbrf_grd(i,j) = cmbrf_grd(i,j) + cmbrfCO (l,m)*sphharm
    enddo
    enddo
    endif
    proc_caller = proc_caller +1
enddo

```