HOWTO g06: Synthesis of planetary topographies

You will learn how to synthesize a planetary topography. In fact, the same approach can be applied to any surface spherical harmonic synthesis of the form

$$f(\varphi,\lambda)=\sum_{n=0}^{n_{\max}}\sum_{m=0}^{n}\left(\bar{C}_{nm}\cos(m\lambda)+\bar{S}_{nm}\sin(m\lambda)\right)\bar{P}_{nm}(\sin\varphi),$$

where $\bar{C}_{nm}$ and $\bar{S}_{nm}$ are 4π-fully-normalized (real) surface spherical harmonic coefficients of the function f, n and m are spherical harmonic degree and order, respectively, $\bar{P}_{nm}(\sin\varphi)$ are the 4π-fully-normalized (real) associated Legendre functions of the first-kind, and, finally, φ and λ are the spherical latitude and longitude, respectively. This means GrafLab can synthesize a wide range of (real) functions given on a sphere.

All the GrafLab input parameters are explained in ../docs/graflab.md.

```
clear; clc; init_checker();
```

Synthesis of the Earth's topography in spherical coordinates

We need to perform the basic *surface* spherical harmonic synthesis shown above. This can be achieved with the *surface* synthesis of the gravitational potential, see the equation for "V" in https://github.com/blazej-bucha/graflab/blob/master/docs/Definition_of_functionals_of_the_geopotential_used_in_GrafLab_software.pdf.

The trick is that we have to set "GM = 1.0", "R = 1.0" and the radius of the evaluation points to "r = 1.0". Obviously, we have to do the synthesis on the unit sphere, so "crd = 1". Finally, we set "quantity" to "11" (see ../docs/graflab.md).

Define the GrafLab inputs.

```
GM                = 1.0;  % Important
R                 = 1.0;  % Important
nmin              = 0;
nmax              = 360;
ellipsoid         = 1;
GGM_path          = '../data/input/DTM2006.mat';
crd               = 1;  % Important
point_type        = 0;
lat_grd_min       = -90.0;
lat_grd_step      =   1.0;
lat_grd_max       =  90.0;
lon_grd_min       =   0.0;
lon_grd_step      = lat_grd_step;
lon_grd_max       = 360.0;
h_grd             =   0.0; % Note that the synthesis is here done at a grid,
                           % so "h_grd" needs to be set to a height above the
                           % sphere with the radius "R", hence "0.0" (see
                           % <../docs/graflab.md ../docs/graflab.md>).  In this
                           % way, the radius of the evaluation points "r" will
                           % be "1.0".  If you do the synthesis at scattered
                           % points, you should set "h_sctr" to "1.0".
out_path          = '../data/output/howto-g06-topography-sph-coord';
```

```
quantity_or_error = 0;
quantity          = 11;  % Gravitational potential; in this case, however,
                         % we synthesize the Earth's topography
fnALFs            = 1;
export_data_txt   = 1;
export_report     = 1;
export_data_mat   = 1;
display_data      = 2;
graphic_format    = 6;
colormap          = 1;
number_of_colors  = 60;
dpi               = 300;
status_bar        = 1;
```

Do the synthesis

```
out = GrafLab('OK', ...
    GM, ...
    R, ...
    nmin, ...
    nmax, ...
    ellipsoid, ...
    GGM_path, ...
    crd, ...
    point_type, ...
    lat_grd_min, ...
    lat_grd_step, ...
    lat_grd_max, ...
    lon_grd_min, ...
    lon_grd_step, ...
    lon_grd_max, ...
    h_grd, ...
    [], ...
    [], ...
    [], ...
    [], ...
    out_path, ...
    quantity_or_error, ...
    quantity, ...
    fnALFs, ...
    [], ...
    export_data_txt, ...
    export_report, ...
    export_data_mat, ...
    display_data, ...
    graphic_format, ...
    colormap, ...
    number_of_colors, ...
    dpi, ...
    status_bar);
```

You may now take a look at the output files.

```
fprintf("The ""%s*_Gravitational_potential.png"" file shows the " + ...
        "synthesized topography.\n", out_path);
```

Note that GrafLab thinks it computed the gravitational potential. It has no idea (how could it?) that we actually computed the Earth's topography. This is why the plot, the report file, the file name, etc. still report the gravitational potential.

Synthesis of the Earth's topography in ellipsoidal coordinates

Now what if you want to synthesize, say, the Earth's topography, but your evaluation points are given in ellipsoidal coordinates rather than in spherical coordinates? In this case, you **cannot** simply set "crd = 0" and use zero ellipsoidal heights. This is because this causes the evaluation points to reside on the ellipsoid, hence the points have (in general) a radius that is not equal to "1.0" (unit sphere). In such cases, GrafLab performs **solid** spherical harmonic synthesis which is not desired here.

We have to fool GrafLab somehow. More specifically, we have to transform the ellipsoidal latitudes of the computation points to their spherical counterpart assuming zero ellipsoidal heights. The spherical coordinates can then be entered to GrafLab similarly as in the previous example.

Let's define grid boundaries in **ellipsoidal** coordinates. Here, we assume the coordinates refer to GRS80. Note that since there is no difference between ellipsoidal and spherical longitudes for biaxial ellipsoids, we use the longitudes from the previous example.

```
% Vector of ellipsoidal latitudes
lat_ell = -90.0:1.0:90.0;

% The first eccentricity of GRS80
eEl = sqrt(0.006694380022903416);

% Now let's transform the ellipsoidal latitudes "lat_ell" to spherical
% latitudes "lat_sph".  The formula holds for points lying on the reference
% ellipsoid only.
lat_sph = atan(tan(lat_ell * pi / 180.0) * (1.0 - eEl^2)) * 180.0 / pi;
```

Note that the "lat_sph" vector does not have an equal spacing. The grid latitudes must be therefore entered in a special way to GrafLab: set the minimum grid latitude "lat_grd_min" to the "lat_sph" vector and then set both the latitudinal grid step "lat_grd_step" and the maximum grid latitude "lat_grd_max" to "'empty'" (see the GrafLab input parameters in ../docs/graflab.md).

```
crd              = 1;  % Important
lat_grd_min      = lat_sph;
lat_grd_step     = 'empty';
lat_grd_max      = 'empty';
h_grd            =   0.0; % We are still on the unit sphere (see above)
out_path         = '../data/output/howto-g06-topography-ell-cord';
```

Do the synthesis

```
out = GrafLab('OK', ...
```

```
    GM, ...
    R, ...
    nmin, ...
    nmax, ...
    ellipsoid, ...
    GGM_path, ...
    crd, ...
    point_type, ...
    lat_grd_min, ...
    lat_grd_step, ...
    lat_grd_max, ...
    lon_grd_min, ...
    lon_grd_step, ...
    lon_grd_max, ...
    h_grd, ...
    [], ...
    [], ...
    [], ...
    [], ...
    out_path, ...
    quantity_or_error, ...
    quantity, ...
    fnALFs, ...
    [], ...
    export_data_txt, ...
    export_report, ...
    export_data_mat, ...
    display_data, ...
    graphic_format, ...
    colormap, ...
    number_of_colors, ...
    dpi, ...
    status_bar);
```

You may now take a look at the output files.

```
fprintf("The ""%s*_Gravitational_potential.png"" file shows the " + ...
        "synthesized topography.\n", out_path);
```