HOWTO i01: Synthesis at grids residing on the Earth's topography

You will learn how to perform an efficient solid spherical harmonic synthesis at grids residing on irregular surfaces, e.g. planetary topographies.

All the isGrafLab input parameters are explained in ../docs/graflab.md and ../docs/isgraflab.md.

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

Synthesis at a grid residing on the Earth's topography in spherical coordinates

The outline of the experiment is as follows. At first, we synthesize heights of the Earth's topography at a grid in spherical coordinates. The heights are used to define the spherical radius of the grid points. Then, we use the point-wise computation in GrafLab to get the *reference* disturbing potential at the grid residing on the previously computed Earth' topography. Finally, we use isGrafLab to compute the disturbing potential at the very same points on the Earth's surface. The synthesis in isGrafLab is approximate but significantly faster than the point-wise evaluation in GrafLab. The approximation errors can be well-controlled and even safely negligible errors for most applications.

Let's start by synthesizing the Earth's topography from "DTM2006" up to degree "360". At first, we need to define some input parameters (see "HOWTO g06").

```
GM                = 1.0;
R                 = 1.0;
nmin              = 0;
nmax              = 360;
ellipsoid         = 1;
GGM_path          = '../data/input/DTM2006.mat';
crd               = 1;
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;
out_path          = '../data/output/howto-i01-topography';
quantity_or_error = 0;
quantity          = 11;
fnALFs            = 1;
export_data_txt   = 0;
export_report     = 0;
export_data_mat   = 0;
display_data      = 0;
status_bar        = 1;
```

Do the synthesize of the Earth's topography.

```
out_t = 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, ...
    [], ...
    [], ...
    [], ...
    [], ...
    status_bar);
```

Now, we have *heights above the geoid* in "out_t", because this is the quantity the "DTM2006.0" model offers. To synthesize the reference disturbing potential at the Earth's topography, we will use GrafLab and the point-wise mode. The grid mode cannot be used, as the radius of the grid points varies irregularly. With the point-wise mode in GrafLab, we need, however, the full spherical radius of each evaluation point. Therefore, for simplicity, we approximate the geoid by a sphere and add its radius "R" to the synthesized heights.

```
% Define the radius of the reference sphere of "EGM96"
R = 6378136.3;

% Get the full spherical radius of the grid points
out_t = R + out_t(:, end);
```

Save the synthesized topography to a format later required by isGrafLab. The data could alternatively be saved in to a text file.

```
is_path = '../data/output/howto-i01-dem.mat';
save(is_path, 'out_t', '-v7.3');
```

Now let's compute the reference disturbing potential on the Earth topography. At first, we define the GrafLab parameters.

```
GM                  = 3986004.415E+8;
GGM_path            = '../data/input/EGM96.mat';
point_type          = 2;
quantity            = 5;
out_path            = '../data/output/howto-i01-ref';

% Next, we create arrays with the coordinates of the evaluation points.
[lon_sctr, lat_sctr] = meshgrid(lon_grd_min:lon_grd_step:lon_grd_max, ...
                                lat_grd_min:lat_grd_step:lat_grd_max);
lat_sctr = lat_sctr(:);  % Spherical latitudes of the grid points
lon_sctr = lon_sctr(:);  % Spherical longitudes of the grid points
h_sctr   = out_t;        % Spherical radii of the grid points
```

Synthesis of the reference disturbing potential in GrafLab using the point-wise mode (see "HOWTO g01").

```
tic
out_sctr = GrafLab('OK', ...
    GM, ...
    R, ...
    nmin, ...
    nmax, ...
    ellipsoid, ...
    GGM_path, ...
    crd, ...
    point_type, ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    lat_sctr, ...
    lon_sctr, ...
    h_sctr, ...
    out_path, ...
    quantity_or_error, ...
    quantity, ...
    fnALFs, ...
    [], ...
    export_data_txt, ...
    export_report, ...
    export_data_mat, ...
    display_data, ...
    [], ...
    [], ...
    [], ...
```

```
    [], ...
    status_bar);
time_sctr = toc;
```

Now, we have the reference disturbing at a grid residing on the Earth's topography. At the very same grid, we can now compute the disturbing potential using isGrafLab. Let's define some isGrafLab-specific inputs parameters.

```
h_grd          = mean(out_t - R);  % The height *above the reference sphere with
                                   % the radius "R"*, from which the Taylor
                                   % continuation is performed.  Here, we
                                   % upward/downward continue from a mean sphere
                                   % passing through our topography.
is_mat_or_vec = 1;
out_path      = '../data/output/howto-i01-sph';
status_bar    = 0;
```

Do the synthesis in isGrafLab. We loop over various orders of the Taylor series. The higher is the order, the more accurate results are expected. For each order, we print RMS of the differences with respect to the reference values from GrafLab.

```
K = [0 1 2 5 10 15 20 25];
n = 1;
time_i = zeros(length(K), 1);
for ts = K

    tic
    out_i = isGrafLab('OK', ...
        GM, ...
        R, ...
        nmin, ...
        nmax, ...
        ellipsoid, ...
        GGM_path, ...
        crd, ...
        ts, ...
        lat_grd_min, ...
        lat_grd_step, ...
        lat_grd_max, ...
        lon_grd_min, ...
        lon_grd_step, ...
        lon_grd_max, ...
        h_grd, ...
        is_path, ...
        is_mat_or_vec, ...
        out_path, ...
        quantity, ...
        fnALFs, ...
        [], ...
        export_data_txt, ...
        export_report, ...
```

```
        export_data_mat, ...
        display_data, ...
        [], ...
        [], ...
        [], ...
        [], ...
        status_bar);
    time_i(n) = toc;


    fprintf("RMS error for Taylor order %d: " + ...
            "%0.16e\n", ts, rms(out_i(:, end) - out_sctr(:, end)));
    n = n + 1;

end
```

Note that for a few last Taylor orders, the RMS did not decrease. This is because we have reached *numerically* convergence of the Taylor series. In practice, the proper value of the Taylor order needs to be tuned. Generally, it depends mostly on the gravity field itself, the maximum degree "nmax" and the shape of the irregular surface.

Now let's look at the computation times.

```
fprintf("The synthesis in GrafLab using the point-wise mode took " + ...
        "%0.1f sec.\n", time_sctr);
for i = 1:length(K)
    fprintf("The synthesis in isGrafLab with Taylor order %d " + ...
            "took %0.1f.\n", K(i), time_i(i));
end
```

The synthesis at grids residing on planetary topographies is significantly faster with isGrafLab than it is with the point-wise mode of GrafLab. Moreover, this can be achieved without compromising the accuracy.

Synthesis in isGrafLab in ellipsoidal coordinates

If we repeat the same experiment but in *ellipsoidal* coordinates, we find that the accuracy is worse than it is in spherical coordinates. This is because with ellipsoidal coordinates, the upward/downward continuation should be done along the ellipsoidal normal but in reality is done along the spherical radius. This introduces approximation errors that cannot be narrowed by the order of the Taylor series. The accuracy is therefore worse. Further details are provided in the paper on isGrafLab (the reference is provided at the "README.md").