HOWTO g09: Exploit the symmetry of Legendre functions

You will learn about the conditions that must be satisfied to take advantage of the symmetry property of Legendre functions.

Legendre functions are symmetric with respect to the equator as follows,

$$\bar{P}_{nm}(\sin\varphi) = (-1)^{(n+m)}\,\bar{P}_{nm}(\sin(-\varphi)).$$

Therefore, if the grid contains both the positive and the negative latitudes, φ and $-\varphi$, respectively, the Legendre function for one of the two needs to be computed only. The other is obtained efficiently by the symmetry property.

In GrafLab, *all* of the following conditions must be satisfied to employ the symmetry of Legendre functions.

- A functional of the geopotential is selected (the symmetry property is not implemented for commission errors).
- The grid-wise computation mode is selected.
- The extended-range arithmetic approach is selected to compute Legendre functions (the whole improvement is tailored to high harmonic degrees, for which the standard and modified forward column approaches do not provide accurate results anyway).
- All positive latitudes have their negative counterpart or vice versa (up to a given threshold to suppress numerical inaccuracies, currently "100.0 * eps" degrees). The zero latitude, i.e., the equator, may be included in the grid (again, within the "100.0 * eps" deg numerical threshold).

If all these conditions are satisfied, the symmetry property is employed *automatically*, meaning that no additional action from the user is required to enable the more efficient variant.

Examples of some symmetric grids (shown are only the latitudes):

- Equator included

```
lat = [-90 -60 -30 0 30 60 90]
```

- Equator included

```
lat = [-80 -60 -40 -20 0 20 40 60 80]
```

- Equator excluded

```
lat = [-35 -25 -15 -5 5 15 25 35]
```

```
lat = [-90 -85 -80 80 85 90]
```

- Varying spacing

```
lat = [-90 -80 -75 -70 -69 -68 -67 -66 -65 65 66 67 68 69 70 75 80 90]
```

Examples of some grids that are not considered as symmetric (shown are only latitudes):

- The negative latitude of -90 deg does not have its positive counterpart

```
lat = [-90 -60 -30 0 30 60]
```

- The difference "abs(abs(-4.99) - 5.0)" is larger than the threshold of "100.0 * eps" degrees

```
lat = [-35 -25 -15 -4.99 5 15 25 35]
```

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

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

Numerical example

Let's define a spherical grid that is symmetric with respect to the equator.

```
% Latitudes
lat = -90.0:0.1:90.0;

% Longitudes.  The grid step is large in this example, as longitudes do not
% affect the grid symmetry in any way
lon = 0.0:5.0:360.0;


% Grid height
h = 0;
```

Now define the GrafLab input parameters.

```
GM                = 3986004.415E+8;
R                 = 6378136.3;
nmin              = 0;
nmax              = 360;
ellipsoid         = 1;
GGM_path          = '../data/input/EGM96.mat';
crd               = 1;
point_type        = 0;
lat_grd_min       = lat;
lat_grd_step      = 'empty';
lat_grd_max       = 'empty';
lon_grd_min       = lon;
lon_grd_step      = 'empty';
lon_grd_max       = 'empty';
h_grd             = h;
out_path          = '../data/output/howto-g09-symm';
quantity_or_error = 0;
quantity          = 5;
fnALFs            = 3;  % Important
export_data_txt   = 0;
export_report     = 1;
export_data_mat   = 0;
display_data      = 0;
status_bar        = 1;
```

Do the synthesis.

```
tic
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);
time_symm = toc;
```

Now let's modify a single latitude, such that the last symmetry property discussed above is not satisfied.

```
lat(1)      = lat(1) + 1000.0 * eps;
lat_grd_min = lat;
```

Do the synthesis.

```
tic
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);
time_nosymm = toc;
```

Now let's compare the computation times.

```
fprintf("Symmetric grid:     %0.1f sec\n", time_symm);
fprintf("Non-symmetric grid: %0.1f sec\n", time_nosymm);
```

The speed-up factor grows with increasing harmonic degree and/or increasing number of latitudes. If possible, you should always try to exploit the symmetry property.

```
% For instance, if, for some reason, you have to slice your grid into
% latitudinal bands and call GrafLab multiple times, slice your grid like this
lat1 = [-90.0: 0.01:-80.0 80.0:0.01:90.0];
lat2 = [-79.99:0.01:-70.0 70.0:0.01:79.99];

% Not like this
lat3 = [-90.0:0.01:-70.0];
lat4 = [ 70.0:0.01: 90.0];
```