HOWTO g08: Grids with varying spacings in latitudes and longitudes

You will learn how to do the efficient grid-wise synthesis, but this time with varying grid step in latitude and/or longitude.

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

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

Varying grid spacing in latitude

Let's assume we want to synthesize the disturbing potential at a grid residing on the GRS80 reference ellipsoid. The tricky part is that we need a constant sampling in *spherical* latitude. In terms of the *ellipsoidal* latitude, the spacing therefore varies. Fortunately, GrafLab makes it possible to define grids with varying spacing in latitude and longitude.

Suppose that our grid resides on GRS80, but is defined by a vector of spherical latitudes "lat_sph" with a constant spacing. Next, we have a vector of spherical longitudes "lon" and the constant height of the grid above the reference ellipsoid "h". The spherical latitudes "lat_sph" can be transformed into ellipsoidal ones "lat_ell", and these ellipsoidal latitudes (with varying spacing) can then be used to define a grid in GrafLab.

```
% Vector of spherical latitudes
lat_sph = -90.0:1.0:90.0;

% Vector of longitudes
lon = 0.0:1.0:360.0;

% Constant ellipsoidal height
h = 0.0;

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

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

The spherical and the ellipsoidal longitudes are equal, so no transformation is required for grid longitudes. 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                 = 0;  % Ellipsoidal coordinates
point_type          = 0;
lat_grd_min         = lat_ell; % Our column vector of ellipsoidal latitudes
```

```
lat_grd_step      = 'empty'; % Required if "lat_grd_min" is an array
lat_grd_max       = 'empty'; % Required if "lat_grd_min" is an array
lon_grd_min       = lon;     % Our column vector of ellipsoidal longitudes
lon_grd_step      = 'empty'; % Required if "lon_grd_min" is an array
lon_grd_max       = 'empty'; % Required if "lon_grd_min" is an array
h_grd             = h;
out_path          = '../data/output/howto-g08-varying-grd-latitude';
quantity_or_error = 0;
quantity          = 5;  % Disturbing potential
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

```
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);
```

In the very same way, you can also work with varying longitudinal step. The spacing may also vary if the coordinates of the evaluation points that enter GrafLab are spherical ("crd = 0"). It doesn't matter.