HOWTO g01: Synthesis at grids and at scattered points

You will learn how to perform a basic spherical harmonic synthesis at grids and scattered points. The scattered points will be loaded from a text file, a binary "mat" file and also will be taken from MATLAB variables.

Throughout the cookbook, we keep the same names of the input parameters to the GrafLab function. They are all explained in detail in ../docs/graflab.md.

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

Synthesis at a grid

Define the GrafLab inputs parameters. Throughout the cookbook, we synthesize the disturbing potential as an example (see the "quantity" variable).

```
GM                  = 3986004.415E+8;
R                   = 6378136.3;
nmin                = 0;
nmax                = 360;
ellipsoid           = 1;
GGM_path            = '../data/input/EGM96.mat';
crd                 = 0;
point_type          = 0;  % Computation at a grid
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-g01-grd';
quantity_or_error = 0;
quantity            = 5;
fnALFs              = 1;
export_data_txt     = 1;
export_report       = 1;
export_data_mat     = 1;
display_data        = 0;
status_bar          = 1;
```

Do the synthesis.

```
tic
out_grd = 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_grd = toc;
```

Now you may take a look at the files generated by GrafLab.

- The report file summarizes details of the synthesis (GGM model path, minimum and maximum harmonic degrees, computation time, etc.):

```
fprintf("""%s_Report.txt"".\n", out_path);
```

- Output numerical data in the text format (the structure of the file is always explained at the end of the report file):

```
fprintf("""%s.txt""\n", out_path);
```

- Output numerical data in the MATLAB binary format (the same structure as that of the text file):

```
fprintf("""%s.mat""\n", out_path);
```

Synthesis at scattered points from MATLAB variables

Let's take the points from the grid-wise synthesis and consider them as scattered points now. We have already did a synthesis at these points using the grid mode. Now, we repeat the same computation, but with the point-wise mode. Both results should therefore be the same.

At first, we have to define three arrays of spherical coordinates of the evaluation points. With the grid-wise computation, we had to define only the grid boundaries. With scattered points, we need to get coordinates of each individual grid point. So let's create the grid and transform it to arrays so that we can pretend the points are scattered. Note that you can pass any points to "lat_sctr", "lon_sctr" and "h_sctr", be them regularly or irregularly distributed. We use regularly sampled points to demonstrate the difference between the computational speed at grids and at scattered 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(:);
lon_sctr = lon_sctr(:);
h_sctr   = zeros(length(lat_sctr), 1);


% Update the GrafLab input parameters
point_type = 2;  % Evaluation points from MATLAB variables
out_path   = '../data/output/howto-g01-sctr-var';
```

Do the synthesis.

```
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;
```

Did you notice how slow the synthesis was at scattered points, despite using the very same points as in the grid-wise synthesis? This is because the synthesis in grids takes advantage of an efficient FFT-based algorithm. This technique cannot be used, however, with scattered points. With increasing number of evaluation points and/or maximum degree of the synthesis, the point-wise computations are therefore slower.

```
fprintf("The grid-wise synthesis took %0.3f sec.\n", time_grd);
fprintf("The point-wise synthesis took %0.3f sec.\n", time_sctr);
```

Now let's compute the RMS of the differences between the synthesis at a grid and at scattered points. The value shoud be very small, say, "10^-12" or less.

```
fprintf("The RMS is: %0.16e\n", rms(out_grd(:, end) - out_sctr(:, end)));
```

```
fprintf("Now you may want to explore the ""%s*"" files.\n\n", out_path);
```

Synthesis at scattered points from a text file

A few variables need to be modified to perform the synthesis at scattered points. We will load the evaluation points from the "sctr_points_path" file (5 points).

```
point_type       = 1;  % Evaluation points from a text file
sctr_points_path = '../data/input/sctr-points.txt';
out_path         = '../data/output/howto-g01-sctr-load-txt';
```

Do the synthesis.

```
GrafLab('OK', ...
    GM, ...
    R, ...
    nmin, ...
    nmax, ...
    ellipsoid, ...
    GGM_path, ...
    crd, ...
    point_type, ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    sctr_points_path, ...
    [], ...
    [], ...
    [], ...
    out_path, ...
    quantity_or_error, ...
    quantity, ...
    fnALFs, ...
    [], ...
    export_data_txt, ...
    export_report, ...
    export_data_mat, ...
    display_data, ...
    [], ...
    [], ...
    [], ...
    [], ...
    status_bar);
```

```
fprintf("Now you may want to explore the ""%s*"" files.\n", out_path);
```

Synthesis at scattered points from a binary mat file

You can import scattered points also from a "mat" file. Let's save now the points from the text file "sctr_points_path" into a binary file and use the new binary file to load the points.

```
sctr_points = load(sctr_points_path);
sctr_points_path = '../data/output/sctr-points.mat';
save(sctr_points_path, 'sctr_points', '-v7.3');
```

Update the GrafLab input parameters.

```
out_path = '../data/output/howto-g01-sctr-load-mat';
```

Do the synthesis.

```
GrafLab('OK', ...
    GM, ...
    R, ...
    nmin, ...
    nmax, ...
    ellipsoid, ...
    GGM_path, ...
    crd, ...
    point_type, ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    [], ...
    sctr_points_path, ...
    [], ...
    [], ...
    [], ...
    out_path, ...
    quantity_or_error, ...
    quantity, ...
    fnALFs, ...
    [], ...
    export_data_txt, ...
    export_report, ...
    export_data_mat, ...
    display_data, ...
    [], ...
    [], ...
    [], ...
    [], ...
    status_bar);
```

```
fprintf("Now you may want to explore the ""%s*"" files.\n", out_path);
```