VISBUNDLE

Visualization tools for geodetic data sets

Nico Sneeuw, Matthias Weigelt, Matthias Roth,
Markus Antoni, Mohammad Tourian et al.

affiliation: Institute of Geodesy, University of Stuttgart

May 4, 2021

Terms of use

In publications, we suggest the following acknowledgment:

The toolbox VISBUNDLE *(version #/date #) was developed by N. Sneeuw, M. Weigelt, M. Roth, M. Antoni, M. Tourian et. al. and its latest version is provided via download from the Institute of Geodesy (GIS), University of Stuttgart:*
`http://www.gis.uni-stuttgart.de/en/research/downloads/`

For feedback, questions or bugs reports, please contact bundle@gis.uni-stuttgart.de

1 Content and history of the VISBUNDLE

MATLAB enables a quick or advanced visualization of data, which is an important reason for its usage in teaching or academic research. Nevertheless, users might miss certain features or built-in routines might be too general or difficult to handle. Hence the question arises, whether someone has faced a similar problem, or if there is a simple solution.

The VISBUNDLE package is a collection of small routines, which support the visualization of "geodetic" data sets in MATLAB. From their concepts, these functions can be sub-divided into

- visualization tools for time series or statistics,
- visualization tools for location dependent data,
- and modified color schemes.

According to the documentation, the first routines were created by Nico Sneeuw since 1993 at the IAPG, TU Mnchen, and collected until 2000. The software was updated and extended — e.g. by map projections or visualization of spherical harmonics — in the following years by Nico Sneeuw and Matthias Weigelt, firstly at the IAPG, TU Mnchen, then at the Department of Geomatics Engineering at the University of Calgary, and latest at the Institute of Geodesy (GIS) at the University of Stuttgart.

In 2013, the MATLAB routines at GIS were reviewed and re-arranged by M. Antoni, B Devaraju and M. Roth. The developed prominent toolbox – the SHBUNDLE for spherical harmonic synthesis and analysis – was published on a webpage of the institute, other tools were collected in their own "bundles" and offered to institutes' members on a local server. The toolbox for visualization was reduced to its core and published with the new name VISBUNDLE on the SHBUNDLE's webpage.

In 2021, we decided to emphasize the toolbox VISBUNDLE with minor updates and an extended documentation on its own webpage.

Please feel free to make comments on the software, notify us about bugs or even provide useful enhancements via bundle@gis.uni-stuttgart.de.

Please consider:

- Ocatve can often substitute MATLAB for calculations without code modification, but the graphic tools and features sometimes differ. Problems in Octave (version 4.2.2) are observed for
  - *graypatch.m*: graphics look similar, but the gray patches are not in the 1st layer,
  - *barplotcol.m*: no graphics due to missing built-in routine *bar3.m*,
  - *plotcolm.m*: no graphics due to missing mapping toolbox or different commands.

  - *sphrwarp.m*: not all options of visualiation on the sphere are possible
- The function *plotcolm.m* requires the mapping toolbox of MATLAB.

2 Time series and statistics

barplotcol.m improves as a 'wrapper function' the behaviour of the built-in function *bar3.m* and provides three-dimensional bar charts per row and column. In opposite to the original function – in its standard call – the color is not chosen depending on the column in the matrix, but depending on the entries' values. Each matrix entry creates a colored block and the color is either defined by the total height of the block ('building'-option), or equal height levels are marked by the same color ('floor'-option).

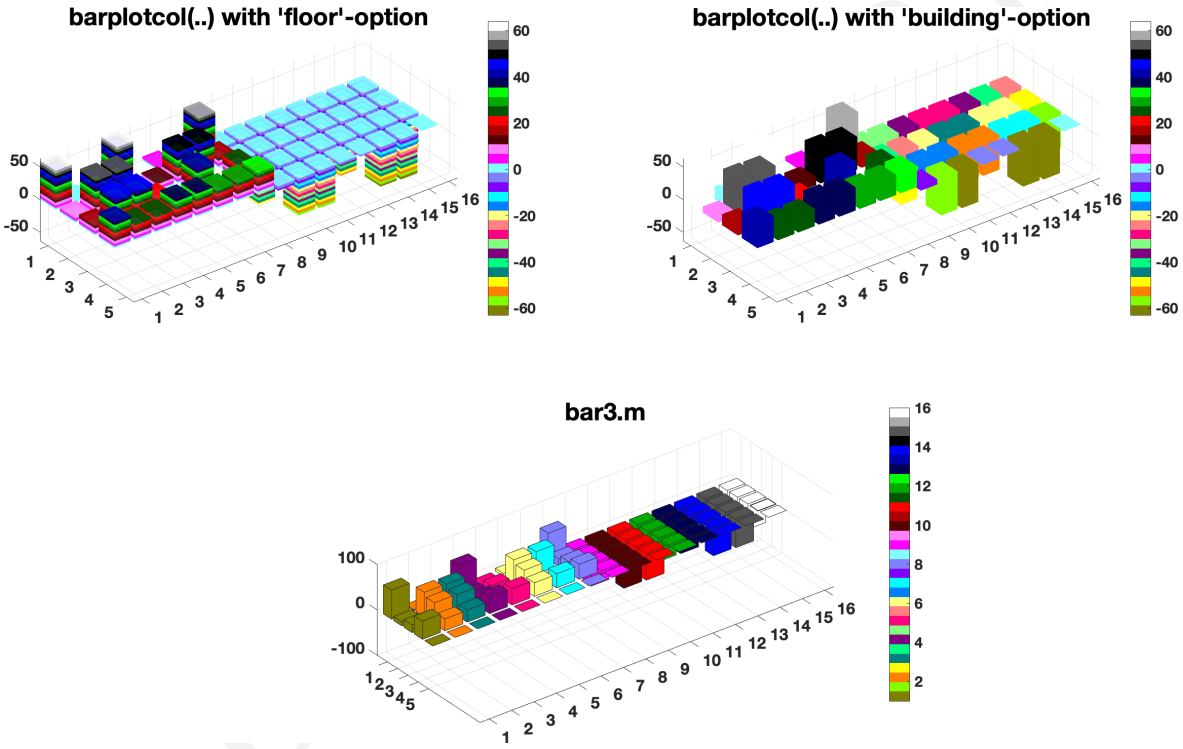

Figure 2.1: These three-dimensional bar charts are produced by the routine *barplotcol.m* – with its two options – and the original function *bar3.m*. The particular colorbar is only chosen to illustrate the 'floor' option better.

graypatch.m highlights user-defined time intervals in one figure by switching the background from white to gray and back to white again. This can be used for emphasizing different behaviors of a signal in user-defined time intervals or for highlighting smaller time spans (month/years) in a longer time series. This feature is strongly recommended for time series on presentation slides for a better readability.

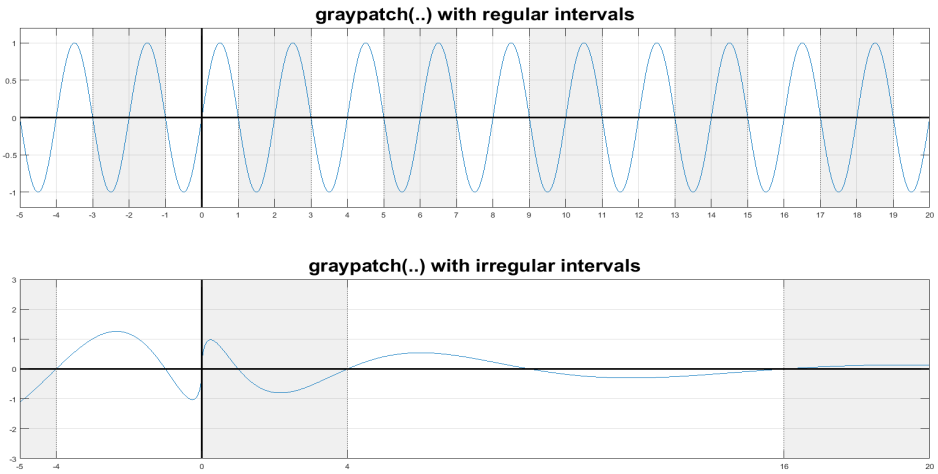

Figure 2.2: The upper figure illustrates the effect of the routine *graypatch.m* for a time series with regular intervals, the lower figure shows the case of irregular time intervals.

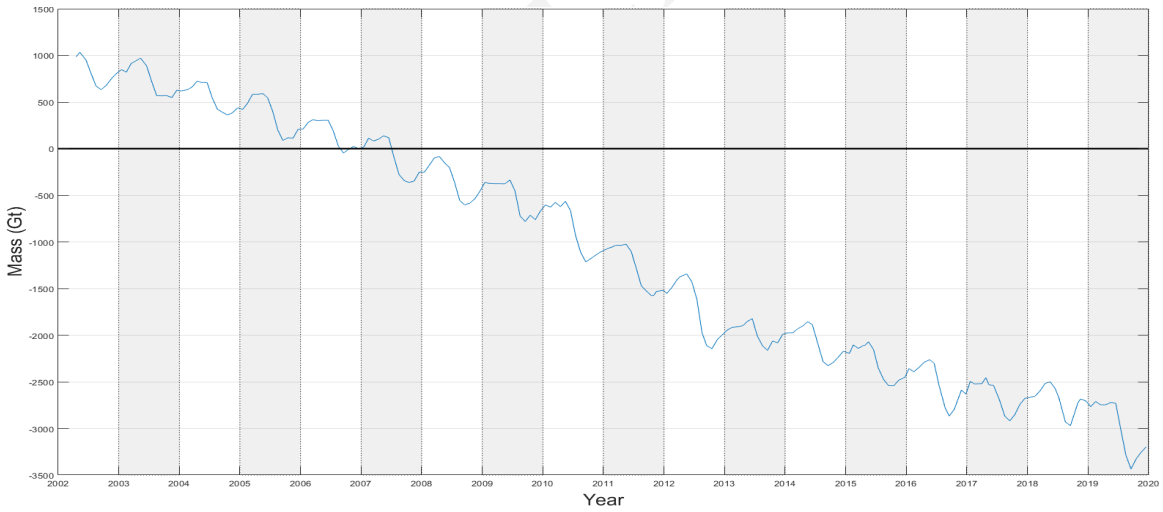

Figure 2.3: Time series of the mass variation in Greenland estimated from GRACE-data by Shuang Yi and re-visualized by the updated *graypatch.m* function

plotmulti.m stacks (similar) signals within the same time interval into one figure, but with copied and shifted x-axes. All sub-figures use the same labeling and the different figures are visually separated by switching the background from white to gray and back to white again. This kind of visualization increases the readability of very similar signals and it is in particular recommended for presenting/comparing data in articles without color printing.

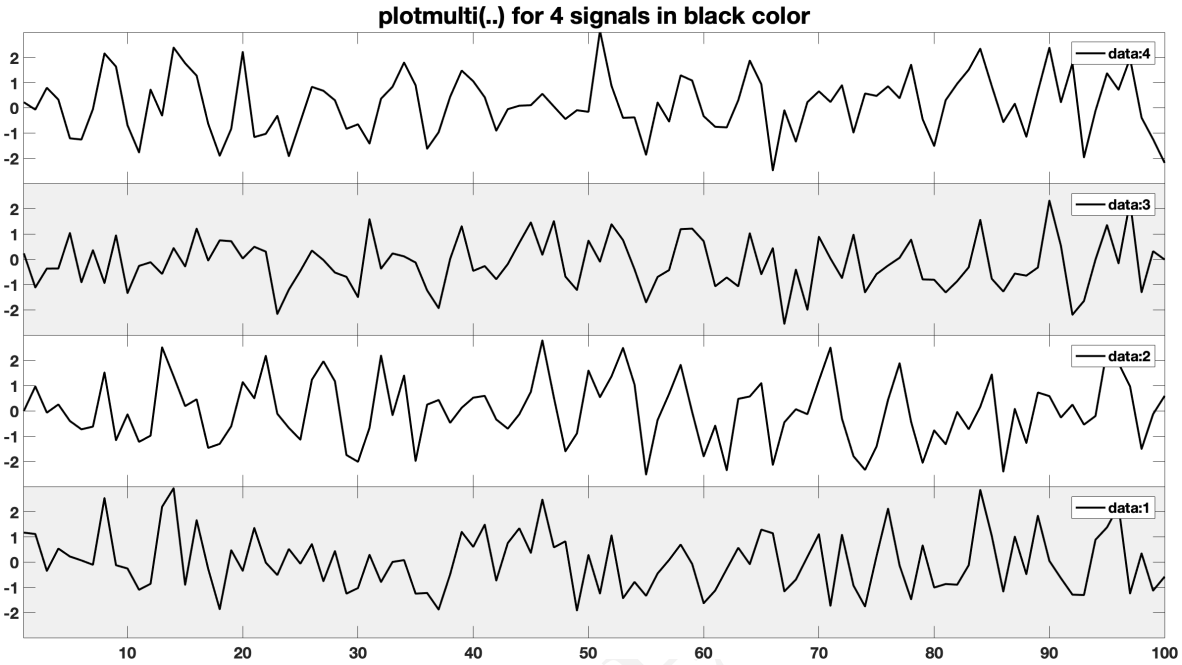

Figure 2.4: Four random signals are stacked together by *plotmulti.m* into one big figure with common labels.

3 Data in the Euclidian space

bubble_voronoi.m calculates a kind of Voronoi diagram for points given by their coordinates in the x-y-plane. In opposite to standard voronoi routines, each cell has a circular shape instead of a polygonal one. The size of the circles is determined by a user-defined maximum radius, and optionally the total amount of polygon points.

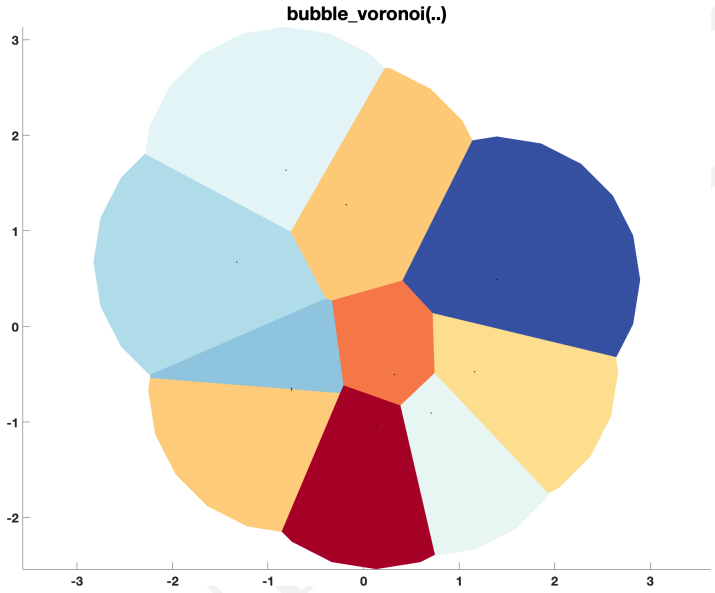

Figure 3.1: Example of a circular Voronoi diagram by *bubble_voronoi.m*

plotcol.m visualizes location dependent data – with 1D, 2D or 3D coordinates – and observed values, which define then the colors. A color is used either along the line between two subsequent points, or in the marker at the location. The program combines the features of the built-in routines *mesh.m*, *scatter.m* and *scatter3.m*.

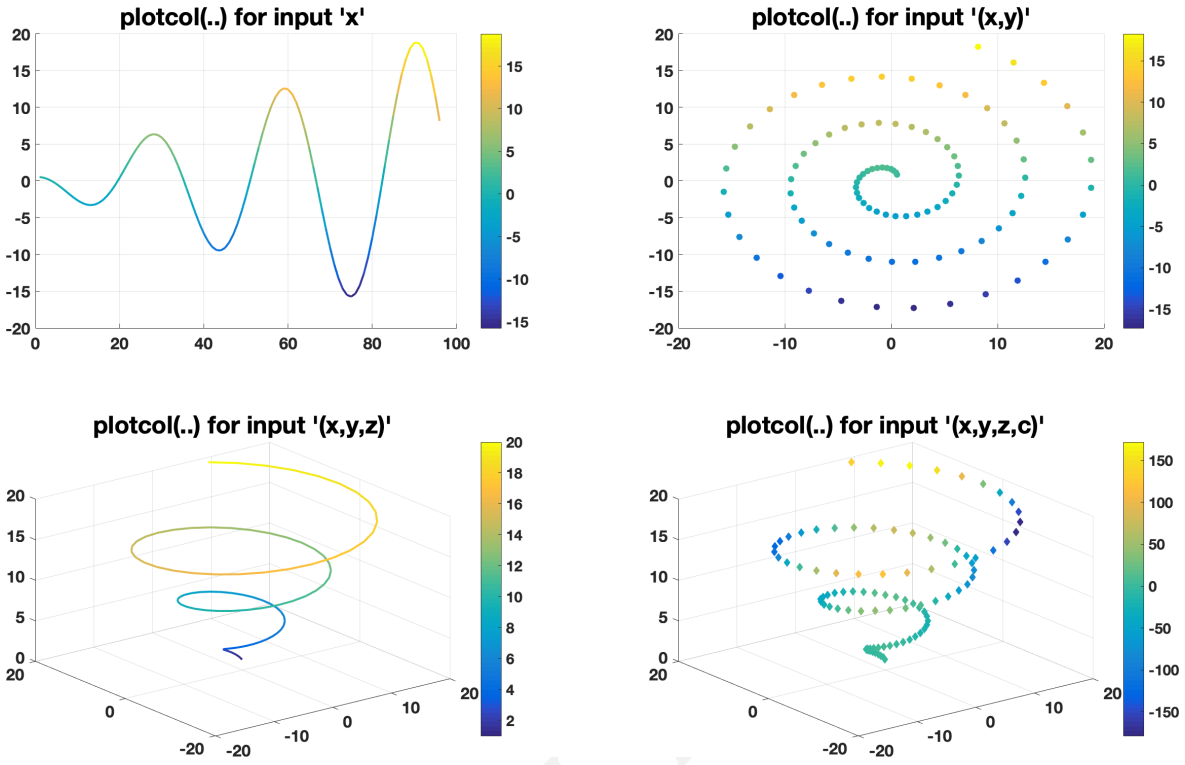

Figure 3.2: Four different signals – for 1D, 2D,3D and 4D data – are visualized by function *plotcol.m*.

4 Data on the sphere

circ4sph.m calculates lines of constant spherical distance and of constant azimuth on a sphere w.r.t. an arbitrary reference point. If the lines pass the mapping boundary or a selected maximum distance around the calculation point, then NaNs are inserted into the vectors to avoid a visual connection. The function requires the UBERALL-bundle in particular the routines *R2.m*, *R3.m* and *jump2nan.m*.

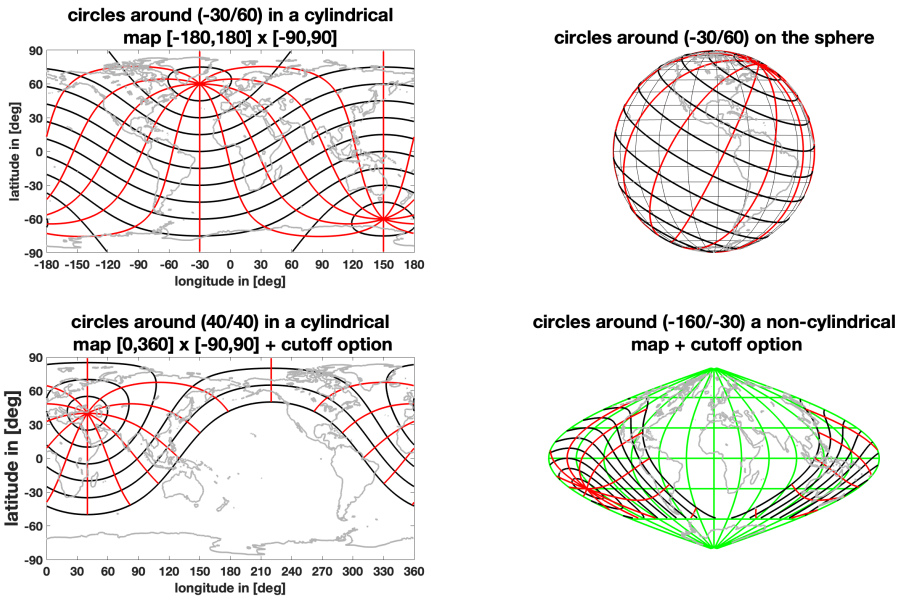

Figure 4.1: The figures illustrates lines of constant azimuth (red) and lines of constant spherical distance (black) around a calculation point in cylindrical and non-cylindrical mappings. The coordinates are determined by the routine *circ4sph.m*

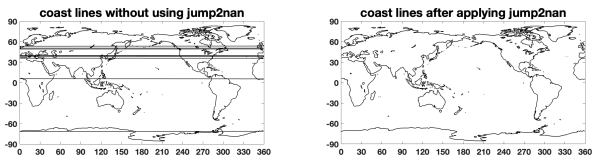

Figure 4.2: The routine *jump2nan.m* – in the UBERALL-bundle – splits lines/vectors by inserting `NaN` based on a threshold condition. This avoids the visual connection of longitude values across the datum switch.

plotcolm.m is a modification of *plotcol.m* for the visualization of location dependent data within a map projection. The function requires the mapping toolbox of MATLAB and combines the features of the built-in routines *mesh.m*, *scatter.m* and *scatter3.m*.

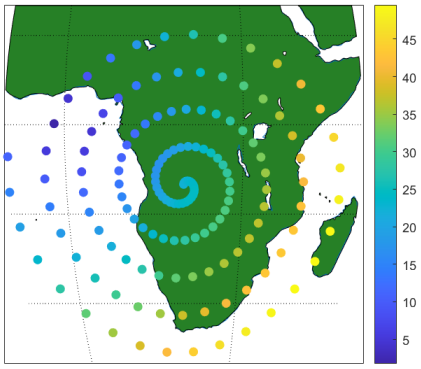

Figure 4.3: The routine *plotcolm.m* visualizes location dependent data within a map projection

sphrwarp.m maps any $N \times M$ matrix onto sphere. The given matrices define the color of the patches or the grid lines, or the height above the sphere. The function uses the sub-routine *sphere2.m*, which is a variation of the built-in routine *sphere.m* but with 2 input arguments.

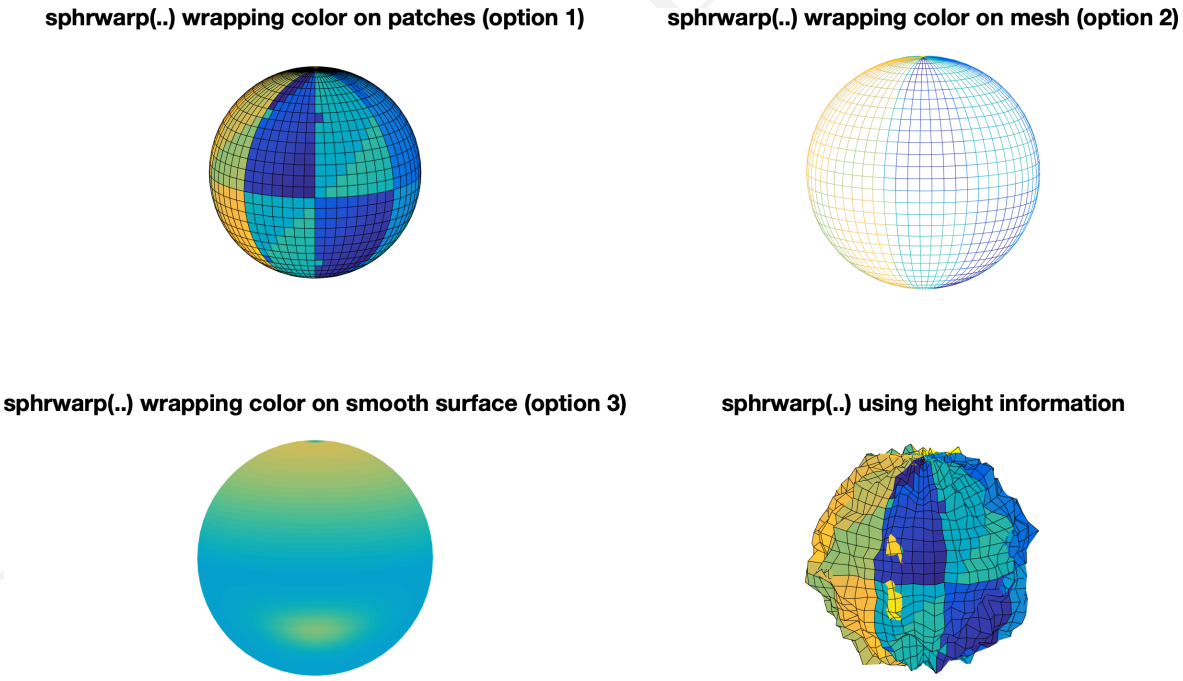

Figure 4.4: A matrix of the dimension 30×60 is mapped with different options of *sphrwarp.m* onto a unit sphere.

skyplot.m visualizes positions (of celestial bodies) in a topocentric coordinate system, i.e the azimuth and elevation angle in the observation location. The function is a modification of the built-in function *polar.m* but in a North-oriented figure and a maximum zenith angle of 90 degrees.

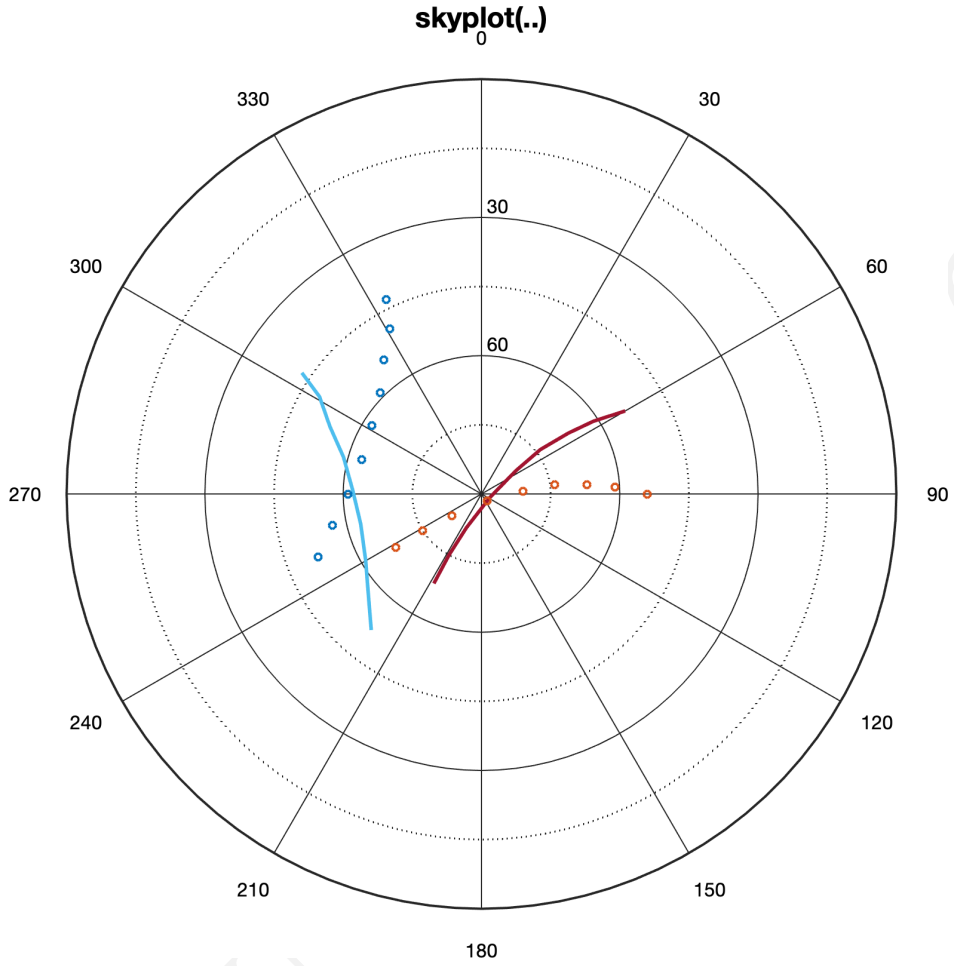

Figure 4.5: The azimuth and elevation angle of 4 objects are visualized by *skyplot.m* in the topocentric system.

5 Color schemes

colbrew.m and *redblue.m* provide a set of diverging color schemes to indicate opposite behaviors like wet-dry, cold-hot, or high-low altitudes. The common idea is it, to highlight the extreme values by opposite and saturated colors, and fill in between only bright colors. Some color schemes are chosen to be colorblind safe according to `www.colorbrewer2.org`. For quick interpretations in hydro geodesy, it is recommended to flip one of the red-blue color schemes, to obtain blue regions for the large water storage and red regions for dry areas.

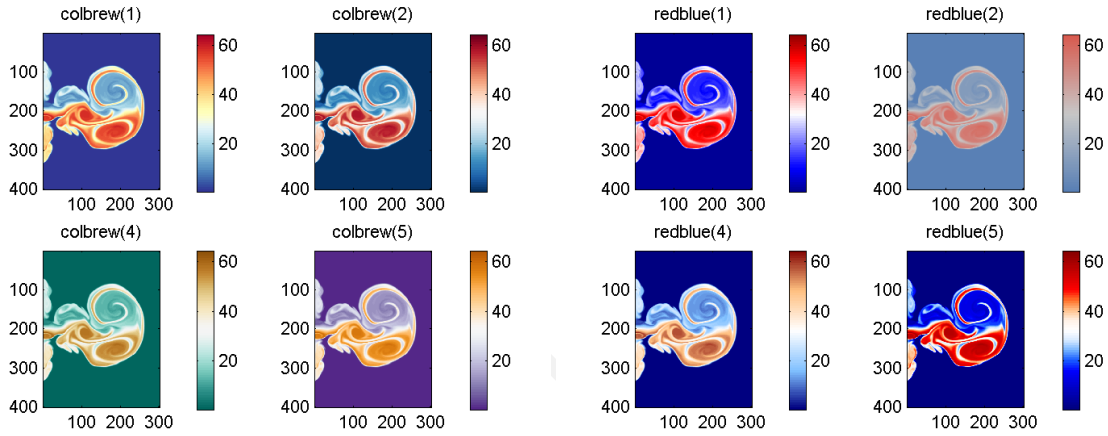

Figure 5.1: Some of the color schemes generated by *colbrew.m* and *redblue.m*

In particular for members of the University of Stuttgart, the routine *colbrew.m* offers another color scheme, which varies between the light blue, white, and the medium gray color of the corporate design:

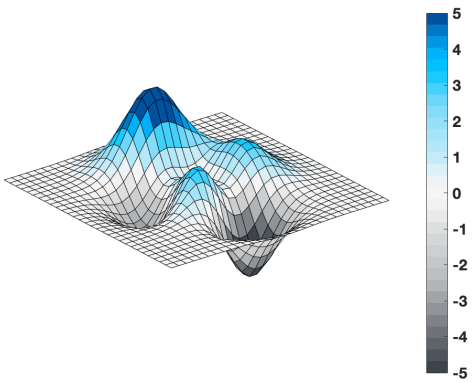

Figure 5.2: Color scheme based on the corporate design at University of Stuttgart

gray1.m and *gray2.m* create two colormaps, where both are ranging from dark gray to white with a clear break at the center.

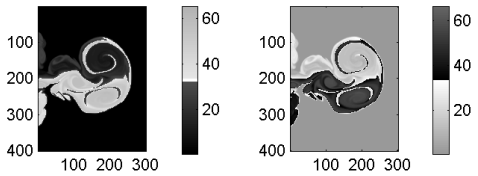

Figure 5.3: Two gray color schemes produced by *gray1.m* and *gray2.m*