NEROGRAV

New Refined Observations of Climate Change from Spaceborne Gravity Missions

**International Spring School**

**10.-14. March 2025**

Anne Springer, Mail: springer@geod.uni-bonn.de Christian Mielke, Mail: mielke@geod.uni-bonn.de

# Practical 4: Regional analysis of gridded total water storage data

**Purpose of the practical:** In this practical a GRACE time series of gridded total water storage anomalies (TWSA) will be analyzed for regional (geophysical) signals. First, hydrological signals in selected river catchments will be evaluated. Then, ice mass changes of the Greenland ice sheet will be computed followed by an investigation of an unusual signal in West Malaysia. Furthermore, the impact of the leakage effect will assessed. Finally, GRACE data over South Africa will be analyzed with respect to droughts.

**Directories:** The data sets needed for this practical can be found in the directory *data_Lab34*. The provided matlab functions can be found in the directory *functions_Lab34*. The provided python functions can be imported from the file `SpringSchoolLib34.py`.

**Exercise 1: Investigation of TWSA in selected hydrological catchments**

1. Load the gridded GRACE TWSA time series from practical 3 (`grace_tws.mat`) and the TWSA data from the GLDAS-NOAH model (`noah_tws.mat`). Both time series are centered with respect to the temporal mean 2003 to 2016.

2. Load polygons from individual hydrological catchments (i.e. area of land where precipitation collects and drains off into a common outlet): `amazon.mat, brahmaputra.mat, danube.mat, volta.mat`. Each file contains a matrix with two columns, the first column consisting of the longitude values of the polygon points, the second column containing the latitude values.

3. Compute catchment averaged time series for GRACE and GLDAS-NOAH data using the function `catchmentAverage` for each of the four catchments and visualize the two time series within one plot for each catchment using the function `showTimeseries`.

4. Compute trend, annual- and semiannual amplitudes of the catchment-averaged time series for GRACE and GLDAS-NOAH from 2003 to 2023 using the functions `time2decimalYears` and `fitTrend` and compare. The amplitude of the annual (semiannual) signal can be computed from corresponding sine and cosine parameters according to $A = \sqrt{p_{sin}^2 + p_{cos}^2}$.

5. Compute the residual signal of the Danube time series (2003 to 2023) by removing trend and seasonal signals using the approximated data provided by the function `fitTrend`. Visualize the residual signal using the function `showTimeseries`.

6. Interpret your results considering also the following aspects:
   - Discuss possible reasons for the different trends of GRACE and GLDAS-NOAH in the Brahmaputra catchment.
   - Discuss possible reasons for the large inter-annual variability indicated by the GRACE time series in the Volta catchment.
   - Analyze the residuals time series of the Danube catchment with respect to hydrological extremes.
   - Discuss limitations of hydrological models and GRACE observations in representing TWSA.

**Exercise 2: Investigation of ice mass changes**

1. Load polygon of Greenland (`greenland.mat`).
2. Compute a GRACE-based time series of total ice mass changes of the Greenland ice sheet in gigatons (Gt) using the function `catchmentAverage` and visualize it using the function `showTimeseries`. Identify periods with pronounced mass balance anomalies.
3. Compute the trend of ice mass changes in Gt/year using the function `fitTrend`. Why does your computed trend differ from trends published in recent papers?

**Exercise 3: Investigation of a mysterious signal**

1. Load the polygon of West Malaysia (`westMalaysia.mat`).
2. Compute a catchment averaged time series of GRACE TWSA using the function `catchmentAverage` and visualize it using the function (`showTimeseries`).
3. Can you explain the signal in winter 2004/2005?

**Exercise 4: Dealing with the leakage effect**

1. Load unfiltered GLDAS-NOAH TWSA (`noah_unfiltered_tws.mat`) additionally to DDK filtered GRACE and GLDAS-NOAH data.
2. Compute TWSA time series of the Rhine catchment for the three data sets (`catchmentAverage`) and visualize them within one plot (`showTimeseries`).
3. Compute annual amplitudes for the two GLDAS-NOAH data sets. Why do they differ?
4. How can we take into account the leakage effect? Implement a (simple) approach for correcting the filtered GRACE time series.
5. (optional) Investigate the leakage effect for other catchments. On which conditions does the magnitude of the leakage effect depend?

**Exercise 5 (optional): Computation of drought severity indices**

1. Load the polygon of South Africa (`southAfrica.mat`).
2. Compute catchment averaged time series of TWSA from GRACE and GLDAS-NOAH using the function `catchmentAverage` and visualize them using the function `showTimeseries`.
3. Compute the drought severity index (DSI) according to $DSI_{i,j} = \frac{TWSA_{i,j} - \overline{TWSA_j}}{\sigma_j}$ from GRACE and GLDAS-NOAH, respectively. $\overline{TWSA_j}$ is the climatology mean of TWSA of month j over all given years and $\sigma_j$ the corresponding climatology standard deviation $i$... year, $j$... month).
4. Visualize the computed DSI using the function `showTimeseries`.
5. A DSI of less than -2.0 is an exceptional drought, from -1.99 to -1.60 is an extreme drought, from -1.59 to -1.30 is a severe drought, from -1.29 to -0.80 is a moderate drought, from -0.79 to -0.50 is abnormally dry, from -0.49 to 0.49 is near normal, from 0.50 to 0.79 is slightly wet, from 0.80 to 1.29 is moderately wet, from 1.30 to 1.59 is very wet, from 1.60 to 1.99 is extremely wet, and higher than 2.0 is exceptionally wet.
   - In which years can you identify extreme droughts?
   - Do GRACE and GLDAS-NOAH agree regarding the drought classification?
   - Which advantage has a drought severity index based on GRACE and based on hydrological models, respectively?

**Functions:**

| function [time] = time2decimalYears (years, months) | |
|---|---|
| Convert time provided as vectors of years and months to decimal years | |
| Input | • `years`: $p \times 1$ vector of years<br>• `months`: $p \times 1$ vector of months |
| Output | • `time`: $p \times 1$ vector of time in decimal years |

| function [twsa] = catchmentAverage (data, lon, lat, polygon) | |
|---|---|
| Computation of catchment averaged TWSA. | |
| Input | • `data`: $n \times m \times p$ matrix containing the gridded data set (p ... # time steps)<br>• `lon` : $n \times 1$ vector containing the longitudes<br>• `lat` : $m \times 1$ vector containing the latitudes<br>• `polygon`: $b \times 2$ matrix which contains the polygon of a selected region (e.g. Amazon). The first column consists of the longitude values of the polygon points, the second column contains the latitude values. |
| Output | • `twsa`: $p \times 1$ vector containing a time series catchment averaged TWSA [mm] |

| function [param,dataApprox] = fitTrend (time, data, mode) | |
|---|---|
| Compute trend, annual and semi-annual signal | |
| Input | • `time`: $p \times 1$ vector of time in decimal years<br>• `data`: $p \times 1$ time series to be approximated by the estimated parameters, with p... number of time steps<br>• `mode`: string, defining if the semiannual signal should be included in the estimation process (choose 'annual' or 'semiannual') |
| Output | • `param`: $\#param \times 1$ vector of estimated parameters (bias, trend, sine and cosine component of the annual signal, sine and cosine component of the semiannual signal)<br>• `dataApprox`: $p \times 1$ vector of data approximated by the estimated parameters |

| function showTimeseries (data, ylim, legendStrings, titleString, ylabelStrings) | |
|---|---|
| Visualization of an arbitrary number of time series also with different number of time steps. | |
| Input | • `data`: struct containing the time and data values in the format [years1,months1,data1],[years2,months2,data2],..., this allows to provide several time series with different number of time steps<br>• `ylim`: lower and upper limit of the y-axis, e.g. [-150 150]<br>• `legendStrings`: legend entries provided as a struct of strings, e.g. {'GRACE', 'NOAH'}<br>• `titleString`: A string describing the Catchment (e.g. "Amazon" or "Rhine"), used for the title of the plot.<br>• `ylabelString`: string with label of the y-axis, e.g. "mm" |
| Output | • Plot of the time series. |