Integrated Geodetic Processing (IGP)
User manual

V1.0, 2 December 2024

This IGP user manual gives a short description of the software setup, and the steps to take to run the software. Detailed descriptions of the algorithms used can be found in the software scrips. The software is written in Matlab.

Software setup

The IGP software enables an integrated processing of various geodetic datasets, acquired by levelling, GNSS and/or InSAR. The software setup is visualized in Figure 1. The process transforms all the original datasets into the common Space-Time Matrix (STM) dataformat (Step 1: Initialization), reduces the GNSS and InSAR data to common evaluation points and epochs (Step 2: Reduction), integrates and tests the datasets (Step 3: Integration), and creates output, e.g., in the form of a map or table (Step 4: Prediction).

As the figure indicates, to enable the integration, the original data volumes are first reduced.

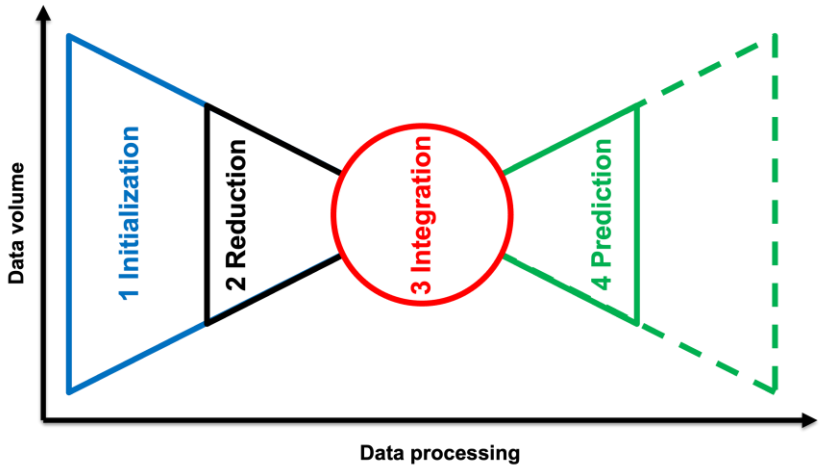

Figure 1 IGP software setup.

In the diagram of Figure 1, step 4, prediction and output generation, is depicted as using the outcome of the integration step. However, the same functionality can be applied to almost *any* space time matrix.

The software comes with a number of utility tools to display and visualize the contents of space time matrices. These tools include a simple display of the space time datasets meta data, plotting of the space time dataset network map, time series, estimated velocities and covariance matrix, and plotting and gridding of predicted displacements.

Each of the four processing steps can be applied within a dedicated *Space* (and equivalently Period), see Figure 2.

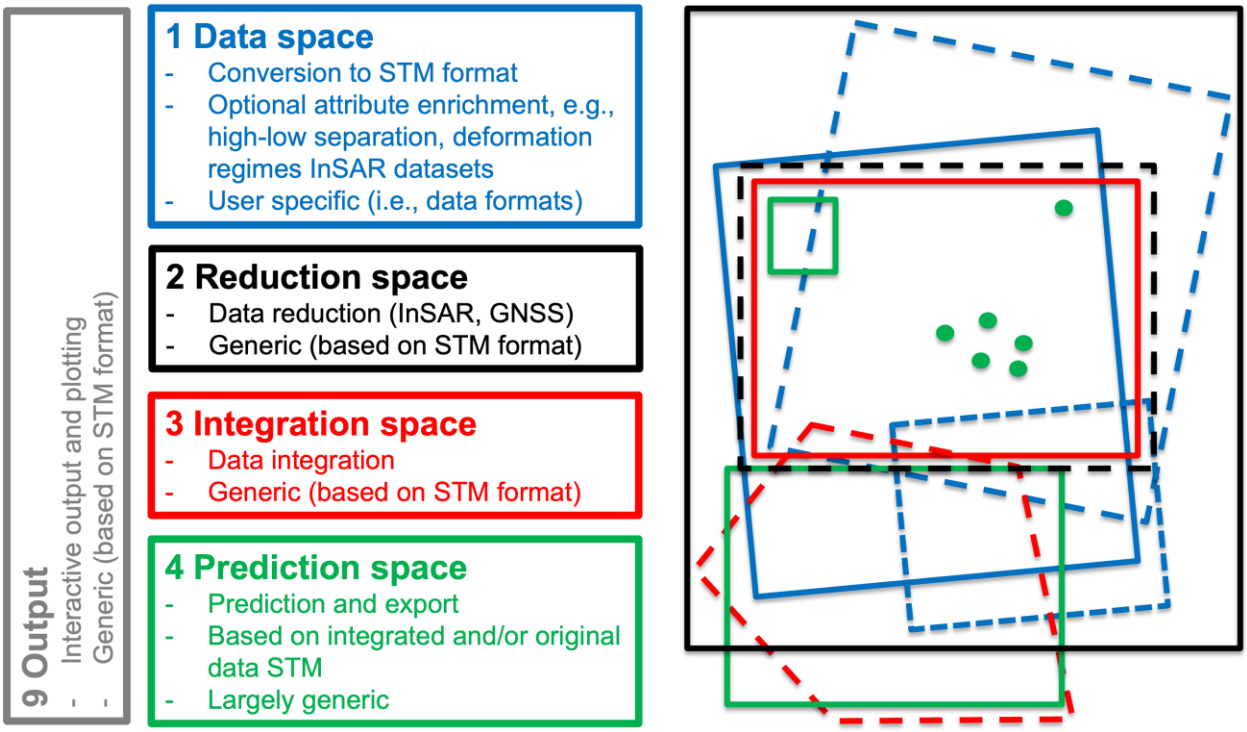

Figure 2 Concept of the different Spaces in the IGP software. For each of the four steps, a different Space may be defined. E.g., the Initialization (Step 1) is typically applied for the extent of each individual dataset, whereas the reduction step could be performed for the full covered area (for consistency of evaluation points and epochs among different projects). The Integration step could be performed on a certain Region of Interest, whereas the extent of the output could be different again (e.g., map, points). The same concept applies in the time domain. The 9 Output functionality can be used throughout the processing flow to output and visualize Space-Time Matrices.

Software Installation

At the location of choice, unzip the IGP package. This results in the following four directories:

igpdata > Placeholder for the original data
igptemplates > Templates for igpprojects (see below)
igpprojects > Initialization of datasets, Reduction, Integration, Prediction and output generation
igpsoftware > Software repository

To complete the installation, you must modify the file **`igpinit.m`** in the `igpprojects` directory. In the file `igpinit.m` the Matlab path must be set to the main `igpsoftware` location, and the structure variable `globalAttributesDefaults` must be defined with default values for the `globalAttributes`.

In principle, the IGP software can be run on any platform. However, especially the initialization and reduction step for InSAR data require a considerable amount of RAM. Therefore, use of a high-performance system may be needed (for these steps). Furthermore, the use of relatively new in-build Matlab functions and toolboxes is avoided as much as possible, so the software should also run with older versions of Matlab.

Directory and project structure

The **igpdata** directory is meant as the location for the original datasets. It can also be a placeholder, contain symbolic links to the original data, or can be empty. This directory is kept separately, to enable the use by different projects, and the efficient update by newly acquired data. It is not necessary to use this directory, the data may also be stored elsewhere.

Within the **igpprojects** directory different **projects** can be setup.

In each project the processing steps 1, 2, 3 and 4 can be run. It is not necessary to run every step in each project; a project may also make use of intermediate outputs of other projects. For example, a project may use data imported in another project.

The igpprojects directory contains a single initialization file, `igpinit.m`, with the path to the software directory (igpsoftware) and initial values for the global attributes. This is the file you have to modify at install time.

A typical layout for the igpprojects and igptemplates directory is

igpprojects/
- **groningen/**
- **simtests/**
- **waddenzee/**
  - **0_import_gnss/**
  - **0_import_insar/**
  - **0_import_levelling/**
  - **1_decompose_gnss/**
  - **2_reduce/**
  - **3_integrate/**
  - **4_predict/**
  - **9_output**
- **igpinit.m**

The difference between `igptemplates` and `igpprojects` is that `igptemplates` contains initial example Matlab scripts, whereas the subfolders in `igpprojects` are working directories which contain in addition to the (modified) Matlab scripts input, intermediate, output and log files. To start a new project in igpprojects, first copy a template from igptemplates, modify the scripts and then run.

The folder structure in igpprojects is quite flexible and allows for various choices. For instance, if you decide to investigate an alternative processing, you can create a new project of you can create a new subfolder within a project (e.g. 3_integrate_alternative), or just create a new output file within an existing subfolder.

The example layout contains a folder `9_output`. This folder contains initially a few mat files with the unstable area and coastlines for plotting. This folder is also intended for interactive plotting.

The actual software directory**, igpsoftware**, contains the following directories and files:

igpsoftware/

stmmain/	> main functions
stmutil/	> Space-Time Matrix utilities toolbox
crsutil/	> Coordinate Reference System utilities toolbox
mht/	> Multiple hypothesis testing toolbox (InSAR)
proj/	> Projection toolbox
rdnaptrans/	> RDNAP transformation toolbox (Dutch reference systems)
tseries2/	> GNSS timeseries toolbox
igptoolbox.cfg	
igpimport.m	

The function `igpimport.m` and file `igptoolbox.cfg` is used by the scripts to import the toolboxes. The file `igptoolbox.cfg` is used to link toolbox names in the scripts, that call `igpimport`, to a specific directory (Note: This file also contains a link to ../**igpdata/basemaps/** with shapefiles for the basemap). This facilitates easy configuration for test without additionally modifying the calling scripts.

Processing steps

The different processing steps are described here.

0. Project initialization

To start a new project copy a `template` directory from `igptemplates` to `igpprojects` and give it a new name. In this section we use the name `waddenzee`; you have now a new directory `igpprojects/waddenzee` .

It consists of several subdirectories, filled with Matlab scripts to run the software, set the input files and input parameters. Input files are sometimes also specified in asci text files.

The main conventions are:

- Run scripts within each subdirectory (e.g. `0_import_gnss`)
- The filenames and input parameters are set inside the script files. Occasionally, input files may be specified in a separate ascii text file
- Each script will write the output space time matrix (stm) files and log files in the subdirectory of the script
- The log files are versioned, i.e. a new log file is created (with a file date) every time the script is run, but the stm files will be overwritten (though this behavior can be changed by an input parameter)
- The input files almost always come from different directories, at the start the could be igpdata (or any other directory), or during the reductions from another sub-directory within the project. This can be specified either by giving the name of the input directory with the filename (full or relative path), or by setting the `inputdir` option.

1. Data initialization

The first step, data import, is done in a project directory. Ideally, each type of data is imported in a separate subdirectory, e.g.

```
igpprojects/
    waddenzee/
        0_import_gnss/
        0_import_insar/
        0_import_levelling/
```

Each directory contains an `igpImport….m` script. Edit this script to provide the proper input files and options as desired, and run the script. The output will be in the same folder as the scripts. Repeat this for each technique.

For gnss and levelling usually one stm output dataset is created. However, for insar multiple files are created. These filenames are stored in a ascii textfile file which can be used by later processing steps.

Whereas the processing scripts for Steps 2 to 4 are generic, this is not the case for this first step. The reason is that the format of the original datasets may be very different, depending on the specific processing tools used to generate them. Therefore, scripts and function they call are meant as an example, which should be adapted to the specific formats used.

2. Data reduction

To select and create the evaluation points and epochs, and to reduce the GNSS and InSAR datasets to these points and epochs, several steps need to be performed.

```
igpprojects/
    waddenzee/
        1_decompose_gnss/
            igpDecompose06gps.m
        2_reduce/
            unreducedfiles.txt
            igpSelect_waddenzee.m
            igpReduceInsar.m
            igpReduceGnss.m
            igpReduceCampaign.m
```

Before starting the reduction we have to decompose the GNSS dataset. This is done using the script `igpDecompose06gps.m` . This is done as a separate activity because it usually needs to be done only once and is independent of the selection of evaluation points and epochs.

The next step is to select evaluation points and epochs. This is done with the script `igpSelect_waddenzee.m` . This script will create a so-called **project file** with the selected evaluation points and epochs. The project file is actually a regular space time matrix file, except, it contains no data, only the evaluation points and epochs with their harmonized point and epoch identifiers, and several point and epoch attributes, including the original point and epoch names. The input file names are specified in `unreducedfiles.txt`.

After evaluation points and epochs have been selected the original space time matrix files are reduced using the scripts `igpReduce….m` . This is repeated for each technique. For campaign data (levelling and GNSS campaigns) this only consists off adding the harmonized point and epoch identifiers to the space time matrix dataset.

In principle it is possible to create multiple `2_reduce` directories (but giving it a different name), using for instance different settings of the parameters.

3. Data integration

The data integration can be carried out in one or multiple subdirectories within the project folder. There is quite some flexibility in organizing the structure. In the Waddenzee `template` only one directory is provided, but with several several scripts each using a different set of input options (mainly period of interest).

igpprojects/
- **waddenzee/**
  - **3_integrate**
    - **reducedfiles.txt**
    - **integrateWaddenzee2006.m**
    - **integrateWaddenzee2015.m**
  - **3_integrate_tests** (this folder is just an example, does not exits)
    - **reducedfiles.txt**
    - **integrateWaddenzeeNoLevelling.m**
    - **integrateWaddenzeeInSAROnly.m**

The input file names are given in a text file. Then each directory may also contain multiple integration runs.

4. Prediction

The prediction and export can be done in one or more subdirectories using two separate scripts within the project folder

igpprojects/
- **waddenzee/**
  - **4_predict**
    - **igpPredictWaddenzee2015SimulatePointEpochList.m** (for demo only)
    - **igpPredictWaddenzee2015.m**
    - **igpExportWaddenzee2015.m**

The export can be in .png and/or csv format, as desired, and set in the script. The export is only compatible with STM files generated with the prediction module, so not for general STM files.

9. Output

`9_output` is used for the interactive plotting and inspection of space time matrices.

igpprojects/
- **waddenzee/**
  - **9_output**
    - **stmplot_examples.m**
    - **…**

It contains a matlab scripts with examples of top level plotting functions and scripts to generate the plot backgrounds with their output. It is possible to do the plotting in `9_ouput` on a different computer (e.g. laptop/desktop with GUI and graphics screen) that the first four steps (batch processing on a cluster).

[EOF]