

IMBIE Phase II

–Software Documentation–

isardSAT Reference: ISARD_ESA_IMBIE2_SDD_549
Issue: 1.a

Prepared by: Gorka Moyano and Mark Pattle
Reviewed and Approved by: Mònica Roca
25 May 2017
Activity: IMBIE Phase II

This page has been intentionally left blank

Change Record

Date	Issue	Section	Page	Comment
25 May 2017	1.a	all	all	Initial Issue

Distribution List

Company	Name
ESA	Marcus Engdahl
isardSAT	Gorka Moyano Mònica Roca
University of Leeds	Andrew Shepherd Kate Briggs

Table of Contents

1	INTRODUCTION	5
1.1	SCOPE AND PURPOSE.....	5
1.2	APPLICABLE DOCUMENTS.....	5
2	SOFTWARE REQUIREMENTS SPECIFICATION	6
2.1	SCOPE	6
2.2	REQUIREMENT NUMBERING CONVENTIONS.....	6
2.3	FUNCTIONAL REQUIREMENTS	6
2.4	SOFTWARE REQUIREMENTS	8
3	SOFTWARE DESIGN	10
3.1	SCOPE	10
3.2	SOFTWARE ARCHITECTURAL DESIGN	10
3.2.1	<i>Processor overview</i>	10
3.2.2	<i>Software components</i>	10
3.3	SOFTWARE DETAILED DESIGN	11
3.3.1	<i>Data flow diagram</i>	11
3.3.2	<i>Reading Input Files</i>	13
3.3.3	<i>Data Normalization</i>	13
3.3.4	<i>Calculating Per-Group Averages</i>	13
3.3.5	<i>Calculating Cross-Group Averages</i>	14
3.3.6	<i>Outputs</i>	14
4	SOFTWARE VERIFICATION PLAN AND REPORT.....	15
4.1	SOFTWARE VERIFICATION PLAN.....	15
4.2	SOFTWARE VERIFICATION REPORT	16
4.2.1	<i>Output Data</i>	16
4.2.2	<i>Comparison Statistics</i>	16
4.2.3	<i>Comparison Plots</i>	17
ANNEX A:	IMBIE 2012 DATA.....	20
	GRAVIMETRY DM/DT DATA	20
	MASS BUDGET DM/DT DATA	20
	RADAR ALTIMETRY DM/DT DATA	20
	LASER ALTIMETRY DM/DT DATA	20
	GRAVIMETRY METADATA	21
	MASS BUDGET METADATA.....	21
	RADAR ALTIMETRY METADATA.....	21
	LASER ALTIMETRY METADATA	21

1 Introduction

1.1 Scope and purpose

This document describes the IMBIE software item in terms of its requirements specification, its design and its verification.

Each one of these sections will be presented as a chapter of this document.

1.2 Applicable Documents

- AD. 1 "Proposal for the ESA-NASA Ice sheet Mass Balance Inter-comparison Exercise (IMBIE) Phase II".
- AD. 2 "SW User Manual", ISARD_ESA_IMBIE2_SUM_533, issue 1.a, 09 May 2017.

2 Software Requirements Specification

2.1 Scope

The scope of this chapter is to describe the functional and software requirements applicable to the IMBIE software item.

2.2 Requirement Numbering Conventions

In this section, the requirements are identified by an annotation in the left margin:

SRS-IMBIE2-CC-p.p-nn

Where:

- CC is a 2-letter category identifier (such as FN for functional).
- p.p is the number of the paragraph containing the requirement.
- nn is a numerical identifier, initially incrementing in steps of 10.

Paragraphs without such annotation provide information.

Comments, if any, appear in an oblique typeface, indented between horizontal rules, as in shown in this paragraph.

Information for the reader, not forming part of the specification, is formatted in this style.

2.3 Functional Requirements

This section will contain a low level description of the functional requirements.

SRS-IMBIE2-FN-2.3-10

The processor shall generate cumulative changes in the mass per ice sheet determined from a reconciliation of measurements acquired with one of the following techniques: Altimetry, Gravimetry, Mass Budget.

SRS-IMBIE2-FN-2.3-20

The processor shall accept as input dM/dt data measured with any of the following techniques: Altimetry, Gravimetry or Mass Budget.

SRS-IMBIE2-FN-2.3-30

The processor shall accept as input many datasets per technique, overlapping or not in time.

SRS-IMBIE2-FN-2.3-40

The processor shall accept as input data computed using either of the following sets of grounded ice area and drainage basins definitions: Zwally or Rignot.

SRS-IMBIE2-FN-2.3-50

The processor shall generate and format the outputs in the form of plots and tables.

SRS-IMBIE2-FN-2.3-60

The interface of the executable binary file shall be in the form:

[processor executable] [configuration file]

The configuration file will contain inputs of the processor, such as the input data path or processing options.

SRS-IMBIE2-FN-2.3-70

The input data path shall be indicated in the configuration file.

SRS-IMBIE2-FN-2.3-80

The output path shall be indicated in the configuration file.

Examples of output files that will be generated are plots and tables.

SRS-IMBIE2-FN-2.3-90

All tuneable processor parameters shall be configurable without the need to change the software.

SRS-IMBIE2-FN-2.3-100

The processor shall print traces to standard output giving information about the current step of the processing.

SRS-IMBIE2-FN-2.3-110

The processor shall exit with a return code.

The return code meaning:

- *Successful termination: 0.*
 - *Failed termination: not 0.*
-

2.4 Software requirements

This section will contain a low level description of the software requirements.

SRS-IMBIE2-SW-2.4-10

The processor shall be written in Python.

SRS-IMBIE2-SW-2.4-20

The Python version shall be 3.5 or newer.

SRS-IMBIE2-SW-2.4-30

The processor shall use permissive licence open source packages, compiler and tools unless duly justified.

Examples of permissive licence open source packages are BSD and LPGL.

SRS-IMBIE2-SW-2.4-40

All code shall be written in a portable way across platforms and OSs.

Code shall be tested in the following platforms: Ubuntu 16.04, Windows 7.

SRS-IMBIE2-SW-2.4-50

Every software delivery shall be identified by a unique and sequential version number.

V01.00, V01.01, V02.00, etc.

The first two digits indicate the major version number, while the second two digits are the minor version number.

SRS-IMBIE2-SW-2.4-60

All software deliveries shall not contain patches.

SRS-IMBIE2-SW-2.4-70

All software shall be kept under strict Configuration using a Configuration Management tool.

Examples of Configuration Management tools are SVN and GIT.

SRS-IMBIE2-SW-2.4-80

The target machine shall have the following specifications: 2 GB of RAM and a processor of 2.4 GHz.

A machine with a worse processor can also be used, but the time performance will be worse.

SRS-IMBIE2-SW-2.4-90

External libraries shall be used to use functions that are not included in the standard packages.

Example of libraries used:

- *numpy.*

A thorough list of the used libraries is described in AD. 2.

SRS-IMBIE2-SW-2.4-100

The type of the variables read from input files shall be double.

3 Software Design

3.1 Scope

The scope of this chapter is to describe the IMBIE software architectural design and the software detailed design.

3.2 Software architectural design

3.2.1 Processor overview

The purpose of the IMBIE processor is to discover and parse data from IMBIE contributions in order to collate, process, and analyse dM/dt time-series for ice sheets in Antarctica and Greenland. The input dM/dt time-series are normalised and combined to provide per-method and cross-method average dM/dt and dM time-series for each geographic region.

These output time-series are plotted and tabulated by a range of methods, in order to present the results to the user.

3.2.2 Software components

The IMBIE processor architecture is formed from a number of different types of component, which are defined here:

- A **Data Manager** searches a target directory for IMBIE submissions and their associated data files. For each data file discovered, it creates a *Data Access Object* to read it.
- A **Data Access Object** is an object that reads a single input file, and parses the contents into individual *data series object* for each geographic region for which the file provides data.
- A **Data Series Object** is a single time-series of data, belonging to a particular geographic area and period of time. A series represents $dM/dt(t)$ data. Each series has a number of metadata parameters, such as geographic region, the methodology used to produce the data, or the name of the contributor who submitted it.
- A **Data Collection Object** is an object that contains a sequence of *data series object*. The collection provides methods to filter its contents into subsets, which will be returned as a new collection. The collection can be filtered by values of the metadata parameters of its members.
- A **Function** performs an operation on a *data collection object*. A function may either transform every series in the input collection to produce an output collection of the transformed series, or it may apply a reduction operation to the input collection, creating a single series as output. Examples of reduction operations are summation or averaging of the input collection.

- An **Analysis Function** provides visual analysis of the input or output data. An analysis function might organise data into a table, or render it as a plot or other visualisation.

Figure 1 shows an abstract diagram of the SW components.

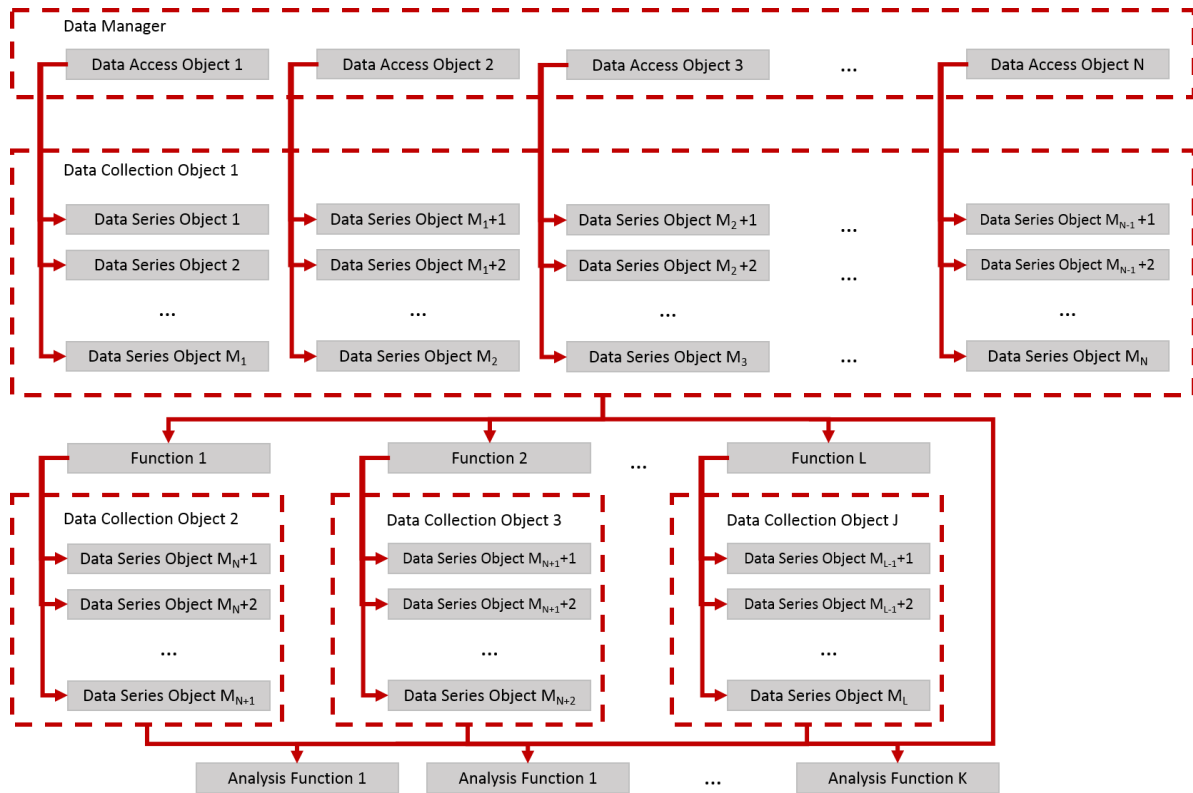


Figure 1 Software components general diagram

3.3 Software detailed design

3.3.1 Data flow diagram

The diagram below shows the processing flow of the IMBIE processor, from reading and normalising the contributions, through the processing chain, to the generation of the output products.

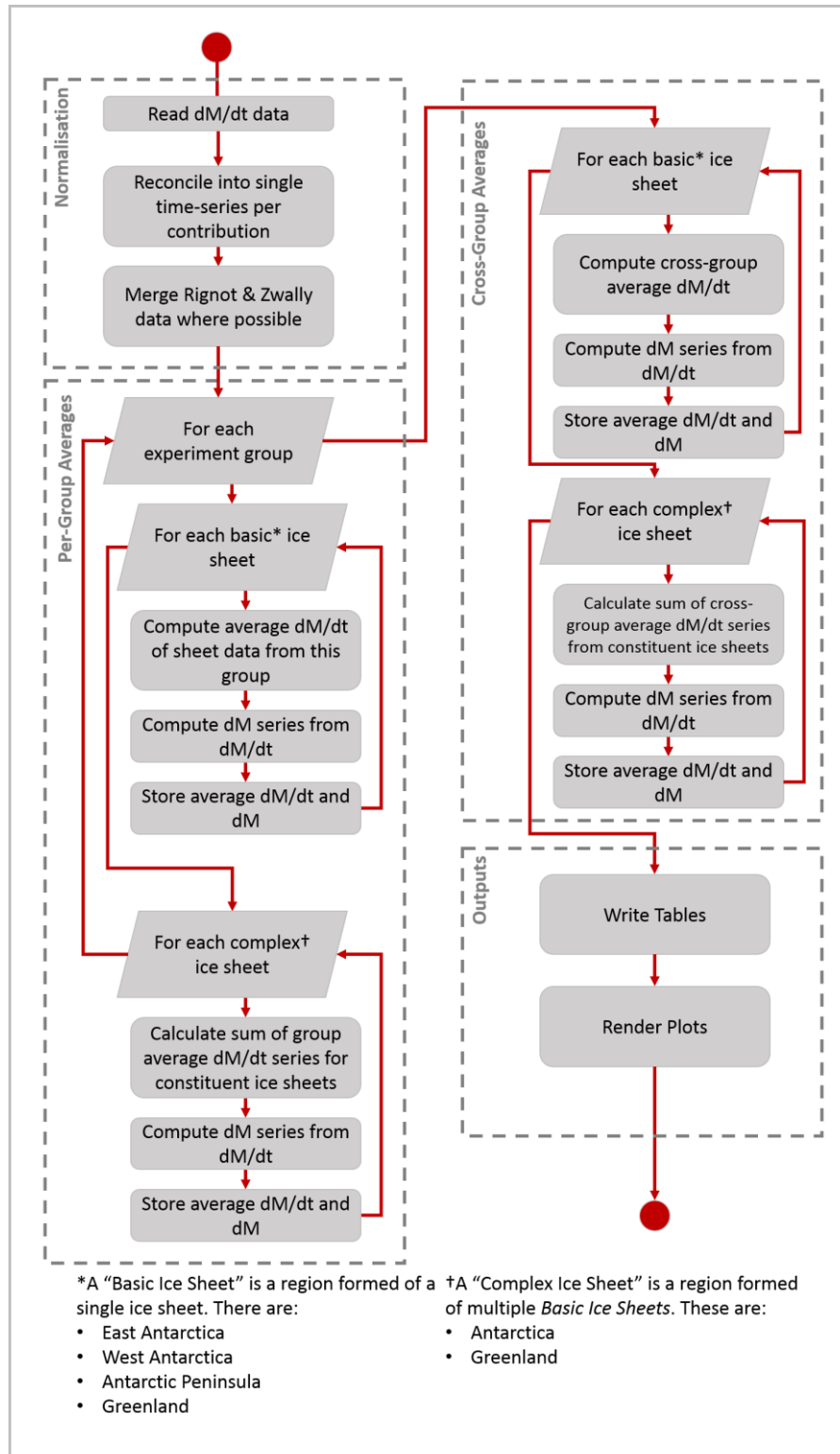


Figure 2 Data flow diagram of the IMBIE processor

The individual steps displayed in the flow chart above are explained in further detail below.

3.3.2 Reading Input Files

Each participant's submission file is read by a data access object. The DAO parses the contents of the CSV file, in which each row is a single datum, with a dM/dt value, a time, an error margin, and is of a particular geographic region. The DAO reads every row of the file and organises the results into a number of time-ordered data series, one for each geographic region that the file provides values for. These series are then inserted into the master data collection.

3.3.3 Data Normalization

Once the data has been read, the first action applied is the normalization of the data. This is a three-step process.

3.3.3.1 Converting dM/dt Time Series

The file format for dM/dt data stipulates that each datum should contain both a start time and an end time, but many of the operations which need to be performed on the data cannot be computed with this ambiguity. Therefore, the first action applied to the input data is to resolve the dM/dt data into series with a single time-value per entry, representing the estimated dM/dt at that particular epoch. The data are also interpolated into monthly intervals, using a step of 1/12th of a year instead of calendar months.

3.3.3.2 Amalgamating Basin Data to Ice Sheets

Where contributors have not provided data at an ice-sheet scale, it is sometimes possible to compute this if the contributor has provided values for all the drainage basins within the ice sheet. This is done by taking the sum of all the per-basin time-series provided by a user for a particular region.

3.3.3.3 Merging Rignot and Zwally Definitions

The geographic regions used by participants are defined by an identifier that either belong to the Zwally or the Rignot set of drainage basin definitions. In some cases, users have provided both Rignot and Zwally-based time series for an ice sheet. Since the geographic difference between Zwally and Rignot ice sheets is very small, it is possible to resolve this by taking the average of the two definitions.

3.3.4 Calculating Per-Group Averages

Once the initial normalization of the data has been performed, the first step of the main computation process is to calculate ice-sheet and regional average time-series for each experiment group.

3.3.4.1 Ice Sheets

For each of the basic Ice Sheets (East Antarctica, West Antarctica, Antarctic Peninsula and Greenland), an average dM/dt time-series is computed for each experiment group. This average time-series runs from the earliest data in the group to the latest. At each epoch within that range, the average of all the inputs is taken, with any series that does not cover that time being ignored.

Once the dM/dt average has been computed for a group, this can be integrated to find the dM series. The integration constant applied is defined by the date given in the 'align_date' configuration option.

3.3.4.2 Regions

Once the ice sheet averages for an experiment group has been calculated, the region dM/dt time-series can be computed. Regions are geographic areas formed of multiple ice sheets, such as Antarctica, formed of East Antarctica, West Antarctica, and Antarctic Peninsula. These are found by taking the sum of the average dM/dt time-series of the constituent ice sheets at each epoch. The summed time-series runs from the latest start-time to the earliest end-time of the contributing averages.

Once the time-series dM/dt for a region has been computed, it is then integrated to find the dM time-series. The integration constant applied is defined by the date given in the 'align_date' configuration option.

3.3.5 Calculating Cross-Group Averages

Once the per-group averages are computed, the cross-group averages can be calculated. This is similar to the process for calculating the per-group averages, but instead of computing the averages of the contributions for a particular group, the average of the group-averages is calculated. Region cross-group time-series can then be calculated by summing the cross-group ice-sheet averages.

3.3.6 Outputs

Once the data computation has been performed, the results are rendered by Analysis Functions. These may express the data in the form of a table, or as a plot.

3.3.6.1 Tables

Tables are generated by table objects. Each class of table object is designed to render a particular table, and ingests the relevant data. The information is then automatically tabulated, and exported in either CSV, plain-text, or HTML format.

3.3.6.2 Plots

Plots are rendered by a plotting object. This service draws a number of different types of plot, including dM/dt and dM time-series per region and ice sheet, both with inter- and intra-group comparisons. Other plots include comparisons of values from Rignot and Zwally basin definitions, Error-bar plots of dM/dt values, and time-coverage plots. The plots can be produced in several different formats, which can be chosen via a configuration parameter.

4 Software Verification Plan and Report

4.1 Software Verification Plan

4.1.1.1 Verification Scenario Description

The verification of the IMBIE processor is performed by comparison to the results of the IMBIE 2012 analysis, which were produced with a version of the code written in IDL code. The processor is provided with a copy of the IMBIE 2012 input data, which has been reformatted to comply with the input file format of the IMBIE processor – this is described in section 4.1.1.2.

In order for the IMBIE processor to match the behavior of the original IDL script used in 2012, a specific set of configuration parameters must be used. These are listed and explained in section 4.1.1.3.

4.1.1.2 Input Data

The input data for the verification are CSV files of the IMBIE 2012 experiment tabulated to match the input format of the IMBIE processor. These data are provided in this document in Annex A. Additionally, metadata files are required in order for the files to be loaded by the IMBIE processor. These are also provided in Annex A.

The data included in these files are split into four experiment groups – Radar Altimetry, Laser Altimetry, Mass Budget and Gravimetry. One file is provided per experiment group, and the data consist of variably-spaced time-series of dM/dt figures (of up to 12 records per year) and associated error margins. Collectively, the time-series cover the period from 1992 to 2011, however some experiment groups do not contain figures for this entire date-range.

4.1.1.3 Configuration Parameters

The configuration used for the software verification is provided below.

```
input_path "~/path/to/input/folder"
output_path "~/path/to/output/folder"
combine_method "imbie1"
export_data True
average_nsigma 3
enable_la_group True
```

Explanations of these parameters and their values are provided below:

- *combine_method "imbie1"* – this setting instructs the IMBIE processor to use a time-series averaging method that reproduces the behaviour of the IMBIE 2012 analysis.
- *export_data True* – this instructs the IMBIE processor to export the final output dM/dt time-series as CSV files, which will be compared against IMBIE 2012 output data to validate the behaviour of the processor.
- *average_nsigma 3* – input data points that are further than three standard deviations away from the mean value will be excluded from the averaged time-series. This replicates the behaviour of the IMBIE 2012 analysis.
- *enable_la_group True* – the IMBIE 2012 analysis divided input data into four experiment groups, including the LA (Laser Altimetry) group. By default, this group is not used by the IMBIE processor. This switch enables the LA group in the processor.

4.2 Software Verification Report

The behaviour of the IMBIE Processor has been verified by comparing its outputs to the output data of the IMBIE 2012 analysis. This has been performed for each of the four basic ice sheets – East Antarctica, West Antarctica, Antarctic Peninsula and Greenland.

4.2.1 Output Data

By using the configuration profile specified in Section 4.1.1.3, the processor will be instructed to produce output data files. These files will be written to the output directory and will contain the final dM/dt time-series and associated error margins for each ice sheet, at a rate of 12 data points per year. The data are stored in CSV format.

These data can be compared against equivalent data files from the IMBIE 2012 Analysis, in order to validate the IMBIE processor.

4.2.2 Comparison Statistics

For each ice sheet, the mean dM/dt values of the IMBIE processor output can be compared against the mean dM/dt value produced by the IMBIE 2012 Analysis. The Table 1 below shows this comparison:

Table 1 Ice sheet comparison statistics

Ice Sheet	Expected mean dM/dt (Gt/yr)	Actual mean dM/dt (Gt/yr)
Greenland	-151.231	-150.119
East Antarctica	16.984	17.371
West Antarctica	-65.162	-65.066
Antarctic Peninsula	-19.704	-19.469

The RMS differences between the output of the IMBIE processor and the IMBIE 2012 results for each of the ice sheets are:

- East Antarctica: 9.175 Gt/yr
- West Antarctica: 3.223 Gt/yr
- Antarctic Peninsula: 1.584 Gt/yr
- Greenland: 3.207 Gt/yr

4.2.3 Comparison Plots

Plots of the IMBIE processor output time-series, compared against the IMBIE 2012 analysis results for each of the four ice-sheets are shown below, in Figure 3, Figure 4, Figure 5 and Figure 6. Each figure shows the expected and actual dM/dt time-series for one ice sheet. Each time-series is drawn as a central coloured line, surrounded by a shaded band of colour. This band indicates the associated error margin calculated by the processor. Where these regions intersect (and the IMBIE processor output matches the IMBIE 2012 results), a third colour is produced by the mixing of the two shaded bands.

As can be seen in the figures, there are some minor differences between the data produced by the IMBIE processor, and those of the IMBIE 2012 analysis. These discrepancies occur at the beginning or end of one of the contributing input time-series, and is believed to be caused by differences between the interpolation methods used by the IDL language (the IMBIE 2012 code) and those used by *scipy* (the python library of scientific and numerical functions used in the IMBIE processor).

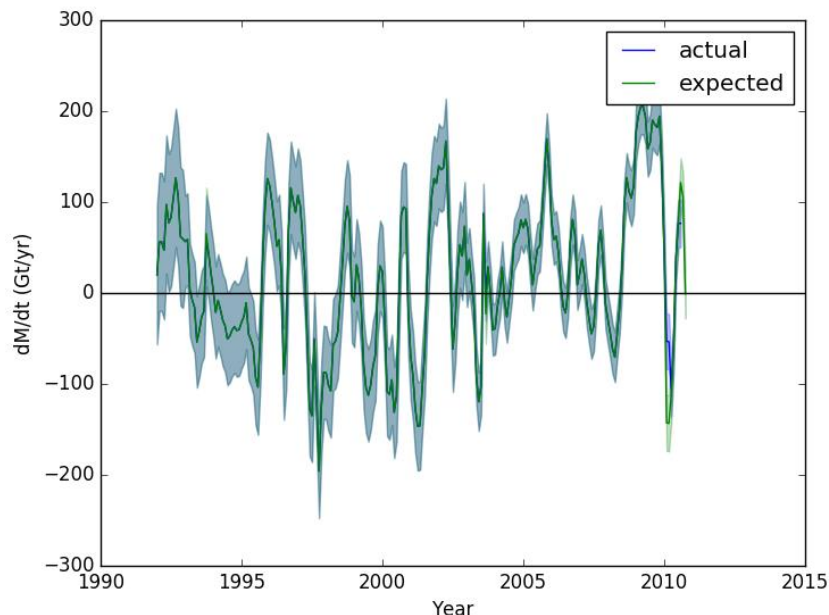


Figure 3 East Antarctica Verification Comparison

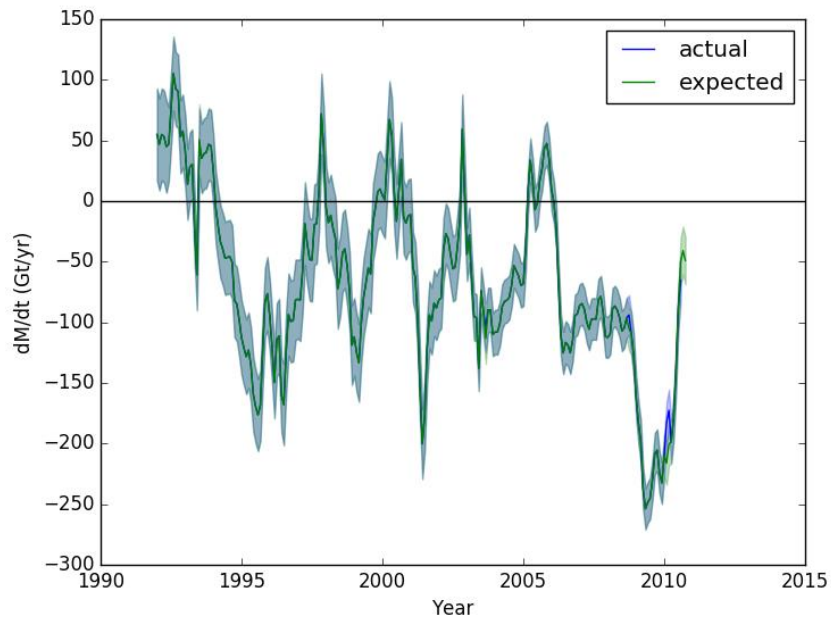


Figure 4 West Antarctica Verification Comparison

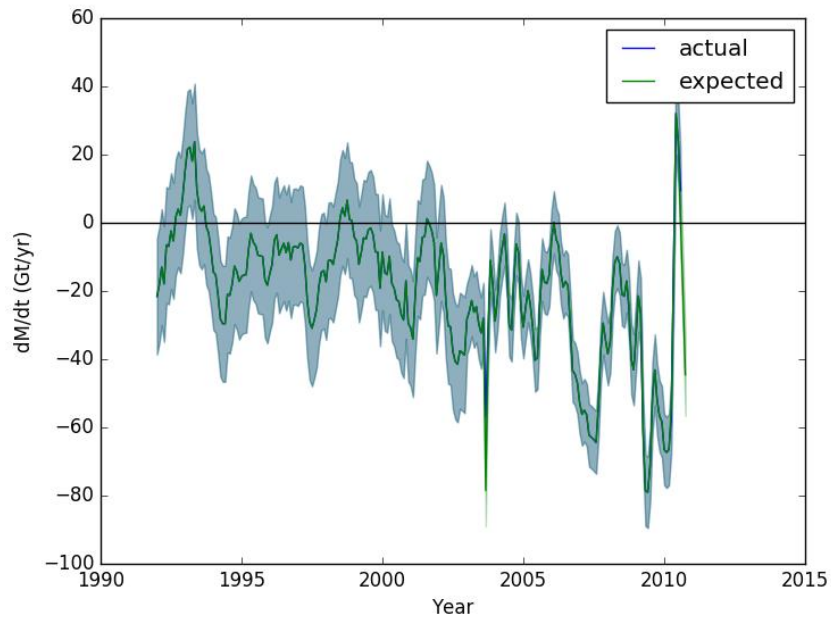


Figure 5 Antarctic Peninsula Verification Comparison

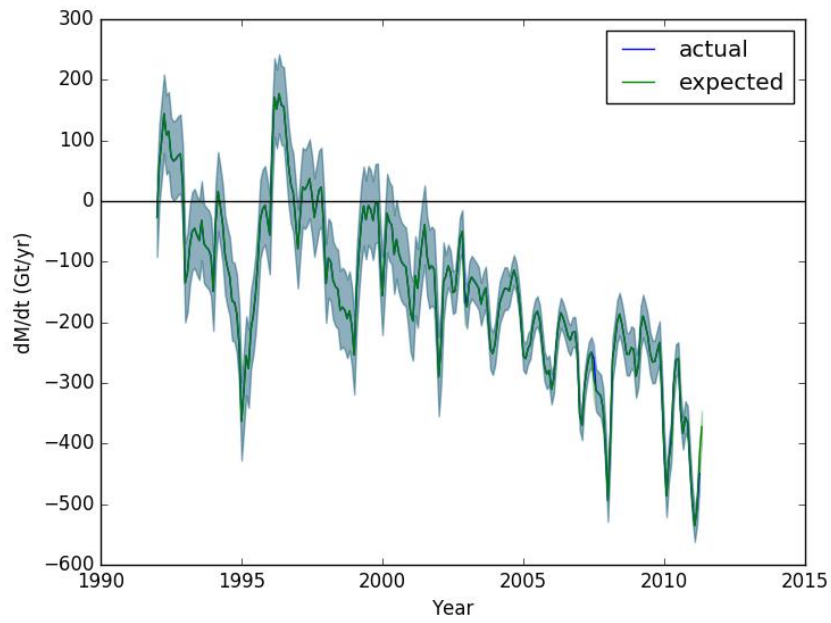


Figure 6 Greenland Verification Comparison

ANNEX A: IMBIE 2012 DATA

Gravimetry dM/dt data

File path: data/GMB/rate.csv



rate.csv

Mass Budget dM/dt Data

File path: data/IOM/rate.csv



rate.csv

Radar Altimetry dM/dt data

File Path: data/RA/rate.csv



rate.csv

Laser Altimetry dM/dt data

File Path: data/LA/rate.csv



rate.csv

Gravimetry Metadata

File path: data/GMB/.answers.json



Mass Budget Metadata

File path: data/IOM/.answers.json



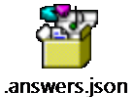
Radar Altimetry Metadata

File path: data/RA/.answers.json



Laser Altimetry Metadata

File path: data/LA/.answers.json



End of document