

Data Assimilation in Elmer/Ice

Samuel Cook and Fabien Gillet-Chaulet

All (or most of) the computations presented in this paper were performed using the GRICAD infrastructure (<https://gricad.univ-grenoble-alpes.fr>), which is supported by Grenoble research communities.



The classic modelling approach

Initial datasets gathered at variety of times

Static inversion to get snapshot of unknown field(s) (typically basal friction)

Relaxation run to dissipate model artefacts and inconsistencies between datasets

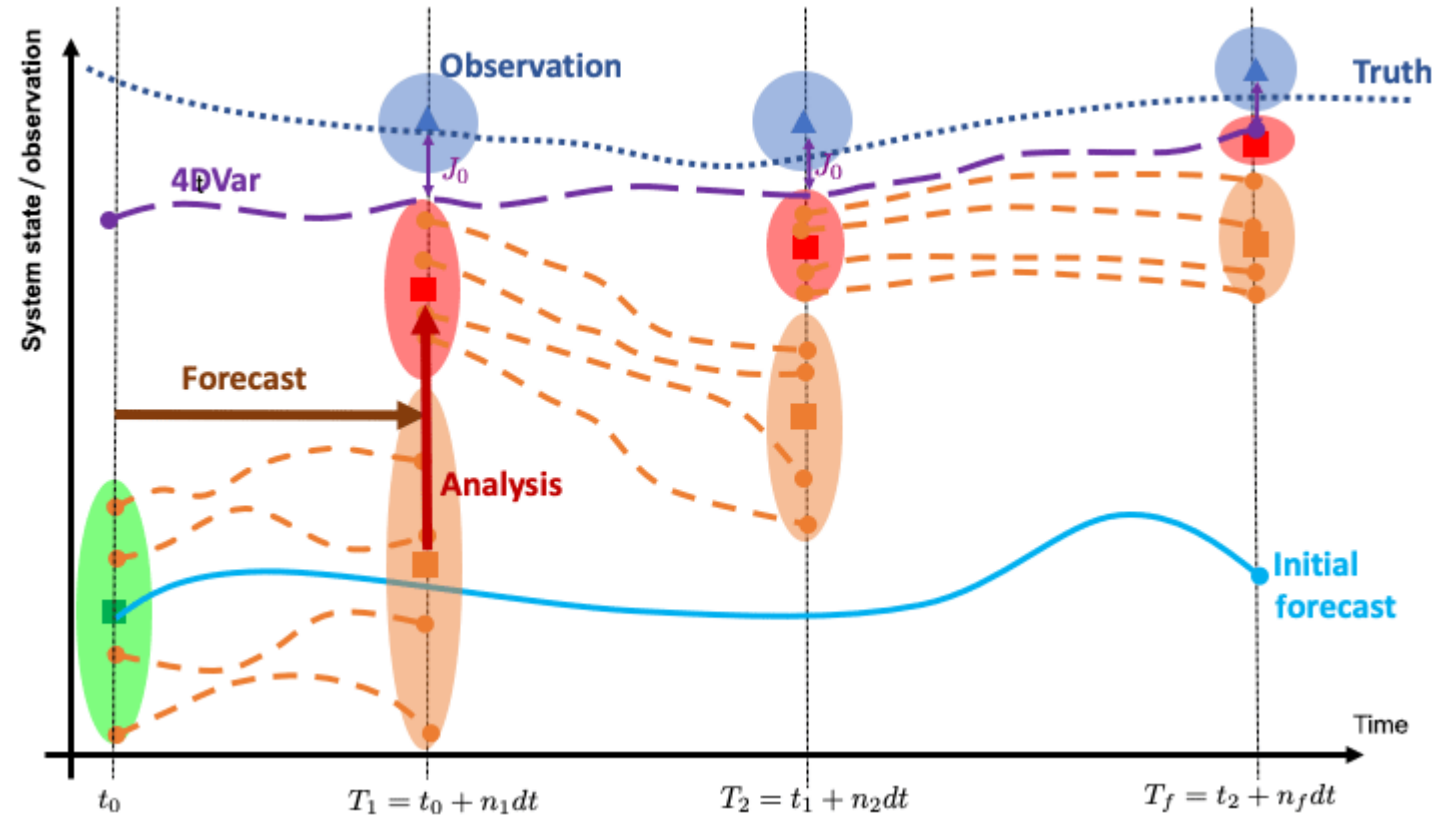
=> Time-consuming, static, and eventual model glacier may differ substantially from real-world equivalent

What is assimilation?

Correcting model state through use of observations

We do this using an Ensemble Kalman filter

Use Elmer/Ice for the forecast steps and PDAF for the analysis (using an error subspace ensemble transform Kalman filter (ESTKF))



Gillet-Chaulet (2020)

Implementation in Elmer/Ice

New external library to perform analysis step : PDAF

Coupled in an offline mode

Routines written to perform I/O between Elmer and PDAF

Nearly zero changes to Elmer/Ice source code

Implementation in Elmer/Ice

Data assimilation works in 2D flowline and in 3D simulations

In 2D, I/O can be using .result files or NetCDF files

In 3D, I/O only using NetCDF files

Updates to GridDataReader.F90 and SaveGridData.F90

If using NetCDF, overall Simulation-section keyword : *Use NetCDF = Logical True*

General workflow

Run Elmer/Ice as forward model normally, including NetCDF output, if required

Run PDAF to perform analysis step, reading Elmer .result files or .nc output

Run Elmer/Ice again, reading PDAF .result files or .nc output

Continue as required

GridDataReader.F90

Minor update to allow solver to be called in a loop over several different files

Three new solver keywords :

Ensemble simulation = Logical True

Ensemble Size = Integer ...

Iterator = File ...

SaveGridData.F90

Major update to allow output in NetCDF format

Functions identically to existing version of SaveGridData, but with additional NetCDF format option

Can output NetCDF of entire mesh or of a specific boundary

Other matters and further work

Analysis-step SIF requires a new solver with specific options to handle Elmer/PDAF interface

Documentation exists in notes ; will be written up properly....

Mostly just about giving it a load of variable names and file paths

Initial ensemble generation in 3D needs to be considered further

We also hope to provide some general post-processing scripts for users

Code availability

All code is on my Grenoble Gitlab account, as well as my personal Github

Ask me if you want to have a look at it (samuel.cook@univ-grenoble-alpes.fr)

Though I might not let you until I've submitted the paper we're writing (shortly, I hope!)