IGE
Institut
des géosciences
de l'environnement

Updates on some Elmer/Ice recent developments

Fabien Gillet-Chaulet

anr agence nationale de la recherche AU SERVICE DE LA SCIENCE
TiPACCs *Tipping Points in Antarctic Climate Components*
Protect CRYOSPHERE & SEA LEVEL

OSUG
cnrs
UGA Université Grenoble Alpes
IRD Institut de Recherche pour le Développement FRANCE
Grenoble INP | UGA

- *I/O with XIOS*
- *(non-linear) Weertman sliding law in* **IncompressibleNSVec**
- Spatial covariance modelling

- ***I/O with XIOS***
    - ***Update on the material presented for ElmerIce User meeting in Dec. 2022***

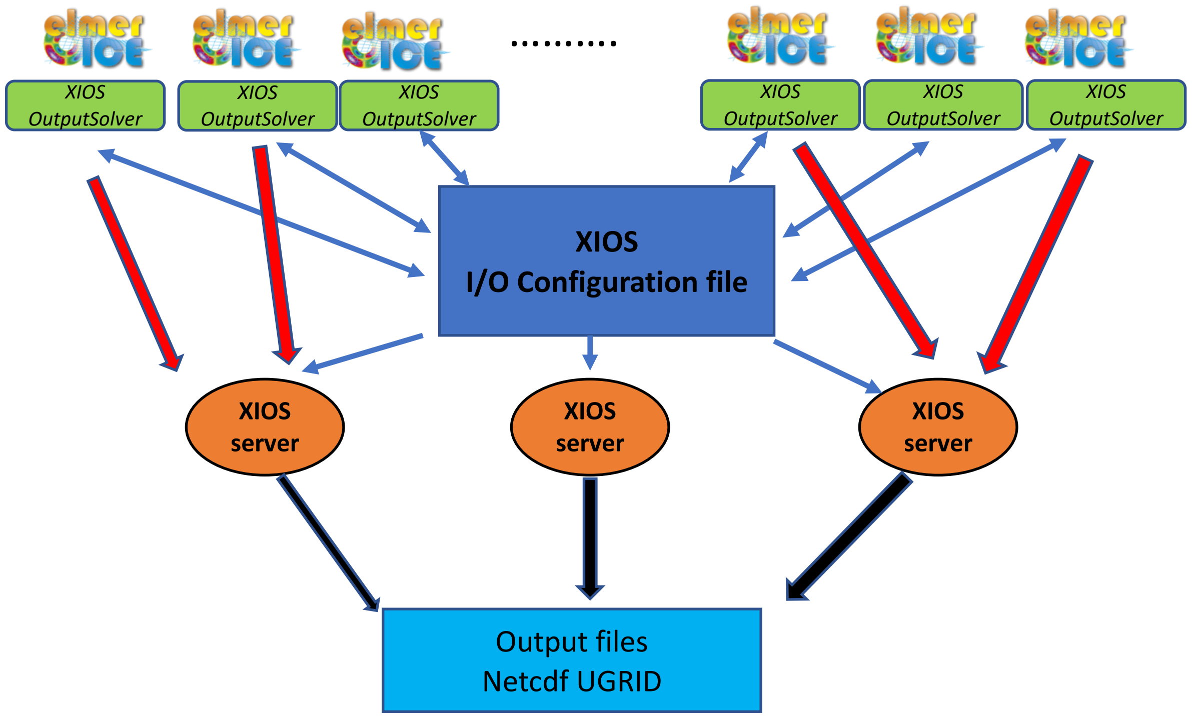
- *(non-linear) Weertman sliding law in* **IncompressibleNSVec**

- Spatial covariance modelling

I/O with XIOS

- *XIOS* is a an **external library for I/O** used in several european climate models (e.g. NEMO, LMDz)
- Objectives of *XIOS* are to adress the following challenges for climate data production
  - *Flexibility in management of I/O and data/metadata definition*
  - *Efficient production on supercomputer parallel file systems*
  - *Complexity and efficiency of post-treatment chain to be suitable for distribution and analysis*

- Interface with XIOS as a new Elmer/Ice solver (***XIOSOutputSolver***)
- In detached mode allocate dedicated cpus for *XIOS*
  - *> mpirun –np XX ElmerSolver_mpi : -np YY xios_server.exe*


XIOS OutputSolver *XIOS OutputSolver* *XIOS OutputSolver* .... *XIOS OutputSolver* *XIOS OutputSolver* *XIOS OutputSolver*

XIOS
I/O Configuration file

XIOS server **XIOS server** **XIOS server**

Output files
Netcdf UGRID

XIOS OutputSolver

- Update configuration from parameters defined in the .sif
- Send the grid definition
- Expose part of the data at each time-step

XIOS I/O Configuration file

- Describe the incoming data flow from Elmer/Ice
- Describe the workflow applied to the incoming data flow:
  - arithemtic filters: C=A+B
  - temporal filters : yearly-averaged values
  - spatial filters : sum over the grid, regridding
- Describe the output files and their content

I/O configuration overview

- SIF file :

```
Solver ....
  Exec Solver = After Timestep

  Equation = "XIOSOutPutSolve"
  Procedure = "ElmerIceSolvers" "XIOSOutputSolver"

 ....
  Keywords related to time unit system, time step, start date
 ...

! node and elem vars; e.g.
  Scalar Field 1 = String "h"
  Scalar Field 1 compute cell average = Logical True

  Scalar Field 2 = String   "acabf "

! Global Variables
  Global Variable 1 = String "time"

End
```

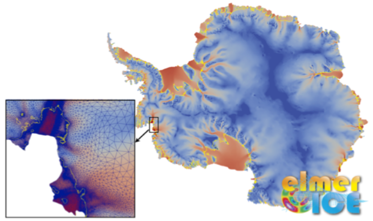
- XIOS xml config file :

```
<context id="elmerice">

<!-- calendar  -->
  <calendar type="NoLeap" time_origin="1995-01-01 00:00:00" />

<!-- fields  -->
  <field_definition>

    <field id="h"      standard_name="land_ice_thickness" unit="m" grid_ref="GridNodes"  operation="instant" />
    <field id="h_elem"  standard_name="land_ice_thickness" unit="m" grid_ref="GridCells"  operation="instant" />

    <field id="acabf"    standard_name="land_ice_surface_specific_mass_balance_flux" unit="m d-1"  grid_ref="GridCells"   operation= "average" />

    <field id="time" name="elmer_time" unit="d" grid_ref="ScalarGrid_sum"  operation="instant" />

     <field id="ismip6_acabf"    field_ref="acabf"    name="acabf"    unit="kg m-2 s-1" > this*$rhoi/$nsec_per_day </field>

   </field_definition>

<!-- files  -->
  <file_definition >

    <file_group id="file01" >
    <file id=   "file01" name= "MyFileName" convention="UGRID"  output_freq="1y"  >
        <field field_ref=" ismip6_acabf "  />
        <variable id="elmerversion" name="model_version" type="string"> elmer ice </variable>
    </file>

   </file_definition>

…

</context>
```

I/O with XIOS used in the most recent ElmerIceSheet configuartions

Codes, tools, files to run and process standard ISMIP6 simulations maintained by IGE:

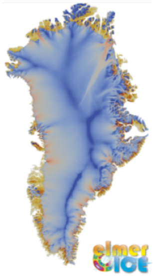
- https://github.com/pmathiot/ELMER_ISMIP6_Antarctica



ISMIP6-Antarctica-2300:

- 2 contributions:
  - IGE; J. Caillet
  - UTAS; C. Zhao
- **SSA** (shelves, 1km-50km)
- I/O-Post-Processing; Melt parameterisations

- https://gricad-gitlab.univ-grenoble-alpes.fr/gilletcf/elmerice_ismip6_gris



ISMIP6-compliant:

- IGE; F. Gillet-Chaulet
- CSC; T. Zwinger
- **SSA** (shelves)
- I/O-Post-Processing; Forced Front Retreat

I/O … an overview of the capabilities and limitations

- **Outputs in netcdf UGRID**
    - contain meta-data
    - Restartable (independent of number of partitions) => *UGridDataReader*
    - Viewable
        - QGIS : as mesh layer with the Crayfish plugin
        - Paraview: reader under development **=> download a recent nighlty build**
    - Can use many functionalities to manipulate netcdfs nco, cdo….
        - e.g. ncdiff => differences between netcdfs => compare simulations, anomalies
        - E.g. ncea => ensemble average
- **Temporal filters**
    - Compute time averaged values (or min, max , cumulative)
- **Calendar management**
- **Used/developped by relatively large community in climate models**

- **Restricted to 2D => 2D boundary of a 3D vertically extruded Mesh**
- **Configuration**
    - Need to read *XIOS* documentation and tutorials
- **Works with geographical coordinates (lon,lat)**
    - Module *ProjUtils* : (lon/lat) <=> (x,y)
    - Analytical implementation for polar stereographic north (Greenland) and south (Antarctica)
    - **Interface with fortran gis and *proj4* for other projections (UTM)**
    - *To see what to do for synthetic experiements…*
- **Calendar management**
    - a year is not a proper time unit (duration depends on the calendar)
    - Time step should be an integer value and a given fraction of the output frequency

> ncdump output_file.nc

```
dimensions:
        axis_nbounds = 2 ;
        Two = 2 ;
        nmesh2D_node = 221562 ;
        nmesh2D_edge = 633466 ;
        nmesh2D_face = 411905 ;
        nmesh2D_vertex = 3 ;
        time = UNLIMITED ; // (86 currently)
variables:
        int mesh2D ;
                mesh2D:cf_role = "mesh_topology" ;
                mesh2D:long_name = "Topology data of 2D unstructured mesh" ;
                mesh2D:topology_dimension = 2 ;
                mesh2D:node_coordinates = "mesh2D_node_x mesh2D_node_y" ;
                mesh2D:edge_coordinates = "mesh2D_edge_x mesh2D_edge_y" ;
                mesh2D:edge_node_connectivity = "mesh2D_edge_nodes" ;
                mesh2D:face_edge_connectivity = "mesh2D_face_edges" ;
                mesh2D:edge_face_connectivity = "mesh2D_edge_face_links" ;
                mesh2D:face_face_connectivity = "mesh2D_face_links" ;
                mesh2D:face_coordinates = "mesh2D_face_x mesh2D_face_y" ;
                mesh2D:face_node_connectivity = "mesh2D_face_nodes" ;
        float mesh2D_node_x(nmesh2D_node) ;
                mesh2D_node_x:standard_name = "longitude" ;
                mesh2D_node_x:long_name = "Longitude of mesh nodes." ;
                mesh2D_node_x:units = "degrees_east" ;
        float mesh2D_node_y(nmesh2D_node) ;
                mesh2D_node_y:standard_name = "latitude" ;
                mesh2D_node_y:long_name = "Latitude of mesh nodes." ;
                mesh2D_node_y:units = "degrees_north" ;
```

```
        double time(time) ;
                time:axis = "T" ;
                time:standard_name = "time" ;
                time:long_name = "Time axis" ;
                time:calendar = "noleap" ;
                time:units = "days since 1995-01-01 00:00:00" ;
                time:time_origin = "1995-01-01 00:00:00" ;
                time:bounds = "time_bounds" ;
        double time_bounds(time, axis_nbounds) ;
```

```
        float xvelmean(time, nmesh2D_face) ;
                xvelmean:standard_name = "land_ice_vertical_mean_x_velocity" ;
                xvelmean:units = "m s-1" ;
                xvelmean:mesh = "mesh2D" ;
                xvelmean:location = "face" ;
                xvelmean:online_operation = "instant" ;
                xvelmean:interval_operation = "1 yr" ;
                xvelmean:interval_write = "1 yr" ;
                xvelmean:cell_methods = "time: point" ;
                xvelmean:_FillValue = 1.e+20f ;
                xvelmean:missing_value = 1.e+20f ;
                xvelmean:coordinates = "time_instant mesh2D_face_y mesh2D_face_x" ;
```

```
// global attributes:
                :name = "ismip6_states_marv3.12_access1.3-rcp85-rhigh_1" ;
                :description = "Created by xios" ;
                :title = "Created by xios" ;
                :Conventions = "UGRID" ;
                :timeStamp = "2022-Nov-09 05:54:35 GMT" ;
                :uuid = "49aa435f-f62d-4b8e-81d3-0191d3ea4f75" ;
                :model_version = "Elmer/Ice v9.0 (Rev: 29fd3bf4)" ;
                :altitude_convention = "altitude reference against geoid EIGEN-EC4" ;
                :projection = "espg:3413" ;
```

- *(non-linear) Weertman sliding law in* **IncompressibleNSVec**
- *I/O with XIOS*
- Spatial covariance modelling

non-linear Weertman sliding law in IncompressibleNSVec

- Implemented a long time ago by Peter
- Previously prescribed as a USF: **Sliding_Weertman** in USF_Sliding.f90

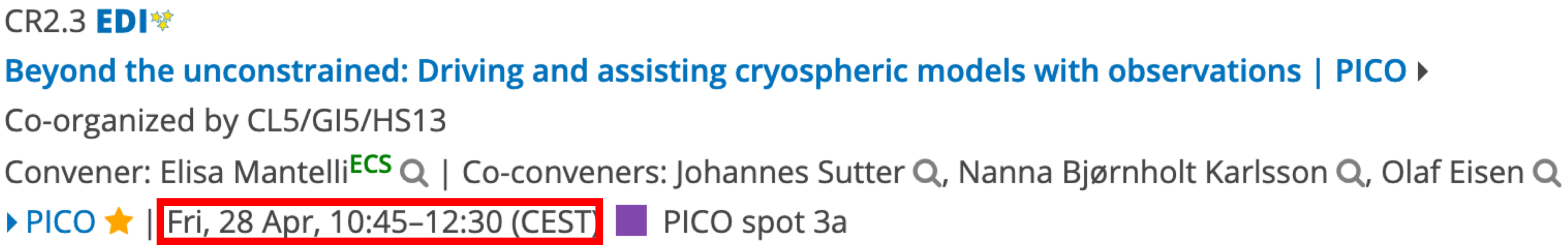
- **Latest elmerice branch** (762e8f2db – March 1rst 2023)
    - Implement **Newton method** for non-linear iterations
        - Shows expected speed-up in the **convergence of non-linear iterations**
        - Examples:
            - elmerice/Tests/Friction_WeertmanNewton2/
            - elmerice/Tests/Friction_WeertmanNewton3D
        - To see for GL applications where contact is tested inside the non-linear iterations loop
    - Update the **adjoint inverse method** to invert for the friction coefficient
        - No proof that it should be better than inverting for the effective friction, but might introduce a discontinuity between inversion and transient simulations
        - Examples:
            - elmerice/examples/Inverse_Methods/StokesWeertman
    - **Need to update the doc**.
    - Implement Coulomb–type friction law (Newton and inversion)?

```
Boundary Condition 5
 Name = "bottom"
 Body Id = 3

 !! Normal-Tangential coordinate system
  Normal-Tangential Velocity = Logical True
  Velocity 1 = Real 0.0

 Weertman Friction Coefficient = Variable alpha
  REAL procedure "ElmerIceUSF" "TenPowerA"
 Weertman Exponent = Real 0.3333333333333333333
! Cut-off such that argument is not smaller than this
 Friction Linear Velocity = Real 1.0e-4

 Slip Coefficient derivative = Variable alpha
  REAL procedure "ElmerIceUSF" "TenPowerA_d"

 Bottom Surface = Variable Coordinate 1
   Real MATC "-tx*tan(Slope)-1000.0"
End
```

- *(non-linear) Weertman sliding law in* **IncompressibleNSVec**

- *I/O with XIOS*

- **Spatial covariance modelling**
  - ***Come to see my PICO about regularisation in the Mass conservation method***



CR2.3 **EDI**

Beyond the unconstrained: Driving and assisting cryospheric models with observations | PICO

Co-organized by CL5/GI5/HS13

Convener: Elisa Mantelli^{ECS} | Co-conveners: Johannes Sutter, Nanna Bjørnholt Karlsson, Olaf Eisen

▸ PICO ★ | Fri, 28 Apr, 10:45–12:30 (CEST) ■ PICO spot 3a

  - **Codes in elmerice branch** since Jan. 2023 (ef1e0b1f1)
    - elmerice/Solvers/Covarianceutils:
      - **BackgroundErrorCostSolver.F90** => Data Assimilation (Regularisation)
      - **GaussianSimulationSolver.F90** => Generate random fields from the given covariance matrix
      - **CovarianceVectorMultiplySolver.F90** => Gaussian Filter

  - **No documentation, no automatic testing and no examples yet** in the distribution; just ask if interested…

Inverse methods from a Bayesian perspective: the linear gaussian case

We consider the following **inverse problem**:

Estimate the « true » **state of a system** $\mathbf{x}^t \in \mathbb{R}^n$, from a set of **observations** $\mathbf{y}^0 \in \mathbb{R}^m$ (in general *m* **<<** *n*), such that:

$$\mathbf{y}^o = \mathbf{H}\mathbf{x}^t + \epsilon^o$$

$\mathbf{H}$ is a linear **observation operator**

ϵ^o is the observation error (assumed unbiaised), with **covariance matrix** $\mathbf{R}$

We also have a **first estimate** $\mathbf{x}^b$ (from lab experiments, a climatology, a previous model forecast, …), such that :

$$\mathbf{x}^b = \mathbf{x}^t + \epsilon^b$$

ϵ^b is the **background error** (assumed unbiaised), with **covariance matrix** $\mathbf{B}$

The **Best Linear Unbiaised Estimation** (BLUE) is given by :

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}\left(\mathbf{y}^o - \mathbf{H}\mathbf{x}^b\right)$$

where the **Kalman gain** is given by $\mathbf{K} = \mathbf{B}\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\right)^{-1}$

After some calculations, it can be shown that **the same estimation minimises** the following **cost function**:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{y}^o - \mathbf{H}\mathbf{x})\mathbf{R}^{-1}(\mathbf{y}^o - \mathbf{H}\mathbf{x}) + \boxed{\frac{1}{2}(\mathbf{x} - \mathbf{x}^b)\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b)}$$

=> Can be seen as a “Regularisation” term

Comments:

=> Require to properly define the covariance matrices R and **B (and the background)**

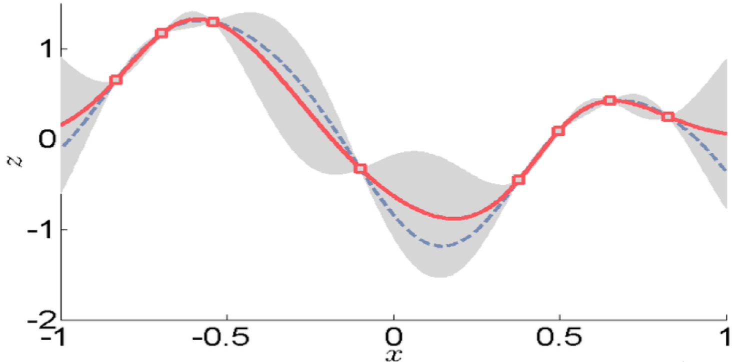
- **=> “scientific issues”** for properties that are very poorly constrained (friction)
- **=> “technical issues”** covariances matrices are full-rank matrices so in general you can not store them

=> in general no proof of optimality for non-linear, non-gaussian, biased cases

Overcoming these issues

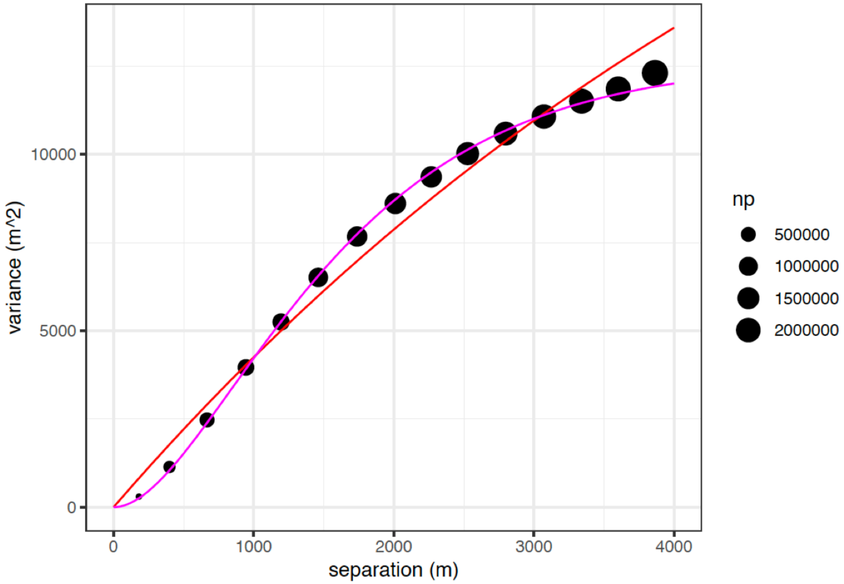
Scientific issues:

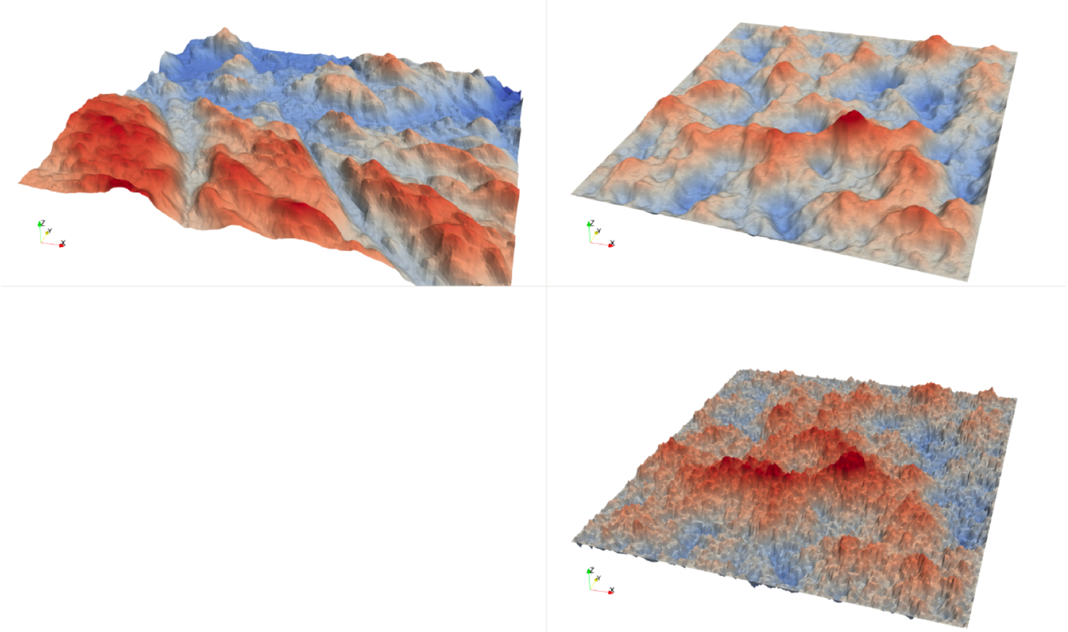
- Analogies with geostatistic:
  - Interpolation by **Kriging** requires to parametrized prior co-variances using standard correlation functions (Exponential, Gaussian, Matérn, …)



source: wikipedia

Correlation function (variograms)





Rq. in general real bed elevations are not gaussian, there is features (valleys, ..)

Overcoming these issues

Scientific issues:

- Analogies with geostatistic:
  - Interpolation by **Kriging** requires to parametrized prior co-variances using standard correlation functions (Exponential, Gaussian, Matérn, …)

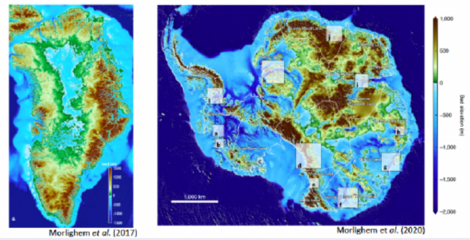
Numerical issues:

- You don't really need B, but an equivalent **operator** (the action of B on a vector)
  - **Diffusion operators** can be used to model a class of correlation functions from the Matérn family (cf e.g. Guillet et al., 2019)
  - It consist in iteratively solving a diffusion-type equation
    - => can be applied on unstructured meshes with the FEM
    - => relatively cheep and memory efficient

- I have implemented the possibility to compute B (and B-1 and L)
  - with **standard correlation functions (**and lapack routines to compute the inverse and square root)
    - **=>** restricted to **small serial problems** (~up to 10-20 knodes)
  - **with the diffusion operator approach** (based on Guillet et al. 2019)
    - => **can be used for large parallel simulations**

Application to the mass conservation method : my PICO summary

The mass conservation method



Morlighem *et al.* (2017)

Morlighem et al. (2020)

- is a method to interpolate radar-derived ice thickness data
- is used in the reference **BedMachine** products

=> is in Elmer/Ice since a while but has never really been used

is a **control method**

- The ice thickness, H, is solution of the **steady-state** continuity equation:

$$\boldsymbol{\nabla}(\bar{\boldsymbol{u}}H) = \dot{a}$$

- The optimal ice thickness H minimizes:

$$J(\bar{\boldsymbol{u}}, \dot{a}) = \frac{1}{2}(H - H^{obs})\boldsymbol{R}^{-1}(H - H^{obs}) + \frac{1}{2}R_{reg}$$

Objective: Test the sensitivity to the regularisation term $\boldsymbol{R_{reg}}$

- is often chosen as a **Tikhonov** regulatrisation term that penalizes spatial derivatives of H:

$$R_T(H) = \lambda \int_{\Omega} ||\nabla H||^2 \mathrm{d}\Omega$$

- In a **Bayesian** framework, regularisation is introduced from prior information about the solution:

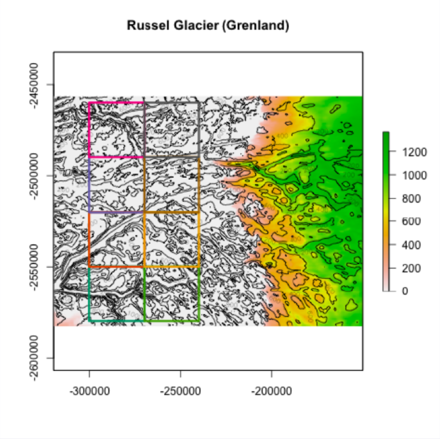
$$R_B(H) = (H - H^b)\boldsymbol{B}^{-1}(H - H^b)$$

Application to the mass conservation method

Synthetic twin experiments

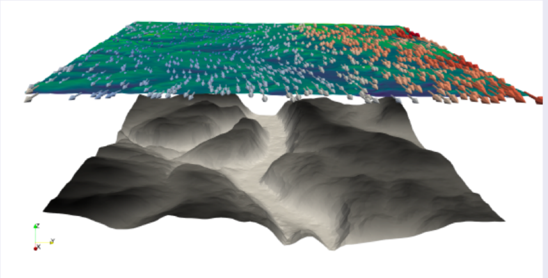
1. Take deglaciated beds

8 test cases: 30 × 30km boxes



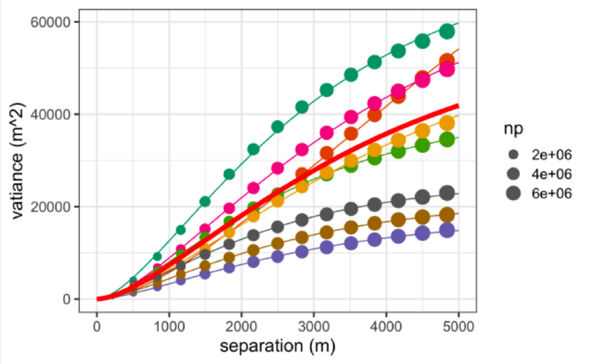
2. Add 1000 m of ice

Generate a perfect model solution



3. Regularisation $R_B(H)$

Parameterize $\boldsymbol{B}^{-1}$

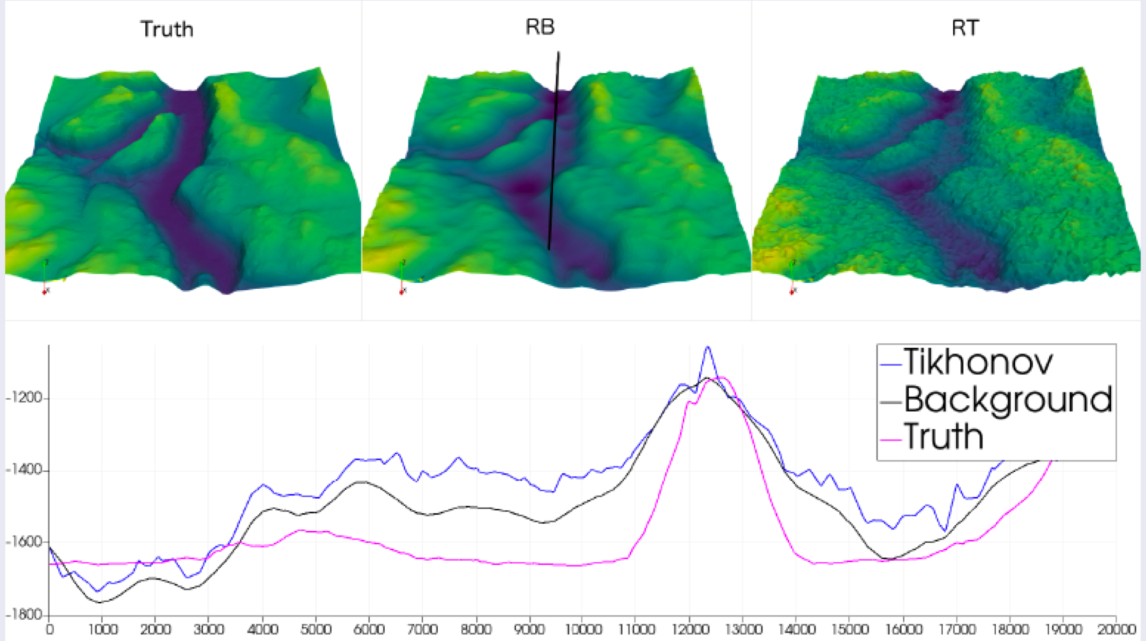


- **spatial correlation** is parametrised using standard correlation functions, as done for kriging
- Fit standard **Matérn covariance** functions from variograms using geostastical modelling tools
- $\boldsymbol{B}^{-1}$ is applied as a diffusion operator on the unstructured FE mesh (Guillet et al., 2019)

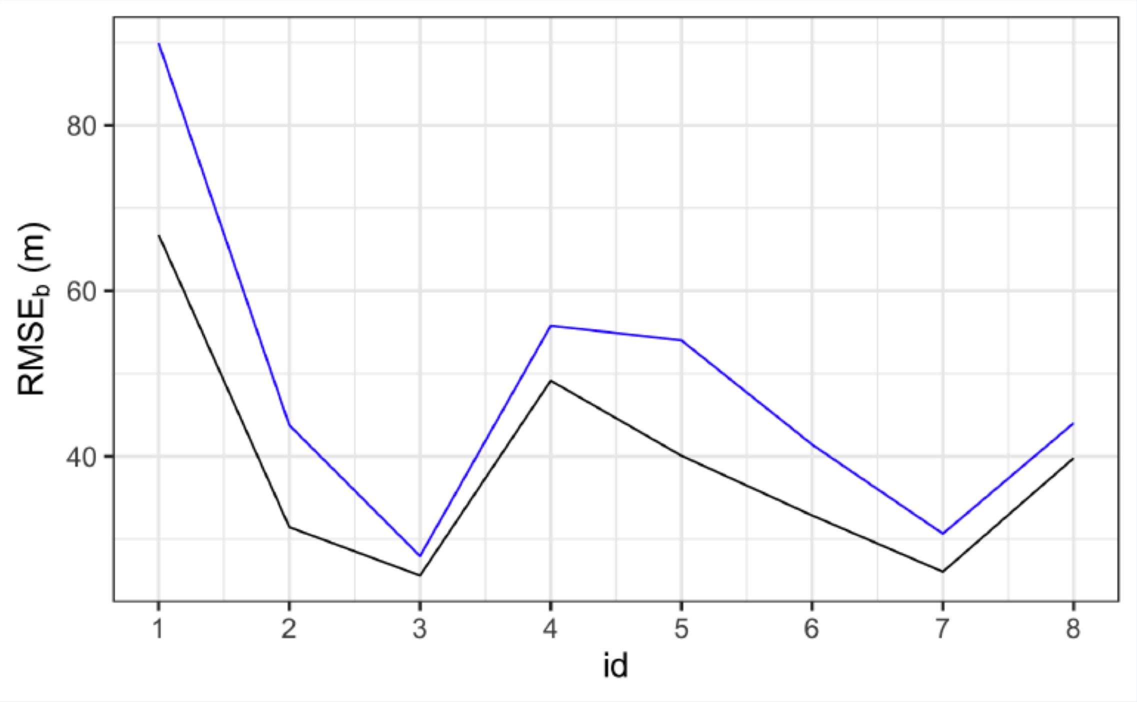
Application to the mass conservation method

Results

Reconstructed beds (id=1)



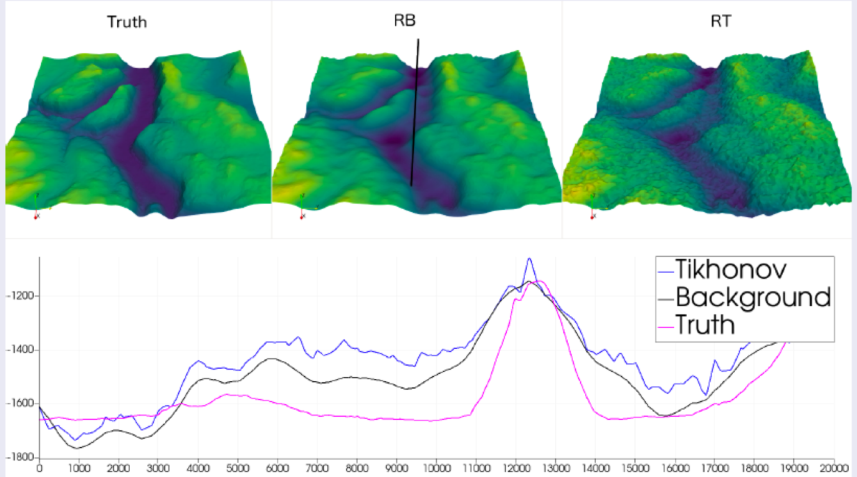
$RMSE_b$ for all cases

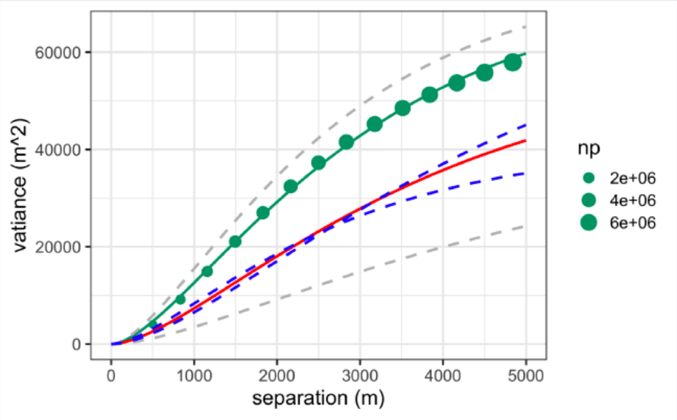


- $R_B(H)$ regularisation **always provides** the best reconstruction
- $R_T(H)$ underestimates correlation at short distances

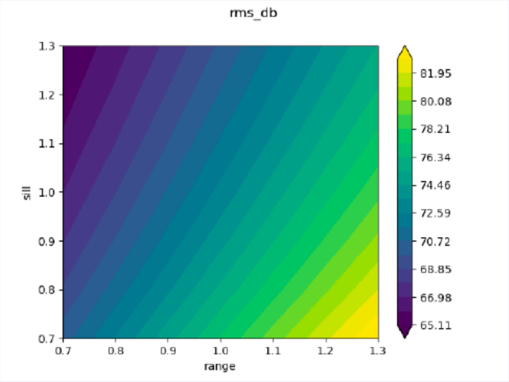
Application to the mass conservation method: sensitivity to range and sill parameters

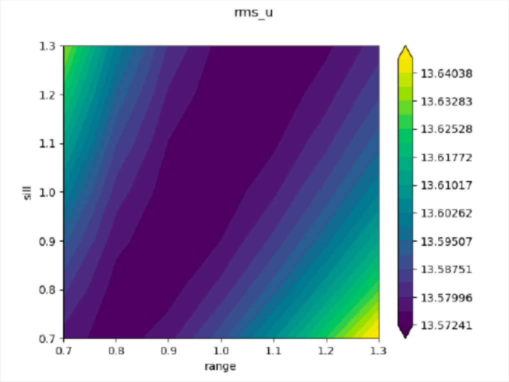
Results: Case 1

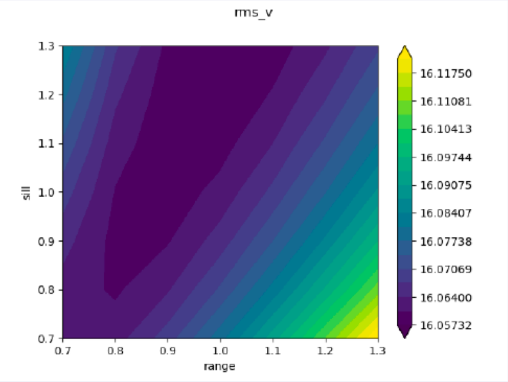




- Sensitivity to range and sill:

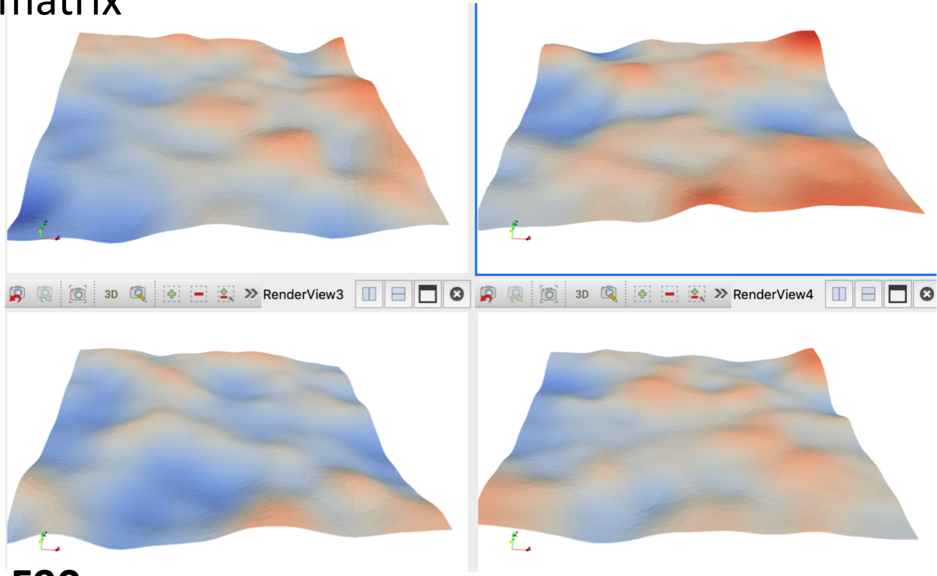






Possible applications with these tools:

- **Data assimilation** (**B^-1**) => **BackgroundErrorCostSolver.F90**
    - More difficult to define B for friction/viscosity inversions
    - But no so different that adjusting the weights in Tikhonov regularisation and you can show that it's equivalent in some cases (regular 1D mesh, …)
    - Parameters (range;sill) have physical meanings and should be consistent across mesh resolution
        - See e.g. A framework for time-dependent Ice Sheet Uncertainty Quantification, applied to three West Antarctic ice streams, Recinos et al., under review (TCD) and presented by J. Maddison in the modelling session
- **Gaussian simulations** (B=LL^T; y=μ + L.z) => **GaussianSimulationSolver.F90**
    - You can **draw random samples** using the same parameterised covariance matrix (requires to compute the square root)
    - **Uncertainty quantification using ensemble methods**
        - See Bulthuis, K., & Larour, E. (2022). Implementation of a Gaussian Markov random field sampler for forward uncertainty quantification in the Ice-sheet and Sea-level System Model v4.19. Geoscientific Model Development, 15(3), 1195–1217. https://doi.org/10.5194/gmd-15-1195-2022



- **Gaussian filters (smoothing noisy data)** (B) => **CovarianceVectorMultiplySolver.F90**