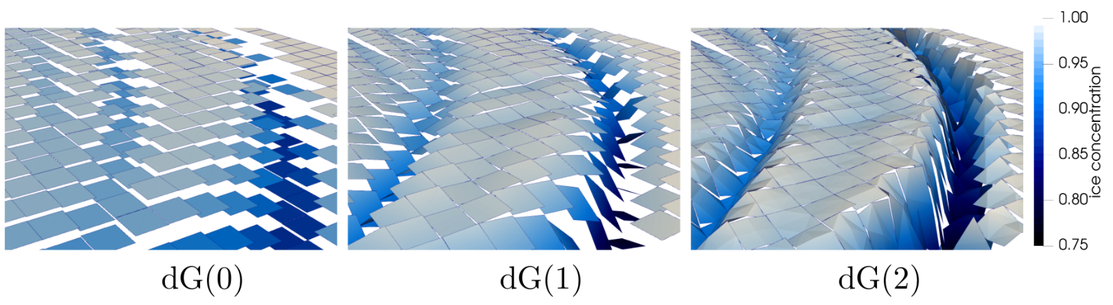
Integrating XIOS into the nextSIM-DG next-generation sea ice model

Joe Wallwork[1], Tom Meltzer[1], Tim Spain[2], Marion Weinzierl[1], Einar Ólason[2]

[1]Institute of Computing for Climate Science, University of Cambridge, U.K.
[2]Nansen Environmental and Remote Sensing Center, Bergen, Norway

Overview

nextSIM-DG is developed as part of the *Scale-Aware Sea Ice Project (SASIP)*. One of the main differences between nextSIM-DG and its predecessor – neXtSIM – is that it uses *discontinuous Galerkin (DG)* methods to better capture fractures in sea ice than with continuous elements.

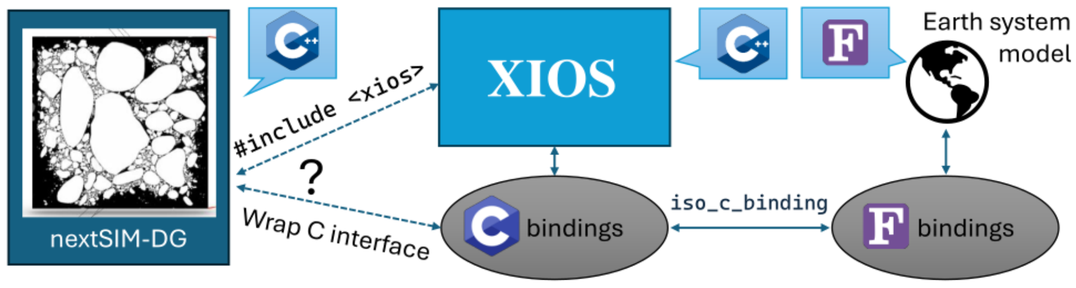


Ice concentration for different discontinuous finite element spaces. Image credit: Richter et al. (2023).

XIOS provides a server-based approach for efficient input/output (I/O), which may be configured either using XML files or with API calls. It supports asynchronous I/O, as well as the NetCDF and HDF5 data formats commonly used by the geoscientific modelling community.

Coupling to XIOS

Both nextSIM-DG and XIOS are written in C++. Should be straightforward, right? Not quite.

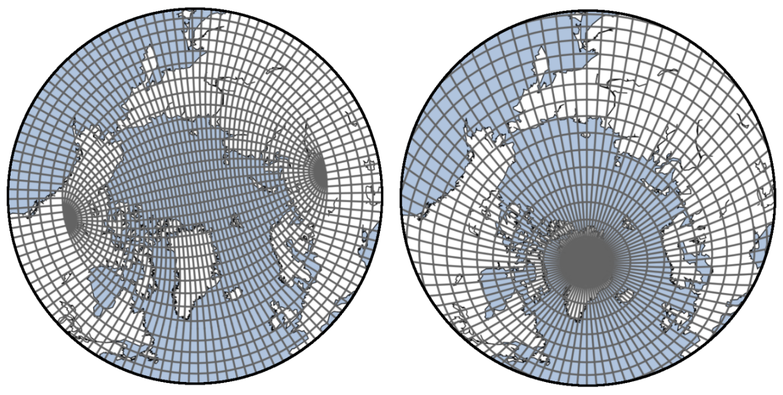


- XIOS' primary user base is the weather and climate modelling community, whose codes are mostly written in Fortran. XIOS provides C bindings and a Fortran interface built from these using `iso_c_binding`.
- Whilst we could make calls to XIOS' C++ API directly in nextSIM-DG, there is a risk that code which isn't in the public C interface might change, especially given the planned major version upgrade to XIOS3.
- Our approach is to wrap the C bindings in nextSIM-DG.

Coupling more generally

Coupling geoscience models

Earth system modelling requires coupling different model components together (e.g., ocean and atmosphere models). This often involves different grids, making it important to choose appropriate interpolation methods. Interpolation between model components can be achieved with *OASIS*.



Tripolar and bipolar grids over the Northern Hemisphere.

Image credit: Winkelbauer, et al. *"StraitFlux–Precise computations of Water Strait fluxes on various Modelling Grids."* EGUsphere 2024 (2024): 1-26.

Data considerations

- Even if grids match, data is an important consideration. A sub-optimal implementation might copy data unnecessarily between grids $\implies$ pass pointers where possible.
- Integrating tools directly into model data structures can be difficult. We found it simpler to map strings and arrays from XIOS' C interface to C++ `std::string`s and `std::vector`s before wrapping in nextSIM-DG syntax.

Summary

Challenges

- (Unexpected) multi-language problem.
- Unintended usage.
- How much of the API should we wrap?

Key lessons

- Use public interfaces where possible.
- Avoid unnecessary copies / modify data in-place.
- Start with standard data types before integrating model data types.

References

- Richter, et al. *"The neXtSIM-DG dynamical core: A Framework for Higher-order Finite Element Sea Ice Modeling."* EGUsphere 2023 (2023): 1-31.
- nextSIM-DG: `github.com/nextsimhub/nextsimdg`
- XIOS: `forge.ipsl.jussieu.fr/ioserver/wiki`
- OASIS: `oasis.cerfacs.fr/en/home`

Acknowledgements

This project is supported by Schmidt Sciences, LLC.