Coupling of the Ice-sheet and Sea-level System Model (version 4.24) with hydrology model CUAS-MPI (version 0.1) using the preCICE coupling library

Daniel Abele[1,2,3], Thomas Kleiner[2], Yannic Fischler[4], Benjamin Uekermann[5], Gerasimos Chourdakis[5], Mathieu Morlighem[6], Achim Basermann[1], Christian Bischof[4], Hans-Joachim Bungartz[3], and Angelika Humbert[2,7]

[1]German Aerospace Center, Institute of Software Technology, Oberpfaffenhofen/Cologne, Germany
[2]Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung, Bremerhaven, Germany
[3]School of Computation, Information, and Technology, Technical University of Munich, Germany
[4]Department of Computer Science, Technical University Darmstadt, Germany
[5]Institute for Parallel and Distributed Systems, University of Stuttgart, Germany
[6]Department of Earth Sciences, Dartmouth College, Hanover, NH, USA
[7]Faculty of Geosciences, University of Bremen, Germany

Correspondence: Daniel Abele (daniel.abele@dlr.de)

Abstract. Accurate earth system models must include interactions between atmosphere, ocean, and continental ice sheets. To build such models, numerical solvers that compute the evolution of the different components are coupled. There are frameworks and libraries for coupling that handle the complex tasks of coordinating the solver execution, communicating between processes, and mapping between different meshes. This allows solvers to be developed independently without compromises on numerical methods or technology. Code reuse is improved, both over large, monolithic software systems that reimplement each coupled model as well as over ad-hoc coupling scripts.

In this work, we use the preCICE coupling library to couple the Ice-sheet and Sea-level System Model (ISSM) with the subglacial hydrology model CUAS-MPI. An adapter for each model is required that passes the meshes and coupled variables between the model and preCICE. We describe the generic, reusable adapters we developed for both models and demonstrate their features experimentally. We also include computational performance results for the coupled system on a high-performance computing cluster. Coupling with preCICE has low computational overhead and does not negatively impact scaling. Therefore, the presented software facilitates studies of the subglacial hydrology systems of continental ice sheets as well as coupling ISSM or CUAS-MPI with other codes such as in global earth system models.

1 Introduction

Ice sheet dynamics is a gravity-driven lubricated flow, forced by changes at their boundaries, such as the ice-atmosphere interface, the ice-ocean interface and the conditions at the ice base. Beneath the ice sheet, a subglacial hydrological system exists that is formed by basal melting due to heat flux from the lithosphere and frictional heat. This hydrological system affects the ice sheet through changes in water pressure, while the ice sheet influences the subglacial system through changes in basal melt rates and ice sheet thickness. While the hydrological system changes in the interior of the ice sheet on long time scales only, the margins, in particular in Greenland, change in the peak melt season in summer on short time scales. Coupled simulations are required to simulate the evolution of both systems and their effect on each other.

The Ice-Sheet and Sea-level System Model (ISSM, Larour et al. (2012)) is a feature-rich ice sheet model. Among its capabilities, it includes the Subglacial Hydrology and Kinetic, Transient Interactions (SHAKTI) subglacial hydrology model (Sommers et al., 2018). SHAKTI is fully integrated into ISSM, using the same mesh and finite-element solvers. In addition to SHAKTI, ISSM also contains an implementation of another hydrology model, Glacier Drainage System (GlaDS), which is also included in the ice sheet model Elmer/Ice (Gagliardini et al., 2013). Other ice sheet models like PISM (Khrulev et al., 2025) also have their own hydrology models. While monolithic software development can be easier (e.g., only one build process, sharing the same data structures), it also comes with disadvantages.

In this paper, we present a different, partitioned approach. Fischler et al. recently published CUAS-MPI (subsequently referred to as CUAS), a stand-alone subglacial hydrology model. ISSM and CUAS rely on different spatial and temporal discretizations, which complicates the coupling. Here, we couple ISSM and CUAS using the preCICE coupling library (Chourdakis et al., 2022). For this, we developed the adapters for both models that link them to preCICE. This approach has a number of advantages over integrating CUAS directly into ISSM. The models can be developed independently and make their own choices regarding discretization and numerical methods. preCICE is highly configurable, so the time points where data are exchanged and other properties of the coupling process can be adapted without code changes. preCICE also provides sophisticated numerical methods that are required for accuracy, performance, and stability of the coupled system. Finally, the adapters aim to be as generic as possible to support other couplings than just ISSM-CUAS. It is therefore easy to either add components like an existing ocean circulation model or recombine the models with other ice sheet or hydrology models, which improves code reuse.

Section 2 of this manuscript covers the software packages that are involved. First, we describe the coupling library preCICE and the existing solvers[1] ISSM and CUAS. Then, we present the newly developed adapters. In the next Sect. 3, we show the setup and results of a few basic experiments we performed to test the coupling and the computational performance. In Sect. 4, we discuss our findings and plans for future development.

2 Software

In the following, we describe all the software components necessary for the coupling of ISSM and CUAS. Figure 1 shows an overview of the coupling. We give short summaries of the existing codes, with a focus on the technical details relevant to coupling: the preCICE coupling library, the ice sheet model ISSM, and the subglacial hydrology model CUAS. The newly developed preCICE adapters for ISSM and CUAS are described in detail.

[1]Throughout the paper, we use preCICE terminology, where *solver* refers to a complete simulation code and not a single numerical routine. A solver that is part of a coupled setup is referred to as *participant*. See https://precice.org/fundamentals-terminology.html.

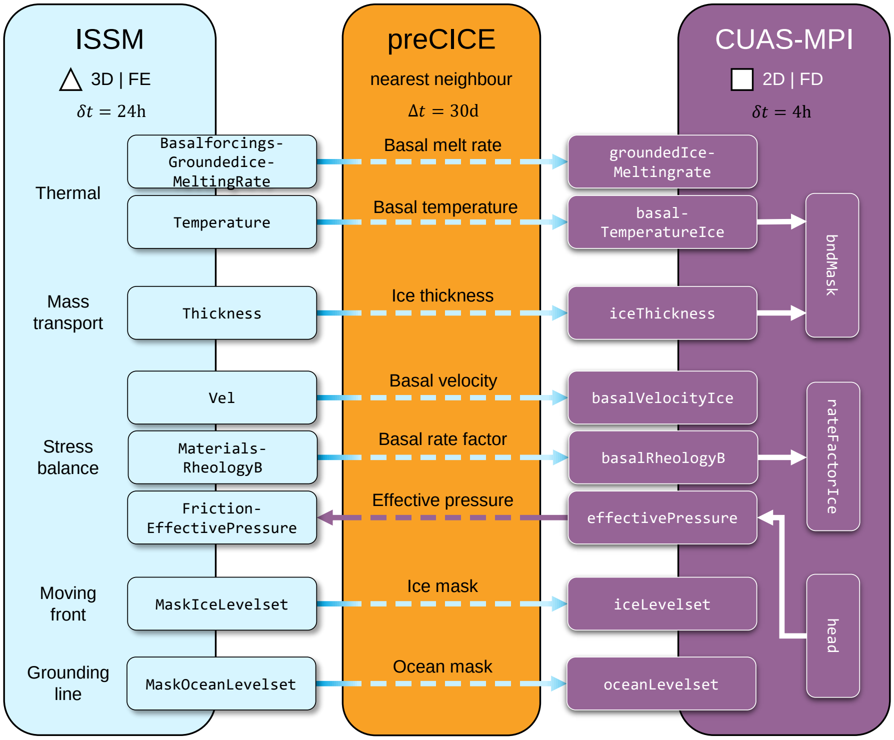

Figure 1. Overview of coupling between ISSM and CUAS-MPI. The coupling library preCICE handles communication and data mapping between the ice sheet and hydrology models. The models are free to use their own mesh (unstructured triangular, regular rectangular) and time steps. Coupled variables are listed by the name internal to the corresponding solver/adapter. Some variables are not directly used by CUAS but are converted first. Details of the exchanged variables and how they are used by the solvers are described in Sec. 2.4. The chosen coupling interval of 30 d is conservative to minimize computational overhead.

2.1 preCICE

preCICE (Chourdakis et al., 2022) is an open source coupling library for multi-physics simulations. The library couples two or more independent parallel solvers (also referred to as participants) and handles communication, data mapping, and coordination of the solvers, as described in the following paragraphs. To start a coupled simulation, the user starts each solver as usual, and each solver calls the preCICE library. This approach requires minimal changes to the source code of the solver, and all options can be configured at runtime. All instances of the library read a shared configuration file, via which the respective algorithms are selected. The code that connects a solver with preCICE (individual lines of code, a dedicated class, or a complete standalone package) is called an adapter. An adapter calls the application programming interface of preCICE, converts between the data structures of the solver and preCICE, and steers the time evolution of the solver (i.e., adapting the time step size, or storing and reloading checkpoints, if necessary). Adapters for a growing number of solvers and numerical frameworks exist.

2.1.1 Communication

preCICE communicates data between coupled solvers in a parallel, peer-to-peer, and point-to-point way. As back-ends, either TCP/IP or MPI can be used. To establish the communication channels, coupling meshes from one solver are repartitioned on the processes of another solver during initialization using a two-level algorithm (Totounferoush et al., 2021).

2.1.2 Data mapping

To handle non-matching coupling meshes, preCICE offers different methods for data mapping, including projection-based methods and kernel methods (radial-basis function interpolation) (Chourdakis et al., 2022; Schneider and Uekermann, 2025). While some projection methods require mesh connectivity information, kernel methods operate solely on point clouds. Each mapping can be configured to be either *conservative* (the total values over the interface are conserved for extensive properties, e.g., mass, forces) or *consistent* (for intensive properties, e.g., temperature, pressure). preCICE supports 2D and 3D Cartesian meshes and surface and volume coupling. In the particular case of this paper, both codes use geographically projected coordinates and can thus apply 2D Cartesian meshes. However, if we wanted to couple ice-sheet codes with Earth system models, we would need to convert between both coordinate systems. This is beyond the scope of this work.

2.1.3 Coordination of participants

To orchestrate the simulation progress of all coupled solvers, preCICE offers different coupling schemes. On the one hand, preCICE distinguishes between serial and parallel coupling: In serial coupling, coupled solvers advance sequentially, one after the other. In parallel coupling, coupled solvers advance concurrently. In both cases, the coupled solvers synchronize and exchange data after each fixed time window. On the other hand, preCICE distinguishes between explicit and implicit coupling. In explicit coupling, each time window is only computed once. In implicit coupling, each time window is repeated, with modified exchanged values, until predefined convergence criteria are met. To this end, solvers need to go back in time, which is typically implemented by checkpointing in the adapter. Implicit coupling increases accuracy and numerical stability. The convergence behavior can be improved with fixed-point acceleration, for instance, with quasi-Newton methods (Mehl et al., 2016). Accuracy and numerical stability can further be improved by sampling time interpolants during each time window (Rüth et al., 2021).

2.1.4 Alternatives

Other coupling libraries, similar to preCICE, exist (e.g., MUI (Tang et al., 2015), OpenPALM (Duchaine et al., 2015), or DTK (Slattery, 2016)), including some that specifically target Earth system modeling (e.g., YAC (Hanke et al., 2016) or MCT (Larson et al., 2005)/cpl7 (Craig et al., 2012)). Most of them focus on handling communication and data mapping, but do not offer advanced coupling methods (e.g., time interpolation or quasi-Newton acceleration). preCICE, as a general-purpose coupling library, does, however, not offer functionality specific to Earth system modeling, such as awareness of calendars (like ISSM) or data mapping tailored to geographic or spherical coordinate systems. One of the goals of this paper is to study to what extent such a general-purpose coupling library can be used for ice-sheet modeling, to potentially benefit from the larger community and maybe more advanced numerical methods.

2.2 ISSM

The Ice-sheet and Sea-level System Model (ISSM) is a well-established, feature-rich code for large-scale simulations of continental ice sheets (Larour et al., 2012). Mathematical ice sheet models consist of balance equations for enthalpy, mass, and momentum and their respective boundary conditions and kinematic boundary conditions for geometry evolution. Ice sheet codes are typically structured in different modules, also referred to as cores, that either solve individual balance equations or deal with the processing of data into forcing fields. A highly versatile ice sheet code, such as ISSM, offers several options for some cores. For example, the momentum balance might solve the full-Stokes equations (FS), higher-order Blatter-Pattyn approximation (HO), shallow-shelf approximation (SSA) or the shallow ice approximation (SIA), as described in the ISSM reference (Larour et al., 2012). Several glaciological processes can to date only be described empirically, for which a code may offer various parameterizations. Calving is such an example, for which ISSM offers a multitude of parameterizations. This configurability makes ISSM a code suitable for applications from mountain glaciers up to continental-scale ice sheets but also leads to large and complex code.

Most use cases of ISSM are large problems in the order of 0.1M - 10M degrees of freedom (DOF). For example, for simulating the Greenland ice sheet in moderate resolution (G4000), the different ISSM cores compute 31.5k - 944k DOF, high resolution (G250) requires 1.1M - 32M DOF (Fischler et al., 2022). Large problem sizes are computationally demanding, so the simulations need to be efficient and scale adequately. Fischler et al. (2022) investigated the performance of ISSM, showing that the code scales well and is not expected to be a significant bottleneck for the scaling of the coupled simulation.

2.2.1 Multi-physics capabilities

Ice sheets are complex systems, so even a standalone ice sheet simulation is already a multi-physics simulation (Fig. 3). In ISSM, the cores can be run individually, e.g., to get the stress balance solution only. However, more often, transient runs are conducted, where most cores are solved. The system of equations of ice sheets is not solved in a numerically monolithic way, but in a sequential (segregated) fashion. In ISSM, a typical sequence is: first, the enthalpy balance is solved, then the stress balance, and afterwards, the geometry is evolved. Each core immediately uses the results of the previous cores.

2.2.2 Mesh and solver

ISSM supports two and three-dimensional meshes. Figure 2 shows the mesh structure. The basic horizontal two-dimensional mesh is an unstructured triangle grid covering the horizontal computational domain, including ice-free regions. The 2D mesh is usually static, as there is limited support for adaptive mesh refinement.

The horizontal 2D mesh can be used in the SSA approximation. For HO or FS, a three-dimensional mesh is required. The 3D mesh is generated by vertically extruding the 2D mesh in multiple layers. The vertices in the top layer are aligned with the

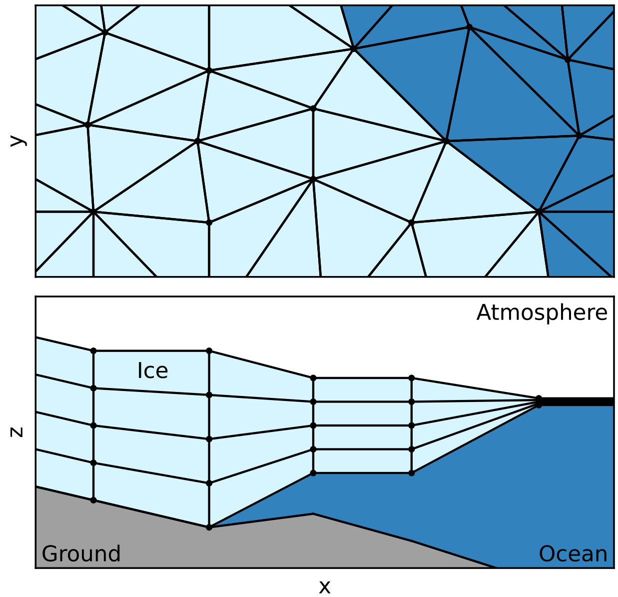

Figure 2. Schematic diagram of the ISSM mesh for 2D and 3D setups. The 2D mesh is an unstructured triangle grid covering the whole domain in horizontal direction, even where there is no ice (top). Triangle elements are completely ice or completely ocean. The 3D mesh is generated by extruding the 2D mesh in multiple layers of triangle prism elements (bottom). The vertices in every layer have the same x and y coordinates. The z coordinate is set so that the mesh always matches the vertical extent of the ice and it is updated when the thickness of the ice changes. In areas without ice, the mesh collapses in vertical direction to a minimal thickness.

surface of the ice. The vertices in the bottom layer are aligned with the base of the ice. The vertices in the layer in between are commonly distributed equally between the top and bottom vertices, but may be unequally distributed as well. Therefore, the vertical (z) coordinate of the vertices changes in every time step as the thickness of the ice changes. The vertices are connected as truncated triangular prisms or tetrahedra.

125 ISSM uses the finite element method (FEM) to solve the partial differential equations (PDE) for each core. The finite element type used can be configured for most cores individually. Linear P1 elements, where nodes are placed exclusively at the vertices, is the default for many cores, but higher order elements are available.

All cores use the same mesh but generally do not have the same finite element types. The time stepping method and step size is also generally identical for every core with fixed or adaptive time steps, but a few cores, e.g., SHAKTI, subdivide the steps
130 further. All cores use the same number of CPUs and same domain decomposition for MPI parallelization. Fischler et al. (2022) showed that the cores that solve two-dimensional problems (e.g., mass transport, moving front) are significant bottlenecks for scaling as they compute fewer DOFs than the three-dimensionsal cores (e.g., thermal, stress balance). This could be resolved with more flexible domain decomposition or CPU allocation.

2.2.3 Architecture

ISSM's architecture is well suited for the development of a generic coupling adapter. Mesh and data access can be implemented based on abstract interfaces for different cores, mesh types, finite element types, etc. Variables are identified by runtime values (strings externally, mapped to enum values internally). With few exceptions that will be noted when describing the adapter in Sect. 2.3, the adapter does not need to include code to handle specific configurations.

2.3 The ISSM-preCICE adapter

The ISSM-preCICE adapter aims to be generic and extensible in order to support different use cases. This section explains how the features of ISSM and preCICE are handled in the implementation. The adapter is an executable that runs in place of the ISSM executable. The adapter configuration file is specified as a command-line parameter, and the command-line parameters of the ISSM executable are part of the adapter configuration file. The configuration of the adapter is done by a file in YAML format. The adapter configuration file is mostly responsible for mapping names specified in the preCICE configuration file to names expected by ISSM and defining the coupling interface. Listing 1 shows an example configuration file. The format conforms to the adapter configuration schema defined by the preECO project[2]. Details of the entries in the file are explained in the following sections. Each section highlights limitations and missing features.

2.3.1 Architecture of the adapter

There are different ways to implement preCICE adapters. Adapters can be directly integrated or patched into the solver's code (e.g., the CalculiX-preCICE adapter (Uekermann et al., 2017)), while other adapters are developed as stand-alone software packages, either as plugins (e.g., the OpenFOAM-preCICE adapter (Chourdakis et al., 2023)) or as orchestration codes that also call the solver (e.g., the CAMRAD II-preCICE adapter (Huang et al., 2021)). As shown in Fig. 3, the ISSM adapter follows the latter approach, structured as a wrapper application that calls ISSM (used as a library) as well as preCICE. This approach allows for independent development and a clean architectural separation between solver and adapter, allowing, e.g., easy support for multiple ISSM versions. However, this relies on a relatively stable API of ISSM and might pose maintenance challenges in the future. Additionally, due to the architectural choice of using ISSM as a library (instead of via its command-line interface), some functionalities internal to ISSM are not available. Note that so far no changes to ISSM or its build process were necessary to support the features of the adapter.

2.3.2 Coupling mesh

Since ISSM could potentially be coupled with many different codes, the coupling interface is configurable. The most important interfaces for coupling with the environment of the ice are the base and surface. So far, these are the only supported interfaces, see the list of limitations below for other interfaces that may be added later. In the configuration file, the user specifies which part of the mesh forms the coupling interface. The vertices of this part of the mesh are passed to preCICE. Mesh connectivity

[2] https://precice.org/couple-your-code-adapter-software-engineering.html

```yaml
precice_config_file_name: precice-config.xml
participant_name: ISSM
issm:
  root_path: some/path/to/model
  model_name: model
interfaces:
  - mesh_name: ISSM-Mesh
    patches:
      - Base # or Surface
    read_data:
      - name: effectivePressure
        solver_name: FrictionEffectivePressure
    write_data:
      - name: iceThickness
        solver_name: Thickness
      - # ...
```

Listing 1. Example adapter configuration file in YAML format. The adapter requires information about the preCICE and ISSM configurations, the coupling mesh and the names of the variables being read or written. Variable names in the preCICE configuration file are mapped to variables known to ISSM.

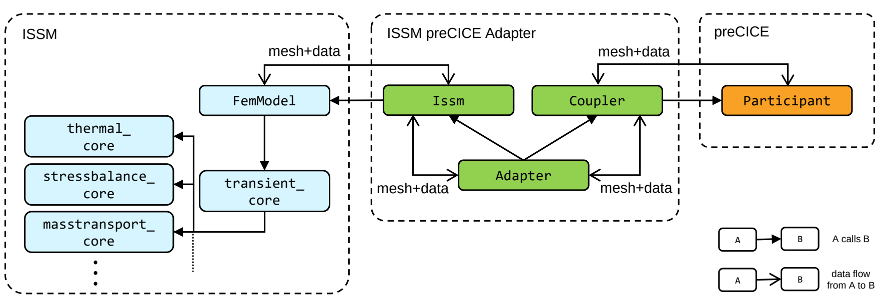

Figure 3. Structure of the ISSM-preCICE adapter. The adapter has its own time loop and calls the ISSM solver through the ISSM `FemModel` class. No interaction with internals of ISSM (such as the constituent cores that make up the transient core) is necessary. The `Adapter` class is the entry point of the program; it coordinates the coupling and the exchange of coupled data (read and write) between ISSM and preCICE. Both ISSM and preCICE are wrapped in the adapter library to improve isolation and testability.

is added to support mapping schemes like linear cell interpolation. The finite element representation of ISSM variables is evaluated at the mesh vertices before writing data to preCICE.

Limitations: The current implementation of the adapter has some limitations regarding the mesh handling, all of which could be addressed in future work, as the technical requirements are already available. These limitations include:

- Some precision of high-order finite element types is lost when evaluating variables at the mesh vertices. Instead, the nodes of the finite elements could be used. For best results, this would require multiple coupling interfaces for different finite element types and mesh connectivity would not be available.

- The adaptive mesh refinement feature of ISSM is not supported, the mesh must be static. Note that preCICE does provide facilities for mesh refinement[3], but the adapter does not yet use them.

- 3D (volume) coupling is not supported. A 3D coupling interface would allow, e.g., to couple ISSM with itself to use different meshes for different cores to optimize precision or performance or to use an external thermal solver. However, as ice is transported by the solver, the z coordinate of the vertices changes in every time step. As the interface changes significantly over time, the coupling mesh could be reset to the new locations of the interface just like for adaptive mesh refinement above. So far, we have not tested whether the computational overhead to update the coupling mesh and corresponding interpolation matrices in preCICE is acceptable.

- Partial interfaces are not supported. For example, it may be beneficial for ice-ocean coupling to only couple over floating ice.

2.3.3 Variables

In ISSM, every state variable is identified by a unique name. These names do not follow any consistent convention, and the other coupling participant may use different names. So, it is necessary to map between names in the preCICE configuration file and ISSM names. The configuration file provides such name mappings for read and written variables.

If ISSM uses a 3D mesh, it is possible to perform depth-averaging of a variable before writing it and extruding a variable (i.e., copying the variable values to the other layers of the mesh) after reading.

Limitations: ISSM accepts time series as input variables. For example, the user can set specific values for these variables at the beginning, middle, and end of the simulation. ISSM calls these "transient variables" and temporally interpolates between these values when necessary. However, ISSM does not allow overwriting the user-provided values of such transient variables. Therefore, the adapter requires coupled variables to be set up as non-transient, i.e., with one fixed value that is overwritten with the value read from preCICE during the simulation. This is a purely technical restriction and does not reduce modelling capabilities, since coupling can also include input variables that change over time.

[3] https://precice.org/couple-your-code-moving-or-changing-meshes.html

2.3.4 Data initialization

In ISSM, most variables are not zero as the simulation starts at some point in time in a defined state. For some variables, e.g., ice thickness, zero is not even valid at all. For non-zero initial values, preCICE allows data initialization[4] by writing variables once in the beginning of the simulation. The ISSM adapter assumes that such initialization is necessary for every variable it writes. For most variables, the adapter simply writes the initial value specified in the ISSM setup. However, users are not required to specify true initial values for variables that are computed by ISSM before they are used. For velocity, the initial value in the setup is merely used as an initial guess for the non-linear iteration of the stress balance core. To initialize velocity, the adapter runs the stress balance core during the initialization phase to get the true initial velocity. This also applies to other variables and cores, e.g., rheology parameter B that is computed by the thermal core.

Limitations: There is no generic way to handle the computation of initial values for all such variables. Special handling has to be added separately for each variable that requires it.

2.3.5 Boundary conditions

ISSM can set discrete Dirichlet boundary conditions (mainly called single point constraints in ISSM) for some cores, e.g., velocity (`Vx`, `Vy`, `Vz`) in the stress balance core. The adapter has limited support to set some of these constraints by coupling. In the ISSM code, the constraints are stored differently from normal variables, so generic support for all constraints that ISSM uses is currently not possible. The association between constraints and the core that they apply to has to be hard-coded for each one. For example, the adapter has no way to find out automatically that constraints on velocity are used by the stress balance core. Further complicating the implementation is that manual MPI communication is required in the adapter to synchronize constraints on ghost vertices. For variables, no such extra communication was necessary.

Limitations:

- Internally to ISSM, constraints are stored per finite element node, not per mesh vertex. Since the coupling mesh is defined at the vertices instead of nodes, only P1 finite elements (or similar) are supported. Unlike for variables, there is no interpolation from P1 to other element types for Dirichlet constraints.

- Only velocity and pressure constraints of the stress balance core are supported so far.

2.3.6 Time stepping

ISSM performs multiple time steps per coupling window depending on the step size set in the ISSM setup.

Limitations:

- Subcycling is not used, i.e., `precice::Participant::advance` is not called on every ISSM time step. Therefore, time interpolation by preCICE is not enabled. This is to avoid excessive file output: ISSM unavoidably writes output at least once per call to the solver routine (in addition to fixed intervals set by the user). So we call the routine only once

[4]https://precice.org/couple-your-code-initializing-coupling-data.html

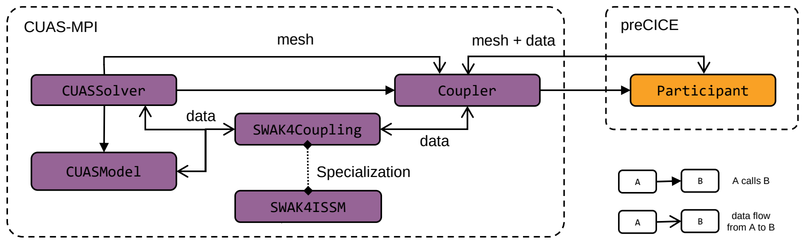

Figure 4. Structure of the CUAS-preCICE adapter. The coupling is integrated into and coordinated by the existing `CUASSolver`, but most of the coupling logic is isolated in added classes. The coupling data that is read and written flows through a utility class (specialized for coupling with ISSM) that handles transformations such as deriving ice pressure and transforming units.

per coupling time window, performing multiple time steps per call, and then call `advance` once after the routine ends. This could only be resolved with code changes in ISSM itself.

– Implicit coupling is not supported. The adapter currently cannot create the necessary checkpoints of ISSM, neither in memory nor on disk. Implicit coupling is required for numerical stability in some setups (see Sect. 2.1). Our experiments show no instability with explicit coupling. This is consistent with the internal explicit coupling of ISSM cores explained in Sect. 2.2.1.

2.4 CUAS and the CUAS-preCICE adapter

CUAS (Fischler et al., 2023) is the MPI parallel implementation of the Confined-Unconfined Aquifer System (CUAS) model for subglacial hydrology. It employs an equivalent porous medium approach in which both a distributed and channelised system are represented by one porous layer. The model solves a vertically integrated groundwater equation (Fischler et al., 2023, Eq. 1) using effective quantities for storativity and transmissivity, which evolve over time based on parameterisations (Fischler et al., 2023, Eqns.2–4). CUAS uses a finite difference spatial approximation on a regular rectangular grid and an implicit Euler time stepping scheme. CUAS solves for the hydraulic head that is proportional to the water pressure. The effective pressure, N (ice overburden pressure minus water pressure), is a diagnostic quantity that is computed in each time step and is used in ISSM for sliding.

We added an experimental preCICE adapter to CUAS. An adapter configuration file has not been specified yet, so the adapter is specific for coupling with ISSM. The adapter currently does not support implicit coupling schemes or interpolations that require mesh connectivity. The CUAS-preCICE adapter is implemented within the CUAS code base and is not a standalone application. This adapter implementation strategy is similar to e.g., the CalculiX-preCICE adapter (Uekermann et al., 2017) and offers a high degree of flexibility. Additionally, the code base of CUAS is much smaller and more modern than of ISSM, so maintenance is not significantly impacted by this choice.

Figure 4 shows an overview of the module structure and data flow. preCICE is integrated directly into the existing CUAS time step iteration, enabling subcycling and time interpolation. The CUAS-preCICE adapter consists of two parts. The `Coupler` class coordinates the main coupling operations of initialization, reading and writing data, and advancing the coupling window. It exchanges data with the CUAS model and solver through the `SWAK4Coupling` interface, which applies necessary transformations to the data based on the model physics. While some transformations are generic, e.g., deriving ice pressure used by CUAS from ice thickness provided by ISSM, others are very specific to the data that the ISSM-preCICE adapter can provide, hence the `SWAK4ISSM` specialization of the interface. Some of the tasks are as simple as converting rheology parameter B from ISSM to `rateFactorIce` (A) in CUAS using $A = B^{-3}$ (assuming a Glen's flow law exponent of $n = 3$). Others are more complex. For example, we use the ice thickness to compute the ice pressure. We further compute the pressure melting point using the ice pressure and the absolute temperature from ISSM to decide whether the ice is frozen at the bed or not, and adjust the mask (active versus inactive) in CUAS. Users can configure whether the mask is allowed to change based on the simulated temperature or not. The transformations applied to the coupling data are implemented directly in the adapter, but the motivation is similar to preCICE Actions[5].

Each time new data is available from preCICE (see Fig. 1 or Fig. 6), the CUAS-preCICE adapter needs to perform several tasks, which are briefly outlined below.

- `iceThickness` is translated into ice overburden pressure using ice density.

- `groundedIceMeltingrate` is rescaled from ms^{-1} ice equivalent to ms^{-1} water equivalent using ice and water density. This is then used as a time-independent (steady) forcing for CUAS during the duration of the current coupling time window.

- Use the `iceThickness` and the steady bed elevation field from CUAS to compute a new `bndMask` using the flotation condition. The `bndMask` contains the information where we have active hydrology (warm base, grounded ice) and where boundary conditions need to be applied (e.g. floating ice or open ocean). Here we also initialize grid points that turned from ocean boundary condition into active CUAS due to grounding line advance. Grounding line retreat is also handled.

- We use `iceLevelset` to disable grounded ice areas in the CUAS domain that are not part of the ISSM domain. We do not use `oceanLevelset` in the prototype implementation of the adapter.

- `basalTemperatureIce` together with `iceThickness` is used to decide if the base is at the pressure melting point to further constrain the `bndMask` in CUAS, if needed.

- The `basalVelocityIce` is copied over without modifications.

- Finally, `rateFactorIce` (A) in CUAS is computed based on `basalRheologyB`.

[5] https://precice.org/configuration-action.html

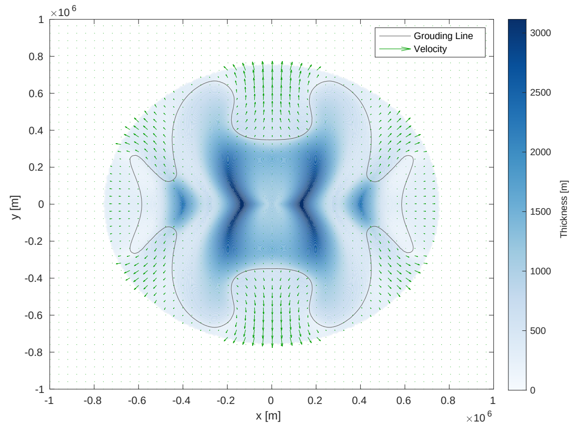

Figure 5. State of the ice sheet in the CalvingMIP Thule domain at steady state. The setup is used to test coupling functionality.

Because the effective pressure is computed directly in CUAS, the adapter can provided this field for coupling without further modifications.

To allow for coupled simulations, other model capabilities of CUAS besides the adapter have been enhanced. We reimplemented the way CUAS is handling the forcing so that different model forcings can be registered and are aggregated. Ice sheet basal melt from coupling is just one source of water for CUAS. This is important because simulations of the Greenland Ice Sheet require water not only from basal melt, but also from surface runoff.

3 Experiments

This section describes experimental verification of the ISSM-preCICE adapter for coupling of ISSM to a model of subglacial hydrology. Here, only synthetic setups are used. Use in real world setups will be explored in future research. Both functionality and computational performance are demonstrated in the following sections.

3.1 Functionality

3.1.1 Experimental setup

We use the synthetic Thule geometry developed for the CalvingMIP project (Jordan, 2024). This setup is based on analytical functions for the bed elevation and results in an ice sheet geometry that contains all the major parts of an ice sheet model domain (grounded ice, floating ice and open ocean). In CalvingMIP it is used to study how different ice sheet models handle calving in a very controlled setup. Instead of fixed thermal and friction conditions as in the CalvingMIP project, we enable the thermal core of ISSM with surface temperature $250\,\mathrm{K}$ and geothermal flux $0.05\,\mathrm{W\,m^{-2}}$ to compute basal melt rates to be used by CUAS and we use the default Budd friction law ($\sigma_\mathrm{b} = C^2 N u_\mathrm{b}$) with coefficient $C = 100$, and effective pressure N supplied

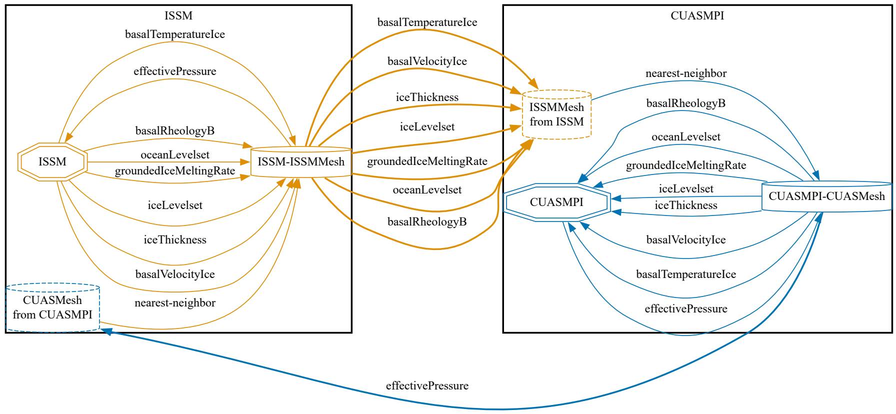

Figure 6. Coupling configuration between ISSM and CUAS. ISSM writes the state of the ice and reads the effective pressure to be used in the friction law. The figure was generated using the preCICE config visualizer (https://precice.org/tooling-config-visualization.html).

by CUAS. Surface mass balance was increased to $8\,\mathrm{m\,a^{-1}}$ ice equivalent to get the balanced geometry shown in 5. The goal is a mix of grounded and floating, fast and slow flowing regions to see the effects of coupling.

The combined model (ISSM + CUAS) initialization consists of two phases. First, we run the ISSM model for 12500 years with 2 year time steps into a steady-state using an effective pressure field of half the ice overburden pressure that is only derived from the ice sheet model geometry. In the second phase, CUAS is coupled just once to ISSM at the beginning of its own spinup to receive a consistent ice sheet model state (see Fig. 6). The CUAS spinup runs for 100 years (resolution: $2.5\,\mathrm{km}$, time step: $4\,\mathrm{h}$) into steady state.

To investigate the effect of coupling, we perform two experiments. In the control simulation, the models run fully coupled for 20 years starting from the end of the spinup. The coupling interval is $720\,\mathrm{h}$ ($30\,\mathrm{d}$). This is a conservative choice and the low overhead for coupling (see Sect. 3.2) shows that the frequency of coupling could be increased without significantly impacting computational performance. The time step of the solvers is still $4\,\mathrm{h}$ in CUAS and $1\,\mathrm{d}$ in ISSM. In the second experiment, we apply an additional, time-dependent water source of about $1\,\mathrm{m\,a^{-1}}$ (water equivalent) for 60 days along a straight line in one half of the domain (Fig. 8) to simulate the effect of surface melt water draining down to the base through a long crack during the summer season. We refer to this additional source of water as an anomaly (anom.) forcing. All other parameters are identical to the control simulation. As mentioned in Sect. 2.4, CUAS has been extended to aggregate water sources from coupling and from input files to enable simulations with anomaly forcing.

Details of the coupling configuration are shown in Fig. 6. As described above, CUAS writes the effective pressure that is used by ISSM to determine basal friction. The basal melting rate from ISSM provides a source of water for CUAS. Basal temperature, ice thickness, and ice and ocean masks are used to update the active CUAS mask. Ice velocity and rheology influence the opening and closing of channels. We are using the simplest mapping scheme, nearest neighbour. Details on how the variables are used in CUAS are described in Sect. 2.4.

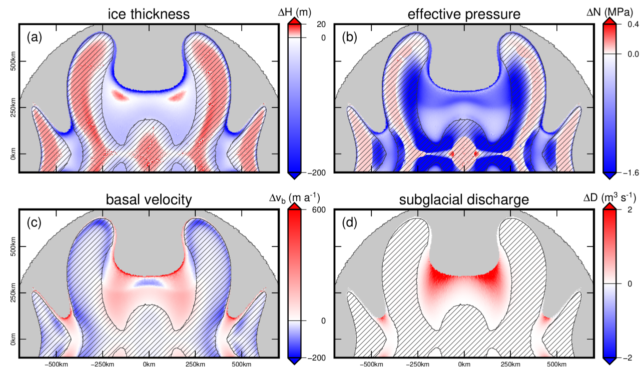

Figure 7. Result of the fully coupled control run using the CalvingMIP setup. The panels a–d show the difference of some selected fields with respect to the initial state (end of spinup) at the end of the 20 year simulation. The gray area indicates the extent of the floating ice. The basal melt rate computed by ISSM is the only water source. The hashed areas indicate the parts of the domain where basal temperatures from ISSM are below the melting point and the basal melt rate is zero.

3.1.2 Results

Figures 7 and 8 show the results of the coupled runs at the end of the simulation with (anom.) or without (control) an additional water source. The ice thickness and basal velocity are selected as the key quantities describing the ice sheet state, while effective pressure and subglacial discharge are selected to describe the state of the subglacial hydrology. The hashed areas in both figures indicate the parts of the domain where basal temperatures from ISSM are below the pressure melting point and ice sheet basal melt is zero (cold base). In a CUAS stand-alone simulation, these areas would be considered too cold to allow for an active hydrology. Nevertheless, ISSM allows for sliding independent of the basal temperature and thus, also the cold-base areas are active in the CUAS setup and feed back the effective pressure to the ice sheet model.

In areas where ISSM is at the pressure melting point (warm base) and ice sheet basal melt provides a source of water for the subglacial hydrology, coupling of ISSM and CUAS results in lower effective pressure (Fig. 7a), leading to enhanced basal sliding (Fig. 7c) and thus higher ice discharge into the ocean. This results in a reduction of ice thickness (Fig. 7a) in most of those areas and grounding line retreat. The increase in subglacial discharge (Fig. 7d) is a direct consequence of higher basal melt rates in the coupled simulations, also contributing to the reduction in ice thickness. In the cold-based areas of the domain, the effective pressure increases compared to the uncoupled simulation, leading to lower basal velocities and thus slightly thicker ice.

The simulation with additional water from the surface (anom.) is intended to test if CUAS can handle different water sources (forcing). Figure 8 illustrates that this is the case and that anomaly forcing leads to an alteration (relative to the control simulation) of the ice thickness and basal velocity. The difference is still visible at the end of the simulation, which is also the end of the year, although the anomaly is only applied during a fraction of the year.

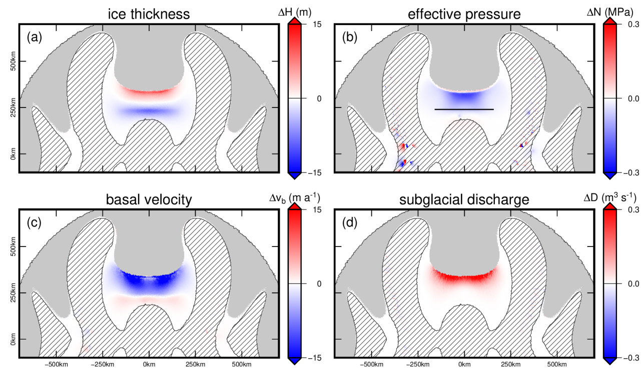

Figure 8. Result of a fully coupled run using the CalvingMIP setup with an additional seasonal water source, comparable to surface melt in the summer in Greenland. The panels a–d show the difference of some selected fields with respect to the control simulation. Both at the end of the 20-year simulation. The hashed areas indicate the parts of the domain where basal temperatures from ISSM are below the melting point and the basal melt rate is zero.

3.2 Performance

The coupled simulation should run with minimal computational overhead and use computing resources efficiently. To demonstrate this, we use a large-scale setup of the Greenland Ice Sheet. As described below, the coupling is simplified to avoid numerical problems, but should have representative performance characteristics. Realistic coupling would require more careful construction of compatible setups and is beyond the scope of this paper. This will be explored in future work.

3.2.1 Experimental setup

The setup for ISSM is mostly the same as G1000 in Fischler et al. (2022). The only modification we made for this paper is to use the default ISSM Budd friction law $\boldsymbol{\sigma}_b = -C^2 N^{\frac{q}{p}} |\boldsymbol{u}_\mathrm{b}|^{\frac{1}{p}-1} \boldsymbol{u}_\mathrm{b}$ with coefficient $C = 12.94$, exponents $p = 3$ and $q = 2$, and effective pressure N provided by CUAS. The spinup process is the same as in Fischler et al. (2022). During the spinup, the effective pressure is $N = N_\mathrm{opc} = \rho_\mathrm{ice} g H - \max(0, \rho_\mathrm{water} g(-z_b))$ with z_b the height of the base above sea level as in Wolovick et al. (2023).

The setup for CUAS is taken from Fischler et al. (2023). We use G500, which has approximately the same resolution as the minimum element size of the ISSM setup. Ice sheet basal melt from ISSM is the only water source for CUAS. No basal temperature constraint is applied to the CUAS mask (`bndMask`) to allow for active subglacial hydrology everywhere under the ice. This is needed because ISSM allows for basal sliding independent of basal temperature and thus requires getting the effective pressure everywhere. The layer thickness was chosen equal to 1 m, and no transition between confined and unconfined. The yield storativity $S_y = 10^{-2}$. For defining the initial active mask in CUAS, a threshold of minimum 10 m of ice thickness was chosen. Similar to ISSM, the initial hydraulic head of CUAS is derived from N_opc.

As experiments, we ran both serial and parallel coupling and increase the number of processes used. We used the results of (Fischler et al., 2022) and (Fischler et al., 2023) to find ranges of process counts where we can reasonably expect the coupled system to scale efficiently. Apart from the coupling scheme, the coupling configuration is the same as in Sect. 3.1.

For serial coupling, both participants run on the same nodes/CPUs and alternate in using all available processes. This minimizes idle time of CPUs. However, this may not work on every system: it may be necessary to allocate separate nodes for each participant to avoid deadlocks in communication.

For parallel coupling, we started with a distribution of two ISSM processes for each CUAS process ($n_{\mathrm{ISSM}} : m_{\mathrm{CUAS}} = 2.0$). Early testing suggested this would lead to approximately equal execution times for each solver and therefore minimize wait times. Based on the results of these experiments, as the solvers exhibit different scaling characteristics, we also tried other distributions of processes for a few selected total process counts. Note that this flexible allocation of CPUs would not be possible without significant development effort if CUAS was integrated into ISSM directly.

We ran the coupled setup for 18 coupling windows of 30 simulated days each, averaging runtimes over five runs. The remaining experiments with parallel coupling and all experiments with serial coupling and are run only once, as randomness in execution times is accounted for by the long runtime and averaging over coupling windows.

We measure the execution time required for each coupling window and how much time is used by each solver and preCICE, respectively. For the measurements, we are using the profiling utility integrated into preCICE[6]. In order to achieve representative results, measurements do not include writing output, as I/O execution times can swing wildly and unpredictably. Adding moderate amounts of data output to both participants should not significantly impact the analysis.

Note that serial and parallel coupling schemes also give different numerical results. Serial coupling is generally more accurate as results from one participant are immediately used by the other participant in the same coupling time window. However, careful investigation of simulation results will be part of future work. Here, we only made sure that both participants use comparable resolution and precision, with a relative solver residual of $\mathrm{rtol} = 10^{-6}$.

The experiments are performed on the Albedo cluster of the Alfred Wegener Institute. The compute nodes of the cluster are equipped with 256 GB of RAM and two AMD Rome Epyc 7702 CPUs for a total of 128 CPU cores per node and are connected by 100 Gb InfiniBand network. The solvers and dependencies are built with GCC version 12.1.0 and OpenMPI version 4.1.3. Care is taken to assign packed blocks of CPUs to participants, i.e., processes used by one participant are as local to nodes as possible, since each participant communicates more internally than with the other participants.

3.2.2 Results

Figures 9 and 10 show the basic time line of coupling experiments with different numbers of processes and coupling schemes. The four basic components of coupling are:

1. Initializing the solver, including loading input data and preparing initial coupling data and mesh for preCICE.

[6] https://precice.org/tooling-performance-analysis.html

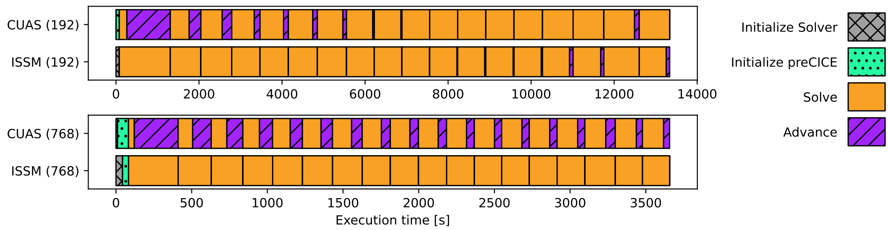

Figure 9. Timeline of two representative experiments with parallel coupling scheme and different numbers of processes (given in parentheses for each solver). Significant times spent in *Advance* indicate wait times where one participant finishes a coupling window before the other. In the first few coupling windows, ISSM's stress balance solver struggles to converge with effective pressure from CUAS. With few processes, wait times are balanced beyond the first coupling windows. With increasing numbers of processes, CUAS needs to wait more often. Initialization time of solvers and preCICE is negligible relative to the simulation.

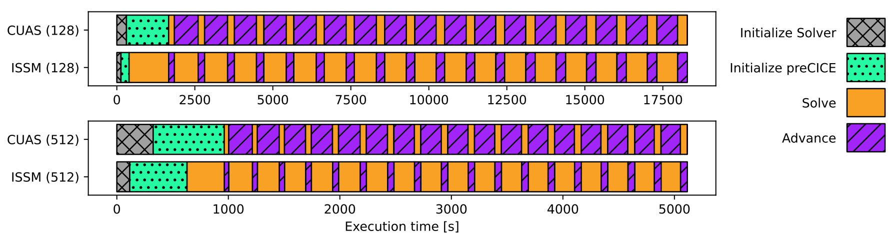

Figure 10. Timeline of two representative experiments with serial coupling scheme and different numbers of processes (given in parentheses for each solver). Like in 9, a few coupling windows are required for stable solve times. During initialization, both participants compete for the shared CPUs, significantly increasing the time required relative to the simulation.

2. Initializing preCICE, where the coupling framework builds its internal data structures for mapping and communication. This also includes wait times if one solver is initialized faster than the other.

3. Running the solver. For legibility we do not display single time steps with preCICE calls in between, but all time steps of one coupling window in aggregate. No relevant information is lost by that since the ISSM adapter calls preCICE only once per coupling window anyway and for CUAS the overhead of calling preCICE in the middle of a coupling window is negligible (on the order of $10\,\mathrm{ms}$ per time step.)

4. Advancing to the next coupling window and sending the data between participants. This mainly consists of wait times if one solver finishes the computation of one coupling window before the other.

During initialization of solvers and preCICE in parallel coupling, participants have dedicated compute and I/O resources. In serial coupling, both participants compete for the same resources, leading to much increased initialization times. Initialization, i.e., everything that happens before the participants or the first participant in a serial coupling, also does not scale well. Exe-

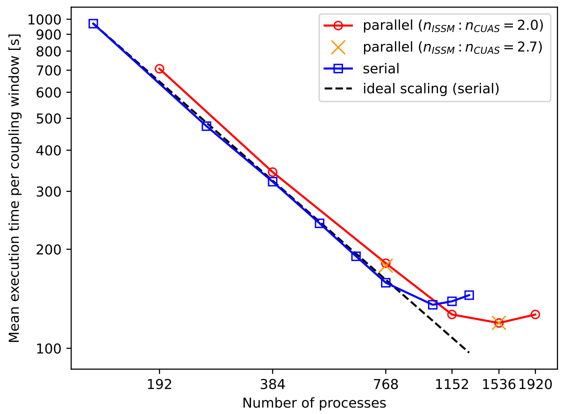

Figure 11. Average execution time for one coupling window (includes solver and communication) using a parallel or serial coupling scheme for different numbers of MPI processes. Parallel coupling is tested with different distributions of processes to participants.

cution times do not change much with more processes. These effects may not be representative of how other clusters handle resource contention, specifically I/O.

The first few coupling windows are clearly dominated by the execution time of ISSM. The stress balance solver struggles to converge, often hitting the maximum number of iterations. The effective pressure delivered by CUAS is too different from the effective pressure used during to set up the ISSM model. To some degree this is unavoidable, but it can probably be reduced with careful construction of compatible solver setups. After a few coupling windows, execution times stabilize.

The following analysis ignores the initialization phase and the first six coupling windows and only includes the remaining 12 coupling windows. This covers 360 simulated days. Simulations usually run for longer, so any one-time effects at the beginning of the simulation do not significantly impact the overall runtime.

Figure 11 shows the comparison of scaling between the serial or parallel coupling scheme. In the range where both parallel and serial coupling exhibit almost perfect strong scaling, serial coupling is faster since it is not possible to entirely eliminate wait times in parallel coupling due to uneven execution times of the solvers.

Figure 12 shows the time required by the solvers to finish one coupling window. CUAS scales very well in the entire tested range, whereas ISSM only scales close to perfect strong scaling up to around 768 processes. As shown in Fig. 13, this leads to increasing wait times for CUAS in parallel coupling. In both cases, execution time of ISSM is the limiting factor.

An attempt was made to improve execution times of parallel coupling by minimizing wait times by assigning more processes to ISSM. For 192 processes, a ratio of 2.0 ISSM processes per CUAS process balances wait times well. For 768 total processes, a ratio of 2.7 works better (see Fig. 13.) However, this leads to only slightly lower execution times. Giving more resources to ISSM when it is at its scaling limit does not seem to significantly improve performance. At 1536 processes, lowering the wait

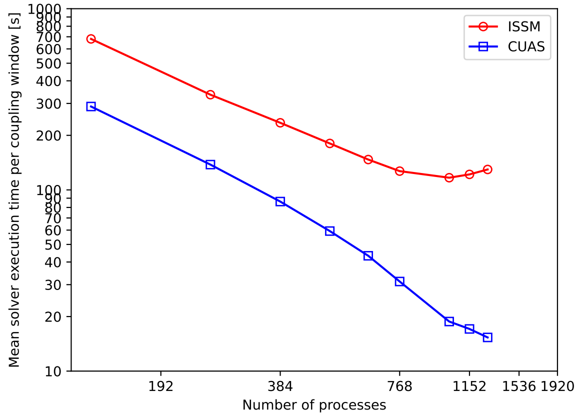

Figure 12. Average execution time required by the CUAS and ISSM solver during one coupling window for CUAS and ISSM for different numbers of MPI processes.

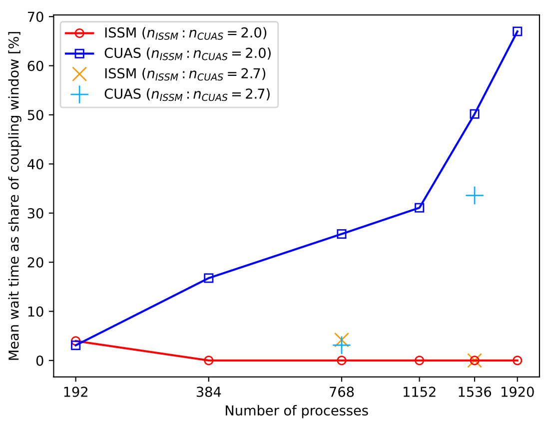

Figure 13. Average communication time relative to execution time of one coupling window for CUAS and ISSM in a parallel coupling scheme for different numbers of MPI processes and two different distributions of processes to participants.

times by redistributing processes actually has a slightly negative effect. These are the experiments that were repeated five times to increase confidence in the results. There is no large effect in either direction.

4 Discussion

We have presented the technical aspects of a new framework for coupling a subglacial hydrology model to the ice sheet model ISSM. Compared to SHAKTI (Sommers et al., 2018) and other hydrology models directly integrated into ISSM, external coupling with preCICE allows greater choice of implementation and in the numerical treatment. CUAS and ISSM use different meshes, time steps, and numerical methods. Development of the models can progress independently and neither model is restricted by the choices of the other. The setup for the coupling is minimal. Coupling scheme, data mapping, and coupling time window are easy to adapt.

Our performance experiments show that preCICE coupling does not negatively affect scaling and has minimal computational overhead. Therefore, our choice of a coupling window size of $30\,\mathrm{d}$ is excessively conservative and coupling on the same time scales as the ISSM time step is feasible. While not a comprehensive review of all possible setups, the scaling results gives future users a basis for how to run the coupling efficiently. Parallel coupling widens the range of CPU counts that can be used efficiently. On the other hand, imbalance in solver run times adds overhead. Therefore, serial coupling is faster for lower CPU counts. This imbalance can not be significantly reduced by redistributing CPUs.

This approach offers many opportunities for the development of ice sheet and earth system models. Many independent models can be more easily recombined to try different setups than tightly integrated models. Rather than reimplementing the CUAS model, only a simple reusable preCICE adapter needs to be developed to add CUAS's capabilities to another ice sheet model. It also would be straightforward to add an ocean circulation model to the experiments presented here. We plan to use preCICE to couple ISSM with a new code for capturing calving dynamics in future work.

The ISSM-preCICE adapter aims to be generic, but still has limitations that need to be resolved in future work as described in Sect. 2.3. Accuracy is lost because the coupling interface is based on the mesh vertices instead of finite element nodes. And it is not yet possible to couple the full three-dimensional volume of the ice sheet. Some features of ISSM and preCICE are not supported at this stage.

The CUAS-preCICE adapter is an early prototype and does not follow most of the guidelines for preCICE adapters. Specific adaptations had to be made to CUAS to support the coupling with ISSM. These can be more cleanly integrated into the code.

preCICE has so far not been widely used in the earth system modeling community. While few (if any) ready-to-use adapters relevant to earth system modeling exist at the moment, developing such adapters is a current opportunity. Besides ESM-specific codes, models in frameworks like OpenFOAM, Elmer, or FEniCS/FEniCSx, for which preCICE adapters already exist, can be coupled with minimal effort. This makes these frameworks a good choice for the development of new models. For example, the ice sheet code Elmer/Ice could easily be coupled to CUAS-MPI using the adapters presented in this work. It is an open question whether a generic coupling library like preCICE has significant downsides over a library like YAC (Hanke et al., 2016) that is specialized on earth system models, at least in global setups. To answer this question, we are in the process of evaluating the data mapping between global meshes using preCICE in a Cartesian coordinate system.

This paper was focused on the technical aspects of the coupling and has neither quantified the numerical accuracy of the coupled system nor fully demonstrated its capabilities to represent real world cases. We have also not fully explored all preCICE features. In particular, we will need to make qualified choices for the coupling scheme and data mapping method.

5 Conclusions

In this paper, we presented the software for coupling the ice sheet model ISSM to the subglacial hydrology model CUAS. The coupling is easy to use, adaptable, and extensible due to the generic coupling library preCICE. We have given performance results that show low computational overhead for communication and data mapping. The parallel scaling results will inform efficient use of the software in the future.

The new preCICE adapters will facilitate studies of the interaction between continental ice sheets and the hydrology systems underneath. The generic ISSM-preCICE adapter can also be used in other setups. For example, we are developing a new solver for capturing the ice sheet calving fronts that can be coupled with ISSM to improve upon its existing moving front core. The use of preCICE to integrate ISSM into a global earth system model will be evaluated. Finally, the adapters can be extended to lift some of the limitations described in this paper and open even more use cases.

Multiphysics problems involving ice sheets have always been solved as split uniphysics problems to date. Keyes et al. (2013) suggests a 'coupled until proven decoupled' strategy for multiphysics problems. The coupling we presented here enables such an investigation.

We hope that the software we developed enables researchers to implement and test new ice sheet model capabilities more quickly. In general, researchers should consider preCICE coupling when developing new models or extending existing ones.

Code and data availability. The current version of the ISSM-preCICE adapter is available at https://git.rwth-aachen.de/terrabyte-dnn2sim/issm-precice. Version 0.3.0 of the ISSM-preCICE adapter used in this paper is available at https://doi.org/10.5281/zenodo.15849145 (Abele and Humbert, 2025). The current version of CUAS-MPI is available at https://git.rwth-aachen.de/yannic.fischler/cuas-mpi. Version 0.1 of CUAS-MPI with added preCICE adapter used in this paper is available at https://doi.org/10.5281/zenodo.15782324 (Fischler et al., 2025). Input data, scripts to run the experiments and produce the plots for all the simulations presented in this paper as well as results of performance measurements are available at https://doi.org/10.5281/zenodo.15849146 (Abele et al., 2025)

Author contributions. DA developed the ISSM-preCICE adapter. YF and TK developed CUAS-MPI and the CUAS-preCICE adapter with contributions by DA. TK and DA ran the experiments to test functionality. DA ran the experiments to measure performance. AH supervised the project. CB supervised YF, HJB and AB supervised DA. MM provided support in the use of ISSM. BU and GC provided support in the use of preCICE. DA prepared the manuscript with significant contributions by TK, AH, BU, GC, and MM. All authors commented on and approved all parts of the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. Part of this work was funded by HELMHOLTZ IMAGING, a platform of the Helmholtz Information & Data Science Incubator [grant number: ZT-I-PF-4-026].

We further thankfully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2075 – 390740016 and under project number 528693298, and the support by the Stuttgart Center for Simulation Science (SimTech).

The authors gratefully acknowledge the computing time provided to them on the high-performance computer Lichtenberg 2 at the NHR Center NHR4CES at TU Darmstadt under grant p0020118. NHR4CES is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities.

References

Abele, D. and Humbert, A.: ISSM-preCICE adapter, https://doi.org/10.5281/zenodo.15785544, Zenodo [code], 2025.

Abele, D., Kleiner, T., Fischler, Y., and Humbert, A.: Coupling ISSM and CUAS-MPI: example cases, https://doi.org/10.5281/zenodo.15849145, Zenodo [data set], 2025.

Chourdakis, G., Davis, K., Rodenberg, B., Schulte, M., Simonis, F., Uekermann, B., Abrams, G., Bungartz, H., Cheung Yau, L., Desai, I., Eder, K., Hertrich, R., Lindner, F., Rusch, A., Sashko, D., Schneider, D., Totounferoush, A., Volland, D., Vollmer, P., and Koseomur, O.: preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved], Open Research Europe, 2, https://doi.org/10.12688/openreseurope.14445.2, 2022.

Chourdakis, G., Schneider, D., and Uekermann, B.: OpenFOAM-preCICE: Coupling OpenFOAM with external solvers for multi-physics simulations, OpenFOAM® Journal, 3, 1–25, https://doi.org/10.51560/ofj.v3.88, 2023.

Craig, A. P., Vertenstein, M., and Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1, The International Journal of High Performance Computing Applications, 26, 31–42, https://doi.org/10.1177/1094342011428141, 2012.

Duchaine, F., Jauré, S., Poitou, D., Quémerais, E., Staffelbach, G., Morel, T., and Gicquel, L.: Analysis of high performance conjugate heat transfer with the openpalm coupler, Computational Science & Discovery, 8, 015 003, https://doi.org/10.1088/1749-4699/8/1/015003, 2015.

Fischler, Y., Rückamp, M., Bischof, C., Aizinger, V., Morlighem, M., and Humbert, A.: A scalability study of the Ice-sheet and Sea-level System Model (ISSM, version 4.18), Geoscientific Model Development, 15, 3753–3771, https://doi.org/10.5194/gmd-15-3753-2022, 2022.

Fischler, Y., Kleiner, T., Bischof, C., Schmiedel, J., Sayag, R., Emunds, R., Oestreich, L. F., and Humbert, A.: A parallel implementation of the confined–unconfined aquifer system model for subglacial hydrology: design, verification, and performance analysis (CUAS-MPI v0.1.0), Geoscientific Model Development, 16, 5305–5322, https://doi.org/10.5194/gmd-16-5305-2023, 2023.

Fischler, Y., Kleiner, T., Abele, D., and Humbert, A.: CUAS-MPI with adapter for the preCICE coupling library, https://doi.org/10.5281/zenodo.15782324, Zenodo [code], 2025.

Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., Durand, G., Favier, L., de Fleurian, B., Greve, R., Malinen, M., Martín, C., Råback, P., Ruokolainen, J., Sacchettini, M., Schäfer, M., Seddik, H., and Thies, J.: Capabilities and performance of Elmer/Ice, a new-generation ice sheet model, Geoscientific Model Development, 6, 1299–1318, https://doi.org/10.5194/gmd-6-1299-2013, 2013.

Hanke, M., Redler, R., Holfeld, T., and Yastremsky, M.: YAC 1.2.0: new aspects for coupling software in Earth system modelling, Geoscientific Model Development, 9, 2755–2769, https://doi.org/10.5194/gmd-9-2755-2016, 2016.

Huang, Q., Abdelmoula, A., Chourdakis, G., Rauleder, J., and Uekermann, B.: CFD/CSD coupling for an isolated rotor using preCICE, 14th World Congress of Computational Mechanics and ECCOMAS Congress, https://doi.org/0.23967/wccm-eccomas.2020.081, 2021.

Jordan, J. R.: Calving Model Intercomparison (CalvingMIP) wiki, https://github.com/JRowanJordan/CalvingMIP/wiki, [Accessed 2025-02-26], 2024.

Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo, A., Connors, J., et al.: Multiphysics simulations: Challenges and opportunities, The International Journal of High Performance Computing Applications, 27, 4–83, 2013.

Khrulev, C., Aschwanden, A., Bueler, E., Brown, J., Maxwell, D., Albrecht, T., Reese, R., Mengel, M., Martin, M., Winkelmann, R., Zeitz, M., Levermann, A., Feldmann, J., Garbe, J., Haseloff, M., Seguinot, J., Hinck, S., Kleiner, T., Fischer, E., Damsgaard, A., Lingle, C., van Pelt, W., Ziemen, F., Shemonski, N., Mankoff, K., Kennedy, J., Blum, K., Habermann, M., DellaGiustina, D., Hock, R., Kreuzer, M., Degregori, E., and Schoell, S.: Parallel Ice Sheet Model (PISM), https://doi.org/10.5281/zenodo.14991122, Zenodo [code], 2025.

Larour, E., Seroussi, H., Morlighem, M., and Rignot, E.: Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), Journal of Geophysical Research: Earth Surface, 117, https://doi.org/10.1029/2011JF002140, 2012.

Larson, J., Jacob, R., and Ong, E.: The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models, The International Journal of High Performance Computing Applications, 19, 277–292, https://doi.org/10.1177/1094342005056115, 2005.

Mehl, M., Uekermann, B., Bijl, H., Blom, D., Gatzhammer, B., and van Zuijlen, A.: Parallel coupling numerics for partitioned fluid–structure interaction simulations, Computers & Mathematics with Applications, 71, 869–891, https://doi.org/10.1016/j.camwa.2015.12.025, 2016.

Rüth, B., Uekermann, B., Mehl, M., Birken, P., Monge, A., and Bungartz, H.-J.: Quasi-Newton waveform iteration for partitioned surface-coupled multiphysics applications, International Journal for Numerical Methods in Engineering, 122, 5236–5257, https://doi.org/10.1002/nme.6443, 2021.

Schneider, D. and Uekermann, B.: Efficient Partition-of-Unity Radial-Basis-Function Interpolation for Coupled Problems, SIAM Journal on Scientific Computing, 47, B558–B582, https://doi.org/10.1137/24M1663843, 2025.

Slattery, S. R.: Mesh-free data transfer algorithms for partitioned multiphysics problems: Conservation, accuracy, and parallelism, Journal of Computational Physics, 307, 164–188, https://doi.org/10.1016/j.jcp.2015.11.055, 2016.

Sommers, A., Rajaram, H., and Morlighem, M.: SHAKTI: Subglacial Hydrology and Kinetic, Transient Interactions v1.0, Geoscientific Model Development, 11, 2955–2974, https://doi.org/10.5194/gmd-11-2955-2018, 2018.

Tang, Y.-H., Kudo, S., Bian, X., Li, Z., and Karniadakis, G. E.: Multiscale universal interface: a concurrent framework for coupling heterogeneous solvers, Journal of Computational Physics, 297, 13–31, 2015.

Totounferoush, A., Simonis, F., Uekermann, B., and Schulte, M.: Efficient and Scalable Initialization of Partitioned Coupled Simulations with preCICE, Algorithms, 14, https://doi.org/10.3390/a14060166, 2021.

Uekermann, B., Bungartz, H.-J., Cheung Yau, L., Chourdakis, G., and Rusch, A.: Official preCICE adapters for standard open-source solvers, pp. 210 – 213, GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industry, https://doi.org/10.18419/opus-9334, 2017.

Wolovick, M., Humbert, A., Kleiner, T., and Rückamp, M.: Regularization and L-curves in ice sheet inverse models: a case study in the Filchner–Ronne catchment, The Cryosphere, 17, 5027–5060, 2023.