https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

A parallel implementation of the confined-unconfined aquifer system model for subglacial hydrology: design, verification, and performance analysis (CUAS-MPI v0.1.0)

Yannic Fischler[1], Thomas Kleiner[2], Christian Bischof[1], Jeremie Schmiedel[4], Roiy Sayag[4], Raban Emunds[1,2], Lennart Frederik Oestreich[1,2], and Angelika Humbert[2,3]

[1]Department of Computer Science, Technical University Darmstadt, Darmstadt, Hesse, Germany
[2]Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung, Bremerhaven, Bremen, Germany
[3]Faculty of Geosciences, University of Bremen, Bremen, Germany
[4]Department of Environmental Physics, BIDR, Ben-Gurion University of the Negev, Sde Boker, Israel

Correspondence: Yannic Fischler (yannic.fischler@tu-darmstadt.de)

Abstract. The subglacial hydrological system affects the motion of ice sheets, the ocean circulation by freshwater discharge, as well as marginal lakes and rivers. For modelling this system a porous medium model has been developed, representing a confined-unconfined aquifer system (CUAS) with evolving transmissivity. To allow for realistic simulations, we developed CUAS-MPI, an MPI-parallel C/C++ implementation, which employs the PETSc infrastructure for handling grids and equation systems. We describe the CUAS model and our software design and validate the numerical result of a pumping test using analytical solutions. We then investigate the scaling behavior of CUAS-MPI and show, that CUAS-MPI scales up to 3840 MPI processes running a realistic Greenland setup. Our measurements show that CUAS-MPI reaches a throughput comparable to the throughput of ice sheet simulations, e.g. the Ice-sheet and Sea-level System Model (ISSM). Lastly, we discuss opportunities for ice-sheet modelling, future coupling possibilities of CUAS-MPI with other simulations, and consider throughput bottlenecks and limits of further scaling.

1 Introduction

The dynamics of ice sheets in Greenland and Antarctica is highly related to the conditions at the ice base. At the base of Greenland, the ice is over at least 33% thawed (MacGregor et al., 2022) at the pressure melting point, with melt rates reaching as much as $0.19\,\mathrm{m\,a^{-1}}$ in the inland (Zeising and Humbert, 2021) and up to $57\,\mathrm{mm\,d^{-1}}$ in outlet glaciers (Young et al., 2022). Hence, an extensive subglacial hydrological system is expected to exist. In addition, the margins of the Greenland Ice Sheet are experiencing massive surface melt in summer (Colosio et al., 2021), which is partially stored in supraglacial lakes (Schröder et al., 2020). Most of them drain eventually and deliver the water to the subglacial hydrological system rapidly (Neckel et al., 2020). As the subglacial system is hidden beneath hundreds to thousands of metres thick ice, observations are extremely

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

sparse and establishing a representative mathematical model is challenging. In the past, glaciologists have adopted equivalent porous media approaches from hydrologists de Fleurian et al. (2014); Beyer et al. (2018), meaning that different elements of the hydrological system, such as a thin water sheet, channels and cavity are not individually simulated, but they are in an integrative way represented by the transmissivity of the system, hence the ability to transport water. For simulating large areas, like entire ice sheets in adequate spatial resolution and in a temporal resolution to allow to represent changes on short temporal change, such as seasonal melt water input, an efficient numerical code is indispensable.

The demand on code performance is mainly driven by the ability to simulate a desired amount of time steps, in a runtime that is affordable in terms of compute time. An example of relevance is the projection of the change in the hydrological system until 2100 reflecting effects of alteration in seasonal melt water supply to the base. Beside the number of affordable time steps, the spatial resolution is of importance. In particular around the ice sheet margins, where seasonal melt water reaches the base, sufficiently high spatial resolution is required. This can only be achieved with a efficient codes.

Therefore we developed CUAS-MPI. CUAS-MPI is a parallel C/C++ implementation of the confined–unconfined aquifer scheme (CUAS) presented by Beyer et al. (2018), employing the MPI-parallelization paradigm to enable the use of large compute clusters. CUAS-MPI allows file in- and output in the NetCDF format and uses PETSc Balay et al. (2021) to handle grids and equation systems. We validated the numerical results of CUAS-MPI by comparing it with the analytic solution of pumping tests. On a realistic setup of Greenland, we employed CUAS-MPI then with up to 3840 processes, gathering performance data with Score-P (Knüpfer et al., 2012).

The paper is structured as follows: first we explain the underlying model and the software design of CUAS-MPI. In Section 3 we then describe a pumping test model problem where analytical solutions are known and compare them to the results of CUAS-MPI. We then describe a realistic Greenland setup in Section 4. This setup then is used in an investigation of the performance and scaling behaviour of CUAS-MPI in Section 5. Finally we discuss and conclude our work.

2 CUAS-MPI

2.1 Model

There are different perspectives on the purpose of simulating the subglacial hydrological environment: for planning hydro power facilities and simulating ocean dynamics, the freshwater flux is the primary interest, while for ice sheet modellers, the water pressure is the key quantity as it affects sliding of glaciers. In the past, subglacial hydrological modelling has developed from simple routing schemes unable to represent channels underneath the ice to simulations representing such features either individually or in an equivalent porous medium (EPM) approach (de Fleurian et al., 2018). The code that we present here is following such an EPM approach, which is a compromise in representing the physics of a hydrological system with a multitude of features and the ability to simulate large areas over in high temporal resolution. The hydrological system beneath the ice sheet is simulated by an EPM layer in which the void space is fully saturated and can thus be described by Darcy flow. The transmissivity of the layer is then representing the different forms of the hydrological system. Areas with high permeability represent very efficient water transport, while low permeability represents an ineffective water system. Very efficient water

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions
Open Access
EGU

transport is thought to take place through a channelised system, while a distributed system is known to lead to inefficient transport. The ice sheet is acting as a confining layer of the aquifer, however, it may happen that water supply is not sufficient to keep the water system fully saturated. To this end, an unconfined layer is incorporated, capturing the dynamics if the head is falling below the layer thickness allowing for further water drainage. The resulting confined-unconfined aquifer system is described following Ehlig and Halepaska (1976).

The model consists of an evolution equation for the hydraulic head for Darcy flow (similar to the groundwater flow equation). This equation is the only PDE to be numerically solved, using a solver provided by PETSc framework. A second major equation is describing the change in transmissivity with time based on melting, creep and cavity formation. However, we update the transmissivity using an explicit Euler-step that does not involve a PETSc solver. We present the details of the model equations in the Appendix A and information on the numerics in Appendix B. More on the model equations can be found in Beyer et al. (2018).

2.2 Software-Design

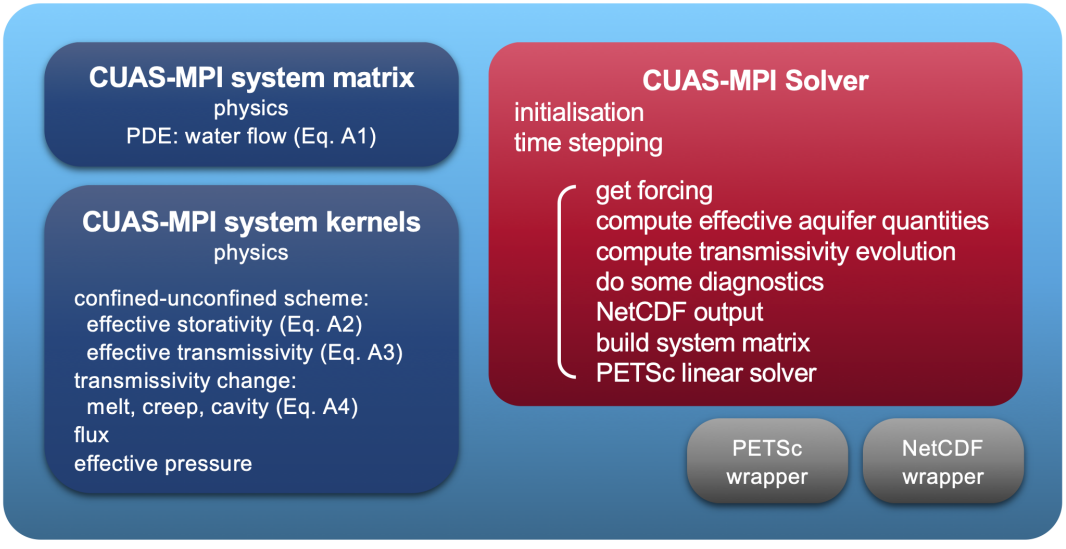

Figure 1. Components of CUAS-MPI. The physics modules of the mathematical model are shown in dark blue. The actual solver is sketched in the red box. Grey boxes denote wrappers interfacing to PETSc and NetCDF.

The starting point of this project was a serial implementation of CUAS, partially written in Python, (Beyer et al., 2018). While producing good numerical results, its performance was too low for larger setups such as Greenland. Our new software design for CUAS-MPI is based on this earlier implementation: CUAS-MPI again uses regular two-dimensional grids and the physics kernels are implemented analogously to the Python implementation. They represent individual equations of CUAS (see Figure 1), computed on data local to an MPI process. The kernels are called from the time stepping sequence of the CUAS-MPI solver, which handles the distributed grids and creates and solves the equation system. Time stepping parameters are optionally

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

described by command line parameters or a time step file. Our implementation is backwards compatible to the setup of the serial version, supporting input data in NetCDF format and the same command line parameters. In addition, CUAS-MPI is able to restart from previous runs.

We use the well-known PETSc parallel math library to handle grids and equation systems in CUAS-MPI, with an object-oriented interface we designed to handle our matrices, vectors and grids. In particular, we employ the distributed memory features of PETSc for grid creation, ghost cell update, and matrix and vector distribution in the context of two-dimensional structured grids. The grid access is guarded by read and write handles which automatically trigger ghost cell updates after data was written. To distribute data across MPI processes, we use a PETSc feature that ensures that the assembly of matrices and vectors for the equation system is compatible with the distribution of data across the processes, thereby ensuring low communication costs during matrix and vector assembly. The solver can be selected by the user from the list of available PETSc solvers. We use the iterative GMRES solver for the Greenland simulations. Within our testing environment we can also use the direct solver MUMPS if the problem size allows for comparison.

The regular two-dimensional grids of CUAS-MPI are stored in the standardized file format NetCDF for in- and output. Different libraries implement parallel read and write operations of NetCDF-files: The NetCDF implementation (Rew and Davis (1990)) with HDF5, PnetCDF (Latham et al. (2003)) and ParallelIO (Hartnett and Edwards (2021)). As all three parallel I/O libraries have advantages and disadvantages, we implement an adapter which provides an uniform interface of the features we require in CUAS-MPI. The NetCDF library is always used for reading, and we also employ it in our experiments. CUAS-MPI supports four levels of output: small, normal, large and xlarge (Table 1). The small output includes only the absolutely necessary fields. All other fields can be derived. In the three more complex output configurations we write additional analytical information, which is computed in the CUAS-MPI solver pipeline.

We read command line parameters through the cxxopts library (https://github.com/jarro2783/cxxopts) and write log output using spdlog (https://github.com/gabime/spdlog).

configuration	enabled fields for output
small	head, transmissivity
normal	bed elevation, water input,
	dT/dt due to channel wall melt & creep opening & cavity opening,
	effective pressure, flux
large	ice thickness, effective layer transmissivity & storativity
xlarge	ice pressure

Table 1. Output options of CUAS-MPI, the categories are inclusive, each options includes the options above.

2.3 Workflow

A typical workflow to run a simulation is described by a setup script. In that script the model domain and the grid are defined and used to create the mask including boundary conditions. This mask is one of the input fields. In addition the fields for

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

bedrock topography, ice thickness and water input are prepared into one NetCDF file. The script setup can be performed in any environment comfortable for the user.

The next step is the initialisation. The initial head and thus the initial water pressure can be selected via specific named CUAS-MPI options to be spatially uniform, following the bed elevation or to be equal to the ice overburden pressure. The initial transmissivity can be set to a spatially uniform value via a command line option. Full control over the initial conditions is further given using the CUAS-MPI "restart from file" capabilities. A restart could be done from e.g., the last time slice of a previous run, from arbitrary fields for head and transmissivity provided by the user or from the output of a coarse resolution spin-up that has be remapped onto the new grid outside of CUAS-MPI. All file names and parameters used for a model run are stored in the NetCDF output file for later reference.

The last step is to set up the actual run script that serves the needs of the cluster environment. In our case this is the slurm scheduler. This step includes setting up the command line options to control CUAS-MPI's physics and time stepping, setting up the the PETSc Solver via the environment variable `PETSC_OPTIONS`, as well as the memory, node, core and runtime configuration on the cluster.

A typical use case may be the simulation of a seasonal hydrological cycle. To this end, a spin-up would be conducted first to retrieve a steady-state system. This may be done on a coarse spatial resolution first, followed by a refinement using another grid using the restart option. Here the seasonal experiment starts, with one simulation without a seasonal forcing serving as the control run and others with seasonal forcing. If a particular target area needs to be simulated in even higher resolution, a nesting approach can be employed. Also the desired output frequency, as well as the variables to be stored, can be adjusted to the needs of the user.

Another typical application is a projection of the change of the hydrological system over a larger time period. For this purpose a spin-up is required, too. To balance the costs for the long simulation period, the user can reduce spatial resolution if the science questions allows, or reduce the output to only annually write files with the essential physical fields. In all cases the user needs to choose an adequate time step.

In order to enable scheduling, we discuss below the simulation costs for a realistic setup with daily and annually written output and different grid resolution. The reader can plan own simulations based on these numbers reasonably well.

3 Validation using analytical solutions

To validate the results of CUAS-MPI we compare the numerical solution of equation (A1) in confined and unconfined cases with suitable analytical and semi-analytical solutions. We test our implementation using an exact steady-state as well as an exact transient solution. Both can be found by the Method of Manufactured Solutions (MMS) for two-dimension diffusion problems (e.g. Oberkampf and Roy, 2010). The exact MMS solutions so far only consider spatially uniform and steady aquifer properties and are limited to the confined aquifer case.

By performing pumping tests we are further able to test the implementations of Dirichlet and Neumann boundary conditions. The problem we simulate in a pumping test involves a horizontal aquifer of uniform thickness b, constant conductivity K and

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

a pump of constant rate Q that is located at the center of the domain and that fully penetrates the aquifer. We consider two situations, one in which the aquifer is confined throughout the flow and has a constant effective transmissivity $T_e = Kb$, and the other in which the aquifer is unconfined and has an effective transmissivity proportional to the head, $T_e = Kh$.

The confined case has an analytical solution on an unbounded domain (Theis, 1935), which can be tested on a bounded numerical domain as long as the flow is far from the boundaries. Moreover, analytical solutions for a bounded domain can be constructed based on the unbounded solution using the method of images, since equation (A1) with constant transmissivity is linear (Ferris et al., 1962). We choose to verify the two types of boundary conditions implemented in CUAS-MPI by considering the case where a pump is at equal distance from two Dirichlet boundaries having zero hydraulic head ($h(x,t) = h(y,t) = 0, \forall\, x,y \to -\infty$), and two Neumann boundaries across which the flow is zero ($\frac{\partial h}{\partial x} = \frac{\partial h}{\partial y} = 0, \forall\, x,y \to \infty$). The analytical solution for such a configuration consists of a superposition of image wells placed across the domain boundaries (Ferris et al., 1962), where the solution accuracy grows with the number of image wells. Specifically, the analytical solution can be described in terms of the drawdown $s = h(x,y,0) - h(x,y,t)$ of an initially uniform hydraulic head, as the series

$$s = \overbrace{\frac{Q}{4\pi T}\left[W\left(\frac{r_{\mathrm{pm}}^2 S}{4Tt}\right) \pm \sum_i W\left(\frac{r_{\mathrm{im}}^2 S}{4Tt}\right)\right]}^{\text{bounded solution}}, \qquad (1)$$

where S is the storativity, t is time, $T = T_e = Kb$ in the confined solution, r_{pm} is the distance from the pump to the point of measurement, r_{im} is the distance from the ith image well to the location of measurement, and W is the well function (Theis, 1935) with $(-)$ referring to the Dirichlet boundaries and $(+)$ referring to the Neumann boundaries.

To verify the results of CUAS-MPI in the non-linear unconfined case we approximate the drawdown s' of an unconfined aquifer from the drawdown of an equivalent confined aquifer s (Eq. 1) with $T = Kh(t=0)$ through the relation $s' = b - \sqrt{b(-2s+b)}$. This relation is based on the Dupuit-Forchheimer assumption that the horizontal flux is greater than the vertical flux, and provides a good prediction to the drawdown at a large distance from the pump compared to the aquifer thickness (Jacob, 1963).

The numerical simulation for the confined case is set up with a domain size of $2000\,\mathrm{m} \times 2000\,\mathrm{m}$, $b = 100\,\mathrm{m}$, $S_s = 1 \times 10^{-6}$, $K = 4.16 \times 10^{-6}\,\mathrm{m\,s^{-1}}$ and an initial hydraulic head of $h(x,y,t=0) = 300\,\mathrm{m}$. For the unconfined case the domain size is $4000\,\mathrm{m} \times 4000\,\mathrm{m}$ and the initial hydraulic head is $h(x,y,t=0) = 99\,\mathrm{m}$. We define the specific yield as $S_y = 0.4\,\mathrm{m}$, which represents a subglacial hydrology system with an EPM approach (de Fleurian et al., 2014). The pumping well in both cases has a constant rate of $Q = 0.1\,\mathrm{m^3 s^{-1}}$.

For the confined case, the CUAS-MPI result is in agreement with the analytical solution on an unbounded domain until $\approx 3 \times 10^4\,\mathrm{s}$ when the flow starts to interact with the boundaries of the numerical domain (Fig. 2). Afterwards, the results follow the bounded solution (Ferris et al., 1962), with an accuracy that grows with the number of image wells (Fig. 2). In the unconfined case the simulation results agree with the approximated analytical solution (Jacob, 1963) (Fig. 2). These CUAS-MPI simulations were performed with the two implemented solvers, the direct (MUMPS) and the iterative (GMRES) one, and the results of the two computational methods were indistinguishable.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

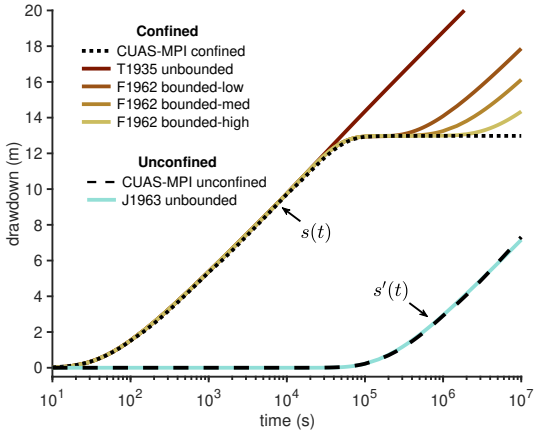

Figure 2. The model solutions of CUAS-MPI for the drawdown s from a confined aquifer and the drawdown s' from an unconfined aquifer. Analytical solutions for a confined aquifer over an unbounded domain (T1935 unbounded) and over a bounded domain, with 24, 80, and 288 image wells (F1962 bounded-low, -med, -high, respectively), and for an unconfined aquifer over an unbounded domain (J1963 unbounded). The point of measurement for both simulations is 80 m away from the pumping well.

4 Greenland Setup

The Greenland setup for this study consists of a rectangular area comprising grid points of the subglacial hydrological system (red in Fig. 3) and the surrounding area, that either consists of ocean or land grid cells. Boundary conditions are determined by the type of grid cell next to the margin grid cells of the subglacial system. In case of a land terminating margin, the boundary condition is no flow (homogeneous Neumann boundary condition), while a transition to the ocean is a Dirichlet boundary condition for the head, namely the ocean water pressure. The type of grid cell (mask), the ice thickness, the bedrock and surface topographies are based on the BedMachine Greenland dataset (Morlighem, 2021; Morlighem et al., 2017, Version 4) also used in (Christmann et al., 2021). To summarise it briefly, the subglacial water is drained into fjords, as land terminating margins are prohibiting outflow. The BedMachine dataset is originally available in 150 m resolution (G150), which we regrid to 300 m (G300), 600 (G600), 1200 m (G1200) and 2400 m (G2400) resolution using conservative remapping for the geometry and near-neighbour interpolation for the mask. Ice sheet basal melt from the Parallel Ice Sheet Model (PISM) output (Aschwanden et al., 2016, 1200 m resolution) is regridded to the CUAS meshes using bi-linear interpolation. Large areas consisting of small glaciers get a spatially uniform ice thickness of 1 m assigned in the BedMachine dataset due to insufficient data coverage. This is particularly visible along the south-eastern and eastern margin of the ice sheet. Those areas are also not well resolved in the PISM model. Therefore a minimum ice thickness constraint of 10 m is applied to eliminate thin marginal areas. The observed surface velocity (MEaSUREs, Joughin et al., 2016, 2018) is used to check, if areas with at least 30 $\mathrm{m\,a^{-1}}$ are included in the mask. If not, the minimum ice thickness constraint is applied and the bed elevation is adjusted to be consistent with the surface elevation from BedMachine. Further, a flood-filling algorithm (van der Walt et al., 2014) is used to select only the connected

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

grid points from the main ice sheets without peripheral ice caps and glaciers to ensure consistent coverage from BedMachine and the PISM model output for ice sheet basal melt. This step is important, because missing data in the basal melt forcing would degrade the solver performance in CUAS-MPI.

The resulting numbers of total and active grid points are given in Table 2. Water input is the basal melt rate presented in Aschwanden et al. (2016). The model parameters (see Appendix A) are mainly taken from Beyer et al. (2018) with only two exceptions. We use a specific yield of 10^{-6} instead of 0.4, and a minimum transmissivity bound $T_{\min}$ of 10^{-8} instead of $10^{-7}\,\mathrm{m^2\,s^{-1}}$. Those changes changes are found to result in smoother hydraulic head in areas of no or only very little basal water supply. For the purpose of this study, the exact representation of the Greenlandic hydrological system is not of primary importance, as we analyse the performance of the model, but we represent the Greenlandic Ice Sheet sufficiently realistic to be able to infer from the outcome in this study the computational costs for other simulations. The hydraulic head is initialized to to be equal to the ice overburden pressure at each grid point and the initial transmissivity is spatially uniform ($T(t=0) = 0.2\,\mathrm{m^2\,s^{-1}}$). The convergence criteria for the iterative GMRES solver are configured as $\mathrm{rtol} = 10^{-7}$ (relative residuum norm) and $\mathrm{atol} = 10^{-5}$ (absolute residuum norm) with a maximum number of iterations set to 10^5.

name	resolution	grid points	% of active cells
G150	150 m	187 459 428	39.08
G300	300 m	46 879 140	39.05
G600	600 m	11 726 928	39.03
G1200	1200 m	2 935 305	39.01
G2400	2400 m	734 720	39.06

Table 2. Characteristics of the Greenland setups.

5 Performance of CUAS-MPI on a representative Greenland Setup

CUAS-MPI was developed to enable high-performance and high-throughput simulations on up-to-date HPC systems, which typically consist of many-core nodes connected by a high-speed network. To assess the performance of CUAS-MPI, we present performance data of CUAS-MPI on the Greenland Setup described in (Section 4). To be able to compare performance and scaling behavior of CUAS-MPI across different resolutions, we run, for every resolution, with the same time step of one hour and compute 24 time steps, i.e. one day. As linear solver, we use GMRES with a Jacobi preconditioner.

We employ GCC 11.2, Open MPI 4.1.4, PETSc 3.17.4 and NetCDF 4.7.4 with HDF5 1.8.22 for our performance experiments. To instrument the code for measurements, we use Score-P 7.1. The code is compiled using optimization level "-O3" and native processor architecture flags "-march=cascadelake". All experiments are conducted on dedicated compute nodes of the Lichtenberg HPC system with two 48-core Intel Xeon Platinum 9242 processors and 384 GB of main memory each, connected with an InfiniBand HDR100 network providing point-to-point connections between nodes. Due to temporary energy saving measures, turbo-mode has been disabled and the base frequency of the chips reduced by 100 Mhz to 2.2 GHz. Every experi-

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

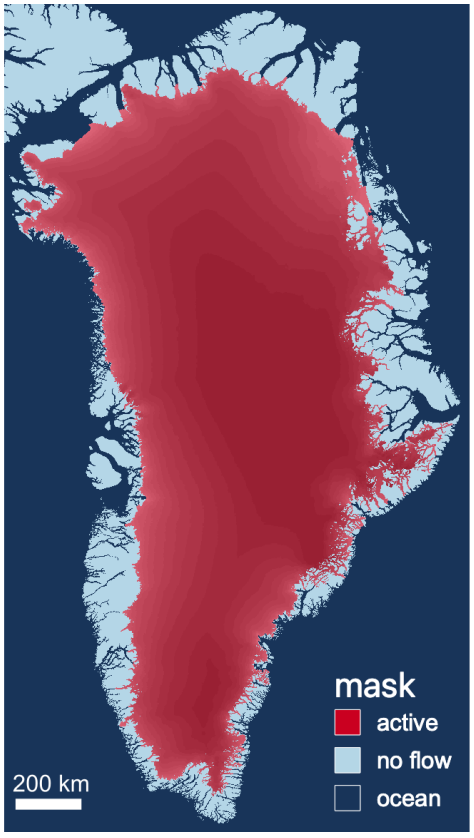

Figure 3. The Greenland setup used for this study. Red color represents the area where the hydrological system is computed, dark blue denotes ocean and pale blue land area. Ocean and land leads to different boundary conditions. The red area is shaded with the bed topography.

ment is repeated three times, and the average runtime is reported. For all runs of the models with G1200 or finer resolution, we see a relative standard deviation of less than 5%. For the coarsest G2400 model, we observed a relative standard deviation of 15 %, which we believe to be in part due to the very short running time of the code in this case.

5.1 Thread Occupancy of Compute Nodes

The two processors on one node share access paths to main memory and they have a sophisticated cache architecture. The sparse matrix setup employed in the numerical solvers requires a significant amount of memory accesses and, employing all cores, will more than saturate the available memory bandwidth. Thus, it is worthwhile to explore whether a lower thread occupancy on a node, which provides more individual bandwidth to the remaining threads, is not, in the end, the better choice. To this end, we tested CUAS-MPI on the Greenland setup with G600 resolution using full, half, quarter and one-eighth occupied compute nodes of the HHLR. Each thread is pinned to one CPU core and realizes one MPI process. Hence, in our discussion and for our setup, the notions of "thread" and "MPI process" are used interchangeably.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions
Open Access
EGU

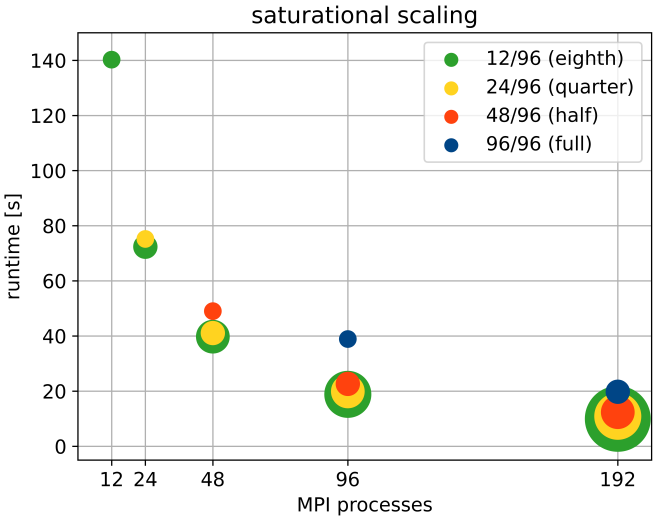

Figure 4. Performance of CUAS-MPI on full (96/96), half (48/96), quarter (24/96), eighth (12/96) occupied nodes of HHLR. A detailed explanation is provided in subsection 5.1.

The result is shown in Figure 4. Here, the color of a circle indicates the thread occupancy ratio, from one-eight, i.e. 12 threads of a 96-core node in green, up to the use of all 96 threads on a node, in blue. The size of the circle indicates the hardware investment. The smallest circle indicates that only one node was used, the next size up indicates two nodes, then four nodes, and lastly 16 nodes. The center of the respective circle indicates the runtime that was achieved with this setup,

Hence, the leftmost green circle (12 threads on one node, resulting in a runtime of about 4900 seconds) is as large as the blue circle in the middle (96 threads one one node, resulting in a runtime of about 1800 seconds). On the other hand, the rightmost blue circle is bigger, because here we need to employ two nodes at full occupancy to realize the 192 threads, resulting in a runtime of about 1000 seconds. Almost concentric circles, such as the red, yellow and green circles for 192 processes, then indicate that the additional hardware investment does not pay off, as the corresponding runtime can be achieved with the setup corresponding to the smallest circle.

We see that there is not much difference in the runtime of one node with 48 MPI processes (red circle at 48 processes) to one node with 96 MPI processes (blue circle at 96 processes), but 96 MPI processes on two nodes (red circle at 96 processes) are about twice as fast. The rightmost circles show that 192 MPI processes are faster than 96 MPI processes and that we should use a distribution of four nodes (red circle), because it is faster than two nodes (blue circle) and not significant slower than four (yellow circle) or eight (green circle) nodes. Similar observations can be made for 96 MPI processes, where an occupancy lower than a half results in some, but not very significant, speedup. Hence, we consider 48 MPI processes per node, i.e. half thread occupancy of a node, as a good trade-off between getting the solution fast and using the hardware reasonably and employ 48 MPI processes on each node to analyze the throughput and scalability of CUAS-MPI in the following subsections.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions
Open Access
EGU

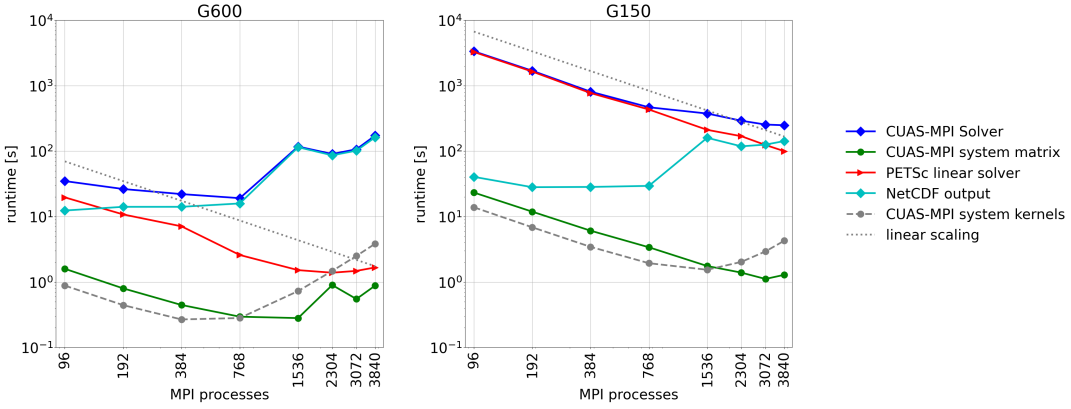

Figure 5. Runtime of 24 time steps of CUAS-MPI solver pipeline writing a single output running G600 and G150.

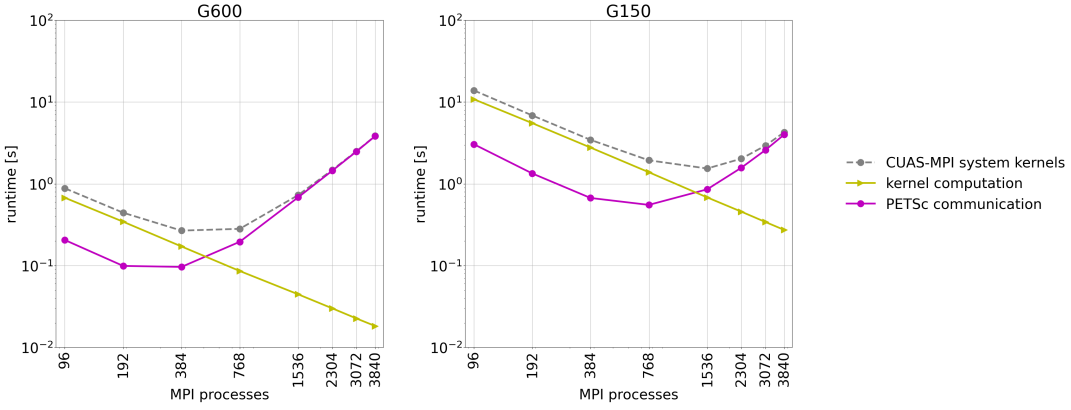

Figure 6. Sum of the CUAS-MPI system kernels runtime of 24 time steps running G600 and G150.

5.2 Runtime and throughput

We identified four functional categories in CUAS-MPI, whose individual performances are worth differentiating: CUAS-MPI setup, CUAS-MPI Solver, PETSc linear solver, and I/O interface. "CUAS-MPI setup" contains initial model loading from disc and the setup of necessary grids. The runtime of this code component is not significant in large productive runs and we do not consider it further. The "CUAS-MPI Solver" category, which is also shown in Fig. 1, includes all kernels running on CUAS-MPI grids, the creation of the equation system and the post-processing of the solution vector. In particular, the "CUAS-MPI solver" also calls the code of the third category, the "PETSc linear solver". This is the library call to the solver, in our case the GMRES implementation. Finally, the "I/O interface" category includes all calls, which write data to file during the simulation. The time taken by I/O will, in general, greatly depends on the I/O and network capabilities of the current HPC system and on the output frequency, which depends on the particular experimental setup in question.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

In Figure 5 we show the runtime of different code categories for 24 time steps of CUAS-MPI with one output to disc. The top line of the runtime plots shows the runtime of "CUAS-MPI Solver", i.e. the aggregate runtime for the entire time stepping code, including all the other routines listed. Here, "CUAS-MPI system matrix" denotes the creation of the equation system, i.e. the computation of the matrix entries and the matrix assembly, that is solved in the "PETSc linear solver". The "CUAS-MPI system kernels" category contains the characteristics of the EPM, such as the confined-unconfined scheme, transmissivity change and flux, as two dimensional fields. They are needed for the computation of the system matrix entries and for diagnostic purposes. Finally we display the runtime of "NetCDF output", which writes a single output of configuration "large" (see table 1).

First, we note that NetCDF output routine does not scale at all. Its runtime is, for both G600 and G150, essentially flat up to 768 MPI processes, and then increases by an order of magnitude. We did not investigate this somewhat surprising issue further, as the performance of NetCDF is not the focus of this work.

Considering the coarser G600 grid, we note that, ignoring NetCDF output, the PETSc linear solver dominates runtime and scales up to 2304 MPI processes, where it is overtaken by the "CUAS-MPI system kernels", whose runtime increases past 768 MPI processes, due to the increasing communication overhead between MPI processes and decreasing computation performed on auch MPI process.

For the finer grid (G150), on the other hand, we see a continual decreases in runtime as we increase the number of MPI processes. Ignoring NetCDF output, the PETSc linear solver dominates the runtime, but scales in an almost linear fashion up to 3840 MPI processes. In particular, we see approximately linear scaling also for the "CUAS-MPI system kernels" and the system matrix routine up to at least 768 MPI processes. Then the "CUAS-MPI system kernels" and thereafter the system matrix creation reach their scaling limit and their respective runtime increases.

Disregarding file output operations, the "CUAS-MPI system kernels" category is the first component reaching its scaling limit. As all kernels behave in the same way, we consider them as a group. Figure 6 shows the accumulated runtime and the separated runtime of kernel computation and grid data exchange communication caused by PETSc. We notice, that the computation scales expectedly linearly, but the grid exchange prevents further scaling.

5.3 Throughput

In our studies of throughput, i.e. how many simulated system years we can run in a day of compute time (simulated years per day, SYPD), we employed the same time step (1 hour) for the different resolutions in order to allow a sensible comparison of the various code components across resolutions and process counts. This is conservative for the lower resolutions, as generally coarser grids allow larger time steps, because time steps are adapted to fit the stability conditions of the simulation. The different grid resolutions require different numbers of iterations for convergence.

We simulate 1 day (24 time steps) and write one output per day using the large configuration (see Table 1)and measure the runtime. Based on that measurement we compute the runtime for 365 days and compute SYPD.

In Figure 7, we compare the throughput and runtime scaling of different grid resolutions on the Greenland setup, from 96 up to 3840 MPI processes. The peak throughput of the coarsest grid G2400 of about 550 simulated years per day is reached at 192 MPI processes and annual writing, while the finest grid G150 can profitably utilize more than 3840 MPI processes for

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

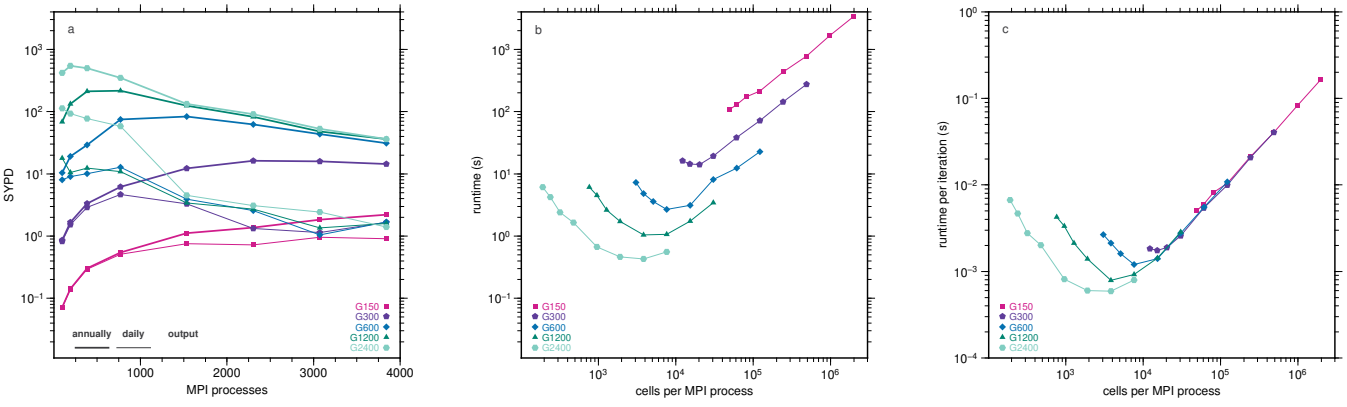

Figure 7. Throughput (panel a) versus MPI process, runtime (panel b) and runtime per iteration (panel c) versus number of cells per MPI process.

both daily and annual writing. For G150, we derive a maximum throughput of approximately one simulated years per day if daily output is written and two simulated years per day if output is written annually.

In general we see that for smaller grids there is no sense in using a large number of processes, as there is not enough work to be done for an efficient parallelization. On the other hand, for the G150 grid (which has almost 100 times as many grid points as the G1200 resolution, see Table 2), it does make sense to use more processes to increase throughput: With 768 and 1536 processes, we can compute 198 and 402 model days per compute day. So, in particular, going from 768 to 1536 processes, we increase throughput by a factor of two.

In the panel b of Figure 7 the runtime of the CUAS-MPI pipeline without output is visualized. We see that larger grids need more cells per MPI process than smaller grids for efficient parallelism. This is caused by the fact that more MPI processes, in general, cause more communication and synchronization overhead and more computation is needed to offset this. While the scaling limits of the four coarse grids is included in the graphs, the finest grid G150 still has potential. We expect, that the sweet spot correlates with the other grids.

In order to remove the effect of increasing number of iterations in the linear solver with resolution in the analysis, we also show in Fig 7c the runtime per iteration versus cells per MPI process. This figure reveals that the minimum of the curves is not affected by the number of iteration, as expected. The minimum of the total runtime per iteration is increasing with spatial resolution, however, the spread is less than the total runtime per year, showing that the total runtime is driven by the increase in number of linear solver iterations with resolution.

6 Discussion

The throughput shown by CUAS-MPI enables ice sheet wide simulations in high, but potentially not highest resolution. Although the code performs well, the number of time steps required for a seasonal cycle limits the number of years that can be

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

simulated with the amount of core-hours usually available for such runs. A simulation covering the 90 years from 2010 to 2100 in 600 m spatial and 1 hour temporal resolution requires a wall-clock time of 1350 hours (56 days) on 384 MPI processes. Such a simulation can be performed a few times, but is not feasible for ensemble simulations for difference atmospheric input. Moreover, a high computational demand arises from spin-ups for having a proper initial state for simulations and a control run needed for assessment of the projection run. So, with CUAS-MPI, panGreenland simulations are still challenging, but they are feasible. The costs are also emphasising the need for efficient coupling of ice-sheet and hydrology models.

Although we have applied the code to an entire ice sheet, applications to alpine glaciers might of interest for a larger community. As an example we consider the Kanderfirn Glacier (Switzerland) which has a size of about $12\,\mathrm{km}^2$. To compute this glacier in the $10\,\mathrm{m}$ resolution would result in 120.000 grid points. Assuming hourly time steps and daily output, one year would require about 3 minutes wallclock time on 384 MPI processes (based on G150 performance).

While the main intention for developing a hydrological model was the influence to the dynamics of the ice sheet via sliding, other disciplines benefit from these simulations, too. Oceanographers are dealing with the simulated water flux across the grounding line or glacier terminus as freshwater input into the ocean/fjord system. Hydrologists need freshwater flux to estimate the discharge available for hydropower as in (Ahlstrøm et al., 2009; Braithwaite and Højmark Thomsen, 1989).

The physical (and mathematical) model of CUAS-MPI may, of course, not be powerful enough to simulate the complex physics sufficiently well in other cases of interest. However, with CUAS-MPI domain scientists now have now a tool to conduct simulations on relevant areas of interest to figure out strengths and weaknesses of the physical basis of the model is. The code has been designed in a modular fashion to allow for extensions of the underlying physical model.

In our work, the next step will be the coupling of CUAS-MPI to an ice sheet model, in our case the finite element based Ice Sheet and Sea Level System Model (ISSM, issm.jpl.nasa.gov) (Larour et al., 2012). There are inherent system dependencies between the physical quantities in both models, ice sheet and hydrology, hence a coupling needs to consider the ingestion of a larger number of fields from the ice sheet code into CUAS-MPI, while there is only the effective normal pressure to be fed into the ice sheet model. To this end, we plan to employ the preCICE coupling library for partitioned multi-physics simulations (precice.org and Bungartz et al. (2016)).

For this endeavour, it is worth comparing the computational costs of the ice sheet and hydrology models. Comparing the SYPD for a Greenland setup in the ice sheet model in the highest tested resolution (G250) presented in (Fischler et al., 2022) with the G150 resolution of CUAS-MPI, we find that the computional costs are comparable for both models. As a consequence, both simulations would need the same time per year, which which is preferable for achieving low idle time at synchronization points in coupled runs.

We also see the possibility to integrate subglacial hydrology of CUAS-MPI directly in ice sheet codes like ISSM. The advantage of coupling the two codes versus a monolithic solution are of different kinds. First a standalone implementation supporting a generalized coupling interface can be used in many other projects. Second it is fully independent of other codes discretization and prevents any inconsistencies. Finally, we see a huge advantage in the implicit additional parallelization. While a monolithic implementation most likely causes a serial execution of modules in a multi-physics environment or additional engineering, a coupled implementation runs ice sheet and subglacial hydrology simulation on dedicated nodes in parallel.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023


Geoscientific Model Development Discussions

Certainly coupling generates additional overhead, but we see no necessity of high frequent data exchange, e.g. each time step, and low frequent data exchange will be affordable.

Considering the performance of CUAS-MPI, we see that the NetCDF performance plays an important role and limits scalability of CUAS-MPI on the HHLR system that we are running on. Hence, we suggest to use low-frequency output, e.g. annually. In addition, the smallest output configuration can be used to reduce the output costs, and then CUAS-MPI restart can be used to compute additional fields afterwards. We have not investigated NetCDF performance further, as I/O performance is highly system dependent. We suspect that more efficient NetCDF implementations are available, and have encapsulated I/O in our software design so that other implementations can be linked in easily.

Outside of I/O, the runtime of CUAS-MPI is dominated by the runtime of the PETSc linear solver (unless the local work becomes too little). Our scaling tests have shown that the solver still has more scaling potential in the case of large setups, but we reached the scaling limit of coarser grids. Future throughput improvements depend on improvements of the PETSc software infrastructure, which, however, is being continually improved and updated.

The scaling of the CUAS-MPI system kernels, which generally is of lower importance overall, might be further improved by asynchronous PETSc communication or additional thread parallelism per MPI process to increase computational granularity. Hybrid parallelism would also enable the opportunity to use less parallelism in less scalable regions like file output and more parallelism in the linear solver.

7 Conclusions

CUAS-MPI is a newly software-engineered code based on the model of Beyer et al. (2018). It has been validated using analytical solutions of pumping tests. The code is instrumented for performance measurements, that have been conducted on the Lichtenberg-2 HPC infrastructure at TU Darmstadt. Our study demonstrate, that it performs well and scales up to 3840 MPI processes. Some performance limitations result from the current implementation of PETSc and NetCDF, and we anticipate that CUAS-MPI will profit from performance improvements in these software infrastructures down the line.

As subglacial hydrology and ice sheet evolution are strongly related to each other, runtime coupling of CUAS-MPI and ice-sheet simulations like ISSM is an important topic. Therefore we see necessary enhancements in the implementation of inter-simulation data exchange and the adaption of the model to support changing simulation domains in transient runs.

Code and data availability. We published our repository on https://github.com/tudasc/CUAS-MPI. For the measurements presented in this paper, version 0.1.0 was used, which is available at github and through the following address https://doi.org/10.5281/zenodo.7554686. Our profiling data is available on https://doi.org/10.48328/tudatalib-1034.

Author contributions. Y.F., C.B., T.K. and A.H. planned the project. Y.F. and C.B. developed the software design. Y.F., L.O., R.E. and T.K. implemented the code. Y.F. conducted all performance measurements. Y.F., C.B., T.K. and A.H. analyzed the performance measurements.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

T.K. developed the Greenland setup. T.K. and A.H. run the code for polar applications. J.S. and R.S. developed the pumping test concept, J.S. conducted the tests. All authors discussed the performance analysis and wrote the manuscript.

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

Acknowledgements. 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.

Parts of this work were funded by the German-Israeli Fund GIF under grant number I-1493-301.8/2019.

The authors thank Vadym Aizinger (University of Bayreuth) for useful discussions. YF thanks Alexander Hück (TU Darmstadt) for helpful discussions.

Appendix A: Model equations

The dynamics of the effective porous medium layer is driven by two main evolution equations: one for the hydrological head h and one for the transmissivity T. Switching between the confined and unconfined aquifer system is facilitated by the effective variables for storativity and transmissivity by evaluating the head above bedrock Ψ in comparison the effective layer thickness b. To allow a smooth transition between the confined and unconfined system, a range d is introduced.

The vertical integrated mass balance for Darcy systems is given by

$$S_\mathrm{e}(h)\frac{\partial h}{\partial t} = \nabla \cdot (T_\mathrm{e}(h)\nabla h) + Q \tag{A1}$$

with the effective storativity S_e, the effective transmissivity T_e and the water input Q. The effective storativity reads as $S_\mathrm{e}(h) = S_\mathrm{s} b + S'(h)$ with the specific storage S_s and

$$S'(h) = \begin{cases} 0, & b \le \Psi \\ (S_\mathrm{y}/d)(b-\Psi), & b-d \le \Psi < b \\ S_\mathrm{y}, & 0 \le \Psi < b-d \end{cases} \tag{A2}$$

in which S_y is the yield storativity that is specific for the porous medium.

The effective transmissivity varies too between the confined and unconfined system:

$$T_\mathrm{e}(h) = \begin{cases} T, & b \le \Psi \\ \dfrac{T}{b}\Psi, & b > \Psi. \end{cases} \tag{A3}$$

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

As soon as the head sinks below the aquifer height, only the saturated section contributes to the estimation of the transmissivity. In the confined case $b \leq \Psi$, the temporal change of transmissivity is computed by

$$\frac{\partial T}{\partial t} = \frac{g\rho_{\mathrm{w}} K T}{\rho_{\mathrm{i}} L}(\nabla h)^2 - 2An^{-n}|N|^{n-1}NT + \beta|\boldsymbol{v}_{\mathrm{b}}|K \tag{A4}$$

in which K is the conductivity. The first term on the right hand side represents melting, the second term creep opening/closure and the last term the formation of cavities. Melting depends on the water and ice density ρ_{w} and ρ_{i} respectively, the gravity acceleration g and the latent heat of fusion L. The creep term incorporates the creep rate factor A , the creep exponent n and the effective normal pressure N. The effective normal pressure is related to the ice overburden pressure p_i and the water pressure p_{w} as $N = p_{\mathrm{i}} - p_{\mathrm{w}}$. Cavity opening is related to the basal ice velocity v_{b} and a parameter β that represents the bed undulation.

Boundary conditions are either a Dirichlet boundaries with a prescribed head that implies an effective pressure of zero or a Neumann boundary condition prescribing no outflow, that is $\nabla h = 0$. The effective transmissivity is at the transition to the ocean maximum transmissivity T_{max}, while the transition to land is represented by the minimum $T_{\mathrm{no\,flow}}$, which is set to $10^{-14}\,\mathrm{m}^2\,\mathrm{s}^{-1}$.

Appendix B: Numerics

The transient flow equation A1 is a two-dimensional diffusion equation with non-uniform and time-dependent hydraulic diffusivity $\alpha(x,y,t) = T_{\mathrm{e}}/S_{\mathrm{e}}$ and a time-dependent the source term $Q(x,y,t)$. The equation is descretized spatially on an equidistant rectangular grid using a second-order central difference scheme (e.g., Ferziger and Perić, 2002). All quantities are co-located on the grid. The implementation allows for first-order approximation (fully-implicit or fully-explicit) and second-order (Crank–Nicolson) approximation in time for the hydraulic head. Nevertheless, only the fully-implicit time stepping is used in this study to make the solution less dependent on the initial conditions for the hydraulic head. The transmissivity is updated using an explicit Euler step. In each time step a system of linear equations is solved for the hydraulic head using one of the PETSc solvers configured by the user. If an iterative solver is used, convergence is decided by the decrease of the residual norm relative to the norm of the right hand side (rtol) and the absolute size of the residual norm (atol).

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Geoscientific Model Development Discussions

References

Ahlstrøm, A., Mottram, R., Nielsen, C., Reeh, N., and Andersen, S.: Evaluation of the future hydropower potential at Paakitsoq, Ilulissat, West Greenland, Tech. Rep. 31, GEUS, https://doi.org/10.22008/gpub/27154, 2009.

Aschwanden, A., Fahnestock, M. A., and Truffer, M.: Complex Greenland outlet glacier flow captured, Nat. Commun., 7, 10 524, https://doi.org/10.1038/ncomms10524, 2016.

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H.: PETSc Users Manual, Tech. Rep. ANL-95/11 - Revision 3.15, Argonne National Laboratory, https://www.mcs.anl.gov/petsc, 2021.

Beyer, S., Kleiner, T., Aizinger, V., Rückamp, M., and Humbert, A.: A confined–unconfined aquifer model for subglacial hydrology and its application to the Northeast Greenland Ice Stream, The Cryosphere, 12, 3931–3947, https://doi.org/10.5194/tc-12-3931-2018, 2018.

Braithwaite, R. J. and Højmark Thomsen, H.: Simulation of Run-Off from the Greenland Ice Sheet for Planning Hydro-Electric Power, Ilulissat/Jakobshavn, West Greenland, Annals of Glaciology, 13, 12–15, https://doi.org/10.3189/S0260305500007540, 1989.

Bungartz, H.-J., Lindner, F., Gatzhammer, B., Mehl, M., Scheufele, K., Shukaev, A., and Uekermann, B.: preCICE - A fully parallel library for multi-physics surface coupling, Computers and Fluids, 141, 250–258, https://doi.org/10.1016/j.compfluid.2016.04.003, advances in Fluid-Structure Interaction, 2016.

Christmann, J., Helm, V., Khan, S. A., Kleiner, T., Müller, R., Morlighem, M., Neckel, N., Rückamp, M., Steinhage, D., Zeising, O., and Humbert, A.: Elastic deformation plays a non-negligible role in Greenland's outlet glacier flow, Communications Earth & Environment, 2, https://doi.org/10.1038/s43247-021-00296-3, 2021.

Colosio, P., Tedesco, M., Ranzi, R., and Fettweis, X.: Surface melting over the Greenland ice sheet derived from enhanced resolution passive microwave brightness temperatures (1979–2019), The Cryosphere, 15, 2623–2646, https://doi.org/10.5194/tc-15-2623-2021, 2021.

de Fleurian, B., Gagliardini, O., Zwinger, T., Durand, G., Le Meur, E., Mair, D., and Råback, P.: A double continuum hydrological model for glacier applications, The Cryosphere, 8, 137–153, https://doi.org/10.5194/tc-8-137-2014, 2014.

de Fleurian, B., Werder, M. A., Beyer, S., Brinkerhoff, D. J., Delaney, I., Dow, C. F., Downs, J., Gagliardinie, O., Hoffman, M. J., Hooke, R. L., and et al.: SHMIP The subglacial hydrology model intercomparison Project, Journal of Glaciology, 64, 897–916, https://doi.org/10.1017/jog.2018.78, 2018.

Ehlig, C. and Halepaska, J. C.: A numerical study of confined-unconfined aquifers including effects of delayed yield and leakage, Water Resources Research, 12, 1175–1183, https://doi.org/10.1029/WR012i006p01175, 1976.

Ferris, J. G., Knowles, D. B., Brown, R. H., and Stallman, R. W.: Theory of Aquifer Tests, Tech. rep., U.S. Government Print. Office, https://doi.org/10.3133/wsp1536E, 1962.

Ferziger, J. H. and Perić, M.: Computational Methods for Fluid Dynamics, Springer, 3rd edn., 2002.

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.

Hartnett, E. and Edwards, J.: THE PARALLELIO (PIO) C/FORTRAN LIBRARIES FOR SCALABLE HPC PERFORMANCE, in: 37th Conference on Environmental Information Processing Technologies, American Meteorological Society Annual Meeting,, 2021.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Jacob, C. E.: Determining the permeability of water-table aquifers, in: Methods of determining permeability, transmissibility and drawdown, edited by Bentall, R., no. 1536 in Geological survey water-supply paper, chap. I, pp. 272–292, US Government Printing Office, https://pubs.usgs.gov/wsp/1536i/report.pdf, 1963.

Joughin, I., Smith, B., Howat, I., and Scambos, T.: MEaSUREs Multi-year Greenland Ice Sheet Velocity Mosaic, Version 1. Boulder, Colorado USA. NASA National Snow and Ice Data Center Distributed Active Archive Center., https://doi.org/10.5067/QUA5Q9SVMSJG., 2016.

Joughin, I., Smith, B. E., and Howat, I. M.: A complete map of Greenland ice velocity derived from satellite data collected over 20 years, J. Glaciol., 64, 243, https://doi.org/10.1017/jog.2017.73, 2018.

Knüpfer, A., Rössel, C., Mey, D. a., Biersdorff, S., Diethelm, K., Eschweiler, D., Geimer, M., Gerndt, M., Lorenz, D., Malony, A., Nagel, W. E., Oleynik, Y., Philippen, P., Saviankou, P., Schmidl, D., Shende, S., Tschüter, R., Wagner, M., Wesarg, B., and Wolf, F.: Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope,Scalasca, TAU, and Vampir, in: Tools for High Performance Computing 2011, edited by Brunst, H., Müller, M. S., Nagel, W. E., and Resch, M. M., pp. 79–91, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

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, n/a–n/a, https://doi.org/10.1029/2011JF002140, 2012.

Latham, R., Zingale, M., Thakur, R., Gropp, W., Gallagher, B., Liao, W., Siegel, A., Ross, R., Choudhary, A., and Li, J.: Parallel netCDF: A High-Performance Scientific I/O Interface, in: SC Conference, p. 39, IEEE Computer Society, Los Alamitos, CA, USA, https://doi.org/10.1109/SC.2003.10053, 2003.

MacGregor, J. A., Chu, W., Colgan, W. T., Fahnestock, M. A., Felikson, D., Karlsson, N. B., Nowicki, S. M. J., and Studinger, M.: GBaTSv2: a revised synthesis of the likely basal thermal state of the Greenland Ice Sheet, The Cryosphere, 16, 3033–3049, https://doi.org/10.5194/tc-16-3033-2022, 2022.

Morlighem, M., Williams, C. N., Rignot, E., An, L., Arndt, J. E., Bamber, J. L., Catania, G., Chauché, N., Dowdeswell, J. A., Dorschel, B., Fenty, I., Hogan, K., Howat, I., Hubbard, A., Jakobsson, M., Jordan, T. M., Kjeldsen, K. K., Millan, R., Mayer, L., Mouginot, J., Noël, B. P. Y., O'Cofaigh, C., Palmer, S., Rysgaard, S., Seroussi, H., Siegert, M. J., Slabon, P., Straneo, F., van den Broeke, M. R., Weinrebe, W., Wood, M., and Zinglersen, K. B.: BedMachine v3: Complete Bed Topography and Ocean Bathymetry Mapping of Greenland From Multibeam Echo Sounding Combined With Mass Conservation, Geophysical Research Letters, 44, https://doi.org/10.1002/2017gl074954, 2017.

Morlighem, M. e. a.: IceBridge BedMachine Greenland, Version 4, https://doi.org/10.5067/VLJ5YXKCNGXO, 2021.

Neckel, N., Zeising, O., Steinhage, D., Helm, V., and Humbert, A.: Seasonal Observations at 79Â°N Glacier (Greenland) From Remote Sensing and in situ Measurements, Frontiers in Earth Science, 8, https://doi.org/10.3389/feart.2020.00142, 2020.

Oberkampf, W. L. and Roy, C. J.: Verification and Validation in Scientific Computing, Cambridge University Press, https://doi.org/10.1017/cbo9780511760396, 2010.

Rew, R. and Davis, G.: NetCDF: an interface for scientific data access, IEEE Computer Graphics and Applications, 10, 76–82, https://doi.org/10.1109/38.56302, 1990.

Schröder, L., Neckel, N., Zindler, R., and Humbert, A.: Perennial Supraglacial Lakes in Northeast Greenland Observed by Polarimetric SAR, Remote Sensing, 12, 2798, https://doi.org/10.3390/rs12172798, 2020.

https://doi.org/10.5194/gmd-2022-312
Preprint. Discussion started: 23 January 2023

Theis, C. V.: The relation between the lowering of the Piezometric surface and the rate and duration of discharge of a well using ground-water storage, Eos, Transactions American Geophysical Union, 16, 519–524, https://doi.org/10.1029/TR016i002p00519, 1935.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors: scikit-image: image processing in Python, PeerJ, p. e453, https://doi.org/10.7717/peerj.453, 2014.

Young, T. J., Christoffersen, P., Bougamont, M., and Stewart, C. L.: Rapid basal melting of the Greenland Ice Sheet from surface meltwater drainage, PNAS, 119, https://doi.org/10.1073/pnas.21160361, 2022.

Zeising, O. and Humbert, A.: Indication of high basal melting at the EastGRIP drill site on the Northeast Greenland Ice Stream, The Cryosphere, 15, 3119–3128, https://doi.org/10.5194/tc-15-3119-2021, 2021.