The next generation sea-ice model neXtSIM, version 2

Einar Ólason[1,2], Guillaume Boutin[1,2], Timothy Williams[1,2], Anton Korosov[1,2], Heather Regan[1,2], Jonathan Rheinlænder[1,2], Pierre Rampal[3], Daniela Flocco[4], Abdoulaye Samaké[5], Richard Davy[1,2], Timothy Spain[1,2], and Sean Chua[1,2]

[1]Nansen Environmental and Remote Sensing Center
[2]Bjerknes Centre for Climate Research, Bergen, Norway
[3]CNRS, Institut des Géosciences de l'Environnement, Grenoble, France
[4]Dipartimento di Scienze della Terra, dell'Ambiente e delle Risorse, Università degli Studi di Napoli "Federico II", Napoli, Italy
[5]Université des Sciences, des Techniques et des Technologies de Bamako, Bamako, Mali

Correspondence: Einar Ólason (einar.olason@nersc.no)

Abstract. Sea ice is a key component of the climate system, and sea-ice models are required to realistically simulate climate, ocean, or atmosphere in general circulation models at high latitudes. In this paper, we present the latest version of the next-generation sea-ice model, neXtSIM, which has been developed with a particular focus on ice dynamics and motion. While many large-scale sea ice models can represent ~~regional to global~~ regional-to-global sea ice evolution, their representation of sea ice dynamics varies little ~~between models. This is because~~ across models, as they all use ~~rheologies based on the hypothesis that sea icebehaves as a visco-plastic solid. This~~ a very similar representation of the ice's internal stresses. While this works reasonably well for several quantities (e.g. sea ice volume)~~but~~ , it fails to capture sea ice deformation features ~~at coarse and moderately high resolutions (i.e. coarser than about 5 km resolution). This may be problematic since these deformations result in the formation of leads and ridges~~and the resulting localised changes in thickness and concentration, which likely play an essential role in ice-atmosphere-ocean interactions~~, and because these are the resolutions at which sea ice models run in coupled models such as Earth System Models. An alternative is to use brittle rheologies that better capture these features independently of the resolution.~~. The neXtSIM model has been at the core of ~~the effort~~ efforts by its developers and users to explore ~~the usage of brittle rheologies and~~ new modelling approaches ~~in geophysical scale simulations of sea ice~~to address this shortcoming. Here, we document neXtSIM, now in version 2 of its development, to foster its use for the sea ice community and release a public version of the model. We describe the sea ice dynamics and the core of the model in detail~~and give insights into the~~, and discuss parameters specific to the brittle rheologies included in neXtSIM. We also document the model's specificity ~~associated with~~ arising from its Lagrangian framework and ~~how it affects the~~ its implications for coupling with other components of Earth system models. We hope that the insights provided in this study and the public release of the model will trigger innovative research in the sea ice modelling community.

1 Introduction

~~The~~ Sea ice is a key component of the climate system, and sea-ice dynamics play a significant role in shaping the sea-ice cover in both hemispheres; therefore, geophysical models must capture these characteristics. These dynamics are strongly influenced by the choice of sea ice rheology, and most models use a rheology based on the same framework as suggested by Hibler (1979). This rheology assumes that sea ice behaves as a visco-plastic (VP) solid and has demonstrated its ability to capture large-scale ice properties (such as area and volume) with reasonable accuracy over the years. However, it generally fails to capture localised deformations of the sea-ice cover, resulting in very homogeneous simulated ice properties, at least for resolutions coarser than about 5 km (e.g. Bouchat et al., 2022; Hutter et al., 2022). This is problematic because heterogeneity in sea ice, such as leads and ridges, is suspected to have a large impact on heat, light, and momentum fluxes exchanged between the ocean and the atmosphere (e.g. Assmy et al., 2017; Itkin et al., 2018; Lapere et al., 2024; Tsamados et al., 2014). This means that current sea ice models in GCMs are currently missing these interactions. A solution could be to modify the rheology, which has motivated the development of brittle rheologies in large-scale sea ice models. This has also been a primary motivation for the development of the next-generation sea-ice model, neXtSIM~~,~~ .

NeXtSIM has been developed at the Nansen Center in Bergen since 2012 at the instigation of ~~P.~~ Pierre Rampal and his group. It was conceived as an opportunity to create a new modelling tool for geophysical scale sea-ice research, offering some unique features, including i) a continuous and fully Lagrangian framework and ii) a new treatment of sea ice dynamics through a different class of rheology inspired by damage mechanics. The dynamical core of neXtSIM was introduced by Bouillon and Rampal (2015). This was based on the pioneering work by Girard et al. (2011), who introduced the first brittle rheology to be applied to sea ice (the elasto-brittle, or EB). Numerical modifications to the initial implementation of Girard et al. (2011) allowed Bouillon and Rampal (2015) to run neXtSIM for ten simulated days, while Girard et al. (2011) only ran the model for four days (including one-day spin-up)~~and assumed a quasi-static regime (i.e. left-end side term of equation 1 in section 3 equal to 0) in their initial work. Thanks to the EB rheology, the new model could reproduce the multifractal spatial scaling properties of sea-ice deformation. Using~~ . These included introducing a Lagrangian advection scheme, ~~the model could preserve the resulting discontinuous fields well.~~ whereby the nodes of the model's triangular grid move with the ice drift. Girard et al. (2011) did not include advection in their work.

Further development of neXtSIM was focused on turning the dynamical core of Bouillon and Rampal (2015) into a fully-fledged sea-ice model. Rampal et al. (2016a) presented a significant step towards this goal with the addition of thermodynamic melt and growth of the ice and dynamical remeshing of the model's triangular mesh. This allowed Rampal et al. (2016a) to run the model for an entire year, showing realistic ice thickness and ice extent evolution while reproducing the multifractal spatial scaling of sea-ice deformation, as before. Following these developments, the model was ported from the original Matlab/C hybrid code to a C++ ~~code base~~ codebase and parallelised using ~~the~~ a distributed-memory approach with the MPI message-passing library (Samaké et al., 2017).

The Maxwell elasto-brittle rheology ~~(MEB Dansereau et al., 2016)~~ (MEB) (Dansereau et al., 2016) was implemented into neXtSIM following its publication, and Rampal et al. (2019) used this latest version of the model to compare for the first time the simulated temporal, as well as the spatial scaling of sea-ice deformation with the same type of scaling relationships found in the observations ~~Marsan et al. (2004); Rampal et al. (2008)~~(Marsan et al., 2004; Rampal et al., 2008). The brittle Bingham-

Maxwell rheology was developed and implemented in neXtSIM, ~~allowing~~ enabling Ólason et al. (2022) to run the model over several years, showing realistic evolution of ice thickness and ice extent ~~evolution~~ while reproducing the multifractal spatial scaling of sea-ice deformation and realistic pan-Arctic deformation patterns. Boutin et al. (2023) extended this work, using neXtSIM coupled to the ocean model of NEMO to explore the mass balance of Arctic sea ice over several decades using BBM. These latest developments have allowed the neXtSIM development team to reach their goal of turning the model into a fully-fledged sea-ice model, albeit with some caveats.

This paper aims to provide an overview of the features of neXtSIM~~that have already been documented and to document those that have remained undocumented~~. It will focus on new features, while still giving a relevant description of those parts of the model already documented elsewhere. The model description is, by its nature, technical, but we will strive to highlight the physical reasoning behind the modelling choices we have made. A core tenet of the neXtSIM development process is to use the simplest modelling approach ~~, which can reproduce observations, of which~~ that reproduces observations. In this, our main focus has been on satellite remote sensing observations~~are most important~~.

2 General model description

In neXtSIM, the model's physical equations are solved in two separate sections: one for the dynamics and one for the thermodynamics. In addition, the model must handle Lagrangian grid advection and perform infrastructural tasks, including initialisation, writing diagnostic outputs, and writing restart files. The model can also be coupled to an ocean model through the OASIS coupler, so OASIS-related operations must also be handled if requested. A typical model workflow is shown in 1, and the remainder of this section outlines the major steps in that workflow, with details provided in later sections of the paper as needed.

2.1 Configuration files

NeXtSIM uses INI-style config files, which are read in using the `boost program_options` library. The neXtSIM config file is sorted into 18 sections: `simul`, `debugging`, `numerics`, `setup`, `mesh`, `moorings`, `drifters`, `restart`, `output`, `ideal_simul`, `dynamics`, `thermo`, `nesting`, `forecast`, `coupler`, `wave_coupling`, `statevector`, and `age`. Importantly, `simul` controls basic parameters of each simulation, such as start date and time step. The `setup` block sets the initial and boundary conditions, and `dynamics` and `thermo` control parameters and options for the model's dynamics and thermodynamics, respectively.

2.2 Initial conditions and restart files

Initial conditions are set depending on whether the user requests restarting from a previous model run or starting from a given set of initial conditions. The restart files are a set of custom binary files that describe both the model mesh and all required fields. Restart files are not guaranteed to work correctly if moved from one computer to another, although in practice, they will as long as the system's CPUs use the same endianness.

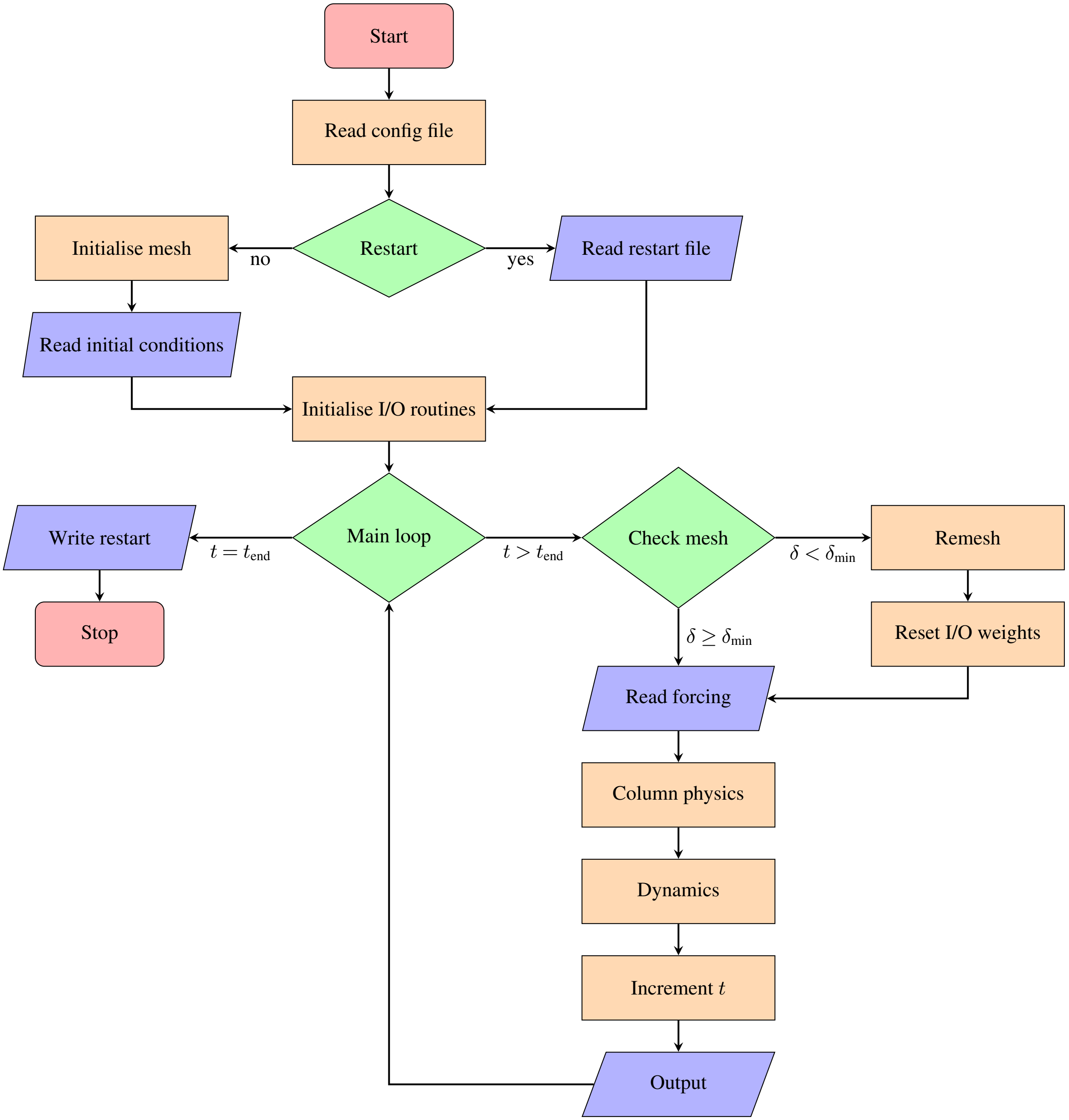

Figure 1. Program flow during an integration of neXtSIM. Purple parallelograms indicate I/O tasks, green triangles indicate boolean checks, and orange boxes indicate other tasks. Here, t indicates the model time and t_{end} the time at which the integration should stop. The smallest angle of any given triangle in the mesh is given by δ, and the smallest allowed such angle by δ_{min}.

NeXtSIM supports a substantial number of different initial conditions when not starting from a restart file, through the `setup.ice-type` option. This includes reading in sea-ice thickness and concentration from the TOPAZ4, GLORYS, and PIOMAS reanalysis, as well as fields from AMSR2, AMSR2, IceSat, and CryoSat2 in various combinations.

2.3 Main loop

The main program loop consists of checking and updating the computational mesh, running the thermodynamics and dynamics routines, reading forcing fields, and outputting diagnostic and restart outputs. Mesh handling, thermodynamics, and dynamics are discussed in more detail in sections 5, 4, and 3. Here, it is important to note that remeshing impacts both reading of forcing files and writing to regularly spaced netCDF files.

While neXtSIM uses a Lagrangian moving mesh, the model assumes that the mesh remains static between remeshings for most input and output operations. In practice, this means that the functions that interpolate the input and output data between the mesh and the source or destination grids use interpolation weights that are recalculated only when the mesh is updated (see 5.1). This also holds for routines interfacing with the OASIS coupler. In addition to gridded outputs (5.4), neXtSIM also provides outputs which are snapshots of the computational grid and preserve the instantaneous positions of the grid nodes. These files are written in the same customised format as the restart files. Finally, restart files are written at the end of the simulation or at given intervals, as requested by the user.

3 Dynamics

The core equation of sea-ice dynamics is the momentum equation. This is Newton's second law, but implementations may vary depending on the level of detail considered (e.g. Hibler, 1979; Connolley et al., 2004; Bouillon and Rampal, 2015; Danilov et al., 2015). The form used in neXtSIM is

$$m\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{\nabla}\cdot(\boldsymbol{\sigma}h) + A(\boldsymbol{\tau}_a + \boldsymbol{\tau}_w) + \boldsymbol{\tau}_b + mf\boldsymbol{k}\times\boldsymbol{u} - mg\boldsymbol{\nabla}\cancel{\eta}H_o, \tag{1}$$

where A is the fraction of the grid cell covered by ice, $m = A\rho h$ is the ice mass per unit area, $\boldsymbol{u}$ is the ice velocity, $\boldsymbol{\sigma}$ is the internal stress tensor, h is the ice slab thickness (not volume per unit area), ρ the ice density, $\boldsymbol{\tau}_a$ and $\boldsymbol{\tau}_w$ are the atmosphere and ocean stress terms, respectively, $\boldsymbol{\tau}_b = -C_b\boldsymbol{u}$ is the basal stress term introduced in Lemieux et al. (2015), $mf\boldsymbol{k}\times\boldsymbol{u}$ is the Coriolis term, with vertical unit vector $\boldsymbol{k}$, and $\cancel{mg\boldsymbol{\nabla}\eta}$ $mg\boldsymbol{\nabla}H_o$ is the ocean-tilt term with g being the acceleration due to gravity and H_o the sea-surface height. We write ~~explicitly~~ the integrated internal stress explicitly as $\boldsymbol{\sigma}h$, following Sulsky et al. (2007) and Bouillon and Rampal (2015). ~~The rheology~~ Rheology determines the internal stress, $\boldsymbol{\sigma}$, which links stress and strain, or strain rates, in the ice. The dynamical core of neXtSIM is built around a finite element discretisation of the momentum equation and a time-splitting approach for time-stepping the model equations.

One primary purpose of neXtSIM has been to explore the performance and effects of using the brittle rheologies (Girard et al., 2011; Dansereau et al., 2016; Ólason et al., 2022) in a large-scale context. In version 2 of neXtSIM, only the BBM rheology is included, out of the brittle rheologies. The EB and MEB rheologies can be recovered using ~~a suitable set of parameters for~~

~~BBM~~ an appropriate set of BBM parameters. Still, the interest in using EB and MEB for large-scale applications is, arguably, limited.

In version 2 of neXtSIM, we also included an implementation of the ~~mEVP rheology (Bouillon et al., 2013). This was initially an EVP implementation as~~ elasto-viscous-plastic (EVP) Hunke and Dukowicz (1997) and modified elasto-viscous-plastic (mEVP) (Lemieux et al., 2012; Bouillon et al., 2013) rheologies. This EVP implementation was initially a technical exercise to aid the BBM development. This was then modified to an mEVP to ~~act~~ serve as a benchmark ~~to compare BBM against~~ for comparing with BBM (Ólason et al., 2022).

3.1 ~~BBM implementation~~Spatial discretisation

The ~~BBM rheology is~~ finite element discretisation is based on a triangular grid, with ice velocities defined at the vertices and tracers treated as constant within the elements (consider e.g. Rao, 2007; Danilov et al., 2015; Bouillon and Rampal, 2015; Bielak, 2024, fo . To solve the momentum equation in this framework, we define a set of test functions to obtain the weak formulation of the equation. Using a continuous Galerkin discretisation, we select the test functions, N_j^e, as linear functions associated with vertex j of element e of the triangular mesh. This function equals one at the vertex j and zero at all other vertices. The ice velocity field in element e is thus

$$\mathbf{u} = \sum_{e(j)} \mathbf{u}_j N_j^e, \tag{2}$$

where the sum is over all the vertices of e.

The gradient of the velocity field is related to the gradient of the test functions, referred to here as the shape coefficients. These are calculated by defining the test functions on the three nodes of a reference element in the coordinates $\{\xi, \upsilon\}$, as

$$N_1(\xi,\upsilon) = 1 - \xi - \upsilon \tag{3}$$

$$N_2(\xi,\upsilon) = \xi \tag{4}$$

$$N_3(\xi,\upsilon) = \upsilon, \tag{5}$$

and then projecting this onto e using the transformation

$$x(\xi,\upsilon) = N_1^e x_1^e + N_2^e x_2^e + N_3^e x_3^e$$

$$y(\xi,\upsilon) = N_1^e y_1^e + N_2^e y_2^e + N_3^e y_3^e, \tag{6}$$

where the coordinates x_i and y_i are those of the three vertices of e. The shape coefficients are then computed as

$$\begin{bmatrix} \frac{\partial N_j^e}{\partial x} \\ \frac{\partial N_j^e}{\partial y} \end{bmatrix} = \frac{1}{\det(\mathbf{J})} \begin{bmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_2 - x_3 & x_3 - x_1 & x_1 - x_3 \end{bmatrix}, \tag{7}$$

where $\det(\mathbf{J})$ is the determinant of the Jacobian of the transformation (6), computed as

$$\det(\mathbf{J}) = x_2y_3 + x_3y_1 + x_1y_2 - x_2y_1 - x_3y_2 - x_1y_3, \tag{8}$$

and is equal to twice the element area. The coordinates x_i and y_i are taken to start at j and go around e in a clockwise fashion.

The strain-rate equations can now be calculated using the shape coefficients. The strain-rate equations are

$$\dot{\varepsilon}_{11} = \frac{\partial u}{\partial x} \tag{9}$$

$$\dot{\varepsilon}_{22} = \frac{\partial v}{\partial y} \tag{10}$$

$$\dot{\varepsilon}_{12} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \tag{11}$$

where $\dot{\varepsilon}_{ij}$ are the components of the strain-rate tensor $\dot{\boldsymbol{\varepsilon}}$, the dot symbolising the time derivative, and u and v are the zonal and meridional components of $\mathbf{u}$. The strain-rate equations are projected onto the test functions to give element-wise strain rates:

$$(\dot{\varepsilon}_{11})_e = \sum_{j(e)}\left(u_j \frac{\partial N_j^e}{\partial x}\right) \tag{12}$$

$$(\dot{\varepsilon}_{22})_e = \sum_{j(e)}\left(v_j \frac{\partial N_j^e}{\partial y}\right) \tag{13}$$

$$(\dot{\varepsilon}_{12})_e = \sum_{j(e)}\left(u_j \frac{\partial N_j^e}{\partial y} + v_j \frac{\partial N_j^e}{\partial x}\right). \tag{14}$$

In neXtSIM, we solve the equations on an $\{x,y\}$ plane (usually using a polar stereographic projection), so we do not need to account for metric terms that arise due to the curvature of the earth.

To solve the momentum equation, we follow Danilov et al. (2015) and use nodal quadrature in all terms that do not involve spatial derivatives. This allows us to solve the BBM, EVP, and mEVP models efficiently using explicit time stepping. Multiplying the momentum equation by the test functions, integrating over the domain, and integrating the rheology term by parts gives the following at vertex j:

$$\int mN_j\left(\frac{\partial \mathbf{u}}{\partial t}\right)dS = \int N_j h\mathbf{n}\boldsymbol{\sigma}d\Gamma - \int(\nabla N_j)h\boldsymbol{\sigma}dS + \int N_j[A(\boldsymbol{\tau}_a + \boldsymbol{\tau}_w) + \boldsymbol{\tau}_b + f\mathbf{k}\times\mathbf{u}]dS - \int mg\nabla H_o N_j dS, \tag{15}$$

where S is the domain, Γ the boundary of the domain and $\mathbf{n}$ the normal vector to that boundary.

The numerical integration of equation (15) proceeds as follows: The left-hand side can be written as

$$\int mN_j\left(\frac{\partial \mathbf{u}}{\partial t}\right)dS = \sum_k M_{jk} m_k \left(\frac{\partial \mathbf{u}_k}{\partial t}\right), \tag{16}$$

where the summation is taken over all vertices, k. The lump mass matrix, M is diagonal and its non-zero entries are

$$M_{ii} = \frac{1}{3}\sum_{e(i)} A_e, \tag{17}$$

where the sum is taken over all elements containing the vertex i. The nodal mass m_j is calculated as the area-weighted mean of all the neighbouring element thickness values. The left-hand side then becomes simply

$$\int mN_j\left(\frac{\partial \mathbf{u}}{\partial t}\right)dS = m_j\left(\frac{\partial \mathbf{u}_j}{\partial t}\right)M_{jj} \tag{18}$$

Similarly, the third term on the right-hand side becomes

$$\int N_j[A(\boldsymbol{\tau}_a+\boldsymbol{\tau}_w)+\boldsymbol{\tau}_b+f\mathbf{k}\times\mathbf{u}]dS = [A(\boldsymbol{\tau}_a+\boldsymbol{\tau}_w)_j+(\boldsymbol{\tau}_b)_j+f\mathbf{k}\times\mathbf{u}_j]M_{jj}. \tag{19}$$

The first term on the right-hand side is the boundary contribution, which we set to zero under the assumption of a no-slip boundary condition.

The second and fourth terms on the right-hand side—the internal stress term and sea-surface tilt term, respectively—are more involved than the terms already discussed and require summation over the elements $e(j)$ surrounding j. The x-component of the internal stress term becomes

$$\left[-\int(\nabla N_j)h\boldsymbol{\sigma}dS\right]_x = -\frac{1}{3}\sum_{e(j)} A_e h_e\left((\sigma_{11})_e\frac{\partial N_j}{\partial x}+(\sigma_{12})_e\frac{\partial N_j}{\partial y}\right) \tag{20}$$

and the y-component becomes

$$\left[-\int(\nabla N_j)h\boldsymbol{\sigma}dS\right]_y = -\frac{1}{3}\sum_{e(j)} A_e h_e\left((\sigma_{12})_e\frac{\partial N_j}{\partial x}+(\sigma_{11})_e\frac{\partial N_j}{\partial y}\right), \tag{21}$$

with σ_{ij} the components of the stress tensor $\boldsymbol{\sigma}$. The sea-surface tilt term requires a double summation, where the x-component becomes

$$\left(-\int mg\nabla H_o N_j dS\right)_x = gm_j\frac{1}{3}\sum_{e(j)}\left(A_e\sum_{k(e)} H_{o,k}\frac{\partial N_k}{\partial x}\right) \tag{22}$$

and the y-component

$$\left(-\int mg\nabla H_o N_j dS\right)_y = gm_j\frac{1}{3}\sum_{e(j)}\left(A_e\sum_{k(e)} H_{o,k}\frac{\partial N_k}{\partial y}\right). \tag{23}$$

3.2 Temporal discretisation

The time-splitting method we use in neXtSIM is based on that proposed by Hunke and Dukowicz (1997) for the EVP model, where they solve the column physics and advection equations on the main model time step Δt, but apply a sub-time stepping to solve the momentum equation at a shorter time step, Δt_m. In neXtSIM, we use this time-stepping scheme for the BBM (as in Ólason et al., 2022) and the EVP model, whereas the mEVP requires a small modification to the time-stepping and internal stress calculation (see section 3.4).

The time stepping of the momentum equation follows the semi-implicit solution derived by Hunke and Dukowicz (1997). Following the discretisation discussion above, this then takes the form

$$(\alpha^2+\beta^2)u_j^{n+1} = \alpha u_j + \beta v_j + \frac{\Delta t_m}{m_j M_{jj}}\left[\alpha\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_x + \tau_x\right) + \beta\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_y + \tau_y\right)\right] \tag{24}$$

and (correcting for a sign error in Hunke and Dukowicz (1997) on the last β term)

$$(\alpha^2+\beta^2)u_j^{n+1} = \alpha u_j - \beta v_j + \frac{\Delta t_m}{m_j M_{jj}}\left[\alpha\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_y + \tau_y\right) - \beta\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_x + \tau_x\right)\right] \tag{25}$$

where all values on the right-hand side are taken at time n to produce u and v at time $n+1$ on the left-hand side. The integrals are calculated following equations (20) and (21). Additionally we have

$$c' = \rho_w C_w |\boldsymbol{u}_w - \boldsymbol{u}| \tag{26}$$

$$\alpha = 1 + \frac{\Delta t_m}{m_j}(c'\cos\theta_w + C_b) \tag{27}$$

$$\beta = \Delta t_m\left(f + \frac{c'\sin\theta_w}{m_j}\right) \tag{28}$$

$$\tag{29}$$

and

$$\tau_x = \tau_{a,x} + c'(u_w\cos\theta_w - v_w\sin\theta_w) - \left(\int mg\nabla H_o N_j dS\right)_x \tag{30}$$

$$\tau_y = \tau_{a,y} + c'(v_w\cos\theta_w - u_w\sin\theta_w) - \left(\int mg\nabla H_o N_j dS\right)_y, \tag{31}$$

where the integrals are calculated following equations (20) to (23). Here, ρ_w is the ocean water density, C_w the ice-ocean drag coefficient, $\mathbf{u}_w$ the ocean velocity (and u_w and v_w its components), θ_w the ocean turning angle, and $\tau_{a,x}$ and $\tau_{a,y}$ are the x and y components of the ice-ocean stress

$$\boldsymbol{\tau} = \rho_a C_m |\mathbf{u}|\mathbf{u}, \tag{32}$$

with ρ_a the atmospheric density and C_m the atmospheric drag coefficient for momentum exchange (e.g. Leppäranta, 2011).

3.3 BBM implementation

The Brittle Bingham-Maxwell rheology (BBM) is described in detail in Ólason et al. (2022)~~, but we will discuss it briefly here~~; here we briefly discuss it for completeness, focusing on ~~any neXtSIM-related specifics.~~

aspects relevant to neXtSIM. The BBM rheology is a damaging Bingham-Maxwell constitutive model. Using damage mechanics for sea-ice modelling was introduced with the ~~EB~~ elasto-brittle (EB) model by Girard et al. (2011), but damage mechanics are widely used in other communities, e.g. in rock and crustal mechanics (e.g. Lyakhovsky et al., 1997; Amitrano et al., 1999; Schapery, 1999)).

The EB model simulates the ice as a damaging elastic sheet, where each grid element can be considered a spring in a mechanics sense. In ~~it~~the EB model, the stress is calculated in every grid cell, and if the stress is outside the Mohr-Coulomb yield criterion, the elasticity is reduced in that grid cell, and the sea ice experiences a reversible deformation. ~~Elasticity~~ The Mohr-Coulomb criterion is

$$\tau = \sigma_N \mu + C, \tag{33}$$

where μ is a constant friction coefficient, C, is the material cohesion, and τ and σ_N are the shear and normal stresses:

$$\tau = \frac{1}{2}\sqrt{(\sigma_{11}-\sigma_{22})^2 + 4\tau_{12}^2} \tag{34}$$

$$\sigma_N = \frac{1}{2}(\sigma_{11}+\sigma_{22}). \tag{35}$$

Elasticity, E, in the model is thus

$$E = Y(1-d), \tag{36}$$

where Y is the ice's Young modulus, and ~~$d \in [0,1[$~~ d with $0 \leq d < 1$, is a scalar damage variable.

At the start of the simulation, $d = 0$ everywhere, but d is then ~~reduced~~ increased to ensure that all stresses in the ice are always within the yield criterion. A local increase in d represents damaging of the ice, which is modelled as a reduction in elasticity. The reduction in elasticity in an element means that this element deforms more easily than before, and so the distribution of stresses in the neighbouring elements must change. This causes a stress redistribution and a cascade of damage ~~reduction~~ increases, emulating the multiplicative cascade that Marsan et al. (2004) suggested was the reason for the ~~observed~~ spatial scaling of sea-ice deformation they observed in satellite remote sensing data.

The MEB model (Dansereau et al., 2016) introduced a viscous element, or a dashpot, in series with the elastic one. This is intended to simulate the viscous dissipation of internal stresses and the larger~~ – irreversible –~~, irreversible deformations (as opposed to the elastic counterpart) ~~occurring~~ that occur along faults once the material is highly damaged. The viscosity of the dashpot is very high when the ice is undamaged, but should decrease faster than the spring's elasticity when damage increases.

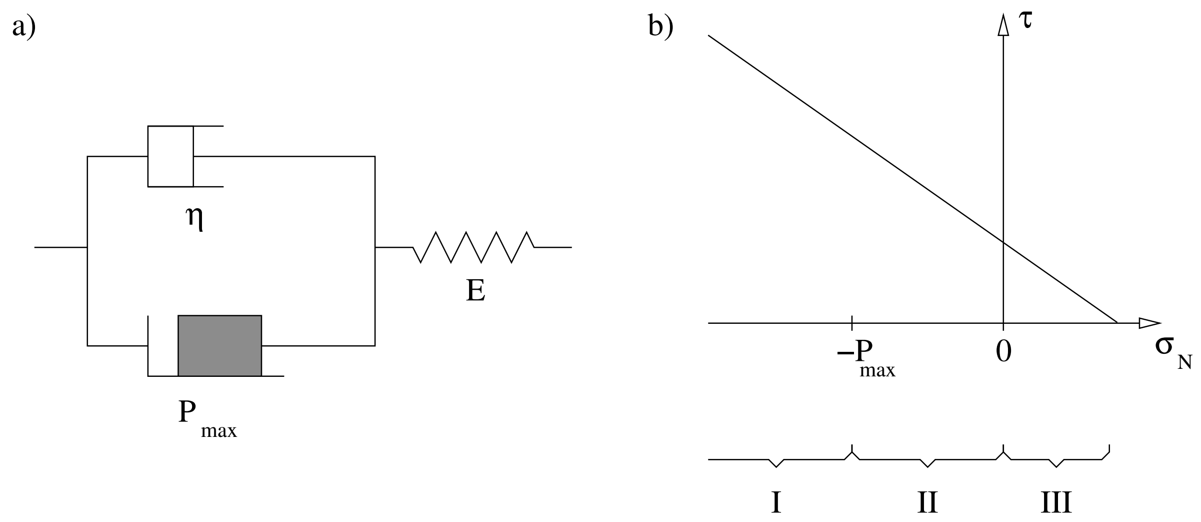

Figure 2. Panel a) A schematic of the Bingham-Maxwell constitutive model showing a dashpot and a friction element connected in parallel, with both connected to a spring in series. Panel b) The yield criterion in the stress invariant plane $\{\sigma_N, \tau\}$, as well as the elastic limit $P_{\max}$, and the ridging (I), elastic (II), and diverging (III) regimes.

This is formulated as:

$$\eta = \eta_0(1-d)^\alpha, \tag{37}$$

where η is the dashpot's viscosity, α is a material-dependent constant~~.~~, and η_0 is the viscosity of undamaged ice. This should tend towards infinity, but is set to 10^7 s^{-1} for practical purposes.

The Bingham-Maxwell model consists of a friction block and a dashpot in parallel, connected with a spring in serial. A schematic of the constitutive model is shown in panel a of figure 2. The spring's elasticity and the dashpot's viscosity evolve as a function of the damage variable, as in the MEB rheology. The friction element is introduced to emulate the resistance to ridging~~so that~~, so that, for small compressive stresses, the ice remains fully elastic ~~,~~ regardless of the level of damage. This leads to the model having three regimes in stress space: a visco-elastic converging one for high compressive normal stress, a purely elastic one for compressive normal stress less than the threshold $P_{\max}$, and a visco-elastic diverging one for divergent normal stress. These three regimes are shown schematically in panel b of figure 2.

The constitutive equation for BBM is

$$\cancel{\sigma}\dot{\boldsymbol{\sigma}} = E\mathbf{K} : \cancel{\varepsilon}\dot{\boldsymbol{\varepsilon}} - \cancel{\frac{\sigma}{\lambda}}\frac{\boldsymbol{\sigma}}{\lambda}\left(1 + \widetilde{P} + \frac{\lambda \dot{d}}{1-d}\right), \tag{38}$$

where ~~σ and $\dot{\varepsilon}$~~ $\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\varepsilon}}$ are the stress and strain rate tensors, respectively, $\lambda = \eta/E = \lambda_0(1-d)^{\alpha-1}$ is the viscous relaxation time, with ~~λ_0~~ $\lambda_0 = \eta_0/Y$ the undamaged viscous relaxation time. ~~The time derivative is noted with a dot~~We assume plane stress conditions, so the stiffness tensor operation $\mathbf{K}:\dot{\boldsymbol{\varepsilon}}$ is

$$\begin{pmatrix}(\mathbf{K}:\dot{\varepsilon})_{11}\\(\mathbf{K}:\dot{\varepsilon})_{22}\\(\mathbf{K}:\dot{\varepsilon})_{12}\end{pmatrix} = \frac{1}{1-\nu^2}\begin{pmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & 1-\nu\end{pmatrix}\begin{pmatrix}\dot{\varepsilon}_{11}\\ \dot{\varepsilon}_{22}\\ \dot{\varepsilon}_{12}\end{pmatrix} \tag{39}$$

where ν is Poisson's ratio. The resistance to ridging is encoded in $\widetilde{P}$ as

$$\widetilde{P} = \begin{cases}\frac{P_{\max}}{\sigma_N} & \text{for } \sigma_N < -P_{\max}\\ -1 & \text{for } -P_{\max} < \sigma_N < 0\\ 0 & \text{for } \sigma_N > 0\end{cases} \tag{40}$$

where

$$P_{\max} = P\left(\frac{h}{h_0}\right)^f e^{-C(1-A)}, \tag{41}$$

with P a constant ice strength parameter, $h_0 = 1$ m, A is ice concentration and C is a constant regulating the decrease of $P_{\max}$ as the concentration decreases. The factor $f \in [1,2]$ was set to $f = 3/2$ by Ólason et al. (2022), who also briefly explored the impact of varying f. The ~~MEB model may be recovered by setting $\widetilde{P} = 0$, while the EB model is recovered when $\widetilde{P} = -1$~~three cases for $\widetilde{P}$ correspond to the cases I, II, and III in panel b of figure 2, corresponding to visco-elastic (ridging), pure elastic, and diverging regimes. Equation (41) for $P_{\max}$ determines the compressive stress required to move from the elastic to the visco-elastic regime, i.e. when the ice can ridge.

Equation (38) is solved together with the momentum equation using equations (24) and (25). The time stepping of ~~equation~~ (38) is done ~~via a time-splitting method, where multiple dynamic time stepsare taken (of order 100) within a model time step. The time stepping is done by first calculating~~ in two steps. First, we calculate an intermediate stress, $\boldsymbol{\sigma}'$ through an Euler forwards iteration of (38), assuming $\dot{d} = 0$, wich gives

$$\cancel{\sigma'}\frac{\boldsymbol{\sigma}'-\boldsymbol{\sigma}^n}{\Delta t} = \cancel{\frac{\lambda(\Delta t E\mathbf{K}:\dot{\varepsilon}+\sigma^n)}{\lambda+\Delta t(1+\widetilde{P})}}.\,E\mathbf{K}:\dot{\boldsymbol{\varepsilon}} - \frac{\boldsymbol{\sigma}'}{\lambda}\left(1+\widetilde{P}\right), \tag{42}$$

~~If this stress~~ or, after rearranging

$$\boldsymbol{\sigma}' = \frac{\lambda(\Delta t_m E\mathbf{K}:\dot{\boldsymbol{\varepsilon}}+\boldsymbol{\sigma}^n)}{\lambda+\Delta t_m(1+\widetilde{P})}, \tag{43}$$

where $\boldsymbol{\sigma}^n$ is the stress at the start of the time step.

If $\boldsymbol{\sigma}'$ is outside the yield criterion, an updated damage value is calculated as

$$d^{n+1} = d^n + (1-d_{\text{crit}})(1-d^n), \tag{44}$$

~~where~~ and a consistent stress correction is applied

$$\boldsymbol{\sigma}^{n+1} = \boldsymbol{\sigma}' d_{\text{crit}}. \tag{45}$$

Here, d_{crit} is the factor needed to relax the stresses back onto the yield criterion, which is calculated as

$$d_{\text{crit}} = \begin{cases} -N/\sigma_N & \text{if } \sigma_N < -N, \\ c/(\tau + \mu\sigma_N) & \text{if } \tau > c - \mu\sigma_N, \end{cases} \tag{46}$$

~~and~~ where ~~σ_N is the mean normal stress , τ is the maximum shear stress, and~~ c and μ are the cohesion and internal friction coefficient values for the Mohr-Coulomb criterion, respectively. A capping with a large value of N is needed to prevent numerical instabilities (see Plante et al., 2020)~~. Damage (and stress) are only updated if the stress is outside the yield criterion, i.e. $0 < d_{\text{crit}} < 1$.~~

~~Following the damage update, a consistent stress correction is applied~~

$$\sigma^{n+1} = \sigma' - (1 - d_{\text{crit}})\sigma'.$$

~~Note that equations (44) and (45) differ from those of Ólason et al. (2022) by a factor of $\Delta t/t_d$ in the second term on the right, where Δt is the dynamic time step and t_d is a damage propagation time scale. Doing so, we follow the damage and stress relaxation~~ , where N is chosen sufficiently large to avoid affecting the solution. In neXtSIM, the default value is four orders of magnitude larger than the cohesion. Equation (46) follows the damaging scheme of Bouillon and Rampal (2015) rather than that of Dansereau et al. (2016), as Ólason et al. (2022) did. The consequences of ~~this~~ which scheme is chosen are discussed in section 6.1. If $\boldsymbol{\sigma}'$ is inside the yield criterion, i.e. $d_{\text{crit}} < 0$ or $d_{\text{crit}} > 1$, damage remains unchanged and $\boldsymbol{\sigma}^{n+1}$ is set to $\boldsymbol{\sigma}'$.

~~The strain rates needed for the stress calculation are spatial derivatives of the velocity, calculated using a spatial discretisation of the equations. The spatial discretisation of neXtSIM is unusual among large-scale sea-ice models and is particularly advantageous for the BBM implementation. NeXtSIM uses a~~ We note that the finite-element discretisation ~~, where tracers are constant on the elements (P0), while the velocities are located at the nodes of the mesh triangles (P1). This configuration means the elements' strain rate and stress tensors are constant, with all their componentsco-located~~used in neXtSIM results in the co-location of all the stress tensor components. This co-location is advantageous for the BBM implementation, because all stress tensor components are needed to calculate σ_N and τ of equation (46). Models using a ~~staggering~~ staggered grid, where different components of the stress tensor lie at different locations ~~in the grid~~ (e.g. the B- or C-grid) must apply extra effort to achieve the co-location of the stress components needed to get an accurate estimate of d_{crit} (e.g. Plante et al., 2020; Brodeau et al., 2024).

3.4 EVP and mEVP implementation

The ~~mEVP rheology (Lemieux et al., 2012; Bouillon et al., 2013) is a~~ elasto-viscous-plastic (EVP Hunke and Dukowicz, 1997) and the modified elasto-viscous-plastic (mEVP Lemieux et al., 2012; Bouillon et al., 2013) rheologies are numerically efficient and easily parallelisable ~~approach~~ approaches to solving the ~~original VP equations (Hibler, 1979)~~equations of the original visco-plastic rheology (VP Hibler, 1979). In the VP model, the internal stress tensor is diagnosed from the strain rates as

$$\sigma_N = \zeta\dot{\varepsilon}_{\mathrm{I}} - P/2 \tag{47}$$

$$\tau = \eta\dot{\varepsilon}_{\mathrm{II}} \tag{48}$$

where ~~$\dot{\varepsilon}_{\mathrm{I}}$ and $\dot{\varepsilon}_{\mathrm{II}}$~~

$$\dot{\varepsilon}_{\mathrm{I}} = \frac{1}{2}(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22}) \tag{49}$$

$$\dot{\varepsilon}_{\mathrm{II}} = \frac{1}{2}\sqrt{(\dot{\varepsilon}_{11} - \dot{\varepsilon}_{22})^2 + 4\dot{\varepsilon}_{12}^2} \tag{50}$$

are the first and second strain rate invariants ~~, and~~ and

$$\zeta = \cancel{\frac{P}{2\Delta}}\frac{P}{2\Delta + \Delta_{\mathrm{min}}} \tag{51}$$

$$\eta = \zeta/e^2 \tag{52}$$

$$\Delta = \sqrt{\dot{\varepsilon}_{\mathrm{I}}^2 + \dot{\varepsilon}_{\mathrm{II}}^2/e^2} \tag{53}$$

$$P = P^* h e^{-C(1-A)} \tag{54}$$

with e and P^* constants controlling the ellipse eccentricity and the ice strength, and Δ_{min} a constant regulating the maximum viscosity in the model.

The VP equations must be solved implicitly ~~with an iteration scheme~~using an iterative scheme, such as a fixed-point iteration or a Picard solver. To avoid using an implicit solver, Hunke and Dukowicz (1997) introduced an elastic term, which allows the equations to be solved explicitly ~~, using~~ through sub-iterations. The VP rheology is recovered as the steady state limit of the EVP rheology of Hunke and Dukowicz (1997) or in the limit of an infinitely large elasticity. For the EVP, the momentum equation is solved using ~~an~~ the explicit time-stepping scheme ~~identical to that used for the BBM~~detailed above in equations (24) and (25).

~~The mEVP, in addition to modifying the constituent model, also modifies the momentum equation.~~ In neXtSIM, we follow Bouillon et al. (2013) in solving the EVP constituent equations by defining $\sigma_1 = \sigma_{11} + \sigma_{22}$ and $\sigma_2 = \sigma_{11} - \sigma_{22}$ and solving for those as

$$\sigma_1^{n+1} = \sigma_1^n + (2\zeta[\varepsilon_{11} + \varepsilon_{22} - \Delta] - \sigma_1^n)/\alpha_1 \tag{55}$$

$$\sigma_2^{n+1} = \sigma_2^n + (2\zeta[\varepsilon_{11} - \varepsilon_{22}]/e^2 - \sigma_2^n)/\alpha_2 \tag{56}$$

$$\sigma_{12}^{n+1} = \sigma_{12}^n + (2\zeta\varepsilon_{12}/e^2 - \sigma_{12}^n)/\alpha_2, \tag{57}$$

where $\alpha_1 = 2T/\Delta t_m$ and $\alpha_2 = \alpha_1/e^2$, with $T = \Delta t/3$ as the elastic time scale.

From a technical standpoint, the mEVP consists of slight modifications of the time stepping of both the momentum and constituent equations of EVP. The constituent equations of mEVP are the same as for EVP (namely equations 55, 56, and 57), except we set $\alpha_1 = \alpha_2 = \alpha'$, where α' is a constant tuning parameter introduced (as just α) by Bouillon et al. (2013). In neXtSIM, we take advantage of this similarity to reduce code duplication.

We also exploit similarities between the EVP/BBM momentum solver and the mEVP ~~one~~solver to reduce code duplication. For mEVP, ~~we can rewrite the modified momentum equation of mEVP (equations (39) and (40) of Lemieux et al., 2012) can be written as~~

$$\beta'(\mathbf{u}^{n+1} - \mathbf{u}^n) = \mathbf{u}^0 - \mathbf{u}^{n+1} - \Delta t \mathbf{f} \times \mathbf{u}^{n+1}$$

$$+ \frac{\Delta t}{m}\left[\mathbf{F} + AC_d\rho_w(\mathbf{u}_w - \mathbf{u}^{n+1})|\mathbf{u}_w - \mathbf{u}^{n+1}| - mg\nabla\eta\right],$$

the momentum equation time stepping takes the form

$$\beta' \frac{m}{\Delta t_m}(\mathbf{u}^{n+1} - \mathbf{u}^n) = \frac{m}{\Delta t_m}(\mathbf{u}^0 - \mathbf{u}^{n+1}) + \boldsymbol{\nabla}\cdot(\boldsymbol{\sigma} h) + A(\boldsymbol{\tau}_a + \boldsymbol{\tau}_w) + \boldsymbol{\tau}_b + mf\boldsymbol{n}\times\boldsymbol{u} - mg\boldsymbol{\nabla}\eta, \tag{58}$$

where β' is the numerical tuning parameter introduced ~~by Lemieux et al. (2012)~~(as just β) by Bouillon et al. (2013), $\mathbf{u}^0$ is the ice velocity at the start of the sub iterations, and $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ ~~is the velocity at~~ are the velocities at the previous and current sub-iteration step, ~~and **F** donates all terms not dependent on, or multiplied with **u**~~respectively. Using the ~~notation and~~ approach of Hunke and Dukowicz (1997), this can be rewritten as

$$(\alpha^2 + \beta^2)u_1^{n+1} = \alpha u_1^n + \beta u_1^n + \frac{u_1^0 - u_1^n}{\beta' + 1}$$

$$+ \frac{\Delta t}{(\beta'+1)m}\left[\alpha\left(\frac{\partial \sigma_{1j}^{n+1}}{\partial x_j} + \tau_1\right) + \beta\left(\frac{\partial \sigma_{2j}^{n+1}}{\partial x_j} + \tau_2\right)\right]$$

$$(\alpha^2 + \beta^2)u_2^{n+1} = \alpha u_1^n + \beta u_2^n + \frac{u_2^0 - u_2^n}{\beta' + 1}$$

$$+ \frac{\Delta t}{(\beta'+1)m}\left[\alpha\left(\frac{\partial \sigma_{2j}^{n+1}}{\partial x_j} + \tau_2\right) + \beta\left(\frac{\partial \sigma_{1j}^{n+1}}{\partial x_j} + \tau_1\right)\right]$$

modified versions of equations (24) and (25):

$$(\alpha^2 + \beta^2)u_j^{n+1} = \alpha u_j + \beta v_j + \alpha\frac{u_j^0 - u_j^n}{b} + \beta\frac{v_j^0 - v_j^n}{b}$$

$$+ \frac{\Delta t_m}{b m_j M_{jj}}\left[\alpha\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma} dS\right]_x + \tau_x\right) + \beta\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma} dS\right]_y + \tau_y\right)\right] \tag{59}$$

~~where~~ and

$$(\alpha^2+\beta^2)v_j^{n+1} = \alpha v_j - \beta u_j + \alpha\frac{v_j^0 - v_j^n}{b} - \beta\frac{u_j^0-u_j^n}{b}$$
$$+\frac{\Delta t_m}{b m_j M_{jj}}\left[\alpha\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_y + \tau_y\right) - \beta\left(\left[-\int(\nabla N_j)h_j\boldsymbol{\sigma}dS\right]_x + \tau_x\right)\right], \quad (60)$$

where $b = \beta' + 1$ and α ~~,~~ and β ~~, and $\tau_{\{1,2\}}$ are as defined in Hunke and Dukowicz (1997).~~ become

$$\alpha = 1 + \frac{\Delta t_m}{b m_j}(c'\cos\theta_w + C_b) \quad (61)$$

$$\beta = \frac{\Delta t_m}{b}\left(f + \frac{c'\sin\theta_w}{m_j}\right). \quad (62)$$

Importantly, the BBM and EVP equations are recovered by setting ~~$\beta' = -1$ and $\mathbf{u}^0 = \mathbf{u}^n$~~ $b = 1$ and $\mathbf{u}^0 = \mathbf{u}^n$.

In neXtSIM, switching between the BBM and mEVP rheologies is done by calling separate functions to calculate the stress tensor, ~~σ~~ $\boldsymbol{\sigma}$. The momentum equation is then solved using equations (59) and (60), setting $\beta' = -1$ and ~~$u^0 = u^n$~~ $\mathbf{u}^0 = \mathbf{u}^n$ when using BBM or EVP.

4 Thermodynamics and column physics

The thermodynamics and column physics in neXtSIM are relatively simple compared to other sea-ice models, focusing on representing the processes most likely to be of interest when using brittle rheology, e.g. atmosphere–ocean–ice interactions in leads and polynyas. In addition, a diagnostic ridge ratio scheme is implemented in the model for the benefit of the sea-ice forecasting platform, neXtSIM-F (Williams et al., 2021) and a melt-pond scheme to improve long-term simulations, such as those of Boutin et al. (2023) and Regan et al. (2023). An ice-age tracer was implemented for Regan et al. (2023), with subsequent minor improvements.

4.1 ~~Column thermodynamics implementation~~Ice categories

~~A~~ In general, a sea-ice model's grid cell must be divided into at least two categories: ice and open water ~~(Hibler, 1979, e.g.)~~ (e.g. Hibler, 1979). Many models also use categories of different ice thicknesses~~(Hibler, 1980, e.g.)~~, with five thickness categories being a common choice ~~(Hunke, 2001, particularly after). NeXtSIM can use the minimal two categoriesof ice~~ (e.g. Bitz et al., 2001; Hunke . Contrary to pure thickness categories, neXtSIM includes a multi-category model based on state of development, considering three categories: consolidated ice, young ice and open water ~~or a three-category approach, categorising ice as either young or consolidated~~ (see appendix A2 of Rampal et al., 2019, who refer to the categories as "thick" and "thin" and not "consolidated" and "young"). ~~Vertical ice growth in the two ice classes is calculated by solving the heat diffusion equation in one dimension, following the zero-layer approach of Semtner (1976) for the young-ice class and the two-layer approach of Winton (2000) for the consolidated ice class~~

In this model, young ice is described by its concentration, A_y, volume per unit area, H_y, and snow volume per unit area, $H_{s,y}$. Consolidated ice is described by the concentration, A, volume per unit area, H, and snow volume per unit area, H_s. We assume that the young ice has no mechanical strength and simply follows the motion of the surrounding consolidated ice. Note the total ice concentration and volume per unit area are $A + A_y$ and $H + H_y$, and the total snow volume per unit area is $H_s + H_{s,y}$.

~~Ice growth~~ Young ice thickness is considered to be uniformly distributed with thickness $h_y = H_y/A_y$ required to be between the parameters h_{min} and h_{max}, which have the default values $h_{min} = 5$ cm and $h_{max} = 27.5$ cm. These parameter values affect the ice thickness in a similar way to Hibler's (1979) h_0 parameter. The evolution equations for A, H, H_s, A_t, H_t and $H_{s,t}$ have the following form:

$$\frac{D\phi}{Dt} = -\phi\nabla\cdot\boldsymbol{u} + \Psi_\phi + S_\phi, \tag{63}$$

where $\frac{D\phi}{Dt}$ is the material derivative that is defined for any scalar as

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + (\boldsymbol{u}\cdot\nabla)\phi. \tag{64}$$

Here $\nabla\cdot\boldsymbol{u}$ is the divergence of the horizontal velocity, Ψ_ϕ a sink/source term due to ridging, and S_ϕ a thermodynamical sink/source term. Volume conservation is imposed by setting $\Psi_H = -\Psi_{H_y}$ and $\Psi_{H_s} = -\Psi_{H_{s,y}}$ and an additional constraint is that $A_y + A \leq 1$.

The evolution of A, H, A_y and H_y is computed following three main steps:

1. Advection: The scheme solves the equation:

$$\frac{D\phi}{Dt} = -\phi\nabla\cdot\boldsymbol{u}, \tag{65}$$

for each conserved scalar quantity (A, H, A_y, H_y, etc.), using the advection Lagrangian scheme (see section 5). After this step, the concentration may exceed 1.

2. Mechanical redistribution: The scheme imposes the limit $A_y + A \leq 1$ on the total ice area by following those steps:

   (a) Compute the new open water concentration as:

$$A_0 = \max(0, 1 - A^n - A_y^n); \tag{66}$$

   (b) Compute the new young ice concentration as:

$$A_y^{n+1} = \min(1, \max(0, 1 - A^n - A_0)) \tag{67}$$

(c) If the young ice concentration has reduced ($A_y^{n+1} < A_y^n$), we also reduce the ice and snow volumes proportionally and transfer the excess to the consolidated ice category, conserving the mean young ice thickness H_y/A_y:

$$H_y^{n+1} = H_y^n \frac{A_y^{n+1}}{A_y^n} \tag{68}$$

$$H_{s,y}^{n+1} = H_{s,y}^n \frac{A_y^{n+1}}{A_y^n} \tag{69}$$

$$H^{n+1} = H^n + (H_y^n - H_y^{n+1}) \tag{70}$$

$$H_s^{n+1} = H_s^n + (H_{s,y}^n - H_{s,y}^{n+1}) \tag{71}$$

$$\Delta A = \frac{A_y^n - A_y^{n+1}}{\gamma}. \tag{72}$$

Here, we have transferred ice and snow volume from young to consolidated ice in a conservative manner, but we do not attempt to conserve ice area: γ is an aspect-ratio parameter (tuned to 10) that preferentially increases ice thickness over the ice area.

(d) Compute the new consolidated ice concentration as:

$$A^{n+1} = \max(0, \min(1, 1 - A_y^{n+1} - A_0 + \Delta A)) \tag{73}$$

(e) Apply ridging of young ice if $(A^{n+1} + A_y^{n+1}) > 1$ by setting $A_y^{n+1} = 1 - A^{n+1}$.

3. Lateral growth and melt: Freezing in open water ~~occurs when the ocean surface becomes supercooled. This ice is either added to the one ice class using Hibler's (1979) collection-depth approach or to the young-ice class following Rampal et al. (2019). In melting conditions, the ice concentration should decrease due to lateral melt. Young ice~~ is computed so that heat loss to the ocean that would otherwise cause supercooling is redirected to ice formation. The newly formed ice is assigned to the young ice category and assumed to have thickness h_{min}. The transfer from the young ice to the consolidated ice and the lateral melting of young ice are computed by applying the bounding limit

h_{max}; if $h_y^n > h_{max}$, then we update the variables as follows:

$$h_y^{n+1} = h_{max}, \tag{74}$$

$$A_y^{n+1} = \frac{h_{max} - h_{min}}{h_y^n - h_{min}} A_y^n, \tag{75}$$

$$H_y^{n+1} = A_y^{n+1} h_y^{n+1}, \tag{76}$$

$$H_{s,y}^{n+1} = \frac{A_y^{n+1}}{A_y^n} H_{s,y}^n, \tag{77}$$

$$H^{n+1} = H^n - (H_y^{n+1} - H_y^n), \tag{78}$$

$$A^{n+1} = A^n - (A_y^{n+1} - A_y^n), \tag{79}$$

$$H_s^{n+1} = H_s^n - (H_{s,y}^{n+1} - H_{s,y}^n). \tag{80}$$

In melting conditions, young ice melts first before the consolidated ice concentration decreases~~when using the young-ice class. NeXtSIM uses~~. NeXtSIM employs the lateral melt scheme of Mellor and Kantha (1989), which uses a fraction of the ice-ocean heat flux to melt ice laterally~~while the rest~~, while the remainder warms the ocean ~~mixed-layer~~mixed layer.

4.2 Column thermodynamics implementation

Vertical ice growth of the ice is calculated by solving the heat diffusion equation in one dimension. We have implemented two classical approaches: the zero-layer approach of Semtner (1976), and the two-layer approach of Winton (2000). The zero-layer approach assumes no heat capacity in the ice and snow, and is generally considered insufficient for longer simulations (e.g. Semtner, 1984). It is, however, sufficient for very thin ice, and so we use this for the young-ice class. We use the two-layer approach for the consolidated ice class.

The two-layer approach of Winton (2000) discretises the heat-diffusion equation using two internal temperature points in the ice and assumes zero heat capacity in the snow. It also accounts for the change in ~~the internal enthalpy in the ice~~ ice's internal enthalpy due to the presence of brine. Although substantially simpler than the more complete schemes (e.g. Bitz and Lipscomb, 1999; Turner and Hunke, 2015), this approach can capture the seasonal cycle and the main characteristics of thermodynamic melt and growth. The zero-layer approach of Semtner (1976) assumes that the ice has no internal temperature and no heat capacity. This straightforward approach is insufficient for general use (Semtner, 1984), but suffices to reproduce the melt and growth of ice in the young ice class, as this is generally thinner than about 30 cm. Therefore, neXtSIM uses the zero-layer approach for ice in the young-ice class, as it is both simpler and faster in execution. It uses the two-layer approach for the consolidated ice.

For both two- and zero-layer models, it is assumed that the ice temperature at the base is at the freezing point of the seawater below. The surface temperature is calculated by balancing the heat flux through the ice with the fluxes from the surface into the atmosphere as (derived from Semtner, 1976)

$$Q + \frac{\partial Q}{\partial T_s}\Delta T_s = \frac{k_i k_s}{k_s h_i + k_i h_s}\left(T_s + \Delta T_s - T_i\right). \tag{81}$$

Here, Q is the sum of the latent, sensible, short-wave, and long-wave fluxes at the surface, T_s is the surface temperature at the previous time step, ΔT_s is the change in surface temperature, k_s and k_i are the snow and ice heat conductivities, respectively, and h_s is the snow thickness. T_i is the ice temperature at the upper-temperature point, T_1 in the case of the two-layer model, or the temperature of the ice base in the case of the zero-layer model. For the two-layer model, h_i is the thickness of ice above the T_1 temperature point, while for the zero-layer model, h_i is the total ice thickness. Once we have calculated the surface fluxes, we calculate ice and snow thickness changes and internal ice temperatures for the two-layer model, following Semtner (1976) for the zero-layer model and Winton (2000) for the two-layer model.

The reanalysis prescribes the incoming short-wave and long-wave fluxes, while the model calculates the albedo, short-wave penetration, and outgoing long-wave radiation. The albedo is calculated using a simplified meltpond scheme (see section 4.3), and the outgoing long-wave radiation and its derivative are simply the black-body radiation and its derivative.

The turbulent fluxes are calculated using bulk formulas ~~relating~~ that relate the near-surface atmospheric and ~~ice surface~~ ice-surface states to surface fluxes, ~~as~~ since neXtSIM has so far always been forced by atmospheric model results. The bulk formulas for sensible and latent heat fluxes are taken from Stull (2012):

$$Q_s = C_h \rho_a c_p |\mathbf{u}_a|(T_s - \theta) \tag{82}$$

$$Q_l = C_h \rho_a L_s |\mathbf{u}_a|(q_s - q_a), \tag{83}$$

where C_h is a drag coefficient for turbulent heat exchange, ρ_a atmospheric density, c_p the specific heat capacity of air, L_s the latent heat of sublimation, $|\mathbf{u}_a|$ is the wind speed, θ the potential temperature at the reference height, q_s the specific humidity at the surface, and q_a the specific humidity at the reference height. The specific humidity may be ~~supplied by the~~ provided by reanalysis or calculated from the mixing ratio or the dew-point temperature.

The drag ~~coefficients~~ coefficient for turbulent heat exchange, C_h ~~and~~ , as well as the one for momentum exchange, C_m ~~used to calculate wind drag do not consider~~ , do not account for surface roughness; they only ~~take~~ consider atmospheric stability. The drag coefficients ~~then become~~

$$\cancel{C = \frac{k^2}{\left[\ln\left(\frac{z}{z_0}\right) - \Psi(\zeta)\right]^2},}$$

at the current time step then become

$$C_h^{n+1} = \frac{k^2}{\left[\ln\left(\frac{z}{z_0}\right) - \Psi_h(\zeta)\right]^2} \tag{84}$$

$$C_m^{n+1} = \frac{k^2}{\left[\ln\left(\frac{z}{z_0}\right) - \Psi_m(\zeta)\right]^2}, \tag{85}$$

where k is the von Kármán constant, z is the reference height, z_0 the surface roughness length, ~~Ψ the stability function~~Ψ_h and Ψ_m the stability functions for heat and momentum, and $\zeta = z/L$ the stability parameter, where L is the Obukov length. We use the stability functions ~~of~~ from Grachev et al. (2007) and Kader and Yaglom (1990) for both momentum and heat~~and calculate~~, and compute separate drag coefficients and stability functions for momentum and heat, using different reference heights (10 m and 2 m by default, respectively). The inverse of the Obukov length is

$$\frac{1}{L} = -\frac{kg(\overline{w'\theta_v'})_s}{u_*^3\bar{\theta}_v}, \tag{86}$$

~~is~~ where g is gravitational acceleration, $(\overline{w'\theta_v'})s$ is the surface virtual potential temperature flux, u_* the friction velocity, and $\bar{\theta}_v$ is the mean virtual temperature. Here, the inverse Obukov length is calculated by approximating some key parameters. The surface virtual potential temperature flux is approximated as

$$(\overline{w'\theta_v'})_s = \overline{w'\theta'}(1+ar) + a\theta\overline{w'r'}, \tag{87}$$

where the potential temperature flux is approximated as

$$\overline{w'\theta'} = C^n_h|\mathbf{u}|(T_s - \theta), \tag{88}$$

the mixing ratio flux as

$$\overline{w'r'} = \hat{C}^n_h|\mathbf{u}|\frac{q_s - q_a}{(1-q_s)(1-q_a)}, \tag{89}$$

~~and~~ where $a = 0.6078$, $r = q_a/(1-q_a)$ is the mixing ratio, $\bar{\theta}_v = \theta(1+aq_a)$ is the virtual potential temperature, and $\theta = T + \Gamma_d z_h$ is the potential temperature, with Γ_d being the adiabatic lapse rate. The frictional velocity is approximated as

$$u_* = \cancel{\sqrt{\hat{C}_m}}\sqrt{C^n_m}|\mathbf{u}|. \tag{90}$$

~~$\hat{C}_m$ and $\hat{C}_h$ are the drag coefficient of the previous time step.~~The inverse Obukov length is limited to $1/L \in [-1, 1]$ m^{-1}.

As advecting additional prognostic variables is very cheap, we set the drag coefficients as prognostic variables and ~~forego iterating over these equations using~~ can calculate C_h^{n+1} and C_m^{n+1} from C_h^n and C_m^n. In models where advection is more expensive, a common approach is to solve for drag iteratively, starting from the neutral drag ~~coefficients at the start.~~ and recalculating it about 5 times to obtain a more accurate estimate of the stability-dependent drag. The atmospheric stability changes slowly enough that using the drag coefficient from the previous time step is sufficient to calculate the Obukov length and the drag coefficient for the current time step.

4.3 Melt pond scheme

The neXtSIM model includes two albedo parameterisations that account for the evolution of melt ponds. The first one is the same as the “CCSM3” albedo scheme (Briegleb et al., 2004). The CCSM3 scheme is a shortwave radiation scheme that includes a melt pond parameterisation, representing the melt pond effect by reducing the albedo value when the surface temperature of sea ice increases. The second one is a recent addition and explicitly represents melt ponds.

The neXtSIM melt-pond scheme largely follows the work of Flocco et al. (2010), Holland et al. (2012) and Flocco et al. (2015). It relies on the following set of equations:

$$V_{\mathrm{pnd}} = h_{\mathrm{pnd}} f_{\mathrm{pnd}} \tag{91}$$

$$h_{\mathrm{pnd}} = \min(a_{\mathrm{emp}} f_{\mathrm{pnd}}, 0.9 h_i) \tag{92}$$

$$\rho_w \frac{dV_{\mathrm{pnd}}}{dt} = (1-r)\left[\rho_i \frac{dh_{\mathrm{top}}}{dt} + \rho_s \frac{dh_s}{dt} + F_{\mathrm{rain}}\right] \tag{93}$$

where V_{pnd} is the melt pond volume, h_{pnd} the pond depth, f_{pnd} the pond fraction, ρ_i and ρ_s the ice and snow densities, a_{emp} is the slope from a linear fit between the pond fraction and the pond depth from SHEBA observations (Perovich et al., 2003), and r is the fraction of surface meltwater that runs off the ice floes and is not collected by the ponds.

Like in Holland et al. (2012), the melt pond volume is a virtual reservoir that does not impact the freshwater and salt fluxes with the ocean. Like Flocco et al. (2010) and Holland et al. (2012), we first estimate the change in V_{pnd} from the accumulated water using (93), then estimate the fraction of ponded ice f_{pnd} by combining (91) and (92). We then check that h_{pnd} is not higher than $0.9h_i$ to obey (92); if this is the case, we reduce h_{pnd} by reducing V_{pnd} while keeping f_{pnd} untouched. This can be interpreted as the vertical draining of ponds that are almost as deep as the ice is thick.

For the sake of simplicity, we use a constant albedo α_p for the melt ponds. To avoid having a significant effect from very shallow ponds, we introduce a low limit threshold to the pond depth set to 0.05m and reduce f_{pnd} if needed.

In contrast to Holland et al. (2012) and Flocco et al. (2010), we consider r, the fraction of meltwater that runs off the ice, to be constant instead of a function of sea ice concentration. This change is motivated by new insights from MOSAIC data (Webster et al., 2022; Smith et al., 2024). Smith et al. (2024) suggest that, after an initial peak at $\simeq$40%, the fraction of surface freshwater that is stored in ponds (i.e., $1-r$) reduces to $\simeq$10%. They acknowledge that their data may be biased low due to high lateral draining at the location of their measurements, close to the edge of a floe. However, they find relatively similar results applying the same analysis to data from SHEBA, even though the fraction bounces back slowly to 20% by the end of the summer. From these observations, a reasonable range for r should be between $\simeq$60% and $\simeq$90%, while it is $\simeq$20% in compact ice using the formulation in Holland et al. (2012) and Flocco et al. (2010).

Since neXtSIM lacks an explicit representation of lateral and vertical draining, obtaining albedo and melt pond fraction values consistent with observations requires using a large value of r to avoid accumulating too much water in ponds by the end of the summer. Therefore, we recommend using a relatively high value for r (e.g., $r \simeq$0.9).

We also allow ~~changing~~ the a_{emp} value to be changed. In Holland et al. (2012) and Flocco et al. (2010), a_{emp} is a constant set to 0.8 from Perovich et al. (2003), but the MOSAIC data presented in Webster et al. (2022) (their Fig. 9 and 10) would suggest a higher slope value ($a_{\mathrm{emp}} \simeq$1) than the one obtained doing linear fit on SHEBA data only. Low ponded-ice fractions associated with deep ponds increase the value of a_{emp} and are also associated with draining, hence strongly depending on the location. Again, as draining is missing in our model, it is possible to increase the value of a_{emp} so that deeper ponds are not associated with large ponded-ice fractions.

In freezing conditions, a lid forms over the pond, which reduces the amount of meltwater in the pond:

$$\frac{dV_{\text{lid}}}{dt} = \frac{Q_{\text{ia}}}{\rho_i L_f} \tag{94}$$

$$\frac{dV_{\text{pnd}}}{dt} = -\frac{dV_{\text{lid}}}{dt}\frac{\rho_i}{\rho_w}, \tag{95}$$

where V_{lid} is the change in lid sea ice volume, Q_{ia} is the heat lost by the ice to the atmosphere, and L_f is the latent heat of fusion. If the meltwater under the lid keeps freezing:

$$\frac{dV_{\text{lid}}}{dt} = \frac{\min(Q_{\text{ia}} - Q_{\text{ic}}, 0) + Q_{\text{ic}}}{\rho_i L_f f_{\text{pnd}}}, \tag{96}$$

with Q_{ic} the heat lost by the meltwater in the pond:

$$Q_{\text{ic}} = \frac{T_{\text{pnd}} - T_i}{k_i V_{\text{pnd}}/f_{\text{pnd}}} \tag{97}$$

with T_{pnd} the temperature of meltwater in the pond that we assume to be equal to the freezing temperature of water with the same salinity as sea ice (constant in neXtSIM). If the ice lid thickness $V_{\text{lid}}/f_{\text{pnd}}$ reaches 1 m, the melt pond and its lid are removed ($V_{\text{pnd}} = V_{\text{lid}} = 0$).

In the end, the total albedo a_{total} is equal to:

$$a_{\text{total}} = A_0 a_{\text{ow}} + a_{\text{cons\_ice}} + a_{\text{young\_ice}}, \tag{98}$$

where a_{ow} is the albedo of open water, $a_{\text{cons\_ice}}$ the total albedo of the consolidated ice-covered area and $a_{\text{young\_ice}}$ the total albedo of the young ice-covered area. These terms are computed as follows:

$$a_{\text{cons\_ice}} = (A - f_s - f_{\text{pnd}})a_i + f_s a_s + f_{\text{pnd}} a_{\text{surf\_pnd}} \tag{99}$$

$$a_{\text{young\_ice}} = (A_y - f_{s,y})a_i + f_{s,y} a_s, \tag{100}$$

with a_i, a_s the albedos of bare ice and snow, respectively, and f_s and $f_{s,y}$ the fractions of consolidated and young sea ice covered by snow, which are estimated as:

$$f_s = \frac{h_s}{h_s + 0.02} \tag{101}$$

$$f_{s,y} = \frac{h_{s,y}}{h_{s,y} + 0.02}, \tag{102}$$

where h_s and $h_{s,y}$ are the snow thickness on consolidated and young ice, respectively. In (99), $a_{\text{surf\_pnd}}$ is the albedo at the surface of the pond (including the lid), which we compute as:

$$a_{\text{surf\_pnd}} = a_i + (a_{\text{pnd}} - a_i)e^{-\kappa * V_{\text{lid}}/f_{\text{pnd}}}, \tag{103}$$

with κ a constant related to the extinction coefficient of sea ice. Taking $\kappa = 4$ and $a_i = 0.76$ gives a dependency of the lid albedo to its thickness, similar to the logarithm law suggested by Ebert et al. (1995).

4.4 Ridge ratio calculations

NeXtSIM tracks the volume ratio of ridged ice throughout the simulation. Convergence in the Lagrangian framework is represented by a shape change of the model element, while mass in the element is conserved. When this happens, the mean thickness in the element increases, which can be ascribed to ridging. This is easily converted to a volume fraction, which can then be tracked. Ridges ~~only form~~ form only when the element is fully ice-covered; otherwise, ~~the~~ convergence reduces the ~~open water fraction. However, the following~~ open-water fraction.

The following is based on the model's prognostic variables for the consolidated ice category: H, the mean ice thickness over the element (or volume per unit area) and $A \in [0,1]$, the fractional ice concentration in the grid cell, as well as the diagnostic variable R, the volumetric ridge ratio. The mean thickness of level ice is thus $(1-R)H/A$, the volume of level ice is $(1-R)H$, and so on. The superscripts n and $n+1$ denote the current and following time steps, respectively. Even though ridging only occurs when the concentration is 100%, the following equations also hold when the concentration is less than ~~100%~~ that.

Tracking the volume ratio in the Lagrangian reference frame requires no assumptions on ridge characteristics such as shape, keel depth, sail height, or frequency, while some assumptions about thermodynamic melt and growth of ridges must be made. Taking into account the ridging of young ice (when using the young-ice class), four cases must be considered~~and associated assumptionsmust be made~~, for which we make the following assumptions:

1. Ridging of consolidated ice: Assume the conservation of mean thickness of level ice

2. ~~Ridging of young ice: Assume the conservation of volume of level ice~~

3. ~~Thermodynamic growth: Assume the conservation of the volume of ridged ice~~

4. ~~Thermodynamic melt: Assume that ridged and level ice melt at the same rate~~

~~When ridges form in consolidated ice due to convergence, we assume that the mean thickness of level ice is conserved. This~~ . This assumption can be written as

$$(1-R^n)\frac{H^n}{A^n} = (1-R^{n+1})\frac{H^{n+1}}{A^{n+1}}\cancel{.}, \tag{104}$$

~~Here, R is the volume ridge ratio, H is the mean ice thickness over the element (or volume per unit area), and $A \in [0,1]$ is the fractional ice concentration in the grid cell. The superscripts n and $n+1$ denote the current and following time steps, respectively. This equation~~ which can be rewritten as

$$R^{n+1} = 1-(1-R^n)\cancel{\frac{A^{n+1}}{A^nF}}\frac{A^{n+1}H^n}{A^nH^{n+1}}, \tag{105}$$

where $F = H^{n+1}/H^n$. ~~Ridges only form when $A^n = A^{n+1} = 1$, but the above equation holds in general due to the way A changes under Lagrangian advection (see section ??). The ratio F~~ The ratio H^n/H^{n+1} can also be related to the Lagrangian change in element area, S, since volume conservation

$$S^nH^n = S^{n+1}H^{n+1} \tag{106}$$

can be written as

$$\frac{S^n}{S^{n+1}} = \frac{H^{n+1}}{H^n} = F. \tag{107}$$

~~When using the young-ice class, we assume that ridged young iceis added to the consolidated ice class without affecting the~~

5. Ridging of young ice: Assume the conservation of mean thickness of level ice~~in that class. This~~. This assumption can be written the same as equation (104), but in this case, we always have ~~$A^n = A^{n+1}$~~$A^{n+1} = A^n$, and the change in H is due to mechanical redistribution ~~, i.e. $H^{n+1} = H^n + \Psi_y$, where Ψ_y is the thickness of ice transferred from the young-ice class due to convergence (see appendix A2 of Rampal et al., 2019)~~as per equation (70). The resulting evolution equation for R is therefore

$$R^{n+1} = 1 - (1 - R^n)\cancel{\frac{H^n}{H^n + \Psi_y}}\cdot\frac{H^n}{H^{n+1}} \tag{108}$$

~~For thermodynamic growthand melt, we first assume that ridged ice does not grow thermodynamically; thus,~~

6. Thermodynamic growth: Assume the conservation of the volume of ridged ice This assumption is equivalent to assuming that all ice added through thermodynamic growth is ~~level~~level-ice growth. This is a reasonable assumption, although it does not consider the consolidation of ridges. This assumption can be interpreted as the conservation of the volume of ridged ice

$$R^n H^n = R^{n+1} H^{n+1}, \tag{109}$$

or

$$R^{n+1} = R^n H^n / H^{n+1}. \tag{110}$$

~~For melt, we assume that both~~

7. Thermodynamic melt: Assume that ridged and level ice ~~melts~~ melt at the same rate ~~.~~This assumption leads us to conserve the ridge ratio at melt, i.e. $R^{n+1} = R^n$. This assumption is not entirely accurate, as ridges are known to melt faster than level ice under certain circumstances (e.g. Rigby and Hanson, 1976; Skyllingstad et al., 2003; Salganik et al., 2023). There is, however, considerable uncertainty related to how ridges change during the melt season, so we content ourselves with the simple assumption of conserving the ridge volume ratio during the melt.

4.5 Ice age tracers

Regan et al. (2023) introduced the tracers A_{my} and H_{my}, which are, respectively, the concentration and volume of multi-year ice (MYI) - defined as ice that survived the summer melt (for a full description of these tracers, see Regan et al. (2023)~~.)~~). They are affected by the following processes:

– *Freezing* contributes to an increase in FYI volume and concentration only.

– *Melting* acts as a sink term for both FYI and MYI concentration and volume.

– *Replenishment* of MYI occurs when the ice in a mesh element has undergone three consecutive days of mean growth following the height of the melt season (set as 1 August).

– *Convergence*, through ridging, of ice acts as a sink term for area only, not affecting volume.

An average sea ice age estimate was not included in the study of Regan et al. (2023). We estimate the sea ice age by a surface average age (~~α_A)~~ α_A) and a volume average age (α_V). The first is intended to relate to the age deduced from satellite observations, while the second is a more physical measure of the total age of the ice. The volume-averaged age tracer follows the same ideas as, e.g. Hunke and Bitz (2009), while we are not aware of an implementation of an area-averaged ice age tracer in the community.

The two tracers evolve according to

$$\alpha_A^{n+1} = w_A(\alpha_A^n + \Delta t) + (1 - w_A)\Delta t, \tag{111a}$$

$$w_A = \min\left\{1, \frac{A^n}{A^{n+1}}\right\}, \tag{111b}$$

$$\alpha_V^{n+1} = w_V(\alpha_V^n + \Delta t) + (1 - w_V)\Delta t, \tag{111c}$$

$$w_V = \min\left\{1, \frac{H^n}{H^{n+1}}\right\}. \tag{111d}$$

The definitions of w_V and w_A ensure that in melting conditions (H and A are decreasing), the age increases by Δt, while in freezing conditions, new ice is given the age Δt and the weights w_V and w_A are the fractions of ice determined by volume or area (respectively) that were already present at the previous time step.

5 Lagrangian mesh

5.1 Lagrangian advection and remeshing

NeXtSIM uses a triangular mesh with moving nodes and a remeshing scheme, as previously documented in Bouillon and Rampal (2015); Rampal et al. (2016b); Samaké et al. (2017). In this section, we ~~will give~~ provide an overview of the Lagrangian advection used in neXtSIM ~~, as well as detailing~~ and detail the conservative remapping scheme added since Samaké et al. (2017).

The neXtSIM mesh is an unstructured triangular mesh on an $\{x, y\}$-plane, usually a polar-stereographic plane, although other projections are technically possible. The initial model mesh can be constructed using tools such as Gmsh (Geuzaine and Remacle, 2009) or any mesh generator that can produce a mesh in the Gmsh format. The ~~finite element method used in neXtSIM stipulates that velocities are calculated on the nodes of the elements while tracers are computed on the element.~~

~~The Lagrangian advection then consists of the following steps:~~ initial mesh sets the mesh resolution, and subsequent adapted meshes retain that same resolution. This is achieved by creating a spatially varying field of mean vertex length for the initial mesh, and then requiring that the lengths of the vertices of later adapted meshes must fall within 20% of the mean vertex length of the initial mesh.

1. Move the nodes of the mesh ~~by $\Delta\mathbf{x} = \Delta t\mathbf{u}$~~
2. Modify concentration and thickness values to conserve volume and area
3. Check the distortion of the resulting mesh
4. ~~If~~ , and if the mesh is too distorted, then
   (a) Adapt the mesh where it has become too distorted
   (b) Copy model fields from old to new mesh, interpolating where new elements are introduced
   (c) Partition the mesh

~~Moving~~ In step 1, we move the mesh nodes by the displacement ~~$\Delta\mathbf{x} = \Delta t\mathbf{u}$~~$\Delta\mathbf{x} = \Delta t\mathbf{u}$, where Δt is the model time step and ~~$\mathbf{u}$~~ $\mathbf{u}$ is the ice velocity, which is trivial. Nodes on land boundaries are fixed with $\mathbf{u} = 0$~~, and while we calculate the velocity at the nodes at open boundaries , those nodes are also kept fixed .~~. For most neXtSIM setups, one or more sections of the domain edge are open, allowing ice to flow in or out of the domain. In- and out-flux through open boundaries is ensured by keeping the open-boundary nodes fixed and not updating these prognostic variables on elements that border an open boundary. This means that ice will flow out as if there was no resistance and flow in as if the ice state outside the boundary was the same as that inside it.

~~Once the nodes have moved, the~~ In step 2, we consider the changes in area of the elements, S ~~, changes~~(using the same notation as in section 4.4). To conserve the ice and snow volume and the ice area in each element, these prognostic variables must be multiplied ~~with~~ by the fraction $F = S^n/S^{n+1}$, where n and $n+1$ denote the state before and after the nodes are moved. This follows directly from volume and area conservation

$$S^n H^n = S^{n+1} H^{n+1}, \tag{112}$$

$$S^n A^n = S^{n+1} A^{n+1}, \tag{113}$$

where H denotes snow or ice volume and $A \in [1,0]$ the concentration fraction. ~~These~~ The ice concentration, ice thickness, and snow thickness are the only prognostic variables affected by ~~the change~~ changes in element area due to ~~the~~ Lagrangian advection. ~~In- and out-flux through open boundaries is ensured by keeping these nodes fixed and not updating these prognostic variables on elements bordering an open boundary. This means that ice will flow out as if there was no resistance and flow in as if the ice state outside the boundary was the same as that inside it~~This is because those are defined per unit area, so when the area changes, their value must change. The value of all other advected variables (such as temperatures and stresses) remains unchanged. Velocities are defined on the nodes and are, therefore, not affected by the deformation of the element.

The mesh adaptation ~~is made~~ in step 3(a) is performed using a port of the bi-dimensional anisotropic mesh generator, ~~bamg~~ BAMG (Hecht, 1998; Larour et al., 2012). The main advantage of ~~bamg~~ BAMG is that it can adapt the mesh only where it is too deformed, leaving other nodes in their original position. The mesher in ~~bamg~~ BAMG returns C++ objects with complete element and node connectivity, making it possible to quickly address nodes and elements connected to any given node or element of the mesh. The mesh adaptation is described in more detail in Rampal et al. (2016b).

In previous versions of neXtSIM (i.e. Rampal et al., 2016b), ~~fields were mapped from the old to the adapted mesh~~ step 3(b) was completed using an algorithm which identified areas where the ~~two~~ old and new meshes differed ("cavities") and then interpolated the fields from the old mesh to the cavities of the new mesh in a conservative manner. This approach was very efficient and highly optimised, but ultimately very complex and occasionally fragile. When developing the coupling routines to couple neXtSIM with an ocean model (see section ~~??~~5.3), we created a more robust conservative remapping approach, described in the following subsection. We could use this approach to map between the old and adapted meshes with only minimal code changes, and the relative simplicity of the scheme and its robustness, in addition to substantial code reuse, more than made up for the fact that the new scheme is less efficient than the old one.

~~While the~~ The conservative remapping approach is fully conservative, but remeshing introduces some local diffusion because a set of deformed elements is replaced with less deformed ones. A quantitative assessment of this diffusion is non-trivial and outside the scope of this paper. We note, however, that by construction, any translation or rotation of features such as leads and ridges is non-diffusive. This is a major advantage for neXtSIM, as the BBM rheology produces particularly well-localised leads.

Step 3(c) is necessary because, although the model is parallelised using MPI (Samaké et al., 2017), the remeshing process is not~~parallelised~~. Therefore, all mesh information and model fields must be gathered ~~to one~~ onto a single MPI processor to adapt the mesh~~,~~: remapping from the old to the new mesh and partitioning the new mesh, which is then scattered to the other MPI processes~~afterwards~~. The mesh is partitioned using the METIS partitioner.

~~Doing the~~ Performing mesh adaptation and field remapping on a single MPI process ~~generates~~ creates a substantial bottleneck for large problems, even ~~if~~ though the Lagrangian advection itself is highly efficient. This was already identified by Samaké et al. (2017), but their solution~~of using~~, an arbitrary Lagrangian-Eulerian (ALE) approach, was technically challenging and was abandoned. As a result, neXtSIM is, in practice, limited to meshes with $\mathcal{O}(10^5)$ elements running on $\mathcal{O}(100)$ MPI processors. For larger problems, the model ~~mostly stops scaling~~ scales poorly with the number of MPI processors, ~~meaning that the execution time stays the same, regardless of how many processorsare used.~~ so execution time remains constant regardless of the number of processors. For regional Arctic setups, this limits the model use to resolutions coarser than about 5 km.

5.2 Conservative remapping

NeXtSIM uses a conservative remapping scheme, ~~where~~ in which values from a donor mesh are mapped to a receiver mesh ~~using weighted averaging~~via weighted averaging, with the area of overlap between elements in the two grids as weights. The weights are ~~calculated only~~ computed once and then applied ~~to~~ across many fields at a very low computational cost. The weight calculation algorithm below relies on two features of ~~bamg~~BAMG: the complete connectivity information of the mesh and a

~~bamg~~ BAMG utility, which allows one to find the element covering any point in the domain. The algorithm is simple (see Algorithm 1) ~~, relying~~ and relies on only three checks for element nodes~~and~~ , element-element ~~intersection~~ intersections, and recursion. The algorithm returns two lists: a list of elements of the donor mesh overlapping a given element of the receiver mesh, $\mathcal{L}$, and a list of the ~~area~~ areas of overlaps, used as weights, $\mathcal{W}$. In practice, a single check routine is called for the receiver element to build $\mathcal{L}$ and $\mathcal{W}$.

Algorithm 1 The algorithm behind the recursive function to generate a list, $\mathcal{L}$, of donor elements, $\mathcal{E}_d \in \Omega_d$, overlapping the receiver element $\mathcal{E}_r$, and the area of overlap, $\mathcal{W}$. $\mathcal{P}$ is a temporary list of points. N is a function returning the nodes of an element, I is a function returning the element number, X is a function returning the intersection point ~~and vertices, $\mathcal{V}$,~~ of two elements, and A is a function returning the area of the polygon formed by the list of points given on input.

```
if Any of N(E_d) ∈ E_r then
    Add all N(E_d) ∈ E_r to P
    for all E'_d ∈ Ω_d sharing N(E_d) and I(E'_d) ∉ L do
        Call self for E'_d
        Append to L and W from recursive call
    end for
end if
if Any of N(E_r) ∈ E_d then
    Add all N(E_r) ∈ E_d to P
end if
if Any X(E_d, E_r) exists then
    Add X(E_d, E_r) to P
    for all E'_d ∈ Ω_d sharing ~~V~~ a side with E_d and I(E'_d) ∉ L do
        Call self for E'_d
        Append to L and W from recursive call
    end for
end if
Add I(E_d) to L
Add A(P) to W
return L, W
```

The first step of the algorithm is to find an element in the receiver mesh, $\mathcal{E}_r \in \Omega_r$, that overlaps with the barycentre of a given element in the donor mesh, $B(\mathcal{E}_d) \in \Omega_d$. This is done using an interpolation utility in ~~bamg~~BAMG, which uses the underlying quadtree to find the element in a given mesh that overlaps a set of points in the domain. This is by far the most time-consuming part of the algorithm. Once all overlapping elements have been found, the algorithm ~~loops through~~ iterates over all donor elements, $\mathcal{E}_d$, performs three simple checks, and ~~initiates checks on~~ then recursively checks neighbour elements, ~~recursively,~~ as needed.

The first check is to test if any of the nodes of the donor element are inside the receiver element, $N(\mathcal{E}_d) \in \mathcal{E}_r$ (line 1 of algorithm 1). The list of element nodes, $N(\mathcal{E})$, is available as a ~~bamg~~ BAMG data structure, and the test is performed using a point inclusion in polygon test (Franklin, 2023). This test is both accurate and extremely fast. It can also be used for ~~any polygon shape~~arbitrary polygonal shapes, which is advantageous when using this algorithm to interpolate between a triangular mesh and a ~~quadrangular~~ quadrilateral grid (see section ~~??~~5.3). Suppose a node of the donor element is inside the receiver element. In that case, this is registered in a list, $\mathcal{P}$, the element number, $I(\mathcal{E}_d)$, is added to the list of elements, $\mathcal{L}$, and the function performing the checks is called for each of the elements of the donor mesh which shares the node.

The second check tests if any of the receiver nodes are inside the donor element, $N(\mathcal{E}_r) \in \mathcal{E}_d$ (line 8). The procedure is identical to the one in the first test, except that no recursive call is necessary in this case. Any $I(\mathcal{E}_r)$ are added to $\mathcal{L}$ and $N(\mathcal{E}_r)$ to $\mathcal{P}$.

The third check is to test if any of the vertices of $\mathcal{E}_d$ and $\mathcal{E}_r$ intersect (line 11). If they do, the intersection point $X(\mathcal{E}_d, \mathcal{E}_r)$ is added to $\mathcal{P}$. The intersection detection is ~~quick~~ fast and can be ~~done~~ performed for any two line segments, so the algorithm can be used for both triangular and ~~quadrangular~~ quadrilateral meshes. If the vertices of $\mathcal{E}_d$ and $\mathcal{E}_r$ intersect, then the function calls itself to perform the checks on the element in Ω_d sharing the intersecting vertex, $\mathcal{V}$, with $\mathcal{E}_d$.

After these three checks, $\mathcal{P}$ is a list of points delineating a polygon of the overlap between $\mathcal{E}_d$ and $\mathcal{E}_r$. The algorithm ends by adding the element number, $I(\mathcal{E}_r)$, to the list $\mathcal{L}$ and the area of the polygon, $A(\mathcal{P})$, to the list $\mathcal{W}$ (lines 18 and 19). The area calculation is done using the shoelace formula (also known as Gauss's area formula or the surveyor's formula, e.g. Braden, 1986). Note that because of the recursive calls within the first and last checks, the lists $\mathcal{L}$ and $\mathcal{W}$ will be fully populated by all overlapping elements at this point. Additional checks and early exits may be implemented when all $N(\mathcal{E}_d) \in \mathcal{E}_r$ or all $N(\mathcal{E}_r) \in \mathcal{E}_d$. This gives a marginal improvement in execution time. Once the lists $\mathcal{L}$ and $\mathcal{W}$ are obtained, calculating the field values in $\mathcal{E}_r$ from the elements $\mathcal{E}_d$ listed in $\mathcal{L}$, using the weights from $\mathcal{W}$, is trivial.

An illustrative example of this algorithm is shown in figure 3, where the donor mesh consists of triangles and the receiver mesh of quadrilaterals - as would be the case when coupling neXtSIM with a typical finite-difference model (see section 5.3). The receiver element, $\mathcal{E}_r$, is drawn in a thick solid line. Its centre point is within the donor element labelled 1, $\mathcal{E}_{d,1}$. The element number for element 1, $I(\mathcal{E}_{d,1})$ is added to the list of elements, $\mathcal{L}$. The first check of the algorithm finds the single node of $\mathcal{E}_{d,1}$, N', which lies within $\mathcal{E}_r$ and adds it to the list of points for element 1, $\mathcal{P}_1$. The program then calls itself on all the potential donor elements, sharing this node, starting with element 2 (the order depends on the internal ordering of triangles given by BAMG). The recursive call to element 2 adds $I(\mathcal{E}_{d,2})$ to $\mathcal{L}$. It then finds again that $N' \in \mathcal{E}_r$ and calls itself on element 3. This recursive call results in a call on element 4, then 5, and finally on element 6. In element 6, the first check triggers no further recursive calls as all the other element numbers are already in $\mathcal{L}$.

After adding N' to $\mathcal{P}_6$, the program proceeds to perform checks two and three on element 6. Check two finds the corner point of the quadrilateral that lies in element 6, since $N(\mathcal{E}_r) \in \mathcal{E}_d$, and adds it to $\mathcal{P}_6$. Check three finds the intersection points between the left-hand side of the quadrilateral and the side shared by elements 5 and 6, and between the bottom and the side shared by elements 6 and 1, as $X(\mathcal{E}_{d,6}, \mathcal{E}_r)$. These are also added to $\mathcal{P}_6$. As the $I(\mathcal{E}_{d,1})$ and $I(\mathcal{E}_{d,5})$ are already in $\mathcal{L}$, no recursive calls

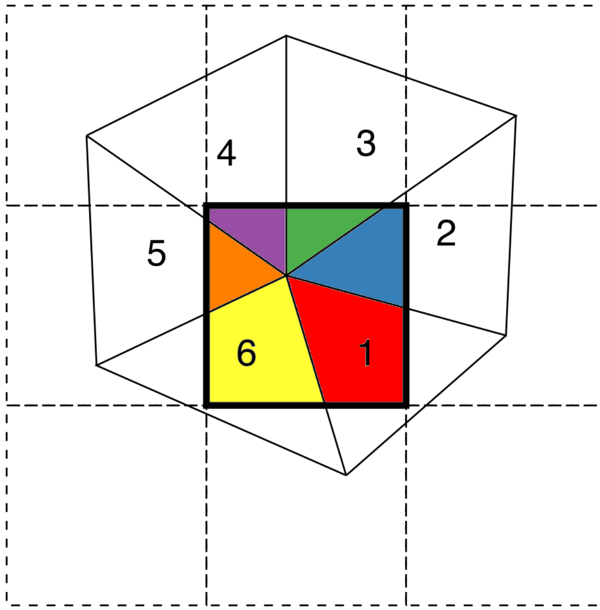

Figure 3. An illustrative example of how the conservative remapping algorithm works. In this case, the donor mesh consists of triangles, while the receiver mesh consists of quadrilaterals. The algorithm calculates the area of overlap for each triangle, starting in the triangle labelled 1, proceeding to no. 2, etc.

are made. The program then calculates the area of the yellow polygon in the figure, as $A(\mathcal{P}_6)$ and adds this to the list $\mathcal{W}$, and returns to the recursive call on element 5.

In element 5, the program completes the tests and computes the area of the orange polygon shown in the figure. No recursion is made, since all element numbers are already in $\mathcal{L}$. The program then exits to the recursive call on element 4, where the area of the purple polygon is computed, and so on. This procedure returns the list $\mathcal{W}$ of the area of all the contributing polygons (coloured in the figure), as well as the list $\mathcal{L}$ of the corresponding element numbers. These lists allow the program to calculate the weighted average contribution of the triangular elements to the resulting value in the quadrilateral. Once the lists $\mathcal{L}$ and $\mathcal{W}$ exist for all the receiver elements, the reverse contribution can be calculated as well.

5.3 OASIS coupling

NeXtSIM can be coupled to both ocean and wave models through the OASIS coupler (e.g. Boutin et al., 2021, 2023). OASIS (Craig et al., 2017) is a coupler that requires minimal code modification in the models being coupled and is very well suited to coupling substantially different code bases, as is the case here. To couple different models, each model must implement initialisation and finalisation routines and include the OASIS `get` and `put` function calls ~~for receiving and sending~~ to receive and send coupling fields.

The OASIS interface implementation in neXtSIM is unusual because of the moving mesh neXtSIM uses. OASIS is designed for fixed meshes, and the OASIS initialisation routine should specify the mesh shape, coordinates, and domain decomposition. This allows OASIS to calculate interpolation weights and the communication pathways between the MPI domains of the different models once at startup. In the case of neXtSIM, the mesh and domain decomposition changes continuously throughout the simulation. Therefore, the OASIS interface in neXtSIM is set up so that the two models communicate through a fixed exchange grid within neXtSIM and on neXtSIM's root MPI processor.

For most setups, the exchange grid is identical to the grid of the other (ocean or wave) model. When coupling to an ocean model, the ice and ocean models must share an identical coastline to ensure the conservation of fluxes between ~~the two models~~them. The initial mesh of neXtSIM is then constructed to exactly trace the coastal boundaries of the exchange and ocean grid. NeXtSIM then uses the conservative remapping algorithm (section ~~??~~5.2) to interpolate between the moving mesh and the fixed exchange grid. To avoid recalculating the exchange weights at every time step (as ~~per section ??~~in section 5.2), the mesh is ~~considered stationary in~~ assumed stationary between remeshing steps.

The coupling procedure described above relies on several compromises in terms of both efficiency and accuracy. Recalculating the interpolation weights ~~every time a~~ each time remeshing occurs is ~~obviously~~ clearly less efficient than ~~only calculating them~~ computing them only at the start of the run. The remapping algorithm we use is, however, efficient enough that this only takes about 5% of the run time in a typical ice–ocean coupled setup (such as the one used by Boutin et al., 2023). Another efficiency loss ~~is due to the fact that~~ arises because all communication must ~~go~~ pass through the neXtSIM root MPI processor. It is difficult to assess this cost, but implementing proper parallel coupling was a significant step forward in the ~~OASIS development~~ development of OASIS (Craig et al., 2017). In a typical ice–ocean setup, about 12% of the total ~~run time~~ runtime is spent in the OASIS `put` and `get` routines, but this includes the time neXtSIM spends waiting for the ocean model. Some loss of accuracy ~~is incurred~~ occurs because the mesh is ~~considered to be~~ assumed to remain stationary between remeshing steps, but this has no greater impact on the solution than numerical noise inside the model itself (discussed in Korosov et al., 2023). In setups ~~at~~ with resolutions between 10 and 20 km ~~resolution~~ (e.g. Ólason et al., 2022; Boutin et al., 2023), remeshing occurs at ~~between three and~~ three to six hourly intervals, resulting in displacement errors no larger than approximately 1/~~3rd~~ 3 of the model resolution.

5.4 Input/Output operations

NeXtSIM supports two ~~types of outputs~~output formats: snapshots of the model state with complete mesh information on the one hand, and netCDF output on a fixed, rectangular output grid on the other. The first is used to output the exact model state and is also used for restart files. The second is used to generate temporal averages and to simplify post-processing and analysis, as it is the standard format used in oceanography and climate science. The mesh-based snapshots are binary files of mesh and model field values, accompanied by ASCII files with the file header information. We have developed a library of Python routines called pynextsim[1] to read, analyse, and modify the mesh-based output from neXtSIM.

In its simplest incarnation, the neXtSIM netCDF output uses a nearest-neighbour interpolation from the neXtSIM mesh to the output grid specified for the netCDF file. At each time step, the output field values in the element covering the centre point

[1]https://github.com/nansencenter/nextsim-tools

of each output grid cell are added to that grid cell's value, and the average is then written to file at the given output interval. ~~This sampling technique provides the same kind of results as moorings in the real ocean. The prefix of the netCDF outputs and functions in the code draw their names from this similarity.~~ Finding the overlap between the model mesh and the output grid is time-consuming, so we assume the mesh is stationary between remeshing steps. This is the same approach as in the coupling code, which is based on the same underlying functions as the netCDF output code.

The common code base ~~of~~ for the coupling functions and netCDF output functions ~~allows us to take advantage of~~ enables us to use the conservative remapping capabilities developed for coupling and remeshing. When in coupled mode, the netCDF output grid is the same as the coupler exchange grid and instead of using a nearest-neighbour interpolation, the conservative remapping approach is used. This has the added benefit that the netCDF outputs are conserving, which is crucial for budget calculations (e.g. Ólason et al., 2021; Boutin et al., 2023). This capacity can also be used in stand-alone mode, provided a specially crafted initial mesh and output grid definitions are provided, following the same requirements as for the mesh and exchange grid in the coupled case.

6 Simulation examples and use-cases

This section gives an overview of the model results and capabilities.

6.1 An idealised test case

While neXtSIM was developed primarily to run large-scale simulations, the model can also be set up for idealised simulations to explore the details of certain physical processes and their parameterisations. To illustrate this, we show the results of an idealised experiment, exploring the difference between two damage update schemes. The two experiments consist of simulating a 1000×1000 km ice block fixed on one side to a coast with ice thickness 1 m, concentration 100 %, damage 0, and wind 5 m/s blowing towards the coast (see Figure 4.a). The domain is located with the geographic North Pole at its lower left corner, resulting in a clear deviation of velocities due to the Coriolis effect. Both experiments run 24 hours on a mesh with ≈10 km resolution, at 5 s time step, with 3 sub-steps. Other parameters are taken from Ólason et al. (2022) and Table 1. In one experiment, the stress and damage are updated using the formulation from Ólason et al. (2022) (i.e., with $\Delta t/t_d$), and in the second one~~- using equations 44and 45 (without~~ , using equation 44.

Of these two schemes, the latter ensures that the stress is always on or inside the failure envelope, while the former is derived from a differential equation for damage evolution. This differential equation is

$$\frac{\partial d}{\partial t} = \frac{1-d_{\text{crit}}}{t_d}(1-d), \tag{114}$$

where $t_d = \frac{\Delta x}{c_E}$ is a characteristic time scale for damaging, based on the grid cell size, Δx and the propagation time of shear waves, c_E. An Euler-forward discretisation of this equation gives

$$d^{n+1} = d^n + (1-d_{\text{crit}})(1-d)\frac{\Delta t}{t_d}. \tag{115}$$

In practice, the only difference between the two approaches is the factor $\Delta t/t_d$ ~~).~~ which, because of the stability requirements of the explicit solver, must always be less than one. However, we can consider equation (44) to be a special case of equation (115) with $\Delta t/t_d = 1$ to force the stresses to always remain within the failure envelope.

Figure 4.a shows the initial concentration and total ice motion over 24 hours, while panels b and c show total deformation from the two experiments. Both deformation fields look somewhat similar; however, the results of the second experiment show narrower deformation zones and more small-scale faults in sea ice. To illustrate the differences in the process of ice faulting, we can analyse the propagation of the internal stress and damage in these experiments (see Figure 5 and the video in the supplement material).

After 2 hours, the fields of internal normal stress (σ_N, Figure 5, a and ~~f~~d) show quite similar patterns of two elastic wave trains propagating from the upper corners, where ice is attached to the coast and where the faulting started. Ice damage increases in the elements where the normal and shear stress exits the Mohr-Coulomb envelope, and lines of enhanced ice damage appear on the maps (Figure 5, b and e). On these maps, the main difference ~~of the second experiment is in the~~ between the experiments is that the second one shows a narrower front of stress waves and, consequently, more localised damage. Analysis of the evolution of normal and shear stress ~~evolution~~ (see Figure 5, c and f) can help ~~understand the reason for~~ explain this difference.

Two points (marked ~~by~~ in blue and red ~~colour~~ in Figure 5, b and e) are located well outside the region of faulting~~and~~ , and the stress behaviour (shown by blue and red trajectories in Figure 5, c and f) is quite similar. Two other points (marked ~~by~~ with orange and green dots) are ~~placed in~~ located within the faulting zone. The orange dot experienced ice damage (and stress relaxation) only once, while the green one underwent several cycles of damage, stress relaxation, and stress build-up. Here, the stress trajectories differ significantly.

In the first experiment, the $\Delta t/t_d$ factor is always below one, and once the stress tensor of the element exits the fault envelope, the damage begins to increase slowly, and the stress begins to decrease slowly. Thus, the stress tensor returns to the envelope in more than one sub-step, manifested in a curved trajectory of normal/shear stress going outside the envelope for several steps and sub-steps ~~.~~ (green line in figure 4.c). At the same time, as stress is still relatively high and damage is not high enough, the stress starts to build up in the neighbouring elements, creating a broad front of the elastic wave. The broad front, in turn, leads to broader lines of increased damage.

In the second experiment, $\Delta t/t_d = 1$, the damage increases sufficiently quickly to allow the stress to return to the envelope in one sub-step. That pushes the stress tensor trajectory on the envelope, creating narrower elastic waves with ~~better damage localisation~~ higher damage localisation and small-scale background deformation lines. Such rapid change, however, tends to generate instabilities, leading to slightly noisier stress and damage fields~~and the appearance of small-scale deformation lines.~~ . The way damage is modelled inevitably causes rapid, local changes in the stress state, which can promote noise in the stress and damage fields, as is visible in figure 4. This noise, however, does not appear to affect large-scale observable metrics, such as deformation statistics, considered in e.g. Rampal et al. (2016b) and Ólason et al. (2022). We therefore have not attempted to damp this noise. It is, however, interesting that the form of damage evolution affects the level of noise in the simulation. Further exploration of the origin, impact, and potential damping methods for this noise is left to future study.

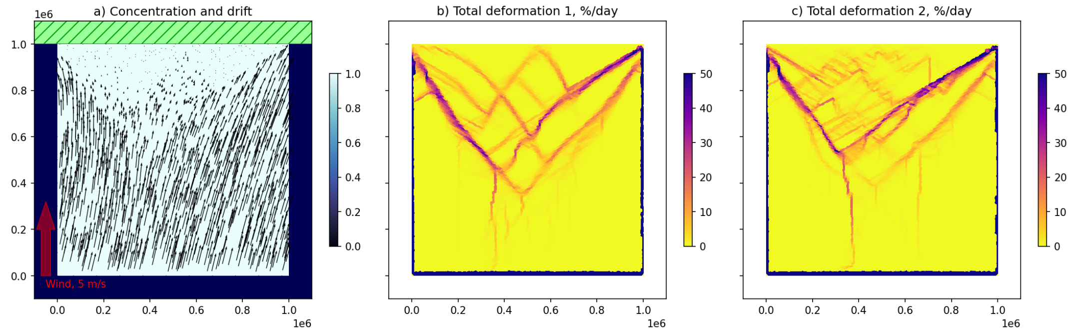

Figure 4. ~~(a) Experiment setup for~~ Drift and deformation after 24 hours of model integration in an idealised experiment. The domain is ~~100×100~~1000×1000 km with a constant wind of 5 m/s blowing along the positive y-direction. Panel (a) shows the modelled concentration and drift, (b) the total deformation using the damaging scheme from Dansereau (2016) and Ólason et al. (2022), and (c) the total deformation using the damaging scheme in equation (44).

These results demonstrate how using a dynamic timestep shorter than the damage propagation time, i.e. $\Delta t < t_d$, allows stresses in the model to temporarily exceed the failure envelope. This results in less localised damage and deformation fields. These supercritical states are short-lived but nonetheless affect the simulation results~~, nonetheless~~. Supercritical states are not physical but could be considered acceptable if the formulation improved model stability without significantly impacting the results. The numerical stability of the model in large-scale setups does, however, not appear to be affected by the formulation of equations (44) and (46), even though the idealised tests do show increased instabilities when omitting $\Delta t/t_d$. The large-scale deformation in our 20 km resolution runs is also not affected to any substantial degree. In neXtSIM, we choose the formulation in equations (44) and (46) to avoid the unphysical supercritical states.

6.2 Stand-alone ~~15 year~~ 15-year simulation

As a large-scale sea ice model, neXtSIM's main goal is to represent the evolution of sea ice's large-scale properties most accurately, for time scales from a few days to decades. Here, we run a 15-year-long simulation of the model in a stand-alone configuration to illustrate the model's capacity to represent different aspects of sea ice properties in the context of a stand-alone simulation.

As in Ólason et al. (2022), we use the hourly ERA5 reanalysis (Hersbach et al., 2020) for atmospheric forcing and the TOPAZ4 reanalysis (Sakov et al., 2012) for oceanic forcing. Initial sea ice thickness and concentration are set from the PIOMAS reanalysis (Zhang and Rothrock, 2003). Initial sea ice damage is set to zero. We use the same domain as Ólason et al. (2022), but with a coarser horizontal resolution of 20 km (instead of 10 km). Key model parameters that differ from Ólason et al. (2022) or were introduced in this study are summarised in 1. The model was tuned to obtain reasonably good results in the observational comparison in the following sections. The tuning is based on the work of Williams et al. (2021), Ólason et al. (2022),Boutin et al. (2023), and Korosov et al. (2025).

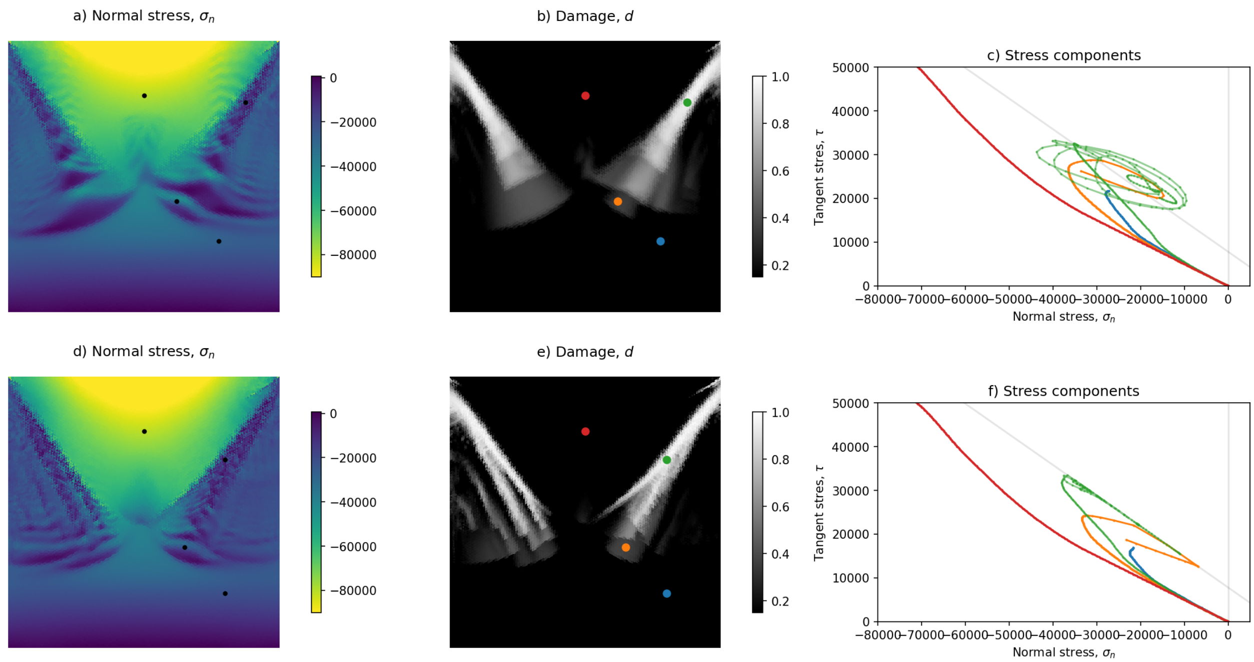

Figure 5. Normal stress, damage, and stress evolution at selected points of the domain using the two different damage and stress update schemes discussed in the text; with $\Delta t/t_d$ (a, ~~e~~b, and ~~e~~c) and without $\Delta t/t_d$ (~~b,~~ d, e, and f). The fields are shown after 24 hours of integration.

Table 1. Parameters used for the 15-year-long simulation in this study that differ from Ólason et al. (2022) (or earlier publications) or that are introduced in this study for the first time.

Parameter	symbol	new value	value in previous neXtSIM studies
Exponent of the thickness dependency for the ridging threshold	n	2	1.5
Cohesion at the reference scale	c_{ref}	1.35 MPa	2 MPa
Sea ice (bare) albedo	a_i	0.76	0.64
Snow albedo	a_s	0.9	0.85
Ratio of ice-ocean heat flux used to melt ice laterally	Φ_m	0.2	0.5
Melt pond albedo	a_{pnd}	0.3	new
Fraction of surface meltwater that runs off the ice	r	0.92	new
Extinction coefficient of sea ice	κ	4 m^{-1}	new
Slope of the linear fit between pond fraction and depth	a_{emp}	0.8	new

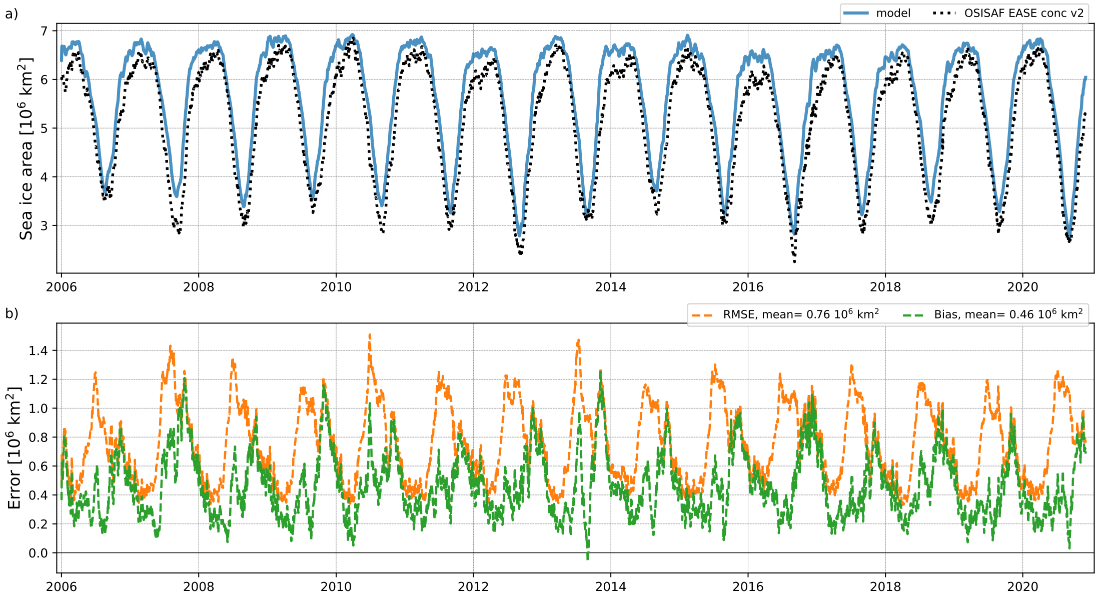

Figure 6. (a) Modelled and observed (OSI-SAF CDR) sea ice extent evolution for the period from January 2006 to December 2021. (b) IIEE and bias between the modelled sea ice drift and the OSI-SAF CDR dataset over the same period. The domain considered is shown in Figure 7e.

To evaluate the consistency of neXtSIM's results, we compare them with various datasets of observed quantities. For sea ice extent, we use sea ice concentration from the climate data record of the EUMETSAT Ocean and Sea Ice Satellite Application Facility (OSI-SAF, Lavergne et al., 2019). For sea ice volume and thickness, we use the dataset combining the observations retrieved from the CryoSAT-2 and SMOS satellites, referred to as CS2SMOS (version 2.6, Ricker et al., 2017). For sea ice drift, we use the OSI-SAF climate data record (v1.0, Lavergne and Down, 2023). We use the dataset from the University of Bremen (Pohl et al., 2020; Istomina et al., 2023) retrieved using Sentinel-3 and ENVISAT for sea ice albedo and melt ponds. In our analysis, we generally compare model results and observations using the bias and RMSE. For sea ice extent, the bias is computed as described in Williams et al. (2021), and like them, we use the Integrated Ice Edge Error (IIEE, Goessling et al., 2016) instead of the RMSE as it provides more information about the model's capacity to capture the extent evolution. We use the same spatial domain for integrated quantities as Boutin et al. (2023), which includes most of the Arctic Ocean but excludes the Greenland Sea and seas south of Bering Strait. This is because we mainly focus on pack ice, for which neXtSIM was originally developed (Rampal et al., 2016a).

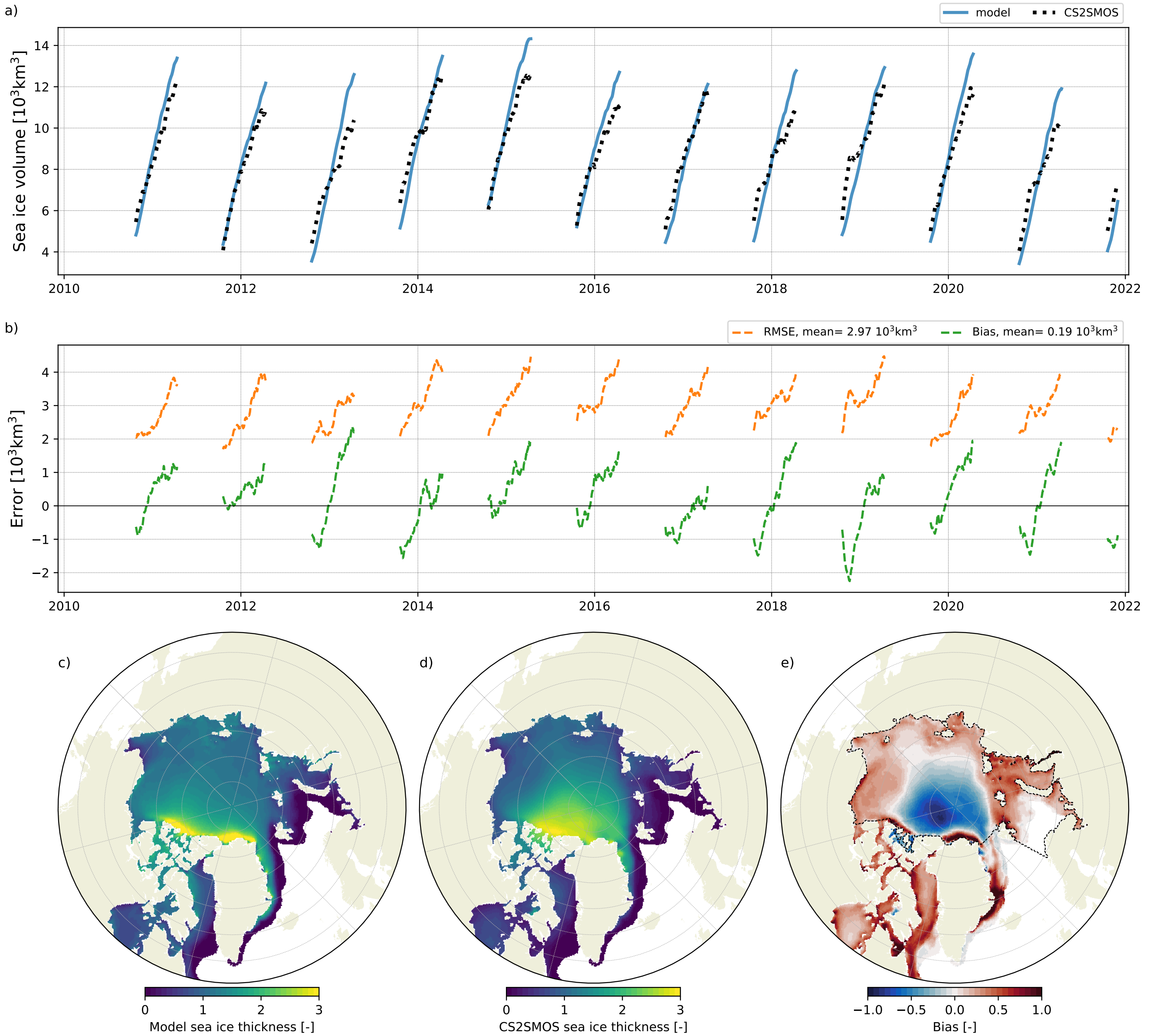

Figure 7. (a) Modelled and observed (CS2SMOS) sea ice extent evolution for the period from January 2006 to December 2021. (b) RMSE and bias between the modelled sea ice drift and the CS2SMOS dataset over the same period. (c,d) show the 2010–2021 November to April sea ice thickness climatology in the model and as estimated by CS2SMOS. (e) is the climatology of the sea ice thickness bias between the model and CS2SMOS. These climatologies are computed from monthly averaged files. The black dashed contour in (e) shows the domain used to compute integrated quantities in section 6.2 (e.g., in panels a,b here).

6.2.1 Extent and volume

Sea ice extent and volume are generally the first quantities to be evaluated due to their effect on climate. As shown in Ólason et al. (2022) and Boutin et al. (2023), neXtSIM can generally simulate their evolution in a way consistent with observations (Figure 6 and 7).

As in the ice-ocean coupled setup presented in Boutin et al. (2023), sea ice extent is generally consistently captured (Figure 6). IIEE peaks at the end of summer, when sea ice extent is minimal, but remains generally under 1M km^2. It is associated with a positive bias, meaning the simulation overestimates the yearly minimum extent. IIEE is lower in winter, as most of the domain used for its computation is covered by stopped ice. As in Boutin et al. (2023), including the totality of the domain has little qualitative effect on the IIEE evolution.

Modelled sea ice thickness also shows reasonable agreement with data from CS2SMOS (Figure 7). The interannual variability is visibly captured, and the slope corresponding to sea ice growth during the freezing season generally agrees with observations. The bias and RMSE both grow from autumn to the end of winter (Figure 7b). In all years but 2011, the simulation first underestimated the amount of ice left at the end of the summer but then ended up growing more ice than in observations by April, when observations stopped being available. Looking at the thickness distribution, the general spatial patterns are captured, with thicker ice north of Greenland where the oldest ice is expected to be found (Figure 7c,d), and thinner ice in areas generally covered by first-year ice. The difference in sea ice thickness distribution reflects the behaviour of bias and RMSE (Figure 7e), with the thickness of older and thicker ice underestimated while the thickness of younger and thinner ice is overestimated. This is a typical bias of sea ice models (Watts et al., 2021), and results can be improved by tuning some parameters, like the albedo, that are not very constrained and have a large impact on the amount of ice surviving the summer. Thinner ice thickness is more sensitive to parameters like the maximum thickness of the young ice class.

6.2.2 Drift

Sea ice drift is also an important metric to assess the quality of a sea ice model, as it significantly impacts sea ice balance through its transport of sea ice and its export through the Fram Strait (in the Arctic). A good representation of sea ice drift also improves the sea ice thickness distribution by properly capturing the location of older and thicker ice (Regan et al., 2023).

The simulation shows an agreement comparable to previous studies using neXtSIM (Rampal et al., 2016b; Williams et al., 2021; Boutin et al., 2023), with a generally low RMSE (less than 4 km/day most of the year, figure 8). Both short-term (a few days) and long-term (seasonal and interannual) variabilities are captured. The bias with the OSI-SAF CDR dataset shows a seasonal cycle as it increases in the summer. However, satellite-derived observations are not available in the summer. Instead, the OSI-SAF CDR uses estimates from a free-drift model with high uncertainties. Lavergne and Down (2023) suggest these estimates may be biased low, meaning it is uncertain whether our simulation is biased in the summer or not.

The mean Arctic-wide drift field also compares well with the OSI-SAF CDR dataset, showing a clear Beaufort Gyre circulation and a Trans-Polar Drift (figure 9). The modelled drift direction is generally very good, but the drift speed in the marginal seas teands to be too slow, while the drift in the central pack is too fast. The reason for this is difficult to pinpoint,

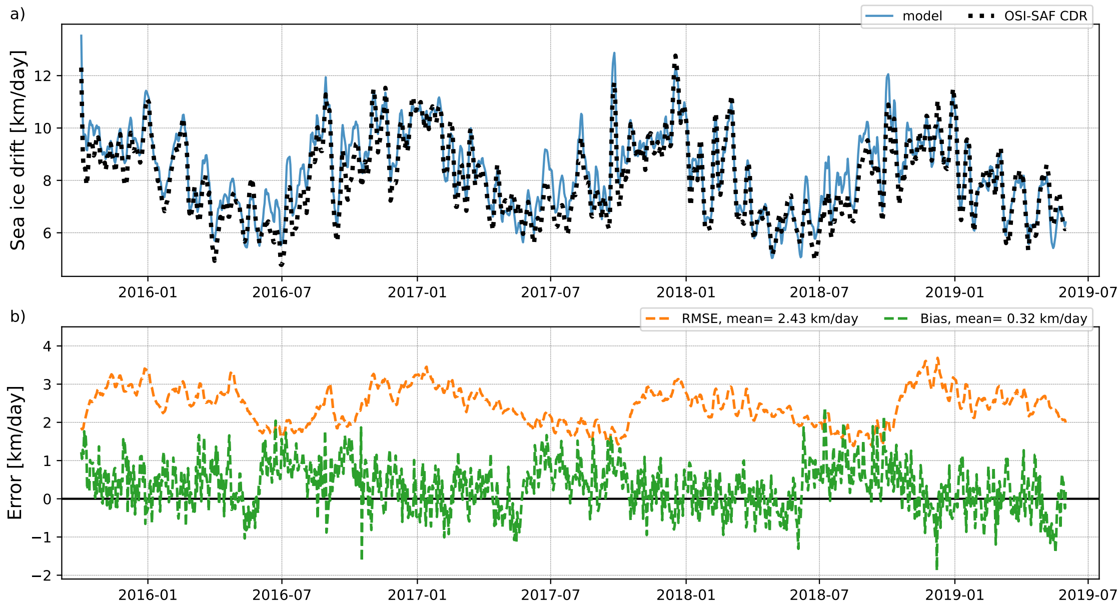

Figure 8. (a) Modelled and observed (OSI-SAF CDR) sea ice drift evolution for the period from October 2015 to June 2019. (b) RMSE (of drift speed) and bias between the modelled sea ice drift and the OSI-SAF CDR dataset over the same period. The domain considered is shown in Figure 7e.

especially since these are the results of a stand-alone model, without an active ocean model beneath. Lacking an active ocean will likely dampen the ice's drift speed.

The simulations used here do not use the stability-dependent drag coefficients introduced in section 4.2. This is because taking atmospheric stability into account slightly increases the bias and RMSE between the modelled sea-ice drift and the OSI-SAF CDR dataset. We have been unable to determine why improving the physical representation in the model in this way gives worse results than the simpler approach of only using neutral drag coefficients. This issue will be investigated further in the near future.

6.2.3 Stress and damage

The BBM rheology in neXtSIM differs from VP and its derivatives in both the calculation of stress and the additional damage variable. No direct observations of these variables exist to compare with the model fields, but it is still instructive to present an example here. In this case, showing a snapshot from a single time step is more informative than showing long-term means, since averaging smooths out any features of interest. Figure 10.a shows the modelled normal stress. We note that this is negative almost everywhere, indicating compressive stress, but that bands of high and low compressive stress are also visible in the field.

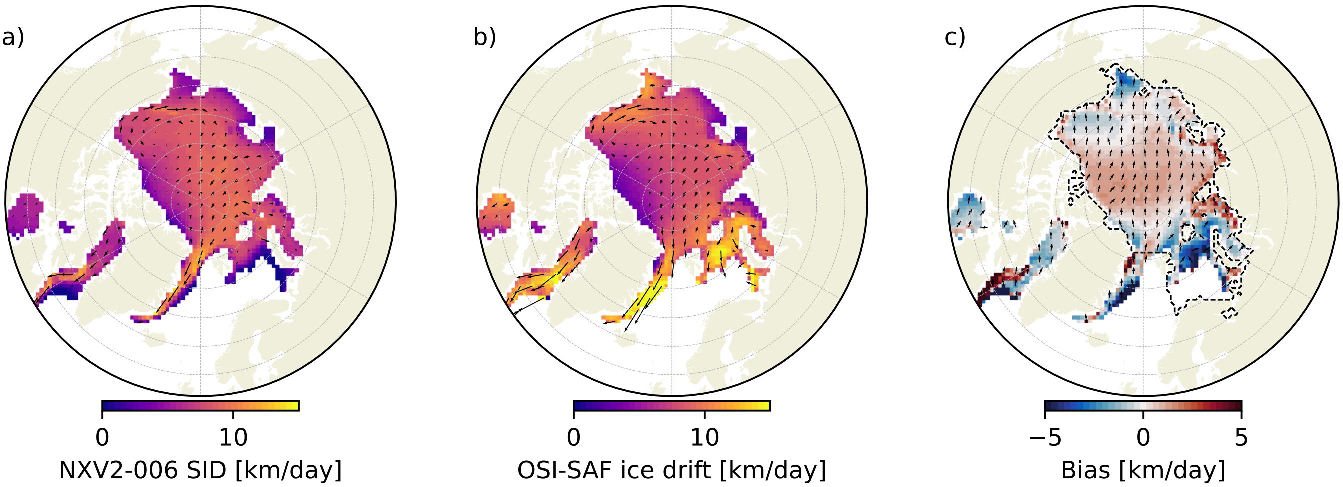

Figure 9. (a) Modelled and (b) observed (OSI-SAF CDR) sea ice drift for the period from October to June, over 2015–19. (c) The model bias compared to observations, with colours showing the bias in speed and arrows bias in direction. Arrows pointing up indicate no directional bias.

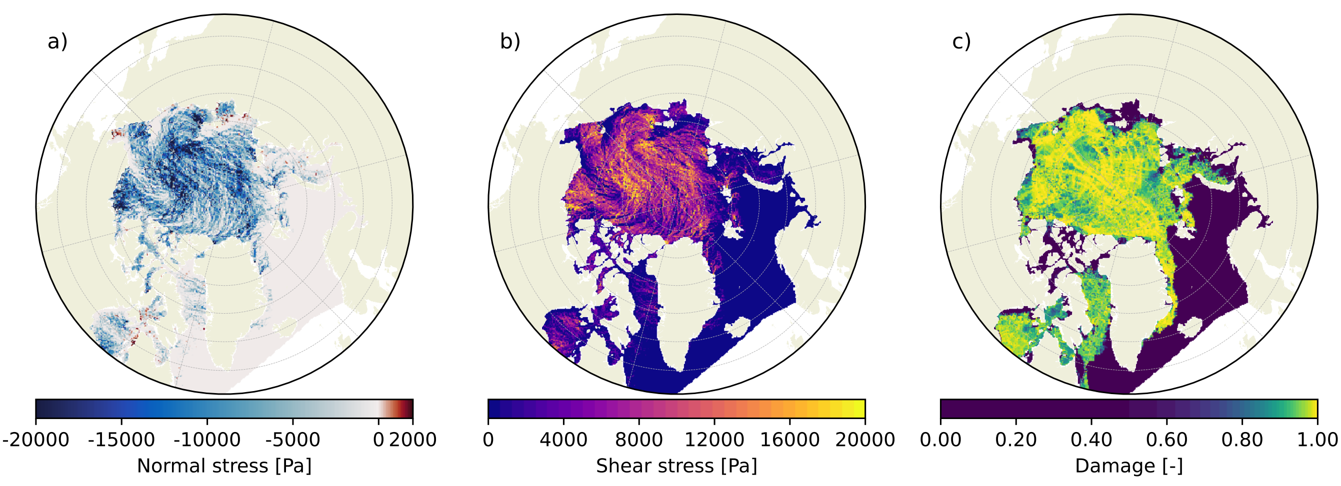

Figure 10. Modelled (a) normal stress, (b) shear stress, and (c) damage snapshots on the first time step of March 3rd, 2008.

These bands can also be seen in the shear stress field (figure 10.b), indicating that the stress is concentrated along bands of shear failure. This feature is also visible in the damage field (figure 10.c), which also shows similar striping. It is interesting to note that there is no direct correspondence between high stress and high damage, as the stresses respond to both the damage field and atmospheric and oceanic forcing. Finally, we note several isolated places along the coast where the normal stress is positive, indicating the presence of land-fast ice.

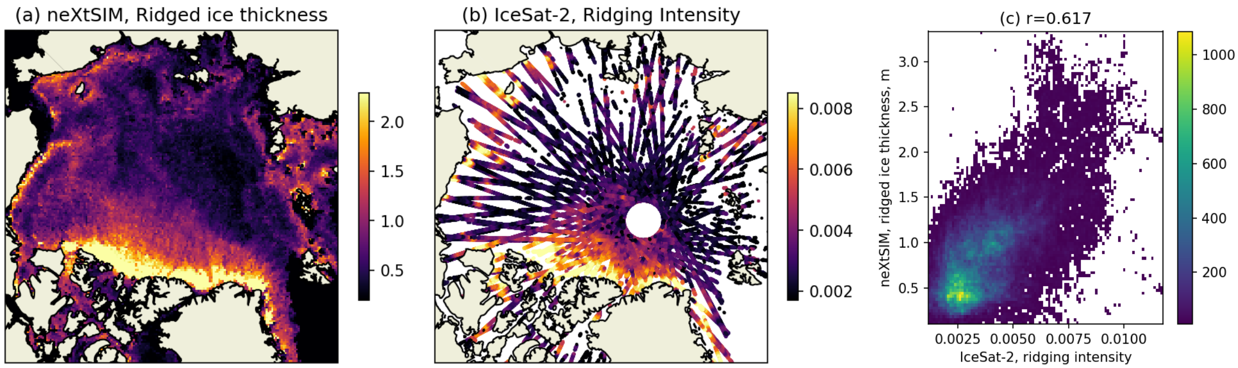

Figure 11. (a) Ridged ice thickness from neXtSIM on 2019-04-04, 12:00:00. (b) Ridging intensity from IceSat-2 orbits acquired between 2019-04-01 and 2019-04-07. (c) Scatter plot of IceSat-2 and neXtSIM products.

Table 2. Pearson's correlation between neXtSIM and IceSat-2 roughness-related products.

	Thickness	Ridged ice ratio	Ridge ice thickness
IS2 Roughness	0.512	0.465	0.587
IS2 Ridging intensity	0.521	0.562	**0.617**

6.2.4 Ridge ratio compared to observed sea-ice roughness

We compared the neXtSIM ridged ice thickness (computed as a product of ice thickness and ridged ice ratio) with ridging intensity from IceSat-2 (calculated as the ridge frequency per 1 km segment multiplied by the mean ridge sail height within the segment, Farrell et al., 2020). The maps in Figure 11 show remarkable similarity in depicting high roughness in perennial ice and coastal areas with first-year ice, with very low ice roughness in the Central Arctic. The correlation between the neXtSIM ridged ice volume and the IceSat-2 ridging intensity is also high (0.617, Figure 11), while correlations of other neXtSIM and IceSAT2 products are lower (see Table 2).

6.2.5 Melt-pond fraction compared to satellite observations

This paper introduced the newly implemented melt pond scheme in neXtSIM and its effect on albedo in section 4.3. Here, we illustrate the behaviour of this scheme by comparing the seasonal distribution of the simulated sea ice albedo with estimates from the University of Bremen (Figure 12). We notice that both the magnitude and the distribution of albedo are generally captured. This distribution is characterised by lower albedo at the margins of the sea ice cover and higher in the pack, with a strong latitude dependency in the observation dataset (see June in particular). The simulated albedo follows a general tendency

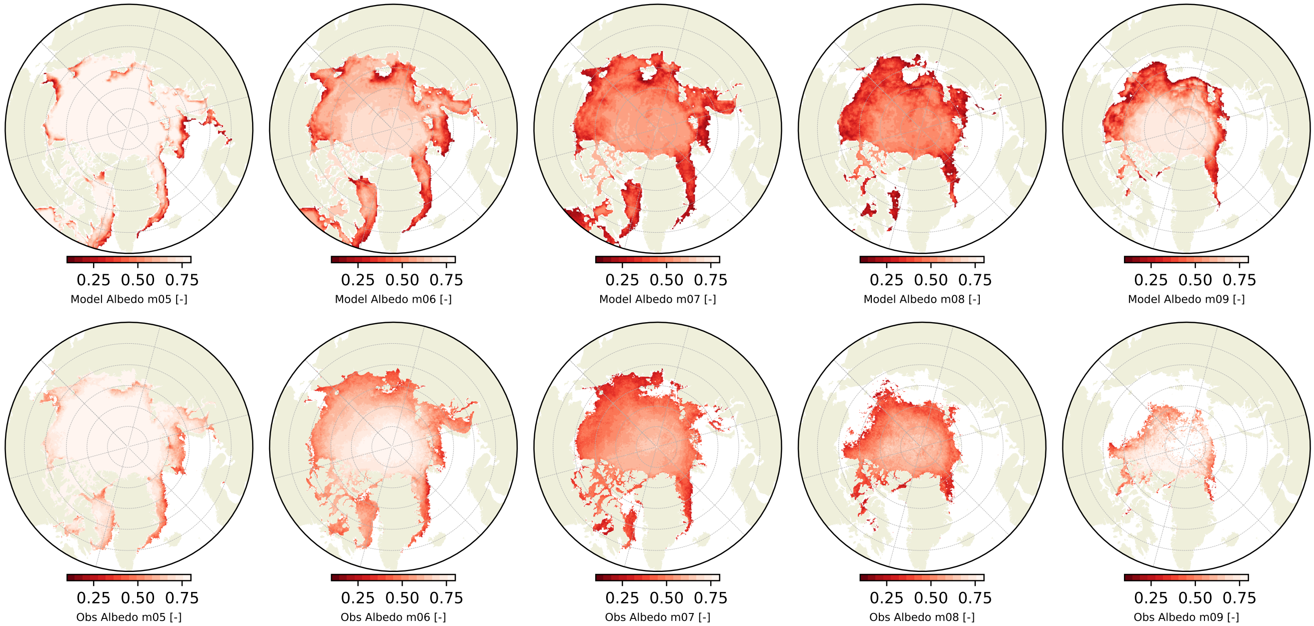

Figure 12. 2017–2021 climatology of the simulated and the observed broadband albedo for the months from May to September, when the observation product from Uni. Bremen is available.

but with a lower contract between pack ice and the ice edge and a generally lower albedo. This is partly explained when we look at the melt pond fraction (MPF) distribution (Figure ~~12~~13). This distribution differs between observations and the simulation. Observed melt pond fraction shows large values close to the sea ice edge and lower values in the interior, while July and August simulated MPF distributions are relatively uniform. We also notice that the modelled MPF is biased low in May–June but biased high later in August and September.

This behaviour of the MPF in the simulation can be explained mainly by simplifications we made in the parameterisation. The main simplifications are the absence of draining and the choice of a constant ratio of melted freshwater ending up in the ponds. Smith et al. (2024) suggest that this ratio is high early in the melt season (up to $\simeq 40\%$) and lower later (down to $\simeq 10\%$). In the absence of draining, using a higher ratio would result in largely overestimated MPF by the end of summer, and the choice of a constant ratio has the advantage of simplicity. Adding a dependency on concentration, as in, e.g., Holland et al. (2012), who assume lower concentrations have a lower ratio due to higher draining, would only increase the current biases by increasing MPF in the pack and reducing it in the margins. In reality, sea ice topography, roughness, and porosity must play a significant role in both this ratio and the amount of draining, but these dependencies are not yet fully understood. Nevertheless, activating the melt pond scheme albedo generally increases the sensitivity of simulated sea ice to melt (not shown), which helps reduce the extent of bias at the end of the summer. Still, it reduces the thickness of older ice, increasing the thickness bias we mentioned earlier. This effect is probably exacerbated by refrozen meltwater not counting towards ice mass in the model.

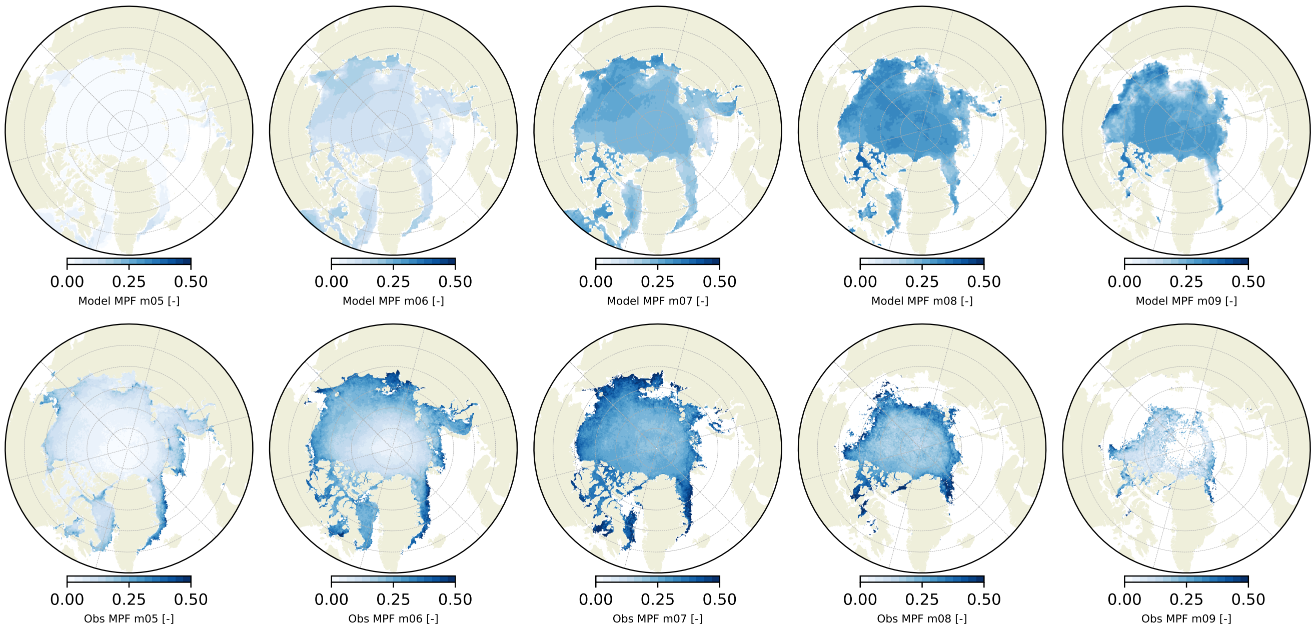

Figure 13. 2017–2021 climatology of the simulated and the observed melt pond fractions for the months from May to September, when the observation product from the University of Bremen is available.

7 Interest of using neXtSIM

In this manuscript, we present the neXtSIM model and demonstrate its ability to simulate sea ice dynamics across a range of spatial and temporal scales. Here, we discuss the main advantages of using neXtSIM and its BBM rheology, as well as some of the current limitations.

Regarding computational performance, Ólason et al. (2022) found that the BBM rheology in neXtSIM is approximately 25% slower than the mEVP rheology at a 10 km resolution, when using the same model time step and number of sub-cycles for the momentum equation solver. However, because BBM is a relatively new rheology in sea ice modelling, its computational efficiency has not yet been fully optimised. Ólason et al. (2022) also suggested parameter choices that could loosen the time-step requirements for BBM and make it faster than mEVP, highlighting its potential for further improvement.

The current version of neXtSIM faces two main computational limitations, both of which are already discussed in this manuscript. First, the remeshing procedure is not yet parallelised (see Section 5.1). Second, when neXtSIM is coupled with other models, the conservative remapping weights must be recomputed at regular intervals. Both limitations stem from the use of a moving mesh, which remains relatively uncommon in large-scale Earth system models. Ongoing developments aim to address these issues and have also motivated the implementation of the BBM rheology in Eulerian sea ice models such as SI3 (Brodeau et al., 2024).

From a sea ice modelling perspective, the main advantage of neXtSIM is its demonstrated ability to accurately reproduce small-scale deformation patterns, even at coarse spatial resolutions (Ólason et al., 2021, 2022; Bouchat et al., 2022; Hutter et al., 2022). This makes it particularly well-suited for investigating the role of small-scale deformations in sea ice evolution (Boutin et al., 2023; Rheinl and their influence on other components of the Earth system, such as the ocean (Regan et al., 2025). Determining when, where, and to what extent these deformations are significant remains an open question—one that neXtSIM should be well equipped to explore.

8 Summary and conclusions

This paper presents the latest version of the next-generation sea-ice model, neXtSIM. It also marks the first open-source release of neXtSIM. This version depends on the core functionality already laid out in Rampal et al. (2016b), Samaké et al. (2017), and Ólason et al. (2022), but contains several significant new developments. The OASIS coupling interface and the associated conservative remapping algorithm are the most important technical developments. This was already used by Boutin et al. (2021, 2022, 2023); Regan et al. (2023), but the technical details are described here for the first time. The ridge ratio and ice age tracers are also currently part of the Copernicus Marine Service's Arctic sea-ice forecast[2], but only described here in detail for the first time. The simulation results presented here also include more recent developments and physics implementations, such as the melt pond ~~scheme~~ and atmospheric drag ~~scheme, which are yet to be~~ schemes, which have not yet been used in scientific publications or forecast products.

. The neXtSIM code used for this paper (version 2.5.0) is available on Zenodo https://doi.org/10.5281/zenodo.14724536. The model is also available on https://github.com/nansencenter/nextsim. The code is released under the MIT licence.

The sea ice concentration dataset from OSI-SAF includes the OSI-450 sea ice concentration product available at ftp://OSI-SAF.met.no/reprocessed/ice/conc/ (last visited September 2024) and the interim version for data from 2021 onwards, OSI-430-b, available at ftp://OSI-SAF.met.no/reprocessed/ice/conc-cont-reproc/ (last visited September 2024). The sea ice drift climate data record is available at ftp://OSI-SAF.met.no/reprocessed/ice/drift_lr/v1/. The CS2SMOS sea ice thickness product is available at ftp://ftp.awi.de/sea_ice/product/cryosat2_smos/ (last visited September 2024). Albedo and melt pond data from the University of Bremen are available at https://data.seaice.uni-bremen.de/olci/ (last visited September 2024). ERA5 is available at https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5. TOPAZ4b reanalysis data are available at https://doi.org/10.48670/moi-00007. PIOMAS outputs are available at http://psc.apl.uw.edu/research/projects/arctic-sea-ice-volume-anomaly/data/model_grid (last visited September 2024).

. EÓ led the writing and wrote most of the text. GB ran the 15-year simulation and wrote the sections on extent, volume, drift, and melt-pond fraction. TW developed and wrote up the ice age tracers and wrote the section on ice categories. HR developed the FYI and MYI category code. AK ran and wrote up the idealised test case and the ridge ratio comparison section. DF, EÓ, and GB developed the melt pond

[2]https://doi.org/10.48670/moi-00004

parameterisation. RD and EÓ developed the drag parameterisation based on atmospheric stability. All authors contributed to the model development through direct code contributions, discussions of physics and parameterisation implementations, or software development issues.

. The authors declare that no competing interests are present

. The development of neXtSIM has been supported by multiple projects, funded nationally in Norway, through European collaborations, and internationally. The writing of this paper was supported by Norges Forskningsråd (grant no. 325292).

References

Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., Dunne, J. P., Griffies, S. M., Hallberg, R., Harrison, M. J., Held, I. M., Jansen, M. F., John, J. G., Krasting, J. P., Langenhorst, A. R., Legg, S., Liang, Z., McHugh, C., Radhakrishnan, A., Reichl, B. G., Rosati, T., Samuels, B. L., Shao, A., Stouffer, R., Winton, M., Wittenberg, A. T., Xiang, B., Zadeh, N., and Zhang, R.: The GFDL global ocean and sea ice model OM4.0: Model description and simulation features, Journal of Advances in Modeling Earth Systems, 11, 3167–3211, https://doi.org/10.1029/2019ms001726, 2019.

Amitrano, D., Grasso, J. R., and Hantz, D.: From diffuse to localised damage through elastic interaction, Geophysical Research Letters, 26, 2109–2112, https://doi.org/10.1029/1999gl900388, 1999.

Assmy, P., Fernández-Méndez, M., Duarte, P., Meyer, A., Randelhoff, A., Mundy, C. J., Olsen, L. M., Kauko, H. M., Bailey, A., Chierici, M., Cohen, L., Doulgeris, A. P., Ehn, J. K., Fransson, A., Gerland, S., Hop, H., Hudson, S. R., Hughes, N., Itkin, P., Johnsen, G., King, J. A., Koch, B. P., Koenig, Z., Kwasniewski, S., Laney, S. R., Nicolaus, M., Pavlov, A. K., Polashenski, C. M., Provost, C., Rösel, A., Sandbu, M., Spreen, G., Smedsrud, L. H., Sundfjord, A., Taskjelle, T., Tatarek, A., Wiktor, J., Wagner, P. M., Wold, A., Steen, H., and Granskog, M. A.: Leads in Arctic pack ice enable early phytoplankton blooms below snow-covered sea ice, Scientific Reports, 7, https://doi.org/10.1038/srep40850, 2017.

Bielak, J.: The finite element method: A primer, Springer International Publishing, ISBN 9783031563690, https://doi.org/10.1007/978-3-031-56369-0, 2024.

Bitz, C. M. and Lipscomb, W. H.: An energy-conserving thermodynamic model of sea ice, Journal of Geophysical Research: Oceans, 104, 15 669–15 677, https://doi.org/10.1029/1999jc900100, 1999.

Bitz, C. M., Holland, M. M., Weaver, A. J., and Eby, M.: Simulating the ice-thickness distribution in a coupled climate model, Journal of Geophysical Research: Oceans, 106, 2441–2463, https://doi.org/10.1029/1999jc000113, 2001.

Bouchat, A., Hutter, N., Chanut, J., Dupont, F., Dukhovskoy, D., Garric, G., Lee, Y., Lemieux, J.-F., Lique, C., Losch, M., Maslowski, W., Myers, P. G., Ólason, E., Rampal, P., Rasmussen, T., Talandier, C., Tremblay, B., and Wang, Q.: Sea ice rheology experiment (SIREx), part I: Scaling and statistical properties of sea-ice deformation fields, Journal of Geophysical Research: Oceans, https://doi.org/10.1029/2021jc017667, 2022.

Bouillon, S. and Rampal, P.: Presentation of the dynamical core of neXtSIM, a new sea ice model, Ocean Modelling, 91, 23–37, https://doi.org/10.1016/j.ocemod.2015.04.005, 2015.

Bouillon, S., Fichefet, T., Legat, V., and Madec, G.: The elastic–viscous–plastic method revisited, Ocean Modelling, 71, 2–12, https://doi.org/10.1016/j.ocemod.2013.05.013, 2013.

Boutin, G., Williams, T., Rampal, P., Ólason, E., and Lique, C.: Wave–sea-ice interactions in a brittle rheological framework, The Cryosphere, 15, 431–457, https://doi.org/10.5194/tc-15-431-2021, 2021.

Boutin, G., Williams, T., Horvat, C., and Brodeau, L.: Modelling the Arctic wave-affected marginal ice zone: a comparison with ICESat-2 observations, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 380, https://doi.org/10.1098/rsta.2021.0262, 2022.

Boutin, G., Ólason, E., Rampal, P., Regan, H., Lique, C., Talandier, C., Brodeau, L., and Ricker, R.: Arctic sea ice mass balance in a new coupled ice–ocean model using a brittle rheology framework, The Cryosphere, 17, 617–638, https://doi.org/10.5194/tc-17-617-2023, 2023.

Braden, B.: The Surveyor's Area Formula, The College Mathematics Journal, 17, 326–337, https://doi.org/10.1080/07468342.1986.11972974, 1986.

Briegleb, B., Bitz, C., Hunke, E., Lipscomb, W., Holland, M., Schramm, J., and Moritz, R.: Scientific Description of the Sea Ice Component in the Community Climate System Model, Version 3, NCAR Technical Note NCAR/TN-463+STR, National Center for Atmospheric Research, Boulder, Colorado, https://doi.org/10.5065/D6HH6H1P, 2004.

Brodeau, L., Rampal, P., Ólason, E., and Dansereau, V.: Implementation of a brittle sea ice rheology in an Eulerian, finite-difference, C-grid modeling framework: impact on the simulated deformation of sea ice in the Arctic, Geoscientific Model Development, 17, 6051–6082, https://doi.org/10.5194/gmd-17-6051-2024, 2024.

Connolley, W. M., Gregory, J. M., Hunke, E., and McLaren, A. J.: On the consistent scaling of terms in the sea-ice dynamics equation, Journal of Physical Oceanography, 34, 1776–1780, https://doi.org/10.1175/1520-0485(2004)034<1776:OTCSOT>2.0.CO;2, 2004.

Craig, A., Valcke, S., and Coquart, L.: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geoscientific Model Development, 10, 3297–3308, https://doi.org/10.5194/gmd-10-3297-2017, 2017.

Danilov, S., Wang, Q., Timmermann, R., Iakovlev, N., Sidorenko, D., Kimmritz, M., Jung, T., and Schröter, J.: Finite-element sea ice model (FESIM), version 2, Geoscientific Model Development, 8, 1747–1761, https://doi.org/10.5194/gmd-8-1747-2015, 2015.

Dansereau, V.: A Maxwell-Elasto-Brittle model for the drift and deformation of sea ice, theses, Université Grenoble Alpes, https://tel.archives-ouvertes.fr/tel-01316987, 2016.

Dansereau, V., Weiss, J., Saramito, P., and Lattes, P.: A Maxwell elasto-brittle rheology for sea ice modelling, Cryosphere, 10, 1339–1359, https://doi.org/10.5194/tc-10-1339-2016, 2016.

Ebert, E. E., Schramm, J. L., and Curry, J. A.: Disposition of solar radiation in sea ice and the upper ocean, Journal of Geophysical Research: Oceans, 100, 15 965–15 975, https://doi.org/10.1029/95jc01672, publisher: American Geophysical Union (AGU), 1995.

Farrell, S. L., Duncan, K., Buckley, E. M., Richter-Menge, J., and Li, R.: Mapping Sea Ice Surface Topography in High Fidelity With ICESat-2, Geophysical Research Letters, 47, e2020GL090 708, https://doi.org/https://doi.org/10.1029/2020GL090708, e2020GL090708 2020GL090708, 2020.

Flocco, D., Feltham, D. L., and Turner, A. K.: Incorporation of a physically based melt pond scheme into the sea ice component of a climate model, Journal of Geophysical Research: Oceans, 115, 1–14, https://doi.org/10.1029/2009JC005568, 2010.

Flocco, D., Feltham, D. L., Bailey, E., and Schroeder, D.: The refreezing of melt ponds on Arctic sea ice, Journal of Geophysical Research: Oceans, 120, 647–659, https://doi.org/10.1002/2014jc010140, 2015.

Franklin, W. R.: PNPOLY - Point Inclusion in Polygon Test, https://wrfranklin.org/Research/Short_Notes/pnpoly.html, [Online; accessed 2-August-2024], 2023.

Geuzaine, C. and Remacle, J.-F.: Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering, 79, 1309–1331, https://doi.org/10.1002/nme.2579, 2009.

Girard, L., Bouillon, S., Weiss, J., Amitrano, D., Fichefet, T., and Legat, V.: A new modeling framework for sea-ice mechanics based on elasto-brittle rheology, Annals of Glaciology, 52, 123–132, https://doi.org/10.3189/172756411795931499, 2011.

Goessling, H. F., Tietsche, S., Day, J. J., Hawkins, E., and Jung, T.: Predictability of the Arctic sea ice edge, Geophysical Research Letters, 43, 1642–1650, https://doi.org/10.1002/2015GL067232, 2016.

Grachev, A. A., Andreas, E. L., Fairall, C. W., Guest, P. S., and Persson, P. O. G.: SHEBA flux–profile relationships in the stable atmospheric boundary layer, Boundary-Layer Meteorology, 124, 315–333, https://doi.org/10.1007/s10546-007-9177-6, 2007.

Hecht, F.: BAMG: bidimensional anisotropic mesh generator, 1998.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 global reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Hibler, W. D.: A dynamic thermodynamic sea ice model, Journal of Physical Oceanography, 9, 815–846, https://doi.org/10.1175/1520-0485(1979)009<0815:adtsim>2.0.co;2, 1979.

Hibler, W. D.: Modeling a Variable Thickness Sea Ice Cover, Monthly Weather Review, 108, 1943–1973, https://doi.org/10.1175/1520-0493(1980)108<1943:MAVTSI>2.0.CO;2, 1980.

Holland, M. M., Bailey, D. A., Briegleb, B. P., Light, B., and Hunke, E.: Improved sea ice shortwave radiation physics in CCSM4: The impact of melt ponds and aerosols on Arctic sea ice, Journal of Climate, 25, 1413–1430, https://doi.org/10.1175/jcli-d-11-00078.1, 2012.

Hunke, E. C.: Viscous–Plastic Sea Ice Dynamics with the EVP Model: Linearization Issues, Journal of Computational Physics, 170, 18–38, https://doi.org/10.1006/jcph.2001.6710, 2001.

Hunke, E. C. and Bitz, C. M.: Age characteristics in a multidecadal Arctic sea ice simulation, Journal of Geophysical Research, 114, https://doi.org/10.1029/2008jc005186, 2009.

Hunke, E. C. and Dukowicz, J. K.: An elastic–viscous–plastic model for sea ice dynamics, Journal of Physical Oceanography, 27, 1849–1867, https://doi.org/10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2, 1997.

Hunke, E. C. and Lipscomb, W. H.: CICE: the Los Alamos sea ice model documentation and software user's manual version 4.1, Tech. Rep. LA-CC-06-012, T-3 Fluid Dynamics Group, Los Alamos National Laboratory, 2010.

Hutter, N., Bouchat, A., Dupont, F., Dukhovskoy, D., Koldunov, N., Lee, Y., Lemieux, J.-F., Lique, C., Losch, M., Maslowski, W., Myers, P. G., Ólason, E., Rampal, P., Rasmussen, T., Talandier, C., Tremblay, B., and Wang, Q.: Sea ice rheology experiment (SIREx), part II: Evaluating linear kinematic features in high-resolution sea-ice simulations, Journal of Geophysical Research: Oceans, https://doi.org/10.1029/2021jc017666, 2022.

Istomina, L., Niehaus, H., and Spreen, G.: Updated Arctic melt pond fraction dataset and trends 2002–2023 using ENVISAT and Sentinel-3 remote sensing data, The Cryosphere Discussions, pp. 1–32, https://doi.org/10.5194/tc-2023-142, publisher: Copernicus GmbH, 2023.

Itkin, P., Spreen, G., Hvidegaard, S. M., Skourup, H., Wilkinson, J., Gerland, S., and Granskog, M. A.: Contribution of deformation to sea ice mass balance: A case study from an N-ICE2015 storm, Geophysical Research Letters, 45, 789–796, https://doi.org/10.1002/2017gl076056, 2018.

Kader, B. and Yaglom, A.: Mean fields and fluctuation moments in unstably stratified turbulent boundary layers, Journal of Fluid Mechanics, 212, 637–662, 1990.

Korosov, A., Rampal, P., Ying, Y., Ólason, E., and Williams, T.: Towards improving short-term sea ice predictability using deformation observations, The Cryosphere, 17, 4223–4240, https://doi.org/10.5194/tc-17-4223-2023, 2023.

Korosov, A., Ying, Y., and Ólason, E.: Tuning parameters of a sea ice model using machine learning, Geoscientific Model Development, 18, 885–904, https://doi.org/10.5194/gmd-18-885-2025, 2025.

Lapere, R., Marelle, L., Rampal, P., Brodeau, L., Melsheimer, C., Spreen, G., and Thomas, J. L.: Modeling the contribution of leads to sea spray aerosol in the high Arctic, Atmospheric Chemistry and Physics, 24, 12 107–12 132, https://doi.org/10.5194/acp-24-12107-2024, 2024.

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, f01022, 2012.

Lavergne, T. and Down, E.: A climate data record of year-round global sea-ice drift from the EUMETSAT Ocean and Sea Ice Satellite Application Facility (OSI SAF), Earth System Science Data, 15, 5807–5834, https://doi.org/10.5194/essd-15-5807-2023, publisher: Copernicus GmbH, 2023.

Lavergne, T., Sørensen, A. M., Kern, S., Tonboe, R., Notz, D., Aaboe, S., Bell, L., Dybkjær, G., Eastwood, S., Gabarro, C., Heygster, G., Killie, M. A., Brandt Kreiner, M., Lavelle, J., Saldo, R., Sandven, S., and Pedersen, L. T.: Version 2 of the EUMETSAT OSI SAF and ESA CCI sea-ice concentration climate data records, The Cryosphere, 13, 49–78, https://doi.org/10.5194/tc-13-49-2019, 2019.

Lemieux, J.-F., Knoll, D. A., Tremblay, B., Holland, D. M., and Losch, M.: A comparison of the Jacobian-free Newton–Krylov method and the EVP model for solving the sea ice momentum equation with a viscous-plastic formulation: A serial algorithm study, Journal of Computational Physics, 231, 5926–5944, https://doi.org/10.1016/j.jcp.2012.05.024, 2012.

Lemieux, J.-F., Tremblay, L. B., Dupont, F., Plante, M., Smith, G. C., and Dumont, D.: A basal stress parameterization for modeling landfast ice, Journal of Geophysical Research: Oceans, 120, 3157–3173, https://doi.org/10.1002/2014JC010678, 2015.

Leppäranta, M.: The drift of sea ice, Springer-Verlag GmbH, second edition edn., ISBN 978-3-642-04683-4, 2011.

Lyakhovsky, V., Ben-Zion, Y., and Agnon, A.: Distributed damage, faulting, and friction, Journal of Geophysical Research: Solid Earth, 102, 27 635–27 649, https://doi.org/10.1029/97jb01896, 1997.

Marsan, D., Stern, H., Lindsay, R., and Weiss, J.: Scale dependence and localization of the deformation of Arctic sea ice, Physical Review Letters, 93, 178 501, https://doi.org/10.1103/physrevlett.93.178501, 2004.

Mellor, G. L. and Kantha, L.: An ice-ocean coupled model, Journal of Geophysical Research, 94, 10 937–10 954, https://doi.org/10.1029/JC094iC08p10937, 1989.

Ólason, E., Rampal, P., and Dansereau, V.: On the statistical properties of sea-ice lead fraction and heat fluxes in the Arctic, The Cryosphere, 15, 1053–1064, https://doi.org/10.5194/tc-15-1053-2021, 2021.

Ólason, E., Boutin, G., Korosov, A., Rampal, P., Williams, T., Kimmritz, M., Dansereau, V., and Samaké, A.: A new brittle rheology and numerical framework for large-scale sea-ice models, Journal of Advances in Modeling Earth Systems, 14, https://doi.org/10.1029/2021ms002685, 2022.

Perovich, D. K., Grenfell, T. C., Richter-Menge, J. A., Light, B., Tucker, W. B., and Eicken, H.: Thin and thinner: Sea ice mass balance measurements during SHEBA, Journal of Geophysical Research: Oceans, 108, https://doi.org/10.1029/2001jc001079, 2003.

Plante, M., Tremblay, B., Losch, M., and Lemieux, J.-F.: Landfast sea ice material properties derived from ice bridge simulations using the Maxwell elasto-brittle rheology, The Cryosphere, 14, 2137–2157, https://doi.org/10.5194/tc-14-2137-2020, 2020.

Pohl, C., Istomina, L., Tietsche, S., Jäkel, E., Stapf, J., Spreen, G., and Heygster, G.: Broadband albedo of Arctic sea ice from MERIS optical data, The Cryosphere, 14, 165–182, https://doi.org/10.5194/tc-14-165-2020, publisher: Copernicus GmbH, 2020.

Rampal, P., Weiss, J., Marsan, D., Lindsay, R., and Stern, H.: Scaling properties of sea ice deformation from buoy dispersion analysis, Journal of Geophysical Research: Oceans, 113, https://doi.org/10.1029/2007JC004143, c03002, 2008.

Rampal, P., Bouillon, S., Bergh, J., and Ólason, E.: Sea ice diffusion in the Arctic ice pack: a comparison between observed buoy trajectories and the neXtSIM and TOPAZ-CICE sea ice models, The Cryosphere, https://doi.org/10.5194/tc-2015-233, 2016a.

Rampal, P., Bouillon, S., Ólason, E., and Morlighem, M.: neXtSIM: a new Lagrangian sea ice model, The Cryosphere, 10, 1055–1073, https://doi.org/10.5194/tc-10-1055-2016, 2016b.

Rampal, P., Dansereau, V., Ólason, E., Bouillon, S., Williams, T., Korosov, A., and Samaké, A.: On the multi-fractal scaling properties of sea ice deformation, The Cryosphere, 13, 2457–2474, https://doi.org/10.5194/tc-13-2457-2019, 2019.

Rao, G. H. S.: Finite element methods vs. Classical methods, New Age International (P) Ltd., Publishers, New Delhi, 1st ed. edn., ISBN 9788122420500, description based on publisher supplied metadata and other sources., 2007.

Regan, H., Rampal, P., Ólason, E., Boutin, G., and Korosov, A.: Modelling the evolution of Arctic multiyear sea ice over 2000–2018, The Cryosphere, 17, 1873–1893, https://doi.org/10.5194/tc-17-1873-2023, 2023.

Regan, H. C., Rheinlaender, J. W., Boutin, G., Semper, S., Willmes, S., and Olason, E.: Rapid winter mixed-layer deepening linked to sea ice breakup in the Beaufort Sea, ESS Open Archive, https://doi.org/10.22541/essoar.175095697.77260616/v1, 2025.

Rheinlænder, J. W., Regan, H., Rampal, P., Boutin, G., Ólason, E., and Davy, R.: Breaking the ice: Exploring the changing dynamics of winter breakup events in the Beaufort Sea, Journal of Geophysical Research: Oceans, 129, https://doi.org/10.1029/2023jc020395, 2024.

Ricker, R., Hendricks, S., Kaleschke, L., Tian-Kunze, X., King, J., and Haas, C.: A weekly Arctic sea-ice thickness data record from merged CryoSat-2 and SMOS satellite data, The Cryosphere, 11, 1607–1623, https://doi.org/10.5194/tc-11-1607-2017, 2017.

Rigby, F. A. and Hanson, A.: Evolution of a large Arctic pressure ridge, AIDJEX bulletin, 34, 43–71, 1976.

Rousset, C., Vancoppenolle, M., Madec, G., Fichefet, T., Flavoni, S., Barthélemy, A., Benshila, R., Chanut, J., Levy, C., Masson, S., and Vivier, F.: The Louvain-La-Neuve sea ice model LIM3.6: global and regional capabilities, Geoscientific Model Development, 8, 2991–3005, https://doi.org/10.5194/gmd-8-2991-2015, 2015.

Sakov, P., Counillon, F., Bertino, L., Lister, K. A., Oke, P. R., and Korablev, A.: TOPAZ4: An ocean sea ice data assimilation system for the North Atlantic and Arctic, Ocean Science, 8, 633–656, https://doi.org/10.5194/os-8-633-2012, 2012.

Salganik, E., Lange, B. A., Katlein, C., Matero, I., Anhaus, P., Muilwijk, M., Høyland, K. V., and Granskog, M. A.: Observations of preferential summer melt of Arctic sea-ice ridge keels from repeated multibeam sonar surveys, The Cryosphere, 17, 4873–4887, https://doi.org/10.5194/tc-17-4873-2023, 2023.

Samaké, A., Rampal, P., Bouillon, S., and Ólason, E.: Parallel implementation of a Lagrangian-based model on an adaptive mesh in C++: Application to sea-ice, Journal of Computational Physics, 350, 84–96, https://doi.org/10.1016/j.jcp.2017.08.055, 2017.

Schapery, R. A.: Nonlinear viscoelastic and viscoplastic constitutive equations with growing damage, International Journal of Fracture, 97, 33–66, https://doi.org/10.1023/a:1018695329398, 1999.

Semtner, A.: On modeling the seasonal thermodynamic cycle of sea ice in studies of climatic-change, Climatic change, 6, 27–37, https://doi.org/10.1007/bf00141666, 1984.

Semtner, A. J.: A model for the thermodynamic growth of sea ice in numerical investigations of climate, Journal of Physical Oceanography, 6, 379–389, https://doi.org/10.1175/1520-0485(1976)006<0379:AMFTTG>2.0.CO;2, 1976.

Skyllingstad, E. D., Paulson, C. A., Pegau, W. S., McPhee, M. G., and Stanton, T.: Effects of keels on ice bottom turbulence exchange, Journal of Geophysical Research: Oceans, 108, https://doi.org/10.1029/2002jc001488, 2003.

Smith, M. M., Fuchs, N., Salganik, E., Perovich, D. K., Raphael, I., Granskog, M. A., Schulz, K., Shupe, M. D., and Webster, M.: Formation and fate of freshwater on an ice floe in the Central Arctic, https://doi.org/10.5194/egusphere-2024-1977, 2024.

Stull, R. B.: An introduction to boundary layer meteorology, vol. 13, Springer Science & Business Media, 2012.

Sulsky, D., Schreyer, H., Peterson, K., Kwok, R., and Coon, M.: Using the material-point method to model sea ice dynamics, Journal of Geophysical Research: Oceans, 112, https://doi.org/10.1029/2005JC003329, 2007.

Tsamados, M., Feltham, D. L., Schroeder, D., Flocco, D., Farrell, S. L., Kurtz, N., Laxon, S. W., and Bacon, S.: Impact of variable atmospheric and oceanic form drag on simulations of Arctic sea ice, Journal of Physical Oceanography, 44, 1329–1353, https://doi.org/10.1175/jpo-d-13-0215.1, 2014.

Turner, A. K. and Hunke, E. C.: Impacts of a mushy-layer thermodynamic approach in global sea-ice simulations using the CICE sea-ice model, Journal of Geophysical Research: Oceans, 120, 1253–1275, https://doi.org/10.1002/2014jc010358, 2015.

Watts, M., Maslowski, W., Lee, Y. J., Kinney, J. C., and Osinski, R.: A spatial evaluation of Arctic sea ice and regional limitations in CMIP6 historical simulations, Journal of Climate, pp. 1–54, https://doi.org/10.1175/jcli-d-20-0491.1, 2021.

Webster, M. A., Holland, M., Wright, N. C., Hendricks, S., Hutter, N., Itkin, P., Light, B., Linhardt, F., Perovich, D. K., Raphael, I. A., Smith, M. M., von Albedyll, L., and Zhang, J.: Spatiotemporal evolution of melt ponds on Arctic sea ice: MOSAiC observations and model results, Elementa: Science of the Anthropocene, 10, 000 072, https://doi.org/10.1525/elementa.2021.000072, 2022.

Williams, T., Korosov, A., Rampal, P., and Ólason, E.: Presentation and evaluation of the Arctic sea ice forecasting system neXtSIM-F, The Cryosphere, 15, 3207–3227, https://doi.org/10.5194/tc-15-3207-2021, 2021.

Winton, M.: A reformulated three-layer sea ice model, Journal of Atmospheric and Oceanic Technology, 17, 525–531, https://doi.org/10.1175/1520-0426(2000)017<0525:ARTLSI>2.0.CO;2, 2000.

Zhang, J. and Rothrock, D. A.: Modeling global sea ice with a thickness and enthalpy distribution model in generalized curvilinear coordinates, Monthly Weather Review, 131, 681–697, https://doi.org/10.1175/1520-0493(2003)131<0845:mgsiwa>2.0.co;2, 2003.