Efficient numerical methods to solve the viscous-plastic sea ice model at high spatial resolutions

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von M.Sc. Carolin Mehlmann

geb. am 16.06.1989 in Worms-Herrnsheim

genehmigt durch die Fakultät für Mathematik

der Otto-von-Guericke Universität Magdeburg

Gutachter: Prof. Dr. Thomas Richter
Prof. Dr. Malte Braack

eingereicht am 22.03.2019

Verteidigung am 29.05.2019

Abstract

In this thesis, we develop efficient numerical methods to solve the *viscous-plastic sea ice model* on high resolution grids, with a cell size up to 2 km. This model describes the *dynamical* and *thermodynamical* large-scale processes in sea ice and plays an important role in climate models. The sea ice component in climate models accounts for more than 20% of the overall computational effort. Thus, the development of efficient numerical methods is topic of current research.

Sea ice dynamics are modeled by a system of equations coupling a nonlinear momentum equation, and transport processes. Currently, existing methods are based on implicit discretizations of the nonlinear momentum equation and converge either poorly or not at all on high resolution grids. Within a *finite element framework*, we present a new efficient Newton solver, globalized with a *line search method* and accelerated, with respect to convergence, by the *operator-related damped Jacobian method*. Using this novel approach we significantly improve the robustness of currently applied Newton solvers. We proove that the Jacobian of the sea ice model is *positive definite*, which provides global convergence of the Newton scheme, assuming an optimal damping parameter.

As the used linear solver in the *Jacobian-free Newton-Krylov* approach is extremely costly, mainly due to the absence of efficient preconditioners, we introduce the *geometric multigrid* method as preconditioner to the linear solver. Analyzing an idealized test case on a 2 km grid, we find that the *multigrid preconditioner* is able to reduce the iteration count of the linear solver by up to 80% compared to an *incomplete lower upper factorization* as preconditioner. As the convergence rate of the *multigrid method* is robust with increasing mesh resolutions, it is a suitable method for sea ice simulations at high spatial resolutions.

In the final part of the thesis, we develop a *goal oriented error estimator for partitioned solution approaches*, which is applicable to the sea ice model. The error estimator is based on the *dual weighted residual method* and derived for a general class of non-stationary differential equations coupled to a transport process. We observe a highly accurate error estimator for a system consisting of Burgers equation and a transport process. While the error estimator is accurate on simple sea ice configurations, efficiency worsens once stronger spatial structure, such as leads, in the ice appear. Finally, we develop a *mesh refinement strategy* that is based on the error estimator and evaluate it for the sea ice model. Applying adaptive meshes, we reach the same accuracy of a functional of intrest using 9 times less nodes.

Zusammenfassung

In der vorliegenden Arbeit entwickeln wir effiziente numerische Methoden zur Lösung des *visko-plastischen Meereismodells* auf hochaufgelösten Gittern, mit einer Gitterweite von bis zu 2 km. Das Meereismodell beschreibt die *Dynamik* und *Thermodynamik* des Meereises und ist ein wichtiger Bestandteil in Klimamodellen. Die Simulation der Meereisdynamik kann in gekoppelten Klimamodellen mehr als 20% des Gesamtaufwandes in Anspruch nehmen, weshalb die Entwicklung effizienter Methoden zur Simulation der Meereisdynamik Teil der gegenwärtigen Forschung ist.

Die Meereisdynamik wird durch ein System bestehend aus einer *nichtlinearen Momentengleichung* und *Transportgleichungen* modelliert. Derzeitige Methoden zur Lösung der implizit diskretisierten Momentengleichung des Meereismodells konvergieren entweder langsam oder gar nicht mit zunehmender örtlicher Auflösung. Im Rahmen eines *Finite-Elemente-Ansatzes* präsentieren wir ein neues Newton-Verfahren, globalisiert mit dem *Liniensuchverfahren* und beschleunigt hinsichtlich der Konvergenz durch die *operator-related damped Jacobian* Methode. Die neue Methode erhöht die Robustheit des derzeitig verwendeten Newton-Lösers signifikant. Wir beweisen, dass die Jacobi-Matrix des Meereismodells *positiv definit* ist, weshalb das Newton-Verfahren unter der Verwendung eines optimalen Dämpfungsparameters global konvergiert.

Die gegenwärtig verwendeten linearen Löser im Newton-Verfahren sind numerisch sehr teuer aufgrund eines fehlenden effizienten Vorkonditionierers. Wir führen das *geometrische Mehrgitterverfahren* als Vorkonditionierer des linearen Lösers ein. In einem idealisierten Test zeigen wir, dass der *Mehrgitter-Vorkonditionierer* im Vergleich zu einem *ILU-Vorkonditionierer* die Anzahl der Iterationen des linearen Lösers um 80% reduziert. Das *Mehrgitterverfahren* ist robust hinsichtlich örtlicher Gitterverfeinerung, daher ist es eine geeignete Methode zur Meereissimulation auf hochaufgelösten Gittern.

Im letzten Teil der Doktorarbeit, entwickeln wir einen *zielorientierten Fehlerschätzer* für *partitionierte Lösungsansätze*, der für das Meereismodell anwendbar ist. Der Fehlerschätzer basiert auf dem *dual gewichteten Fehlerschätzer* und ist für ein gekoppeltes System, bestehend aus instationären partiellen Differentialgleichungen und einem Transportprozess, hergeleitet. Am Beispiel der Burgersgleichung gekoppelt an einen Transportprozess, zeigen wir, dass der Fehlerschätzer akkurat ist. Im Fall des Meereismodells ist der Fehlerschätzer akkurat für einfache Konfigurationen des Modells. Die Genauigkeit des Fehlerschätzers nimmt ab, sobald sich starke Strukturen, beispielsweise Risse im Meereis, entwickeln. Abschließend leiten wir, basierend auf dem Fehlerschätzer, einen Gitterverfeinerungsalgorithmus her und werten diesen für das Meereismodell aus. Unter der Verwendung von adaptiven Gittern können wir die gleiche Genauigkeit eines Zielfunktionals mit neun mal weniger Knoten erreichen.

Contents

1. Introduction

Since the first satellite measurements have been recorded in 1979, we observe a steadily decreasing sea ice extent in the Arctic [91]. During a year, the maximal sea ice extent is observed in March, where the ice extents to approximately 15×10^6 km^2 [32]. The minimal sea ice extent is reached in September, at the end of the summer season. Based on time series from 1979 to 2018, the lowest record was reached in September 2012, when sea ice extent reduced to 3.41×10^6 km^2 [7]. The September average sea ice extent decreases by nearly 14% per decade based on a regression through the entire time series. The mean March sea ice extent reduces by 2.4% per decade from 1979 to 1999 and 3.4% per decade from 2000 onwards [92]. Following Stroeve and Notz [92], the decreasing sea ice extent in the Arctic is one of the most striking changes in the climate.

The role of sea ice in the climate system Sea ice plays an important role in the Earth's climate. The climate system consists of the atmosphere, the oceans, the biosphere and the cryosphere. As described by Kreyscher et al. [45], sea ice like glaciers belong to the cryosphere, which contains all types of water in a frozen stadium. Sea ice is a thin layer between the atmosphere and the ocean. It covers up to 10% of the world ocean [32]. Snow covered sea ice reflects a significant amount of solar radiation, while open water absorbs most of it [32]. A shrinking ice layer will reduce the reflectivity and thus, reinforce the reduction, the so-called *ice-albedo feedback* [32, 77] Thus, sea ice is very sensitive to global warming and plays a critical role in the climate system. As described by Hunke et al. [32], sea ice acts as an insulator between the atmosphere and the ocean. In the Arctic winter, sea ice isolates the cold atmosphere from the warmer ocean water. In the Arctic summer, sea ice reflects the solar radiation. Further, the extent of sea ice and the amount of open water as well as the thickness of the ice layer have a large impact on the heat and moisture exchange between the atmosphere and the ocean [54].

Sea ice models Modern sea ice models include *thermodynamical* and *dynamical processes*. In this thesis, we focus on the dynamical part of the sea ice models. Thus, we give an overview of development of the dynamical core in sea ice models in the last 40 years.

In the 1970s, the observations of the Arctic Ice Dynamics Joint Experiment (AIDJEX) greatly improved the modeling of the sea ice dynamics [32]. Based on the data of the AIDJEX, Coon et al. [13] developed an *elastic-plastic sea ice model*. Here, sea ice is assumed to be elastic material up to a certain internal stress level. Above this stress level, sea ice is modeled as a plastic material. In the plastic regime, the ice deforms irreversibly, whereas in the elastic regime it returns to the principal shape, if the stress

is removed [78]. Following Coon et al. [13], the elasticity assumption is numerically expensive to realize. Therefore in 1979, Hibler [25] replaced the elasticity by a viscosity assumption. He modeled sea ice as a viscous material at very low internal stresses.

The *viscous-plastic sea ice model* introduced by Hibler [25] reduces the numerical costs of a simulation and is easier to couple with atmosphere and ocean models [78]. The model consists of a nonlinear *momentum equation* and two *transport equations*. The nonlinearity of the momentum equation stems from the viscous-plastic material law. Initially, the model was developed for meshes in range of 100 kilometers. To solve the momentum equation at this spatial resolution, explicit time stepping schemes would require a time step of less than 1 second for simulation times ranging from months to years [34]. Thus, it is recommended by Ip et al. [34] to discretize the momentum equation with an implicit time stepping scheme. Due to the nonlinear character of the momentum equation, implicit solvers are needed to solve the viscous-plastic sea ice model in each time step.

Nowadays, with increasing mesh resolutions of up to 10 km [53], the performance of these solvers are getting poorer and this leads to an extreme increase of the numerical cost [32]. Therefore, Hunke and Dukowicz [31] proposed a commonly used variant to solve the viscous-plastic (VP) sea ice model, the *elastic-viscous-plastic* (EVP) model. The authors added an artificial elastic term to the VP material law (rheology), which allows an explicit discretization at relatively large time steps at high resolutions. The additional elastic term produces noticeable errors in the sea ice dynamics compared to an implicit discretization of the VP model [8, 54, 63]. Kimmritz et al. [38] showed that for a simplified model the EVP solution converges to the VP solution, if a sufficiently large number of sub-cycling steps are taken. Convergence of models can be accelerated by an adaptive control of the sub-cycling [39]. Nowadays, nearly all modern sea ice models are based on a viscous-plastic material assumption and include the EVP formulation, e.g., the *CICE, LIM3, MITgcm* [1, 29, 103].

It has been shown by Kwok et al. [49] that many sea ice simulations significantly differ from satellite observations. How much of this discrepancy must be attributed to a *modeling error* and how much to the *numerical approximation error* remains an open question [55]. In 2009, Girard et al. [59] pointed out that the representation of ice drift, the deformation statistics and the scaling of the VP and the EVP model differ from satellite observations. In 2011, Girard et al. [21] developed the *elasto-brittle* material law. Based on that rheology, Bouillon and Rampal [78] formulated a new dynamical sea ice model. Spreen et al. [90] could not reproduce the findings of Girard and his co-authors. In Spreen et al. [90], the best fit of simulations of the VP model to satellite observation is achieved with an increasing resolution. Here, the study was obtained with spatial resolutions up to 4.5 km. It is subject to future research to identify which sea ice rheology is the most appropriate.

Sea ice simulations at high spatial resolutions As the dynamical core of most sea ice models is based on the viscous-plastic sea ice model developed by Hibler 1979 in [56], we numerically analyze this model on high spatial resolutions. In this thesis, we focus on

efficiently simulating sea ice dynamics at 10 km resolutions and under. This includes the development of efficient and accurate numerical methods as well as the use of adaptive spatial meshes to increase the efficiency of a simulation.

We begin with outlining the importance of short-term forecasts on high resolutions, before we present the numerical challenges and the novel numerical methods to simulate sea ice dynamics at high spatial resolutions.

Following Williams and Tremblay [108] there is an increasing interest in estimating the sea ice extent in the Arctic, in particular in forecasting the minimum sea ice extent. In 2012, Massonnet et al. [67] predicted an ice-free Arctic summer between 2041 and 2060, based on a time series from 1979 to 2010. As described by Losch et al. [73], the strong decrease of the sea ice extent in the Arctic leads to an increase of economic activities such as tourist cruises and maritime transport. Not only the ice extent is decreasing, the sea ice is getting thinner and the amount of multi-year ice is reducing [100]. Ships can travel through *leads ("big cracks")* and first year ice more easily, as it is not as "stiff" as multi-year ice. Due to the increased maritime activities, short-term forecasts of sea ice deformations on high spatial resolutions become more and more important [73].

In the following, we address the topical numerical challenges in simulating sea ice dynamics at high resolutions.

- Currently, most applied methods to solve the viscous-plastic (VP) sea ice model converge either poorly, as the EVP model and the *Picard solver* (a fixed-point method), or not at all with increasing spatial resolutions, as the *Jacobian-free Newton-Krylov* (JFNK) solver [55, 56]. In practice, usually only few iterations of the Picard solver or small number of sub-cycles in the EVP model are considered such that the computed numerical solution contains numerical errors that accumulate over time [53]. On the other hand, the convergence of the JFNK solver suffers if the Jacobian is ill-conditioned or in the absence of good initial conditions. To improve the robustness of the Newton solver, Lemieux et al. [53] introduced a *line search method* and an *inexact solving of the linear subproblem* as accelerating and globalization techniques. The linear problem is solved with a *generalized minimal residual method* (GMRES) preconditioned with a *line successive over-relaxation method* (LSOR) [64]. Solving the linear problems is numerically extremely costly due to the absence of an efficient preconditioner.

- As outlined above, there have been some numerical studies on the convergence of sea ice solvers, e.g. [38, 39, 54, 55, 56, 64]. Besides that, there exists one study that analyzes the temporal convergence of the numerical solution [53], and another study that investigates the spatial convergence [107]. So far, no analysis includes both aspects at the same time. Lemieux et al. [53] analyzed the convergence of the sea ice model in time. The authors applied an approximation of a *monotholitic solution approach* (IMEX) to solve the sea ice model and compared it to the standard *partitioned solution approach* (SIT). They found no superior convergence behavior of the IMEX scheme when a first order discretization in time was applied. A convergence study in space for a one-dimensional reduction of the VP model was

done in [107]. They showed that the simulated velocity field depends on the spatial resolution and observed that the mean sea ice drift speed increases by 32% by increasing resolution from 40 km to 5 km.

- So far, unstructured grids for a better representation of irregular coastal geometries such as the Canadian Arctic Archipelago have been applied in some sea ice models, e.g., in *FESOM, CICE, SLIM* [15, 20, 61]. However, unstructured grids have not been used yet in the context of *adaptive mesh refinement strategies*. Adaptive mesh refinement strategies aim to reduce the overall numerical error by refining the grid in a region where the largest numerical error is measured. In feedback loops, the mesh refinement strategies are usually based on the evaluation of an error estimator [6]. So far, error estimators have not been used in the context of large-scale sea ice models.

- Finally, we want to emphasize that the question of existence and uniqueness of a solution for the viscous-plastic sea ice model, has not been answered. Some effort has been made to analyze existence and uniqueness for simplified or modified models [42, 89]. However, in this thesis we do not investigate this topic.

Novel numerical methods to simulate sea ice dynamics at high spatial resolutions
This thesis provides the following novel numerical methods to simulate sea ice dynamics based on the viscous-plastic rheology on high resolutions:

- To solve the nonlinear momentum equation, we introduce a Newton solver accelerated with the *operator-related damped* Jacobian method introduced by Horn, Ouazzi and Turek [14]. We prove that the Jacobian of viscous-plastic sea ice model is *positive definite* such that global convergence is provided by theory for an optimal damping parameter. Furthermore, we show that the Jacobian can be divided into a *positive semidefinite* and a *negative semidefinite* part. The idea of the *operator-related Jacobian* method is to adaptively control the negative semidefinite part of the Jacobian. We compare the new solver to the existing *Jacobian-free Newton-Krylov* (JFNK) method and observe an increased robustness and faster convergence with the *operator-related Jacobian method* on an idealized test case and on a realistic simulation on a pan-Arctic grid.

- We establish the *multigrid method* as a preconditioner to the GMRES method, which is applied to solve the linear problems arising in each Newton iteration. On an idealized test case, we show that the multigrid preconditioner needs about 80% less iteration counts than an *incomplete lower upper factorization* (ILU) as preconditioner to the GMRES method. The convergence of multigrid preconditioner is robust with increasing resolutions, as the iteration count is only slightly increasing.

- We develop a *goal oriented error estimator* for *partitioned solution approaches*, which is applicable to the sea ice model. The a posteriori error estimator is based on the *dual weighted residual method* introduced by Becker and Rannacher [6].

The error estimator is derived for a general class of partial differential equations that are coupled to transport processes. It consists of three parts, the *temporal discretization error*, the *spatial discretization error* and the *splitting error*, which is introduced by using a partitioned solution approach. We apply this error estimator to the sea ice model to quantify the error contribution due to spatial and temporal discretization. Further, we analyze the effect of using a partitioned solution approach instead of a monotholitic scheme. On an idealized test cases, the error estimator shows high accuracy. Based on this estimator, we present an adaptive mesh refinement algorithm to reduce the numerical costs of sea ice simulations.

The structure of this thesis The thesis can be divided into four parts:

I **The viscous-plastic sea ice model**

II **Discretization of the sea ice model**

III **An efficient Newton solver for the viscous-plastic sea ice model**

IV **A goal oriented error estimator for partitioned solution approaches**

(I) **The viscous-plastic sea ice model** of Hibler [25] is introduced in **Chapter 2**. Here, sea ice dynamics are modeled in a two dimensional framework. The averaged properties of sea ice are described by the *ice concentration*, the *mean ice thickness* and the *horizontal ice velocity*. The model consists of a momentum equation that includes a nonlinear material law, the *viscous-plastic rheology*. The momentum equation is coupled to transport equations, which advect the ice concentration and ice thickness in time. Sea ice dynamics are driven by an atmospheric and an oceanic forcing. We present the complete systems of equations in a *variational formulation* and a *non-dimensional framework*. Finally, we introduce a model problem which is used for the numerical evaluation of methods introduced in this thesis.

(II) **Discretization of the sea ice model** is presented in **Chapter 3**. The sea ice model is implemented in the Software library Gascoigne 3D which is based on a quadrilateral mesh. We introduce a *partitioned solution approach* to solve the sea ice model, i.e., we decouple the system in time and compute the numerical solution of the momentum equation and the transport equations in each time step separately. An *implicit Euler* scheme is used to discretize the momentum in time, where as the transport equations are discretized with an *explicit flux-limited Taylor-Galerkin* scheme. For the spatial discretization of the system we apply a *finite element* framework. As a direct finite element discretization of transport equations gives rise to instabilities in space, we analyze in **Chapter 4** the effect of different spatial stabilizations on the numerical solution. First, we evaluate a body rotation problem, then we transfer the result to the sea ice model.

(III) **An efficient Newton solver for the viscous-plastic sea ice model**, namely the *operator-related damped Jacobian* method is presented in **Chapter 5**. We prove that the Jacobian of the sea ice model is *positive definite*. Based on the analysis of the

Jacobian, we introduce the *operator-related damped Jacobian* method, which is an acceleration method for the Newton solver and compare it to the currently used acceleration method in the *Jacobian-free Newton-Krylov* (JFNK) solver. To solve the linear subproblems arising in each Newton iteration, we introduce a multigrid preconditioner to the GMRES method and compare its performance to the behavior of a GMRES method preconditioned with an ILU method. In **Chapter 6**, we analyze the performance of the *operator-related damped Jacobian* method in a realistic setting on a pan-Arctic grid. We implement the method into the software framework of the McGill University and compare its performance to the performance of the already implemented JFNK solver in partitioned and monotholitic solving approach on a 10 km mesh, which is the highest available resolution in this model.

(IV) **A goal oriented error estimator for partitioned solution approaches** is derived in **Chapter 7**. The error estimator is based on *the dual weighted residual method* which is presented by Becker and Rannacher [6]. We introduce an approximation of the error estimator as it includes non-computable quantities, e.g., interpolation errors. Hereby, we follow the work of Becker and Rannacher [6], Schmich and Vexler [88] and Richter and Meidner [72]. Finally, we introduce an adaptive refinement strategy which balances the different contributions of the error estimator. In **Chapter 8**, we evaluate the error estimator. First, on a simplified example, namely Burgers equation coupled to a transport process, and then we analyze it on the sea ice model. In order to apply the error estimator, we introduce a $dG(0)G(0)$ discretization in time for both examples. In case of the sea ice model, we evaluate an adaptive refinement strategy and discuss the possible benefits of adaptive computations.

Published results We want to outline that some of our results have already been published by Mehlmann and Richter in [69] and [70]. This is especially true for Chapter 5 and Section 2.6.

Part I.

The viscous-plastic sea ice model

2. The viscous-plastic sea ice model

In this chapter, we describe the *viscous-plastic sea ice model* which was introduced by Hibler in 1979 [25] to simulate sea ice dynamics. Sea ice dynamics are mainly modeled by a *momentum equation* (Section 2.2), a *rheology* (Section 2.4), and the *balance laws* (Section 2.3). The rheology is a constitutive law that describes the material behavior of the sea ice and its mechanical deformation. Sea ice dynamics are driven by the wind and the ocean current. The transportation of ice is modeled by the balance laws.

In Section 2.1, we present the assumptions of Hibler's *viscous-plastic sea ice model*. This is followed by the derivation of the *sea ice momentum equation* in Section 2.2 and the presentation of the *balance laws* in Section 2.3. In Section 2.4, we derive the *viscous-plastic sea ice rheology*. In Section 2.5, we present the complete system of equations and its *variational formulation*. Finally, we define a model problem for the sea ice model in Section 2.6. The latter is used to analyze the numerical methods in Chapter 4, Chapter 5 and Chapter 8.

2.1. Model variables

In this Section, we follow the derivation of Harder in [24]. To model sea ice in the climate system, it is reasonable to choose a large-scale approach that describes the average properties of sea ice. For example, the impact that a group of ice floes have on the interaction between the atmosphere and the ocean is considered in this large scale approach, whereas small-scale details like single ice floes are not taken into account [24].

We begin with introducing the first variable of the sea ice model, the *ice concentration* A. Following Hibler [25], the ice concentration A defines the averaged horizontal fraction of a control area A_r which is covered with ice. It is defined as

$$A = \frac{1}{|A_r|} \int_{A_r} I(x,y)\, dx\, dy, \quad \text{with} \quad I(x,y) = \begin{cases} 1, & \text{if a point is covered with ice} \\ 0, & \text{else.} \end{cases}$$

We assume that A_r is big enough to contain a meaningful number of small objects (ice floes), whose size is much smaller than the considered region. There are two thickness levels *thin* and *thick* ice that are described by the model. The fraction of a control volume covered with *thick* ice is given by A. The area of *thin* ice is given by $1 - A$.

Using the *continuums hypothesis*, we assume that all model variables are averaged variables and continuous differentiable functions in space and time [24]. For large-scale sea ice simulations, the continuum's hypothesis is a suitable approximation. In the original work of Hibler, the size of a grid cell is about $100\,\mathrm{km}^2$, whereas the size of an ice floe ranges between square meters and square kilometers. It needs to be investigated if

the continuum's hypothesis is suitable for simulations at high spatial resolutions up to 1 km.

The mean horizontal properties of sea ice are modeled by three variables, the *ice concentration* A which has already been introduced, the *mean ice thickness* H and the *horizontal ice velocity* $\mathbf{v}$. The *mean ice thickness* H describes the average ice volume per control area A_r. The horizontal ice velocity $\mathbf{v}$ is a two-dimensional vector, which models the sea ice velocity in x and y direction.

2.2. Sea ice momentum equation

Sea ice dynamics are taking place in a three-dimensional frame. As the vertical sea ice velocity is smaller than the horizontal ice velocity, the motion of sea ice in a large-scale model is described in two dimensions. To derive the two dimensional viscous-plastic sea ice model [25], we begin with presenting the three dimensional sea ice momentum equation.

$$\rho_{ice}(\partial_t \tilde{\mathbf{v}} + \tilde{\mathbf{v}} \cdot \nabla \tilde{\mathbf{v}}) = \operatorname{div} \tilde{\boldsymbol{\sigma}} + \tilde{F}_c + \tilde{F}_H, \tag{2.1}$$

with the sea ice density ρ_{ice}, the three-dimensional ice velocity $\tilde{\mathbf{v}}$ and three-dimensional symmetric stress tensor $\tilde{\boldsymbol{\sigma}}$. The Coriolis force is denoted by $\tilde{F}_c$ and the ocean tilt by $\tilde{F}_H$, which stems from the changing sea surface height. As the horizontal sea ice velocity dominates the vertical velocity, we integrate through the ice thickness H and obtain

$$\int_0^{H(x,y)} \rho_{ice}(\partial_t \tilde{\mathbf{v}} + \tilde{\mathbf{v}} \cdot \tilde{\mathbf{v}})\, dz = \int_0^{H(x,y)} \operatorname{div} \tilde{\boldsymbol{\sigma}} + \tilde{F}_c + \tilde{F}_H \, dz.$$

The surface boundary conditions τ_{atm} and $\tau_{\text{ocean}}(\mathbf{v})$ are included by integration through the stress tensor $\tilde{\boldsymbol{\sigma}}$. A detailed derivation of the stresses is given in the textbook of Leppäranta [58].

In two dimensions the motion of sea ice obeys

$$\rho_{ice} H(\partial_t \mathbf{v} + \mathbf{v} \cdot \mathbf{v}) = \operatorname{div} \boldsymbol{\sigma} + Fc + F_H + \tau_{atm} + \tau_{\text{ocean}}(\mathbf{v}). \tag{2.2}$$

We are referring to the horizontal sea ice velocity by $\mathbf{v}$ and to the stress tensor by $\boldsymbol{\sigma}$. Following Hibler and Zhang [111] the advection term $\mathbf{v} \cdot \nabla \mathbf{v}$ can be neglected due to scaling properties. The internal ice forces are modeled by $\operatorname{div} \boldsymbol{\sigma}$. All body forces acting on sea ice are included in F,

$$F = F_c + F_H + \tau_{atm} + \tau_{ocean}(\mathbf{v}),$$

where F_c is the Coriolis force, F_H the force to the surface height, τ_{atm} the atmospheric stress and the τ_{ocean} oceanic stress.

Particularly, the Coriolis force is given by

$$F_c = -\rho_{ice} H f \boldsymbol{e}_r \times \mathbf{v},$$

where $\boldsymbol{e}_r$ is the unit vector normal to the surface and f is the Coriolis parameter. The force arising from changing sea surface tilt is described by $F_H = \rho_{ice} H g \nabla H_s$, where g is the acceleration due to gravity and H_s describes the sea surface height. In order to express the surface height with the Coriolis term, Coon [12] introduced the following approximation

$$F_H \approx m f \boldsymbol{e}_r \times \mathbf{v}_{\text{ocean}},$$

where $\mathbf{v}_{\text{ocean}}$ is the ocean current. In [12] Coon modeled the effect of atmospheric and oceanic boundary layer on ice dynamics as

$$\begin{aligned} \tau_{atm} &= C_{\text{atm}} \rho_{\text{atm}} \|\mathbf{v}_{\text{atm}}\|_2 B_{\theta_{\text{atm}}}(\mathbf{v}_{\text{atm}}), \\ \tau_{\text{ocean}}(\mathbf{v}) &= C_{\text{ocean}} \rho_{\text{ocean}} \|\mathbf{v}_{\text{ocean}} - \mathbf{v}\|_2 B_{\theta_{\text{ocean}}}(\mathbf{v}_{\text{ocean}} - \mathbf{v}), \end{aligned}$$

where by $\|\cdot\|_2$ we denote the Euclidean norm. C_{atm} and C_{ocean} are the air and water drag coefficients, ρ_{atm} and ρ_{ocean} are the densities and $\mathbf{v}_{\text{atm}}$ describes the atmospheric flow. The air and water stress is turned at an angle θ_{atm} with the rotation matrix

$$B_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The model can be further simplified by assuming zero turning angles such that $B_\theta = I$ [25]. Expressing the terms in equation (2.2) explicitly, the momentum equation reads as

$$\rho_{\text{ice}} H \partial_t \mathbf{v} = \operatorname{div} \boldsymbol{\sigma} - \rho_{\text{ice}} H f \boldsymbol{e}_r \times (\mathbf{v} - \mathbf{v}_{\text{ocean}}) + \tau_{\text{atm}} + \tau_{\text{ocean}}(\mathbf{v}). \tag{2.3}$$

In this two-dimensional framework, sea ice is modeled as a compressible material. The ice thickness can be interpreted as a material behavior of ice and not as the third dimension [13].

2.3. Balance laws

In the following, we derive the balance laws, which advect the mean ice thickness H and the ice concentration A in time. Let $V(t)$ be a two-dimensional domain. The ice mass is given as

$$m_{ice} = \int_{V(t)} \rho_{ice} H.$$

Assuming only mechanical effects, the sea ice mass should be conserved

$$D_t m_{\text{ice}} = 0,$$

where the total derivative is given by $D_t = \partial + \mathbf{v} \cdot \nabla$. As the ice density ρ_{ice} is constant in time, the conservation of mass is equivalent to the conservation of the mean ice thickness

H [58]. Applying Reynolds transport theorem [80] and assuming continuity of H and $\mathbf{v}$, we get

$$\partial_t H + \operatorname{div}(H\mathbf{v}) = 0. \tag{2.4}$$

Following Kuzmin [46, Theorem 3.10], we find that equation (2.4) has a positive solution.

The balance law for ice concentration A is derived simultaneously.

$$\partial_t A + \operatorname{div}(A\mathbf{v}) = 0, \quad A \leq 1. \tag{2.5}$$

The restriction $A \leq 1$ is always fulfilled [25] by including thermodynamic growth S_H and S_A at the right hand side of equation (2.4) and (2.5). A description of the thermodynamics is given by Hibler in [25]. As this thesis focuses only on sea ice dynamics we neglect the thermodynamics in our analysis and we set $S_H = S_A = 0$.

2.4. Sea ice rheology

The sea ice motion described by equation (2.3) is driven by external forces F and internal forces $\operatorname{div}\boldsymbol{\sigma}$. We already discussed the external forces in Section 2.2. The internal forces, described in Section 2.4.1, are modeled by the stress tensor $\boldsymbol{\sigma}(\dot{\boldsymbol{\epsilon}})$, which is a function of the strain rates $\dot{\boldsymbol{\epsilon}}$. The strain rates, presented in Section 2.4.2, are part of the *kinematics*, while the stress tensor is part of the *dynamics*. The relation between kinematics and dynamics is given by the sea ice *rheology* and described in Section 2.4.3. The presentation of this Section follows Harder [24] and Leppäranta [58].

2.4.1. Internal forces

The simplest sea ice models consider free ice drift and neglect the internal forces, $\operatorname{div}\boldsymbol{\sigma} = 0$. In areas with compact ice cover, the free drift assumption is however not physical [24]. During the Arctic Ice Dynamics Joint Experiment in the 1970s, various measurements on sea ice dynamics haven taken place. Since then, the internal forces are included in sea ice models by $\operatorname{div}\boldsymbol{\sigma}$. The two-dimensional symmetrical stress tensor is given as

$$\boldsymbol{\sigma} = \begin{pmatrix} \boldsymbol{\sigma}_{11} & \boldsymbol{\sigma}_{12} \\ \boldsymbol{\sigma}_{12} & \boldsymbol{\sigma}_{22} \end{pmatrix}.$$

The stress tensor can be transformed to its principal components, the eigenvalues, by

$$\tilde{\boldsymbol{\sigma}} = \begin{pmatrix} \tilde{\boldsymbol{\sigma}}_1 & 0 \\ 0 & \tilde{\boldsymbol{\sigma}}_2 \end{pmatrix} = \mathbf{Q} \begin{pmatrix} \boldsymbol{\sigma}_{11} & \boldsymbol{\sigma}_{12} \\ \boldsymbol{\sigma}_{12} & \boldsymbol{\sigma}_{22} \end{pmatrix} \mathbf{Q}^T, \tag{2.6}$$

with the transformation matrix $\mathbf{Q}$ and angle θ.

$$\mathbf{Q} = \begin{pmatrix} \sin(\theta) & \cos(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}, \quad \theta = \frac{1}{2}\tan^{-1}\left(\frac{2\boldsymbol{\sigma}_{12}}{\boldsymbol{\sigma}_{11} - \boldsymbol{\sigma}_{22}}\right).$$

A detailed derivation can be found in [35] and [97]. Finally, we obtain the principal components as

$$\tilde{\boldsymbol{\sigma}}_1 = \frac{\boldsymbol{\sigma}_{11} + \boldsymbol{\sigma}_{22}}{2} + \sqrt{\Big(\frac{\boldsymbol{\sigma}_{11} - \boldsymbol{\sigma}_{22}}{2}\Big)^2 + \boldsymbol{\sigma}_{12}^2}, \tag{2.7}$$

$$\tilde{\boldsymbol{\sigma}}_2 = \frac{\boldsymbol{\sigma}_{11} + \boldsymbol{\sigma}_{22}}{2} - \sqrt{\Big(\frac{\boldsymbol{\sigma}_{11} - \boldsymbol{\sigma}_{22}}{2}\Big)^2 + \boldsymbol{\sigma}_{12}^2}. \tag{2.8}$$

The eigenvalues contain information about the frame independent properties of the material. To analyze the stress tensor, the following two invariants are introduced,

$$\begin{aligned} \boldsymbol{\sigma}_I &= \frac{1}{2}(\tilde{\boldsymbol{\sigma}}_1 + \tilde{\boldsymbol{\sigma}}_2) = \frac{1}{2}(\boldsymbol{\sigma}_{11} + \boldsymbol{\sigma}_{22}), \\ \boldsymbol{\sigma}_{II} &= \frac{1}{2}(-\tilde{\boldsymbol{\sigma}}_1 + \tilde{\boldsymbol{\sigma}}_2) = \sqrt{(\frac{\boldsymbol{\sigma}_{11} - \boldsymbol{\sigma}_{22}}{2})^2 + \boldsymbol{\sigma}_{12}^2} = \sqrt{-\det \boldsymbol{\sigma}'}, \end{aligned} \tag{2.9}$$

where $\boldsymbol{\sigma}_I$ is the average normal stress - also called volume stress - and $\boldsymbol{\sigma}_{II}$ describes the shear stress [84, 97]. A useful relation for analysis is the splitting of the stress tensor in its trace $\operatorname{tr}(\boldsymbol{\sigma})$ and the trace-free deviatoric part $\boldsymbol{\sigma}'$.

$$\boldsymbol{\sigma} = \frac{1}{2}\operatorname{tr}(\boldsymbol{\sigma}) + \boldsymbol{\sigma}'. \tag{2.10}$$

We apply this property when deriving the theoretical properties of the sea ice model in Section 5.1.

2.4.2. Kinematics

The theory of *kinematics* describes the movement of ice in space and time, relative to a frame system, without discussing the forces generating this movement. Among others, a good introduction to continuums mechanics can be found in the textbook of Richter [80]. The strain rate tensor $\dot{\boldsymbol{\epsilon}}$ is given as

$$\dot{\boldsymbol{\epsilon}} = \begin{pmatrix} \dot{\boldsymbol{\epsilon}}_{1,1} & \dot{\boldsymbol{\epsilon}}_{1,2} \\ \dot{\boldsymbol{\epsilon}}_{2,1} & \dot{\boldsymbol{\epsilon}}_{2,2} \end{pmatrix},$$

with $\dot{\boldsymbol{\epsilon}}_{i,j} = \frac{1}{2}\{\partial_{\mathbf{x}_j}\mathbf{v}_i + \partial_{\mathbf{x}_i}\mathbf{v}_j\}$ and $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$. As $\dot{\boldsymbol{\epsilon}}$ is symmetric, we transform the strain rate tensor into its principal axis, analogous to the transformation of the stress tensor (2.6),

$$\begin{aligned} \dot{\boldsymbol{\epsilon}}_I &= \dot{\epsilon}_1 + \dot{\epsilon}_2 = \dot{\boldsymbol{\epsilon}}_{1,1} + \dot{\boldsymbol{\epsilon}}_{2,2} = \operatorname{tr}(\dot{\boldsymbol{\epsilon}}) = \operatorname{div}\mathbf{v}, \\ \dot{\boldsymbol{\epsilon}}_{II} &= \dot{\epsilon}_1 - \dot{\epsilon}_2 = \sqrt{(\dot{\boldsymbol{\epsilon}}_{1,1} - \dot{\boldsymbol{\epsilon}}_{2,2})^2 - 4\dot{\boldsymbol{\epsilon}}_{1,2}^2} = 2\sqrt{-\det \dot{\boldsymbol{\epsilon}}'}. \end{aligned} \tag{2.11}$$

The divergence of the ice is described by $\dot{\boldsymbol{\epsilon}}_I$ and models a compression or expansion of the ice, as for a fixed point in time,

$$\int_V \operatorname{div}\mathbf{v}\, dx\, dy = \int_{\partial V} n \cdot \mathbf{v}.$$

The second strain invariant $\dot{\boldsymbol{\epsilon}}_{II}$ is the shear deformation. A rate of the total deformation is given by $|\dot{\boldsymbol{\epsilon}}| = \sqrt{\dot{\boldsymbol{\epsilon}}_I^2 + \dot{\boldsymbol{\epsilon}}_{II}^2}$ [24]. Another useful representation of strain tensor is the decomposition in its trace $\mathrm{tr}(\dot{\boldsymbol{\epsilon}})$, and trace-free deviatoric part $\dot{\boldsymbol{\epsilon}}'$,

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\,\mathrm{tr}(\dot{\boldsymbol{\epsilon}}). \tag{2.12}$$

This relationship is used by deriving the properties of the sea ice momentum equation in Section 5.1.

2.4.3. The viscous-plastic rheology

Following Hibler [25], we assume that ice does not have any elastic response. We use a Reiner-Rivlin fluid model to express the relation of $\boldsymbol{\sigma}$ and $\dot{\boldsymbol{\epsilon}}$

$$\boldsymbol{\sigma} = \alpha I + \beta\dot{\boldsymbol{\epsilon}} + \gamma\dot{\boldsymbol{\epsilon}}^2,$$

where α, β, γ depend on state variables and strain rate invariants [58]. Following Hibler [104], the γ term is neglected and a general *constitutive law* is assumed

$$\boldsymbol{\sigma} = \zeta\,\mathrm{tr}(\dot{\boldsymbol{\epsilon}})I + 2\eta\dot{\boldsymbol{\epsilon}}' - \frac{P}{2}, \tag{2.13}$$

where P is the *ice strength* that models the maximum stress due to compression. The relation of η to ζ is constant. Further, it holds that $\frac{\eta}{\zeta} > 1$. To achieve the viscous-plastic constitutive law, it is missing to determine the viscosities η and ζ in (2.13). In the *plastic state* the viscosities are modeled by a *normal flow rule*. The latter relates the strain rates to the stresses that are prescribed by a *yield curve*. In the *viscous regime* the viscosities are chosen as constant values. A detailed derivation of this relationship is given below.

The yield curve Many sea ice models assume that sea ice follows a plastic rheology [58]. To illustrate the plastic behavior of a material, we first analyze a one-dimensional case. In Figure 2.1, we present a one-dimensional stress model. Here, the *yield function* consists of two points. One stress state models compression ($-\boldsymbol{\sigma}^\star$) and the other stress state describes tension of ice ($\boldsymbol{\sigma}^\star$). There is no change in strain up to the critical stress. If the strain rates are not zero, the material is in a continual state of *yielding*. In two dimensions the *yield function* consists of a curve. Here, the ice is *yielding*, e.g., is in a plastic state if the stress states are lying on the curve. In the principal stress state the yield function is determined by

$$F(\boldsymbol{\sigma}_I, \boldsymbol{\sigma}_{II}) = 0, \tag{2.14}$$

where $\boldsymbol{\sigma}_I = \frac{1}{2}(\tilde{\boldsymbol{\sigma}}_1 + \tilde{\boldsymbol{\sigma}}_2)$ is the stress due to compression and tension, and $\boldsymbol{\sigma}_{II} = \frac{1}{2}(-\tilde{\boldsymbol{\sigma}}_1 + \tilde{\boldsymbol{\sigma}}_2)$ describes the shear stress.

Based on observations the yield curve must fulfill the following requirements [58]. Sea ice is isotropic, thus the yield curve should be symmetric throughout the axis $\tilde{\boldsymbol{\sigma}}_1 = \tilde{\boldsymbol{\sigma}}_2$.

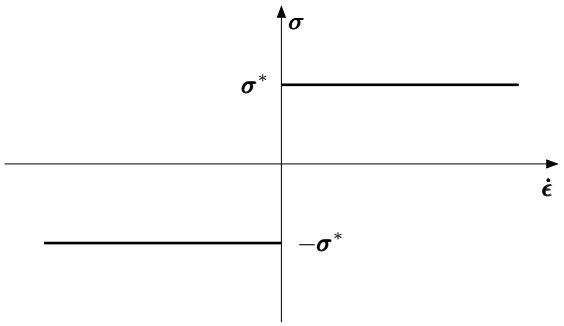


Figure 2.1.: One dimensional ideal plastic state.

Sea ice has almost no tensile *strength*, e.g, the yield curve should contain only a few combinations of $(\tilde{\boldsymbol{\sigma}}_1, \tilde{\boldsymbol{\sigma}}_2) > 0$. Generally spoken, the strength describes the limit at which sea ice is in a plastic flow. The compressive strength in sea ice is higher than the shear strength, thus the extension of the yield curve should be larger in the direction of $\boldsymbol{\sigma}_I$ than in the $\boldsymbol{\sigma}_{II}$-axis.

Hibler [25] assumed that the yield curve has the form of an ellipse with the eccentricity $e = 2$. The eccentricity is an empirical value and still a subject of dispute, as well as the whole form of the yield curve. The ellipse is chosen for mathematical convenience as it agrees with the observed behavior of sea ice. Some authors suggest using a Mohr-Coulomb, a teardrop or a modified Coulomb yield curve, [26, 58].

The compressive ice strength

$$P = P^{\star} H \exp(C(1 - A))$$

with constants $P^{\star}$ and C, determines the length of the major axis of the ellipse. The compressive strength decreases, when the ice thickness H and ice concentration A are decreasing. In this case, the ellipse gets smaller without changing the eccentricity and the ratio of the minor axis, compare the blue ellipse to the green ellipse in Figure 2.2. Considering Figure 2.2, pure shear is modeled in point S, where the gradient is parallel to the $\boldsymbol{\sigma}_I$-axis. The maximal compression with value P is given in point C. Pure divergence is received in D (in the origin of the graph) as $\boldsymbol{\sigma}_I = \boldsymbol{\sigma}_{II} = 0$.

Following Hibler [25], the elliptic yield curve is given by

$$F(\boldsymbol{\sigma}_I, \boldsymbol{\sigma}_{II}) = \left(\boldsymbol{\sigma}_I + \frac{P}{2}\right)^2 + \left(e\boldsymbol{\sigma}_{II}\right)^2 - \left(\frac{P}{2}\right)^2. \tag{2.15}$$

Normal flow rule To connect the strain rates to the stress, a *normal flow rule* is assumed, which can be motivated as follows: For simplicity, we write $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_I, \boldsymbol{\sigma}_{II})$. Using (2.14), we obtain $F(\boldsymbol{\sigma}) = 0$. In an ideal plastic state, a change in stress state $d\boldsymbol{\sigma}$ must generate a stress state which lies again on the yield curve, $F(\boldsymbol{\sigma} + d\boldsymbol{\sigma}) = 0$. We

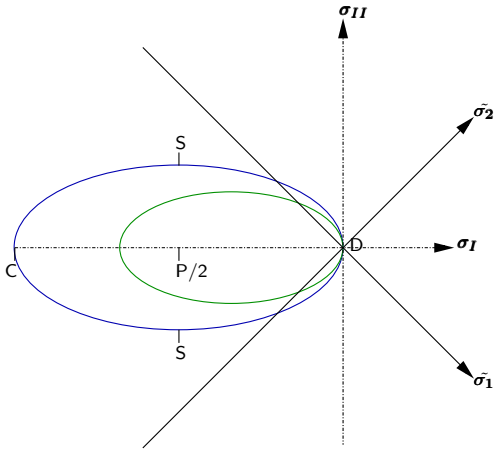


Figure 2.2.: Two-dimensional yield curve in principal stress space. The length of the major axis is determined by the compressive strength P. Pure divergence is obtained in D (in the origin of the graph), whereas pure convergence is modeled in C and pure shear is obtained in S.

expand this expression to

$$F(\boldsymbol{\sigma} + d\boldsymbol{\sigma}) = F(\boldsymbol{\sigma}) + \frac{\partial F}{\partial \boldsymbol{\sigma}} d\boldsymbol{\sigma} + O(|d\boldsymbol{\sigma}|^2).$$

and obtain, combining all properties,

$$\frac{\partial F}{\partial \boldsymbol{\sigma}} d\boldsymbol{\sigma} = 0.$$

By *Drucker's postulate* for stable inelastic materials, it holds that $d\boldsymbol{\sigma} d\dot{\boldsymbol{\epsilon}} \geq 0$ [58]. As we assume ideal plastic flow $d\boldsymbol{\sigma} = 0$ if $d\dot{\boldsymbol{\epsilon}} \geq 0$, thus we get $d\boldsymbol{\sigma} d\dot{\boldsymbol{\epsilon}} = 0$. Therefore, the normal $\frac{\partial F}{\partial \boldsymbol{\sigma}}$ to the yield surface and the strain are perpendicular to the surface. This relationship is shown in Figure 2.3. It is assumed that the stress does not depend on the level of strain rate, but on the direction of the strain rate. This property is given by the *normal flow rule*, which reads as

$$\dot{\boldsymbol{\epsilon}}_i = \gamma \partial_i F(\boldsymbol{\sigma}_I, \boldsymbol{\sigma}_{II}), \quad i = I, II,$$

with a positive parameter γ, the *flow rate*. Using the flow rule, we derive the yield curve (2.15) and get

$$\dot{\boldsymbol{\epsilon}}_I = \gamma 2(\boldsymbol{\sigma}_I + \frac{P}{2}), \quad \dot{\boldsymbol{\epsilon}}_{II} = \gamma 2 e^2 \boldsymbol{\sigma}_{II}. \tag{2.16}$$

To determine γ, we reformulate (2.16) to

$$\boldsymbol{\sigma}_I = \frac{\dot{\boldsymbol{\epsilon}}_I}{2\gamma} - \frac{P}{2}, \quad \boldsymbol{\sigma}_{II} = \frac{e^{-2}}{2\gamma} \dot{\boldsymbol{\epsilon}}_{II}, \tag{2.17}$$

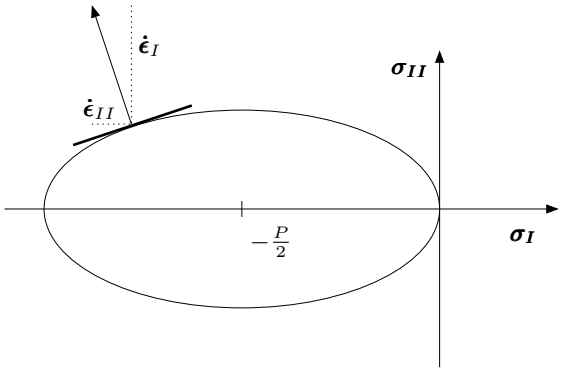


Figure 2.3.: Two-dimensional yield curve. The change of strain in the plastic state is only possible in normal direction. The stress increment can only change in tangential direction.

and insert $\boldsymbol{\sigma}_I$, $\boldsymbol{\sigma}_{II}$ in the yield curve (2.15),

$$\Big(\frac{\dot{\boldsymbol{\epsilon}}_I}{2\gamma}\Big)^2 + \Big(\frac{e^{-1}\dot{\boldsymbol{\epsilon}}_{II}}{2\gamma}\Big)^2 = \Big(\frac{P}{2}\Big)^2.$$

Finally, we obtain

$$\gamma = \frac{\Delta(\dot{\boldsymbol{\epsilon}})}{P}, \quad \Delta(\dot{\boldsymbol{\epsilon}}) := \sqrt{\dot{\boldsymbol{\epsilon}}_I^2 + \dot{e}^{-2}\boldsymbol{\epsilon}_{II}^2}. \tag{2.18}$$

Using (2.9) and (2.11), we reformulate the constitutive law (2.13) in stress and strain invariant coordinates to

$$\boldsymbol{\sigma}_I = \zeta\dot{\boldsymbol{\epsilon}}_1 - \frac{P}{2}, \quad \boldsymbol{\sigma}_{II} = \eta\dot{\boldsymbol{\epsilon}}_{II}. \tag{2.19}$$

By comparing (2.19) to (2.17), we find that the constitutive law is connected to the yield curve if the viscosities are chosen as

$$\zeta = \frac{1}{2\gamma}, \quad \eta = \frac{e^{-2}}{2\gamma}. \tag{2.20}$$

Doing so, the constitutive law allows exclusive stress states that lie on the yield curve. Thus, sea ice is modeled as a plastic material.

Viscous regime Assuming that sea ice is in a plastic state we observe that the viscosities (2.20) are degenerating, $\eta, \zeta \to \infty$, if $\Delta(\dot{\boldsymbol{\epsilon}})$ (2.18) is going to zero for $\dot{\boldsymbol{\epsilon}} \to 0$. Besides the numerical problems, the physical assumption that ice tends to display plastic behavior for arbitrary small deformations is not suitable. Thus, a description is needed for the limiting case $\dot{\boldsymbol{\epsilon}} \to 0$. First, Coon et al. [13] suggested modeling sea ice as an *elastic* material for $\dot{\boldsymbol{\epsilon}} \to 0$. As the elasticity assumption is numerically costly, Hibler introduced

a viscous closure by assuming that sea ice shows viscous behavior for $\dot{\boldsymbol{\epsilon}} \to 0$ [25]. Doing so, he limited $\Delta(\dot{\boldsymbol{\epsilon}})$ by

$$\Delta(\dot{\boldsymbol{\epsilon}}) \geq \Delta_{min} := 2 \cdot 10^{-9} s^{-1}. \tag{2.21}$$

The limitation of $\Delta(\dot{\boldsymbol{\epsilon}})$ is related to maximal values for the viscosities η and ζ and avoids the degeneration for $\Delta(\dot{\boldsymbol{\epsilon}}) \to 0$. For $\Delta(\dot{\boldsymbol{\epsilon}}) > \Delta_{min}$ sea ice is modeled as a plastic material, whereas for $\Delta(\dot{\boldsymbol{\epsilon}}) \leq \Delta_{min}$ sea ice behaves like a viscous material with constant viscosities. In the viscous state, the model generates stress states that are lying in the inside of the ellipse such that $F(\boldsymbol{\sigma}_I, \boldsymbol{\sigma}_{II}) < 0$.

The full viscous-plastic rheology reads as

$$\boldsymbol{\sigma} = \frac{P}{2}\Big(\frac{\mathrm{tr}(\dot{\boldsymbol{\epsilon}})I}{\max(\Delta(\dot{\boldsymbol{\epsilon}}), \Delta_{min})} + 2e^{-2}\frac{\dot{\boldsymbol{\epsilon}}'}{\max(\Delta(\dot{\boldsymbol{\epsilon}}), \Delta_{min})} - I\Big). \tag{2.22}$$

Replacement closure In the limiting case $\Delta(\dot{\boldsymbol{\epsilon}}) \to 0$, the internal stresses are not zero as

$$\lim_{\dot{\boldsymbol{\epsilon}} \to 0} \boldsymbol{\sigma}_I = -\frac{P}{2}.$$

This is avoided by a so-called *replacement closure* [45]. The idea of the replacement closure is to replace the ice strength by an ideal plastic ice strength, P_p

$$P = r_p(\Delta(\dot{\boldsymbol{\epsilon}}))P_p,$$

with

$$r_p(\Delta(\dot{\boldsymbol{\epsilon}})) = \begin{cases} 1, & \text{if } \Delta(\dot{\boldsymbol{\epsilon}}) \geq \Delta_{min} \\ 0, & \text{else.} \end{cases}$$

For a smooth and differentiable transition between plastic and viscous regime, Kreyscher et al. [45] replaced $\Delta(\dot{\boldsymbol{\epsilon}})$ by

$$r_p(\Delta(\dot{\boldsymbol{\epsilon}})) = \frac{\Delta(\dot{\boldsymbol{\epsilon}})}{\Delta(\dot{\boldsymbol{\epsilon}}) + \Delta_{min}}.$$

To be closer to the original non-smooth proposal of Hibler (2.22), Lemieux and Tremblay [53] introduced the viscosities as

$$\eta = \eta_{max} \tanh\Big(\frac{P_p}{2\Delta(\dot{\boldsymbol{\epsilon}})\eta_{max}}\Big),$$

where P_p is chosen similar as in Kreyscher et al. [45]. We use a similar approach as Kreyscher et al. [45] to smooth the transition from the viscous to the plastic regime and modify $\Delta(\dot{\boldsymbol{\epsilon}})$ by

$$\Delta_T(\dot{\boldsymbol{\epsilon}}) := \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \mathrm{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}, \tag{2.23}$$

where $A : B := \sum_{ij} A_{ij}B_{ij}$ is the inner product of two tensors. However, we do not use a replacement pressure for the limiting case $\Delta(\boldsymbol{\epsilon}) \to 0$. For a better readability, we will write $\Delta(\dot{\boldsymbol{\epsilon}})$ instead of $\Delta_T(\dot{\boldsymbol{\epsilon}})$, if it is not strictly necessary to distinguish between $\Delta(\dot{\boldsymbol{\epsilon}})$ and $\Delta_T(\dot{\boldsymbol{\epsilon}})$.

2.5. Complete system of equations

Let $\Omega \subset \mathbb{R}^2$ be the spatial domain and $I = [0, T] \subset \mathbb{R}$ be the time interval of interest. The boundary of the domain is denoted by $\partial\Omega$ and the inflow part of the boundary by $\Gamma_{in} := \{(x, y) \in \partial\Omega | \, \mathbf{n} \cdot \mathbf{v} < 0\}$, where $\mathbf{n}$ is the outward unit normal vector. We search $\mathbf{v}(x, y, t) \in \mathbb{R}^2$, $A(x, y, t) \in [0, 1]$ and $H(x, y, t) \in [0, \infty)$ such that

$$\rho_{\text{ice}} H \left(\partial_t \mathbf{v} + f_c \boldsymbol{e}_r \times (\mathbf{v} - \mathbf{v}_{ocean})\right) - \operatorname{div} \boldsymbol{\sigma} - \boldsymbol{\tau}_{\text{ocean}}(\mathbf{v}) - \boldsymbol{\tau}_{\text{atm}} = 0, \tag{2.24}$$

$$\mathbf{v} = \mathbf{v}^0 \text{ on } \{0\} \times \Omega, \quad \mathbf{v} = 0 \text{ on } I \times \partial\Omega. \tag{2.25}$$

$$\begin{aligned} \partial_t A + \operatorname{div}(\mathbf{v}A) = 0, \quad A = A^0 \text{ on } \{0\} \times \Omega, \quad A = A^{in} \text{ on } I \times \Gamma^{in} \\ \partial_t H + \operatorname{div}(\mathbf{v}H) = 0, \quad H = H^0 \text{ on } \{0\} \times \Omega, \quad H = H^{in} \text{ on } I \times \Gamma^{in} \end{aligned} \tag{2.26}$$

with the viscous-plastic rheology

$$\boldsymbol{\sigma} = 2\eta \dot{\boldsymbol{\epsilon}}' + \zeta \operatorname{tr}(\dot{\boldsymbol{\epsilon}}) I - \frac{P}{2} I, \tag{2.27}$$

and the viscosities

$$\eta = e^{-2}\zeta, \quad \zeta = \frac{P}{2\Delta(\dot{\boldsymbol{\epsilon}})}, \quad \Delta(\dot{\boldsymbol{\epsilon}}) := \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}. \tag{2.28}$$

The ice strength is modeled as

$$P = P^\star H \exp\big(-C(1 - A)\big).$$

The forcing from ocean and atmosphere is given as

$$\begin{aligned} \tau_{\text{ocean}}(\mathbf{v}) &= C_{\text{ocean}} \rho_{\text{ocean}} \|\mathbf{v}_{\text{ocean}} - \mathbf{v}\|_2 (\mathbf{v}_{\text{ocean}} - \mathbf{v}), \\ \tau_{\text{atm}} &= C_{\text{atm}} \rho_{\text{atm}} \|\mathbf{v}_{\text{atm}}\|_2 (\mathbf{v}_{\text{atm}}), \end{aligned}$$

where by $\|\cdot\|_2$ we denote the Euclidean norm. All constants are defined in Table 2.1.

2.5.1. Variational formulation

The presentation of the variational formulation is based on the paper of Mehlmann and Richter [70]. As we discretize the sea ice model with a *finite element approach*, a *variational formulation* of the set of equations is needed. In the following, we present the sea ice model in a variational formulation and discuss its similarities to well known problems.

We denote the usual Sobolev space of $L^2(\Omega)$-functions with first weak derivatives in $L^2(\Omega)$ and with trace zero on the boundary Γ by $H_0^1(\Omega)$. We prescribe homogeneous Dirichlet conditions $\mathbf{v} = 0$ on the whole boundary $\Gamma := \partial\Omega$.

Then, the velocity is found in

$$\mathbf{v}(t) \in \mathcal{V},$$

Parameter	Definition	Value
ρ_{ice}	sea ice density	$900\,\text{kg/m}^3$
ρ_{atm}	air density	$1.3\,\text{kg/m}^3$
ρ_{ocean}	water density	$1026\,\text{kg/m}^3$
C_{atm}	air drag coefficient	$1.2\cdot10^{-3}$
C_{ocean}	water drag coefficient	$5.5\cdot10^{-3}$
f_c	Coriolis parameter	$1.46\cdot10^{-4}\,\text{s}^{-1}$
$P^\star$	ice strength parameter	$27.5\cdot10^3\,\text{N/m}^2$
C	ice concentration parameter	20
e	ellipse ratio	2

Table 2.1.: Physical parameters of the momentum equation.

where $\mathcal{V} \subset H_0^1(\Omega)^2$ is a function space with adequate regularity for the viscous-plastic rheology. The ice height and the ice concentration are found in

$$H(t) \in \mathcal{V}^H(t) := \{\phi \in L^2(\Omega),\ \mathbf{v}(t)\cdot\nabla\phi \in L^2(\Omega),\ \phi \geq 0 \text{ a.e.}\},\ \partial_t H(t) \in L^2(\Omega),$$
$$A(t) \in \mathcal{V}^A(t) := \{\phi \in L^2(\Omega),\ \mathbf{v}(t)\cdot\nabla\phi \in L^2(\Omega),\ 0 \leq \phi \leq 1 \text{ a.e.}\},\ \partial_t A(t) \in L^2(\Omega).$$

To derive the variational formulation, we multiply the momentum equation with $\boldsymbol{\phi} \in \mathcal{V}$ and apply partial integration. Further, the balance laws are multiplied with $\phi_H, \phi_A \in \mathcal{L} := L^2(\Omega)$ and integrated over Ω.

Finally, for all $t \in [0,T]$ we find $\mathbf{v}$, H and A such that

$$\begin{aligned}(\rho_{ice}H\partial_t\mathbf{v},\boldsymbol{\phi}) + (\rho_{ice}Hf_c\boldsymbol{e}_r\times(\mathbf{v}-\mathbf{v}_{\text{ocean}}),\boldsymbol{\phi}) - (\boldsymbol{\tau}_{\text{atm}},\boldsymbol{\phi})& \\ -(\boldsymbol{\tau}_{\text{ocean}}(\mathbf{v}),\boldsymbol{\phi}) + (\boldsymbol{\sigma}(\mathbf{v},H,A),\nabla\boldsymbol{\phi}) = 0 \quad \forall\boldsymbol{\phi}\in\mathcal{V},& \\ (\partial_t H + \mathbf{v}\cdot\nabla H + H\,\text{div}(\mathbf{v}),\phi_H) = 0 \quad \forall\phi_H\in\mathcal{L},& \\ (\partial_t A + \mathbf{v}\cdot\nabla A + A\,\text{div}(\mathbf{v}),\phi_A) = 0 \quad \forall\phi_A\in\mathcal{L},&\end{aligned} \tag{2.29}$$

whereby $(\cdot,\cdot)$ we denote the L^2- inner product on Ω. The constraints $H(x,y,t) \geq 0$ and $A(x,y,t) \in [0,1]$ are embedded in the trial-spaces $\mathcal{V}^A$ and $\mathcal{V}^H$ and must be realized by a projection of the solution.

Note that until now, there are no existence and regularity results on the viscous-plastic (VP) sea ice model. The viscous-plastic rheology in the variational formulation has similarities to the regularized p-Laplacian

$$(\mu(\Delta_{min}^2 + |\nabla u|^2)^{\frac{p-2}{2}}\nabla u, \nabla\phi) = (f,\phi),$$

which has a unique solution for $p > 1$. However, the limit case $p \to 1$ is not covered by theory [102]. Another similarity to the viscous plastic rheology is found with the time-dependent minimal surface problem, see [60] and Chapter 6 and Chapter 7 of [2]. Here, for the scalar equation

$$(\partial_t u,\phi) + ((1+|\nabla u|^2)^{-\frac{1}{2}}\nabla u,\nabla\phi) = (f,\phi)$$

the existence of solutions in the space $u \in W^{1,1}(\Omega)$ is shown. We believe that this result can be extended to the sea ice momentum equation assuming that A, H are smooth with $A, H > 0$, such that the operator does not degenerate. The assumption is motivated by the numerical treatment of the sea ice model. Here it is common to apply a partitioned solution approach e.g., in [25, 30, 53, 96] such that the momentum equation and the balance laws are decoupled in time and the momentum equation is solved with smoothed A, H. The major difficulty in proofing existence and uniqueness for the sea ice momentum equation is to extend this scalar result outlined above to a system of equations.

2.5.2. Sea ice model in a non-dimensional framework

For a better evaluation of the numerical discretization, we derive a non-dimensional form of the sea ice model. By L we define a characteristic horizontal length scale, by G a typical ice thickness scale and by T a characteristic time scale. The length and the thickness is given in meters, whereas the time is chosen in seconds. We define the non-dimensional quantities

$$\bar{t} := \frac{t}{T}, \quad \bar{x} := \frac{x}{L}, \quad \bar{H} := \frac{H}{G}.$$

It holds that

$$\operatorname{div}(\cdot) = L^{-1}\bar{\operatorname{div}}(\cdot), \quad \partial_t = T^{-1}\bar{\partial}_t.$$

Then, the non-dimensional velocity $\bar{\mathbf{v}}$ and the strain rates are given by

$$\mathbf{v} = \frac{L}{T}\bar{\mathbf{v}}, \quad \partial_t \mathbf{v} = \frac{L}{T^2}\partial_{\bar{t}}\bar{\mathbf{v}}, \quad \dot{\boldsymbol{\epsilon}} = T^{-1}\bar{\dot{\boldsymbol{\epsilon}}}.$$

We begin with transforming the sea ice rheology. The nonlinearity

$$\Delta_T(\dot{\boldsymbol{\epsilon}}) := \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}$$

transforms to

$$\Delta_T(\dot{\boldsymbol{\epsilon}}) = T^{-1}\bar{\Delta}_T(\bar{\dot{\boldsymbol{\epsilon}}}), \quad \Delta_{min}^2 = T^{-2}\bar{\Delta}_{min}^2, \quad \bar{\Delta}_T(\bar{\dot{\boldsymbol{\epsilon}}}) = \sqrt{(2e^{-2}\bar{\dot{\boldsymbol{\epsilon}}}' : \bar{\dot{\boldsymbol{\epsilon}}}' + \operatorname{tr}(\bar{\dot{\boldsymbol{\epsilon}}})^2) + \bar{\Delta}_{min}^2}.$$

The ice strength is given as

$$P(A, H) = P^\star H \exp(-C(1-A)) = P^\star G\bar{H} \exp(-C(1-A)).$$

Thus, the viscous-plastic sea ice rheology,

$$\boldsymbol{\sigma} = \frac{P}{2\Delta_T(\dot{\boldsymbol{\epsilon}})}\Big(\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I + 2e^{-2}\dot{\boldsymbol{\epsilon}}' - \Delta_T(\dot{\boldsymbol{\epsilon}})I\Big),$$

is transformed to

$$\boldsymbol{\sigma} = \frac{P^* G}{2} \bar{\boldsymbol{\sigma}},$$

with

$$\bar{\boldsymbol{\sigma}} := \frac{1}{\bar{\Delta}_T(\bar{\dot{\boldsymbol{\epsilon}}})} \bar{H} \exp(-C(1-A)) \left(\mathrm{tr}(\bar{\dot{\boldsymbol{\epsilon}}}) I + e^{-2} \bar{\dot{\boldsymbol{\epsilon}}}' - \Delta_T(\bar{\dot{\boldsymbol{\epsilon}}}) I \right).$$

We proceed with transforming the momentum equation and obtain

$$\rho_{ice} H \partial_t \mathbf{v} = \frac{\rho_{\mathrm{ice}} G \bar{H} L}{T^2} \partial_t \bar{\mathbf{v}}, \tag{2.30}$$

$$\rho_{\mathrm{ice}} H f_c \times (\mathbf{v} - \mathbf{v}_{\mathrm{ocean}}) = \frac{G \bar{H} L}{T} f_c \times (\bar{\mathbf{v}} - \bar{\mathbf{v}}_{\mathrm{ocean}}), \tag{2.31}$$

$$\mathrm{div}(\boldsymbol{\sigma}) = \frac{P^* G}{2L} \,\mathrm{div}\, \bar{\boldsymbol{\sigma}}, \tag{2.32}$$

$$\boldsymbol{\tau}_{\mathrm{ocean}}(\mathbf{v}) = \frac{L^2}{T^2} \rho_{\mathrm{ocean}} C_{\mathrm{ocean}} \underbrace{\|\bar{\mathbf{v}}_{\mathrm{ocean}} - \bar{\mathbf{v}}\|_2 (\bar{\mathbf{v}}_{\mathrm{ocean}} - \bar{\mathbf{v}})}_{=:\bar{\boldsymbol{\tau}}_{\mathrm{ocean}}(\bar{\mathbf{v}})}, \tag{2.33}$$

$$\boldsymbol{\tau}_{\mathrm{atm}} = \frac{L^2}{T^2} \rho_{\mathrm{atm}} C_{\mathrm{atm}} \underbrace{\|\bar{\mathbf{v}}_{\mathrm{atm}}\|_2 \bar{\mathbf{v}}_{\mathrm{atm}}}_{=:\bar{\boldsymbol{\tau}}_{\mathrm{atm}}}, \tag{2.34}$$

with $\bar{\mathbf{v}}_{\mathrm{ocean}} = \frac{T}{L} \mathbf{v}_{\mathrm{ocean}}$ and $\bar{\mathbf{v}}_{\mathrm{atm}} = \frac{T}{L} \mathbf{v}_{\mathrm{atm}}$.

By scaling (2.30)-(2.34) with $\rho_{ice} G L / T^2$, we obtain the non-dimensional momentum equation as

$$\bar{H}(\partial_{\bar{t}} \bar{\mathbf{v}} + T f_c \times (\bar{\mathbf{v}} - \bar{\mathbf{v}}_{\mathrm{ocean}})) - C_a \bar{\boldsymbol{\tau}}_{\mathrm{atm}} - C_o \bar{\boldsymbol{\tau}}_{\mathrm{ocean}}(\bar{\mathbf{v}}) - C_r \,\mathrm{div}(\bar{\boldsymbol{\sigma}}) = 0, \tag{2.35}$$

with the constants

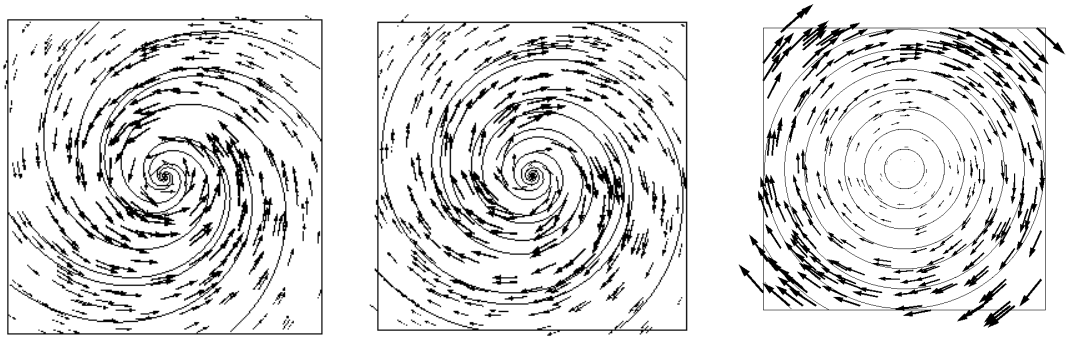
$$C_o := \frac{\rho_{\mathrm{ocean}} C_{\mathrm{ocean}} L}{\rho_{\mathrm{ice}} G}, \quad C_a := \frac{\rho_{\mathrm{atm}} C_{\mathrm{atm}} L}{\rho_{\mathrm{ice}} G}, \quad C_r := \frac{P^\star T^2}{2 \rho_{ice} L^2}.$$

Here, C_r describes the ratio between inertia and internal stress, C_o and C_a the ratio between inertia and external forcing. The balance equations are given as

$$\frac{G}{T}[\partial_{\bar{t}} \bar{H} + \bar{\mathbf{v}} \cdot \bar{\nabla} \bar{H} + \bar{H} \bar{\mathrm{div}}(\bar{\mathbf{v}})] = 0,$$
$$\frac{1}{T}[\partial_{\bar{t}} \bar{A} + \bar{\mathbf{v}} \cdot \bar{\nabla} \bar{A} + A \bar{\mathrm{div}}(\bar{\mathbf{v}})] = 0.$$

Finally, we choose $G = 1\,\mathrm{m}$, $L = 10^6\,\mathrm{m}$ and $T = 10^3\,\mathrm{s}$, and obtain, with literature values for $P^\star$, the densities and the drag coefficients,

$$C_r \approx 1.5 \cdot 10^{-5}, \quad C_o \approx 6270, \quad C_a \approx 1.73.$$



Cyclone at $t = 0\,\text{days}$ | Anticyclone at $t = 8\,\text{days}$ | Water stress

Figure 2.4.: Forcing by wind and water. The center of the wind field moves in time. The water forcing is stationary.

As a typical wind velocity is factor $10 \sim 100$ higher than the ocean velocity, the impact of the atmospheric and ocean forcing is comparable. Further, note that

$$\bar{\Delta}_{min} = T\Delta_{min} = 2 \cdot 10^{-6} \tag{2.36}$$

causes substantial internal stress. We omit the bars $\bar{\cdot}$, if it is not strictly necessary to distinguish between dimensional and non-dimensional quantities. Furthermore, we use the dimensional form of the equations for better readability.

2.6. Model problem

In this section, we start with introducing a model problem which will be used as a test case to evaluate the novel numerical methods. We present the numerical solution of the problem on different spatial resolutions. The presentation of the model problem is similar to the presentation in our paper [70].

The model problem is designed to have large deformations at the beginning of the simulation. A time-depending wind forcing avoids stationary solutions. We present a simple analytical wind and ocean forcing that makes the model problem easy to reproduce. Further, we consider the quadratic domain $\Omega = (0, 500\,\text{km})^2$ and measure the time t in days. Similar to Hunke [30], we prescribe a circular steady ocean current

$$\mathbf{v}_{ocean} = \bar{v}_{ocean}^{max} \begin{pmatrix} 2y/500\,\text{km} - 1 \\ 1 - 2x/500\,\text{km} \end{pmatrix}, \tag{2.37}$$

where the maximal ocean velocity is given as

$$\bar{v}_{ocean}^{max} = 0.01\,\text{m}\,\text{s}^{-1}.$$

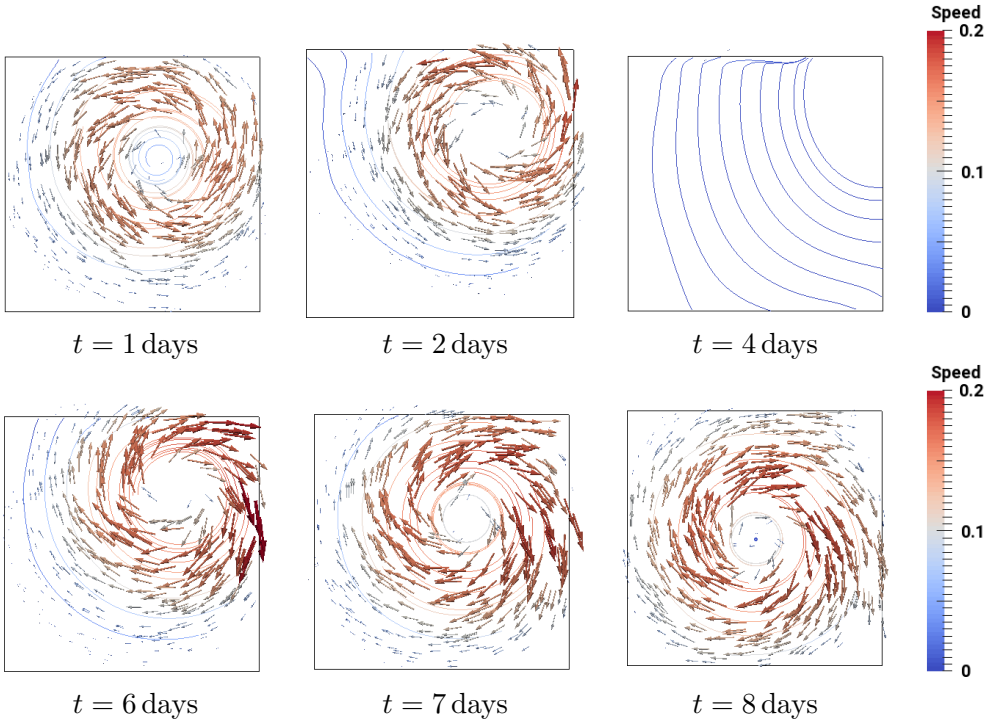


Figure 2.5.: Ice velocity vector field (in $\mathrm{m\,s^{-1}}$) and ice velocity streamlines. Evaluating the test case at $t = 1, 2, 4, 6, 7, 8\,\text{days}$ using a step size of $k = 0.5\,\text{h}$ and a 2 km mesh.

The wind field is diagonally passing from the midpoint to the edge or the edge to the midpoint of the computational domain,

$$v_{atm}(t) = \bar{v}_{atm}^{max} \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x - m_x(t) \\ y - m_y(t) \end{pmatrix} \omega(x, y).$$

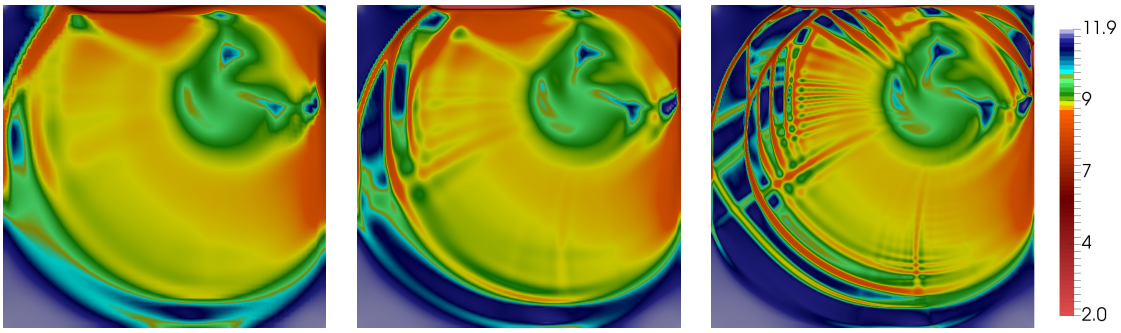
The wind forcing alternates between cyclonic to anticyclonic within a period of 4 days. The maximum wind velocity (in $\mathrm{m\,s^{-1}}$) is given by

$$\bar{v}_{atm}^{max}(t) = 15\,\mathrm{m\,s^{-1}} \begin{cases} -\tanh\left(\frac{(4-t)(4+t)}{2}\right) & t \in [0, 4], \\ \tanh\left(\frac{(12-t)(t-4)}{2}\right) & t \in [4, 8]. \end{cases}$$

The center of the cyclone is a function of time and reads as

$$m_x(t) = m_y(t) = \begin{cases} 250\,\text{km} + 50t\,\text{km/day} & t \in [0, 4], \\ 650\,\text{km} - 50t\,\text{km/days} & t \in [4, 8]. \end{cases}$$

Divergence and convergence angles of the anticyclone and the cyclone are set to 9° and



mesh size 8 km | mesh size 4 km | mesh size 2 km

Figure 2.6.: Plot of η in a logarithmic scale to the basis 10. Evaluating the test case at $t = 2$ days using a time step size of $k = 0.5$ h. On a 2 km mesh the viscosities vary by nearly 10 orders of magnitude on very small spatial scales.

18°, which correspond to

$$\alpha = \begin{cases} 90° - 18° = 72° & t \in [0, 4], \\ 90° - 9° = 81° & t \in [4, 8]. \end{cases}$$

To reduce the wind strength away from the center, we choose $\omega(x, y)$ as

$$\omega(x, y) = \frac{1}{50} \exp\left(-\frac{r(x, y)}{100\,\text{km}}\right), \quad r(x, y) = \sqrt{(x - m_x(t))^2 + (y - m_y(t))^2}.$$

Snapshots of the wind and the water forcing are shown in Figure 2.4.

As initial condition, we use zero velocity, a constant ice concentration $A(0, x, y) = 1$ and a small variation of the ice height at $H(0, x, y) = 0.3\,\text{m}$,

$$\begin{aligned} \mathbf{v}(0, x, y) = \mathbf{v}^0(x, y) &:= 0\,\text{m/s}, \quad A^0 := A(0, x, y) = 1, \\ H(0, x, y) = H^0(x, y) &:= 0.3\,\text{m} + 0.005\,\text{m}\left(\sin\left(\frac{x}{2\,\text{km}}\right) + \sin\left(\frac{y}{2\,\text{km}}\right)\right). \end{aligned}$$

At the boundary of the domain we apply a no-slip condition for the velocity

$$\mathbf{v} = 0 \text{ on } \partial\Omega.$$

In Figure 2.5, we show snapshots of the ice velocity field at $t = 1, 2, 4, 6, 7, 8$ days.

Numerical solution with increasing spatial resolutions In this thesis, we analyze the model problem at different mesh resolutions,

$$\frac{500\text{km}}{2^6} \approx 7.81\,\text{km}, \quad \frac{500\text{km}}{2^7} \approx 3.91\,\text{km}, \quad \frac{500\text{km}}{2^8} \approx 1.95\,\text{km}.$$

For simplicity, we refer to these meshes as 8 km, 4 km and 2 km. Further, we use time steps of

$$k = 4000\,\text{sec} \approx 1\,\text{h},\ k = 2000\,\text{sec} \approx 0.5\,\text{h},\ k = 1000\,\text{sec} \approx 0.25\,\text{h}.$$

Figure 2.6 shows the bulk viscosity η in logarithmic scale to the basis 10 under mesh refinement. The values of η vary over nearly 10 magnitudes. As the mesh is refined, the values of η change more rapidly. This effect is most visible on the 2 km mesh. The increasing variation of viscosities on high spatial resolutions cause difficulties for the nonlinear solvers and makes the momentum equation very hard to solve. In applications, where these small-scale patterns are of interest, robust solvers are required.

To evaluate the discretization of the model problem we follow Lemieux and Tremblay [55] and show the states of stress in the stress-invariant space $(\frac{\sigma_I}{P}, \frac{\sigma_{II}}{P})$ with

$$\begin{aligned} \sigma_I &= \zeta\dot{\boldsymbol{\epsilon}}_I - \frac{P}{2}, \qquad & \sigma_{II} &= \eta\dot{\boldsymbol{\epsilon}}_{II}, \\ \dot{\boldsymbol{\epsilon}}_I &= \mathbf{v}_x^1 + \mathbf{v}_y^2, \qquad & \dot{\boldsymbol{\epsilon}}_{II} &= \sqrt{(\mathbf{v}_x^1 - \mathbf{v}_y^2)^2 + (\mathbf{v}_y^1 + \mathbf{v}_x^2)^2}. \end{aligned}$$

Figure 2.7, shows the stress states in the stress-invariant space for different mesh levels. Details on the ellipse are given in Section 2.4.3. The maximal residual of equation (2.14) is bounded by $5 \cdot 10^{-6}$. While the maximal residual does not increase on fine meshes, the number of nodes with high residual is slightly enlarged. On a 2 km mesh 1.7% of stress states, on a 4 km mesh 1.4% of stress states and on a 8 km mesh 0.8% of stress states lying outside of the elliptical curve. Following Lemieux et al. [57] an approximation is considered to be a solution of the viscous-plastic sea ice model if the following conditions are fulfilled:

$$\begin{aligned} 1 \leq &\frac{\boldsymbol{\sigma}_I}{P} \leq 0 \ \text{ for at least 99\% of the } \boldsymbol{\sigma}_I, \\ 0 \leq &\frac{\boldsymbol{\sigma}_{II}}{P} \leq f(\frac{\boldsymbol{\sigma}_I}{P}) + 0.005, \quad \text{with} \quad f(\frac{\boldsymbol{\sigma}_I}{P}) = 0.25\sqrt{-(2\frac{\boldsymbol{\sigma}_I}{P} + 1)^2 + 1}. \end{aligned}$$

We fulfill both criteria for every grid point on all mesh levels.

Solution of the nonlinear and linear systems In every time step, we solve the nonlinear momentum equation until either a global residual condition is fulfilled

$$\|\mathcal{R}(\mathbf{v}^{(l)})\| \leq tol_g, \quad tol_g := 10^{-13},$$

or until a relative reduction of the residual with respect to the initial residual is obtained

$$\|\mathcal{R}(\mathbf{v}^{(l)})\| \leq \|\mathcal{R}(\mathbf{v}^{(0)})\| \cdot \gamma_{nl}, \quad \gamma_{nl} := 10^{-4}.$$

The residual is defined in equation (5.23) in Section 5.2. The linear problems $\mathbf{A}[\mathbf{v}^{(l-1)}]\mathbf{w} = \mathbf{b}$ arising within the Newton iteration are approximated with the tolerance $tol := 10^{-4}$,

$$\|\mathbf{b} - \mathbf{A}[\mathbf{v}^{(l-1)}]\mathbf{w}\| \leq tol\|\mathbf{b}\|, \quad tol := 10^{-4}.$$

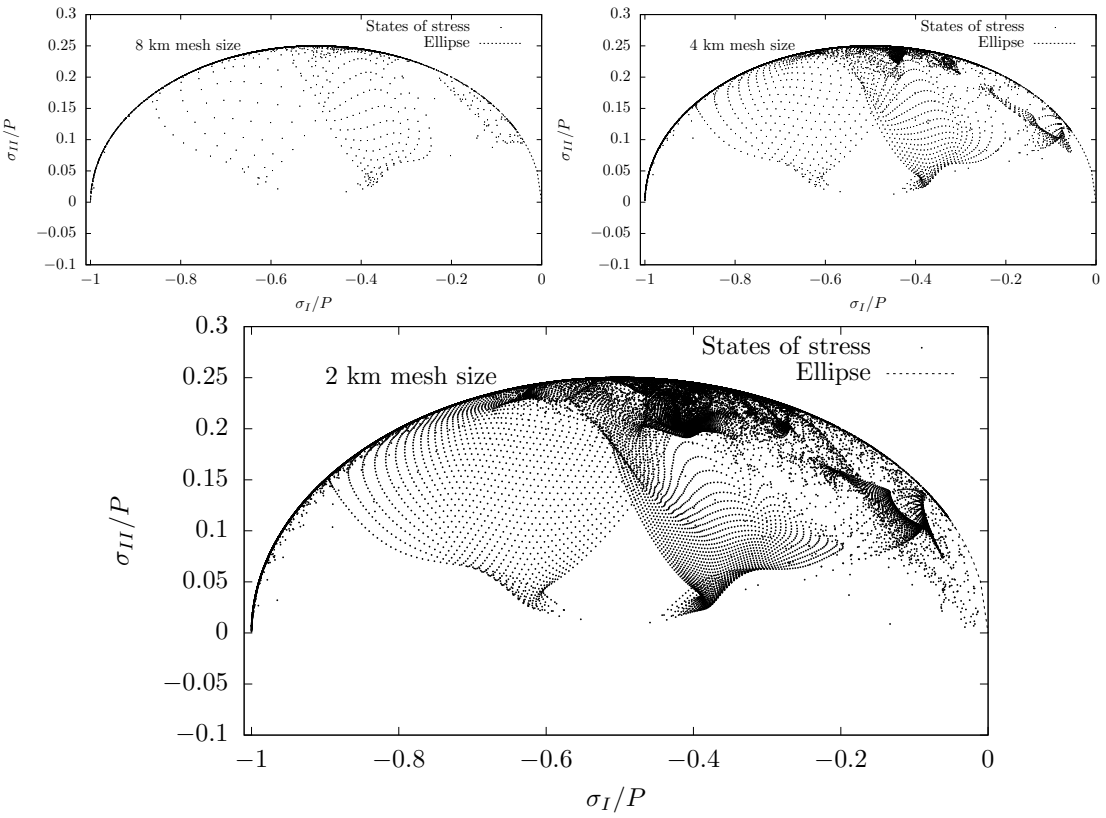


Figure 2.7.: States of stress in the stress-invariant space. Evaluating the test case at $t = 2$ days using a time step size of $k = 0.5$ h.

We limit the number of nonlinear steps to 200 iterations and the number of linear steps to 50 iterations. The nonlinear system is solved with the *operator-related damped Jacobian* method as described in Section 5.2. The solution of the linear system is computed with a *generalized minimal residual method* with a *multigrid preconditioner*, see Section 5.4.

Part II.

Discretization of the sea ice model

3. Discretization

Following Lietaer et al. [61], the first finite element sea ice models were proposed after the AIDJEX (Arctic Ice Dynamics Joint Experiment) expeditions in the 1970s. During the 1980s to the early 2000s finite elements have been applied in many idealized test cases, as they can be easily used on unstructured meshes and complex geometries [61, 95]. In the beginning of the 21 century some regional sea ice models based on a finite element approach have been proposed by Kliem in [43], by Lietaer et al. in [61], and Yakovlev in [109]. A global finite element sea ice model was published by Danilov and Timmermann [95] in 2009. It was developed in connection with the *Finite Element Sea Ice-Ocean Model* (FESOM) [96] and is based on the *viscous-plastic* (VP) rheology with an *elastic-visco-plastic* (EVP) formulation developed by Hunke [31]. A short description of the EVP formulation is given in Chapter 5. In FESOM, linear finite elements on an unstructured triangular mesh are used for the spatial discretization of the ice velocity, ice thickness and ice concentration. This was the first time that a global sea ice ocean simulation was done on an unstructured mesh. In [95] the authors apply a *Galerkin least squares stabilization* to the balance laws. In a second version Danilov and Timmermann [15] use a *flux-corrected Taylor-Galerkin* scheme to discretize the balance laws. Furthermore, in addition to the EVP formulation the authors implemented an implicit discretization of the VP rheology in [15].

This chapter is structured as follows. In Section 3.1 we present a *partitioned solution approach* to decouple the system consisting of a momentum equation and the balance laws. Section 3.2 is devoted to spatial and temporal discretization of the momentum equation. In Section 3.3 we discuss the discretization of the balance laws.

3.1. Partitioned solution approach

To discretize the viscous-plastic sea is model (2.29), we decouple the momentum equation and the balance laws. This approach is standard to cope with the complex coupled system and has been used, e.g., in [25, 30, 53, 96]. In [68], we considered a fully coupled monolithic and implicit solution approach, where both equations are treated as one unit. However, this procedure significantly increases the computational cost. In order to enhance the stability of an approximation a time step of less than 9 seconds has been applied. Therefore, the advantage of an implicit time stepping scheme, which allows stable discretizations with arbitrary large time steps, is lost. Thus, we recommend to decouple the system.

Let $I = [0, T]$ be the time span of interest. We start by introducing a discretization

of this period into sub intervals $I_n = [t_{n-1}, t_n]$, where

$$0 = t_0 < t_1 < \cdots < t_N = T.$$

The time step size is denoted by $k_n := t_n - t_{n-1}$. For simplicity, we assume that $k \equiv k_n$ is constant for all steps. At time t_n we denote the velocity, the ice concentration and the thickness by $\mathbf{v}_n, A_n, H_n$. To solve the sea ice model (2.29) we choose the following partitioned solution approach:

Algorithm 1 (Partitioned Solution Approach)**.** *Let $\mathbf{v}_0, A_0, H_0$ be the initial solutions at time $t = 0$. Iterate for $n = 1, 2, \ldots, N$:*

1. *Solve the balance equations based on the velocity $\mathbf{v}_{n-1}$*

$$A_{n-1} \mapsto A_n, \quad H_{n-1} \mapsto H_n.$$

2. *Solve the momentum equations based on A_n and H_n*

$$\mathbf{v}_{n-1} \mapsto \mathbf{v}_n.$$

In contrast to Lemieux et al. [53], we first solve the momentum equations with A_{n-1} and H_{n-1} from the previous time step and then update the balance laws. As we consider an *explicit Taylor-Galerkin* method for the approximation of ice thickness and ice concentration the presented sequence is more natural. Lemieux et al. [53] observed that using the ice strength P from the previous time step in the momentum equation can cause instabilities, if larger time steps ($\sim 1h$) are applied. This instabilities especially occur in boundary layers [33]. To avoid this instability, Lemieux et al. [53] developed an iterated implicit-explicit time stepping scheme, where the update of ice thickness H and ice concentration A is moved to the inside of the Newton iteration, that is used to solve the momentum equation. This method is an approximation to a monotholitic solution approach and is summarized in Algorithm 2:

Algorithm 2 (Iterated Implicit Solution Approach)**.** *Let $\mathbf{v}_0, A_0, H_0$ be the initial solution at time $t = 0$. In each time step $n = 1, 2, \ldots, N$ we start the Newton iteration with the initial solution $\mathbf{v}_n^{(0)} = \mathbf{v}_{n-1}$, $H_n^{(0)} = H_{n-1}$, $A_n^{(0)} = A_{n-1}$ and iterate for $l = 1, 2, \cdots, l_{max}$:*

1. *Update the balance laws based on the velocity $\mathbf{v}_n^{(l-1)}$*

$$A_{n-1} \mapsto A_n, \quad H_{n-1} \mapsto H_n.$$

2. *Solve the Newton update of the momentum equation based on $H_n^{(l)}, A_n^{(l)}$*

$$\mathbf{v}_n^{(l-1)} \mapsto \mathbf{v}_n^{(l)}.$$

3. *Stop, if the residual $\mathcal{R}(\cdot)$ of the momentum equation is reduced by a certain tolerance γ_{nl}*

$$||\mathcal{R}(\mathbf{v}_n^{(l)})|| \leq \gamma_{nl}||\mathcal{R}(\mathbf{v}_n^{(0)})||.$$

In Section 6, we will compare this two algorithms on a pan-Arctic sea ice model. In Section 7, we estimate the error contribution due to the decoupling in time, as described in Algorithm 1.

3.2. Discretization of the momentum equation

We discretize the momentum equation using Rothe's method [50], i.e, we begin by discretizing the momentum equation in time. After that we discretize the semi-discrete problem in space.

Discretization in time To derive stable and accurate discretizations of the sea ice deformations an implicit treatment of the momentum equation is recommended, see Lemieux et al. [57] or Hibler and Zhang [111]. Ip et al. [34] pointed out that a fully explicit time stepping scheme for the momentum equation with a VP rheology would require a small time step of less than a second - even on a grid resolution as coarse as 100 km by 100 km. Therefore, they recommend to use an implicit time stepping scheme. Like Lemieux and Tremblay [56], we use the simple *implicit Euler* time stepping method. The implicit Euler scheme is a *strongly A-stable* method [101]. This property allows a stable approximation with arbitrary large time steps. We note that due to the strong nonlinearity of the momentum equation, the time step size of the implicit Euler method is limited by the convergence of the nonlinear solver. We observe that the convergence of the modified Newton scheme in Chapter 5 worsens for time steps larger than 1 h. In Chapter 4, we find that the spatial discretization error is dominating the overall numerical error, even on highly resolved meshes. Thus, the use of a first order implicit Euler method is sufficient. It is known by theory that strongly A-stable methods have a smoothing property [101], i.e., strongly A-stable methods allow for energy estimates for all time steps sizes. Here, they are suitable to model processes with locally distorted problem data, e.g., by measurement errors. The coupling of the balance laws and the atmospheric and the oceanic forcing gives disturbed data in this application. This makes the use of an implicit Euler method appropriate. The semi-discrete momentum equation reads as

$$\mathbf{v}_n \in \mathcal{V}: \quad A(\mathbf{v}_n, \boldsymbol{\phi}) = F(\boldsymbol{\phi}), \quad \forall \boldsymbol{\phi} \in \mathcal{V}, \tag{3.1}$$

for $n = 1, ..., N$ with

$$\begin{aligned} A(\mathbf{v}_n, \boldsymbol{\phi}) :=& (\rho_{\text{ice}} H_n \mathbf{v}_n, \boldsymbol{\phi}_{\mathbf{v}}) + k(\rho_{ice} H_n f_c \boldsymbol{e}_r \times \mathbf{v}_n, \boldsymbol{\phi}_{\mathbf{v}}) \\ &+ k(\boldsymbol{\sigma}(\mathbf{v}_n, H_n, A_n), \nabla \boldsymbol{\phi}) - k(\boldsymbol{\tau}_{\text{ocean}}(t_n, \mathbf{v}_n), \boldsymbol{\phi}), \\ F(\boldsymbol{\phi}) :=& (\rho_{\text{ice}} H_n \mathbf{v}_{n-1}, \boldsymbol{\phi}) + k(\boldsymbol{\tau}_{\text{atm}}(t_n), \boldsymbol{\phi}) + k(f_c \boldsymbol{e}_r \times \mathbf{v}_{\text{ocean}}, \boldsymbol{\phi}). \end{aligned} \tag{3.2}$$

Discretization in space The presentation of the following is similar to Mehlmann and Richter [70], where we first described the Newton-Multigrid framework for sea ice dynamics. For spatial discretization we use *finite elements* on a quadrilateral mesh. The model is implemented in the software library Gascoigne [5].

We consider a domain with a polygonal boundary. By Ω_h we introduce a triangulation of Ω into open quadrilaterals $K \in \Omega_h$ that fulfills the usual assumptions of structural-regularity and shape-regularity, i.e., it holds

$$\bar{\Omega} = \bigcup_{K \in \Omega_h} \bar{K}, \quad K \cap K' = \emptyset \; \forall K \neq K' \in \Omega_h,$$

and $\partial K \cap \partial K'$ (for $K \neq K'$) is either empty, a common vertex, a common edge of both quadrilaterals or – to include hanging nodes – the edge of one and half of the other's edge [19]. For shape-regularity we request that all interior angles are close to 90°. Further, we assume bounded aspect-ratios of all elements. All these conditions must hold uniform in $h > 0$ [19].

On Ω_h we denote the function space of bi-linear (r=1) or bi-quadratic (r=2) continuous and parametric finite elements.

$$V_h^{(r)} := \{\phi \in C(\bar{\Omega}),\ \forall K \in \Omega_h :\ \phi\big|_K \circ T_K \in Q^r\}, \quad Q^r := \text{span}\{x^i y^j,\ 0 \leq i, j \leq r\},$$

where by $T_K : (0,1)^2 \to K$ we describe the iso-parametric reference element map, i.e., $T_K \in [Q^r]^2$ [80]. For the ice velocity we define the space

$$V_h^{\mathbf{v}} := (\mathcal{V} \cap V_h^{(r)})^2, \quad r = 1, 2.$$

For simplicity, we assume that the finite element meshes and therefore the finite element spaces are static and do not change in time. However, an extension to changing meshes is possible. The fully discrete system for $n = 1, ..., N$ reads as

$$\mathbf{v}_{n,h} \in V_h^{\mathbf{v}} : \quad A(\mathbf{v}_n, \boldsymbol{\phi}_h) = F(\boldsymbol{\phi}_h), \quad \forall \boldsymbol{\phi}_h \in V_h^{\mathbf{v}}. \tag{3.3}$$

To complete the spatial discretization we introduce the nodal basis

$$V_h^{\mathbf{v}} = \text{span}\{\boldsymbol{\phi}_h^i, \quad i = 1, ..., N\},$$

and rewrite the solution of (3.3) as

$$\mathbf{v}_{n,h} = \sum_{i=1}^{N} \mathbf{v}_n^i \boldsymbol{\phi}_{n,h}^i,$$

where each coefficient $\mathbf{v}_h^i$ is a vector in $\mathbb{R}^2$. After discretization in space a nonlinear system of algebraic equations arises. In Section 5, we present the solver that is applied to the discrete momentum equation. The nonlinearity will be treated by a Newton method and the resulting linear systems of equations are approximated with the GMRES method preconditioned by a multigrid method. Details follow in Section 5.

3.3. Discretization of the balance laws

In analogy to the discretization of the momentum equation in Section 3.2, we discretize the balance laws with Rothe's method. In Section 3.3.1 we discuss the time discretization of the balance laws and present the *implicit Euler* scheme, the *Crank-Nicolson* method as well as a *second order Taylor-Galerkin* scheme as possible choices. For spatial discretization we use linear finite elements. As a direct finite element discretization of the balance laws gives rise to instabilities, we discuss different stabilization schemes such as *artificial diffusion*, a *streamline upwind Petrov-Galerkin method* and a *flux limiter*. These different approaches are presented and discussed in Section 3.3.2. In the following, we cover the discretization of the ice concentration, while the ice thickness can be derived analogously.

3.3.1. Discretization in time and space

θ-scheme We start by introducing the θ-scheme for the temporal discretization of the ice concentration. Thus, the time-discretized balance law (2.29) reads as

$$A_n - \theta k \operatorname{div}(\mathbf{v}_n A_n) = A_{n-1} + (1-\theta) k \operatorname{div}(\mathbf{v}_{n-1} A_{n-1}).$$

The variational formulation is then given by

$$A_n \in \mathcal{V}^{\mathcal{A}}: \quad \Big(A_n - \theta k \operatorname{div}(\mathbf{v}_n A_n), \phi\Big) = \Big(A_{n-1} + (1-\theta) k \operatorname{div}(\mathbf{v}_{n-1} A_{n-1}), \phi\Big) \quad \forall \phi \in \mathcal{L}. \tag{3.4}$$

For $\theta = 1$ we obtain the implicit Euler scheme. The properties of the implicit Euler scheme have been discussed in Section 3.2. For $\theta = \frac{1}{2}$ we get the Crank-Nicolson method. The latter is a second order scheme with no numerical diffusion and it maintains the energy of the system. Further, as an A-stable method it allows stable temporal discretizations for large time step, but due to the missing strong A-stability the Crank-Nicolson scheme can produce instabilities if the problem data is disturbed, see Turek [101]. The extension to second order methods with better stability properties like the *stabilized Crank-Nicolson* method presented by Luskin and Rannacher [65] or the *fractional step theta* scheme discussed by Glowinski [22] is possible.

Due to the application of a partitioned solution approach the velocity $\mathbf{v}_n$ is not at hand. Thus, we use either the last time step, an extrapolation or we updated the velocity in each Newton iteration as described in Algorithm 2.

Second order explicit Taylor-Galerkin method The underlying idea of a Taylor-Galerkin method is to use a Taylor expansion of the quantities of interest and to substitute the higher-order temporal derivatives with the spatial differential operator. The formal Taylor expansion in time is given as ($A_n = A(t_n)$)

$$A_n = A_{n-1} + k\partial_t A_{n-1} + \frac{k^2}{2}\partial_{tt} A_{n-1} + O(k^3). \tag{3.5}$$

To express the derivatives in (3.5), we insert the balance law given in (2.29).

$\partial_t A = \operatorname{div}(\mathbf{v}A)$, $\partial_{tt} A = \partial_t \operatorname{div}(\mathbf{v}A) = \operatorname{div}(\partial_t \mathbf{v} A + \mathbf{v}\partial_t A) = \operatorname{div}\ (\partial_t \mathbf{v} A) - \operatorname{div}(\mathbf{v}(\operatorname{div}(\mathbf{v}A))$.

Using the explicit Euler method the temporal discretized balance law (3.4) is given as

$$\frac{1}{k}(A_n - A_{n-1}) + \operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) = 0.$$

Thus, the ice concentration at time t_n is approximated as

$$\begin{aligned}
A_n =& A_{n-1} - k\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) - \frac{k^2}{2}\partial_t \operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) \\
=& A_{n-1} - k\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) - \frac{k^2}{2}\operatorname{div}(\partial_t \mathbf{v}_{n-1}A_{n-1} + \mathbf{v}_{n-1}\partial_t A_{n-1}) \\
=& A_{n-1} - k\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) - \frac{k^2}{2}\operatorname{div}\ (\partial_t \mathbf{v}_{n-1}A_{n-1} - \mathbf{v}_{n-1}(\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}))) \\
=& A_{n-1} - k\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) - \frac{k^2}{2}\operatorname{div}\ (\partial_t \mathbf{v}_{n-1}A_{n-1}) \\
&+ \frac{k^2}{2}\operatorname{div}(\mathbf{v}_{n-1}(A_{n-1}\operatorname{div}\mathbf{v}_{n-1})) + \frac{k^2}{2}\operatorname{div}(\mathbf{v}_{n-1}(\mathbf{v}_{n-1}\cdot\nabla A_{n-1}))\,.
\end{aligned} \tag{3.6}$$

The extra term

$$\frac{k^2}{2}\operatorname{div}(\mathbf{v}_{n-1}(\mathbf{v}_{n-1}\cdot\nabla A_{n-1})),$$

add stability to the system, since $\Big(\operatorname{div}(\mathbf{v}_{n-1}(\mathbf{v}_{n-1}\cdot\nabla A_{n-1})), \phi\Big) = \Big(\mathbf{v}\cdot\nabla A, \mathbf{v}\cdot\nabla\phi\Big)$. We follow Danilov and Timmermann [15] and neglect the term $\operatorname{div}(\partial_t \mathbf{v}A)$ as the variation in the velocity is small in time. We conclude with the approximative time stepping scheme

$$A_n = A_{n-1} - k\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}) + \frac{k^2}{2}\operatorname{div}\ (\mathbf{v}_{n-1}\operatorname{div}(\mathbf{v}_{n-1}A_{n-1})).$$

.

For $n = 1, ..., N$ the variational formulation is given as

$$\begin{aligned}
A_n \in \mathcal{V}^A :\ (A_n, \phi) =& (A_{n-1}, \phi) \\
&- k(\operatorname{div}(\mathbf{v}_{n-1}A_{n-1}), \phi) - \frac{k^2}{2}(\operatorname{div}(\mathbf{v}_{n-1}\nabla A_{n-1}), \mathbf{v}_{n-1}\cdot\nabla\phi) \quad \forall\phi \in \mathcal{L}.
\end{aligned} \tag{3.7}$$

After every time step, the new solution A_n is projected onto the set of admissible functions, such that $A_n : \Omega \to [0, 1]$. The discretization of the balance equations is completely explicit. For stability reasons, small time steps are required. Therefore, we apply a subcycling procedure, such that every time step $I_n = [t_{n-1}, t_n]$ is further split into

$$t_{n-1} = t_{n-1}^0 < t_{n-1}^1 < \cdots < t_{n-1}^S = t_n, \quad k_S = \frac{k}{S},$$

where $S \in \mathbb{N}$. To reach stability, we usually take $S = 20$ and get

$$A_{n-1} = A_{n-1}^{(0)} \mapsto A_{n-1}^{(1)} \mapsto \cdots \mapsto A_{n-1}^{(S)} = A_n.$$

Each sub-step is convected with the velocity prediction

$$\mathbf{v}_{n-1}^{(l)} = \frac{S-l}{S}\mathbf{v}_{n-1} + \frac{l}{S}\mathbf{v}_n^{ex}, \quad l = 1, ..., S, \tag{3.8}$$

$$\mathbf{v}_n^{ex} := 2\mathbf{v}_{n-1} - \mathbf{v}_{n-2}. \tag{3.9}$$

Discretization in space Analogous to the spatial discretization of the ice velocity in Section 3.2, the finite element spaces of the ice concentration is

$$V_h^A := \mathcal{V}^A \cap V_h^{(r)}, \quad r = 1, 2.$$

The fully discretized balance law is then given as

$$\begin{aligned} A_h^n \in V_h^A : \quad (A_{n,h}, \phi) = (A_{n-1,h}, \phi_h) - k(\operatorname{div}(\mathbf{v}_{n-1,h}A_{n-1,h}), \phi_h) \\ - \frac{k^2}{2}(\operatorname{div}(\mathbf{v}_{n-1,h}A_{n-1,h}), \mathbf{v}_{n-1,h} \cdot \nabla\phi_h) \quad \forall \phi_h \in V_h. \end{aligned} \tag{3.10}$$

To complete the spatial discretization, we choose the nodal basis as

$$V_h^A = \operatorname{span}\{\phi_A^k, \quad k = 1, ..., M_A\}$$

and rewrite the solution of (3.7) as

$$A_{n,h} = \sum_{k=1}^{N} A_n^k \boldsymbol{\phi}_{A_{n,h}}^k,$$

where A_n^k is a scalar. The ice thickness is discretized analogously.

3.3.2. Stabilization techniques for the balance laws

In this section, we compare different spatial stabilization techniques for the balance laws (2.29). As we considered two different types of time discretization in Section 3.3.1, we present distinct stabilization approaches. The θ-scheme is stabilized with residual based methods. In particular, we present *artificial diffusion* and the *streamline upwind Petrov-Galerkin* scheme. The Taylor-Galerkin method on the other hand is stabilized with a *flux limiter*. A detailed introduction into stabilization methods can be found in the books of Johnson [37] and Kuzmin [48].

We formulate the balance law discretized with a θ-scheme in time and finite elements in space as

$$\Big(A_{n,h}, \phi_h\Big) - \theta B(A_{n,h}, \phi_h) = \Big(A_{n-1,h}, \phi_h\Big) + (1-\theta)B(A_{n-1,h}, \phi_h), \tag{3.11}$$

with

$$B(A_{n,h}, \phi_h) := \Big(\mathbf{v}_{n,h} \cdot \nabla A_{n,h} + A_{n,h} \operatorname{div}(\mathbf{v}_{n,h}), \phi_h\Big).$$

The ice thickness is discretized analogously.

Artificial diffusion A classical way to stabilize the bilinearform $B(A_h, \phi_h)$ is to add artificial diffusion. We introduce

$$\tilde{B}(A_{n,h}, \phi_h) = B(A_{n,h}, \phi_h) + \underbrace{\gamma_c |\mathbf{v}_{n,h}| \frac{h}{2} (\nabla A_{n,h}, \nabla \phi_h)}_{:= S_h(A_{n,h}, \phi_h)},$$

where γ_c is a positive constant and h the mesh size. This technique is non-consistent. Due to the presence of the stabilization term, the approximation of the solution is only first order accurate.

Streamline-upwind Petrov-Galerkin method (SUPG) The idea of the streamline-upwind Petrov-Galerkin method is to add a diffusive term in direction of the streamlines. Following Johnson [37], this is done by a modification in the test function

$$\tilde{\phi}_h = \phi_h + \delta_s (\mathbf{v}_h \cdot \nabla \phi_h).$$

The stabilized balance law then read as

$$\left(A_{n,h}, \tilde{\phi}_h\right) - \theta k B(A_{n,h}, \tilde{\phi}_h) = \left(A_{n-1,h}, \tilde{\phi}_h\right) + (1-\theta) k B(A_{n-1,h}, \tilde{\phi}_h). \tag{3.12}$$

The streamline-upwind Petrov-Galerkin method is consistent as

$$\left(\partial_t A, \phi + \delta_s (\mathbf{v} \cdot \nabla \phi)\right) + B(A, \phi + \delta_s (\mathbf{v} \cdot \nabla \phi)) = 0, \quad \forall \phi \in \mathcal{L}.$$

We follow John et al. [36] and choose

$$\delta_s = \delta_d \min \left(\frac{h}{2\theta k \|\mathbf{v}_h\|}, \frac{1}{(1+\theta k) \| \operatorname{div}(\mathbf{v}) A \|} \right), \tag{3.13}$$

with the time step size k, the mesh size h and $\theta = 1$ for the implicit Euler scheme, and $\theta = \frac{1}{2}$ for the Crank-Nicolson method. We simplified (3.13) to

$$\delta_s = \delta_d \left(\frac{k 2\theta \|\mathbf{v}_h\|}{h} + (1+\theta k) \| \operatorname{div}(\mathbf{v}) A \| \right)^{-1}, \tag{3.14}$$

where δ_d is a positive constant.

Comparing the SUPG stabilization with the Taylor-Galerkin discretization derived in Section 3.3.1, we find that the stabilizing second order term of the Taylor-Galerkin discretization,

$$\frac{k^2}{2} (\mathbf{v} \cdot \nabla A, \mathbf{v} \cdot \nabla \phi),$$

corresponds to the stabilizing part of the SUPG approach,

$$\delta_s (\mathbf{v} \cdot \nabla A, \mathbf{v} \cdot \nabla \phi), \tag{3.15}$$

with the difference that the parameter δ_s of the SUPG method depends on the mesh size h, whereas the stabilizing term of the Taylor-Galerkin scheme is related to the time step size k.

Flux limiter We apply a flux limiter to stabilize the Taylor-Galerkin discretization introduced in Section 3.3.1. The idea of flux correction is to combine a stable, but not accurate *low-order solution* A^{low} with a possibly unstable, but accurate *high-order solution* A^{high}. The low-order solution is computed with *row-sum mass lumping*. The diffusive properties of *row-sum mass lumping* adds additional dissipation close to discontinuities and sharp edges [47]. A detailed introduction into flux-limited Taylor-Galerkin methods is given in the book of Kuzmin [48].

First, we define the *high-order solution* as the pure Taylor-Galerkin approximation of (3.7)

$$\mathbf{M}A_n^{\text{high}} = \mathbf{M}A_{n-1} + k\mathbf{B}(\mathbf{v}_{n-1})A_{n-1}, \tag{3.16}$$

$\mathbf{M}$ denotes the mass matrix and $\mathbf{B}(\mathbf{v})$ the stiffness matrix

$$\mathbf{M} = (m_{ij})_{i,j=1}^{N}, \quad \mathbf{B}(\mathbf{v}) = (b_{ij}(\mathbf{v}))_{i,j=1}^{N},$$

with

$$m_{ij} = (\phi_h^j, \phi_h^i), \quad b_{ij}(\mathbf{v}) = -(\text{div}(\mathbf{v}\phi_h^j), \phi_h^i) - \frac{k}{2}(\mathbf{v}\cdot\nabla\phi_h^j + \text{div}(\mathbf{v})\phi_h^j, \mathbf{v}\cdot\nabla\phi_h^i).$$

The solution of equation (3.16) also fulfills

$$\mathbf{M}^L A_n^{\text{high}} = \mathbf{M}^L A_{n-1} + (\mathbf{M}^L - \mathbf{M})(A_n^{\text{high}} - A_{n-1}) + k\mathbf{B}(\mathbf{v}_{n-1})A_{n-1}, \tag{3.17}$$

where the lumped version of the mass matrix is given as

$$\mathbf{M}^L = \text{diag}(m_i^L), \quad m_i^L = \sum_{j=1}^{N} m_{ij}.$$

Then, we define the *low-order solution* by means of mass lumping through

$$\mathbf{M}^L A_n^{\text{low}} = \mathbf{M}A_{n-1} + k\mathbf{B}(\mathbf{v}_{n-1})A_{n-1}. \tag{3.18}$$

This low-order solution is smooth and stable, but suffers from reduced accuracy as the right hand side is still evaluated with the consistent mass matrix $\mathbf{M}$. The low-order scheme can be written as

$$\mathbf{M}^L A_n^{\text{low}} = \mathbf{M}^L A_{n-1} + k\mathbf{B}(\mathbf{v}_{n-1})A_{n-1} + (\mathbf{M} - \mathbf{M}^L)A_{n-1}. \tag{3.19}$$

The difference between the *high-order* (3.17) and the *low-order* scheme (3.19) is given by

$$\mathbf{M}^L(A_n^{\text{high}} - A_n^{\text{low}}) = (\mathbf{M}^L - \mathbf{M})A_n^{\text{high}}. \tag{3.20}$$

The *high-order solution* can be written in terms of the *low-order solution* as

$$\mathbf{M}^L A_n^{\text{high}} = \mathbf{M}^L A_n^{\text{low}} + (\mathbf{M}^L - \mathbf{M})A_n^{\text{high}}. \tag{3.21}$$

For ease of notation we denote $\tilde{A} := A_n^{\text{high}}$. The anti-diffusive part, $\mathbf{M}^L - \mathbf{M}$, in the right hand side can be formulated as

$$-[(\mathbf{M}-\mathbf{M}^L)\tilde{A}]_i = -(\sum_j m_{ij}\tilde{A}_j - m_i\tilde{A}_i) = -(\sum_{j\neq i} m_{ij}(\tilde{A}_j - \tilde{A}_i)) = \sum_{j\neq i} m_{ij}(\tilde{A}_i - \tilde{A}_j). \tag{3.22}$$

The idea of flux correction is to regulate the amount of anti-diffusion $f_{i,j} = m_{i,j}(\tilde{A}_i - \tilde{A}_j)$ by a parameter α_{ij} which is added to the *low-order* solution. This limiting is applied node-wise as

$$m_i^L A_{n,i} = m_i^L A_{n,i}^{\text{low}} + \sum_{j=1}^{N} \alpha_{ij} m_{ij}(\tilde{A}_i - \tilde{A}_j). \tag{3.23}$$

For $\alpha_{ij} = 1$, we reconstruct the *high-order* solution, $\alpha_{ij} = 0$ corresponds to the *low-order* solution. The solution constructed with the flux limiter can be seen as a limited correction of A_n^{low} towards A_n^{high}. The coefficients $\alpha_{ij} = \alpha_{ji} \in \mathbb{R}$ are chosen such that no new local minima $A_i^{\min}$ and maxima $A_i^{\max}$ in node i are generated. At time step n we define

$$A_i^{\min} = \min_{j\in S_i} A_j^{\text{low}}, \quad A_i^{\max} = \max_{j\in S_i} A_j^{\text{low}},$$

where $S_i = (j \neq i | m_{ij} \neq 0)$ is the stencil of node i.

Based on Zalesak's multidimensional flux correction algorithm [110] we compute $\alpha_{i,j}$ from equation (3.23) as given in Algorithm 3. The presentation follows [46].

Algorithm 3 (Zalesak's multidimensional flux correction algorithm)**.**

1. *Compute the sum of positive and negative anti-diffusive fluxes $f_{ij} = m_{ij}(\tilde{A}_i - \tilde{A}_j)$ that contribute to node i:*

$$P_i^+ = \sum_{i\neq j} \max\{0, f_{i,j}\}, \quad P_i^- = \sum_{i\neq j} \min\{0, f_{i,j}\}.$$

2. *Compute the maximal admissible increment Q_i^+ and minimal admissible increment Q_i^- for node i:*

$$Q_i^+ = A_i^{\max} - A_{n,i}^{\text{low}}, \quad Q_i^- = A_i^{\min} - A_{n,i}^{\text{low}}.$$

3. *Bound the correction factors to guarantee that there are no new over- and undershoots generated at node i:*

$$R_i^+ = \begin{cases} \min\{1, \frac{Q_i^+}{P_i^+}\}, & \text{if} \quad P_i^+ \neq 0, \\ 0, & \text{else}, \end{cases}$$

$$R_i^- = \begin{cases} \min\{1, \frac{Q_i^-}{P_i^-}\}, & \text{if} \quad P_i^- \neq 0, \\ 0, & \text{else}. \end{cases}$$

4. Define a suitable limiter by

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if} \quad f_{i,j} \geq 0, \\ \min\{R_i^-, R_j^+\}, & \text{if} \quad f_{i,j} < 0. \end{cases}$$

Numerical implementation The numerical realization of this limiting approach is very efficient. We need to solve two problems

$$\mathbf{M}^L A_n^{\text{low}} = \mathbf{b}_1, \quad \mathbf{M} A_n^{\text{high}} = \mathbf{b}_2,$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the right hand side vectors as given by (3.16) and (3.18) respectively. The *low-order* solution is simply given by diagonal scaling of the right hand side. For computing the *high-order* solution, we employ about 10 to 20 steps of a *Conjugate Gradient* method (CG), preconditioned with the lumped mass matrix $\mathbf{M}^L$. In the CG iterations, we set the global tolerance to 10^{-13} and prescribe a relative residual reduction of 10^{-4}. It is noted in Danilov et al. [15], that 3 steps of a simple Richardson iteration, preconditioned with $\mathbf{M}^L$, should be also sufficient,

$$\mathbf{M}_L^{-1} \mathbf{M} A_n^{\text{high}} = \mathbf{M}_L^{-1} \mathbf{b}. \tag{3.24}$$

Even with intense sub-cycling, e.g. doing 20 sub-steps in each time step, the cost for approximation of the balance laws is not dominant compared to solving the nonlinear momentum equation. To enhance the stability of the nonlinear solver we use the *low-order* solution of A and H to calculate the velocity field, while the solution of the balance laws is calculated with the flux correction scheme detailed above. This *discretized partitioned solution approach* is summarized in Algorithm 4.

Algorithm 4 (Discretized Partitioned Solution Approach)**.** *Let $\mathbf{v}_0, A_0, H_0$ be the initial solutions at time $t = 0$. Iterate for $n = 1, 2, \ldots, N$:*

1. *Solve the balance laws with $\mathbf{v}_n^{ex} := 2\mathbf{v}_{n-1} - \mathbf{v}_{n-2}$ and Algorithm 3*

$$A_{n-1}^{\text{flux}} \mapsto A_n^{\text{flux}}, \quad H_{n-1}^{\text{flux}} \mapsto H_n^{\text{flux}},$$

2. *Solve the momentum equation based on the low-order solutions $A_n^{\text{low}}, H_n^{\text{low}}$ as calculated in the flux correction framework*

$$\mathbf{v}_{n-1} \mapsto \mathbf{v}_n.$$

Due to the decoupling in Algorithm 4 the new velocity $\mathbf{v}_n$ is not at hand when solving the balance laws (2.29). We observed that the nonlinear solver is less stable if we solve the balance laws with extrapolation of the velocity $\mathbf{v}^{ex} = 2\mathbf{v}_{n-1} - \mathbf{v}_{n-2}$, instead of using the linear extrapolation $\mathbf{v}^{ex} = \mathbf{v}_{n-1}$. It has shown to be essential to couple the low-order solutions $A_n^{\text{low}}, H_n^{\text{low}}$ into the momentum equation for obtaining a robust approximation of the momentum equation. It is a subject to future work to analyze the impact on the accuracy of the velocity field. In Section 4, we visualize the low-order, high-order and flux-limited solution of an idealized test case.

4. Numerical benchmarking of different transport stabilizations for the balance laws

In this section, we analyze the convergence and the robustness of different numerical solutions of the sea ice model (2.29) using the different discretizations of the balance laws that were introduced in Section 3.3. In Section 4.1, we analyze the convergence of the numerical solution of a simple linear transport problem. In Section 4.2, we solve the sea ice model problem that has been introduced in Section 2.6 and analyze the convergence of the numerical solution.

4.1. A body rotation problem

In this section, we evaluate a *body rotation problem*, a linear transport problem as described by John et al. [36], where a quantity $A(t,x,y) \in \mathbb{R}^2$ is transported with the velocity $\mathbf{v}(t,x,y) \in \mathbb{R}^2$. The motion is modeled by

$$\partial_t A + \operatorname{div}(\mathbf{v}A) = 0, \tag{4.1}$$

where the stationary velocity field is given as

$$\mathbf{v}(x,y) = 2\pi \begin{pmatrix} -y \\ x \end{pmatrix}. \tag{4.2}$$

We consider (4.1) in the domain $\Omega = \{(x,y) \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1\}$ with Dirichlet zero conditions on the boundary. As shown in Figure 4.1, the initial condition of $A(0,x,y)$ consists of three different bodies: a *box*, a *hump* and a *cone*. The bodies are rotated in a circle with radius $r = 1.5$. The position of a body is described by its center (x_0, y_0). Outside of the bodies the initial condition is zero. The bodies are rotated in counterclockwise direction. One full revolution is achieved for $T = 1$. The distance from the midpoint to the center of a body is defined as

$$d(x,y) := \Big((x-x_0)^2 + (y-y_0)^2\Big)^{\frac{1}{2}}.$$

We start by presenting the initial position of the first body, the *box*, which is described by

$$A(0,x,y) = \begin{cases} 0.5, & \text{if} \quad \|(x,y) - (x_0,y_0)\|_\infty < 0.2, \\ 0, & \text{else}, \end{cases}$$

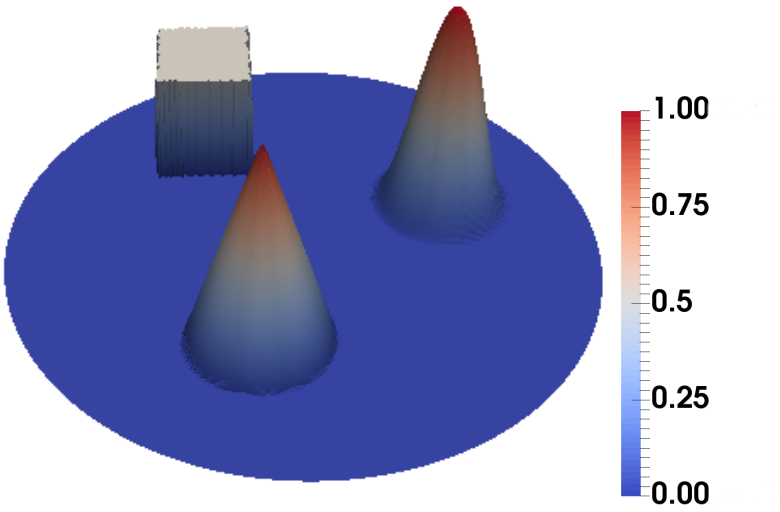


Figure 4.1.: Initial condition of the rotating body problem

with initial center at $(x_0, y_0) = (-0.4, 0.7)$.

The second body, the *hump*, is given by

$$A(0,x,y) = \begin{cases} 0.5 + 0.5\cos(\pi d/0.3) & \text{if} \quad d(x,y) < 0.3, \\ 0 & \text{else}, \end{cases}$$

where the center of the *hump* is given by $(x_0, y_0) = (0.6, 0.3)$.

The third body is the cone, which is described by

$$A(0,x,y) = \begin{cases} 1.0 - d/0.3 & \text{if} \quad d(x,y) < 0.3, \\ 0 & \text{else}, \end{cases}$$

with the initial center at $(x_0, y_0) = (-0.2, -0.5)$.

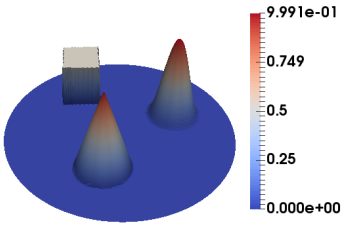
The problem is discretized with bi-linear finite elements in space. For the temporal discretization we apply the *implicit Euler* scheme, the *Crank-Nicolson* method and a second order explicit *Tayor-Galerkin* method. Depending on the temporal discretization, we stabilize the discretized balance laws 4.1 with *artificial diffusion*

$$S_{ad}(A_{n,h}, \phi_{n,h}) = |\mathbf{v}_{n,h}| \frac{h}{2} \Big(\nabla A_{n,h}, \nabla \phi_h \Big), \tag{4.3}$$

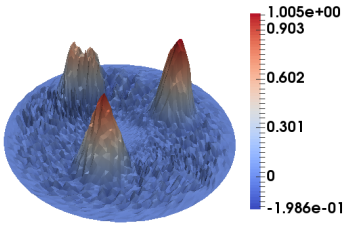
the *streamline-upwind Petrov-Galerkin* method

$$S_{SUPG}(A_{n,h}, \phi_{n,h}) = (\delta_d R^{-1} \Big(\operatorname{div}(\mathbf{v}_{n,h} A_{n,h}), \mathbf{v}_h \cdot \nabla \phi_h \Big), \tag{4.4}$$

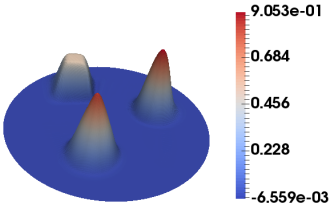
with $R = \frac{k2\theta\|\mathbf{v}_h\|}{h} + (1 + \theta k)\|\operatorname{div}(\mathbf{v})A\|$ and a *flux correction scheme* based on Zalesak's limiter [110]. The different discretization and stabilization schemes are introduced in Chapter 3.3.2.



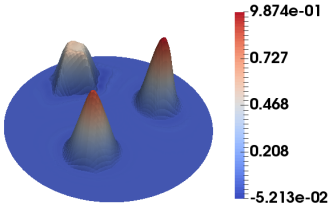
Initial condition



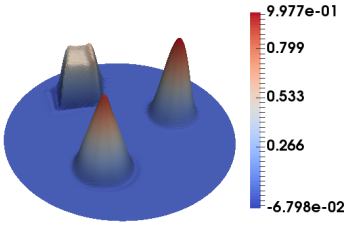
CN solution without stabilization



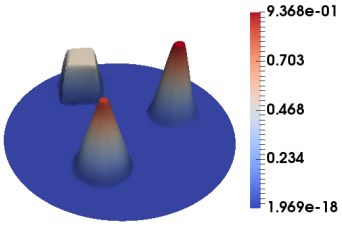
CN with artificial diffusion using $\gamma_c = 0.6$ (4.3)



CN with SUPG using $\delta_d = 0.06$ (4.4)

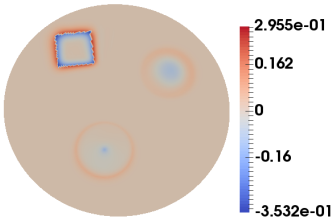


TG

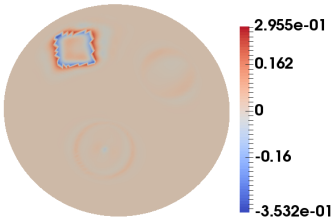


TG with flux limiter

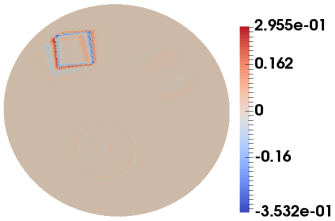
Figure 4.2.: The numerical solution of the rotated body problem after one revolution, by applying different stabilization techniques. We stabilize the system with artificial diffusion, the streamline-upwind Petrov-Galerkin (SUPG) method or a flux-corrected transport algorithm. For time discretization we use a Crank-Nicolson scheme (CN) or a second order explicit Tayor-Galerkin scheme (TG).



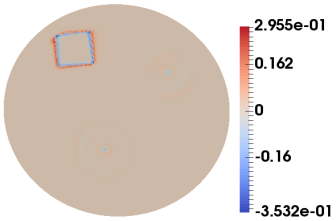
CN with artificial diffusion using $\gamma_c = 0.6$



CN with SUPG using $\delta_d = 0.06$



TG without stabilization



TG with flux-corrected transport

Figure 4.3.: We calculate the difference of the approximation and the initial condition after one revolution of the rotated body problem. The system is stabilized with artificial diffusion, a streamline-upwind Petrov-Galerkin (SUPG) method or a flux-corrected transport algorithm. For the temporal discretization we use a Crank-Nicolson scheme (CN) or a second order explicit Tayor-Galerkin method (TG).

Stabilization method	e_h^2
CN with artificial diffusion	0.00444047
CN with SUPG	0.00244914
TG	0.00170643
TG with flux-corrected transport	0.00152172

Table 4.1.: We show the L^2-error e_h^2 of the difference between the initial data and the approximation of the solution after one revolution of the rotated body problem. For the temporal discretization, a Crank-Nicolson scheme (CN) or an explicit second order Taylor-Galerkin scheme (TG) is applied.

We aim to choose a stabilization such that the approximation of the solution after one revolution is very close to the initial condition. We use a grid with 5185 nodes and a time step of $k = 0.001$.

In Figure 4.2, we present the numerical solutions after one revolution using the different stabilizations. In the upper row of Figure 4.2, we show the initial condition of the rotated body problem and the solution discretized with a Crank-Nicolson scheme without stabilization. As the Crank-Nicolson shows instabilities in space, we add a stabilization to the balance laws by using either artificial diffusion or the SUPG method. Considering all three bodies, we observe that artificial diffusion is more damping than the SUPG stabilization, whereas the approximation with SUPG method still contains small oscillations.

We compare these two residual based techniques to a second order Taylor-Galerkin method (with and without flux correction). The temporal discretization with the Taylor-Galerkin method enables a sharper representation of the box than using a Crank-Nicolson scheme with spatial stabilization. However, small oscillations still appear. Thus, we add Zalesak's flux limiter described in [110] to the Taylor-Galerkin scheme and observe that the overshoots vanish, the peaks of the cone and the hump are damped by the flux limiter.

Figure 4.3 shows, the difference between the numerical solution after one revolution at $T = 1$ and the initial condition for each stabilization method. We obtain the sharpest contours for all three bodies with the flux-limited Taylor-Galerkin scheme. Even so, the peaks of the cone and the hump are damped. The cone and the hump are maintained best with a Tayler-Galerkin method without flux correction, compare to Figure 4.3 bottom row. In Table 4.1, we present the square error of the L^2-error

$$e_h^2 := \int_\Omega \Big(A(1,x,y) - A(0,x,y)\Big)^2 \, d(x,y).$$

The smallest L^2-error is obtained with a second order explicit Taylor-Galerkin scheme with a flux-limited transport, followed by a Taylor-Galerkin method without stabilization, and the Crank-Nicolson method with a SUPG stabilization. By using artificial diffusion we observe the largest error. Note that by using the implicit Euler discretiza-

tion in time, the numerical solution is diffusive such that no additional stabilization is required.

4.2. Application to the sea ice model

Based on the results from Section 4.1, we analyze the impact of the SUPG stabilization and the flux correction scheme on the numerical solution of the coupled sea ice model. Furthermore, we investigate the convergence of the numerical solutions by analyzing two functionals under spatial and temporal refinement. Finally, we discuss a defect called *terracing* that appears in the approximation of the ice thickness and the ice concentration under the application of the flux limiter.

Numerical solution of the coupled sea ice model with different stabilization methods
We discuss approximation achieved with the implicit Euler scheme, the Crank-Nicolson scheme stabilized with a SUPG method and a flux-corrected Taylor-Galerkin scheme, as it is introduced in Section 3.3.2. We analyze the implicit Euler method as it provides without spatial stabilization a robust but diffusive approximation of the body rotation problem in Section 4.1.

With the different schemes, we computed the solution of the model problem defined in Section 2.6. Snapshots of the calculated ice concentration after iteration 172, nearly $T = 4$ days of simulation are shown in Figure 4.4. The solution is computed with $k = 0.5$ h on 8 km, 4 km and 2 km meshes.

Analyzing Figure 4.4, we observe that the approximation with the implicit Euler scheme is more diffusive than the approximations with the other schemes. The diffusion can be explained with the damping property of the implicit Euler scheme that is described in Section 3.2. Applying an implicit Euler discretization without stabilization, the nonlinear solver which is applied to compute the velocity field $\mathbf{v}$ from the momentum equation failed on a 2 km mesh. Failing means that the nonlinear solver could not reduce the residual of the momentum equation to the required tolerance. Hence, the simulation was terminated. Note that the implicit Euler scheme with SUPG stabilization provides a robust approximation. In comparison to the approximation achieved with the implicit Euler scheme, we find that the solution computed with the Crank-Nicolson scheme stabilized with a SUPG method is less diffusive and contains more structure. On the other hand, we observe that the ice concentration achieved with the stabilized Crank-Nicolson method strongly differs from the ice concentration calculated with the flux-limited Taylor-Galerkin scheme. The Crank-Nicolson method combined with the SUPG stabilization conserves the structure of the initial condition of the ice thickness, whereas the structure is vanishing with the flux-limited Taylor-Galerkin scheme. However, the solution computed with the flux-corrected Taylor-Galerkin method contains more cracks in the ice cover.

In Figure 4.5, we show the ice concentration at $T = 8$ days of simulation. We observe a huge difference between the approximation computed with the stabilized Crank-Nicolson scheme and the solution calculated with the flux-limited Taylor-Galerkin method. Ap-

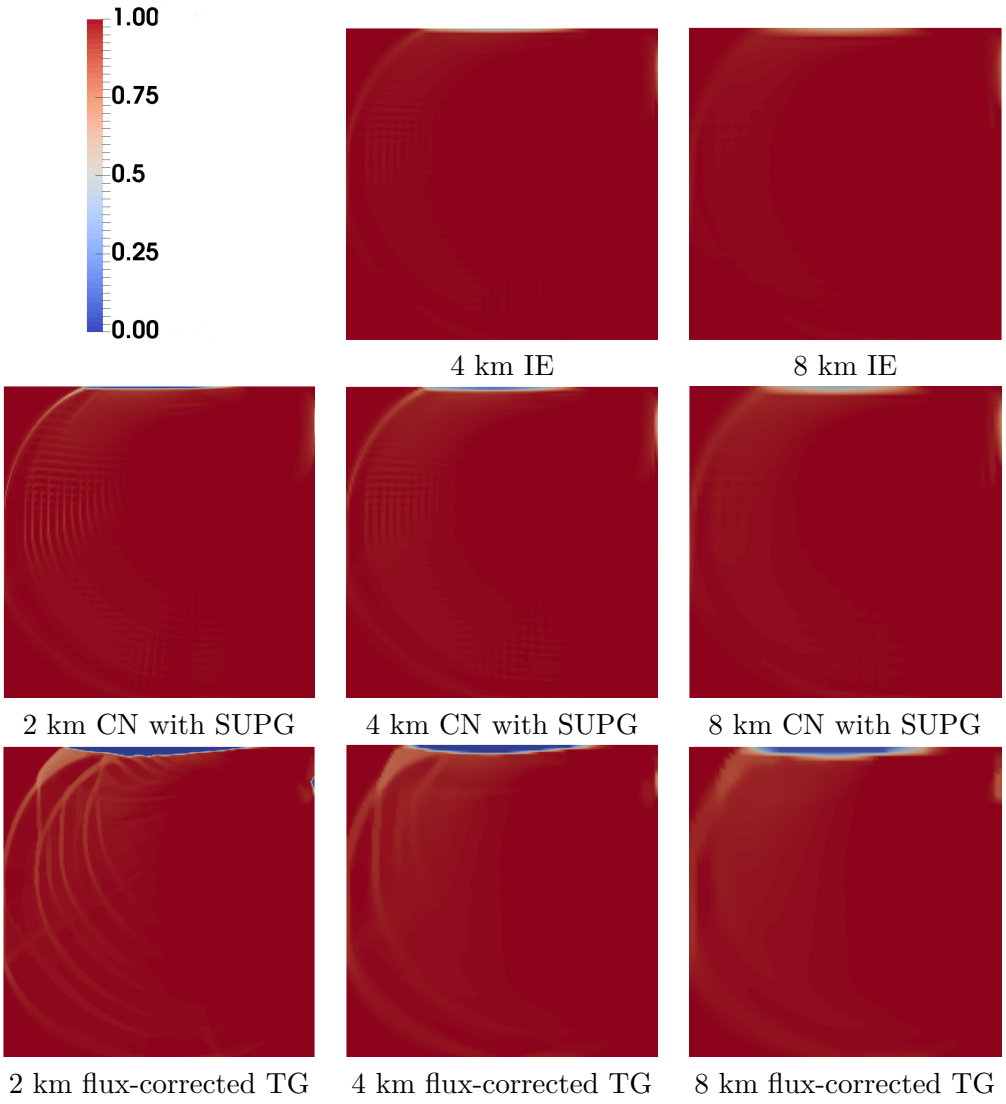
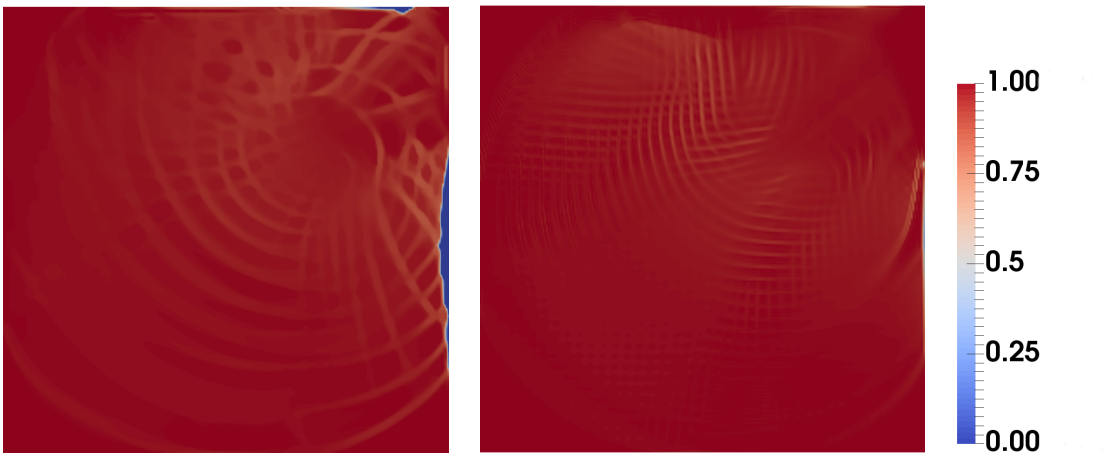


Figure 4.4.: We solve the model problem of Section 2.6 and show snapshots of the ice concentration A at iteration $n = 172$, after nearly $T = 4$ days of simulation. The solution is calculated with an explicit second order flux-corrected Taylor-Galerkin scheme (TG), the Crank-Nicolson scheme (CN) stabilized with the SUPG method, and the implicit Euler scheme (IE) without stabilization. Using the latter, the nonlinear solver failed on a 2 km mesh.



2 km flux-corrected TG 2 km CN with SUPG

Figure 4.5.: We solve the model problem of Section 2.6 on a 2 km mesh and show snapshots of the ice concentration A at iteration $n = 330$. The solution is calculated with an explicit second order flux-corrected Taylor-Galerkin scheme (TG) and with the Crank-Nicolson scheme (CN) stabilized with the SUPG method.

plying a Crank-Nicolson scheme smaller cracks and lesser open water areas occur in comparison to a discretization with a flux-limited Taylor Galerkin scheme.

Besides the visual differences in the solution, these two discretization approaches have a different impact on the performance of the nonlinear solver which is used to solve the sea ice momentum equation (2.29). Applying the flux-corrected Taylor-Galerkin scheme, the nonlinear solver failed 6 times in one simulation on a 2 km mesh, whereas no failures of the nonlinear solver occur with a stabilized Crank-Nicolson method in time. This might be due to the fact that less structure is developing in the approximation of the balance laws when a Crank-Nicolson scheme is applied.

Convergence of the numerical solution of the sea ice model In order to analyze the convergence of the numerical solution of the sea ice model, we evaluate two different functionals under mesh refinement. Similar to the comparison of satellite data and model simulations, e.g. in [52, 62], we compare the total sea ice deformation $|\dot{\boldsymbol{\epsilon}}|$ and the total sea ice extent A_{total}.

We choose $I = [0, 4]$ days and $\Omega = (0, 500) \times (0, 500)\,\text{km}^2$. The space and time averaged total sea ice deformation is given as

$$|\dot{\boldsymbol{\epsilon}}| = J_{\boldsymbol{\epsilon}}(\dot{\boldsymbol{\epsilon}}) = \int_I \int_\Omega \left((\dot{\boldsymbol{\epsilon}}_I)^2 + (\dot{\boldsymbol{\epsilon}}_{II})^2 \right)^{\frac{1}{2}} d(x, y)\, dt.$$

h	$k = 1$ h	$k = 0.5$ h	$k = 0.25$ h	q_k
16 km	0.13829	0.13867	0.13885	1
8 km	0.14120	0.14161	0.14180	1
4 km	0.14392	0.14433	0.14455	2
2 km	0.14815	0.14770	0.14759	2

Table 4.2.: We analyze the space and time averaged total ice deformation $|\dot{\boldsymbol{\epsilon}}|$ under refinement. By h we denote the mesh size by k the time steps size and by q_k the convergence order in time.

h	$k = 1$h	$k = 0.5$h	$k = 0.25$h	q_h
16 km	1.01257	1.01235	1.01216	2
8 km	1.00669	1.00645	1.00622	2
4 km	1.00918	1.00888	1.00871	2
2 km	1.00969	1.00943	1.00923	2

Table 4.3.: We analyze the space and time averaged total ice extent A_{total} under refinement. By h we denote the number of nodes, by k the time steps size, and by q_h the convergence order in space.

The total ice extent is averaged over a subdomain $\Omega_1 = (375, 500) \times (375, 500)\,\text{km}^2$ and over a time span of $I_1 = [3.125, 4]$ days.

$$A_{\text{total}} = J_A(A) = \int_{I_1} \int_{\Omega_1} \chi_{0.15}(A) A \, d(x, y) dt, \quad \text{if} \quad \chi_{0.15}(A) := \begin{cases} 1, & A(x, y) \geq 0.15, \\ 0, & A(x, y) < 0.15. \end{cases}$$

We evaluate the functional on mesh resolutions decreasing from 16 km to 2 km and time steps varying from $k = 1$ h to $k = 0.25$ h. To enhance the stability of the solver we slightly modify the initial condition of the model problem described in Section 2.6, by removing the oscillations in H and decreasing A to

$$H^0 = 0.3, \quad A^0 = 0.8. \tag{4.5}$$

The convergence order in space is calculated by the following approach:

$$\begin{aligned} J(h) &= a + C, \\ J(h/2) &= a + C \left(\frac{1}{2}\right)^q, \\ J(h/4) &= a + C \left(\frac{1}{4}\right)^q, \end{aligned} \tag{4.6}$$

where a is the limit, C is a constant, and q is the convergence order. The mesh size is

denoted by h. By reformulating system (4.6), the convergence order is given as

$$q = \frac{-\ln\left(\frac{J(h/2) - J(h/4))}{J(h) - J(h/2)}\right)}{\ln(2)}. \tag{4.7}$$

Like the convergence order in space, we calculate the convergence order in time.

We present the values for the total ice deformation $|\dot{\boldsymbol{\epsilon}}|$ in Table 4.2. On 2 km and 4 km meshes, we observe quadratic convergence in time. On 8 km and 16 km meshes, the convergence in time is linear. In space, the functional value is increasing on finer meshes. However, we could not observe a clear convergence order by approach (4.6) at higher spatial resolutions.

In Table 4.3, we present the total ice extent A_{total} in the time span of interest. We observe convergence as the value of the functional is increasing for smaller time steps. We do not observe a clear convergence order by (4.7). In space we observe nearly quadratic convergence. This is expected as we use linear finite elements for spatial discretization.

The impact of the spatial discretization in both functionals dominates the impact of the temporal discretization. Thus, the spatial refinement has a bigger influence on the approximation compared to the temporal refinement. We observed the same behavior in our analysis in [68].

Terracing effects of the flux-limited Taylor-Galerkin method We analyze the test case presented in [68]. As in the model problem described in Section 2.6, we consider a stationary circular clockwise ocean current. The wind pushes the ice with a constant force to the upper right corner of the domain. As shown in Figure 4.6, the ice thickness decreases behind the island due to the circular ocean current. The wind is the dominant external force in this example. After about 81 days of simulation, we observe a defect in the approximation of the ice thickness and the ice concentration, called *terracing*. Small *terraces* in the length of a grid cell appear in the upper right corner of the domain.

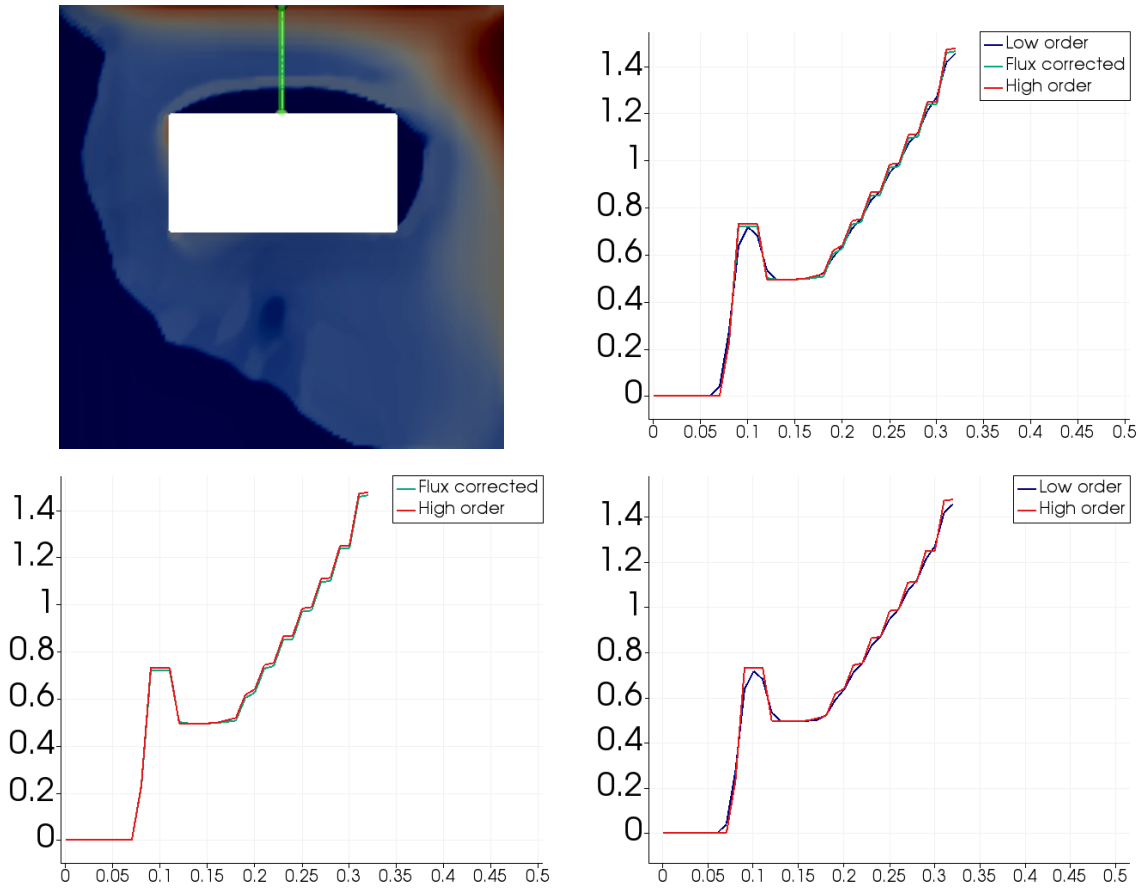


Figure 4.6.: We plot the ice thickness H along the green line marked in the upper left image. In the surrounding three graphs, we present the *low-order*, the *high-order* and the *flux-corrected* approximation. On the horizontal axis, we show the grid points in y-direction. On the vertical axis, we present the ice thickness in meters.

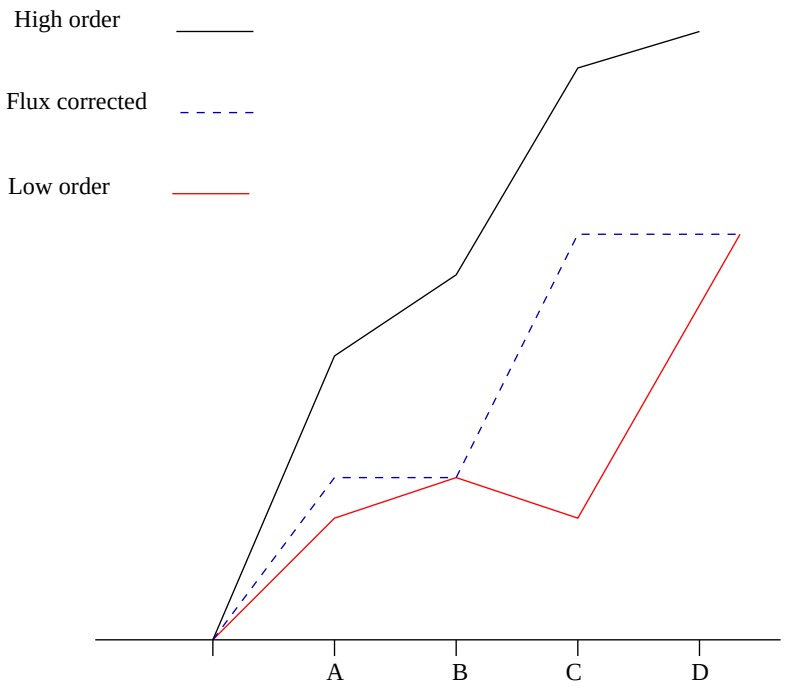


Figure 4.7.: Illustration of the terracing effect.

For a better visualization of the defect, we plot the ice thickness along the line shown in the Figure 4.6. In the upper right plot of Figure 4.6, we present the *high-order*, the *low-order* and the *flux-corrected solution* as described in Section 3.3.2. In the lower right graph, we observe that the low-order solution does not contain any terraces. In the lower left graph, we see that the high-order solution based on the flux-corrected approximation from the previous time step already includes terraces. The flux correction described in Algorithm 3 in Section 3.3.2 limits the flux $f_{i,j}$ to the local extremum. By doing so terraces might occur.

Those effect is visualized in Figure 4.7. Here, Algorithm 3 limits the maximal allowed increment of point A and B by the height of B, which is the local maximum, such that a terrace is built. The same argumentation holds for points C and D, where D is the local maximum. Here, a second terrace is occurring. It needs to be investigated if this effect can be avoided by modifying the limitation in Algorithm 3 in Section 3.3.2. Note that the *high-order* solution calculated without flux limiting is oscillating and the *low-order* solution is smeared out such that all the structures in the approximation are vanishing.

However, by using Algorithm 3, where we use the intermediate low-order solutions $A_n^{\text{low}}, H_n^{\text{low}}$ to couple into the momentum equation, the negative effect of *terraces* in solving the sea ice model is circumvented.

4.3. Conclusion

In this section, we have solved the balance laws which are part of the sea ice model (2.29) using different time discretizations and varying spatial stabilizations. First, we analyzed the impact that different discretization schemes have on the computed solution of a body rotation problem. Here, we observed the best performance with the flux-corrected Taylor-Galerkin discretization scheme and the Crank-Nicolson scheme stabilized with a SUPG method.

In the case of the sea ice model, the different discretizations show similar behavior as in the body rotation problem. We analyzed the model problem presented in Section 2.6 and observed that the implicit Euler scheme is more damping than the other time stepping schemes. A discretization with the Crank-Nicolson method and the Taylor-Galerkin scheme produced oscillations in space. Even the approximation with the implicit Euler method required spatial stabilization. We observed robust approximations with the implicit Euler method, the Crank-Nicolson scheme stabilized with a SUPG method, and a flux-limited Taylor-Galerkin discretization.

We find that the approximation of the ice concentration achieved with the stabilized Crank-Nicolson scheme differs a lot from the approximation conducted with the flux-limited Taylor-Galerkin method. The latter contains bigger cracks and a greater amount of open water. It is subject to future work to test this discretization in a more realistic setting.

Similar to the evaluation of satellite data, we analyze the averaged total ice deformation $|\dot{\boldsymbol{\epsilon}}|$ and the averaged total sea ice extent A_{total} under mesh refinement to investigate the convergence of the numerical solutions of the sea ice model. We solved the model problem described in Section 2.6 and observed that both functionals converge in space and time where the impact of the spatial discretization of the functional values is higher than the impact on the temporal discretization. Thus, we conclude that the spatial error dominates the temporal error. The approximation of the total sea ice extent is converging quadratic in space. This is expected as we use linear finite elements in space. The total ice deformation is converging on high mesh resolutions quadratically in time as we use a second order Taylor-Galerkin scheme.

By discretizing the balance laws with the Zalesak's flux limiter, we observed artifacts in the approximation of the ice concentration and ice thickness, called *terracing*. It is subject to future work to develop a modified limiting strategy in the flux correction scheme to avoid this effect.

It remains an open question, which discretization of the balance laws is optimal. The effects are manifold as the choice of the balance laws discretization has an influence on the robustness of the momentum equation, on the accurancy of the coupled simulation, but also on the global structure of the solution, such as the evolution of cracks in the ice.

Part III.

An efficient solver for the sea ice model

5. A Newton scheme to solve the momentum equation

Parts of this chapter have been published in Mehlmann and Richter [69, 70]. Using linear finite elements in space and an implicit Euler method in time, the sea ice momentum equation turns into a nonlinear algebraic system which needs to be solved at every time step. In context of sea ice dynamics there are mainly two schemes discussed in the literature to solve implicit discretizations of nonlinear momentum equation. First, a fixed-point method, denoted as *Picard solver*, and second an inexact Newton method. The Picard iteration was presented by Zhang and Hibler [111]. The idea of the standard Picard solver is to repeatedly solve a simple linearized system of equations. Lemieux and Tremblay [55] showed that the convergence of the Picard solver is slow. Therefore, they developed a second approach, an inexact Newton method, realized as the *Jacobian-free Newton-Krylov* (JFNK) solver [53, 56].

To avoid the implicit methods, currently many sea ice models apply the *elastic-viscous-plastic* formulation [40]. Here, an artificial elastic term is added to the viscous-plastic rheology to use an explicit discretization of the momentum equation, that allows relatively large time steps [30]. However, the EVP model produces large difference in the numerical solution, compared to the approximations computed with the *viscous-plastic* (VP) model [40, 63]. Lemieux et al. [54] modified the EVP solver by adding an inertia term to the momentum equation to compare the EVP and the VP solutions. Bouillon et al. [8] formulated the modified EVP solver as an iterative process converging to the VP solution. Kimmritz et al. [38] revised the convergence analysis of the modified EVP solver and showed that for a simplified model, the EVP solution converges to the VP solution, if a sufficiently large number of sub-cycling steps are taken. The convergence can be accelerated by an adaptive control of the sub-cycling [39].

The EVP and the Picard approach suffer from their poor convergence behavior. Typically, only a small number of Picard iterations or sub-cycling steps of the EVP model are considered. Thus the approximated solution still has a large numerical error which is accumulated over time [53]. This may result in large discrepancies of the simulated sea ice dynamics [55]. Newton solvers on the other hand tend to have difficulties if the Jacobian is ill-conditioned, nearly singular or if good initial values are not at hand. The nonlinearity coming from the viscous-plastic material tensor $\boldsymbol{\sigma}(\mathbf{v}^n)$ is severe and the efficient treatment of this term is still under active research [39]. The convergence of sea ice solvers usually worsens on high spatial resolutions [55, 56, 54]. Lemieux et al. [56] mentioned that

> Nonlinear solvers such as the JFNK method tend to have difficulties when

> there are such sharp structures in the solution. This lack of robustness of both solvers is however a debatable problem as it mostly occurs for large required drops in the residual norm. [...] Globalization approaches for the JFNK solver, such as the line search method, have not yet proven to be successful. Further investigation is needed.

We develop a robust modification of a Newton solver that is able to cope with high spatial resolutions. For the solution of the linear systems at each Newton iteration, it is recommended using a *generalized minimal residual method* (GMRES) preconditioned with a *line successive over-relaxation method* (LSOR) [57, 86]. The linear problems, arising in each Newton or Picard iteration, are very stiff and ill-conditioned. Losch et al. [64] observe that the use of an indirect LSOR preconditioner is computationally expensive. Especially on fine meshes, the LSOR preconditioner needs to be revised [53]. They suggested testing an *incomplete lower upper factorization* (ILU) as preconditioner to reduce computational cost. We introduce a multigrid solver as a preconditioner to the GMRES method and compare it to an ILU preconditioner.

In this section, we discuss different Newton approaches to solve the discretized sea ice model. In Section 5.1, we derive the analytical Jacobian of the system and analyze its properties. In Section 5.2, we present different approaches to globalize and accelerate the Newton method. These approaches are numerically evaluated in Section 5.3. In Section 5.4, we present the GMRES method with a multigird preconditioner to solve the linear subproblems in the Newton scheme. The linear solver is numerically evaluated in Section 5.5. We conclude in Section 5.6.

5.1. Newton solver

In the following, we write each time step using the abstract notation

$$A(\mathbf{v}_n, \boldsymbol{\phi}) = F(\boldsymbol{\phi}), \tag{5.1}$$

where

$$\begin{aligned} A(\mathbf{v}_n, \boldsymbol{\phi}_\mathbf{v}) :=& (\rho_{\text{ice}} H_n \mathbf{v}_n, \boldsymbol{\phi}_\mathbf{v}) + k(\rho_{ice} H f_c \boldsymbol{e}_r \times \mathbf{v}_n, \boldsymbol{\phi}_\mathbf{v}) \\ &+ k(\boldsymbol{\sigma}_n(A_n, H_n, \mathbf{v}_n), \nabla \boldsymbol{\phi}_\mathbf{v}) - k(\boldsymbol{\tau}_{\text{ocean}}(t_n, \mathbf{v}_n), \boldsymbol{\phi}_\mathbf{v}), \\ F(\boldsymbol{\phi}_\mathbf{v}) :=& (\rho_{\text{ice}} H_n \mathbf{v}_{n-1}, \boldsymbol{\phi}_\mathbf{v}) + k(\boldsymbol{\tau}_{\text{atm}}(t_n), \boldsymbol{\phi}_\mathbf{v}) + k(f_c \boldsymbol{e}_r \times \mathbf{v}_{\text{ocean}}, \boldsymbol{\phi}_\mathbf{v}). \end{aligned} \tag{5.2}$$

Then the Newton scheme to problem (5.1) reads as

$$\begin{aligned} \mathbf{w}_n^{(l)} \in \mathcal{V} \quad A'(\mathbf{v}_n^{(l-1)})(\mathbf{w}_n^{(l)}, \boldsymbol{\phi}) = F(\boldsymbol{\phi}) - A(\mathbf{v}_n^{(l-1)}, \boldsymbol{\phi}) \quad \forall \boldsymbol{\phi} \in \mathcal{V}, \\ \mathbf{v}_n^{(l)} = \mathbf{v}_n^{(l-1)} + \mathbf{w}_n^{(l)}, \quad l = 1, 2, \dots \end{aligned} \tag{5.3}$$

where by $\mathbf{v}_n^{(0)}$ we denote an initial guess which is usually taken as $\mathbf{v}_n^{(0)} = \mathbf{v}_{n-1}$, the solution of the last time step. By $A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi})$ we denote the Gâteaux derivative of $A(\mathbf{v}, \boldsymbol{\phi})$ in direction $\mathbf{w}$, which is defined as

$$A'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) := \frac{d}{ds}A(\mathbf{v}+s\mathbf{w},\boldsymbol{\phi})\Big|_{s=0}.$$

This derivative can be approximated by finite differences [44], by automatic differentiation [27] or analytically. Here, an analytical evaluation of the Jacobian is used for a better understanding of its structure. It is the first time that the Jacobian is derived analytically. Although the VP model involves strong nonlinearities and a complex viscosity structure, the resulting Jacobian is a positive definite matrix and is well-structured.

Theorem 1 (Jacobian of the viscous-plastic sea ice model)**.** *Let* $\mathbf{v} \in W^{1,\infty}(\Omega)\cap H_0^1(\Omega)$, $\mathbf{w} \in H_0^1(\Omega)$, $H \in L^\infty(\Omega)$ *and* $H > 0$. *Then, it holds that*
(i) *The derivative* $A'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})$ *of the implicit Euler discretization of the viscous-plastic sea ice model (5.2) is given by*

$$\begin{aligned} A'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) = (\rho_{ice}H\mathbf{w},\boldsymbol{\phi}) + k(\rho_{ice}Hf_c\mathbf{e}_r\times\mathbf{w},\boldsymbol{\phi}) \\ + k\boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) + k\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) - k\boldsymbol{\tau}_{ocean}(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}), \end{aligned} \tag{5.4}$$

where the derivatives of the stress-tensor $\boldsymbol{\sigma}$ *and the derivatives of the ocean forcing* $\boldsymbol{\tau}_{ocean}$ *are given by*

$$\begin{aligned} \boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) &= \Big(2e^{-2}\zeta\dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \zeta\,\mathrm{tr}\,(\dot{\boldsymbol{\epsilon}}(\mathbf{w}))I, \nabla\boldsymbol{\phi}\Big) \\ \boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) &= \Big(-\frac{\zeta}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\Big(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \mathrm{tr}(\dot{\boldsymbol{\epsilon}})\,\mathrm{tr}\,(\dot{\boldsymbol{\epsilon}}(\mathbf{w}))\Big)\Big(2e^{-2}\dot{\boldsymbol{\epsilon}}' + \mathrm{tr}(\dot{\boldsymbol{\epsilon}})I\Big), \nabla\boldsymbol{\phi}\Big) \\ \boldsymbol{\tau}_{ocean}'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) &= \Big(-\rho_{ocean}C_{ocean}\left(\|\mathbf{v}_{ocean}-\mathbf{v}\|\mathbf{w} + \frac{(\mathbf{v}_{ocean}-\mathbf{v})\cdot\mathbf{w}}{\|\mathbf{v}_{ocean}-\mathbf{v}\|}(\mathbf{v}_{ocean}-\mathbf{v})\right), \boldsymbol{\phi}\Big). \end{aligned} \tag{5.5}$$

(ii) *Apart from the Coriolis term, the Jacobian is symmetric*

$$A'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) = A'(\mathbf{v})(\boldsymbol{\phi},\mathbf{w}) - 2k(\rho_{ice}f_c\mathbf{e}_r\times\boldsymbol{\phi},\mathbf{w}).$$

(iii) *Furthermore,* $\boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})$*, the derivative with respect to the strain rates, as well as the complete Jacobian are positive definite, while* $\boldsymbol{\tau}_{ocean}'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})$ *is positive semidefinite and* $\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})$*, the derivative with respect to the viscosities, is negative semidefinite.*

(iv) *It holds that*

$$\delta_f\|\mathbf{w}\|^2 + c_K\min_\Omega\left\{\frac{\Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2 \le A'(\mathbf{v})(\mathbf{w},\mathbf{w}) \le 2\delta_f\|\mathbf{w}\|^2 + 2\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2, \tag{5.6}$$

where $\delta_f := \rho_{ice}H + k\rho_oC_o\|\mathbf{v}-\mathbf{v}_{ocean}\|$ *and the constant* $c_K > 0$ *depends on the domain* Ω. *Further,* $\|\mathbf{w}\|_{H^1(\Omega)}^2 := \|\mathbf{w}\|^2 + \|\nabla\mathbf{w}\|^2$.

Proof. We frequently split the strain rate $\dot{\boldsymbol{\epsilon}}$ (2.12) into the deviatoric part $\dot{\boldsymbol{\epsilon}}'$ and the trace $\operatorname{tr}(\dot{\boldsymbol{\epsilon}})$. For better readability, we repeat the important definitions made in Section 2.4.2.

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I, \tag{5.7}$$

The deviatoric part is given as

$$\dot{\boldsymbol{\epsilon}}' = \dot{\boldsymbol{\epsilon}} - \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I = \begin{pmatrix} \frac{1}{2}(\dot{\boldsymbol{\epsilon}}_{11} - \dot{\boldsymbol{\epsilon}}_{22}) & \dot{\boldsymbol{\epsilon}}_{12} \\ \dot{\boldsymbol{\epsilon}}_{12} & \frac{1}{2}(\dot{\boldsymbol{\epsilon}}_{22} - \dot{\boldsymbol{\epsilon}}_{11}) \end{pmatrix}. \tag{5.8}$$

Thus,

$$\dot{\boldsymbol{\epsilon}}'(\boldsymbol{\phi}) : I = \sum_{i,j} \dot{\boldsymbol{\epsilon}}'_{i,j} I_{i,j} = \dot{\boldsymbol{\epsilon}}'_{1,1} + \dot{\boldsymbol{\epsilon}}'_{2,2} = 0, \quad \text{as } \operatorname{tr}(\dot{\boldsymbol{\epsilon}}') = 0.$$

We start with proving the first claim of Theorem 1.
(i)Derivatives. The derivative $A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi})$ can be split into two parts.

$$\begin{aligned} &A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) = A'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) + A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}), \\ &A'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) := (\rho_{ice} H \mathbf{w}, \boldsymbol{\phi}) + k(f_c \boldsymbol{e}_r \times \mathbf{w}, \boldsymbol{\phi}) + k\boldsymbol{\sigma}'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) - k\boldsymbol{\tau}'_{ocean}(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}), \\ &A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) := k\boldsymbol{\sigma}'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}). \end{aligned} \tag{5.9}$$

We introduce the abbreviations

$$\boldsymbol{\sigma}' := \boldsymbol{\sigma}'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}), \quad \boldsymbol{\sigma}'_i := \boldsymbol{\sigma}'_i(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}),\ i = 1, 2.$$

The splitting of the stresses $\boldsymbol{\sigma}' = \boldsymbol{\sigma}'_1 + \boldsymbol{\sigma}'_2$ is organized such that $\boldsymbol{\sigma}'_1$ includes the derivatives with respect to the strain rate $\dot{\boldsymbol{\epsilon}}$ while $\boldsymbol{\sigma}'_2$ includes the derivatives with respect to the viscosities $\zeta(\dot{\boldsymbol{\epsilon}})$ and $\eta(\dot{\boldsymbol{\epsilon}})$. Here, the fundamental expression is the derivative of $\Delta(\dot{\boldsymbol{\epsilon}})^{-1}$. For a better readability, we recall the formulation of the stress tensor and the definition of $\Delta(\dot{\boldsymbol{\epsilon}})$ and rewrite them using their symmetric structure. Applying the symmetry of the inner product, we get

$$(\boldsymbol{\sigma}, \nabla\phi) = \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^T, \nabla\phi) = \frac{1}{2}(\boldsymbol{\sigma}, \nabla\phi + \nabla\phi^T) = (\boldsymbol{\sigma}, \dot{\boldsymbol{\epsilon}}(\phi)), \quad \dot{\boldsymbol{\epsilon}}(\phi) := \frac{1}{2}\left(\nabla\phi + \nabla\phi^T\right). \tag{5.10}$$

Considering the structure of the stress tensor and applying that $\eta = e^{-2}\zeta$ it holds that

$$\begin{aligned} (\boldsymbol{\sigma}, \nabla\phi) = (\boldsymbol{\sigma}, \dot{\boldsymbol{\epsilon}}(\phi)) &= (2\eta\dot{\boldsymbol{\epsilon}}' + \zeta \operatorname{tr}(\dot{\boldsymbol{\epsilon}})I - \frac{P}{2}I, \dot{\boldsymbol{\epsilon}}(\phi)) \\ &= (\zeta(2e^{-2}\dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})I) - \frac{P}{2}I, \dot{\boldsymbol{\epsilon}}(\phi)). \end{aligned} \tag{5.11}$$

In the following, we replace the strain rate $\dot{\boldsymbol{\epsilon}}$ by (5.7) and make frequently use of

$$\boldsymbol{\epsilon}' : \operatorname{tr}(\dot{\boldsymbol{\epsilon}})I = 0 \text{ and } I : I = 2.$$

We proceed with deriving (5.11) to

$$\begin{aligned}\Big(\boldsymbol{\sigma}, \nabla\phi\Big) &= 2\Big(\zeta(e^{-2}\dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I), \dot{\boldsymbol{\epsilon}}'(\phi) + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))I\Big) - \Big(P, \operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))\Big) \\ &= 2\Big(\zeta e^{-2}\dot{\boldsymbol{\epsilon}}', \dot{\boldsymbol{\epsilon}}'(\phi)\Big) + 2\Big(\frac{1}{2}\zeta\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I, \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))I\Big) - \Big(P, \operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))\Big) \\ &= 2\Big(\zeta(\underbrace{e^{-1}\dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I}_{=:\boldsymbol{\tau}(\mathbf{v})}), \underbrace{e^{-1}\dot{\boldsymbol{\epsilon}}'(\phi) + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))I}_{=:\boldsymbol{\tau}(\phi)}\Big) - \Big(P, \operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))\Big) \\ &= 2\Big(\zeta\boldsymbol{\tau}(\mathbf{v}), \boldsymbol{\tau}(\phi)\Big) - \Big(P, \operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))\Big), \qquad \boldsymbol{\tau}(\mathbf{v}) := e^{-1}\dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I.\end{aligned}$$

Using the notation of $\boldsymbol{\tau}(\mathbf{v})$, we reformulate $\Delta(\dot{\boldsymbol{\epsilon}})$ to

$$\begin{aligned}\Delta(\dot{\boldsymbol{\epsilon}}) &= \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}, \\ &= \sqrt{2\tau(\mathbf{v}) : \tau(\mathbf{v}) + \Delta_{\min}^2}.\end{aligned} \tag{5.12}$$

Finally, we receive the following compact representation of the stress tensor,

$$(\boldsymbol{\sigma}, \nabla\phi) = \left(\frac{P}{\sqrt{\Delta_{\min}^2 + 2\tau(\mathbf{v}) : \tau(\mathbf{v})}}\tau(\mathbf{v}), \tau(\phi)\right) - \Big(P, \operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\phi))\Big). \tag{5.13}$$

Using (5.13), we compute the derivative of the stress tensor with respect to $\mathbf{v}$ in direction $\mathbf{w}$.

$$\boldsymbol{\sigma}'(\mathbf{v})(\mathbf{w}, \phi) = \underbrace{\left(\frac{P}{\sqrt{\Delta_{\min}^2 + 2\boldsymbol{\tau}(\mathbf{v}) : \boldsymbol{\tau}(\mathbf{v})}}\boldsymbol{\tau}(\mathbf{w}), \boldsymbol{\tau}(\phi)\right)}_{:=\boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})} \tag{5.14}$$

$$\underbrace{- \left(\frac{P}{\sqrt{\Delta_{\min}^2 + 2\boldsymbol{\tau}(\mathbf{v}) : \boldsymbol{\tau}(\mathbf{v})}^3}(2\boldsymbol{\tau}(\mathbf{v}) : \boldsymbol{\tau}(\mathbf{w}))\boldsymbol{\tau}(\mathbf{v}), \boldsymbol{\tau}(\phi)\right)}_{:=\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi})}. \tag{5.15}$$

We insert $\tau(\mathbf{v})$ into the derivative of $\boldsymbol{\sigma}'$ and obtain the expression proposed in the Theorem 1.

$$\begin{aligned}\boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) &= \Big(2\zeta(e^{-1}\dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \frac{1}{2}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)I, e^{-1}\dot{\boldsymbol{\epsilon}}'(\boldsymbol{\phi}) + \frac{1}{2}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\boldsymbol{\phi})\right)I\Big) \\ &= \Big(2e^{-2}\zeta\dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \zeta\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)I, \dot{\boldsymbol{\epsilon}}(\boldsymbol{\phi})\Big).\end{aligned} \tag{5.16}$$

For $\boldsymbol{\sigma}_2'$, we get

$$\begin{aligned}\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) &= -\left(\frac{P}{\Delta^3}[2\tau(\mathbf{v}) : \tau(\mathbf{w})](e^{-2}\dot{\boldsymbol{\epsilon}}' + \frac{1}{2}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I), \dot{\boldsymbol{\epsilon}}'(\boldsymbol{\phi}) + \frac{1}{2}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\boldsymbol{\phi})\right)I\right) \\ &= -\left(\frac{\zeta}{\Delta^2}[2\tau(\mathbf{v}) : \tau(\mathbf{w})](2e^{-2}\dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})I), \dot{\boldsymbol{\epsilon}}'(\boldsymbol{\phi}) + \frac{1}{2}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\boldsymbol{\phi})\right)I\right).\end{aligned} \tag{5.17}$$

We insert $\boldsymbol{\tau}(\mathbf{v})$ and $\tau(\mathbf{w})$ and get

$$\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\boldsymbol{\phi}) = \\ -\left(\frac{\zeta}{\Delta^2}\Big[2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)\Big], \Big[2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}'(\boldsymbol{\phi}) + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\boldsymbol{\phi})\right)\Big]\right). \tag{5.18}$$

The derivative of the ocean forcing,

$$(\tau_{\text{ocean}}(\mathbf{v}), \boldsymbol{\phi}) = \Big(C_{\text{ocean}}\rho_{\text{ocean}}\|\mathbf{v}_{\text{ocean}} - \mathbf{v}\|(\mathbf{v}_{\text{ocean}} - \mathbf{v}), \boldsymbol{\phi}\Big),$$

is given as

$$\tau'_{ocean}(\mathbf{v})(\mathbf{w}, \phi) = -p_{ocean}C_{ocean}\Big(\mathbf{w}\|v_{ocean} - \mathbf{v}\| + (\mathbf{v}_{ocean} - \mathbf{v})\frac{(\mathbf{v}_{ocean} - \mathbf{v})}{\|\mathbf{v}_{ocean} - \mathbf{v}\|}\mathbf{w}, \boldsymbol{\phi}\Big)$$

Finally, we calculate the derivatives of the ice mass term, $(\rho_{\text{ice}}H_n\mathbf{v}_n, \boldsymbol{\phi}_{\mathbf{v}})$, and the Coriolis forcing, $(\rho_{ice}Hf_c\boldsymbol{e}_r \times \mathbf{v}_n, \boldsymbol{\phi}_{\mathbf{v}})$, as

$$(\rho_{ice}H\mathbf{w}, \boldsymbol{\phi}) + k(f_c\boldsymbol{e}_r \times \mathbf{w}, \boldsymbol{\phi}).$$

(ii) Symmetry. Next, we analyze the different terms in the Jacobian $A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi})$ and check for symmetry. As the triple product $(\vec{a}\times\vec{b})\cdot\vec{c} = -(\vec{a}\times\vec{c})\cdot\vec{b}$ is anti-commutative, the Coriolis term is non-symmetric. The term stemming from the time derivative $(\rho_{\text{ice}}H\mathbf{w}, \boldsymbol{\phi})$ is clearly symmetric. Using the formulation of the stress tensor (5.14) we see that $\boldsymbol{\sigma}_1'$ is symmetric. The symmetry of $\boldsymbol{\sigma}_2'$ can be observed in representation (5.18). Likewise, symmetry holds for $\boldsymbol{\tau}'_{\text{ocean}}$ given by (5.5). Finally, all terms except the Coriolis term are symmetric.

(iii) Definiteness. To show definiteness of the Jacobian we first note that $(\boldsymbol{e}_r \times \mathbf{w}, \mathbf{w}) = 0$. Positive definiteness of $(\rho_{ice}H\mathbf{w}, \mathbf{w})$ is obvious as long as $H > 0$. The forcing term $-\boldsymbol{\tau}'_{\text{ocean}}(\mathbf{v})(\mathbf{w}, \mathbf{w})$ is positive semidefinite as

$$-\boldsymbol{\tau}'_{\text{ocean}}(\mathbf{v})(\mathbf{w}, \mathbf{w}) = \rho_{ocean}C_{ocean}\left(\|\mathbf{v}_{ocean} - \mathbf{v}\|\|\mathbf{w}\|^2 + \frac{\|(\mathbf{v}_{ocean} - \mathbf{v})\cdot\mathbf{w}\|^2}{\|\mathbf{v}_{ocean} - \mathbf{v}\|}\right).$$

Next, we prove positive definiteness of $\boldsymbol{\sigma}'$. With (5.16) and (5.18) we obtain

$$\begin{aligned}\boldsymbol{\sigma}_1'(\mathbf{v})(\mathbf{w}, \mathbf{w}) &= \|\sqrt{2e^{-2}\zeta}\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\|^2 + \|\sqrt{\zeta}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)\|^2, \\ \boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w}, \mathbf{w}) &= -\|\sqrt{\zeta/\Delta(\dot{\boldsymbol{\epsilon}})^2}\big[2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\right)\big]\|^2,\end{aligned} \tag{5.19}$$

and observe positivity of $\boldsymbol{\sigma}_1'$. To proceed with $\boldsymbol{\sigma}_2'$ we define

$$A := \sqrt{2e^{-2}}\dot{\boldsymbol{\epsilon}}' + \frac{1}{\sqrt{2}}\operatorname{tr}(\dot{\boldsymbol{\epsilon}})I : \Omega \to \mathbb{R}^{2\times2}, \quad B := \sqrt{2e^{-2}}\dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \frac{1}{\sqrt{2}}\operatorname{tr}(\dot{\boldsymbol{\epsilon}}(\mathbf{w}))I : \Omega \to \mathbb{R}^{2\times2}. \tag{5.20}$$

By Cauchy-Schwarz inequality, it holds that $\left(\sum_{ij} A_{ij}B_{ij}\right)^2 \leq \sum_{ij} A_{ij}^2 \sum_{ij} B_{ij}^2$. Using the notation of $\Delta(\dot{\boldsymbol{\epsilon}})$ given in (5.12), we get

$$\begin{aligned}\frac{1}{\sqrt{\zeta/\Delta^2}}[\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w}) : \nabla \mathbf{w}] &= -\left[2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\right)\right]^2 \\ &= -(A:B)^2 \geq -(A:A)\cdot(B:B) \\ &\geq -[\underbrace{2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2}_{=:\Delta^2(\dot{\boldsymbol{\epsilon}})-\Delta_{min}^2}] \cdot [2e^{-2}\dot{\boldsymbol{\epsilon}}'(\mathbf{w}) : \dot{\boldsymbol{\epsilon}}'(\mathbf{w}) + \operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)^2]. \quad (5.21)\end{aligned}$$

The combination of (5.19) and (5.21) gives

$$\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\mathbf{w}) \geq -\max_{\Omega}\left\{\frac{\Delta^2(\dot{\boldsymbol{\epsilon}}) - \Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\left(\|\sqrt{2e^{-2}\zeta}\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\|^2 + \|\sqrt{\zeta}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)\|^2\right),$$

with a positive constant $c > 0$. Together with the estimate of $\boldsymbol{\sigma}_1'$ in (5.19), we obtain

$$\begin{aligned}\boldsymbol{\sigma}'(\mathbf{v})(\mathbf{w},\mathbf{w}) &\geq \min_{\Omega}\left\{\frac{\Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\left(\|\sqrt{2e^{-2}\zeta}\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\|^2 + \|\sqrt{\zeta}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)\|^2\right) \\ &\geq c_K \min_{\Omega}\left\{\frac{\Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2,\end{aligned}$$

where $c_K > 0$ is the constant of Korn's inequality [11].

(iv) Upper and lower bound of the Jacobian. The upper bound of the $\boldsymbol{\sigma}_2'$ is obtained in the same manner as the lower bound such that

$$|\boldsymbol{\sigma}_2'(\mathbf{v})(\mathbf{w},\mathbf{w})| \leq \max_{\Omega}\Big\{\underbrace{\frac{\Delta^2(\dot{\boldsymbol{\epsilon}}) - \Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}}_{\leq 1}\Big\}\left(\|\sqrt{2e^{-2}\zeta}\dot{\boldsymbol{\epsilon}}'(\mathbf{w})\|^2 + \|\sqrt{\zeta}\operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}(\mathbf{w})\right)\|^2\right).$$

We combine $\boldsymbol{\sigma}_1'$ and $\boldsymbol{\sigma}_2'$ and get

$$c_K \min_{\Omega}\left\{\frac{\Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2 \leq \boldsymbol{\sigma}'(\mathbf{v})(\mathbf{w},\mathbf{w}) \leq 2\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2.$$

By applying the Cauchy-Schwarz inequality, it holds that

$$k\rho_o C_o\|\mathbf{v}_{ocean} - \mathbf{v}\|\|\mathbf{w}\|^2 \leq -\tau_{ocean}'(\mathbf{v})(\mathbf{w},\mathbf{w}) \leq k\rho_o C_o 2\|\mathbf{v}_{ocean} - \mathbf{v}\|\|\mathbf{w}\|^2.$$

Taking into account that $(\boldsymbol{e}_r \times \mathbf{w}, \mathbf{w}) = 0$, we achieve the whole inequality

$$\delta_f\|\mathbf{w}\|^2 + c_K \min_{\Omega}\left\{\frac{\Delta_{min}^2}{\Delta^2(\dot{\boldsymbol{\epsilon}})}\right\}\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2 \leq A'(\mathbf{v})(\mathbf{w},\mathbf{w}) \leq 2\delta_f\|\mathbf{w}\|^2 + 2\|\sqrt{\zeta}\mathbf{w}\|_{H^1(\Omega)}^2, \quad (5.22)$$

where $\delta_f := \rho_{ice} H + k\rho_o C_o \|\mathbf{v} - \mathbf{v}_{ocean}\|$.

□

The theorem has several implications for the design of numerical methods to approximate the VP sea ice model. Inequality (5.6) shows that all eigenvalues of the Jacobian are bound from below and above. For $\zeta > 0$, which is the case for $H > 0$, all eigenvalues are positive such that the matrix is positive definite. For functions with a positive definite and Lipschitz continuous Jacobian, a damped Newton method is globally convergent (see Lemma 3.5 of [16]). We will use a line search method to damp each Newton iteration, see Section 5.2. The ratio of largest and smallest eigenvalue indicates the condition number of the matrix. This number is important, as it describes the propagation of errors when solving linear systems. Usually, the convergence rate of iterative solvers worsens if the condition number is large. The convergence rate of the Newton method also depends on the smallest eigenvalue. This shows the role of limiting $\Delta(\dot{\boldsymbol{\epsilon}}) \geq \Delta_{min}$ in the lower bound of (5.6).

Finally, we showed that the Jacobian is symmetric if we neglect the Coriolis term. An explicit discretization of this term allows the use the *Conjugate Gradient* method as linear solver which is more efficient than GMRES method. In particular, it does not require additional storage for orthogonalization, see [86].

5.2. Globalization and acceleration methods

Existing solvers for the sea ice momentum equation are not robust with increasing resolutions [56]. In this context, the most discussed Newton solver is the *Jacobian-free Newton-Krylov* (JFNK) method developed by Lemieux and Tremblay [56]. There are mainly two acceleration strategies implemented in the framework of the JFNK solver to improve the numerical convergence. First, an inexact solving of the linear problems in a Newton iteration and second, a line search approach [53]. Lemieux et al. [64] suggested testing a combination of the JFNK with a Picard solver to further improve the convergence properties of the solver. A combination of a Newton method and a Picard solver for *multidimensional variably saturated flow problems* was carried out in [76]. First, a Picard solver is used until the iteration converges steadily, and then the Newton method is applied. While this approach has similarities to the *operator-related damped Jacobian* method, the adaptive control is different. A numerical evaluation of the globalization and the acceleration strategies presented in this Section is given in Section 5.3.

Inexact solving of the linear subproblems

The JFNK solver is based on an inexact Newton method. In the l-th iteration of the Newton method (5.3), one needs to solve the linear subproblem

$$\underbrace{A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}_i)}_{:=[\mathcal{A}'(\mathbf{v})(\mathbf{w})]_i} = \underbrace{F(\boldsymbol{\phi}_i) - A(\mathbf{v}, \boldsymbol{\phi}_i)}_{:=[\mathcal{R}(\mathbf{v})]_i}, \tag{5.23}$$

with the basis function $\boldsymbol{\phi}_i \in V_h^{\mathbf{v}}$. The idea of the inexact soling approach is to accelerate the convergence of the Newton solver by a coarse approximation of the linear subproblem (5.23). This technique has been applied in the first version of the JFNK solver [56] with an adaptive control of the linear residual. Lemieux and Tremblay [56] found that the convergence of the nonlinear solver is slow in the beginning and fast at the end of a time step. Additionally, they observed that in some cases a high initial tolerance criterion ($tol \sim 0.001$) can lead to a non-converging Newton solver due to *over-solving*. Over-solving describes the effect that solving the linear subproblem up to small residual can cause an inaccurate nonlinear correction of the solution [85]. Therefore, Lemieux and Tremblay suggest choosing the initial tolerance low ($tol \sim 1$) and reduce it as one gets in the area of fast convergence [53]. In the l-th iteration of the Newton method (5.3) Lemieux et al. [56] require that

$$\|\mathcal{A}'(\mathbf{v}_n^{(l-1)})(\mathbf{w}_n^{(l)}) - \mathcal{R}(\mathbf{v}_n^{(l-1)})\| \leq \mathrm{tol}(l)\|\mathcal{R}(\mathbf{v}_n^{(l-1)})\|, \tag{5.24}$$

with the adaptive tolerances $\mathrm{tol}(l)$. The tolerances for the linear problems are chosen similar to Walker and Eisenstat [17] as

$$\mathrm{tol}(l) = \begin{cases} 0.99, & \text{if } ||\mathcal{R}(\mathbf{v}_n^{(l-1)})|| \geq \frac{2}{3}||\mathcal{R}(\mathbf{v}_n^{(0)})||, \\ \left(\frac{||\mathcal{R}(\mathbf{v}_n^{(l-1)})||}{||\mathcal{R}(\mathbf{v}_n^{(l-2)})||}\right)^{1.5} & \text{else.} \end{cases}$$

Further, if the Newton iteration count exceeds $l > 100$, the tolerance is taken as $\mathrm{tol}(l) = 0.99$ to improve robustness.

Line search method

In the framework of the JFNK solver a line search method is applied. The general Newton method with the line search approach for the solution of variational problems is given as

$$\begin{aligned} \mathbf{w}_n^{(l)} \in \mathcal{V} \quad A'(\mathbf{v}_n^{(l-1)})(\mathbf{w}_n^{(l)}, \boldsymbol{\phi}) = F(\boldsymbol{\phi}) - A(\mathbf{v}_n^{(l-1)}, \boldsymbol{\phi}) \quad \forall \boldsymbol{\phi} \in \mathcal{V}, \\ \mathbf{v}_n^{(l)} = \mathbf{v}_n^{(l-1)} + \omega \mathbf{w}_n^{(l)}. \end{aligned} \tag{5.25}$$

By $\omega \in (0, 1]$ we denote the *line search* parameter to relax the Newton update $\mathbf{w}_n^{(l)}$. This parameter is chosen as large as possible (close to one) but small enough such that the Newton residual decays

$$\omega \in \{1, (1-\gamma), (1-\gamma)^2, \dots\} : \quad \|\mathcal{R}(\mathbf{v}_n^{(l-1)} + \omega \mathbf{w}_n^{(l)})\| < \|\mathcal{R}(\mathbf{v}_n^{(l-1)})\|,$$

where $\gamma \in (0, 1)$. Lemieux et al. [53] choose $\gamma = 0.5$ and perform a maximum of 4 iterations of the line search approach. The line search method strongly increases the robustness of the JFNK solver [53]. As shown in Theorem 1, the Jacobian of the sea ice momentum equation is positive definite. For functions with a positive definite and Lipschitz continuous Jacobian, a damped Newton method is globally convergent. Therefore, we also apply a line search method as globalization strategy to the modified

Newton solver. We choose $\gamma = 0.75$ for better robustness and perform a maximum of 10 linear search iterations. Afterwards, we proceed with the next Newton step even if the residual is still increasing.

Operator-related damped Jacobian method

The *operator-related damped* Jacobian method was introduced by Horn, Ouazzi and Turek [28, 75]. A detailed description of the approach can be found in [74]. As presented in Theorem 1, the Jacobian A' can be split into a positive definite part A'_1 and a negative semidefinite part A'_2,

$$A'(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) = A'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) + A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}). \tag{5.26}$$

We refer to the positive definite part as the *good, stabilizing* part of the Jacobian

$$A'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) = (\rho_{ice} H \mathbf{w}, \boldsymbol{\phi}) + k(\rho_{ice} H f_c \boldsymbol{e}_r \times \mathbf{w}, \boldsymbol{\phi}) + k\boldsymbol{\sigma}'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) - k\boldsymbol{\tau}'_{ocean}(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}),$$

whereas the negative semidefinite part is possibly *troublesome*

$$A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) = k\boldsymbol{\sigma}'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}).$$

To improve the robustness of the Newton method, we adaptively control the negative semidefinite part of the Jacobian. The reason for splitting the Jacobian is twofold. First the convergence rate of the Newton method depends on the bound of the inverse of the Jacobian, which is given by the inverse of its lowest eigenvalue. By decreasing the influence of $A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi})$, the lowest eigenvalue in (5.22) is increased and the convergence of the method is improved as detailed in Richter and Mehlmann [81]. A complete analysis of the operator-related damped Newton method is still subject to future work. Some first results are given by Richter and Mehlmann in [81]. A second benefit of adaptively controlling $A'_2(\mathbf{v})(\mathbf{w})$ is the reduced condition number of the system matrix as described in Section 5.1.

The Jacobian of the modified Newton solver is given by

$$A'_\delta(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) := A'_1(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}) + \delta A'_2(\mathbf{v})(\mathbf{w}, \boldsymbol{\phi}),$$

where the parameter $\delta \in [0, 1]$ is adaptively chosen depending on the convergence history. The idea is to choose $\delta = 1$ if the reduction rate of the Newton method is good. If not, or only very slow convergence is observed, we reduce δ to get closer to a robust fixed-point iteration, which is for $\delta = 0$ very similar to typical Picard solvers [56].

The choice of δ is crucial for the convergence rate of the modified Newton solver. With the residual (5.23) of the Newton scheme, we define the reduction rate in the l-th iteration of the Newton method as

$$Q_l := \frac{\|\mathcal{R}(\mathbf{v}_n^{(l)})\|}{\|\mathcal{R}(\mathbf{v}_n^{(l-1)})\|}. \tag{5.27}$$

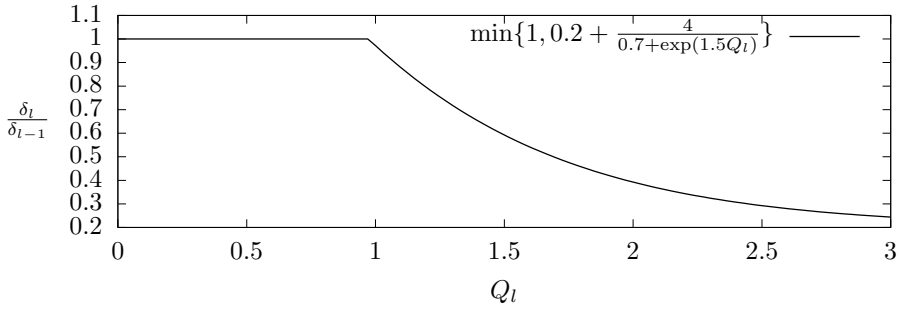


Figure 5.1.: Relation of damping parameter $\frac{\delta^{(l)}}{\delta^{(l-1)}}$ and reduction rate Q_l.

Similar to Turek et al. [66], we choose the damping parameter in the l-th step as

$$\delta^{(l)} = \begin{cases} \min\left\{1, \delta^{(l-1)}\left(0.2 + \frac{4}{0.7+\exp(1.5Q_l)}\right)\right\} & \delta^{(l-1)} \geq 0.2, \\ 1 & \delta^{(l-1)} < 0.2. \end{cases} \tag{5.28}$$

This relation is found by experimental tuning in [66]. Other feedback functions give similar results without a distinct superiority. Therefore, we adapted the original suggestion. A constant choice of δ does not increase the convergence rate of the modified Newton solver. In opposition to Turek et al. [66], we start with $\delta^{(0)} = 1$ to achieve rapid Newton convergence if possible. The damping parameter is reduced if the reduction rate exceeds $Q_l > \tilde{Q} \approx 0.97$, see Figure 5.1. Otherwise, we increase $\delta^{(l)}$ up to $\delta^{(l)} = 1$, which corresponds to the standard Newton scheme. To avoid very rapid changes of $\delta^{(l)}$, we modify the parameter relative to the last value. In contrast to the original work of Turek et al., we reset $\delta^{(l)}$ to 1 once $\delta^{(l-1)}$ reaches a limit of 0.2, and restart the Newton iteration with probably better initial values. This is done to avoid a stagnation of the convergence.

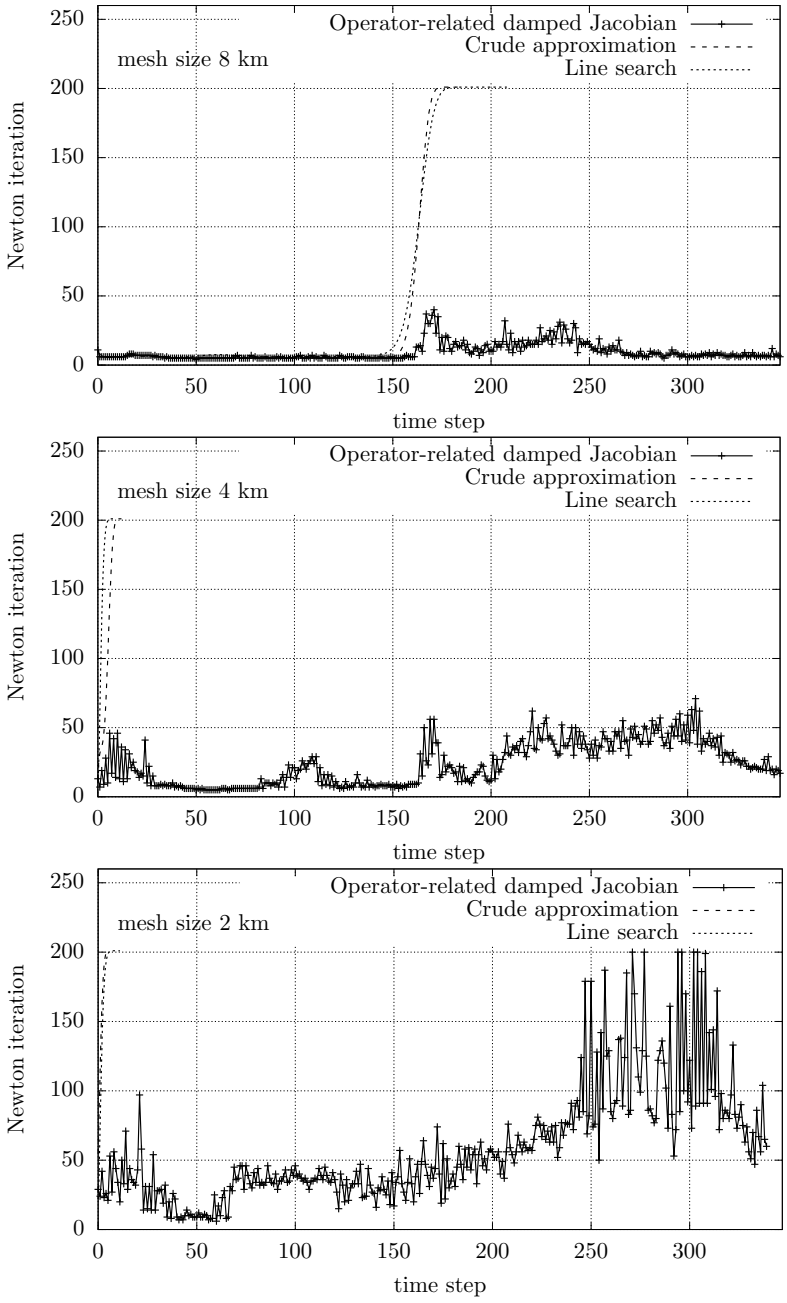


Figure 5.2.: Inexact Newton method with different globalization and acceleration strategies under mesh refinement. By crude approximation, we are referring to an inexact solving of the linear subproblems. We solve the test problem of Section 2.6 and evaluate the number of Newton steps per time step. The relative nonlinear tolerance is set to $tol_r = 10^{-4}$ and the global tolerance to $tol_g = 10^{-13}$. The simulation is terminated if the solver cannot reduce the residual to the desired tolerance in 10 consecutive time steps.

5.3. Numerical evaluation of the Newton solver

First, we compare the performance of the Newton solver with the different globalization techniques presented in Section 5.2, namely a *line search globalization*, a *inexact solving of the linear problems* and the *operator-related damped Jacobian* method. Then, we give a detailed numerical analysis of the operator-related damped Jacobian method on different mesh resolutions. By refining, the mesh structures get sharper and solving the implicit momentum equation becomes increasingly difficult. Linear and nonlinear tolerances are set as described in Section 2.6.

Comparison of the different globalization and acceleration strategies

In Figure 5.2, we show the performance of the globalization and acceleration methods that are discussed in Section 5.2. The performance is analyzed on 8 km, 4 km, 2 km meshes. We run the test case for a period of $I = [0, 8]$ days. In the top of Figure 5.2, we show the performance of the solver on an 8 km mesh. The inexact Newton method globalized by a line search approach (with and without exact solving of the linear subproblems) fails to reach the desired nonlinear residual after about $n = 160$ time steps. However, the operator-related damped Jacobian method with a line search globalization provides a robust strategy on the 8 km mesh.

On finer meshes, this result becomes even clearer. In the center image of Figure 5.2 we compare all three approaches on a 4 km mesh. The inexact Newton method with a line search approach fails after two time steps. The inexact Newton method with a line search approach and an inexact approximation of the linear subproblem is more robust, but it is not converging after 6 time steps. Using the operator-related damped Jacobian approach with a line search globalization the solver is robust for all time steps.

In the image at the bottom of Figure 5.2 we compare the performance of the Newton solver on a 2 km mesh. The inexact Newton method globalized by a line search approach (with and without inexact solving of the linear subproblems) could not reduce the nonlinear residual to the required tolerance after two time steps. The operator-related damped Jacobian method with a line search strategy solves the test problem until time step $n = 272$ (≈ 6 days), where the desired tolerance could not be reached in the given 200 steps. On the 2 km mesh, small-scale patterns evolve. The term $\Delta^2_{min}/\Delta^2(\dot{\boldsymbol{\epsilon}})$ becomes small and the lower bound of the Jacobian (5.6) turns towards zero. This could be one reason for the slower convergence of the Newton solver with the operator-related damped Jacobian method. If we continue the simulation for the full temporal interval of $I = [0, 8\,\text{days}]$, we note only 6 failures out of 350 steps. In comparison, if a failure occurs using a standard Newton solver globalized with a line search method (with and without inexact solving of the linear subproblems), the residual cannot be reduced in the subsequent 10 time steps. Therefore, the simulation was terminated.

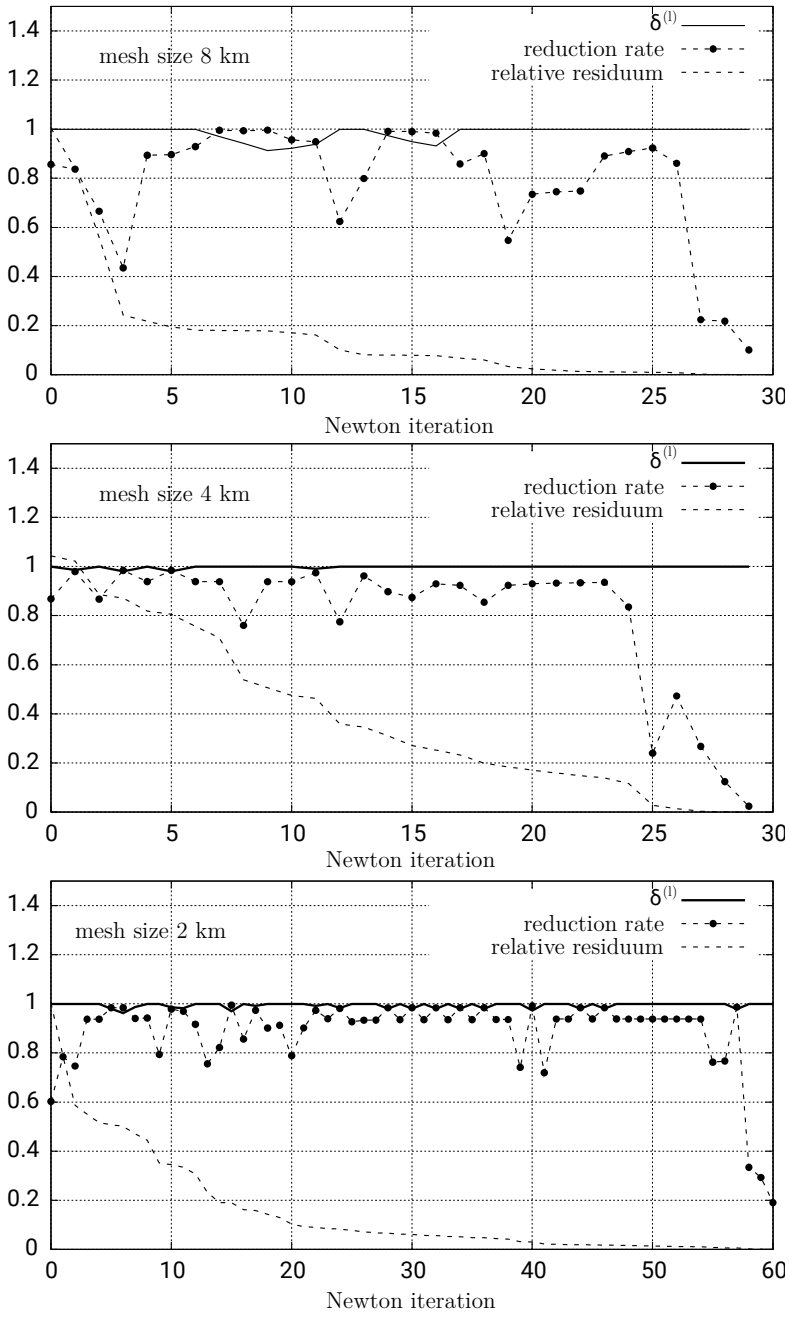


Figure 5.3.: Evaluating the damping parameter $\delta^{(l)}$, the reduction rate Q_l and the residual of the operator-related damped Jacobian method at time step $n = 160$ on an 8 km, 4 km, 2 km mesh. The residual is normalized by the initial residual.

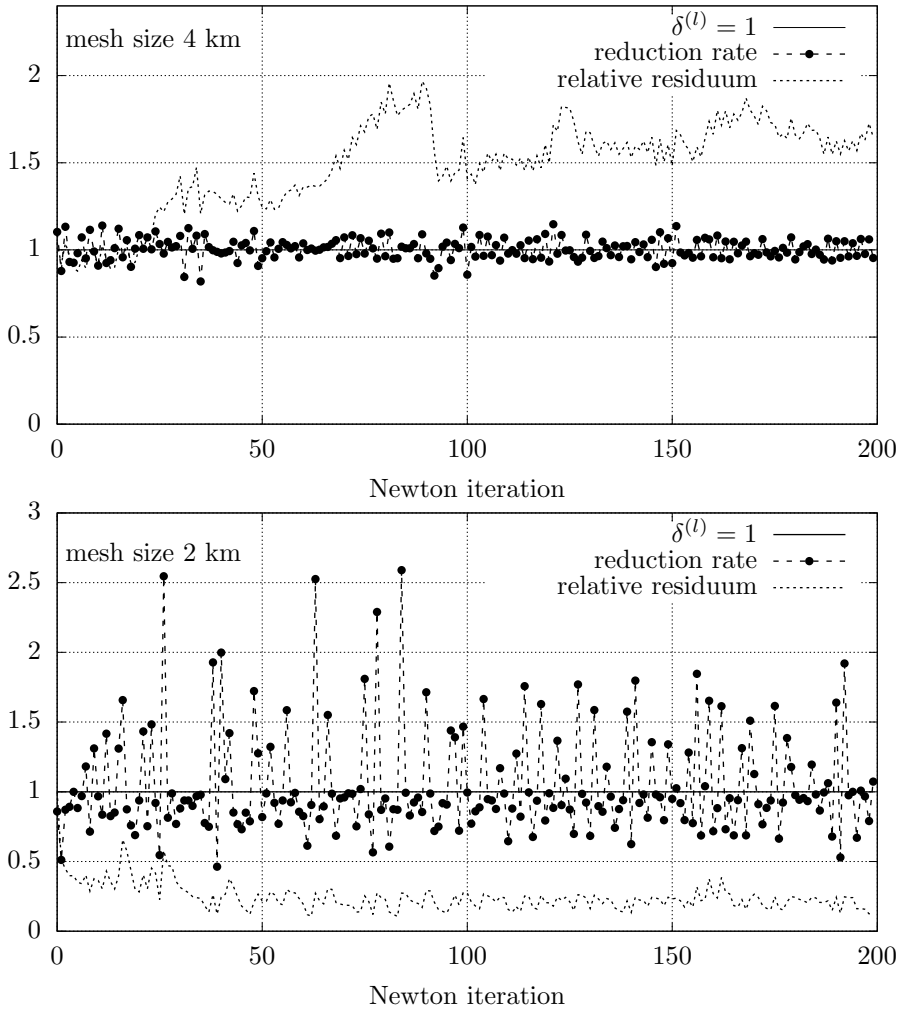


Figure 5.4.: Convergence of the full Newton scheme at time step $n = 160$ ($\delta^{(l)} = 1$ fixed). No convergence is observed on the 4 km mesh (top) and on the 2 km mesh (bottom). Compare to Figure 5.3 for the operator-related damped Jacobian method with an adaptive $\delta^{(l)}$.

It is typical to all existing implicit sea ice solvers [56] that the desired residual cannot be reached in all time steps. Usually, a simulation will not be canceled after one time step fails, but it will be continued based on a coarse approximation in the "hope" that the solver will recover within the following steps. As the sea ice problem is mostly data driven, with forcing from ocean, atmosphere and thermodynamics, this approach is reasonable.

The operator-related Newton scheme has no computational overhead. As it is able to accelerate the convergence, it can reduce the required number of Newton iterations and therefore decrease the overall computational cost. Every change of the damping parameter δ will call for a new assembly of the Jacobian. This may limit the possible savings. In Chapter 6, we evaluate the operator-related damped Jacobian method on a realistic arctic simulation and compare it to the currently applied *Jacobian-free Newton-Krylov* (JFNK) solver.

Analysis of the operator-related damped Jacobian method

Figure 5.2 shows that the standard Newton scheme with line search globalization (with and without inexact solving of the linear systems) fails after $n = 160$ time steps on an 8km mesh. On finer meshes, these approaches fail even quicker. Therefore, we declare this time step $n = 160$ as difficult and analyze the convergence behavior of the operator related Jacobian method in this single step.

In Figure 5.3, we show the relation between the damping parameter $\delta^{(l)}$ and the reduction rate Q_l (5.27) of the operator-related damped Jacobian method for different mesh resolutions. For large reduction rates close to 1, the damping parameter $\delta^{(l)}$ is decreased. Whenever the Newton convergence is good and the reduction rates are small, $\delta^{(l)}$ is enlarged and chosen close to 1. The operator-related damped Jacobian method is robust under mesh refinement. We obtain a reduction of the nonlinear residual on all mesh levels. The damping parameter, $\delta^{(l)}$, is not decreased on finer mesh, see Figure 5.3.

Even if $\delta^{(l)}$ is close to 1 in all iterations shown in Figure 5.3, an adaptive control of the Jacobian is necessary to achieve convergence. For comparison, we show the behavior of the full Newton method in Figure 5.4. Comparing the convergence of the Newton method without adaptive δ control (Figure 5.4) to the Newton method with adaptive δ control (lower two graphs of Figure 5.3), we find that adaptive control is necessary. Without the proper choice of the damping parameter $\delta^{(l)}$, no convergence to the required tolerance is obtained.

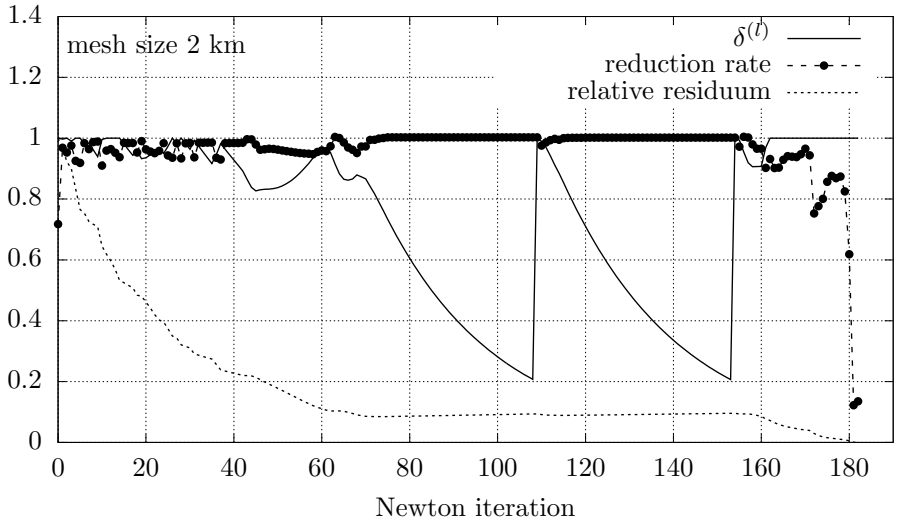


Figure 5.5.: Residual of the operator-related damped Jacobian method at time step $n = 280$. The parameter $\delta^{(l)}$ reaches the limit $\delta^{(l)} < 0.2$ twice and is reset to 1.

Finally in Figure 5.5, we present the damping of the operator-related damped Jacobian method. As the operator-related damped Jacobian method converges in time step $n = 160$ with less than 60 iterations on the 2 km mesh, we analyze time step $n = 280$, exemplary for a time step in which the solver needs a lot of iterations to converge.

One problem of Turek's original damping strategy is that too low values of $\delta^{(l)}$ will limit the possible convergence to that of the simple Picard iteration. This results in reduction rates so close to 1 that the adaptive strategy (5.28) will never increase $\delta^{(l)}$ again. Therefore, once $\delta^{(l-1)}$ reaches the critical value of $\delta^{(l-1)} < 0.2$, we continue the procedure with $\delta^{(l)} = 1$. The idea is to restart the Newton scheme with a new initial value. In our test cases, we observed that the operator-related damped Jacobian method would not converge to the required tolerance within 200 iterations without limiting $\delta^{(l)}$.

Figure 5.5 shows a case where $\delta^{(l)}$ is reset to 1 twice. Using the operator-related Jacobian method with adaptive choice of $\delta^{(l)}$, we can reduce the residual to a required tolerance in a limited number of iterations. Initially, $\delta^{(l)} < 1$ must be chosen to obtain convergence. Finally, transition to fast quadratic convergence is reached.

Note that Lemieux et al. [53] allow maximal 100 nonlinear Newton steps per time step. It needs to be investigated which nonlinear and linear tolerances are suitable for a realistic setting and how many Newton steps are appropriate in practice. However, the operator-related damped Jacobian method provides a tool to calculate highly accurate solutions faster than the existing solvers.

5.4. Solving linear systems within the Newton scheme

In [57], the authors recommend calculating the solution of the linear systems at each iteration with a *generalized minimal residual method* (GMRES) preconditioned with *line successive over relaxation method* (LSOR). In the framework of the sea ice momentum equation, the GMRES method was first applied by Lemieux et al. [57]. The linear problems arising in each Newton or Picard iteration are very ill-conditioned. Losch and Lemieux [64] observe that the use of an indirect LSOR preconditioner is computationally expensive, in particular on high resolution meshes [53]. The authors propose to test an *incomplete lower upper factorization* (ILU) as preconditioner to reduce computational cost.

In this section, we introduce the *geometric multigrid method* as preconditioner to the GMRES method. The linear solver can be used for Newton and Picard solvers. In Section 5.5, we compare the performance of the multigrid preconditioner with the performance of an ILU preconditioner.

We have already presented the operator-related damped Jacobian method and the linearization of the momentum equation in Section 5.2. It remains to describe the solution of the arising linear systems of equations $\mathbf{Ax} = \mathbf{b}$. We denote the coefficient vector of the two velocity components by $\mathbf{x}$, the (possibly damped) Jacobian by $\mathbf{A}$ and the Newton residual by $\mathbf{b}$, see (5.3). The idea of the *multigrid preconditioner* is to enhance the convergence of the GMRES solver by applying an operator $\mathbf{P}_{MG} \approx \mathbf{A}^{-1}$, which transforms the linear system to

$$\mathbf{P}_{MG}\mathbf{Ax} = \mathbf{P}_{MG}\mathbf{b}.$$

The GMRES method is a Krylov method projects the residual $\mathbf{r}$ in the m dimensional subspace

$$K^m = (\mathbf{r}, \mathbf{P}_{MG}\mathbf{Ar}, (\mathbf{P}_{MG}\mathbf{A})^2\mathbf{r}, (\mathbf{P}_{MG}\mathbf{A})^3\mathbf{r}, .., (\mathbf{P}_{MG}\mathbf{A})^{m-1}\mathbf{r}), \quad \mathbf{r} = \mathbf{A} - \mathbf{b},$$

and approximate the solution in K^m. We use a Krylov space of dimension $m - 1 = 10$. and apply a start residual $\mathbf{r} = -\mathbf{b}$, with the first guess $\mathbf{x}_0 = 0$. The multigrid preconditioner is applied once for each iteration of the GMRES method. A detailed description of the GMRES method can be found in Saad [86].

Multigrid solution

Multigrid solvers are well established for the solution of various partial differential equations [23, 105]. Based on a hierarchical mesh setup, they offer the potential to circumvent the bad influence of the worsening conditioning on finer meshes. The main numerical effort consists in residual evaluations, i.e. matrix-vector products, and the smoother, a usually local operator that approximates the problem on every mesh-level. Depending on the specific choice of smoother, they are well-suited for parallelization [3, 41].

First, we assume that by

$$\Omega_h =: \Omega_L \succ \Omega_{L-1} \succ \cdots \succ \Omega_0,$$

a nested hierarchy of finite element meshes is given. An element $K \in \Omega_l$ stems from uniform refinement of an element $K' \in \Omega_{l-1}$ into four similar elements. This construction gives a nested hierarchy of finite element spaces

$$V_h =: V_L \supset V_{L-1} \supset \cdots \supset V_0.$$

Modifications for locally refined meshes are described by Becker and Braack [4]. On this sequence of function spaces, we define a sequence of linear problems

$$\mathbf{A}_l \mathbf{x}_l = \mathbf{b}_l, \quad l = 0, \ldots, L,$$

where $\mathbf{A}_l$ is the Jacobian evaluated on the mesh Ω_l. Then, the standard multigrid algorithm in the $(i+1)$-st iteration is given as

$$\mathbf{x}_L^{(0)} = 0, \quad \mathbf{x}_L^{(i+1)} = \mathcal{MG}(L, \mathbf{x}_L^{(i)}, \mathbf{b}_L) \quad i = 1, 2, \ldots, \tag{5.29}$$

where each iteration $\mathcal{MG}(l, \mathbf{x}_l, \mathbf{b}_l)$ is defined as

Algorithm 5 (Multigrid iteration $\mathcal{MG}(l, \mathbf{x}_l, \mathbf{b}_l)$)**.**

Coarse mesh if $l = 0$	$\mathbf{x}_0 = \mathbf{A}_0^{-1}\mathbf{b}_0$
otherwise iterate:	
1. Smooth	$\mathbf{s}_l = \mathcal{S}(\mathbf{x}_l, \mathbf{b}_l, \mathbf{A}_l)$
2. Residual	$\mathbf{d}_l = \mathbf{b}_l - \mathbf{A}_l \mathbf{s}_l$
3. Restrict	$\mathbf{d}_{l-1} = \mathcal{R}_l \mathbf{d}_l$
4. Coarse-mesh correction	$\mathbf{c}_{l-1} = \mathcal{MG}(l-1, 0, \mathbf{d}_{l-1})$
5. Prolongate	$\mathbf{c}_l = \mathbf{s}_l + \mathcal{P}_l \mathbf{c}_{l-1}$
6. Smooth	$\mathbf{x}_l = \mathcal{S}(\mathbf{c}_l, \mathbf{b}_l, \mathbf{A}_l)$

The multigrid iteration is recursively constructed. This is the *V-cycle* of multigrid, as the coarse-mesh solution is approximated by one single call in Step ***4***. On the coarsest mesh level $l = 0$, the linear system is usually solved exactly. In Section 5.5, we will see that a direct solver is not necessary but can be replaced by a few steps of an approximative iteration. The *restriction* $\mathcal{R}_l : V_l \to V_{l-1}$ is defined as the L^2-projection

$$d_{l-1} := \mathcal{R}_l d_l \in V_{l-1}, \quad (\mathcal{R}_l d_l, \phi_{l-1}) = (d_{l-1}, \phi_{l-1}) \quad \forall \phi_{l-1} \in V_{l-1}.$$

The *prolongation operator* $\mathcal{P}_l : V_{l-1} \to V_l$ is the embedding as long as the meshes are nested which is always the case in our setting. As $\mathbf{d}_l$ is the variational residual, it is possible to compute the restriction by local operations without inverting the mass matrix [79]. The *smoothing operation* $\mathcal{S}(\cdot)$ is a simple iterative scheme with the purpose of approximating all those high frequent error contributions, that are visible on mesh level Ω_l, but not anymore representable on the next coarse mesh level Ω_{l-1}.

Multigrid smoothing

As smoothing operator $\mathcal{S}(\cdot)$, we always consider a Richardson iteration, preconditioned with an ILU decomposition of the system matrix $\mathbf{A}$ on the corresponding mesh level. On level $l = 0, \ldots, L$, we introduce the iteration

$$\mathbf{s}_l^{(i)} = \mathcal{S}(\mathbf{s}_l^{(l-1)}, \mathbf{b}_l, \mathbf{A}_l) := \mathbf{s}_l^{(i-1)} + \tilde{\mathbf{R}}_l^{-1}\tilde{\mathbf{L}}_l^{-1}(\mathbf{b}_l - \mathbf{A}_l\mathbf{s}_l^{(i-1)}), \quad i = 1, \ldots, \nu, \tag{5.30}$$

where $\mathbf{A}_l \approx \tilde{\mathbf{L}}_l\tilde{\mathbf{R}}_l$ is the stabilized incomplete decomposition of $\mathbf{A}_l$ into lower left and upper right triangular matrices and by ν we denote the number of smoothing steps. For pre-smoothing and post-smoothing, we apply a small number of steps of this simple iteration (about $\nu_{pre} = \nu_{post} = 4$ each).

ILU preconditioner

By ILU preconditioning, we denote the iteration

$$\mathbf{x}_l^{(i)} = \mathbf{x}_l^{(i-1)} + \tilde{\mathbf{R}}_l^{-1}\tilde{\mathbf{L}}_l^{-1}(\mathbf{b}_l - \mathbf{A}_l\mathbf{x}_l^{(i-1)}), \quad i = 1, \ldots, \nu_{ILU}. \tag{5.31}$$

The same iteration is applied as multigrid smoother (5.30). For a fair comparison of the results, we choose $\nu_{ILU} = \nu_{pre} + \nu_{post}$. We use $\nu_{ILU} = 8$ steps of the ILU decomposition and therefore $\nu_{pre} = 4$ *pre-* and $\nu_{post} = 4$ *post-smoothing* steps in the multigrid method, to have similar computational cost.

5.5. Numerical evaluation of the multigrid preconditioner

In this section, we analyze the performance of the linear solver on the test problem defined in Section 2.6. We evaluate the performance of the multigrid method as preconditioner within the GMRES method and compare it to the performance of the ILU preconditioner. As the cost of existing linear solvers increases with increasing resolution [64], we investigate the behavior of the multigrid preconditioner at high spatial resolutions.

Comparison of a multigrid method and an ILU decomposition as preconditioner to GMRES method

For comparison of the ILU preconditioner with the multigrid preconditioner within the GMRES method, we evaluate the test case of Section 2.6 at time $t = 1\,\text{day}$ on a $2\,\text{km}$ mesh with $k = 0.5\,\text{h}$. We use $\nu_{ILU} = 8$ steps of the ILU decomposition and $\nu_{pre} = 4$ *pre-smoothing* and $\nu_{post} = 4$ *post-smoothing* steps in the multigrid method. The computational cost by handling the full mesh hierarchy in the multigrid method increases by about 30%.

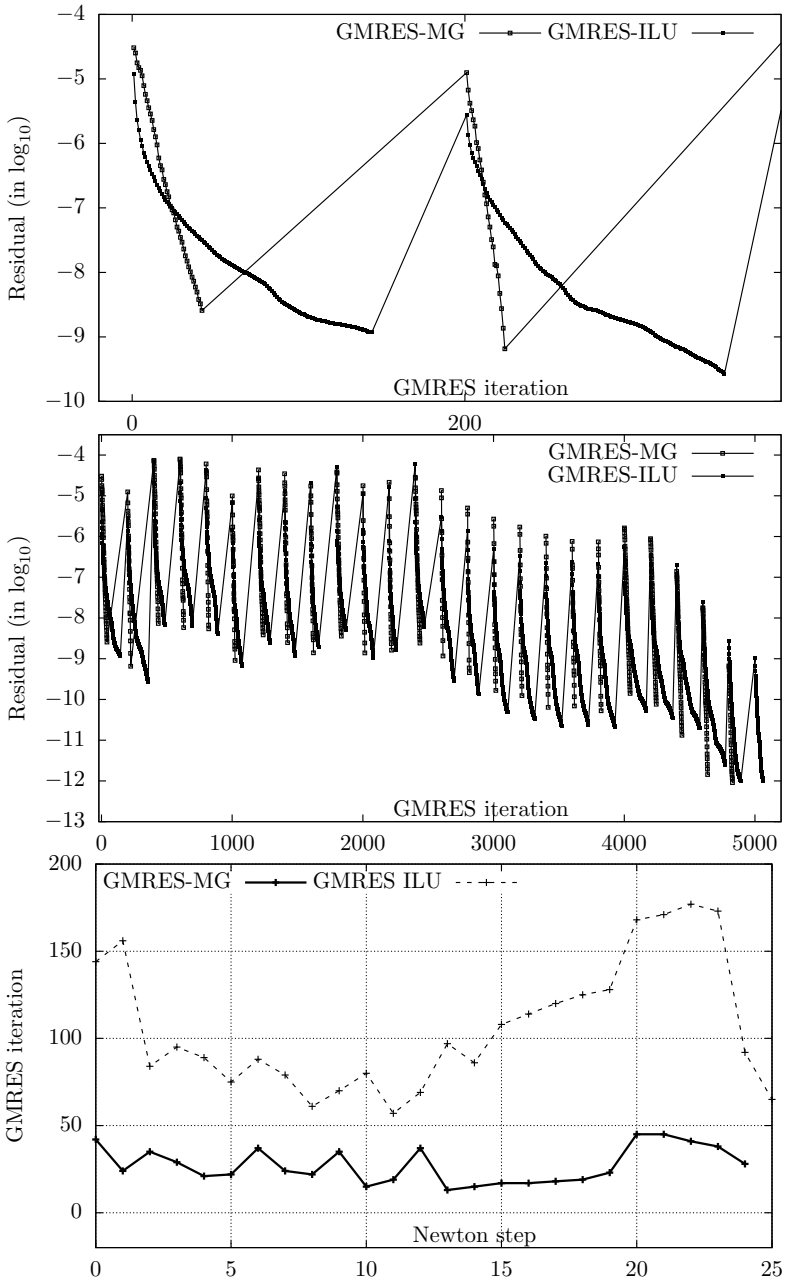


Figure 5.6.: Comparison of an ILU and a multigrid preconditioner. Evaluation of the test case at time $t = 1\,\text{day}$ on a 2 km mesh. Top: GMRES convergence of the first two Newton iterations. Middle: GMRES convergence over all Newton iterations of the time step. Bottom: summed number of GMRES iterations in every Newton iteration.

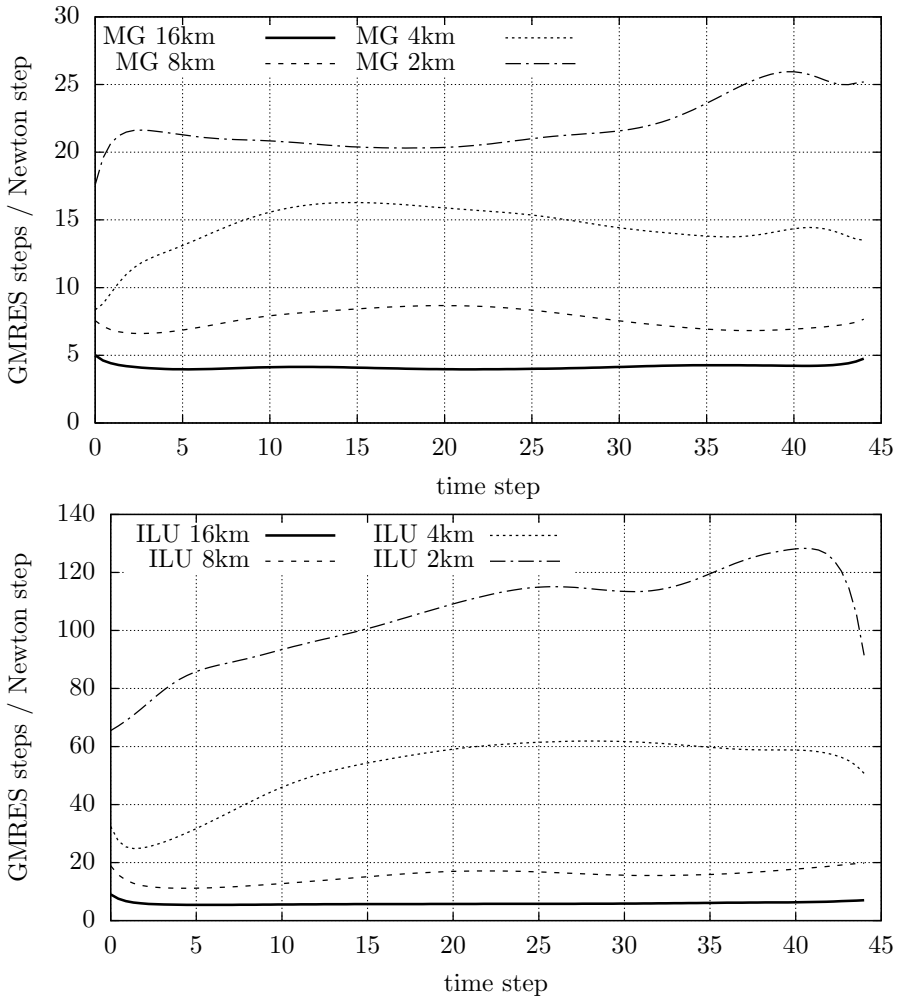


Figure 5.7.: We evaluate the test case of Section 2.6 on 16 km, 8 km, 4 km and 2 km meshes. The time step is set to $k = 0.5\,\mathrm{h}$. We display the average number of GMRES iterations per Newton iteration at each time step. Top: GMRES method preconditioned with the multigrid method. Bottom: GMRES method preconditioned with the ILU decomposition.

In Figure 5.6, we present the residual drop of the GMRES method in every Newton iteration. Using an ILU preconditioner, 26 Newton iterations are required. By applying a multigrid preconditioner, 25 Newton iterations are needed. No substantial difference in the nonlinear iteration count is expected as the linear problems are solved with high demands on the tolerance. In the upper plot of Figure 5.6, we show the GMRES convergence of the first two Newton iterations. The convergence of the GMRES solver preconditioned with the multigrid method is linear. The convergence of the GMRES solver with an ILU preconditioner is much slower.

In the middle plot of Figure 5.6, we observe linear convergence for all Newton steps of one time step. In the lower plot of Figure 5.6, we show the total number of GMRES iterations per Newton step at $t = 1\,\text{day}$. In average, we need 26 GMRES iterations per Newton step using a multigrid preconditioner and 106 GMRES iterations per Newton step by applying an ILU preconditioner.

Further, we analyze the behavior of the GMRES method under mesh refinement using an ILU decomposition and a multigrid method as preconditioner. In Figure 5.7, we evaluate the first 45 iterations (1 day) of the test case. We measure the average amount of GMRES iterations per Newton iteration in each time step. We introduce the GMRES-Newton ratio as the total amount of GMRES steps per time step divided by the total number of Newton iterations per time step.

In the upper plot of Figure 5.7, we show the performance of the GMRES solver preconditioned with a multigrid method at different mesh levels. The lower plot of Figure 5.7 displays the performance of the GMRES method preconditioned with ILU decomposition on refining meshes. We observe that on fine meshes, the GMRES-Newton ratio with an ILU preconditioner is up to factor 6 times higher than the GMRES-Newton ratio with a multigrid preconditioner. The convergence rate of the GMRES solver preconditioned with a multigrid method is very robust regarding mesh refinement, and shows only a small increase in the iteration count per Newton step. By using ILU preconditioning, we observe less robustness.

Coarse mesh solution

The multigrid theory asks for an exact - or at least accurate - approximation of the coarse mesh problem in Algorithm 5. Here, we test the robustness of the multigrid method with respect to different kinds of coarse mesh approximation. We solve the test case of Section 2.6 and choose a 2 km mesh with a time step size of $k = 0.5\,\text{h}$.

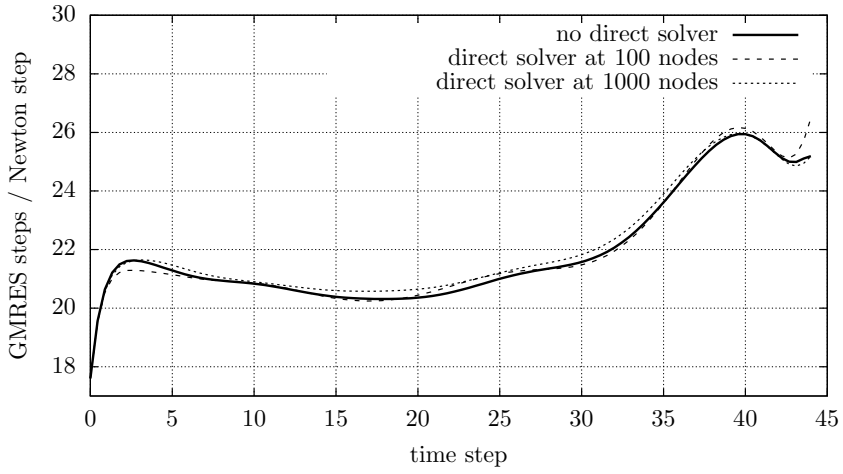


Figure 5.8.: Evaluating the test case of Section 2.6 using $k = 0.5\,\mathrm{h}$ and a 2 km mesh. We use a multigrid method with a direct solver at 1000 nodes and at 1000 nodes and without directly solving the coarse mesh problem.

We analyze the behavior of the multigrid preconditioner under application of a direct solver on different coarse mesh levels. In Figure 5.8, we show the GMRES-Newton ratio under application of a direct solver at 100 nodes, at 1000 nodes and without directly solving the problem. The solver performance does not change in any case. This shows that the multigrid method is robust with respect to the coarse mesh problem. It is not necessary to directly solve the coarse problem. Thus, we recommend using the multigrid method without directly solving the problem.

Number of pre and post smoothing steps

We study the computational cost of the GMRES iteration. Usually, taking less pre-smoothing and post-smoothing steps (5.30) of the multigrid method is more efficient, while a higher iteration number increases the robustness of the solver. Most of the work is done during the smoothing operation, as it consists of matrix-vector products as well as the application of the incomplete decomposition.

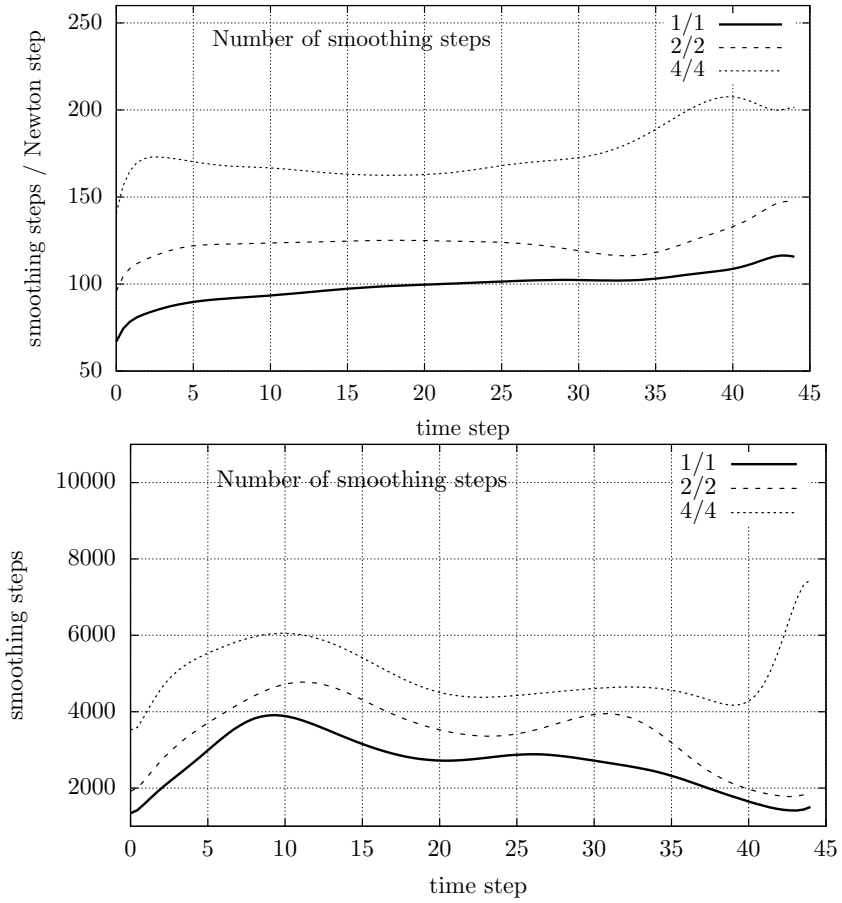


Figure 5.9.: We evaluate the test case of Section 2.6 using $k = 0.5\,\mathrm{h}$ and a 2 km mesh with different numbers of pre-smoothing and post-smoothing steps in the multigrid algorithm.

To analyze the computational cost, we compare in Figure 5.9 the total number of smoothing steps per Newton iteration and the total number of smoothing steps per time step. The average amount of smoothing steps per Newton iteration increases by increasing the number of smoothing steps. Analogously, the average amount of smoothing steps per time step increases by increasing the number of smoothing steps. The overall cost is best when using only one step for pre- and post-smoothing. Our experience shows best robustness – while still obtaining good performance – if 4 pre-smoothing and 4 post-smoothing steps are taken.

5.6. Conclusion

The nonlinearity of the sea ice momentum equation, coming from the viscous-plastic material tensor $\boldsymbol{\sigma}(\mathbf{v}^n)$, is severe and the efficient treatment of this term is still under active research [39]. Picard or Newton solvers are mainly used to solve implicit discretizations of the momentum equation. The linear systems resulting from Newton linearization are very stiff and usually approximated with the GMRES method using different preconditioners like a LSOR method [57].

In this chapter, we have presented and tested a Newton solver, accelerated with the *operator-related damped Jacobian* method and globalized with a *line search method*. Further, we introduced a multigrid preconditioner to the GMRES method for solving the linear subproblems. The idea of the *operator-related damped Jacobian* method is to combine a Newton solver with a Picard method. The method is based on the analytical evaluation of the Jacobian. We proved that the Jacobian of the viscous-plastic model is a positive definite matrix. For a positive definite Jacobian, a damped Newton iteration will converge globally if a good damping parameter is chosen, see [16]. The matrix is symmetric apart from the Coriolis term. Given an explicit treatment as used by Zhang and Hibler [34], this allows for simpler solvers like Conjugate Gradient method [86]. Further, we showed that the Jacobian of the sea ice model can be divided into a positive definite part and a negative semidefinite matrix, which contains the derivatives of the viscosities. By an adaptive controlling of the negative semidefinite part in the Jacobian, we were able to accelerate the convergence of the Newton iteration. In the regime of quadratic convergence, rapid decline of the residual is reached by a transition to the full Jacobian.

We compared our globalization strategy to a standard Newton method with a line search method and a Newton method with an inexact solving of the linear subproblem and a line search damping. The second approach is used in the framework of the *Jacobian-free Newton Krylov* (JFNK) solver [54]. Using the model probelm introduced in 2.6 we verified that Newton solver accelerated with the operator-related damped Jacobian method gives better robustness. This is particularly true on fine meshes. In contrast to the latter two approaches, we could solve the test problem on an 8 km mesh and a 4 km mesh without failures. On a 2 km mesh, the inexact Newton method with the operator-related damped Jacobian method and line search globalization only fails 6 out of 350 time steps. Here, we emphasize that "failure" just means, we did not reach the desired tolerance within a fixed number of iterations. On all mesh levels, the other globalization strategies failed many times in a row without recovery, such that the problem can no longer be considered as "solved". By evaluating the model problem, we show that the operator-related damped Jacobian method is robust under mesh refining as the damping parameter $\delta^{(l)}$ is not chosen to be smaller on fine meshes. In Chapter 6, we will compare the operator-related damped Jacobian method to the JFNK solver evaluating a simulation computed on a pan-Arctic.

As the currently used linear solver in the JFNK solver is numerically costly, we introduced a geometric multigrid method as preconditioner to the GMRES method. We

showed that the convergence rate of the multigrid method is robust with regard to mesh refinement. This makes it an appealing method for sea ice simulations at high resolutions. We found that it is not necessary to directly solve the coarse mesh problem of the multigrid method. In terms of robustness and performance, we recommend 4 pre-smoothing and 4 post-smoothing steps. We compared the residual reduction of the GMRES method with ILU and multigrid preconditioning. The reduction of the GMRES with a multigrid preconditioner is linear and much faster than using an ILU preconditioner. With increasing spatial resolution the ILU preconditioner is less robust than the multigrid preconditioner. Especially on fine meshes, the multigrid preconditioner reduces the iteration counts of the GMRES method by 80 % in comparison to an ILU preconditioner.

6. Performance of the operator-related damped Jacobian method in a stand-alone sea ice model

In this section, we describe the implementation of the operator-related damped Jacobian method into the software framework used by the working group of Prof. Tremblay at the McGill University. In this framework, the stand-alone sea ice model is applied to investigate seasonal and sub-seasonal forecasts of sea ice conditions. The model is implemented in FORTRAN and applies the *Jacobian-free Newton-Krylow* solver (JFNK) to compute the solution of the momentum equation. The aim of this section is to show that the *operator-related damped Jacobian* method works in a realistic setting, and to compare its performance to the globalization and acceleration approaches of the JFNK solver, that are implemented in the McGill sea ice model.

A detailed derivation of the operator-related damped Jacobian method can be found in Section 5.2. The JFNK solver is a modified Newton method that has been introduced by Lemieux et al. [56] to solve the viscous-plastic (VP) sea ice momentum equation. Besides the stand-alone sea ice models [53, 54, 56], the JFNK solver is applied in global climate models [39, 63, 64]. The McGill sea ice model is based on a finite difference discretization in space and an implicit Euler approximation in time. To save computational costs in the Newton scheme, a Jacobian-free approach is applied. Here, the matrix vector product of the Jacobian and the direction of descent of the Newton scheme are approximated by a differential quotient. The linear problems arising in each Newton iteration are solved with a *flexible generalized minimal residual* method (FGMRES) [57] that is preconditioned with a *line successive over-relaxation* method (LSOR) [57]. Further, the Newton method is globalized with a *line search* approach and accelerated by an *inexact solving of the linear subproblems* [53]. A detailed description of the globalization and acceleration approach is given in Section 5.2. In the standard version of the software, the balance laws are advanced in time with a first order upwind scheme. Likewise, a second order backward difference scheme can be applied.

This chapter is structured as follows. In Section 6.1, we briefly recapitulate the idea of the Jacobian-free Newton approach and explain how the operator-related damped Jacobian method is realized in this framework. Section 6.2 is devoted to the numerical evaluation of the JFNK solver accelerated with the operator-related damped Jacobian method. The chapter closes with a conclusion.

6.1. The operator-related damped Jacobian method in a Jacobian-free Newton solver

We begin with a brief introduction to the Jacobian-free Newton solver, before we present the operator-related damped Jacobian method for this approach. As it is done for the presentation of the Newton scheme in Section 5.1, we introduce the following abstract notation of the fully discretized momentum equation

$$\mathbf{A}(\mathbf{v}_n) = \mathbf{b},$$

where $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^N$ contains all terms of the momentum equation that depend on the velocity $\mathbf{v}$. The right hand side of the momentum equation $\mathbf{b}$ includes all terms that are independent of $\mathbf{v}$. In order to apply Newton's method, we introduce the residual as

$$\mathbf{F}(\mathbf{v}_n) = \mathbf{A}(\mathbf{v}_n) - \mathbf{b}. \tag{6.1}$$

Thus, the l-th Newton iteration at time step n reads as

$$\begin{aligned} \mathbf{F}'(\mathbf{v}_n^{(l-1)})\mathbf{w}_n^{(l)} &= -\mathbf{F}(\mathbf{v}_n^{(l-1)}), \\ \mathbf{v}_n^{(l)} &= \mathbf{v}_n^{(l-1)} + \omega \mathbf{w}_n^{(l)}, \quad l = 1, 2, ... \end{aligned} \tag{6.2}$$

Here, the initial guess for Newton's method is the solution of the last time step $\mathbf{v}_n^{(0)} = \mathbf{v}_{n-1}$. In (6.2), we denote the Jacobian by $\mathbf{F}'$ and the Newton update by $\mathbf{w}$. Further, by ω we describe the line search parameter $\omega \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$. The line search parameter is reduced until $||\mathbf{F}(\mathbf{v}_n^{(l)})|| \leq ||\mathbf{F}(\mathbf{v}_n^{(l-1)})||$ or $\omega = \frac{1}{8}$ is reached. Here, by $||\cdot||$ we refer to the Euclidean norm.

The linear system $\mathbf{F}'(\mathbf{v}_n^{(l-1)})\mathbf{w}_n^{(l)} = -\mathbf{F}(\mathbf{v}_n^{(l-1)})$ is solved with a *flexible generalized minimal residual method* (FGMRES). More details on the solver and the discretization can be found in the work of Lemieux et al. [57].

The matrix vector multiplication arising in (6.2) is approximated by

$$\mathbf{F}'(\mathbf{v}_n^{(l-1)})\mathbf{w}_n^{(l)} \approx \frac{\mathbf{F}(\mathbf{v}_n^{(l-1)} + \gamma_\epsilon \mathbf{w}_n^{(l)}) - \mathbf{F}(\mathbf{v}_n^{(l-1)})}{\gamma_\epsilon}.$$

We choose $\gamma_\epsilon = 10^{-7}$ as it is suggested in Losch et al. [64]. In order to derive the operator-related damped Jacobian method for the Jacobian-free approach, we recall the viscous-plastic rheology.

$$\begin{aligned} \boldsymbol{\sigma} &= 2\eta\dot{\boldsymbol{\epsilon}}' + \zeta \operatorname{tr}(\dot{\boldsymbol{\epsilon}})I - \frac{P}{2}I, \\ \eta &= \frac{\zeta}{4}, \quad \zeta = \frac{P}{2\Delta(\dot{\boldsymbol{\epsilon}})}, \quad \Delta(\dot{\boldsymbol{\epsilon}}) := \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}. \end{aligned}$$

The idea of the operator-related damped Jacobian method is to split the Jacobian of the momentum equation (2.24) into two parts

$$\mathbf{F}' = \mathbf{F}'_1 + \delta \mathbf{F}'_2, \tag{6.3}$$

where $\mathbf{F}'_2$ contains the derivative of the viscosities. Here, $\mathbf{F}_2$ is approximated by the forward differential quotient

$$\begin{aligned}\mathbf{F}'_2(\mathbf{v}_n^{(l-1)})\mathbf{w}_n^{(l)} \approx & \frac{\zeta(\mathbf{v}_n^{(l-1)} + \gamma_\epsilon \mathbf{w}_n^{(l)}) - \zeta(\mathbf{v}_n^{(l-1)})}{\gamma_\epsilon} \operatorname{tr}\left(\dot{\boldsymbol{\epsilon}}\left(\mathbf{v}_n^{(l-1)}\right)\right) I \\ & + \frac{\eta(\mathbf{v}_n^{(l-1)} + \gamma_\epsilon \mathbf{w}_n^{(l)}) - \eta(\mathbf{v}_n^{(l-1)})}{\gamma_\epsilon} 2\dot{\boldsymbol{\epsilon}}'\left(\mathbf{v}_n^{(l-1)}\right).\end{aligned}$$

All the remaining derivatives of the momentum equation (2.24) are stored in $\mathbf{F}'_1$ and computed as it was done in the original Jacobian-free approach from Lemieux et al. [56].

As described in Section 5.2, we define the control parameter δ in (6.3) in the l-th Newton iteration as

$$\delta^{(l)} = \begin{cases} \min\left\{1, \delta^{(l-1)}\left(0.2 + \frac{4}{0.7+\exp(1.5Q_l)}\right)\right\} & \delta^{(l-1)} \geq \delta_{\min} = 0.55, \\ 1 & \delta^{(l-1)} < \delta_{\min}, \end{cases} \tag{6.4}$$

where Q_l is the reduction rate. In comparison to (5.28) in Section 5.2, we modified δ_{min} to reduce the computational costs of the nonlinear solver while maintaining its stability. A detailed analysis of the parameter setting is given in Section 6.2.1.

As in the model problem introduced in Section 2.6, we apply a maximal number of 200 Newton iterations per time step and set the nonlinear tolerance as

$$\mathbf{F}||(\mathbf{v}_n^{(l)})|| \leq \text{tol}_{nl}||\mathbf{F}(\mathbf{v}_n^{(0)})||, \quad \text{tol}_{nl} = 10^{-3}. \tag{6.5}$$

Further, we allow a maximum of 100 GMRES iterations per Newton step. The linear tolerance of the GMRES method is chosen as

$$||\mathbf{F}'(\mathbf{v}_n^{(l-1)})\mathbf{w}_n^{(l)} + \mathbf{F}(\mathbf{v}_n^{(l-1)})|| \leq \text{tol}||\mathbf{F}(\mathbf{v}_n^{(l-1)})||, \quad \text{tol} = 0.67. \tag{6.6}$$

6.2. Numerical analysis of the operator-related damped Jacobian method in a stand-alone sea ice model

The aim of this analysis is twofold. First, we show that the operator-related damped Jacobian method works in a realistic setting. Second, we compare its performance to the globalization and acceleration methods that are implemented in the McGill software. In particular, we compare it to a *line search approach* and an *inexact solving of the linear problem*, which are introduced in Section 5.2.

Our analysis is based on the pan-Arctic sea ice model provided by the working group of Prof. Tremblay at McGill University in Montreal. The model includes a viscous-plastic

tol	Newton	GMRES	failure
0.01	775	22715	0
0.1	701	10665	0
0.25	1364	13827	4
0.67	856	6043	0
0.7	850	5944	0
0.75	870	5502	0

Table 6.1.: For $k = 2$ h, we analyze the number of Newton and GMRES iterations achieved with operator-related damped JFNK solver for varying linear tolerances tol (6.6). We set the minimal control parameter of the operator-related damped Jacobian method to $\delta_{\min} = 0.55$. The simulation covers one day on a 10 km mesh, starting at 01.01.2002.

δ_{min}	tol	Newton	GMRES	failure
0.7	0.67	31803	248615	5
0.7	0.6	30213	256590	8
0.6	0.67	30126	238178	3
0.6	0.6	28952	249249	7
0.6	0.5	28886	272798	18
0.55	0.67	29657	238032	1
0.5	0.67	29735	238047	2

Table 6.2.: We analyze the number of Newton and GMRES iterations achieved with the operator-related damped JFNK solver by applying different values for the minimum control parameter $\delta_{\min}$ (6.4) and a variation of the linear tolerances tol (6.6). We cover one month of simulation, starting on 01.01.2002 with $k = 2$ h on a 10 km mesh.

sea ice model that is driven by atmospheric data. For our test we set ocean currents and thermodynamic fluxes to zero. A more detailed description of the model is given by Lemieux et al. [53]. We simulate the period from 01.01.2002 to 01.02.2002 which according to the authors shows typical and challenging conditions of the Arctic. Our analysis is done on a 40 km, 20 km and 10 km mesh, where the 10 km mesh is the highest available mesh resolution of the model. The tolerances are set as described in (6.5) and (6.6).

6.2.1. Parameter setting in operator-related damped Jacobian method

In this section, we adjust the parameter setting of the operator-related damped Jacobian method, in order to apply it in the stand-alone sea ice model. For our analysis, we use a 10 km mesh and a time step of $k = 2$ h. We simulate the sea ice dynamics over one

month, starting on 01.01.2002. First, we evaluate the behavior of the *operator-related damped* JFNK solver globalized with a *line search approach* and varying linear tolerance tol (6.6). We analyze the number of Newton and GMRES iterations as well as the number of failures arising in a simulation. We consider the method to fail if the solver cannot reduce the residual of the momentum equation to the desired tolerance in 200 iterations.

In Table 6.1, we present the number of Newton and GMRES iterations that are computed with the *operator-related damped* JFNK solver in a $T = 1$ day simulation. The solvers show the same robustness for high tolerances ($tol \approx 0.001$) and low tolerances (about $tol \approx 0.75$). However, Lemieux et al. [54] observed a better robustness of the JFNK solver when using low linear tolerances close to 1 and applying *an adaptive control of the linear tolerances*. In Section 5.3, we analyzed the solver robustness of the operator-related damped Jacobian method on an idealized test case. We observe that the approach fails on 4 km and 2 km mesh if the linear tolerance is set close to 1. In this test case, we find that the nonlinear solver is more robust if the linear problem is solved with high demands on the accuracy, i.e., $tol = 0.0001$. In contrast to the JFNK solver, our approach approximates the linear problems with a multigrid method as preconditioner to the GMRES solver, see Section 5.4. This strongly reduces the computational costs in comparison to an ILU preconditioner. Therefore, we suggest testing the multigrid method as preconditioner to the GMRES method in the JFNK framework.

Approximating the linear subproblems is the most costly part in solving the momentum equation in the McGill model. In order to use as few linear iterations as possible, we set the linear tolerance as low as possible ($tol \approx 1$) while maintaining stability of the nonlinear solver. In Table 6.2, we adjust the parameters in the *operator-related damped* JFNK solver and evaluate the number of Newton and GMRES iterations with different linear tolerances and varying values for $\delta_{\min}$ (6.4). In this test case, the simulation covers $T = 1$ month starting on 01.01.2002. We observe the most robust performance of the JNFK solver combined with the operator-related damped Jacobian method by choosing $\delta_{\min} = 0.55$ and setting the linear tolerance to tol$= 0.67$, see Table 6.2. Thus, we proceed with this parameter setting. For our purposes, this straightforward way to implement the operator-related damped Jacobian method in the McGill model is sufficient.

Note that we do not observe an enhanced robustness or a faster converging behavior of the JFNK solver by using a combination of *operator-related damped Jacobian* method, and an *adaptive control of the linear tolerance*.

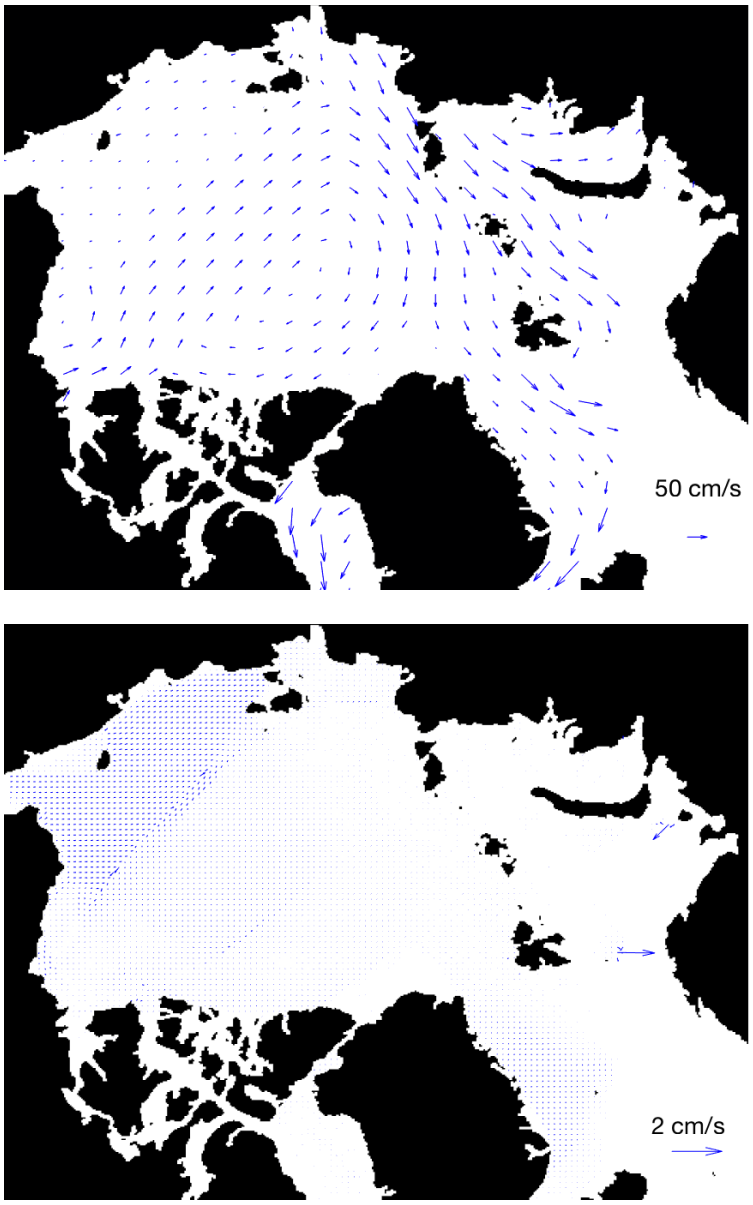


Figure 6.1.: Top: The velocity field achieved with the orJFNK solver after $T = 1$ month of simulation, starting on 01.01.2002. Bottom: The difference of the solutions calculated with the *orJFNK* and the *JFNKac* solver. We use $k = 2$ h and a 10 km mesh.

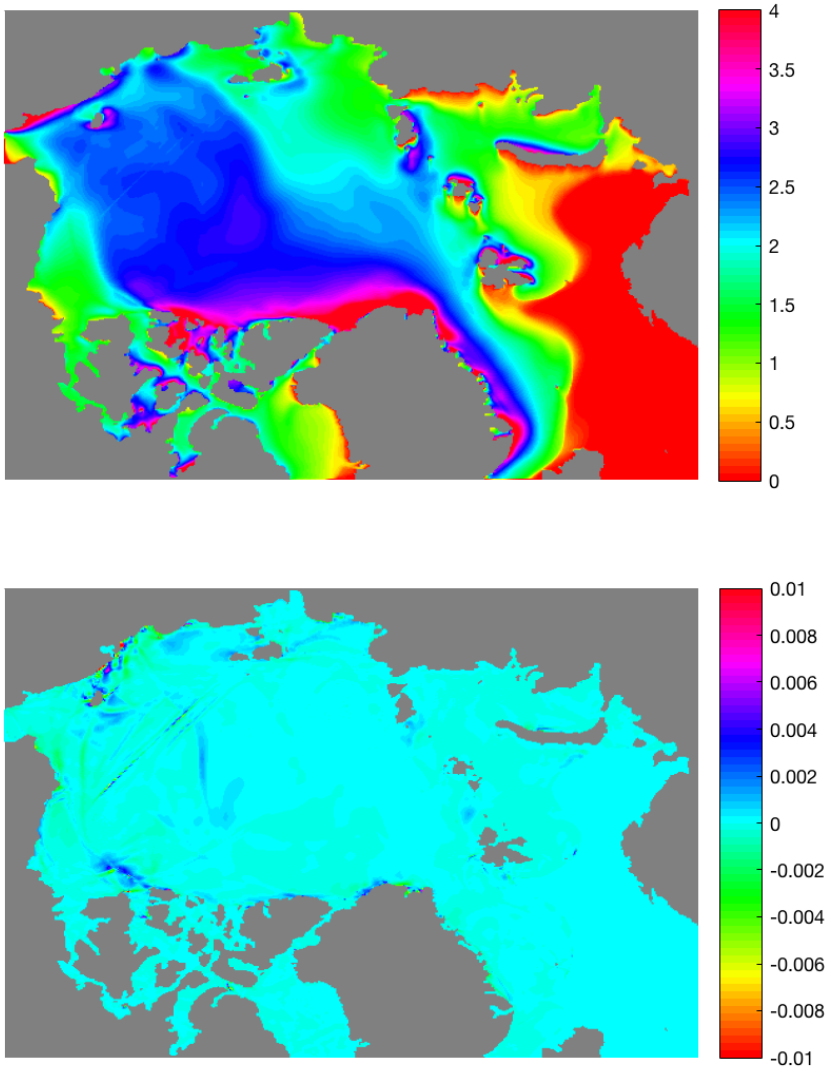


Figure 6.2.: Top: The ice thickness computed with the orJFNK solver after $T = 1$ month of simulation, starting on 01.01.2002. Bottom: The difference of the solutions calculated with the orJFNK solver and the JFNKac solver. We use a time step of $k = 2$ h and a 10 km mesh. The ice thickness is given in meter.

6.2.2. Solver comparison on varying spatial and temporal resolutions

In this section, we analyze the performance of the *operator-related damped* JFNK solver with a fixed tolerance for the linear problems (orJFNK). Further, we compare the orJFNK solver to a JFNK solver with a fixed linear tolerance (JFNKf) and to a JFNK solver with an *adaptive control* of the linear tolerance (JFNKac). The *JFNKac* solver is currently applied by Lemieux et al. [56] and Losch et al. [64] to solve the sea ice momentum equation. A line search globalization is used in all three inexact Newton methods.

We begin by comparing the velocity field and the ice thickness, computed with the *orJFNK* solver and the *JFNKac* method. In the upper image of Figure 6.1, we show the velocity field achieved after $T = 1$ month of simulation. In the lower image, we present the difference of the velocity field computed with the *orJFNK* solver and the *JFNKac* solver. The difference is taken on 31.01.2002. During the simulation, the *orJFNK* solver failed twice and the *JFNKac* method failed 90 times. We observe a small difference in the velocity field up to 2cm/s which is around 5% in comparison to the maximum ice velocity.

In the upper image in Figure 6.2, we present the ice thickness distribution on 31.01.2002, computed with the *orJFNK* solver. In the lower image of Figure 6.2, we show the differences in the ice thickness field computed with the *orJFNK* solver and the *JFNKac* method. We observe differences locally up to 1 cm which is about 0.1% of the maximum ice thickness. The differences are based on the failures of the solvers.

Actually, the choice of the solver should not have an impact on the solution. However, so far, no robust solvers exist to compute an accurate numerical solution of the viscous-plastic sea ice model such that different implementations produce different approximations. Further, the different discretizations of the sea ice equations are considered as different sea ice models.

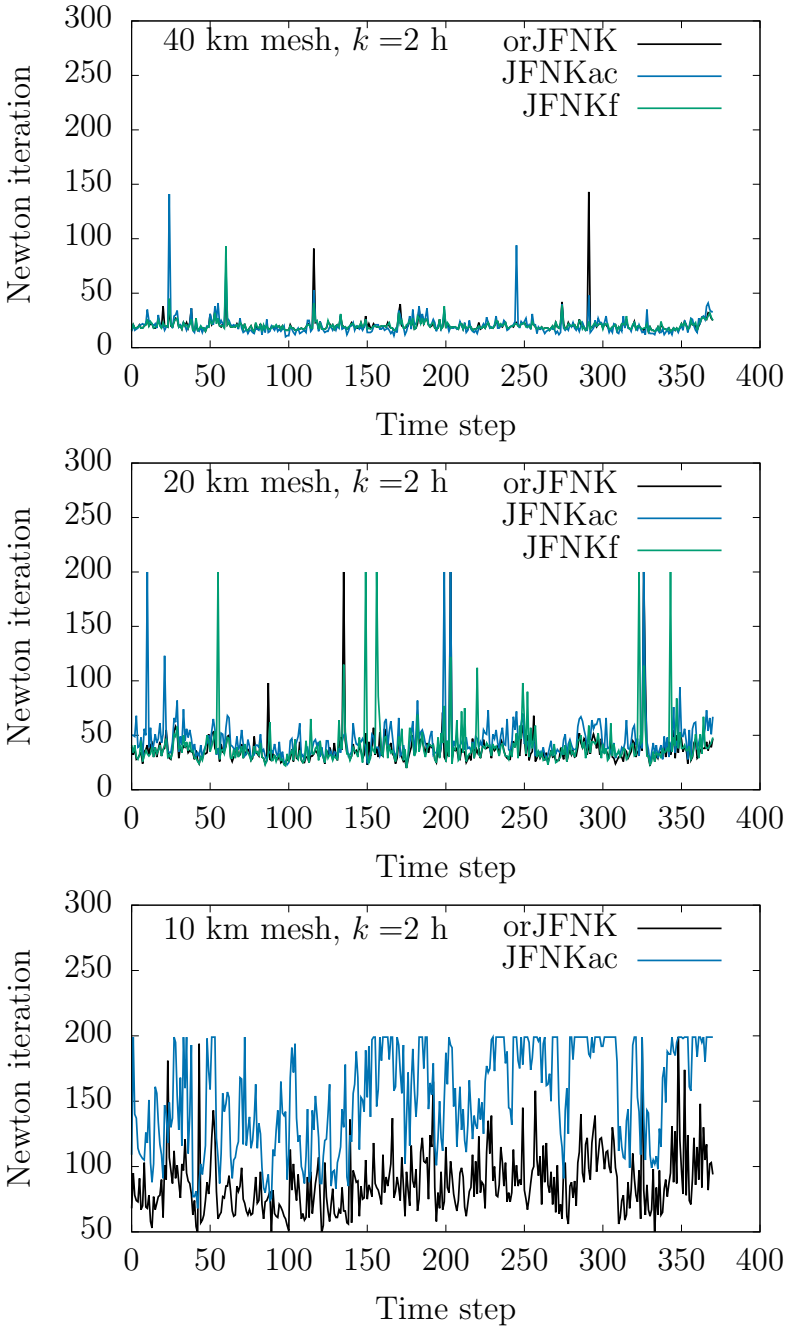


Figure 6.3.: We compare the JFNK solver with a *fixed linear tolerance* (JFNKf) to an *operator-related damped* JFNK solver (orJFNK) with *fixed linear tolerance* and to the JFNK solver with *adaptive control of the linear tolerance* (JFNKac).

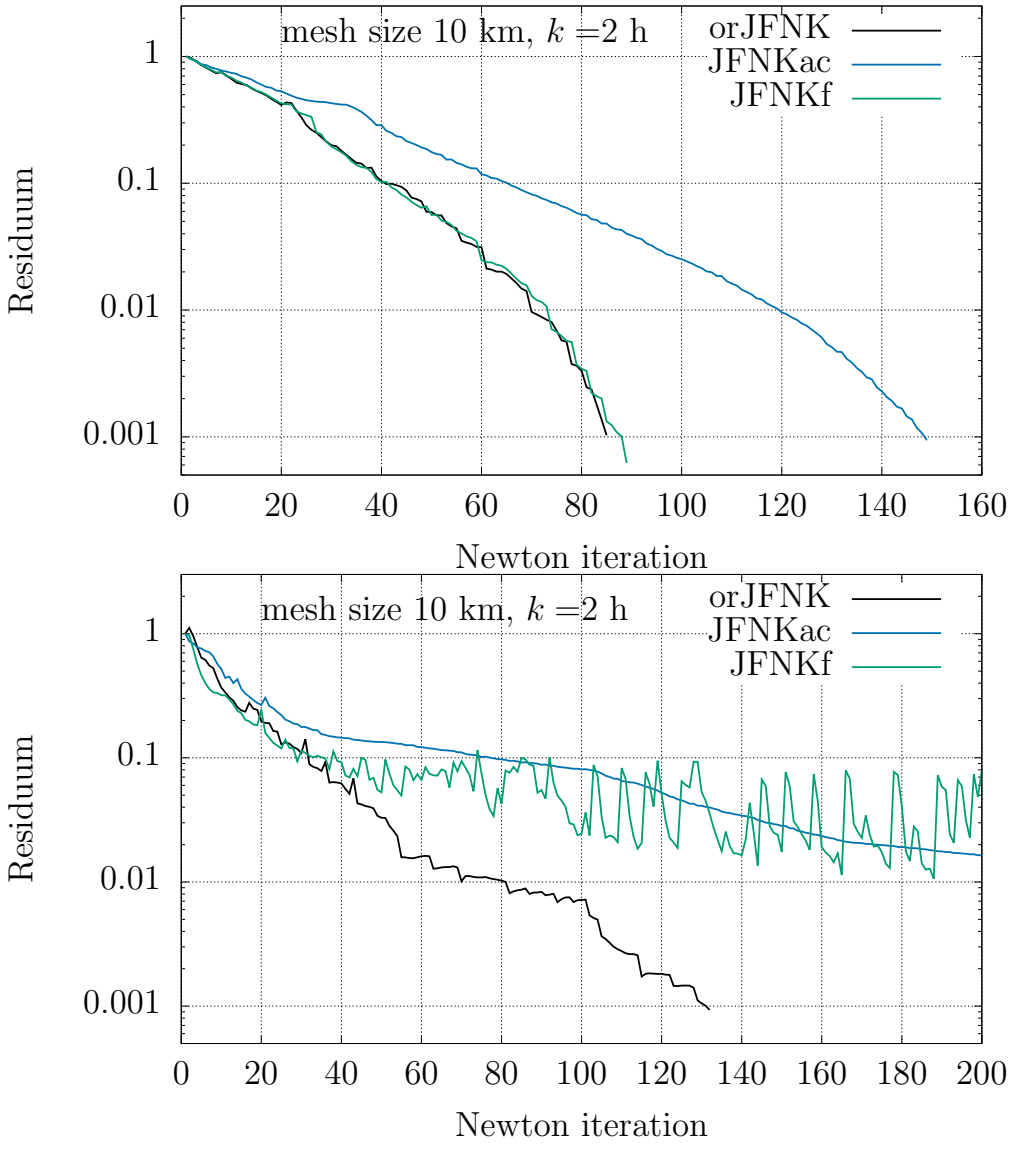


Figure 6.4.: We compare the JFNK solver with a *fixed linear tolerance* (JFNKf) to the *operator-related damped* JFNK solver (orJFNK) with *fixed linear tolerance* and to the JFNK solver with *adaptive control of the linear tolerance* (JFNKac). Here, we plot the residuum reduction per Newton iteration for one time step in logarithmic scale. A maximal number of 200 Newton iterations is applied. The residual is normalized by the initial residual.

40 km		Newton	GMRES	CPU [s]	failure
	JFNKf	7560	25881	524	0
	JFNKac	7614	28159	929	0
	orJFNK	7689	25612	593	0
20 km					
	JFNKf	14008	70008	9184	5
	JFNKac	16060	56596	7635	4
	orJFNK	13229	64787	9160	3
10 km					
	JFNKf	32572	313581	159590	60 (16%)
	JFNKac	57097	210371	98151	90 (24%)
	orJFNK	32428	260334	128660	02 (0.5%)

Table 6.3.: We simulate $T = 1$ month with $k = 2$ h. A time step is considered as a failure if the desired residual can not be reached in 200 Newton iterations.

time step	solver	Newton	GMRES	failure
2 h	JFNKac	57097	210371	90 (24%)
	orJFNK	32428	260334	2 (0.5%)
0.5 h	JFNKac	117473	490451	29 (1.9%)
	orJFNK	70560	519779	5 (0.3 %)
0.16 h	JFNKac	183485	822605	84 (1.8%)
	orJFNK	134077	838770	39 (0.8%)

Table 6.4.: We use a 10 km mesh and simulate the sea ice dynamics $T = 1$ month.

time step	solver	Newton	GMRES	failure
2 h	JFNKac	59100	275183	105 (28%)
	or JFNK	34497	340984	13 (3.5 %)
0.5 h	JFNKac	181229	684602	19 (1.27%)
	orJFNK	67429	498942	6 (0.4 %)
0.16 h	JFNKac	180561	800809	50 (1.12%)
	orJFNK	134260	850155	26 (%0.5)

Table 6.5.: We use the monotholitic solving approach presented in Algorithm 2 in Section 3.1 to simulate the sea ice dynamics for $T = 1$ month on a 10 km mesh.

Solver comparison on increasing spatial resolutions In the following, we analyze the performance of the JFNK solver accelerated with the *operator-related damped Jacobian* method (orJFNK), the JFNK solver accelerated with an *adaptive control of the linear tolerance* (JFNKac) and the JFNK solver with a *fixed linear tolerance* (JFNKf).

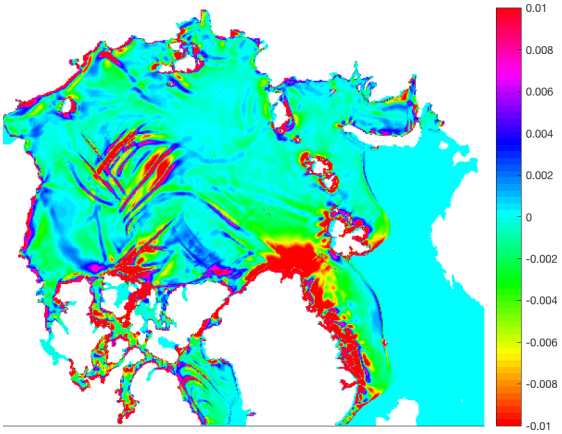
In Figure 6.3 we compare the number of Newton steps applied by the different approaches to solve the sea ice model for $T = 1$ month on 40 km, 20 km and 10 km meshes. We observe that the number of Newton iterations increases on higher resolutions and find that the *orJFNK* solver needs less Newton iterations than the other two solvers in nearly all the time steps, see Figure 6.3. We proceed with analyzing the robustness of the solvers on 40 km, 20 km and 10 km meshes. As shown in Table 6.3, the robustness of the solver decreases with increasing resolution. On a 10 km mesh, we observe the biggest difference in terms of solver robustness. The orJFNK solver fails in less than 1% of the iterations, whereas the other two approaches fail in more than 15% of the iterations, see Table 6.3.

Next, we compare the computational time that the three solvers need to solve the test case on different mesh levels. In Table 6.3, we present the CPU time of the solvers. On the 40 km mesh the JFNKac solver is factor 1.7 slower than the other solvers, but on the other meshes the JFNKac solver is up to 38% faster. This is reflected in the number of GMRES iterations that are applied by each solver. Even if the *JFNKac* solver applies more Newton iterations than the other two solvers, the much smaller number of GMRES iterations makes a quicker performance possible, see Table 6.3.

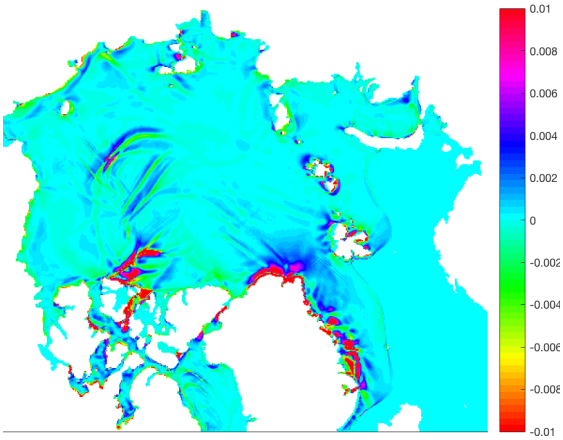
In Figure 6.4, we show the relative residuum reduction of the three solvers in one time step on a 10 km mesh. In the upper graph, we plot the performance in time step $n = 61$, a case in which all three solvers converge. To reach the desired tolerance, the *JFNKac* solver needs double the amount of Newton iterations, but requires 27% less GMRES iterations than the other two solvers.

In the lower graph in Figure (6.4), we present time step $n = 345$, a time step in which the *JFNKac* solver and the *JFNKf* solver fail. In between 200 iterations, only the *orJFNK* method reduces the residuum to the desired tolerance. The *orJFNK* solver requires 55 Newton steps and 435 GMRES iterations to reach the tolerance at which the *JFNKac* solver fails. Here, the *JFNKac* solver applies 200 Newton steps and 571 GMRES iterations.

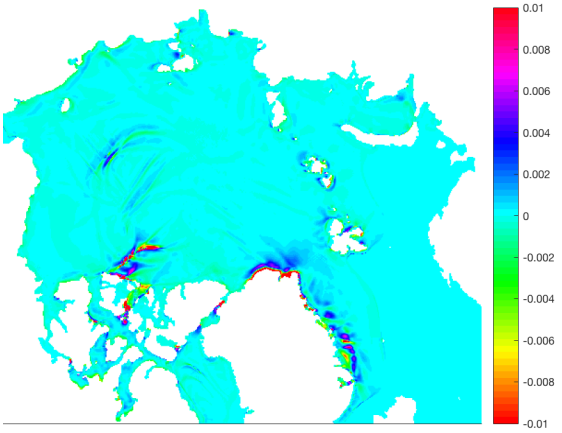
Solver comparison with reducing time steps We analyze the robustness of the *orJFNK* solver and the *JFNKac* solver with increasing temporal resolution. In Table 6.4, we evaluate the number of Newton and GMRES iterations after $T = 1$ month of simulation. The robustness of the *JFNKac* solver increases with smaller time steps. The failure rate decreases from 24% to 1.8%. Using the *orJFNK* solver, the failure rate is below 1% for all considered time steps. With a decreasing time step, the number of GMRES and Newton iterations are increase by factor 2 to factor 4. For all considered time steps, the *orJFNK* is solver is more robust than the *JFNKac* solver.



$k = 2$ h



$k = 0.5$ h



$k = 0.16$ h

Figure 6.5.: We simulate the sea ice dynamics for 8 days on a 10 km mesh and plot the difference of the ice thickness A calculated with the orJFNK solver in a partitioned and a monotholitic solution approach. A is given in meters.

6.2.3. Operator-related damped Jacobian method in a partitioned and monotholitic solution approach

We compare the performance of the operator-related damped JFNK solver in a *monotholitic* and a *partitioned* solution approach. In the partitioned solution approach (SIT) the system of equations is decoupled such that first the momentum equation and then the balance laws are solved [53]. By decoupling the system, a *splitting error* is introduced. This error would be avoided by solving the momentum and the balance laws simultaneously in a monotholitic approach. The IMEX method introduced by Lemieux et al. [53] and described in Algorithm 2 in Section 3.1, is a monotholitic solution approach. It aims to reduce the splitting error by iteratively solving the sea ice momentum equation with updated balance laws in each Newton iteration. We run the system with the IMEX method and analyze the performance of the operator-related damped Newton method in the IMEX framework and we compare it to the behavior in the partitioned solution approach.

By comparing Table 6.4 and Table 6.5, we observe that for $k = 2$ h the *orJFNK* solver and the *JFNKac* solver are less robust in the monotholitic approach than in the partitioned solution method. For decreasing time steps, both solvers show the same robustness in the monotholitic and in the partitioned solution approach. As in the partitioned solution approach, the *orJFNK* solver is more robust than the *JFNKac* solver.

In Figure 6.5, we show the differences of ice thickness fields calculated with operator-related damped JFNK solver in the partitioned and in the monotholitic approach. The difference is taken after T=8 days of simulation. Considering $k = 2$ h, we find that the solution differs in larger areas especially on the boundaries and at cracks in the ice. By decreasing the time step, the difference in the solutions is reduced.

Finally, we compare Figure 6.2 and Figure 6.5 and observe that the difference in the solution conducted with a partitioned and a monotholitic approach seems to be bigger than the error of non-converging Newton solver. Here, further investigation is needed to prove this hypothesis.

6.3. Conclusion

In this section, we described the implementation of the operator-related damped Jacobian method in the JFNK solver and tested it on a pan-Arctic sea ice model. This model is used by the working group of Prof. Tremblay at the McGill University to investigate the ice conditions in short-term and seasonal forecasts. In this sea ice model, the main computational effort in solving the sea ice momentum equation lies in the approximation of the linear system in every Newton iteration. The linear systems are solved with a GMRES method preconditioned by a LSOR iteration. To keep the number of linear iterations as low as possible, we choose the linear tolerance as low (close to 1) as possible while maintaining stability of the solver. In doing so, we observed the most robust performance of the *operator-related damped* JFNK (orJFNK) solver with a fixed linear

tolerance at tol=0.67 (6.6).

In contrast to Lemieux et al. [56] who applied the JFNK solver with an *adaptive control of the linear tolerance* (JFNKac), we do not observe an enhanced robustness by solving the linear problem in the *operator-related damped JFNK* (orJFNK) solver, neither with an adaptive control, nor by using a linear tolerance close to 1. In order to reduce the overall computational cost, we suggest testing a multigrid preconditioner to the GMRES method in the JFNK solver as it is done in Section 5.4. Here, especially at high resolutions, the use of the multigrid preconditioner strongly decreases the iteration count of the linear solver in comparison to the use of an ILU preconditioner.

We analyzed the robustness the *orJFNK* solver, the *JFNKac* approach and the JFNK solver with a *fixed linear tolerance* (JFNKf) under mesh refinement. To analyze the performance, we simulated the ice dynamics over $T = 1$ month with $k = 2$ h and started the simulation on 01.01.2002. We showed that the robustness of the solver is decreasing with increasing spatial resolutions. In terms of robustness, we noted the biggest difference on the 10 km mesh. We observed failure rates of 0.5% of the *orJFNK* solver and of 24% of the *JFNKac* solver. We assume that the increase in robustness will be even higher on finer meshes as it was observed on an idealized test case in Section 5.3.

It needs to be investigated what effect the increased robustness has on the overall numerical error, e.g, how big the approximation error is due to failures of the solver in comparison to either the *splitting error* achieved by decoupling the system of equations or the *discretization error* in space and time. We will estimate the *splitting error* and the *discretization error* for the sea ice model in Chapter 8.

By applying a partitioned solution approach (SIT) as introduced by Lemieux et al. [53] a *splitting error* is introduced. This error can be avoided by solving the whole system simultaneously. As Ip et al. [34], we found in [68] that a monotholitic framework is not suitable for the sea ice equations due to the large computational costs. Lemieux et al. [53] introduced an IMEX scheme as an iterative solution method to solve the viscous-plastic sea ice model. This monotholitic approach is described in Algorithm 2 in Section 3.1. We found that the *JFNKac* solver and the *orJFNK* method are less stable when using the IMEX method than solving the system with a partitioned solution approach. The robustness of both solvers in the IMEX method is increased by decreasing the time step. In the partitioned approach as well as the monotholitic approach, the *orJFNK* solver is more robust than the *JFNKac* solver.

Finally, we observed that the error introduced by decoupling the system is bigger than the error that is introduced by a non-converging Newton solver. Further investigation is needed to prove this hypothesis. Note that the use of the IMEX method with many advected quantities such as the ice thickness, the ice concentration or the salinity, is very costly. In order to apply the IMEX method in global climate simulations, the efficiency of this method needs to be improved. Alternatively, the time step size can be decreased to reduce the *splitting error* in the numerical approximation.

Part IV.

A goal oriented error estimator for partitioned solution approaches

7. A posteriori error estimator

The aim of this chapter is to derive an abstract representation of a *goal oriented error estimator* for *partitioned solution approaches* of coupled systems. For some applications as the sea ice model [52] or the Navier-Stokes flow coupled to a transport process [10], it is desirable to apply a partitioned solution approach to decouple the system of equations and solve the different equations separately.

The concept of the error estimator is based on the *dual weighted residual method* which goes back to Becker and Rannacher [6]. The error estimator will be derived for a general class of coupled non-stationary (parabolic) partial differential equations that are solved with a partitioned solution approach. Particularly in Chapter 8, we evaluate the error estimator for a coupled system consisting of Burgers equation and a transport process, before we apply it to the more complex sea ice model. Due to the decoupling of the system, the error estimator consists of a *splitting error* in addition to the *spatial* and *temporal discretization error*. We introduce a projection operator to decouple the system in time. The decoupled system is then discretized with a discontinuous Galerkin method. Particularly, we present the discontinuous Galerkin method of degree zero. More details on discontinuous Galerkin methods in time can be found in the books of Eriksson et al. [18] and Thomée [94].

The chapter is structured as follows. First in Section 7.1, we introduce an abstract problem formulation and present the Galerkin discretization in space and time. Section 7.2 is devoted to the derivation of the a posteriori error estimator. In Section 7.2.1, we introduce the adjoint equation and recall the idea of the dual weighted residual method. We derive the goal oriented error estimator for partitioned solution approaches in Section 7.2.2. In order to compute the a posteriori error estimator, we present an approximation of the error estimator in Section 7.3. In Section 7.4, we describe, based on the error estimator, an adaptive mesh refinement algorithm, that aims to reduce the computational costs of a coupled simulation. The numerical evaluation of the error estimator is given in Chapter 8. This chapter closes with a conclusion.

7.1. Notation and problem formulation

We recall the notation of Section 3. Let $\Omega \in \mathbb{R}^d$ be a Lipschitz domain. For a time interval $I = [0, T]$, we introduce

$$X(I) := \Big\{ v \in L^2(I; \mathcal{V}), \partial_t v \in L^2(I, \mathcal{V}^*) \Big\},$$

with a Hilbert space $\mathcal{V}$ and its dual space $\mathcal{V}^*$. Usually $\mathcal{V} = H_0^1(\Omega)$, a Sobolev space of $L^2(\Omega)$-functions with weak first derivative in $L^2(\Omega)$ and zero trace on the boundary

$\partial\Omega$. In case of $H_0^1(\Omega)$, the dual space is given by $H^{-1}(\Omega)$. We assume that our coupled system can be written in the variational space-time framework as

$$U \in X(I): \int_I \{\Big(g(U(t))\partial_t U(t), \Phi(t)\Big) + a\Big(U(t)\Big)\Big(\Phi(t)\Big)\}dt = \int_I \Big(f(t), \Phi(t)\Big), \quad \forall \Phi \in X(I), \tag{7.1}$$

with $f \in L^2(I, \mathcal{V}^*)$ and $g : \mathcal{V} \to \mathbb{R}$. The semilinear form which may be nonlinear in the first argument and which is linear in the second argument is denoted by $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$. For a more detailed analysis, we reformulate equation (7.1) to find $(u, v) \in X(I)$ such that

$$\begin{aligned} \int_I \{\Big(g_u(u(t), v(t))\partial_t u(t), \phi(t)\Big) + a_u\Big(u(t), v(t)\Big)\Big(\phi(t)\Big)\}dt &= \int_I \Big(f_u(t), \phi(t)\Big)dt, \\ \int_I \{\Big(g_v(u(t), v(t))\partial_t v(t), \psi(t)\Big) + a_v\Big(u(t), v(t)\Big)\Big(\psi(t)\Big)\}dt &= \int_I \Big(f_v(t), \psi(t)\Big)dt, \\ \forall (\phi, \psi) \in X(I), \end{aligned} \tag{7.2}$$

with $U = (u, v)$ and $g(U(t)) = g_u(u(t), v(t)) + g_v(u(t), v(t))$. The sesquilinearform and the right hand side can be written as $a(U(t), \boldsymbol{\phi}(t)) = a_u(u, v)(\phi, \psi) + a_v(u, v)(\phi, \psi)$, with $\boldsymbol{\phi} = (\phi, \psi)$ and $f(t) = (f_u(t), f_v(t))$.

Galerkin discretization in time We partition the time interval $I = [0, T]$

$$I = \{0\} \cup I_1 \cup I_2 \cup ... \cup I_N, \tag{7.3}$$

into the sub-intervals $I_n = (t_{n-1}, t_n]$ and the corresponding time steps $k_n = t_n - t_{n-1}$. To apply a discontinuous Galerkin method in time, we define the non-conforming space

$$X_I := \Big\{v : I \times \Omega \to \mathbb{R}, v|_{I_n} \in X(I_n)\Big\},$$

which is not necessarily continuous at t_n. Also, note that $X_I \not\subset X(I)$, but $X(I) \subset X_I$. For $v \in X_I$ we introduce

$$v_n^+ := v(t_n)^+ := \lim_{s \to 0, s \geq 0} v(t_n + s), \quad v_n^- := v(t_n)^- := \lim_{s \to 0, s \geq 0} v(t_n - s), \quad [v]_n := v_n^+ - v_n^-. \tag{7.4}$$

On the space X_I, we formulate a second variational formulation

$$U \in X_I, \quad B(U)(\Phi) = F(\Phi), \quad \forall \Phi \in X_I, \tag{7.5}$$

with

$$\begin{aligned} B(U)(\Phi) := \sum_{n=1}^N \int_{I_n} \{g(U(t))\Big(\partial_t U(t), \Phi(t)\Big) + a\Big(U(t)\Big)\Big(\Phi(t)\Big)\}dt \\ + \sum_{n=1}^N \Big(g(U_{n-1}^+))[U]_{n-1}, \phi_{n-1}^+\Big), \end{aligned} \tag{7.6}$$

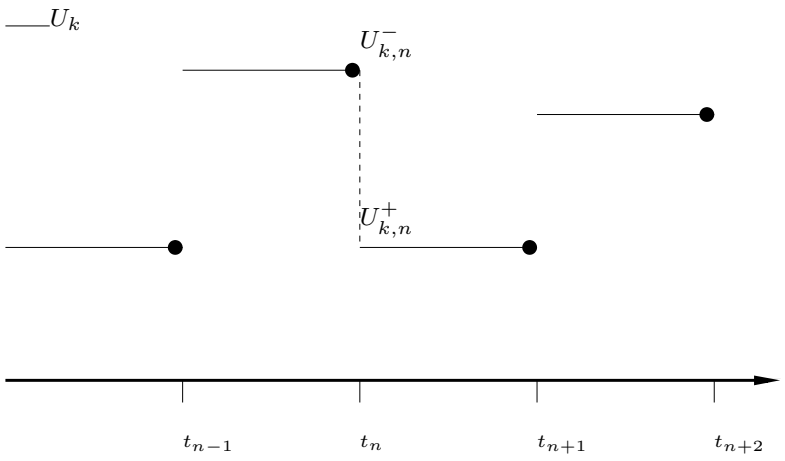


Figure 7.1.: Visualization of the temporal jump of piecewise constant functions at time point $t = t_n$.

and

$$F(\Phi) := \sum_{n=1}^{N} (f(t), \Phi(t))dt. \tag{7.7}$$

The continuous solution $U \in X(I)$ of (7.1) also solves formulation (7.5) as all jumps in (7.6) are zero.

For time discretization we apply a *discontinuous Galerkin discretization of degree zero* ($dG(0)dG(0)$) with piecewise constant ansatz and test functions. The space of the piecewise constant functions on I_n is given as

$$X_k \subset X_I, \qquad X_k := \Big\{ v \in X_I, \quad v_{|_{I_n}} \in P^{(0)}(I_n, \mathcal{V}), \quad n = 1, ..., N \quad v(0) \in \mathcal{V} \Big\}.$$

The space of polynomials with degree zero in time is denoted by $P^{(0)}(I, \mathcal{V})$. For a better understanding, we visualize the piecewise constant functions in Figure 7.1. The semi-discrete formulation of problem (7.5) reads as

$$U_k \in X_k : \quad B(U_k)(\Phi_k) = F(\Phi_k), \quad \forall \Phi_k \in X_k. \tag{7.8}$$

Galerkin discretization in space Let $V_h \subset \mathcal{V}$ be a conforming finite element subspace as introduced in Section 3. The discrete space is defined as

$$X_{k,h} := \Big\{ v \in X_I, \quad v_{|_{I_n}} \in P^0(I_n, V_h), \quad n = 1, ..., N \quad v(0) \in V_h \Big\}. \tag{7.9}$$

Finally, the discrete formulation of problem (7.5) is given by

$$U_{k,h} \in X_{k,h} : \quad B(U_{k,h})(\Phi_{k,h}) = F(\Phi_{k,h}), \quad \forall \Phi_{k,h} \in X_{k,h}. \tag{7.10}$$

Considering conforming test and ansatz spaces, *Galerkin orthogonality* holds for the solutions $U \in X_I$ of (7.5) and $U_{k,h} \in X_{k,h}$ of (7.10),

$$B(U)(\Phi_{k,h}) - B(U_{k,h})(\Phi_{k,h}) = 0, \quad \forall \Phi_{k,h} \in X_{k,h}. \tag{7.11}$$

7.2. Goal oriented error estimator for partitioned solution approaches

By the application of a *partitioned solution approach*, the coupled system (7.8) is decoupled in time such that the different equations can be solved separately. The numerical solution of this decoupled system $U^s_{k,h}$ does not solve the coupled system (7.10). Further, using a partitioned solution approach, the discrete solution $U_{k,h}$ of the coupled system (7.10) is not at hand such that, besides the discretization error in space and time, an additional non-conformity error due to the *splitting* in time is introduced.

To address this, we define a second sesquilinearform $B^s(\cdot)(\cdot)$ that describes the decoupling of system (7.8) in time. Based on the $dG(0)dG(0)$ discretization, the splitting is realized by the projection

$$\mathcal{S} : X_I \to P^{(0)}(I_n, \mathcal{V}), \quad \mathcal{S}(v)|_{I_n} = v^-(t_{n-1}). \tag{7.12}$$

We formulate the decoupled variational problem based on the notation introduced in (7.2) as

$$U^s \in X_I : B^s(U^s)(\Phi) = F(\Phi), \quad \forall \Phi \in X_I, \tag{7.13}$$

with

$$\begin{aligned} B^s(U^s)(\Phi) := &\sum_{n=1}^{N} \int_{I_n} \{ g_u\Big(u^s(t), \mathcal{S}(v^s(t))\Big)\Big(\partial_t u^s(t), \phi(t)\Big) + a_u\Big(u^s(t), \mathcal{S}(v^s(t))\Big)\Big(\phi(t)\Big)\} dt \\ &+ \sum_{n=1}^{N} g_u\Big(u^{s+}_{n-1}, \mathcal{S}(v^s)^+_{n-1}\Big)\Big([u^s]_{n-1}, \phi^+_{n-1}\Big) \\ &\sum_{n=1}^{N} \int_{I_n} \{ g_v\Big(u^s(t), (v^s(t))\Big)\Big(\partial_t v^s(t), \psi(t)\Big) + a_v\Big(u^s(t), v^s(t))\Big)\Big(\psi(t)\Big)\} dt \\ &+ \sum_{n=1}^{N} g_v\Big(u^{s+}_{n-1}, v^{s+}_{n-1})\Big)\Big([v^s]_{n-1}, \phi^+_{n-1}\Big). \end{aligned}$$

The *splitting error* introduced from the decoupling in time is given by

$$\beta(U^s, \Phi) := B^s(U^s)(\Phi) - B(U^s)(\Phi). \tag{7.14}$$

In this section, we derive a *goal oriented error estimator* for *partitioned solution approaches*. The new error estimator will be based on concepts of the *dual weighted residual method* introduced by Rannacher and Becker [6]. Before deriving the error estimator for decoupled systems, we briefly recapitulate the idea of the dual weighted residual method.

7.2.1. The dual weighted residual method

The derivation follows Becker and Rannacher [6]. Let U and $U_{k,h}$ be the solution of (7.5) and of (7.10). The idea of the dual weighted residual method for time-dependent problems in an interval $I = [0, T]$ is to estimate the error $e_u = U - U_{k,h}$ in a goal functional $J : X_I \to \mathbb{R}$, i.e., $J(U) - J(U_{k,h})$. We consider functionals like

$$J(U) = \int_I j(U(t))\, dt,$$

where $j : \mathcal{V} \to \mathbb{R}$. The estimator is derived by introducing a Lagrange functional $L : X \times X \to \mathbb{R}$ and by formally minimizing the goal functional $J(U)$. The constraint of the minimization problem is the variational problem (7.5). Following Becker and Rannacher [6], we define the Lagrange functional as

$$L(U, Z) := J(U) + F(Z) - B(U)(Z). \tag{7.15}$$

The stationary point of the Lagrange functional $L(U, Z)$ (7.15),

$$L'(U, Z)(\delta_u, \delta_z) = \Big(F(\delta_z) - B(U, \delta_z)\Big) + \Big(J'(U)(\delta_u) - B'(U)(Z, \delta_u)\Big) = 0, \quad \forall \delta_u, \delta_z \in X,$$

defines the solution $U \in X$ of equation (7.5) as well as the adjoint solution $Z \in X$ of the adjoint equation

$$J'(U)(\delta_u) = B'(U)(Z, \delta_u), \quad \forall \delta_u \in X. \tag{7.16}$$

Adjoint problem Let U_k be the semi-discrete solution of (7.8). Analogous to the primal equation, we introduce the semi-discrete adjoint equation as

$$Z_k \in X_k : J'(U_k)(\delta_{u_k}) = B'(U_k)(Z_k, \delta_{u_k}), \quad \forall \delta_{u_k} \in X_k, \tag{7.17}$$

and the fully discrete problem as

$$Z_{k,h} \in X_{k,h} : J'(U_{k,h})(\delta_{u_{k,h}}) = B'(U_{k,h})(Z_{k,h}, \delta_{u_{k,h}}), \quad \forall \delta_{u_{k,h}} \in X_{k,h}. \tag{7.18}$$

Galerkin orthogonality also holds for the adjoint problem

$$B'(U)(Z - Z_{k,h}, \delta_{u_{k,h}}) = 0, \quad \forall \delta_{u_{h,k}} \in X_{k,h}. \tag{7.19}$$

The dual weighted residual method For a better readability, we introduce $x = (U, Z)$, $x_{k,h} = (U_{k,h}, Z_{k,h})$ and $e = x - x_{k,h}$. Using the Lagrange functional defined in (7.15), it holds that

$$J(U) - J(U_{k,h}) = L(x) - L(x_{k,h}) = \int_0^1 L'(x_{k,h} + \lambda e)(e) d\lambda.$$

By approximating the integral with the trapezoidal rule, we obtain

$$J(U) - J(U_{k,h}) = \frac{1}{2}\Big\{L'(x)(e) + L'(x_{k,h})(e)\Big\} + \mathcal{R}(x_{k,h}, e), \tag{7.20}$$

with the remainder

$$\mathcal{R}(x_{k,h}, e) = \int_0^1 L'''(x_{k,h} + \lambda e)(e, e, e) \cdot \lambda(1 - \lambda) d\lambda. \tag{7.21}$$

Let x and $x_{k,h}$ be the stationary points. By Applying Galerkin orthogonality given in (7.11) and in (7.18) to equation (7.20), it holds that

$$\begin{aligned} J(U) - J(U_{k,h}) = L(x) - L(x_{k,h}) =& \frac{1}{2}\Big\{ \underbrace{L'(x)(x - x_{k,h})}_{=0} + L'(x_{k,h})(x - x_{k,h}) \Big\} + \mathcal{R}(x_{k,h}, e) \\ =& \frac{1}{2} L'(x_{k,h})(x - \mathcal{I}_{k,h} x) + \mathcal{R}(x_{k,h}, e), \end{aligned} \tag{7.22}$$

with an interpolation $\mathcal{I}_{k,h} : X \times X \to X_{k,h} \times X_{k,h}$. The exact solution x and the interpolation $\mathcal{I}_{k,h} x$ is not at hand. Thus, for an evaluation of the dual weighted residual method, it remains to approximate the interpolation error $x - \mathcal{I}_{k,h} x$. Here, various possibilities are discussed in literature, i.e., by Becker and Rannacher [6], by Schmich and Vexler [88] and by Richter and Meidner [71]. We describe the approximation of the weight $x - \mathcal{I}_{k,h} x$ in Section 7.3.

7.2.2. Derivation of the goal oriented error estimator for partitioned solution approaches

In this section, we derive a *goal oriented error estimator* for *partitioned solution approaches* of coupled systems. As the numerical solution $U^s_{k,h}$ of the decoupled system

$$U^s_{k,h} \in X_{k,h} : B^s(U_{k,h})(\Phi_{k,h}) = F(\Phi_{k,h}), \quad \forall \Phi_{k,h} \in X_{k,h}, \tag{7.23}$$

does not solve the coupled system (7.10), a *splitting error* is introduced in addition to the *temporal* and *spatial discretization error*. The a posteriori error estimator is given as follows.

Theorem 2. *Let $U \in X$ be the primal solution to (7.5) and Z be the dual solution to (7.16). Further, $U^s_k \in X_k$ is the time discrete solution achieved with a partitioned solution approach*

$$U^s_k \in X_k : \quad F(\Phi_k) = B^s(U^s_k)(\Phi_k), \quad \forall \Phi_k \in X_k, \tag{7.24}$$

and Z^s_k the corresponding adjoint solution

$$Z^s_k \in X_k : J'(U^s_k)(\Psi_k) = B^{s'}(U^s_k)(Z^s_k, \Psi_k), \quad \forall \Psi_k \in X_k. \tag{7.25}$$

Let $U^s_{k,h}$ and $Z^s_{k,h}$ be the fully discrete solutions to equations (7.24) and (7.25). Then, it holds that

$$\begin{aligned} J(U) - J(U^s_{k,h}) =& \frac{1}{2}\Big\{ \rho(U^s_k)(Z - \mathcal{I}_k Z) + \rho^*(U^s_k, Z^s_k)(U - \mathcal{I}_k U) \Big\} + \mathcal{R}_k \\ &+ \frac{1}{2}\Big\{ \rho(U^s_{k,h})(Z^s_k - \mathcal{I}_h Z^s_k) + \rho^*(U^s_{k,h}, Z^s_{k,h})(U^s_k - \mathcal{I}_h U^s_k) \Big\} + \mathcal{R}_h \\ &+ \frac{1}{2}\Big\{ \beta(U^s_k)(Z + Z^s_k) + \beta^*(U^s_k)(Z^s_k, U - U^s_k) \Big\}, \end{aligned}$$

with the primal and dual residuals

$$\rho(U)(\Phi) := F(\Phi) - B^s(U)(\Phi), \quad \rho^*(U)(Z) := J'(U)(Z,\Psi) - B^{s'}(U)(Z,\Psi), \tag{7.26}$$

as well as the primal and dual splitting errors

$$\beta(U)(\Phi) := B^s(U)(\Phi) - B(U)(\Phi), \quad \beta^*(U)(Z) := B^{s'}(U)(Z,\Psi) - B'(U)(Z,\Psi). \tag{7.27}$$

The remainders $\mathcal{R}_k := \mathcal{R}_k(x_k, x - x_k)$ and $\mathcal{R}_h := \mathcal{R}_h(x_{k,h}, x_k - x_{k,h})$ have the same structure as in (7.21). By $\mathcal{I}_k$ and $\mathcal{I}_h$ we denote pointwise nodal interpolations in time and space.

Proof. Similar to the Lagrange functional (7.15) defined in Section 7.2, we introduce a Lagrange functional for coupled systems that are solved with a partitioned solution approach

$$L^s : X_k \times X_k \to \mathbb{R} : \quad L^s(U_k^s, Z_k^s) := J(U_k^s) + F(Z_k^s) - B^s(U_k^s)(Z_k^s). \tag{7.28}$$

We begin with splitting the error into the temporal and spatial contribution. As U and Z are the solutions to (7.5) and (7.16), and $U_{k,h}^s$ and $Z_{k,h}^s$ are the discrete solutions to (7.24) and (7.25), we obtain

$$\begin{aligned} J(U) - J(U_{k,h}^s) =& J(U) \pm J(U_k^s) - J(U_{k,h}^s) = L(U,Z) \pm L^s(U_k^s, Z_k^s) - L^s(U_{k,h}^s, Z_{k,h}^s) \\ =& \underbrace{L(U,Z) - L(U_k^s, Z_k^s)}_{(I)} + \underbrace{L(U_k^s, Z_k^s) - L^s(U_k^s, Z_k^s)}_{(II)} \\ &+ \underbrace{L^s(U_k^s, Z_k^s) - L^s(U_{k,h}^s, Z_{k,h}^s)}_{(III)}. \end{aligned}$$

The *splitting error* is estimated by the second term (II). Given that U_k^s is the solution to (7.24) and applying the notation of (7.14), the second term (II) reduces to

$$(II) = \underbrace{J(U_k^s) - J(U_k^s)}_{=0} + F(Z_k^s) - B(U_k^s)(Z_k^s) + \underbrace{F(Z_k^s) - B^s(U_k^s)(Z_k^s)}_{=0} = \beta(U_k^s)(Z_k^s).$$

Using the trapezoidal rule, the first term (I) is given as

$$L(U,Z) - L(U_k^s, Z_k^s) = \frac{1}{2}\Big\{ \underbrace{L'(U_k^s, Z_k^s)(U - U_k^s, Z - Z_k^s)}_{(*)} \Big\} + \mathcal{R}_k(x_k, x - x_k),$$

where the remainder is of third order (7.21). We proceed with expanding $(*)$ to

$$\begin{aligned} L'(U_k^s, Z_k^s)(U - U_k^s, Z - Z_k^s) =& [L' - L^{s'}](U_k^s, Z_k^s)(U - U_k^s, Z - Z_k^s) \\ &+ L^{s'}(U_k^s, Z_k^s)(U - \mathcal{I}_k U, Z - \mathcal{I}_k Z) \\ =& \beta(U_k^s)(Z - Z_k^s) + \beta^*(U_k^s)(Z_k^s, U - U_k^s) \\ &+ \rho(U_k^s)(Z - \mathcal{I}_k Z) + \rho^*(U_k^s)(Z_k^s, U - \mathcal{I}_k U). \end{aligned} \tag{7.29}$$

The third term (III) is treated with the usual concept of the dual weighted residual method as described in Section 7.2.1. Thus, we obtain

$$\begin{aligned} L^s(U_k^s, Z_k^s) - L^s(U_{k,h}^s, Z_{k,h}^s) =& \frac{1}{2}\Big\{ L^{s'}(U_{k,h}^s, Z_{k,h}^s)(U_k^s - \mathcal{I}_h U_k^s, Z_k^s - \mathcal{I}_h Z_k^s)\Big\} \\ &+ \mathcal{R}_h(x_{k,h}, x - x_{k,h}) \\ =& \frac{1}{2}\Big\{ \rho(U_{k,h}^s)(Z_k^s - \mathcal{I}_h Z_k^s) + \rho^*(U_{k,h}^s, Z_{k,h}^s)(U_k^s - \mathcal{I}_h U_k^s)\Big\} \\ &+ \mathcal{R}_h(x_{k,h}, x - x_{k,h}). \end{aligned}$$

□

7.3. Approximation of the error estimator

By neglecting the remainders, the error estimator derived in Theorem 2 can be approximated as

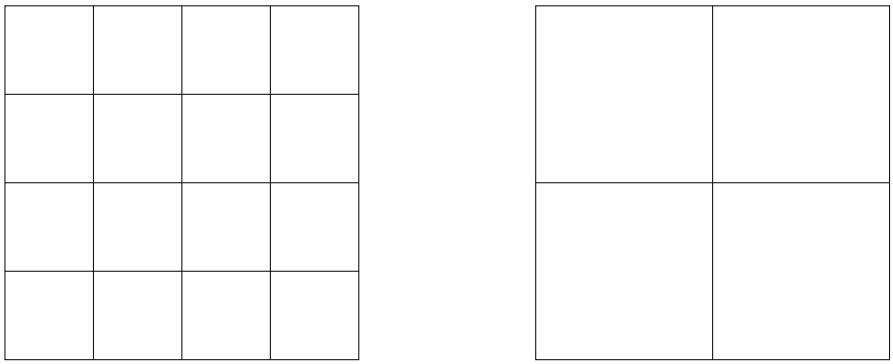
$$\begin{aligned} J(U) - J(U_{k,h}^s) \approx& \frac{1}{2}\Big\{ \rho(U_k^s)(Z - \mathcal{I}_k Z) + \rho^*(U_k^s)(Z_k^s, U - \mathcal{I}_k U)\Big\} \\ &+ \frac{1}{2}\Big\{ \rho(U_{k,h}^s)(Z_k^s - \mathcal{I}_h Z_k^s) + \rho^*(U_{k,h}^s)(Z_{k,h}^s, U_k^s - \mathcal{I}_h U_k^s)\Big\} \\ &+ \frac{1}{2}\Big\{ \beta(U_k^s)(Z + Z_k^s) + \beta^*(U_k^s)(Z_k^s, U - U_k^s)\Big\}. \end{aligned} \tag{7.30}$$

In the following, we again use the short notation $x = (U, Z)$. As we only compute the discrete solution of the decoupled system, $U_{k,h}^s$ and $Z_{k.h}^s$, the remaining quantities in (7.30) need to be reconstructed. In particular, in order to achieve a computable error estimator based on formulation (7.30), it remains to describe the evaluation of the temporal interpolation error $x - \mathcal{I}_k x$ and spatial interpolation error $x_k^s - \mathcal{I}_h x_k^s$ as well as the *weights* of the primal and dual *splitting error*, $Z + Z_k^s$ and $U - U_k^s$. We refer by *weights* to the second argument of the sesquilinearform.

We start with approximating the *primal* and *dual residual error* in time and space. Then, we continue with the approximation of the *splitting error*.

Approximation of residual error We begin with prescribing the approximation of the weights of the *temporal* and *spatial residuals*, $x - \mathcal{I}_k x$ and $x_k^s - \mathcal{I}_h x_k^s$. As we apply a partitioned solution approach, our approximation is based on the discrete solution of the decoupled system $x_{k,h}^s$. Thus, we express $x - \mathcal{I}_k x$ by the interpolation errors $x^s - \mathcal{I}_k x^s$ of the partitioned solution approach by inserting $\pm x^s - \mathcal{I}_k x^s$.

$$\begin{aligned} x - \mathcal{I}_k x &= x^s - \mathcal{I}_k x^s + x - x^s - \mathcal{I}_k\Big(x - x^s\Big) \\ &= x^s - \mathcal{I}_k x^s + \underbrace{(id - \mathcal{I}_k)(x - x^s}_{=O(k^2)} \\ &\approx x^s - \mathcal{I}_k x^s, \end{aligned}$$



Ω_h Ω_{2h}

Figure 7.2.: Two dimensional patch-wise organized mesh Ω_h and the corresponding coarse mesh Ω_{2h}.

where we assume that $\|x - x^s\| = O(k)$.

To approximate the interpolation error $x^s - \mathcal{I}_k x^s$, we introduce a higher order reconstruction in space, Π_h, and a high-order reconstruction in time, Π_k.

$$\begin{aligned} U^s - \mathcal{I}_k U^s \approx & \Pi_k U_k^s, \qquad & Z^s - \mathcal{I}_k Z^s \approx \Pi_k Z_k^s, \\ U_k^s - \mathcal{I}_h U_k^s \approx & \Pi_h U_{k,h}^s, \qquad & Z_k^s - \mathcal{I}_h Z_k^s \approx \Pi_h Z_{k,h}^s. \end{aligned} \tag{7.31}$$

We proceed with defining the high-order reconstructions Π_k and Π_h. For a better readability, we recall the definition of the finite element space. We assume that Ω_h is structured in a patch-wise manner such that we obtain Ω_h as a refinement from Ω_{2h}. It means, we can always combine four cells of Ω_h and receive one cell of Ω_{2h}. In Figure 7.2, we show the patch-wise structure. In Section 3.2, we introduced $V_h(\Omega_h)$ as the bi-linear continuous finite element space with the nodal basis $\{\phi_1, ..., \phi_n\}$. In order to define the high-order reconstruction of $U_{k,h}^s$ and $Z_{k,h}^s$ in space we make use of the bi-quadratic finite element space $V_{2h}^{(2)}$ with the basis $\{\phi_1^{(2)}, ..., \phi_n^{(2)}\}$. By $i_{2h}^{(2)} : V_h \to V_{2h}^{(2)}$, we define the nodal interpolation from the bi-linear finite elements to bi-quadratic finite elements on Ω_{2h}. The interpolation is visualized in Figure 7.3.

Based on this interpolation, the spatial interpolation error is approximated as

$$\Pi_h U_{k,h}^s := i_{2h}^{(2)} U_{k,h}^s - U_{k,h}^s, \qquad \Pi_h Z_{k,h}^s := i_{2h}^{(2)} Z_{k,h}^s - Z_{k,h}^s.$$

Next, we describe the approximation of the temporal interpolation error. Based on the $dG(0)dG(0)$ discretization in time, we define

$$\Pi_k U_k^s := i_k^{(1)} U_k^s - U_k^s, \qquad \Pi_k Z_k^s := i_k^{(1)} Z_k^s - Z_k^s, \tag{7.32}$$

where $i_k^{(1)} U_k^s$ and $i_k^{(1)} Z_k^s$, are piecewise linear reconstructions of U_k^s and Z_k^s in time. As in Besier and Rannacher [87], we introduce on each interval $I_n = [t_{n-1}, t_n)$ the

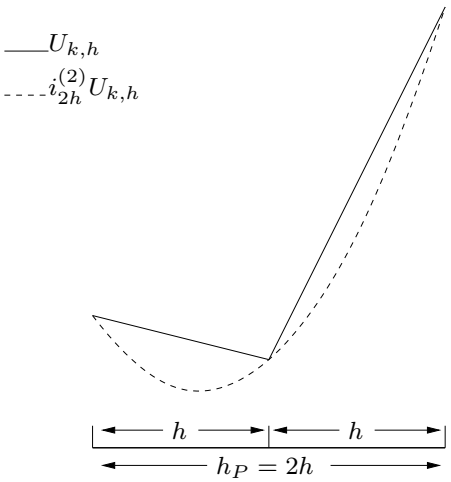


Figure 7.3.: Spatial interpolation of linear finite elements into the space of quadratic finite elements on every patch of the mesh.

reconstruction as

$$i_k^{(1)} U_k^s | I_n := \frac{t - t_{n-1}}{t_n - t_{n-1}} U_k^{s-}(t_n) + \frac{t - t_n}{t_{n-1} - t_n} U_k^{s-}(t_{n-1}), \tag{7.33}$$

and respectively

$$i_k^{(1)} Z_k^s | I_n := \frac{t - t_{n-1}}{t_n - t_{n-1}} Z_k^{s-}(t_n) + \frac{t - t_n}{t_{n-1} - t_n} Z_k^{s-}(t_{n-1}). \tag{7.34}$$

This piecewise linear interpolation is visualizes in Figure 7.4.

It is left to approximate the semi-discrete solutions U_k^s and Z_k^s in time. For the reconstruction, we make use of the high-order interpolation in space and approximate

$$U_k^s \approx i_{2h}^{(2)} U_{k,h}^s, \quad Z_k^s \approx i_{2h}^{(2)} Z_{k,h}^s. \tag{7.35}$$

In Section 8.1.2, we compare alternative methods for an approximation of the temporal interpolation error (7.32).

Finally, it remains to outline the evaluation of the temporal integrals in the error estimator. Among others a good introduction into quadrature rules is provided by the textbook of Richter and Wick [83]. All time-dependent integrals without linear reconstruction $i_k^{(1)}$ are approximated with the first order box rule. This ensures an exact evaluation of the integrals as we use a $dG(0)dG(0)$ discretization in time. Further, all temporal integrals that involve a high-order reconstruction $i_k^{(1)}$ are evaluated with the second order trapezoidal rule. This ensures an exact evaluation of these temporal integrals as $i_k^{(1)}$ is a first order interpolation time.

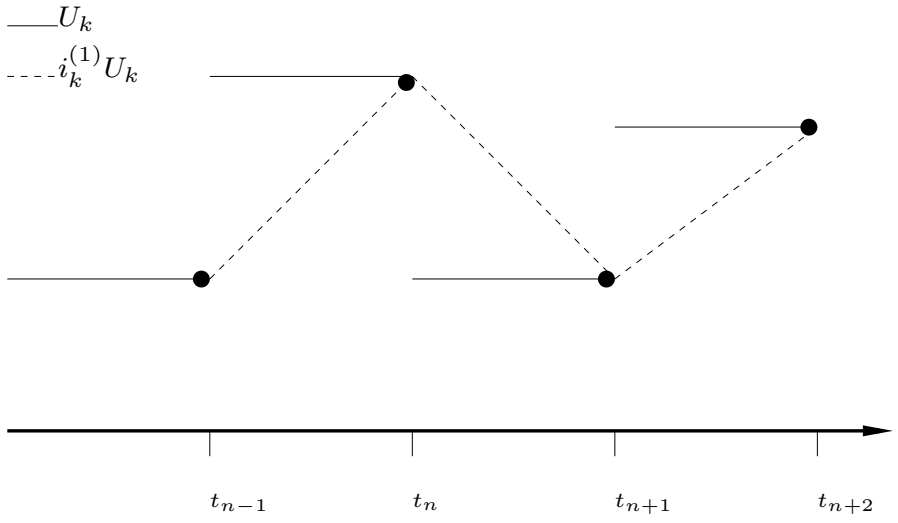


Figure 7.4.: Piecewise linear interpolation of piecewise constant functions in time.

Approximation of the splitting error In order to evaluate the *splitting error*, we introduce the following abbreviation

$$\tilde{\Pi}_k Z^s_{k,2h} := i_k^{(1)} Z^s_{k,2h} + Z^s_{k,2h}.$$

Based on the interpolations (7.32), (7.33) and the approximation in (7.35), we evaluate the *splitting error* as

$$\begin{aligned}\beta(U^s_k)(Z + Z^s_k) + \beta^*(U^s_k)(Z^s_k, U - U^s_k) \approx &\beta(i^{(2)}_{2h} U^s_{k,h})(\tilde{\Pi}_k i^{(2)}_{2h} Z^s_{k,h}) \\ &+ \beta^*(i^{(2)}_{2h} U^s_{k,h})(i^{(2)}_{2h} Z^s_{k,h}, \Pi_k i^{(2)}_{2h} U^s_{k,h}),\end{aligned}$$

where $i^{(2)}_{2h}$ describes the nodal interpolation from the bi-linear finite elements to bi-quadratic finite elements. To summarize, we compute the error estimator as

$$J(U) - J(U^s_{k,h}) \approx \eta_{\rho_k} + \eta_{\rho_h} + \eta_\beta,$$

with

$$\begin{aligned}\eta_{\rho_k} :=& \frac{1}{2}\Big\{\rho(i^{(2)}_{2h} U^s_{k,h})(\Pi_k i^{(2)}_{2h} Z^s_{k,h}) + \rho^*(i^{(2)}_{2h} U^s_{k,h})(i^{(2)}_{2h} Z^s_{k,h}, \Pi_k i^{(2)}_{2h} U^s_{k,h})\Big\}, \\ \eta_{\rho_h} :=& \frac{1}{2}\Big\{\rho(U^s_{k,h})(\Pi_h Z^s_{k,h}) + \rho^*(U^s_{k,h})(Z^s_{k,h}, \Pi_h U^s_{k,h})\Big\}, \\ \eta_\beta :=& \frac{1}{2}\Big\{\beta(i^{(2)}_{2h} U^s_{k,h})(\tilde{\Pi}_k i^{(2)}_{2h} Z^s_{k,h}) + \beta^*(i^{(2)}_{2h} U^s_{k,h})(i^{(2)}_{2h} Z^s_{k,h}, \Pi_k i^{(2)}_{2h} U^s_{k,h})\Big\}.\end{aligned} \tag{7.36}$$

7.4. Adaptive refinement strategy

In this section, we describe an adaptive refinement algorithm which is based on the approximation of the a posteriori error estimator given in (7.36). The aim of the algorithm

is to efficiently increase the accuracy of the numerical solution by reducing the time step size and by automatically refining the mesh where large errors occur.

In each iteration of the refinement algorithm, we compute the error estimator which consists of the *splitting error* η_β as well as the *temporal* and *spatial residual error*, η_{ρ_k} and η_{ρ_h}. If temporal errors dominate the spatial error, we reduce the time step until the error is balanced. In case of a dominating spatial error, we calculate which cells $K_i \in \Omega_h$ need to be refined. Doing so, we localize the information of the error estimator η by splitting it into cell wise quantities η_i. A detailed description of the localization is given by Braack et al. [9]. A study on different error localizations is presented by Richter and Wick in [82]. Our adaptive mesh algorithm refines those cells K_i which provide an error contribution that is larger than the average mean value of η.

$$\bar{\eta} = \frac{1}{M}|\eta|_1. \tag{7.37}$$

Here, we denote the number of cells by M and the l_1-norm by $|\cdot|_1$. Further refinement strategies are presented by Becker and Braack in [4].

Algorithm 6. *Let Ω_0 be the initial mesh and $k_0 > 0$ the initial time step size. For $l = 0, .., n_r$ iterate*

1. *Compute the solution $U^s_{k_l,h_l}$ of the primal system (7.24).*
2. *Compute the solution $Z^s_{k_l,h_l}$ of the dual system (7.25).*
3. *Compute the goal function $J(U^s_{k_l,h_l})$.*
4. *Evaluate the error estimator η_l according to (7.36).*
5. *Localize the error estimator $\eta_{l,i}$ with $i = 1, .., M_l$ as described in Braack et al. [9].*
6. *Refine all cells K_i with $\eta_{l,i} \geq \bar{\eta}_l$ and proceed with the refined mesh Ω_{l+1}.*
7. *If $\eta_{\rho_k} + \eta_\beta \geq \eta_{\rho_h}$, set $k_{l+1} = \frac{1}{2}k_l$.*

7.5. Conclusion

In this chapter, we introduced an abstract representation of a *goal oriented error estimator* that is applicable to the sea ice model. The error estimator is derived for a general class of coupled non-stationary partial differential equations that are solved with a *partitioned solution approach*, as it is done for the sea ice model.

The new error estimator is based on the concept of the *dual weighted residual method* that has been introduced by Becker and Rannacher [6]. The error estimator consists mainly of two parts, the *primal* and *dual residual error* that arise in the framework of the dual weighted residual method, and an additional *splitting error* which stems from the application of the partitioned solution approach. In the following chapter, we present numerical studies on the performance of the error estimator. Here, we also discuss the

application to the sea ice model. Before applying the concept to the sea ice model, we will test the error estimator on a simplified problem, namely Burgers equation coupled to a transport process.

We decoupled the system by using an operator projection and discretized the system with a $dG(0)dG(0)$ method in time. Here, an extension to higher order schemes is possible. The extension will only effect the approximation of the error estimator described in Section 7.3, where the primal and dual high-order reconstructions need to be adjusted.

As the derived error estimator consists of an interpolation error as well as the continuous and semi-discrete solutions of primal and dual systems, we introduced approximations to make the error estimator computable. The approximation is based on spatial and temporal interpolations of the discrete primal and dual solution of the decoupled system, $U^s_{k,h}$ and $Z^s_{k,h}$.

Finally, we presented an adaptive mesh refinement algorithm which is based on the new a posteriori error estimator (7.36). The aim of the refinement strategy is to decrease the numerical cost of coupled simulations. By balancing the temporal and spatial error contributions, we efficiently increased the accuracy of the numerical solution. The temporal error is reduced by uniform refinement of the time steps, whereas the spatial error is decreased by an adaptive mesh refinement algorithm. An extension to temporal adaptivity is possible.

8. Numerical evaluation of the error estimator on two test cases

In this chapter, we evaluate the *goal oriented error estimator* for *partitioned solution approaches* by considering two test cases, the complex sea ice model and a simplified example, namely Burgers equation coupled to a transport process. Both models involve a nonlinear momentum equation. For these two examples, we analyze the different contributions of the error estimator, the *temporal* and *spatial residual error*, η_{ρ_k} and η_{ρ_h}, as well as the *splitting error* η_β. Furthermore, for the sea ice model, we evaluate the adaptive mesh refinement strategy presented in Algorithm 6 in Section 7.4.

This chapter is structured as follows. Section 8.1 is devoted to the analysis of Burgers equation coupled to a transport process. In Section 8.1.1, we introduce a partitioned solution approach to solve this coupled system. The decoupled system is discretized with a $dG(0)dG(0)$ scheme in time. In Section 8.1.2, we give details on the numerical evaluation of the error estimator. Section 8.2 is devoted to the sea ice model. In particular, in Section 8.2.1, we introduce a partitioned solution approach for the decoupled sea ice model and discretize it with a $dG(0)dG(0)$ scheme in time. In Section 8.2.2, we present the numerical evaluation of the error estimator and the adaptive mesh refinement strategy. In Section 8.2.3, we discuss the practical relevance of adaptive meshes in large-scale sea ice simulations. Finally, we conclude in Section 8.3.

8.1. Burgers equation coupled to a transport process

We consider the following coupled problem

$$\begin{aligned} \partial_t \mathbf{v}(1+20h) + \epsilon\Delta\mathbf{v} + (\mathbf{v}\cdot\nabla)\,\mathbf{v} = \mathbf{f}, \quad \mathbf{v}(0) = \mathbf{v}^0, \\ \partial_t h + \operatorname{div}(h\mathbf{v}) + \epsilon_t\Delta h = g, \quad h(0) = h^0, \end{aligned} \tag{8.1}$$

on $\Omega = (-1,1)^2$, with $\epsilon = 0.01$ and $\epsilon_t = 0.1$. The initial conditions are given as

$$\mathbf{v}(0,x,y) = \mathbf{v}^0(x,y) := 0, \quad h(0,x,y) = h^0(x,y) := 0.1\exp\Big(-(x+0.0125)^2 - (y-0.0125)^2\Big).$$

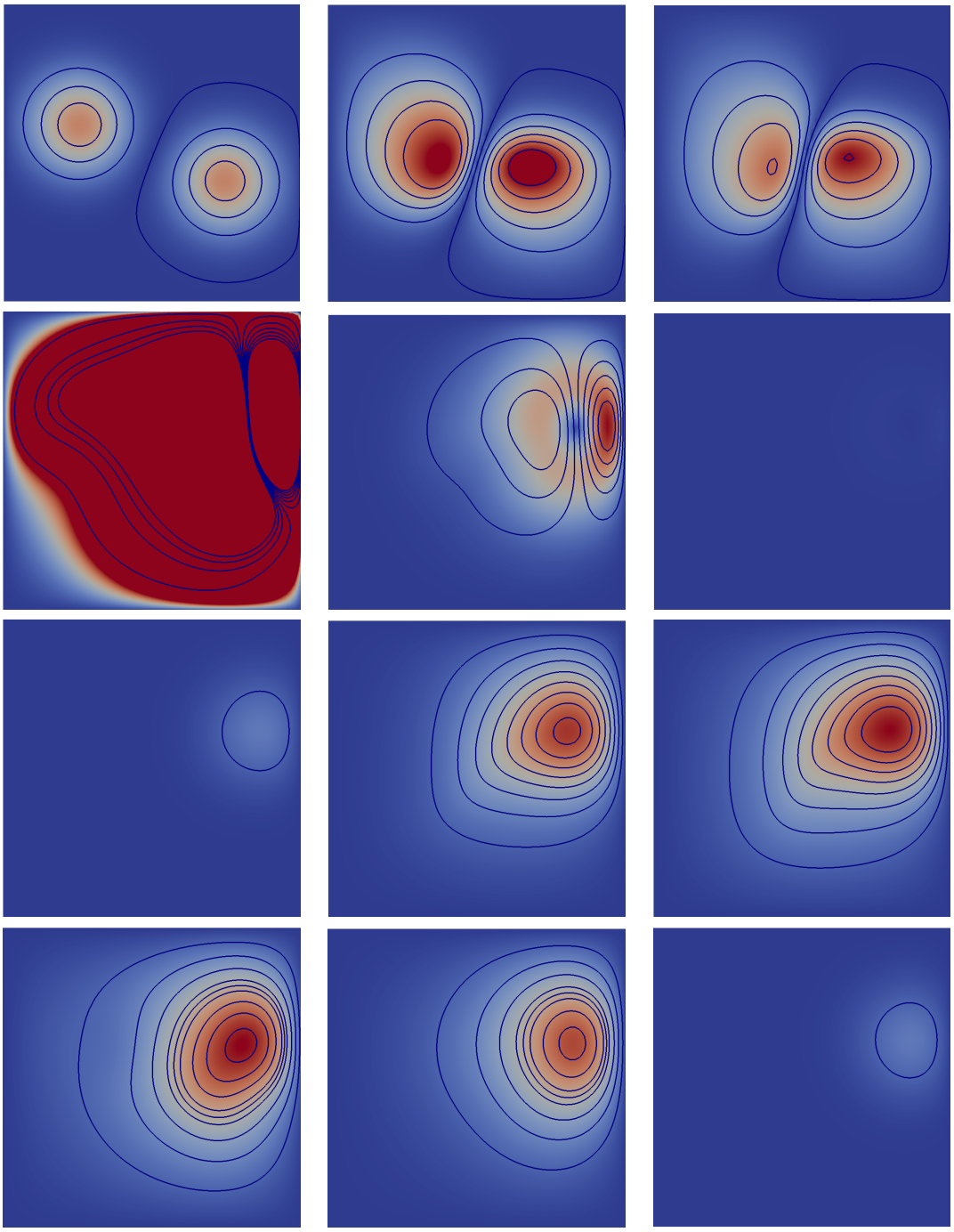


Figure 8.1.: In the top row, we show the numerical solution of Burgers equation $|\mathbf{v}_{k,h}|$. In the second row, we present the adjoint solution $|\mathbf{z}_{k,h}|$. In the third row, we plot the numerical solution of the transport process $h_{k,h}$. In the bottom row, we show the corresponding adjoint solution $q_{k,h}$.

As right hand side, we choose

$$\mathbf{f}(x,y,t) = \omega(t)\begin{pmatrix}\exp(-2\tilde{x}) - \exp(-2\tilde{y})\\ \exp(-2\tilde{y}) - \exp(-2\tilde{x})\end{pmatrix},$$

with

$$\omega(t) = \begin{cases}\cos(2\pi t), & t \le 0.5,\\ 0, & t > 0.5,\end{cases} \quad \text{and} \quad \begin{aligned}\tilde{x} &= \Big((x+0.5)^2 + (y-0.2)^2\Big),\\ \tilde{y} &= \Big((x-0.5)^2 + (y+0.2)^2\Big),\end{aligned}$$

and $g = 0$. At the boundaries, we set Dirichlet zero data for the velocity $\mathbf{v}$. We present snapshots of the primal and adjoint numerical solution of system (8.1) in Figure 8.1.

Variational formulation The variational formulation of problem (8.1) reads as

$$\begin{aligned}(\mathbf{v},h) \in X(I) : \int_I \Big(\partial_t \mathbf{v}(1+20h), \boldsymbol{\phi}\Big) + \epsilon\Big(\nabla\mathbf{v}, \nabla\boldsymbol{\phi}\Big) + \Big(\,(\mathbf{v}\cdot\nabla)\,\mathbf{v}, \boldsymbol{\phi}\Big)dt\\ + \Big(\mathbf{v}(0) - \mathbf{v}^0, \phi(0)\Big) = \int_I (\mathbf{f}, \boldsymbol{\phi})dt,\\ \int_I \Big((\partial_t h, \psi\Big) + \epsilon_t\Big(\nabla h, \nabla\psi\Big) + \Big(\,\mathrm{div}(\mathbf{v}h), \psi\Big)dt + \Big(h(0) - h^0, \psi(0)\Big) = 0,\\ \forall(\boldsymbol{\phi},\psi) \in X(I).\end{aligned} \tag{8.2}$$

For the definition of the function space $X(I)$ we refer to Section 7.1. Using the space X_I of piece wise continuous functions defined in Section 7.1, we formulate problem (8.2) as

$$\begin{aligned}(\mathbf{v},h) \in X_I : \quad \sum_{n=1}^{N}\int_{I_n} \Big(\partial_t \mathbf{v}(1+20h), \boldsymbol{\phi}\Big) + \epsilon\Big(\nabla\mathbf{v}, \nabla\boldsymbol{\phi}\Big) + \Big((\mathbf{v}\cdot\nabla)\mathbf{v}, \boldsymbol{\phi}\Big)dt\\ + \sum_{n=2}^{N}\Big((1+20h^+_{n-1})[\mathbf{v}]_{n-1}, \boldsymbol{\phi}^+_{n-1}\Big) + \Big((1+20h^+_0)(\mathbf{v}^+_0 - \mathbf{v}^0), \boldsymbol{\phi}^+_0\Big) = \int_{I_n}(\mathbf{f},\boldsymbol{\phi})dt\\ \sum_{n=1}^{N}\int_{I_n} \Big(\partial_t h, \psi\Big) + \Big(\,\mathrm{div}(\mathbf{v}h), \psi\Big) + \epsilon_t\Big(\nabla h, \nabla\psi\Big)dt\\ + \sum_{n=2}^{N}\Big([h]_{n-1}, \psi^+_{n-1}\Big) + \Big(h^+_0 - h^0, \psi^+_0\Big) = 0,\\ \forall(\boldsymbol{\phi},\psi) \in X_I.\end{aligned}$$

8.1.1. Partitioned solution approach and dG(0)dG(0) scheme in time

To solve the primal and adjoint system, we decouple it in time. This is realized by a projection operator (7.12) introduced in Section 7.2.

$$\mathcal{S} : X \to X_k, \quad \mathcal{S}(v)|_{I_n} = v^-(t_{n-1}),$$

where the space X_k is defined in Section 7.1. Doing so, the decoupled system is given as follows:

Algorithm 7 (Partitioned solution approach for the primal system)**.** *Let* $\mathbf{v}^0$ *and* h^0 *be the initial solutions at time* $t = 0$*. We iterate for* $n = 1, ..., N$*:*

1. *First, we solve Burgers equation,*

$$\begin{aligned}\int_{I_n} (\partial_t \mathbf{v}(1 + 20\mathcal{S}(h)), \boldsymbol{\phi}) + \epsilon(\nabla \mathbf{v}, \nabla \boldsymbol{\phi}) + ((\mathbf{v} \cdot \nabla)\mathbf{v}, \boldsymbol{\phi}) dt \\ + ((1 + 20\mathcal{S}(h)^+_{n-1})[\mathbf{v}]_{n-1}, \boldsymbol{\phi}^+_{n-1}) = \int_{I_n} (\mathbf{f}, \boldsymbol{\phi}) dt,\end{aligned} \tag{8.3}$$

   with $[\mathbf{v}]_0 := \mathbf{v}^+_0 - \mathbf{v}^0$*.*

2. *After that, we compute the solution of the transport equation,*

$$\begin{aligned}\int_{I_n} (\partial_t h, \psi) + \epsilon_t \Big(\nabla h, \nabla \psi\Big) + (\mathrm{div}(\mathbf{v}h), \psi) dt \\ + ([h]_{n-1}, \psi^+_{n-1}) = 0,\end{aligned} \tag{8.4}$$

   with $[h]_0 := h^+_0 - h^0$*.*

Based on the decoupling of the primal equation, we derive the decoupled dual problem. We differentiate the semilinear form (8.3) with respect to $\mathbf{v}$ and h in direction of Φ and Ψ. Analogously, we differentiate the variational formulation (8.4) with respect to $\boldsymbol{\phi}$ and ψ in direction $\mathbf{z}$ and q. Doing so, the order, in which the equations are solved, changes. The dual decoupled system reads as follows:

Algorithm 8 (Partitioned solution approach for the dual system)**.** *Let* $\mathbf{v}_n$ *and* q_n *be the discrete solution of the primal problem at* $t = t_n$*. We iterate for* $n = N, ..., 1$*:*

1. *First, we solve the dual transport equation,*

$$\begin{aligned}\int_{I_n} (\partial_t \Psi, q) + (\mathrm{div}(\mathbf{v}\Psi), q) + \epsilon_t \Big(\nabla \Psi, \nabla h\Big) + (\partial_t \mathbf{v} 20\mathcal{S}(\Psi), \mathbf{z}) dt \\ + \Big(20\mathcal{S}(\Psi)^+_{n-1}[\mathbf{v}]_{n-1}, \mathbf{z}^+_{n-1}\Big) + \Big([\Psi]_{n-1}, q^+_{n-1}\Big) = J(\Psi),\end{aligned} \tag{8.5}$$

   with

$$\Big(\mathcal{S}(\Psi)^+_0[\mathbf{v}]_0, \mathbf{z}^+_0\Big) := \Big(\mathcal{S}(\Psi)^+_0 \mathbf{v}^+_0, \mathbf{z}^+_0\Big)$$

   and

$$\Big([\Psi]_0, q^+_{n-1}\Big) := \Big(\Psi^+_0, q^+_0\Big),$$

   where $J(\cdot)$ *is the output functional.*

2. *Next, we compute the numerical solution of the adjoint Burgers equation,*

$$\begin{aligned}\int_{I_n} (\partial_t \Phi(1 + 20\mathcal{S}(h)), \mathbf{z}) + \epsilon((\nabla \Phi, \nabla \mathbf{z}) + ((\Phi \cdot \nabla)\mathbf{v}, \mathbf{z}) + ((\mathbf{v} \cdot \nabla)\Phi, \mathbf{z}) \\ + (\mathrm{div}(\Phi h), q) dt + \Big((1 + 20\mathcal{S}(h)^+_{n-1})[\Phi]_{n-1}, \mathbf{z}^+_{n-1}\Big) = 0,\end{aligned} \tag{8.6}$$

   with

$$\Big((1 + 20\mathcal{S}(h)^+_{n-1})[\Phi]_0, \mathbf{z}^+_0\Big) := \Big((1 + 20\mathcal{S}(h)^+_{n-1})\Phi^+_0, \mathbf{z}^+_0\Big).$$

Formulation as a time stepping scheme The system is discretized with a $dG(0)dG(0)$ method in time. Therefore, the time derivatives in (8.3), (8.4), (8.5) and (8.6) vanish, and based on the definition of the discontinuous functions (7.4) in Section 7.1, the temporal jump terms are given by

$$\begin{aligned}
\Big([\mathbf{v}]_{n-1}(1+20\mathcal{S}(h)^+_{n-1}),\phi^+_{n-1}\Big) &= \Big((\mathbf{v}_n-\mathbf{v}_{n-1})(1+20h_{n-1}),\boldsymbol{\phi}_n\Big),\\
\Big([h]_{n-1},\psi^+_{n-1}\Big) &= \Big(h_n-h_{n-1},\psi_n\Big),\\
\Big((1+20\mathcal{S}(h)^+_{n-1})[\Phi]_n,\mathbf{z}^+_{n-1}\Big) &= \Big(1+20h_{n-1}(\Phi_n-\Phi_{n-1}),\mathbf{z}_n\Big),\\
\Big(20\mathcal{S}(\Psi)^+_{n-1}[\mathbf{v}]_{n-1},\mathbf{z}^+_{n-1}\Big) &= \Big(20\Psi_{n-1}(\mathbf{v}_n-\mathbf{v}_{n-1}),\mathbf{z}_n\Big),\\
\Big([\Psi]_{n-1},q^+_{n-1}\Big) &= \Big(\Psi_n-\Psi_{n-1},q_n\Big).
\end{aligned} \tag{8.7}$$

The temporal integrals in (8.3), (8.4), (8.5) and (8.6) are approximated with the first order box rule, which is exact for a $dG(0)dG(0)$ time approximation if the problem parameters do not explicitly depend on time. The evaluation of the time-dependent right hand side

$$\int_{I_n}(\mathbf{f}(t),\phi_n)\,dt,$$

with the box rule would correspond to an approximation with the implicit Euler scheme in time. However, we choose the second order midpoint rule to evaluate the right hand side. Thus, we ensure that the error contribution due to the evaluation of the temporal integral is not dominant.

Next, we formulate the $dG(0)dG(0)$ discretization as a time stepping scheme.

Algorithm 9 (Discretization scheme of the forward problem)**.** *Given the time step $k_n = t_n - t_{n-1}$ and the initial solutions h^0 and $\mathbf{v}^0$ at $t = 0$, we compute for $n = 1, ..., N$:*

- *the numerical solution of Burgers equation*

$$\begin{aligned}
k_n\epsilon\Big(\nabla\mathbf{v}_n,\nabla\boldsymbol{\phi}_n\Big) + k_n\Big((\mathbf{v}_n\cdot\nabla)\mathbf{v}_n,\boldsymbol{\phi}_n\Big)&\\
+\Big((1+20h_{n-1})\mathbf{v}_n,\boldsymbol{\phi}_n\Big) &= \Big((1+20h_{n-1})\mathbf{v}_{n-1},\boldsymbol{\phi}_n\Big) + \int_{I_n}(\mathbf{f},\boldsymbol{\phi}_n)\,dt,
\end{aligned}$$

  with $\mathbf{v}_0 := \mathbf{v}^0$ and $h_0 := h^0$,

- *and the approximation of the transport process*

$$\epsilon_t\Big(\nabla h_n,\nabla\psi_n\Big) + k_n\Big(\operatorname{div}(\mathbf{v}_n h_n),\psi_n\Big) + \Big(h_n,\psi_n\Big) = \Big(h_{n-1},\psi_n\Big),$$

  with $h_0 := h^0$.

We continue with reformulating the dual equations (8.5) and (8.6) to a time stepping scheme. The adjoint Burgers equation discretized with a $dG(0)dG(0)$ scheme is given by

$$\sum_{n=1}^{N} k_n\Big(\epsilon(\nabla\Phi_n,\nabla\mathbf{z}_n)+\big((\Phi_n\cdot\nabla)\mathbf{v}_n,\mathbf{z}_n\big)+\big((\mathbf{v}_n\cdot\nabla)\Phi_n,\mathbf{z}_n\big)+\big(\operatorname{div}(\Phi_n h_n),q_n\big)\Big) + \sum_{n=2}^{N}\Big((1+20h_{n-1})(\Phi_n-\Phi_{n-1}),\mathbf{z}_n\Big)+\Big((1+20h^0)\Phi_1,\mathbf{z}_1\Big)=0, \tag{8.8}$$

whereas the adjoint transport equation reads as

$$\sum_{n=1}^{N} k_n\Big(\epsilon_t(\nabla\Psi_n,\nabla q_n)+\operatorname{div}(\mathbf{v}_n\Psi_n),q_n\Big)+\sum_{n=2}^{N}\Big(20\Psi_{n-1}(\mathbf{v}_n-\mathbf{v}_{n-1}),\mathbf{z}_n\Big) + \sum_{n=2}^{N}\Big((\Psi_n-\Psi_{n-1}),q_n\Big)+\Big(\Psi_1,q_1\Big)=J(\Psi_n). \tag{8.9}$$

In order to formulate the adjoint system as time stepping scheme, we sort the formulations (8.8) and (8.9) by the test functions. Then, the discretization scheme reads as follows:

Algorithm 10 (Discretization scheme of the backward problem)**.** *Let $\mathbf{v}_n$ and h_n be the discrete solution of the primal system. Given a time step $k_n = t_n - t_{n-1}$, we compute:*

1. *For $n = N$*
   - *the approximation of the adjoint transport equations*
   $$k_N\Big(\epsilon_t(\nabla\Psi_N,\nabla q_N)+(\operatorname{div}(\mathbf{v}_N\Psi_N),q_N)\Big)+(\Psi_N,q_N)=J(\Psi_N)$$
   - *and the solution of the adjoint Burgers equation*
   $$k_N\Big(\epsilon(\nabla\Phi_N,\nabla\mathbf{z}_N)+\big((\Phi_N\cdot\nabla)\mathbf{v}_N,\mathbf{z}_N\big)+\big((\mathbf{v}_N\cdot\nabla)\Phi_N,\mathbf{z}_N\big) +\big(\operatorname{div}(\Phi_N h_N),q_N\big)\Big)+(1+20h_{N-1})(\Phi_N,\mathbf{z}_N)=0.$$
2. *For $n = N-1, ..., 1$*
   - *the solution of the adjoint transport equations*
   $$k_n\Big(\epsilon_t(\nabla\psi_n,q_n)+\big(\operatorname{div}(\mathbf{v}_n\Psi_n),q_n\big)\Big)+(\Psi_n,q_n)=\Big(\Psi_n,q_{n+1}\Big)+J(\Psi_n) -\Big(20\Psi_n([\mathbf{v}]_n),\mathbf{z}_{n+1}\Big),$$
   - *and the approximation of the adjoint Burgers equation*
   $$k_n\Big(\epsilon(\nabla\Phi_n,\nabla\mathbf{z}_n)+\big((\Phi_n\cdot\nabla)\mathbf{v}_n,\mathbf{z}_n\big)+\big((\mathbf{v}_n\cdot\nabla)\Phi_n,\mathbf{z}_n\big) +\big(\operatorname{div}(\Phi_n h_n),q_n\big)\Big)+\Big(\Phi_n,(1+20h_{n-1})\mathbf{z}_n\Big)=\Big(\Phi_n,(1+20h_n),\mathbf{z}_{n+1}\Big),$$
   with $h_0 := h^0$.

8.1.2. Numerical evaluation of the error estimator

The aim of this section is to evaluate the accuracy of the new error estimator and to analyze under which conditions the *splitting error* dominates the overall numerical error. The error estimator, derived in Theorem 2 in Section 7.2.2, can be divided into two error contributions, the *splitting error* η_β due to the application of a partitioned solution approach, and the *residual error* η_ρ, due to the approximation of the primal and the dual residual. The latter can be split into the *spatial residual error*, η_{ρ_h} and the *temporal residual error* η_{ρ_k}. The different parts of the error estimator are defined in (7.36).

In the following, we evaluate the error estimator for Burgers equation coupled to transport process (8.1). We set $U = (\mathbf{v}, h)$. As functional of interest, we consider the time averaged functional

$$J(U) = \int_I \int_\Omega h(x, y, t) \exp(-10\bar{x}^2 - 10\bar{y}^2) dx\, dy\, dt, \quad \bar{x} = x - 0.75, \quad \bar{y} = y - 0.25, \tag{8.10}$$

which measures the solution h, averaged in space and time, with a weight on the upper right part of the domain. For the evaluation of the error estimator, we use the reference value

$$\tilde{J} = 0.319180 \pm 10^{-5},$$

received on very high temporal and spatial resolutions. We analyze the error estimator by using different time steps k, and by varying spatial resolutions, which are characterized by the number of nodes N. To analyze the effectivity of the error estimator, we introduce the effectivity index

$$\text{eff} := \frac{\tilde{J} - J(U^s_{k,h})}{\eta}. \tag{8.11}$$

For an accurate error estimator the index should be converging to 1.

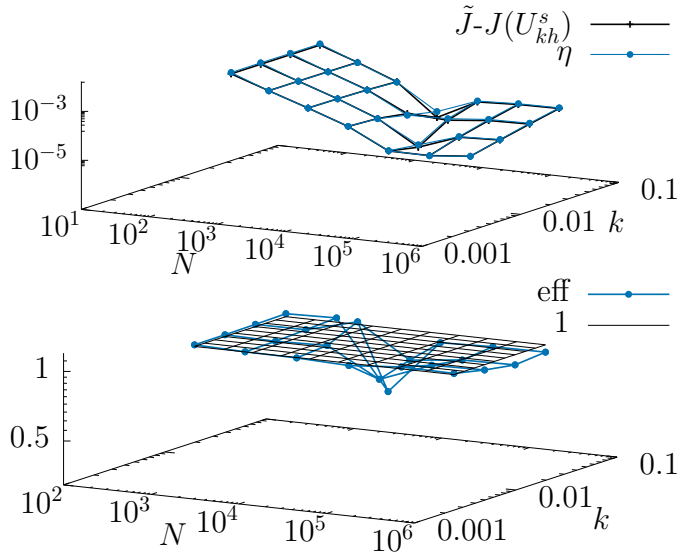


Figure 8.2.: In the upper graph, we plot the functional error $\tilde{J} - J(U^s_{k,h})$ and the error estimator η. In the lower graph, we plot the effectivity index eff and a plane with value 1. The number of nodes varies between $N = 81$ and $N = 263169$, whereas the time step decreases from $k = 0.125$ to $k = 0.0078125$.

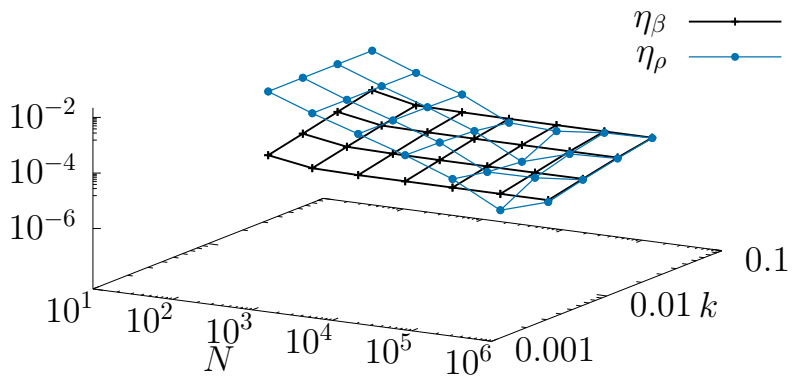


Figure 8.3.: We plot the *splitting error* η_β and the error of the *temporal* and *spatial residual* η_ρ for decreasing $k = 0.125$ to $k = 0.00782125$ and increasing nodes from $N = 81$ to $N = 263169$.

k	N	$\tilde{J} - J(U^s_{k,h})$	η	η_ρ	η_β
0.0625	81	$2.12{\cdot}10^{-2}$	$2.41{\cdot}10^{-2}$	$2.46{\cdot}10^{-2}$	$-5.67{\cdot}10^{-4}$
	289	$5.12{\cdot}10^{-3}$	$4.97{\cdot}10^{-3}$	$5.20{\cdot}10^{-3}$	$-2.34{\cdot}10^{-4}$
	1089	$1.00{\cdot}10^{-3}$	$9.59{\cdot}10^{-4}$	$1.17{\cdot}10^{-3}$	$-2.14{\cdot}10^{-4}$
	4225	$-3.38{\cdot}10^{-5}$	$-6.39{\cdot}10^{-5}$	$1.43{\cdot}10^{-4}$	$-2.07{\cdot}10^{-4}$
	16641	$-2.92{\cdot}10^{-4}$	$-3.21{\cdot}10^{-4}$	$-1.16{\cdot}10^{-4}$	$-2.05{\cdot}10^{-4}$
	66049	$-3.57{\cdot}10^{-4}$	$-3.86{\cdot}10^{-4}$	$-1.80{\cdot}10^{-4}$	$-2.05{\cdot}10^{-4}$
	263169	$-3.73{\cdot}10^{-4}$	$-4.01{\cdot}10^{-4}$	$-1.97{\cdot}10^{-4}$	$-2.05{\cdot}10^{-4}$
0.03125	81	$2.14{\cdot}10^{-2}$	$2.47{\cdot}10^{-2}$	$2.49{\cdot}10^{-2}$	$-2.64{\cdot}10^{-4}$
	289	$5.29{\cdot}10^{-3}$	$5.18{\cdot}10^{-3}$	$5.30{\cdot}10^{-3}$	$-1.26{\cdot}10^{-4}$
	1089	$1.17{\cdot}10^{-3}$	$1.15{\cdot}10^{-3}$	$1.27{\cdot}10^{-3}$	$-1.16{\cdot}10^{-4}$
	4225	$1.44{\cdot}10^{-4}$	$1.23{\cdot}10^{-4}$	$2.36{\cdot}10^{-4}$	$-1.13{\cdot}10^{-4}$
	16641	$-1.15{\cdot}10^{-4}$	$-1.36{\cdot}10^{-4}$	$-2.28{\cdot}10^{-5}$	$-1.12{\cdot}10^{-4}$
	66049	$-1.79{\cdot}10^{-4}$	$-2.00{\cdot}10^{-4}$	$-8.78{\cdot}10^{-5}$	$-1.12{\cdot}10^{-4}$
	263169	$-1.96{\cdot}10^{-4}$	$-2.17{\cdot}10^{-4}$	$-1.04{\cdot}10^{-4}$	$-1.13{\cdot}10^{-4}$
0.015625	81	$2.15{\cdot}10^{-2}$	$2.49{\cdot}10^{-2}$	$2.50{\cdot}10^{-2}$	$-1.21{\cdot}10^{-4}$
	289	$5.40{\cdot}10^{-3}$	$5.30{\cdot}10^{-3}$	$5.36{\cdot}10^{-3}$	$-6.04{\cdot}10^{-4}$
	1089	$1.28\ {\cdot}10^{-3}$	$1.27{\cdot}10^{-3}$	$1.32{\cdot}10^{-3}$	$-5.59{\cdot}10^{-5}$
	4225	$2.42{\cdot}10^{-4}$	$2.37{\cdot}10^{-4}$	$2.92{\cdot}10^{-4}$	$-5.46{\cdot}10^{-5}$
	16641	$-1.73{\cdot}10^{-5}$	$-2.25{\cdot}10^{-5}$	$3.18{\cdot}10^{-5}$	$-5.43{\cdot}10^{-5}$
	66049	$-8.21{\cdot}10^{-5}$	$-8.74{\cdot}10^{-5}$	$-3.32{\cdot}10^{-5}$	$-5.42{\cdot}10^{-5}$
	263169	$-9.83{\cdot}10^{-5}$	$-1.04{\cdot}10^{-4}$	$-4.94{\cdot}10^{-5}$	$-5.42{\cdot}10^{-5}$
0.0078125	81	$2.16{\cdot}10^{-2}$	$2.50{\cdot}10^{-2}$	$2.50{\cdot}10^{-2}$	$-5.77{\cdot}10^{-5}$
	289	$5.45{\cdot}10^{-3}$	$5.36{\cdot}10^{-3}$	$5.39{\cdot}10^{-3}$	$-2.90{\cdot}10^{-5}$
	1089	$1.33{\cdot}10^{-3}$	$1.33{\cdot}10^{-3}$	$1.35{\cdot}10^{-3}$	$-2.67{\cdot}10^{-5}$
	4225	$2.93{\cdot}10^{-4}$	$2.93{\cdot}10^{-4}$	$3.19{\cdot}10^{-4}$	$-2.61{\cdot}10^{-5}$
	16641	$3.28{\cdot}10^{-5}$	$3.36{\cdot}10^{-5}$	$5.96{\cdot}10^{-5}$	$-2.60{\cdot}10^{-5}$
	66049	$-3.20{\cdot}10^{-5}$	$-3.14{\cdot}10^{-5}$	$-5.45{\cdot}10^{-6}$	$-2.60{\cdot}10^{-5}$
	263169	$-4.82{\cdot}10^{-5}$	$-4.78{\cdot}10^{-5}$	$-2.17{\cdot}10^{-5}$	$-2.60{\cdot}10^{-5}$

Table 8.1.: We compare the functional error $\tilde{J} - J(U^s_{k,h})$ to the error estimator η, which can be decomposed in the splitting error η_β and the primal and dual residual error η_ρ. We denote the time step size by k and the number of nodes by N.

In the upper image in Figure 8.2, we plot the functional error $\tilde{J} - J(U^s_{k.h})$ and the error estimator η for decreasing time steps k and increasing nodes N. We observe a kink passing diagonally through the graph. The kink is due to a changing sign of the error functional. The changing sign is also reproduced by the error estimator. The accuracy of the error estimator suffers close to the kink. This can be seen in the lower image in Figure 8.2. Here, we plot the effectivity index eff (8.11). On coarse meshes with $N = 289$ and $N = 1089$ nodes, we observe high effectivity indices close to 1. The index is decreasing when the sign of the error functional is changing. After the change, the index is again converging to 1.

Analyzing Table 8.1, we observe quadratic convergence of the error functional $\tilde{J} - J(U^s_{k,h})$ and of the error estimator η as long as the spatial error is dominant. This is expected as we use bi-linear finite elements in space. Comparing the functional error $\tilde{J} - J(U^s_{k,h})$ and the error estimator η on high spatial resolutions, we observe linear convergence for decreasing time steps. Here, the temporal error dominates the spatial discretization error. The linear temporal convergence stems from the use of the $dG(0)dG(0)$ method.

In Figure 8.3, we compare the *residual error* η_ρ and the *splitting error* η_β for varying temporal and spatial resolution. As shown in Table 8.1, we observe that the *splitting error* is converges linear in time due to the first order projection (7.12). Further, it is constant in space. This is expected, as the *splitting error* is a pure temporal error.

Different linear high-order reconstructions As the error estimator (7.30) consists of interpolation errors $x^s - \mathcal{I}_k x^s$ as well as semi-discrete solutions U^s_k and Z^s_k, an approximation of these terms is necessary to compute the error estimator. We analyze the effect of different linear interpolations in time to reconstruct these quantities. In particular, we compare linear interpolations with different interpolation points, set at the left boundary, the right boundary and the center of an interval I_n.

$$
\begin{aligned}
\textbf{left:} \quad & i^{(1)}_{k,l} U|I_n := \frac{t - t_{n-1}}{t_n - t_{n-1}} U^-(t_n) + \frac{t - t_n}{t_{n-1} - t_n} U^-(t_{n-1}), \\
\textbf{right:} \quad & i^{(1)}_{k,r} U|I_n := \frac{t - t_{n-1}}{t_n - t_{n-1}} U^+(t_n) + \frac{t - t_n}{t_{n-1} - t_n} U^+(t_{n-1}), \\
\textbf{center:} \quad & i^{(1)}_{k,m} U|I_n := \frac{t - t_{n-1}}{t_n - t_{n-1}} \left(\frac{U^-(t_n) + U^+(t_n)}{2} \right) + \frac{t - t_n}{t_{n-1} - t_n} \left(\frac{U^-(t_n) + U^+(t_n)}{2} \right).
\end{aligned}
\tag{8.12}
$$

We visualize the different high-order reconstructions in Figure 8.4.

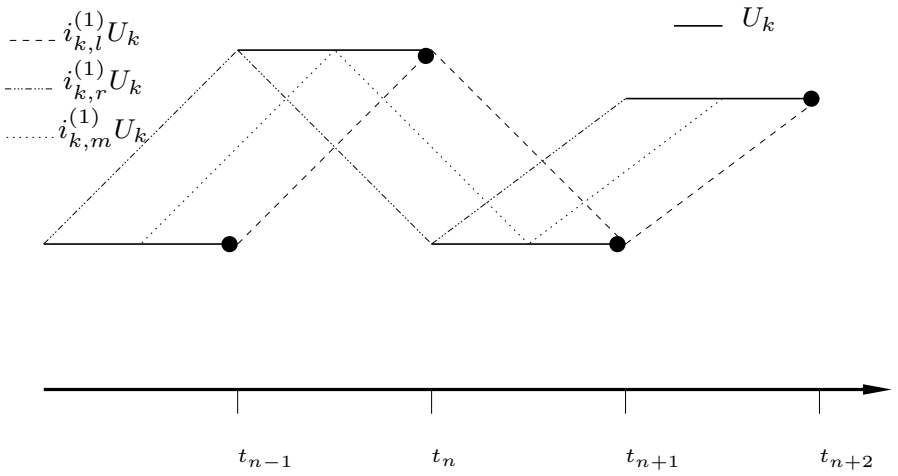


Figure 8.4.: Piecewise linear interpolations of piecewise continuous function. Using different interpolation points set at the left boundary ($i^{(1)}_{k,l}U_k$), at the right boundary ($i^{(1)}_{k,r}U_k$) and at the midpoint ($i^{(1)}_{k,m}U_k$) of an interval I_n.

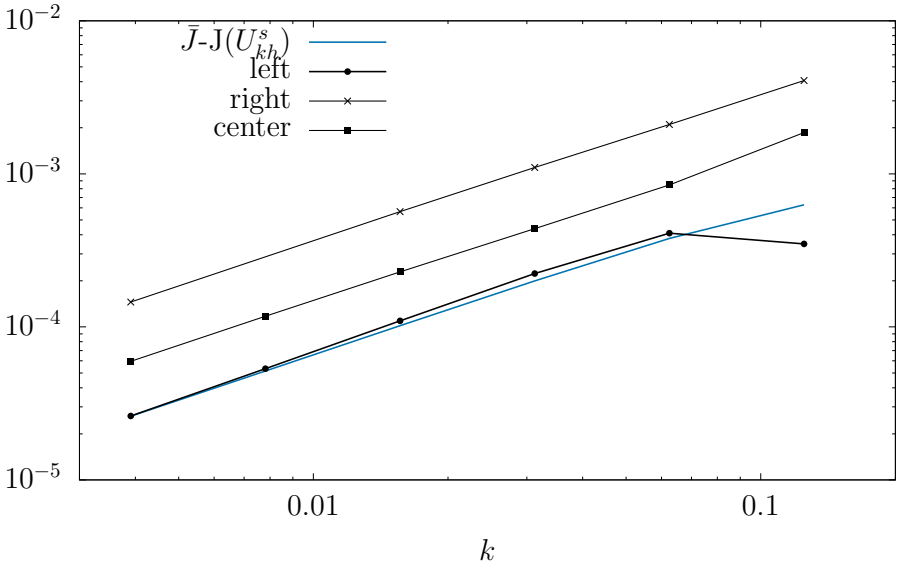


Figure 8.5.: We compare the temporal functional error $\bar{J} - J(U^s_{k,h})$ and the error estimator η on a mesh with $N = 1089$ nodes using different time steps k. The error estimator is computed with varying high-order reconstructions of $U^s_{k,h}$ and $Z^s_{k,h}$. The reconstructions use different interpolation points: *Left* refers to the interpolation point at the left boundary of the temporal interval, *right* to the point at the right boundary of the temporal interval and *center* to the interval midpoint.

To analyze the effect that different high-order reconstructions have on the numerical approximation, we chose a fixed grid with $N = 1089$ nodes and identify for this mesh resolution the reference value of the goal functional (8.10) by choosing the time step k very small.

$$\bar{J} = 0.317800899 \pm 10^{-6}.$$

In Figure 8.5, we compare the error functional to the error estimator η using the different linear interpolations introduced in (8.12). We observe that the chosen interpolation point has a huge impact on the constant of the error estimator. As shown in Figure 8.5, only the error estimator using a high-order reconstruction with an interpolation point at the left boundary of I_n yields highly accurate results. Hence, we use this construction for all further test cases.

8.2. Sea ice model

In this section, we evaluate the error estimator derived in Theorem 2 for the viscous-plastic sea ice model. Further, we analyze the potential of the adaptive refinement strategy introduced in Algorithm 6 in Section 7.4.

This section is structured as follows. In Section 8.2.1, we introduce the partitioned solution approach for the sea ice model and derive its primal and adjoint formulation. For simplicity, we discretize the sea ice momentum equation and the balance laws with a $dG(0)dG(0)$ formulation in time. Since the implicit Euler scheme is standard in the sea ice community [53], the extension to $cG(1)dG(0)$ scheme is subject to future work and could follow the work of Richter and Meidner [79]. In Section 8.2.2, we numerically evaluate the error estimator and analyze the potential of the adaptive mesh refinement strategy.

To simplify the notation of the balance laws in the sea ice model, we introduce $\mathbf{h} = (H, A)$. Following the derivation and the notation introduced in Section 7.1, we reformulate the sea ice model to

$$\begin{aligned}
&\sum_{n=1}^{N} \int_{I_n} \{(\rho_{ice} H \partial_t \mathbf{v}, \boldsymbol{\phi}) + (f_c \rho_{ice} H \boldsymbol{e}_r \times (\mathbf{v} - \mathbf{v}_{\text{ocean}}), \boldsymbol{\phi}) - (\boldsymbol{\tau}_{\text{atm}}, \boldsymbol{\phi}) \\
&-(\boldsymbol{\tau}_{\text{ocean}}(\mathbf{v}), \boldsymbol{\phi}) + (\boldsymbol{\sigma}(\mathbf{v}, H, A), \nabla \boldsymbol{\phi})\} dt + \sum_{n=1}^{N} \left(\rho_{ice} H_{n-1}^{+} [\mathbf{v}]_{n-1}, \boldsymbol{\phi}_{n-1}^{+} \right) = 0, \\
&\sum_{n=1}^{N} \int_{I_n} \{(\partial_t \mathbf{h} + \operatorname{div}(\mathbf{v}\mathbf{h}), \boldsymbol{\psi})\} dt + \sum_{n=1}^{N} \left([\mathbf{h}]_{n-1}, \boldsymbol{\psi}_{n-1}^{+} \right) = 0,
\end{aligned} \tag{8.13}$$

where the stress tensor and the ice strength are given as

$$\begin{aligned}
\boldsymbol{\sigma}(\mathbf{v}, H, A) &= 2\eta(\dot{\boldsymbol{\epsilon}}) \dot{\boldsymbol{\epsilon}}' + \zeta(\dot{\boldsymbol{\epsilon}}) \operatorname{tr}(\dot{\boldsymbol{\epsilon}}) I - \frac{P(H, A)}{2} I, \\
P(H, A) &= P^{\star} H \exp\big(- C(1 - A)\big).
\end{aligned} \tag{8.14}$$

Ocean and atmospheric stress are described as

$$\tau_{\text{ocean}} = C_{\text{ocean}}\rho_{\text{ocean}}\|\mathbf{v}_{\text{ocean}} - \mathbf{v}\|(\mathbf{v}_{\text{ocean}} - \mathbf{v}), \quad \tau_{\text{atm}} = C_{\text{atm}}\rho_{\text{atm}}\|\mathbf{v}_{\text{atm}}\|\mathbf{v}_{\text{atm}}.$$

8.2.1. A partitioned solution approach for the primal and dual system

To solve the sea ice model in time, we apply an operator splitting approach and use the projection introduced in (7.12) in Section 7.2,

$$\mathcal{S} : X \to X_k, \quad \mathcal{S}(v)|_{I_n} = v^-(t_{n-1}),$$

where the space X_k is defined in Section 7.1. Introducing the projection $\mathcal{S}(\cdot)$, the sea ice momentum equation reads as

$$\begin{aligned}\sum_{n=1}^{N}\int_{I_n}\{(\rho_{ice}\mathcal{S}(H)\partial_t\mathbf{v},\boldsymbol{\phi}) + (f_c\rho_{ice}\mathcal{S}(H)\boldsymbol{e}_r\times(\mathbf{v}-\mathbf{v}_{\text{ocean}}),\boldsymbol{\phi}) - (\boldsymbol{\tau}_{\text{atm}},\boldsymbol{\phi})\\ -(\boldsymbol{\tau}_{\text{ocean}}(\mathbf{v}),\boldsymbol{\phi}) + (\boldsymbol{\sigma}(\mathbf{v},\mathcal{S}(H),\mathcal{S}(A)),\nabla\boldsymbol{\phi})\}dt + \sum_{n=1}^{N}\left(\rho_{ice}\mathcal{S}(H)^+_{n-1}[\mathbf{v}]_{n-1},\boldsymbol{\phi}^+_{n-1}\right) = 0.\end{aligned} \tag{8.15}$$

The balance laws are given as

$$\sum_{n=1}^{N}\int_{I_n}\{(\partial_t\mathbf{h} + \operatorname{div}(\mathbf{v}\mathbf{h}),\psi)\}dt + \sum_{n=1}^{N}\left([\mathbf{h}]_{n-1},\psi^+_{n-1}\right) = 0. \tag{8.16}$$

To derive the adjoint sea ice model, we differentiate the semilinear forms (8.15) and (8.16) with respect to $\mathbf{v}$ and $\mathbf{h}$ in direction Φ and Ψ, with $\boldsymbol{\psi} = (\psi_H, \psi_A)$. Analogously, we differentiate the variational formulation with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ in direction of $\mathbf{z}$ and $\mathbf{q}$. The adjoint momentum equation reads as

$$\begin{aligned}\sum_{n=1}^{N}\int_{I_n}\{(\rho_{ice}\mathcal{S}(H)\partial_t\Phi,\mathbf{z}) + (f_c\rho_{ice}\mathcal{S}(H)\boldsymbol{e}_r\times\Phi,\mathbf{z}) - \boldsymbol{\tau}'_{\text{ocean}}(\mathbf{v})(\Phi,\mathbf{z})\\ +\boldsymbol{\sigma}'(\mathbf{v},\mathcal{S}(H)\mathcal{S}(A))(\Phi,\nabla\mathbf{z}) + (\operatorname{div}(\Phi\mathbf{h}),\mathbf{q})\}dt + \sum_{n=1}^{N}\left(\rho_{ice}\mathcal{S}(H)^+_{n-1}[\Phi]_{n-1},\mathbf{z}^+_{n-1}\right) = 0,\end{aligned} \tag{8.17}$$

whereas the adjoint balance laws are given by

$$\begin{aligned}\sum_{n=1}^{N}\int_{I_n}\{(\partial_t\Psi + \operatorname{div}(\mathbf{v}\Psi),\mathbf{q})\}dt + \sum_{n=1}^{N}\left([\Psi]_{n-1},\mathbf{q}^+_{n-1}\right)\\ + \sum_{n=1}^{N}\int_{I_n}\{(\partial_t\mathbf{v}\rho_{ice}\mathcal{S}(\Psi_H),\mathbf{z}) + (f_c\rho_{ice}\mathcal{S}(\Psi_H)\boldsymbol{e}_r\times(\mathbf{v}-\mathbf{v}_{ocean}),\mathbf{z})\\ +\boldsymbol{\sigma}'(\mathbf{v},\mathcal{S}(H),\mathcal{S}(A))(\mathcal{S}(\Psi_H),\nabla\mathbf{z}) + \boldsymbol{\sigma}'(\mathbf{v},\mathcal{S}(H),\mathcal{S}(A))(\mathcal{S}(\Psi_A),\nabla\mathbf{z})dt\}\\ + \sum_{n=1}^{N}\left(\rho_{ice}\mathcal{S}(\Psi_H)^+_{n-1}[\mathbf{v}],\mathbf{z}^+_{n-1}\right) = 0.\end{aligned}$$

Note that the derivatives of $\boldsymbol{\sigma}'$ and τ'_{ocean} are given in Theorem 1. The *partitioned solution algorithm* for the primal system is given by Algorithm 1 in Section 3.1. The *partitioned solution algorithm* for the dual sea ice equations is derived analogously to Algorithm 8 in Section 8.1.1.

8.2.2. Numerical evaluation of the error estimator

In this section, we numerically analyze the error estimator derived in Section 7.2.2 for the sea ice model. In order to apply the error estimator (7.30), we solve the sea ice model problem introduced in Section 2.6 without a stabilization of the balance laws. By neglecting the stabilization an additional consistency error is avoided.

Like in Section 4.2, as functional of interest, we evaluate the sea ice extent measured in a certain area for a time span of interest I_J.

$$J_A(A) = \int_{I_J} \int_{\Omega_2} A(x, y, t)\, d(x, y)\, dt, \tag{8.18}$$

with $\Omega_2 = (375, 500) \times (375, 500)$. I_J is specified later.

Test cases *We analyze the following two modifications of the model problem introduced in Section 2.6,*

A1 *As in Section 4.2, we use $A^0 = 0.8$ and $H^0 = 0.3$ as initial conditions for our model problem to ensure that A, H do not reach the limits $0 < A, H$ and $A < 1$. To avoid failures of the solver, we reduce the simulation time to $I = [0, 2]$ days and the temporal interval of interest where the functional is evaluated, is set to $I_J = [1, 2]$ days. The reference value achieved by spatial and temporal extrapolation, is given as*

$$\tilde{J}_{A1} = 0.942083 \pm 10^{-4}. \tag{8.19}$$

A2 *We modify the initial values of the model problem and use a small variation of the ice concentration at $A = 0.9$ to increase the spatial variation in the beginning of the simulation,*

$$A^0(x, y) = 0.9 + 0.005 \left(\cos\left(\frac{x}{5km}\right) + \cos\left(\frac{y}{5km}\right) \right).$$

We simulate the model problem for $I = [0, 6]$ days where temporal interval of interest is set to $I_J = [3, 6]$. Here, we identify the reference value by an extrapolation as

$$\tilde{J}_{A2} = 3.62 \pm 10^{-2}. \tag{8.20}$$

k	h	$J_{A1}(U^s_{k,h})$	$\tilde{J}_{A1} - J_{A1}(U^s_{k,h})$	η	η_{ρ_h}	η_{ρ_k}	η_β
1 h	8 km	0.94166	$4.20{\cdot}10^{-4}$	$4.74{\cdot}10^{-4}$	$-9.27{\cdot}10^{-6}$	$4.73{\cdot}10^{-4}$	$9.40{\cdot}10^{-6}$
	4 km	0.94164	$4.47{\cdot}10^{-4}$	$4.78{\cdot}10^{-4}$	$-6.29{\cdot}10^{-6}$	$4.75{\cdot}10^{-4}$	$9.45{\cdot}10^{-6}$
	2 km	0.94162	$4.61{\cdot}10^{-4}$	$4.83{\cdot}10^{-4}$	$-4.57{\cdot}10^{-6}$	$4.78{\cdot}10^{-4}$	$9.27{\cdot}10^{-6}$
0.5 h	8 km	0.94189	$1.83{\cdot}10^{-4}$	$2.31{\cdot}10^{-4}$	$-8.91{\cdot}10^{-6}$	$2.37{\cdot}10^{-4}$	$2.55{\cdot}10^{-4}$
	4 km	0.94187	$2.09{\cdot}10^{-4}$	$2.35{\cdot}10^{-4}$	$-6.01{\cdot}10^{-6}$	$2.38{\cdot}10^{-4}$	$2.58{\cdot}10^{-6}$
	2 km	0.94186	$2.23{\cdot}10^{-4}$	$2.38{\cdot}10^{-4}$	$-4.36{\cdot}10^{-6}$	$2.40{\cdot}10^{-4}$	$2.50{\cdot}10^{-6}$
0.25 h	8 km	0.94201	$6.57{\cdot}10^{-5}$	$1.10{\cdot}10^{-4}$	$-8.74{\cdot}10^{-6}$	$1.19{\cdot}10^{-4}$	$7.37{\cdot}10^{-7}$
	4 km	0.94199	$9.11{\cdot}10^{-5}$	$1.14{\cdot}10^{-4}$	$-5.87{\cdot}10^{-6}$	$1.19{\cdot}10^{-4}$	$7.51{\cdot}10^{-7}$
	2 km	0.94197	$1.04{\cdot}10^{-4}$	$1.16{\cdot}10^{-4}$	$-4.25{\cdot}10^{-6}$	$1.20{\cdot}10^{-4}$	$7.19{\cdot}10^{-7}$

Table 8.2.: We compare the functional errors $\tilde{J}_{A1} - J_{A1}(U^s_{k,h})$ to the error estimator η, which consists of the *spatial* and *temporal residual error*, η_{ρ_h} and η_{ρ_k}, as well as the *splitting error* η_β. The time step size is denoted by k and the mesh size by h. The reference value $\tilde{J}_{A1}$ is achieved by temporal and spatial extrapolation.

k	h	$J_{A2}(U^s_{k,h})$	$\tilde{J}_{A2} - J_{A2}(U^s_{k,h})$	η	η_{ρ_h}	η_{ρ_k}	η_β
1 h	8 km	3.5709	$4.91{\cdot}10^{-2}$	$2.68{\cdot}10^{-3}$	$2.53{\cdot}10^{-3}$	$-9.12{\cdot}10^{-4}$	$1.06{\cdot}10^{-3}$
	4 km	3.5908	$2.92{\cdot}10^{-2}$	$2.01{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$-6.98{\cdot}10^{-4}$	$9.23{\cdot}10^{-4}$
	2 km	3.5952	$2.48{\cdot}10^{-2}$	$-1.21{\cdot}10^{-3}$	$2.04{\cdot}10^{-4}$	$-9.03{\cdot}10^{-5}$	$-1.33{\cdot}10^{-3}$
0.5 h	8 km	3.5702	$4.98{\cdot}10^{-2}$	$2.12{\cdot}10^{-3}$	$2.05{\cdot}10^{-3}$	$-4.56{\cdot}10^{-4}$	$5.24{\cdot}10^{-4}$
	4 km	3.5928	$2.71{\cdot}10^{-2}$	$1.97{\cdot}10^{-3}$	$1.85{\cdot}10^{-3}$	$-3.59{\cdot}10^{-4}$	$4.81{\cdot}10^{-4}$
.	2 km	3.6037	$1.63{\cdot}10^{-2}$	$-6.9{\cdot}10^{-4}$	$7.24{\cdot}10^{-5}$	$8.79{\cdot}10^{-5}$	$-8.55{\cdot}10^{-4}$
0.25 h	8 km	3.5697	$5.03{\cdot}10^{-2}$	$1.82{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$-2.31{\cdot}10^{-4}$	$2.60{\cdot}10^{-4}$
	4 km	3.5940	$2.60{\cdot}10^{-2}$	$2.10{\cdot}10^{-3}$	$2.04{\cdot}10^{-3}$	$-1.88{\cdot}10^{-4}$	$2.46{\cdot}10^{-4}$
	2 km	3.6096	$1.04{\cdot}10^{-2}$	$-5.53{\cdot}10^{-4}$	$-1.43{\cdot}10^{-4}$	$4.08{\cdot}10^{-5}$	$-4.51{\cdot}10^{-4}$

Table 8.3.: We compare the functional errors $\tilde{J}_{A2} - J_{As}(U^s_{k,h})$ to the error estimator η, which consists of the *spatial* and *temporal residual error*, η_{ρ_h} and η_{ρ_k}, as well as the *splitting error* η_β. The time step size is denoted by k and the mesh size by h. The reference value $\tilde{J}_{A2}$ is achieved by temporal and spatial extrapolation.

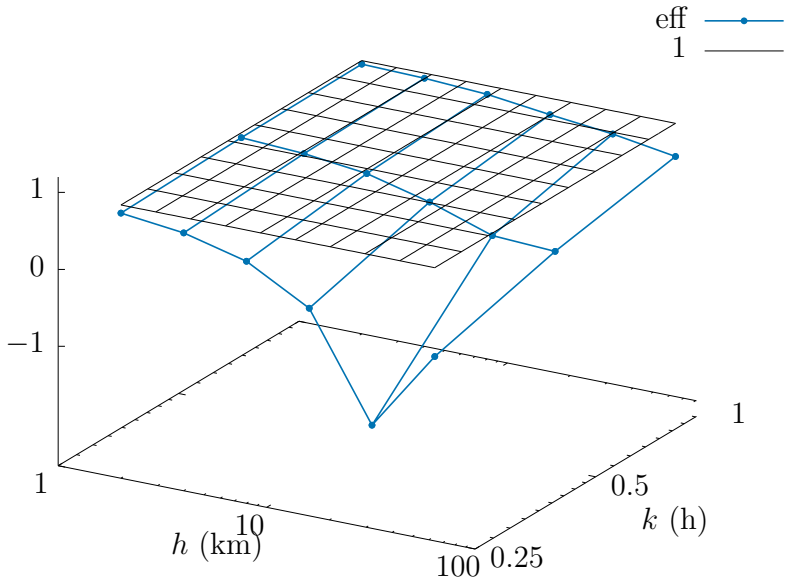


Figure 8.6.: We plot the effectivity index eff for test case A1.

We begin by analyzing test case A1. In Table 8.2, we evaluate the functional error $\tilde{J}_{A1} - J_{A1}(U^s_{k,h})$ and the error estimator η. The functional error and the error estimator converge linear in time and without a specific order in space. Further, analyzing the different error contributions of the error estimator, we observe that the *temporal residual error* dominates the whole error estimator η on all considered mesh levels and for all time steps. This is unusual for sea ice simulations. In most of our test cases, we observe a dominating *spatial error* as discussed in [68] or shown in Table 4.2 and Table 4.3, in Section 4.2. The dominating *temporal residual error* might stem from the short simulation time of $T = 2$ days and the idealized initial conditions.

In Figure 8.6, we plot the effectivity index eff as defined in (8.11),

$$\text{eff} := \frac{\tilde{J} - J(U^s_{k,h})}{\eta},$$

and observe a highly accurate estimator with increasing spatial and temporal resolutions.

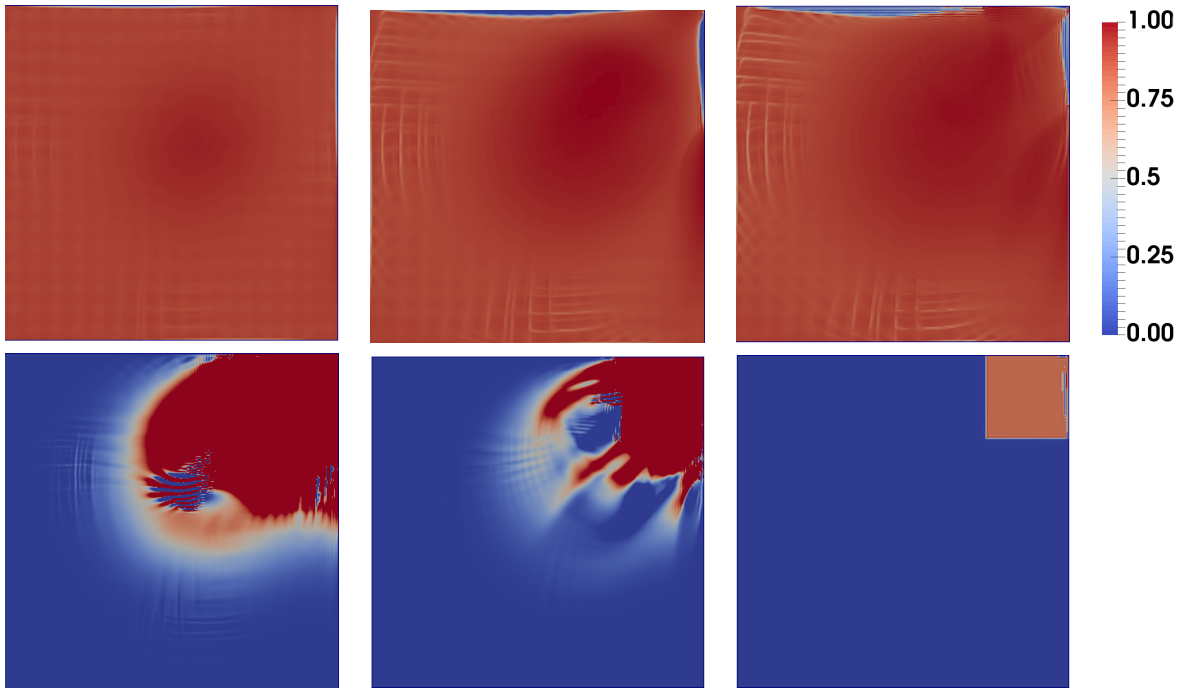


Figure 8.7.: For test case A2, we show snapshots of the computed ice concentration and the corresponding adjoint approximation on a 2 km mesh. The adjoint approximation ranges between 6 and 600. For a better visualization we limit the scale from 0 to 6.

We proceeding by discussing test case A2. In Figure 8.7, we show snapshots of the ice concentration and the adjoint approximation. Analyzing Table 8.3, we find that the functional error $\tilde{J}_{A2} - J_{A2}(U^s_{k,h})$ converges linear in space and time. We observe that the error estimator η is smaller by a factor 20 compared to the functional error.

The decreased performance might stem from the fact that the limits of $0 \leq H, A$ and $A \leq 1$ are reached in the second example but in the first example. We drop the constrains in the derivation of the dual system. The implementation of the constrains in the error estimator increases the complexity of the estimator as it requires considering a system of variational inequalities [93]. The decreased performance might be also explained by the increased spatial variation of the approximation. The derivation of the error estimator involves the third order derivative of the semilinear form (7.21). In case of the sea ice momentum equation the third derivative can scales as Δ_{min}^{-5} with $\Delta_{min} = 2 \cdot 10^{-9}$ as it forms part of the viscosities,

$$\zeta = \frac{P}{2\Delta(\dot{\boldsymbol{\epsilon}})}, \quad \Delta(\dot{\boldsymbol{\epsilon}}) := \sqrt{(2e^{-2}\dot{\boldsymbol{\epsilon}}' : \dot{\boldsymbol{\epsilon}}' + \operatorname{tr}(\dot{\boldsymbol{\epsilon}})^2) + \Delta_{min}^2}.$$

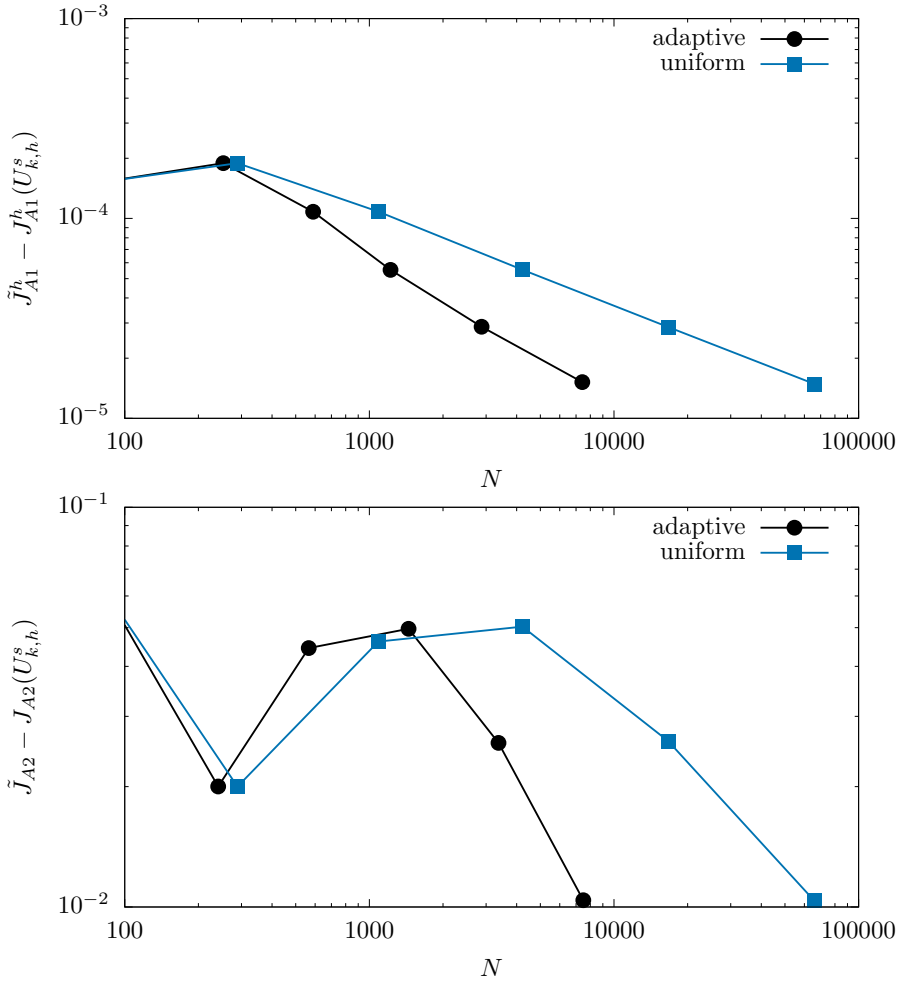


Figure 8.8.: Comparison of the functional error on locally refined and uniform meshes. The number of nodes is denoted by N. We evaluate the first test case A1 in the upper graph and the second test case A2 in the lower graph.

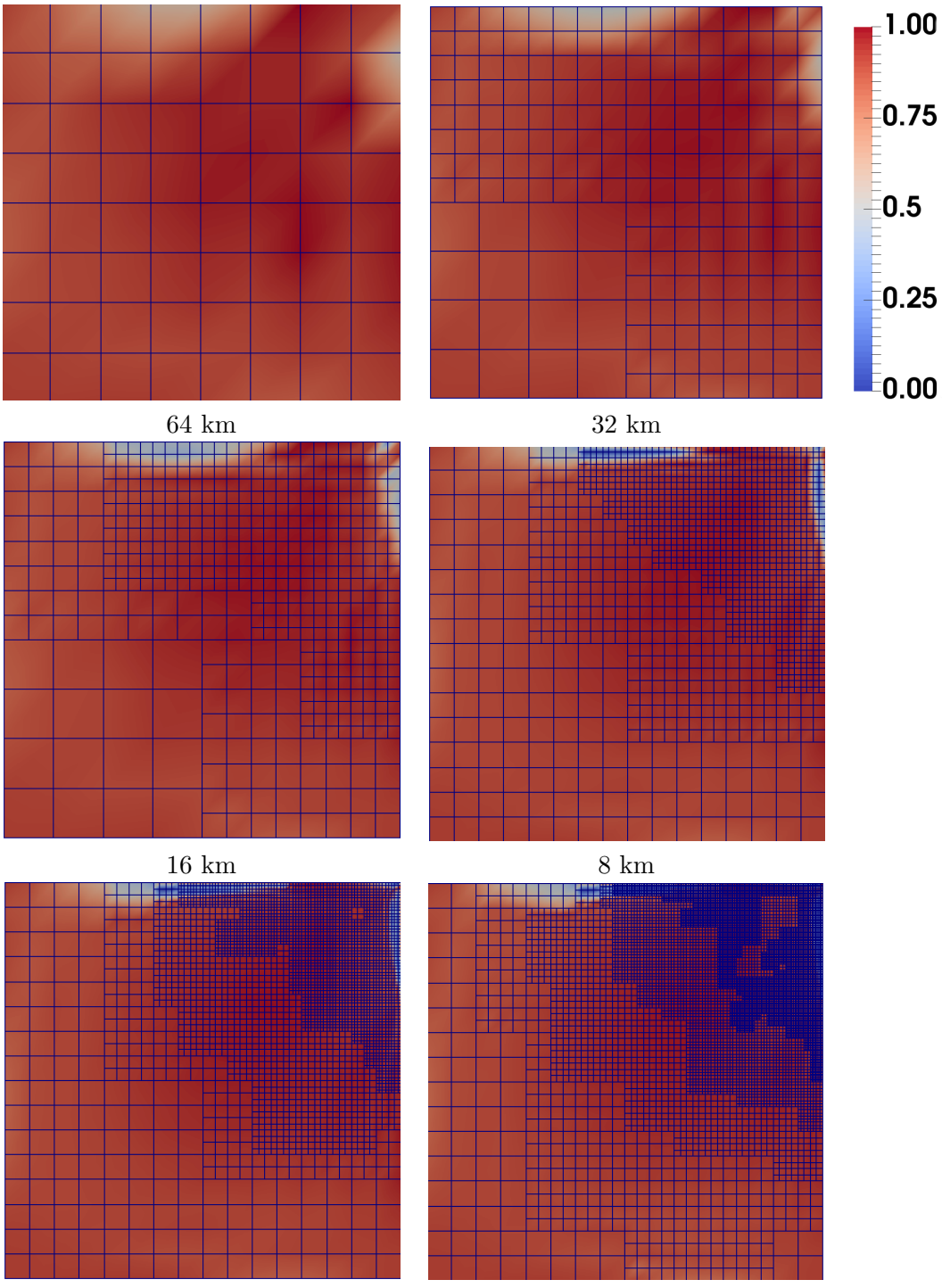


Figure 8.9.: We show adaptive spatial meshes for test case A2, achieved by using Algorithm 6 presented in Section 7.4.

In the derivation of the error estimator we drop the third order derivative (7.21). In case of the sea ice model this might be an unsuitable approximation. As it is shown in Figure 2.6 in Section 2.6, the spatial variability of the viscosity enlarges with increasing resolution. So far, it is uncertain if it is possible to observe convergence in space as high resolution meshes of the Arctic and robust solvers are required. In approximation of the solution of the sea ice problem, one might get similar problems as they exist in simulating turbulence, where meshes of required resolution are not feasible. In addition, the robustness of nonlinear solvers of the sea ice model decreases at high spatial resolutions, such that computing a numerical solution becomes more difficult.

We proceed with analyzing the error estimator η and observe that the error estimator η converges in space and in time, for time steps smaller than $k = 1$ h without a specific convergence order. However, analyzing the different error contributions of the error estimator on coarse meshes, we find that the *splitting error* η_β and the *temporal residual error* η_{ρ_k} converge linear in time. The *spatial error* dominates the latter two error contributions, except of the highest spatial resolution. Here the *splitting error* dominates the overall error. However, we do not observe explicit spatial convergence of the *spatial residual error* η_{ρ_h}. This might be due to the fact that the spatial variation of the numerical solution is increasing.

As the functional error in the first test case is always below 0.1%, the second test case might be more relevant for practice. Here, the error ranges between 8 % and 1 %, which is relevant order of magnitude in applications.

To evaluate the adaptive mesh refinement strategy presented in Algorithm 6 in Section 7.4, we keep the time step k fixed and compare the approximation accuracy of the functional of interest by using uniform and adaptive meshes. On uniform meshes, we identify the reference value of the goal functional for the first test case A1 as $J^h_{A1} = 0.941607 \pm 10^{-5}$ by an extrapolation in space. As the spatial error is dominant in the second test case A2, we use $\tilde{J}_{A2}$ as a reference value. Analyzing Figure 8.8, we observe that using adaptive meshes, we reach the same approximation accuracy of the goal functional as on uniform meshes with 9 times less nodes. In Figure 8.9, we show snapshots of the different adaptive meshes used to compute the numerical solution of the test case A2. The mesh size decreases from 64 km to 2 km.

8.2.3. Practical relevance of adaptive meshes in large-scale sea ice models

Considering numerical methods, the topical focus of the sea ice community lies in efficiently simulating sea ice dynamics [31, 64]. Here, the topics are the development of fast solvers that are easy to parallelize [64], the investigation in new rheologies that are simple to solve and reasonably accurate [98], or even whole sea ice model approximations such as the EVP model [31]. Hereby, only a few researchers address the accuracy of numerical simulations [40, 55]. In that light, the implementation of a technically challenging error estimator might not be the prior task.

- Parallelized sea ice models take advantage of regular grid structure, e.g., in setting up the LSOR preconditioner in the *Jacobian-free Newton-Krylov* solver [64]. An

adaptive mesh algorithm changes the structure of the grid. Thus, implementing it in a parallelized model might require a modification of the parallelization structure.

- Furthermore, when coupling sea ice models to atmospheric and oceanic models, the complexity of the sea ice model increases. Following Lemieux [51], the coupling of CICE (sea ice model) to an oceanic and atmospheric model produces instabilities in the boundary layer when high spatial resolutions are applied. The use of adaptive meshes in sea ice models might increase the difficulties in coupling them to oceanic and atmospheric models, as the latter two are solved on uniform meshes with different resolutions.

- Applying the a posteriori error estimator in coupled sea ice models requires the derivation of a fully coupled dual model. Meaning, in order to compute the dual system backwards in time, the ocean and atmosphere models also need to be solved backwards in time. This will extremely increase the computational cost in applying adaptive meshes. Possible simplifications to approximate the dual solution need to be investigated, e.g., using time averaged solutions for the dual system as presented by Braack et al. [9].

Nevertheless, the error estimator can be used to balance the overall numerical error in sea ice simulations when measuring a quantity of interest. It provides information about the different error contributions and thus, it is a useful tool to increase the efficiency in sea ice simulations by determining and controlling stopping criteria.

8.3. Conclusion

In this chapter, we evaluated an error estimator for strongly nonlinear momentum equations that are coupled to a transport process and solved with a partitioned solution approach. In particular, we consider a model consisting of Burgers equation coupled to a transport process and the sea ice model. For both models, we presented the primal and dual variational formulation. In case of Burgers equation, we described the $dG(0)dG(0)$ formulation of the coupled system and formulated it as a time stepping scheme.

The error estimator introduced in Theorem 2 in Section 7.2 is based on the evaluation of the interpolation error $x^s - \mathcal{I}x^s$, where $x^s = (U^s, Z^s)$, and the semi-discrete solution, U_k^s and Z_k^s. To make the error estimator computable, we use a linear interpolation of $U_{k,h}^s$ and $Z_{k,h}^s$. We numerically analyzed the effect of different linear interpolations to calculate the error estimator and observed that the linear interpolation with the interpolation point at the left boundary of the temporal interval yields the approximation with the highest accuracy.

To evaluate the error estimator for Burgers equation coupled with a transport process, we introduced a space and time averaged goal functional that has a weight on the upper right region of the domain. We showed high accuracy of the error estimator η as the effectivity index converges to 1 for increasing spatial and temporal resolutions. For large

time steps and high spatial resolutions, we found that the *splitting error* η_β dominates the *residual error* η_ρ.

In contrast to Burgers equation, which is coupled to the transport process in the temporal differential operator, the coupling in the sea ice model is taking place in the spatial and temporal differential operator. This increases the difficulty of error estimation. In case of the sea ice model, we considered a goal functional that measures the sea ice extent averaged over a subdomain and time span of interest. We analyzed two modifications of the model problem introduced in Section 2.6 and observed a highly accurate error estimator η as long as the *temporal residual error* η_{ρ_k} dominates the error estimator. This is untypical for sea ice simulations as shown in [68] and might be due to the short simulation time and the idealized initial conditions. For increased spatial variation, we observed that the accuracy of the error estimator decreases, meaning the effectivity index is about 20. Here, we observed that the *splitting error* dominates the overall numerical error on a 2 km mesh.

In addition to the error estimator, we apply the adaptive mesh refinement algorithm to the sea ice problem. We found that the adaptive mesh refinement algorithm yields the same approximation accuracy of the functional of interest as the uniform mesh approach but using 9 times less nodes. Assuming solvers of linear complexity, these savings carry over to the computational time.

As efficiently solving sea ice dynamics on high resolutions is still a big issue, including adaptive meshes may not be the prior task in sea ice modeling, e.g., including the error estimator in parallelized models is technically challenging as it might require a modification of the parallelization structure. Further, the coupling of sea ice, oceanic and atmospheric models at high resolutions is still under active research as instabilities are occurring. Introducing adaptive meshes for the sea ice model might further increase these difficulties. For an evaluation of the goal oriented error estimator in coupled sea ice models, the oceanic and the atmospheric models need to be calculated backwards in time. This extremely increases the computational costs of a simulation.

Nevertheless, using adaptive meshes in stand-alone sea ice models to simulate short-term forecasts with focus on small spatial subdomains might be useful. Further, the goal oriented error estimator gives us the opportunity to balance the different error contributions by measuring a quantity of interest. Doing so, the efficiency of sea ice simulations is increased.

9. Conclusion and outlook

In this thesis, we presented robust and efficient numerical methods to simulate sea ice dynamics on high resolution grids, with a cell wise up to 2 km. The simulations are based on the *viscous-plastic* sea ice model introduced by Hibler in [25], which is currently applied in most sea ice models [56]. The model consists of a nonlinear momentum equation that is coupled with two transport equations. To discretize the sea ice model, we used a *finite element approach* in space and applied a *partitioned solution approach* in time. Doing so, the momentum and transport equations are solved separately in each time step.

We found that the approximation of the transport equations are very sensitive to the applied time stepping method and the used stabilization scheme. In particular, we compared the approximation achieved with the *second order flux corrected Taylor-Galerkin scheme* to numerical solution computed with the *Crank-Nicolson* scheme stabilized with a SUPG method, and observed large variations between the numerical solutions of the different schemes. We chose these two discretization schemes as they showed the best performance in a simplified test case. Using a *second order flux corrected Taylor-Galerkin* scheme, we observed artifacts in the approximation of the transport equations, called *terracing*. These stem from the use of *Zalesak's flux limiter* and might be avoided by developing a new limiting strategy for the fluxes. By modifying the partitioned solution approach we were able to circumvent the negative effect of *terraces* when solving the sea ice model. In order to obtain a robust approximation of the momentum equation, it has shown to be essential to couple the intermediate solutions $A_n^{\text{low}}, H_n^{\text{low}}$ into the momentum equation. It is subject to future work to analyze the impact on the accuracy of the velocity field in this new approach.

To solve the nonlinear momentum equation discretized with an implicit Euler scheme in each time step, we applied a Newton solver, globalized with a *line search method* and accelerated with the *operator-related damped Jacobian* scheme. We showed that the Jacobian of the sea ice model is *positive definite*, thus global convergence of the Newton method is provided by theory from cited literature assuming an optimal damping parameter. We proved that the Jacobian of the sea ice model is *symmetric* apart from the Coriolis term, and can be divided into a *positive definite* part and a *negative semidefinite* part. The idea of the *operator-related damped Jacobian* method is to adaptively control the negative semidefinite part of the Jacobian by evaluating the convergence history of the Newton solver. This new acceleration method significantly increases the robustness of the Newton scheme on high spatial resolutions. Analyzing the performance of the solver on a realistic simulation on a 10 km pan-Arctic grid, we found that the new Newton approach could reduce the number of failures of the currently applied JFNK solver from 24% solver to less than 1%. We assume that the increase in robustness will

be even larger with increasing resolutions.

As the existing linear solver in the JFNK approach is extremely costly due to the absence of efficient preconditioners, we showed the potential of the *multigrid preconditioner* to increase the efficiency of the linear solver. By solving an idealized test case, the multigrid preconditioner reduces the iteration count by up to 80% on the highest resolution in comparison to the use of an *incomplete lower upper factorization* as a preconditioner. As the iteration count increases slightly with higher resolutions, we concluded that the multigrid preconditioner is a suitable method for sea ice simulations on high spatial resolutions.

In the last part of this thesis, we derived a *goal oriented error estimator for partitioned solution approaches* that is applicable to the viscous-plastic sea ice model. The error estimator is able to approximate the different error contributions that occur in measuring a functional of interest. It consists of the *spatial* and *temporal discretization error* as well as the *splitting error* due to the use of a partitioned solution approach. In terms of the sea ice model, we measured the sea ice extent, which is considered in seasonal and short-term forecastings. The error estimator is based on the *dual weighted residual method* introduced by Becker and Rannacher [6] and is applicable for a general class of non-stationary differential equations that are coupled to a transport process. Based on the approximation of the error estimator, we derived an *adaptive refinement strategy* that balances the different error contribution to reduce the overall numerical error.

Finally, we evaluated the error estimator on a model consisting of Burgers equation coupled to a transport process and observed a highly accurate estimator. In case of the sea ice model, the error estimator is highly accurate as long as the temporal error is dominating the error estimator. For numerical solutions with larger variations in space, the performance of the error estimator is decreasing. For both models, we observed a dominating *splitting error* on high resolution grids.

Furthermore, using adaptive meshes in sea ice simulations, we where able to reach the same accuracy in measuring the sea ice extent with 9 times less nodes compared to uniform meshes. Here, we reduced the error in measuring the sea ice extent from 7% to 2%, which is a magnitude of interest when estimating the minimal sea ice extent in the Arctic.

As currently, most applied methods to solve the viscous-plastic sea ice model either converge poorly or are not at all at high spatial resolutions, we conclude by recommending the application of the novel Newton approach with the multigrid method as a preconditioner to the linear solver in sea ice models. Both methods showed a huge potential to increase the efficiency of a simulation and they are relatively easy to realize. Regarding the use of adaptive meshes, it might be technically too complex to integrate them into coupled sea ice models. Nevertheless, the error estimator gives us information on the magnitude of the different error contributions when measuring a quantity of interest. This can be used to balance the overall numerical error, i.e., select the correct choice of spatial and temporal resolutions to increase the efficiency in sea ice simulations.

Outlook We introduced a *non-dimensional variational formulation* of the sea ice model for a better evaluation of the numerical discretization. Hereby, we derived three dimensionless quantities that describe the external ocean and atmospheric forcing as well as the forcing due to internal stresses. It remains to analyze the physical meaning of these properties and to investigate how they are related to the convergence issues of the numerical solvers at high spatial resolutions.

In fluid mechanics it is common to design *benchmark problems* that verify the quality of the discretization and validate the efficiency of new numerical methods. So far, no benchmark problem exists to validate the numerical methods when solving sea ice dynamics. The formulation of our model problem may be a first step in developing a benchmark problem. As we observed a huge variability in the approximation of transport equations a benchmark problem might be helpful to verify discretization methods. Regarding the discretization of the transport equations, it is left to analyze the effect of the different discretizations in realistic model simulations.

Before the *geometric multigrid preconditioner* can be applied in realistic sea ice simulations, it needs to be investigated how the nested mesh hierarchy can be set up for complex geometries, e.g., how to approximate regions that contain small islands and complex structured coastlines. With respect to that, an overview of applications of the multigrid method on complex geometries using unstructured grids is given by Wesseling and Oosterlee in [106].

We already outlined that implementation of the *goal oriented error estimator* for *partitioned solution approaches* might be technically too complex to integrate it into parallelized coupled climate models. It is subject to future work to analyze the potential of the error estimator in short-term forecasts in stand-alone sea ice models. It needs to be investigated in which form the error estimator could be approximated in these models. The use of averaged dual quantities, as is proposed by Braack et al. [9], might be a good starting point. Nevertheless, we suggest investigating the potential of *energy error estimators* in sea ice simulations as they are simpler in derivation and more robust in application.

As we observed a high accurate error estimator for Burgers equation coupled to transport process, it is subject to future work to evaluate the potential of the *goal oriented error estimator* for *partitioned solution approach* in a $cG(1)dG(0)$ discretization. A possible test case are the incompressible Navier-Stokes equations coupled to a transport process.

Finally, we note that the Newton solver accelerated with the *operator-related damped Jacobian* method can be applied to sea ice models with other rheologies or different plastic yield curves such as the Mohr-Coulomb curve [99], as long as the Jacobian has the same structure. So far, a first analysis has been conducted by Richter and Mehlmann [81] to theoretically explain the increased robustness of the *operator-related damped Jacobian* method. However, it is not fully understood yet and subject to future work.

Acknowledgement

A special thanks goes to the *Deutsche-Bundesstiftung Umwelt* for the support of my PhD project and the great opportunities offered in their PhD program. Furthermore, I would like to thank the DFG for the support of the graduate college MathCoRe, which enabled me to participate in several conferences. Finally, I thank the BMBF *Professorinnen Programm II* for supporting my research stay at *Environmental Canada*.

Bibliography

[1] A. Adcroft, J.M. Campin, S. Dutkiewicz, C. Evangelinos, D D. Ferreira, G. Forget, B. Fox-Kemper, P. Heimbach, C. Hill, E Hill, H. Hill, O. Jahn, M. Losch, J. Marshall, G. Maze, D. Menemenlis, and A. Molod. *MITgcm User Manual*, 2018.

[2] F. Andreu-Vaillo, V. Caselles, and J.M. Mazón. *Parabolic Quasilinear Equations Minimizing Linear Growth Functionals*. Progress in Mathematics. Springer, 2004.

[3] P. Bastian, K. Birken, K. Johannsen, S. Lang, V. Reichenberger, C. Wieners, G. Wittum, and C. Wrobel. Parallel Solutions of Partial Differential Equations with Adaptive Multigrid Methods on Unstructured Grids. In *High performance computing in science and engineering II*, 1999.

[4] R. Becker and M. Braack. Multigrid techniques for finite elements on locally refined meshes. *Numerical Linear Algebra with Applications*, 7:363–379, 2000. Special Issue.

[5] R. Becker, M. Braack, D. Meidner, T. Richter, and B. Vexler. The finite element toolkit Gascoigne. www.gascoigne.de.

[6] R. Becker and R. Rannacher. An optimal Control Approach to A Posteriori Error Estimation in Finite Element Methods. In A. Iserles, editor, *Acta Numerica 2001*, volume 37, pages 1–225. Cambridge University Press, 2001.

[7] J. Beitler. Minimum Seaice Extend. http://nsidc.org/arcticseaicenews/2012/09/arctic-sea-ice-extent-settles-at-record-seasonal-minimum/, accessed Feburary 25, 2019.

[8] S. Bouillon, T. Fichefet, V. Legat, and G. Madec. The elastic-viscous-plastic method revisited. *Ocean Modelling*, 71:2–12, 2013.

[9] M. Braack, E. Burman, and N. Taschenberger. Duality Based A Posteriori Error Estimation for Quasi-Periodic Solutions Using Time Averages. *SIAM J. Sci. Comput.*, 33:2199–2216, 2011.

[10] M.B. Cardenas, D. T. Slottke, R. A. Ketcham, and J. M. Sharp. Navier-Stokes flow and transport simulations using real fractures shows heavy tailing due to eddies. *Geophysical Research Letters*, 34(14), 2007.

[11] P.G. Ciarlet. On Korn's inequality. *Chin. Ann. Math.*, 31:607–618, 2010.

[12] M.D. Coon. A review of AIDJEX modeling. In *Sea Ice Processes and Models: Symposium Proceedings*, pages 12–27. Univ. of Wash. Press, Seattle., 1980.

[13] M.D. Coon, G.A. Maykut, R.S. Pritchard, D.A. Rothrock, and A.S. Thorndike. Modeling the pack ice as an elastic-plastic material. *AIDJEX Bull.*, 24:1–05, 1974.

[14] H. Damanik, J. Hron., A. Ouazzi, and S. Turek. Monolithic Newton–multigrid solution techniques for incompressible nonlinear flow models. *Int. J. Numer. Math. Fluids.*, 71:208–222, 2013.

[15] S. Danilov, Q. Wang, R. Timmermann, N. Iakovlev, D. Sidorenko, M. Kimmritz, T. Jung, and J. Schröter. Finite-Element Sea Ice Model (FESIM), version 2. *Geosci. Model Dev.*, 8:1747–1761, 2015.

[16] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*, volume 35 of *Computational Mathematics*. Springer, 2011.

[17] S.C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput*, 17:16–32, 1996.

[18] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Computational differential equations. *Cambridge Univ. Press*, 1996.

[19] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Applied Mathematical Sciences, 159, Springer, 2004.

[20] G. Gao, C. Chen, J. Qi, and R. C. Beardsley. An unstructured-grid, finite-volume sea ice model: Development, validation, and application. *Journal of Geophysical Research: Oceans*, 116(C8), 2011.

[21] L. Girard, S. Bouillon, J. Weiss, D. Amitrano, T. Fichefet, L. Thierry, and V. Legat. A new modeling framework for sea-ice mechanics based on elasto-brittle rheology. *Annals of Glaciology*, 52:123–132, 2011.

[22] R. Glowinski. Finite element methods for the numerical simulation of incompressible viscous flow. introduction to the control of the Navier-Stokes equations. *Lect. Appl. Math.*, 28:219–301, 1991.

[23] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, 1985.

[24] M. Harder. *Dynamics,roughness, and age of Arctic sea ice - Numerical investigations with a large-scale model*. PhD thesis, University of Bremen, 1996.

[25] W.D. Hibler. A dynamic thermodynamic sea ice model. *J. Phys. Oceanogr*, 9:815–846, 1979.

[26] W.D. Hibler and E. M. Schulson. On modeling the anisotropic failure and flow of flawed sea ice. *J. Geophys. Res.*, 105(C7):17105–17120, 2000.

[27] P.D. Hovland and L.C. Mc Innes. Parallel simulation of compressible flow using automatic differentiation and PETSc. *Parallel Computing*, 27:503–519, 2001.

[28] J. Hron, A. Ouazzi, and S. Turek. A computational comparison of two FEM solvers for nonlinear incompressible flow. In *Lecture Notes in Computational Science and Engineering*, volume 35, pages 87–109. Springer, 2003. New York.

[29] E. Hunke, W.H. Lipscomb, A.K. Turner, N. Jeffery, and S. Elliott. *CICE: the Los Alamos Sea Ice Model Documentation and Software User's Manual Version 5.1 LA-CC-06-012*, 2015.

[30] E.C. Hunke. Viscous-plastic sea ice dynamics with the EVP model: Linearization issues. *J. Comp. Phys.*, 170:18–38, 2001.

[31] E.C. Hunke and J.K. Dukowicz. An elastic-viscous-plastic model for sea ice dynamics. *J. Phys. Oceanogr.*, 27:1849–1867, 1997.

[32] E.C. Hunke, W.H. Lipscomb, and A.K. Tuner. Sea-ice models for climate study: retrospective and new directions. *Journal of Glaciology*, 200:1162–1172, 2010.

[33] J.K. Hutchings, H. Jasak, and S.W. Laxon. A strength implicit correction scheme for the viscous-plastic sea ice model. *Ocean Modelling*, 7:111–133, 2004.

[34] C.F. Ip, W.D. Hibler, and G.M. Flato. On the effect of rheology on seasonal sea-ice simulations. *Annals of Glaciology*, 15:17–25, 1991.

[35] F. Irgens. *Continuum Mechanics*. Springer-Verlag Berlin Heidelberg, 2008.

[36] V. John and E. Schmeyer. Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg*, 198:475 – 495, 2008.

[37] C. Johnson. *Numerical Solution of partial Differential equations by the Finite Element Method*. Dover Publications, Inc. Mineola, New York, 2009.

[38] M. Kimmritz, S. Danilov, and M. Losch. On the convergence of the modified elastic-viscous-plastic method for solving the sea ice momentum equation. *J. Comp. Phys.*, 296:90–100, 2015.

[39] M. Kimmritz, S. Danilov, and M. Losch. The adaptive EVP method for solving the sea ice momentum equation. *Ocean Modelling*, 101:59–67, 2016.

[40] M. Kimmritz, M. Losch, and S. Danilov. A comparison of viscous-plastic sea ice solvers with and without replacement pressure. *Ocean Modelling*, 115:59–69, 2017.

[41] M. Kimmritz and T. Richter. Parallel multigrid method for finite element simulations of complex flow problems on locally refined meshes. *Numerical Linear Algebra with Applications*, 18:615–636, 2010.

[42] E. Kleine and S. Sklyar. Mathematical features of Hibler's model of large-scale sea-ice dynamics. *Deutsche Hydrographische Zeitschrift*, 47:179–230, 1995.

[43] N. Kliem. A sea ice forecasting system for the waters around Greenland. *6th International Conference on Port and Ocean Engineering under Arctic Conditions*, 3:1217–1226, 2001.

[44] D. Knoll and D. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comp. Phys.*, 193:357–397, 2004.

[45] M. Kreyscher, M. Harder, P.Lemke, G. Flato, and M. Gregory. Results of the Sea Ice Model Intercomparison Project: Evaluation of sea ice rheology schemes for use in climate simulations. *J. Geophys. Res.*, 105:11299–11320, 2000.

[46] D. Kuzmi. *A Guide to Numerical Methods for Transport Equations*. Friedirch-Alexander Universität Erlangen-Nürnberg, 2010.

[47] D. Kuzmin. Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comp. Phys.*, 228:2517–2534, 2009.

[48] D. Kuzmin and J.Hämäläinen. *Finite Element Methods for Computational Fluid Dynamics*. SIAM, 2015.

[49] R. Kwok, E. C. Hunke, D. Maslowski, and J. Zhang. Variability of sea ice simulations assessed with RGPS kinematics. *J. Geophys. Res.*, 113(C11), 2008.

[50] J. Lang and A. Walter. An adaptive Rothe method for nonlinear reaction-diffusion systems. *Applied Numerical Mathematics*, 13(1):135 – 146, 1993.

[51] J.F. Lemieux. Environment Canada, April 2018. private communication.

[52] J.F. Lemieux, C. Beaudoin, F. Dupont, F Roy, G. Smith, A. Shlyaeva, M Buehner, A. Caya, J. Chen, T. Carrieres, L. Pogson, P. DeRepentigny, A. Plante, P. Pestieau, P. Pellerin, H. Ritchie, G. Garric, and N. Ferry. The Regional Ice Prediction System (RIPS): verification of forecast sea ice concentration. *Quarterly Journal of the Royal Meteorological Society*, 142(695):632–643, 2016.

[53] J.F. Lemieux, D. Knoll, M. Losch, and C. Girard. A second-order accurate in time IMplicit–EXplicit (IMEX) integration scheme for sea ice dynamics. *J. Comp. Phys.*, 263:375–392, 2014.

[54] J.F. Lemieux, D. Knoll, B. Tremblay, D. Holland, and M. Losch. 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. *J. Comp. Phys.*, 231:5926–5944, 2012.

[55] J.F. Lemieux and B. Tremblay. Numerical convergence of viscous-plastic sea ice models. *J. Geophys. Res.*, 114(C5), 2009.

[56] J.F. Lemieux, B. Tremblay, J. Sedláček, P. Tupper, S. Thomas, D. Huard, and J.P. Auclair. Improving the numerical convergence of viscous-plastic sea ice models with the Jacobian-free Newton-Krylov method. *J. Comp. Phys.*, 229:2840–2852, 2010.

[57] J.F. Lemieux, B. Tremblay, S. Thomas, J. Sedlaček, and L.A. Mysak. Using the preconditioned Generalized Minimum RESidual (GMRES) method to solve the sea-ice momentum equation. *J. Geophys. Res.*, 113(C10), 2008.

[58] M. Leppäranta. *The Drift of Sea Ice*. Springer-Verlag Berlin Heidelberg, 2011.

[59] L.Girard, J.Weiss, J.M.Molines, B. Barnier, and S. Bouillon. Evaluation of high-resolution sea ice models on the basis of statistical and scaling properties of arctic sea ice drift and deformation. *Journal of Geophysical Research: Oceans*, 114, 2009.

[60] A. Lichnewsky and R. Temam. Psuedosolutions of the Time-Dependent Minimal Surface Problem. *J. of Differential Equations*, 30:340–364, 1978.

[61] O. Lietaer, T. Fichefet, and V. Legat. The effects of resolving the Canadian Arctic Archipelago in a finite element sea ice model. *Ocean Modelling*, 24:140–152, 2008.

[62] R.W. Lindsay, J. Zhang, and D. A. Rothrock. Sea-ice deformation rates from satellite measurements and in a model. *Atmosphere-Ocean*, 41:35–47, 2003.

[63] M. Losch and S. Danilov. On solving the momentum equations of dynamic sea ice models with implicit solvers and the elastic–viscous–plastic technique. *Ocean Modelling*, 41:42–52, 2012.

[64] M. Losch, A Fuchs, J.F. Lemieux, and A. Vanselow. A parallel Jacobian-free Newton-Krylov solver for a coupled sea ice-ocean model. *J. Comp. Phys.*, 257:901–911, 2014.

[65] M. Luskin and R. Rannacher. On the smoothing property of the Crank-Nicolson scheme. *Applicable Anal.*, 14:117–135, 1982.

[66] S. Mandal, A. Ouazzi, and S. Turek. Modified Newton solver for yield stress fluids. In *Proceedings of ENUMATH 2015, the 11th European Conference on Numerical Mathematics and Advanced Applications*, volume 112 of *Lecture Notes in Computational Science and Engineering*, pages 481–490. Springer, 2016.

[67] F. Massonnet, T. Fichefet, H. Goosse, C. M. Bitz, G. Philippon-Berthier, M. M. Holland, and P.Y. Barriat. Constraining projections of summer Arctic sea ice. *The Cryosphere*, 6:1383–1394, 2012.

[68] C. Mehlmann. Numerische Analyse eines Meereis-Modells mit visko-plastischem Materialgesetz. Master's thesis, Universität Heidelberg, 2015.

[69] C. Mehlmann and T. Richter. A finite element multigrid-framework to solve the sea ice momentum equation. *J. Comp. Phys.*, 348:847–861, 2017.

[70] C. Mehlmann and T. Richter. A modified global Newton solver for viscous-plastic sea ice models. *Ocean Modeling*, 116:96–107, 2017.

[71] D. Meidner and T. Richter. Goal-Oriented Error Estimation for the Fractional Step Theta Scheme. *Comp. Meth. Appl. Math.*, 14:203–230, 2014.

[72] D. Meidner and T. Richter. A Posteriori Error Estimation for the Fractional Step Theta discretization of the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engrg.*, 288:45–59, 2015.

[73] M. Mohammadi-Aragh, F. H. Goessling, M. Losch, N. Hutter, and T. Jung. Predictability of Arctic sea ice on weather time scales. *Scientific Reports*, 8, 2018.

[74] A. Ouazzi. *Finite Element Simulation of Nonlinear Fluids. Application to Granular Material and Powder*. Shaker, 2006. ISBN 3-8322-5201-0.

[75] A. Ouazzi and S. Turek. Numerical methods and simulation techniques for flow with shear and pressure dependent viscosity. In M. Feistauer, V. Dolejsi, P. Knobloch, and K. Najzar, editors, *Numerical Mathematics and Advanced Applications*, pages 668–676. Springer, August 2003. Enumath 2003 Prague; ISBN-Nr. 3-540-21460-7.

[76] C. Paniconi and M. Putti. A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems. *Water Resources Research*, 30:3357–3374, 1994.

[77] D.K. Perovich, B. Light, H. Eicken, K. F. Jones, K. Runciman, and S. V. Nghiem. Increasing solar heating of the Arctic Ocean and adjacent seas, 1979–2005: Attribution and role in the ice-albedo feedback. *Geophysical Research Letters*, 34, 2007.

[78] P.Rampal, S. Bouillon, E. Olason, and M. Morlighem. neXtSIM: a new lagrangian sea ice model. *The Cryosphere*, 10:1055–1073, 2016.

[79] T. Richter. A Monolithic Geometric Multigrid Solver for Fluid-Structure Interactions in ALE formulation. *Int. J. Numer. Meth. Engrg.*, 104:372–390, 2015.

[80] T. Richter. *Fluid-structure Interactions*. Springer International Publishing, 2017.

[81] T. Richter and C. Mehlmann. An accelerated Newton method for nonlinear materials in structure mechanics and fluid mechanics. In F.A. Radu, K.Kumar, I. Berre, J.M Nordbotten, and I.S. Pop, editors, *Numerical Mathematics and Advanced Applications*, pages 345–353. Springer, 2017. Enumath 2017 Bergen.

[82] T. Richter and T. Wick. Variational Localizations of the Dual Weighted Residual Method. *Journal of Computational and Applied Mathematics*, pages 192–208, 2015.

[83] T. Richter and T. Wick. *Einführung in die Numerische Mathematik*. Springer Spektrum, Berlin, Heidelberg, 2017.

[84] D.A. Rothrock. The energetics of the plastic deformation of pack ice by ridging. *J. Geophys. Res.*, 80:4514–4519, 1975.

[85] J.N. Shadid R.S. Tuminaro, H.F. Walker. On backtracking failure in Newton–GMRES methods with a demonstration for the Navier–Stokes equations. *J. Comp. Phys.*, 180:549–558, 2002.

[86] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[87] M. Schmich and R. Rannacher. Goal-oriented space–time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow. *Int. J. Numer. Meth. Fluids*, 70(1):1139–1166, 2012.

[88] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space–time finite element discretizations of parabolic equations. *SIAM Journal on Scientific Computing*, 30(1):369–393, 2008.

[89] J. Sirven and B. Tremblay. Analytical Study of an Isotropic Viscoplastic Sea Ice Model in Idealized Configurations. *J. Phys. Oceanogr.*, 45(2):331–354, 2015.

[90] G. Spreen, R. Kwok, and D. Menemenlis A.T Nguyen. Sea-ice deformation in a coupled ocean-sea-ice model and in satellite remote sensing data. *The Cryosphere*, 11:1553–1573, 2017.

[91] J.C. Stroeve, V. Kattsov, A. Barrett, M. Serreze, T. Pavlova, M. Holland, and W.N. Meier. Trends in Arctic sea ice extent from CMIP5, CMIP3 and observations. *Geophysical Research Letters*, 39(16), 2012.

[92] J.C. Stroeve and D. Notz. Changing state of Arctic sea ice across all seasons. *Environmental Research Letters*, 13(10):103001, 2018.

[93] F.T. Suttmeier. *The dual-weighted-residual method. In: Numerical solution of Variational Inequalities by Adaptive Finite Elements*, chapter 3. Vieweg+Teubner, 2008.

[94] V. Thomée. Galerkin Finite Element Methods for Parabolic Problems. *Springer Series in Computational Mathematics*, 25, 1997.

[95] R. Timmermann, S. Danilov, J. Schröter, C. Boening, D. Sidorenko, and K. Rollenhagen. Ocean circulation and sea ice distribution in a finite element global sea ice ocean model. *Ocean Modelling*, 27:114–129, 2009.

[96] R. Timmermann, S. Danilov, J. Schröter, C. Boening, D. Sidorenko, and K. Rollenhagen. The Finite Element Sea Ice-Ocean Model (FESOM) v.1.4: formulation of an ocean general circulation model. *Geosci. Model Dev.*, 7:663–693, 2014.

[97] S.P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill International Editions, 1970.

[98] B. Tremblay. McGill University, April 2018. private communication.

[99] B. Tremblay and L.A. Mysak. Modeling sea ice as a granular material, including the dilatancy effect. *J. Phys. Oceanogr.*, 27:2342–2360, 1997.

[100] M. Tschudi, C. Fowler, J. Maslanik, and J. Stroeve. Tracking the movement and changing surface characteristics of arctic sea ice. *Numerical Linear Algebra with Applications*, 4:536–540, 2010.

[101] S. Turek. A comparative study of time-stepping techniques for the incompressible Navier-Stokes equations: From fully implicit non-linear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids*, 22(10):987–1011, 1996.

[102] K. Uhlenbeck. Regularity for a class of non-linear elliptic systems. *Acta Math*, 138:219–240, 1977.

[103] M. Vancoppenolle, T. Fichefet, H. Goosse, S. Bouillon, G. Madec, and M.A. Morales Maqueda. *Simulating the mass balance and salinity of Arctic and Antarctic sea ice. 1. Model description and validation*, 2009.

[104] W.F Weeks and W.D. Hibler. *On Sea Ice*. University of Alaska Press, 2010.

[105] P. Wesseling. *An introduction to multigrid methods*. Wiley, Chichester, 1991.

[106] P. Wesseling and C.W. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311 – 334, 2001.

[107] J. Williams and B. Tremblay. The dependence of energy dissipation on spatial resolution in a viscous-plastic sea-ice model. *Ocean Modelling*, 130:40 – 47, 2018.

[108] J. Williams, B. Tremblay, R. Newton, and R. Allard. Dynamic Preconditioning of the Minimum September Sea-Ice Extent. *Journal of Climate*, 29(16):5879–5891, 2016.

[109] N.G. Yakovlev. Coupled model of Ocean General Circulation and sea ice evolution in the Arctic Ocean. *Izvestiya Atmospheric and Oceanic Physic*, 39:355–386, 2003.

[110] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comp. Phy.*, 31:335–362, 1979.

[111] J. Zhang and W.D. Hibler. On an efficient numerical method for modeling sea ice dynamics. *J. Geophys. Res.*, 102:8691–8702, 1991.