Coupled Systems Mechanics, Vol. 1, No. 4 (2012) 325-343
DOI: http://dx.doi.org/10.12989/csm.2012.1.4.325

Operator-splitting methods respecting eigenvalue problems for shallow shelf equations with basal drag

Jürgen Geiser*[1] and Reinhard Calov[2]

[1]*Department of Physics, Ernst-Moritz-Arndt University of Greifswald, Felix-Hausdorff-Str. 6, D-17489 Greifswald, Germany*
[2]*Potsdam Institute for Climate Impact Research, PO Box 60 12 03, D-14412 Potsdam, Germany*

(Received June 25, 2012, Revised November 8, 2012, Accepted November 14, 2012)

Abstract. We present different numerical methods for solving the shallow shelf equations with basal drag (SSAB). An alternative approach of splitting the SSAB equation into a Laplacian and diagonal shift operator is discussed with respect to the underlying eigenvalue problem. First, we solve the equations using standard methods. Then, the coupled equations are decomposed into operators for membranes stresses, basal shear stress and driving stress. Applying reasonable parameter values, we demonstrate that the operator of the membrane stresses is much stiffer than the operator of the basal shear stress. Here, we could apply a new splitting method, which alternates between the iteration on the membrane-stress operator and the basal-shear operator, with a more frequent iteration on the operator of the membrane stresses. We show that this splitting accelerates and stabilize the computational performance of the numerical method, although an appropriate choice of the standard method used to solve for all operators in one step speeds up the scheme as well.

AMS subject classications. 35K25, 35K20, 74S10, 70G65

Keywords: partial differential equations; operator-splitting methods; laplacian operator, diagonal shift operator iterative methods; eigenvalue approach; ice streams; membrane stress; shallow shelf approximation

1. Introduction

The shallow shelf approximation with basal drag (SSAB) finds its application in the simulation of ice streams, which are regions of fast ice flow in ice sheets like Greenland or Antarctica, but also former ice bodies like the Laurentide ice sheet or the Fennoscandinavian ice sheet. Due to rather low but still present basal drag, the speed of such streams is one to two orders of magnitude higher than that of ordinary ice flow.

In this paper, we will discus different numerical solutions of the SSAB equations. They follow from the balance of momentum and the flow law of ice after a number of approximations (scaling, introducing vertically averaged stress, e.g., (Greve and Blatter 2009)). u and v are the x- and y-components of (average) horizontal velocity, respectively. Following (MacAyeal 1995), we have

* Corresponding author, Dr., E-mail: geiser@mathematik.hu-berlin.de

$$\frac{\partial}{\partial x}\left(2\nu H\left(2\frac{\partial u}{\partial x}+\frac{\partial \upsilon}{\partial y}\right)\right)+\frac{\partial}{\partial y}\left(\nu H\left(\frac{\partial u}{\partial y}+\frac{\partial \upsilon}{\partial x}\right)\right)-\rho g H\frac{\partial h}{\partial x}-\tau^{x(b)}=0 \tag{1.1}$$

$$\frac{\partial}{\partial y}\left(2\nu H\left(2\frac{\partial \upsilon}{\partial y}+\frac{\partial u}{\partial x}\right)\right)+\frac{\partial}{\partial x}\left(\nu H\left(\frac{\partial u}{\partial y}+\frac{\partial \upsilon}{\partial x}\right)\right)-\rho g H\frac{\partial h}{\partial y}-\tau^{y(b)}=0 \tag{1.2}$$

where g, ρ, H and h are the **gravitational acceleration**, density of ice, ice thickness and ice surface elevation, respectively. In physical terms, the above equations balance membrane stresses (longitudinal stresses and lateral shear stress) with driving stress resulting from the Earth's acceleration and the basal shear stress. For the basal shear stress, we assume the simple expression

$$\tau^{x(b)}=\frac{c_b u}{(u^2+\upsilon^2)^{p/(2(p+1))}} \tag{1.3}$$

$$\tau^{y(b)}=\frac{c_b v}{(u^2+\upsilon^2)^{p/(2(p+1))}} \tag{1.4}$$

with the basal shear parameter c_b and the power parameter p. In practice, different values for p apply. Pollard and DeConto (2007) use $p=1.25$. For $p=0$ Eq. (1.4) reduces to a linear relation between velocity and basal shear stress.

The effective viscosity reads

$$\nu=\frac{\bar{B}}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial \upsilon}{\partial y}\right)^2+\frac{1}{4}\left(\frac{\partial u}{\partial y}+\frac{\partial \upsilon}{\partial x}\right)^2+\frac{\partial \upsilon}{\partial x}\frac{\partial \upsilon}{\partial y}\right]^{(1-n)/(2n)}$$

with $n=3$ and

$$\bar{B}=\int_b^h B\,dz$$

where b denotes the elevation of the base of the ice stream.

For our numerical schemes, we assume in our first approach that the average inverse rate factor $\bar{B}$ is constant. In general, B can depend on temperature and water content in the ice and is calculated via separate equations.

The outline of the paper is as follows. The mathematical methods are introduced in Section 2. The solver schemes are discussed in Section 3. In Section 4, we discuss the algorithm and the assembling of the splitting schemes. In Section 5, we validate our basic scheme, compute the condition of our split operators and determine the performance of the different schemes. Finally, we draw our conclusions and discuss the results including an outlook on future work.

2. Mathematical method

In the following, we present the iterative splitting methods for elliptic and parabolic operators, see Geiser (2011). To optimize the application of iterative splitting methods, we have taken into account the eigenvalues and conditions of matrices, which are used to control the iterative steps.

2.1 Splitting of elliptic operators

In the following, we discuss the standard iterative splitting with respect to one operator

$$Au_i + Bu_{i-1} = 0,\ i = 1,2,...,m \tag{2.1}$$

Iterative splitting with respect to alternating operators

$$Au_i + Bu_{i-1} = 0,\ i = 1,2,...,j \tag{2.2}$$

$$Au_{i-1} + Bu_i = 0,\ i = j+1, j+2,...,m \tag{2.3}$$

For Eq. (2.1) and Eqs. (2.2) and (2.3), $u_0(x) = u_{\text{initial}}$ is an initial starting solution for $u_0 = 0$ and the initial conditions and boundary conditions are given by $u_i(x) = u_{\text{init}}(x)$, for $x \in \Omega_{\text{init}}$ and $u_i(x) = u_{\text{bound}}(\text{x})$, for $x \in \partial\Omega$ for all $i = 0,1,...,m$. Here, Ω_{init} is the domain of the initial condition and $\partial\Omega$ the boundary of Ω.

2.2 Iterative operator-splitting method as fixed-point scheme

In this section, we examine parabolic equations and assume that the elliptic equations are only a special parabolic case $t \to \infty$, see Barbu (1998). The results can be achieved for the elliptic part with the assumption that the solutions are embedded into the parabolic part.

The iterative operator-splitting method is used as a fixed-point scheme to linearize the nonlinear operators (Geiser 2011, Kanney *et al*. 2003).

We consider nonlinear differential equations of the form

$$\frac{du}{dt} = A(u(t))\ u(t) + B(u(t))\ u(t), \text{ with } u(t^n) = u^n \tag{2.4}$$

where $A(u)$, $B(u)$ are matrices with nonlinear entries and densely defined, where we assume that the entries involve the spatial derivatives of u (Zeidler 1990). In the following, we discuss the standard iterative operator-splitting method as a fixed-point iteration method to linearize the operators.

We split our nonlinear differential Eq. (2.4) by applying

$$\frac{du_i(t)}{dt} = A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } u_i(t^n) = u^n \tag{2.5}$$

$$\frac{du_{i+1}(t)}{dt} = A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i+1}(t), \text{ with } u_{i+1}(t^n) = u^n \tag{2.6}$$

where the time-step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1,3,...,2m+1$. $u_0(t) = u_n$ is the starting solution, where we assume that the solution u^{n+1} is near u^n, or $u_0(t) = 0$. Therefore, we have to solve the local fixed-point problem. u^n is the known split approximation at time-level $\text{t} = t^n$.

The split approximation at time-level $t = t^{n+1}$ is defined as $u^{n+1} = u_{2m+2}(t^{n+1})$. We assume that the operators $A(u_{i-1}(t^{n+1}))$, $\text{B}(u_{i-1}(t^{n+1}))$ are constant for $i = 1,3,...,2m+1$. Here the linearization is done with respect to the iterations, such that $A(u_{i-1})$, $\text{B}(u_{i-1})$ are at least non-dependent operators in the iterative equations, and we can apply the linear theory. For the linearization we assume at least in the

first equation $A(u_{i-1}(t)) \approx A(u_i(t))$, and in the second equation $B(u_{i-1}(t)) \approx B(u_{i+1}(t))$, where $t = t^n + \tau$, for small τ.
We have

$$\|A(u_{i-1}(t^{n+1}))u_i(t^{n+1}) - A(u^{n+1})u(t^{n+1})\| \leq \varepsilon$$

for sufficient iterations $i \in \{1,3,...,2m+1\}$.

The linearization with the fixed-point scheme can be used for smooth or weak nonlinear operators, otherwise convergence is lost, because we did not converge to the local fixed point (Kanney *et al*. 2003).

2.3 Decoupling ideas based on eigenvalue problems

To distinguish the operators in the differential equation as stiff or non-stiff, we can determine the eigenvalues of each operator and use them as reciprocal time scales.

The operator equations are analyzed with the eigenvalue problem

$$(A + B)u(t) = (\lambda_A + \lambda_B)u(t) \tag{2.7}$$

where the operators A and B result from the spatial discretization and λ_A and λ_B are eigenvalues.

2.4 Stiffness and non-stiffness of elliptic operators

Here, we consider the stiffness of operator A compared to operator B. An operator is stiff if its matrix is huge in the condition, see Hackbusch (1986).

The maximal and minimal eigenvalues are determined by

$$\lambda_{\max} = \|A\|_2,\ \lambda_{\min} = 1/\|\mathrm{A}^{-1}\|_2 \tag{2.8}$$

$$\chi_A = \lambda_{\max} / \lambda_{\min} \tag{2.9}$$

where χ_A is the condition of matrix A. For the matrix B analogous equations with the condition χ_B of matrix B apply.

For the condition numbers of the matrices, we have

$$\chi_A >> 1,\ \chi_B \approx 1 \tag{2.10}$$

where the eigenvalues of A vary strongly and the eigenvalue of B vary weakly, i.e., the respective operators are regarded as stiff and non-stiff.

3. Applied iterative methods

In the following, we present the used linear iterative methods, which are adapted and programmed in an own novel library in FORTRAN90 code. Our code, which is written modular, can be imple-

mented like standard libraries as EISPACK (see LIS11. LIS (2011)), LAPACK (see Smith *et al*. (1976)) or LIS (see LAPACK92. LAPACK (1992)). For the computation of eigenvalues of matrices, we used the free software OCTAVE (http://www.gnu.org/software/octave/).

Here, we restricted us to standard iterative schemes, as Jacobi, Gauss-Seidel and SOR method, see Thomas (1999), and take into account the control of these iterative scheme via the eigenvalues of the underlying operators, see Geiser (2008, 2009, 2011).

Competitive methods, as multigrid (Hackbusch 1985, Trottenberg *et al*. 2001) and multi-level domain (Smith *et al*. 1999, Toselli and Widlund 2005), also Krylov methods (Hestenes and Stiefel 1952, Saad 2003) are also very attractive. But the applicability of these schemes in the framework of operator splitting are planned to be studied in our future work.

Due to their advantages as scale-independent solvers, while respecting different grids, we also have in mind the problems in their implementations to our underlying FORTRAN90 code (done with finite difference schemes):

· Arranging different hierarchical grids
· Time-consuming assembling of sub-routines for grids

while improvement are also done, e.g., for adaptive finite elements, see Storti *et al*. (2001). Such discussions, we would like to shift into a next paper when we will investigate multigrid and Krylov schemes.

Moreover, we have taken into account the special characteristics of the SSAB equations, see MacAyeal (1995), and adapted the standard successive over-relaxation (SOR) as one possible submethod in our splitting approach. We have included the following characteristics of the SSAB equation to optimize the splitting method:

· Decoupling into linearized and nonlinearized parts (Laplacian and diagonal shifted part).
· Perturbed part (nonlinearized part) is non-stiff, the non-perturbed part (linearized part) is stiff and solved as main operator in the splitting schemes.
· Eigenvalues are used to quantify the stiffness of the operator.

In the next section, we apply our theoretical results to the shallow shelf approximation and propose a decomposition of the underlying operators.

4. Algorithm and assembling of the splitting method with respect to the eigenvalues of the operators

In the following, we discuss the adapted algorithms and their implementation in our modular FORTRAN90 programming code with respect to our SSAB equation, see MacAyeal (1995).

We start with introducing the operators relevant to the SSAB equations.

Linearized Part (membrane stresses)

$$M(\nu) = \begin{pmatrix} 4\frac{\partial}{\partial x}\nu H\frac{\partial}{\partial x} & 2\frac{\partial}{\partial x}\nu H\frac{\partial}{\partial y} \\ 2\frac{\partial}{\partial y}\nu H\frac{\partial}{\partial x} & 4\frac{\partial}{\partial y}\nu H\frac{\partial}{\partial y} \end{pmatrix} \tag{4.1}$$

The operator containing the driving stress

$$S(h) = -\begin{pmatrix} \rho g H \frac{\partial}{\partial x} h \\ \rho g H \frac{\partial}{\partial y} h \end{pmatrix} \tag{4.2}$$

The nonlinear part

$$B(\mu) = -\mu(u, v)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.3}$$

with

$$\mu(u, v) = \frac{c_b}{(u^2 + v^2)^{p/(2(p+1))}} \tag{4.4}$$

which is related to the basal shear stress as

$$B(\mu)(u, v)^t = \begin{pmatrix} \tau^{x(b)} \\ \tau^{y(b)} \end{pmatrix} \tag{4.5}$$

4.1 One-side iterative schemes

The operators are split into the following schemes, where both operators M and B are solved together in one iterative step

$$M(\nu_{i-1})\,(u_i, v_i)^t + S(h) + B(\mu_{i-1})\,(u_i, v_i)^t = 0 \tag{4.6}$$

or with explicitly expressed operators

$$\frac{\partial}{\partial x}\left(2\nu_{i-1}H\left(2\frac{\partial u_i}{\partial x} + \frac{\partial v_i}{\partial y}\right)\right) + \frac{\partial}{\partial y}\left(\nu_{i-1}H\left(\frac{\partial u_i}{\partial y} + \frac{\partial v_i}{\partial x}\right)\right) - \rho g H\frac{\partial h}{\partial x} - \mu_{i-1}u_i = 0 \tag{4.7}$$

$$\frac{\partial}{\partial y}\left(2\nu_{i-1}H\left(2\frac{\partial u_i}{\partial y} + \frac{\partial u_i}{\partial x}\right)\right) + \frac{\partial}{\partial x}\left(\nu_{i-1}H\left(\frac{\partial u_i}{\partial y} + \frac{\partial u_i}{\partial x}\right)\right) - \rho g H\frac{\partial h}{\partial y} - \mu_{i-1}v_i = 0 \tag{4.8}$$

where ν_{i-1} is given with μ_{i-1} for the previous solution.

Here, Eqs. (4.7) and (4.8) are discretized using centred differences in the Arakawa C grid (Arakawa 1972) and, following the standard procedure, are rewritten by collecting the coefficients of the discrete velocities

$$\begin{aligned}&\beta^0_{l,m}u_{l+1/2,m}+\beta^1_{l,m}u_{l+3/2,m}+\beta^2_{l,m}u_{l-1/2,m}+\beta^3_{l,m}u_{l+1/2,m+1}\\&+\beta^4_{l,m}u_{l+1/2,m-1}+\beta'^5_{l,m}v_{l,m+1/2}+\beta'^6_{l,m}v_{l,m-1/2}+\beta'^7_{l,m}v_{l+1,m+1/2}\\&+\beta'^8_{l,m}v_{l+1,m-1/2}=f_{l,m}\end{aligned}\tag{4.9}$$

$$\begin{aligned}&\beta'^0_{l,m}v_{l,m+1/2}+\beta'^1_{l,m}v_{l+1,m+1/2}+\beta'^2_{l,m}v_{l-1,m+1/2}+\beta'^3_{l,m}v_{l,m+3/2}\\&+\beta'^4_{l,m}v_{l,m-1/2}+\beta^5_{l,m}u_{l-1/2,m+1}+\beta^6_{l,m}u_{l-1/2,m}+\beta^7_{l,m}u_{l+1/2,m+1}\\&+\beta^8_{l,m}u_{l+1/2,m}=f'_{l,m}\end{aligned}\tag{4.10}$$

where the integers l, m refer to the two horizontal coordinates.

The discretized effective viscosity together with the velocities which appear therein are taken from the previous solution at step $i-1$. It reads

$$\begin{aligned}\nu_{l,m}=\frac{\overline{B}}{2}\Bigg\{&\left(\frac{u_{l+1/2,m}-u_{l-1/2,m}}{\Delta x_{l,m}}\right)^2+\left(\frac{v_{l,m+1/2}-v_{l,m-1/2}}{\Delta y_{l,m}}\right)^2\\&+\frac{1}{4}\left(\frac{u_{l-1/2,m+1/2}-u_{l-1/2,m-1/2}}{\Delta y_{l,m}}+\frac{v_{l+1/2,m-1/2}-v_{l-1/2,m-1/2}}{\Delta x_{l,m}}\right)^2\\&+\frac{u_{l+1/2,m}-u_{l-1/2,m}}{\Delta x_{l,m}}\frac{v_{l,m+1/2}-v_{l,m-1/2}}{\Delta v_{l,m+1/2}}+\epsilon_\nu^2\Bigg\}^{(1-n)/(2n)}\end{aligned}\tag{4.11}$$

As in Bueler and Brown (2009) we introduce a regularization as $\varepsilon_\nu = 3\cdot 10^{-14}\mathrm{s}^{-1}$.

Using these Eqs. (4.9) and (4.10) as a basis, we will construct the matrix corresponding to the operator $A := M + B$ in Eq. (4.6) of the linear system $A\hat{u} = f$. The right-hand side column vector f is the discretized representation of the two-column vector $-S$ in Eq. (4.6).

We order the velocities and the right-hand side column vector f as

$$\hat{u} = (v_{1,1.5},\, u_{1.5,1},\, v_{1,2.5},\, \ldots,\, v_{l_{\max}-1},\, m_{\max-0.5},\, u_{l_{\max}-0.5},\, m_{\max-1})^t \tag{4.12}$$

$$f = (f'_{1.1}, f_{1.1}, f'_{1.2}, \cdots, f'_{l_{\max}-1,\, m_{\max}-1}, f_{l_{\max}-1,\, m_{\max}-1})^t \tag{4.13}$$

and find the matrix

$$A = \begin{pmatrix} D & R & & & \\ L & D & R & & \\ & L & D & R & \\ & & \ddots & & \\ & & & L & D \end{pmatrix} \tag{4.14}$$

where the elements L, D and R denote

$$D = \begin{pmatrix} \beta'^{0}_{l,1} & \beta^{8}_{l,1} & \beta'^{3}_{l,1} & \beta^{7}_{l,1} & & \\ \beta'^{5}_{l,1} & \beta^{0}_{l,1} & \cdot & \beta^{3}_{l,1} & \cdot & \\ \beta'^{4}_{l,2} & \cdot & \beta'^{0}_{l,2} & \beta^{8}_{l,2} & \beta'^{3}_{l,2} & \beta^{7}_{l,2} \\ \beta^{6}_{l,2} & \beta^{4}_{l,2} & \beta'^{5}_{l,2} & \beta^{0}_{l,2} & \cdot & \beta^{3}_{l,2} \\ \cdot & \cdot & \beta'^{4}_{l,3} & \cdot & \beta'^{0}_{l,3} & \beta^{8}_{l,3} \\ & & \beta^{6}_{l,3} & \beta^{4}_{l,3} & \beta'^{5}_{l,3} & \beta^{0}_{l,3} \\ & & & & \ddots & \end{pmatrix}$$

$$R = \begin{pmatrix} \beta'^{1}_{l,1} & & & \\ \beta'^{7}_{l,1} & \beta^{1}_{l,1} & & \\ & \cdot & \beta'^{1}_{l,2} & \\ \beta'^{8}_{l,2} & & \beta'^{7}_{l,2} & \beta^{1}_{l,2} \\ & \cdot & & \cdot \\ & & & \ddots \end{pmatrix}, \quad L = \begin{pmatrix} \beta'^{2}_{l,1} & \beta^{6}_{l,1} & \cdot & \beta^{5}_{l,1} \\ & \beta^{2}_{l,1} & & \\ & & \beta'^{2}_{l,2} & \beta^{6}_{l,2} \\ & & & \ddots \end{pmatrix}$$

The coefficients of the velocities are given as

$$\beta^{0}_{l,m} = -\frac{4\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta x_{l+1,m}} - \frac{4\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta x_{l,m}} - \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m-1/2}} - \mu_{l+1/2,m} \tag{4.15}$$

$$\beta^{1}_{l,m} = \frac{4\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta x_{l+1,m}} \tag{4.16}$$

$$\beta^{2}_{l,m} = \frac{4\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta x_{l,m}} \tag{4.17}$$

$$\beta^{3}_{l,m} = \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m+1/2}} \tag{4.18}$$

$$\beta^{4}_{l,m} = \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m-1/2}} \tag{4.19}$$

$$\beta'^{5}_{l,m} = -\frac{2\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta y_{l,m}} - \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta x_{l+1/2,m+1/2}} \tag{4.20}$$

$$\beta'^{6}_{l,m} = \frac{2\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta y_{l,m}} + \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta x_{l+1/2,m-1/2}} \tag{4.21}$$

$$\beta'^{7}_{l,m} = \frac{2\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta y_{l+1,m}} + \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta x_{l+1/2,m+1/2}} \tag{4.22}$$

$$\beta'^{8}_{l,m} = -\frac{2\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta y_{l+1,m}} - \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta x_{l+1/2,m-1/2}} \tag{4.23}$$

and

$$\beta'^{0}_{l,m} = -\frac{4\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta y_{l,m+1}} - \frac{4\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta y_{l,m}}$$
$$-\frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l-1/2,m+1/2}} - \mu_{l,m+1/2} \tag{4.24}$$

$$\beta'^{1}_{l,m} = \frac{4\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta y_{l,m+1}} \tag{4.25}$$

$$\beta'^{2}_{l,m} = \frac{4\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta y_{l,m}} \tag{4.26}$$

$$\beta'^{3}_{l,m} = \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l+1/2,m+1/2}} \tag{4.27}$$

$$\beta'^{4}_{l,m} = \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l-1/2,m+1/2}} \tag{4.28}$$

$$\beta^{5}_{l,m} = -\frac{2\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta x_{l,m+1}} - \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta y_{l-1/2,m+1/2}} \tag{4.29}$$

$$\beta^{6}_{l,m} = \frac{2\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta x_{l,m}} + \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta y_{l-1/2,m+1/2}} \tag{4.30}$$

$$\beta^{7}_{l,m} = \frac{2\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta x_{l,m+1}} + \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta y_{l+1/2,m+1/2}} \tag{4.31}$$

$$\beta^{8}_{l,m} = -\frac{2\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta x_{l,m}} + \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta y_{l+1/2,m+1/2}} \tag{4.32}$$

with the abbreviation

$$\hat{\nu}_{\tilde{l}\tilde{m}} = \nu_{\tilde{l}\tilde{m}} H_{\tilde{l}\tilde{m}}$$

for the indices

$$\tilde{l} = l-1/2, l, l+1/2, l+1, \tilde{m} = m-1/2, m, m+1/2, m+1$$

Further, the components of the right-hand side column vector read

$$f_{l,m} = \rho g H_{l+1/2,m}\left(\frac{h_{l+1,m}-h_{l,m}}{\Delta x_{l+1/2,m}}\right) \tag{4.33}$$

$$f'_{l,m} = {}_{\rho g}H_{l,m+1/2}\left(\frac{h_{l,m+1}-h_{l,m}}{\Delta y_{l,m+1/2}}\right) \tag{4.34}$$

The first method we use to solve the system is the weighted Jacobi method as described in Thomas (1999). Applied to our problem, we obtain the component-wise iteration steps for u and v

$$\begin{aligned} v^i_{l,m+1/2} = (1-\omega)v^{i-1}_{l,m+1/2} + \frac{\omega}{\beta'^0_{l,m}}[f'_{l,m} - \beta'^1_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^2_{l,m}v^{i-1}_{l-1,m+1/2} \\ -\beta'^3_{l,m}v^{i-1}_{l,m+3/2} - \beta'^4_{l,m}v^{i-1}_{l,m-1/2} - \beta^5_{l,m}u^{i-1}_{l-1/2,m+1} - \beta^6_{l,m}u^{i-1}_{l-1/2,m} \\ -\beta^7_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^8_{l,m}u^{i-1}_{l+1/2,m}] \end{aligned} \tag{4.35}$$

$$\begin{aligned} u^i_{l+1/2,m} = (1-\omega)u^{i-1}_{l+1/2,m} + \frac{\omega}{\beta^0_{l,m}}[f_{l,m} - \beta^1_{l,m}u^{i-1}_{l+3/2,m} - \beta^2_{l,m}u^{i-1}_{l-1/2,m} \\ -\beta^3_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^4_{l,m}u^{i-1}_{l+1/2,m-1} - \beta'^5_{l,m}v^{i-1}_{l,m+1/2} - \beta'^6_{l,m}v^{i-1}_{l,m-1/2} \\ -\beta'^7_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^8_{l,m}v^{i-1}_{l+1,m-1/2}] \end{aligned} \tag{4.36}$$

with $m = 1,...,m_{\max} - 1$, $l = 1,...,l_{\max} - 1$.

An appropriate value for the weight ω is found via testing (section 5.1).

For the second method, we apply the SOR algorithm Thomas (1999) to solve the system and construct the following algorithm

$$\begin{aligned} v^i_{l,m+1/2} = (1-\omega)v^{i-1}_{l,m+1/2} + \frac{\omega}{\beta'^0_{l,m}}[f'_{l,m} - \beta'^1_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^2_{l,m}v^{i}_{l-1,m+1/2} \\ -\beta'^3_{l,m}v^{i-1}_{l,m+3/2} - \beta'^4_{l,m}v^{i}_{l,m-1/2} - \beta^5_{l,m}u^{i-1}_{l-1/2,m+1} - \beta^6_{l,m}u^{i-1}_{l-1/2,m} \\ -\beta^7_{l,m}u^{i}_{l+1/2,m+1} - \beta^8_{l,m}u^{i}_{l+1/2,m}] \end{aligned} \tag{4.37}$$

$$\begin{aligned} u^i_{l+1/2,m} = (1-\omega)u^{i-1}_{l+1/2,m} + \frac{\omega}{\beta^0_{l,m}}[f_{l,m} - \beta^1_{l,m}u^{i-1}_{l+3/2,m} - \beta^2_{l,m}u^{i}_{l-1/2,m} \\ -\beta^3_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^4_{l,m}u^{i}_{l+1/2,m-1} - \beta'^5_{l,m}v^{i-1}_{l,m+1/2} - \beta'^6_{l,m}v^{i-1}_{l,m-1/2} \\ -\beta'^7_{l,m}v^{i}_{l+1,m+1/2} - \beta'^8_{l,m}v^{i}_{l+1,m-1/2}] \end{aligned} \tag{4.38}$$

with $m = 1,...,m_{\max} - 1$, $l = 1,...,l_{\max} - 1$.

For both of the methods, the stop criterion is given as

$$\max_{l,m}(|u_{l+1/2,m,i+1} - u_{l+1/2,m,i}|, |v_{l+1/2,m,i+1} - v_{l+1/2,m,i}|) \le err \tag{4.39}$$

for all l, m, where err is a given bound. Later, we will provide a concrete value for err.

We will use both methods, Jacobi and SOR, with appropriate modifications in our splitting scheme described in the next section.

4.2 Two-side iterative schemes

Two-side iterative schemes have the benefit of computationally balancing the different operators

such that we obtain a predictor-corrector scheme to improve a previous solution.

We start with $i = 1$ and $u_0 = v_0 = 0$.

The steps are given for $i > 1$:

First step (respecting M)

$$M(v_{i-1})\ (u_i,\ v_i)^t + S(h) + B(\mu_{i-1})\ (u_{i-1},\ v_{i-1})^t = 0 \tag{4.40}$$

where ν_{i-1} is given with μ_{i-1} for the previous solution.

Second step (respecting B)

$$M(\nu_i)\ (u_i,\ v_i)^t + S(h) + B(\mu_i)\ (u_{i+1},\ v_{i+1})^t = 0 \tag{4.41}$$

where ν_i is given with μ_i in the last step (i-th iterative step). And we obtain the next solution u_{i+1}, v_{i+1} which is used to iterate the solution of u_{i+2}, v_{i+2}

In terms of computational time, it makes sense that we have one dominant equation, here Eq. (4.40), such that the main part of the iterative steps are taken for this equation. Alternation between each iterative scheme is expensive and we should reduce it to a minimum, e.g., 10 iterative steps with Eq. (4.40), while only one iterative step with Eq. (4.41). This is because the spectrum of the operators is not the same in our case (see Section 5.2). Therefore, it makes sense to employ more iteration steps over the larger spectrum (means stiffer part), see Hackbusch (1985).

As before, we substitute the expressions for the operators into their equations (here Eqs. (4.40) and (4.41)) and develop the algorithm.

We start with the splitting step in M. As for the one-side scheme (Eq. (4.6)), one can iterate Eq. (4.40) using the weighted Jacobi scheme or, alternatively, utilizing the SOR scheme; but the term $B(\mu_{i-1})\ (u_{i-1},\ v_{i-1})^t$ remains explicit in this splitting step.

Redefinition of Eqs. (4.15) and (4.24) yield

$$\begin{aligned}\beta^0_{l,m} &= -\frac{4\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta x_{l+1,m}} - \frac{4\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta x_{l,m}} \\ &-\frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m-1/2}}\end{aligned} \tag{4.42}$$

$$\begin{aligned}\beta'^0_{l,m} &= -\frac{4\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta y_{l,m+1}} - \frac{4\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta y_{l,m}} \\ &-\frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l-1/2,m+1/2}}\end{aligned} \tag{4.43}$$

Additionally, we introduce the new coefficients

$$\beta^9_{l,m} = -\mu_{l+1/2,m},\ \beta'^9_{l,m} = -\mu_{l,m+1/2} \tag{4.44}$$

For the splitting step in M, we use the SOR algorithm, which is proved to be faster than the Jacobi procedure.

$$v^{i}_{l,m+1/2} = (1-\omega_M)v^{i-1}_{l,m+1/2} + \frac{\omega_M}{\beta'^{0}_{l,m}}[f_{l,m} - \beta'^{1}_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^{2}_{l,m}v^{i}_{l-1,m+1/2}$$
$$-\beta'^{3}_{l,m}v^{i-1}_{l,m+3/2} - \beta'^{4}_{l,m}v^{i}_{l,m-1/2} - \beta^{5}_{l,m}u^{i-1}_{l-1/2,m+1} - \beta^{6}_{l,m}u^{i-1}_{l-1/2,m}$$
$$-\beta^{7}_{l,m}u^{i}_{l+1/2,m+1} - \beta^{8}_{l,m}u^{i}_{l+1/2,m} - \beta'^{9}_{l,m}v^{i-1}_{l,m+1/2}] \tag{4.45}$$

$$u^{i}_{l+1/2,m} = (1-\omega_M)u^{i-1}_{l+1/2,m} + \frac{\omega_M}{\beta^{0}_{l,m}}[f_{l,m} - \beta^{1}_{l,m}u^{i-1}_{l+3/2,m} - \beta^{2}_{l,m}u^{i}_{l-1/2,m}$$
$$-\beta^{3}_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^{4}_{l,m}u^{i}_{l+1/2,m-1} - \beta'^{5}_{l,m}v^{i-1}_{l,m+1/2} - \beta'^{6}_{l,m}v^{i-1}_{l,m-1/2}$$
$$-\beta'^{7}_{l,m}v^{i}_{l+1,m+1/2} - \beta'^{8}_{l,m}v^{i}_{l+1,m-1/2} - \beta^{9}_{l,m}u^{i-1}_{l+1/2,m}] \tag{4.46}$$

with $m = 1,...,m_{\max} - 1$, $l = 1,...,l_{\max} - 1$.

Together with the coefficients $\beta^{9}_{l,m}$ and $\beta'^{9}_{l,m}$, the central velocities $u_{l+1/2,m}$ and $v_{l,m+1/2}$ are additional terms in Eqs. (4.45) and (4.46) compared to Eqs. (4.37) and (4.38) and by the definition of the splitting step in M they are taken from the previous iteration step $i - 1$.

The splitting step in B can be expressed straightforwardly by solving Eq. (4.41)

$$u_{l+1/2,m,i+1} = \frac{1}{\mu_{l,m+1/2,i}}\left[\frac{\partial}{\partial x}\left(2\,\nu_i H\left(2\frac{\partial u_i}{\partial x} + \frac{\partial v_i}{\partial y}\right)\right)\right.$$
$$\left. + \frac{\partial}{\partial y}\left(\nu_i H\left(\frac{\partial u_i}{\partial y} + \frac{\partial v_i}{\partial x}\right)\right) - \rho g H\frac{\partial h}{\partial x}\right] \tag{4.47}$$

$$v_{l,m+1/2,i+1} = \frac{1}{\mu_{l,m+1/2,i}}\left[\frac{\partial}{\partial y}\left(2\,\nu_i H\left(2\frac{\partial v_i}{\partial y} + \frac{\partial u_i}{\partial x}\right)\right)\right.$$
$$\left. + \frac{\partial}{\partial x}\left(\nu_i H\left(\frac{\partial u_i}{\partial y} + \frac{\partial v_i}{\partial x}\right)\right) - \rho g H\frac{\partial h}{\partial y}\right] \tag{4.48}$$

Eqs. (4.47) and (4.48) describe an application of the ordinary Jacobi scheme, which is the weighted Jacobi scheme with $\omega = 1$. In principle, weighting of the velocities at the iteration step i and at the iteration step i+1 by Eqs. (4.47) and (4.48) in discretized form would lead to a solution with the weighted Jacobi scheme. Here, we will express the weighted Jacobi algorithm applied to the splitting in B in the framework of the coefficients of the velocities, because Eqs. (4.9) and (4.10) are already discretized.

Again we redefine the coefficients $\beta^{0}_{l,m}$ and $\beta'^{0}_{l,m}$ in Eqs. (4.15) and (4.24)

$$\beta^{0}_{l,m} = -\mu_{l+1/2,m},\ \beta'^{0}_{l,m} = -\mu_{l,m+1/2} \tag{4.49}$$

and we redefine the coefficients $\beta^{9}_{l,m}$ and $\beta'^{9}_{l,m}$ from Eqs. (4.44) and find

$$\beta^{9}_{l,m} = -\frac{4\hat{\nu}_{l+1,m}}{\Delta x_{l+1/2,m}\Delta x_{l+1,m}} - \frac{4\hat{\nu}_{l,m}}{\Delta x_{l+1/2,m}\Delta x_{l,m}}$$
$$-\frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l+1/2,m-1/2}}{\Delta y_{l+1/2,m}\Delta y_{l+1/2,m-1/2}} \tag{4.50}$$

$$\beta'^{9}_{l,m} = -\frac{4\hat{\nu}_{l,m+1}}{\Delta y_{l,m+1/2}\Delta y_{l,m+1}} - \frac{4\hat{\nu}_{l,m}}{\Delta y_{l,m+1/2}\Delta y_{l,m}} - \frac{\hat{\nu}_{l+1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l+1/2,m+1/2}} - \frac{\hat{\nu}_{l-1/2,m+1/2}}{\Delta x_{l,m+1/2}\Delta x_{l-1/2,m+1/2}} \tag{4.51}$$

One may recognize that the redefinitions expressed by Eqs. (4.49) to (4.51) are complementary to the redefinitions and new-definitions in Eqs. (4.42) to (4.44).

Reformulation of the Eqs. (4.35) and (4.36) leads to a weighted Jacobi algorithm with the additional terms in $\beta^{9}_{l,m}$ and $\beta'^{9}_{l,m}$

$$\begin{aligned} v^{i}_{l,m+1/2} &= (1-\omega_B)v^{i-1}_{l,m+1/2} + \frac{\omega_B}{\beta'^{0}_{l,m}}[f'_{l,m} - \beta'^{1}_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^{2}_{l,m}v^{i-1}_{l-1,m+1/2} \\ &\quad -\beta'^{3}_{l,m}v^{i-1}_{l,m+3/2} - \beta'^{4}_{l,m}v^{i-1}_{l,m-1/2} - \beta^{5}_{l,m}u^{i-1}_{l-1/2,m+1} - \beta^{6}_{l,m}u^{i-1}_{l-1/2,m} \\ &\quad -\beta^{7}_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^{8}_{l,m}u^{i-1}_{l+1/2,m} - \beta'^{9}_{l,m}v^{i-1}_{l,m+1/2}] \\ u^{i}_{l+1/2,m} &= (1-\omega_B)u^{i-1}_{l+1/2,m} + \frac{\omega_B}{\beta^{0}_{l,m}}[f_{l,m} - \beta^{1}_{l,m}u^{i-1}_{l+3/2,m} - \beta^{2}_{l,m}u^{i-1}_{l-1/2,m} \\ &\quad -\beta^{3}_{l,m}u^{i-1}_{l+1/2,m+1} - \beta^{4}_{l,m}u^{i-1}_{l+1/2,m-1} - \beta'^{5}_{l,m}v^{i-1}_{l,m+1/2} - \beta'^{6}_{l,m}v^{i-1}_{l,m-1/2} \\ &\quad -\beta'^{7}_{l,m}v^{i-1}_{l+1,m+1/2} - \beta'^{8}_{l,m}v^{i-1}_{l+1,m-1/2} - \beta^{9}_{l,m}u^{i-1}_{l+1/2,m}] \end{aligned}$$

with $m = 1,...,m_{\max} - 1$, $l = 1,...,l_{\max} - 1$.

Again, we apply the stop criterion according to Eq. (4.39), but now after the two splitting steps have been evaluated.

5. Numerical experiments

In the following, we introduce our numerical experiments. The computations were performed on the IBM iDataPlex Cluster (IPLEX).

5.1 Validation of the schemes with the MacAyeal schematic setup

In most of literature regarding ice streams (e.g., Joughin *et al*. 2001, MacAyeal 1995), the SSAB equations are used for inverse modeling of a basal friction parameter from the measured ice surface velocity. Without knowledge of the basal friction, a validation of our model via measured velocities would be difficult. Here, we will not perform a strong model validation in the sense that we assess the modeling errors numerically. We rather make use of the schematic setup for the ice topography and basal friction (Fig. 1) by MacAyeal (1993), a tutorial on control methods. As a byproduct, MacAyeal (1993) computes the velocity fields and we use the plots of these velocities for a comparison with our modeled velocities. At this stage of development, it is not our intention to perform a complete model validation and/or verification as described in Bueler *et al*. (2005). Such an approach would imply the knowledge of an exact solution of the two dimensional SSAB equations, which is not available yet.

Instead of the non-linear relation for the basal shear stress in Eq. (1.4) (formally $p = 0$ therein) the

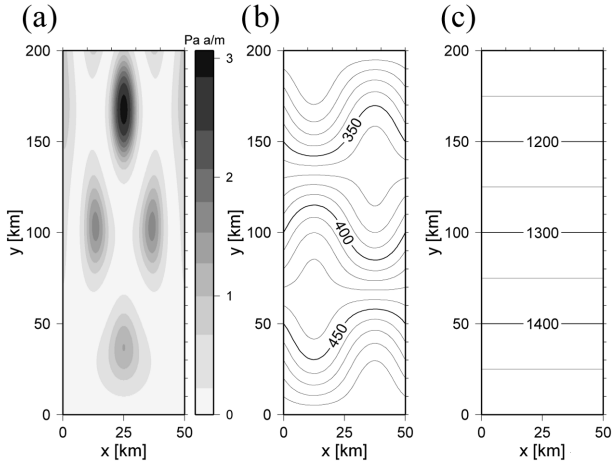

Fig. 1 MacAyeal's schematic model setup. (a) shading plot of basal friction parameter $c_b(x,y)$ in Pa a m^{-1}, isolines of (b) ice surface elevation in m and (c) ice thickness in m

linear relation

$$\tau^{x(b)} = c_b(x,y)u,\ \tau^{y(b)} = c_b(x,y)v,\ c_b(x,y) = \mu(x,y) > 0 \tag{5.1}$$

is applied by MacAyeal (1993) and we follow that publication here.

Dirichlet conditions apply for the boundary, with $u = 0$, $v = 0$ at the lateral boundaries of the stream, $u = 0$ at the influx and outflux line of the stream. The velocity v at the influx and outflux line is defined by a linear combination of sinusoidal functions, which higher velocities at outflux compared to that at influx line, see MacAyeal (1993) for more details. For the average inverse rate factor $\bar{B} = 1.8 \times 10^8 \mathrm{Pas}^{1/3}$ applies.

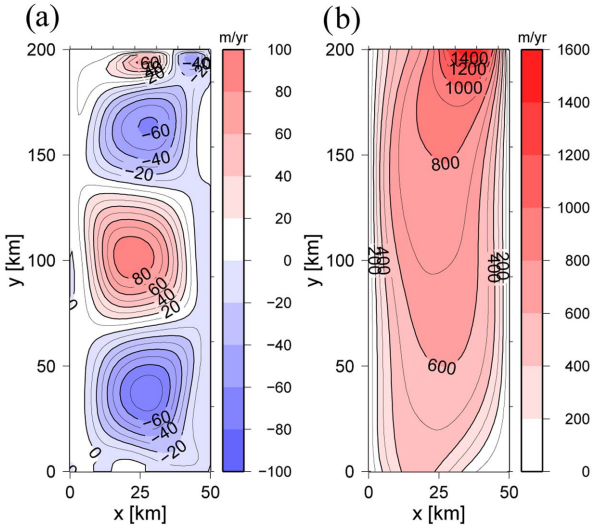

Fig. 2 Simulated velocity. (a) transverse velocity u and (b) longitudinal velocity v in m/a

Fig. 2 shows the velocity fields computed with our weighted Jacobi method. As in reality, the ice-stream velocity is dominated by its longitudinal component. Its magnitude increase towards the middle of the ice stream and downstream. The different signs of the transverse velocity component component reflect the undulations in the surface elevation, which affect the velocity via the driving stress due to the gradient of the surface elevation. When using the same scale and orientation, the isolines of our simulated surface velocity match well to results printed in an earlier publication MacAyeal (1993). The respective fields yielded with the SOR or the splitting method are not displayed explicitly here, because they are almost identical to those found with the weighted Jacobi method.

We also found the right ω for Jacobi (as promised above) and SOR in several numerical experiments. For Jacobi a value of $\omega = 0.6$ is safe, while higher values lead to instability. For SOR we choose $\omega = 1.4$, although we are aware that there is still room for fine tuning to reach better performance. Further simulations (not presented explicitly here) showed a strong dependence of the convergence on the spatial resolution and we are motivated to inspect the eigenvalues of operator M more closely (Section 5.2), which is stiff.

5.2 Eigenvalues of the SSAB equations

We will later demonstrate that the splitting leads to faster convergence and reduced run times compared to the one-side iterative schemes. But first, we show that Eqs. (4.40) and (4.41) are indeed a reasonable splitting due to the far larger condition of matrix M compared to matrix B.

In the computations of the condition numbers, the average inverse rate factor has the same value as before. But for the basal shear stress (Eq. (1.4), we use the non-linear relation Pollard and DeConto (2007) now and set $p = 1.25$ and $c_b = 5.4 \times 10^6$ kgm$^{-13/9}$s$^{-14/9}$. Our c_b is somewhat smaller that that of Pollard and DeConto (2007), but still within the range of measurements (Jenson 1994).

As before, we use Dirichlet boundary conditions, but here we reduce the boundary values by a factor of 0.5 compared to MacAyeal (1993), because such a setting is consistent with the stronger basal drag leading to smaller velocities as MacAyeal (1993), which compares better with observations.

The spatial resolution of the simulation shown in Fig. 3 is $\Delta x = 5000$ m corresponding to 10×40 grid cells and we run the model until the stop criterion Eq. (4.39) with $err = 10^{-3}$ m/a is fulfilled. Then the condition numbers are computed 100 times for equally spaced iterations steps from the

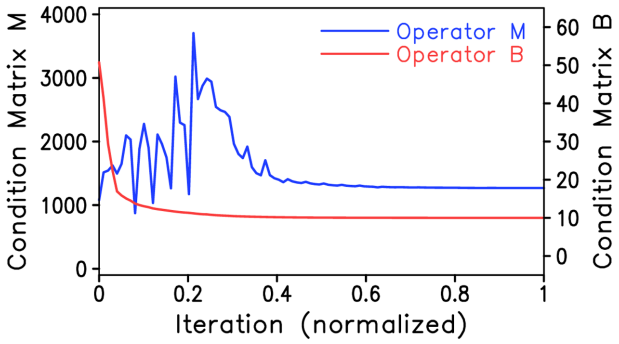

Fig. 3 Conditions of M and B of the SSAB equation with a spatial resolution of $\Delta x = 5000$ m. Please, note the different scales of the y-axes (left and right side)

Table 1 Condition numbers and relative condition numbers for different grid spacings at the end of iteration

Δx	χ_M	χ_B	χ_M / χ_B
5000	1300	10	130
2500	10000	18	570
1250	45000	30	1500

beginning to the end of the entire computation.

We determine the conditions of the matrices via their maximal and minimal eigenvalues (Eqs. (2.8) and (2.9) with the free software OCTAVE (Section 3), which reads and evalutes the output of the simulations with the FORTRAN90 code.

The condition of the respective matrices show a strong dependence on the iteration steps (Fig. 3). But the condition of matrix M is for nearly all iteration steps two orders of magnitude higher that the condition of matrix B; for the very first iteration steps, the condition of the matrix M is still about a factor of 20 higher compared to that of matrix B.

We also investigated the dependence of the condition numbers on the spatial resolution at the end of the iteration. In addition the the run with $\Delta x = 5000$ m, we performed simulations with $\Delta x = 2500$ m and $\Delta x = 1250$ m corresponding to 20×80 and 40×160 grid cells, respectively. For these finer resolutions, we computed the condition number at the end of the simulations (see Table 1).

We found increasing condition numbers with decreasing grid spacing. But still with a value of about 30 for $\Delta x = 1250$ m, the condition of the matrix B remains small. Furthermore, the relative conditions (condition number of matrix M divided by condition number of matrix B) also increase with decreasing grid spacing. Because a reasonable resolution of an ice stream is rather finer than $\Delta x = 5000$ m, the higher (relative) condition numbers will appear in practice.

These very clear results motivate a stronger iteration on matrix M than that on matrix B with fixed iteration steps for the splittings; i.e., for now we assume that the dominance of the condition of matrix M will never change.

5.3 Performance studies for weighted Jacobi, SOR and the M-B splitting

Here, we investigate the performance and dependence on spatial resolution of our applied schemes using the same model parameters and boundary conditions as in section 5.2. Again, our stop criterion with $err = 10^{-3}$ m/a applies. Also, for the SOR scheme we use the same weights as in its non-splitting application. To determine the iteration steps n_M of the split in M and the weights ω_B of the split in B for different spatial resolutions, we performed simulations for $\Delta x = 1250$, 2500, 5000 m with varying n_M and ω_B. For a preliminary fixing of the optimal parameter values, we started with the rather coarse steppings $n_M = 10$, 20, 40 and $\omega_B = 0.005$, 0.01, 0.05. Then we applied refined steppings for each individual resolution. Because we determined ω_B only with an accuracy of one significant digit, we still left room for further improvement. As already mentioned, we use SOR as sub-method to solve for the split in M. Here, we assumed that the weight $\omega_M = 1.4$ is as optimal as for SOR in its non-split application. For fixed spatial resolution, we chose those weights ω_B and iteration steps n_M as optimal, for which the run time is smallest and the scheme stays still stable. To avoid interference with different nodes of the machine all nodes which share the same memory except for one are switched off. We checked the accuracy of the run time with the same numerical method for different nodes on the IPLEX and found that the run times compared, at least

Table 2 Performance of the different numerical methods with applied weights of the operators. ω is the weight/relaxation parameter for the the Jacobi/SOR algorithm in the one-side schemes. For the M split of the M-B splitting, n_M and ω_M denote the iteration steps and the relaxation parameter, respectively. For the B split, ω_B describes the weight and the iterations step is always $n_B = 1$ here

Method	ω	n_M	ω_M	ω_B	Run time (s)
Grid spacing 5000 m:					
Jacobi	0.6	-	-	-	6.4
SOR	1.4	-	-	-	1.7
Splitting	-	15	1.4	0.09	0.6
Grid spacing 2500 m:					
Jacobi	0.6	-	-	-	91
SOR	1.4	-	-	-	25
Splitting	-	30	1.4	0.05	8.5
Grid spacing 1250 m:					
Jacobi	0.6	-	-	-	1120
SOR	1.4	-	-	-	335
Splitting	-	50	1.4	0.02	115

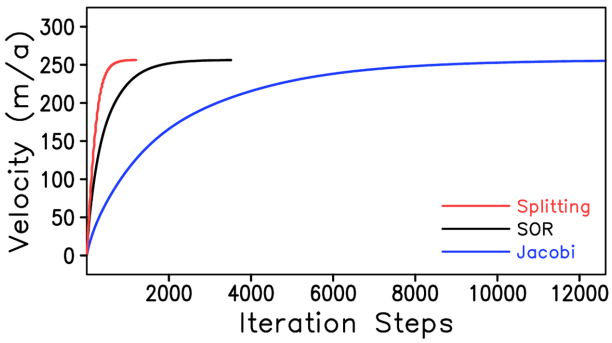

Fig. 4 Average of absolute value of the velocity with respect to the iterative steps for different schemes. For depicted curves a medium spatial resolution of $\Delta x = 2500$ m applied in the simulations. The splitting scheme (red) converges the fastest

for the accuracy of the significant digits listed for our run times.

Table 2 shows the run times yielded with the weights which we found. For the non-splitting schemes, we have a gain in performance by a factor of about 3.3 to 3.8 (depending on resolution) if we compare SOR with Jacobi. Here, we used fixed weights for all resolutions. In the splitting scheme, we had to apply smaller weights in the B split with decreasing grid spacing in order to have stability. The splitting scheme is faster by a factor of 2.9 to 3.0 than the non-splitting SOR scheme. This gain in performance is comparable with that which we reached without splitting by the choice of the standard method and its parameters.

Fig. 4 shows the iteration steps, which are needed to obtain convergent results. The gain in performance of the different schemes is clearly seen. The best results are obtained with the proposed iterative splitting scheme. All curves are very smooth indicating stable schemes. Further on, all curves do asymptotically approach the same limit. This is also true for the entire two dimensional velocity fields, which we have inspected for all simulations.

6. Conclusions

We presented an iterative operator-splitting method to solve the SSAB equations with respect to their underlying spatial scales. In the our splitting, one operator for membrane stresses and another operator for basal shear stress are solved for in an alternating method.

The scheme was validated by comparison with resulting velocity fields from previous research using an artificial ice-stream topography. Our simulated velocity fields compared with excellent agreement to those by MacAyeal (1993).

The computation of the eigenvalues of the two split operators shows a much higher condition number for the operator belonging to the membrane stresses compared to the operator associated with the basal shear stress for a broad range of spatial resolutions. An embedded (expensive) computation of the eigenvalues in the algorithm is most probably not necessary. But in our computations we used standard values for the physical parameter like the average inverse rate factor and the basal shear parameter. In practice, these parameters can vary strongly. Therefore, it is planned to investigate the validity of our split for the full range of physical parameters in the future.

The splitting performed here was shown to be successful; compared to the SOR scheme in its non-splitting application, the gain in performance amounts to a factor of three. We reached such gain in performance despite the rather small weights we had to apply in the Jacobi scheme, which is our solving scheme of the B split, in order to guarantee stability. Therefore, the underlying splitting has the potential to considerably speed up the SSAB solving, but it needs improvements in the sub-methods, in particular in the method solving for the B split. But still one should keep in mind, that the splitting is relatively easy to implement: only a simple rearrangement of expressions from implicit to explicit treatment is necessary. Therefore, the actual gain in performance is easy to achieve.

We also see the benefits in accelerating our iterative schemes with respect to Krylov-subspace iterations (see Hestenes and Stiefel 1952, Saad 2003) and multigrid methods (see (Hackbusch 1985, Trottenberg *et al*. 2001)) which allows to reduce the iteration count to $O(h^{-1})$ ($O(h^{-\mathrm{d}-1}\})$ time in serial instead of our iterative counts with $O(h^{-2})$ ($O(h^{-\mathrm{d}-2})$ time in serial. In future we consider such acceleration techniques.

Moreover, we will pay more attention to the solving of the non-linear part of the equations by introducing other methods in addition to the splitting, e.g., Newton's method, in order to gain efficient solver methods for highly spatially resolved ice stream equations. In future, we develop a novel method, that taken into account the benefit of split approximations (decouple to simpler problems) with the shifted Laplacian idea (improved matrix conditions) to establish a stable and fast SSAB solver, see van Gijzen *et al*. (2007).

Acknowledgments

R. Calov was funded by the Deutsche Forschungsgemeinschaft RA 977/6-1. We are grateful to T. Recknagel for programming assistance and A. Robinson for editing our manuscript. Further, we would like to thank C. Linstead for assistance with the IBM iDataPlex Cluster.

References

Arakawa, A. (1972), *Design of the ucla general circulation model*, Technical report no. 7, Department of

Meteorology, University of California.
Barbu, V. (1998), *Partial differential equations and boundary value problems*, Kluwer Academic Publishers, Dordrecht.
Bueler, E. and Brown, J. (2009), "Shallow shelf approximation as a "sliding law" in a thermomechanically coupled ice sheet model", *J. Geophys. Res*., **114**(3), F03008.
Bueler, E., Lingle, C.S., Kallen-Brown, J.A., Covey, D.N. and Bowman, L.N. (2005), "Exact solutions and verification of numerical models for isothermal ice sheets", *J. Glaciol*., **51**(173), 291-306.
Geiser, J. (2008), "Operator-splitting methods respecting eigenvalue problems for convection diffusion and wave equations", *Int. J. Appl. Math. Mech*., **4**(5), 93-104.
Geiser, J. (2009), "Operator-splitting methods in respect of eigenvalue problems for nonlinear equations and applications to burgers equations", *J. Comput. Appl. Math*., **231**(2), 815-827.
Geiser, J. (2011), *Iterative splitting methods for differential equations*, CRC Press, New York, USA.
Geiser, J. (2010), *Iterative splitting methods for differential equations. series: numerical analysis and scientific computing*, CRC Press, Taylor and Franics Group, London.
Greve, R. and Blatter, H. (2009), *Dynamics of ice sheets and glaciers*, Springer, Berlin, Germany.
Hackbusch, W. (1985), *Multi-grid methods and applications, Springer*, Berlin, Germany.
Hackbusch, W. (1986), *Theorie und numerik elliptischer differentialgleichungen*, Teubner, Stuttgart, Germany.
Hestenes, M. and Stiefel, E. (1952), "Methods of conjugate gradients for solving linear systems", *J. Res. Nat. Bureau Standards*, **49**(6), 409-438.
Jenson, J.W. (1994), *A nonlinear numerical model of the Lake Michigan lobe*, Laurentide ice sheet. Doctoral thesis, Department of Geology, Oregon State University, Corvallis, OR, USA.
Joughin, I., Fahnestock, M. MacAyeal, D.R., Bamber, J.L. and Gogenini, S.P. (2001), "Observation and analysis of ice flow in the largest Greenland ice stream", *J. Geophys. Res*., **106**(24), 34021-34034.
Kanney, J., Miller, C. and Kelley, C. (2003), "Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems", *Adv. Water. Resour*., **26**(3), 247-261.
LAPACK92. LAPACK (1992), *Linear algebra package*, Software Library, http://www.netlib.org/lapack/.
LIS11. LIS (2011), *A library of iterative solvers for linear systems*, Software Library, http://www.ssisc.org/.
Storti, M.M., Nigro, N. and Idelsohn, S. (1991), "Multigrid methods and adaptive refinement techniques in elliptic problems by finite element methods", *Comput. Method. Appl. M*., **93**(1),13-30.
MacAyeal, D.R. (1993), "A tutorial in the use of control methods in ice-sheet modeling", *J. Glaciol*., **39**(131), 91-98.
MacAyeal, D.R. (1995), "Basal friction of Ice Stream E,West Antarctica", *J. Glaciol*., **41**(138), 247-262.
Pollard, D. and DeConto, R.M. (2007), *A coupled ice-sheet/ice-shelf/sediment model applied to a marine-margin flowline: forced and unforced variations*, (Eds., Hambrey, M.J., Christoffersen, P., Glasser, N.F. and Hubbard, B.), Glacial Sedimentary Processes and Products, International Association of Sedimentologists Special Publication No. 39, 37-52. Blackwell Publ.
Saad, Y. (2003), *Iterative methods for sparse linear systems,*Society for Industrial Mathematics, Philiadelphia, USA.
Smith, B. Bjoerstad, P. and Gropp, W. (1999), *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge Univ Press, New York, USA.
Smith, B., Boyle, J., Dongarra, J., Garbow, B., Ikebe, Y., Klema, V. and Moler, C. (1976), *Matrix eigensystem routines*, EISPACK Guide. Lecture Notes in Computer Science, Springer, New York.
Thomas, J.W. (1999), *Numerical partial differential equations: conservation laws and elliptic equations*, Springer, New York.
Toselli, A. and Widlund, O. (2005), *Domain decomposition methods: algorithms and theory*. Springer, Berlin, Germany.
Trottenberg, U., Oosterlee, C. and Schüller, A. (2001), *Multigrid*, Academic Press, San Diego, USA.
van Gijzen, M.B., Erlangga, Y.A. and VuikKanney, C. (2007), "Spectral analysis of the discrete helmholtz operator preconditioned with a shifted laplacian", *SIAM J. Sci. Comput*., **29**(5),1942-1958.
Zeidler, E. (1990), *Nonlinear functional analysis and its applications II/B: nonlinear monotone operators*, Springer, New York.