Manuscript prepared for Geosci. Model Dev.
with version 2014/09/16 7.15 Copernicus papers of the LaTeX class copernicus.cls.
Date: 15 May 2015

Finite-Element Sea Ice Model (FESIM), version 2

Sergey Danilov[1,2], Qiang Wang[1], Ralph Timmermann[1], Nikolay Iakovlev[3], Dmitry Sidorenko[1], Madlen Kimmritz[1], Thomas Jung[1], and Jens Schröter[1]

[1]Alfred Wegener Institute for Polar and Marine Research, Bremerhaven, Germany
[2]A. M. Obukhov Institute of Atmospheric Physics RAS, Moscow, Russia
[3]Institute of Numerical Mathematics RAS, Moscow, Russia

Correspondence to: S. Danilov (sergey.danilov@awi.de)

Abstract. The Finite-Element Sea-Ice Model, used as a component of the Finite-Element Sea ice Ocean Model, is presented. Version 2 includes the elastic-viscous-plastic (EVP) and viscous-plastic (VP) solvers and employs a flux corrected transport algorithm to advect the ice and snow mean thicknesses and concentration. The EVP part also includes a modified approach proposed recently by Bouillon et al., which is characterized by an improved stability compared to the standard EVP approach. The model is formulated on unstructured triangular meshes. It assumes a collocated placement of ice velocities, mean thicknesses and concentration at mesh vertices, and relies on a piecewise-linear (P_1) continuous elements. Simple tests for the modified EVP and VP solvers are presented to show that they may produce very close results provided the number of iterations is sufficiently high.

1 Introduction

The Finite-Element Sea Ice Model (FESIM) was developed as a component of the Finite-Element Sea Ice Ocean circulation Model (FESOM) (for a recent description see Wang et al. (2014)) in 2003. Its basis was the standard zero-layer thermodynamical component, and an elastic-viscous-plastic (EVP) solver coded following Hunke and Dukowicz (1997) and the early version of CICE documentation (see Hunke and Lipscomb (2008) for the current one). It was the first unstructured-mesh sea ice model used for global ocean – sea ice simulations. The description of the first version was only available as an internal technical report (Danilov and Iakovlev 2003, unpublished manuscript) and in a brief form was presented by Timmermann et al. (2009). The $P_1 - P_1$ (linear polynomials on triangles for velocities and scalars) continuous representation used in the dynamical core led to a very compact code relying on the numerical infrastructure of FESOM. The components of stresses

and strain rate tensors are elementwise constant, which makes the numerical implementation very straightforward.

Version 2 of the model is augmented by a new viscous-plastic (VP) solver, while the Galerkin-Least-Squares stabilized advection scheme inherited from early versions of FESOM is replaced by the FE flux corrected transport (FCT) scheme by Löhner et al. (1987), which warrants better numerical stability. It also contains the new EVP solver by Bouillon et al. (2013), which puts the EVP and VP approaches on the same footing. The model reached a high level of maturity and shows a robust behavior in numerous simulations performed with FESOM (see, e. g., Sidorenko et al. (2011), Wang et al. (2012), Wekerle et al. (2013), Timmermann and Hellmer (2013), Wang et al. (2014) and Sidorenko et al. (2014)). It may serve as a prototype for other groups developing unstructured-mesh models intended for large-scale ocean sea-ice simulations.

The intention of this paper is to present the description of the dynamical part of the model (momentum balance and tracer advection), and illustrate the performance of the solver algorithms implemented in the model. The thermodynamical part will not be described here, as its implementation is standard (pointwise) and is not affected by the unstructured character of the surface mesh. It follows Parkinson and Washington (1979) and includes a prognostic snow layer (Owens and Lemke (1990)).

Several approaches to sea-ice modelling on unstructured meshes have been proposed recently. Hutchings et al. (2004) describes an approach based on a finite-volume (FV) cell-centered discretization. Another finite-volume implementation is that by FVCOM, which follows CICE (see Hunke and Lipscomb (2008)), but employs cell-vertex discretization, i. e., velocities are on cells (triangles), and tracers are on vertices (see Gao et al. (2011)). Next to FESIM, another finite-element (FE) model has been proposed by Lietaer et al. (2008). It relies on linear non-conforming elements for velocities (full velocity vectors are associated with the edges of the triangular mesh) and elementwise constant tracers. We comment on these discretizations later.

Sections 2 and 3 introduce the basic equations and present the description of model's numerical part. We discuss some aspects of model performance in section 4, and conclude the presentation in section 5.

2 Governing equations, VP and EVP methods

2.1 Governing equations

The sea-ice motion equation is

$$m(\partial_t + \boldsymbol{f}\times)\boldsymbol{u} = a\boldsymbol{\tau} - aC_d\rho_o(\boldsymbol{u}-\boldsymbol{u}_o)|\boldsymbol{u}-\boldsymbol{u}_o| + \boldsymbol{F} - mg\nabla H. \quad (1)$$

Here m is the ice plus snow mass per unit area, C_d the ice-ocean drag coefficient, ρ_o the water density, a the sea ice concentration, $\boldsymbol{u}=(u,v)$ and $\boldsymbol{u}_o$ the ice and ocean velocities, $\boldsymbol{\tau}$ the wind

stress, H the sea surface elevation, g the acceleration due to gravity and $F_j = \partial_i \sigma_{ij}$ is the force from stresses within the ice. We use Cartesian coordinates for brevity, with $i, j = 1, 2$ implying x and y directions; the implementation of spherical coordinates will be discussed later. Summation over repeating coordinate indices is implied. The total mass m is

$$m = \rho_{ice} h_{ice} + \rho_s h_s, \tag{2}$$

with ρ_{ice} and ρ_s, respectively, the densities of ice and snow and h_{ice} and h_s their mean thicknesses (volumes per unit area).

The internal ice stresses are computed assuming the VP rheology (Hibler (1979)). One writes

$$\sigma_{ij} = 2\eta(\dot{\epsilon}_{ij} - (1/2)\delta_{ij}\dot{\epsilon}_{kk}) + \zeta\delta_{ij}\dot{\epsilon}_{kk} - (1/2)\delta_{ij}P, \tag{3}$$

where

$$\dot{\epsilon}_{ij} = (1/2)(\partial u_i/\partial x_j + \partial u_j/\partial x_i) \tag{4}$$

is the strain rate tensor, η and ζ are the moduli ('viscosities') and P is the ice strength. Both the stress and the strain rate tensors are symmetric, so they are characterized by only three independent components. The standard VP rheology adopts the following scheme of computing the ice strength P and moduli η and ζ:

$$P = P_0, \quad \zeta = (P_0/2)/(\Delta + \Delta_{min}), \quad \eta = \zeta/e^2, \tag{5}$$

where

$$P_0 = h_{ice} p^* e^{-C(1-a)}, \quad \Delta^2 = (\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2)(1 + 1/e^2) + 4\dot{\epsilon}_{12}^2/e^2 + 2\dot{\epsilon}_{11}\dot{\epsilon}_{22}(1 - 1/e^2), \tag{6}$$

$e = 2$ (the ellipticity parameter) and $C = 20$; the default values in FESOM for Δ_{min} and $p*$ are $\Delta_{min} = 2 \cdot 10^{-9}$ s^{-1} and $p^* = 27500$ N/m^2. In this scheme, Δ_{min} serves for a viscous regularization of plastic behavior in areas where Δ is very small. The ice strength can be modified as $P = P_0\Delta/(\Delta + \Delta_{min})$ for stresses to remain on the elliptic yield curve even if Δ is small, and we will follow this variant below. We note that multi-category ice implementations (such as CICE, see Hunke and Lipscomb (2008)) use different parameterizations for P_0, which take into account the distribution of ice over thickness categories. This does not change the basic equations (1, 3).

In our case we deal with three tracers, the concentration a, ice mean thickness (volume per unit area) h_{ice} and snow mean thickness h_s. They are advected by the ice velocities and modified through thermodynamical forcing

$$\partial_t a + \nabla \cdot (\boldsymbol{u} a) = S_a, \quad \partial_t h_{ice} + \nabla \cdot (\boldsymbol{u} h_{ice}) = S_{ice}, \quad \partial_t h_s + \nabla \cdot (\boldsymbol{u} h_s) = S_s \tag{7}$$

with S_a, S_{ice} the sources related to sea ice melting and freezing, and S_s the sources due to snow precipitation and melting. The system (1), (3) and (7), augmented with an appropriate model of sources and boundary conditions, defines the sea ice model. We use the no-slip boundary conditions for momentum and no-flux condition for tracers at lateral walls.

2.2 VP and EVP methods

The well known difficulty in solving the ice momentum equation is related to the internal stress term, which makes this equation very stiff and would require time steps of fractions of second if stepped explicitly. There are two common ways of handling this difficulty. The first one treats a part of stress divergence in an implicit way, with linearization for the moduli, as suggested by Zhang and Hibler (1997). As mentioned by Lemieux and Tremblay (2009), it does not warrant full convergence, and a full nonlinear solver (for example, a Jacobian-free Newton-Krylov solver, see Lemieux et al. (2012)) has to be used for that. This strategy is still too expensive computationally, so the VP solver adopted by us is similar in spirit to that of Zhang and Hibler (1997) (see section 3.4). The second way is to reformulate the VP approach by adding pseudo-elasticity, which leads to the so-called elastic–viscous–plastic (EVP) method. It raises the order of the system (1, 3) with respect to time, which makes the CFL limitation on the explicit time step much less severe than in the original VP framework.

The EVP approach, as proposed by Hunke and Dukowicz (1997) (see also Hunke and Lipscomb (2008)), is described as follows. One first defines the combinations

$$\sigma_1 = \sigma_{11} + \sigma_{22}, \quad \sigma_2 = \sigma_{11} - \sigma_{22} \tag{8}$$

and similar combinations for the strain rate components:

$$\dot{\epsilon}_1 = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \quad \dot{\epsilon}_2 = \dot{\epsilon}_{11} - \dot{\epsilon}_{22}. \tag{9}$$

In this notation, the EVP approach is

$$\frac{\partial \sigma_1}{\partial t} + \frac{\sigma_1}{2T} + \frac{P}{2T} = \frac{P}{2T\Delta}\dot{\epsilon}_1, \tag{10}$$

$$\frac{\partial \sigma_2}{\partial t} + \frac{e^2\sigma_2}{2T} = \frac{P}{2T\Delta}\dot{\epsilon}_2, \tag{11}$$

$$\frac{\partial \sigma_{12}}{\partial t} + \frac{e^2\sigma_{12}}{2T} = \frac{P}{2T\Delta}\dot{\epsilon}_{12}, \tag{12}$$

where T is the relaxation time. It determines the time scale of transition from elastic behavior to the VP rheology. The default value is $T = \Delta t/3$, where Δt is the external time step (set by the ocean model). It can be easily seen that the EVP 'rheology' becomes equivalent to the VP rheology if the contribution from the time derivatives are negligible on the time scale given by Δt. The equations for stresses are time stepped together with the momentum equation (1) at a shorter time step Δt_{EVP}, so that $N_{EVP} = \Delta t/\Delta t_{EVP}$ is a large number (about one hundred or more). A caveat of this approach is that by the end of the external time step the stresses may still differ from the VP solution, and the difference may accumulate with time. So in practice the EVP solution may slightly deviate from the VP one. Because of purely explicit time stepping for the stress-velocity pair (velocity is considered known in stress computations and vice versa), the EVP approach must respect the CFL limitation on the subcycling time step Δt_{EVP} (see Hunke and Dukowicz (1997), Hunke (2001)).

It can be circumvented by limiting 'viscosities' ($\zeta = P_0/2(\Delta + \Delta_{min}), \eta = \zeta/e^2$) so that they stay below some level (see Hunke (2001)) $P_0/((\Delta + \Delta_{min}) < C_{lim} T \Delta x^2/(\Delta t_{EVP})^2$, where C_{lim} is the limiting constant and Δx the grid cell size. However, on unstructured meshes this can modify solutions simply because of varying resolution (see the discussion by Losch and Danilov (2012)). Limiting is therefore not used by us. The stability condition then demands that Δt_{EVP} remains small. Note that the limitation on Δt_{EVP} becomes more restrictive for finer meshes, and would require to use larger N_{EVP}.

If not observed, the CFL limitation may lead to noisy fields of velocity divergence and viscosities in practical applications in the areas where Δ is low. The code remains stable in most cases (because of stability added through time stepping, see further) and produces relatively smooth results for the ice thickness and area coverage. Clearly, the noise may affect the ice dynamics, and a user must be aware of that. Fully eliminating it could be both difficult and expensive in terms of CPU time.

Bouillon et al. (2013) proposed a modified EVP approach in which subcycling is fully detached from the physical time stepping. It can be considered as a pseudo-time solver for the VP rheology. In this case one writes

$$\alpha(\sigma_1^{p+1} - \sigma_1^p) = \frac{P_0}{\Delta^p + \Delta_{min}}(\dot{\epsilon}_1^p - \Delta^p) - \sigma_1^p, \tag{13}$$

$$\alpha(\sigma_2^{p+1} - \sigma_2^p) = \frac{P_0}{(\Delta^p + \Delta_{min})e^2}\dot{\epsilon}_2^p - \sigma_2^p, \tag{14}$$

$$\alpha(\sigma_{12}^{p+1} - \sigma_{12}^p) = \frac{P_0}{(\Delta^p + \Delta_{min})e^2}\dot{\epsilon}_{12}^p - \sigma_{12}^p, \tag{15}$$

for stresses and

$$\beta(\boldsymbol{u}^{p+1} - \boldsymbol{u}^p) = -\boldsymbol{u}^{p+1} + \boldsymbol{u}^n - \Delta t \boldsymbol{f} \times \boldsymbol{u}^{p+1}$$
$$+ (\Delta t/m)[\boldsymbol{F}^{p+1} + a\boldsymbol{\tau} + C_d a \rho_o(\boldsymbol{u}_o^n - \boldsymbol{u}^{p+1})|\boldsymbol{u}_o^n - \boldsymbol{u}^p| - mg\nabla H^n] \tag{16}$$

for the velocity. Here α and β are some large constants. The superscript p is related to pseudotime iterations, replacing the subcycling of the standard EVP, and n is the index of external time stepping. Fields are initialized with values at time step n for $p = 1$, and their values for the last iteration $p = N_{EVP}$ are taken as solutions for time step $n + 1$. In order that CFL limitations be satisfied, the product $\alpha\beta$ should be sufficiently large compared to $\pi^2 P_0 \Delta t(\Delta + \Delta_m in)^{-1} m^{-1} \Delta x^{-2}$ (see Bouillon et al. (2013) and further comments by Kimmritz et al. (2014)). The regime of the standard EVP scheme ($N_{EVP} = 120$ and $T = \Delta t/3$) will be approximately recovered for $\alpha = \beta = 80$ (for σ_1) and $N_{EVP} = 120$, but much larger values have to be used on fine meshes to warrant the absence of noise in strain rates and viscosities. The stability requirements here are very similar to those of the standard EVP method if expressed in terms of N_{EVP}, and, likewise, become more restrictive for finer meshes. For numerical convergence, N_{EVP} should exceed α and β (for the same reason that T is a fraction of Δt in the standard EVP).

One expects that if this scheme is stable and converged, it would produce solutions identical to those of a converging VP solver, while the standard EVP scheme may slightly deviate. We will return

to this in sections 3.4 and 3.5 where the time stepping is discussed. In practice, it will seldom be run for full convergence, which is rather expensive, and some difference will be preserved.

FESIM implements the three approaches mentioned above, which will be referred further as VP, EVP and mEVP. The reason for keeping all them is two-fold. First, it facilitates the comparison of results with other models which may use one of these approaches. Second, their numerical efficiency and performance depend on applications, and one may wish to select the most appropriate one for a particular application.

3 Numerical implementation

We first describe spatial discretization, and then the discretization in time. The easiest way of introducing the FE method is through considering transport equations. For this reason we begin with advection, and then continue with the motion equation.

3.1 Finite-element discretization of ice transport equations

This section explains the FE spatial discretization, which is based on linear continuous functions defined on triangles. The original motivation for this choice was the ability to share the infrastructure with the ocean model, which is based on the same discretization. The transport equations (7) are solved in two steps: first scalar quantities are advanced with the right hand sides (rhs) of tracer equations set to zero. Then tracers are updated with account for thermodynamic sources and sinks in a pointwise manner. We therefore limit ourselves to homogeneous equations. In what follows, the superscript n will denote external time steps, and p subcycling time steps in solvers, as in the discussion above. Subscripts j and k will denote quantities related to vertices (nodes) of triangular mesh. It is hoped that they will not be mixed with the notation for coordinate indices of tensors. For the mesh indices the agreement on summation over repeating indices will only be kept for matrix–vector products.

The tracer equations are solved with the FE Taylor–Galerkin (TG) method (see, e. g., Zienkiewicz and Taylor (2000), p. 47), which is analogous to that of Lax–Wendroff for FV. One writes for the concentration

$$a^{n+1} = a^n + \Delta t \partial_t a|_n + (\Delta t^2/2)\partial_{tt} a|_n, \tag{17}$$

and substitutes

$$\partial_t a = -\nabla \cdot (\boldsymbol{u}^n a^n), \tag{18}$$

and

$$\partial_{tt} a = \nabla \cdot (\boldsymbol{u}^n \nabla \cdot (\boldsymbol{u}^n a^n)). \tag{19}$$

In the last case the velocity is considered steady during the tracer time step. This still provides the second order in time if velocity and tracers are considered to be shifted by a half time step (asynchronous time stepping). The resulting equation

$$a^{n+1} = a^n - \Delta t \nabla \cdot \boldsymbol{G}^n, \quad \boldsymbol{G}^n = \boldsymbol{u}^n a^n - (\Delta t/2)\boldsymbol{u}^n \nabla \cdot (\boldsymbol{u}^n a^n) \tag{20}$$

provides the second order in both time and space (for linear functions). Here $\boldsymbol{G}$ is the modified flux vector, with a diffusive flux that exactly compensates for the first-order error in the time derivative. Note that it does not introduce dissipation. The ice and snow thickness equations are solved similarly.

To solve the tracer equation (20) with the FE method one first projects it on an appropriate set of test functions M_j:

$$\int M_j(a^{n+1} - a^n + \Delta t \nabla \cdot \boldsymbol{G}^n) dS = 0 \tag{21}$$

and then integrates it by parts to obtain

$$\int (M_j(a^{n+1} - a^n) - \Delta t \nabla M_j \cdot \boldsymbol{G}^n) dS = -\Delta t \int M_j \boldsymbol{G}^n \cdot \boldsymbol{n} d\Gamma, \tag{22}$$

where Γ is the boundary of the domain S. At the solid boundary ($\boldsymbol{G} \cdot \boldsymbol{n} = 0$) or an open boundary located far from the ice-covered region (so that $a = 0$), the boundary integral is zero. We will assume that this is the case.

The procedure outlined above gives the equation in a so-called weak form. The discretization is obtained by expanding scalar fields and velocities into series

$$a(x, y, t) = \sum_k a_k(t) N_k(x, y), \tag{23}$$

and similarly for h_{ice}, h_s, and components u, v of the velocity vector $\boldsymbol{u}$. We use continuous Galerkin discretization implying that $M_j = N_j$, and that functions N_j are continuous across the boundaries of triangles. We select N_j as a linear function associated to vertex j of the triangular mesh. It equals one at vertex j and decays linearly to zero at all neighboring vertices; the expansion above is simply the linear interpolation and summation is over all vertices. As a result, the Galerkin system of equations on nodal values of ice concentration a_k (same for $(h_{ice})_k$ and $(h_s)_k$) is obtained

$$\mathbf{M}_{jk}(a_k^{n+1} - a_k^n) + \mathbf{A}_{jk} a_k^n = 0, \tag{24}$$

where

$$\mathbf{M}_{jk} = \int N_j N_k dS, \quad \mathbf{A}_{jk} = -\Delta t \int \nabla N_j (\boldsymbol{u} N_k - (\Delta t/2)\boldsymbol{u} \nabla \cdot (\boldsymbol{u} N_k)) dS. \tag{25}$$

Note that summation is implied over k (matrix–vector product). It will be reminded in some cases below too. A similar procedure is used to obtain discretized momentum equations. The mass matrix $\mathbf{M}_{jk}$ is not diagonal, but has a limited bandwidth (defined by the number of neighbors). Its appearance is what makes the method different from the FV Lax–Wendroff implementation. Indeed,

it is easy to check that the latter would lead to the same result on median-dual control volumes (obtained by connecting triangle centroids with mid-edge points), but with the diagonal-lumped mass matrix $\mathbf{M}^L_{jk}$, whose diagonal entries are sums of rows of $\mathbf{M}_{jk}$, and other entries are zeros. Two points should be mentioned here on practical implementation. First, the velocity field is linear on triangles, so computations of operator $\mathbf{A}_{jk}$ should be formally done with account for this. Doing so would not, however, improve accuracy compared to just using mean velocities on triangles, which simplifies computations. Second, true iterative solution of equations involving mass matrices, written schematically as $\mathbf{M}_{jk}b_k = c_j$, is expensive and is never attempted. Instead, one does three iterations $\mathbf{M}^L_{jk}b^{p+1}_k = (\mathbf{M}^L_{jk} - \mathbf{M}_{jk})b^p_k + c_j$ starting with $b^0_k = 0$. Doing more iterations does not improve dispersive properties of the method, yet doing just one (lumping) deteriorates the method rather noticeably.

The presence of a consistent mass matrix in the TG method effectively removes a significant portion of dispersion related to the Lax–Wendroff method. However, remaining dispersive errors may still be damaging. For this reason, the approach is augmented to the FE-FCT method as proposed by Löhner et al. (1987). In this method, the TG solution above serves as the high-order one, and will be denoted as $\tilde{a}^{n+1}_k$. The low order solution $\overline{a}^{n+1}_k$ is obtained by adding artificial dissipation to the rhs and replacing the consistent mass matrix with the lumped one on the left hand side (lhs),

$$\mathbf{M}^L_{jk}(\overline{a}^{n+1}_k - a^n_k) + \mathbf{A}_{jk}a^n_k = \gamma_{FCT}(\mathbf{M}_{jk} - \mathbf{M}^L_{jk})a^n_k, \tag{26}$$

which leads to a monotonic solution provided the parameter γ_{FCT} is sufficiently high (about one). The difference between the high-order solution $\tilde{a}^{n+1}_k$ and the monotonic low-order solution $\overline{a}^{n+1}_k$ is due to the antidiffusive flux contribution,

$$\mathbf{M}^L_{jk}(\tilde{a}^{n+1}_k - \overline{a}^{n+1}_k) = -(\mathbf{M}_{jk} - \mathbf{M}^L_{jk})((\gamma_{FCT} - 1)a^n_k + \tilde{a}^{n+1}_k) \tag{27}$$

The rhs of the last expression is split into contributions from separate elements. They are limited as detailed in Löhner et al. (1987) and assembled back to recover a monotonic solution a^{n+1}_k instead of $\tilde{a}^{n+1}_k$.

By construction, the solution method is conserving. Indeed, because $\sum_j N_j(x,y) = 1$, $\sum_j \mathbf{A}_{jk} = 0$, and $\sum_j \mathbf{M}_{jk}a_k$ is the area integral. Also $\sum_j \mathbf{M}_{jk}a_k = \sum_j \mathbf{M}^L_{jk}a_k$, so that the simple iterative procedure above preserves conservation. According to Budgell et al. (2007) the FCT method shows second order convergence in simple advection tests. Note, however, that the ice velocity is divergent and may thus lead to the formation of local extrema in scalar fields. The FCT scheme may therefore result in excessive smoothing of extrema. Yet it does so for the antidiffusive fluxes only, the low-order solution will react to the divergence of the velocity field. For this reason the parameter γ_{FCT} should be taken at minimum compatible with stability and preservation of positivity.

Despite the fact that the FCT limiting doubles the computational cost of advection (compared to using solely the TG method), the burden remains small compared to the cost of solving for ice velocities.

3.2 Computation of strain rates and stresses

Similar to the thicknesses and concentration, ice velocities are considered to be linear functions on elements:

$$(u, v) = \sum_k (u_k, v_k) N_k. \tag{28}$$

The strain rates are therefore elementwise constant. At this point we need to take into account sphericity and peculiarities coming from the derivatives of metric terms. We use spherical coordinate system with poles at land to avoid the pole singularity. In spherical coordinates (ϕ, θ)

$$\dot{\epsilon}_{11} = \frac{1}{R\cos\theta}\left(\frac{\partial u}{\partial \phi} + v\frac{\partial \cos\theta}{\partial \theta}\right), \quad \dot{\epsilon}_{22} = \frac{1}{R}\frac{\partial v}{\partial \theta}, \tag{29}$$

and

$$\dot{\epsilon}_{12} = \frac{1}{2R}\frac{\partial u}{\partial \theta} + \frac{1}{2R\cos\theta}\left(-u\frac{\partial \cos\theta}{\partial \theta} + \frac{\partial v}{\partial \phi}\right). \tag{30}$$

Here R it the Earth radius. We approximate the geometry as locally flat on triangles, which is equivalent to replacing $\cos\theta$ in $(1/\cos\theta)\partial/\partial\phi$ by its estimate on elements. If we use a *local* Cartesian frame of reference on *each element* with the x and y axes oriented along the directions $\boldsymbol{e}_\phi$ and $\boldsymbol{e}_\theta$, we can then write ∂_x and ∂_y instead of $(1/R\cos\theta)\partial/\partial\phi$ and $(1/R)\partial/\partial\theta$ respectively. With the same accuracy we make an elementwise-constant estimate of the metric differentiation term, so that the expressions above become

$$\dot{\epsilon}_{11} = \partial_x u - v m_f, \quad \dot{\epsilon}_{22} = \partial_y v, \quad \dot{\epsilon}_{12} = (1/2)(\partial_y u + \partial_x v + u m_f), \tag{31}$$

where $m_f = \tan\theta/R$ is the metric factor. These expression for the strain rates are further used to compute the components of stresses which would then be naturally treated as elementwise constant too. Although the ice strength P would be more naturally modelled as a linear function because h_{ice} and a are represented in that way, the estimate of the ice strength gradient at vertex points will be the same if P is averaged to triangles, i. e., treated as elementwise constant. To further simplify computations we estimate h_{ice} and a on triangles as the mean over vertices. This makes all components of stresses elementwise quantities, so that time stepping of stresses in EVP and mEVP becomes an algebraic operation on triangles. Formally projecting the last equations on functions $M_c = 1$ on triangle (cell) c gives

$$(\dot{\epsilon}_{11})_c = \sum_{k(c)} (\partial_x N_k u_k - m_f v_k/3), \quad (\dot{\epsilon}_{22})_c = \sum_{k(c)} v_k \partial_y N_k \tag{32}$$

and

$$(\dot{\epsilon}_{12})_c = (1/2)\sum_{k(c)} (u_k \partial_y N_k + m_f u_k/3 + v_k \partial_x N_k). \tag{33}$$

Here summation is over vertices k of cell c, hence the symbolic notation $k(c)$. The expression for the ice strength is computed as $P_c = (h_{ice})_c(\Delta + \Delta_{min})_c^{-1} p^* \exp(-C(1-a_c))$ with $(h_{ice})_c = \sum_{k(c)}(h_{ice})_k/3$ and $a_c = \sum_{k(c)} a_k/3$. With the strain rates and ice strength known, Δ and the stress components are easily computed on elements.

3.3 Spatial discretization of momentum equation

Rigorous finite-element implementation of the momentum equation would involve mass matrices and would be too time consuming in the case of EVP and mEVP solvers. For that reason some simplifications are required. Luckily, mass matrices are not important here, as no compensation of discrete errors can be achieved with their help. We therefore use nodal quadratures in all terms that do not involve spatial derivatives. Multiplying (1) with test functions, integrating over the domain, and integrating the rheology term by parts, one gets:

$$\int m\boldsymbol{N}_j(\partial_t + \boldsymbol{f}\times)\boldsymbol{u}dS = \int \boldsymbol{N}_j(a\boldsymbol{\tau} - C_d a\rho_o(\boldsymbol{u}-\boldsymbol{u}_o)|\boldsymbol{u}-\boldsymbol{u}_o|)dS - \int(\nabla \boldsymbol{N}_j)\boldsymbol{\sigma}dS - \int mg\nabla H\boldsymbol{N}_j dS + \int \boldsymbol{N}_j(\boldsymbol{n}\boldsymbol{\sigma})d\Gamma. \tag{34}$$

Here $\boldsymbol{N}_j$ is a shortcut for either $(N_j, 0)$ or $(0, N_j)$, so that (34) is a set of two equations obtained by projecting on x and y directions, the second term on the rhs involves dyadic product of two tensors and the last term involves the contraction of the stress tensor with the unit vector normal to the boundary. On substituting the expansions in N_k for velocities, we approximate the lhs of (34) as

$$\int m\boldsymbol{N}_j(\partial_t + \boldsymbol{f}\times)\boldsymbol{u}dS = \boldsymbol{M}^L_{jk} m_k(\partial_t + \boldsymbol{f}\times)\boldsymbol{u}_k, \tag{35}$$

where $m_k = \rho_{ice}(h_{ice})_k + \rho_s(h_s)_k$ and $\boldsymbol{M}^L_{jk}$ is a shortcut for two 'vectors' $(\mathbf{M}^L_{jk}, 0)$ and $(0, \mathbf{M}^L_{jk})$. Similarly, the first term on the rhs is

$$\int \boldsymbol{N}_j(a\boldsymbol{\tau} - C_d a\rho_o(\boldsymbol{u}-\boldsymbol{u}_o)|\boldsymbol{u}-\boldsymbol{u}_o|) = \boldsymbol{M}^L_{jk}(a_k\boldsymbol{\tau}_k - C_d a_k\rho_o(\boldsymbol{u}-\boldsymbol{u}_o)_k|\boldsymbol{u}-\boldsymbol{u}_o|_k). \tag{36}$$

Summation over k implied in these equations is trivial because the lumped mass matrix is diagonal. The entries of the diagonal lumped mass matrix (for j=k) are just the areas of median-dual control volumes associated with vertices, i. e., one third of the sums of areas of triangles containing the vertex considered.

The second term on the rhs of (34) leads to the following contributions to equations for local x and y directions:

$$-\int(\nabla \boldsymbol{N}_j)\boldsymbol{\sigma}dS = \begin{array}{l}(\sum_{c(j)} A_c(-(\sigma_{11})_c\partial_x N_j - (\sigma_{12})_c\partial_y N_j - (\sigma_{12}m_f)_c/3),\\ \sum_{c(j)} A_c(-(\sigma_{12})_c\partial_x N_j - (\sigma_{22})_c\partial_y N_j + (\sigma_{11}m_f)_c/3)).\end{array} \tag{37}$$

Here $c(j)$ are the indices of cells containing vertex j (spanned by test function N_j) and A_c is the area of cell c. Notice that, because of metric differentiation, applying ∇ to any of $(N_j, 0)$ or $(0, N_j)$ also gives a contribution projecting on the other vector.

In the third term on the rhs of (34) computations of the slope term are simpler because the gradient of scalar field H does not involve differentiation of metrics. We use the nodal quadrature for the mass, which results in

$$-\int mg\nabla H \boldsymbol{N}_j dS = gm_j(G^x_{jk}, G^y_{jk})H_k, \tag{38}$$

with summation over k implied. Here $G^x_{jk} = \int N_j \partial_x N_k dS$, and similarly for the y-equation component. Clearly, $G^x_{jk}H_k = \sum_{c(j)}(A_c/3)\sum_{k(c)} H_k \partial_x N_k$ and likewise for the y equation.

The last term in (34) involves only vertices j on the boundary. We do not need equations there in the no-slip case, which is used by us, because zero velocity will be prescribed by the virtue of boundary conditions. Leaving equations there, but omitting the tangent component of this term would impose free-slip boundary conditions.

3.4 Time stepping and the implementation details of VP solver

As mentioned above, large values for viscosities in the VP case would lead to severe CFL limitations in the case of explicit time stepping. This suggests to account for the stress term in the ice motion equation implicitly,

$$m^n(1/\Delta t + \boldsymbol{f}\times)\boldsymbol{u}^{n+1} - m^n\boldsymbol{u}^n/\Delta t = \\ a^n\boldsymbol{\tau} - C_d a^n \rho_o(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n_o)|\boldsymbol{u} - \boldsymbol{u}_o|^n + \nabla\cdot\boldsymbol{\sigma}^{n+1} - m^n g\nabla H^n. \tag{39}$$

However, since the viscosities in $\boldsymbol{\sigma}$ are functions of the velocity field, the expression for $\boldsymbol{\sigma}$ should be linearized (by estimating viscosities on time step n) in order to use standard iterative solvers. The 'implicitness' is recovered by doing (Picard) iterations, when the velocity of the previous iteration is used to estimate the viscosities for the current iteration. Note that friction between ice and ocean is linearized and taken implicitly too.

This approach is suboptimal because of the need to solve a problem for a matrix of dimension $2N$, where N is the number of surface nodes (vertices). The non-zero entries in each row come from both u and v contributions in this case, which would make matrix–vector multiplications more expensive too.

The now traditional way of handling this problem was proposed by Zhang and Hibler (1997). In that case one makes implicit the terms involving u in the x-equation and terms involving v in the y-equation. This still requires assembling two matrices and preconditioning them. The approach employed by us was formulated by Hutchings et al. (2004). It is similar in spirit to that of Zhang and Hibler (1997), but allows us to use the same matrix for u and v. This considerably reduces the computational cost if general-purpose iterative solvers (like PETSc) are used. Its essence is in

writing the stress tensor (3) in the form

$$\sigma_{ij} = (\eta+\zeta)(\partial_i u_j) + \eta(\partial_j u_i) - \zeta(\partial_i u_j) + (\zeta-\eta)\delta_{ij}\dot{\epsilon}_{kk} - (1/2)\delta_{ij}P, \tag{40}$$

and making implicit only the first term on the rhs of this expression. Since the eigenvalue of the implicit operator is larger in this case than in the algorithm of Zhang and Hibler (1997), the method is stable. Yet its convergence rate is not necessarily better because it introduces an artificial residual through $\zeta(\partial_i u_j)$. The rest of the implementation resembles that of Zhang and Hibler (1997). It consist of three steps. The first two of them are iterations of the scheme

$$m^n \boldsymbol{u}^p/\Delta t - \partial_i(\eta+\zeta)^*\partial_i \boldsymbol{u}^p =$$
$$m^n(\boldsymbol{f}\times\boldsymbol{u}^* + \boldsymbol{u}^n/\Delta t) + a^n\boldsymbol{\tau} - C_d a^n \rho_o(\boldsymbol{u}^p - \boldsymbol{u}_o^n)|\boldsymbol{u}^* - \boldsymbol{u}_o^n| + \tilde{\boldsymbol{F}}^* - m^n g\nabla H^n, \tag{41}$$

where, as above, p is the index of iterations, and n of time stepping. In the original procedure $p=1,2$, but (Picard) iterations can be repeated to arbitrary high $p=N_p$. For $p=1$ the superscript $*$ implies that the quantity is estimated at time step n. For $p=2$, $\boldsymbol{u}^* = (\boldsymbol{u}^{p-1}+\boldsymbol{u}^n)/2$, $\tilde{\boldsymbol{F}}^* = \tilde{\boldsymbol{F}}(\boldsymbol{u}^*)$, and same for the viscosities on the lhs, following Zhang and Hibler (1997). For $p>2$ (if $N_p>2$) the starred quantities are those at iteration $p-1$. In the expressions above, $\tilde{\boldsymbol{F}}$ denotes the explicit part of the ice reaction. The final (third) step updates the Coriolis term to the implicit one:

$$m^n(\boldsymbol{u}^{n+1} - \boldsymbol{u}^{N_p})/\Delta t + m^n \boldsymbol{f}\times(\boldsymbol{u}^{n+1} - \boldsymbol{u}^*) = -C_d a^n \rho_o(\boldsymbol{u}^{n+1} - \boldsymbol{u}^{N_p})|\boldsymbol{u}^* - \boldsymbol{u}_o^n|. \tag{42}$$

Because of the need to keep the same matrix in u and v equations, the terms associated with metric differentiation in the lhs operator are all put on the rhs (added to those of $\tilde{\boldsymbol{F}}$), and the discretization of the operator part is straightforward. For convenience, we write down $\tilde{\boldsymbol{F}}$ in the finite-element discretization. We first omit the terms arising from metrics differentiation, as they are more conveniently taken into account separately all together. Since

$$\tilde{\sigma}_{11} = \eta\partial_x u - \zeta\partial_x u + (\zeta-\eta)(\partial_x u + \partial_y v) - P/2 = (\zeta-\eta)\partial_y v - P/2, \tag{43}$$

$$\tilde{\sigma}_{12} = \eta\partial_y u - \zeta\partial_x v, \quad \tilde{\sigma}_{21} = \eta\partial_x v - \zeta\partial_y u \tag{44}$$

and

$$\tilde{\sigma}_{22} = \eta\partial_y v - \zeta\partial_y v + (\zeta-\eta)(\partial_x u + \partial_y v) - P/2 = (\zeta-\eta)\partial_x u - P/2, \tag{45}$$

the divergence of stress tensor multiplied with test function $\boldsymbol{N}_j$ and integrated by parts, will lead to the following contributions to the rhs of the u and v components of the momentum equations

$$-\int(\nabla\boldsymbol{N}_j)\tilde{\boldsymbol{F}}dS = \quad (\int(-\partial_x N_j[(\zeta-\eta)\partial_y v - P/2] - \partial_y N_j(\eta\partial_x v - \zeta\partial_y u))dS,$$
$$\int(-\partial_x N_j(\eta\partial_y u - \zeta\partial_x v) - \partial_y N_j[(\zeta-\eta)\partial_x u - P/2])dS). \tag{46}$$

All derivatives and P are elementwise constant, so the integrals are equivalent to summation over the cells spanned by N_j.

It is easy to see that all 'metric differentiation terms' lead to the additional contributions

$$\int m_f[(\eta+\zeta)v\partial_x N_j - \eta u\partial_y N_j - \sigma_{12}N_j]dS \tag{47}$$

and

$$\int m_f[-\eta u\partial_x N_j + \sigma_{11}N_j]dS, \tag{48}$$

respectively, to u and v equations. The last terms in both contributions require integration of test functions, which gives $A_c/3$ on each cell involved.

The operator matrix is assembled in the standard sparse format on each time step. In order to reduce the computational load in the course of iterative solution, the matrix entries in the rows corresponding to nodes where the ice concentration is less than a small critical value are set to one at the diagonal, and zero otherwise. The rhs vector is corrected accordingly, and set to zero (default) or to the ocean velocity or to the velocity of the previous time step. ThePETSc solver with ILU preconditioning is used to solve the resulting matrix problem.

In theory, the tolerance does not necessarily need to be very small as the solution procedure is repeated on every time step, and the solution cannot diverge very much from the previous solution. However, on unstructured meshes a small tolerance can sometimes be required to achieve an acceptable accuracy on elements of differing size. Also, higher solver accuracy can be needed in quasistationary regimes, to properly handle areas where Δ is small. Our experience with PETSc is that while a tolerance of 10^{-6} may be sufficient on relatively uniform meshes, it should be at least two orders smaller if mesh size varies by a factor of 5 or more (see also discussion of convergence below).

There is always some sensitivity to the mesh, domain geometry and preconditioning; users are advised to experiment with the available options of the solver.

3.5 EVP and mEVP time stepping

In the EVP case equations (10–12) are time stepped implicitly on each cell (cell index c is omitted):

$$\sigma_1^{p+1} = d_1\left(\sigma_1 + \Delta t_{EVP}\frac{P_0}{2T}\frac{\dot\epsilon_1 - \Delta}{\Delta+\Delta_{min}}\right)^p, \tag{49}$$

$$\sigma_2^{p+1} = d_2\left(\sigma_2 + \Delta t_{EVP}\frac{P_0}{2T}\frac{\dot\epsilon_2}{\Delta+\Delta_{min}}\right)^p, \tag{50}$$

$$\sigma_{12}^{p+1} = d_2\left(\sigma_{12} + \Delta t_{EVP}\frac{P_0}{2T}\frac{\dot\epsilon_{12}}{\Delta+\Delta_{min}}\right)^p. \tag{51}$$

Here $d_1 = (1+\Delta t_{EVP}/2T)^{-1}$ and $d_2 = (1+\Delta t_{EVP}e^2/2T)^{-1}$. The initial value for $p=1$ is that from the previous time step n.

Pseudo-time stepping of the stress equations of mEVP is given by equations (13 - 15). It can also be made implicit as

$$\sigma_1^{p+1} = d_1\sigma_1^p + d_2\frac{P_0^p}{\Delta^p + \Delta_{min}}(\epsilon_1^p - \Delta^p), \tag{52}$$

$$\sigma_2^{p+1} = d_1\sigma_2^p + d_2\frac{P_0^p}{(\Delta^p + \Delta_{min})e^2}\epsilon_2^p, \tag{53}$$

$$\sigma_{12}^{p+1} = d_1\sigma_{12}^p + d_2\frac{P_0^p}{(\Delta^p + \Delta_{min})e^2}\epsilon_{12}^p, \tag{54}$$

where $d_1 = \alpha/(1+\alpha)$ and $d_2 = 1/(1+\alpha)$. This has however a very small impact on stability.

Time stepping of momentum equations is implicit for the Coriolis term and the part of ice-ocean stress. In the case of EVP the equations at each vertex j are

$$\begin{aligned}&\boldsymbol{u}_j^{p+1} + \Delta t \boldsymbol{f} \times \boldsymbol{u}_j^{p+1} + (C_d\Delta t\rho_o a m^{-1}\boldsymbol{u}^{p+1}|\boldsymbol{u}^p - \boldsymbol{u}_o|)_j = \\ &\boldsymbol{u}_j^p + (\Delta t a(\boldsymbol{\tau} + C_d a\rho_o \boldsymbol{u}_o|\boldsymbol{u}^p - \boldsymbol{u}_o|)m^{-1})_j - \\ &gM_j^{-1}\int \boldsymbol{N}_j^n \nabla H dS - M_j^{-1}m_j^{-1}\int \nabla \boldsymbol{N}_j \boldsymbol{\sigma}^{p+1} dS.\end{aligned} \tag{55}$$

The expressions for the two last terms have been given above (equations (37,38) and $M_j = M_{jj}^L$ with no summation (it is the area associated with vertex j). The fields are initialized with values at time step n. Pseudo-time stepping of the momentum part of mEVP is given by (16) with the terms interpreted similarly as in the equations above.

Now, when all equations are written, we can discuss the differences between the methods. The differences between the EVP and mEVP are subtle (apart from the difference in variables used to organize subcycling). First, (i) as can be seen comparing equations (10-12) with (13-15), the EVP uses different rates for σ_1 on one hand and σ_2 with σ_{12} on the other to approach the VP rheology. Second, (ii) after N_{EVP} iterations are done, the EVP scheme estimates the time derivative of velocity based on the last substep, while mEVP employs the estimate over the entire time step Δt. Third, (iii) there is damping in mEVP introduced by β, which helps to equilibrate the solution over the places where ice is weak. One does not expect large discrepancies between both approaches. However, it turns out that (i) has a negative impact on stability (cf. Bouillon et al. (2013)), which is why mEVP is more robust, as will be demonstrated below. At the end of the external time step the VP and mEVP solutions satisfy the same equations. To summarize, all three methods are expected to behave approximately similar, and the main point is the convergence of their solutions (and hence stability).

4 Box test case

The model described above is routinely used with FESOM both in an ice/ocean-only version or in a version coupled to an atmosphere model, so that its practical performance can be judged by the results of respective papers (see, e. g., Sidorenko et al. (2011),Wang et al. (2012), Wekerle et al. (2013), Timmermann and Hellmer (2013), Wang et al. (2014), Sidorenko et al. (2014)) and is

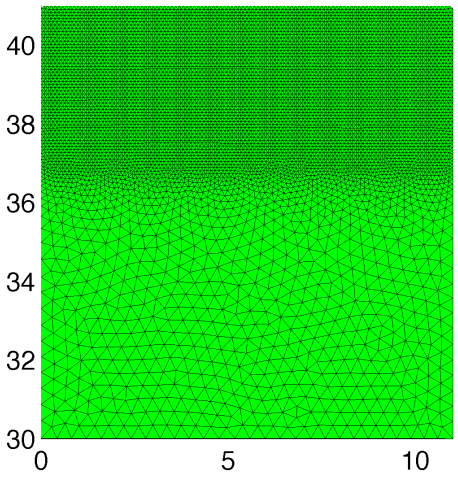

Figure 1. Triangular mesh used in simulations. The resolution varies from approximately 40 to 10 km. Stability of EVP and mEVP on the fine mesh requires that α, β and N_{EVP} be sufficiently large.

not repeated here. Thus far FESOM was run only with the EVP solver (since it was the first one available) and the comparison of the performance of the three available versions in the global setup is the subject of future work. Here we will use a box test case without thermodynamic forcing, with an intention to demonstrate similarities and disparities in the performance of VP, mEVP and EVP algorithms. This will be more difficult for realistic simulations where many other factors may contribute.

The setup follows that used by Hunke (2001), with the difference that islands are removed, geometry is spherical and the mesh is an unstructured one with variable resolution as used in Losch and Danilov (2012). The square box is of approximately 11 by 11 degrees in size (with the side lengths L_x and L_y) and the resolution is varied from approximately 40 to 10 km from the south to the north, as shown in Fig. **??**. It will be seen below that noise, if excited, appears at the fine mesh part, as could be anticipated. Apart from this, no other implications of mesh unstructuredness will be mentioned here to keep discussion concise and concentrated on the algorithm performance issues. Ice is driven by the wind stress $\boldsymbol{\tau} = C_a \rho_a \boldsymbol{u}_a |\boldsymbol{u}_a|$, with $C_a = 0.00225$. Here ρ_a is the air density and the wind velocity (in m/s) is taken as $u_a = 5 + (\sin(2\pi t/T) - 3)\sin(2\pi x/L_x)\sin(\pi y/L_y)$ and $v_a = 5 + (\sin(2\pi t/T) - 3)\sin(2\pi y/L_y)\sin(\pi x/L_x)$, where T =4 days. The ocean velocity (in m/s) is selected as $u_o = 0.1(2y - L_y)/L_y, \quad v_o = -0.1(2x - L_x)/L_x$, and the elevation H is computed by geostrophy. The coordinates x, y are the longitude and latitude counted from the south-west corner of the box. The ice thickness is 2 m initially and the ice concentration grows linearly from 0 to 1 in the west-east direction. The results of simulations at the end of one month are shown.

We start from comparing VP and mEVP solutions. In case A advection is switched off, and we compare the convergence of solutions obtained with different methods. In cases B and C the advection is switched on, they differ by the value of Δ_{min}: 2×10^{-9} s^{-1} (B) and 2×10^{-11} s^{-1} (C). Case A takes Δ_{min} of case B.

Figures 2 and 3 show, respectively, the zonal velocity and Δ (upper left panels) and the differences in solutions obtained by different methods in case A. We take the mEVP solution with $\alpha = \beta = 500$ and $N_{EVP} = 1000$ as a reference one (mEVP500), for modifications seen in runs with higher α, β and N_{EVP} are very small. The other solutions shown are those obtained with mEVP, but $\alpha = \beta = 250$ and $N_{EVP} = 250$ (mEVP250), and with VP, but in the regime with 2 (VP2p) and 10 (VP10p) additional Picard iterations (which means that $N_p = 4$ and 12 respectively). It is immediately seen from the velocity comparison that mEVP250 is far from convergence (there is a large-scale pattern in the velocity difference), and that it contains noise in the field of Δ. Note that the noise is seen over the fine part of the mesh, as stressed in Losch and Danilov (2012), because it is more difficult to satisfy the stability requirement when the mesh is refined. So the parameters of the mEVP and the number of subcycles should be sufficiently large. Note that the same is also true for the standard EVP. The traditional practice of running it with relatively low subcycling numbers ($N_{EVP} = 120$ is commonly used) may lead to noise in Δ over places where it is sufficiently small.

The difference between the two VP solutions and mEVP500 is much smaller, and is largely concentrated at the front between the moving and nearly stopped ice. However, one sees that there is a basin-scale pattern in the velocity difference in the bottom left panel of Fig. 2, which is the indication of the lack of convergence of the VP solution over the area where ice is moving. Indeed, it almost disappears on increasing the number of Picard iterations (bottom right panel). Simultaneously we see the substantially improved agreement between the patterns of Δ in Fig. 3. The remaining discrepancy is due to errors in both, EVP500 and VP10p, solutions, eliminating it will require increasing the number of subcycling steps and iterations even further, and is not pursued. We conclude that mEVP and VP converge to each other if one takes care that both are sufficiently accurate. Reaching full agreement between mEVP and VP solutions is more difficult if the ice advection is on, because errors may accumulate in this case with time. Smaller values of Δ_{min} additionally complicate the issue. In the presence of advection, ice is pressed into the north-east corner of the mesh, piling up there. The western part of the basin becomes free of ice, so that there are two fronts *no ice – moving ice* and *moving ice – nearly stopped ice*. We concentrate on the differences in the north-east corner, errors along the fronts depend on minor details and are difficult to eliminate.

The results of case B are given in Fig. 4 and 5 which present h_{ice} and Δ respectively after 1 month of model time. Here we compare three VP solutions with the mEVP500 reference simulation. We checked that increasing α and β to 1000 with subsequent increase of N_{EVP} to 2000 in mEVP does only small changes to the field of Δ compared to those seen for the VP solutions. The solution labelled VPb is obtained with the basic algorithm ($N_p = 2$), and VP10p and VP25p correspond

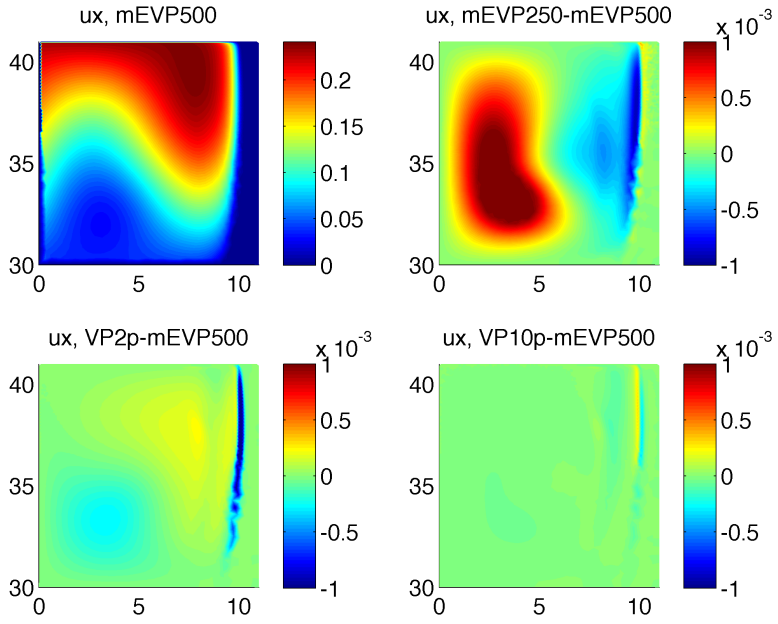

Figure 2. Ice zonal velocity (m/s) in case A (advection is switched off) after one month of simulations in mEVP500 (top, left) and differences between the solutions obtained by different methods: mEVP250-mEVP500 (top right), VP2p-mEVP500 (bottom left) and VP10p-mEVP500 (bottom right). mEVP250 does not converge, VP2p is closer to convergence, but still with noticeable errors. Additional Picard iterations in PV10p substantially reduced differences between the mEVP and VP solutions.

to using 10 and 25 additional Picard iterations respectively. While the difference in ice thicknesses remains small and is only slightly affected by the number of iterations in the VP solutions (patchiness in the difference panels of Fig. 4 is due to the finite accuracy of output), there is substantial improvement in the correspondence between the mEVP and VP solutions for Δ as the number of Picard iterations is increased. The fact that the differences in the ice thickness in the north-east corner stagnate hints that they evolved from some minor implementation details. Since the total ice volume is conserved, these errors are connected to those in the front position. They are rather small to be of practical importance.

Finally, case C (Fig 6, 7) shows that reaching agreement between the mEVP and VP runs for a much smaller Δ_{min} requires even larger number of Picard iterations (and also more subcycling in mEVP, although the improvements seen are less substantial). The mEVP500 solution in this case contains some noise in Δ, and is replaced by mEVP1000 obtained with $\alpha = \beta = 1000$ and $N_{EVP} = 2000$. We also consider the standard VPb solution and the solutions obtained with 100 (VP100p) and 200 (VP200p) Picard iterations. As in case B, the Picard iterations do not change the difference between ice volumes very much, but have substantial impact on the field of Δ. Similarly,

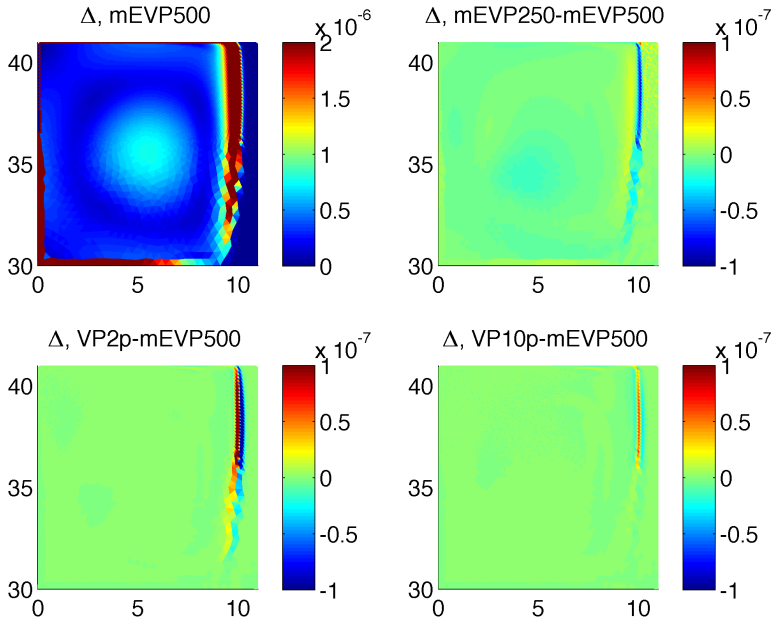

Figure 3. Same as in Fig. 2 but for the 'divergence' Δ (1/s) after one month of simulations. Additional Picard iterations in the PV method lead to very good agreement between mEVP and VP solutions.

VPb produces a stronger ice (smaller Δ) in the north-west corner, which is partly made weaker by increased number of Picard iterations. Of particular interest is the structure in the compression zone of VP solutions, which is sensitive to the number of iterations. There is some sensitivity of band structure to the change of solver tolerance and time step. This hints that one deals here either with incomplete convergence or some internal instabilities in the iterative procedure, a question we postpone for the future. We see that it is much more difficult to minimize the difference between mEVP and VP solutions if Δ_{min} is taken smaller.

Since the intention of Δ_{min} is to provide regularization, it should not be made excessively small unless there is motivation for that.

The next pair of figures (Fig. 8, 9) compares the performance of EVP and mEVP solvers. We use $\Delta_{min} = 2 \times 10^{-9}$ s^{-1}, and three EVP solutions denoted EVP3_100 ($\Delta t/T = 3$, $N_{EVP} = 100$), EVP3_500 (N_{EVP}=500) and a special solution, EVP4_1000 ($\Delta t/T = 4$, $N_{EVP} = 1000$), obtained by removing e^2 from the second terms on the left hand side of (11,12) and putting it to the denominator of the right hand side. After this manipulation EVP becomes almost identical to mEVP (all components of the stress tensor decay to the VP limit at the same rate), except for the differences in the velocity time stepping. In this case one may identify α with $2TN_{EVP}/\Delta t$. Solution EVP3_100, with parameters typical for large-scale applications, shows noisy Δ over the area with compressed ice. Increasing the number of subcycle steps improves the agreement (Fig. 9, bottom left), but it

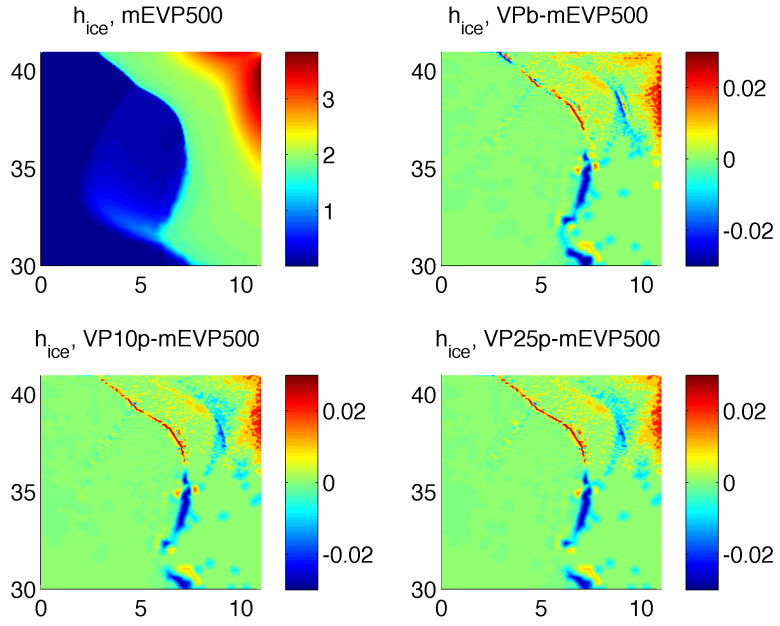

Figure 4. Ice thickness h_{ice} (m) after one month of simulations in case B in mEVP500 (top, left) and the differences between solutions obtained by different methods: VPb-mEVP500 (top right), VP10p-mEVP500 (bottom left) and VP25p-mEVP500 (bottom right). Additional Picard iterations in the VP method only slightly affect the differences.

still remains noisy. The noise takes the form of a wave structure. Simulations with further increased N_{EVP} (1000 and 2000, not shown) improve the agreement, but only slightly. Similarly, varying T is of no avail. However, the situation improves dramatically if the decay rates for stresses in (10-12) are made similar, as indicated by the bottom right panel in Fig. 9. The noise disappears. While the remaining discrepancy in Δ over the stiff ice can be further reduced, some differences will persist because of the different treatment of the momentum equation. The central circular spot over weak ice is common to all three solutions. Here the contribution from rheology is not dominant, and the difference is entirely due to the time stepping of momentum equation. We therefore conclude that it is the difference in the damping rates in the equations for stresses (10-12) in the standard EVP which is the main factor responsible for the noise seen in the field of Δ. More detailed analysis of this statement is needed. If we now turn to the patterns of ice thickness, we see that even in EVP3_100 and EVP3_500 with noisy Δ the simulated mean ice thickness agrees rather well with the mEVP solution, with differences of about 10 cm at maximum. The difference virtually disappears for the special case of EVP4_1000.

In summary, given the sensitivity of the field of Δ to the solution procedure, one should be cautious to discuss its detail unless the convergence has been tested. Judged from this perspective, the VP and

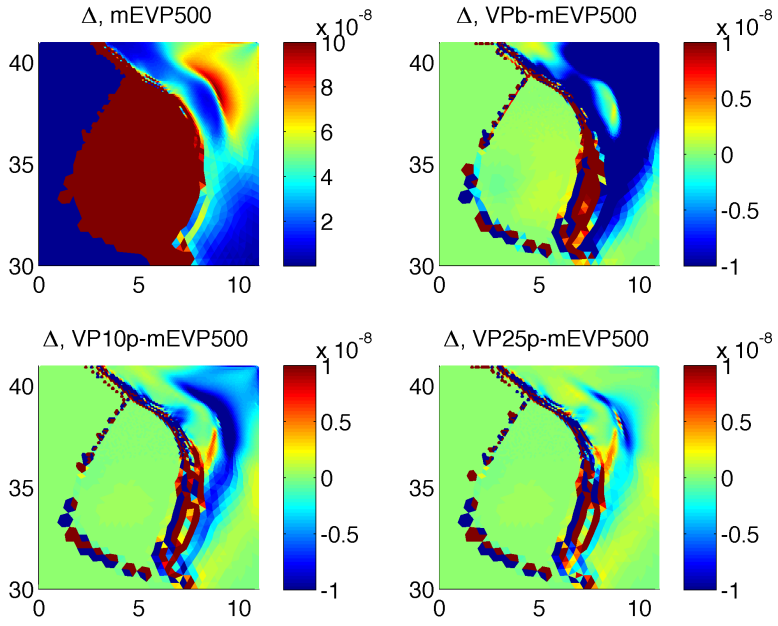

Figure 5. Δ (1/s) after one month of simulations in case B in mEVP500 (top, left) and differences between the solutions: VPb-mEVP500 (top right), VP10p-mEVP500 (bottom left) and VP25p-mEVP500 (bottom right). Additional Picard iterations in the VP method lead to substantially reduced differences between the solutions in the north-east corner. VPb reproduces a much stronger ice (smaller Δ), but additional Picard iterations make it weaker and closer to mEVP500.

mEVP approaches provide more consistent behavior than the EVP. However, even with them, one should realize that there might be some sensitivity to the implementation detail. For example, the VP solutions discussed here have been obtained with a tolerance of 10^{-8} in the PETSc solver; using a tolerance of 10^{-6} leads to changes in Δ comparable in magnitude to the effect of varying the number of Picard iterations. We have not seen benefits from making the tolerance even smaller (10^{-10}), but this may change in other applications. Additionally, there is some sensitivity to the time step interval Δt. Finally, the lack of full agreement in the pattern of Δ in VP and mEVP simulations, especially for the low $\Delta_{min} = 2 \times 10^{-11}$ s^{-1} in case C, can partly be due to the particular implicit/explicit splitting of the stresses, and we cannot exclude that the original splitting of Zhang and Hibler (1997) will converge somewhat differently. Note that the mEVP method shows less sensitivity to details than the VP method if α and β are sufficiently large to ensure the absence of noise in the solutions, and if N_{EVP} is sufficient for convergence.

The ice mean thickness and concentration, in contrast, show a much more robust behavior, and are much more consistent, even in the presence of noise in Δ. Still, the presence of noise pushes simulations on a dangerous ground and should be avoided. In many practical cases the VP, mEVP or

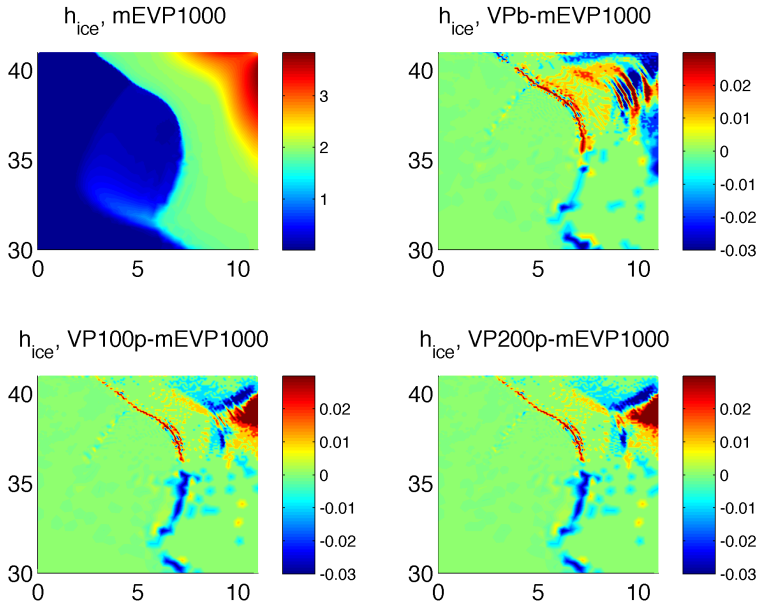

Figure 6. Ice thickness (m) after one month of simulations in case C ($\Delta_{min} = 2 \times 10^{-11}$ s^{-1} in mEVP1000 (top, left) and differences between the solutions obtained by different methods: VPb-mEVP1000 (top right), VP100p-mEVP1000 (bottom left) and VP200p-mEVP1000 (bottom right). The differences are small and additional Picard iterations only slightly change them.

EVP solvers will be run in a 'partially converging mode' when accuracy is achieved over a number of steps under conditions that forcing does not change much over a time step. Numerical stability and lack of noise (for the EVP and mEVP methods) will remain an issue to pay attention to.

5 Discussion

5.1 Numerical aspects: spatial discretization

The finite-element discretization of sea-ice dynamics employed by FESIM works in a robust way on unstructured triangular meshes. We now discuss how it relates to other unstructured-mesh discretizations proposed in the literature.

We first note that the FE $P_1 - P_1$ implementation can easily be cast in a FV form as explained in the Appendix. As concerns the purely dynamical (momentum) part, there is almost no difference in the final result to the FE discretization because of the lumping of the mass matrices we use for dynamics. One may wish to select a transport scheme that differs from FE-FCT, but the only motivation behind this can be the availability of a a more accurate and efficient FV scheme. Our experience is that reaching the accuracy of the FE-FCT scheme would require a better than third-order method in the

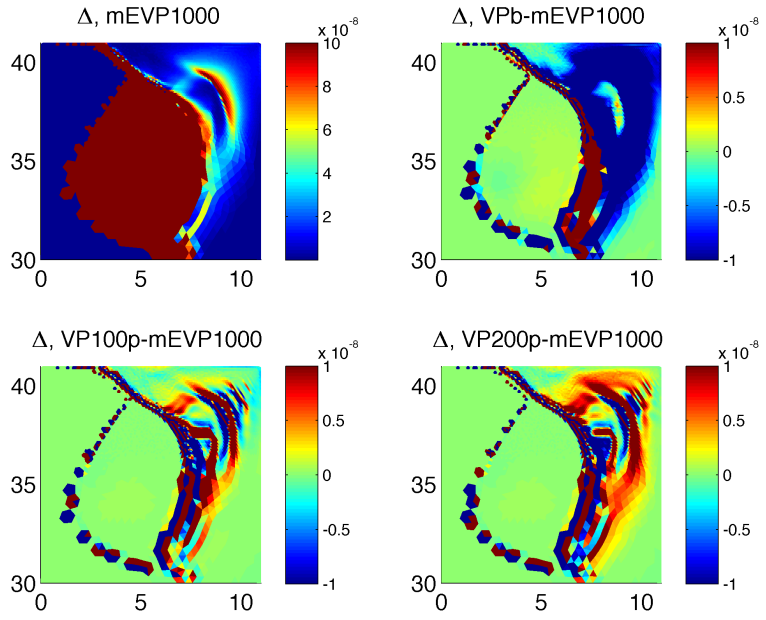

Figure 7. Δ (1/s) after one month of simulations in case C in mEVP1000 (top, left) and differences between the solutions: VPb-mEVP500 (top right), VP100p-mEVP1000 (bottom left) and VP200p-mEVP1000 (bottom right). Additional Picard iterations in the VP method substantially modify the differences, reducing them in the north-east corner. The convergence is not reached even for 200 Picard iterations.

respective FV FCT algorithm. As mentioned above, the presence of a consistent mass matrix in the FE transport equation efficiently compensates for a significant part of dispersion, which explains its good performance.

The vertex placement of variables we used is an analogue of A-grid in the traditional (Arakawa) terminology. A different A-grid implementation with the cell (triangle centroid) placement of variables was proposed by Hutchings et al. (2004). The discretization is straightforward if the FV approach is used and if the velocity derivatives on each triangle are computed by, e. g., the least square fit using the velocities on this and three neighboring triangles. The potential problem of the cell-based placement is a somewhat unfavorable stencil used in the computation of stress divergence. Indeed, it involves not only the nearest neighbors, but the neighbors of neighbors. We therefore consider the vertex placement of variables to be an easier choice.

The implementation adopted by FVCOM (Gao et al. (2011)) is also a FV one, with velocities placed at cells and scalars at vertices. We tested this variable placement while developing the sea-ice model to complement the ocean circulation model based on the staggered cell-vertex discretization. Because of an excessively large velocity space (the number of triangles is approximately twice that of vertices) it is prone to noise in velocities along the ice edge and was therefore abandoned in favor

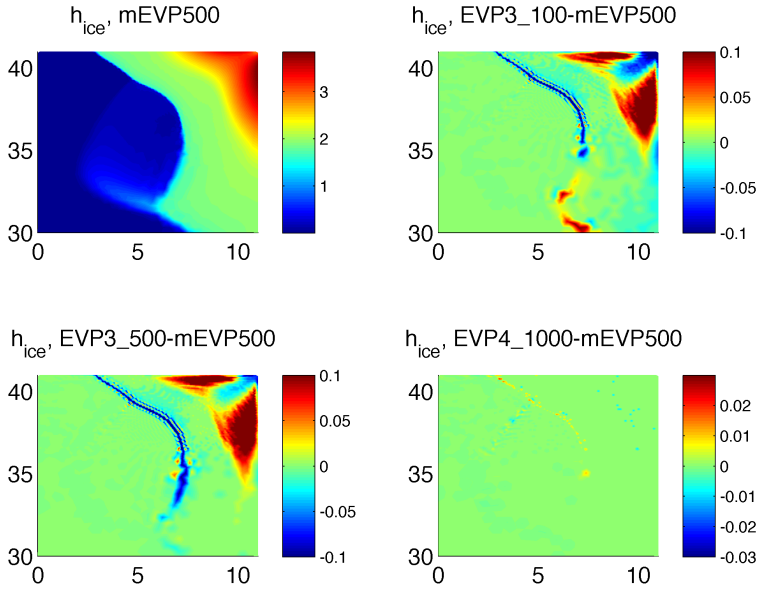

Figure 8. Ice mean thickness h_{ice} (m) after one month of simulations in mEVP500 (top, left) and differences between it and EVP solutions: EVP3_100-mEVP500 (top right), EVP3_500-mEVP500 (bottom left) and EVP4_1000p-mEVP500 (bottom right). The last EVP solution (bottom right), obtained with modified equations for stresses, shows the results almost identical to mEVP (see the text for details).

of the vertex–vertex scheme. Once again the vertex placement of velocities and scalars seems to be a more robust option.

Finally, the discretization proposed by Lietaer et al. (2008) is a FE one, based on nonconforming linear functions to represent the velocity vectors, with velocity degrees of freedom placed at the edges, and elementwise-constant scalars. It also has a too large velocity space, and is not optimal in this respect. Additionally, placing scalars at centers would be suboptimal for representing the ice strength gradients: a nonconforming linear function used for velocity spans only two elements with a common edge, and two ice strength values at centroids give only one component of the gradient.

Thus, despite its simplicity the discretization in FESIM deserves attention as a balanced choice. Work is planned on augmenting it with a multi-category ice functionality.

5.2 Numerical aspects: VP/EVP convergence

There is ongoing discussion on the convergence of traditional implementations of VP and EVP, with indications that convergence is lacking (see, e. g., Lemieux and Tremblay (2009), Lemieux et al. (2012)). It partly motivated the development of new approaches such as the Jacobian-free Newton–Krylov solver (see Lemieux et al. (2012)), which intends to improve the convergence of the VP

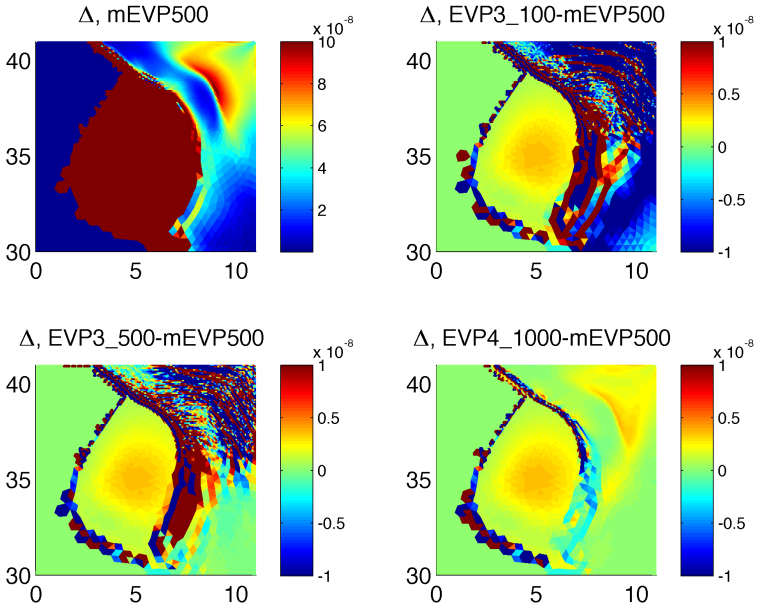

Figure 9. Same as Fig. 8 but for Δ (s^{-1}). Only the special solution obtained with the same decay rates in equations for stresses (bottom right) compares well to the mEVP solution.

method, but is too CPU-demanding, and also served as a motivation behind the new formulation of EVP in Bouillon et al. (2013), referred to as mEVP here. However Bouillon et al. (2013) mention that they fail to reach converging mEVP solutions. The analysis of Kimmritz et al. (2014) shows that mEVP does provide converging solutions, but only when α and β are sufficiently large and N_{EVP} is larger than any of them. From the theoretical viewpoint the mEVP and VP methods should lead to identical solutions if converged, and the solutions obtained with EVP may slightly deviate from them. The box test cases above illustrate that the solutions can be made rather close, but reaching full agreement between them might be too expensive computationally and require adjusting minor details of the algorithms.

The stability (and convergence as a result) of (m)EVP solvers is sensitive to the mesh size, and will generally deteriorate if the mesh is refined. Larger α, β, N_{EVP} are to be expected on finer meshes, and it is user's responsibility to select values providing the absence of noise in the fields of divergence and Δ. Note that the issues mentioned here are in full measure relevant for other models, including those formulated on structured meshes. While in realistic applications they can be hidden behind much larger uncertainties in parameterizations of mechanical and thermodynamical forcing, one should be sure that the dynamical operators the model relies on behave in a predictable and understandable way.

5.3 Practical aspects: CPU load

Computations of stresses and their contributions to the rhs of momentum equation are rather expensive in models formulated on unstructured meshes (compared to their structured-mesh counterparts) mainly because of the lack of directional splitting and, in the case presented, also because the number of triangles is twice as large as the number of scalar degrees of freedom. For this reason, one computation of the rhs (done N_{EVP} times per external time step in EVP and mEVP solvers) is substantially more expensive than one matrix-vector multiplication in the iterative matrix solver in the VP method. On the other hand, the number of iterations needed to reach convergence to the specified tolerance may depend on the ice distribution and domain geometry and the number of required Picard iterations can be high. One has to take into account the time spent on assembling the stiffness matrix and preconditioning it. Any comparison is even more complicated because full convergence of mEVP and VP methods will not necessarily be attempted in practice. For this reason no general recommendation can be given here. Just for information, we present the results for the box test case above: the time step of mEVP500 with N_{EVP}=1000 takes 0.55 s on 8 cores of old IBM BladeCenter JS22, to be compared with 0.88 s for VP25p and only 0.065 s for VPb, and there is approximately linear dependence on N_{EVP} and the number of Picard iterations N_p. Since VPb (N_p=2) provides a very reasonable solution for the ice mean thickness, and since the field of Δ, despite the lack of convergence, is smooth in this case, it can still be used and will be a faster option than mEVP500 with N_{EVP}=1000. They will be close to each other if we run mEVP500 with N_{EVP}=120, sacrificing convergence but keeping stability. As mentioned, the comparison in realistic global configuration is the subject of future work.

6 Code availability

The code of the model can be obtained on request from the first author (sergey.danilov@awi). It has also been uploaded as a supplement to this paper.

7 Conclusions

FESIM, the sea-ice component of FESOM v.1.4, is described here. We focus on the dynamical part of the model in this documentation. The new EVP solver (mEVP) proposed by Bouillon et al. (2013) leads to solutions approaching those of the VP solver if both are run toward convergence. However, it is expected that some differences between their results would still persist in practical usage. While the mEVP (13-16) algorithm shows better stability in our tests than the standard EVP algorithm (49-55), the performance of mEVP and VP is rather similar, and the CPU efficiency becomes the criterion to select between them. The mEVP method can be more convenient on massive parallel computers. As concerns the unstructured character of meshes, the implementation based on linear continuous

elements is perhaps the easiest among the other possible choices. It shows robust behavior and serves well the tasks of multi-resolution modeling, as indicated by a growing list of practical applications using FESOM. An important issue to be kept in mind with respect to multi-resolution simulations is the sensitivity of stability and hence convergence to the mesh resolution.

Appendix A: Finite-volume formulation

The finite-element implementation described above can be recast in a finite-volume form, as briefly described below.

In a FV implementation one deals with median-dual cells formed around vertices. They are formed by connecting mid-edges with centroids of mesh cells. The area of the median-dual cell associated to vertex v is the sum $A_v = \sum_{c=c(v)} A_c/3$, which coincides with the respective diagonal entry of the lumped mass matrix ($c(v)$ is the list of cells (triangles) containing vertex v). Since the force $\boldsymbol{F}$ is given by the divergence of stresses, by integrating it over A_v one gets

$$\int_v F_i dS = \sum_{e=e(v)} [(n_j \sigma_{ij} l)_l + (n_j \sigma_{ij} l)_r]_e, \tag{A1}$$

where the notation $e(v)$ implies the list of edges emanating from vertex v, the indices i, j denote directions, the subscripts l and r denote the left and right segments of the boundary around cell v which is associated to edge e (they connect the mid-edge point to the centroids of cells on both sides of the edge), l is the length of the respective segment and $\boldsymbol{n}$ is the outer normal. The stresses σ_{ij} are constant on triangles, so the computations with the last formula are straightforward, but involve a cycle over edges instead of that over elements in the FE implementation. The contribution from the elevation gradient is computed by averaging the gradients on triangles

$$\int_v \nabla H dS = \sum_{c=c(v)} (\nabla H)_c A_c/3. \tag{A2}$$

Note that the gradient computed by the last formula will be slightly different from its true FV counterpart in the spherical geometry. The latter can be recovered by using

$$\int_v \nabla H dS = \sum_{e=e(v)} [(\boldsymbol{n} H l)_l + (\boldsymbol{n} H l)_r]_e, \tag{A3}$$

with H estimated on edges as average over the vertices forming the edge.

The modifications of the transport scheme are as wel straightforward, but it is recommended to keep the consistent mass matrix of the FE case, which will augment the FV Lax–Wendroff scheme to the FE Taylor–Galerkin one. The FCT scheme in that case should follow the FE logics, because the mass matrix will mix the fluxes associated to boundaries. Other positivity preserving schemes are possible too, but have to be tested.

References

Bouillon, S., Fichefet, T., Legat, V., Madec, G., 2013. The elastic-viscous-plastic method revisited, Ocean Modelling 71, 2–12.

Budgell, W.P., Oliveira, A., Skogen, M.D., 2007. Scalar advection schemes for ocean modeling on unstructured triangular grids. Ocean Dyn. 57, 339–361.

Lietaer, O., Fichefet, T., Legat, V. 2008. The effects of resolving the Canadian Arctic Archipelago in a finite element sea ice model. Ocean Modelling, 24, 140–152.

Losch, M., Danilov, S., 2012. On solving the momentum equations of dynamic sea ice models with implicit solvers and the elastic–viscous–plastic technique. Ocean Modelling 41, 42–52.

Hibler, W. D., 1979. A dynamic thermodynamic sea ice model. J. Phys. Oceanogr., 9, 817–846.

Hutchings, J. K., Jasak, H., and Laxon, S. W., A strength implicit correction scheme for the viscous-plastic sea ice model, Ocean Modell., 7, 111–133, 2004.

Hunke, E. C., Viscous-plastic sea ice dynamics with the EVP model: Linearization issues, J. Comp. Phys., 170, 18–38, 2001.

Hunke, E. C., and Dukowicz, J. K., An elastic-Viscous-Plastic model for sea ice dynamics, J. Phys. Oceanogr., 27, 1849–1867, 1997.

Hunke, E.C., Lipscomb, W.H., 2008. Cice: the Los Alamos sea ice model documentation and software user's manual. Tech. Rep., T-3 Fluid Dynamics Group, Los Alamos National Laboratory, Los Alamos NM 87545.

Gao, G., Chen, C., Qi, J., Beardsley, R. C., 2011. An unstructured-grid, finite-volume sea ice model: Development, validation, and application. J. Geophys. Res., 116, doi:10.1029/2010JC006688.

Kimmritz, M., Danilov, S., Losch, M., 2014. On the convergence of EVP solvers. To be submitted.

Large, W.G., Yeager, S.G., 2009. The global climatology of an interannually varying air-sea flux data set. Climate Dynamics 33, 341–364.

Lemieux, J.-F., Knoll, D., Tremblay, B., Holland, D., Losch, M., 2012. A comparison of the Jacobian-free Newton–Krylov method and the EVP model for solving the sea ice momentum equation with a viscous-plastic formulation: a serial algorithm study. Journal of Computational Physics 231 (17), 5926–5944.

Lemieux, J.-F., Tremblay, B., 2009. Numerical convergence of viscous–plastic sea ice models. Journal of Geophysical Research 114, C05009.

Löhner, R., Morgan, K., Peraire, J., and Vahdati, M.: Finite-element flux-corrected transport (FEM-FCT) for the Euler and Navier–Stokes equations, Internat. J. Numer. Methods Fluids, 7, 1093—1109, 1987.

Owens W.B. Lemke, P., 1990 Sensitivity studies with a sea ice-mixed layer-picnocline modelin the Weddell Sea, J. Geophys. Res., 95, 9527–9538.

Parkinson, C. L., Washington, W.M., 1979. A large-scale numerical model of sea ice. J. Geophys. Res., 84, 311–337.

Sidorenko, D., Wang, Q., Danilov, S., Schröter, J., 2011, FESOM under Coordinated Ocean-ice Reference Experiment forcing, Ocean Dynamics, doi:10.1007/s10236-011-0406-7

Sidorenko, D., Rackow, T., Jung, T., Semmler, T., Barbi, D., Danilov, S., Dethloff, D., Dorn, W., Fieg, K., Goessling, H. F., Handorf, D., Harig, S., Hiller, W., Juricke, S, Losch, M., Schröter, J., Sein, D. V., Wang, Q., 2014. Towards multi-resolution global climate modeling with ECHAM6-FESOM. Part I: model formulation and mean climate. Clim. Dyn., DOI 10.1007/s00382-014-2290-6.

Timmermann, R., Danilov, S., Schröter, J., Böning, C. , Sidorenko, D., Rollenhagen, K., 2009. Ocean circulation and sea ice distribution in a finite element global sea ice – ocean model, Ocean Modell., doi:10.1016/j.ocemod.2008.10.009.

Timmermann, R., Hellmer, H. H., 2013. Southern Ocean warming and increased ice shelf basal melting in the 21st and 22nd centuries based on coupled ice-ocean finite-element modelling, Ocean Dynamics, 63(9), 1011-1026, doi:10.1007/s10236-013-0642-0.

Wang, X., Wang, Q., Sidorenko, D., Danilov, S., Schröter, J., Jung, T. (2012) Long-term ocean simulations in FESOM: evaluation and application in studying the impact of Greenland Ice Sheet melting, Ocean Dynamics. doi:10.1007/s10236-012-0572-2

Wang, Q., Danilov, S., Sidorenko, D., Timmermann, R., Wekerle, C., Wang, X., Jung, T., Schröter, J., 2014. The Finite Element Sea Ice-Ocean Model (FESOM) v.1.4: formulation of an ocean general circulation model, Geoscientific Model Development, 7 (2), 663–693, doi:10.5194/gmd-7-663-2014.

Wekerle, C., Wang, Q., Danilov, S., Jung, T., Schröter, J., 2013, The Canadian Arctic Archipelago throughflow in a multiresolution global model: Model assessment and the driving mechanism of interannual variability, J. Geophys. Res.: Oceans, doi:10.1002/jgrc.20330.

Zhang, J., Hibler, W.D, On an efficient numerical method for modeling sea ice dynamics, J. Geophys. Res., 102, 8691–8702, 1997.

Zienkiewicz, O.C., Taylor, R. L., 2000. The Finite Element Method, Fifth Edition, V. 3: Fluid Dynamics, Oxford: Butterworth-Heinemann.