Journal of Computational Physics 229 (2010) 2840–2852

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Improving the numerical convergence of viscous-plastic sea ice models with the Jacobian-free Newton–Krylov method

Jean-François Lemieux [a,*,1], Bruno Tremblay [a], Jan Sedláček [b], Paul Tupper [c], Stephen Thomas [d], David Huard [a], Jean-Pierre Auclair [e]

[a] Department of Atmospheric and Oceanic Sciences, McGill University, 805 Sherbrooke Street West, Montréal, QC, Canada H3A 2K6
[b] Institute for Atmospheric and Climate Science, ETH Zurich, Universitaetstrasse 16, Zurich CH-8092, Switzerland
[c] Department of Mathematics, Simon Fraser University, 8888 University Drive, Burnaby, BC, Canada V5A 1S6
[d] Acceleware Corp., 1600 37th Street SW, Calgary, AB, Canada T3C 3P1
[e] Department of Oceanography, Dalhousie University, 1355 Oxford Street, Halifax, NS, Canada B3H 4J1

ARTICLE INFO

Article history:
Received 19 August 2009
Received in revised form 27 November 2009
Accepted 8 December 2009
Available online 16 December 2009

Keywords:
Sea ice
Viscous-plastic rheology
Newton–Krylov method
Numerical convergence

ABSTRACT

We have implemented the Jacobian-free Newton–Krylov (JFNK) method to solve the sea ice momentum equation with a viscous-plastic (VP) formulation. The JFNK method has many advantages: the system matrix (the Jacobian) does not need to be formed and stored, the method is parallelizable and the convergence can be nearly quadratic in the vicinity of the solution. The convergence rate of our JFNK implementation is characterized by two phases: an initial phase with slow convergence and a fast phase for which the residual norm decreases significantly from one Newton iteration to the next. Because of this fast phase, the computational gain of the JFNK method over the standard solver used in existing VP models increases with the required drop in the residual norm (termination criterion). The JFNK method is between 3 and 6.6 times faster (depending on the spatial resolution and termination criterion) than the standard solver using a preconditioned generalized minimum residual method. Resolutions tested in this study are 80, 40, 20 and 10 km. For a large required drop in the residual norm, both JFNK and standard solvers sometimes do not converge. The failure rate for both solvers increases as the grid is refined but stays relatively small (less than 2.3% of failures). With increasing spatial resolution, the velocity gradients (sea ice deformations) get more and more important. Nonlinear solvers such as the JFNK method tend to have difficulties when there are such sharp structures in the solution. This lack of robustness of both solvers is however a debatable problem as it mostly occurs for large required drops in the residual norm. Furthermore, when it occurs, it usually affects only a few grid cells, i.e., the residual is small for all the velocity components except in very localized regions. Globalization approaches for the JFNK solver, such as the line search method, have not yet proven to be successful. Further investigation is needed.

1. Introduction

Sea ice dynamics plays a key role in shaping the Arctic and Antarctic sea ice covers. Indeed, it strongly affects the sea ice thickness field which then influences the exchange of heat and momentum between the polar oceans and the atmosphere.

* Corresponding author.
E-mail address: lemieux@cims.nyu.edu (J.-F. Lemieux).
[1] Now at the Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012-1185, USA.

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.12.011

Crucial to the model representation of sea ice dynamics is the rheology, i.e., the relationship between applied stresses and the resulting deformations. Owing to their very sporadic behavior, ridging and lead formation events, respectively associated with convergence and divergence, suggest plasticity: as if critical stresses must be reached before these deformations can occur [1]. Most of the sea ice dynamic models used today are based on a viscous-plastic (VP) rheology. For example, among the sea ice models (coupled to general circulation models) participating in the Intergovernmental Panel on Climate Change (IPCC) fourth assessment, 4 have no dynamics or a very crude representation of the rheology, 4 use the cavitating fluid model (i.e., without shear strength) [2], and the remaining 15 models are physically based on a viscous-plastic rheology [3].

A very common nonlinear solver used for VP sea ice models, referred to as the standard solver in this paper, is based on an implicit solution of a linearized system of equations and an outer loop (OL) iteration (two OL iterations = one pseudo time step as introduced by [4]). The linearized system of equations formed at each OL iteration is solved by a linear solver such as a successive overrelaxation (SOR) method [3], a line SOR (LSOR, [4]) or the preconditioned generalized minimum residual (GMRES) method [5].

In a recent paper, Lemieux and Tremblay [6] investigated the numerical convergence of VP models. They showed that the numerical convergence of the VP model standard solver is very slow, i.e., the errors on the calculated approximate velocity field decrease very slowly with the number of OL iterations. This slow convergence is an issue at all spatial resolution but gets more severe as the grid is refined. Lemieux and Tremblay [6] pointed out that the sea ice momentum equation with a VP formulation is intrinsically difficult to solve because it is highly nonlinear. Indeed, sea ice has very little tensile strength while it can resist large stresses before yielding in compression. Furthermore, they argued that the convergence properties could be improved by having a continuously differentiable momentum equation. The sea ice momentum equation with a VP formulation is indeed not continuously differentiable because of a capping of the viscous coefficients. To get a continuously differentiable momentum equation, Lemieux and Tremblay [6] have replaced the formulation of the viscous coefficients with capping [3] by a hyperbolic tangent function. They also argued (without implementing it), that the convergence properties of VP models could be improved by using the Newton method instead of the standard nonlinear solver which is based on a simpler linearization approach.

We introduce a nonlinear solver that is based on the Newton method. The Newton method for solving a nonlinear system of equations $\mathbf{F}(\mathbf{u}) = 0$ is based on a multivariate Taylor expansion around an approximate solution $\mathbf{u}^k$ (where the superscript k indicates the iteration number). Theoretically, $\mathbf{u}^k$ converges quadratically to the solution $\mathbf{u}$ if $\mathbf{u}^k$ is sufficiently close to $\mathbf{u}$ and if $\mathbf{F}'$, the Jacobian, is Lipschitz continuous at $\mathbf{u}$ [7]. This is however one drawback of the Newton method: the convergence is quadratic in the vicinity of the solution (local convergence) but obtaining the solution from any initial iterate (global convergence) can be challenging for some problems. In these cases, a globalization approach is needed.

Because of the complexity of the sea ice momentum equation with a VP formulation, finding the Jacobian is a difficult task that is prone to errors. For this reason, we adopt a Jacobian-free approach: the Jacobian-free Newton–Krylov (JFNK) method [8]. The JFNK method has many advantages: the system matrix (the Jacobian) does not need to be formed and stored, the method is parallelizable and the convergence can be nearly quadratic in the vicinity of the solution. Because of these advantages, the JFNK method has been applied to many different nonlinear problems such as hurricane dynamics [9], solar magnetic field calculations [10] and simulations of chemical reactions [11].

In this paper, we also explore ways to improve convergence properties of the existing VP solver. We first revisit the hyperbolic tangent approach for the viscous coefficients proposed by Lemieux and Tremblay [6]. Globalization techniques for both standard and JFNK solvers are also discussed.

The elastic-viscous-plastic (EVP) model developed by Hunke and Dukowicz [12] is based on a fully explicit time-stepping scheme. With this approach, there is an extra elastic term in the constitutive equation which allows for a larger sub-time step using a fully explicit time-stepping scheme. This elastic term does not represent the physical elastic stress. It is an artificial term introduced in order to relax the stability condition with a fully explicit time-stepping scheme. The comparison of the convergence properties of the EVP model to the ones of the JFNK solver is kept for future work.

This paper is structured as follows. Section 2 describes the sea ice momentum equation with a VP formulation. The continuously differentiable formulation for the viscous coefficients, introduced by Lemieux and Tremblay [6] is also given in this section. In Section 3, the discretization of the momentum equation is presented and a comparison between the standard solver and the JFNK method is made. The algorithms for both methods are given in the same section. In Section 4, we describe the model and the forcing fields used for the simulations. Section 5 shows the impact of using the continuously differentiable formulation for the viscous coefficients on the convergence properties of the standard solver. This section also provides a comparison of the computational efficiency of the JFNK method and the one of the standard solver. Global convergence issues are discussed in this section. Concluding remarks and a description of future work are given in Section 6.

2. Sea ice momentum equation

A two-dimensional treatment is usually employed to model the dynamic of sea ice. This is justified on the basis of the large ratio of the horizontal and the vertical scales O(1000 km/10 m) = O(10^5) [1]. The two-dimensional sea ice momentum equation is given by

$$\rho h \frac{D\mathbf{u}}{Dt} = -\rho h f \mathbf{k} \times \mathbf{u} + \boldsymbol{\tau}_a - \boldsymbol{\tau}_w + \nabla \cdot \boldsymbol{\sigma} - \rho h g \nabla H_d, \tag{1}$$

where ρ is the density of the ice, h the sea ice thickness, f the Coriolis parameter, $\mathbf{u} = u\mathbf{i} + v\mathbf{j}$ the horizontal sea ice velocity vector, $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are the unit vectors aligned respectively with the x, y and z axis, τ_a is the wind stress, τ_w the water drag, $\boldsymbol{\sigma}$ the internal ice stress tensor ($\nabla \cdot \boldsymbol{\sigma}$ is defined as the rheology term), g the gravity and H_d the sea surface height. Following Tremblay and Mysak [13], the sea surface tilt is expressed in terms of the geostrophic ocean current. Using a simple quadratic law with a constant turning angle, τ_a and τ_w are expressed as [14]

$$\boldsymbol{\tau}_\mathbf{a} = \rho_a C_{da} |\mathbf{u}_a^g| (\mathbf{u}_a^g \cos\theta_a + \mathbf{k} \times \mathbf{u}_a^g \sin\theta_a), \tag{2}$$

$$\boldsymbol{\tau}_\mathbf{w} = C_w [(\mathbf{u} - \mathbf{u}_w^g) \cos\theta_w + \mathbf{k} \times (\mathbf{u} - \mathbf{u}_w^g) \sin\theta_w], \tag{3}$$

where $C_w = \rho_w C_{dw} |\mathbf{u} - \mathbf{u}_w^g|$, ρ_a and ρ_w are the air and water densities, C_{da} and C_{dw} are the air and water drag coefficients, and $\mathbf{u}_a^g$ and $\mathbf{u}_w^g$ are the geostrophic wind and ocean current. Because $\mathbf{u}$ is much smaller than $\mathbf{u}_a^g$, $\mathbf{u}$ has been neglected in the expression for the wind stress.

With a VP formulation, the constitutive law, that relates the stresses and the strain rates, can be written as [3]

$$\sigma_{ij} = 2\eta\dot{\epsilon}_{ij} + [\zeta - \eta]\dot{\epsilon}_{kk}\delta_{ij} - P\delta_{ij}/2, \quad i,j = 1,2, \tag{4}$$

where σ_{ij} are the internal ice stresses, δ_{ij} is the Kronecker delta, $\dot{\epsilon}_{ij}$ are the strain rates defined by $\dot{\epsilon}_{11} = \frac{\partial u}{\partial x}, \dot{\epsilon}_{22} = \frac{\partial v}{\partial y}$ and $\dot{\epsilon}_{12} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \dot{\epsilon}_{kk} = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \zeta$ is the bulk viscosity and η is the shear viscosity. Following Hibler [3], the pressure term P is parameterized by

$$P = P^* h \exp[-C(1 - A)], \tag{5}$$

where P^* is the ice strength parameter, A is the sea ice concentration and C is the ice concentration parameter, an empirical constant characterizing the dependence of the compressive strength on sea ice concentration.

The rheology term ($\nabla \cdot \boldsymbol{\sigma}$) depends on the yield curve and the flow rule, through the formulation of the bulk and shear viscosities. In the following, we use the elliptical yield curve with a normal flow rule [3]. In this case, the bulk and shear viscosities are given by

$$\zeta = \frac{P}{2\triangle}, \tag{6}$$

$$\eta = \zeta e^{-2}, \tag{7}$$

where $\triangle = [(\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2)(1 + e^{-2}) + 4e^{-2}\dot{\epsilon}_{12}^2 + 2\dot{\epsilon}_{11}\dot{\epsilon}_{22}(1 - e^{-2})]^{\frac{1}{2}}$, and e is the ratio of the long axis and the short axis of the elliptical yield curve.

In the limit where $\triangle$ tends to zero, Eqs. (6) and (7) become singular. To avoid this problem, the values of ζ and η are capped at maximum values of $\zeta_{max} = (2.5 \times 10^8)P$ and $\eta_{max} = \zeta_{max} e^{-2}$ in the original VP model [3]. Eqs. (6) and (7) therefore become

$$\zeta = min\left(\frac{P}{2\triangle}, \zeta_{max}\right), \tag{8}$$

$$\eta = min\left(\frac{P}{2e^2\triangle}, \eta_{max}\right). \tag{9}$$

This formulation of the viscous coefficients with capping leads to a rheology term that is not continuously differentiable with respect to the velocity. To obtain a smooth formulation of the viscous coefficients, Lemieux and Tremblay [6] have replaced the expression of ζ (Eq. (8)) by

$$\zeta = \zeta_{max} \tanh\left(\frac{P}{2\triangle\zeta_{max}}\right). \tag{10}$$

As in Eq. (7), $\eta = \zeta e^{-2}$. Lemieux and Tremblay [6] have shown that this smooth formulation reduces the number of OL iterations required to reach full convergence by a factor of $\sim$2 with the standard solver.

3. Numerical implementations

3.1. Temporal and spatial discretizations

Following Zhang and Hibler [4], we neglect the advection of momentum which is a lot smaller than the other terms in the momentum equation. The u and v momentum equations can then be written as

$$-\rho h\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left[(\eta + \zeta)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[\eta\frac{\partial u}{\partial y}\right] + \frac{\partial}{\partial x}\left[(\zeta - \eta)\frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial y}\left[\eta\frac{\partial v}{\partial x}\right] + \rho h f v - C_w(u\cos\theta_w - v\sin\theta_w) = -C_w(u_w^g\cos\theta_w - v_w^g\sin\theta_w) + r_u, \tag{11}$$

$$-\rho h\frac{\partial v}{\partial t}+\frac{\partial}{\partial y}\left[(\eta+\zeta)\frac{\partial v}{\partial y}\right]+\frac{\partial}{\partial x}\left[\eta\frac{\partial v}{\partial x}\right]+\frac{\partial}{\partial y}\left[(\zeta-\eta)\frac{\partial u}{\partial x}\right]+\frac{\partial}{\partial x}\left[\eta\frac{\partial u}{\partial y}\right]$$
$$-\rho hfu-C_w(v\cos\theta_w+u\sin\theta_w)=-C_w(v_w^g\cos\theta_w+u_w^g\sin\theta_w)+r_v, \tag{12}$$

where r_u and r_v include all the terms that do not depend on u and v, i.e., the pressure gradient term, the sea surface tilt and the wind stress. Note that η, ζ and C_w are function of u and v.

Given the high values of viscosity in regions where small deformations (small values of $\triangle$) are present, the stability criterion for a fully explicit time-stepping scheme requires a time step on the order of a second for a 100-km resolution grid [15] and a hundredth of a second for a resolution of 10 km, a typical spatial resolution for current regional models. Because of this stringent stability condition, we solve the momentum equation implicitly.

Eqs. (11) and (12) are discretized in time by using a backward Euler differencing and discretized spatially by using centered finite differences. The components of the velocity (u and v) are positioned on the Arakawa C-grid (see Fig. 7(a) of [13] for details). We therefore solve the momentum equation at times $\ldots t-\Delta t, t, t+\Delta t, t+2\Delta t, \ldots$ where Δt is the time step. At time t, Eqs. (11) and (12) become

$$-\rho h\frac{u^t}{\Delta t}+\frac{\partial}{\partial x}\left[(\eta+\zeta)\frac{\partial u^t}{\partial x}\right]+\frac{\partial}{\partial y}\left[\eta\frac{\partial u^t}{\partial y}\right]+\frac{\partial}{\partial x}\left[(\zeta-\eta)\frac{\partial v^t}{\partial y}\right]+\frac{\partial}{\partial y}\left[\eta\frac{\partial v^t}{\partial x}\right]$$
$$+\rho hf\,v_{avg}^t-C_{uw}(u^t\cos\theta_w-v_{avg}^t\sin\theta_w)=-\rho h\frac{u^{t-\Delta t}}{\Delta t}-C_{uw}(u_w^g\cos\theta_w-v_{wavg}^g\sin\theta_w)+r_u, \tag{13}$$

$$-\rho h\frac{v^t}{\Delta t}+\frac{\partial}{\partial y}\left[(\eta+\zeta)\frac{\partial v^t}{\partial y}\right]+\frac{\partial}{\partial x}\left[\eta\frac{\partial v^t}{\partial x}\right]+\frac{\partial}{\partial y}\left[(\zeta-\eta)\frac{\partial u^t}{\partial x}\right]+\frac{\partial}{\partial x}\left[\eta\frac{\partial u^t}{\partial y}\right]$$
$$-\rho hfu_{avg}^t-C_{vw}(v^t\cos\theta_w+u_{avg}^t\sin\theta_w)=-\rho h\frac{v^{t-\Delta t}}{\Delta t}-C_{vw}(v_w^g\cos\theta_w+u_{wavg}^g\sin\theta_w)+r_v, \tag{14}$$

where the superscript indicates the time at which the u and v components are considered, v_{avg}^t is the spatial averaging of the four neighboring v components of velocity at the u location (and vice versa for u_{avg}^t). The same idea applies for u_{wavg}^g and v_{wavg}^g for the ocean currents. C_{uw} and C_{vw} are respectively C_w evaluated at the u and v locations. Note that C_{uw}, C_{vw}, η and ζ are function of u^t and v^t. r_u and r_v are evaluated at time t.With the spatial discretization, the derivative terms are evaluated at the u location for Eq. (13) and at the v location for Eq. (14).

We introduce b_u and b_v which are respectively the sum of all the terms on the right hand side of Eqs. (13) and (14). A f-plane approximation is used with $f=1.46\times10^{-4}\ \mathrm{s}^{-1}$. A Dirichlet boundary condition is applied at an ocean-land boundary ($\mathbf{u}=0$) and a Neumann condition at an open boundary (i.e., the spatial derivatives of the components of velocity in the normal direction with the open boundary are chosen to be zero). For stability, the pressure P is set to zero at the open boundaries [16]. Close to model boundaries, proper left and right difference schemes are used in the Taylor series expansion to evaluate the spatial derivatives.

3.2. The nonlinear system of equations

The number of grid cells in the domain is defined by the parameters nx and ny. Recall that u and v are not collocated on a C-grid. The components of velocity on the grid are u_{ij}^t where $i=1,nx+1$ and $j=1,ny$ and v_{ij}^t where $i=1,nx$ and $j=1,ny+1$. The spatial discretization leads to a system of n nonlinear equations where $n=(ny(nx+1)+nx(ny+1))$. Dropping the superscript t and defining $u_1=u_{11},u_2=u_{12},\ldots u_{ny(nx+1)+1}=v_{11},u_{ny(nx+1)+2}=v_{12},\ldots$, we form the vector $\mathbf{u}^T=[u_1,u_2,\ldots u_n]$. This means that the u_{ij} are stacked first followed by the v_{ij}.

We can then write in a compact form the nonlinear system of n equations resulting from the discretization of the momentum equation at time t as

$$\mathbf{F}(\mathbf{u})=\mathbf{A}(\mathbf{u})\mathbf{u}-\mathbf{b}(\mathbf{u})=0, \tag{15}$$

where $\mathbf{A}(\mathbf{u})$ is an $n\times n$ matrix function of $\mathbf{u}$. The vector $\mathbf{u}$ is the solution we are looking for. $\mathbf{A}(\mathbf{u})\mathbf{u}$ defines the left hand side of the set of discretized momentum equations (Eqs. (13) and (14)). Note that the spatial derivative terms in Eqs. (13) and (14) are symmetric operators. However, because of the Coriolis term and part of the water drag term, the matrix $\mathbf{A}(\mathbf{u})$ is not symmetric. The vector $\mathbf{b}(\mathbf{u})$ is formed as the vector $\mathbf{u}$, i.e., the b_u are stacked first followed by the b_v.

3.3. Comparison of the standard and JFNK solvers

Both standard and JFNK solvers are based on an outer loop (we call it a Newton loop for the JFNK method), a linearization and an implicit solution of a linearized system of equations. In other words, both methods solve the nonlinear system of equations by solving a succession of linearized systems of equations.

For the JFNK solver, the continuously differentiable formulation of the viscous coefficients is used. However, to study the effect of the continuously differentiable formulation on the computational properties of the standard solver, both approaches (continuous and discontinuous) are used with the standard solver. Note that the solution $\mathbf{u}$ is slightly different whether the

capping or the hyperbolic tangent formulation is used for the viscous coefficients. In order to compare the solvers in this section, both solvers are assumed to solve the same nonlinear system of equations (the continuously differentiable formulation). To explain the differences between the standard and JFNK solvers, we look at the linearized system of equations to be solved by both methods at iteration k (of the OL or Newton loop). In this paper, by linearization we refer to both the velocity field used to get the linearized system of equations and also the matrix form of this same linearized system of equations.

In the standard method, we want to solve for $\mathbf{u}^k$ the following linear system

$$\mathbf{A}(\mathbf{u}^{k-1})\mathbf{u}^k = \mathbf{b}(\mathbf{u}^{k-1}), \tag{16}$$

where the coefficients of the system matrix $\mathbf{A}(\mathbf{u}^{k-1})$ are formed from the previous iterate (or the initial iterate if $k = 1$). Note that Hibler and Ackley [17] found a splitting problem under certain free drift conditions when linearizing the system of equations with $\mathbf{u}^{k-1}$. They observed that the water drag terms were very different between the two steps (the two OL iterations) of their modified Euler procedure. We follow Hibler and Ackley [17] in the standard model (unless otherwise specified), and use the average of the past two iterates $\left(\frac{\mathbf{u}^{k-1}+\mathbf{u}^{k-2}}{2}\right)$ for the linearization in the simulations presented. However, for the sake of comparison between the standard solver and the JFNK method, we use $\mathbf{u}^{k-1}$ in the mathematical development of the standard approach presented in this subsection in a manner analogous to the JFNK method (see below). The kth iterate can hence be written as

$$\mathbf{u}^k = \mathbf{A}^{-1}(\mathbf{u}^{k-1})\mathbf{b}(\mathbf{u}^{k-1}). \tag{17}$$

Using $\mathbf{F}(\mathbf{u}^{k-1}) = \mathbf{A}(\mathbf{u}^{k-1})\mathbf{u}^{k-1} - \mathbf{b}(\mathbf{u}^{k-1})$, Eq. (17) becomes

$$\mathbf{u}^k = \mathbf{u}^{k-1} - \mathbf{A}^{-1}(\mathbf{u}^{k-1})\mathbf{F}(\mathbf{u}^{k-1}). \tag{18}$$

On the other hand, the Newton method for solving the nonlinear system of Eq. (15) is based on a multivariate Taylor expansion around a previous iterate:

$$\mathbf{F}(\mathbf{u}^{k-1} + \boldsymbol{\delta}\mathbf{u}^k) \approx \mathbf{F}(\mathbf{u}^{k-1}) + \mathbf{F}'(\mathbf{u}^{k-1})\boldsymbol{\delta}\mathbf{u}^k. \tag{19}$$

The higher order terms are neglected in the expression above. Setting $\mathbf{F}(\mathbf{u}^{k-1} + \boldsymbol{\delta}\mathbf{u}^k) = 0$, the correction $\boldsymbol{\delta}\mathbf{u}^k = \mathbf{u}^k - \mathbf{u}^{k-1}$ can be obtained by solving the linear system of n equations:

$$\mathbf{J}(\mathbf{u}^{k-1})\boldsymbol{\delta}\mathbf{u}^k = -\mathbf{F}(\mathbf{u}^{k-1}), \tag{20}$$

where the system matrix $\mathbf{J} \equiv \mathbf{F}'$ is the Jacobian, an $n \times n$ matrix whose entries are $J_{qr} = \partial F_q(\mathbf{u}^{k-1})/\partial(u_r^{k-1})$ (where $q = 1, n$ and $r = 1, n$). The next iterate can therefore be written as

$$\mathbf{u}^k = \mathbf{u}^{k-1} - \mathbf{J}^{-1}(\mathbf{u}^{k-1})\mathbf{F}(\mathbf{u}^{k-1}). \tag{21}$$

The matrix $\mathbf{J}$ can be divided into two matrices:

$$\mathbf{J}(\mathbf{u}^{k-1}) = \mathbf{A}(\mathbf{u}^{k-1}) + \mathbf{G}(\mathbf{u}^{k-1}), \tag{22}$$

where $\mathbf{A}(\mathbf{u}^{k-1})$ is the matrix of Eq. (15) linearized with $\mathbf{u}^{k-1}$ and $\mathbf{G}(\mathbf{u}^{k-1})$ is a matrix with coefficients made up of sums of terms like $u_r^{k-1}\partial a_{qr}(\mathbf{u}^{k-1})/\partial(u_r^{k-1})$ (the a_{qr} being the coefficients of the matrix $\mathbf{A}$) and to some terms related to the water drag. Eq. (21) can therefore be written as

$$\mathbf{u}^k = \mathbf{u}^{k-1} - (\mathbf{A}(\mathbf{u}^{k-1}) + \mathbf{G}(\mathbf{u}^{k-1}))^{-1}\mathbf{F}(\mathbf{u}^{k-1}). \tag{23}$$

Both methods are valid (Eqs. (18) and (21)). They are fixed point iteration methods [18]. Indeed, they can be written as $\mathbf{u}^k = \boldsymbol{\phi}(\mathbf{u}^{k-1})$ (with their respective $\boldsymbol{\phi}$). A comparison of Eqs. (18) and (23), shows that the matrix $\mathbf{G}(\mathbf{u}^{k-1})$ is neglected in the linearization process of the standard method. Lemieux and Tremblay [6] argue that the slow convergence rate of the standard solver is in part due to this simpler linearization approach (i.e., due to the omission of the matrix $\mathbf{G}$).

For both standard and JFNK solvers, we use the same linear solver: the preconditioned GMRES method [19,20], a Krylov subspace method. For both methods, all the terms in the momentum equation are treated implicitly. If we look at the "line" q coming from the multiplication of the matrix $\mathbf{A}(\mathbf{u}^{k-1})$ and the vector $\mathbf{u}^k$, we get the following sum: $a_{q1}u_1^k + \cdots + a_{qq}u_q^k + \cdots + a_{qr}u_r^k + \cdots + a_{qn}u_n^k$ where the superscript k for all the terms shows the implicit treatment. For example, treating the $a_{qr}u_r$ term explicitly would mean that it would be written as $a_{qr}u_r^{k-1}$ and would be sent to the right hand side to be included in the $\mathbf{b}$ vector.

In our case, the implicit treatment of the Coriolis term and the off-diagonal part of the water drag term leads to a non symmetric system matrix. This is not a problem because the GMRES method, as opposed to the conjugate gradient method for example, does not require a symmetric system matrix [20] and therefore permits an implicit treatment of the off-diagonal terms. Our implementation of the preconditioned GMRES method is described in Lemieux et al. [5].

In both methods, it is desirable to use a matrix-free approach [21]. It is especially true for the Newton method. Indeed, forming and storing the Jacobian matrix is prohibitively expensive in CPU time and storage. Also, because of the complexity of the sea ice momentum equation with a VP formulation, finding the Jacobian is a difficult task that is prone to errors. For these reasons, we adopt a Jacobian-free approach. Krylov methods to solve systems of linear equations do not need the

system matrix (in our case **J**) to be formed and stored. These methods only require the results of matrix-vector products. With a Jacobian-free approach, the product of **J** times a vector **v** can be approximated by

$$\mathbf{J}(\mathbf{u}^{k-1})\mathbf{v} \sim \frac{\mathbf{F}(\mathbf{u}^{k-1}+\epsilon\mathbf{v})-\mathbf{F}(\mathbf{u}^{k-1})}{\epsilon}, \tag{24}$$

where ϵ is a small number (10^{-6} in our implementation). This is a first-order Taylor series expansion: a forward difference approach. The extra accuracy obtainable with a centered difference is negligible for our purposes and so we use the more computationally efficient forward difference to approximate $\mathbf{J}(\mathbf{u}^{k-1})\mathbf{v}$. Note that in the standard method, we also use a matrix-free approach but we calculate the exact product of the system matrix **A** and a vector.

We also need to specify a preconditioner for GMRES. Our implementation of GMRES is the flexible GMRES which implies right preconditioning [20]. Preconditioning is a technique for accelerating the convergence rate of an iterative method by defining a transformed system that has the same solution as the original problem but which is easier to solve. The preconditioner matrix, denoted by **M**, is usually a simpler matrix than the system matrix. Again, because we want to have a Jacobian-free approach, we choose a simpler matrix than **J**. Recall from Eq. (22) that $\mathbf{J}=\mathbf{A}+\mathbf{G}$. Therefore, only the simpler matrix **A** could be used as a preconditioner. However, we make a further simplification by setting the Coriolis parameter and the water drag turning angle to zero (only in the preconditioner). This is done in order to increase the diagonal dominance of the preconditioner matrix. Therefore, $\mathbf{M}\sim\mathbf{A}$. One can get an idea of the matrix **M** by setting $f=0$ and $\theta_w=0$ on the left hand side of Eqs. (13) and (14). The advantage of using this matrix **M** is that we use the same preconditioner as used in the standard method. During preconditioning, a fixed number of iterations (8 iterations) of a LSOR solver are performed on $\mathbf{M}\mathbf{z}_\mathbf{j}=\mathbf{v}_\mathbf{j}$ where $\mathbf{z}_\mathbf{j}$ and $\mathbf{v}_\mathbf{j}$ are vectors formed during the jth Krylov iteration [5].

3.4. The standard and JFNK solvers

Recall that we use the past two iterates in the standard model for the linearization. Here is the algorithm that describes the standard solver:

1. Start with an initial iterate $\mathbf{u}^0$ and calculate $\|\mathbf{F}(\mathbf{u}^0)\|$
do $k=1,\ k_{max}$
2. "Solve" $\mathbf{A}(\mathbf{u}_l)\mathbf{u}^k=\mathbf{b}(\mathbf{u}_l)$ using the preconditioned GMRES method
3. if $\|\mathbf{F}(\mathbf{u}^k)\|<\gamma_{nl}\|\mathbf{F}(\mathbf{u}^0)\|$ stop
enddo

where $\mathbf{u}_l$ is the average of the past two iterates except for $k=1$ where it is the initial iterate. Note that the subscript "l" stands for linearization.

And here is the algorithm that describes the JFNK solver:

1. Start with an initial iterate $\mathbf{u}^0$ and calculate $\|\mathbf{F}(\mathbf{u}^0)\|$
do $k=1,\ k_{max}$
2. "Solve" $\mathbf{J}(\mathbf{u}^{k-1})\boldsymbol{\delta}\mathbf{u}^k=-\mathbf{F}(\mathbf{u}^{k-1})$ using the preconditioned GMRES method
3. $\mathbf{u}^k=\mathbf{u}^{k-1}+\boldsymbol{\delta}\mathbf{u}^k$
4. if $\|\mathbf{F}(\mathbf{u}^k)\|<\gamma_{nl}\|\mathbf{F}(\mathbf{u}^0)\|$ stop
enddo

In these algorithms, $\|\|$ is the 2-norm (simply referred to as the norm for the rest of this paper). For a vector **v** of dimension n, the norm is given by $\left(\sum_{r=1}^{n} v_r^2\right)^{1/2}$ where the v_r are the components of **v**. $\|\mathbf{F}(\mathbf{u}^0)\|$ is the initial residual norm. In step 2, the convergence criterion for the linear solver is $\|\mathbf{J}(\mathbf{u}^{k-1})\boldsymbol{\delta}\mathbf{u}^k+\mathbf{F}(\mathbf{u}^{k-1})\|<\gamma(k)\|\mathbf{F}(\mathbf{u}^{k-1})\|$ for the JFNK solver and $\|\mathbf{A}(\mathbf{u}_l)\mathbf{u}^k-\mathbf{b}(\mathbf{u}_l)\|<\gamma(k)\|\mathbf{F}(\mathbf{u}^{k-1})\|$ for the standard solver. The parameter $\gamma(k)$, that could change from one iteration to the next, is a constant smaller than unity. For the standard solver, $\gamma(k)$ is constant and equal to 0.04 throughout the whole OL iteration process. This value provides the steepest decrease of the nonlinear residual norm as a function of the CPU time. For the JFNK method, solving $\mathbf{J}(\mathbf{u}^{k-1})\boldsymbol{\delta}\mathbf{u}^k=-\mathbf{F}(\mathbf{u}^{k-1})$ to a very high accuracy (oversolving: $\gamma(k)\to 0$) might overall increase the total CPU time in solving the nonlinear system of equations. In other words, one might obtain an accurate linear solution that leads to an inaccurate correction to the nonlinear problem. As in [22], we have observed that oversolving in early Newton iterations can be detrimental on the CPU time and can even prevent the method to converge. When the approximate solution to the linear system of equations is not "accurate", the approach is referred to as an inexact Newton method [8]. Our tests have shown that the convergence rate with the JFNK solver is characterized by two phases: a slow initial phase for which a large value of $\gamma(k)$ should be used (to improve global convergence) and a fast phase for which smaller values of $\gamma(k)$ can be used. These two phases are separated by the residual norm res_t where the subscript "t" stands for transition. $\gamma(k)$ is given by

$$\gamma(k)=\begin{cases}\gamma_{ini}, & \text{if } \|\mathbf{F}(\mathbf{u}^{\mathbf{k-1}})\|\geqslant res_t,\\ \frac{\|\mathbf{F}(\mathbf{u}^{\mathbf{k-1}})\|}{\|\mathbf{F}(\mathbf{u}^{\mathbf{k-2}})\|}, & \text{if } \|\mathbf{F}(\mathbf{u}^{\mathbf{k-1}})\|<res_t,\end{cases} \tag{25}$$

where we have used an expression of $\gamma(k)$ similar to the one of [23] for the fast phase. γ_{ini} is set to 0.99. res_t is the only value that changes with the spatial resolution. res_t is set to 0.05 at 80-km resolution, 0.25 at 40 km, 0.625 at 20 km and 1.25 at 10 km. These values roughly correspond to a quarter of the typical initial residual norm (with a 30-minute time step). Note that at 20-km and 10-km resolutions, $\gamma(k)$ is forced to be larger than 0.3 to prevent excessive use of the linear solver which tends to slow down the nonlinear solver.

A termination criterion for solving the nonlinear system of equations is also needed. Hence, the nonlinear solver (either JFNK or the standard solver) stops iterating after a required drop in the residual norm is obtained (step 3 for the standard solver and step 4 for the JFNK method). γ_{nl}, which is similar to $\gamma(k)$ for the linear problem, is a factor characterizing the quality (set by the user) of the nonlinear approximate solution (the quality increases as γ_{nl} decreases). A solver fails to converge when the termination criterion is not reached in k_{max} iterations.

This paper is not intended to provide the acceptable magnitude of errors on the approximate solution (which depends on γ_{nl}) for VP models but to study and compare the convergence properties of the VP model standard solver to the ones of the JFNK method. As an indication, for γ_{nl}=0.2, the standard solver performs on average 16 OL iterations at 80-km resolution, 24 at 40-km resolution, 36 at 20-km resolution and 58 at 10-km resolution.

4. The model and the forcing fields

The model has four possible spatial resolutions: 80 km, 40 km, 20 km and 10 km. The Canadian Arctic Archipelago has open channels only for the 20-km and 10-km resolution versions. The model has two thickness categories and a zero-layer thermodynamics [3]. The advection is performed after the last OL (or Newton) iteration with a simple upstream scheme. More details on the model can be found in Lemieux et al. [5].

In the original VP model [3], the viscous coefficients were also limited to minimum values ($\zeta_{min} = 4 \times 10^8$ kg s^{-1}) to prevent nonlinear instabilities. Hunke [24] pointed out that $\zeta_{min} = 4 \times 10^8$ kg s^{-1} leads to boundary layers that are too wide (near a land boundary) and to significant velocity errors (compared to the case of $\zeta_{min} = 0$ kg s^{-1}) in marginal zones. Contrary to Lemieux et al. [5], we now set ζ_{min} to zero. We did not encounter any problems due to the fact that ζ_{min} is now equal to zero.

The wind stress is calculated from geostrophic winds derived from the National Centers for Environmental Prediction and National Center for Atmospheric Research (NCEP/NCAR) 6 h reanalysis of sea level pressure [25]. The interpolation of the sea level pressure on our grids is performed using the Akima scheme [26]. The geostrophic winds at a specific time step are linearly interpolated between the previous and subsequent 6 h geostrophic wind fields. The climatological ocean currents were obtained from the steady-state solution of the Navier–Stokes equation in which the advection of momentum is neglected, a 2-D non-divergent field is assumed and a quadratic drag law is used. The forcing used to derive the ocean currents is a 30-year climatological wind stress field. The sea ice model is coupled thermodynamically to a slab ocean model [13]. The thermodynamics is forced by NCEP/NCAR reanalysis of monthly mean surface air temperature. All NCEP/NCAR reanalysis data are found at http://www.cdc.noaa.gov/.

Starting with a constant sea ice thickness of 1 m and a concentration of 100%, the standard model was run (at each spatial resolution) for 10 years from 1980 to 1990 with a six-hour time step and 10 OL iterations. The fields obtained on December 31, 1989 are used as the initial conditions for the simulations presented here. The rheology parameters are taken as $P^* = 30 \times 10^3$ N m^{-2}, $C = 20$ and $e = 2$. Values for the other model parameters are the same as used by Tremblay and Mysak [13].

5. Simulation results

5.1. Convergence properties of the standard and JFNK solvers

A time step of 30 min is used for all the runs. This is a typical time step used in current regional ice-ocean models [27]. The previous time step solution is the initial iterate. All four spatial resolutions are used for the runs. The use of multiple spatial resolutions allows one to assess the behavior and the robustness of a solver.

All real variables were defined as double precision. All simulations were performed on a desktop computer (one dual-core Intel Xeon CPU Model 5140 2.33 GHz, cache of 4096 Kb with a RAM of 4 Gb). The fortran compiler is gfortran 4.3.0, 64 bits. The optimization option O3-ffast-math was used for all the runs.

Fig. 1 shows a typical velocity field solution. This solution is obtained with the JFNK solver with $\gamma_{nl} = 0.001$. The time is 7 January 1990 08Z and the spatial resolution is 10 km. The continents are shown in grey on the figure. Note that for clarity, only one out of a 100 velocity vectors is plotted. The velocity vectors are plotted only in regions where the sea ice concentration is higher than 50%. Note also that there are regions where the concentration is higher than 50% but for which the velocity vectors are very small and therefore are not visible (e.g., North of Canada and Alaska). The average velocity is 9.5 cm s^{-1}.

We first investigate the sole effect of using the hyperbolic tangent for the viscous coefficients as opposed to the capping. For the rest of this paper, when the standard solver is used and the viscous coefficients are capped, we refer to these experiments as Stand-cap while we refer to Stand-tanh when the standard solver uses the continuously differentiable formulation for the viscous coefficients, i.e., the hyperbolic tangent. In a first set of runs, the Stand-cap and Stand-tanh solvers were used for a 1-month (January 1990) integration at different spatial resolutions (80 km, 40 km, 20 km and 10 km) and for different required drops in the residual norm ($\gamma_{nl} = 0.2, 0.1, 0.01, 0.001$). Due to computational constraints with the standard model,

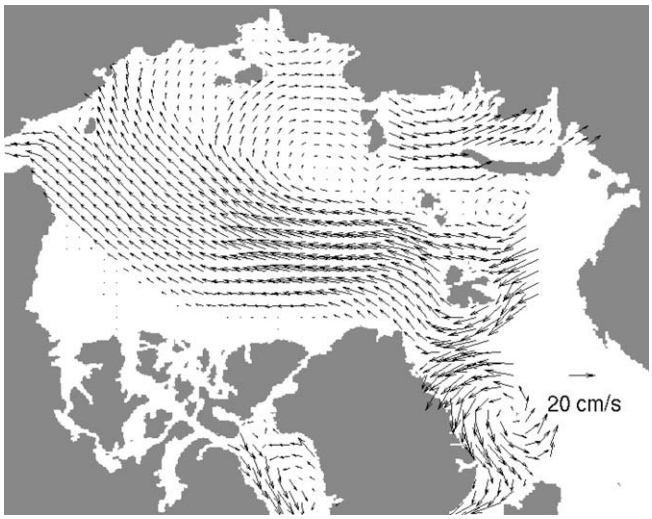

Fig. 1. Simulated velocity field on 7 January 1990 08Z. The JFNK solver is used with $\gamma_{nl} = 0.001$. The spatial resolution is 10 km.

the simulations with $\gamma_{nl} = 0.01$ and $\gamma_{nl} = 0.001$ at 10-km resolution were not performed. For each experiment, the number of failures (convergence not achieved) and the mean CPU time (excluding failure cases) needed to reach the termination criterion were calculated. The mean CPU time is the total CPU time needed by a nonlinear solver for a certain experiment divided by the number of time steps in the 1-month integration (1488 time steps). Results of a certain experiment (e.g. 80 km, $\gamma_{nl} = 0.1$) with Stand-cap are then compared with results of the very same experiment (i.e., 80 km, $\gamma_{nl} = 0.1$) with Stand-tanh. A failure occurs when a solver is not able to reach the desired residual norm before the maximum number of iterations is attained. For the standard solver, the maximum number of iterations is set to 2000.

Fig. 2(a) shows the ratios of the mean CPU times for Stand-cap and the mean CPU times for Stand-tanh as a function of resolution. The different curves correspond to different values of γ_{nl}. Note that the curves for $\gamma_{nl} = 0.2$ and $\gamma_{nl} = 0.1$ are very similar and overlay each other. The grey dashed line defines a CPU time ratio of 1.0, i.e., above this line the Stand-tanh solver is more efficient than the Stand-cap solver. Fig. 3(a) shows the percentage of failures (number of failures $\times 100/1488$) as a function of the resolution for the Stand-cap solver and Fig. 3(b) for the Stand-tanh solver. Note that a portion of these failures are due to a very slow convergence rate (i.e., the solver would converge if the maximum number of iterations was higher). The rest of the failures are cases for which the residual norm reaches a plateau (maybe a local minimum in the residual norm) and does not decrease anymore. A test with the Stand-cap solver at 40-km resolution for $\gamma_{nl} = 0.001$ shows that there are 12 failures during the 1-month of integration when the maximum number of OL (k_{max}) is set to 5000 as opposed to 14 for $k_{max} = 2000$. We observe that having a continuously differentiable formulation of the viscous coefficients increases the computational efficiency and decreases the number of failures. The computational gain associated with the hyperbolic tangent slightly increases with resolution and with the quality of the approximate solution (as γ_{nl} decreases).

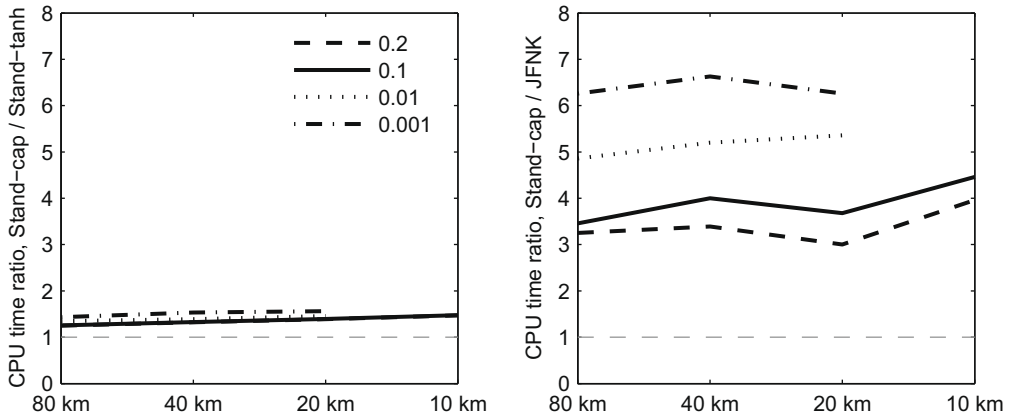

Fig. 2. (a) CPU time needed by the Stand-cap solver divided by the CPU needed by the Stand-tanh solver for the residual to decrease by a factor of γ_{nl} as a function of the resolution. (b) CPU time needed by the Stand-cap solver divided by the CPU needed by the JFNK solver for the residual to decrease by a factor of γ_{nl} as a function of the resolution. The values of γ_{nl} are the same for both panels and are given in the legend of the left panel. The dashed grey lines indicate a CPU time ratio of 1.

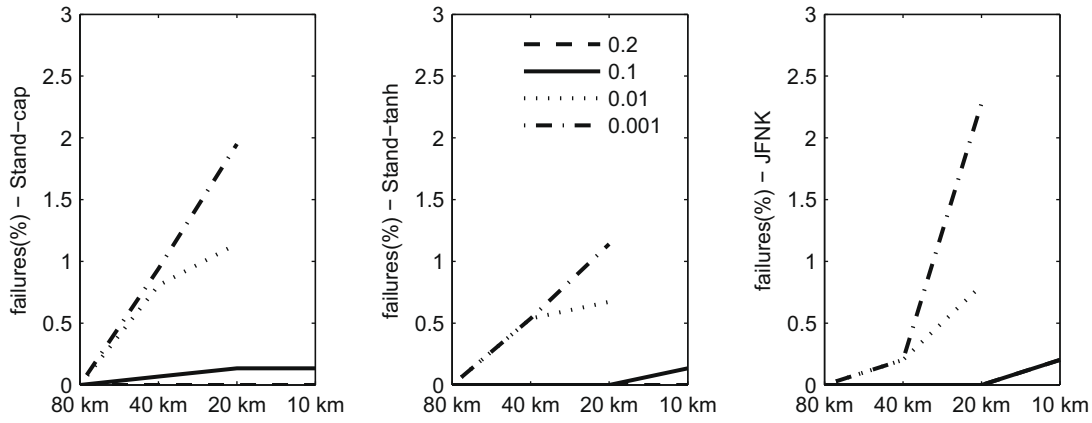

Fig. 3. Percentage of failures as a function of resolution for the Stand-cap (a), Stand-tanh (b) and the JFNK solver (c). A failure occurs when a solver is not able to obtain a sufficient drop in the residual norm as defined by γ_{nl}. The values of γ_{nl} are the same for the three panels and are given in the legend of the middle panel.

We also compare the computational efficiency and robustness of the JFNK method to the ones of the Stand-cap solver. Fig. 2(b) shows the ratios of the mean CPU times for the Stand-cap solver and the mean CPU times for the JFNK method as a function of the resolution. Again, the different curves correspond to different values of γ_{nl}. The grey dashed line defines a CPU time ratio of 1.0, i.e., above this line the JFNK solver is more efficient than Stand-cap. Fig. 3(c) shows the percentage of failures as a function of the resolution for the JFNK solver. For the JFNK solver, the maximum number of iterations is set to 1000.

Due to the limited length of the simulations (1 month: 1488 time steps), only qualitative conclusions should be drawn about the failure rate results. Longer simulations would be needed to assess the exact values of the failure rates.

Fig. 2(b) shows that for all spatial resolutions, the computational gain of the JFNK solver over the standard solver increases as γ_{nl} decreases. This means that the JFNK method is more and more efficient compared to the standard solver as a better approximate solution is required. Also, at all spatial resolutions, the JFNK solver is significantly more efficient than both Stand-cap and Stand-tanh solvers. Indeed, the JFNK solver is 3.0–6.6 times faster than Stand-cap (depending on the resolution and the required drop in the residual norm).

Fig. 3 shows that for both standard and JFNK solvers, the number of failures increases with the spatial resolution. The fact that the number of failures increases as the grid is refined is related to sharper solution structures at high-resolution. Lemieux and Tremblay [6] have shown that the largest errors with the standard solver are in zones of large sea ice deformations (large velocity gradients). The JFNK method is also known to have convergence problems when the solution has sharp

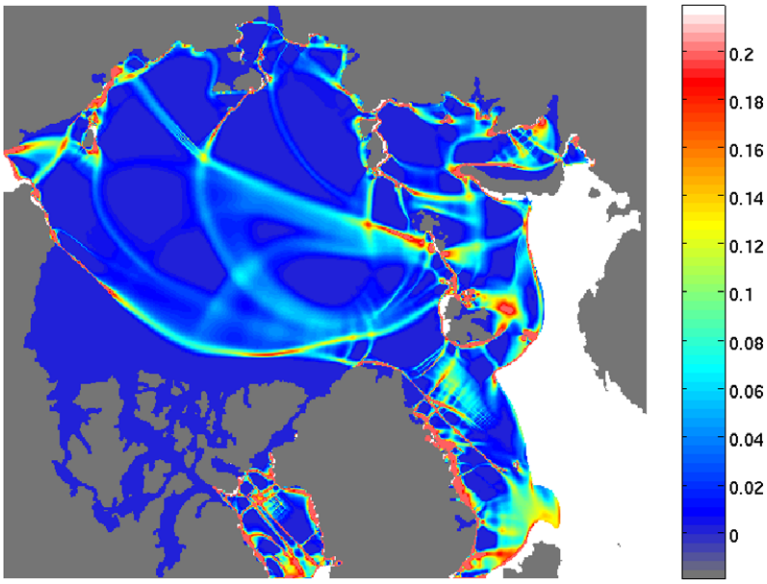

Fig. 4. Simulated shear deformation (capped at 0.2 day^{-1} for clarity) field on 7 January 1990 08Z. The JFNK solver is used with $\gamma_{nl} = 0.001$. The spatial resolution is 10 km.

structures [8]. Fig. 4 illustrates what we mean by sharp solution structures. It shows the shear deformation field on 7 January 1990 08Z simulated by the JFNK solver when using the 10-km resolution model and a γ_{nl} of 0.001. The shear deformation (second strain rate invariant) is given by $\sqrt{\left(\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right)^2+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2}$. As in Maslowski and Lipscomb [28], who used a model with about the same spatial resolution (9 km), our model simulates basin scale linear kinematic features that resemble the observed ones [29]. Note that the existence of these strong velocity gradients is physically based (VP rheology) and is not a consequence of residual errors in the velocity field approximate solution.

Note that for the JFNK solver, the computational efficiency and failure rate depend on the chosen value of res_t (Eq. (25)) and that some tuning might slightly modify these results. A larger res_t tends to increase the computational efficiency and the failure rate.

The lack of convergence (failures) of the JFNK solver and the standard solver is a global convergence issue. When the initial iterate is "sufficiently close" to the solution, the solvers always converge. The quality of the initial iterate is determined by the time step compared to the forcing time scale and to the level of convergence of the previous time step solution. A 1-month integration at 40-km resolution with a 1-minute time step (44,640 time steps) for $\gamma_{nl}=0.001$ shows that both solvers always converge. Unfortunately, the use of such a small time step represents a prohibitive computational approach. We have not investigated what is the maximum time step allowed (between 1 and 30 min at 40-km resolution) for the solvers to converge in all cases.

To illustrate the high convergence rate of the JFNK method as opposed to the ones of Stand-cap and Stand-tanh, Fig. 5 shows the residual norm of the nonlinear system of equations as a function of the iteration (Newton iteration or OL iteration) down to a small residual norm (10^{-6}). This typical result is for 1 January 1990 18Z. The Stand-cap solver needs in this case 2631 OL iterations to reach a residual norm of 10^{-6} while it takes 24 Newton iterations for JFNK to satisfy the same criterion. This might suggest that JFNK is more than a 100 times faster than the Stand-cap solver. This is however not the case because one JFNK iteration involves more calculation (in the fast phase) than one OL iteration. JFNK is ~23 times faster than the Stand-cap solver to reach a residual norm of 10^{-6}. Compared to the Stand-tanh solver, JFNK is 6.4 times faster. The required CPU time for JFNK is 2.41 s, 15.49 s for Stand-tanh and 55.07 s for Stand-cap.

Even though the convergence rate of the JFNK solver is high (especially in the fast phase), it is not quadratic because an inexact Newton approach is used. Asymptotic quadratic convergence could be possible but at the expense of very small $\gamma(k)$ values [8].

5.2. Discussion about the robustness of the standard and JFNK solvers

Both standard and JFNK solvers show a lack of robustness. Moreover, the failure rate for both solvers increases as the grid is refined. However, the lack of robustness of the solvers might not be so dramatic for practical considerations. First, $\gamma_{nl}=0.2$

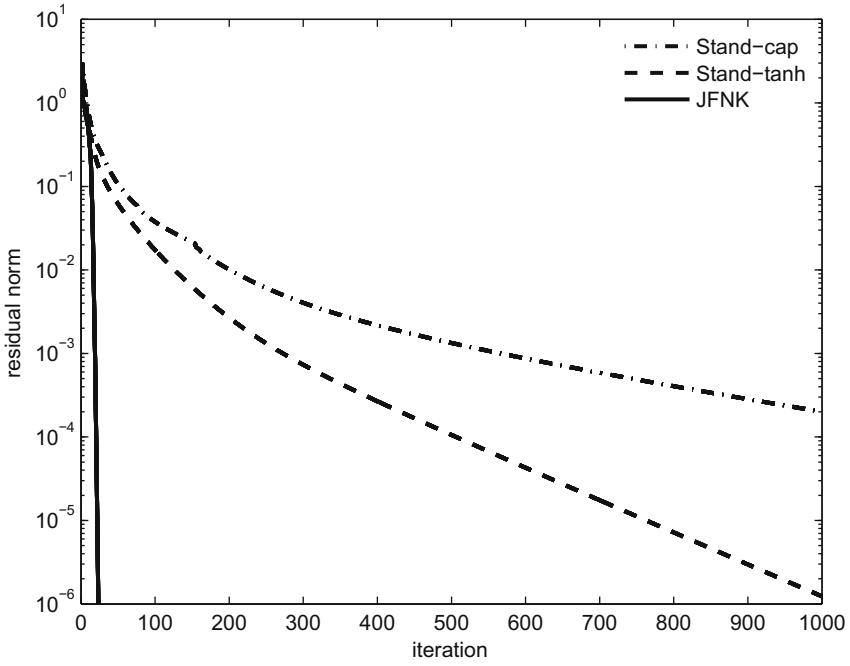

Fig. 5. Residual norm (N m^{-2}) of the nonlinear system of equations as a function of the OL iteration (or Newton iteration) on 1 January 1990 18Z. The spatial resolution is 40 km.

might lead to errors that are small enough on the approximate solution for typical applications of VP models. Lemieux and Tremblay [6] have shown that 10 OL iterations with a 30-minute time step at 10-km resolution lead to errors smaller than 1 cm s^{-1} everywhere on the domain. Our results show that on average, 58 OL iterations at 10-km resolution are required for $\gamma_{nl} = 0.2$. For this drop in the residual ($\gamma_{nl} = 0.2$), the Stand-cap and Stand-tanh solvers always converge at all resolutions. On the other hand, JFNK always converges at 80, 40 and 20-km resolutions and fails only three times at 10-km resolution. Secondly, for both solvers, our tests show that when a failure occurs, the residual norm initially decreases and then flattens out (with an oscillation superimposed). A plot of the residual on the grid (not shown) indicates that for failure cases, the lack of convergence usually only affects a few grid cells, i.e., the residual is small for all the velocity components except in very localized regions. Even though the lack of robustness is a debatable problem, we still have investigated ways to improve the robustness of both solvers.

We propose a globalization "trick" for the standard solver. Recall that with the standard solver, the past two iterates are used to get the linearized system of equations. The linearization velocity field at iteration k can therefore be written as

$$\mathbf{u}_l^k = \frac{(\mathbf{u}^{k-1} + \mathbf{u}^{k-2})}{2}, \tag{26}$$

where the subscript "l" refers to the linearization. The globalization "trick" we propose is to switch to a different linearization field (a smoother field) when the residual norm stagnates. When the number of OL iterations is larger than 500, we use the following linearization

$$\mathbf{u}_l^k = \frac{(\mathbf{u}_l^{k-1} + \mathbf{u}^{k-1})}{2}, \tag{27}$$

which has "memory" of other iterates than just $\mathbf{u}^{k-1}$ and $\mathbf{u}^{k-2}$. Note that $\mathbf{u}_l^1 = \mathbf{u}^0$.

We have conducted a test at 40-km resolution with $\gamma_{nl} = 0.001$ for the same month of integration. The Stand-tanh solver previously had 8 failures (over the 1488 cases). With the globalization "trick", the solver always converges. Fig. 6 shows the residual of the nonlinear system of equations as a function of the OL iteration on 21 January 1990 07Z. The residual norm stagnates (and slightly oscillates) after ~100 OL iterations. The linearization is switched after 500 iterations and the residual norm quickly decreases. We argue that, when the residual oscillates and stagnates, the smoother linearization prevents the solver from overshooting and therefore allows convergence.

Various approaches have been tried so far to improve the robustness of the JFNK solver. With the line search method [30], once a correction $\boldsymbol{\delta}\mathbf{u}^k$ is obtained, it is scaled by a factor s (1.0, 0.5, 0.25...) until $\|\mathbf{F}(\mathbf{u}^{k-1} + s\boldsymbol{\delta}\mathbf{u}^k)\|$ is smaller than $\|\mathbf{F}(\mathbf{u}^{k-1})\|$. Unfortunately, the line search method does not decrease the number of failures as the residual norm still stagnates.

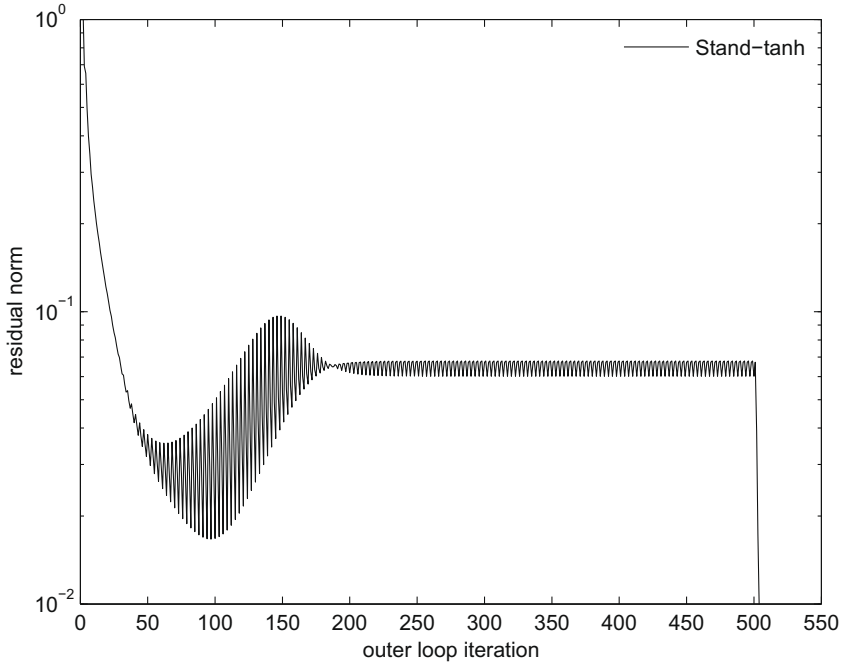

Fig. 6. Residual norm (N m^{-2}) of the nonlinear system of equations as a function of the OL iteration on 21 January 1990 07Z. The globalization "trick" is turned on at OL iteration 500. The spatial resolution is 40 km. The Stand-tanh solver is used.

6. Conclusion

The nonlinear solver of viscous-plastic (VP) models, referred to as the standard solver, is based on an implicit solution of a linearized system of equations and an outer loop (OL) iteration. To prevent the viscous coefficients with a VP formulation to become singular when the sea ice deformations are small, a capping limits their value [3]. This capping of the viscous coefficients however leads to a momentum equation that is not continuously differentiable with respect to the velocity.

Replacing the formulation of the viscous coefficients with capping by a hyperbolic tangent function [6] leads to a continuously differentiable momentum equation. When this latter formulation is used, the standard model is about 1.5 times faster than when the former approach is employed. In other words, for a fixed number of OL iterations performed, the approximate solution is a better converged solution when the momentum equation is continuously differentiable. Furthermore, the computational gain slightly increases as the grid is refined.

Apart from leading to an improved computational efficiency of the standard solver, the continuously differentiable formulation of the viscous coefficients decreases the number of failures. We therefore advise modelers to use this continuously differentiable formulation for the viscous coefficients as it involves the modification of only one line of code.

Modelers should also be careful in formulating the yield curve. Jennifer Hutchings (personal communication) observed a much lower convergence rate of a VP model with a lens shaped yield curve (which is not continuously differentiable) as opposed to the elliptical yield curve. Whether other yield curves that are not continuously differentiable (e.g. [13,31]) also lead to a slower convergence remains to be investigated.

We have also implemented the Jacobian-free Newton–Krylov (JFNK) method to solve the sea ice momentum equation with a VP formulation. The JFNK method is advantageous as the system matrix (the Jacobian) does not need to be formed and stored, it is parallelizable and the convergence can be nearly quadratic in the vicinity of the solution. The computational gain of the JFNK method over the standard solver used in existing VP models increases with the required drop in the residual norm (termination criterion). When the termination criterion requires the initial residual norm to be divided by 5 ($\gamma_{nl} = 0.2$), the JFNK method is 3.25 times faster than the standard solver at 80-km resolution. This computational gain is 3.4 at 40-km, 3.0 at 20-km and 4.0 at 10-km resolution.

For a large required drop in the residual norm, both JFNK and standard solvers sometimes do not converge. The failure rate for both solvers increases as the grid is refined but stays relatively small (less than 2.3% of the 1488 cases are failures). This lack of robustness of the standard and JFNK solvers is a global convergence issue. When the initial iterate is "close" to the solution, the solvers always converge to this solution. The quality of the initial iterate can be improved by reducing the size of the time step. Tests at 40-km resolution with a time step of one minute show that both solvers always converge. Unfortunately, such a small time step leads to a significant increase in computational time.

The fact that failures to converge are sometimes present is a matter of debate as it mostly occurs for large required drops in the residual norm. The "sufficient" drop in the residual norm for typical VP model applications remains to be investigated. Furthermore, when a failure occurs, it usually affects only a few grid cells, i.e., the residual is small for all the velocity components except in very localized regions. As the number of failures increases with the spatial resolution, we argue that they are related to sea ice deformations that become more important as the grid is refined. Nonlinear solvers such as the JFNK method tend to have difficulties when there are sharp structures in the solution [8]. To cure this problem of failures, both solvers require a globalization method.

For the standard solver, a globalization "trick" that involves a switch to a different linearization approach when the residual norm stagnates, eliminates the failures. Globalization methods for the JFNK solver have not yet proven to be successful. For example, a line search method slightly reduces the computational efficiency and does not improve the robustness of the JFNK solver. Further investigation is needed to cure this globalization issue with the JFNK solver.

In this paper, the computational gain of the JFNK solver was given with respect to a standard solver using the preconditioned GMRES method for the linear solver. Lemieux et al. [5] have shown that the preconditioned GMRES method is 16 times faster than an SOR solver and 3 times faster than a LSOR solver-two linear solvers that are often used in the sea ice modeling community. While we cannot provide the exact gain of the JFNK solver over existing VP models using an SOR or LSOR linear solver (because the experiments described here are slightly different than the ones performed by [5]), the gain is multiplicative and would be of the order of 10–50 times depending on the linear solver used.

The JFNK method is therefore a significant improvement in terms of computational efficiency when compared to the standard solver. To improve the robustness of the JFNK solver, we are currently implementing a pseudo-transient continuation method [8]. This method introduces an artificial time derivative term that goes to zero as the approximate solution approaches the solution. It has proven to be useful for problems where line search methods fail to converge to a global minimum. It is especially useful when the solution includes sharp structures [32].

Acknowledgments

We would like to thank Einar Olason, Jennifer Hutchings and Gleb Panteleev for many useful discussions during the course of this work. We also thank two anonymous reviewers for very interesting comments. Bruno Tremblay is grateful to NSERC for a Discovery Grant, and to the National Science Foundation Office of Polar Program (OPP-0230325) and Arctic Science Program (ARC-0520496). Paul Tupper was supported by an NSERC Discovery Grant. David Huard thanks NSERC for a

Postdoctoral fellowship. This work was also partially supported by the National Science Foundation under Grant No. 0804010.

References

[1] M.D. Coon, G.A. Maykut, R.S. Pritchard, D.A. Rothrock, A.S. Thorndike, Modeling the pack ice as an elastic-plastic material, AIDJEX Bull. 24 (1974) 1–105.
[2] G.M. Flato, W.D. Hibler, Modeling pack ice as a cavitating fluid, J. Phys. Oceanogr. 22 (1992) 626–651.
[3] W.D. Hibler, A dynamic thermodynamic sea ice model, J. Phys. Oceanogr. 9 (1979) 815–846.
[4] J. Zhang, W.D. Hibler, On an efficient numerical method for modeling sea ice dynamics, J. Geophys. Res. 102 (C4) (1997) 8691–8702.
[5] J.-F. Lemieux, B. Tremblay, S. Thomas, J. Sedláček, L.A. Mysak, Using the preconditioned Generalized Minimum RESidual (GMRES) method to solve the sea-ice momentum equation, J. Geophys. Res. 113 (2008) C10004, doi:10.1029/2007JC004680.
[6] J.-F. Lemieux, B. Tremblay, Numerical convergence of viscous-plastic sea ice models, J. Geophys. Res. 114 (2009) C05009, doi:10.1029/2008JC005017.
[7] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton methods, SIAM J. Optim. 4 (1994) 393–422.
[8] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[9] J.M. Reisner, V.A. Mousseau, A.A. Wyszogrodzki, D.A. Knoll, An implicitly balanced hurricane model with physics-based preconditioning, Mon. Weather Rev. 133 (2005) 1003–1022.
[10] W.P. Abbett, The magnetic connection between the convection zone and corona in the quiet sun, Astrophys. J. 665 (2007) 1469–1488.
[11] L. Ferm, P. Lötstedt, Numerical method for coupling the macro and meso scales in stochastic chemical kinetics, BIT Numer. Math. 47 (4) (2007) 735–762.
[12] E.C. Hunke, J.K. Dukowicz, An elastic-viscous-plastic model for sea ice dynamics, J. Phys. Oceanogr. 27 (1997) 1849–1867.
[13] B. Tremblay, L.A. Mysak, Modeling sea ice as a granular material, including the dilatancy effect, J. Phys. Oceanogr. 27 (1997) 2342–2360.
[14] M.G. McPhee, Ice-ocean momentum transfer for the AIDJEX ice model, AIDJEX Bull. 29 (1975) 93–111.
[15] C.F. Ip, W.D. Hibler, G.M. Flato, On the effect of rheology on seasonal sea-ice simulations, Ann. Glaciol. 15 (1991) 17–25.
[16] J. Dukowicz, Comments on the "stability of the viscous-plastic sea ice rheology", J. Phys. Oceanogr. 27 (1997) 480–481.
[17] W.D. Hibler, S.F. Ackley, Numerical simulation of the Weddell Sea pack ice, J. Geophys. Res. 88 (1983) 2873–2887.
[18] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer, 2000.
[19] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[20] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS, 1996.
[21] N. Qin, D.K. Ludlow, S.T. Shaw, A matrix-free preconditioned Newton/GMRES method for unsteady Navier–Stokes solutions, Int. J. Numer. Meth. Fluids 33 (2000) 223–248.
[22] R.S. Tuminaro, H.F. Walker, J.N. Shadid, On backtracking failure in Newton-GMRES methods with a demonstration for the Navier–Stokes equations, J. Comput. Phys. 180 (2002) 549–558, doi:10.1006/jcph.2002.7102.
[23] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[24] E.C. Hunke, Viscous-plastic sea ice dynamics with the EVP model: linearization issues, J. Comput. Phys. 170 (2001) 18–38.
[25] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, A. Leetmaa, R. Reynolds, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K.C. Mo, C. Ropelewski, J. Wang, R. Jenne, D. Joseph, The NCEP/NCAR 40-year reanalysis project, Bull. Amer. Meteorol. Soc. 77 (1996) 437–470.
[26] H. Akima, Rectangular-grid-data surface fitting that has the accuracy of a bicubic polynomial, Trans. Math. Softw. 22 (3) (1996) 357–361.
[27] M. Johnson, S. Gaffigan, E. Hunke, R. Gerdes, A comparison of Arctic Ocean sea ice concentration among the coordinated AOMIP model experiments, J. Geophys. Res. 112 (2007) C04S11, doi:10.1029/2006JC003690.
[28] W. Maslowski, W.H. Lipscomb, High resolution simulations of Arctic sea ice, 1979–1993, Polar Res. 22 (2003) 67–74.
[29] R. Kwok, Deformations of the Arctic Ocean sea ice cover between November 1996 and April 1997: a survey, J. Dempsey, H. Shen, L. Shapiro, 2001.
[30] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[31] W.D. Hibler, E.M. Schulson, On modeling the anisotropic failure and flow of flawed sea ice, J. Geophys. Res. 105 (C7) (2000) 17105–17120.
[32] T.S. Coffey, C.T. Kelley, D.E. Keyes, Pseudo-transient continuation and differential-algebraic equations, SIAM J. Sci. Comput. 25 (2) (2003) 553–569.