Using the preconditioned Generalized Minimum RESidual (GMRES) method to solve the sea-ice momentum equation

Jean-François Lemieux,[1] Bruno Tremblay,[1] Stephen Thomas,[2] Jan Sedláček,[1] and Lawrence A. Mysak[1]

Received 6 December 2007; revised 14 May 2008; accepted 5 June 2008; published 9 October 2008.

[1] We introduce the preconditioned generalized minimum residual (GMRES) method, along with an outer loop (OL) iteration to solve the sea-ice momentum equation. The preconditioned GMRES method is the linear solver. GMRES together with the OL is used to solve the nonlinear momentum equation. The GMRES method has low storage requirements, and it is computationally efficient and parallelizable. It was found that the preconditioned GMRES method is about 16 times faster than a stand-alone successive overrelaxation (SOR) solver and three times faster than a stand-alone line SOR (LSOR). Unlike stand-alone SOR and stand-alone LSOR, the cpu time needed by the preconditioned GMRES method for convergence weakly depends on the relaxation parameter when it is smaller than the optimal value. Results also show that with a 6-hour time step, the free drift velocity field is a better initial guess than the previous time step solution. For GMRES, the symmetry of the system matrix is not a prerequisite. The Coriolis term and the off-diagonal part of the water drag term can then be treated implicitly. The implicit treatment eliminates an instability characterized by a residual oscillation in the total kinetic energy of the ice pack that can be present when these off-diagonal terms are handled explicitly. Treating these terms explicitly prevents one from obtaining a high-accuracy solution of the sea-ice momentum equation unless a corrector step is applied. In fact, even after a large number of OL iterations, errors in the drift of the same magnitude as the drift itself can be present when these terms are treated explicitly.

Citation: Lemieux, J.-F., B. Tremblay, S. Thomas, J. Sedláček, and L. A. Mysak (2008), Using the preconditioned Generalized Minimum RESidual (GMRES) method to solve the sea-ice momentum equation, *J. Geophys. Res.*, *113*, C10004, doi:10.1029/2007JC004680.

1. Introduction

[2] *Hibler* [1979] developed a viscous-plastic (VP) sea-ice model which became, over the years, the standard model used by many ocean and climate modelers. The VP formulation introduced by *Hibler* [1979] is based on an elliptical yield curve and a normal flow rule. In the VP approach, the elastic strains are neglected [*Hibler*, 1977]. With an ideal plastic formulation, the viscosities, which depend nonlinearly on the strain rates, become singular when the sea-ice deformations (strain rates) tend to zero. To avoid this difficulty, the viscosities are bounded at very high values when the sea-ice deformations are small. This latter regime mimics the rigid behavior of sea ice. In this case, sea ice is treated as a very viscous Newtonian fluid and experiences a slow creep. Given these high values of viscosity in regions where small deformations are present, the stability criterion for a fully explicit time-stepping scheme requires a time step on the order of a second or smaller for a 100-km resolution grid [*Ip et al.*, 1991]. Because of this stringent stability condition, modelers often solve the sea-ice momentum equation using an implicit scheme.

[3] The numerical scheme introduced by *Hibler* [1979] is based on a modified Euler time step and an successive overrelaxation (SOR) solver. In this procedure, the nonlinear solution is first approximated by advancing the solution to the middle of the time step by solving the linearized equation with an SOR solver. The nonlinear viscosities are then updated and the water drag linearized using the newly calculated velocities, and the new linearized equation is again solved with the SOR solver. *Oberhuber* [1993] also introduced a predictor-corrector scheme but chose to solve the linearized equation with an line SOR (LSOR) solver. With this approach, the linearized equation is solved line by line with a tridiagonal matrix solver. Following *Oberhuber* [1993], *Zhang and Hibler* [1997] also solved the linearized equation using an LSOR. However, by treating many terms

[1]Department of Atmospheric and Oceanic Sciences, McGill University, Montréal, Quebec, Canada.
[2]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, USA.

0148-0227/08/2007JC004680

explicitly, they completely decoupled the u and v momentum equations. They showed that the linear solver converges faster when the equations are decoupled. They have also demonstrated that the two steps of the modified Euler time step do not lead to a VP solution, i.e., a solution for which all the stress states lie either inside or on the yield curve. To insure proper convergence, they proposed the use of pseudo time steps, i.e., the repetition of the modified Euler time step. *Tremblay and Mysak* [1997] solved the steady-state momentum equation using an OL along with an SOR solver. First, the linearized equation is solved for the pressure using an SOR solver following *Flato and Hibler* [1992], considering the off-diagonal terms to be constant. Then, the internal ice pressure is used to update the nonlinear viscosities and the linearized equation is solved again with an SOR solver. *Tremblay and Mysak* [1997] proposed to repeat these two steps four times to obtain the solution of the nonlinear momentum equation. Subsequently, *Zhang and Rothrock* [2000] introduced the alternate direction implicit (ADI) method to solve the linearized equations. As in the study by *Zhang and Hibler* [1997], the u and v equations are completely decoupled. They claim that the ADI method is significantly faster than the LSOR method to solve the linearized equation, but that the advantage of ADI over LSOR is reduced when many pseudo time steps are performed.

[4] The common feature of all these numerical schemes is that they produce an approximate solution of the nonlinear momentum equation by solving implicitly the linearized momentum equation a few times per time step. They might differ in the terms treated explicitly or implicitly and in the method used to solve the linearized equation.

[5] *Hunke and Dukowicz* [1997] introduced an extra elastic term in the constitutive equation. 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 scheme. The elastic-viscous-plastic (EVP) approach permits an explicit solution of the momentum equation with a large sub time step. This method is also appropriate for parallelization [*Hunke and Zhang*, 1999].

[6] In recent years, sea-ice models have been developed at higher and higher spatial and temporal resolutions [e.g., *Maslowski and Lipscomb*, 2003; *Hunke and Zhang*, 1999]. These high-resolution models need a considerable amount of floating point operations. Efficient and parallelizable numerical methods therefore need to be employed in order to deal with the enormous amount of computations required.

[7] In this article, we introduce a new numerical scheme that uses an outer loop (OL) iteration along with the preconditioned generalized minimum residual (GMRES) method [*Saad and Schultz*, 1986] to solve the sea-ice momentum equation. We implemented this numerical scheme in order to deal with the intensive computational needs of a newly developed 10-km resolution sea-ice model of the Arctic and Canadian Arctic Archipelago. To solve the nonlinear momentum equation, the equation is first linearized and then solved using the preconditioned GMRES method. These two steps are repeated multiple times (OL) until a chosen convergence criterion is met. The OL iterations are similar to the pseudo time steps (two OL iterations correspond to one pseudo time step) introduced by *Zhang and Hibler* [1997] and to the OL iterations used by *Tremblay and Mysak* [1997].

[8] The GMRES method belongs to the family of Krylov subspace methods [*Saad*, 2003]. Krylov subspace methods are particularly useful for solving large systems of equations. First, they have low memory requirements. Indeed, the system matrix does not need to be formed and stored explicitly. Instead, only the results of matrix-vector multiplications are required. Second, when the system is suitably preconditioned, Krylov methods exhibit very good convergence properties [*Liesen and Tichy*, 2004]. Finally, these methods are well-suited for parallelization [e.g., *da Cunha and Hopkins*, 1994; *DeLong and Ortega*, 1998]. The general idea of Krylov subspace methods is that instead of solving the very large system, the solution is approximated in a subspace of small dimension.

[9] Krylov subspace methods are often used in physics and engineering applications [e.g., *Freund*, 1999] for solving large systems of equations. They have, however, rarely been employed for sea-ice modeling. *Hunke and Dukowicz* [1997] implemented the preconditioned conjugate gradient (CG) method, which is also a Krylov method, to solve the sea-ice momentum equation in an idealized experiment. They observed that the computational efficiency of the preconditioned CG method is comparable to that of the EVP model. The CG method with proper preconditioning is a very efficient method; however, it requires the system matrix and the preconditioner to be symmetric. For this reason, *Hunke and Dukowicz* [1997] treated the Coriolis term and the off-diagonal part of the water drag term explicitly. The explicit treatment of these terms, however, can cause an instability if no corrector step is applied [*Zhang and Hibler*, 1997]. Indeed, as will be shown in section 5.5, the explicit treatment of the Coriolis term and of the off-diagonal part of the water drag term can cause a residual oscillation (observed in the average kinetic energy of the ice pack or in the residual norm) that prevents full convergence of the nonlinear solution when a corrector step is not employed. Note that the corrector step treats the Coriolis term and the off-diagonal part of the water drag term implicitly and the rheology term explicitly.

[10] For GMRES, symmetry is not a prerequisite. This is an advantage as will be demonstrated in section 5.5. To the best of our knowledge, this is the first time that the GMRES method is used to solve the sea-ice momentum equation. In this article, we will discuss the convergence properties of the preconditioned GMRES method and compare the method with other numerical methods used to solve the sea-ice momentum equation. Two preconditioners in combination with GMRES are examined: the SOR method and the LSOR method. It was previously shown that SOR is an efficient preconditioner for GMRES and that this approach is well suited for parallelization when using coloring [*DeLong and Ortega*, 1995]. We refer to GMRES-SOR as the GMRES method with SOR as the preconditioner. Similarly, GMRES-LSOR is the GMRES method with LSOR as the preconditioner.

[11] The article is structured as follows. In section 2, a review of the sea-ice momentum equation with a VP formulation is given. In section 3, we describe the numerical

scheme and introduce the GMRES method. In section 4, we describe the model and the forcing fields used for the simulations. In section 5, we present results on the convergence properties of the preconditioned GMRES method and compare these to the ones of a stand-alone SOR and a stand-alone LSOR. The impact of a partly implicit treatment of the linearized equation versus a fully implicit treatment is also discussed in section 5. Concluding remarks and a description of future work are given in section 6.

2. Sea-ice Momentum Equation

[12] The two-dimensional sea-ice momentum equation is given by

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

where ρ_i is the density of the ice, h the sea-ice thickness, f the Coriolis parameter, $\mathbf{u}$ the horizontal sea-ice velocity vector, τ_a the wind stress, τ_w the water drag, σ the internal ice stress tensor ($\nabla \cdot \sigma$ is defined as the rheology term), g the gravity and H_d the sea surface height. Following *Tremblay and Mysak* [1997], 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 [*McPhee*, 1975]

$$\tau_{\mathbf{a}} = \rho_a C_{da} |\mathbf{u}_a^g| \left(\mathbf{u}_a^g \cos\theta_a + \mathbf{k} \times \mathbf{u}_a^g \sin\theta_a\right), \quad (2)$$

$$\tau_{\mathbf{w}} = \rho_w C_{dw} |\mathbf{u} - \mathbf{u}_w^g| \left[\left(\mathbf{u} - \mathbf{u}_w^g\right) \cos\theta_w + \mathbf{k} \times \left(\mathbf{u} - \mathbf{u}_w^g\right) \sin\theta_w\right], \quad (3)$$

where 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 $\mathbf{u}_a$, $\mathbf{u}$ has been neglected in the expression for the wind stress.

[13] With a VP formulation, the constitutive law can be written as [*Hibler*, 1979]

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

where 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 $\dot{\epsilon}_{12} = \frac{1}{2}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$, $\dot{\epsilon}_{kk} = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}$, ζ is the bulk viscosity and η is the shear viscosity. According to *Hibler* [1979], the pressure term P is parameterized by the following equation:

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

where P^* is the ice strength per meter, 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.

[14] The rheology term ($\nabla \cdot \sigma$) 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 [*Hibler*, 1979]. In this case, the bulk and shear viscosities are given by

$$\zeta = \frac{P}{2\triangle}, \quad (6)$$

$$\eta = \zeta e^{-2}, \quad (7)$$

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

[15] In the limit where Δ tends to zero, equations (6) and (7) become singular. Following *Hibler* [1979], the values of ζ and η are capped at maximum values given by

$$\zeta_{\max} = \left(2.5 \times 10^8\right) P, \quad (8)$$

$$\eta_{\max} = \zeta_{\max} e^{-2}. \quad (9)$$

[16] ζ is also limited to a minimum value of 4×10^8 kg s^{-1} in order to avoid potential numerical instabilities [*Hibler*, 1979]. This value is several orders of magnitude smaller than what describes strong ice interactions and results in a virtually free drift behavior. A scale analysis of equation (1) for a time step of 6 hours shows that the acceleration term is at least two orders of magnitude smaller than typical wind stress values. For this reason, we neglect the acceleration term.

3. Numerical Scheme

3.1. Solving the Nonlinear Equation: The OL Iterations

[17] Because of the water drag and rheology terms, equation (1) is strongly nonlinear. At any given time step, the equation is linearized using previous u and v components of ice velocity (or the initial guess velocity field for the first iteration). The linear system of equations is then solved using the preconditioned GMRES method until a convergence criterion is satisfied. The new solution is used to update the viscosities and $|\mathbf{u} - \mathbf{u}_w^g|$ in the water drag term. This process is repeated multiple times (OL) until a chosen convergence criterion is met for the nonlinear equation. Note that there are two convergence criteria: one for the linear problem (that might change from one OL iteration to the next) and one for the nonlinear problem. In this article, we refer to the linear convergence criterion as the tolerance.

[18] The momentum equation is discretized using finite differences. The u and v components of the velocity are positioned on the Arakawa C-grid. 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 derivative 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 [*Dukowicz*, 1997]. Close to model boundaries,

proper left and right difference schemes are used in the Taylor series expansion to evaluate the spatial derivatives. At the k^{th} OL iteration, the u and v momentum equations are written as

$$\frac{\partial}{\partial x}\left[\left(\eta(\mathbf{u}_l^k)+\zeta(\mathbf{u}_l^k)\right)\frac{\partial u^k}{\partial x}\right]+\frac{\partial}{\partial y}\left[\eta(\mathbf{u}_l^k)\frac{\partial u^k}{\partial y}\right] + \frac{\partial}{\partial x}\left[\left(\zeta(\mathbf{u}_l^k)-\eta(\mathbf{u}_l^k)\right)\frac{\partial v^k}{\partial y}\right]+\frac{\partial}{\partial y}\left[\eta(\mathbf{u}_l^k)\frac{\partial v^k}{\partial x}\right] + \rho_i h f v_{avg}^k - C_w(\mathbf{u}_l^k)\left(u^k\cos\theta_w - v_{avg}^k\sin\theta_w\right)=b_u, \quad (10)$$

$$\frac{\partial}{\partial y}\left[\left(\eta(\mathbf{u}_l^k)+\zeta(\mathbf{u}_l^k)\right)\frac{\partial v^k}{\partial y}\right]+\frac{\partial}{\partial x}\left[\eta(\mathbf{u}_l^k)\frac{\partial v^k}{\partial x}\right] + \frac{\partial}{\partial y}\left[\left(\zeta(\mathbf{u}_l^k)-\eta(\mathbf{u}_l^k)\right)\frac{\partial u^k}{\partial x}\right]+\frac{\partial}{\partial x}\left[\eta(\mathbf{u}_l^k)\frac{\partial u^k}{\partial y}\right] - \rho_i h f u_{avg}^k - C_w(\mathbf{u}_l^k)\left(v^k\cos\theta_w + u_{avg}^k\sin\theta_w\right)=b_v, \quad (11)$$

where b_u and b_v are the sum of all the terms that do not depend on $\mathbf{u}^k$, $\mathbf{u}_l^k$ is the velocity vector used at the k^{th} iteration to linearize the momentum equation (the subscript l refers to the linearization) and v_{avg} is the spatial averaging of the four neighboring v components of velocity at the u location (and vice versa for u_{avg}). In equations (10) and (11), $C_w(\mathbf{u}_l^k)$ is defined as

$$C_w(\mathbf{u}_l^k)=\rho_w C_{dw}|\mathbf{u}_l^k-\mathbf{u}_w^g|. \quad (12)$$

[19] The linearization $\mathbf{u}_l^k$ is chosen to be

$$\mathbf{u}_l^k=\frac{\left(\mathbf{u}^{k-1}+\mathbf{u}^{k-2}\right)}{2}, \quad k=2,3..,k_{\max}, \quad (13)$$

where for $k = 1$, $\mathbf{u}_l^1$ is the initial guess velocity ($\mathbf{u}^0$).

[20] All the terms in equations (10) and (11) are written implicitly. Some authors treat some of these terms explicitly [e.g., *Zhang and Hibler*, 1997; *Tremblay and Mysak*, 1997]. The consequences of using one approach as opposed to the other will be discussed in section 5.5.

[21] The structure of the numerical scheme is as follows.

[22] 1. Start with an initial guess $\mathbf{u}^0$
do $k = 1$, $k_{\max}$
[23] 2. Calculate $\zeta(\mathbf{u}_l^k)$, $\eta(\mathbf{u}_l^k)$ and $C_w(\mathbf{u}_l^k)$
[24] 3. Solve equations (10) and (11) for u^k and v^k using the preconditioned GMRES method
[25] enddo

[26] We refer to this "do loop" as the OL. The initial guess can be the previous time step solution, a zero velocity field or the free drift solution. The free drift velocity field is the solution to equation (1) when the rheology term is neglected. The method to obtain the free drift velocity field follows that used in the study of *Tremblay and Mysak* [1997]. Using a B-grid (where u and v are collocated), the u and v equations at a velocity point do not depend on the u and v components at other locations on the grid. This allows one to get the free drift solution very efficiently using a Newton-Raphson method [*Press et al.*, 1989]. The effect of the initial guess on the convergence properties of the numerical scheme is investigated in section 5.4.

3.2. Solving the Linearized Equation: The Preconditioned GMRES Method

[27] The linear system of equations at OL iteration k is written as

$$\mathbf{A}\mathbf{x}^k=\mathbf{b}, \quad (14)$$

where $\mathbf{A}$ is the $n \times n$ system matrix, $\mathbf{x}^k$ is the solution vector in which the u elements appear first, followed by the v elements, and $\mathbf{b}$ is the forcing vector composed of b_u and b_v (see equations (10) and (11)). The elements of the matrix $\mathbf{A}$ are denoted a_{qr} where $q = 1, 2 \ldots n$ and $r = 1, 2 \ldots n$.

[28] Like other Krylov subspace methods, GMRES may exhibit slow convergence unless preconditioning is employed. Preconditioning is a technique for accelerating the convergence rate of an iterative method by solving a transformed system that has the same solution as the original problem but which is easier to solve. The convergence rate depends on the spectrum of the transformed system [*Liesen and Tichy*, 2004]. Typically, a preconditioner reduces the number of distinct eigenvalues of the transformed system and clusters them together.

[29] During the iteration process, the vectors forming the Krylov subspace are stored and used to find the j^{th} iterate. Each iterate is the vector that minimizes the residual. As the dimension of the subspace increases, there could be a storage issue. To overcome this difficulty, a restarted version of GMRES is employed. After m iterations, the m^{th} iterate is used as the initial condition and the GMRES method is restarted from scratch. The parameter m is called the restart value. To solve the linear system of equation (14), the restarted preconditioned flexible GMRES algorithm (see below) has been implemented [*Saad*, 2003]. This flexible version allows one to use a different preconditioner at each iteration. However, in this work, this capability is not used, and we will simply refer to this method as the preconditioned GMRES method for the rest of the article. The GMRES algorithm that we use is as follows:

[30] 1. Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$ and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
do j = 1, m
[31] 2. do k_p preconditioner iterations with $\mathbf{A}\mathbf{z}_j = \mathbf{v}_j$ to get $\mathbf{z}_j$
[32] 3. Compute $\mathbf{w} = \mathbf{A}\mathbf{z}_j$
do i = 1, j
[33] 4. $h_{i,j} = (w, v_i)$
[34] 5. $\mathbf{w} = \mathbf{w} - h_{i,j}\,\mathbf{v}_i$

[35] enddo

[36] 6. Compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
[37] 7. Define $\mathbf{Z}_j = [\mathbf{z}_1,\ldots,\mathbf{z}_j]$, $\overline{\mathbf{H}}_j$
[38] 8. Compute $\mathbf{y}_j = argmin_y\|\beta\mathbf{e}_1 - \overline{\mathbf{H}}_j\mathbf{y}\|_2$ and $\mathbf{x}_j^k = \mathbf{x}_0 + \mathbf{Z}_j\mathbf{y}_j$
[39] 9. If $R_j <$ tolerance Stop
[40] If $j = m$, set $\mathbf{x}_0 = \mathbf{x}_m^k$ and goto 1

[41] enddo

[42] In the above, $\mathbf{x}_0$ is the initial guess, $\mathbf{r}_0$ is the initial residual vector, $\|\ \|_2$ is the 2-norm and (,) is the dot product. The initial guess is always the solution of the previous OL

iteration ($\mathbf{x}^{k-1}$) or for the first OL iteration, it is the initial guess of the nonlinear system of equations (in our case it is the free drift velocity field). The Arnoldi process (steps 4–6) forms an orthonormal basis for the Krylov subspace. The $\mathbf{z}_j$ vectors form the basis of this Krylov subspace. $\overline{\mathbf{H}}_j$ is an $(j+1) \times j$ Hessenberg matrix whose entries are the $h_{i,j}$. The preconditioner, often denoted as $\mathbf{M}$, is in some Krylov method implementations a simpler matrix than the system matrix $\mathbf{A}$. For example, *Hunke and Dukowicz* [1997] used a tridiagonal matrix extracted from the system matrix. Here the preconditioning matrix is the whole system matrix. However, instead of iterating to solve for $\mathbf{z}_j$ with a very high precision, only a few iterations (order of 10) of the SOR or LSOR preconditioner are performed [*DeLong and Ortega*, 1995]. For the SOR preconditioner, the elements of the $\mathbf{z}_j$ vector for the p^{th} iteration are relaxed by performing $z^p_{j,r} = z^{p-1}_{j,r} + \omega\, res_r/a_{rr}$. This expression shows that the p^{th} iterate is obtained from the $(p-1)^{th}$ iterate and a correction that depends on the residual for the r^{th} component and on the relaxation parameter ω. In the case of the LSOR preconditioner, once a line is solved (the $z^p_{j,r}$ are obtained), the elements are relaxed by doing $z^p_{j,r} = z^{p-1}_{j,r} + \omega(z^p_{j,r} - z^{p-1}_{j,r})$.

[43] Note that in GMRES, the residual norm R_j is expressed as $\|\beta\, \mathbf{e}_1 - \overline{\mathbf{H}}_j \mathbf{y}\|_2$, where $\mathbf{e}_1$ is a vector for which the elements are all zero except the first one that has a value of one. In step 8, $\mathbf{y}_j$ is the unique vector that minimizes the residual. The vector $\mathbf{y}_j$ is inexpensive to compute because it requires the solution of an $(j+1) \times j$ least squares problem. The GMRES iteration is stopped when the residual norm (R_j) is lower than a chosen tolerance.

[44] Once the iterate $\mathbf{x}^k$ is obtained, the residual norm of the nonlinear system of equations can be calculated according to:

$$R_{nl} = \sqrt{\sum_{q=1}^{n} \left(\sum_{r=1}^{n} \left(a_{qr} x_r^k - b_r \right)^2 \right)}, \qquad (15)$$

where the elements of $\mathbf{A}$ are calculated with $\mathbf{x}^k$.

[45] The SOR preconditioner was implemented by making slight modifications to the SOR solver of *Tremblay and Mysak* [1997]. The LSOR preconditioner solves row-by-row for the u component and column-by-column for the v component with a tridiagonal solver and applies relaxation. Our implementation of GMRES only requires the user to provide the initial guess vector and the $\mathbf{b}$ vector, a preconditioner and a matrix-vector multiplier. Because an SOR or an LSOR preconditioner can be employed, this GMRES code can be easily implemented from an existing sea-ice model making use of existing model routines. This version of GMRES is a publicly available code obtained directly from Yousef Saad (http://www-users.cs.umn.edu/~saad/).

4. Model and the Forcing Fields

[46] The model has a resolution of 10 km and we use a time step of 6 hours. Various yield curves and flow rules defining many nonlinear VP formulations can be used with this platform. The definitions of the bulk and shear viscosities for other formulations than those of *Hibler* [1979] are given in the study of *Zhang and Rothrock* [2005] and that of *Ip et al.* [1991]. The thermodynamics part of the model is the one used by *Tremblay and Mysak* [1997]. The advection is performed at the end of the $k_{\max}$ OL iterations with a simple upstream scheme. A lower resolution (110 km) version of the model is also sometimes used to perform additional tests.

[47] The 10-km land-ocean mask was derived from a merged IBCAO/ETOPO5 topographic data set [*Holland*, 2000]. The data set can be accessed at http://www.ngdc.noaa.gov/mgg/bathymetry/arctic/ibcaorelatedsites.html. The wind stress is calculated from geostrophic winds derived from the National Center for Environmental Prediction and National Center for Atmospheric Research (NCEP/NCAR) 6-hour reanalysis of sea level pressure [*Kalnay et al.*, 1996]. 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 was a 30-year climatological wind stress. The sea-ice model is coupled thermodynamically to a slab ocean model. The thermodynamics are forced by NCEP/NCAR reanalysis of monthly mean surface air temperature. All NCEP/NCAR reanalysis data are found at http://www.cdc.noaa.gov/.

[48] Starting with a constant sea-ice thickness of 1 m and a concentration of 100%, the model was run for 5 years from 1992 to 1996. The fields obtained on 31 December 1996, represent the initial conditions used for the simulations. 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* [1997].

5. Simulation Results

[49] Convergence properties of the numerical scheme are analyzed for the month of January 1997. All real variables were defined as double precision. All simulations were performed on a desktop computer (2 quad Intel(R) Xeon(R) 2.33 GHz, cache of 4096 Kb with a RAM of 3.9 Gb). The fortran compiler is gfortran 4.1.2, 64 bits. The optimization option O3-ffast-math was used for all the runs.

5.1. The Outer Loop

[50] *Ip* [1993] and *Zhang and Hibler* [1997] have shown that a single modified Euler time step does not lead to a VP solution and that the average kinetic energy (KE) of the pack has not converged. To ensure that the average KE has converged and that the stress states lie either inside or on the yield curve, they propose to repeat multiple times the modified Euler time step. *Zhang and Hibler* [1997] refer to these repeated modified Euler steps as pseudo time steps. In the following, the number of OL iterations ($k_{\max}$) is chosen such that the average KE is within 2% of the fully converged average KE. Note that the fully converged average KE is defined as the average KE of the pack after 100 OL iterations. The average KE is given by $\frac{1}{N}\sum_{j=1}^{N} \frac{\rho_i h_j \left(u_j^2 + v_j^2\right)}{2}$, where N is the number of ice-covered grid cells and u and v are the components of velocity interpolated at the tracer point.

[51] The number of iterations needed for the OL also depends on the convergence criterion of the linear solver (the tolerance). For GMRES, this criterion is expressed by the residual norm. To meet the convergence criterion for the KE and to insure computational efficiency, we adopt what we refer to as a progressive tolerance. At the k^{th} OL

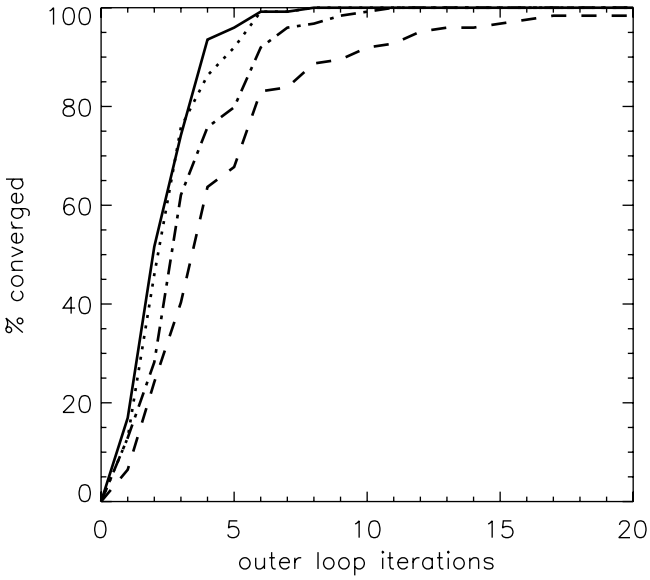

Figure 1. Percentage of time steps that have converged for the month of January 1997 (124 time steps) as a function of the number of OL iterations. Dashed curve: α = 20, dash-dot curve: α = 50, dotted curve: α = 100 and solid curve: α = 200.

iteration, the initial residual norm (R_0^k) of the linearized system of equations is calculated. The linear solver then iterates until the residual norm is lower than R_0^k/α, where α is a free parameter. Figure 1 shows the percentage of time steps for a January 1997 simulation that have converged (within 2% of the fully converged average KE) as a function of the number of OL iterations for different values of α. GMRES-LSOR is used for this sensitivity study. Note that these results are not dependent on the linear solver employed.

[52] Results show that for $\alpha \geq 100$ (α = 200: solid curve, α = 100: dotted curve), only eight OL iterations are needed to achieve the 2% convergence criterion. Similarly, 11 OL iterations are needed for α = 50 (dash-dot curve) and more than 20 are needed for α = 20 (dashed curve). Figure 2 shows, for α = 100, the normalized average KE as a function of the number of OL iterations for two particular time steps: one for which the 2% criterion is achieved in just one OL iteration (dashed curve) and one that requires eight OL iterations to achieve the same criterion (solid curve).

[53] We found that a more efficient way, even at the expense of a few more OL iterations, to achieve the 2% convergence criterion is to start with a low value of α and increase the value of α during the OL iteration process. The 2% convergence criterion can be reached in 10 OL iterations for values of α of: 10, 20, 20, 30, 30, 30, 30, 40, 50, 80. We will refer to this progressive tolerance as the progressive α tolerance. For reference, $R_0^1/10$ (first OL iteration) corresponds approximately to a maximum velocity correction of 5×10^{-3} m s^{-1} for one sweep of the grid when using a stand-alone SOR solver, $R_0^6/30$ (sixth OL iteration) gives a maximum velocity correction of 10^{-6} m s^{-1} and a residual norm of $R_0^{10}/80$ (last OL iteration) gives a maximum velocity correction lower than 10^{-7} m s^{-1}. All the results presented later use the progressive α tolerance with 10 OL iterations unless otherwise specified.

5.2. Optimization

[54] In this section, we investigate the computational efficiencies of GMRES-SOR and GMRES-LSOR with the restart value m, the number of preconditioner iterations k_p and the relaxation parameter. It was found that the optimal k_p and m describe very broad minima of computational efficiency. Slightly different values do not change dramatically the computational efficiency. Accordingly, we use a restart value m of 15 for both GMRES-SOR and GMRES-LSOR methods. The optimal values of k_p for GMRES-LSOR and GMRES-SOR are respectively 10 and 20 iterations (results not shown). The rest of the optimization is done with the relaxation parameter ω. In the following, results from GMRES-LSOR and GMRES-SOR are compared to a stand-alone SOR and a stand-alone LSOR. Note that the four linear solvers converge to the same solution of the linearized system of equations when iterated to a very low value of residual norm (results not shown).

[55] Figure 3 shows the average cpu time per time step, needed to perform 10 OL iterations with the progressive α tolerance, for the month of January 1997, as a function of the relaxation parameter for GMRES-LSOR (solid curve) and GMRES-SOR (dashed curve). The average cpu time per time step for the stand-alone SOR (dotted) and the stand-alone LSOR (dash-dot) are also shown on Figure 3. In order to limit the total integration time, the simulations were done only for the first time step of each day (as opposed to every 6 hours). The initial guess, in all cases, is the free drift velocity field.

[56] For GMRES-LSOR and GMRES-SOR, the average cpu time is weakly dependent on the choice of relaxation parameter, provided the relaxation parameter is smaller than their respective optimal value. *DeLong and Ortega* [1995] found the same behavior for GMRES-SOR for a simple

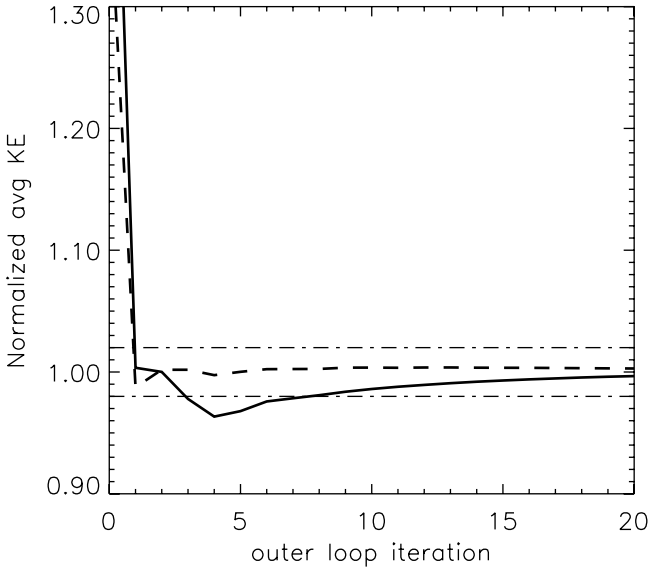

Figure 2. Average KE of the ice pack normalized by the average KE after 100 OL iterations (considered to be the truth) as a function of the number of OL iterations for 4 January 1997 18Z (solid curve) and for 16 January 1997 00Z (dashed curve). The 2% convergence criterion is represented by the two dash-dot lines.

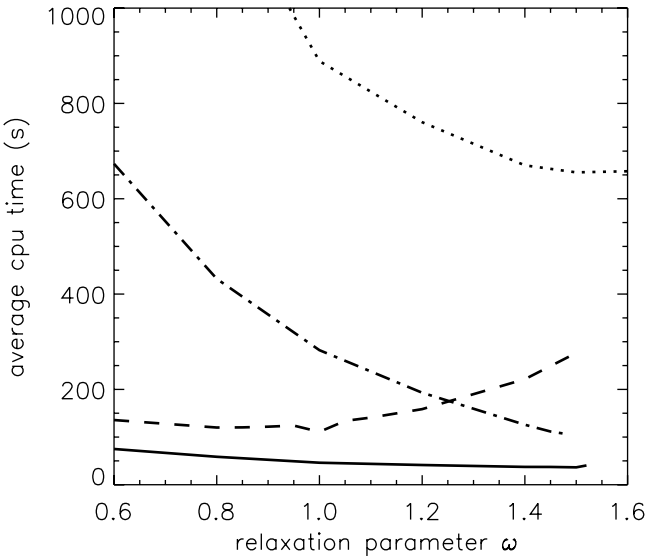

Figure 3. Average cpu time per time step for January 1997 as a function of the relaxation parameter ω. Solid curve: GMRES-LSOR, dashed curve: GMRES-SOR, dash-dot curve: stand-alone LSOR and dotted curve: stand-alone SOR.

convection-diffusion equation. These results suggest that a crude approximation of the optimal relaxation parameter, that is smaller than the optimal value, is almost as good and certainly safer for converging in all cases. On the other hand, the choice of ω for the stand-alone LSOR and SOR solvers is critical as the computational efficiency drops quickly when the relaxation parameter is decreased from the optimal value. For ω greater than 1.52, the GMRES-LSOR simulation crashes while this occurs for $\omega > 1.48$ for the stand-alone LSOR.

[57] The computational efficiency of GMRES-SOR is about the same as that of the stand-alone LSOR when their respective optimal relaxation parameter is used. However, the stand-alone LSOR solver would not be used with the optimal ω for robustness reasons. Also, GMRES-LSOR is about three times faster than the stand-alone LSOR and about 16 times faster than a stand-alone SOR. These results show that LSOR is a better preconditioner than SOR on a serial computer. While the efficiency of the stand-alone SOR and LSOR solvers cannot be further improved, improvements to the GMRES method may still be possible with the use of a different preconditioner (e.g., algebraic multigrid, *Saad* [2003]).

[58] Note that the ice conditions do not affect equally the computational efficiency of each of the solvers. The average cpu time per time step for a 10-day simulations (1–10 January 1997) using GMRES-LSOR and the two stand-alone solvers was calculated for the standard case, a case for which the sea-ice thickness is multiplied by two (referred to as "thick"), and a case for which the sea-ice thickness is divided by two (referred to as "thin"). The relaxation parameter was set to 1.40 for the three solvers. The computational time of the stand-alone LSOR changes by more than a factor of two (2.05) between the "thin" sea-ice cover experiment and the "thick" sea-ice cover experiment (the thicker sea ice leading to a higher average cpu time). For the stand-alone SOR solver, the computational time is multiplied by 2.67 between the "thin" experiment and the "thick" experiment. On the other hand, GMRES-LSOR leads to a cpu time that is weakly dependent on the sea-ice conditions (ratio of 1.26). It was observed that the amount of overrelaxation allowed for the stand-alone LSOR solver is reduced for thicker sea-ice conditions. The fully implicit treatment reduces the diagonal dominance [*Press et al.*, 1989] of the system matrix and probably explains this lack of robustness of the stand-alone LSOR method. The preconditioned GMRES is less sensitive to this issue as GMRES can deal with a non-symmetric system. A 2-year simulation was performed with GMRES-LSOR with the progressive α tolerance and with $\omega = 1.45$ (slightly smaller value than the optimal value) and the simulation ran smoothly.

5.3. Convergence Rate of the Linear Solvers

[59] Using the free drift solution as the initial guess, the convergence properties of the linear solvers are further explored. In Figure 4, the residual norm of GMRES-LSOR, GMRES-SOR, stand-alone LSOR and stand-alone SOR is plotted as a function of the cpu time for the first linear solve (first OL iteration) on 1 January 1997 00Z. The relaxation parameters were set to the optimal values found in section 5.2. In this figure, the behavior of the linear solvers is shown over a wider range of tolerance values (extending to 10^{-10}). Both stand-alone SOR and LSOR solvers exhibit an efficient initial phase followed by a change of slope (more pronounced for the stand-alone SOR solver) when the residual norm reaches a value of about 10. Further reduction in the residual norm is done less efficiently. This behavior is not seen for GMRES-LSOR and GMRES-SOR. The convergence rate is relatively constant throughout the whole iteration process, that is, the amount of time to reduce the

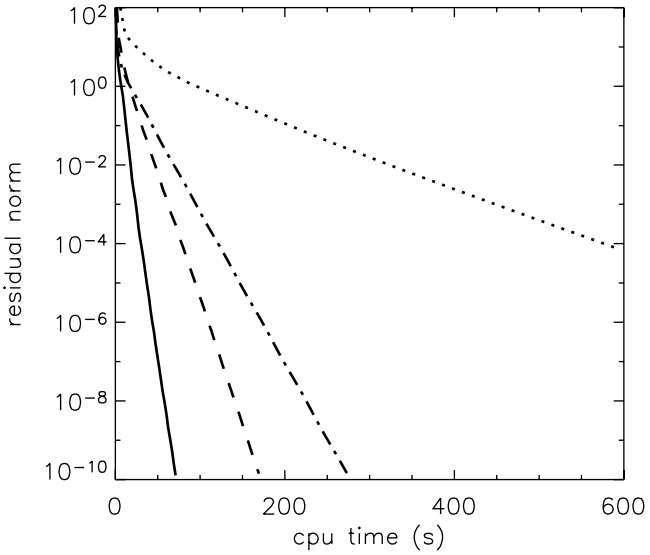

Figure 4. Residual norm (N m^{-2}) as a function of the cpu time for the first linear solve, 1 January 1997 00Z. Solid curve: GMRES-LSOR; dashed curve: GMRES-SOR; dash-dot curve: stand-alone LSOR and dotted curve: stand-alone SOR.

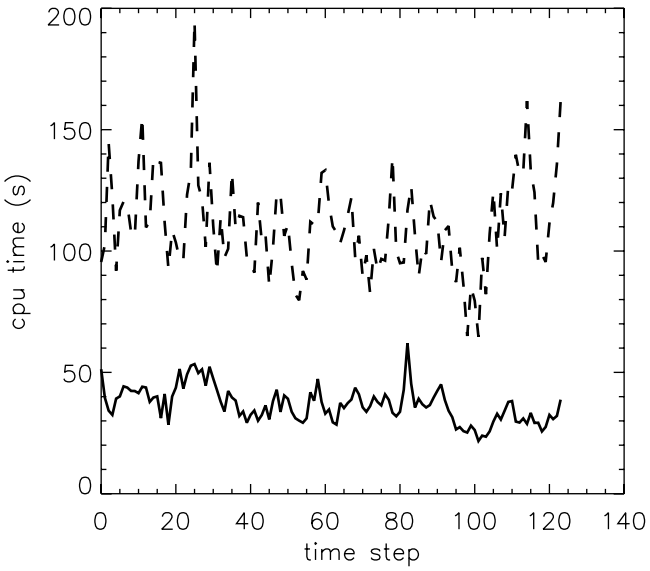

Figure 5. Cpu time per time step using GMRES-LSOR for January 1997 with the free drift velocity field (solid curve) or the previous time step solution (dashed curve) as the initial guess.

residual norm by one decade is constant. These results are robust and irrespective of the time step analyzed.

[60] Stationary iterative methods such as SOR or LSOR rapidly reduce the amplitudes of the residual error Fourier modes associated with the highest wave numbers at the grid scale. These correspond to the largest eigenvalues of the system matrix. However, the large-scale modes are not reduced as quickly with these iterative methods. GMRES is a non-stationary method and is based on an expansion of the residual error as a polynomial. Indeed, the Krylov subspace forms a basis for the space containing these polynomials. The GMRES iteration computes the optimal coefficients for this polynomial in order to minimize the residual [*Liesen and Tichy*, 2004]. In this sense, GMRES works on all the eigenvalues simultaneously. This is the main reason for the better convergence properties of the preconditioned GMRES method compared to the ones of the two stand-alone solvers.

5.4. The Initial Guess

[61] In many studies, the previous time step solution is often used as the initial guess [e.g., *Zhang and Hibler*, 1997]. Some authors also employ the free drift solution [e.g., *Tremblay and Mysak*, 1997]. Figure 5 shows the cpu time per time step using GMRES-LSOR for January 1997, to perform the 10 OL iterations (with the progressive α tolerance and with $\omega = 1.45$). The solid curve is the cpu time when using the free drift solution as the initial guess and the dashed curve is for the previous time step solution as the initial guess. Figure 5 shows that with a 6-hour time step, the model needs less cpu time when using the free drift velocity field than when using the previous time step solution as the initial guess. Figure 5 also demonstrates that the cpu time is much less sensitive to the variations in the forcing when the free drift is used as the initial guess.

[62] We recall that with the free drift velocity field as the initial guess and using the progressive α tolerance, 100% of the solutions obtained achieve the 2% criterion in 10 OL iterations or less. This percentage drops to 92% when using the previous time step solution as the initial guess. This means that the previous time step solution leads to a higher cpu time and the convergence of the KE is not as good as when using the free drift velocity field.

[63] Understanding the convergence properties of the GMRES method for solving a linear system of equations $\mathbf{Ax} = \mathbf{b}$ is non-trivial for a non-normal system matrix. For a normal matrix (recall that a normal matrix satisfies $\mathbf{AA}^T = \mathbf{A}^T\mathbf{A}$), the convergence behavior is solely determined by the distribution of eigenvalues of $\mathbf{A}$ (or of the transformed system matrix if preconditioning is used) in the complex plane. However, for a non-normal system matrix (i.e., our case because of the fully implicit treatment), the convergence rate depends on the eigenvalues but is also influenced by the initial guess and the forcing vector [*Liesen and Tichy*, 2004]. In our case, the situation is further complicated because the linear systems of equations are preconditioned and especially because we are solving a system of nonlinear equations. Consequently, the elements of the system matrix are a function of the initial guess and/or subsequent solution vectors $\mathbf{x}^k$ during the OL iteration process. This implies that the system matrix and the solution vectors $\mathbf{x}^k$ are different whether the free drift velocity field or the previous time step solution is used as the initial guess. Hence the system matrix has a "memory" of the initial guess. When performing a large number of OL iterations (a test was done with 500 with the 110 km model), the two system matrices eventually become identical and the two different cases converge to the same solution.

[64] Finding a theoretical explanation of the better computational efficiency when using the free drift as opposed to the previous time step solution is beyond the scope of this article. However, our tests have shown that the clustering of

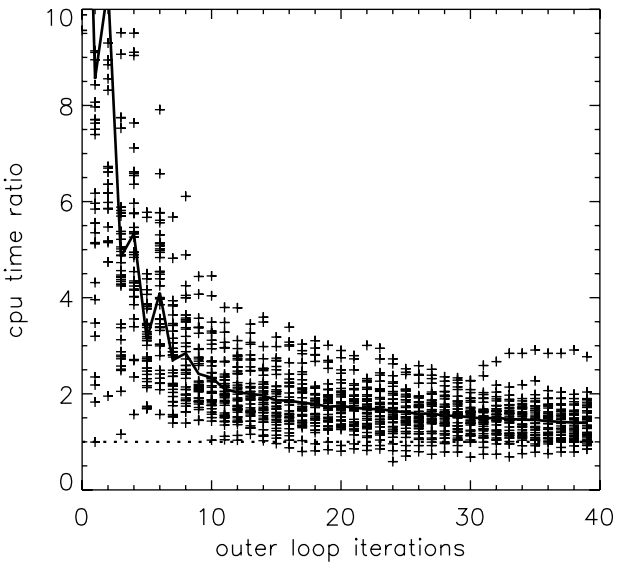

Figure 6. Ratio of the cpu time needed when using the previous time step solution as the initial guess and of the cpu time needed when using the free drift solution as the initial guess for each OL iteration for the first 40 time steps of January 1997. The solid curve is the mean for each OL iteration. The dotted line marks a ratio of 1.0.

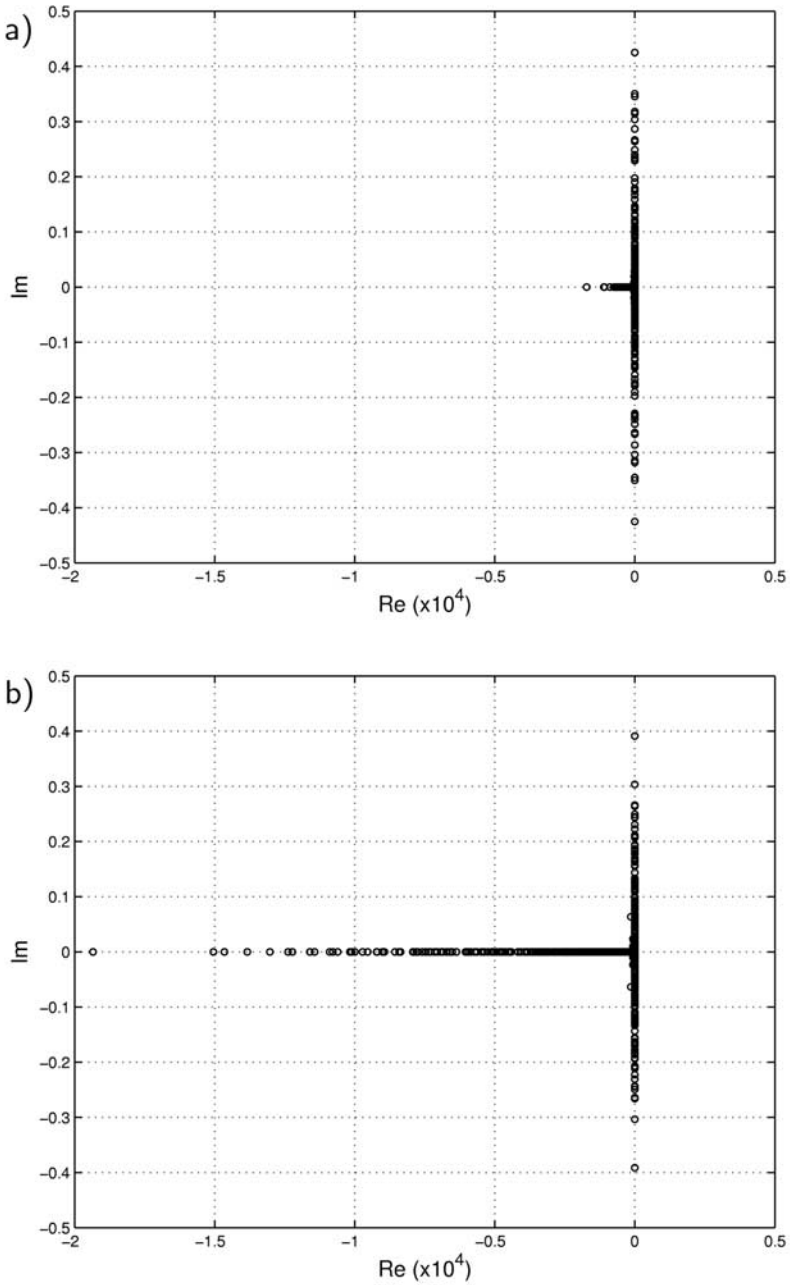

Figure 7. (a) Eigenvalues of the system matrix for the first OL iteration when the free drift velocity field is used as the initial guess. (b) Same as Figure 7a but using the previous time step solution as the initial guess.

the eigenvalues of the system matrix is a good indicator of the convergence properties of GMRES-LSOR for solving the linearized system of equations.

[65] Figure 6 shows the cpu time needed when using the previous time step as the initial guess divided by the cpu time needed when using the free drift solution as the initial guess as a function of the OL iteration. The simulations were done for 40 time steps each using 40 OL iterations. The solid curve is the mean of the cpu time ratio at each OL iteration. Even after 40 OL iterations, the free drift leads to better convergence properties than the previous time step but the gain is more pronounced for the first five OL iterations. Using the 110 km model (to reduce the computational time), we have calculated the eigenvalues of the system matrix. Figures 7a and 7b show the eigenvalues of the system matrix for the first OL iteration when the free drift is used as the initial guess and when the previous time step solution is employed. On these two graphs, the presence of pairs of imaginary eigenvalues indicates the nonsymmetry of the system matrix. These results also demonstrate that for the first OL iteration, the eigenvalues are more clustered together when the free drift is used as the initial guess as compared to the previous time step. The spectra of eigenvalues after 10 OL iterations (not shown) resemble

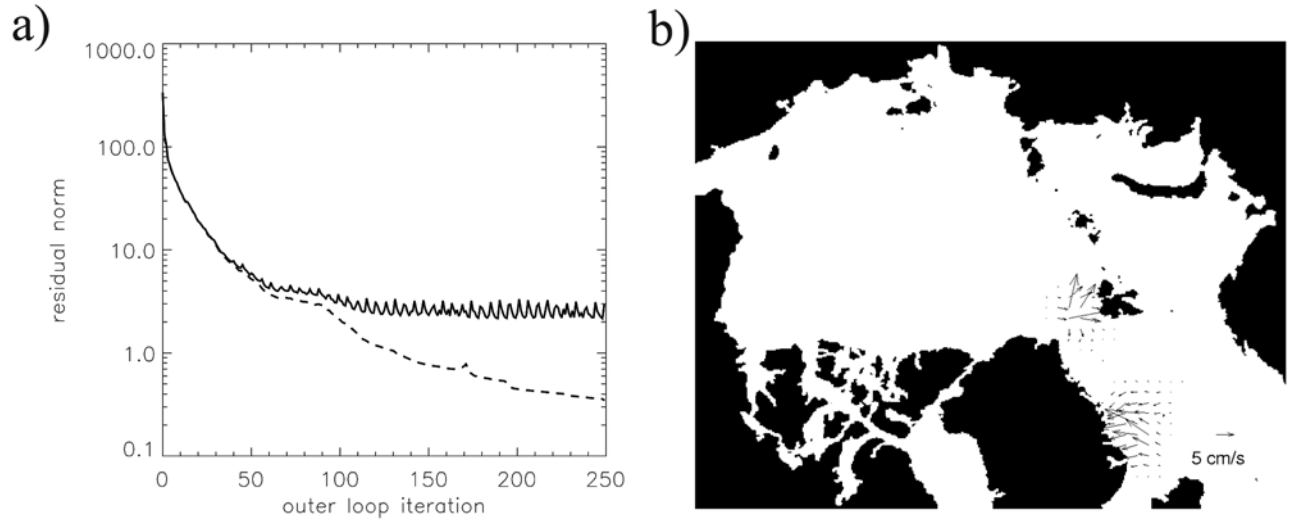

Figure 8. (a) Residual norm (N m^{-2}) of the nonlinear system of equations as a function of the number of OL iterations. Solid curve: partly implicit; dashed curve: fully implicit. Results are for 20 September 1996 00Z. (b) Difference between the velocity field obtained with the partly implicit treatment after 250 OL iterations and the velocity field obtained with the fully implicit treatment after 250 OL iterations on 20 September 1996 00Z.

each other more and explain the more similar convergence properties at this stage of the process. The fact that the eigenvalues are closer to zero and more clustered together for the first OL iterations when the free drift is used is caused by larger deformations in the free drift solution which lead to smaller viscous coefficients (see equations (6) and (7)) as opposed to the ones obtained when the previous time step solution is used. Larger viscous coefficients lead to larger negative eigenvalues. These results indicate that with a 6-hour time step on a 10-km grid, the free drift solution is a better initial guess than the previous time step solution. As the time step is reduced, the previous time step solution will eventually be the best initial guess. We have not determined at what time step this transition occurs (future work).

5.5 Partly Implicit or Fully Implicit Treatment?

[66] As mentioned earlier, symmetry is not a prerequisite for the GMRES method. The linearized equation can therefore be solved fully implicitly. Most numerical methods used to solve the sea-ice momentum equation consider some terms explicitly to increase the diagonal dominance of the system matrix [e.g., *Tremblay and Mysak*, 1997] and also to improve the convergence properties [e.g., *Zhang and Hibler*, 1997]. *Hibler* [1979] considered the Coriolis term implicitly but he treated the momentum advection and the off-diagonal part of the rheology term explicitly. *Tremblay and Mysak* [1997] solved the steady state momentum equation with the Coriolis term and the off-diagonal part of the water drag term written explicitly. In order to completely decouple the u and v equations, *Zhang and Hibler* [1997] treated explicitly all the v terms in the u equation (and vice versa for the v equation). *Zhang and Rothrock* [2000] later introduced the ADI technique to improve the computational efficiency of the method by *Zhang and Hibler* [1997]. Again, the two equations are completely decoupled.

[67] *Stössel et al.* [1996] have found that treating the off-diagonal part of the strain rate tensor explicitly can cause anomalous drifts of up to 6 cm s^{-1} in the case of zero forcing. The difference between a fully implicit treatment and a partly implicit treatment with realistic forcing is investigated in this section. The dashed curve in Figure 8a shows the residual norm as a function of the number of OL iterations for the fully implicit treatment on 20 September 1996 00Z, while the solid curve is for a partly implicit treatment: the Coriolis term and the off-diagonal part of the water drag term are considered explicitly. Note that the residual norm of the nonlinear system of equations is plotted in this figure. It gives an indication of the quality of the converged solution for the nonlinear system of equations. It is obtained at the end of each linear solution with the updated viscosities and relative ice-ocean velocities ($|\mathbf{u}-\mathbf{u}_w^g|$) in the water drag term calculated with the newly calculated velocity field (equation (15)). At the k^{th} OL iteration, the explicit terms in the u equation are written with $(v_{avg}^{k-2} + v_{avg}^{k-1})/2$ (same approach for the v equation). For both simulations, a progressive tolerance of $R_0^k/50$ is used. There is a clear undamped oscillation present in the residual norm of the partly implicit case. This behavior is also observed in the average KE of the pack (not shown). The velocity field obtained after 250 OL iterations with the partly implicit treatment is compared with the one obtained after 250 OL iterations with the fully implicit treatment (considered the truth here because its residual norm is one order of magnitude smaller that the residual norm with the partly implicit treatment). When using the partly implicit treatment, anomalous drifts of up to 14 cm s^{-1} are present along the East Greenland coast and north of Svalbard (see Figure 8b). This behavior is not observed at each time step but depends on the sea-ice conditions and on the wind stress. It occurs more often in summer or at the beginning of fall in regions of low sea-ice concentration and low wind stress. The low

sea-ice concentration makes the rheology term negligible in the momentum balance and reduces the relative importance of the dissipative terms. The low wind stress leads to low relative ice-ocean velocities and therefore to a small ocean drag. When the instability occurs, it causes anomalous drifts that have the same order of magnitude as the drift itself.

[68] The problem is even more critical when the explicit term is written in terms of v_{avg}^{k-1} and u_{avg}^{k-1} (as usually done in the sea-ice modeling community). In this case, the residual norm increases until the model crashes. Note that this instability does not depend on the linear solver employed. Other sea-ice models may have this instability because of the partly implicit treatment [e.g., *Zhang and Hibler*, 1997; *Tremblay and Mysak*, 1997] if no corrector step is used.

[69] Following *Zhang and Hibler* [1997], we perform a stability analysis to understand the problem. We assume a low sea-ice concentration (the rheology term is then negligible), that u and v are collocated (B-grid) for simplicity, and that $C_w = \rho_w C_{dw} |\mathbf{u}_I^k - \mathbf{u}_w^g|$ is constant. With these assumptions and treating the off-diagonal terms explicitly, we get according to equations (10) and (11)

$$\rho_i h f v^{k-1} - C_w \cos\theta_w u^k + C_w \sin\theta_w v^{k-1} = b_u, \tag{16}$$

$$-\rho_i h f u^{k-1} - C_w \cos\theta_w v^k - C_w \sin\theta_w u^{k-1} = b_v. \tag{17}$$

[70] Note that the model is unstable under the same assumptions (except that u and v are not collocated). Substituting $u^k = u_{sol} + \xi_u^k$ and $v^k = v_{sol} + \xi_v^k$ where u_{sol} and v_{sol} represent the true solution, we can write the errors ξ_u^k and ξ_v^k at iteration k as a function of the errors at iteration $k-2$ as:

$$\xi_u^k = -\left(\frac{\rho_i h f + C_w \sin\theta_w}{C_w \cos\theta_w}\right)^2 \xi_u^{k-2}, \tag{18}$$

$$\xi_v^k = -\left(\frac{\rho_i h f + C_w \sin\theta_w}{C_w \cos\theta_w}\right)^2 \xi_v^{k-2}. \tag{19}$$

[71] In order for the numerical scheme to be stable, the absolute value of the error has to decrease with the number of OL iterations. This means that $\frac{\rho_i h f + C_w \sin\theta_w}{C_w \cos\theta_w}$ has to be smaller than one. We find that

$$|\mathbf{u}_I^k - \mathbf{u}_w^g| > \frac{\rho_i h f}{\rho_w C_{dw}(\cos\theta_w - \sin\theta_w)}. \tag{20}$$

[72] For typical values of ρ_i (900 kg m^{-3}), ρ_w (1026 kg m^{-3}), f (1.46×10^{-4}), C_{dw} (5.5×10^{-3}) and using a θ_w of 25°, the scheme is stable if $|\mathbf{u}_I^k - \mathbf{u}_w^g| > 0.048\ h$. Modelers might get around this instability issue by limiting $|\mathbf{u}_I^k - \mathbf{u}_w^g|$ to a minimum value (which then acts as an artificial dissipative term). It is even possible that they never encounter this problem if they use a low number of OL iterations (or of pseudo time steps) but then this raises the question about the level of convergence of the solution to the nonlinear system of equations.

[73] The fact that the model uses a C-grid and that the stability analysis was performed on a B-grid suggests that the problem exists irrespectively of the grid used, when the Coriolis term and the off-diagonal part of the water drag term are treated explicitly. Whether other explicit treatments [e.g., *Hibler*, 1979] lead to a similar instability remains to be investigated. *Zhang and Hibler* [1997] proposed a corrector step in which the Coriolis term and the off-diagonal part of the water drag term are treated implicitly. They either suggested to use the corrector step at the end of each pseudo time step or to perform many pseudo time steps and then to finish with a corrector step. On the basis of the results shown in Figures 8a and 8b, the former approach should be employed. The model of *Tremblay and Mysak* [1997] which treats the Coriolis and the off-diagonal part of the water drag term explicitly should also include a corrector step to eliminate this potential instability.

6. Conclusions

[74] We have implemented the preconditioned GMRES method along with an OL iteration to solve the sea-ice momentum equation. The preconditioned GMRES method is the linear solver. GMRES together with the OL is used to solve the nonlinear momentum equation. We have demonstrated that the preconditioned GMRES method is significantly more computationally efficient than the stand-alone SOR and LSOR solvers on a serial computer when the linearized equation is treated fully implicitly. With 10 OL iterations and the use of a progressive tolerance, it was found that GMRES-LSOR is three times faster than the stand-alone LSOR solver and 16 times faster than the stand-alone SOR solver. It was found that for GMRES-LSOR and GMRES-SOR, the average cpu time is weakly dependent on the value of the relaxation parameter provided it is smaller than the optimal value. This behavior is very different than that of the stand-alone SOR and stand-alone LSOR solvers for which the cpu time depends strongly on the relaxation parameter. It was also found that the cpu time needed by GMRES-LSOR is less sensitive to the sea-ice conditions then the cpu times needed by the two stand-alone solvers.

[75] Our results also demonstrate that the free drift velocity field is a better initial guess than the previous time step solution when using a time step of 6 hours. Furthermore, the cpu time is not really sensitive to the variations in the forcing when the free drift is used as the initial guess. We therefore recommend its use since the calculation of the free drift solution requires a negligible amount of cpu time. As the time step is reduced, the previous time step solution will eventually be the best initial guess. We have not determined at what time step this transition occurs (future work).

[76] The impact of a fully implicit or partly implicit treatment of the linearized momentum equation was also investigated. As it was found from simulation results and from a stability analysis, an undamped oscillation (or instability) can be present in the average KE of the ice pack and in the nonlinear residual norm when treating the Coriolis term and the off-diagonal part of the water drag term explicitly. This oscillation is not present when using a fully implicit treatment. If the partly implicit treatment is used, the nonlinear momentum equation cannot be solved to

low residual norm values unless a corrector step is employed. With this treatment, errors on the drift of the same order of magnitude than the drift itself can still be present. This is a clear advantage of the preconditioned GMRES method; because symmetry is not a prerequisite, the fully implicit treatment can be employed.

[77] Our results have shown that the preconditioned GMRES method is a very efficient linear solver. However, Figure 8a (the dashed curve) suggests that the nonlinear equation is not solved efficiently. Indeed, the residual norm of the nonlinear equation decreases slowly with the number of OL iterations. This result is independent of the linear solver used. In other words, the linear solver can be very efficient (e.g., the preconditioned GMRES method) but a large number of OL iterations need to be performed to solve accurately the nonlinear system of equations. Future work will focus on further investigating the nonlinear convergence on the solution of the momentum equation. Preliminary results suggest that the number of OL iterations (or the number of pseudo time steps) presently used in the modeling community leads to large errors on the drift. This is the case even if a much larger number of OL iterations (than presently used in current sea-ice models) are performed. This raises the question whether sea-ice models are presently iterated to convergence. In order to address these issues, we are currently developing a Jacobian free Newton-Krylov method [*Knoll and Keyes*, 2004]. This will involve a Newton iteration along with the already implemented preconditioned GMRES method. Finally, a parallel version of the code will be developed to take full advantage of the GMRES method presented in this article.

[78] **Acknowledgments.** We would like to thank Paul Tupper from the Department of Mathematics and Statistics at McGill and David Straub from the Department of Atmospheric and Oceanic Sciences at McGill for interesting discussions and for their help in the treatment of boundary conditions. Jean-François Lemieux would like to thank NSERC, FQRNT and the McGill University J. W. McConnell foundation for fellowships received during the course of this project. Bruno Tremblay was supported by a NSERC Discovery grant and by the National Science Foundation Office of Polar Program (OPP-0230325) and Arctic Science Program (ARC-0520496). This work was also supported by an NSERC Discovery grant and NSERC/CFCAS-funded Canadian CLIVAR Research Network grant awarded to Lawrence A. Mysak. Stephen Thomas was supported by the US Department of Energy.

References

da Cunha, R. D., and T. Hopkins (1994), A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations, *Adv. Comput. Math.*, *2*, 261–277.

DeLong, M. A., and J. M. Ortega (1995), SOR as a preconditioner, *Appl. Numer. Math.: Trans. IMACS*, *18*, 431–440.

DeLong, M. A., and J. M. Ortega (1998), SOR as a preconditioner II, *Appl. Numer. Math.: Trans. IMACS*, *26*, 465–481.

Dukowicz, J. (1997), Comments on the stability of the viscous-plastic sea ice rheology, *J. Phys. Oceanogr.*, *27*, 480–481.

Flato, G. M., and W. D. Hibler (1992), Modeling pack ice as a cavitating fluid, *J. Phys. Oceanogr.*, *22*, 626–651.

Freund, R. W. (1999), Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation, in *Applied and Computational Control, Signals, and Circuits*, vol. 1, edited by B. N. Datta, pp. 435–498, Birkhäuser, Boston, Mass.

Hibler, W. D. (1977), A viscous sea ice law as a stochastic average of plasticity, *J. Geophys. Res.*, *82*, 3932–3938.

Hibler, W. D. (1979), A dynamic thermodynamic sea ice model, *J. Phys. Oceanogr.*, *9*, 815–846.

Holland, D. M. (2000), *Merged IBCAO/ETOPO5 Global Topographic Data Product*, Natl. Geophys. Data Cent., Boulder, Colo.

Hunke, E. C., and J. K. Dukowicz (1997), An elastic-viscous-plastic model for sea ice dynamics, *J. Phys. Oceanogr.*, *27*, 1849–1867.

Hunke, E. C., and Y. Zhang (1999), A comparison of sea ice dynamics models at high resolution, *Mon. Weather Rev.*, *127*, 396–408.

Ip, C. F. (1993), Numerical investigation of different rheologies on sea-ice dynamics, Ph.D. thesis, Darmouth College, Hanover, N. H.

Ip, C. F., W. D. Hibler, and G. M. Flato (1991), On the effect of rheology on seasonal sea-ice simulations, *Ann. Glaciol.*, *15*, 17–25.

Kalnay, E., et al. (1996), The NCEP/NCAR 40-year reanalysis project, *Bull. Am. Meteorol. Soc.*, *77*, 437–470.

Knoll, D. A., and D. E. Keyes (2004), Newton-Krylov methods: A survey of approaches and applications, *J. Comput. Phys.*, *193*, 357–397.

Liesen, J., and P. Tichy (2004), Convergence analysis of Krylov subspace methods, *GAMM Mitt.*, *27*, 153–173.

Maslowski, W., and W. H. Lipscomb (2003), High resolution simulations of Arctic sea ice, 1979–1993, *Polar Res.*, *22*, 67–74.

McPhee, M. G. (1975), Ice-ocean momentum transfer for the AIDJEX ice model, *AIDJEX Bull.*, *29*, 93–111.

Oberhuber, J. (1993), Simulation of the Atlantic circulation with a coupled sea ice-mixed layer-isopycnal general circulation model. Part I: Model description, *J. Phys. Oceanogr.*, *23*, 808–829.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1989), *Numerical Recipes, The Art of Scientific Computing*, Cambridge Univ. Press, New York.

Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, Pa.

Saad, Y., and M. H. Schultz (1986), GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, *7*, 856–859.

Stössel, A., J. M. Oberhuber, and E. Maier-Reimer (1996), On the representation of sea ice in global ocean general circulation models, *J. Geophys. Res.*, *101*, 18,193–18,212.

Tremblay, L.-B., and L. A. Mysak (1997), Modeling sea ice as a granular material, including the dilatancy effect, *J. Phys. Oceanogr.*, *27*, 2342–2360.

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

Zhang, J., and D. A. Rothrock (2000), Modeling Arctic sea ice with an efficient plastic solution, *J. Geophys. Res.*, *105*, 3325–3338.

Zhang, J., and D. A. Rothrock (2005), Effect of sea ice rheology in numerical investigations of climate, *J. Geophys. Res.*, *110*, C08014, doi:10.1029/2004JC002599.

J.-F. Lemieux, L. A. Mysak, J. Sedláček, and B. Tremblay, Department of Atmospheric and Oceanic Sciences, McGill University, 805 Sherbrooke Street West, Montréal, QC, Canada H3A 2K6. (lemieux@zephyr.meteo.mcgill.ca)

S. Thomas, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439-4803, USA.