Available online at www.sciencedirect.com
ScienceDirect
Ocean Modelling 21 (2008) 97–105
Ocean Modelling
www.elsevier.com/locate/ocemod

Fast dynamical spin-up of ocean general circulation models using Newton–Krylov methods

Timothy M. Merlis [a,*], Samar Khatiwala [b]

[a] Department of Applied Physics and Mathematics, Columbia University, New York, NY 10027, USA
[b] Lamont–Doherty Earth Observatory, Columbia University, Palisades, NY 10964, USA

Received 26 September 2007; received in revised form 19 December 2007; accepted 19 December 2007
Available online 31 December 2007

Abstract

Numerical models of the ocean play an important role in efforts to understand past climate variability and predict future climate changes. In many studies, ocean models are driven by forcings that are either time-independent or vary periodically (seasonally) and it is often highly desirable or even essential to obtain equilibrium solutions of the model. Existing methods, based on the simple, expedient idea of integrating the model until the transients have died out, are too expensive to use routinely because the ocean takes several thousand years to equilibrate. Here, we present a novel approach for efficiently computing equilibrium solutions of ocean models. Our general approach is to formulate the problem as a large system of nonlinear algebraic equations to be solved with a class of methods known as matrix-free Newton–Krylov, a combination of Newton-type methods for superlinearly convergent solution of nonlinear equations, and Krylov subspace methods for solving the Newton correction equations. As an initial demonstration of the feasibility of this approach, we apply it to find the equilibrium solutions of a quasi-geostrophic ocean model for both steady forcing and seasonally-varying forcing. We show that the matrix-free Newton–Krylov method converges to the solutions obtained by direct time integration of the model, but at a computational cost that is between 10 and 100 times smaller than direct integration. A key advantage of our approach is that it can be applied to any existing time-stepping code, including ocean general circulation models and biogeochemical models. However, effective preconditioning of the linear equations to be solved during the Newton iteration remains a challenge.

1. Introduction

Ocean general circulation models are an essential tool in climate research. Commonly, such models are forced by time-independent or periodically (seasonally) varying surface fluxes, and for a variety of problems, it is highly desirable or even essential to compute equilibrium solutions of the model. For instance, in order to more accurately simulate changes in climate due to changes in forcing, a model should be initialized from a quasi-equilibrium state of the ocean model to prevent "model drift" that may otherwise confuse the results. Similarly, equilibrium solutions of the ocean model are needed to simulate (steady state) biogeochemical tracers, to systematically explore sensitivity to parameters and sub-grid scale parameterizations, and to understand past climate regimes (such as that of the last glacial period).

An equilibrium solution (time-periodic or steady) of a dynamical system can be obtained by integrating from an appropriate initial value until the transients have died out. If the transients have disappeared after a few periods, this is the simplest and most efficient approach for the computation of the steady state solution. The time-scale for dynamical adjustment of the ocean is, however, rather long. Computing an equilibrium solution by direct integration of an ocean general circulation model (GCM) requires "running" the model for several thousand years rendering this brute force approach impractical, at least on a routine basis. For instance, given typical allocations of supercomputing resources at U.S. climate modeling centers and

* Corresponding author. Present address: California Institute of Technology, Pasadena, CA, USA. Tel.: +1 626 395 6982; fax: +1 626 585 1917.
E-mail address: tmerlis@caltech.edu (T.M. Merlis).

1463-5003/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ocemod.2007.12.001

conventional schemes for accelerating dynamical spin-up, it takes on the order of 20–30 days of wall-clock time to integrate a state-of-the-art ocean GCM to equilibrium (Dr. Gokhan Danabasoglu, National Center for Atmospheric Research, personal communication).

A number of ad-hoc acceleration techniques have been proposed for the dynamical adjustment problem. The most widely applied method uses unequal time-steps in the tracer and momentum equations (Bryan and Lewis, 1979; Bryan, 1984; Danabasoglu et al., 1994). This use of "distorted physics" or "asynchronous" time integration has the effect of slowing down gravity and Rossby waves while speeding up abyssal processes (by decreasing the local heat capacity). While this can considerably speed-up the approach to steady state, it still requires long integrations. Additionally, the distortion of time-scales caused by the acceleration can severely alter the seasonal cycle (e.g., Wang, 2001; Huang and Pedlosky, 2003; Danabasoglu, 2004), and it is customary to extend the asynchronous simulation with a shorter synchronous integration that adds to the computational burden. A more recent development is the use of implicit time-stepping in ocean models. The nonlinear solver required to implement an implicit scheme can also be used to directly obtain steady state solutions, as has been demonstrated by Dijkstra et al. (2001) and Weijer et al. (2003) in an ocean model constructed specifically for this purpose. While promising, there are several limitations of this approach, most notably the fact that this approach cannot deal with time-dependent forcing. Seasonality in forcing has a strong impact on the response, rectifying it in ways that fundamentally changes the nature of the equilibrium response. It is also not clear how this method can be applied to existing ocean GCM codes.

Here, we propose a novel solution to the dynamical spin-up problem. Our general approach, described below, is to formulate the problem as solving a nonlinear equation of the form $\mathbf{F}(\mathbf{u}) = \mathbf{0}$. To solve this equation, we apply a class of methods known as matrix-free Newton–Krylov (MFNK), a combination of Newton-type methods for superlinearly convergent solution of nonlinear equations, and Krylov subspace methods for solving the Newton correction equations (see Knoll and Keyes (2004) for a recent review). The basic link between the two methods is the Jacobian-vector product, which may be probed approximately without forming and storing the elements of the true Jacobian. The "matrix-free" nature of this approach makes it extremely flexible, allowing its use with any ocean model. As an initial demonstration of the feasibility of this approach, and to illustrate its use, we apply it to a single-layer quasi-geostrophic ocean model on the beta-plane. The resolution is low enough to have well-defined equilibrium[1] solutions for both steady forcing and seasonally-varying forcing. Though this model is much simpler than a primitive equation GCM, it is a relevant demonstration because it includes the advective nonlinearity of the full equations of motion, and we test it at resolutions approaching that of a coarse resolution GCM.

[1] Throughout the text, the term "equilibrium" will be used for both steady (time-independent) and seasonally (time-dependent) forced problems. For seasonally forcing, the equilibrium solution may be called a periodic orbit, limit cycle, or cyclo-stationary.

2. Matrix-free Newton–Krylov for time-steppers

Our approach to computing the periodic steady state solution of an ocean model involves finding an initial condition such that the solution at the end of the period exactly matches the initial condition. More precisely, let $\mathbf{u}(t)$ represent the state of the OGCM (i.e., a vector of all the prognostic fields) at time t. Given $\mathbf{u}(t=0)$, the ocean model solves an initial value problem to predict the state of the system at any future time $t \geqslant 0$. We write this as

$$\mathbf{u}(t) = \boldsymbol{\Phi}(\mathbf{u}(0), t), \tag{1}$$

where $\boldsymbol{\Phi}$ is the state-transition function of the system of differential equations governing the model. In this "state-space" representation, the OGCM can be viewed as a nonlinear mapping $\boldsymbol{\Phi}$ from $\mathbf{R}^N$ to $\mathbf{R}^N$. (For a typical global model, the dimension of the state-space, $N = \mathrm{O}(10^6)$.) The steady state problem can now be formulated as follows. We seek a state $\mathbf{u}$ that repeats after time T, where T is the (known) period, i.e., $\mathbf{u}(T) = \mathbf{u}(0)$, which we write as

$$\mathbf{F}(\mathbf{u}) = \mathbf{u} - \boldsymbol{\Phi}(\mathbf{u}, T) = \mathbf{0}. \tag{2}$$

Note that since the periodically forced system is nonautonomous, fixing the definition of $t = 0$ (e.g., January 1) renders the solution unique. When the forcing is time-independent, the problem can still be formulated as above, but Eq. (2) must now be valid for any $T > 0$. We have thus transformed the problem into a nonlinear algebraic system. Interestingly, this idea appears to be well known in the solution of ordinary differential equations (ODEs) (Deuflhard, 2004), in particular, those arising in electrical engineering (e.g., Aprille and Trick, 1972; Skelboe, 1980; Telichevesky et al., 1995). (The problem there is to find the periodic steady state response of analog circuits described by a system of nonlinear ODEs.) However, we are unaware of applications in other areas or to partial differential equations (PDEs).

One of the most common methods for solving a system of nonlinear equations of the form $\mathbf{F}(\mathbf{u}) = \mathbf{0}$ is Newton's method (Kelley, 1995; Dennis and Schnabel, 1996). The basic idea is to construct a local linear model of the function $\mathbf{F}$ and then iterate. Algorithmically: given an initial guess, $\mathbf{u}_0$, repeat for $k = 0, 1, \ldots$ until convergence

$$(1) \quad \text{Solve } \mathbf{J}(\mathbf{u}_k)\boldsymbol{\delta}\mathbf{u}_k = -\mathbf{F}(\mathbf{u}_k), \tag{3}$$

$$(2) \quad \mathbf{u}_{k+1} = \mathbf{u}_k + \boldsymbol{\delta}\mathbf{u}_k, \tag{4}$$

where $\mathbf{J}$ is the Jacobian matrix ($J_{ij} = \partial F_i/\partial u_j$). The iteration is terminated when the norm of the nonlinear residual, $\mathbf{F}$, falls below a specified value. One of the main attractions of this method is that it is quadratically convergent. (See

Kelley (1995) for a detailed discussion of the local convergence theory of this method.)

Much of the computational effort required in Newton's method is in solving the linear system Eq. (3). In the present case, the Jacobian matrix is both large ($N \times N$) and dense (since the influence of the initial condition is spread throughout the domain in 1 period). Explicit storage and factorization of **J**, as would be required for a direct solution method, is therefore quite impractical. Instead, we consider the use of iterative methods based on Krylov subspace projection (Barrett et al., 1994; Saad, 2003). Krylov methods have the property that the coefficient matrix is only accessed via matrix–vector products. (Methods in which the coefficient matrix is not formed or stored explicitly, are known as "matrix-free".) In the present context this implies that only the action of the Jacobian **J** times a vector is required. But this product, $\mathbf{J}\boldsymbol{\delta}\mathbf{u}$, is simply the directional derivative of **F** and can be approximated by a difference quotient of the form (Chan and Jackson, 1984; Brown and Saad, 1990)

$$\mathbf{J}(\mathbf{u})\boldsymbol{\delta}\mathbf{u} \approx \frac{\mathbf{F}(\mathbf{u}+\sigma\boldsymbol{\delta}\mathbf{u}) - \mathbf{F}(\mathbf{u})}{\sigma}, \tag{5}$$

where the differencing parameter σ is chosen dynamically (Brown, 1987; Brown and Saad, 1990). This has the advantage of requiring only function evaluations and avoiding explicit computation and storage of the Jacobian matrix. The overall nonlinear solver is termed matrix-free Newton–Krylov (MFNK) or Jacobian-free Newton–Krylov (JFNK).

Note that, in contrast to direct methods, iterative schemes only yield an approximate solution. That is, Eq. (3) for the Newton step is effectively replaced by what is known as the inexact Newton condition

$$\|\mathbf{J}(\mathbf{u}_k)\boldsymbol{\delta}\mathbf{u}_k + \mathbf{F}(\mathbf{u}_k)\| \leqslant \eta_k \|\mathbf{F}(\mathbf{u}_k)\|. \tag{6}$$

(For this reason, such schemes are also known as "inexact Newton" or "truncated Newton" methods (Dembo et al., 1982; Dembo and Steihaug, 1983).) The parameter η_k is known as the forcing term and is a measure of the tradeoff between the effort required to solve the linear system to a tight tolerance and the number of nonlinear iterations.

When applying this method to an ocean model, each "function evaluation" (Eq. (2)) requires a forward integration of the GCM with the given initial condition for a period T. The difference between the final solution (after time T) and the initial condition is the desired function value **F**. As a result, the MFNK solver needs to know nothing of the dynamical equations that are being advanced by the GCM, enabling the same MFNK software (see Appendix for details) to be applied to different GCMs without modification.

3. An illustrative example

3.1. Model formulation

To illustrate the MFNK method, we apply it to a single-layer (barotropic) quasi-geostrophic (QG) ocean basin forced by surface Ekman pumping and damped by a biharmonic viscosity term. The nondimensional governing equation for the streamfunction ψ is

$$\frac{\partial \nabla^2 \psi}{\partial t} + \delta_i^2 J(\psi, \nabla^2\psi + y) = w_e(x,y,t) + \delta_m^3 \nabla^4 \psi, \tag{7}$$

where, δ_m and δ_i are nondimensional scales for the "Munk" and inertial boundary layers, respectively, and w_e is the prescribed Ekman pumping velocity. The damping term is harmonic viscosity acting on the relative vorticity (as $\nabla^2\psi = \zeta$) of the flow. (For more details on this formulation see Pedlosky (1996), Cessi and Ierley (1995).) The lateral boundary condition is no-slip and the aspect ratio of the domain is 1. The inertial scale, δ_i, is 0.02 in all cases, while the viscosity, δ_m, is tuned at each grid resolution to permit as close to inviscid dynamics as the numerical discretization will allow for a fixed time-step. As is conventional, the surface forcing is a westerly windstress pattern that gives a profile of Ekman pumping that varies sinusoidally with latitude

$$w_e(x,y,t) = -\sin(\pi y)[1 + \epsilon\cos(\omega t)], \tag{8}$$

with $\epsilon = 0$ in the steady forcing experiments, and $\epsilon = 0.25$ in the seasonally-forced simulations (with a period of 1 year).

The equations are discretized in space with a second-order centered finite-difference scheme that is energy and enstrophy conserving (Arakawa, 1966). A second-order Runge–Kutta scheme with a (nondimensional) time-step of 0.1 (corresponding to a dimensional time-step of $\sim$1/3000 year) is used to integrate the resulting system of ODEs.

This model with time-independent Ekman pumping has been extensively studied before. For example, Sheremet et al. (1997) computed steady solutions over the parameter space spanned by δ_i and δ_m and Kamenkovich et al. (1995) computed time-dependent solutions of Eq. (7) and showed that the flows develop eddies as the ratio of δ_i to δ_m increases. While our goal here is not to reproduce those calculations, we note that the spatial discretization that we use is inadequate to study the eddying solutions which limits our computations to the $\delta_m < \delta_i$ region of parameter space (Kamenkovich et al., 1995). As stated above, we choose the minimum value of δ_m for which the model can be stably integrated in time which is typically $\delta_m \approx \delta_i$. This issue is further discussed in Section 3.2.3.

3.2. Results

We performed two sets of experiments with the QG model, one with steady forcing, and the other with seasonally-varying forcing. For each set, we investigate the performance of the MFNK relative to direct time integration (DI) as a function of spatial resolution, which we vary from 32×32 grid points to 256×256 points. In an additional experiment, we show that the MFNK method is capable of computing solutions at lower levels of viscosity

than possible by direct integration. In all simulations, we start from an initial condition/iterate of $\psi = 0$.

3.2.1. Steady Ekman forcing

In the first set of experiments, the surface Ekman forcing is time-independent. Fig. 1 (left panel) shows the equilibrium solution obtained using direct integration at the highest resolution (256^2). The right panel shows the difference between the corresponding MFNK and DI solutions. As is evident, the MFNK solution converges to the DI solution, with a difference approaching machine precision.

To compare the computational efficiency of the two methods, we recorded the total number of time-steps the QG model was integrated to reach a given residual (defined as the vector 2-norm of **F**). This is in fact the quantity the Newton method uses to determine convergence. (The time-stepper cannot achieve a residual lower than $\sim 10^{-13}$ because of machine round-off.) While there is a small computational cost associated with the MFNK algorithm itself, the total expense is dominated by that of evaluating the residual function. Since each function evaluation involves integrating the model for a fixed number of time-steps, the total number of time-steps the model was integrated is an accurate measure of the computational expense of the MFNK method, allowing us to directly compare it with the DI method. As discussed in Section 2, for time-independent forcing, the time T in Eq. (2) is a free parameter. The results presented here were obtained with $T = 0.1$, i.e., 1 time-step.

Fig. 2 compares the convergence of the DI and MFNK schemes for different spatial resolutions. We also display the speed-up of the MFNK method relative to DI, i.e. the ratio of the integration time required to reach a given residual with the DI method to that needed by MFNK. In the figures, the speed-up is plotted at each Newton step to demonstrate the computational savings possible at less strict definitions of equilibrium that may be sufficient for some applications. It is clear from the plots that MFNK is significantly more efficient than DI at all resolutions that we have tested. However, the speed-up does decrease as the resolution is increased. Thus, at the lowest resolution, 32^2, MFNK is roughly 1000 times faster than DI, while at the highest resolution, 256^2, it is about 100 times more efficient.

There are two competing factors that affect the speed-up as the resolution increases. First, each linear solve in the MFNK method requires more Krylov iterations to converge, which increases the number of function evaluations. Second, the viscosity is reduced so direct integration takes more time to converge. In our experiments, the number of Krylov iterations grew faster than the model adjustment time, hence reducing the speed-up as resolution increases. Nonetheless, the speed-up on the largest grid is still quite substantial.

As mentioned above, the time T for which the model is integrated to evaluate the residual function is a free parameter. To investigate how the convergence of the MFNK method depends on it, we tried $T = 0.1$, 1, and 10, and found (Fig. 3) that convergence is fastest for the smallest value of T. In Fig. 3, the residuals are rescaled by a factor of $1/T$ to account for the dependence of the residual on the integration length T in Eq. (2). The total integration required for convergence is less for smaller values of T in spite of the fact that for smaller values of T, the linear solves took more iterations (as convergence theory predicts (Kelley et al., 2004)). (The number of iterations for linear solves is not shown here, but can be inferred from the spacing between Newton iterations along the integration time axis.) However, the advantage of using a smaller T diminishes on the larger grids. For example, with the 256^2 grid, the curves for $T = 0.1$ and $T = 1$ are co-incident.

3.2.2. Seasonal Ekman forcing

We now turn to the periodically forced problem. Fig. 4 (top left panel) shows the spun-up streamfunction at the start of the forcing period obtained using direct integration at the highest resolution (256^2) and (bottom left panel) the time series of the domain integrated (nondimensional) kinetic energy over the period. The right panels show the difference between the DI and MFNK solutions. We emphasize that MFNK does not actually compute a periodic orbit. What it computes is an initial condition that,

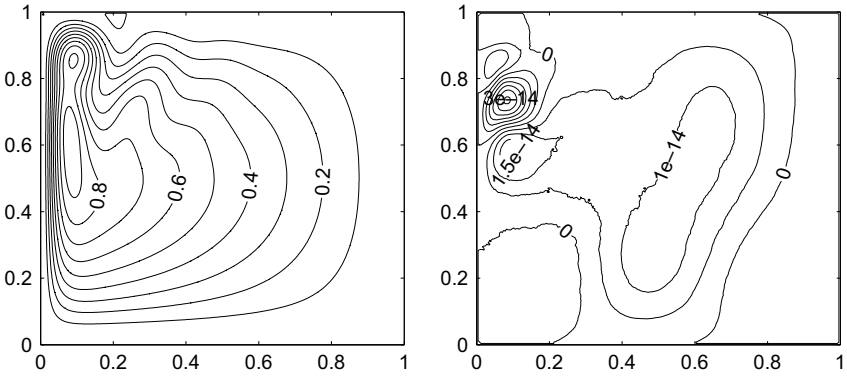

Fig. 1. Left: Equilibrium streamfunction for steady Ekman pumping on 256^2 grid with $\delta_m = 0.0232$ and $\delta_i = 0.02$ obtained using direct integration. Right: Difference between the MFNK and DI solutions.

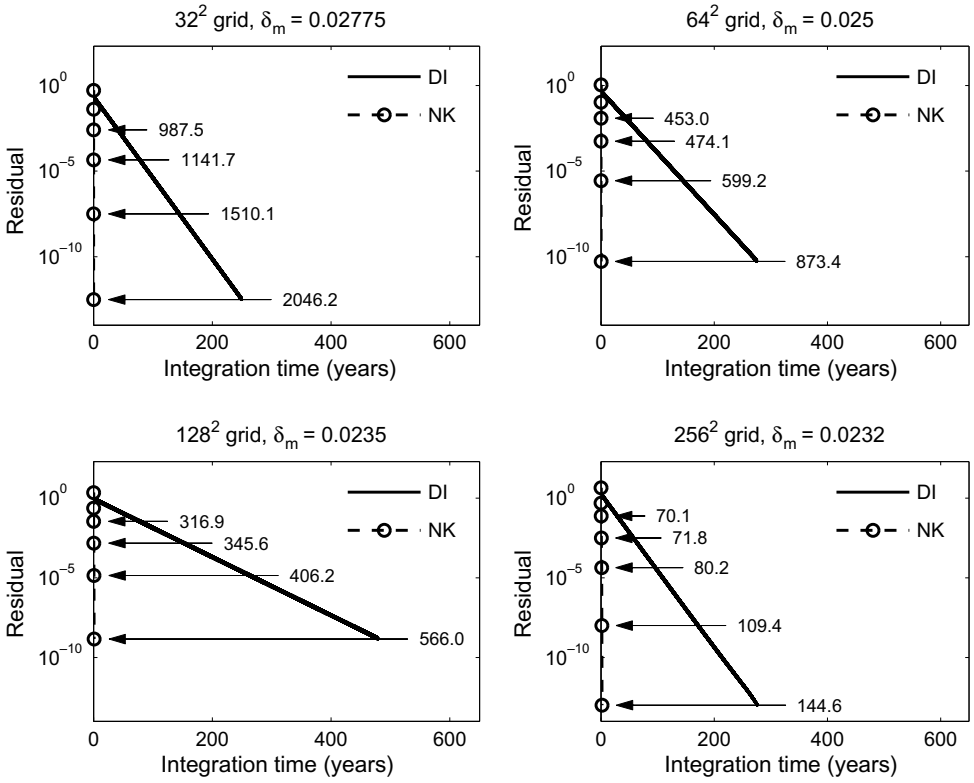

Fig. 2. Comparison of DI and MFNK method for the steady forcing problem. Circles mark each Newton iteration. Numbers next to arrows are the speed-up of the MFNK method relative to DI, computed as the ratio (DI/MFNK) of the number of time-steps to reach the same residual.

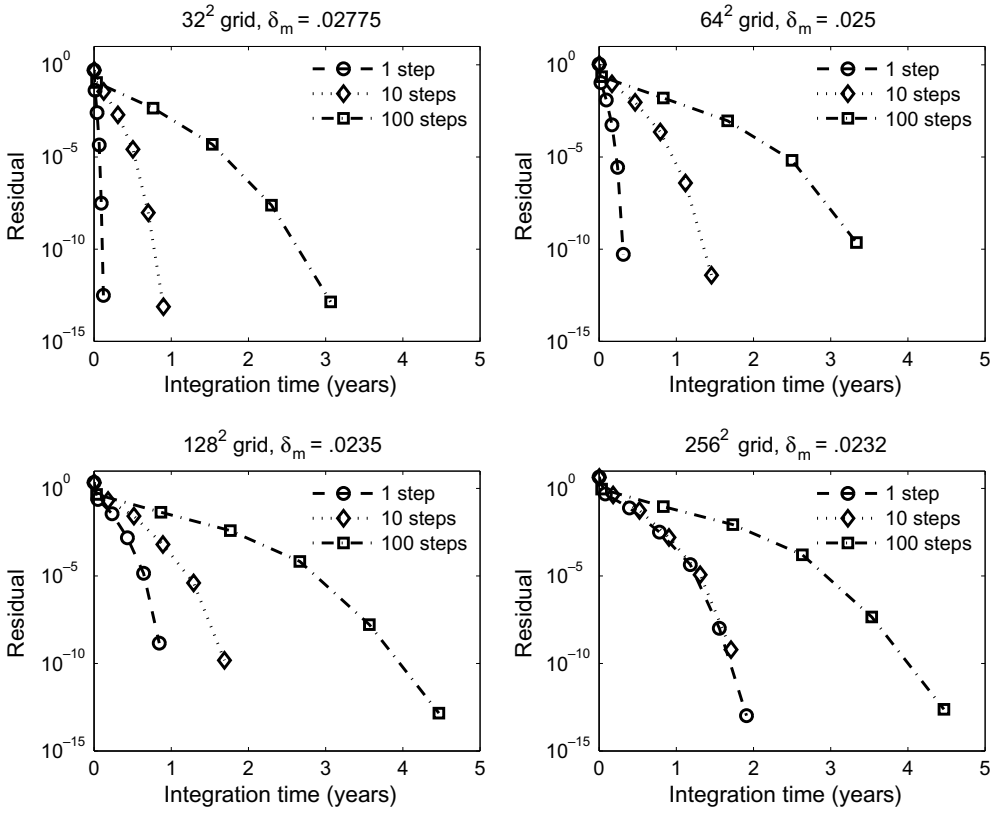

Fig. 3. Convergence of the MFNK method for the steady forcing problem for $T = 0.1$ (1 time-step), $T = 1$ (10 time-steps), and $T = 10$ (100 time-steps). Symbols mark each Newton iteration. Different panels correspond to different resolution and viscosity.

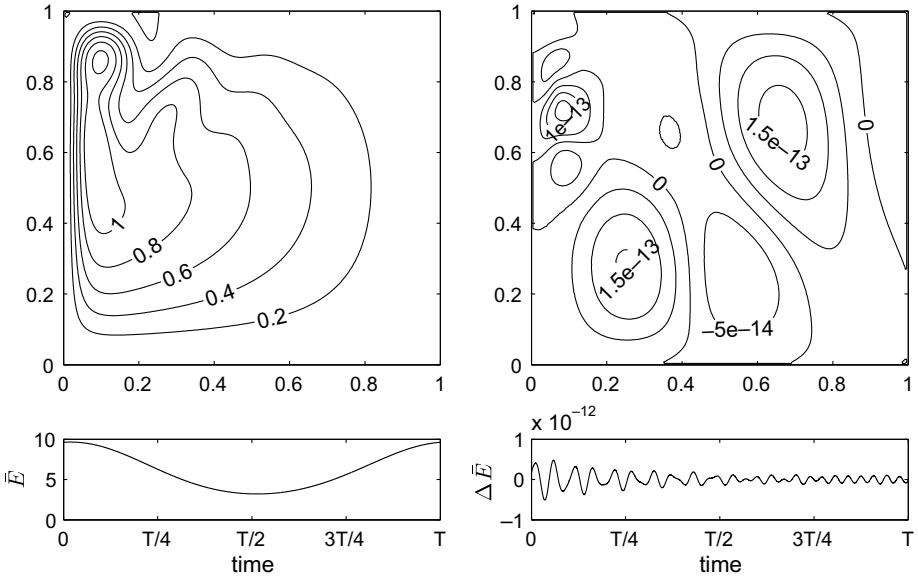

Fig. 4. Top left: Instantaneous, equilibrated streamfunction at the start of the forcing period for periodic Ekman pumping as computed by direct integration. Results are for a 256^2 grid with $\delta_m = 0.0232$ and $\delta_i = 0.02$. Top right: Difference between the MFNK and DI solutions at the start of the forcing period. Bottom left: Time series of domain integrated kinetic energy over forcing period computed by direct integration. Bottom right: Difference between MFNK and DI kinetic energy time series.

when "inserted" into the time-stepping code, will repeat after one period. Thus, to obtain the time series of energy corresponding to MFNK, we simply used the MFNK solution as an initial condition for the time-stepping model and integrated the model for one period. Evidently, as for the steady problem, the MFNK solution converges to the DI solution.

Fig. 5 compares the convergence of the DI and MFNK methods for the periodic problem. Compared with direct integration, the MFNK method is about 10 times faster, converging in between 19 and 31 years at all grid resolutions. The linear solves converged more rapidly in this problem, requiring no more than six Krylov iterations to converge. In the steady problem, at least 22 iterations were needed. The speed-up of MFNK over DI is somewhat larger at higher resolution. This dependence on resolution is largely a consequence of the fact that at higher resolution, the damping is smaller and DI takes longer to converge. In contrast, the integration time needed for MFNK was insensitive to resolution.

In the steady forcing case, the length of time for which the model is integrated to evaluate the residual function is a free parameter. Here, this time is restricted to an integer multiple of the forcing period. This corresponds to replacing T in Eq. (2) with nT where $n = 1, 2, \ldots$. Because the algorithm converged quite rapidly when integrating for one period (which will not be generically true), it is unlikely that the overall integration time can be reduced by choosing $n > 1$. To verify this, we applied MFNK on a 64^2 grid using $2T$ per Newton–Krylov iteration. The algorithm converged after 19 combined NK iterations (38 periods of integration) compared to 21 iterations (21 periods) for $n = 1$ (Fig. 5).

3.2.3. Towards the inviscid limit: Steady Ekman forcing

In the calculations of the preceding sections, care was taken to ensure that the viscosity was large enough to ensure numerical stability and the solutions were stable. The viscosity used is not the minimum for which steady solutions exist (Sheremet et al., 1997), but is at the boundary for which steady solutions are stable (Kamenkovich et al., 1995). As the viscosity decreased, steady solutions of Eq. (7) are unstable and form eddies. Kamenkovich et al. (1995) have pointed out that Arakawa's discretization of the Jacobian term in Eq. (7) is inadequate to compute time-dependent solutions in the eddying regime. However, we have found that the MFNK algorithm is able to converge to steady solutions at levels of viscosity that are below those for which eddies form. In this section, we present some of the MFNK-computed solutions for steady Ekman forcing at lower viscosities.

Fig. 6 shows the MFNK-computed streamfunction for different values of viscosity on a 128^2 grid. As viscosity is reduced, the solutions start to more closely resemble those obtained at higher resolution, for example, Fig. 1 (left panel). (We note that the $\delta_m = 0.02$ solution is quite similar to Fig. 1a in Ierley (1987), with differences possibly arising from different resolutions.) When the streamfunction computed by MFNK is integrated forward in time, the trajectory diverges from the steady solution, which is an explicit demonstration that it is unstable.

As the viscosity was reduced, the NK algorithm required more iterations to converge. For example, it took approximately twice as many iterations to compute the solution for the lowest value of δ_m in Fig. 6 than to compute the solution for the highest value of δ_m in Fig. 6. The MFNK method cannot efficiently calculate solutions for viscosities

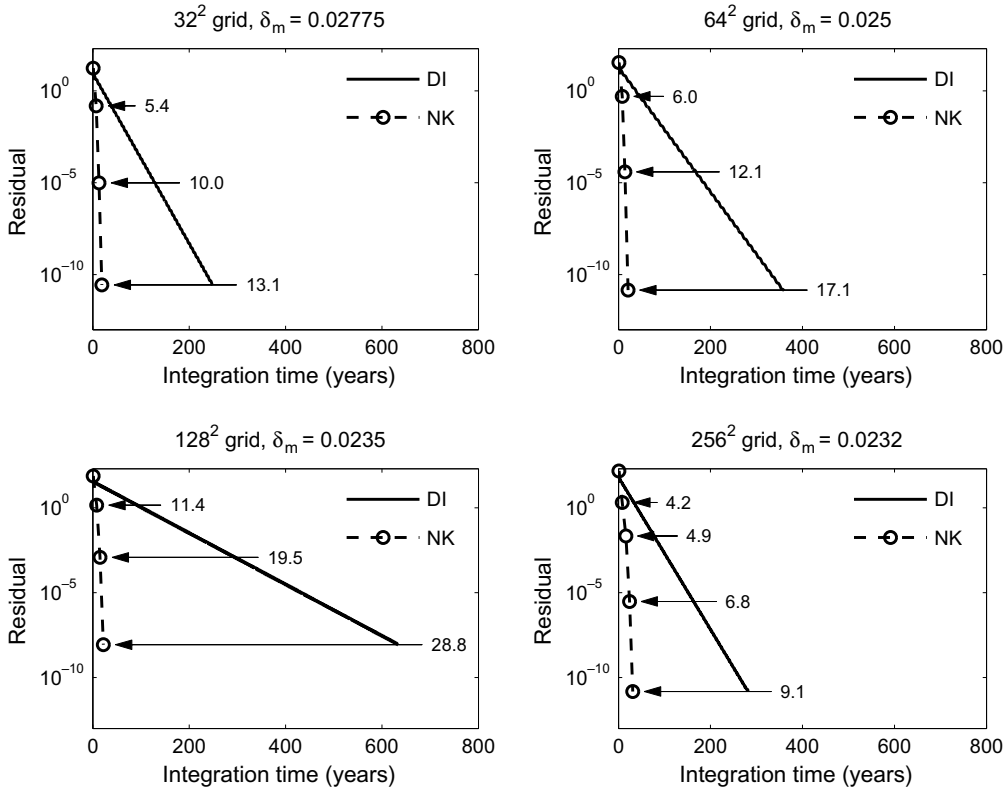

Fig. 5. Comparison of DI and MFNK for the seasonally-forced problem. Circles mark each Newton iteration. Numbers next to arrows are the speed-up of the Newton method, computed as the ratio (DI/MFNK) of the number of time-steps to reach the same residual.

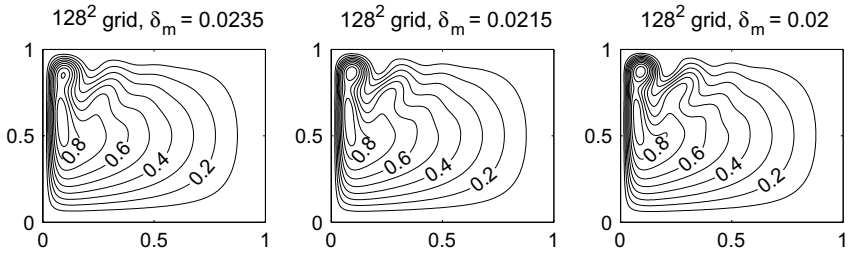

Fig. 6. Streamfunction for different viscosities computed by Newton–Krylov method all 128^2 grid.

much lower than those shown in this section because the Jacobian matrix becomes increasingly ill-conditioned and the Krylov solver converges impractically slowly. As a result, a preconditioning technique is needed in these cases. The novelty of the results presented in this section is that the MFNK algorithm can access regions of the parameter space which direct time integration of the same numerical model cannot, and that even though the MFNK method uses the time-dependent model, it can compute steady solutions that are unstable.

4. Discussion and conclusions

In this paper, we have introduced a novel approach to the spin-up of steady and periodically forced dynamical models defined via a time-stepper. We have shown that this approach, based on the matrix-free Newton–Krylov method, is quite effective at spinning-up a quasi-geostrophic ocean model to equilibrium. The equilibrium solutions obtained with this method agree, to within round-off error, with those obtained by direct time integration of the model. Additionally, in some cases, MFNK successfully computed equilibrium solutions that could not be accessed via direct integration. Our results also show that MFNK is significantly more efficient than direct integration. At the highest resolution we tested, speed-up ranged from $\approx$100 for the steady forcing problem to $\approx$10 for the periodic case. A key advantage of our approach is that it can be applied to any existing time-stepping code, such as ocean GCMs and biogeochemical models.

The algorithm as formulated here (Eq. (2)) applies for dynamical systems that are steady or cyclo-stationary in time. While the ocean and high resolution ocean models have eddying flows, the current generation of ocean climate

models are generally run at resolutions which do not resolve eddies, and as a result, this form of the algorithm applies without modification. The results in Section 3.2.3 show that the Newton method can compute unstable steady solutions, though these cases have relatively few unstable modes (Sheremet et al., 1997). This is promising for the prospect of using the MFNK algorithm for more turbulent flows, which we leave for future research. For longer time-scale variability such as the El Niño-Southern Oscillation or internal ocean variability, the MFNK algorithm is still an appealing alternative initialization procedure compared to a long-time integration, even if the stationary circulation itself is not the desired final calculation. Consequently, the MFNK algorithm can be useful for a range of different types of simulations.

While our initial results are quite encouraging, successful application of the MFNK method to more realistic problems such as global ocean GCMs is dependent on adequate preconditioning of the linear Newton equations (Eq. (3)). Preconditioning leads to efficient clustering of the eigenvalues of the iteration matrix, so as to reduce the number of iterations in the Krylov solver. In the QG problem, the MFNK solver converged rapidly without any preconditioning. However, it is unlikely that will be the case as we go to larger systems with many more prognostic variables and degrees of freedom. Traditionally, Newton–Krylov has been applied to solving nonlinear equations arising from the discretization of PDEs. In such problems, the Jacobian is generally quite sparse, and so called "physics-based" preconditioners can be brought to bear on the problem (e.g., Mousseau et al., 2000; Reisner et al., 2003; Knoll and Keyes, 2004). This approach exploits the nature of the PDE operator to explicitly compute an approximation to the Jacobian that can be stored and factored for easy application. However, this approach is simply not feasible for the dense Jacobian in the periodic spin-up problem. Furthermore, a physics-based preconditioner, if one could be developed, would be closely tied to the underlying ocean model, and would thus not be truly matrix-free or sufficiently general to be used with existing ocean GCMs.

Fortunately, there are potential alternatives that maintain much of the "black box" nature of MFNK. One approach that we are currently exploring is based on "polynomial preconditioning." In the present context, this involves constructing a convergent expansion for $\mathbf{J}^{-1} = (\mathbf{I} - \boldsymbol{\Phi}')^{-1}$ in terms of $\boldsymbol{\Phi}'$. To apply this preconditioner requires a fast, possibly approximate, method to repeatedly multiply $\boldsymbol{\Phi}'$ with a vector. This is relatively straightforward if the underlying model is implicit in time. However, even for models that are explicit in time, as most current GCMs are, it may be possible to build a linearized approximation to the model operator by using the "transport matrix" approach of Khatiwala et al. (2005) or automatic differentiation (AD) software. We have successfully applied this idea to a somewhat different problem, namely an ocean biogeochemical model, where the polynomial preconditioner significantly accelerated the Krylov solver (Khatiwala, to be submitted). The challenge now is to extend these ideas to a full ocean GCM.

Acknowledgements

We thank Alistair Adcroft for suggesting the QG ocean basin to us. The QG model was written by Karl Helfrich and modified by David Holland. We thank David Keyes for useful suggestions. The comments of two anonymous reviewers helped improve the presentation of the paper. Both the MFNK and QG program codes are available upon request. This work was funded by NSF Grants OCE-0449703 and OCE-0623366 and a NDSEG fellowship to TM. LDEO contribution number 7119.

Appendix. Algorithm details

The matrix-free Newton–Krylov solver that we use is that implemented in PETSc ("Portable, Extensible Toolkit for Scientific Computation") (Balay et al., 2003, 2001). PETSc is a freely available, open source suite of data structures and routines for solution of large scale linear and nonlinear problems. In our numerical experiments, the default solver parameters (residual tolerances, Krylov subspace method (GMRES), number of basis vectors retained, maximum number of linear iterations, etc.) are used. Since Newton's method is a locally convergent method, that is, it is possible for it to fail if the initial iterate is not sufficiently close to the solution, we also use a line search scheme to "globalize" it. (Another effective approach would be to carry out some preliminary time integration to bring the initial iterate closer to the solution.)

One area of potential improvement to applying the method to ocean GCMs is to optimize these parameters, though it is likely that the optimal set is problem dependent, so it is not the focus for this initial demonstrative example.

References

Aprille, T., Trick, T., 1972. A computer algorithm to determine the steady-state response of nonlinear oscillators. IEEE Trans. Circuit. Syst. 19, 354–360.

Arakawa, A., 1966. Computational design for long-term numerical integration of the equations of fluid motion: Two dimensional incompressible flow. J. Comput. Phys 1, 119–143.

Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F., 2001. PETSc home page, <http://www.mcs.anl.gov/petsc>.

Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F., 2003. PETSc users manual. Tech. Rep. ANL-95/11 – Revision 2.1.5, Argonne National Laboratory.

Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V., 1994. Templates for the solution of linear systems: Building blocks for iterative methods, second ed. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Brown, P.N., 1987. A local convergence theory for combined inexact-Newton/finite-difference projection methods. SIAM J. Numer. Anal. 24, 407–434.

Brown, P.N., Saad, Y., 1990. Hybrid Krylov methods for nonlinear systems of equations. SIAM J. Sci. Statist. Comput. 11, 450–481.

Bryan, K., 1984. Accelerating the convergence to equilibrium of ocean-climate models. J. Phys. Oceanogr. 14, 666–673.

Bryan, K., Lewis, L.J., 1979. A water mass model of the world ocean. J. Geophys. Res. 84, 2503–2517.

Cessi, P., Ierley, G., 1995. Symmetry-breaking multiple equilibria in quasigeostrophic, wind-driven flows. J. Phys. Oceanogr. 25 (6), 1196–1205.

Chan, T.F., Jackson, K.R., 1984. Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms. SIAM J. Sci. Statist. Comput. 5, 533–542.

Danabasoglu, G., McWilliams, J.C., Gent, P.R., 1994. The role of mesoscale tracer transports in the global ocean circulation. Science 264, 1123–1126.

Danabasoglu, G., 2004. A comparison of global ocean general circulation model solutions obtained with synchronous and accelerated integration methods. Ocean Modell. 7, 323–341.

Dembo, R.S., Eisenstat, S.C., Steihaug, T., 1982. Inexact Newton methods. SIAM J. Numer. Anal. 19, 400–408.

Dembo, R.S., Steihaug, T., 1983. Truncated-Newton algorithms for large-scale unconstrained optimization. Math. Prog. 26, 190–212.

Dennis, J.E., Schnabel, R.B., 1996. Numerical methods for unconstrained optimization and nonlinear equations. In: Classics in Applied Mathematics, vol. 16. Society for Industrial and Applied Mathematics.

Deuflhard, P., 2004. Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms. In: Springer Series in Computational Mathematics, vol. 35. Springer Verlag.

Dijkstra, H.A., Oksuzoglu, H., Wubs, F.W., Botta, E.F.F., 2001. A fully implicit model of the three-dimensional thermohaline ocean circulation. J. Comput. Phys. 173, 685–715.

Huang, R., Pedlosky, J., 2003. On aliasing Rossby waves induced by asynchronous time stepping. Ocean Modell. 5, 65–76.

Ierley, G., 1987. On the onset of inertial recirculation in barotropic general circulation models. J. Phys. Oceanogr. 17 (12), 2366–2374.

Kamenkovich, V., Sheremet, V., Pastushkov, A., Belotserkovsky, S., 1995. Analysis of the barotropic model of the subtropical gyre in the ocean for finite Reynolds numbers. Part I. J. Mar. Res. 53 (6), 959–994.

Kelley, C.T., 1995. Iterative Methods for Linear and Nonlinear Equations. In: Frontiers in Applied Mathematics, vol. 16. Society for Industrial and Applied Mathematics.

Kelley, C., Kevrekidis, I., Qiao, L., 2004. Newton–Krylov solvers for time-steppers. Available from: Arxiv:math.DS/0404374.

Khatiwala, S., Visbeck, M., Cane, M., 2005. Accelerated simulation of passive tracers in ocean circulation models. Ocean Modell. 9, 51–69.

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

Mousseau, V.A., Knoll, D.A., Rider, W.J., 2000. Physics-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion. J. Comput. Phys. 160, 743–765.

Pedlosky, J., 1996. Ocean Circulation Theory. Springer Verlag.

Reisner, J., Wyszogrodzki, A.A., Mousseau, V.A., Knoll, D.A., 2003. An efficient physics-based preconditioner for the fully implicit solution of small-scale thermally driven atmospheric flows. J. Comput. Phys. 189, 30–44.

Saad, Y., 2003. Iterative Methods for Sparse Linear Systems. In: Classics in Applied Mathematics, vol. 16. Society for Industrial and Applied Mathematics.

Sheremet, V., Ierley, G., Kamenkovich, V., 1997. Eigenanalysis of the two-dimensional wind-driven ocean circulation problem. J. Mar. Res. 55 (1), 57–92.

Skelboe, S., 1980. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. IEEE Trans. Circuit. Syst. 27, 161–175.

Telichevesky, R., Kundert, K.S., White, J.K., 1995. Efficient steady-state analysis based on matrix-free Krylov-subspace methods. In: 32nd ACM/IEEE Design Automation Conference.

Wang, D., 2001. A note on using the accelerated convergence method in climate models. Tellus 53A, 27–34.

Weijer, W., Dijkstra, H.A., Oksuzoglu, H., Wubs, F.W., de Niet, A.C., 2003. A fully-implicit model of the global ocean circulation. J. Comput. Phys. 192, 452–470.