

Fast spin up of Ocean biogeochemical models using matrix-free Newton–Krylov

Samar Khatiwala*

Lamont–Doherty Earth Observatory, Columbia University, Palisades, NY 10964, USA

ARTICLE INFO

Article history:

Received 3 January 2008
 Received in revised form 3 March 2008
 Accepted 5 May 2008
 Available online 15 May 2008

ABSTRACT

A novel computational approach is introduced for the efficient computation of equilibrium solutions of seasonally forced ocean biogeochemical models. The essential idea 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 (MFNK). MFNK is a combination of Newton-type methods for superlinearly convergent solution of nonlinear equations, and Krylov subspace methods for solving the Newton correction equations. 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. To render this approach practical for global models with $O(10^6)$ degrees of freedom, a flexible preconditioning strategy is developed. The result is an essentially “black-box” numerical scheme than can be applied to most existing biogeochemical models. The method is illustrated by applying it to find the equilibrium solutions of two realistic biogeochemical problems. Compared with the conventional approach of direct time integration, the preconditioned-MFNK scheme is shown to be roughly two orders of magnitude more efficient. Several potential refinements of the basic algorithm that may yield further performance gains are discussed. The numerical scheme described here addresses a fundamental challenge to using ocean biogeochemical models more effectively.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Ocean biogeochemical models are an important tool in efforts to understand the cycling of chemical and biological tracers such as carbon and nutrients. Given the vital role of the ocean carbon cycle in the climate system, such models are also a key component of coupled climate and earth system models. One of the principal challenges to using ocean biogeochemical models more effectively and systematically is their computational expense, as many biogeochemical tracers require several thousand years to reach equilibrium (Fig. 1; see also Wunsch and Heimbach, 2008). Equilibrium solutions are needed to develop, evaluate, and optimize new biogeochemical parameterizations, or to explore the sensitivity of simulated tracer fields to the underlying physical circulation model. The conventional practice of integrating the model until the transients have died out is, thus, prohibitively expensive to use on a routine basis, especially for the current generation of ocean climate models (typically 1° horizontal resolution).

Here, I present a novel approach to the biogeochemical “spin up” problem that builds on the recent proposal by Merlis and Khatiwala (2008) (MK07, hereafter) to apply Newton’s method to compute periodic solutions of ocean general circulation models

(GCMs). The basic idea is to seek solutions that repeat after one period. This transforms the time-dependent spin up problem into a nonlinear algebraic equation implicitly defined via the time-stepping model. To solve this equation, MK07 propose the use of an inexact Newton method, specifically, “matrix-free Newton–Krylov” (MFNK). As noted by MK07, the key advantage of this approach is that it can be applied to any existing time-stepping code, including biogeochemical models. However, MK07 also note that a naive application of this scheme to realistic problems is likely to fail because of the slow convergence of the linear iterative solver used to solve the Newton equation. As with most iterative methods, adequate preconditioning is absolutely essential for good performance, but, as explained below, is especially challenging for the spin up problem. To address this, I have developed a general preconditioning strategy that can be applied in a “black-box” fashion to efficiently finding the periodic equilibrium solutions of many, perhaps most, existing ocean biogeochemical model.

I should mention the work of Li and Primeau (2008), who have independently applied MFNK to the spin up of biogeochemical models. Li and Primeau (2008) also recognize the importance of preconditioning. However, as discussed in more detail below, their preconditioning strategy is quite different from that taken here. In particular, I believe that the scheme introduced here is considerably more general, and can be applied to a wide class of numerical models, including ocean GCMs.

* Tel.: +1 845 365 8454; fax: +1 845 365 8736.
 E-mail address: spk@ldeo.columbia.edu.

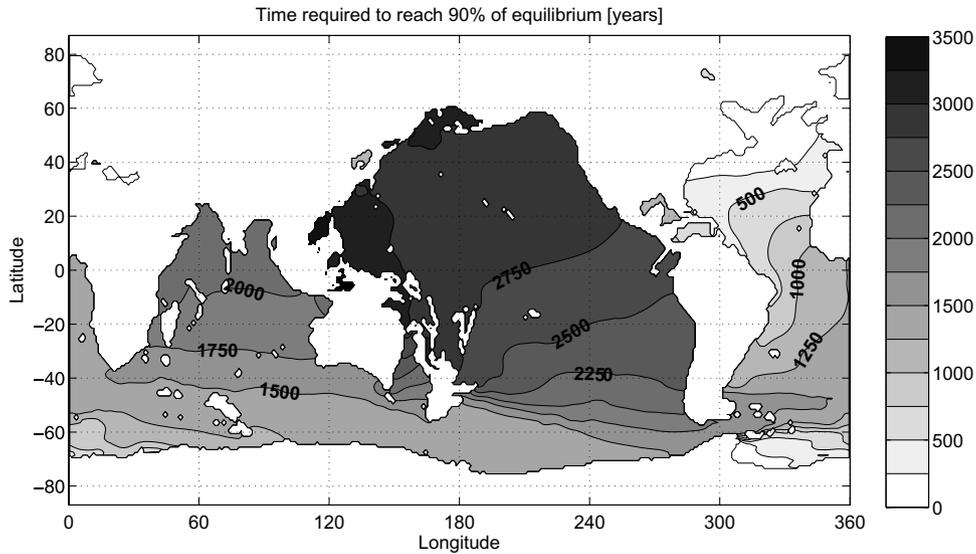


Fig. 1. Figure illustrating the long time scales required for tracers to reach equilibrium in the ocean. Shown is the time required to reach 90% of equilibrium at a depth of 1975 m for a passive tracer with a step boundary condition at the surface. The equilibrium solution is “1” everywhere. The simulation was performed with a 1° , data-assimilated configuration of the MIT GCM using the transport matrix method (Khatiwala et al., 2005).

In the next section, I describe the essential idea of matrix-free Newton–Krylov, followed by a description of the preconditioner. Finally, I illustrate the new scheme by applying it to two typical biogeochemical problems, and present numerical results demonstrating the superior performance of this method compared with direct time integration.

2. Matrix-free Newton–Krylov

To compute the equilibrium solutions of ocean GCMs, the so called dynamical spin up problem, MK07 proposed to apply a scheme known as matrix-free Newton–Krylov. The general idea is to find an initial condition such that the solution at the end of the period exactly matches the initial condition. I present a brief outline of MFNK, modified appropriately for the present biogeochemical context.

Let $\mathbf{u}(t)$ represents the state of the biogeochemical model, i.e., a vector of all the prognostic tracer concentrations, at time t . Given $\mathbf{u}(0)$, the model solves an initial value problem to predict the state of the system at any future time $t \geq 0$. In “state-space” notation, this may be written as

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

where Φ is the state-transition function of the system of differential equations governing the model. For a typical biogeochemical model, the dimension of the state-space, N , is $O(10^6)$. 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., $\mathbf{u}(T) = \mathbf{u}(0)$. This may be written as

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

thus transforming the problem into a nonlinear algebraic system. Note that since the periodically forced system is nonautonomous, fixing the definition of $t = 0$ (e.g., January 1) renders the solution unique. In the following, we drop the second (time) argument in $\Phi(\mathbf{u}, t)$. Unless stated otherwise, $t = T$ is implied, i.e., $\Phi(\mathbf{u})$ is the solution of the model after 1 period starting with an initial condition of \mathbf{u} . As MK07 point out, the above idea is not new. In fact, it is quite well known in the solution of ordinary differential equations (ODEs) (Deuflhard, 2004), in particular, those arising from

nonlinear analog circuits (e.g., Aprile and Trick, 1972; Skelboe, 1980; Telichevsky et al., 1995). However, the systems being considered in these works have generally been quite small (several hundred degrees of freedom, at most).

To solve the nonlinear system $\mathbf{F}(\mathbf{u}) = \mathbf{0}$, MK07 proposed to apply a class of methods known as matrix-free Newton–Krylov (MFNK) or Jacobian-free Newton–Krylov (JFNK). The MFNK method is an iterative method consisting of two primary components: an outer loop over the Newton correction and an inner loop building up the Krylov subspace out of which each Newton correction is drawn. The origins of this method can be traced back to publications motivated by the solution of ODEs (Gear and Saad, 1983; Brown and Hindmarsh, 1986) and PDEs (Brown and Saad, 1990; Chan and Jackson, 1984). (See Knoll and Keyes (2004) for a recent review.) The primary motivation in all cases appears to be the ability to perform a Newton iteration without forming the Jacobian.

To understand MFNK, recall, that the basic idea behind Newton’s method is to construct a local linear model of the function \mathbf{F} and then iterate (Kelley, 1995; Dennis and Schnabel, 1996). Algorithmically, given an initial guess, \mathbf{u}_0 , repeat for $k = 0, 1, \dots$ until convergence:

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

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

where \mathbf{J} is the Jacobian matrix, $\partial \mathbf{F} / \partial \mathbf{u}$. The iteration is terminated when the norm of the nonlinear residual, \mathbf{F} , falls below a specified value. A principal attraction of this method is that it is locally q -quadratically convergent, that is, the error in the solution is roughly squared with each iteration (Kelley, 1995). Newton’s method can also be “globalized” by combining it with various line-search and trust region techniques, as is done in most modern implementations.

While conceptually simple, Newton’s method can become prohibitively expensive for large problems. This is because much of the computational effort is in solving the linear system Eq. (3). In the periodic spin up problem, 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). Thus, explicit storage and factorization of \mathbf{J} , as would be required for a direct solution

method is quite impractical. It is then natural to consider iterative methods in which the linear system is solved only approximately. Specifically, Eq. (3) for the Newton step is replaced by the inexact Newton condition

$$\|\mathbf{J}(\mathbf{u}_k)\delta\mathbf{u}_k + \mathbf{F}(\mathbf{u}_k)\| \leq \eta_k \|\mathbf{F}(\mathbf{u}_k)\|. \quad (5)$$

Such a scheme is known as an “inexact Newton method” (Dembo et al., 1982) or (in the optimization literature) as a “truncated Newton method” (Dembo and Steihaug, 1983). The parameter η_k is known as the forcing term and is a measure of the trade-off between the effort required to solve the linear system to a tight tolerance and the number of nonlinear iterations. In practice, η is varied as the Newton iteration progresses (Eisenstat and Walker, 1996).

Among the most popular and successful iterative methods for solving large sparse linear systems are methods based on Krylov subspace projection (Barrett et al., 1994; Saad, 2003). Krylov methods such as GMRES have the property that the coefficient matrix is only accessed via matrix–vector products. This is the key to their use with Newton’s method, since the required Jacobian–vector product, $\mathbf{J}\delta\mathbf{u}$, can be accurately approximated by a difference quotient of the form (Chan and Jackson, 1984; Brown and Saad, 1990)

$$\mathbf{J}(\mathbf{u})\delta\mathbf{u} \approx \frac{\mathbf{F}(\mathbf{u} + \sigma\delta\mathbf{u}) - \mathbf{F}(\mathbf{u})}{\sigma}. \quad (6)$$

The differencing parameter σ is typically chosen dynamically (Brown, 1987; Brown and Saad, 1990). Since Eq. (6) only requires function evaluations, explicit computation and storage of the Jacobian matrix can be avoided. Note that, in the present context, each “function evaluation” (Eq. (2)) requires a forward integration of the biogeochemical model with a given initial condition for time T . The overall nonlinear solver is termed matrix-free Newton–Krylov or Jacobian-free Newton–Krylov. (Methods in which the coefficient matrix is not formed or stored explicitly, are known as “matrix-free”.)

3. Preconditioning of the Krylov solver

While, in principle, use of MFNK requires nothing more than the ability to compute the residual \mathbf{F} , a naive application of this method is likely to fail. The reason for this is the slow rate of convergence of the Krylov method used to solve the linear Newton equation. The convergence of iterative schemes depends critically on adequate preconditioning of the iteration matrix (e.g., Saad, 2003). Preconditioning leads to efficient clustering of the eigenvalues of the iteration matrix, so as to reduce the number of iterations of the Krylov solver. Preconditioning the iteration matrix \mathbf{J}

means multiplying \mathbf{J} from the right, left, or both sides by a preconditioner \mathbf{M} . The basic idea is that systems with the coefficient matrix $\mathbf{M}\mathbf{J}$ or $\mathbf{J}\mathbf{M}$ may be easier to solve than those with \mathbf{J} . For the spin up problem, each iteration requires a function evaluation, i.e., integrating the model for one period, something we wish to minimize. Thus, for MFNK to be practical, an appropriate preconditioner for the linear Krylov solver is absolutely essential.

Traditionally, Newton–Krylov has been applied to solving nonlinear equations arising from the discretization of PDEs. In such problems, the Jacobian is generally very 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; Bernsen et al., 2008). 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, the *periodically* forced spin up problem leads to a dense Jacobian that simply cannot be computed or stored explicitly, making this approach unfeasible for the problem we wish to address. Furthermore, it is desirable to have an approach that can be applied to any biogeochemical problem, and not one tied to a specific model. Here, I describe a flexible preconditioning strategy that addresses this problem.

Before describing the preconditioner, it is useful to examine the eigenvalue spectrum of the Jacobian in the spin up problem. As an illustrative example, I have computed the spectrum for a simple carbon system model based on the OCMIP-2 protocol (Orr et al., 1999) (described more fully later). The ocean model is a 2.8° global configuration of the MIT model with 15 vertical levels, and forced with monthly mean climatological fluxes of momentum, heat, and freshwater. In addition, surface temperature and salinity are weakly restored to the *Levitus* climatology (Levitus et al., 1998). To compute the full dense Jacobian for this problem would require as many function evaluations (years of integration) as the number of grid points, which at this resolution is roughly 53,000. The resulting Jacobian would occupy ≈ 22 GB of memory. Since this is clearly not feasible, I used the “transport matrix method” of Khatiwala et al. (2005) to perform simulations at a much lower horizontal resolution (16°) (see Section 3.2). This method allows one to generate a coarse grained circulation field at any desired resolution. Using finite-differences, I have computed the Jacobian of \mathbf{F} for this coarser problem which has a much more tractable dimension of ≈ 2000 .

Fig. 2 shows the spectrum of this Jacobian. As is evident, there is a large cluster of eigenvalues near zero. These are responsible for the slow convergence of the Krylov method. The goal of preconditioning is to move the eigenvalues toward the unit circle, thus accelerating convergence.

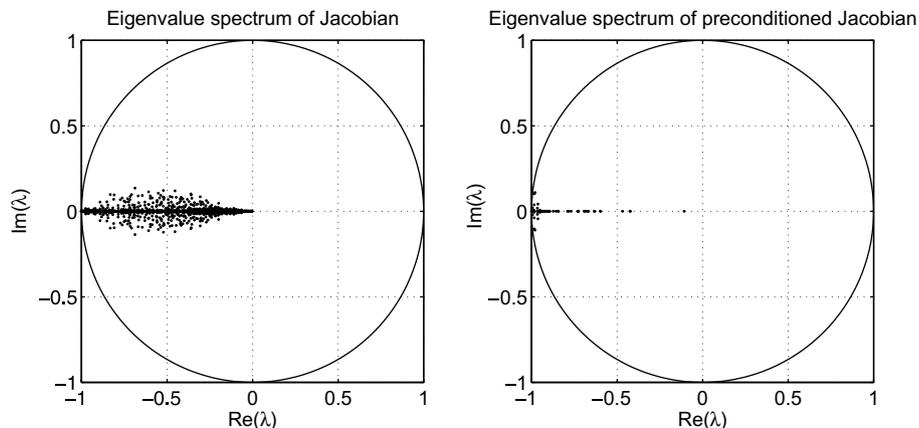


Fig. 2. Left: Eigenvalue spectrum of the Jacobian matrix of the residual function for the coarse-resolution OCMIP-2 abiotic carbon model. Right: Spectrum of the Jacobian preconditioned using the implicit preconditioner discussed in the text.

I should also mention the related work by Li and Primeau (2008) (LP08, hereafter) on applying MFNK to biogeochemical models. There are a number of differences between the present paper and their study, the principal one being the preconditioner. As described below, the approach taken here is quite general. It makes no assumptions about the underlying model, which could be a biogeochemical model, a GCM, or any other dynamical system. It is also essentially a black-box scheme that can be applied to existing codes. In contrast, LP08 develop a “physics-based” preconditioner that is constructed “by hand”, and is directly based on, and tied to, a particular biogeochemical model. A second difference is that the model they apply their scheme to is a relatively simple one with two tracers, inorganic and dissolved organic phosphate (PO_4 and DOP, respectively). The only nonlinearity in the problem comes from a function that switches off surface biological production when PO_4 falls below a prescribed value. In contrast, I have evaluated the present approach on two different models (see Section 4), both of which include nonlinear terms such as equilibrium carbonate chemistry, and one of which has several interacting tracers. For their biogeochemical model, LP08 obtain a speed-up of over two orders of magnitude over direct time integration. However, it is important to note that for this particular problem, MFNK converged 10 times faster than direct integration even *without* a preconditioner. Thus, their preconditioner yields an effective speed-up factor of “only” 10. In contrast, neither of the two biogeochemical models I evaluated converge at all without preconditioning. In light of the large cluster of eigenvalues near zero seen in Fig. 2, this is not surprising, and would suggest that the results obtained by LP08 may not be representative of more complex biogeochemical models. Lastly, LP08 simply “write down” their preconditioning operator without providing any details of how it was obtained. Nor do they provide any justification for why their method should work, or how well. Thus, it is not clear how their scheme can be generalized to other biogeochemical models.

3.1. An implicit preconditioner

An ideal preconditioner for a linear system should be a close approximation to the coefficient matrix, but one that is much easier to “invert”. The fact that the Jacobian is dense, severely limits our choice to matrix-free methods in which the preconditioner is applied implicitly, i.e., by solving a linear system. That is the strategy taken here. Specifically, consider writing the model as:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, t). \quad (7)$$

This equation can be envisaged as arising from a “method-of-lines” discretization of the underlying PDEs in which the spatial operator is first discretized, resulting in a system of ODEs. This approach is common to many ocean models. The function \mathbf{f} represents advective–diffusive transport of the tracer by the ocean circulation, as well as biogeochemical sources and sinks. Consider time-stepping this equation implicitly with Euler backward differencing, with a time step $\Delta t = T$. Updating from $t = 0$ to $t = T$ we have,

$$\mathbf{u}(T) - \mathbf{u}(0) = \Delta t \mathbf{f}(\mathbf{u}(T)). \quad (8)$$

But note that, by definition, the LHS of this equation is simply an approximation of $\mathbf{F}(\mathbf{u}(0))$ (Eq. (2)). Differentiating w.r.t. $\mathbf{u}(0)$ we have,

$$\mathbf{F}'(\mathbf{u}(0)) \approx \hat{\mathbf{J}} \equiv \Delta t \mathbf{f}'(\mathbf{u}(T)) \frac{\partial \mathbf{u}(T)}{\partial \mathbf{u}(0)}. \quad (9)$$

(A prime denotes the derivative of a function w.r.t. the argument.) From Eq. (8) we also have:

$$\frac{\partial \mathbf{u}(T)}{\partial \mathbf{u}(0)} = [\mathbf{I} - \Delta t \mathbf{f}'(\mathbf{u}(T))]^{-1}.$$

Substituting the above equation into Eq. (9), approximating $\mathbf{u}(T)$ by $\Phi(\mathbf{u}, T)$, and taking the inverse we finally have:

$$\mathbf{F}'(\mathbf{u})^{-1} \approx \hat{\mathbf{J}}^{-1} = \mathbf{P}\mathbf{Q}^{-1}, \quad (10)$$

where,

$$\mathbf{P} = [\mathbf{I} - \Delta t \mathbf{f}'(\Phi(\mathbf{u}, T))] \text{ and } \mathbf{Q} = [\Delta t \mathbf{f}'(\Phi(\mathbf{u}, T))]. \quad (11)$$

Eq. (10) is an expression for the *approximate inverse* of the Jacobian of \mathbf{F} . To apply it as a preconditioner, i.e., to compute $\mathbf{w} = \hat{\mathbf{J}}^{-1}\mathbf{v}$ for a given \mathbf{v} , requires the following steps:

- (1) Solve $\mathbf{Q}\mathbf{z} = \mathbf{v}$ (equivalent to $\mathbf{z} = \mathbf{Q}^{-1}\mathbf{v}$)
- (2) Compute $\mathbf{w} = \mathbf{P}\mathbf{z}$

I emphasize the following aspects of the above procedure:

- To assemble \mathbf{P} and \mathbf{Q} , the full, *sparse* matrix \mathbf{f}' is needed. This can be done very efficiently by using the sparsity pattern of \mathbf{f}' (known from the stencil of the spatial discretization of the model) to partition \mathbf{f}' into “structurally orthogonal” columns identified using graph coloring methods (Curtis et al., 1974; Coleman and Moré, 1983; Gebremedhin et al., 2005). These columns may then be computed simultaneously, rather than one at a time, either via finite-differences or differentiation of the model code using automatic differentiation software (AD) (Bischof et al., 1997; Giering and Kaminski, 1998). Thus, if \mathbf{f}' has n groups of structurally orthogonal columns, where $n \ll N$, then, using finite-differences, \mathbf{f}' can be computed with $n + 1$ evaluations of \mathbf{f} . As a practical matter, n is *independent* of horizontal resolution (e.g., Khatiwala et al., 2005), but proportional to the number of vertical levels and the number of tracers (see below).
- Since \mathbf{Q} is sparse and explicitly available, linear systems involving this matrix can be solved very efficiently. Specifically, we can either use a sparse direct solver such as SuperLU (Demmel et al., 1999; Li and Demmel, 2003) or MUMPS (Amestoy et al., 2000), or a Krylov solver such as GMRES preconditioned by an incomplete LU factorization of \mathbf{Q} . Both approaches scale well to large problems, and much of the overhead of setting up the solvers can be amortized. Efficiency can be further improved by reordering the state vector \mathbf{u} so as to aggregate the different variables together. This has the effect of reducing the bandwidth of \mathbf{Q} and minimizing the fill-in of its sparse factors.
- The matrix \mathbf{f}' (and \mathbf{P} and \mathbf{Q}) only needs to be computed once per Newton iteration. Numerical experiments suggests that the number of Newton iterations in a typical spin up calculation to be ≈ 10 – 20 .
- The spatial discretization used in Eq. (7) to compute the preconditioner need *not* be the same as that used in the “true” model whose equilibrium solutions we are computing. For example, the ocean model used to transport tracers might use a 3d order flux-limited advection scheme for tracers, while for the preconditioner we might use a 2d order centered scheme, which has a narrower stencil. Other simplifications are possible, but will likely slightly degrade the performance of the preconditioner.
- For models lacking an explicit method-of-lines discretization, \mathbf{f} can be computed using finite-differences (essentially taking an infinitesimal time step of the model).
- Since the method works by “probing” the function \mathbf{f} , it is relatively straightforward to apply this scheme to an existing model code.
- The implicit preconditioner can be used for both periodic and time-independent spin up problems.

To summarize, the cost of computing the implicit preconditioner is no more, and generally much smaller, than the cost of performing function evaluations (integrating the model for 1 year).

The cost of applying the preconditioner is negligible, especially if using a sparse direct solver.

To illustrate the impact of this preconditioner on the spectrum of the Jacobian, the right panel in Fig. 2 shows the spectrum of the preconditioned Jacobian of the coarse resolution biogeochemical model described above. As is evident, the preconditioner significantly improves the spectrum of the iteration matrix by moving the small eigenvalues toward the unit circle.

3.2. Efficient computation of \mathbf{f}'

While the matrix \mathbf{f}' can be constructed by directly probing the coupled circulation-biogeochemistry model, its computation can be considerably simplified by exploiting the transport matrix method (TMM). The TMM is a scheme for efficient “offline” simulation of passive tracers in ocean models. The essential idea of the TMM is that the discrete tracer transport operator of an ocean general circulation model can be written as a sparse matrix, which may be accurately and efficiently constructed by “probing” the GCM with a passive tracer. In the transport matrix formulation, \mathbf{f} can be expressed as

$$\mathbf{f}(\mathbf{u}, t) = \text{diag}(\mathbf{A}(t))\mathbf{u} + \mathbf{q}(\mathbf{u}, t). \quad (12)$$

Here, \mathbf{A} is the transport matrix (“TM”) which results from discretization of the advection–diffusion operator and includes the effects of advection, diffusion, and various (parameterized) sub-grid scale processes. ($\text{diag}(\mathbf{A})$ is a block diagonal matrix with \mathbf{A} on the diagonal. The number of blocks is equal to the number of tracers.) \mathbf{q} represents biogeochemical sources and sinks. In the following, we simplify the above equation by replacing $\mathbf{A}(t)$ with an annual mean TM. Using the fact that \mathbf{A} is extremely sparse, Khatiwala et al. (2005) have developed an empirical procedure that allows us to efficiently construct \mathbf{A} by “probing” the GCM with a passive tracer. The algorithm can be applied to most ocean GCMs, requiring little or no modification to the underlying GCM code. It has been implemented in the MIT GCM and MOM4.

Using Eq. (12), \mathbf{f}' can be written as:

$$\mathbf{f}'(\mathbf{u}) = \text{diag}(\mathbf{A}) + \mathbf{q}'(\mathbf{u}), \quad (13)$$

where $\mathbf{q}' = \partial\mathbf{q}/\partial\mathbf{u}$. In this equation, the term involving \mathbf{A} is sparse because of the inherent locality of the advection–diffusion operator. The term involving the Jacobian of the biogeochemical source/sink term is also sparse. The reason for this is that in most biogeochemical models, the source/sink term at a grid point depends only on tracer concentrations in the same vertical column. Consequently, \mathbf{q}' has a *known* sparse block structure, with different blocks corresponding to different tracers. Furthermore, each block itself has a sparse block structure, with the sub-blocks corresponding to differ-

ent vertical profiles. This sparsity pattern is schematically illustrated in Fig. 3. Using graph coloring methods, \mathbf{q}' can be computed in at most (number of vertical grid points) \times (number of tracers) evaluations of \mathbf{q} . Thus, with the TMM, \mathbf{f}' , and the matrices \mathbf{P} and \mathbf{Q} required for preconditioning, can be computed at negligible expense.

Thus, once we have computed the TM of an ocean model (online or offline), it can be used (as a preconditioner) for the spin-up of any biogeochemical model coupled to that particular ocean model. If the TM is not already available, then it is advantageous to first compute it using the procedure developed by Khatiwala et al. (2005). The cost is roughly $m \times$ (number of vertical grid points) \times (number of tracers) evaluations of \mathbf{f} , with a single passive tracer and $\mathbf{q} = \mathbf{0}$ (no source/sink term). The proportionality constant m depends only on the geometry (coastline) and the stencil of the spatial discretization of the advection–diffusion operator (which need not be the same as that of the “true” model). Crucially, it is independent of the horizontal resolution. For a global domain and 2d order centered advection scheme, $m \approx 15$.

3.3. Singularity of the Jacobian matrix

Newton’s method breaks down if the Jacobian is singular. As mentioned above, this situation, which is due to tracer conservation, is in fact quite common in biogeochemical problems. To alleviate this problem, Li and Primeau (2008) suggest adding a very weak relaxation term to the biogeochemical source/sink term. This approach removes the singularity without appreciably changing the final equilibrium solution.

4. Numerical results

To illustrate the preconditioned-MFNK (PMFNK) method, I have applied it to two representative biogeochemical problems. In both cases, I computed periodic equilibrium solutions starting with a uniform initial condition using PMFNK and, for comparison, direct time integration (DI) of the biogeochemical model. To compare the computational efficiency of the two methods, I recorded the number of function evaluations, for PMFNK, and years of integration, for DI, to reach a given residual. The residual is defined as the vector 2-norm of \mathbf{F} and is a measure of model drift. Recall that each “function evaluation” in MFNK requires integrating the model for one period. Here, I have used two different configurations of the offline version of the MIT GCM to perform the simulations. The first is the 2.8° model described above. The second is a 1° resolution, global configuration with 23 vertical levels. The GCM is forced by a monthly climatology of heat, freshwater, and momentum fluxes derived from the ECCO-GODAE data assimilation project (Wunsch

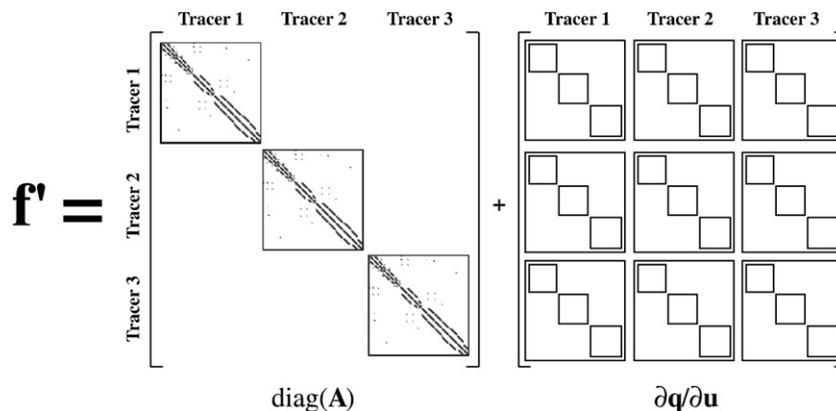


Fig. 3. Sparsity structure of \mathbf{f}' for a biogeochemical model with 3 tracers. Each sub-block in $\partial\mathbf{q}/\partial\mathbf{u}$ corresponds to a vertical profile.

and Heimbach, 2007). A time step of 1/2 day was used to perform tracer simulations. Transport matrices were already available for both configurations, and were used in constructing the preconditioner (Section 3.2).

A number of freely available implementations of inexact Newton methods exist. To obtain the results shown here, I used the KINSOL component of SUNDIALS (“SUite of Nonlinear and Differential/Algebraic equation Solvers”), a freely available, numerical software library from Lawrence Livermore National Laboratory (Hindmarsh et al., 2005). The advantage of using KINSOL, as opposed to some other implementations, is that it incorporates preconditioning and scaling of the variables in a self consistent manner. Scaling is of particular importance in solving systems of nonlinear equations, especially when the independent variables have widely disparate scales (Dennis and Schnabel, 1996), as is often the case in biogeochemical models with multiple tracers. Scaling is also important when evaluating the Jacobian-vector product using finite-differences (Eq. (6)). One remedy is to rescale the independent variables, typically by multiplying \mathbf{u} by a diagonal scaling matrix. The basic idea is to ensure that each component of \mathbf{u} has roughly the same magnitude. While the nonlinear residual \mathbf{F} can be similarly scaled, a practical difficulty is that this requires a scaling of the Newton equations. However, since we are using a matrix-free Krylov algorithm, the Jacobian matrix is not available, and hence an explicit scaling of it cannot be performed. KINSOL implements a scaled, right-preconditioned GMRES algorithm based on Brown and Hindmarsh (1986) that effectively performs a scaling of the Newton equations without ever scaling the Jacobian. KINSOL also incorporates a robust line-search globalization method.

To apply KINSOL to a biogeochemical problem, we specify routines to (1) compute the residual function \mathbf{F} , (2) setup the preconditioner, and (3) apply the preconditioner to a given vector. The setup routine is called at the start of each nonlinear iteration, and allows us to recompute the preconditioner. In addition, scale factors for both the independent and dependent variables can be optionally specified. As a practical matter, KINSOL can be conveniently accessed through a MATLAB interface distributed with the SUNDIALS package.

In applying KINSOL to the examples shown below, \mathbf{u} was scaled by the initial iterate (uniform tracer distribution). The residual vector was not scaled. Additionally, to apply the preconditioner, I use a parallel sparse direct solver (MUMPS). The primary overhead incurred here is that of factoring \mathbf{Q} at the start of each Newton iteration. Subsequent linear solves have negligible cost. Also, since the sparsity of \mathbf{Q} does not change over the course of the calculation, the expensive permutation and symbolic factorization steps under-

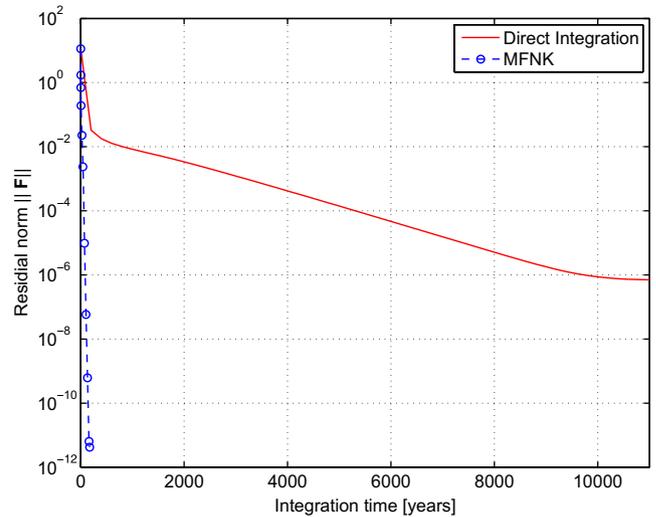


Fig. 4. Residual norm $\|\mathbf{F}\|$ as a function of integration time for DI and PMFNK for the OCMIP-2 spin up problem. The residual was scaled by the initial iterate. Symbols mark each Newton iteration.

taken by MUMPS only need to be performed once. In the first example, I estimate the cost of factoring \mathbf{Q} as only a few percent of the cost of a function evaluation. In the second (larger) example, this cost was roughly 50% of that of a function evaluation. In the following, the integration time for PMFNK has been adjusted to account for this overhead.

4.1. OCMIP-2 abiotic carbon model

As a first example, we consider a simple model of ocean carbon based on the OCMIP-2 protocols (Abiotic model) (Orr et al., 1999). This model simulates a single tracer, total dissolved inorganic carbon (DIC), and has a parameterization for air-sea exchange of CO_2 at the surface (the so-called solubility pump). No biological effects are included. However, the model is nonlinear because of the nonlinear equilibrium carbonate chemistry equations that are solved at each time step. The dimension of this problem is $N = 52749$.

As seen in Fig. 4, a direct forward integration of this model, starting with uniform DIC concentration, takes several thousand years to reach equilibrium. Specifically, to reach a residual norm of 5×10^{-5} (the norm of the difference in solutions one year apart), takes roughly 5800 years of integration. In contrast, to reach the same residual norm, PMFNK takes 42 function evaluations, equivalent to integrating the model for 42 years. Thus, PMFNK is about

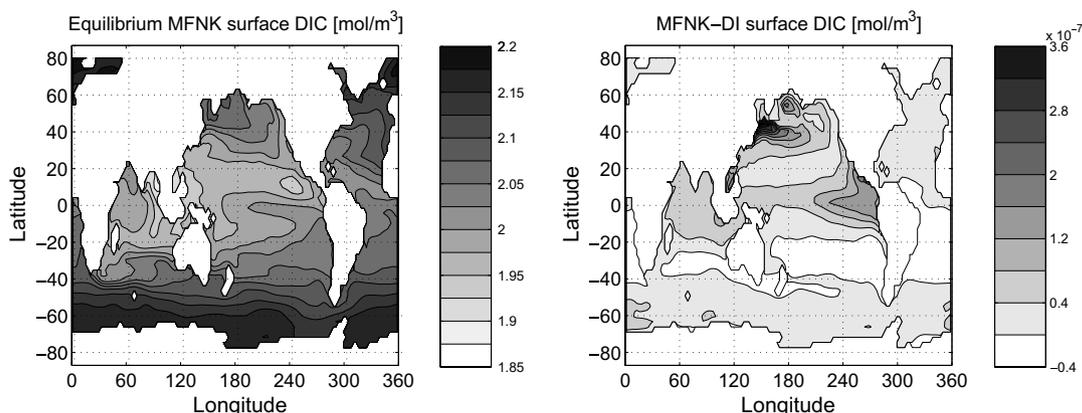


Fig. 5. Equilibrium surface DIC concentration on January 1. Left: solution computed using PMFNK. Right: difference between the PMFNK and DI equilibrium solutions.

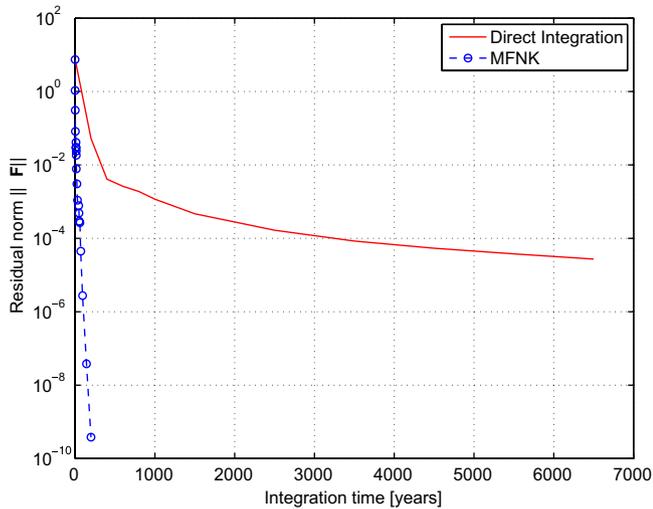


Fig. 6. Comparison of DI and PMFNK method for the coupled biogeochemical spin up problem. Shown is the residual norm as a function of integration time for DI and PMFNK. The residual was scaled by the initial iterate. Symbols mark each Newton iteration.

138 times faster than direct time integration. I should note that the final equilibrium solution obtained by PMFNK is essentially identical from the DI solution (Fig. 5).

It should also be noted that PMFNK can achieve a much lower drift than DI. In this experiment, for example, even after 11,000 years of integration, the time stepping model was not able to achieve a residual norm lower than 7×10^{-7} . (This is likely due to numerical round-off that accumulates over time.) With PMFNK, a norm of 10^{-12} was readily reached.

4.2. Coupled biogeochemical model

As a second, more complex, example, we consider a coupled biogeochemical model of carbon, phosphorus, and iron, based on that of Dutkiewicz et al. (2005). The model includes 6 prognostic tracers: inorganic and organic forms of phosphorus, carbon, oxygen, alkalinity, and iron. As documented in Dutkiewicz et al. (2005) and Dutkiewicz et al. (2006), the model includes a simplified parameterization for net community production, B , which is regulated by the availability of light (photosynthetically active radiation), phosphate, and iron. Scavenging and complexation of iron is represented according to Parekh et al. (2005). In addition, iron is supplied to the ocean via an aeolian dust source prescribed according to Mahowald et al. (2003). The dimension of the state space is $N \approx 4.1 \times 10^6$. This model conserves a number of quantities, giving rise to a null space. Following Li and Primeau (2008), I account for this by adding a weak restoring term, with a time scale of 10^4 years, to the tracer equations.

Fig. 6 compares the residual norm as a function of integration time (equivalent integration years for PMFNK). To reach a residual norm of 3×10^{-5} , the norm at the end of 6000 years of direct integration, requires the equivalent of 80 years of integration using MFNK, a speed-up factor of ~ 75 . Again, the PMFNK equilibrium solution is essentially identical to that obtained with DI (not shown).

5. Alternative preconditioning strategies

The implicit preconditioner presented here was shown to be quite effective on two representative biogeochemical problems, even at a relatively high horizontal resolution. It is, however, not the only approach one can take to preconditioning. Here, I briefly

describe a number of other potential schemes to accelerate convergence that I am currently exploring.

5.1. Polynomial preconditioning

The implicit preconditioner does not resolve the seasonal cycle. To derive one that is more accurate we turn to Eq. (1), which may be viewed as a fixed point iteration. If the map Φ is stable and convergent, the eigenvalues, λ , of $\Phi' \equiv \partial \mathbf{u}(T)/\partial \mathbf{u}(0)$ must lie on or within the unit circle. (The eigenvalues of Φ' and \mathbf{J} are shifted relatively to each other by 1.) If the λ 's are strictly within the unit circle, then $\mathbf{J} = \Phi' - \mathbf{I}$ is invertible, and the inverse may be approximated by the truncated Neumann expansion:

$$\mathbf{J}^{-1} = (\Phi' - \mathbf{I})^{-1} \approx - \sum_{k=0}^{K-1} (\Phi')^k. \quad (14)$$

(Alternative convergent expansions, using a matrix-valued z-transform, are readily found for cases when some eigenvalues lie outside the unit circle (Tylavsky and Sohie, 1986).) This assumes that \mathbf{J} is nonsingular, i.e., $\lambda = 1$ is not an eigenvalue, and therefore Eq. (14) converges to the inverse as $K \rightarrow \infty$. The essential idea is to apply Eq. (14) as a preconditioner, an approach known as “polynomial preconditioning” (e.g., Saad, 2003).

To apply \mathbf{J}^{-1} to a vector via Eq. (14) requires multiple Φ' -vector products. To efficiently evaluate this matrix–vector product, we once again use Euler backward differencing on Eq. (7), but this time break the time interval $[0, T]$ into M segments. Updating from time step m to $m + 1$ we have,

$$\mathbf{u}_{m+1} - \mathbf{u}_m = \Delta t \mathbf{f}(\mathbf{u}_{m+1}).$$

The time step $\Delta t \equiv T/M$ need not be the same as that used by the “true” model. Indeed, for preconditioning, we can use a Δt much larger than the model time step. Differentiating the above equation w.r.t. \mathbf{u}_m and using the chain rule we obtain,

$$\frac{\partial \mathbf{u}_{m+1}}{\partial \mathbf{u}_m} = [\mathbf{I} - \Delta t \mathbf{f}'(\mathbf{u}_{m+1})]^{-1}.$$

Defining $\mathbf{P}_i \equiv \mathbf{I} - \Delta t \mathbf{f}'(\mathbf{u}_i)$ and making repeated use of the chain rule finally gives

$$\Phi' = \mathbf{P}_M^{-1} \mathbf{P}_{M-1}^{-1} \cdots \mathbf{P}_1^{-1}. \quad (15)$$

In summary, to compute $\mathbf{w} = \mathbf{J}^{-1} \mathbf{v}$, we need to solve $M \times K$ linear systems, with the coefficient matrices \mathbf{P}_i . As discussed earlier, the \mathbf{P}_i are sparse and relatively inexpensive to compute.

For a given M , how rapidly Eq. (14) converges to the inverse of \mathbf{J} depends on the spectrum of Φ' . To gain insight into this, I have compared the performance of the polynomial preconditioner with $M = 1$ and increasing K , with the implicit one, for the OCMIP-2 abiotic problem (Fig. 7). As expected, the accuracy of the Neumann expansion, and hence the speed-up over DI, increases with K . In particular, with $K = 30$, the speed up factor is 92, with $K = 100$ it is 112, and with $K = 150$, it is 120. Not surprisingly, the implicit preconditioner, which corresponds to $K = \infty$, is the most efficient. These preliminary results are reassuring, but it remains to be seen whether the potentially increased accuracy of this scheme outweighs the extra overhead of solving the linear systems, particularly as M is increased. Future work will examine this issue.

5.2. Quasi-Newton preconditioner

Another possibility is to apply a quasi-Newton method to improve the implicit preconditioner. Quasi-Newton schemes, such as Broyden's method, have conventionally been used as a linear solver as part of an inexact Newton strategy (e.g., Dennis and Schnabel, 1996). However, the approach envisaged here is quite

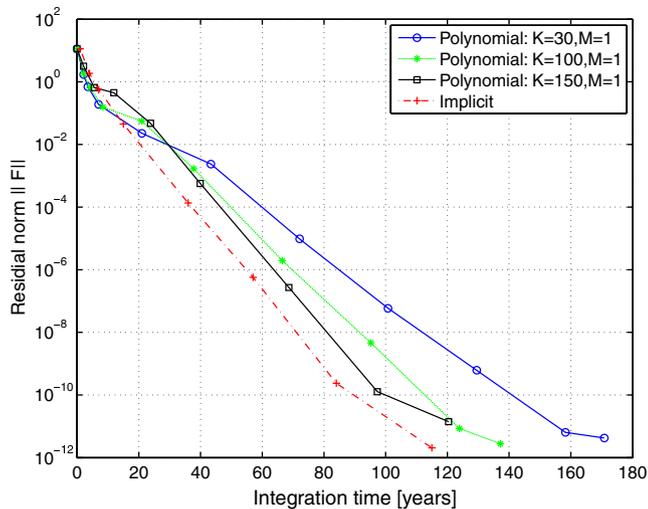


Fig. 7. Comparison of the implicit and polynomial preconditioners for $M = 1$, and different values of K .

different. The idea is to start with an initial preconditioner \mathbf{J}_0^{-1} , for example Eq. (10), and improve on it at each nonlinear iteration via a quasi-Newton, low-rank update based on the history of the nonlinear iterations. The overhead of this scheme is quite small: storage of 2 vectors and one extra application of \mathbf{J}_0^{-1} per Newton iteration.

5.3. Recursive projection method

A third direction being investigated is the application of the recursive projection method (RPM) of Shroff and Keller (1993) to the spin-up problem. RPM is a scheme to accelerate the convergence of (possibly unstable) fixed-point iterations such as $\mathbf{u}^{k+1} = \Phi(\mathbf{u}^k)$, by adaptively constructing a low-dimensional invariant subspace corresponding to the large eigenvalues of Φ' . In this subspace, a Newton iteration is performed, while the fixed-point iteration is used on its complement. RPM is particularly effective when slow convergence is due to a few slowly decaying modes. It is an entirely “black-box” procedure, requiring only the function Φ . However, initial experiments with using RPM on the biogeochemical problems discussed here, show only slight performance gains (roughly a factor of 4) over direct integration. Future work will therefore involve combining RPM with a preconditioner (e.g., Davidson, 1997).

6. Discussion and conclusions

In this paper, I have introduced a new approach to the spin up of periodically forced global ocean biogeochemical models. This approach, based on the matrix-free Newton–Krylov method, was shown to be significantly more efficient at computing periodic equilibrium solutions than the conventional brute force scheme of direct time integration. Application of MFNK to two realistic biogeochemical problems shows a speed up of roughly 2 orders of magnitude over direct integration. Much of the efficiency of this scheme is due to a novel preconditioner, which can be applied in an almost “black-box” manner to accelerate the spin up of a wide class of numerical codes, including ocean biogeochemical and general circulation models.

The focus of this paper has been on seasonally forced models. However, the basic approach can be applied to models with any forcing period, although it is not clear if the computational gains will extend to, say, ecosystem models with fast time scales or to

models with diurnal forcing. A different approach may be required to spinning up such systems. For instance, in such models the surface biology often equilibrates rapidly, while the deep ocean nutrient field responds on much longer time scales. A possible solution is an iterative scheme that alternates between (1) running the ecosystem component for a few years with a given nutrient field until it equilibrates, and (2) applying MFNK to the nutrients with a prescribed ecology from archived model output.

Acknowledgement

I would like to thank Tim Merlis, Andreas Oschlies, and two anonymous referees for their comments and suggestions to improve the text. This work was funded by NSF grant OCE-0623366. LDEO contribution number 7164.

References

- Amestoy, P.R., Duff, I.S., LéXcellent, J.-Y., 2000. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Method. Appl. Mech. Eng.* 184, 501–520.
- Aprille, T., Trick, T., 1972. A computer algorithm to determine the steady-state response of nonlinear oscillators. *IEEE Trans. Circuits Syst.* 19, 354–360.
- 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.
- Bernsen, E., Dijkstra, H.A., Wubs, F.W., 2008. A method to reduce the spin-up time of ocean models. *Ocean Modell.* 20, 380–392.
- Bischof, C.H., Roh, L., Mauer-Oats, A.J., 1997. ADIC: an extensible automatic differentiation tool for ANSI-C. *Softw. Pract. Exper.* 27 (12), 1427–1456.
- 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., Hindmarsh, A.C., 1986. Matrix-free methods for stiff systems of ODE's. *SIAM J. Numer. Anal.* 23, 610–638.
- Brown, P.N., Saad, Y., 1990. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.* 11, 450–481.
- Chan, T.F., Jackson, K.R., 1984. Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms. *SIAM J. Sci. Stat. Comput.* 5, 533–542.
- Coleman, T.F., Moré, J.J., 1983. Estimation of sparse jacobian matrices and graph coloring problems. *SIAM J. Numer. Anal.* 20, 187–209.
- Curtis, A.R., Powell, M.J.D., Reid, J.K., 1974. On the estimation of sparse jacobian matrices. *J. Inst. Math. Appl.* 13, 117–119.
- Davidson, B.D., 1997. Large-scale continuation and numerical bifurcation for partial differential equations. *SIAM J. Numer. Anal.* 34, 2008–2027.
- 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.
- Demmel, J.W., Eisenstat, S.C., Gilbert, J.R., Li, X.S., Liu, J.W.H., 1999. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Anal. Appl.* 20 (3), 720–755.
- Dennis, J.E., Schnabel, R.B., 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics, vol. 16. Society for Industrial and Applied Mathematics.
- Deuffhard, P., 2004. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, vol. 35. Springer Verlag.
- Dutkiewicz, S., Follows, M., Parekh, P., 2005. Interactions of the iron and phosphorus cycles: A three-dimensional model study. *Glob. Biogeochem. Cy.* 19 (doi:10.1029/2004GB002342).
- Dutkiewicz, S., Follows, M.J., Heimbach, P., Marshall, J., 2006. Controls on ocean productivity and air-sea carbon flux: An adjoint model sensitivity study. *Geophys. Res. Lett.* 33. doi:10.1029/2005GL024987.
- Eisenstat, S.C., Walker, H.F., 1996. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.* 17 (1), 16–32.
- Gear, C.W., Saad, Y., 1983. Iterative solution of linear equations in ODE codes. *SIAM J. Sci. Stat. Comput.* 4, 583–601.
- Gebremedhin, A., Manne, F., Pothén, A., 2005. What color is your Jacobian? Graph coloring for computing derivatives. *SIAM Rev.* 47, 629–705.
- Giering, R., Kaminski, T., 1998. Recipes for adjoint code generation. *ACM Trans. Math. Softw.* 24, 437–474.
- Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S., 2005. SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.* 31, 363–396.
- Kelley, C.T., 1995. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in applied mathematics, vol. 16. Society for Industrial and Applied Mathematics.
- Khatiwala, S., Visbeck, M., Cane, M., 2005. Accelerated simulation of passive tracers in ocean circulation models. *Ocean Modell.* 32, 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.

- Levitus et al., 1998. World Ocean Database 1998. NOAA Atlas NESDIS 18, US Government Printing Office, Washington, DC.
- Li, X., Primeau, F., 2008. A fast Newton–Krylov solver for seasonally varying global ocean biogeochemistry models. *Ocean Modell.*, in press.
- Li, X.S., Demmel, J.W., 2003. SuperLU_DIST: a scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Math. Softw.* 29 (2), 110–140. June.
- Mahowald, N., Lou, C., del Corral, J., Zender, C., 2003. Interannual variability in atmospheric mineral aerosols from a 22-year model simulation and observational data. *J. Geophys. Res.* 108. doi:10.1029/2002JD002821.
- Merlis, T., Khatiwala, S., 2008. Fast dynamical spin-up of ocean general circulation models using Newton–Krylov methods. *Ocean Modell.* 21, 97–105.
- 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.
- Orr, J.C., Najjar, R.G., Sabine, C.L., Joos, F., 1999. Abiotic-HOWTO: Internal OCMIP Report. LSCE/CEA Saclay, Gif-sur-Yvette, France, 25 pp.
- Parekh, P., Follows, M.J., Boyle, E.A., 2005. Decoupling of iron and phosphate in the global ocean. *Global Biogeochem. Cy.* 19. doi:10.1029/2004GB002280.
- Reisner, J., Mousseau, V.A., Wyszogrodzki, A.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. *Classics in Applied Mathematics*, vol. 16. Society for Industrial and Applied Mathematics.
- Shroff, G.M., Keller, H.B., 1993. Stabilization of unstable procedures: the recursive projection method. *SIAM J. Numer. Anal.* 30, 1099–1120.
- Skelboe, S., 1980. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. *IEEE Trans. Circuits 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.
- Tylavsky, D.J., Sohie, G.R.L., 1986. Generalization of the matrix inversion lemma. *Proc. IEEE* 74, 1050–1052.
- Wunsch, C., Heimbach, P., 2007. Practical global oceanic state estimation. *Physica D* 230, 197–208.
- Wunsch, C., Heimbach, P., 2008. How long to oceanic tracer and proxy equilibrium? *Quat. Sci. Rev.* 27, 637–651.