Ocean Modelling 109 (2017) 33–43

Contents lists available at ScienceDirect

Ocean Modelling

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

A Newton–Krylov solver for fast spin-up of online ocean tracers

Keith Lindsay[*]

National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307-3000, USA

ARTICLE INFO

Article history:
Received 31 March 2016
Revised 1 November 2016
Accepted 1 December 2016
Available online 2 December 2016

Keywords:
Newton–Krylov
Tracers
Ideal age
Ocean modelling

ABSTRACT

We present a Newton–Krylov based solver to efficiently spin up tracers in an online ocean model. We demonstrate that the solver converges, that tracer simulations initialized with the solution from the solver have small drift, and that the solver takes orders of magnitude less computational time than the brute force spin-up approach. To demonstrate the application of the solver, we use it to efficiently spin up the tracer ideal age with respect to the circulation from different time intervals in a long physics run. We then evaluate how the spun-up ideal age tracer depends on the duration of the physics run, i.e., on how equilibrated the circulation is.

1. Introduction

Analysis of ocean tracer simulations is complicated by the presence of drift (i.e., the simulated ocean tracers are not in balance with their boundary conditions and the ocean circulation). Drift in simulated tracers adds uncertainty to comparisons with observations and complicates the interpretation of tracer processes (e.g., source-sink terms and couplings between tracers), particularly when these processes are non-linear. Tracer drift also complicates the evaluation of tracer's response to transient forcing (e.g., rising atmospheric CO_2 or anthropogenic nutrient inputs), because the tracer's response can depend on the state of the tracer itself. Therefore, the capacity to generate tracer distributions in balance with their boundary conditions and the ocean circulation is advantageous. Because ventilation of the deep ocean takes in excess of 1000 years (Holzer and Primeau, 2010; DeVries and Primeau, 2011; Khatiwala et al., 2012), brute force tracer spin-up, running the tracers forward in time until this balance is reached, takes thousands of simulated years (i.e., multiple ventilation timescales). Running ocean models at a resolution typical of contemporary climate models (about 1° or $\mathcal{O}(10^6)$ grid points) for this duration is computationally expensive, and often prohibitive.

Previous efforts have overcome this hurdle, in the context of offline tracer transport, by using Newton–Krylov based solvers to directly solve for the spun-up tracer distributions (Li and Primeau, 2008; Khatiwala, 2008; Bardin et al., 2014). One approach to spin up tracers for use in an online ocean model is to use a Newton–Krylov solver in conjunction with an offline tracer transport model, where the offline model is using the circulation of the online model. A disadvantage of this approach is that offline models typically use approximations in their representation of the online model's circulation (e.g., using temporally averaged circulation operators), and it is unclear how these approximations affect the spun-up tracer distributions. While (Bardin et al., 2016) shows that the impact of these approximations on spun-up ideal age is small if the circulation is sampled frequently enough, the significance of the impact likely depends on the timescales of the processes affecting the tracers being spun up. Another disadvantage of this approach is that in practice, the tracers are either implemented separately in both the offline and online models, or the tracers are implemented in a way that is compatible with both models. For complex tracers, such as ecosystem models, the burden of having multiple tracer implementations, or of implementing the tracers compatibly with both models, can be excessive. Furthermore, it would not be easy to extend this approach to spinning up the active tracers of an ocean model. Because of these disadvantages, it is desirable to have a fast spin-up technique that is directly applicable to online ocean tracer simulations.

We present here the successful application of a Newton–Krylov based solver to efficiently spin up passive tracers in online ocean tracer simulations. We demonstrate this tool by efficiently spinning up the tracer ideal age (Thiele and Sarmiento, 1990) with respect to the circulation from different time intervals in a long physics run. We further evaluate how the spun-up ideal age tracer depends on the duration of the physics run, i.e., on how equilibrated the circulation is.

We describe, in Section 2, our formulation of the spin-up problem and the Newton–Krylov solver that we have developed to solve it, deferring some of the technical implementation details

[*] Corresponding author. Fax: +1 303 497 1700.
E-mail address: klindsay@ucar.edu

http://dx.doi.org/10.1016/j.ocemod.2016.12.001
1463-5003/

into appendices. We also describe in that section the computational experiments in which we use the solver. We present the results of these experiments in Section 3, discuss these results in Section 4, and give some concluding remarks in Section 5.

2. Methods

We first state the definition that we use of the spin-up problem for generic ocean tracers, mostly following the notation of Khatiwala (2008). Tracer concentrations, denoted by $\boldsymbol{c}(t)$, evolve in time according to the equation

$$\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{c}), \tag{1}$$

where $\boldsymbol{f}$ denotes the sum of mixing, advective, and source-sink tendencies of the tracers. In practice, $\boldsymbol{c}$ is a vector whose component values correspond to the ocean model's spatial discretization of tracers (e.g., grid cell means or spectral element decomposition coefficients). We use bold to denote vectors, such as $\boldsymbol{c}$, matrices, and other vector-valued operators. The solution of Eq. (1), with tracer concentrations at time t_0 specified by $\boldsymbol{c}(t_0)$, can be written as

$$\boldsymbol{c}(t) = \boldsymbol{\Phi}(t, t_0, \boldsymbol{c}(t_0)). \tag{2}$$

The spin-up problem is that for a given t_0 and time duration Δt, we seek initial tracer concentrations $\boldsymbol{c}(t_0)$ that leads to $\boldsymbol{c}(t_0 + \Delta t) = \boldsymbol{c}(t_0)$. That is, we seek a solution $\boldsymbol{c}(t_0)$ of the equation

$$\boldsymbol{G}(\boldsymbol{c}(t_0)) \equiv \boldsymbol{\Phi}(t_0 + \Delta t, t_0, \boldsymbol{c}(t_0)) - \boldsymbol{c}(t_0) = \boldsymbol{0}. \tag{3}$$

Such a solution is said to be spun-up with respect to the ocean model's circulation over the time interval t_0 to $t_0 + \Delta t$. Δt is typically an integer number of years. We refer the reader to Section 4.1 for some practical implications of this formal definition of the tracer spin-up problem.

The brute force approach to generating a spun-up solution to Eq. (3) is to run the tracers forward in time repeatedly from t_0 to $t_0 + \Delta t$, initializing the tracer concentrations at t_0 from the previous values at $t_0 + \Delta t$. Mathematically, this is equivalent to using fixed-point iterations on Eq. (2) to solve Eq. (3),

$$\boldsymbol{c}_{n+1} = \boldsymbol{\Phi}(t_0 + \Delta t, t_0, \boldsymbol{c}_n). \tag{4}$$

We note that repeatedly performing fixed-point iterations is equivalent to running the tracers forward in time from t_0 to $t = \infty$, if one repeats periodically the model's circulation and physical state variables forward in time from the interval t_0 to $t_0 + \Delta t$. As stated in Section 1, it takes thousands of model years to spin up tracers with such an approach, which is computationally expensive.

A different approach to find spun-up tracer concentrations is to apply Newton's method to Eq. (3). Newton's method is an iterative method to approximate the solution of a system of equations that proceeds as follows. Given the current iterate, $\boldsymbol{c}_n$, $\boldsymbol{G}(\boldsymbol{c})$ is approximated with a linearization in the neighborhood of $\boldsymbol{c}_n$,

$$\boldsymbol{G}(\boldsymbol{c}_n + \delta\boldsymbol{c}_n) \approx \boldsymbol{G}(\boldsymbol{c}_n) + \boldsymbol{J}_n \delta\boldsymbol{c}_n, \tag{5}$$

$\delta\boldsymbol{c}_n$ is found such that the right-hand side of Eq. (5) is zero,

$$\boldsymbol{J}_n \delta\boldsymbol{c}_n = -\boldsymbol{G}(\boldsymbol{c}_n), \tag{6}$$

and the next Newton iterate is set to

$$\boldsymbol{c}_{n+1} = \boldsymbol{c}_n + \delta\boldsymbol{c}_n. \tag{7}$$

In these equations, $\boldsymbol{J}_n$ is the Jacobian of $\boldsymbol{G}$ evaluated at $\boldsymbol{c}_n$,

$$\boldsymbol{J}_n \equiv \left.\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{c}}\right|_{\boldsymbol{c}_n}. \tag{8}$$

It represents how increments, $\delta\boldsymbol{c}$, to the tracer concentrations, $\boldsymbol{c}$, lead to changes in $\boldsymbol{G}$, as they evolve over the time interval t_0 to $t_0 + \Delta t$. Because the duration of the interval, Δt, spans many model time steps (i.e., we are not considering instantaneous changes), $\boldsymbol{J}_n$ is not sparse, making it difficult to compute all of its entries. This precludes using a direct method to solve Eq. (6). We instead use an iterative method.

Krylov subspace methods (Saad, 2003) are a class of iterative methods for solving $\boldsymbol{Ax} = \boldsymbol{b}$ that work by minimizing the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{Ax}$ over vectors from a Krylov subspace

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{b}) = \mathrm{span}\left\{\boldsymbol{b}, \boldsymbol{Ab}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{b}\right\}. \tag{9}$$

A notable feature of these methods is that they only utilize the matrix $\boldsymbol{A}$ through matrix-vector multiplications with $\boldsymbol{A}$, they do not directly access the individual entries of $\boldsymbol{A}$. Because of this feature, they are well suited for solving $\boldsymbol{Ax} = \boldsymbol{b}$ when the entries of $\boldsymbol{A}$ are not available, provided that one can compute matrix-vector multiplications with $\boldsymbol{A}$. Matrix-vector multiplications with our Jacobian matrix $\boldsymbol{J}_n$ can be approximated with the finite difference approximation

$$\boldsymbol{J}_n \delta\boldsymbol{c} \approx (\boldsymbol{G}(\boldsymbol{c}_n + \sigma\delta\boldsymbol{c}) - \boldsymbol{G}(\boldsymbol{c}_n))/\sigma, \tag{10}$$

for a suitably chosen scalar σ. For our $\boldsymbol{G}$, the right-hand side of Eq. (10) can be evaluated with two model runs, one to evaluate each $\boldsymbol{G}$ term, or a single model run if $\boldsymbol{G}(\boldsymbol{c}_n)$ has been previously computed. Krylov subspace methods are thus a feasible way to solve Eq. (6). The particular Krylov subspace method that we use is GMRES (Saad and Schultz, 1986), following the presentation in Saad (2003).

Summarizing, we use Newton's method, Eqs. (6) and (7), to solve Eq. (3). The increment in Newton's method, $\delta\boldsymbol{c}_n$, is obtained by using GMRES to solve Eq. (6). The matrix-vector multiplications required by GMRES are evaluated using Eq. (10). There are a number of technical details involved in implementing this Newton–Krylov solver. In particular, the batch computing environment that our ocean model is run within leads to some complications. We refer the reader to Appendix A for these technical details.

2.1. Krylov preconditioner

Following Li and Primeau (2008) and Khatiwala (2008), we apply a preconditioner to Eq. (6) to improve the convergence rate of the Krylov solver. We use a left preconditioner, yielding

$$\boldsymbol{P}_n \boldsymbol{J}_n \delta\boldsymbol{c}_n = -\boldsymbol{P}_n \boldsymbol{G}(\boldsymbol{c}_n), \tag{11}$$

where $\boldsymbol{P}_n$ is the preconditioner matrix. In order to improve the convergence rate of the Krylov solver, $\boldsymbol{P}_n$ should approximate the inverse of $\boldsymbol{J}_n$. With the introduction of a left preconditioner into the Krylov solver, matrix operations using the matrix $\boldsymbol{J}_n$, such as matrix-vector multiplications, are replaced with matrix operations using the matrix $\boldsymbol{P}_n\boldsymbol{J}_n$. These latter matrix-vector multiplications are implemented by first evaluating a matrix-vector multiplication using $\boldsymbol{J}_n$, evaluated using Eq. (10), and then multiplying the result by $\boldsymbol{P}_n$. In order to be of practical benefit, computing matrix-vector multiplications with $\boldsymbol{P}_n$ needs to be computationally feasible.

We use the same type of conditioner that (Li and Primeau, 2008) and (Khatiwala, 2008) both use. Following the derivation in Khatiwala (2008), we construct $\boldsymbol{P}_n$ by introducing $\tilde{\boldsymbol{\Phi}}$, an approximation to the ocean model's evaluation of $\boldsymbol{\Phi}$, and set $\boldsymbol{P}_n$ to be inverse of $\tilde{\boldsymbol{J}}$, the Jacobian that corresponds to $\tilde{\boldsymbol{\Phi}}$. As shown by Khatiwala (2008), a choice for $\tilde{\boldsymbol{\Phi}}$ that works well is to replace the ocean model's time stepping with a single backward Euler time step over the time interval t_0 to $t_0 + \Delta t$,

$$\tilde{\boldsymbol{\Phi}}_K - \boldsymbol{c}(t_0) = \Delta t \boldsymbol{f}(t_0 + \Delta t, \tilde{\boldsymbol{\Phi}}_K), \tag{12}$$

where the K subscript denotes that this is the approximation of Khatiwala (2008). In order to better capture the mean of the ocean model's circulation, we replace the instantaneous mixing and advective operators that effectively appear on the right-hand side