Available online at www.sciencedirect.com

SCIENCE DIRECT®

Mathematical and Computer Modelling 44 (2006) 870–886

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

Ocean modelling for climate studies: Eliminating short time scales in long-term, high-resolution studies of ocean circulation

Balasubramanya T. Nadiga[a], Mark Taylor[b], Jens Lorenz[c,*]

[a] *Los Alamos National Laboratory, MS-B296, Los Alamos, NM 87545, United States*
[b] *Sandia National Laboratories, Albuquerque, NM 87185, United States*
[c] *Department of Mathematics and Statistics, UNM, Albuquerque, NM 87131, United States*

Received 13 February 2006; accepted 22 February 2006

Abstract

On the decadal to centennial time scale, changes in climate are controlled strongly by changes in ocean circulation. However, because of limitations inherent to the time integration schemes used in present-day ocean models, state-of-the-art climate change simulations resolve the oceans only very coarsely. With an aim to enable long-term simulations of ocean circulation at the high resolutions required for a better representation of global ocean dynamics, we have implemented fully-implicit time integration schemes in a version of the popular ocean general circulation model POP (Parallel Ocean Program), employing Jacobian-free Newton–Krylov techniques. Here, we describe the numerical principles underlying iPOP in some detail and present a few computational results. While there are many advantages to this approach, including a consistent and uniform treatment of the terms in the governing equations, the primary advantage lies in the ability to take time steps that are of relevance to the physical phenomenon that is being studied. The time step is not limited (for stability reasons) by the fastest modes of the system.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Ocean general circulation models; Implicit parallel ocean program; Implicit time stepping; Jacobian-free Newton–Krylov iteration

1. Introduction

Studies of the variability of our climate system suggest that changes in the oceanic thermohaline circulation (THC) have the potential to trigger severe climate change events. A popular example of this is a scenario in certain climate models wherein a weakening of the thermohaline circulation in the North Atlantic from its present level due to doubled CO_2 levels in the atmosphere, triggers a change to ice-age-like conditions [1,2].

The myriad physical processes occurring on a vast range of spatial and temporal scales and coupling to each other, compounded by a paucity of long-term observations, make computational modelling of the climate system an indispensable tool in understanding climate variability. An important component of the climate system is the ocean, and on the decadal to centennial time scales, variability of the climate system is crucially controlled by the ocean. The computational challenge of ocean modelling arises from the need to accurately represent the immense range of spatial

* Corresponding author.
E-mail addresses: balu@lanl.gov (B.T. Nadiga), mataylo@sandia.gov (M. Taylor), lorenz@math.unm.edu (J. Lorenz).

0895-7177/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2006.02.021

and temporal scales that interact to produce the intricate and complex behavior of the ocean system. In this sense, the dynamics of ocean circulation is a prototypical *strongly coupled multiscale phenomenon*, and the modelling of such systems in general is an area of great current scientific interest.

While early studies of the weakening of the THC were based on *box models* of oceans (e.g. [3]) wherein the global ocean is idealized as a few (typically two or three) interconnected rectangular basins, the more recent studies (see [4] and references therein) use realistic Ocean General Circulation Models (OGCM) but at very coarse resolutions (300–400 km at the equator corresponding to about 3°–4° spacing in terms of latitude and longitude). While studies using these models have given insights into possible mechanisms for variability of the thermohaline circulation, the lack of detailed representation of dynamics, e.g. mesoscale eddy fields and topography in these models is a shortcoming that needs to be further worked on [4]. On the other hand, notwithstanding numerous issues regarding closure of the equations at the smallest resolved scales, studies of ocean circulation on the shorter (annual to decadal) time scale have shown that the global ocean circulation can be modelled quite realistically using high resolution (grid spacings of around 10 km or approximately 0.1° of latitude and longitude at the equator (e.g., see [5,6]). An important result coming out of these studies is the major role played by detailed topography. Furthermore, these results also seem to suggest the importance of the mesoscale eddy field in shaping the mean circulation. Thus, further systematic studies of possible climate change scenarios would be greatly helped by the ability to carry out long-term (centennial to millennial time scale) simulations of the world oceans using fairly high resolutions.

The adjustment of dynamical and thermodynamical processes in the ocean to external forcings occur over a wide range of time scales. While barotropic adjustment occurs on the order of days, slow, mid-latitude, planetary waves may take years to adjust, and the thermohaline processes equilibrate only on the centennial–millennial time scale. Furthermore, the adjustment or quasi-equilibration of these processes are mediated by a variety of inertia-gravity waves (e.g. see [7]). To deal with these different characteristic times, most present-day ocean models split the governing equations, in a rather ad hoc fashion, into subsystems that purport to include dynamics over a limited range of time scales, e.g., fast, slow, very slow, etc. Computational savings are then realized when these subsystems are solved using different time steps, with the subsystems with slower dynamics being solved with longer time steps.

As an example, the external gravity waves that are the fastest dynamics of the primitive equation system (with wave speeds that are given by $\sqrt{gD}$, where g is the acceleration due to gravity and D is the depth of the ocean, of about 200 m/s) are isolated into a two-dimensional system, the so-called barotropic mode. With such a restriction of the fastest modes of the full system to a two-dimensional subsystem, substantial savings in computational costs can be achieved since, now, the rest of the system, the so-called baroclinic subsystem, can be evolved using longer time steps. The fast two-dimensional subsystem (in which the baroclinic forcing term is held constant) is solved either implicitly in the semi-implicit formulation, e.g., as in POP (Parallel Ocean Program) [8] or explicitly using shorter time steps in the split-explicit formulation, e.g., as in MICOM (Miami Isopycnal Coordinate Ocean Model) [6], with the slower 3D baroclinic subsystem being treated explicitly in both cases.

While this type of splitting is natural in a linear system with horizontally-homogeneous stratification, and a flat top and bottom, it is ad hoc in the presence of horizontally-variable stratification, realistic bottom topography, free-surface top boundary condition, and nonlinearities of the governing equations. These latter factors are likely to lead to a leakage of the fast modes into the intended slow system, and therefore require further ad hoc fixes so that the baroclinic subsystem can actually be stably integrated using a longer time step. (In a further extension of this idea, but at the cost of introducing additional ad hoc approximations and errors, along with the barotropic modes, the first few internal gravity wave modes can similarly be isolated into a hierarchy of two-dimensional subsystems.)

As previously mentioned, present day OGCMs when used for climate studies resolve the oceans only very coarsely (3°–4° in longitude and latitude), and in the process fail to represent the most energetic part of the kinetic energy spectrum, the mesoscales. Consequently, the flow velocities encountered in these simulations are rather small and the resulting circulation sluggish. With this coupled to the barotropic–baroclinic splitting used in most OGCMs, one would expect to be able to take reasonably long time steps—time steps that would correspond to advective CFL numbers of about 1. However, the time step actually used at these coarse resolutions tends to be much shorter – about an hour(e.g., see [9,10]) – corresponding to advective CFL numbers much smaller than one, clearly demonstrating limitations of the presently used techniques of time integration used in present day OGCMs.

Finally, long term studies of ocean circulation typically use an artificial acceleration of the slow evolution of tracers (sometimes with an acceleration that is depth-dependent) below the upper ocean [11–14], at least in the spin-up phase, to further reduce computational costs. This kind of tracer acceleration in possible conjunction with artificial slow down of momentum modes is called the distorted physics technique. It is untenable either when the transients are of interest or are of importance (as due to the presence of mesoscale eddies) or when the thermodynamics of the ocean system itself is not in a state of equilibrium, but rather engenders variability as in climate change scenarios. Furthermore, these kinds of techniques additionally introduce time step sensitivities and other unfavorable interactions with other physical processes in OGCMs [15]. Thus, the presently used techniques of time integration in ocean models are rather ill-suited for climate scale studies.

Thus, a fully implicit approach, in which all components of ocean circulation are handled in a consistent and uniform fashion, may be well-suited for the centennial–millennial scale simulations of ocean circulation. With implicit methods, the size of the time step is restricted only by accuracy, not by stability. This allows us to "step over" the fast time scales and use a smaller number of larger time steps to achieve long-term simulations. Physically, this may be thought of as effectively evolving the slow modes of the system while continually equilibrating the faster modes. To illustrate, consider a model time-dependent problem that includes both fast and slow time scales:

$$\mathbf{u}_t = \mathbf{f}(\mathbf{u}), \tag{1.1}$$

where $\mathbf{u}$ is a state vector and $\mathbf{f}$ is nonlinear. The backward Euler (BE) method is the simplest approach for implicitly advancing the solution in time. For BE, the time-advanced solution $\mathbf{u}_h^n$ is the approximate solution of

$$\mathbf{F}(\mathbf{u}_h) \equiv \mathbf{u}_h - \mathbf{u}_h^{n-1} - \Delta t \mathbf{f}_h(\mathbf{u}_h) = 0, \tag{1.2}$$

where Δt is the time step, the subscript h refers to spatially discretized quantities, and the superscript refers to the discrete time step; $\mathbf{u}_h$ is the exact solution of (1.2).

In (1.2), if Δt is taken to infinity, then both the fast scales and the slow scales are exactly equilibrated and a steady state is obtained. It is in this limit that Dijkstra, Weijer and co-workers [16–19] have extensively investigated the stability of the global overturning circulation. In these works, the steady states and their local bifurcation structure were computed using a Newton–Krylov formulation in conjunction with continuation techniques. However, if Δt is chosen to resolve the slow dynamics, but at the same time is much larger than the fast time scale, then a solution of the nonlinear algebraic system (1.2) results in only the fast dynamics being equilibrated. That is to say, the main dynamical evolution is that of the slow modes, and they are accurately resolved. Clearly, the evolution of the fast modes is distorted, but it is important to note that in this implicit setting the fast modes can drive the slow modes. This is in contrast to a situation where the fast modes are eliminated and hence cannot possibly influence the slow modes. Finally, in the implicit setting, the time step is not limited by the fast dynamics.

It is natural to employ Newton's method to solve the nonlinear algebraic system (1.2). In such a case, a linear system involving the Jacobian of $\mathbf{F}$ with respect to $\mathbf{u}$ arises. Considering the complexity of $\mathbf{F}$ in an OGCM, it would seem difficult to compute and store the Jacobian matrix. However, what is natural in OGCMs is for a subroutine to act on a state vector and transform it into another state vector. Thus we would prefer not to explicitly compute and store the Jacobian matrix, but rather deal only with matrix–vector products. Finally, Krylov space techniques are attractive for solving the resulting linear systems. All of these ingredients have been combined into what are called Jacobian-Free Newton–Krylov (JFNK) methods to solve systems of nonlinear algebraic equations in the field of Computational Fluid Dynamics (CFD) (e.g., see [20] or, for a recent review, see [21]). JFNK-based time-integration has been applied in the geophysical context to one-dimensional and two-dimensional shallow water equations by Reisner et al. [22] and Mousseau et al. [23], and, as mentioned earlier, Dijkstra, Weijer and co-workers [16–19] apply the Newton–Krylov methodology to compute steady states but also make explicit use of the Jacobian.

We have implemented, using JFNK techniques, fully-implicit evolution schemes in a version of POP that we call iPOP. This approach allows for high-resolution modelling of ocean circulation using a computational time step that is relevant to the problem under consideration, rather than one based on stability requirements dictated by the fastest process—be it a day, a month, or a year. However, we stress that for the large-scale simulations that we envisage for studying the variability of the global thermohaline circulation, the viability of this approach depends not just on the ability to take long time steps but also on the efficiency and scalability of the method. While scalability to massively parallel machines is assured by the Jacobian-free nature of the method (e.g., see [20]), efficiency is a more difficult issue. The efficiency of this approach is presently not good in the sense that it is about an order of magnitude slower compared to semi-implicit time stepping, and clearly requires more work. Improved efficiency is contingent upon better matrix-free preconditioning and better scalable (parallel) matrix-free linear solvers—both areas of intense ongoing mathematical research, and beyond the scope of the present article. In fact, we intend this article to be of a descriptive nature, presenting various aspects of the algorithm that go into the fully-implicit time integration framework and showing some results. We defer a discussion of important issues such as those related to accuracy and efficiency to future articles.

2. JFNK-based fully-implicit time integration

2.1. Governing equations

The starting point for describing the fluid dynamics of the ocean are the Navier–Stokes equations with rotation. However, scale analysis indicates that certain approximations can be made if the interest is in describing large scale circulation. In such a case, with circulation velocities everywhere being much smaller than the propagation speed of sound, the flow may be considered incompressible, with changes in density limited to those acting as sources or sinks of buoyancy. This leads to the Boussinesq approximation. Further, since the vertical scales of motion are much smaller (the oceans are typically only about 5 km deep) than the corresponding horizontal scales of motion (that can extend to the global scales), the vertical momentum equation can be well-approximated by hydrostatic balance—a balance between the vertical pressure gradient term and buoyancy. Computationally, the hydrostatic approximation removes the need to invert a three-dimensional elliptic equation for the pressure that is used to ensure incompressibility otherwise; instead incompressibility is now used to diagnose the vertical velocity w at each time step given the evolving horizontal velocity $\mathbf{u}$. The equations resulting from an application of the Boussinesq and hydrostatic approximations to the Navier–Stokes equations are termed the Hydrostatic Primitive Equations (HPE).

When the primitive equations are considered with the vertical coordinate being depth z, the primary unknowns are the horizontal velocity $\mathbf{u}$, and a pair of tracers, temperature, T, and salinity, S. These variables are to be determined in the three-dimensional region of the global ocean that is laterally interrupted by continents and bounded at the bottom by specified topography, $H(x, y)$. The upper surface of the domain η is however free to evolve dynamically, $\eta = \eta(x, y, t)$, and as such the primitive equations include a free boundary problem. The boundary conditions include: no normal flow at all solid boundaries and no slip at lateral boundaries. Finally, the boundary conditions at the top surface are determined from interactions of the oceanic mixed layer with the overlying atmosphere; these are the main drivers of circulation.

The governing equations may be thought of conveniently in terms of the explicit evolution of the variables as follows. Tracers, which serve as the source of buoyancy changes, evolve according to the advection–diffusion equation

$$\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}t} = \kappa_h \nabla_h^2 \mathbf{T} + \frac{\partial}{\partial z}\kappa_v \frac{\partial \mathbf{T}}{\partial z} + \mathbf{F_T} \tag{2.1}$$

where ∇_h is the horizontal gradient operator, $\frac{\mathrm{d}}{\mathrm{d}t}$ is the substantial derivative given by

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla_h + w\frac{\partial}{\partial z}, \tag{2.2}$$

$\mathbf{u}$ is the horizontal velocity and w the vertical velocity. The dissipative terms in the tracer evolution equation are thought of as parametrizations of turbulent diffusion. They are split into horizontal and vertical components to reflect the stratified nature of the flow: with vertical (more correctly diapycnal) motion being inhibited and turbulent mixing in the horizontal (more correctly isopycnal) direction being much greater. In practice, the mixing terms are usually rotated to be in the isopycnal–diapycnal directions rather than in the vertical–horizontal directions as written. These terms are, of course, open to further modeling. The forcing terms $\mathbf{F_T}$ are parametrizations of heat and salinity fluxes at the air–sea interface and are specified.

While the horizontal velocity $\mathbf{u}$ evolves under the Boussinesq approximation as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + f\mathbf{k}\times\mathbf{u} = \frac{1}{\rho_r}\nabla_h p + \nabla\cdot(\nu\nabla\mathbf{u}) + \mathbf{F_u}, \tag{2.3}$$

the vertical velocity is diagnosed from the incompressibility condition

$$\nabla_h\cdot\mathbf{u} + \frac{\partial w}{\partial z} = 0. \tag{2.4}$$

The dissipative term in the horizontal momentum equations (2.3) is thought of as a turbulence closure and is open to modeling, but is here simply represented by a specified turbulent eddy viscosity. The forcing $\mathbf{F_u}$ is again specified and primarily consists of parametrizations of momentum exchange at the air–sea interface.

The pressure required to evolve the horizontal velocity consists of two components—the surface component and the hydrostatic component. The surface component is found from the sea-surface height (SSH) η that also constitutes the upper free boundary of the domain, as previously mentioned. The evolution of SSH η is related to the (previously diagnosed) vertical velocity w at the surface, but may be conveniently expressed by a vertical integration of the incompressibility condition (or continuity equation) as follows

$$\frac{\partial \eta}{\partial t} + \nabla \cdot [(H + \eta)\widehat{\mathbf{u}}] - q_w = 0, \tag{2.5}$$

where $H(x, y)$ is specified topography, $\widehat{\mathbf{u}}$ is the vertically averaged horizontal velocity and q_w are specified sources due to evaporation and precipitation. The surface component then acts as the boundary condition and the full pressure is diagnosed from the hydrostatic relation

$$\frac{\partial p}{\partial z} = -\rho g, \tag{2.6}$$

where ρ is the density and is determined from temperature, salinity and pressure according to the equation of state

$$\rho = \rho(T, S, p). \tag{2.7}$$

Finally, having made the hydrostatic approximation, as is appropriate for the dynamics of the large scales that we are interested in, any unstable stratification that may result, as for example from buoyancy forcing at the surface, will have to be additionally resolved. This is achieved through a parametrization of unresolved small-scale convective processes, and is called convective adjustment. Convective adjustment involves either a vertical mixing of the water column using a diagnosed vertical diffusivity that is large in regions of unstable stratification or an iterative mixing of vertically-adjacent cells that are unstably stratified.

2.2. *Numerical approach*

In this section we describe, in some detail, the numerical approach that we took to approximate the governing equations described in the previous section. Since our starting point is the POP OGCM, the POP reference manual [24] gives all relevant details about the governing equations, approximations and spatial discretization. The basic ingredients are well-known. After discretizing in space, one obtains a method-of-lines system that we write in the form

$$\frac{\mathrm{d}U}{\mathrm{d}t} = f(t, U), \quad U(t) \in \mathbb{R}^N. \tag{2.8}$$

This system of ODEs is discretized in time using an implicit method, ensuring that the time step Δt is not restricted by stability requirements. In each time step one has to solve a nonlinear system of equations with N unknowns. We apply a preconditioned Jacobian-free Newton–Krylov iteration. The aim of this section is to describe the numerical properties of the ingredients in some detail.

Space discretization

We use the second-order difference formulas as provided by POP [24].

Time-stepping

Let $U_n \in \mathbb{R}^N$ denote the computed numerical approximation at time $t = t_n$ and let Δt denote the current time step. We want to obtain an approximation U_{n+1} for $U(t_{n+1})$ where $t_{n+1} = t_n + \Delta t$. In the simplest case, U_{n+1} is the solution of the backward Euler equation,

$$U_{n+1} - U_n = \Delta t\, f(t_{n+1}, U_{n+1}). \tag{2.9}$$

The local truncation error of the formula is $\mathcal{O}(\Delta t)$. The Crank–Nicholson formula

$$U_{n+1} - U_n = \frac{\Delta t}{2} \left(f(t_n, U_n) + f(t_{n+1}, U_{n+1}) \right) \tag{2.10}$$

has a local truncation error of order $\mathcal{O}((\Delta t)^2)$. In future work, we also plan to use backward differentiation formulas (BDF) and, possibly, methods of Radau type. We refer to [26] for details of these methods.

Assuming $t_{n-1} = t_n - \Delta t$, the 2nd order BDF formula reads

$$\frac{3}{2} U_{n+1} - 2U_n + \frac{1}{2} U_{n-1} = \Delta t \, f(t_{n+1}, U_{n+1}). \tag{2.11}$$

Remarks on stability

It is common to discuss stability properties of time-stepping processes like (2.9)–(2.11) by applying them to the scalar model equation

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \lambda U \tag{2.12}$$

where λ is a complex parameter. One may think of λ as an eigenvalue of the Jacobian $f_U(t, U)$ for $t \sim t_n$ and $U \sim U_n$. Precise relations between numerical properties of a method when applied to the model problem (2.12) and when applied to a nonlinear system (2.8) are non-trivial, however. We ignore this complication in the following discussion.

If the backward Euler formula (2.9) is applied to (2.12), one obtains

$$U_{n+1} = g_{BE}(\lambda \Delta t) U_n \quad \text{with } g_{BE}(z) = \frac{1}{1-z}.$$

The growth function $g_{BE}(z)$ satisfies

$$|g_{BE}(z)| < 1 \quad \text{for } \operatorname{Re} z < 0,$$

which makes the backward Euler formula A-stable. We also note that

$$g_{BE}(z) \to 0 \quad \text{as } |z| \to \infty,$$

which implies good damping properties of the method.

An application of the Crank–Nicolson formula (2.10)–(2.12) yields

$$U_{n+1} = g_{CN}(\lambda \Delta t) U_n \quad \text{with } g_{CN}(z) = \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}}.$$

Again, the growth function satisfies

$$|g_{CN}(z)| < 1 \quad \text{for } \operatorname{Re} z < 0$$

implying A-stability of the Crank–Nicholson method. We note that

$$g_{CN}(z) \to -1 \quad \text{as } |z| \to \infty,$$

thus there is some danger that unresolved time oscillations remain undamped.

An application of the backward differentiation formula (2.11), (2.12) yields the second order difference formula

$$\left(\frac{3}{2} - z \right) U_{n+1} = 2U_n - \frac{1}{2} U_{n-1} \quad \text{with } z = \lambda \Delta t.$$

Assuming

$$U_n = \zeta^n, \quad n = 1, 2, \ldots$$

where ζ^n is the n-th power of a complex number ζ, one obtains the quadratic equation

$$\left(\frac{3}{2} - z\right)\zeta^2 - 2\zeta + \frac{1}{2} = 0 \tag{2.13}$$

for ζ. Let $\zeta_{1,2}(z)$ denote the two roots of (2.13). It is elementary to show that

$$|\zeta_{1,2}(z)| < 1 \quad \text{for } \operatorname{Re} z < 0$$

which makes the BDF formula (2.11) A-stable. Also,

$$\zeta_{1,2}(z) \to 0 \quad \text{as } |z| \to \infty,$$

which gives the method better damping properties than the Crank–Nicolson formula. Of course, an advantage of the Crank–Nicolson formula (2.10) over the BDF formula (2.11) is storage: (2.10) requires storage of the solution at a single previous time level, whereas (2.11) requires storage of solutions at two previous time levels.

Solution of nonlinear systems

Any of the time discretization methods described above lead to nonlinear systems of equations that need to be solved at each time step. We write such a system as

$$F(x) = 0, \quad x \in \mathbb{R}^N. \tag{2.14}$$

For example, if the backward Euler formula (2.9) is employed for time-stepping, then one must solve the system

$$F(U_{n+1}) = 0$$

where

$$F(x) = x - U_n - \Delta t f(t_{n+1}, x), \quad x \in \mathbb{R}^N.$$

A basic iterative scheme, and the starting point of our discussion, is the classical Newton method. If x_k is the current approximation to an exact solution x^* of (2.14), then one solves the linear system

$$F'(x_k)s_k = -F(x_k), \quad s_k \in \mathbb{R}^N, \tag{2.15}$$

for s_k and sets

$$x_{k+1} = x_k + s_k.$$

In (2.15), $F'(x_k)$ denotes the Jacobian of $F(x)$ evaluated at $x = x_k$. A main advantage of Newton's method is its fast local convergence. If $F'(x^*)$ is nonsingular and if x_0 is sufficiently close to x^*, then

$$\|x^* - x_{k+1}\| \le C\|x^* - x_k\|^2, \quad k = 0, 1, \ldots. \tag{2.16}$$

Here C is a constant which is independent of k and x_0. Roughly speaking, the estimate (2.16) says that the number of significant digits in the approximation x_k doubles with each Newton iteration.

Inexact Newton iterations

However, given the difficulty of finding a good initial guess, inexact Newton iterations are commonly used and may be briefly described as first solving for an approximate Newton direction s_k such that

$$\|F'(x_k)s_k + F(x_k)\| \le \eta_k \|F(x_k)\| \tag{2.17}$$

and then going only partway along that direction:

$$x_{k+1} = x_k + \lambda_k s_k. \tag{2.18}$$

Jacobian-free Newton–Krylov

For large scale systems $F(x) = 0$ it is not feasible to build the Jacobian matrix $F'(x_k)$ and to solve the linear system (2.15) using direct methods like Gaussian elimination or LU-factorization. There are many ways to compute approximations for s_k, be it either the exact solution of the Newton system (2.15), or the solution of the inexact Newton system (2.17). Krylov subspace methods are attractive since they require only computations of matrix–vector products $F'(x_k)v$, and these may be approximated by the difference formula

$$F'(x_k)v \sim \frac{1}{\varepsilon}(F(x_k + \varepsilon v) - F(x_k)).$$

Using this approximation, it is clear that the Jacobian $F'(x_k)$ never has to be formed explicitly. Such an approach to approximating the solution of the system $F(x) = 0$ is commonly referred to as the Jacobian-free Newton–Krylov method.

For ease of reference and to gain a better understanding of the expected properties of the whole iterative process, we next recall the GMRES algorithm for solving a linear system, $Ax = b$.

GMRES: an iterative solver for linear systems

The generalized minimum residual method GMRES, which belongs to the class of Krylov based iterative methods, was proposed in [27] to solve large, sparse, non-symmetric linear systems $Ax = b$. In our code we use the public domain software GMRES as provided in [25].

We first describe the algorithm in its well-known *basic form*. We then comment on preconditioning and restarting. We use the notation

$$\langle x, y\rangle = \sum x^{(j)} y^{(j)}, \quad |x| = \langle x, x\rangle^{1/2}$$

for the Euclidean inner product and norm on $\mathbb{R}^N$. Here $x^{(j)}$ is the j-th component of the column vector $x \in \mathbb{R}^N$.

Let $x_0 \in \mathbb{R}^N$ be an initial guess for the solution of the system

$$Ax = b \tag{2.19}$$

and let

$$r_0 = b - Ax_0$$

denote its residual. GMRES builds approximate solutions for (2.19) of the form

$$x_m = x_0 + V_m y_m, \quad m = 1, 2, \ldots \tag{2.20}$$

where $V_m \in \mathbb{R}^{N\times m}$ is a matrix with m orthonormal columns which form a basis of the Krylov space

$$\mathcal{K}_m = \mathcal{K}_m(x_0) = \mathrm{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}.$$

The vector $y_m \in \mathbb{R}^m$ is determined so that the Euclidean norm of the residual $r_m = b - Ax_m$ is minimized, i.e.,

$$|b - x_0 - AV_m y_m| < |b - x_0 - AV_m y| \quad \text{for all } y \in \mathbb{R}^m,\ y \neq y_m.$$

Computation of V_m and H_m

The m orthonormal columns $v_j \in \mathbb{R}^N$ of V_m can be computed by the well-known Arnoldi iteration. We describe this iteration here using the *modified Gram–Schmidt process*, which proceeds as follows:

```
 $v_1 = r_0/|r_0|$
for $j = 1$ to $m$
 $v = Av_j$
for $i = 1$ to $j$
 $h_{ij} = \langle v_i, v\rangle$
 $v = v - h_{ij} v_i$
end i
 $h_{j+1,j} = |v|$
 $v_{j+1} = |v|/h_{j+1,j}$
end j
```

Remark. The software [25] offers other options besides modified Gram–Schmidt, like iterative classical Gram–Schmidt. The latter is attractive in a distributed memory environment.

Upon completion of the above MGS process, one has obtained $m + 1$ orthonormal vectors

$$v_1, \dots, v_{m+1} \in \mathbb{R}^N$$

and numbers h_{ij} for $j = 1$ to m and $i = 1$ to $j + 1$. The numbers h_{ij} can be used to form the upper Hessenberg matrices

$$\tilde{H}_m \begin{pmatrix} h_{11} & & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ & \ddots & \ddots & \vdots \\ & & h_{m,m-1} & h_{mm} \\ 0 & & & h_{m+1,m} \end{pmatrix} \in \mathbb{R}^{(m+1)\times m}$$

and

$$H_m \begin{pmatrix} h_{11} & & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ & \ddots & \ddots & \vdots \\ & & h_{m,m-1} & h_{mm} \end{pmatrix} \in \mathbb{R}^{m\times m}.$$

Note that the square matrix H_m is obtained from $\tilde{H}_m$ by dropping the last row.

Let

$$V_m = (v_1, \dots, v_m), \quad V_{m+1} = (v_1, \dots, v_m, v_{m+1}).$$

It is not difficult to show that

$$AV_m = V_{m+1}\tilde{H}_m. \tag{2.21}$$

In fact, the j-th column of (2.21) states that

$$Av_j = \sum_{i=1}^{j+1} h_{ij} v_i,$$

which follows directly from the definition of the numbers h_{ij} and the orthonormal vectors v_i in the modified Gram–Schmidt process.

Eq. (2.21) implies that

$$V_m^T A V_m = H_m. \tag{2.22}$$

Roughly speaking, Eq. (2.22) says that the low-dimensional matrix H_m represents the action of A restricted to the Krylov space $\mathcal{K}_m$, followed by the orthogonal projection onto $\mathcal{K}_m$. Therefore, one can expect that the m eigenvalues of H_m are approximations to m eigenvalues of A. In fact, typically, the extreme eigenvalues of A are approximated by the eigenvalues of H_m. This is reasonable since the Krylov space $\mathcal{K}_m$ encodes the *repeated* action of A, which emphasizes the extreme eigenvalues. It is therefore reasonable to monitor if the eigenvalues of H_m lie in the stability regions of the time stepping method employed.

Remark. A precise interpretation of $H_m = V_m^T A V_m$ is as follows. Let $V_m y \in \mathcal{K}_m$ be arbitrary and let $z = H_m y$. The vectors

$$V_m y = \sum_j y^{(j)} v_j \quad \text{and } V_m z = \sum_j z^{(j)} v_j$$

lie in $\mathcal{K}_m$. The numbers $y^{(j)}$ and $z^{(j)}$ are their coordinates in the basis $v_1, \dots, v_m$ of $\mathcal{K}_m$. We have

$$V_m z = V_m H_m y = V_m V_m^T A V_m y.$$

Here $Q := V_m V_m^T$ satisfies $Q^2 = Q = Q^T$ and $range Q = \mathcal{K}_m$. Thus $Q \in \mathbb{R}^{N\times N}$ is the orthogonal projection of $\mathbb{R}^N$ onto $\mathcal{K}_m$. If one takes an arbitrary $V_m y \in \mathcal{K}_m$, applies A and then the projection Q, then one obtains $V_m z \in \mathcal{K}_m$, where $z = H_m y$. In other words, the transformation

$$y \to z = H_m y$$

expresses the action of A on $\mathcal{K}_m$, followed by Q, in the basis $v_1, \ldots, v_m$.

The least-squares problem

First let $y \in \mathbb{R}^m$ be arbitrary and set

$$x = x_0 + V_m y.$$

The residual of x is

$$\begin{aligned} b - Ax_0 - AV_m y &= r_0 - AV_m y \\ &= r_0 - V_{m+1}\tilde{H}_m y. \end{aligned}$$

We want to solve the minimization problem. Find $y \in \mathbb{R}^m$ with

$$|r_0 - V_{m+1}\tilde{H}_m y| = \min. \tag{2.23}$$

Set $\beta = |r_0|$. Since $v_1 = r_0/\beta$ is the first column of V_{m+1} we have

$$r_0 = \beta V_{m+1} e_1, \quad e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{m+1}.$$

Therefore,

$$r_0 - V_{m+1}\tilde{H}_m y = V_{m+1}(\beta e_1 - \tilde{H}_m y).$$

Finally, since the columns of V_{m+1} are orthonormal, we have

$$|V_{m+1}(\beta e_1 - \tilde{H}_m y)| = |\beta e_1 - \tilde{H}_m y|.$$

Consequently, the minimization problem (2.23) is equivalent to: find $y \in \mathbb{R}^m$ with

$$|\beta e_1 - \tilde{H}_m y| = \min \quad \text{where } \beta = |r_0|. \tag{2.24}$$

This least-squares problem can be solved as usual via QR-factorization of $\tilde{H}_m$. Thanks to the Hessenberg structure of $\tilde{H}_m$ the number of floating point operations is $\mathcal{O}(m^2)$. Further savings are possible if one does not factorize the successive matrices $\tilde{H}_1, \tilde{H}_2, \ldots$ independently, but implements an updating process to get the QR-factorization of $\tilde{H}_{m+1}$ from that of $\tilde{H}_m$. Only a single Givens rotation with $\mathcal{O}(m)$ floating point operations is required. For details we refer to [25] where this updating process is implemented.

Preconditioning

The convergence of GMRES applied to a system $Ax = b$ may be slow. To accelerate convergence of the iteration one often transforms the given system to an equivalent one by so-called preconditioning. The general ideas are well-known. Nevertheless, a theoretical analysis is typically not possible, and the design of good preconditioners remains a field of numerical analysis that requires experience and testing.

Let us first describe *left preconditioning*. If $M \in \mathbb{R}^{N\times N}$ is nonsingular, one replaces the given system $Ax = b$ by

$$M^{-1}Ax = M^{-1}b \tag{2.25}$$

and applies GMRES to (2.25). The matrix M^{-1} is not formed explicitly. Instead, in the modified Gram–Schmidt process given above, where the assignment $v = Av_j$ must be replaced by $v = M^{-1}Av_j$, one solves the linear system

$$Mv = Av_j \tag{2.26}$$

for v. To obtain a meaningful algorithm it is necessary that (1) the system $Mv = Av_j$ is much easier to solve than the given system $Ax = b$; (2) the GMRES iteration for $M^{-1}Ax = M^{-1}b$ converges faster than the GMRES iteration for $Ax = b$.

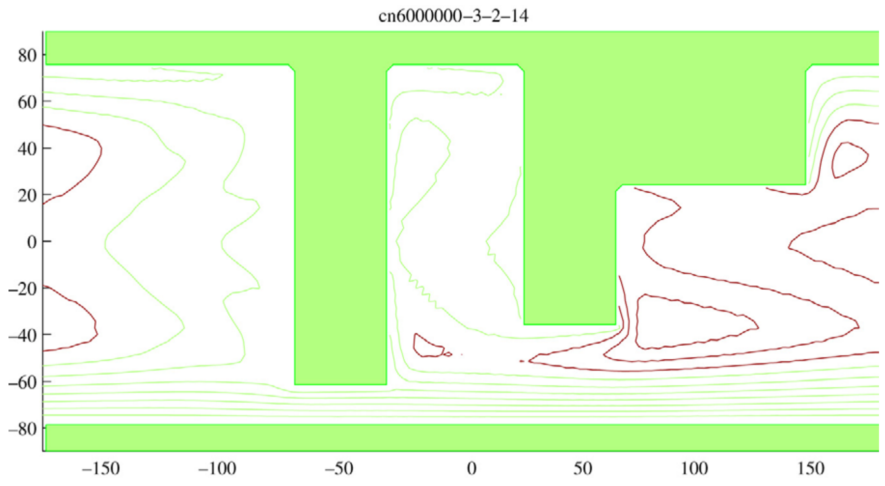

Fig. 1. Sea-surface height (SSH) from a preliminary iPOP run demonstrating the capability of iPOP to take arbitrarily large time steps. In practice, the time step is chosen to resolve only the processes of interest in the study. The iPOP time step is about 800 times that used in POP and other traditional OGCMs.

Second, *right preconditioning* proceeds as follows. The given system $Ax = b$ is replaced by

$$AM^{-1}\xi = b \quad \text{for } \xi = Mx. \tag{2.27}$$

In the modified Gram–Schmidt process the assignment $v = AM^{-1}v_j$ is carried out by first solving $Mw = v_j$ for w and then computing $v = Aw$.

In practice one often applies *left and right preconditioning* together. If M_1 and M_2 are nonsingular matrices, then the given system $Ax = b$ is replaced by

$$M_1^{-1}AM_2^{-1}\xi = M_1^{-1}b \quad \text{with } \xi = M_2x.$$

In the modified Gram–Schmidt process the assignment

$$v = M_1^{-1}AM_2^{-1}v_j$$

is carried out as follows: (1) solve $M_2w = v_j$ for w; (2) compute Aw; (3) solve $M_1v = Aw$ for v.

GMRES with restart

For simplicity, let us assume that GMRES is applied without preconditioning. (Similar considerations hold for the preconditioned iteration.) After the vector $y_m \in \mathbb{R}^m$ is computed by solving the least-squares problem (2.24), the vector

$$x_m = x_0 + V_m y_m$$

(see (2.20)) can be formed to yield the current approximation of the solution of $Ax = b$. Here the matrix V_m has m orthonormal columns, $v_1, \ldots, v_m$, each of length N. Since N is typically very large, the storage of the vectors v_j is prohibitive for larger m. To save storage it is common to restart GMRES after a rather small number of iterations, e.g., after $m = 20$ iterations. One can restart at the last computed approximation, x_m.

2.3. Example computations

To illustrate a first use of iPOP, Fig. 1 shows the sea-surface height (SSH) in an idealized configuration of continental geometry and wind forcing. This flow was computed at a time step that is roughly 800 times larger than with the traditional OGCMs—presently POP. While the flow is qualitatively similar to that obtained with the much smaller time step, the differences will be the subject of further study. Table 1 gives details about the full range of time integration schemes that we have used for this problem. Note that the time step is measured in terms of the

Table 1
Crank–Nicholson (NK) is the time stepping method of choice in iPOP, and semi-implicit is the method of choice in POP

Integration scheme	$\sqrt{gH}\frac{\Delta t}{\Delta x}$	vis CFL	adv CFL
2nd order Runge–Kutta	0.49	0.07	0.006
Explicit-leapfrog	0.53	0.07	0.006
Semi-implicit	4.40	0.60	0.05
Semi-implicit (Robert)	8.5	1.8	0.1
Backward Euler (NK)	176	24	2.0
Crank–Nicholson (NK)	3520	480	40

Note therefore, that the flow in Fig. 1 was run with an iPOP time step about 800 times larger than the POP time step.

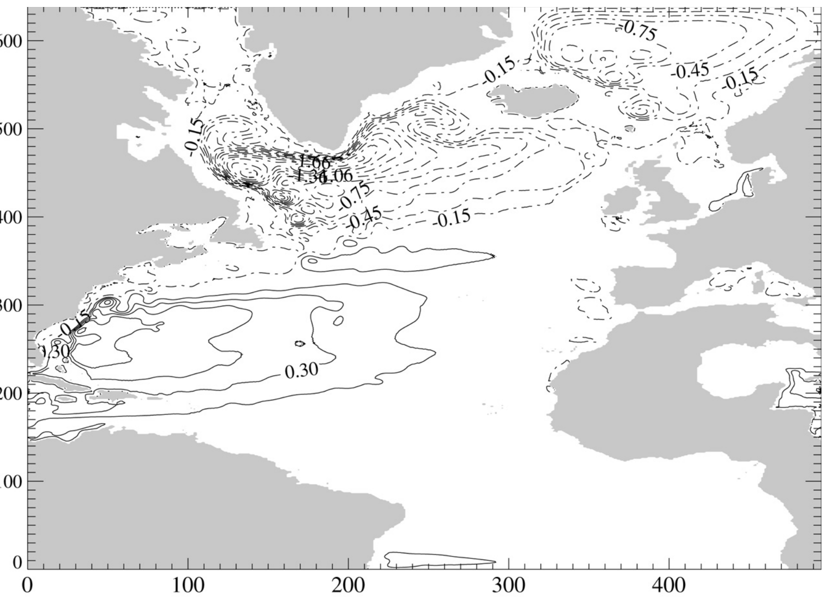

Fig. 2. Contours of simulated sea-surface height in the North Atlantic. The closed contours in the region to the East of the south-eastern US is the subtropical gyre and its western boundary marks the Gulf Stream. The wind-driven circulation is simulated here on a 0.2° grid using a shallow layer representing the upper ocean overlying a deep quiescent layer representing the ocean below the thermocline, and with a fully implicit time stepping scheme. Reanalyzed, time-averaged, steady winds are used.

Courant–Friedrichs–Levy (CFL) number, which is a measure of the stability limit for explicit schemes. While in the traditional OGCMs, neither the advective CFL number nor the viscous CFL number can be greater than unity, they are seen to be vastly greater than unity in the new scheme.

As an example of a case with realistic continental geometry and wind forcing, Fig. 2 shows the SSH in the North Atlantic. In that figure, contours of simulated sea-surface height (in meters) are shown, with continuous contours indicating elevation and dashed–dotted contours indicating depression from reference. The closed contours in the region to the East of the south-eastern US is the subtropical gyre and its western boundary marks the Gulf Stream. The wind-driven circulation is simulated here on a 0.2° grid using a shallow layer representing the upper ocean overlying a deep quiescent layer representing the ocean below the thermocline. Reanalyzed, time-averaged, steady winds are used for the forcing. While Fig. 2 was produced using a time step of a day, corresponding to a gravity wave CFL number of about 184, and an advective CFL number of about 4, this problem was also run using much longer time steps which are discussed further in the section on preconditioning.

Finally, as an example of a case with all ingredients that go into a fully-realistic OGCM, Figs. 3 and 4 show the sea-surface temperature and a meridional cross-section of temperature through the Pacific and Indian oceans, respectively. In this case, the convective adjustment procedure was applied explicitly after the dynamics was fully-implicitly evolved over a time step. Results shown in Figs. 3 and 4 were obtained with a time step about 25 times larger than in traditional OGCMs. However, to run at much longer time steps like a 1000 times longer than in traditional

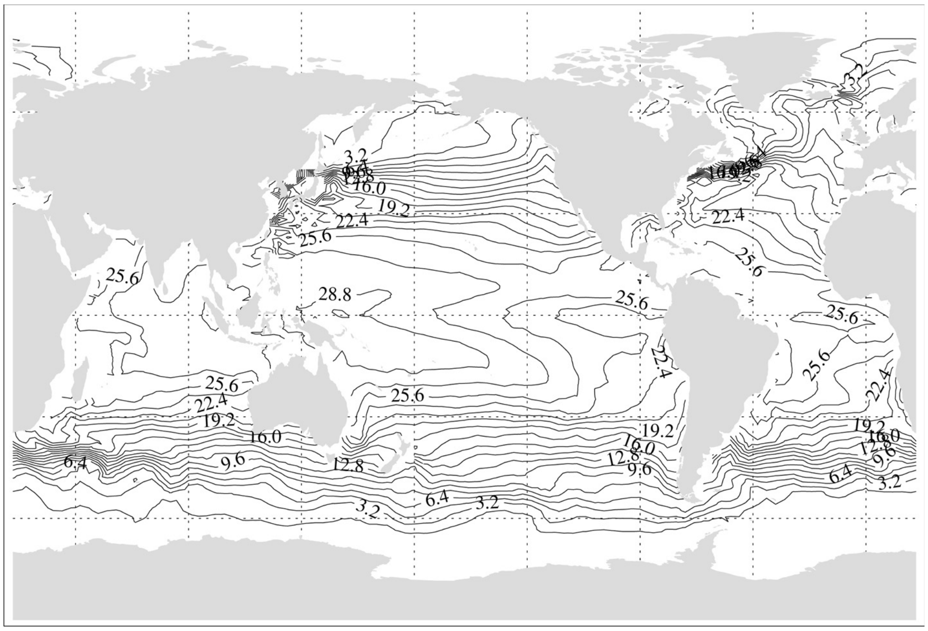

Fig. 3. The sea-surface temperature from iPOP in a realistic configuration using actual topography and continental geometry, initial Levitus temperature and salinity, observed annual averaged wind stress and restoring boundary conditions at the surface for temperature and salinity. The iPOP time step used here is about 25 times that in other OGCMs.

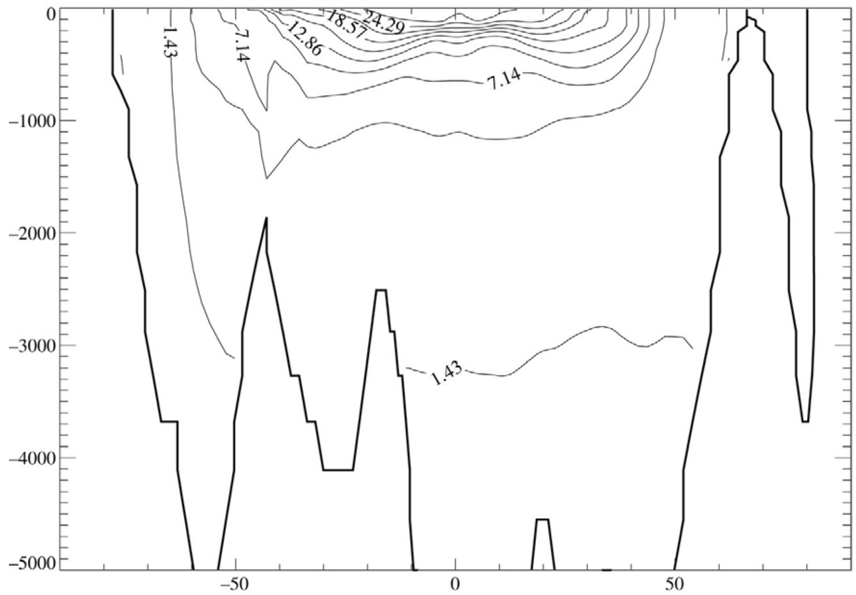

Fig. 4. A meridional cross-section of temperature through the Pacific and Indian oceans (same run as in Fig. 3). The structure of the thermocline (lighter waters of the upper ocean in the equatorial, tropical and subtropical oceans) is properly represented.

OGCMs the treatment of convective adjustment within the Newton–Krylov framework needs to be investigated further. This is because the density field can change significantly over such a time step.

Scalability to massively parallel machines is assured by the Jacobian-free nature of the method. The type of communication required closely mimics that found in POP: predominantly nearest neighbor to update ghost cells and a small amount of collectives for global sums required during the iterations. Thus we expect iPOP will have

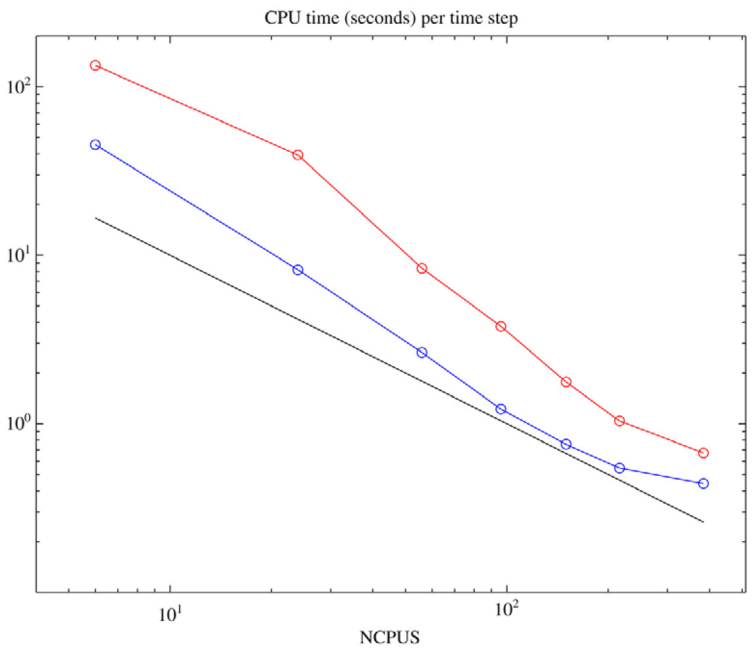

Fig. 5. Parallel performance: perfect scaling is represented by the lowest straight line. Actually realized parallel scaling on an HP Alpha cluster is represented by the middle line and on an SGI Ultrix by the upper line.

scalability similar to POP. We give scalability results for iPOP on leading computational platforms (HP Alpha cluster and SGI Altrix) in Fig. 5. The results show good parallel scaling obtained on both platforms.

3. Remarks on preconditioning

The efficiency of iPOP depends on effective preconditioning. The strongly-coupled, multi-scale dynamics of ocean circulation render the linear systems in iPOP stiff. We are developing preconditioners based on physical processes like gravity waves, planetary geostrophy, and advection–diffusion to solve these systems efficiently. This is a tough problem that requires cross-disciplinary expertise and collaborations. The gravity wave preconditioner is complete and tested, while the latter ones require further work. Here we briefly describe the formulation of the gravity-wave preconditioner.

In the semi-implicit formulation, the integration of the model over one time step may be thought of as solving a *linear* system for $\mathbf{U}^{n+1}$:

$$\mathbf{G}(\mathbf{U}^{n+1}) \equiv \frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\Delta t} - \mathbf{g}(\mathbf{U}^{n+1}, \mathbf{U}^{n}) = 0. \tag{3.1}$$

Here $\mathbf{g}$ depends *linearly* on $\mathbf{U}^{n+1}$; the implicit terms in $\mathbf{g}$ result from linear gravity wave dynamics whereas all the other terms in the governing equations, including the nonlinear terms, are updated explicitly. Consequently, $\mathbf{g}(\mathbf{U}^{n+1}, \mathbf{U}^{n})$ has the form

$$\mathbf{g}(\mathbf{U}^{n+1}, \mathbf{U}^{n}) = \mathcal{P}_0 \mathbf{U}^{n+1} + C(\mathbf{U}^{n}) \tag{3.2}$$

with a constant matrix $\mathcal{P}_0$.

On the other hand, the second-order, fully-implicit, Crank–Nicholson time step would correspond to solving a nonlinear system of the form

$$\mathbf{F}(\mathbf{U}^{n+1}) = \frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\Delta t} - \frac{\mathbf{f}(\mathbf{U}^{n+1}) + \mathbf{f}(\mathbf{U}^{n})}{2} = 0 \tag{3.3}$$

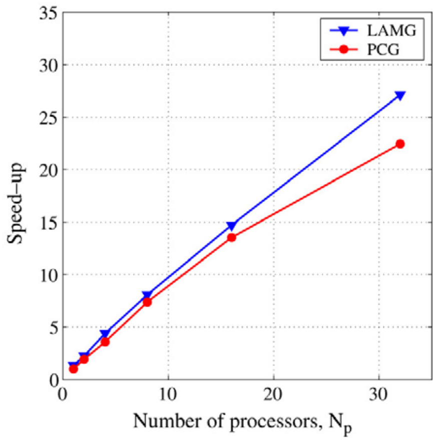

Fig. 6. Speed-up is defined as the time per Newton iteration using N_p processors, divided by the time for a single processor using PCG. Since LAMG is more efficient than PCG, we obtain a speed-up even for the single processor job just by switching from PCG to LAMG.

using the Newton–Krylov iterations. This would involve solving linear systems of the form (2.15), written here for convenience as

$$\mathcal{J}(\mathbf{x}^k)\delta\mathbf{x}^k = -\mathbf{F}(\mathbf{x}^k); \quad \mathbf{x}^\infty = \mathbf{U}^{n+1}. \tag{3.4}$$

Since the semi-implicit scheme **G** effectively allows us to take time steps which exceed the gravity wave CFL restriction, a natural question is how one may use **G** as a gravity wave preconditioned for **F**. Here we introduce such a gravity wave preconditioner as a right preconditioner as in (2.27), where the above equation may be rewritten as

$$\mathcal{J}(\mathbf{x}^k)\mathcal{P}^{-1}\mathcal{P}\delta\mathbf{x}^k = -\mathbf{F}(\mathbf{x}^k); \quad \mathbf{x}^\infty = \mathbf{U}^{n+1}. \tag{3.5}$$

Since **G** can be considered an easy-to-invert approximation to **F**, it is reasonable to assume that the Jacobian of **G** will be a reasonable preconditioner for $\mathcal{J}$.

In light of this, we take

$$\mathcal{P}_{ij} = \frac{\partial \mathbf{G}_i}{\partial \mathbf{x}_j}. \tag{3.6}$$

Since **G** is itself linear in $\mathbf{U}^{n+1}$ it can be written in the form

$$\mathbf{G}(\mathbf{U}^{n+1}) = (\mathcal{I} - \mathcal{A})\mathbf{U}^{n+1} + b(\mathbf{U}^n). \tag{3.7}$$

In this form the Jacobian is evident and thus the preconditioner $\mathcal{P}$ is given by

$$\mathcal{P} = \mathcal{I} - \mathcal{A}. \tag{3.8}$$

Before we computationally examine the use of the above gravity-wave preconditioner, we briefly digress to consider two different methods of inverting the above linear operator. The traditional method for solving the elliptic problem resulting from linear gravity waves is the Preconditioned Conjugate Gradient (PCG) method. For example, this is the method of choice in POP. However, this method has poor scalability and efficiency and its applicability is limited to symmetric operators. Hierarchical solution methods may provide a more robust, efficient, and scalable alternative. To study this, we have interfaced iPOP with the Los Alamos Algebraic Multigrid (LAMG) package to solve the linear elliptic equation that results from the gravity wave preconditioner. Fig. 6 shows speed-up as a function of the number of processors N_p, where speed-up is defined as the time per Newton iteration using N_p processors divided by the time for a single processor using PCG. It shows that not only does algebraic multigrid scale better at larger processor count, but also that its convergence is better even at small processor counts.

Returning to a computational examination of the gravity wave preconditioner, Fig. 7 shows the effectiveness of the gravity wave preconditioner in reducing the number of GMRES iterations required. The lower line corresponds to the

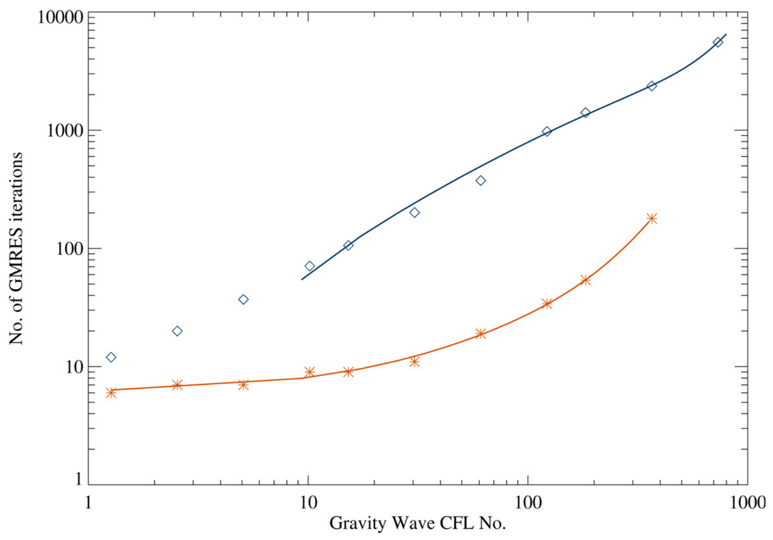

Fig. 7. The impact of the gravity wave preconditioner on the efficiency of the linear solution in the Newton–Krylov methodology. The upper line shows the number of iterations of GMRES required *without* the gravity wave preconditioner and the lower line *with* the gravity wave preconditioner. (GMRES is a costly component of the present scheme. Note, however, that this figure does not account for the cost of the preconditioner.) In particular, considering the range between CFL numbers of 1 and 30, while the number of iterations without the preconditioner is continuously rising with CFL number, it stays almost constant with the preconditioner. The subsequent rise of the lower curve at larger CFL numbers is indicative of the lack of preconditioning of other physical processes.

number of GMRES iterations *with* the preconditioner while the upper line corresponds to computations *without* the preconditioner. In particular, considering the range of CFL numbers between 1 and 30, while the number of iterations without the preconditioner is continuously rising with CFL number, it stays almost constant with the preconditioner. The subsequent rise of the lower curve at larger CFL numbers is indicative of the lack of preconditioning of other physical processes. Note, however, that while GMRES is a costly component of the present scheme, this figure does not account for the cost of the preconditioner. The setup corresponds to the North Atlantic case considered earlier.

4. Conclusion

In the present article we have dealt with computational aspects of time integration in OGCMs. This is complicated by the fact that ocean circulation engenders dynamics on a very wide range of time scales that interact with each other. A simple example of this is the mediation by inertia-gravity waves, a fast process, of the change from one geostrophically balanced (slow) state to another. Justified by scale analysis and limitations of available computational resources, it is presently accepted that the hydrostatic primitive equations (HPE) constitute a reasonable framework to study the dynamics of ocean circulation at scales larger than about ten kilometers. Notwithstanding, such an elimination of even faster dynamics using the Boussinesq and hydrostatic approximations, accurate time integration of HPE is still difficult, particularly on the climate (centennial to millennial) time scales, and at high spatial resolutions.

Present day OGCMs typically use semi-implicit time integration, wherein only the gravity waves are treated implicitly, in conjunction with possible "distorted-physics" techniques for climate studies. Such approaches require ad hoc splitting of the HPE system and are likely to introduce significant distortions. An obvious alternative to this approach is a fully-implicit time integration scheme. It is also well-recognized that the problem with this approach is the need to efficiently solve large nonlinear (algebraic) systems of equations. In fact, historically, solving this nonlinear system has been considered as too difficult and/or too expensive, and this is what has led to the development of the time-split and operator-split approaches presently in use. However, in the intervening decades available computational resources and algorithmic advancements have evolved to render this problem somewhat more tractable. Algorithmically, among these newer approaches are nonlinear multigrid methods and Jacobian-free Newton–Krylov methods. We have successfully implemented, in a version of POP, fully-implicit time-integration schemes based on the latter JFNK methods. In this article, we have mostly described this implementation and then gone on to verify it using a variety of test cases. Considering the unified temporal treatment of the different terms in the governing equations in our approach, which is in contrast to present day OGCMs, we hope to demonstrate possible numerical accuracy advantages of our approach in a future article.

While arbitrarily long time steps are now a possibility in the model, by far the most pressing issue before the model can be competitive at long time steps is the need to efficiently solve possibly poorly-conditioned sparse, but large, linear systems in a scalable fashion. As a first step in this direction, we have implemented the gravity wave preconditioner that works well up to gravity wave CFL numbers of about 60. For the method to be efficient at even longer time steps, other physical processes have to be preconditioned as well. This is an area of ongoing research. Looking further ahead to when the time step in the model can be extended to time scales on which thermodynamical processes relax, the issue of convective adjustment (presently handled explicitly) will have to be revisited.

Acknowledgments

This work was supported by the Department of Energy's Laboratory Directed Research and Development program at Los Alamos National Laboratories.

References

[1] J. Marotzke, Abrupt climate change and thermohaline circulation: Mechanisms and predictability, Proc. Natl. Acad. Sci. USA 97 (2000) 1347–1350.

[2] R.A. Wood, A.B. Keen, J.F.B. Mitchell, J.M. Gregory, Changing spatial structure of the thermohaline circulation in response to atmospheric CO2 forcing in a climate model, Nature 399 (1999) 572–575.

[3] J.R. Scott, J. Marotzke, P.H. Stone, Interhemispheric thermohaline circulation in a coupled box model, J. Phys. Oceanogr. 29 (1999) 351–365.

[4] Intergovernmental Panel on Climate Change (IPCC), Report on Climate Change 2001: The Scientific Basis, Section F.6.

[5] R.D. Smith, M.E. Maltrud, F.O. Bryan, M.W. Hecht, Numerical simulation of the North Atlantic Ocean at $1/10^\circ$, J. Phys. Oceanogr. 30 (2000) 1532–1561.

[6] A.M. Paiva, J.T. Hargrove, E.P. Chassignet, R. Bleck, Turbulent behavior of a fine mesh $(1/12^\circ)$ numerical simulation of the North Atlantic, J. Marine Systems 21 (1) (1999) 307–320.

[7] A.E. Gill, Atmosphere-Ocean Dynamics, Academic Press, San Diego, 1982.

[8] J.K. Dukowicz, R.D. Smith, Implicit free-surface method for the Bryan–Cox–Semtner ocean model, J. Geophys. Res. 99 (C4) (1994) 7991–8014.

[9] T. Johns et al., HadGEM1 model description and analysis of preliminary experiments for the IPCC Fourth Assessment, Hadley Center Technical Note 55 (2005).

[10] W. Cheng, R. Bleck, C. Rooth, Multi-decadal thermohaline variability in an ocean–atmosphere general circulation model, Clim. Dyn. 22 (2004) 573–590.

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

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

[13] K. Bryan, S. Manabe, R.C. Pacanowski, A global ocean–atmosphere climate model. Part II: The oceanic circulation, J. Phys. Oceanogr. 5 (1975) 30–46.

[14] P.D. Killworth, J.M. Smith, A.E. Gill, Speeding up ocean circulation models, Ocean Modell. 56 (1984) 1–4.

[15] R.A. Wood, Time step sensitivity and accelerated spinup of an ocean GCM with a complex mixing scheme, J. Atmos. Ocean. Technol. 15 (2) (1998) 482–495.

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

[17] W. Weijer, H.A. Dijkstra, A bifurcation study of the three-dimensional thermohaline ocean circulation: The double hemispheric case, J. Marine Res. 59 (2001) 599–631.

[18] W. Weijer, H.A. Dijkstra, Multiple oscillatory modes of the global ocean circulation, J. Phys. Oceanogr. 33 (2003) 2197–2213.

[19] H.A. Dijkstra, W. Weijer, Stability of the global ocean circulation: Basic bifurcation diagrams, J. Phys. Oceanogr. 35 (2005) 933–948.

[20] X.C. Cai, D.E. Keyes, V. Venkatakrishnan, Newton–Krylov–Schwarz: An implicit solver for CFD, in: R. Glowinski (Ed.), Domain Decomposition Methods in Sciences and Engineering, John Wiley & Sons Ltd., 1996.

[21] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: A survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.

[22] J. Reisner, V. Mousseau, D. Knoll, Application of the Newton–Krylov method to geophysical flows, Mon. Weather Rev. 129 (2001) 2404–2415.

[23] V.A. Mousseau, D.A. Knoll, M. Reisner, An implicit nonlinearly consistent method for the two-dimensional shallow-water equations with Coriolis force, Mon. Weather Rev. 130 (2002) 2611–2625.

[24] R. Smith, P. Gent, Reference Manual for the Parallel Ocean Program (POP), LAUR-02-2484.

[25] V. Frayssé, L. Giraud, S. Gratton, J. Langou, A set of GMRES routines for real and complex arithmetics on high performance computers, CERFACS Technical Report **TR/PA/03/3**, (2003). Public domain software available at http://www.cerfacs/algor/Softs.

[26] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer Verlag, 1996.

[27] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.