Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 654–679
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A tailored solver for bifurcation analysis of ocean-climate models

Arie de Niet [a], Fred Wubs [a,*], Arjen Terwisscha van Scheltinga [b], Henk A. Dijkstra [b]

[a] Department of Mathematics and Computer Science, University of Groningen, Groningen, The Netherlands
[b] Institute for Marine and Atmospheric Research Utrecht, Department of Physics and Astronomy, Utrecht University, Utrecht, The Netherlands

Received 22 June 2006; received in revised form 17 May 2007; accepted 7 August 2007
Available online 22 August 2007

Abstract

In this paper, we present a new linear system solver for use in a fully-implicit ocean model. The new solver allows to perform bifurcation analysis of relatively high-resolution primitive-equation ocean-climate models. It is based on a block-ILU approach and takes special advantage of the mathematical structure of the governing equations. In implicit models Jacobian matrices have to be constructed. Analytical construction is hard for complicated but more realistic representations of mixing. This is overcome by evaluating the Jacobian in part numerically. The performance of the new implicit ocean model is demonstrated using (i) a high-resolution model of the wind-forced double-gyre flow problem in a (relatively small) midlatitude spherical basin, and (ii) a medium-resolution model of thermohaline and wind-driven flows in an Atlantic size single-hemispheric basin.

1. Introduction

Models of the large-scale ocean circulation form interesting dynamical systems showing transition behavior on many spatial and temporal scales when parameters are varied [8]. A well known large-scale transition is the collapse of the North Atlantic meridional overturning circulation, which occurs when the freshwater flux amplitude increases in the northern part of the basin [3]. Another, less known transition, is that between different separation patterns of western boundary currents, such as those of the Kuroshio and Gulf Stream [18,29]. Dynamical systems theory has also been used to investigate the physics of low-frequency climate variability. In several examples, such as the multidecadal variability in the North Atlantic [35] and the El Niño variability in the equatorial Pacific [25], the origin of the variability could be traced to the occurrence of a Hopf bifurcation.

* Corresponding author. Tel.: +31 (0)50 3633994; fax: +31 (0)50 3633800.
E-mail address: wubs@math.rug.nl (F. Wubs).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.08.006

For the North Atlantic ocean circulation, focus of dynamical systems studies has been to determine the bifurcation behavior of two limiting cases of the 'real' system: (i) the pure wind-driven circulation and (ii) the pure thermohaline circulation. In the first type of studies, the thermodynamic degrees of freedom are limited (e.g. in isothermal situations) and focus is on the transition behavior of solutions of the momentum equations. In the second type of studies, focus has been on the transition behavior of the solutions of the temperature and salinity equations under coupling to (approximately) linear momentum equations. The numerical techniques used are pseudo-arclength continuation (see Section 2.3 below) in which steady states are determined versus parameters with a Newton–Raphson iterative method [19].

For the pure wind-driven ocean circulation, so far only relatively simple models, such as single- or multilayer quasi-geostrophic models and shallow-water models, have been used [34,9,23,31,32,24]. The main limitation in the degree of complexity of these models is the necessary horizontal resolution needed (about 10 km) to adequately capture the bifurcations and the transitions between the different flow regimes. One of the systems studied in detail is the double-gyre flow in a rectangular basin. When the lateral friction is decreased for this flow, it undergoes several local bifurcations before a homoclinic (either Shilnikov or Lorenz type) bifurcation occurs [20,22,33]. For smaller values of the lateral friction, eventually a so-called coherent vortex regime is found, with vortices of about 10 km diameter dominating the flow [30].

For pure thermally (or thermohaline) driven flows, several studies have analyzed the instabilities of single-hemispheric flows [4,16,17,35]. The density driven flow may undergo a Hopf bifurcation when the lateral (thermal) diffusion decreases, leading to either multidecadal or centennial variability. The patterns and time scales of these instabilities are already resolved at a horizontal resolution of about 4°, they can eventually be connected to those of the Atlantic Multidecadal Oscillation [11] and have therefore lead to a mechanistic view of this oscillation. Also the double-hemispheric thermohaline-driven case has been much studied with the strength of the freshwater flux forcing as a control parameter. In the equatorially symmetric case, a pitchfork bifurcation leads to pole-to-pole solutions for the meridional overturning circulation (MOC) and possibilities of transitions between different equilibria [3]. These transitions can eventually be connected to hysteresis behavior of the global ocean circulation as seen in many low-resolution climate models [8].

Several numerical problems have limited the applicability of numerical bifurcation techniques to more sophisticated primitive-equation ocean models:

(i) The inclusion of more realistic representations of mixing processes for heat and salt (than simple constant) of spatially dependent (prescribed) eddy coefficients was difficult because of the analytic evaluation of the Jacobian matrix. State-of-the-art ocean-climate models certainly incorporate mixing along density surfaces and a representation of the effects of the meso-scale eddies [12].
(ii) The representation of convective overturning (so-called 'convective adjustment') needed to obtain stably stratified solutions has given trouble due to convergence problems in the linear system solvers. Here, the vertical mixing depends in an approximately discontinuous way on the vertical density gradient. In Dijkstra et al. [10], we presented the Global Adjustment Procedure (GAP) as an ad-hoc method to obtain stably stratified solutions from unstably stratified ones. The GAP cannot be combined, however, with a pseudo-arclength continuation procedure and is computationally expensive.
(iii) It was difficult to carry out numerical bifurcation analysis for ocean-climate models with more than 10^5 degrees of freedom because of the slow convergence of the iterative solvers of the linear systems (which arise from the Newton–Raphson approach to find steady states). In Weijer et al. [38], we used the Matrix Renumbering Incomplete LU factorization (MRILU [2]) as a preconditioner combined with the Generalized Mimimal Residual method (GMRES [28]) to accomplish this. This solver is a general sparse matrix solver and hence no specific properties of the physical problem at hand are exploited.

Implicit models are useful to understand the origin of low-frequency variability in the combined wind- and buoyancy-driven ocean circulation. In these situations, the MOC has to be computed while narrow western boundary currents have to be resolved. For these advectively dominated flows, a more realistic representation of mixing is needed, convective adjustment is essential and the MRILU preconditioner is inadequate. In these cases, MRILU needs too much time to construct the incomplete factorization simply because it generates too much fill. This fill in the factorization also causes a relatively expensive solution process.

Recently, Nadiga et al. [21] suggested an approach for implicit integration of the POP model using Jacobian-free Newton–Krylov methods. They were able to take much larger time steps than in the explicit version of the POP model, but so far no bifurcation diagrams were computed. In this paper, we describe methodology to tackle all the problems (i–iii) mentioned above by a partial numerical evaluation of the Jacobian matrix and by the development of a tailored linear system solver. The efficient numerical approximation of the Jacobian matrix solves the problems (i) and (ii) above. The solver to tackle problem (iii) above consists of a new block-ILU preconditioner combined with flexible GMRES. This preconditioner exploits the structure of the equations. The block approach leads to a very efficient method, which allows bifurcation analysis of state-of-the-art ocean-climate models with up to a million degrees of freedom.

In Section 2, we present the formulation of the ocean model. The new solver is presented in detail in Section 3 and its numerical properties are demonstrated in Section 4 through two example flows. A short summary and discussion of the results is provided in Section 5.

2. Ocean flows in a single-hemispheric basin

In this section, we describe the ocean model configuration used (Section 2.1), the formulation of the mixing of heat and salt (Section 2.2) and details on the numerical implementation (Section 2.3).

2.1. Model formulation

We consider flows in a spherical sector bounded by longitudes ϕ_w and ϕ_e and by latitudes θ_s and θ_n. The ocean basin has a constant depth D and is bounded vertically by $z = -D$ and a nondeformable ocean-atmosphere boundary $z = 0$. The flows in this domain are forced by a heat flux Q_H (in W m^{-2}), a zonal wind stress field (τ^ϕ, τ^θ) (in Pa) and a virtual salt flux Q_S (in m s^{-1}).

The heat flux Q_H is proportional to the temperature difference between the sea-surface temperature T and a prescribed atmospheric temperature T_S, i.e.

$$Q_H = -\lambda_T(T - T_S) \tag{1}$$

where λ (in W m^{-2} K^{-1}) is a constant exchange coefficient. The virtual salt flux Q_S is similarly formulated as a restoring condition and is given by

$$Q_S = -\lambda_S(S - S_S) \tag{2}$$

where λ_S (in m s^{-1}) is an exchange coefficient. Both wind and buoyancy forcing are distributed as a body forcing over the first (upper) layer of the ocean having a depth H_m.

Temperature and salinity differences in the ocean cause density differences according to

$$\rho = \rho_0(1 - \alpha_T(T - T_0) + \alpha_S(S - S_0)) \tag{3}$$

where α_T and α_S are the volumetric expansion coefficients and T_0, S_0 and ρ_0 are reference quantities. We use the Boussinesq and hydrostatic approximations. With r_0 and Ω being the radius and angular velocity of the Earth, the governing equations for the zonal, meridional and vertical velocity u, v and w and the dynamic pressure p (the hydrostatic part has been subtracted) become

$$\frac{Du}{dt} - uv\tan\theta - 2\Omega v\sin\theta + \frac{1}{\rho_0 r_0\cos\theta}\frac{\partial p}{\partial\phi} = A_V\frac{\partial^2 u}{\partial z^2} + A_H L_u(u,v) + \frac{\tau_0}{\rho_0 H_m}\tau^\phi\mathcal{G}(z) \tag{4a}$$

$$\frac{Dv}{dt} + u^2\tan\theta + 2\Omega u\sin\theta + \frac{1}{\rho_0 r_0}\frac{\partial p}{\partial\theta} = A_V\frac{\partial^2 v}{\partial z^2} + A_H L_v(u,v) + \frac{\tau_0}{\rho_0 H_m}\tau^\theta\mathcal{G}(z) \tag{4b}$$

$$\frac{\partial p}{\partial z} = \rho_0 g(\alpha_T T - \alpha_S S) \tag{4c}$$

$$\frac{\partial w}{\partial z} + \frac{1}{r_0\cos\theta}\left(\frac{\partial u}{\partial\phi} + \frac{\partial(v\cos\theta)}{\partial\theta}\right) = 0 \tag{4d}$$

$$\frac{DT}{\mathrm{d}t} + R_{\mathrm{T}}(T,S) = \frac{(T_{\mathrm{S}} - T)}{\tau_{\mathrm{T}}}\mathcal{G}(z) \tag{4e}$$

$$\frac{DS}{\mathrm{d}t} + R_{\mathrm{S}}(T,S) = \frac{(S_{\mathrm{S}} - S)}{\tau_{\mathrm{S}}}\mathcal{G}(z) \tag{4f}$$

where $\mathcal{G}(z) = \mathcal{H}(z/H_m + 1)$ and $\mathcal{H}$ is a continuous approximation of the Heaviside function. Furthermore, C_p is the constant heat capacity and $\tau_{\mathrm{T}} = \rho_0 C_p H_m/\lambda_{\mathrm{T}}$ and $\tau_{\mathrm{S}} = H_m/\lambda_{\mathrm{S}}$ are the surface adjustment time scales of heat and salt, respectively. In addition,

$$\frac{D}{\mathrm{d}t} = \frac{\partial}{\partial t} + \frac{u}{r_0\cos\theta}\frac{\partial}{\partial\phi} + \frac{v}{r_0}\frac{\partial}{\partial\theta} + w\frac{\partial}{\partial z}$$

$$L_u(u,v) = \nabla_{\mathrm{H}}^2 u + \frac{u}{r_0^2\cos^2\theta} - \frac{2\sin\theta}{r_0^2\cos^2\theta}\frac{\partial v}{\partial\phi}$$

$$L_v(u,v) = \nabla_{\mathrm{H}}^2 v + \frac{v}{r_0^2\cos^2\theta} + \frac{2\sin\theta}{r_0^2\cos^2\theta}\frac{\partial u}{\partial\phi}$$

$$\nabla_{\mathrm{H}}^2 = \frac{1}{r_0^2\cos\theta}\left[\frac{\partial}{\partial\phi}\left(\frac{1}{\cos\theta}\frac{\partial}{\partial\phi}\right) + \frac{\partial}{\partial\theta}\left(\cos\theta\frac{\partial}{\partial\theta}\right)\right]$$

In Eqs. (4a) and (4b) A_{H} and A_{V} are the horizontal and vertical momentum (eddy) viscosity which we will take constant. The terms R_{T} and R_{S} in (4e) and (4f) represent the mixing of heat and salt and will be discussed in the next subsection.

Slip conditions are assumed at the bottom boundary, while at all lateral boundaries no-slip conditions are applied.At all lateral boundaries and the bottom boundary, the heat flux is zero.As the forcing is represented as a body force over the first layer, slip and no-flux conditions apply at the ocean surface. Hence, the boundary conditions are

$$z = -D, 0: \quad \frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = w = \frac{\partial T}{\partial z} = \frac{\partial S}{\partial z} = 0 \tag{5a}$$

$$\phi = \phi_{\mathrm{w}}, \phi_{\mathrm{e}}: \quad u = v = w = \frac{\partial T}{\partial\phi} = \frac{\partial S}{\partial\phi} = 0 \tag{5b}$$

$$\theta = \theta_{\mathrm{s}}, \theta_{\mathrm{n}}: \quad u = v = w = \frac{\partial T}{\partial\theta} = \frac{\partial S}{\partial\theta} = 0 \tag{5c}$$

Parameters that are fixed in the calculations described in Section 4 are the same as in typical large-scale low-resolution ocean general circulation models and their values are listed in Table 1. Other parameter values will be specified in Section 4.

2.2. Tracer mixing representation

To avoid spurious diapycnal mixing in the model of the previous section, so-called neutral physics fluxes have to be specified. Background and details of these fluxes and of the so-called small slope approximation used can be found in Griffies [13]. Expressions are summarized in Section 14.3 of Griffies [13] and in our notation these become

Table 1
Values of fixed parameters used in the numerical calculations in Section 4

$2\Omega = 1.4\times10^{-4}$ [s^{-1}]	$r_0 = 6.4\times10^{6}$ [m]
$C_p = 4.2\times10^{3}$ [J (kg K)$^{-1}$]	$\tau_{\mathrm{T}} = \tau_{\mathrm{S}} = 7.5\times10^{1}$ [days]
$\alpha_{\mathrm{T}} = 1.0\times10^{-4}$ [K^{-1}]	$\alpha_{\mathrm{S}} = 7.6\times10^{-4}$ [–]
$\rho_0 = 1.0\times10^{3}$ [kg m^{-3}]	$A_{\mathrm{V}} = 1.0\times10^{-3}$ [$\mathrm{m}^2\,\mathrm{s}^{-1}$]

$$R_T(T,S) = \nabla_H \cdot \left(K_H \nabla_H T + \eta_M (K_H - \eta_G \kappa) \mathbf{S} \cdot \frac{\partial T}{\partial z} \right) + \left(\frac{\partial}{\partial z} \left[\eta_M (K_H + \eta_G \kappa) \mathbf{S} \cdot \nabla_H T + \eta_M K_H \mathbf{S} \cdot \mathbf{S} \frac{\partial T}{\partial z} \right] \right) + \frac{\partial}{\partial z} \left(K_V \frac{\partial T}{\partial z} \right) \tag{6a}$$

$$R_S(T,S) = \nabla_H \cdot \left(K_H \nabla_H S + \eta_M (K_H - \eta_G \kappa) \mathbf{S} \cdot \frac{\partial S}{\partial z} \right) + \left(\frac{\partial}{\partial z} \left[\eta_M (K_H + \eta_G \kappa) \mathbf{S} \cdot \nabla_H S + \eta_M K_H \mathbf{S} \cdot \mathbf{S} \frac{\partial S}{\partial z} \right] \right) + \frac{\partial}{\partial z} \left(K_V V \frac{\partial S}{\partial z} \right) \tag{6b}$$

where K_H is the neutral diffusivity and κ is the [12] skew diffusive mixing coefficient. In addition, the neutral slope vector $\mathbf{S}$ is given by

$$\mathbf{S} = \frac{-\alpha_T \nabla_H T + \alpha_S \nabla_H S}{-\alpha_T \frac{\partial T}{\partial z} + \alpha_S \frac{\partial S}{\partial z}} \tag{7}$$

Note that we have introduced two homotopy parameters η_M and η_G to be able to continue smoothly between constant horizontal diffusivity ($\eta_M = 0$) and neutral mixing ($\eta_M = 1$) and from no meso-scale eddy representation ($\eta_G = 0$) to the [12] representation ($\eta_G = 1$); the latter is referred below as GM-mixing.

We can only apply the representation above when the stratification is stable and when the slope of the isopycnals is small. To limit the slopes, we apply a similar procedure as in Danabasoglu and McWilliams [6] by defining α_m as a maximum permissible slope angle and then multiplying η_M by the function $f(\frac{\partial \rho}{\partial z})$, where

$$f(x) = 3\left(\frac{x+\xi}{\delta}\right)^2 - 2\left(\frac{x+\xi}{\delta}\right)^3, \quad -\xi \leqslant x \leqslant -\xi + \delta \tag{8a}$$

$$f(x) = 1, \quad -\xi + \delta \leqslant x \leqslant -\delta \tag{8b}$$

$$f(x) = 1 - 3\left(\frac{x+\delta}{\delta}\right)^2 + 2\left(\frac{x+\delta}{\delta}\right)^3, \quad -\delta \leqslant x \leqslant 0 \tag{8c}$$

$$f(x) = 0, \quad x \leqslant -\xi \text{ or } x > 0 \tag{8d}$$

where

$$\xi = \frac{|\nabla_H \rho|}{\tan \alpha_m} \tag{9}$$

and $\delta = 0.05\xi$. In this way, the mixing coefficients are smoothly tapered to zero in regions where the slope becomes too large or where the stratification is unstable. In the results below, we will use different α_m to investigate the sensitivity to this parameter.

The coefficient K_V in (6b) is the vertical mixing coefficient. Whereas in many ocean-climate models, usually K_V is taken constant (or spatially prescribed), too high values (in the order of $10^{-4}\,\mathrm{m^2\,s^{-1}}$) were needed to obtain a realistic strength of the meridional overturning. Mixing in a stratified ocean, however, requires a transfer from kinetic to potential energy. The available energy for mixing, say e, is a topic of current research [39] and it is supplied by the wind, the tides and buoyancy forcing. An overview of the processes responsible for the spatial pattern of e is, for example, given in Huang [15]. According to a production–dissipation equilibrium of the turbulent kinetic energy [36], the vertical mixing coefficient is defined as

$$K_V = K_V^0 + \frac{\Gamma e}{N_b^2}; \quad N_b^2 = -\frac{g}{\rho_0} \frac{\partial \rho}{\partial z} \tag{10}$$

where $\Gamma = 0.2$ is the mixing efficiency and N_b is the buoyancy frequency. A background value K_V^0 can be attributed to internal wave breaking and it is taken to be constant.

In case of an unstable stratification, $N_b^2 < 0$, additional mixing occurs through convective overturning. We can take this mixing into account through an additional mixing coefficient $K_V^c \gg K_V^0$ by formulating K_V as

$$K_V = K_V^0 + \mathcal{F}(N_b^2) K_V^c + \eta_V \mathcal{F}(-N_b^2) \frac{\Gamma e}{N_b^2} \tag{11}$$

where η_V is another homotopy parameter which can be used to study the situations with only constant mixing coefficient ($\eta_V = 0$) and variable mixing ($\eta_V = 1$). Furthermore, $\mathcal{F}$ is a mixing profile function which we take as

$$\mathcal{F}(x) = \max\{\tanh(-x^3), 0\} \tag{12}$$

such that additional convective mixing is generated smoothly as soon as $N_b^2 < 0$.

2.3. Numerical implementation

The equations are discretized in space using a second-order accurate control volume discretization method on a staggered Arakawa B-grid in the horizontal with $i = 1,\ldots,N$, $j = 1,\ldots,M$, and a C-grid in the vertical $k = 1,\ldots,L$; this combination is called a Lorenz grid (see Fig. 1). The spatially discretized model equations can be written in the form

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{F}(\mathbf{u}) = \mathbf{L}(\mathbf{u}) + \mathbf{N}(\mathbf{u},\mathbf{u}) \tag{13}$$

where the vector $\mathbf{u}$ contains the unknowns (u, v, w, p, T, S) at each grid point and hence has dimension $d = 6 \times N \times M \times L$. The operators $\mathbf{M}$ and $\mathbf{L}$ are linear and $\mathbf{N}$ represents the nonlinear terms in the equations.

Steady state solutions lead to a set of nonlinear algebraic equations of the form

$$\mathbf{F}(\mathbf{u},\mathbf{p}) = 0 \tag{14}$$

Here the parameter dependence of the equations is made explicit through the p-dimensional vector of parameters $\mathbf{p}$ and hence $\mathbf{F}$ is a nonlinear mapping from $\mathbf{R}^{d+p} \to \mathbf{R}^d$. To determine branches of steady solutions of Eq. (14) as one of the parameters, say μ, is varied, the pseudo-arclength method [19] is used. The branches $(\mathbf{u}(s), \mu(s))$ are parameterized by an 'arclength' parameter s. An additional equation is obtained by 'normalizing' the tangent

$$\dot{\mathbf{u}}_0^{\mathrm{T}}(\mathbf{u} - \mathbf{u}_0) + \dot{\mu}_0(\mu - \mu_0) - \Delta s = 0 \tag{15}$$

where $(\mathbf{u}_0, \mu_0)$ is an analytically known starting solution or a previously computed point on a particular branch and Δs is the step-length. Euler–Newton continuation is used to solve the system of Eqs. (14) and (15). The $(d+1)\times(d+1)$ Jacobian matrix $\mathcal{J}(s)$ of (14) and (15) along a branch is given by

$$\mathcal{J}(s) = \begin{bmatrix} \Phi & \mathbf{F}_\mu \\ \dot{\mathbf{u}}_0^{\mathrm{T}} & \dot{\mu}_0 \end{bmatrix} \tag{16}$$

where Φ is the matrix of derivatives of $\mathbf{F}$ to $\mathbf{u}$ and $\mathbf{F}_\mu$ the derivative to the parameter μ.

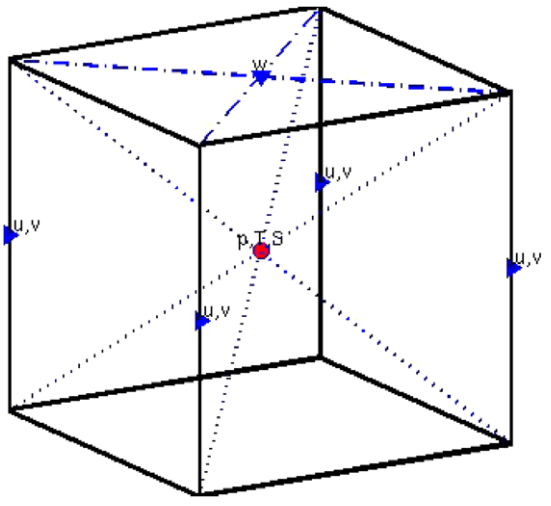

Fig. 1. Grid cell showing positioning of the variables.

In previous formulations of this ocean model [38], the entries in the Jacobian matrix were computed analytically and directly from the discretized equations. As the mixing formulations (6b) and (11) are quite complicated, it turned out to be advantageous to use an additional numerical evaluation of the Jacobian matrix. In Coleman et al. [5], the problem of how to estimate the Jacobian matrix $\mathbf{J}$ of a general mapping $\mathbf{F}$: $\mathbf{R}^n \to \mathbf{R}^m$ using the least number of function evaluations is discussed. The jth column of $\mathbf{J}$ is approximated by

$$\mathbf{J}_j = \frac{\mathbf{F}(\ldots, x_j + \epsilon, \ldots) - \mathbf{F}(\ldots, x_j, \ldots)}{\epsilon} \tag{17}$$

for small ϵ. In the results in Section 4, we tested values of $\epsilon = 10^{-7} - 10^{-5}$ and found that the results were not sensitive over this range; a value of $\epsilon = 10^{-6}$ was chosen. In general, we need to calculate m of such differences to approximate the Jacobian. For sparse Jacobian matrices, a group of columns can be determined such that no two columns in this group have a nonzero element in the same row position. This is because, for large-scale problems, the elements of $\mathbf{F}$ are dependent on a limited number of variables x_j. Let C be such a group of columns and let $\mathbf{d} \in \mathbf{R}^m$ be a vector with $d_j = \epsilon$ if $\mathbf{J}_j \in C$ and $d_j = 0$ otherwise. Then the difference

$$\mathbf{J}_C = \frac{\mathbf{F}(\mathbf{x} + \mathbf{d}) - \mathbf{F}(\mathbf{x})}{\epsilon} \tag{18}$$

contains the non-zero elements of the columns belonging to C. By partitioning the columns of the Jacobian matrix in this way, we can more efficiently evaluate the Jacobian since the number of differences we need to calculate are strongly reduced. For the implementation of this approach, we used subroutines provided by Coleman et al. [5]. These subroutines consists of (i) routines that determine the groups C, such that the number of groups is minimal; and (ii) routines that determine for a given group C the vector $\mathbf{d}$ and extract the columns $\mathbf{J}_j$ from $\mathbf{J}_C$. These routines require the sparsity pattern of the Jacobian matrix, i.e. the variables on which $\mathbf{F}$ depends, and a routine that evaluates $\mathbf{F}(\mathbf{x})$.

During one Newton iteration, linear systems of the form

$$\mathcal{J}\begin{pmatrix} \Delta\mathbf{u} \\ \Delta\mu \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ r_{d+1} \end{pmatrix} \tag{19}$$

have to be solved where $\Delta\mathbf{u}$ and $\Delta\mu$ are updates during the Newton process and $\mathbf{r}$ and r_{d+1} derive from the left-hand sides of (14) and (15), respectively. For these linear systems of equations the special solver, as presented in the next section, is required.

3. The tailored solver

In each step of the Newton process we have to solve one or two linear systems of equations with the matrix Φ in (16), which is by far the most expensive part of the computation. The size of each linear system is such that direct solvers are not applicable and we have to use iterative methods. Because the matrix is non-symmetric and has indefinite parts, so it has a high condition number, any Krylov subspace method directly applied to the equations will at best converge very slowly, so we have to accelerate the method with a preconditioner. In Weijer et al. [38] we used matrix renumbering incomplete LU (MRILU [2]), a black box preconditioner that is comparable to algebraic multigrid. MRILU is able to build an efficient preconditioner, but the construction takes too much CPU time and it requires much memory because a high amount of fill is needed for an accurate factorization, i.e. a factorization that keeps the number of iterations in the solution process low.

In order to get a better preconditioner, one that is cheaper to construct and requires less memory, we have to look more closely to the discrete ocean model equations and exploit their structure.

3.1. The structure of the equations

The equation $\Phi\mathbf{u} = \mathbf{b}$ has the following structure

$$\begin{bmatrix} \mathbf{A}_{uv} & \mathbf{E}_{uv} & \mathbf{G}_{uv} & 0 \\ 0 & 0 & \mathbf{G}_w & \mathbf{B}_{\mathrm{ST}} \\ \mathbf{D}_{uv} & \mathbf{D}_w & 0 & 0 \\ \mathbf{B}_{uv} & \mathbf{B}_w & 0 & \mathbf{A}_{\mathrm{ST}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{uv} \\ \mathbf{u}_w \\ \mathbf{u}_p \\ \mathbf{u}_{\mathrm{ST}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{uv} \\ \mathbf{b}_w \\ \mathbf{b}_p \\ \mathbf{b}_{\mathrm{ST}} \end{bmatrix} \tag{20}$$

The four rows in (20) are associated to the equations for conservation of momentum in longitudinal and latitudinal direction (Eqs. (4a) and (4b)), hydrostatic equilibrium (Eq. (4c)), conservation of mass (Eq. (4d)) and conservation of heat and salt (Eqs. (4e) and (4f)), respectively.

Each $\mathbf{G}_*$ represents the discrete gradient operator and each $\mathbf{D}_*$ the discrete divergence operator. Here, $\mathbf{G}_w$ is the transpose of $\mathbf{D}_w$, but, due to the spherical coordinate system, $\mathbf{G}_{uv}$ is not the transpose of $\mathbf{D}_{uv}$, which is often the case in the plane and important for numerical stability. However, here the equivalent property holds that there exist diagonal matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ such that $\mathbf{G}_{uv}^{\mathrm{T}} = \mathbf{\Lambda}_1 \mathbf{D}_{uv} \mathbf{\Lambda}_2$ The matrices $\mathbf{A}_{\mathrm{ST}}$ and $\mathbf{A}_{uv}$ are of convection–diffusion type, and $\mathbf{A}_{uv}$ includes the Coriolis terms. The matrices $\mathbf{B}_*$ represent the couplings between the dynamic and thermodynamic variables. The matrix $\mathbf{E}_{uv}$ represents the coupling to w in the material derivative in Eqs. (4a) and (4b). As these terms are small with respect to Coriolis terms we will ignore the block in the design of the preconditioner. It is treated as another zero block which simplifies the calculations. As one can see, next to $\mathbf{E}_{uv}$ there are six zero blocks, which means that severable variables are not coupled. All the submatrices are structured and sparse and there are only $\mathcal{O}(d)$ nonzeros in each matrix.

These zero blocks are currently not exploited enough in MRILU and therefore too much fill is generated. We will try to overcome this shortcoming of MRILU and exploit the empty blocks in the design of a new preconditioner. We will use clustering for u, v, because the Eqs. (4a) and (4b) have the same structure. As the same holds for (4e) and (4f), also the variables T, S are clustered.

At the start of the section we mentioned that there are $d = 6 \times N \times M \times L$ unknowns. However, due to the staggered grid and the boundary treatment some of the unknowns are not part of the computational domain. These unknowns are already excluded from the system (20). Immediate consequence is that the dimensions of the vectors $\mathbf{u}_*$ in (20) are not the same. Let d_* be the dimension of each $\mathbf{u}_*$, then it holds that $d_{uv} = d_{\mathrm{ST}} = 2 \times N \times M \times L$ and $d_w < d_p \leqslant N \times M \times L$. The inequality $d_w < d_p$ holds because we have a staggered grid in the vertical direction. Per water column, there is one internal pressure node more than there are vertical velocity nodes.

3.2. Rewriting the equations to ease the solution process

Before we describe the preconditioner for (20) we first rewrite the system in three subsequent steps: (i) we split the pressure in a depth-averaged pressure and a component perpendicular to that, (ii) transform the continuity equation and (iii) rearrange columns and rows. The aim of all these operations is to get a system that is a better starting point for the development of a preconditioner.

3.2.1. Step 1: depth-average the pressure

In the continuous formulation the hydrostatic pressure equation, see Eq. (4c), contains the term p_z. Note that we can decompose the pressure in $p = \bar{p} + \tilde{p}$, where $\bar{p}(\phi, \theta) = \int_{-h}^{0} p(\phi, \theta, z) \mathrm{d}z$ is a depth-averaged pressure. The averaged pressure vanishes in the hydrostatic pressure equation, because it is in the kernel of the operator $\frac{\partial}{\partial z}$. From this it follows that $\int_{-h}^{0} \tilde{p}(\phi, \theta, z) \mathrm{d}z = 0$. The consequence of this decomposition is that the problem $f_z = g$ in the space of all functions with $\int_{-h}^{0} f \, \mathrm{d}z = 0$, is now uniquely solvable.

We can do something similar on the discrete equations. As we pointed out in Section 3.1 the matrix $\mathbf{G}_w$ has dimensions $d_w \times d_p$ and because $d_w < d_p$ it is not square. The matrix has full rank, so the kernel has dimension $d_p - d_w$. Because of the structured grid and the fact that $\mathbf{G}_w$ is a discrete gradient operator, it is quite easy to construct an orthonormal basis for this kernel. Given such a basis we can build the operator $\mathbf{M}_p$, who's columns are precisely these orthonormal basis vectors. Consequently $\mathbf{G}_w \mathbf{M}_p = 0$ and $\mathbf{M}_p^{\mathrm{T}} \mathbf{M}_p = \bar{\mathbf{I}}_p$. The operator $\mathbf{M}_p^{\mathrm{T}}$ acts on the pressure space and is up to a factor a depth-averaging of the pressure field.

Let $\bar{\mathbf{u}}_p \doteq \mathbf{M}_p^{\mathrm{T}}\mathbf{u}_p$ be the depth-averaged pressure. Furthermore, we define $\hat{\mathbf{u}}_p \doteq \mathbf{u}_p - \mathbf{M}_p\bar{\mathbf{u}}_p = \left(\mathbf{I}_p - \mathbf{M}_p\mathbf{M}_p^{\mathrm{T}}\right)\mathbf{u}_p$, a vector that lives in the same space as $\mathbf{u}_p$ but that satisfies the extra condition $\mathbf{M}_p^{\mathrm{T}}\hat{\mathbf{u}}_p = 0$. We can decompose the pressure in the following way

$$\mathbf{u}_p = \hat{\mathbf{u}}_p + \mathbf{M}_p\bar{\mathbf{u}}_p$$

Next, $\mathbf{u}_p$ is substituted into Eq. (20) and the equation that restricts $\hat{\mathbf{u}}_p$ in the pressure space $\left(\mathbf{M}_p^{\mathrm{T}}\hat{\mathbf{u}}_p = 0\right)$ is added to the system. In this way the number of both unknowns and equations is increased with the same amount.

The advantage of the splitting of the pressure space is that in the discrete hydrostatic pressure equation we have a term $\mathbf{G}_w\,\mathbf{u}_p$ that now becomes $\mathbf{G}_w\hat{\mathbf{u}}_p + \mathbf{G}_w\mathbf{M}_p\bar{\mathbf{u}}_p = \mathbf{G}_w\hat{\mathbf{u}}_p$ and the system $\mathbf{G}_w\hat{\mathbf{u}}_p = b$ is now uniquely solvable for any right-hand side b.

3.2.2. Step 2: transform the continuity equation

Since $w_z = f$ with $w(0) = w(-h) = 0$ is overdetermined, there is no solution if not $\int_{-h}^{0} f\,\mathrm{d}z = 0$. In our case, the latter is obtained from the depth-integration of the continuity equation, see Eq. (4d), where w drops out. This gives an equation for the horizontal velocities. With the horizontal velocities satisfying this equation, one can solve w uniquely and leave out even one of the boundary conditions or even more rigorously, cut the z-interval in two parts and let the upper part be determined by the boundary condition at the surface and the lower part by that at the bottom.

We can perform this depth-integration as well on the discrete equations. In analogy to Step 1, we use the operator $\mathbf{M}_p$, who's rows form an orthonormal basis for the kernel of $\mathbf{G}_w = \mathbf{D}_w^{\mathrm{T}}$. So each row of $\mathbf{M}_p^{\mathrm{T}}$ is a left-singular vector of $\mathbf{D}_w$. Consequently $\mathbf{M}_p^{\mathrm{T}}\mathbf{D}_w = 0$ and $\mathbf{M}_p^{\mathrm{T}}\mathbf{M}_p = \bar{\mathbf{I}}_p$. With $\mathbf{M}_p^{\mathrm{T}}$ we define the following transformation matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_w & & 0 \\ & \mathbf{M}_p^{\mathrm{T}} & \end{bmatrix} \tag{21}$$

With this matrix an ordering of $\mathbf{u}_p$ is associated such that by the identity part in every vertical column in the grid all but one elements of $\mathbf{u}_p$ are picked. Then the matrix $\mathbf{T}$ is square and invertible. Moreover the rows of $\mathbf{D}_w$ picked by the same identity part will constitute an invertible matrix, this is the discrete counterpart of the continuous case discussed above.

Therefore, we define

$$\widetilde{\mathbf{D}}_{uv} = [\,\mathbf{I}_w \quad 0\,]\mathbf{D}_{uv}, \quad \mathbf{A}_w = [\,\mathbf{I}_w \quad 0\,]\mathbf{D}_w \quad \text{and} \quad \tilde{\mathbf{b}}_p = [\,\mathbf{I}_w \quad 0\,]\mathbf{b}_p \tag{22}$$

and hence $\mathbf{A}_w$ is the "square part" of $\mathbf{D}_w$ which contains only the first d_w rows of $\mathbf{D}_w$. The matrices $\widetilde{\mathbf{D}}_{uv}$ and $\tilde{\mathbf{b}}_p$ are the corresponding parts of $\mathbf{D}_{uv}$ and $\mathbf{b}_p$, respectively. As the transformation matrix $\mathbf{T}$ is invertible, we can apply it to the continuity equation: $\mathbf{T}\mathbf{D}_{uv}\mathbf{u}_{uv} + \mathbf{T}\mathbf{D}_w\mathbf{u}_w = \mathbf{T}\mathbf{b}_p$, which becomes

$$\begin{bmatrix} \widetilde{\mathbf{D}}_{uv} \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} \end{bmatrix}\mathbf{u}_{uv} + \begin{bmatrix} \mathbf{A}_w \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_w \end{bmatrix}\mathbf{u}_w = \begin{bmatrix} \tilde{\mathbf{b}}_p \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{b}_p \end{bmatrix}$$

Now the matrix product $\mathbf{M}_p^{\mathrm{T}}\mathbf{D}_w$ is the depth averaging of the discrete vertical divergence operator. It is similar to $(\mathbf{G}_w\mathbf{M}_p)^{\mathrm{T}}$, which was equal to *0*; the same holds here and hence $\mathbf{M}_p^{\mathrm{T}}\mathbf{D}_w = 0$. The transformation gives an extra zero block.

So by Steps 1 and 2 we arrive at the following system

$$\begin{bmatrix} \mathbf{A}_{uv} & 0 & \mathbf{G}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p & 0 \\ 0 & 0 & \mathbf{G}_w & 0 & \mathbf{B}_{\mathrm{ST}} \\ \widetilde{\mathbf{D}}_{uv} & \mathbf{A}_w & 0 & 0 & 0 \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{M}_p^{\mathrm{T}} & 0 & 0 \\ \mathbf{B}_{uv} & \mathbf{B}_w & 0 & 0 & \mathbf{A}_{\mathrm{ST}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{uv} \\ \mathbf{u}_w \\ \hat{\mathbf{u}}_p \\ \bar{\mathbf{u}}_p \\ \mathbf{u}_{\mathrm{ST}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{uv} \\ \mathbf{b}_w \\ \tilde{\mathbf{b}}_p \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{b}_p \\ 0 \\ \mathbf{b}_{\mathrm{ST}} \end{bmatrix}$$

3.2.3. Step 3: rearrange columns and rows

As a subsequent step, we perform a non-symmetric rearrangement of columns (variables) and rows (equations) in order to bring the matrix to a lower block-triangular matrix as much as possible. A lower block-triangular matrix shows a way to solve the system in subsequent steps with smaller systems. Numbering the rows from 1 to 6 and columns from 1 to 5, we use for the rows the permutation $(2,5,1,4,3,6)$ and for the columns $(3,1,4,2,5)$. This rearrangement gives

$$\begin{bmatrix} \mathbf{G}_w & 0 & 0 & 0 & \mathbf{B}_{\mathrm{ST}} \\ \mathbf{M}_p^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{G}_{uv} & \mathbf{A}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p & 0 & 0 \\ 0 & \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 & 0 & 0 \\ 0 & \tilde{\mathbf{D}}_{uv} & 0 & \mathbf{A}_w & 0 \\ 0 & \mathbf{B}_{uv} & 0 & \mathbf{B}_w & \mathbf{A}_{\mathrm{ST}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_p \\ \mathbf{u}_{uv} \\ \bar{\mathbf{u}}_p \\ \mathbf{u}_w \\ \mathbf{u}_{\mathrm{ST}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_w \\ 0 \\ \mathbf{b}_{uv} \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{b}_p \\ \tilde{\mathbf{b}}_p \\ \mathbf{b}_{\mathrm{ST}} \end{bmatrix}$$

In the first two rows, we now have in the first column the matrices $\mathbf{G}_w$ and $\mathbf{M}_p^{\mathrm{T}}$. Because of the special relation between these two matrices (remember that $\mathbf{G}_w\mathbf{M}_p = 0$) they form together a square and invertible matrix. To make this structure more explicit we define

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{G}_w \\ \mathbf{M}_p^{\mathrm{T}} \end{bmatrix}, \quad \widetilde{\mathbf{B}}_{\mathrm{ST}} = \begin{bmatrix} \mathbf{B}_{\mathrm{ST}} \\ 0 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{b}}_w = \begin{bmatrix} \mathbf{b}_w \\ 0 \end{bmatrix}$$

With the introduction of these matrices the matrix system above becomes

$$\begin{bmatrix} \mathbf{A}_p & 0 & 0 & 0 & \widetilde{\mathbf{B}}_{\mathrm{ST}} \\ \mathbf{G}_{uv} & \mathbf{A}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p & 0 & 0 \\ 0 & \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 & 0 & 0 \\ 0 & \widetilde{\mathbf{D}}_{uv} & 0 & \mathbf{A}_w & 0 \\ 0 & \mathbf{B}_{uv} & 0 & \mathbf{B}_w & \mathbf{A}_{\mathrm{ST}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_p \\ \mathbf{u}_{uv} \\ \bar{\mathbf{u}}_p \\ \mathbf{u}_w \\ \mathbf{u}_{\mathrm{ST}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{b}}_w \\ \mathbf{b}_{uv} \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{b}_p \\ \tilde{\mathbf{b}}_p \\ \mathbf{b}_{\mathrm{ST}} \end{bmatrix} \tag{23}$$

We will call the matrix in this equation $\widetilde{\Phi}$. Observe that apart from $\widetilde{\mathbf{B}}_{\mathrm{ST}}$ it is of block lower triangular form.

Note that so far we did not essentially change the equation; Eqs. (20) and (23) have the same solution (knowing that $\mathbf{u}_p = \hat{\mathbf{u}}_p + \mathbf{M}_p\bar{\mathbf{u}}_p$), although they look quite different. There are two advantages of the new form. The first is that in a block LU factorization of the matrix, apart from the last diagonal block, the L matrix is equal to the block lower triangular part of $\widetilde{\Phi}$. The second advantage of (23) is that the amount of zeros on the diagonal has reduced dramatically. The matrix in (23) has only one zero block where the one in (20) has two. Furthermore, the zero block in the rewritten system is rather small because it is part of the depth-averaged continuity equation. Hence, the rewritten system is a more convenient starting point for the development of a preconditioner than (20).

3.3. The preconditioner and its implementation

If we drop the block $\widetilde{\mathbf{B}}_{\mathrm{ST}}$ in system (23), the matrix becomes lower block-triangular. This would be a natural choice for a preconditioner. It is easy to solve a system with that preconditioner: we only have to solve systems with the diagonal blocks consecutively. Unfortunately the performance of this preconditioner deteriorates when the coupling between dynamical en thermodynamical variables becomes stronger. In that case the block $\widetilde{\mathbf{B}}_{\mathrm{ST}}$ becomes important. This is certainly the case in the flows of interest. Hence, we have to use a more sophisticated preconditioner.

First of all one should note that the second and third row of (23) contain the saddle point matrix

$$\mathbf{K} = \begin{bmatrix} \mathbf{A}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 \end{bmatrix} \tag{24}$$

An LU factorization of $\mathbf{K}$ is given by $\widehat{\mathbf{K}}\widehat{\mathbf{U}}$, where

$$\widehat{\mathbf{K}} = \begin{bmatrix} \mathbf{A}_{uv} & 0 \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & \mathbf{C}_{uv} \end{bmatrix}, \quad \widehat{\mathbf{U}} = \begin{bmatrix} \mathbf{I}_{uv} & \mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{M}_p \\ 0 & \bar{\mathbf{I}}_p \end{bmatrix}$$

Here, the Schur complement $\mathbf{C}_{uv}$ is given by

$$\mathbf{C}_{uv} = -\mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv}\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{M}_p$$

Now, we approximate $\mathbf{K}$ by $\widehat{\mathbf{K}}$ only. Note that if $\widehat{\mathbf{K}}$ is used as a preconditioner for $\mathbf{K}$ in a Krylov subspace method, then the convergence is determined by the eigenvalues of the preconditioned matrix $\widehat{\mathbf{K}}^{-1}\mathbf{K} = \widehat{\mathbf{U}}$, which are all simply 1 and their multiplicity is at most 2. Hence the method will converge in two steps.

We replace the saddle point problem $\mathbf{K}$ with the approximation $\widehat{\mathbf{K}}$ in the matrix of system (23). Now we can perform Gaussian elimination with the diagonal blocks. This will result in the following factorization

$$\begin{bmatrix} \mathbf{A}_p & 0 & 0 & 0 & 0 \\ \mathbf{G}_{uv} & \mathbf{A}_{uv} & 0 & 0 & 0 \\ 0 & \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & \mathbf{C}_{uv} & 0 & 0 \\ 0 & \widetilde{\mathbf{D}}_{uv} & 0 & \mathbf{A}_w & 0 \\ 0 & \mathbf{B}_{uv} & 0 & \mathbf{B}_w & \mathbf{S}_{\mathrm{ST}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_p & 0 & 0 & 0 & \mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & \mathbf{I}_{uv} & 0 & 0 & -\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & 0 & \bar{\mathbf{I}}_p & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_w & \mathbf{A}_w^{-1}\widetilde{\mathbf{D}}_{uv}\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & 0 & 0 & 0 & \mathbf{I}_{\mathrm{ST}} \end{bmatrix}$$

where

$$\mathbf{S}_{\mathrm{ST}} = \mathbf{A}_{\mathrm{ST}} + (\mathbf{B}_{uv} - \mathbf{B}_w\mathbf{A}_w^{-1}\widetilde{\mathbf{D}}_{uv})\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \tag{25}$$

Surprisingly except in $\widehat{\mathbf{K}}$, the matrix $\mathbf{C}_{uv}$ does not occur anywhere in the factorization. This is due to the zero blocks in the column of $\mathbf{C}_{uv}$. We do not like to construct $\mathbf{C}_{uv}$ explicitly, because it is probably a full matrix and expensive to construct. Fortunately we can get rid of $\widehat{\mathbf{K}}$ for the same reason we introduced it. The matrix $\mathbf{K}$ is a good approximation of $\widehat{\mathbf{K}}$ so we put back the original saddle point problem in the matrix. This gives us the following block-ILU factorization as preconditioner

$$\widehat{\Phi} = \mathbf{L}_\Phi\mathbf{U}_\Phi \tag{26}$$

with

$$\mathbf{L}_\Phi = \begin{bmatrix} \mathbf{A}_p & 0 & 0 & 0 & 0 \\ \mathbf{G}_{uv} & \mathbf{A}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p & 0 & 0 \\ 0 & \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 & 0 & 0 \\ 0 & \widetilde{\mathbf{D}}_{uv} & 0 & \mathbf{A}_w & 0 \\ 0 & \mathbf{B}_{uv} & 0 & \mathbf{B}_w & \mathbf{S}_{\mathrm{ST}} \end{bmatrix} \tag{27}$$

and

$$\mathbf{U}_\Phi = \begin{bmatrix} \mathbf{I}_p & 0 & 0 & 0 & \mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & \mathbf{I}_{uv} & 0 & 0 & -\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & 0 & \bar{\mathbf{I}}_p & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_w & \mathbf{A}_w^{-1}\widetilde{\mathbf{D}}_{uv}\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & 0 & 0 & 0 & \mathbf{I}_{\mathrm{ST}} \end{bmatrix} \tag{28}$$

To determine the quality of $\widehat{\Phi}$ as a preconditioner for $\tilde{\Phi}$ (the transformed matrix Φ, see (23)) we have to look at the generalized eigenvalue problem $\widetilde{\Phi}\mathbf{x} = \lambda\widehat{\Phi}\mathbf{x}$. For $\lambda = 1$ we can compute the difference $\widetilde{\Phi} - \widehat{\Phi}$:

$$\widetilde{\Phi} - \mathbf{L}_{\Phi}\mathbf{U}_{\Phi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv}\mathbf{A}_{uv}^{-1}\mathbf{G}_{uv}\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This shows that all eigenvalues of the preconditioned matrix are one. Unfortunately $\mathbf{L}_{\Phi}$ is not the true lower triangular factor of $\widetilde{\Phi}$, hence we cannot give the maximum geometric multiplicity of these eigenvalues. In the results section, we will see that this multiplicity cannot be very high since we do not find very large numbers of iterations.

The application of $\widehat{\Phi}$ as preconditioner requires the solution of two large systems involving $\mathbf{L}_{\Phi}$ and $\mathbf{U}_{\Phi}$, respectively. The last is the most easy to solve. Note that $\mathbf{U}_{\Phi}$ is a projection matrix. The inverse is the same as $\mathbf{U}_{\Phi}$ except for a minus-sign in front of the off-diagonal blocks in the last column. In other words $\mathbf{U}_{\Phi}^{-1} = (2\mathbf{I} - \mathbf{U}_{\Phi})$. There is no need to explicitly construct $\mathbf{U}_{\Phi}$ or its subblocks. We only need to apply its inverse to a vector. For example, $\mathbf{A}_p^{-1}\widetilde{\mathbf{B}}_{\mathrm{ST}}\mathbf{x}$ can be decomposed in a matrix vector product $\mathbf{y} = \widetilde{\mathbf{B}}_{\mathrm{ST}}\mathbf{x}$ and a solve of the system $\mathbf{A}_p\mathbf{z} = \mathbf{y}$ for $\mathbf{z}$. This holds for all subblocks in $\mathbf{U}_{\Phi}$. We can even reuse intermediate results, because the subblocks have a lot in common.

The system with $\mathbf{L}_{\Phi}$ is more difficult to solve, but, because the matrix is block lower triangular, we can subsequently solve equations with $\mathbf{A}_p$, the saddle point problem with $\mathbf{A}_{uv}$, $\mathbf{A}_w$ and $\mathbf{S}_{\mathrm{ST}}$. We discuss the solution of each of these smaller matrix problems in more detail below. We note that in practice the application of $\mathbf{L}_{\Phi}^{-1}$ means that we solve the different variables in the following order

$$\tilde{\mathbf{u}}_p \Rightarrow \mathbf{u}_{uv} + \bar{\mathbf{u}}_p \Rightarrow \mathbf{u}_w \Rightarrow \mathbf{u}_{\mathrm{TS}}$$

The application of $\mathbf{U}_{\Phi}^{-1}$ then gives a correction for $\tilde{\mathbf{u}}_p, \mathbf{u}_{uv}, \bar{\mathbf{u}}_p$ and $\mathbf{u}_w$ based on the computed $\mathbf{u}_{\mathrm{TS}}$.

3.3.1. Solving the system for the pressure

In this section, we show how the system with $\mathbf{A}_p$ is solved, i.e. that part of the pressure that is perpendicular to the null space of the gradient. We first split $\mathbf{G}_w = [\mathbf{G}_1\mathbf{G}_2]$ and $\mathbf{M}_p^{\mathrm{T}} = [\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_2^{\mathrm{T}}]$ in such a way that $\mathbf{G}_1$ and $\mathbf{M}_2$ are square. Hence, we would like to find the inverse of the matrix

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{G}_w \\ \mathbf{M}_p^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \\ \mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_2^{\mathrm{T}} \end{bmatrix}$$

with extra conditions $\mathbf{G}_w\mathbf{M}_p = 0$ and $\mathbf{M}_p^{\mathrm{T}}\mathbf{M}_p = \bar{\mathbf{I}}_p$, i.e.

$$\mathbf{G}_1\mathbf{M}_1 + \mathbf{G}_2\mathbf{M}_2 = 0$$
$$\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1 + \mathbf{M}_2^{\mathrm{T}}\mathbf{M}_2 = \bar{\mathbf{I}}_p$$

We can use these conditions to rewrite the matrix $\mathbf{A}_p$. The first condition implies that $\mathbf{G}_2 = -\mathbf{G}_1\mathbf{M}_1\mathbf{M}_2^{-1}$ and the second condition that $\mathbf{M}_2^{\mathrm{T}} = (\bar{\mathbf{I}}_p - \mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1)\mathbf{M}_2^{-1}$. If we substitute these expressions into the matrix $\mathbf{A}_p$, we can compute the following factorization, i.e.

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{G}_1 & -\mathbf{G}_1\mathbf{M}_1\mathbf{M}_2^{-1} \\ \mathbf{M}_1^{\mathrm{T}} & (\bar{\mathbf{I}}_p - \mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1)\mathbf{M}_2^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 & 0 \\ \mathbf{M}_1^{\mathrm{T}} & \bar{\mathbf{I}}_p \end{bmatrix} \begin{bmatrix} \mathbf{I}_w & -\mathbf{M}_1 \\ 0 & \bar{\mathbf{I}}_p \end{bmatrix} \begin{bmatrix} \mathbf{I}_w & 0 \\ 0 & \mathbf{M}_2^{-1} \end{bmatrix} \tag{29}$$

In our case $\mathbf{M}_2$ is a diagonal matrix, so we explicitly have the inverses of the last two factors. The only complication could come from the first factor where we have to solve a system with $\mathbf{G}_1$. Fortunately this appears to be quite easy, because $\mathbf{G}_1$ is by definition the square part of the discrete gradient $\mathbf{G}_w$, which is an upper triangular matrix with only one nonzero off-diagonal. The action of the inverse of $\mathbf{A}_p$ on a vector can be computed exactly at low cost.

3.3.2. Solving the saddle point problem

The next system is the saddle point problem

$$\begin{bmatrix} \mathbf{A}_{uv} & \mathbf{G}_{uv}\mathbf{M}_p \\ \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{uv} \\ \bar{\mathbf{u}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{uv} \\ \bar{\mathbf{b}}_p \end{bmatrix}$$

The system is queer in the sense that we have a full velocity field coupled to a depth-averaged pressure field. Hence, there are much more velocity nodes than pressure nodes. We can simplify the system by depth-integration of the velocity field. Let $\mathbf{M}_{uv}^{\mathrm{T}}$ be the depth-averaging operator for the velocities such that $\mathbf{M}_{uv}^{\mathrm{T}}\mathbf{M}_{uv} = \bar{\mathbf{I}}_{uv}$. Then the depth-averaged velocity field is $\bar{\mathbf{u}}_{uv} = \mathbf{M}_{uv}\mathbf{u}_{uv}$. Now define

$$\overline{\mathbf{A}}_{uv} = \mathbf{M}_{uv}^{\mathrm{T}}\mathbf{A}_{uv}\mathbf{M}_{uv}$$
$$\overline{\mathbf{G}}_{uv} = \mathbf{M}_{uv}^{\mathrm{T}}\mathbf{G}_{uv}\mathbf{M}_p$$
$$\overline{\mathbf{D}}_{uv} = \mathbf{M}_p^{\mathrm{T}}\mathbf{D}_{uv}\mathbf{M}_{uv}$$

Depth-integration of the momentum equations for the velocity gives the following saddle point problem

$$\begin{bmatrix} \overline{\mathbf{A}}_{uv} & \overline{\mathbf{G}}_{uv} \\ \overline{\mathbf{D}}_{uv} & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_{uv} \\ \bar{\mathbf{u}}_p \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{b}}_{uv} \\ \bar{\mathbf{b}}_p \end{bmatrix} \tag{30}$$

This problem, which is related to a barotropic equation, is different from the original one. The solution for $\bar{\mathbf{u}}_p$ is not the same, but hopefully the approximation is good enough. By depth integration, we at least use information that is spread all over the velocity field. One could choose a different operator than $\mathbf{M}_{uv}^{\mathrm{T}}$, for example one that picks a single horizontal velocity field instead of averaging. Then the approximation becomes worse and the overall performance of the preconditioner deteriorates.

The depth-averaged saddle point problem can be solved with a Krylov subspace method using a preconditioner. In Benzi et al. [1] one can find an overview of saddle point preconditioner in literature. Most of the preconditioners do not work well in our application because of the dominant Coriolis force. We propose the following alternative preconditioner for the matrix in (30)

$$\begin{bmatrix} \overline{\mathbf{A}}_{uv} & \overline{\mathbf{G}}_{uv} \\ \overline{\mathbf{D}}_{uv} & -\bar{\mathbf{I}}_p/\omega \end{bmatrix}$$

This preconditioner is based on artificial compressibility and application requires the solution of a system with the grad–div stabilized matrix

$$\overline{\mathbf{A}}_{uv}^{\mathrm{GD}} = \overline{\mathbf{A}}_{uv} + \omega\overline{\mathbf{G}}_{uv}\overline{\mathbf{D}}_{uv}$$

The spectral properties of this preconditioner and the motivation for its use are discussed in de Niet and Wubs [7]. Note that the size of the depth averaged saddle point problem is L (the number of cells in the vertical) times smaller than the original saddle point problem. The time to solve it is only a fraction of the time needed to solve the much bigger systems with $\mathbf{A}_{uv}$ and $\mathbf{S}_{\mathrm{ST}}$. Therefore, the choice of the preconditioner is not a crucial factor.

Given $\bar{\mathbf{u}}_p$ we can compute $\mathbf{u}_{uv}$ from the equation $\mathbf{A}_{uv}\mathbf{u}_{uv} = \mathbf{b}_{uv} - \mathbf{G}_{uv}\mathbf{M}_p\bar{\mathbf{u}}_p$. This is again done with a Krylov subspace method involving a preconditioner for the matrix $\mathbf{A}_{uv}$.

3.3.3. Solving the system for the vertical velocity

The vertical velocity is solved from an equation involving $\mathbf{A}_w$. The equation is easy to solve, because $\mathbf{A}_w$ is lower triangular.

3.3.4. Solving the Schur complement

The most difficult system to solve is the one with the Schur complement matrix $\mathbf{S}_{\mathrm{ST}}$, that we defined in (25). Fortunately there is no need to construct this matrix explicitly. In a Krylov subspace method we only need to be able to apply it to a vector. The matrix vector product $\mathbf{S}_{\mathrm{ST}}\mathbf{x}$ can be decomposed in a number of easier matrix vector products and a few system solves.

The next question we have to answer is: how do we obtain a good preconditioner for $\mathbf{S}_{\mathrm{ST}}$. It appears that in our case an incomplete LU factorization of $\mathbf{A}_{\mathrm{ST}}$ suffices. One can get a better preconditioner by constructing a

factorization for a better approximation to the Schur-complement. For example one can take in account more of the terms of (25). In general the construction of such a better approximation will be quite expensive and the number of nonzeros will increase rapidly, which makes the construction of an incomplete factorization much more expensive. We tried several approximations, but none of them did beat the relatively cheap incomplete LU factorization of $\mathbf{A}_{ST}$.

3.3.5. Nested iterations

The systems with $\mathbf{A}_{uv}$, $\mathbf{A}_{ST}$ and the depth averaged saddle point problems are solved with right-preconditioned GMRES using the MRILU preconditioner [2]. On these subsystems MRILU performs very well. It is able to produce a good incomplete factorization with modest fill-in in short time. In all cases we use clustering of the two variables u, v and T, S. Because we apply nested iterations, we choose a flexible Krylov method (FGMRES [27]) for the outer iteration.

3.4. Advantages of the block preconditioner over MRILU

At the end of this section, we want to point at a few advantages of the block-ILU preconditioner compared to MRILU.

We expect that the block-ILU solver scales almost linearly with the problem size. Most of the operations we have to perform for the construction and application of the preconditioner, scale linearly with the problem size. Extraction of submatrices, computation of sparse matrix transposes, matrix vector products and the solution of systems with lower or upper-triangular matrices can all be computed in $\mathcal{O}(d)$ time. This linear behavior with the problem size can only be violated by the computation and application of the MRILU factorization of the matrices $\mathbf{A}_{uv}$, $\mathbf{A}_{ST}$ and $\overline{\mathbf{A}}_{uv}^{GD}$. Fortunately, these matrices are of convection–diffusion type and MRILU shows almost grid-independent convergence for that kind of problems [2].

We also expect that less memory is needed for the block-ILU solve than with MRILU applied directly to (20). MRILU requires much memory because a high amount of fill is needed for an accurate factorization. The block-ILU preconditioner requires the storage of the subblocks and the MRILU preconditioners for the much smaller matrices $\mathbf{A}_{uv}$, $\mathbf{A}_{ST}$ and $\overline{\mathbf{A}}_{uv}^{GD}$. Because MRILU is a suitable preconditioner for these matrices, the fill generated by the MRILU factorization will be rather small. Immediate consequence is that both construction and storage of the MRILU factors is much cheaper. Overall, we expect a serious decrease of the memory requirements.

The last advantage we want to mention is that parallelization of block-ILU preconditioner seems straightforward. If we have good parallel solvers for the solution of the subsystems involving $\mathbf{A}_{uv}$, $\mathbf{A}_{ST}$ and $\overline{\mathbf{A}}_{uv}^{GD}$, we will be able to build a good parallel code. Because all these matrices are of convection–diffusion type we expect that we can find suitable parallel solvers for them.

4. Numerical results

In this section, we apply the implicit ocean model in a pseudo-arclength continuation set-up with the block-ILU solver to two problems. The first problem is the double-gyre wind-driven circulation in an idealized relatively small basin (Section 4.1) and the second problem is the computation of wind- and buoyancy-driven single-hemispheric flows in an Atlantic size single-hemispheric basin. The aim of the presentation of the results is to demonstrate the capabilities of the implicit model and hence the physics of the solutions will not be discussed in detail.

4.1. The double-gyre problem

As in the studies with quasi-geostrophic (QG) and shallow-water (SW) models, a basin of 10° length and 10° width centered around 45°N is considered. In longitude ϕ and latitude θ, the boundaries of the domain are given by $\phi_w = 300°E$, $\phi_e = 310°E$, $\theta_s = 40°N$ and $\theta_n = 50°N$; the basin has a constant depth $D = 2400$ m. The flow is only forced by a steady wind stress and otherwise has a constant temperature and salinity. A so-called double-gyre wind forcing is prescribed of the form

$$\tau^{\phi}(\phi,\theta) = -\tau_0 \cos\left(2\pi \frac{\theta-\theta_s}{\theta_n-\theta_s}\right) \tag{31}$$

where τ_0 is a typical amplitude.

In the case of no thermal and freshwater forcing, $T_S = S_S = 0$, we know from QG and SW models that multiple western boundary current separation paths may exist due to symmetry breaking of the wind-driven flow. This has, as far as we know, never been demonstrated in primitive-equation models. Due to spherical coordinates, in the latter equation there is no reflection symmetry with respect to the mid axis of the basin (here $\theta = 45°$) and an imperfect pitchfork bifurcation is expected [8]. An imperfect pitchfork bifurcation is a generalization of a pitchfork bifurcation. If the domain would have had a reflection symmetry then we would have had a symmetric solution for large A_H. If in this case by decreasing A_H a nonsymmetric solution shows up then it must have also a counterpart due to the reflection symmetry. These solutions diverge in a continuous way from the symmetric solution at the bifurcation point which in a plot visualizes as a pitchfork. In our problem we have approximately a reflection symmetry, therefore for large A_H we also find a solution which is nearly symmetric (see Fig. 4a). When decreasing A_H we may end up in a strongly non-symmetric solution (Fig. 4b). It is to be expected there also exists a variant which is about the reflection of this non-symmetric solution (Fig. 4c). In this case there is no argument that they should depart both from the "symmetric" solution, simply because the latter only exists approximately. So the branch of the "reflection" is not attached to the nearly symmetric solution for large A_H (Fig 3). It is however, attached to another nearly symmetric solution (Fig. 4d). The plot will now show an imperfect pitchfork.

We fix the vertical resolution at $L = 12$, such that $H_m = 200$ m. To test the effects of the horizontal resolution on the pure wind-driven solutions of the model and the performance of the new block-ILU solver, we start with a value of $A_H = 1600\ \mathrm{m^2\ s^{-1}}$. For this value of A_H, a 40 km horizontal resolution (25 gridpoints) will still resolve the western boundary (Munk) layer, such that the solutions at three resolutions $25 \times 25 \times 12$, $50 \times 50 \times 12$ and $100 \times 100 \times 12$ can be compared. The Munk layer profile has an oscillatory spatial structure in zonal direction [26] and hence one can do with only 1–2 gridpoints over a distance δ_M. Note that in the last case the number of unknowns (six per grid cell) is about 0.7 million.

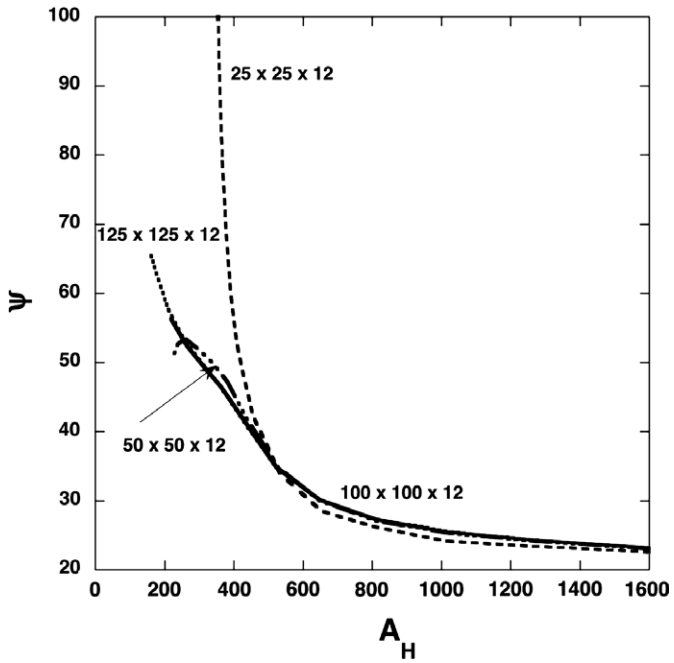

Fig. 2. Solution branch of the constant density, wind-driven ocean circulation with control parameter A_H for the four different resolutions indicated.

We increase the wind stress parameter τ_0 from zero to the value of 0.1 Pa and then decrease A_H. Along the solution branch, the maximum value of the barotropic streamfunction is monitored and the result is plotted in Fig. 2. For $A_H = 1600\ m^2\ s^{-1}$ and $\tau_0 = 0.1$ Pa, there is a clear quadratic convergence in the maximum value of the barotropic streamfunction ψ_B, as shown in the second column of Table 2. The curves in Fig. 2 remain close to a value of $A_H \approx 650\ m^2\ s^{-1}$. The curve for the $25 \times 25 \times 12$ grid starts to diverge first, since the resolution is not sufficient anymore to resolve the Munk boundary layer. The curve for the 20 km horizontal resolution starts to diverge near $A_H \approx 400\ m^2\ s^{-1}$ for the same numerical reason. Based on the dependence of the Munk boundary layer thickness $\delta_M = (A_H/\beta_0)^{1/3}$ (with $\beta_0 = 2\Omega\cos\theta_0/r_0$) on A_H, the solutions for the $100 \times 100 \times 12$ grid are expected to be accurate for values of A_H down to $A_H \approx 250\ m^2\ s^{-1}$. This is confirmed with the results of the $125 \times 125 \times 12$ grid.

For each resolution, about 10–20 branch steps were taken to compute the curves in Fig. 2. To provide an impression of the performance of the block-ILU solver, we provide in the other entries of Table 2 the average CPU time (in seconds on a 1 GHz HP-DS25 workstation with 16 GB of memory) of the building of the matrix, the construction of the block-ILU preconditioner and the FGMRES solution of the linear system (16). For all these timing results, the parameters in the block-ILU solver are such that the accuracy of FGMRES iterate has improved by two decimal digits and that of the saddle point problem by 6. In the first three cases the number of unknowns increases by a factor 4. This factor is observed in the construction of the

Table 2
Overview of CPU time (in s) needed for the first iteration within the Newton–Raphson process with a step $ds = -10.0$ in A_H starting at $A_H = 1600\ m^2\ s^{-1}$

Grid	ψ_B	d	Matrix	Block-ILU	#it	FGMRES
$25 \times 25 \times 12$	22.53	45,000	0.9	1.8	7	0.8
$50 \times 50 \times 12$	23.01	180,000	3.8	9.1	9	5.5
$100 \times 100 \times 12$	23.14	720,000	16.3	48.5	11	42.9
$125 \times 125 \times 12$	23.15	1,125,000	24.9	84.7	8	48.7

Also the value of the maximum of the barotropic streamfunction ψ_B is shown for the four different resolutions at $A_H = 1600\ m^2\ s^{-1}$. The quantity d is the number of degrees of freedom and the number of iterations (#it) in the FMGRES iterative process is also shown.

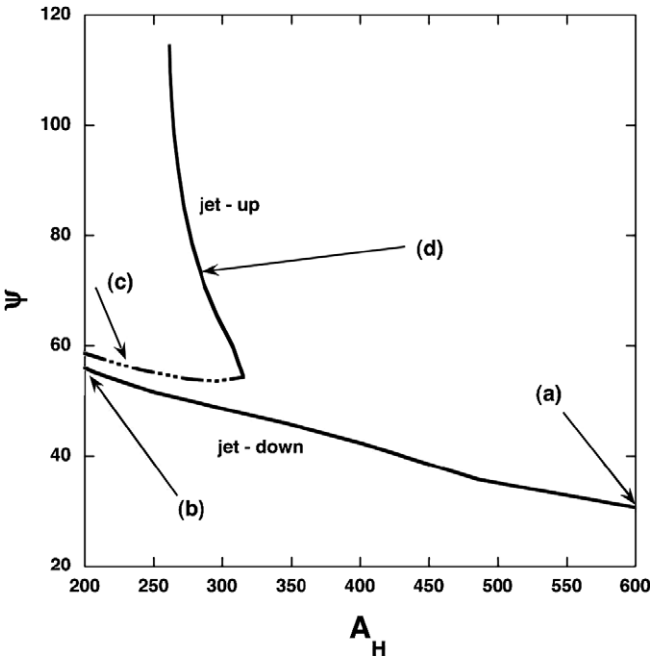

Fig. 3. Bifurcation diagram showing the maximum of the barotropic streamfunction ψ_B (in Sv, 1 Sv $= 10^6\ m^3\ s^{-1}$) of the constant density circulation versus the control parameter A_H (in $m^2\ s^{-1}$). Solutions at the labeled locations are presented in Fig. 4.

matrix, it is about a factor 5 and 7 for the construction of the block ILU preconditioner and the solution process, respectively. Hence the total CPU time per Newton step increases slightly more than linearly with the number of degrees of freedom.

In a typical bifurcation diagram, we use the lateral friction coefficient A_H as a parameter and a resolution of $100 \times 100 \times 12$ points (Fig. 3). In agreement to what is found in shallow-water models, we here find an imperfect pitchfork bifurcation, which leads to the existence of multiple patterns below a value of $A_H = 320\ \mathrm{m^2\ s^{-1}}$. The isolated branch in Fig. 3 was found by the residue continuation algorithm as described in chapter 4 of Dijkstra [8] and analyzed in Gruais et al. [14]. Note that explicit models will not be able to find the dashed-dotted branch of solutions since it is unstable.

The barotropic streamfunction of steady solutions at specific labeled locations in Fig. 3 are plotted in Fig. 4. The solution for large A_H on the connected branch is the near anti-symmetric double-gyre flow (Fig. 4a). When A_H decreases along this branch, a so-called jet-down solution appears (Fig. 4b), a solution very well known from QG and SW models [8]. Along the isolated branch, a jet-up solution exists (Fig. 4c) along the lower branch and after the saddle-node bifurcation at $A_H = 320\ \mathrm{m^2\ s^{-1}}$, the flow becomes more inertially controlled (Fig. 4d). Although these results were expected, it is the first time that such a multiple equilibria structure is computed for a full primitive-equation model.

4.2. Wind- and thermohaline flows

Similar to the study in Te Raa and Dijkstra [35], we consider a basin of 64° length and 64° width centered around 40°N. In longitude ϕ and latitude θ, the boundaries of the domain are given by $\phi_w = 286°\mathrm{E}$, $\phi_e = 350°\mathrm{E}$, $\theta_s = 10°\mathrm{N}$ and $\theta_n = 74°\mathrm{N}$; the basin has a constant depth $D = 4000$ m. As forcing we choose

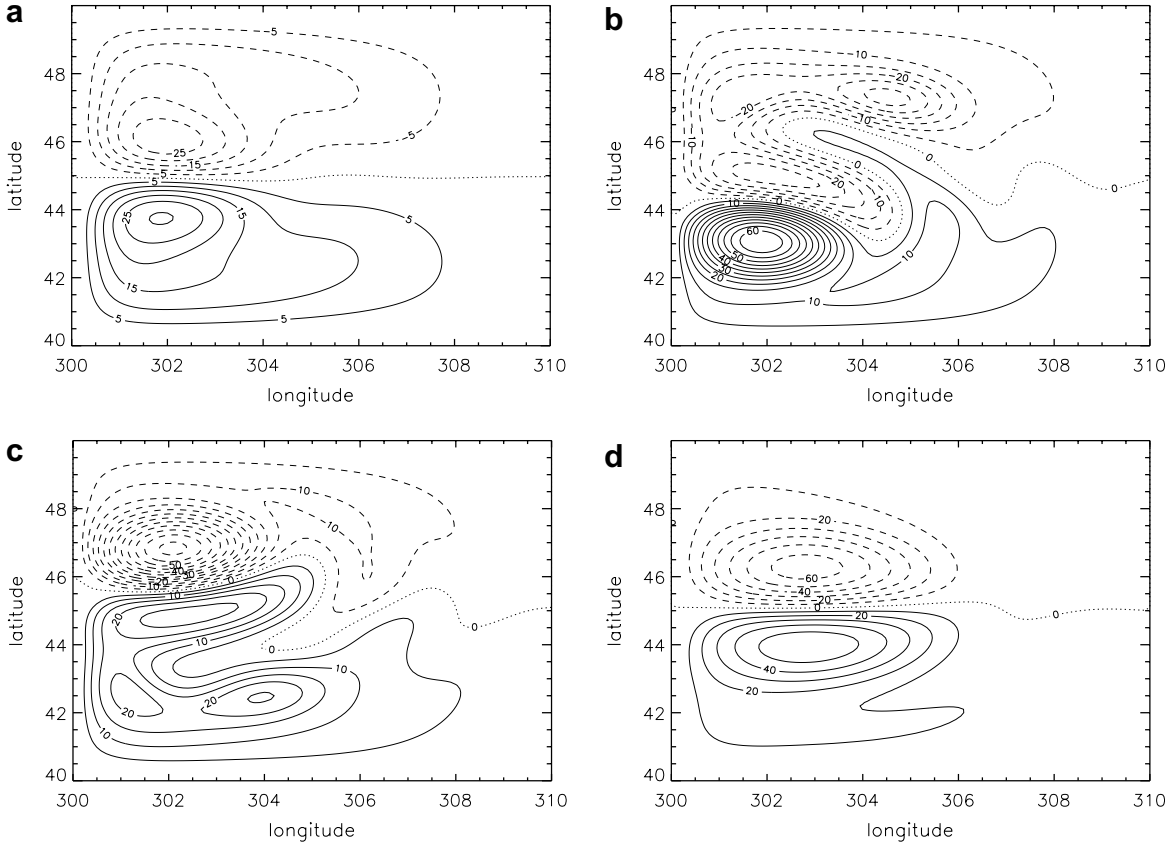

Fig. 4. Patterns of the barotropic streamfunction of labeled locations in Fig. 3. Contour values are in Sv.

$$T_S(\phi,\theta) = T_0 + \eta\frac{\Delta T}{2}\cos\left(\pi\frac{\theta-\theta_s}{\theta_n-\theta_s}\right) \tag{32a}$$

$$S_S(\phi,\theta) = S_0 + \eta\frac{\Delta S}{2}\cos\left(\pi\frac{\theta-\theta_s}{\theta_n-\theta_s}\right) \tag{32b}$$

where we have introduced another homotopy parameter η such that $\eta = 0$ represents no forcing and $\eta = 1$ is a realistic strength of the forcing. Furthermore, we use interpolated values of the annual mean wind stress dataset of Trenberth et al. [37] and multiply the amplitude of this forcing by η. In this way, we can vary η and increase all three forcing functions simultaneously.

4.2.1. Constant mixing coefficients

We first take a constant vertical mixing coefficient, i.e. $\eta_V = 0$ in (11) with $K_V^0 = 10^{-4}\ \mathrm{m^2\ s^{-1}}$ and $K_V^c = 10^2\ \mathrm{m^2\ s^{-1}}$. In addition, we use values $A_H = 2.5\times10^5\ \mathrm{m^2\ s^{-1}}$, $K_H = 1.0\times10^3\ \mathrm{m^2\ s^{-1}}$, $\Delta T = 20\ ^\circ\mathrm{C}$ and $\Delta S = 1.0$ psu. These are values quite similar to those used in ocean-climate models of similar resolution. Furthermore, we do not yet consider neutral mixing ($\eta_M = 0$) and GM-mixing ($\eta_G = 0$). The vertical resolution is chosen as $L = 16$ such that $H_m = 250$ m. Maximum values of the meridional overturning streamfunction (ψ_M) are plotted versus η in Fig. 5 for three different resolutions; the highest resolution corresponds to a horizontal resolution of 1°. Beyond $\eta = 0.3$, the three curves start to deviate from each other and a smaller value of the meridional overturning occurs for the higher resolution model at larger values of η.

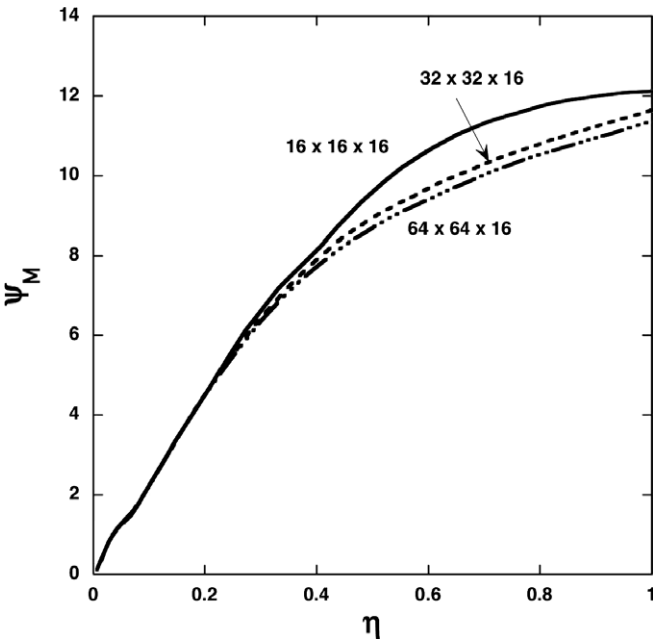

Fig. 5. Solution branch, showing the maximum of the meridional overturning streamfunction ψ_M (in Sv) of the thermohaline and wind-driven ocean circulation versus the control parameter η for the three different resolutions indicated.

Table 3
Overview of CPU time (in s) needed for the first iteration in the Newton–Raphson process for a step $\Delta s = 0.01$ in η starting at $\eta = 1$

Resolution	ψ_M	d	Matrix	Block-ILU/MRILU	#it	FGMRES/GMRES
$16\times16\times16$	12.12	24,576	2.0	9.1/260.6	37	44.2/81.1
$32\times32\times16$	11.64	98,304	8.6	70.6	60	330.2
$64\times64\times16$	11.36	393,216	36.9	557.2	35	882.9

The format is the same as in Table 2. For the lowest resolution, also timing results for the original MRILU solver are provided. The maximum of the meridional overturning streamfunction ψ_M for three different resolutions and $\eta = 1$ is also presented. No restart is used for (F)GMRES.

The maximum of meridional overturning streamfunction (ψ_M) converges with the horizontal resolution as can be seen from the second entry of Table 3. From each solution at $\eta = 1$, one step with $\Delta s = 0.01$ was performed in η and the CPU time required for the first iteration in the Newton–Raphson process is shown in Table 3. It is clear that computations involving these thermohaline and wind-driven flows are more expensive than those for only wind-driven flows, since the flow is simply more complex with salt and temperature variations over the computational domain. In column 4 one observes that the time for constructing the matrix is

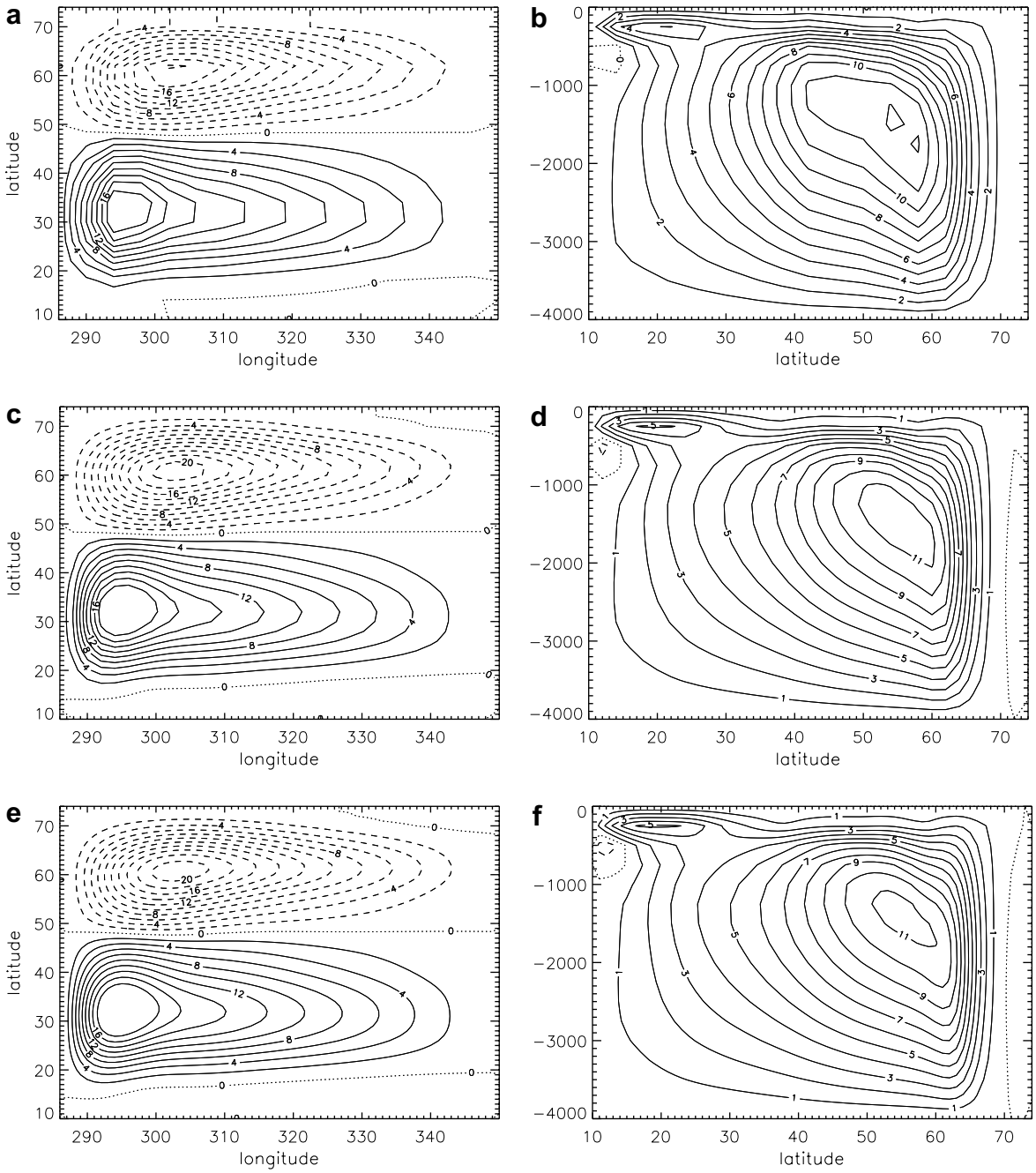

Fig. 6. Patterns of the barotropic streamfunction (a, c, e) and meridional overturning streamfunction (b, d, f) at $\eta = 1$. (a and b) $16 \times 16 \times 16$, (c and d) $32 \times 32 \times 16$ and (e and f) $64 \times 64 \times 16$.

linear in the number of unknowns as may be expected. However, this is not the case for the construction of the preconditioner. A similar behavior is shown in Dijkstra et al. [10] for MRILU applied to the whole system at once, and will also occur if a direct method with fill-reducing order is used. However, the coefficient of this behavior is in the new solver more than an order of magnitude less than that in the original MRILU approach. The behavior results from the construction of the MRILU preconditioner for A_{ST}, and is due to the dropping parameters used. These are currently not optimized. We see that the number of iterations in the 6th column has an outlier, which also makes the solution phase relatively expensive here. Correcting for this number of iterations, we observe a solution time which increases almost linear with the number of unknowns. In any case we see that the geometric multiplicity of the eigenvalues of the preconditioned matrix (see the discussion in Section 3.3) cannot be very high since the number of iterations is still relatively small.

For the lowest resolution, a comparison with the old MRILU method is provided in the 5th and 7th column of Table 3. The new solver is a significant improvement over the old approach, in particular the construction of the preconditioner is more than a magnitude faster needing also less computing time for the solution phase. Not shown is that also the use of memory is substantially less. With the original approach we run much earlier out of core than the new approach if the number of freedoms is increased. This allows us to run problems with higher resolution needed to catch important physical phenomena.

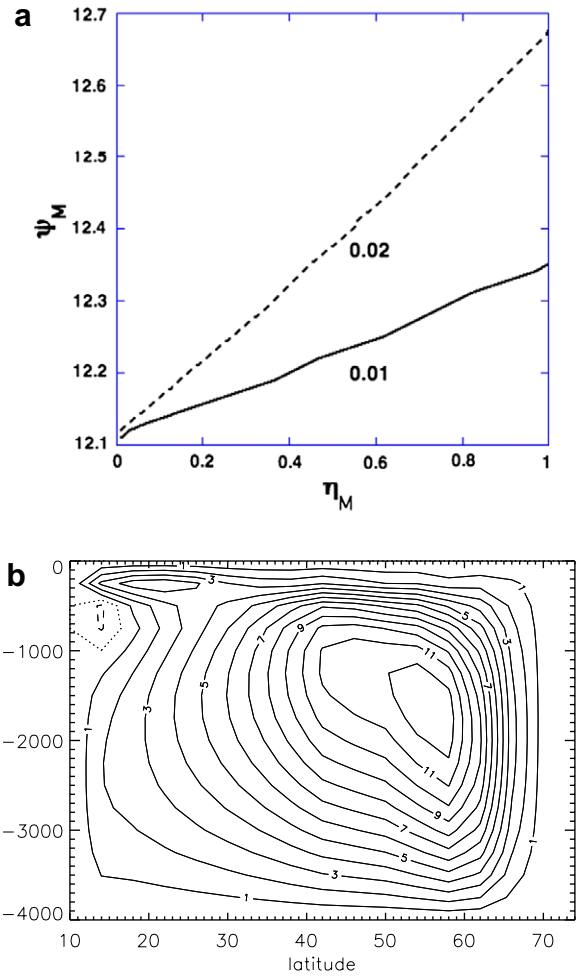

Fig. 7. (a) Dependence of ψ_M on η_M for $\eta = 1$ for two values of the slope limiter $\tan\alpha_m$. (b) Pattern of the meridional overturning streamfunction for $\eta_M = 1$, $\eta = 1$, $\tan\alpha_m = 0.02$; contours in Sv.

Patterns of the meridional overturning streamfunction ψ_M and the barotropic streamfunction ψ_B are plotted in Fig. 6 at $\eta = 1$ for each of the three resolutions. Each solution is completely stably stratified and sinking occurs north of 60°N. In principle, the solutions for the coarsest grid are already sufficiently accurate for this value of K_V^0 and A_H and they are just smoothed at higher resolution. This justifies the resolution used in many of the earlier results on single-hemispheric flows [35].

4.2.2. Spatially variable mixing

So far, only results for a constant mixing coefficients K_V were presented i.e. $\eta_V = 0$ in (11) and no neutral and GM-mixing was applied ($\eta_M = \eta_G = 0$). We first consider the effect of neutral mixing on the solutions on a $16 \times 16 \times 16$ grid for $\eta = 1$ (i.e. Fig. 6a and b). For two values of the slope limiter $\tan\alpha_m = 0.01$ and 0.02, we increase η_M from zero to one, while keeping $\eta_G = 0$ and $\kappa = K_H$. The presence of neutral mixing leads to a slight increase in the strength of the meridional overturning circulation (Fig. 7a). Allowing larger slopes removes spurious diapycnal mixing over a larger area and hence the increase in meridional overturning is larger. In the solution for $\eta_M = \eta = 1$ in Fig. 7b, the small equatorial cell has more or less disappeared indicating that it is due to spurious diapycnal mixing.

From the solution in Fig. 7, we increase η_G to include GM-mixing and see that for both slope limiters, the strength of the MOC further increases (Fig. 8a). Although the Newton–Raphson method has no trouble

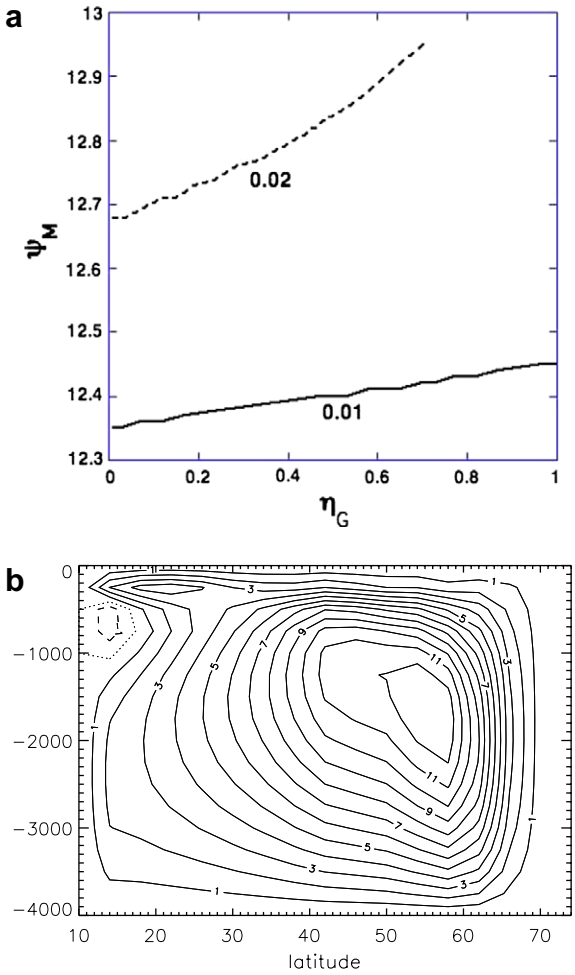

Fig. 8. (a) Dependence of ψ_M on η_G for $\eta = 1$ for two values of the slope limiter $\tan\alpha_m$. (b) Pattern of the meridional overturning streamfunction for $\eta_G = 0.5$, $\eta = 1$, $\tan\alpha_m = 0.02$; contours in Sv. The solution for $\tan\alpha_m = 0.01$, $\eta = 1.0$ is very similar.

converging, small local differences in mixing create the small wiggles in the curves in Fig. 8a. The solution for $\eta_G = 0.5$ (Fig. 8b) is hardly distinguishable from that of $\eta_G = 0$ (Fig. 7b). We checked the results of our model with respect with neutral and GM mixing by comparing them, for only temperature forcing, to steady solutions obtained with version 3.1 of the GFDL Modular Ocean Model.

We next consider a spatially variable K_V while choosing $\eta_M = \eta_G = 0$. Using a $16 \times 16 \times 16$ resolution, we first change the value of K_V^0 to 2.0×10^{-5} m^2 s^{-1} which is a more realistic value of the background vertical mixing coefficient. Again with the same forcing functions of wind, temperature and salinity, we increase η from $\eta = 0$ to 1 under a constant K_V^0. The maximum value of the meridional overturning streamfunction is plotted versus η in Fig. 9a. The maximum meridional overturning increases up to about 6.8 Sv which is much smaller than the earlier value of 12.12 Sv (found for $K_V^0 = 2.0 \times 10^{-5}$ m^2 s^{-1}). The pattern of the MOC at $\eta = 1$ for this case is shown in Fig. 9b and displays a large subsurface near-equatorial cell indicating a larger effect of spurious diapycnal mixing on the solution.

We next choose the field e in (11) as

$$e(\phi, \theta, z) = \left[\frac{-ze_0}{D}\right]\left(\cos\frac{\pi}{2}\frac{\theta - \theta_s}{\theta_n - \theta_s} + 0.2\right) \tag{33}$$

with $e_0 = 10^{-9}$ m^2 s^{-3}. This representation mimics the larger available energy for mixing at the bottom of the domain, thought as being caused by the interaction of tides and topography [39]. We next continue η_V from $\eta_V = 0$ to 1 to allow for mixing as determined from the background stratification according (11) and the en-

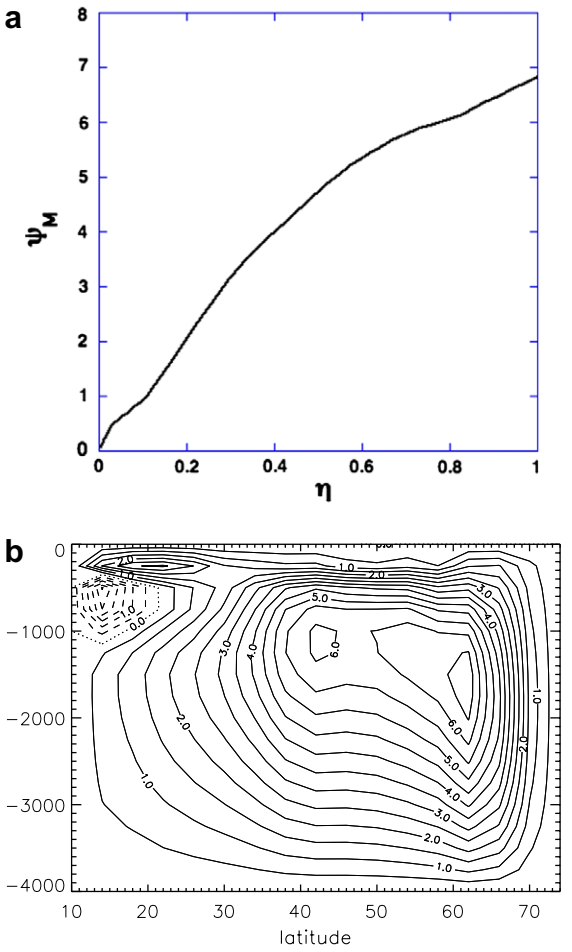

Fig. 9. (a) Dependence of ψ_M on η for $\eta_V = 0$. (b) Pattern of the meridional overturning streamfunction for $\eta_V = 0$, $\eta = 1$; contours in Sv.

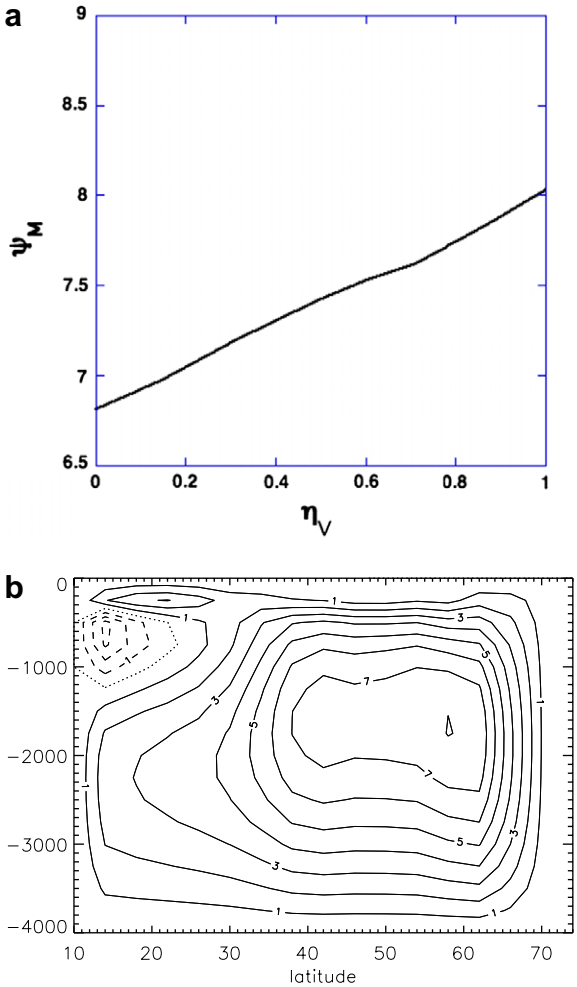

Fig. 10. (a) Dependence of ψ_M on η_V for $\eta = 1$. (b) Patterns of the meridional overturning streamfunction for $\eta_V = 1$, $\eta = 1$; contours in Sv.

ergy field e as in (33). The dependence of the maximum of ψ_M on η_V is plotted in Fig. 10a and shows an increase in overturning with η_V of about 1.5 Sv. Because of the larger mixing at the bottom of the domain, the overturning increases at depth as can be seen in Fig. 10b. This has also an effect on the equatorial cell, which increases in strength. We next vary e_0 from the solution in Fig. 10a for $\eta_V = 1$, $\eta = 1$. The dependence of the strength of the meridional overturning on e_0 is plotted in Fig. 11a. As expected, the overturning strength increases with increasing e_0 as more energy is available for mixing.

5. Summary and discussion

Techniques from numerical bifurcation theory have demonstrated their usefulness to provide more insight into the physics of transition behavior in ocean models [8]. So far, they have been applied to ocean models with a relatively small number of degrees of freedom (up to $\mathcal{O}(10^5)$), such as 1.5-layer shallow-water models, 2-layer quasi-geostrophic models and low-resolution three-dimensional primitive equation models [35,38].

Major weaknesses in previous models was the representation of horizontal and vertical mixing processes and the application of convective adjustment. In the model formulation used here, this has been overcome by evaluating the Jacobian matrix in part numerically. This enables to include more realistic representations

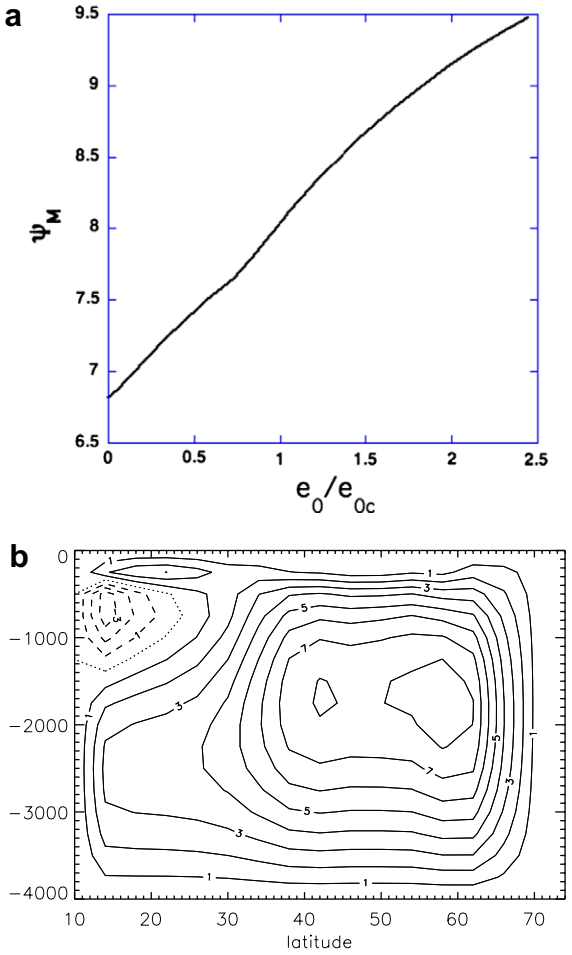

Fig. 11. (a) Dependence of ψ_M on e_0/e_{0c}, with $e_{0c} = 10^{-9}\,\mathrm{m^2\,s^{-3}}$ for $\eta = 1$ and $\eta_V = 1$. (b) Patterns of the meridional overturning streamfunction for $e_0/e_{0c} = 2.5$, $\eta_V = 1$, $\eta = 1$; contours in Sv.

of oceanic mixing processes. We have shown the capabilities of this approach by considering energy consistent vertical mixing in combination with convective adjustment. The solvers are seen to handle these cases efficiently. The tailored linear system solver presented in this paper is therefore a major step forward to apply bifurcation analysis to ocean models with up to $\mathcal{O}(10^6)$ degrees of freedom. An alternative approach would be the one chosen in Nadiga et al. [21], but this has so far not been applied for bifurcation studies.

The solver here is based on the fact that splitting the pressure in an average pressure and a part perpendicular to it leads to a system with more zeros in it which can be put almost in lower triangular block form. In fact, a lower triangular system would occur if the fluid is barotropic, hence if the pressure does not depend on salinity and temperature. However, in the ocean model it does and it is critical. Therefore, instead of approximating by a lower triangular matrix, we approximate the matrix by a block-ILU factorization, which is used as a preconditioner. The solver applies this preconditioner within FGMRES, a variant of GMRES that allows for iterative subprocesses, as iterative method. This is needed because the application of the block-ILU preconditioner requires the solution of a number of subsystems, some of which can be solved exactly but others are solved by GMRES using the MRILU preconditioner. With the block-ILU preconditioner, the FGMRES (the outer iterative method) converges very fast to an accurate solution.

Compared to the previous solver used (MRILU with GMRES [38]), the block-ILU approach has the advantage that the construction of the preconditioner is an order of magnitude faster, storage of the preconditioner requires less memory and consequently the application of the preconditioner is much cheaper. The

construction time for the preconditioner for the thermohaline flow increases like $\mathcal{O}(d^{3/2})$ and for the solution almost like $\mathcal{O}(d)$. The latter is very well acceptable. We think we can also improve on the former, which is a subject of further study. The critical systems to be solved are of convection–diffusion type. This also eases the path to parallelization since many advanced parallel methods exist for that in the literature.

The parallelization is the next step in our model development, which should allow us to run problems with 10^7 degrees of freedom. For the parallelization we will use domain decomposition; each domain is assigned to a processor which computes the right-hand side function, the Jacobian matrix and the matrices needed for the incomplete block factorization for its own domain. This will be implemented in Trilinos, which has a variety of parallel solvers available for the solution of the subsystems.

Two typical flow examples were used to demonstrate the performance of the new solver. We were able to compute the bifurcation structure of the isothermal wind-driven double-gyre flow using a 10 km resolution involving the analysis of a system of 720,000 degrees of freedom. This is the first time that an imperfect pitch-fork bifurcation is computed in a primitive-equation ocean model. Encouraging result is here that the computational cost increases approximately linearly with the number of degrees of freedom. Furthermore, we were able to compute steady wind- and buoyancy-driven flows in a single-hemispheric Atlantic size basin with up to 1° horizontal resolution and represent horizontal and vertical mixing processes in a state-of-the-art way. The latter result opens the way to study the interaction of the thermohaline and wind-driven ocean circulation in more detail using tools of dynamical systems theory.

Acknowledgments

This work was supported by the Technology Foundation STW within project GWI 5798, the Applied Science Division of NWO and the Technology Program of the Ministry of Economic Affairs.

References

[1] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[2] E.F.F. Botta, F.W. Wubs, Matrix renumbering ILU: an effective algebraic multilevel ILU preconditioner for sparse matrices, SIAM J. Matrix Anal. Appl. 20 (4) (1999) 1007–1026 (electronic). Sparse and structured matrices and their applications (Coeur d'Alene, ID, 1996).
[3] F.O. Bryan, High-latitude salinity effects and interhemispheric thermohaline circulations, Nature 323 (1986) 301–304.
[4] F. Chen, M. Ghil, Interdecadal variability of the thermohaline circulation and high-latitude surface fluxes, J. Phys. Oceanogr. 22 (1995) 161–167.
[5] T. Coleman, B.S. Garbow, J.J. Moré, Software for estimating sparse Jacobian matrices, ACM Trans. Math. Software 10 (1984) 329–345.
[6] G. Danabasoglu, J.C. McWilliams, Sensitivity of the global ocean circulation to parameterizations of mesoscale tracer transports, J. Phys. Oceanogr. 8 (1995) 2967–2987.
[7] A. de Niet, F. Wubs, Two saddle point preconditioners for fluid flows, Int. J. Numer. Method Fluid 54 (2007) 355–377.
[8] H.A. Dijkstra, Nonlinear Physical Oceanography: A Dynamical Systems Approach to the Large Scale Ocean Circulation and El Niño, second ed., Springer, Dordrecht, The Netherlands, 2005.
[9] H.A. Dijkstra, C.A. Katsman, Temporal variability of the wind-driven quasi-geostrophic double gyre ocean circulation: basic bifurcation diagrams, Geophys. Astrophys. Fluid Dynam. 85 (1997) 195–232.
[10] H.A. Dijkstra, H. Öksüzŏglu, F.W. Wubs, E.F.F. Botta, A fully implicit model of the three-dimensional thermohaline ocean circulation, J. Comput. Phys. 173 (2001) 685–715.
[11] H.A. Dijkstra, L.A. Te Raa, M. Schmeits, J. Gerrits, On the physics of the Atlantic Multidecadal Oscillation, Ocean Dynam. 56 (2006) 36–50.
[12] P.R. Gent, J. Willebrand, T.J. McDougall, J.C. McWilliams, Parameterizing eddy-induced tracer transports in ocean circulation models, J. Phys. Oceanogr. 25 (1995) 463–474.
[13] S.M. Griffies, Fundamentals of Ocean-Climate Models, Princeton University Press, Princeton, USA, 2004.
[14] I. Gruais, N. Rittemard, H.A. Dijkstra, A priori estimations of a global homotopy residue continuation method, Nonlinear Funct. Anal. Optim. 4–5 (2005) 507–521.
[15] R.X. Huang, Mixing and energetics of the oceanic thermohaline circulation, J. Phys. Oceanogr. 29 (1999) 727–746.
[16] T. Huck, A. Colin de Verdiére, A.J. Weaver, Interdecadal variability of the thermohaline circulation in box-ocean models forced by fixed surface fluxes, J. Phys. Oceanogr. 29 (1999) 865892.
[17] T. Huck, G. Vallis, A. Colin de Verdiére, On the robustness of interdecadal oscillations of the thermohaline circulation, J. Climate 14 (2001) 940–963.
[18] S. Jiang, F.-F. Jin, M. Ghil, Multiple equilibria and aperiodic solutions in a wind-driven double-gyre, shallow-water model, J. Phys. Oceanogr. 25 (1995) 764–786.

[19] H.B. Keller, Numerical solution of bifurcation and nonlinear eigenvalue problems, in: P.H. Rabinowitz (Ed.), Applications of Bifurcation Theory, Academic Press, New York, USA, 1977.
[20] S.P. Meacham, Low frequency variability of the wind-driven circulation, J. Phys. Oceanogr. 30 (2000) 269–293.
[21] B. Nadiga, M. Taylor, J. Lorenz, Ocean modelling for climate studies: eliminating short time scales in long-term, high-resolution studies of ocean circulation, Math. Comp. Model. 44 (2006) 870–886.
[22] B.T. Nadiga, B. Luce, Global bifurcation of Shil nikov type in a double-gyre model, J. Phys. Oceanogr. 31 (2001) 2669–2690.
[23] J. Nauw, H.A. Dijkstra, The origin of low-frequency variability of double-gyre wind-driven flows, J. Mar. Res. 59 (2001) 567–597.
[24] J. Nauw, H.A. Dijkstra, E. Chassignet, Frictionally induced asymmetries in wind-driven flows, J. Phys. Oceanogr. 34 (2004) 2057–2072.
[25] J.D. Neelin, D.S. Battisti, A.C. Hirst, F.-F. Jin, Y. Wakata, T. Yamagata, S.E. Zebiak, ENSO Theory, J. Geophys. Res. 103 (1998) 14261–14290.
[26] J. Pedlosky, Geophysical Fluid Dynamics, second ed., Springer-Verlag, New York, 1987.
[27] Y. Saad, A flexible inner–outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469.
[28] Y. Saad, M. Schultz, A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[29] M.J. Schmeits, H.A. Dijkstra, Bimodality of the Kuroshio and the Gulf Stream, J. Phys. Oceanogr. 31 (2001) 2971–2985.
[30] A. Siegel, J.B. Weiss, J. Toomre, J.C. McWilliams, P. Berloff, I. Yavneh, Eddies and coherent vortices in ocean basin dynamics, Geophys. Res. Lett. 28 (2001) 3183–3186.
[31] E. Simonnet, M. Ghil, K. Ide, R. Temam, S. Wang, Low-frequency variability in shallow-water models of the wind-driven ocean circulation. Part I: Steady-state solutions, J. Phys. Oceanogr. 33 (2003) 712–728.
[32] E. Simonnet, M. Ghil, K. Ide, R. Temam, S. Wang, Low-frequency variability in shallow-water models of the wind-driven ocean circulation. Part II: Time dependent solutions, J. Phys. Oceanogr. 33 (2003) 729–752.
[33] E. Simonnet, M. Ghil, H.A. Dijkstra, Homoclinic bifurcations of barotropic QG double-gyre flows, J. Mar. Res. 63 (2005) 931–956.
[34] S. Speich, H.A. Dijkstra, M. Ghil, Successive bifurcations of a shallow-water model with applications to the wind driven circulation, Nonlinear Proc. Geophys. 2 (1995) 241–268.
[35] L.A. Te Raa, H.A. Dijkstra, Instability of the thermohaline ocean circulation on interdecadal time scales, J. Phys. Oceanogr. 32 (2002) 138–160.
[36] S.A. Thorpe, The Turbulent Ocean, Cambridge University Press, UK, 2005, 437pp.
[37] K.E. Trenberth, J.G. Olson, W.G. Large, A global ocean wind stress climatology based on ECMWF analyses, Technical Report, National Center for Atmospheric Research, Boulder, CO, USA, 1989.
[38] W. Weijer, H.A. Dijkstra, H. Oksuzoglu, F.W. Wubs, A.C. deNiet, A fully-implicit model of the global ocean circulation, J. Comp. Phys. 192 (2003) 452–470.
[39] C. Wunsch, R. Ferrari, Vertical mixing, energy and the general circulation of the oceans, Annu. Rev. Fluid Mech. 36 (2004) 281–314.