Journal of Computational Physics 398 (2019) 108900

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Exponential time differencing for mimetic multilayer ocean models

Konstantin Pieper[a,*], K. Chad Sockwell[a,b], Max Gunzburger[a]

[a] Florida State University, Department of Scientific Computing, 400 Dirac Science Library, Tallahassee, FL 32306, United States of America
[b] Los Alamos National Laboratory, P.O. Box 1663, T-3, MS-B216, Los Alamos, NM 87545, United States of America

ARTICLE INFO

Article history:
Received 15 December 2018
Received in revised form 13 July 2019
Accepted 14 August 2019
Available online 21 August 2019

Keywords:
Exponential time differencing
Ocean modeling
Mimetic methods
Time stepping

ABSTRACT

A framework for exponential time discretization of the multilayer rotating shallow water equations is developed in combination with a mimetic discretization in space. The method is based on a combination of existing exponential time differencing (ETD) methods and a careful choice of approximate Jacobians. The discrete Hamiltonian structure and conservation properties of the model are taken into account, in order to ensure stability of the method for large time steps and simulation horizons. In the case of many layers, further efficiency can be gained by a layer reduction which is based on the vertical structure of fast and slow modes. Numerical experiments on the example of a mid-latitude regional ocean model confirm long term stability for time steps increased by an order of magnitude over the explicit CFL, while maintaining accuracy for key statistical quantities.

1. Introduction

Despite their relevance in climate modeling, the numerical solution of the primitive equations used for the modeling of global or regional oceanic circulation remains challenging. This is due to the fact that the partial differential equations underlying the derivation of the primitive equations are of hyperbolic type, since physical diffusion terms are negligible at practically feasible grid resolutions. Concerning time discretization, a particular challenge lies in the presence of multiple time scales (due to, e.g., fast free-surface wave modes or locally refined meshes near coastal boundaries), which requires special schemes to take advantage of this structure. Otherwise, straightforward explicit integrators/Runge-Kutta schemes – which are usually very effective for problems of hyperbolic character – are restricted to an excessively small time step, degrading performance. Due to these requirements, specialized implicit methods based on the structure of the fast vertical mode have been developed; see, e.g. [1]. However, they can be affected by loss of accuracy due to high frequency error for large time steps. Moreover, scalability concerns arise on parallel computers due to the requirement of solving large linear systems. Subsequently, split-explicit methods [2,3] have been developed and applied with great success, which treat fast and slow modes with different explicit time discretization schemes. For an overview over the earlier developments in implicit and split-explicit time stepping methods for atmosphere and ocean models, we also refer to [4] and [5, Section 5].

Recently, exponential integrators (see, e.g., [6]), also called exponential time differencing methods (ETD), have gained attention in the context of circulation models [7–10]. Due to the presence of multiple time scales, ETD methods seem well suited to enable efficient large time step computations together with a reasonably accurate representation of high frequency

[*] Corresponding author.
E-mail addresses: kpieper@fsu.edu (K. Pieper), sockwell@lanl.gov (K.C. Sockwell), mgunzburger@fsu.edu (M. Gunzburger).

https://doi.org/10.1016/j.jcp.2019.108900
0021-9991/© 2019 Elsevier Inc. All rights reserved.

dynamics. For the purposes of this paper, we consider a simplified ocean model which still exhibits all of the difficulties mentioned above. Concretely, we restrict attention to the rotating shallow water equation (RSWE) with multiple horizontal layers, which corresponds to a vertical discretization of the primitive equations cast in an isopycnal vertical coordinate system. Concerning the spatial discretization, mimetic finite difference/finite volume (FD/FV) schemes have proven to be very effective here. Specifically, we work with the TRiSK scheme [11,12], which has many of the features of classical FD/FV schemes on Cartesian grids but additionally allows for the use of multi-resolution meshes. We emphasize that the resulting discretization can be set up to inherit the Hamiltonian structure of the underlying RSWE, which leads to exact energy conservation of the space discrete model. Based on this, we develop a framework for exponential time discretization which relies on a combination of existing exponential Runge-Kutta (ETD-RK) methods (see, e.g., [6,13]), developed for semi-linear equations of the form

$$\partial_t \boldsymbol{V} = \boldsymbol{F}[\boldsymbol{V}] = \boldsymbol{A}\boldsymbol{V} + \boldsymbol{r}[\boldsymbol{V}]$$

with an appropriate choice of the linear operator $\boldsymbol{A}$. Here, we prefer an approximation to the Jacobian of the forcing term over the full Jacobian $\boldsymbol{F}'$ (which would result in a Rosenbrock-ETD method), due to favorable properties concerning the implementation and structure of the linear operator and its numerical treatment. Physically, the proposed choice of $\boldsymbol{A}$ neglects the linearized advection and potential vorticity dynamics, which typically evolve on a relatively slow time scale, while still capturing the fast external (and internal) gravity waves. This leads to a class of explicit exponential Runge-Kutta methods which can take time steps significantly increased over an explicit integrator, while still maintaining stability and sufficient accuracy.

On the discrete level, the proposed class of linear operators inherits the Hamiltonian structure and corresponds to a skew-symmetric matrix with respect to an appropriate inner product. In turn, this enables the use of specialized efficient skew-Lanczos methods for the practical evaluation of the matrix exponentials and φ-functions, which are required for the implementation of an ETD method. Moreover, the matrix exponential $\exp(\Delta t \boldsymbol{A})$ maintains the linearized energy of the RSWE for all Δt, which improves numerical stability for large time steps. Since $\boldsymbol{A}$ describes the linearized free-surface and internal gravity waves around a reference configuration, we can additionally use the knowledge of the approximate structure of the fast and slow wave modes to further reduce the computational complexity. This is done by performing an additional projection of the linear operator onto the fast subspace, where we take special care to preserve the symmetry properties of $\boldsymbol{A}$. In a typical configuration of a global ocean model, the difference in the free-surface speed and speed of internal gravity waves is greater that an order of magnitude. Thereby, this projection enables computational savings proportional to the number of layers, while still capturing the free-surface dynamics in the linear operator. Thus, we obtain a faster method at the cost of additional restrictions on the maximal stable time step, due to the neglected internal modes.

In order to enable stable computations for very long simulation horizons (typically decades, in the context of climate models), additional diffusion terms have to be incorporated into the discrete model, in order to prevent a build-up of turbulent energy in the smallest (grid-level) scales. Here, we employ a variant of the classical biharmonic smoothing. Since optimal choices of the parameters of these diffusion terms are typically not stiff when compared to the fastest gravity waves, for efficiency we treat them explicitly, by adding them to the residual $\boldsymbol{r}$. However, since $\boldsymbol{A}$ does not take into account the dissipation, this can lead to a spurious build-up of kinetic energy in the smallest scales for large time step simulations (over the course of several months). To remedy this, we describe a simple method of adding a minimal amount of artificial high-frequency dissipation at minimal cost, by tuning the matrix φ-functions occurring in the method. The described procedure can be set up to maintain the formal order of accuracy of the scheme.

Finally, since exact mass conservation on the discrete level is an essential requirement for long-running simulations, we take care that the proposed methods fulfill this basic requirement. This is obtained by proving that the considered exponential integrators preserve linear invariants for an appropriate choice of $\boldsymbol{A}$.

This paper is structured as follows: In section 2 we introduce the concrete space and time continuous model and the underlying Hamiltonian structure. Section 3 summarizes the necessary details on the spatial discretization scheme. In section 4 the relevant background on exponential integrators, the efficient evaluation of the matrix exponential, and the proposed artificial dissipation strategy is given. Section 5 is devoted to the layer reduction strategy, which allows to take advantage of the vertical structure of the fast modes. In Section 6, we test the methods based on a simplified regional mid-latitude ocean model. In particular, we show that the methods deliver high order accuracy for large time step configurations, and investigate the effect of the artificial diffusion. Moreover, we perform decade long simulations with several configurations of the methods. Here, single trajectories can not be compared anymore due to the underlying chaotic structure of the model. However, we verify that key statistical quantities, such as mean flow and variance of the sea-surface height are accurately replicated in each simulation, while significant cost reductions are achieved over an explicit time discretization scheme.

2. Continuous equations

The governing equations used in this work are the multilayer rotational shallow water equations, which serve as proxy to the primitive equations in the MPAS-O model [3]. For the sake of readability, we first explain the single-layer model, and then the multilayer extension.

2.1. Single-layer rotating shallow water equations

The model equations for this work are defined on a spherical surface, with a variable bottom topography, and with multiple layers considered. We will start with the simple case of the single-layer equations. We denote by $\mathbb{S}^2$ the two-sphere with outward oriented surface-orthogonal unit vector $\hat{k}$, and by Ω an open sub-manifold with boundary $\partial\Omega$ and outer normal $\hat{n}$. The time variable is denoted by $t \in \mathbb{R}$. The single-layer rotating shallow water equations can now be expressed in terms of the fluid thickness $h\colon \mathbb{R} \times \Omega \to \mathbb{R}$ and the velocity $u\colon \mathbb{R} \times \Omega \to \mathbb{R}^3$ in the vector-invariant form as

$$\begin{cases} \partial_t h = -\nabla \cdot (hu) & \text{in } \Omega, \\ \partial_t u = -\nabla(K[u] + g(h+b)) - q[h,u]\hat{k} \times (hu) + \mathcal{G}[h,u] & \text{in } \Omega, \end{cases} \tag{2.1}$$

together with the constraint that the velocity should be tangential to the surface $u \cdot \hat{k} = 0$ in Ω, the no normal flow boundary condition $u \cdot \hat{n} = 0$ on $\partial\Omega$, and appropriate initial conditions on h and u. Here, $K[u] = |u|^2/2 = (u \cdot u)/2$ is the kinetic energy, $\hat{k} \times u$ the perpendicular velocity, $q[h,u] = (\hat{k} \cdot \nabla \times u + f)/h$ is the potential vorticity with f the Coriolis parameter. The bathymetry $b < 0$ encodes the bottom topography. The differential operators are defined in the canonical way on $\mathbb{S}^2$. The term $\mathcal{G}(h,u)$ contains additional forcing, arising either from wind or bottom drag or possible diffusion terms, which will be detailed later. For now, we only assume that $\hat{k} \cdot \mathcal{G}(h,u) = 0$, to ensure the consistency of the momentum equation with the constraint on the velocity.

The rotating shallow water equations (2.1) can also be given in a more abstract form, using a Hamiltonian framework. This also provides an abstract way to guarantee energy conservation (in the case $\mathcal{G} \equiv 0$). Consider the total energy over the domain as given by the Hamiltonian

$$\mathcal{H}[h,u] = \int_\Omega \left(hK[u] + \frac{g}{2}(h+b)^2 \right). \tag{2.2}$$

Furthermore, introduce a skew-symmetric operator $\mathcal{J}$ given formally by

$$\mathcal{J}[h,u] = \begin{pmatrix} 0 & -\nabla\cdot \\ -\nabla & -q[h,u]\hat{k}\times \end{pmatrix}.$$

In the following, we abbreviate the solution variables by $V = (h,u)$. Furthermore, we endow the solution space $\mathcal{X} = L^2(\Omega) \times L^2(\Omega,\mathbb{R}^3)$ by its canonical Hilbert space structure. We denote the inner product by

$$(V,W)_{\mathcal{X}} = \int_\Omega (v^h w^h + v^u \cdot w^u)\,\mathrm{d}x,$$

where $V = (v^h, v^u)$ and $W = (w^h, w^u)$ are elements of $\mathcal{X}$. The shallow water equations can then be formed using the functional derivative of $\mathcal{H}$ given as

$$\delta\mathcal{H}[V] = \frac{\delta\mathcal{H}}{\delta V}[V] = \begin{pmatrix} K[u] + g(h+b) \\ hu \end{pmatrix}, \tag{2.3}$$

which is identical to the Hilbert space gradient of the energy functional. In detail, for a perturbation $W = (w^h, w^u)$ the directional derivative of the Hamiltonian $\mathcal{H}$ (if it is well-defined) fulfills

$$\mathcal{H}'[V;W] = \frac{\mathrm{d}}{\mathrm{d}\tau}\mathcal{H}[V + \tau W]\Big|_{\tau=0} = \int_\Omega \Big((K[u] + g(h+b))w^h + hu \cdot w^u \Big) = \left(\frac{\delta\mathcal{H}}{\delta V}[V], W \right)_{\mathcal{X}}.$$

The first identity shows that (2.3) indeed gives the functional derivative from the calculus of variations, the second identity gives the interpretation as a gradient with respect to the space $\mathcal{X}$. Note that we abbreviate the functional derivative by $\delta\mathcal{H}[V]$, since the argument of differentiation is clear. In the following, we will also need the Jacobian of the functional derivative of the Hamiltonian (the Hessian), denoted by $\delta^2\mathcal{H}[V]$ where $\delta^2\mathcal{H}[V]W = \delta\mathcal{H}'[V;W]$.

Then, we interpret the boundary conditions as incorporated into the solution space, and obtain (2.1) in abstract form as

$$\partial_t V = \mathcal{J}[V]\delta\mathcal{H}[V] + \begin{pmatrix} 0 \\ \mathcal{G}[h,u] \end{pmatrix}. \tag{2.4}$$

In this work we will often appeal to the Hamiltonian framework to formulate the main ideas in a concise way. We note that the formal continuous description given above serves also serves as a motivation for the employed discrete scheme introduced below, which inherits the Hamiltonian structure. However, all of the developments can also be carried out without this formalism (and transferred to different discretization schemes under certain assumptions), but in a less direct way. Conversely, the main ideas apply also to different sets of equations, provided they can be written in terms of this framework.

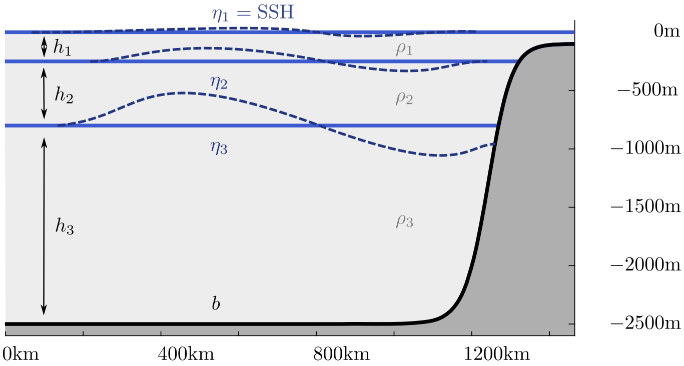

Fig. 1. Visualization of a three layer configuration on the SOMA bathymetry.

2.2. Extension to multiple layers

In order to model discrete stratification of bodies of water, multiple layers can be stacked on top of each other with each layer being modeled by its own set of shallow water equations. Although layered stratification does not fully represent continuous stratification, it is an efficient and robust model for describing ocean dynamics. In the multilayer system, each layer's density is set a priori and is considered the average density of the layer. The density is increasing with water depth in order to produce a stable configuration. Knowing the density ρ of each layer a priori gives a convenient set of Lagrangian coordinates, namely the isopycnal contours that separate the layers with different densities. This gives what is known as isopycnal coordinates.

The rotating shallow water equations (2.1) can easily be extended to deal with a body of water with multiple layers, using isopycnal coordinates. Consider a water basin that is separated into L layers, and posses L distinct densities ρ_k, with ρ_1 being the density for the top layer and $\rho_{k-1} < \rho_k$ for $1 \le k \le L$ and the zeroth density defined as $\rho_0 = 0$ for convenience (which represents the density of air, much smaller than that of water). The velocities u_k and h_k are now defined for each layer, and the corresponding layer coordinates η_k are defined by

$$\eta_{L+1} = b, \quad \eta_k[h] = b + \sum_{l=k}^{L} h_l, \quad k = 1, \dots, L.$$

A visualization is given in Fig. 1. In the following, we denote by u, h and η the vectors containing all layer variables, and by $V = (h, u)$ and $V_k = (h_k, u_k)$ the combined solution variable. The solution space is $\mathcal{X}^L = L^2(\Omega)^L \times L^2(\Omega, \mathbb{R}^3)^L$, which is endowed with the canonical inner product. The multilayer rotating shallow water equations for L layers are now given by

$$\begin{cases} \partial_t h_k = -\nabla \cdot (h_k u_k) & \text{in } \Omega\,, \\ \partial_t u_k = -\nabla(K[u_k] + (g/\rho_k) p_k[h]) - q[h_h, u_k]\hat{k} \times (h_k u_k) + \mathcal{G}_k(h, u) & \text{in } \Omega\,, \end{cases} \tag{2.5}$$

where the main modification with respect to the single-layer case is in the pressure term, which is defined in each layer as

$$p_k[h] = \rho_k \eta_{k+1}[h] + \sum_{l=1}^{k} \rho_l h_l = \rho_k \eta_k[h] + \sum_{l=1}^{k-1} \rho_l h_l.$$

The Hamiltonian framework can be extended for the multilayer case (see, e.g., [14]). The multilayer Hamiltonian $\mathcal{H}$ can be defined as

$$\mathcal{H}(h, u) = \sum_{k=1}^{L} \rho_k \int_{\Omega} \left(h_k K[u_k] + g h_k \left(\eta_{k+1}[h] + h_k/2 \right) \right) \mathrm{d}x\,. \tag{2.6}$$

The skew-symmetric operator $\mathcal{J}[h, u]$, consists of L copies of $\mathcal{J}[h_k, u_k]$, represented schematically by a block diagonal matrix

$$\mathcal{J}[V] = \mathrm{diag}_{k=1,\dots,L} \frac{1}{\rho_k} \mathcal{J}[h_k, u_k],$$

with the scaled single-layer operator on the diagonal. In a weak formulation this corresponds to

$$
\begin{aligned}
(W, \mathcal{J}[V]Y)_{\mathcal{X}^L} &= \sum_{k=1}^{L} \frac{1}{\rho_k} (W_k, \mathcal{J}_k[V_k]V_k)_{\mathcal{X}} \\
&= -\sum_{k=1}^{L} \frac{1}{\rho_k} \int_{\Omega} \left(w_k^h \nabla \cdot y_k^u + w_k^u \cdot \nabla y_k^h + q[h_k, u_k]\, w_k^u \cdot \hat{k} \times y_k^u \right) \mathrm{d}x,
\end{aligned}
$$

where $W = (w_k^h, w_k^u)_{k=1,\ldots,L}$ and $Y = (y_k^h, y_k^u)_{k=1,\ldots,L}$. The concrete form of (2.5) can now be derived from (2.4), as before. Note that the multilayer case contains the single-layer case for the special choice of one layer with arbitrary density.

Alternatively, another version of the multilayer Hamiltonian can be given as

$$
\widetilde{\mathcal{H}}(h, u) = \sum_{k=1}^{L} \int_{\Omega} \left(\rho_k h_k K[u_k] + \frac{g}{2} \Delta\rho_k \, (\eta_k[h])^2 \right), \tag{2.7}
$$

where $\Delta\rho_k = \rho_k - \rho_{k-1}$ for $k = 1, \ldots, L$. Further structure can be exposed by introducing the summation matrix T with entries

$$
T_{k,j} = \begin{cases} 1 & \text{if } k \le j, \\ 0 & \text{else.} \end{cases}
$$

It allows to express the layer coordinates as $\eta_k = b + (Th)_k$, where the matrix T operates on the layer variables in an obvious way. More abstractly, we also write $\eta = b + Th$. This allows to rewrite the last term in (2.7) as

$$
\frac{g}{2} \sum_{k=1}^{L} \Delta\rho_k (\eta_k[h])^2 = \frac{g}{2} \, \| b + Th \|_{\Delta\rho}^2 \, ,
$$

where $\|\eta\|_{\Delta\rho}^2 = \sum_{k=1}^{L} \Delta\rho_k \eta_k^2$ corresponds to a weighted Euclidean norm. Taking the functional derivative of this term, we obtain $g\, T^\top \operatorname{diag}(\Delta\rho)(Th + b) = g\widetilde{p}_k[h]$, and the corresponding pressure can be rewritten as:

$$
\widetilde{p}_k[h] = \rho_1 \eta_1[h] + \sum_{l=2}^{k} (\rho_l - \rho_{l-1}) \eta_l[h] = \rho_k \eta_k[h] + \sum_{l=1}^{k-1} \rho_l h_l = p_k[h].
$$

Thus, both Hamiltonians lead to the same pressure and it holds $\delta\mathcal{H} = \delta\widetilde{\mathcal{H}}$. Consequently, both Hamiltonians are equal up to a constant value (i.e. $\widetilde{\mathcal{H}} \equiv \mathcal{H} + const$).

Since $\Delta\rho_1 = \rho_1$ is typically much larger than $\Delta\rho_k = \rho_k - \rho_{k-1}$ for $k > 1$, the pressure differences induced by the free surface are much larger than the pressure differences stemming from perturbations of the internal layers. This gives rise to the well-known separation of vertical modes into a fast barotropic mode, and the remaining slow baroclinic modes; see, e.g., [15]. We give an independent exposition that is relevant for the development of the paper in the next section.

2.3. Linearization of the model and modes

We perform a linearized perturbation analysis of (2.5) for $\mathcal{G} = 0$, in order to understand the structure of the fastest modes of certain linearizations of (2.5). This will be used later to develop appropriate linear operators to be used for the exponential time integrators. For a more in-depth analysis of the fast barotropic mode arising in ocean models; cf. also [15]. Using the Hamiltonian formalism, the linearized equation for a perturbation $V = V^{\mathrm{ref}} + W$ can be written as

$$
\partial_t W = \mathcal{J}'[V^{\mathrm{ref}}; W] \delta\mathcal{H}[V^{\mathrm{ref}}] + \mathcal{J}[V^{\mathrm{ref}}] \delta^2 \mathcal{H}[V^{\mathrm{ref}}] \, W, \quad W(0) = W_0 \tag{2.8}
$$

by an application of the product rule, recalling the convention $\delta^2\mathcal{H}[V]W = (\delta\mathcal{H})'[V; W]$. The first term contains the derivatives of $\mathcal{J}$ with respect to V and is given for any perturbation W as

$$
\mathcal{J}'[V^{\mathrm{ref}}; W] = \operatorname{diag}_{k=1,\ldots,L} \frac{1}{\rho_k} \begin{pmatrix} 0 & 0 \\ 0 & -q'[V^{\mathrm{ref}}; W] \hat{k} \times \end{pmatrix} .
$$

This, in turn, contains the derivatives of the potential vorticity with respect to the solution variables, which are given as $q'[V; W] = -(\hat{k} \cdot \nabla \times u + f) h^{-2} \, w^h + (\hat{k} \cdot \nabla \times w^u)/h$, where $W = (w^h, w^u)$.

In the following, we linearize around a zero flow, i.e., $V^{\mathrm{ref}} = (h^{\mathrm{ref}}, u^{\mathrm{ref}})$ with $u^{\mathrm{ref}} = 0$. Then, (2.8) simplifies to

$$
\partial_t W = \mathcal{J}[V^{\mathrm{ref}}] \delta^2 \mathcal{H}[V^{\mathrm{ref}}] \, W, \qquad \text{for } V^{\mathrm{ref}} = (h^{\mathrm{ref}}, 0), \tag{2.9}
$$

since the derivative of $\mathcal{J}$ contains only entries in the lower right block (containing the derivatives of the potential vorticity), which are multiplied by the second entry of $\delta\mathcal{H}[V^{\mathrm{ref}}]$, which is given by $\rho_k u^{\mathrm{ref}} h^{\mathrm{ref}} = 0$. This system has again Hamiltonian structure, with a fixed $\mathcal{J}$-operator and a quadratic approximation to the energy:

$$\mathcal{J}^{\mathrm{ref}} = \mathcal{J}[V^{\mathrm{ref}}] = \mathrm{diag}_{k=1,\dots,L}\ \frac{1}{\rho_k}\begin{pmatrix} 0 & -\nabla\boldsymbol{\cdot} \\ -\nabla & -(f/h_k^{\mathrm{ref}})\,\hat{k}\,\boldsymbol{\times} \end{pmatrix},$$

$$\mathcal{H}^{\mathrm{ref}}(W) := \frac{1}{2}(W, \delta^2\mathcal{H}[V^{\mathrm{ref}}]\,W)_{\mathcal{X}^L} = \frac{1}{2}\sum_{k=1}^{L}\int_{\Omega}\left(\rho_k h_k^{\mathrm{ref}}\left|w_k^u\right|^2 + g\Delta\rho_k (Tw^h)_k^2\right),$$

where $W = (w^h, w^u)$. Thus, (2.9) reads as

$$\begin{cases} \partial_t w_k^h = -\nabla\boldsymbol{\cdot}\left(h_k^{\mathrm{ref}} w_k^u\right) & \text{in } \Omega\,, \\ \partial_t w_k^u = -(g/\rho_k)\nabla\left(T^\top \mathrm{diag}(\Delta\rho) T w^h\right)_k - f\,\hat{k}\,\boldsymbol{\times}\, w_k^u & \text{in } \Omega\,. \end{cases} \tag{2.10}$$

Under the simplifying assumption that the Coriolis term and the bathymetry b are flat, i.e. $f \equiv \mathit{const}$, $b \equiv \mathit{const}$, the eigenmodes of (2.10) can be easily computed for the stable reference configuration $h^{\mathrm{ref}} = h^0$. This reference configuration sets the heights h_k^0 in such a way that $\eta_k[h^0] = \max(b, \eta_k^0)$, where $\eta_k^0 \in \mathbb{R}_{<0}$ is a decreasing sequence of constant negative values with $\eta_1^0 = 0$, which is the total sea-surface height (SSH) at rest; see Fig. 1. Note that this configuration is just dependent upon the choice of the total layer volumes $\int_\Omega h_k$. Clearly, if the bathymetry is constant, the same holds for the reference heights h_k^0. Now the eigenmodes $W^\lambda = (h^\lambda, u^\lambda)$ associated to the imaginary eigenvalue $\lambda = \pm\mathrm{i}\,|\lambda|$ of $\mathcal{A}^{\mathrm{ref}} = \mathcal{J}^{\mathrm{ref}}\delta^2\mathcal{H}^{\mathrm{ref}}$ can be grouped into two sets: First, there are stationary (geostrophic) modes, which are obtained by setting $\lambda = 0$ in the eigenvalue equation. Secondly, there are instationary modes, which can be shown (by taking the $\nabla\boldsymbol{\cdot}$ and $\hat{k}\boldsymbol{\cdot}\nabla\boldsymbol{\times}$ of the momentum equation in (2.10) and algebraic manipulations) to solve

$$-g\,\mathrm{diag}(h^0/\rho)R\,\boldsymbol{\Delta}h^\lambda = \left(|\lambda|^2 - f^2\right)h^\lambda, \quad \text{and} \quad \left(f\,\hat{k}\,\boldsymbol{\times}\,\boldsymbol{\cdot} + \lambda\right)u^\lambda = -g\,\mathrm{diag}(1/\rho)R\nabla h^\lambda,$$

where $R = T^\top \mathrm{diag}(\Delta\rho)T$ is the matrix arising from the layer coupling through the pressure term. By using the fact that the layer operator and the spatial Laplacian commute, one can further decouple the above eigenvalue problem to obtain

$$|\lambda|^2 - f^2 = \mu^{\mathrm{vert}}\mu^{\mathrm{horiz}} > 0, \qquad h^\lambda = h^{\mu,\mathrm{vert}}h^{\mu,\mathrm{horiz}}$$
$$\text{with} \quad g\,\mathrm{diag}(h^0/\rho)R\,h^{\mu,\mathrm{vert}} = \mu^{\mathrm{vert}}h^{\mu,\mathrm{vert}} \quad \text{and} \quad -\boldsymbol{\Delta}h^{\mu,\mathrm{horiz}} = \mu^{\mathrm{horiz}}h^{\mu,\mathrm{horiz}},$$

where $h^{\mu,\mathrm{horiz}} \in L^2(\Omega)$ with $\int_\Omega h^{\mu,\mathrm{horiz}} = 0$ and $h^{\mu,\mathrm{vert}} \in \mathbb{R}^L$. Due to the fact that variations of the density $\Delta\rho$ are much smaller than density itself, the layer matrix

$$\mathrm{diag}(1/\rho)R = \mathrm{diag}(1/\rho)T^\top \mathrm{diag}(\Delta\rho)T = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \rho_1/\rho_2 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1/\rho_L & \rho_2/\rho_L & \cdots & 1 \end{pmatrix} \tag{2.11}$$

can be well-approximated by a rank-one matrix, e.g. the matrix with all entries equal to one. Therefore, the largest mode of the vertical eigenvalue problem and the associated u-mode fulfill approximately

$$h_k^{\mathrm{max,vert}} \approx \frac{h_k^0}{-b}, \quad u_k^{\mathrm{max,vert}} = (1/\rho_k)(Rh^{\mathrm{max,vert}})_k \approx 1.$$

This leads to the well-known (fast) barotropic free-surface mode with wave-speed $\sqrt{\mu^{\mathrm{max,vert}}} \approx \sqrt{-bg}$, corresponding to uniformly contracting and expanding layers, and approximately constant velocities in the vertical. The remaining modes are associated to (relatively slow) baroclinic modes, which approximately correspond to internal layer perturbations leaving the free surface constant.

Note that the vertical eigenvalue problem can be rewritten in terms of the generalized eigenvalue problem for the vertical η-mode $\eta^{\mu,\mathrm{vert}} = Th^{\mu,\mathrm{vert}}$ as

$$g\,\mathrm{diag}(\Delta\rho)\,\eta^{\mu,\mathrm{vert}} = \mu^{\mathrm{vert}}\,D^\top \mathrm{diag}(1/h^0)D\,\eta^{\mu,\mathrm{vert}},$$

where $D = T^{-1}$ is the discrete difference matrix with entries

$$T_{k,j}^{-1} = D_{k,j} = \begin{cases} 1 & \text{if } k = j, \\ -1 & \text{if } k = j-1, \\ 0 & \text{else.} \end{cases}$$

This formulation relates the vertical layer-modes to discrete approximations of solutions to the generalized eigenvalue problem from, e.g., [16,15].

3. Discretization by the TRiSK scheme

The rotating shallow water equations (2.1) and the multilayer version (2.5) will be discretized by a mimetic scheme. This ensures that properties of the continuous equation, such as energy conservation, are preserved on the discrete level. In this paper, we will employ the mimetic TRiSK scheme [11,12] (see also [17]). In the following, we briefly introduce a high level notation for employed differential operators that we will use to describe and analyze the time integration schemes. For a detailed exposition, we refer to the literature above.

Since the employed scheme is only developed in the literature for unbounded domains, we will in the following assume that $\Omega = \mathbb{S}^2$. This also simplifies the notation. Comments on the adaptation to a bounded domain can be found in Appendix A.

3.1. Discrete quantities and notation

The spatial discretization is defined on staggered C-grid that is comprised of spherical (centroidal) Voronoi tessellations, serving as the primal grid, and a Delaunay triangulation serving as the dual grid. The discrete quantities are defined at different locations on the grid, such as the edges, cell centers, and cell vertices. The edges of the grid will be denoted by $e \in \mathsf{E}$ (associated to the point of intersection of primal and dual grid edge x_e), primal cell grids will be denoted by $i \in \mathsf{I}$ (associated to cell centers x_i), and the primal cell vertices will be denoted with $v \in \mathsf{V}$ (associated to the circum-center x_v of a dual grid triangle, which is required to lie inside the triangle). Discrete quantities are denoted by bold vectors, and lie in corresponding cell, vertex, or edge space $X_\mathsf{I} = \mathbb{R}^{N_\mathsf{I}}$, $X_\mathsf{V} = \mathbb{R}^{N_\mathsf{V}}$, and $X_\mathsf{E} = \mathbb{R}^{N_\mathsf{E}}$, respectively. Discrete quantities are denoted in the following by bold symbols. The scheme is built upon the fundamental interpretation of these quantities as piece-wise constant on the primal or dual cell, and the edge degrees of freedom are associated to a flow across an interior primal edge (from one primal cell to another, in edge normal direction) or across a dual edge. Thus for a continuous vector field y, we have $\boldsymbol{y}_e \approx n_e \cdot y(x_e)$, where n_e is the geodesic normal to the primal edge e. Corresponding inner products on these spaces are given by

$$(\boldsymbol{y}, \boldsymbol{\varphi})_\mathsf{I} = \sum_{i \in \mathsf{I}} A_i \boldsymbol{y}_i \boldsymbol{\varphi}_i, \quad (\boldsymbol{y}, \boldsymbol{\varphi})_\mathsf{V} = \sum_{v \in \mathsf{V}} A_v \boldsymbol{y}_v \boldsymbol{\varphi}_v, \quad \text{and } (\boldsymbol{y}, \boldsymbol{\varphi})_\mathsf{E} = \sum_{e \in \mathsf{E}} A_e \boldsymbol{y}_e \boldsymbol{\varphi}_e,$$

respectively, where A_i denotes the area of a primal cell, A_v the area of a dual cell, and $A_e = l_e d_e$ the area of the square with side lengths given by the lengths of primal and dual edges (l_e and d_e). Note that the sum of the "edge areas" A_e corresponds to two times the volume of the domain, which is a peculiarity of this scheme, and corresponds to the fact that the velocities encode only one direction of the flow (edge normal).

The differential operators are built upon the fundamental relations that for any discrete variable $\boldsymbol{y} \in X_\mathsf{E}$, and test functions $\boldsymbol{\varphi} \in X_\mathsf{I}$ and $\boldsymbol{\psi} \in X_\mathsf{V}$ we have

$$(\boldsymbol{\varphi}, \underset{\mathsf{E}\to\mathsf{I}}{\nabla\cdot}\, \boldsymbol{y})_\mathsf{I} = \sum_{i \in \mathsf{I}} \sum_{e \in \mathsf{EoI}(i)} \boldsymbol{\varphi}_i n_{e,i} l_e \boldsymbol{y}_e, \quad (\boldsymbol{\psi}, \underset{\mathsf{E}\to\mathsf{V}}{(\hat{k}\cdot\nabla\times)}\, \boldsymbol{y})_\mathsf{V} = \sum_{v \in \mathsf{V}} \sum_{e \in \mathsf{EoV}(v)} \boldsymbol{\psi}_v t_{e,v} d_e \boldsymbol{y}_e,$$

where EoI and EoV denote the edges adjacent to each primal or dual cell, respectively, and $n_{e,i}$, $t_{e,v} \in \{+1, -1\}$ encodes the sign convention used for the direction of the edge normal velocity $\boldsymbol{y}_e$. For further details, we refer to [12]. We note that in [17], the schemes are built upon the integrated quantities $Y_e = l_e \boldsymbol{y}_e$ and $\tilde{Y}_e = d_e \boldsymbol{y}_e$, whereas we follow the convention used in [12].

Moreover, a discrete gradient is defined for a cell-wise quantity on the primal grid (across a primal edge): For each $\boldsymbol{\varphi} \in X_\mathsf{I}$ and test function $\boldsymbol{y} \in X_\mathsf{E}$ we set

$$(\underset{\mathsf{I}\to\mathsf{E}}{\nabla}\boldsymbol{\varphi}, \boldsymbol{y})_\mathsf{E} = -(\boldsymbol{\varphi}, \underset{\mathsf{E}\to\mathsf{I}}{\nabla\cdot}\, \boldsymbol{y})_\mathsf{I},$$

which mirrors the continuous integration by parts formula. Similarly, a perpendicular gradient $\nabla^\perp{}_{\mathsf{V}\to\mathsf{E}}$ can be defined on X_V (across a dual edge). Additionally, we define the interpolation operators

$$\{\boldsymbol{\varphi}\}_\mathsf{E} \in X_\mathsf{E}, \quad \{\boldsymbol{\psi}\}_\mathsf{E} \in X_\mathsf{E},$$

for $\boldsymbol{\varphi} \in X_\mathsf{I}$ and function $\boldsymbol{y} \in X_\mathsf{E}$ that average the values of the two adjacent primal and dual cells to the corresponding edge. Interpolation operators from edges to cells are defined by transposition as

$$(\{\boldsymbol{y}\}_\mathsf{I}, \boldsymbol{\varphi})_\mathsf{I} = (\boldsymbol{y}, \{\boldsymbol{\varphi}\}_\mathsf{E})_\mathsf{E}, \quad (\{\boldsymbol{y}\}_\mathsf{V}, \boldsymbol{\psi})_\mathsf{V} = (\boldsymbol{y}, \{\boldsymbol{\psi}\}_\mathsf{E})_\mathsf{E}.$$

We refer to [12] for the concrete expressions. Finally, a reconstruction of tangential velocities is needed (for the implementation of the perpendicular velocity $u^\perp$). This is realized by the reconstruction operator

$$\underset{\mathsf{E}\to\mathsf{E}}{\hat{k}\times} : X_\mathsf{E} \to X_\mathsf{E}, \quad \text{with} \quad (\underset{\mathsf{E}\to\mathsf{E}}{\hat{k}\times}\, \boldsymbol{y})_e \approx t_e \cdot y(x_e)$$

for any $\boldsymbol{y} \in X_{\mathsf{E}}$ representing a continuous vector field y, with $t_e = \hat{k} \times n_e$ the tangent to the primal edge e. We refer to [11] for a derivation and the concrete expressions, which in particular ensure the skew-symmetry of the reconstruction operator $\hat{k}\underset{\mathsf{E}\to\mathsf{E}}{\times}$ on the edge space X_{E}. For convenience, the specific form of all required operators is also summarized in Appendix A.

3.2. Discrete multilayer equations

We describe the scheme for the general multilayer case, which contains the single-layer rotating shallow water equations as a special case. The prognostic variables of the equations are the fluid heights $\boldsymbol{h}_k \in X_{\mathsf{I}}$ and the velocities $\boldsymbol{u}_k \in X_{\mathsf{E}}$, where the degree of freedom for the edge encodes the (point-wise) velocity in primal cell normal direction. Diagnostic quantities are the kinetic energy and the potential vorticity, defined by:

$$\begin{aligned} &\boldsymbol{K}[\boldsymbol{u}] = (1/2)\{\boldsymbol{u} * \boldsymbol{u}\}_{\mathsf{I}} = \{\boldsymbol{u}^2/2\}_{\mathsf{I}} && \text{(kinetic energy in } X_{\mathsf{I}}\text{)},\\ &\boldsymbol{q}[\boldsymbol{h},\boldsymbol{u}] = \left((\hat{k}\cdot\underset{\mathsf{E}\to\mathsf{V}}{\nabla\times})\,\boldsymbol{u} + \boldsymbol{f}\right)/\{\boldsymbol{h}\}_{\mathsf{V}} && \text{(potential vorticity in } X_{\mathsf{V}}\text{)}. \end{aligned}$$

Here, $*$ denotes the point- or entry-wise product (Hadamard product), and / the point-wise division and $\boldsymbol{f} \in X_{\mathsf{V}}$ is an interpolant of the Coriolis parameter.

The discrete equations are now given as:

$$\begin{cases} \partial_t \boldsymbol{h}_k = -\underset{\mathsf{E}\to\mathsf{I}}{\nabla\cdot}\left(\{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{u}_k\right) & \text{in } X_{\mathsf{I}},\\ \partial_t \boldsymbol{u}_k = \boldsymbol{Q}\left[\boldsymbol{h}_k,\boldsymbol{u}_k\right]\left(\{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{u}_k\right) - \underset{\mathsf{I}\to\mathsf{E}}{\nabla}\left(\boldsymbol{K}[\boldsymbol{u}_k] + (g/\rho_k)\,\boldsymbol{p}_k[\boldsymbol{h}]\right) + \boldsymbol{G}_k[\boldsymbol{h},\boldsymbol{u}] & \text{in } X_{\mathsf{E}}. \end{cases} \tag{3.1}$$

Here, the pressure is computed as in the continuous case as

$$\boldsymbol{p}_k[\boldsymbol{h}] = \rho_k \boldsymbol{\eta}_k[\boldsymbol{h}] + \sum_{l=1}^{k-1} \rho_l \boldsymbol{h}_l, \quad \boldsymbol{\eta}_k[\boldsymbol{h}] = \boldsymbol{b} + \sum_{l=k}^{L} \boldsymbol{h}_k,$$

where $\boldsymbol{b} \in X_{\mathsf{I}}$ is an interpolant of the bathymetry. The operator $\boldsymbol{Q}\,[\cdot,\cdot]\colon X_{\mathsf{E}} \to X_{\mathsf{E}}$ is defined as

$$\boldsymbol{Q}\,[\boldsymbol{h}_k,\boldsymbol{u}_k]\boldsymbol{y} = \frac{1}{2}\left(\{\boldsymbol{q}[\boldsymbol{h}_k,\boldsymbol{u}_k]\}_{\mathsf{E}} * \left(\hat{k}\underset{\mathsf{E}\to\mathsf{E}}{\times}\boldsymbol{y}\right) + \hat{k}\underset{\mathsf{E}\to\mathsf{E}}{\times}\left(\{\boldsymbol{q}[\boldsymbol{h}_k,\boldsymbol{u}_k]\}_{\mathsf{E}} * \boldsymbol{y}\right)\right),$$

where $\boldsymbol{y} \in X_{\mathsf{E}}$ is a discrete flux. The construction of this operator ensures energy conservation; see [12]. In terms of the Hamiltonian framework, this follows from the fact that the operator is skew-symmetric: for any $\boldsymbol{w},\boldsymbol{v} \in X_{\mathsf{E}}$ we have $(\boldsymbol{v},\ \boldsymbol{Q}\,[\boldsymbol{V}_k]\boldsymbol{w})_{\mathsf{E}} = -(\boldsymbol{Q}\,[\boldsymbol{V}_k]\boldsymbol{v},\boldsymbol{w})_{\mathsf{E}}$, using the skew-symmetry of $\hat{k}\underset{\mathsf{E}\to\mathsf{E}}{\times}$.

Energy conservation follows directly by introducing a discrete Hamiltonian framework for (3.1). We define the combined solution variable as $\boldsymbol{V} = (\boldsymbol{h},\boldsymbol{u}) \in X^L = X_{\mathsf{I}}^L \times X_{\mathsf{E}}^L$ analogous to the continuous case. It is endowed with the discrete inner product

$$(\boldsymbol{V},\boldsymbol{W})_{X^L} = \sum_{k=1}^{L} (\boldsymbol{h}_k,\boldsymbol{w}_k^h)_{\mathsf{I}} + (\boldsymbol{u}_k,\boldsymbol{w}_k^u)_{\mathsf{E}},$$

where $\boldsymbol{W} = (\boldsymbol{w}_k^h,\boldsymbol{w}_k^u)_{k=1,\dots,L}$. The discrete Hamiltonian has the form

$$\boldsymbol{H}[\boldsymbol{V}] = \frac{1}{2}\sum_{k=1}^{L}\left(\rho_k(\boldsymbol{h}_k,\boldsymbol{K}[\boldsymbol{u}_k])_{\mathsf{I}} + g\Delta\rho_k\,(\boldsymbol{\eta}_k[\boldsymbol{h}],\boldsymbol{\eta}_k[\boldsymbol{h}])_{\mathsf{I}}\right). \tag{3.2}$$

Mirroring the continuous case, the functional derivative of the Hamiltonian fulfills the identity

$$(\boldsymbol{W},\delta\boldsymbol{H}[\boldsymbol{V}])_{X^L} = \boldsymbol{H}'[\boldsymbol{V};\boldsymbol{W}] = \sum_{k=1}^{L}\left((\boldsymbol{w}_k^h,\rho_k\boldsymbol{K}[\boldsymbol{u}_k] + g\,\boldsymbol{p}_k[\boldsymbol{h}])_{\mathsf{I}} + \rho_k(\boldsymbol{w}_k^u,\{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{u}_k)_{\mathsf{E}}\right),$$

using that $(\boldsymbol{h}_k,\boldsymbol{K}'[\boldsymbol{u}_k;\boldsymbol{w}_k^u])_{\mathsf{I}} = (\boldsymbol{w}_k^u,\{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{u}_k)_{\mathsf{E}}$ and the definition of $\boldsymbol{\eta}$. Thus, we can write

$$\delta\boldsymbol{H}[\boldsymbol{V}] = \begin{pmatrix} \rho_k\boldsymbol{K}[\boldsymbol{u}_k] + g\,\boldsymbol{p}_k[\boldsymbol{h}] \\ \rho_k\{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{u}_k \end{pmatrix}_{k=1,2,\dots,L}. \tag{3.3}$$

From the concrete form of the equations as given above, one can infer the discrete analogue of the operator $\mathcal{J}$, which is given by

$$\boldsymbol{J}[\boldsymbol{V}] = \operatorname{diag}_{k=1,2,\dots,L} \frac{1}{\rho_k} \begin{pmatrix} 0 & -\nabla\boldsymbol{\cdot}_{\mathsf{E}\to\mathsf{I}} \\ -\nabla_{\mathsf{I}\to\mathsf{E}} & \boldsymbol{Q}\,[\boldsymbol{h}_k, \boldsymbol{u}_k] \end{pmatrix}. \tag{3.4}$$

Using the discrete identities for $\nabla_{\mathsf{I}\to\mathsf{E}}$ and $\nabla\boldsymbol{\cdot}_{\mathsf{E}\to\mathsf{I}}$, the definition of $\boldsymbol{Q}$, and the skew-symmetry of $\hat{k}\boldsymbol{\times}_{\mathsf{E}\to\mathsf{E}}$, the skew symmetry of $\boldsymbol{J}$ can be verified by considering a discrete weak formulation. Together, this shows that (3.1) can be described by

$$\partial_t \boldsymbol{V} = \boldsymbol{J}[\boldsymbol{V}]\delta\boldsymbol{H}(\boldsymbol{V}) + \begin{pmatrix} 0 \\ \boldsymbol{G}[\boldsymbol{V}] \end{pmatrix}, \tag{3.5}$$

which directly yields energy conservation in the case $\boldsymbol{G}[\boldsymbol{V}] = 0$. Additional source and dissipation terms can be added to the momentum equation in the term $\boldsymbol{G}$. We detail some particular choices in Appendix A.1.

4. Exponential time integration

Exponential integrators or exponential time differencing methods (ETD) are a special class of time integration methods; see [6] and the references therein. We briefly summarize the relevant content for this manuscript, in the context of the discrete system introduced in (3.5). We will focus only on the case without forcing or dissipation, $\boldsymbol{G} \equiv 0$. Additional forcing terms can be easily added to the following derivation, but are omitted from the derivation, since they are usually much less stiff than the core ocean dynamics, and will be added back at the end.

4.1. Exponential integrators

Exponential integrators are based on a splitting of the forcing term into a linear part, and a remainder. Denote by $\boldsymbol{V}_n \approx \boldsymbol{V}(t_n)$ the current solution at time t_n, $n = 0, 1, 2, \dots$, and write

$$\partial_t \boldsymbol{V} = \boldsymbol{F}[\boldsymbol{V}] = \boldsymbol{A}_n \boldsymbol{V} + \boldsymbol{r}_n[\boldsymbol{V}] \tag{4.1}$$

with the nonlinear remainder defined by

$$\boldsymbol{r}_n[\boldsymbol{V}] = \boldsymbol{F}[\boldsymbol{V}] - \boldsymbol{A}_n \boldsymbol{V}.$$

Such a splitting is natural for many problems, where $\boldsymbol{F}$ is given as the sum of a stiff linear, and a nonlinear term, e.g., semilinear parabolic problems [13]. However, for the present case a suitable choice of $\boldsymbol{A}_n$ is less obvious. Another point of view is to perform an affine linear expansion of $\boldsymbol{F}$ around $\boldsymbol{V}_n$, which leads to

$$\partial_t \boldsymbol{V} = \boldsymbol{F}[\boldsymbol{V}] = \boldsymbol{F}[\boldsymbol{V}_n] + \boldsymbol{A}_n(\boldsymbol{V} - \boldsymbol{V}_n) + \boldsymbol{R}_n[\boldsymbol{V}] \tag{4.2}$$

with the nonlinear residual defined by

$$\boldsymbol{R}_n[\boldsymbol{V}] = \boldsymbol{F}[\boldsymbol{V}] - \boldsymbol{F}[\boldsymbol{V}_n] - \boldsymbol{A}_n(\boldsymbol{V} - \boldsymbol{V}_n) = \boldsymbol{r}_n[\boldsymbol{V}] - \boldsymbol{r}_n[\boldsymbol{V}_n].$$

Clearly, both forms only differ in the constant term $\boldsymbol{r}_n[\boldsymbol{V}_n]$ and are thus very similar. However, the second form immediately suggests to choose $\boldsymbol{A}_n = \boldsymbol{F}'[\boldsymbol{V}_n]$, the Jacobian of $\boldsymbol{F}$, which corresponds to a Taylor expansion in (4.2). This leads to the development of Rosenbrock type methods.

The idea behind ETD methods, more specifically exponential RK methods, is to treat the (affine) linear and nonlinear part in different ways: the linear term involving $\boldsymbol{A}_n$ will be treated exactly, using matrix exponentials, and only the remainder will be approximated by internal stages of the (exponential) RK method. In particular, an affine linear problem (i.e. when $\boldsymbol{R}_n \equiv 0$) will be solved exactly under reasonable assumptions on the methods. By now, there is a well-developed theory of order conditions for such methods, and several classes of appropriate methods are known; see, e.g., the overview in [6]. If $\boldsymbol{A}_n$ can be chosen in a way that the residual is significantly less stiff than the linear part, then the CFL conditions that limit the time step size of explicit methods are less restrictive for exponential RK methods. However, while the Jacobian $\boldsymbol{F}'[\boldsymbol{V}_n]$ always constitutes a mathematically optimal choice in terms of stiffness reduction and accuracy, it is not necessarily the best choice in terms of practical performance.

In the specific setting with $\boldsymbol{G} \equiv 0$, due to the product structure of $\boldsymbol{F}[\boldsymbol{V}] = \boldsymbol{J}[\boldsymbol{V}]\delta\boldsymbol{H}[\boldsymbol{V}]$ we have

$$\boldsymbol{F}'[\boldsymbol{V}_n; \boldsymbol{W}] = \boldsymbol{J}'[\boldsymbol{V}_n; \boldsymbol{W}]\delta\boldsymbol{H}[\boldsymbol{V}_n] + \boldsymbol{J}[\boldsymbol{V}_n]\delta^2\boldsymbol{H}[\boldsymbol{V}_n]\boldsymbol{W}; \tag{4.3}$$

by the product rule; cf. section 2.3. The concrete expressions for the Jacobians of $\boldsymbol{J}$ and $\delta\boldsymbol{H}$ on the discrete level are given in Appendix B. Instead of the Jacobian at the current time step, we will consider choices of linear operator that correspond to Jacobians that are evaluated at a reference configuration $\boldsymbol{V}^{\text{ref}} = (\boldsymbol{h}^{\text{ref}}, \boldsymbol{0})$. This leads to

$$\boldsymbol{F}'[\boldsymbol{V}^{\text{ref}}] = \boldsymbol{J}[\boldsymbol{V}^{\text{ref}}]\delta^2\boldsymbol{H}[\boldsymbol{V}^{\text{ref}}], \tag{4.4}$$

since the first term in (4.3) is zero (cf. section 2.3). Note that the reference point can be chosen differently in each time step, in order to take updated height variables into account. This leads to a choice of $\boldsymbol{A}_n = \boldsymbol{A}_n^{\text{ref}} = \boldsymbol{F}'[\boldsymbol{V}_n^{\text{ref}}]$, which leads to a structurally simpler and computationally more efficient linear operator, at the cost of an increased approximation error. As we will demonstrate, in the context of global ocean models, this still captures enough of the fast dynamics to enable stable and accurate simulations with large time steps. We note that (4.4) has again Hamiltonian structure, which can be exploited in computations. Additional approximations of $\boldsymbol{A}^{\text{ref}}$, which further decrease the cost of the practical evaluation in the multilayer case, but keep the underlying structure of the linear operator intact, will be discussed in section 5. Note that, if we were to employ Rosenbrock methods, the Jacobian of any additional nonlinear terms occurring in $\boldsymbol{G}[\boldsymbol{V}]$ would need to be included in $\boldsymbol{A}_n$. Since we use approximate Jacobians, any additional forces that are not stiff can be neglected in $\boldsymbol{A}_n$.

Finally, we note that, on the continuous level, the splitting (4.2) with the linear operator (4.4) introduced above corresponds (up to constant terms) to the splitting of the original equations of the form

$$\begin{aligned}\partial_t h_k + \nabla \cdot \left(h_k^{\text{ref}} u_k\right) &= -\nabla \cdot \left(\left(h_k - h_k^{\text{ref}}\right) u_k\right),\\ \partial_t u_k + \nabla(1/\rho_k) p_k[h] + f\,\hat{k} \times u_k &= -u_k \cdot \nabla u_k + \mathcal{G}_k(h, u).\end{aligned}$$

Here, we have used the vector identity $u \cdot \nabla u = \nabla(|u|^2/2) + (\hat{k} \cdot \nabla \times u)\,(\hat{k} \times u)$. In the splitting above, the terms on the left are linear (affine linear in the case of the pressure) and correspond to a multilayer rotating wave equation, and the remaining terms on the right are nonlinear advection terms. Roughly speaking, the former will always be solved exactly in theory and treated with matrix exponentials in practice, whereas the latter will be approximated by the internal stages of an exponential Runge-Kutta method. Therefore, a method based on (4.4) can be expected to have no time step restrictions associated to the wave phenomena, whereas it would likely still be subject to CFL conditions associated to the advective processes and other physics contained in $\mathcal{G}$.

4.1.1. Approximation of the residual

To obtain an exponential integrator, the variation of constants formula is applied to the continuous equation (4.1) to obtain the solution at time $t_{n+1} = t_n + \Delta t$ as

$$\begin{aligned}\boldsymbol{V}(t_{n+1}) &= \exp(\Delta t\,\boldsymbol{A}_n)\boldsymbol{V}_n + \int_0^{\Delta t} \exp((\Delta t - \tau)\boldsymbol{A}_n)\,\boldsymbol{r}_n[\boldsymbol{V}(t_n + \tau)]\,\mathrm{d}\tau\\ &= \boldsymbol{V}_n + \int_0^{\Delta t} \exp((\Delta t - \tau)\boldsymbol{A}_n)\,(\boldsymbol{F}[\boldsymbol{V}_n] + \boldsymbol{R}_n[\boldsymbol{V}(t_n + \tau)])\,\mathrm{d}\tau.\end{aligned} \tag{4.5}$$

For further details on the derivations in this section we refer to [6]. This formula for the exact solution is further approximated by replacing the residual term (which still depends on the unknown solution) by a polynomial in time given as

$$\boldsymbol{R}_n[\boldsymbol{V}(t_n + \tau)] \approx \sum_{s=2}^{S} \frac{\tau^{s-1}}{(\Delta t)^{s-1}(s-1)!}\boldsymbol{b}_{n,s},$$

where the coefficients $\boldsymbol{b}_{n,s}$ should approximate the derivatives $\mathrm{d}^{s-1}/\mathrm{d}\tau^{s-1}\,\boldsymbol{R}_n[\boldsymbol{V}(t_n + \tau)]|_{\tau=0}$. Note that, since $\boldsymbol{R}_n[\boldsymbol{V}_n] = 0$, the constant term in the polynomial can be omitted. For exponential Runge-Kutta methods these coefficients will be determined as linear combinations of the residual $\boldsymbol{R}_n$ evaluated at the internal stages of the method; see [6].

Consequently, by inserting the above approximation into the solution formula (4.5) (see also Proposition 4.1 below) we obtain one time step of the underlying method as:

$$\boldsymbol{V}(t_{n+1}) \approx \boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t\left(\varphi_1(\Delta t\boldsymbol{A}_n)\boldsymbol{F}[\boldsymbol{V}_n] + \sum_{s=2}^{S}\varphi_s(\Delta t\boldsymbol{A}_n)\boldsymbol{b}_{n,s}\right), \tag{4.6}$$

where the φ-functions are defined as

$$\varphi_s(z) = \int_0^1 \exp\left((1-\sigma)z\right)\frac{\sigma^{s-1}}{(s-1)!}\,\mathrm{d}\sigma = \sum_{k=0}^{\infty}\frac{z^k}{(k+s)!} \quad \text{for } z \in \mathbb{C},\ s = 1, 2, \dots. \tag{4.7}$$

In the case $s = 0$, we set $\varphi_0(\cdot) = \exp(\cdot)$. Note that the above definition of φ_s generalizes to matrix arguments either by replacing z by a matrix in the above definition, or by applying the matrix functional calculus. Based on the construction, there is a simple correspondence between φ-functions and inhomogeneous linear equations; cf., e.g., [18]:

Proposition 4.1. *Let* $\boldsymbol{x} = \sum_{s=0}^{S} \varphi_s(\Delta t \boldsymbol{A})\boldsymbol{b}_s$ *for arbitrary* $\boldsymbol{b}_s$, $s = 0, 1, \ldots, S$. *Then it holds* $\boldsymbol{x} = \boldsymbol{w}(\Delta t)$, *which is the terminal value of the solution to the linear differential equation*

$$\begin{cases} \partial_t \boldsymbol{w}(\tau) = \boldsymbol{A}\boldsymbol{w}(\tau) + \sum_{s=1}^{S} \dfrac{\tau^{s-1}}{\Delta t^{s-1}(s-1)!}\boldsymbol{b}_s, \qquad (0 < \tau < \Delta t) \\ \boldsymbol{w}(0) = \boldsymbol{b}_0. \end{cases} \tag{4.8}$$

Certainly, the efficient computation of these matrix functions is important for the practical success of ETD methods. Since even for sparse $\boldsymbol{A}$ the matrix $\varphi_s(\Delta t \boldsymbol{A})$ is generally a full matrix, they can not be assembled in practice in the large-scale context. Therefore, we will employ iterative methods; see section 4.2.

4.1.2. Example methods

In the following, we will briefly present specific exponential integrators employed in this work. In the simplest case, the residual is simply neglected, and the exponential Euler method is obtained as

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t \varphi_1(\Delta t \boldsymbol{A}_n)\boldsymbol{F}[\boldsymbol{V}_n]. \tag{4.9}$$

In the case that $\boldsymbol{A}_n$ is only an approximation to the Jacobian, this method is only first-order accurate in Δt, and not attractive for practical computations. We remark that this method is second-order for $\boldsymbol{A}_n = \boldsymbol{F}'[\boldsymbol{V}_n]$, which highlights the fact that Rosenbrock-ETD methods have different order conditions.

For the case of approximate Jacobians, e.g., for (4.4), a family of two-stage, second-order methods (fulfilling the stiff order conditions) is given by the one-parameter family

$$\begin{aligned} \boldsymbol{v}_{n,2} &= \boldsymbol{V}_n + c_2 \Delta t \varphi_1(c_2 \Delta t \boldsymbol{A}_n)\boldsymbol{F}[\boldsymbol{V}_n], \\ \boldsymbol{V}_{n+1} &= \boldsymbol{V}_n + \Delta t \varphi_1(\Delta t \boldsymbol{A}_n)\boldsymbol{F}[\boldsymbol{V}_n] + (\Delta t / c_2)\varphi_2(\Delta t \boldsymbol{A}_n)\boldsymbol{R}_n[\boldsymbol{v}_{n,2}], \end{aligned} \tag{4.10}$$

for the parameter $c_2 \in (0, 1]$; see [13]. In the case $c_2 = 1$, we obtain the exponential version of Heun's method, while $c_2 = 2/3$ corresponds to Ralston's method. More details and the description of a three stage third-order method that will be used in the computational experiments are given in Appendix C. A general discussion of higher order methods can be found in, e.g., [13,6].

4.2. Approximation of the matrix functions

As previously mentioned, the most challenging aspect of ETD methods is efficiently evaluating the matrix functions φ_s. This challenge made ETD methods computational infeasible for many years after their discovery due to a lack of efficient matrix function evaluation methods [19]. However, in recent years more efficient ways to approximate φ_s have been found such as Krylov subspace projections [20], which can be combined with sub-stepping algorithms [18], or instead found with Leja-point interpolation [21], or Chebyshev polynomial approximations [22]. Due to the optimality of the matrix polynomials produced by Krylov methods, it is likely that these methods will provide an advantage over the other approximation methods (requiring only matrix vector products), therefore they will be the focus from this point forward. Methods based on rational approximation are advantageous from a theoretical standpoint and also promising from a practical standpoint [23]. However, in the context of the spatial scheme employed in this work, an efficient parallel way of solving the large sparse linear systems remains challenging.

4.2.1. Polynomial Krylov methods

Krylov subspace methods, or Krylov methods, provide an efficient way to approximate matrix functions. This is done by projecting the matrix into a Krylov subspace and then evaluating the function in a much smaller space than the original. Another benefit of Krylov methods is the fact that the matrix itself is never explicitly required throughout the method, only it's action upon single vectors.

The Krylov subspace of dimension M for a matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and a vector $\boldsymbol{b} \in \mathbb{R}^N$ is defined as

$$K_M(\boldsymbol{A}, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{M-1}\boldsymbol{b}\} = \{\, p(\boldsymbol{A})\boldsymbol{b} \mid p \in \mathcal{P}_{M-1}\} \tag{4.11}$$

Essentially, Krylov methods find the optimal polynomial to approximate a matrix function applied to a matrix applied to a single vector. In the case of the linear systems arising in linearly implicit methods, the Krylov method approximates a rational function to form $\boldsymbol{x} = (\mathrm{Id} + \Delta t \boldsymbol{A})^{-1}\boldsymbol{b}$; for ETD the expressions $\varphi_s(\Delta t \boldsymbol{A})\boldsymbol{b}$ are approximated. For this purpose, an orthonormal basis of $K_M(\boldsymbol{A}, \boldsymbol{b})$ is constructed, which is typically done by the Arnoldi process. The approximation of matrix functions by Krylov methods is well documented in the literature. However, we will employ an inner product induced by another matrix, which is usually not discussed. Thus, we briefly summarize the necessary extensions for this case.

In this work, we will mostly employ linear operators of the form (4.4), which have the product structure

$$\boldsymbol{A} = \boldsymbol{J}\,\delta^2 \boldsymbol{H}. \tag{4.12}$$

Additionally the matrix $\boldsymbol{J}$ and $\delta^2\boldsymbol{H}$ have symmetry properties with respect to the inner product of the space $X^L = X_\mathsf{I}^L \times X_\mathsf{E}^L$, namely

$$(\boldsymbol{W},\ \boldsymbol{J}\boldsymbol{V})_{X^L} = -(\boldsymbol{J}\boldsymbol{W},\ \boldsymbol{V})_{X^L}, \quad (\boldsymbol{W},\ \delta^2\boldsymbol{H}\,\boldsymbol{V})_{X^L} = (\delta^2\boldsymbol{H}\,\boldsymbol{W},\ \boldsymbol{V})_{X^L}$$

for all $\boldsymbol{V}, \boldsymbol{W} \in X^L$. To express this in terms of linear algebra, we introduce the symmetric and diagonal mass matrix $\boldsymbol{M}_{X^L}$ of the solution space X^L, containing L copies of the cell and edge areas on the diagonal. In terms of linear algebra, we can now reformulate the symmetry properties above as

$$\boldsymbol{M}_{X^L}\boldsymbol{J} = -\boldsymbol{J}^\top \boldsymbol{M}_{X^L}, \quad \boldsymbol{M}_{X^L}\delta^2\boldsymbol{H} = \delta^2\boldsymbol{H}^\top \boldsymbol{M}_{X^L}, \tag{4.13}$$

where $\cdot^\top$ denotes the transpose. This yields the skew-symmetry of $\boldsymbol{A}$ with respect to the inner product induced by the symmetric matrix induced by the second variation of the Hamiltonian.

Proposition 4.2. *The operator $\boldsymbol{A}$ given in (4.12) with (4.13) fulfills*

$$\boldsymbol{M}_H\boldsymbol{A} = -\boldsymbol{A}^\top \boldsymbol{M}_H \quad \textit{where } \boldsymbol{M}_H = \boldsymbol{M}_{X^L}\delta^2\boldsymbol{H}.$$

In order to take advantage of this symmetry, we will describe the following orthogonalization procedures for a general operator $\boldsymbol{A}$ with respect to the inner product and norm

$$(\boldsymbol{x}, \boldsymbol{y})_\mathbb{M} = \boldsymbol{x}^\top \mathbb{M}\boldsymbol{y}, \quad \|\boldsymbol{x}\|_\mathbb{M} = \sqrt{\boldsymbol{x}^\top \mathbb{M}\boldsymbol{x}}$$

induced by another symmetric matrix $\mathbb{M}$, which can be chosen as either $\boldsymbol{M}_X$ (corresponding to the L^2 space X^L) or $\boldsymbol{M}_H$ (corresponding to a norm induced by a quadratic approximation of the Hamiltonian).

The orthonormal basis vectors $\boldsymbol{v}_m$, $m = 1, \ldots, M$ can be found through the iterative Arnoldi process:

$$\begin{aligned} \widetilde{\boldsymbol{v}}_{m+1} &= \boldsymbol{A}\boldsymbol{v}_m - \sum_{j=1}^{m} \boldsymbol{h}_{j,m}\boldsymbol{v}_j, & \boldsymbol{h}_{j,m} &= (\boldsymbol{v}_j, \boldsymbol{A}\boldsymbol{v}_m)_\mathbb{M}, \quad j = 1, 2, \ldots, m\,, \\ \boldsymbol{v}_{m+1} &= \frac{1}{\boldsymbol{h}_{m+1,m}}\widetilde{\boldsymbol{v}}_{m+1}, & \boldsymbol{h}_{m+1,m} &= \|\widetilde{\boldsymbol{v}}_{m+1}\|_\mathbb{M}\,, \end{aligned}$$

where $\boldsymbol{v}_1 = \boldsymbol{b}$. The Arnoldi process can be collectively given by the Arnoldi decomposition

$$\begin{aligned} \boldsymbol{A}\mathbb{V}_M &= \mathbb{V}_M\mathbb{H}_M + \boldsymbol{h}_{M+1,M}\boldsymbol{v}_{M+1}\boldsymbol{e}_M^\top = \mathbb{V}_{M+1}\widetilde{\mathbb{H}}_M\,, \\ \widetilde{\mathbb{H}}_M &= \begin{pmatrix} \mathbb{H}_M \\ \boldsymbol{h}_{M+1,M}\boldsymbol{e}_M^\top \end{pmatrix} \in \mathbb{R}^{(M+1)\times M}\,, \end{aligned}$$

where $\mathbb{V}_M \in \mathbb{R}^{N\times M}$ contains the M orthogonal basis vectors, i.e., $\mathbb{V}_M^\top \mathbb{M}\mathbb{V}_M = \mathrm{Id}_M$, $\boldsymbol{e}_M$ is the M-th canonical basis vector with entries $\boldsymbol{e}_{M,i} = \delta_{M,i}$, and $\mathbb{H}_M \in \mathbb{R}^{M\times M}$ is the Hessenberg matrix given by $\mathbb{H}_M = \mathbb{V}_M^\top \mathbb{M}\boldsymbol{A}\mathbb{V}_M$. Finally, the Krylov approximation of a matrix function $\varphi_s(\Delta t\boldsymbol{A})$ (see, e.g., [6, Section 4.2]) is given by

$$\varphi_s(\Delta t\boldsymbol{A})\boldsymbol{b} \approx \mathbb{V}_M\varphi_s(\mathbb{V}_M^\top \mathbb{M}\Delta t\boldsymbol{A}\mathbb{V}_M)\mathbb{V}_M^\top \mathbb{M}\boldsymbol{b} = \|\boldsymbol{b}\|_\mathbb{M}\,\mathbb{V}_M\varphi_s(\Delta t\mathbb{H}_M)\boldsymbol{e}_1\,, \tag{4.14}$$

where $\boldsymbol{e}_1$ is the first canonical basis vector. Here, $\varphi_s(\Delta t\mathbb{H}_M)\boldsymbol{e}_1$ can be computed using a dense Padé approximation or an exponential of an augmented matrix (see, e.g., [24]).

For an operator $\boldsymbol{A}$ that is skew-symmetric with respect to $\mathbb{M}$, there exists a more efficient method known as the skew-Lanczos process (see, e.g., [25,26]). In the situation of Proposition 4.2, the Hessenberg matrix produced by the Arnoldi process is skew-symmetric and tri-diagonal, and the recurrence relation simplifies to the skew-Lanczos process given for $m = 1, \ldots, M-1$ by:

$$\begin{aligned} \widetilde{\boldsymbol{v}}_{m+1} &= \boldsymbol{A}\boldsymbol{v}_m - \boldsymbol{h}_{m-1,m}\boldsymbol{v}_{m-1}, & \boldsymbol{h}_{m-1,m} &= -\boldsymbol{h}_{m,m-1}, \\ \boldsymbol{v}_{m+1} &= \frac{1}{\boldsymbol{h}_{m+1,m}}\widetilde{\boldsymbol{v}}_{m+1}, & \boldsymbol{h}_{m+1,m} &= \|\widetilde{\boldsymbol{v}}_{m+1}\|_\mathbb{M}\,, \end{aligned}$$

where $\widetilde{\boldsymbol{v}}_1 = \boldsymbol{b}$ and $\boldsymbol{v}_{-1}$ and $\boldsymbol{h}_{0,1}$ are defined as zero, for convenience. In the case of a tri-diagonal Hessenberg matrix, a diagonalization can be performed in time $\mathcal{O}(M^2)$, which also makes a direct evaluation using the eigen-decomposition of $\mathbb{H}_M$ practically efficient.

Thus, the skew-Lanczos process avoids most of the reorthogonalization steps, which reduces the computational cost of the Arnoldi-method from $\mathcal{O}(M^2N)$ to $\mathcal{O}(MN)$. Therefore, it is preferable to use an appropriate inner product for computations, if possible. If such symmetry cannot be found (for instance in the case of a full Jacobian), an alternative is to use the incomplete orthogonalization method IOM [9,27], which performs orthogonalization only with respect to the last p Arnoldi vectors, while maintaining an exponential asymptotic convergence rate towards the exact solution; see [27].

Concerning the convergence behavior of the methods, we note that, according to the theoretical estimates, an exponential convergence rate of the Krylov approximation towards the matrix φ-function holds; see the overview in [6, Section 4.2]. For instance, in the skew-symmetric case [28, Theorem 4], after a minimum of $M \geq \Delta t\,|\boldsymbol{A}|$ iterations, where $|\boldsymbol{A}|$ is the spectral radius of $\boldsymbol{A}$, the error decreases at an exponential rate. We note that this error estimate couples the effort for an accurate approximation of the matrix exponential of $\boldsymbol{A}$ to a proportional factor of the time step size; cf. also section 6.1.3.

4.3. Artificial numerical dissipation

Both the modeling concerns and considerations of numerical efficiency favor a skew-symmetric choice of the linear operator as given in (4.4). In fact, the dissipation terms contained in $\boldsymbol{G}$ correspond to numerical closure terms and are usually relatively slow processes (with the possible exception of vertical diffusion in the case of a very fine vertical discretization, which we do not consider here). Moreover, horizontal diffusion can even be set up to be perfectly energy conserving, which leads to the development of the anticipated potential vorticity method; see, e.g., [29]. Therefore, a skew-symmetric operator, which conserves a linearized energy, appears the most reasonable choice and also provides algorithmic benefits. However, if the methods are employed together with very large step-sizes – which is the desired configuration – the temporal discretization error can lead to a build-up of spurious energy in high scales. If no dissipation term is present in the linear operator, this can lead to an eventual breakdown of the method due to nonlinear interaction over very long simulation horizons (of several months).

To remedy this, additional diffusion or high-frequency filtering techniques can be employed. In the following, we describe a simple technique which can be easily analyzed and ties into the ETD-Krylov approach described above. For the rest of this section, we assume that $\boldsymbol{A}_n = \boldsymbol{A} = \boldsymbol{J}\delta^2\boldsymbol{H}$, which is skew-symmetric with respect to the $\delta^2\boldsymbol{H}$ inner product. We note that this implies that $\boldsymbol{A}$ has purely imaginary spectrum, and the associated eigen-vectors are $\delta^2\boldsymbol{H}$ orthogonal. To dampen the high frequencies, we replace any occurrence of a matrix function $\varphi_s(cz)$ appearing in the scheme by a modified function $\varphi_{s,\gamma}(c,z)$ defined by

$$\varphi_{s,\gamma}(c,z) = \varphi_s(c\,(z - (-z^2/\gamma^2)^p)), \tag{4.15}$$

for some fixed $p \geq 1$ (e.g., $p = 2$) and time scale selective parameter $\gamma > 0$. Here, $0 \leq c \leq 1$ represents the constant appearing in the internal stage of the ETD-RK method, or $c = 1$ for the final stage. This change is motivated by the following result, which is simple to derive and given here without proof.

Proposition 4.3. *Let $\boldsymbol{V}_{\gamma,n}$ be computed by an ETD-RK method, where each occurrence of $\varphi_s(c\Delta t\boldsymbol{A})$ is replaced by $\varphi_{s,\gamma}(c,\Delta t\boldsymbol{A})$. Then $\boldsymbol{V}_{n,\gamma} \approx \boldsymbol{V}_\gamma(t_n)$ approximates the solution of the modified problem*

$$\partial_t \boldsymbol{V}_\gamma = \boldsymbol{F}(\boldsymbol{V}_\gamma) - (\Delta t^{2p-1}/\gamma^{2p})\,(-\boldsymbol{A}^2)^p\boldsymbol{V}_\gamma, \quad \boldsymbol{V}_\gamma(0) = \boldsymbol{V}_0, \tag{4.16}$$

at the same order as the underlying ETD-RK method.

Thus, if s is the convergence order of the underlying ETD-RK scheme, the modified scheme is of order $\min\{2p-1, s\}$. In particular, the order of convergence is maintained for $p \geq (s+1)/2$. We further comment on the structure of the perturbation term. Due to the skew-symmetry of $\boldsymbol{A}$, it follows that $-\boldsymbol{A}^2 = \boldsymbol{J}^\top\delta^2\boldsymbol{H}\boldsymbol{J}\delta^2\boldsymbol{H}$ is a $\delta^2\boldsymbol{H}$-symmetric positive operator, and thus the appearance of its p-th power in (4.16) dissipates the quadratic energy induced by $\delta^2\boldsymbol{H}$.

Remark 1. Additionally, in the concrete case of $\boldsymbol{A}$ derived as (4.4) from (3.5) it can be further verified that $-\boldsymbol{A}^2$ behaves similar to a second-order differential operator (a weighted negative Laplacian). In this case, for $p = 2$ the artificial dissipation is given by an additional biharmonic diffusion with coefficient proportional to $\Delta t^3/\gamma^4$.

Concerning the numerical implementation, a direct application of the Krylov method to (4.15) would lead to a significant increase in computation times if the Krylov space is constructed for $\Delta t\boldsymbol{A}_\gamma = \Delta t\boldsymbol{A} - (\Delta t/\gamma)^{2p}(-\boldsymbol{A}^2)^p$, since this requires additional multiplications by $\boldsymbol{A}$. Instead, we build the Krylov space as before for $\boldsymbol{A}$, and apply the modified φ-function, i.e.,

$$\varphi_{s,\gamma}(c,\Delta t\boldsymbol{A})\boldsymbol{b} \approx \|\boldsymbol{b}\|\,\mathbb{V}_M\varphi_{s,\gamma}(c,\Delta t\mathbb{H}_M)\boldsymbol{e}_1 = \|\boldsymbol{b}\|\,\mathbb{V}_M\varphi_s(c\,\Delta t(\mathbb{H}_M - \Delta t^{2p-1}/\gamma^{2p}(-\mathbb{H}_M^2)^p)\boldsymbol{e}_1\,,$$

where $\mathbb{H}_M$ is the Hessenberg matrix from section 4.2. In this way, the additional cost for a Krylov approximation with M vectors is limited to computing powers of $\mathbb{H}_M$, which is usually negligible.

4.4. Conservation of mass

The property of a scheme to be exactly mass conserving is a basic and important requirement for global ocean models. We note that the multilayer TRiSK-scheme is layerwise mass conserving in continuous time. Most commonly employed time integration methods such as explicit Runge-Kutta or implicit methods preserve this property, which makes it desirable also in the context of exponential integrators. More generally, this corresponds to the preservation of linear invariants present in the semidiscrete problem. Fortunately, under a simple requirement on the linear operator, which are fulfilled for the choices made above, many exponential integrators share this property as well.

We begin by summarizing the mass conserving properties in the multilayer model.

Proposition 4.4. *The evolution of the model* (2.5) *is layer-wise volume conserving, i.e.,*

$$(\mathbf{1}, \boldsymbol{h}_k(t))_I = const \quad \text{for all } k = 1, 2, \ldots, L,$$

where $\mathbf{1} \in X_\mathsf{I}$ *denotes the constant one cell-vector.*

Proof. The property follows by testing the k-th component of the mass equation with $\mathbf{1}$ to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{1}, \boldsymbol{h}_k(t))_I = (\mathbf{1}, \partial_t \boldsymbol{h}_k(t))_I = (\mathbf{1}, \underset{\mathsf{E}\to\mathsf{I}}{\nabla\cdot}(\{\boldsymbol{h}_k\}_\mathsf{E} * \boldsymbol{u}_k))_\mathsf{I} = -(\underset{\mathsf{I}\to\mathsf{E}}{\nabla}\mathbf{1}, \{\boldsymbol{h}_k\}_\mathsf{E} * \boldsymbol{u}_k)_\mathsf{E} = 0,$$

using the discrete adjoint relation between $\nabla\cdot_{\mathsf{E}\to\mathsf{I}}$ and $\nabla_{\mathsf{I}\to\mathsf{E}}$. □

Consequently, also the total mass $\sum_k \int_\Omega \rho_k h_k$, given in the discrete equation by

$$\boldsymbol{m}[\boldsymbol{h}] = \sum_{k=1}^{L} \rho_k (\mathbf{1}, \boldsymbol{h}_k(t))_I, \tag{4.17}$$

is conserved. We note that these conservation properties can be expressed more generally as linear invariants,

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{l}, \boldsymbol{V}(t))_X = (\boldsymbol{l}, \partial_t \boldsymbol{V}(t))_X = (\boldsymbol{l}, \boldsymbol{F}[\boldsymbol{V}(t)])_X = 0, \tag{4.18}$$

where $(\boldsymbol{l}, \cdot)_X$ is a linear functional, represented by testing with the vector $\boldsymbol{l}$. For instance, in the case of (4.17), we choose $\boldsymbol{l} = (\boldsymbol{l}^h, \boldsymbol{l}^u) = (\boldsymbol{\rho}, \mathbf{0})$, where $\boldsymbol{\rho}_k = \rho_k \mathbf{1}$.

Under the appropriate assumption on the linear operator $\boldsymbol{A}$, and the underlying ETD method, linear invariants remain preserved in the time discrete system.

Theorem 4.5. *Assume that* $(\boldsymbol{l}, \boldsymbol{F}(\boldsymbol{V}))_X = 0$ *for all* $\boldsymbol{V} \in X$ *(which implies the linear invariant* (4.18)*). Assume further that*

$$(\boldsymbol{l}, \boldsymbol{A}_n \boldsymbol{V})_X = 0 \quad \text{for all } \boldsymbol{V} \in X.$$

Then, the ETD methods presented in this section preserve the same linear invariant.

Proof. We use the explicit formula for the final stage (4.6) to obtain

$$(\boldsymbol{l}, \boldsymbol{V}_{n+1} - \boldsymbol{V}_n)_X = \sum_{s=1}^{S} (\boldsymbol{l}, \varphi_s(\boldsymbol{A}_n)\boldsymbol{b}_s)_X,$$

where $\boldsymbol{b}_1 = \boldsymbol{F}(\boldsymbol{V}_n))$ and $\boldsymbol{b}_s$ for $s > 1$ is a linear combination of the residuals $\boldsymbol{R}_n(\boldsymbol{v}_n^i)$ evaluated at the internal stages $\boldsymbol{v}_n^i$. Now, we directly obtain $(\boldsymbol{l}, \boldsymbol{b}_s)_X = 0$ using the properties of $\boldsymbol{F}$ and $\boldsymbol{A}_n$. Then, testing the linear equation (4.8) which corresponds to the expressions involving the φ-functions by $\boldsymbol{l}$ reveals that also $(\boldsymbol{l}, \varphi_s(\boldsymbol{A}_n)\boldsymbol{b}_s)_X = 0$ for all s, which yields

$$(\boldsymbol{l}, \boldsymbol{V}_{n+1})_X = (\boldsymbol{l}, \boldsymbol{V}_n)_X,$$

which is the desired property. □

Thereby, we directly obtain mass conservation for the specific ETD schemes.

Corollary 4.6. *For the choices* $\boldsymbol{A}_n = \boldsymbol{F}'[\boldsymbol{V}_n]$ *and* $\boldsymbol{A}_n = \boldsymbol{A}^{ref} = \boldsymbol{F}'[\boldsymbol{V}^{ref}]$*, see* (4.3) *and* (4.4)*, we obtain*

$$(\mathbf{1}, \boldsymbol{h}_{k,n+1})_\mathsf{I} = (\mathbf{1}, \boldsymbol{h}_{k,n})_\mathsf{I} \quad k = 1, 2, \ldots, L, \quad \text{and} \quad \boldsymbol{m}[\boldsymbol{h}_{n+1}] = \boldsymbol{m}[\boldsymbol{h}_n],$$

where $(\boldsymbol{h}_n, \boldsymbol{u}_n) = \boldsymbol{V}_n$ *are the time steps* (4.6)*.*

5. Energetically consistent layer reduction

Motivated by the mode analysis of section 2.3, we propose a layer reduction technique to project the linear operator used in the ETD method to a subspace corresponding to the fastest modes. Here, we take special care to perform this projection in an energetically consistent way and to account for invariants responsible for mass conservation. In this section, we focus attention on the linear operators of the structure (4.4) and let $\boldsymbol{J} = \boldsymbol{J}[\boldsymbol{V}^{\mathrm{ref}}]$ and $\delta^2\boldsymbol{H} = \delta^2\boldsymbol{H}[\boldsymbol{V}^{\mathrm{ref}}]$ for some reference configuration $\boldsymbol{V}^{\mathrm{ref}} = (\boldsymbol{h}^{\mathrm{ref}}, \boldsymbol{0}) \in X$.

5.1. A general class of linear operators

At first, we describe a general method for layer reduction, which is based on a projection preserving the Hamiltonian structure of the linearized equation. For an ETD method, it is desirable to only approximate the fast modes of the system. Based on the eigenvalue analysis of section 2.3, we can exploit the fact that the vertical modes are decreasing in magnitude with finer vertical resolution (in contrast to the horizontal modes, which are increasing with finer horizontal resolution).

We introduce a reduced-layer space

$$X^{\widehat{L}} = X_{\mathsf{I}}^{\widehat{L}} \times X_{\mathsf{E}}^{\widehat{L}} \quad \text{for } 1 \le \widehat{L} \ll L,$$

which is supposed to parametrize the fastest vertical modes of the discrete linear operator $\boldsymbol{A} = \boldsymbol{J}\,\delta^2\boldsymbol{H}$ as in (4.4). We define a corresponding ansatz by the linear mapping $\boldsymbol{\Psi}\colon X^{\widehat{L}} \to X^L$ with the structure

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_h & 0 \\ 0 & \boldsymbol{\Psi}_u \end{pmatrix}.$$

Here, the matrix $\boldsymbol{\Psi}_h$ (and similarly, $\boldsymbol{\Psi}_u$) is defined by

$$\left[\boldsymbol{\Psi}_h\widehat{\boldsymbol{h}}\right]_i = \sum_{j=1}^{\widehat{L}} \boldsymbol{\Psi}_h^{j,i}\widehat{\boldsymbol{h}}_{k,i}, \quad \text{for all } i \in \mathsf{I},$$

and any $\widehat{\boldsymbol{h}} \in X_{\mathsf{I}}^{\widehat{L}}$. Here, the vector $\boldsymbol{\Psi}_h^{j,i} \in \mathbb{R}^L$ should roughly correspond to the j-th fastest vertical height mode, which can be different in each cell $i \in \mathsf{I}$.

Remark 2. Consider the simplified case of a constant Coriolis term and bathymetry, i.e. $f \equiv \mathit{const}$, $b \equiv \mathit{const}$ with constant reference heights $h^0 \equiv \mathit{const}$. Let $\mu_j > \mu_{j+1} > 0$, $j = 1, 2, \ldots, L$ be the eigenvalues of the matrix $A^{\mathrm{vert}} = g\,\mathrm{diag}(h^0/\rho)R$ with R defined as in (2.11), corresponding to the eigenvectors $w^{j,h,vert} \in \mathbb{R}^L$. These modes correspond to the height variable, whereas the corresponding modes for the velocity are given as $w^{j,u,\mathrm{vert}} = \mathrm{diag}(1/\rho)Rw^{j,h,\mathrm{vert}}$. In this case, we can choose $\boldsymbol{\Psi}_h^{j,i} = w^{j,h,\mathrm{vert}}$ and $\boldsymbol{\Psi}_u^{j,e} = w^{j,u,\mathrm{vert}}$ for all cells and edges. In this particular case, the following arguments would greatly simplify. However, since b and f are generally not constant, the horizontal modes can only be defined in an approximate sense, separately in each cell and edge.

While the ansatz functions $\boldsymbol{\Psi}$ can be chosen freely, in principle, it is important to obtain appropriate dynamics for the reduced linear system. Here, since we are dealing with a linear Hamiltonian system (the rotating multilayer wave equation), we will take care to preserve this structure when deriving the reduced dynamics. The special case of using only the single fastest barotropic mode, which leads to simple concrete formulas, will be discussed further in section 5.3. We start by defining the reduced linearized Hamiltonian, which arises from the canonical ansatz

$$\widehat{\boldsymbol{H}}^{\mathrm{ref}}(\widehat{\boldsymbol{V}}) = \boldsymbol{H}^{\mathrm{ref}}(\boldsymbol{\Psi}\widehat{\boldsymbol{V}}) = \frac{1}{2}\Big(\boldsymbol{\Psi}\widehat{\boldsymbol{V}}, \delta^2\boldsymbol{H}\,\boldsymbol{\Psi}\widehat{\boldsymbol{V}}\Big)_{X^L}$$

for $\widehat{\boldsymbol{V}} \in X^{\widehat{L}}$. Now, we use additionally the fact that the mass matrix $\boldsymbol{M}_{X^L}$ of the space X^L commutes with $\boldsymbol{\Psi}^\top$ in the sense that $\boldsymbol{M}_{X^L}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{M}_{X^{\widehat{L}}}$. In fact, for $\boldsymbol{\Psi}_h$ (and similarly for $\boldsymbol{\Psi}_u$) it simply holds for any $\widehat{\boldsymbol{h}} \in X_{\mathsf{I}}^{\widehat{L}}$ that

$$\left[\boldsymbol{M}_{X_{\mathsf{I}}^L}\boldsymbol{\Psi}_h\widehat{\boldsymbol{h}}\right]_i = A_i\left[\boldsymbol{\Psi}_h\widehat{\boldsymbol{h}}\right]_i = \left[\boldsymbol{\Psi}_h\boldsymbol{M}_{X_{\mathsf{I}}^{\widehat{L}}}\widehat{\boldsymbol{h}}\right]_i \quad \text{for all } i \in \mathsf{I},$$

where A_i is the area of the i-th cell. This corresponds to the fact that $\boldsymbol{\Psi}$ operates only on the layer indices for each cell and edge stack, and that both mass matrices consist of multiple identical copies of the (diagonal) single-layer mass-matrix. Thus, it follows that

$$\widehat{\boldsymbol{H}}^{\mathrm{ref}}(\widehat{\boldsymbol{V}}) = \frac{1}{2}(\boldsymbol{M}_{X_{\mathsf{I}}^L}\boldsymbol{\Psi}\widehat{\boldsymbol{V}})^\top(\delta^2\boldsymbol{H}\,\boldsymbol{\Psi}\widehat{\boldsymbol{V}}) = \frac{1}{2}\Big(\widehat{\boldsymbol{V}}, \boldsymbol{\Psi}^\top\delta^2\boldsymbol{H}\,\boldsymbol{\Psi}\widehat{\boldsymbol{V}}\Big)_{X^{\widehat{L}}} = \frac{1}{2}\Big(\widehat{\boldsymbol{V}}, \delta^2\widehat{\boldsymbol{H}}\widehat{\boldsymbol{V}}\Big)_{X^{\widehat{L}}},$$

with the corresponding reduced (linearized) Hamiltonian matrix defined as

$$\delta^2 \widehat{\boldsymbol{H}} = \boldsymbol{\Psi}^\top \delta^2 \boldsymbol{H} \boldsymbol{\Psi}.$$

We note that the above projection can be computed separately for each cell- and edge-stack, since for fixed layer coordinates $\delta^2 \boldsymbol{H}$ is a diagonal matrix in the horizontal dimensions. We also note that $\delta^2 \widehat{\boldsymbol{H}}$ shares the same property.

In order to derive a projection operator, we first introduce the transpose matrix of $\boldsymbol{\Psi}$, denoted by $\boldsymbol{\Psi}^\top \colon X^L \to X^{\widehat{L}}$. Note that for the particular setting considered here, it is identical to the adjoint of $\boldsymbol{\Psi}$, i.e., it fulfills

$$(\boldsymbol{\Psi}^\top \boldsymbol{V}, \widehat{\boldsymbol{V}})_{X^{\widehat{L}}} = (\boldsymbol{V}, \boldsymbol{\Psi} \widehat{\boldsymbol{V}})_{X^L},$$

for any $\boldsymbol{V} \in X^L$ and $\widehat{\boldsymbol{V}} \in X^{\widehat{L}}$. Here, we have used again the fact that the mass matrix commutes with $\boldsymbol{\Psi}$. Although $\boldsymbol{\Psi}^\top$ maps from the full to the reduced space, it is not appropriate to map solution variables from the full to the reduced space. Instead, in order to provide an energetically consistent projection of the linearized equation to the reduced space, we introduce $\boldsymbol{\Psi}^\dagger$, the (generalized) Moore–Penrose pseudoinverse

$$\boldsymbol{\Psi}^\dagger \colon X^L \to X^{\widehat{L}}, \quad \boldsymbol{\Psi}^\dagger = (\delta^2 \widehat{\boldsymbol{H}})^{-1} \boldsymbol{\Psi}^\top \delta^2 \boldsymbol{H}.$$

Hence, $\boldsymbol{\Psi}^\dagger$ gives the reduced layer coordinates for any discrete solution variable. Again, we note that the inverse $(\delta^2 \widehat{\boldsymbol{H}})^{-1}$ can be computed in a cell- and edge-stack wise fashion, reducing the solution to a large number of $\widehat{L} \times \widehat{L}$ sized systems. The definition of $\boldsymbol{\Psi}^\dagger$ is motivated by the following derivation:

Proposition 5.1. *The restriction matrix $\boldsymbol{\Psi}^\dagger$ gives the solution to the following minimization problem: For any $\boldsymbol{V} \in X^L$ we have $\boldsymbol{\Psi}^\dagger \boldsymbol{V} = \widehat{\boldsymbol{V}}$ where*

$$\widehat{\boldsymbol{V}} = \operatorname*{arg\,min}_{\widehat{\boldsymbol{V}} \in X^{\widehat{L}}} \| \boldsymbol{V} - \boldsymbol{\Psi} \widehat{\boldsymbol{V}} \|_{\delta^2 \boldsymbol{H}} = \operatorname*{arg\,min}_{\widehat{\boldsymbol{V}} \in X^{\widehat{L}}} \frac{1}{2} (\boldsymbol{V} - \boldsymbol{\Psi} \widehat{\boldsymbol{V}}, \delta^2 \boldsymbol{H} (\boldsymbol{V} - \boldsymbol{\Psi} \widehat{\boldsymbol{V}}))_{X^L} . \tag{5.1}$$

Proof. We define the energy norm $\| \boldsymbol{V} \|_{\delta^2 \boldsymbol{H}} = \sqrt{(\boldsymbol{V}, \delta^2 \boldsymbol{H}\, \boldsymbol{V})_X}$ in the canonical way. The derivation is then standard, by writing out the optimality condition of (5.1), given by

$$\boldsymbol{\Psi}^\top \boldsymbol{M}_X \delta^2 \boldsymbol{H} \boldsymbol{\Psi} \widehat{\boldsymbol{V}} = \boldsymbol{\Psi}^\top \boldsymbol{M}_X \delta^2 \boldsymbol{H}\, \boldsymbol{V}.$$

Now, we use again the fact that the mass matrix $\boldsymbol{M}_{X^L}$ of the space X^L commutes with $\boldsymbol{\Psi}^\top$ in the sense that $\boldsymbol{\Psi}^\top \boldsymbol{M}_{X^L} = \boldsymbol{M}_{X^{\widehat{L}}} \boldsymbol{\Psi}^\top$. □

Additionally, we introduce the orthogonal projection

$$\boldsymbol{P} = \boldsymbol{\Psi} \boldsymbol{\Psi}^\dagger \colon X^L \to X^L .$$

By Proposition 5.1, the projection is given as $\boldsymbol{P} \boldsymbol{V} = \boldsymbol{\Psi} \widehat{\boldsymbol{V}}$, where $\widehat{\boldsymbol{V}}$ is the minimizer of (5.1), and thus minimizes the projection error in the canonical linearized energy norm. Based on this choice of the projection, a projected linear operator $\boldsymbol{A}_P$ can be defined for $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{ref}} = \boldsymbol{J} \delta^2 \boldsymbol{H}$ as

$$\begin{aligned} \boldsymbol{A}_P &= \boldsymbol{P} \boldsymbol{A} \boldsymbol{P} = \boldsymbol{\Psi} \widehat{\boldsymbol{A}} \boldsymbol{\Psi}^\dagger, \\ \text{where} \quad \widehat{\boldsymbol{A}} &= \boldsymbol{\Psi}^\dagger \boldsymbol{A} \boldsymbol{\Psi}. \end{aligned} \tag{5.2}$$

Here, $\widehat{\boldsymbol{A}} \colon X^{\widehat{L}} \to X^{\widehat{L}}$ is the corresponding reduced layer operator. Using the properties of $\boldsymbol{\Psi}^\dagger$, we now verify that the operators $\widehat{\boldsymbol{A}}$ and $\boldsymbol{A}_P$ again have Hamiltonian structure, together with an appropriate definition of the reduced $\mathcal{J}$-operator.

Proposition 5.2. *Define the skew-adjoint operators*

$$\begin{aligned} &\boldsymbol{J}_P \colon X^L \to X^L \quad \boldsymbol{J}_P = \boldsymbol{P} \boldsymbol{J} \boldsymbol{P}^\top \\ &\widehat{\boldsymbol{J}} \colon X^{\widehat{L}} \to X^{\widehat{L}} \quad \widehat{\boldsymbol{J}} = \boldsymbol{\Psi}^\dagger \boldsymbol{J} (\boldsymbol{\Psi}^\dagger)^\top. \end{aligned} \tag{5.3}$$

Then, it holds that

$$\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{J}} \delta^2 \widehat{\boldsymbol{H}} \quad \textit{and} \quad \boldsymbol{A}_P = \boldsymbol{J}_P \delta^2 \boldsymbol{H}_P,$$

where $\delta^2 \boldsymbol{H}_P = \boldsymbol{P}^\top \delta^2 \boldsymbol{H} \boldsymbol{P}$.

Proof. By definition of $\boldsymbol{\Psi}^\dagger$ we have $(\boldsymbol{\Psi}^\dagger)^\top \delta^2 \widehat{\boldsymbol{H}} = \delta^2 \boldsymbol{H} \boldsymbol{\Psi}$ and thus

$$\boldsymbol{\Psi}^\dagger \boldsymbol{A} \boldsymbol{\Psi} = \boldsymbol{\Psi}^\dagger \boldsymbol{J} \delta^2 \boldsymbol{H} \boldsymbol{\Psi} = \boldsymbol{\Psi}^\dagger \boldsymbol{J} (\boldsymbol{\Psi}^\dagger)^\top \delta^2 \widehat{\boldsymbol{H}} = \widehat{\boldsymbol{J}} \delta^2 \widehat{\boldsymbol{H}}$$

Concerning the second case, we compute

$$\boldsymbol{A}_P = \boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger \boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger = \boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger \boldsymbol{J}(\boldsymbol{\Psi}^\dagger)^\top \boldsymbol{\Psi}^\top \delta^2 \boldsymbol{H}\boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger,$$

using the previous result. Now we observe that $(\boldsymbol{\Psi}^\dagger)^\top \boldsymbol{\Psi}^\top = \boldsymbol{P}^\top = \boldsymbol{P}^\top \boldsymbol{P}^\top$, owing to the fact that $\boldsymbol{P}$ is a projection. □

Thereby, the structural properties important for stability of the solutions are preserved. Thus, we can easily employ the derived linear operator in the context of an ETD method, by simply replacing the operator $\boldsymbol{A}$ by $\boldsymbol{A}_P$. By the construction the adaptation is straightforward. However, for the practical use of the method it is important that the computation of the matrix φ-functions can be reduced to a smaller-size problem, based on the following observation.

Proposition 5.3. *Let $\boldsymbol{A}_P = \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}$ and $\widehat{\boldsymbol{A}} = \boldsymbol{\Psi}^\dagger \boldsymbol{A}\boldsymbol{\Psi}$ be defined as above. Then, it holds for any $s \geq 0$ that*

$$\varphi_s(\boldsymbol{A}_P) = \frac{1}{s!}(\mathrm{Id} - \boldsymbol{P}) + \boldsymbol{\Psi}\,\varphi_s(\widehat{\boldsymbol{A}})\,\boldsymbol{\Psi}^\dagger.$$

Proof. Due to (4.7), we have

$$\varphi_s(\boldsymbol{A}_P) = \sum_{k=0}^{\infty} \frac{\boldsymbol{A}_P^k}{(k+s)!} = \frac{1}{s!}(\mathrm{Id} - \boldsymbol{P}) + \boldsymbol{\Psi} \sum_{k=0}^{\infty} \frac{(\boldsymbol{\Psi}^\dagger \boldsymbol{A}\boldsymbol{\Psi})^k}{(k+s)!} \boldsymbol{\Psi}^\dagger,$$

using the identities $\boldsymbol{P} = \boldsymbol{\Psi}\boldsymbol{\Psi}^\dagger$, $\boldsymbol{A}_P^0 = \mathrm{Id}$ and $\boldsymbol{A}_P^k = \boldsymbol{\Psi}(\boldsymbol{\Psi}^\dagger \boldsymbol{A}\boldsymbol{\Psi})^k \boldsymbol{\Psi}^\dagger$ for $k \geq 1$. □

Thereby, the computation of $\varphi_s(\Delta t \boldsymbol{A}_P)\boldsymbol{b}_s$ in an ETD method can be reduced to the computation of $\varphi_s(\Delta t \widehat{\boldsymbol{A}})\widehat{\boldsymbol{b}}_s$, the restriction $\widehat{\boldsymbol{b}}_s = \boldsymbol{\Psi}^\dagger \boldsymbol{b}_s$, and another application of the prolongation $\boldsymbol{\Psi}^\dagger$. In the context of an ETD-Krylov method, this significantly reduces the required computational work.

5.2. Implementation of the resulting ETD methods

Finally, for illustrative purposes, we explicitly write the exponential Euler method using the projected operator $\boldsymbol{A}_P$. On a high level, we simply replace the operator $\boldsymbol{A}$ by $\boldsymbol{A}_P$ in (4.9). Using Proposition 5.3 it can then be rewritten as

$$\begin{aligned} \boldsymbol{V}_{n+1} &= \boldsymbol{V}_n + \Delta t\,\varphi_1(\Delta t \boldsymbol{A}_P)\boldsymbol{F}[\boldsymbol{V}_n] \\ &= \boldsymbol{V}_n + \Delta t(\mathrm{Id} - \boldsymbol{P})\boldsymbol{F}[\boldsymbol{V}_n] + \Delta t\,\boldsymbol{\Psi}\,\varphi_1(\Delta t \widehat{\boldsymbol{A}})\,\boldsymbol{\Psi}^\dagger \boldsymbol{F}[\boldsymbol{V}_n]\,. \end{aligned}$$

Thus, the method performs an explicit Euler step with the forcing term projected to the orthogonal complement of the span of $\boldsymbol{\Psi}$, whereas the part of the forcing term in the space spanned by $\boldsymbol{\Psi}$ is treated with the matrix exponential associated to the projected matrix $\widehat{\boldsymbol{A}}$. If the matrix $\widehat{\boldsymbol{A}}$ is assembled ahead of time, the evaluation of the matrix exponential of the reduced matrix is more efficient, since the number of degrees of freedom and the nonzero entries of the projected matrix is much smaller than the number of degrees of freedom of the original one. Similarly, the higher order ETD methods can be rewritten in the above way to a form that is suitable for implementation purposes.

5.3. Barotropic ETD method

Due to the fact that the quotient of the first mode (the fast barotropic mode) and the second mode (the fastest baroclinic mode) is usually much bigger than one in realistic global ocean simulations, the stiffest parts of the linear operator can be captured by a particularly simple choice of $\boldsymbol{\Psi}$, which exploits the analytical structure of this mode. We note that state-of-the art global models exploit this splitting as well. In particular, we refer to the widely used split-explicit scheme; see [2]. Here, a suitable method arises from a direct application of an exponential integrator with a particular choice of $\boldsymbol{\Psi}$, which we refer to as barotropic ETD (B-ETD) method.

For a reference configuration $\boldsymbol{h}^{\mathrm{ref}} \in X_{\mathrm{I}}^L$ define the corresponding total height and average density as

$$\widehat{\boldsymbol{h}} = \sum_{k=1}^{L} \boldsymbol{h}_k^{\mathrm{ref}}, \qquad \widehat{\boldsymbol{\rho}} = \sum_{k=1}^{L} \frac{\rho_k \boldsymbol{h}_k^{\mathrm{ref}}}{\widehat{\boldsymbol{h}}} \in \mathrm{X}_{\mathrm{I}}.$$

Based on the approximate form of the fastest vertical mode (see section 2.3), we consider the concrete choice with $\widehat{L} = 1$ given by

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{y} & 0 \\ 0 & \boldsymbol{1} \end{pmatrix}, \quad \text{where } \boldsymbol{1} \equiv 1,\ \boldsymbol{y}_k = \boldsymbol{h}_k^{\mathrm{ref}}/\widehat{\boldsymbol{h}}. \tag{5.4}$$

In the following, we compute the concrete form of the layer-reduced Hamiltonian and the operator $\boldsymbol{\Psi}^\dagger$, containing the test-functions. Due to the fact that the average density is given as $\widehat{\boldsymbol{\rho}} = \rho^\top \boldsymbol{y}$, the concrete form of the reduced Hamiltonian is readily derived as

$$\delta^2 \widehat{\boldsymbol{H}} = \begin{pmatrix} g\,\mathrm{diag}(\widehat{\boldsymbol{r}}) & 0 \\ 0 & \mathrm{diag}\left(\{\widehat{\boldsymbol{\rho}} * \widehat{\boldsymbol{h}}\}_{\mathsf{E}}\right) \end{pmatrix}, \quad \text{where } \widehat{\boldsymbol{r}} = \boldsymbol{y}^\top R \boldsymbol{y} \in X_{\mathsf{I}},$$

with the matrix $R = T^\top \mathrm{diag}(\Delta\rho) T = (\rho_{\min\{k,l\}})_{k,l}$ introduced in section 2.3. We note that $\delta^2\widehat{\boldsymbol{H}}$ simply corresponds to a quadratic approximation of a single-layer Hamiltonian, albeit with variable densities. In fact, $\widehat{\boldsymbol{h}}$ is the total column height of the reference configuration, and both $\widehat{\boldsymbol{\rho}}$ and $\widehat{\boldsymbol{r}}$ are average values of the density over each stack. A simple computation now yields the concrete form of $\boldsymbol{\Psi}^\dagger$ as

$$\boldsymbol{\Psi}^\dagger = \begin{pmatrix} (1/\widehat{\boldsymbol{r}}) * \boldsymbol{y}^\top R & 0 \\ 0 & \{\rho * \boldsymbol{h}^{\mathrm{ref}}\}_{\mathsf{E}}^\top / \{\widehat{\boldsymbol{h}} * \widehat{\boldsymbol{\rho}}\}_{\mathsf{E}} \end{pmatrix}.$$

At first glance, the concrete form of $\boldsymbol{\Psi}^\dagger$ is not very instructive, even though it can be easily computed in practice. However, in the special case of constant densities, it simplifies further.

Remark 3. In the case where $\rho_k = \rho^{\mathrm{ref}}$ for $k = 1, 2, \dots, L$, we obtain that

$$\boldsymbol{\Psi}^\dagger = \begin{pmatrix} \boldsymbol{1}^\top & 0 \\ 0 & \{\boldsymbol{h}^{\mathrm{ref}}\}_{\mathsf{E}}^\top / \{\widehat{\boldsymbol{h}}\}_{\mathsf{E}} \end{pmatrix}.$$

Thus, the roles of the test-functions in $\boldsymbol{\Psi}^\dagger$ and ansatz-functions in $\boldsymbol{\Psi}$ are simply interchanged with respect to the continuity and momentum equation. We note that this closely resembles the averaging operators employed in the split-explicit scheme; cf. [2].

5.4. Total mass conservation

One drawback of the outlined approach is that the form of the test functions in $\boldsymbol{\Psi}^\dagger$ can not be controlled directly, rather they arise from the choice of $\boldsymbol{\Psi}$ in an indirect way. This is a problem for instance for exact mass-conservation as considered in section 4.4. We briefly describe a simple remedy for this in the context of the barotropic method.

In a first step, we replace the second variation of the Hamiltonian by the modified version

$$\delta^2 \widetilde{\boldsymbol{H}} = \begin{pmatrix} g\,\mathrm{diag}\left(\rho\rho^\top / \widehat{\boldsymbol{\rho}}\right) & 0 \\ 0 & \mathrm{diag}\left(\rho * \{\boldsymbol{h}\}_{\mathsf{E}}\right) \end{pmatrix},$$

where the matrix R is replaced by the cell-wise defined rank-one matrix $\rho\rho^\top/\widehat{\boldsymbol{\rho}}_i \in \mathbb{R}^{L\times L}$ for every $i \in \mathsf{I}$. The reduced operator is then derived in the same way as before, based on this modified Hamiltonian. Using again the ansatz $\boldsymbol{\Psi}$ from (5.4), we obtain now

$$\delta^2 \widehat{\boldsymbol{H}} = \begin{pmatrix} g\,\mathrm{diag}(\widehat{\boldsymbol{\rho}}) & 0 \\ 0 & \mathrm{diag}\left(\{\widehat{\boldsymbol{\rho}}\widehat{\boldsymbol{h}}\}_{\mathsf{E}}\right) \end{pmatrix} \text{ and } \boldsymbol{\Psi}^\dagger = \begin{pmatrix} \rho^\top/\widehat{\boldsymbol{\rho}} & 0 \\ 0 & \{\rho * \boldsymbol{h}^{\mathrm{ref}}\}_{\mathsf{E}}^\top / \{\widehat{\boldsymbol{h}} * \widehat{\boldsymbol{\rho}}\}_{\mathsf{E}} \end{pmatrix}.$$

The resulting linear operator from this choice leads to global mass conservation.

Proposition 5.4. *For a reference configuration, define as before $\widetilde{\boldsymbol{A}}_P = \boldsymbol{P}\,\boldsymbol{J}\,\delta^2\widetilde{\boldsymbol{H}}\,\boldsymbol{P} = \boldsymbol{\Psi}\,\widehat{\boldsymbol{J}}\,\delta^2\widehat{\boldsymbol{H}}\,\boldsymbol{\Psi}^\dagger$. Then, a corresponding ETD-method with $\boldsymbol{A}_n = \widetilde{\boldsymbol{A}}_P$ preserves the total mass;*

$$\boldsymbol{M}[\boldsymbol{h}_{n+1}] = \boldsymbol{M}[\boldsymbol{h}_n], \quad \textit{where } \boldsymbol{M}[\boldsymbol{h}] = \sum_{k=1}^{L} \rho_k (\boldsymbol{1}, \boldsymbol{h}_k)_{\mathsf{I}},$$

and $(\boldsymbol{h}_n, \boldsymbol{u}_n) = \boldsymbol{V}_n$ are the time steps (4.6).

Proof. Defining the vector $\boldsymbol{l} = (\boldsymbol{\rho}, \boldsymbol{0})$, mass can be computed as $\boldsymbol{M}(\boldsymbol{V}) = (\boldsymbol{l}, \boldsymbol{V})_X$, and with Theorem 4.5, we have to verify that

$$(\boldsymbol{l}, \widetilde{\boldsymbol{A}}_P \boldsymbol{V})_X = (\boldsymbol{\rho}, \boldsymbol{\Psi}_h \boldsymbol{\Psi}_h^\dagger \boldsymbol{f}_h[\boldsymbol{V}])_{X_{\mathsf{I}}^L} = 0 \quad \text{for all } \boldsymbol{V} \in X,$$

where $\boldsymbol{f}_h[\boldsymbol{V}] \in X_{\mathsf{I}}^L$ is the first component (corresponding to height variables) of $\boldsymbol{f}[\boldsymbol{V}] = \boldsymbol{J}\,\delta^2\widetilde{\boldsymbol{H}}\,\boldsymbol{P}\boldsymbol{V}$. We note that $\boldsymbol{\Psi}_h \boldsymbol{\Psi}_h^\dagger = \boldsymbol{y}\rho^\top/\widehat{\boldsymbol{\rho}}$, from the concrete form of $\boldsymbol{\Psi}^\dagger$. Thus, it follows that $\rho^\top \boldsymbol{\Psi}_h \boldsymbol{\Psi}_h^\dagger = \rho^\top \boldsymbol{y}\rho^\top/\widehat{\boldsymbol{\rho}} = \rho^\top$, since $\widehat{\boldsymbol{\rho}} = \rho^\top \boldsymbol{y}$. Therefore, it indeed follows $(\boldsymbol{\rho}, \boldsymbol{\Psi}_h \boldsymbol{\Psi}_h^\dagger \boldsymbol{f}_h[\boldsymbol{V}])_{X_{\mathsf{I}}^L} = (\boldsymbol{\rho}, \boldsymbol{f}_h[\boldsymbol{V}])_{X_{\mathsf{I}}^L} = 0$, due to the properties of $\boldsymbol{J}$. □

Remark 4. Similarly, replacing the matrix R by the constant rank-one matrix $(\boldsymbol{R}_i)_{k,l} = \widehat{\boldsymbol{\rho}}_i$ in every cell $i \in \mathrm{I}$ results in an ETD method which exactly conserves the total layer volume, defined as $\sum_k (\mathbf{1}, \boldsymbol{h}_k)_{\mathrm{I}}$.

We note that this method incurs an additional approximation error. However, since the approximation of R with a rank-one matrix is well justified, and the reduced system can only capture the single fast mode contained in this space, we still expect good properties from this linear operator in the context of an ETD-scheme.

6. Numerical results

In this section, we numerically demonstrate the stability and performance of the ETD methods described in this work. We first given an overview of the simulation setup, which is based on a simplified version of the SOMA testcase [30]. The computational domain is given by a circular basin on the surface of the sphere of radius 6371.22 km, centered at (35°N, 0°W) longitude-latitude. The basin is 2500 km in diameter, has a depth ranging from 2.5 km at the center to 100 m on the coastal shelf. The concrete form of the bathymetry can be found in [30, Appendix A]; see also Fig. 1. We consider a single-layer and a three-layer configuration.

6.1. Algorithmic details

In the following, we detail the numerical setup employed in the computational experiments.

6.1.1. Spatial mesh

A quasi-uniform mesh is constructed from a centroidal Voronoi tessellation [31], where the distance of the cell centers d_e is ca. 16 km resolution. In order to obtain an initial condition for the initial layer configuration, we interpolate the initial heights as in Fig. 1 to the cell centers. Then, all cell variables in each layer that correspond to zero heights are marked as dry. Additionally, all edges adjacent to a dry cell are marked as boundary edges. Subsequently, the degrees of freedom corresponding to those cells and edges are fixed to zero and thus eliminated from the computation; cf. also Appendix A.

In order to compare different time stepping methods at different CFL-numbers, we introduce the reference time step and the Courant number. In this context, we define it for simplicity as

$$\Delta t_{\mathrm{C}} = 1/\,|\boldsymbol{A}_0|\,, \quad \mathrm{C}(\Delta t) = \Delta t\,|\boldsymbol{A}_0| = \Delta t/\Delta t_{\mathrm{C}}$$

where $|\cdot| = \sigma_{\max}(\cdot)$ denotes the largest magnitude eigenvalue, and $\boldsymbol{A}_0 = \boldsymbol{F}'(\boldsymbol{V}^0)$ is the linearized operator at the stable reference configuration. We note that Δt_{C} is determined (up to a constant factor) by the largest quotient of the local mesh-width and the local free-surface wave-speed $\sqrt{g\,b(x)}$. Concretely, we obtain $\Delta t_{\mathrm{C}} \approx 37.9$ [s] on the given domain, mesh, and bathymetry.

6.1.2. Considered time stepping methods

In the tests, the explicit fourth-order Runge-Kutta method (RK4) serves as a base-line, since it is explicit (thus easy to efficiently implement in a parallel environment), sufficiently high order accurate (in combination with the second-order TRiSK scheme), and includes an imaginary interval in its stability region. Specifically, stability of RK4 (for the linearized equation $\partial_t \boldsymbol{V} = \boldsymbol{A}_0 \boldsymbol{V}$) is given for Courant numbers $\mathrm{C}(\Delta t) \leq \sqrt{8}$. Thus, the maximal RK4 stepsize is given as $\Delta t_{\mathrm{RK4}} = \sqrt{8}\Delta t_{\mathrm{C}} \approx 107.2$ [s] for the 16 km grid, which is used as a reference time step for performance considerations. We remark that, in practice, a stable simulation is only obtained for slightly smaller Courant numbers, since the definition employed above ignores the nonlinearity in the forcing term. Concerning the choice of RK4 over lower order methods, we note that optimal order one and two stage RK schemes are unconditionally unstable for imaginary eigenvalues, and that RK4 delivers a better ratio of the number of internal stages to the maximal CFL-compliant time step than RK3.

The ETD methods described in this work can be separated into two classes. The first class of methods is constructed by choosing the linear operator as $\boldsymbol{A}_n = \boldsymbol{A}^{\mathrm{ref}}$, as in (4.4), linearized either at the reference configuration $\boldsymbol{V}^{\mathrm{ref}} = (-\boldsymbol{b}, 0)$ or updated in each time step with the current height $\boldsymbol{V}_n^{\mathrm{ref}} = (\boldsymbol{h}_n, 0)$. Since $\boldsymbol{A}_n^{\mathrm{ref}}$ corresponds to a first-order wave operator, this class of methods will be called ETDSwave, where S refers to the number of internal stages. The second class of methods are based on section 5. Within this class, we will focus on the methods where the linear operator is projected onto the barotropic mode 5.3. For this reason we will refer to these methods as B-ETDSwave.

6.1.3. Implementation of the Krylov methods

For both classes of ETD methods, the Krylov subspace method from section 4.2.1 is used to evaluate the φ_s functions. Because both classes of methods possess the properties in Proposition 4.2, the more efficient skew-Lanczos process, described in section 4.2, is chosen over the Arnoldi process or IOM. The cost of evaluating the inner products is not significant, since the corresponding mass matrix is diagonal in the single-layer case and involves only a vertical translation between height and layer coordinates using the layer matrix T in the multilayer case. For each stage of the considered methods, matrix functions of $\boldsymbol{A}_n$ need to be computed for additional right-hand sides. Specifically, in the second stage the right-hand side is $\boldsymbol{b}_1 = \boldsymbol{F}[\boldsymbol{V}_n]$, and in each subsequent stage the right-hand side $\boldsymbol{b}_s = \boldsymbol{R}_n[\boldsymbol{v}_{n,s-1}]$ is introduced, where $\boldsymbol{v}_{n,s-1}$ is the previous

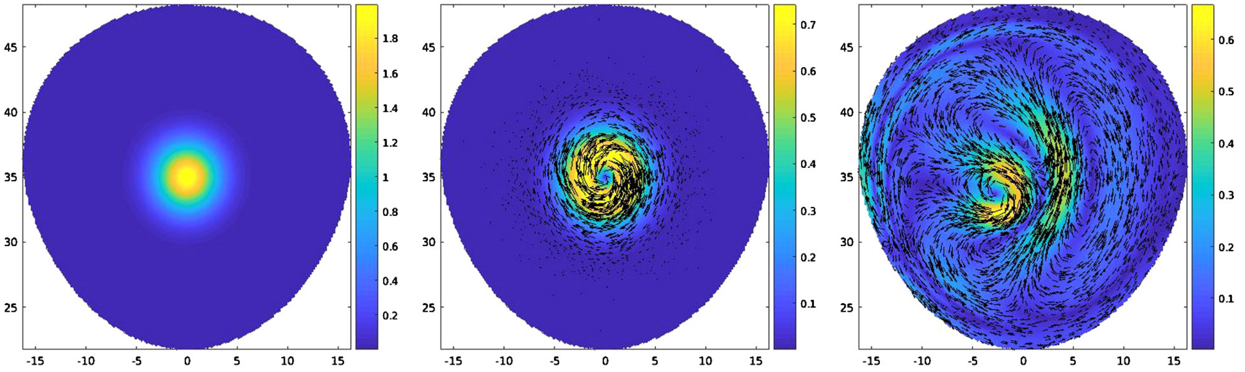

Fig. 2. Snapshots of the initial sea surface height $\eta_0 = h_0 + b$ [m], the initial velocity u_0 [m/s] and the velocity after ten days of simulation time for the geostrophic testcase. Latitude and Longitude are in degrees.

stage; cf. Appendix C. Additionally, possibly different matrix functions of $\boldsymbol{A}_n$ need to be applied to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{s-1}$. Here, the previous Krylov spaces can be reused; only the matrix function of the Hessenberg matrix needs to be recomputed. In cases where additional Krylov vectors are required, the Arnoldi process can be continued. Thus, for each additional stage, effectively one additional skew-Lanczos process needs to be computed, and the matrix exponentials of the Hessenberg matrix and linear combinations in (4.14) need to be updated at most $s-1$ times.

Additionally, we comment on the number of Krylov iterations per evaluation of a matrix φ-function. The theoretical estimates (see, e.g., [6, Section 4.2]) suggest that the required number of Krylov vectors effectively depends linearly on the Courant number, before an exponential rate of convergence sets in. In practice, we also employ the adaptive a posteriori error criterion suggested in [32], based on [33,34]. However, in the numerical experiments, we found that the convergence behavior suggested by theory was sharp: e.g., an error tolerance of 10^{-6} was usually met after $M \geq a_1\,\mathrm{C}(\Delta t) + a_2$ iterations, where appropriate constants $a_1 \approx 1.25$, $a_2 \approx 15$ were determined empirically. Moreover, using less than $\mathrm{C}(\Delta t)$ Krylov iterations usually lead to completely inaccurate solutions and even unstable simulations. This can be contrasted with an approximation of the matrix φ-function based on RK4 time stepping using (4.8), which requires at a minimum a number of $4/\sqrt{8} = \sqrt{2} \approx 1.41$ matrix multiplications per unit timestep (with $\mathrm{C}(\Delta t) = 1$), just to obtain basic stability, and then converges at fourth order.

6.2. *Discussion of results*

In the first two test cases, obtained using the single-layer configuration, the order of convergence and energy conservation of the ETD*S*wave methods are investigated. The third and final test case uses a three-layer configuration and a spin-up initial condition (over a ten year horizon), and investigates the performance and accuracy of the methods over a ten year simulation time, including additional forcing and biharmonic smoothing terms.

6.2.1. *Single-layer scenario*

The first test scenario is used to verify the accuracy and the energy conservation properties of the ETD*S*wave methods. For simplicity, this scenario is implemented using the single-layer configuration. We consider an unforced problem either without or with a minimal amount of biharmonic smoothing added to the problem. We consider two initial conditions corresponding to fast and slow modes of the single-layer equation, respectively. The initial condition for the height $h_0 = -b + \eta_0$ is a Gaussian perturbation of the stable reference height with $\eta_0 = \bar{\eta}\, \exp\left(-(x - x_{\mathrm{center}})^2/(2\sigma^2)\right)$, where the radius is $\sigma = 200$ km, the total perturbation height is $\bar{\eta} = 2$ m and x_{center} is the location at the center of the domain.

In the first case, this is combined with a zero initial velocity $u_0 = 0$. This then leads to a free-surface gravity wave emanating from the center of the domain. Over the simulation horizon of six hours the wave spreads out from the center of the domain, is reflected at the coastal boundaries, and roughly ends up back at the center of the domain. In the second case, the initial height is chosen in the same way. However, now the initial velocity is given as $u_0 = (g/f(x_{\mathrm{center}}))\,\hat{k} \times \nabla\eta_0$. This choice ensures that $\nabla \cdot u_0 = 0$ and the pressure gradient $g\nabla\eta_0$ balances the Coriolis force $f\hat{k} \times u_0$, which is referred to as *geostrophic balance*. The dynamics of this solution evolve on a slower time scale; a snapshot of the solution after ten days is given in Fig. 2. In the following, we will refer to the former as the *gravity wave*, and to the latter as the *geostrophic* testcase.

Convergence test. The errors of RK4 and various ETD*S*wave methods are computed using a reference solution computed with RK4 using a time step size of $\Delta t = (1/4)\Delta t_{\mathrm{C}}$. The time step sizes for the tested methods are chosen as $\Delta t = 2^j \Delta t_{\mathrm{C}}$, $j = 0, 1, \ldots, 7$, and the two values $\Delta t = 0.9\Delta t_{\mathrm{RK4}}$ and $\Delta t = 1.1\Delta t_{\mathrm{RK4}}$ are added to verify the stability region of RK4. Additionally, we consider the first-order ETDwave method, the second-order ETD2wave method with $c_2 = 1$ and $c_2 = 2/3$ and the third-order ETD3wave method (detailed in Appendix C), where the coefficients are chosen to be $(c_2, c_3) = (1/2, 3/4)$.

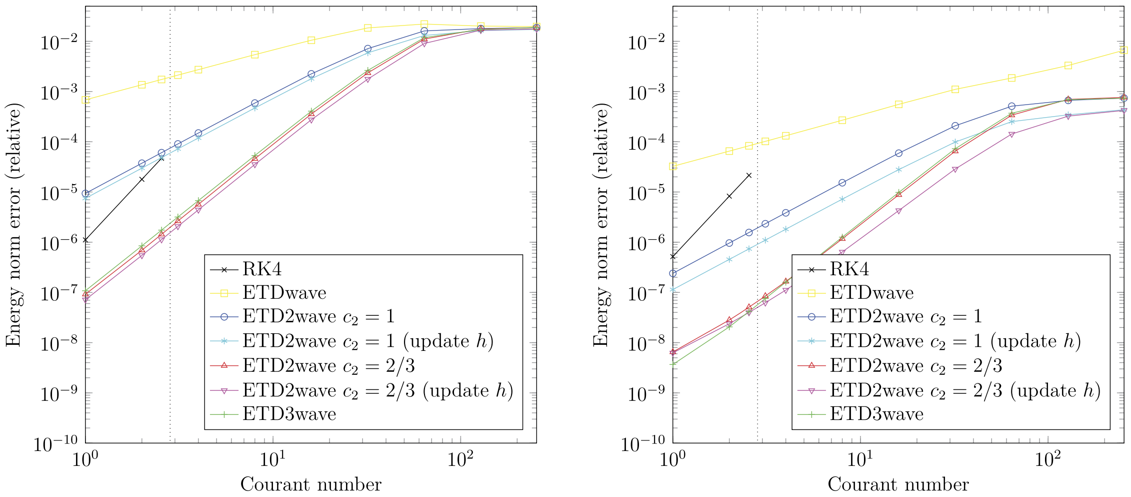

Fig. 3. The relative error in the solution (given in terms of the SSH $\eta = h + b$ and the velocities u) at the final time in a (linearized) energy norm for various methods and Courant numbers. The vertical dotted line denotes the maximal stable time step for RK4 at Courant number $\sqrt{8}$. Left: for the gravity wave initial condition $(h_0, 0)$ after six hours. Right: for the geostrophic initial condition (h_0, u_0) after ten days.

We also consider the impact of using updated heights for the ETD2wave methods. A small amount of biharmonic smoothing is added to the problem (see Appendix A.1) with horizontal viscosity $\boldsymbol{\nu}_{\mathrm{h}} = 2 \times 10^9$. Note that, for all experiments, we use a number of Krylov iterations set to $M = 1.3\,C(\Delta t) + 15$ (rounded to the nearest integer), which ensures an accurate evaluation of the matrix φ-functions; see section 6.1.3.

To compare the methods, in Fig. 3 we show the relative solution error at the final time in the discrete linearized energy norm as a function of the Courant number. The discrete linearized energy norm is induced by the mass matrix $\boldsymbol{M}_H = \boldsymbol{M}_{\chi^L} \delta^2 \boldsymbol{H}^{\mathrm{ref}}$ (cf. sections 2.3 and 4.2.1), and locally combines the weighted errors in heights and weights in a way that more closely matches their contribution to the total energy (up to a linearization error). We note that RK4 is unstable for time steps larger than Δt_{RK4} (as predicted by theory) but the ETD methods remain stable for all time steps considered. Within the regions of stability, the methods exhibit the expected convergence order. Only the ETD2wave method with $c_2 = 2/3$ is noteworthy, since it appears to have almost third-order convergence for a large regime of time step sizes. Moreover, we note that all ETD methods deliver solutions that are accurate up to the second significant digit for all time step sizes. Surprisingly, some of the second- and third-order ETD methods are more accurate than RK4 at the same time step size, despite being of lower order than RK4.

We can also notice that the ETD methods are more accurate for the geostrophic testcase, whereas the accuracy of RK4 is similar in both cases. Moreover, differences can be seen among the ETD methods: In the gravity wave test-case the error of all methods is similar at a Courant number of ca. 100. In the range $1 \leq \Delta t \leq 50$ a clear benefit in accuracy can be seen for the three stage method and ETD2wave with $c_2 = 2/3$ over the second- and first-order methods. Updating the reference height appears to only provide a marginal benefit in the first test-case, which can be attributed to the fact that the perturbation of the height compared to the stable reference height changes appreciably over each timestep, due to the fast free surface wave. In contrast, for the geostrophic test-case, we observe an improvement due to incorporating the current reference height $\boldsymbol{h}_n$ into $\boldsymbol{A}$. We note that this improvement is also present for large Courant numbers.

Performance test. From a practical point of view, the most interesting question is the performance of the methods. Thus, we also plot the errors as a function of the wall clock time, measured in simulated years per day (SYPD); see Fig. 4. Here, the purely explicit RK4 method has an advantage over the ETD methods at moderate time step sizes, since no φ-functions need to be evaluated. In fact, in the gravity wave testcase, we observe that RK4 outperforms the proposed ETD methods in terms of accuracy in the range of SYPD that it can achieve. In the geostrophic testcase, the ETD2wave methods with $c_2 = 2/3$ outperform RK4, using fewer but larger, more expensive time steps. In all cases, the maximal SYPD that can be achieved with RK4 is bounded by the maximal time step, whereas the ETD methods achieve higher SYPD in the large time step regime. However, we also notice that performance gains decrease at higher Courant numbers. In the gravity wave testcase, the simulation horizon is very short, so that SYPD actually decreases around Courant number 100. In the geostrophic case, where the horizon is longer, SYPD continues to increase with larger time steps, but at Courant number 200, the gain of doing fewer nonlinear forcing term evaluations per time interval is increasingly balanced by the increased cost of computing the φ-functions, since the Krylov method requires a number of iterations that is roughly proportional to the length of the time step. Note that this also highlights that the performance gains of the ETD methods exploit the fact that the cost of a matrix-vector product with $\boldsymbol{A}_n$ is cheaper than an evaluation of $\boldsymbol{F}$. For Rosenbrock-ETD methods, this is more difficult to achieve, since the Jacobian of every term in $\boldsymbol{F}$ also needs to be included in the linearization.

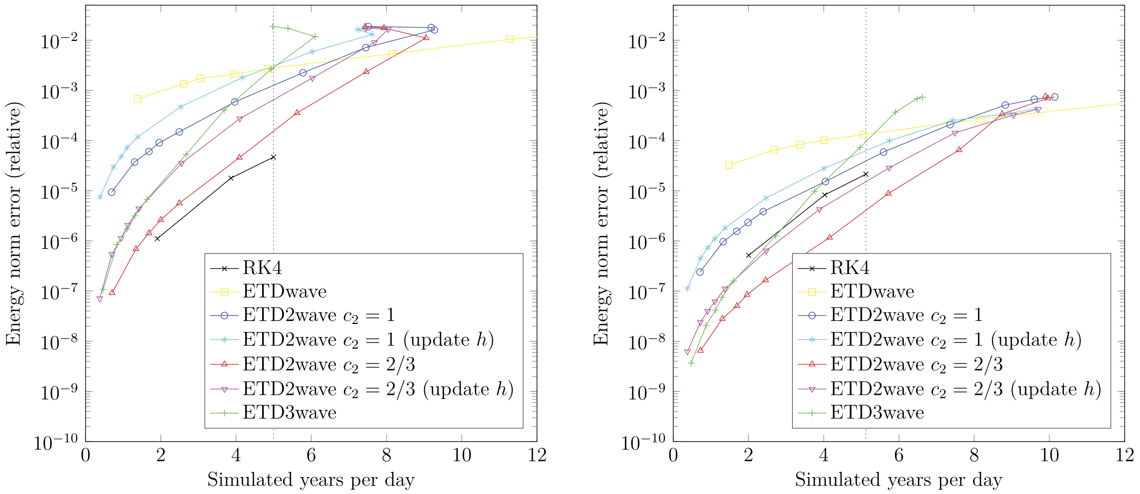

Fig. 4. The relative error in the solution (given in terms of the SSH $\eta = h + b$ and the velocities u) at the final time in the (linearized) energy norm for various methods as a function of the wall clock time, measured in simulated years per day (SYPD). The vertical dotted line denotes the maximal SYPD that could be obtained with RK4. Left: for gravity wave initial condition $(h_0, 0)$ after six hours. Right: for the geostrophic initial condition (h_0, u_0) after ten days.

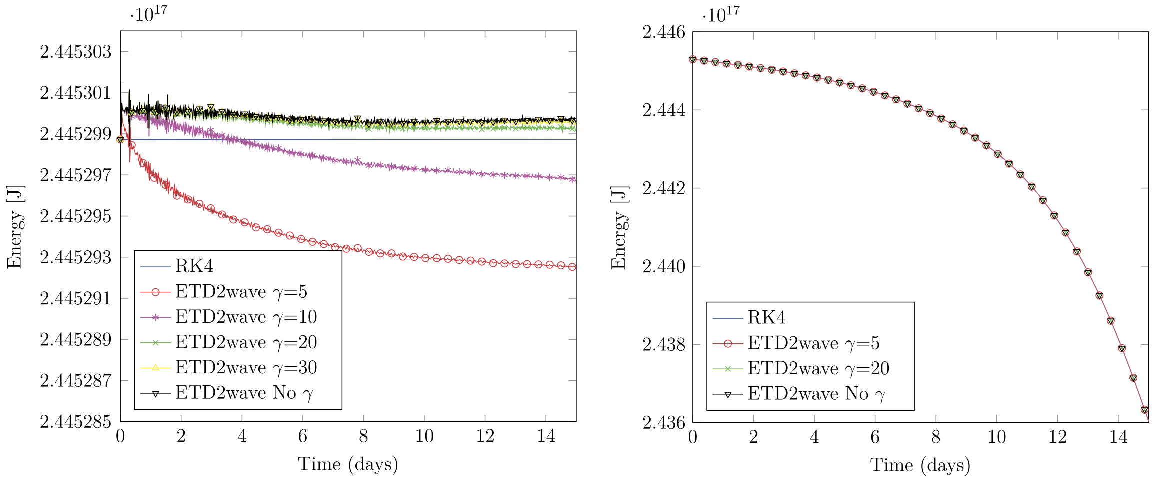

Fig. 5. Energy (2.6) for RK4 at Courant number $C(\Delta t) = (3/4)\sqrt{8} \approx 2.1$ and ETD at $C(\Delta t) = 10\sqrt{8} \approx 28.3$ for varying γ values over a time horizon of fifteen days. Left, without biharmonic smoothing. Right, with biharmonic smoothing.

Artificial dissipation test. The next test focuses on energy conservation in the ETD methods and the effect of the artificial numerical dissipation described in section 4.3. In particular, we investigate its effect on the total energy. Concretely, we fix the parameter $p = 2$, and consider different values of the spectral cut-off parameter γ. Again, we use the initial condition from Fig. 2, which is in geostrophic balance. First, the evolution of the energy from a simulation using RK4 close to the maximal time step $\Delta t_{\mathrm{RK4}} = \sqrt{8}\Delta t_{\mathrm{C}}$, is compared to the energy obtained using ETD with various γ values. The time step size for ETD2wave is chosen as $\Delta t = 10\Delta t_{\mathrm{RK4}}$ (using the reference heights, $c_1 = 1$, and $M = 45$). The values $\gamma \in \{5, 10, 20, 30\}$ are employed for ETD2wave, and also the unmodified case without artificial dissipation is considered. Secondly, we repeat the same test, but add a biharmonic smoothing to the model (see Appendix A.1) with horizontal viscosity $\boldsymbol{\nu}_{\mathrm{h}} = 2 \times 10^{10}$. This is motivated by the fact that the same term will be included in the decade long simulations in section 6.2.2. There, it provides a necessary turbulence closure, which prevents an unphysical build-up of vorticity in the finest grid cells. We note that the concrete viscosity value for this grid resolution is taken from [30].

In Fig. 5 we plot the evolution of the energy for all methods. Note that, in the case of additional biharmonic smoothing, the curves visually coincide. This shows that the energy dissipating effect of the biharmonic viscosity is stronger than either the time discretization error or the artificial numerical diffusion. Concerning the case without biharmonic smoothing, we observe that the ETD methods are affected by a larger time discretization error than RK4. This is not surprising, due to the much larger time step employed by these methods. Concerning the influence of γ, we observe that for the largest value of $\gamma = 30$ the energy is barely affected, which is explained by the fact that γ is bigger than the Courant number $C(\Delta t) \approx 28.3$.

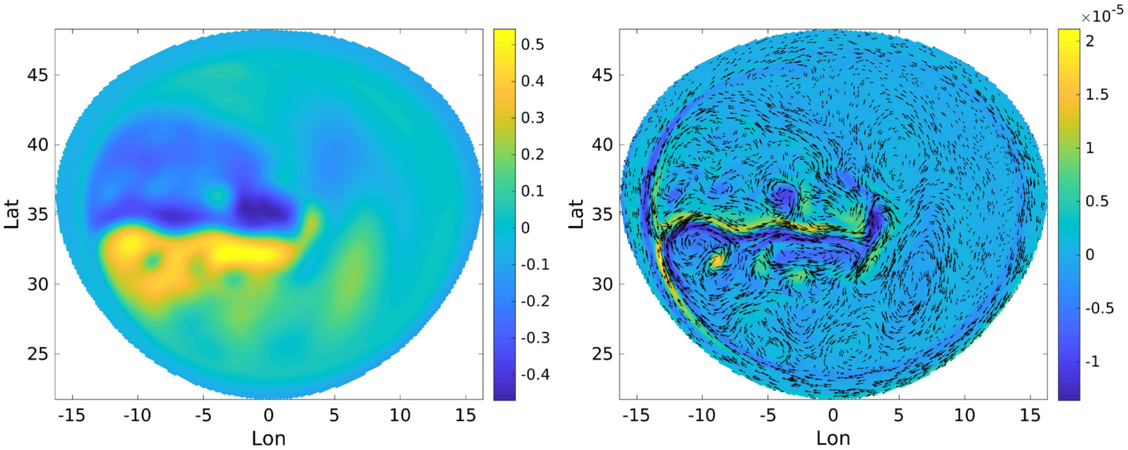

Fig. 6. Spin-up initial condition: Left, SSH. Right, velocity vectors [m/s] and relative vorticity $\hat{k} \cdot \nabla \times u$ [1/s] (density plot) after ten years in the top layer. Latitude and Longitude are in degrees.

For smaller γ, there is an increasing effect on the energy, which tends to be dissipative on average. However, we note that only for the smallest value of γ, the effect of the artificial dissipation is noticeably larger than the time discretization error.

6.2.2. Multilayer scenario

The second scenario is used to investigate the long term stability and accuracy of the methods over simulation horizons of decades. The scenario tries to represent a realistic simulation in the context of climate studies and, in addition to the bathymetry, shares the same forcing and smoothing terms as the SOMA test case in [30, Appendix A]. The wind stress τ_λ is in the easterly direction in the center of the domain in the westerly at the top and bottom of the domain. This induces a double-gyre mean circulation pattern. To extract energy from the system, a quadratic bottom drag with coefficient $c_{\text{drag}} = 10^{-3}$ is added. Also, a vertical Laplacian is implemented such that the bottom drag term can be interpreted as a Robin-like bottom boundary condition. The concrete form of these terms is given in Appendix A.1. The horizontal and vertical viscosities are set to $\boldsymbol{\nu}_{\text{h}} = 2 \times 10^{10}$ and $\boldsymbol{\nu}_{\text{v}} = 10^{-4}$, respectively (which are the values given in [30] for the 16 km grid). The three layer configuration for this scenario has initial layer interfaces located at $\eta_1^0 = 0$, $\eta_2^0 = -25/3$, and $\eta_3^0 = -50/3$ [km]. This evenly distributed layer configuration is chosen to avoid the possible out-cropping of layers (which refers to the vanishing of one of the layer heights on some part of the domain), which could lead to a breakdown of the simulation. The layer densities are set to $\rho = (1025, 1027, 1028)$ $[\text{kg}\,\text{m}^{-3}]$.

The initial condition is obtained from a ten year spin-up simulation initiated at the resting state and using RK4 with a time step of $\Delta t = (3/4)\Delta t_{\text{RK4}}$. The SSH and top layer velocity of the resulting spin-up initial condition are shown in Fig. 6. This process ensures that the system is in dynamic equilibrium, which means that the long-term statistics, such as the mean flow or the root mean square (RMS) of the sea surface height (SSH), have coherent structure. This is important since, over time horizons of years and longer, it is expected that the trajectories computed with different methods will drift apart. Thus, a comparison of instantaneous values of the solution becomes meaningless, and only the behavior of the long-term statistics can be used to assess the quality of the different time discretization methods. A second motivation for evaluating solution statistics is that climate-ocean models are concerned with long time scale changes, not instantaneous phenomena. Therefore, a method's ability to accurately predict these long-term statistics is important.

Results. We consider a simulation starting from the spin-up initial condition over the horizon of ten simulation years. We employ ETD2wave, B-ETD2wave, and B-ETD3wave (using the reference heights and $c_1 = 1$) with time steps increased above the maximal RK4 time step $\Delta t_{\text{RK4}} = \sqrt{8}\Delta t_{\text{C}} \approx 107.2$ [s] for the 16 km grid. For ETDwave we were not able to obtain stable simulations in any configuration. For ETD2wave the time step is increased 10 and 15 times, for B-ETD2wave 5 and 7 times, and for B-ETD3wave 10 and 12.5 times over Δt_{RK4}. Additionally, to avoid spurious high-frequency oscillations, the artificial dissipation from section 4.3 is employed, using $\gamma = 20$ and $p = 2$. We note that the larger time step for each ETD method reflects the largest time step that was empirically found to be stable over the entire time horizon in combination with a value of $\gamma = 20$.

For the purposes of comparison, two additional simulations are performed with RK4 at 1/4 and 3/4 the maximal time step, respectively. We note that global mass is conserved up to machine precision over the whole simulation horizon for all considered methods, as predicted by theory. Due to the wind forcing and smoothing, we can not expect conservation of energy. The global energy evolution for all methods is given in Fig. 7. From this, it is evident that the solutions differ significantly after the first years of simulation time. Therefore, we consider the statistical quantities mean flow and SSH RMS (to be precise, we compute the RMS of the deviation of the SSH from its temporal mean, which corresponds to the statistical variance). The mean and variance are approximated by the statistical mean and the sample variance, with snapshots taken

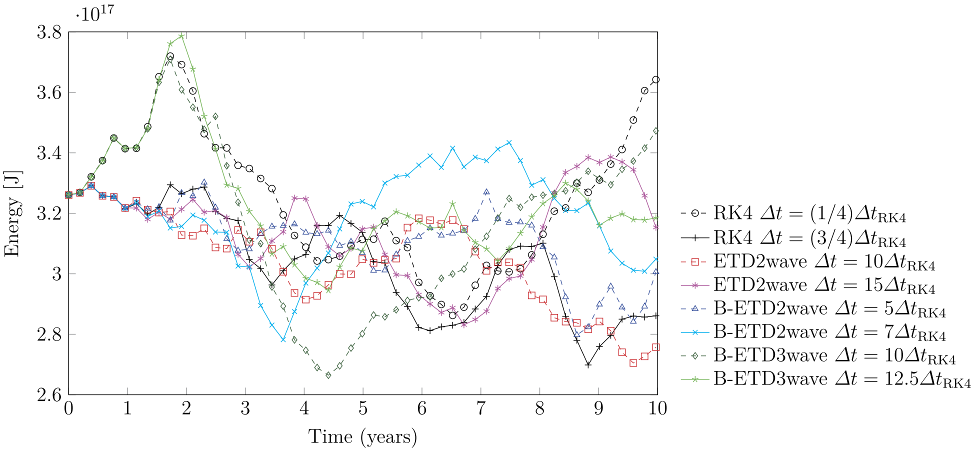

Fig. 7. Global energy evolution over ten years in the SOMA simulations for different methods.

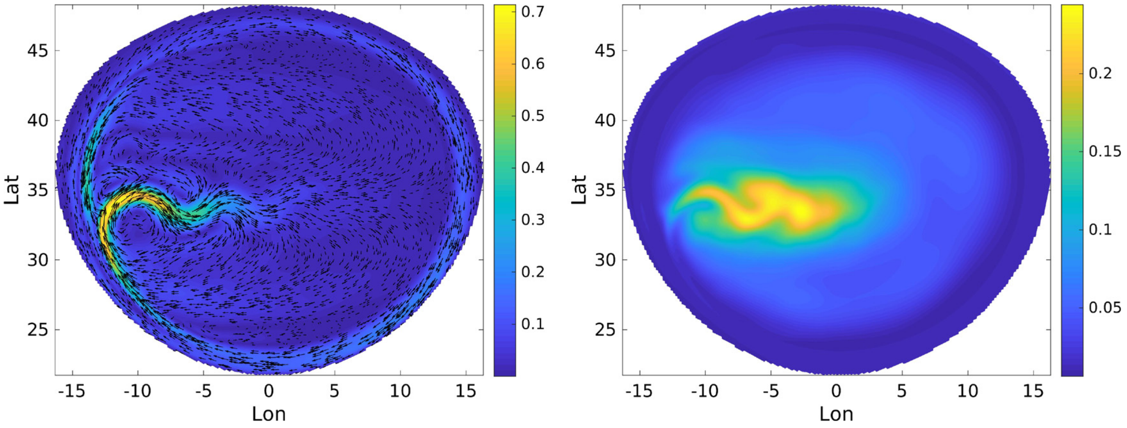

Fig. 8. SOMA simulation statistics computed with RK4: The mean flow of the velocity, vectors and magnitude [m/s] (left) and the RMS of the SSH [m] (right) computed as the sample mean and variance with snapshots of the solution taken every two weeks over ten years. Latitude and longitude are in degrees.

Table 1
Comparison of the ETD methods against the base-line RK4 method: the Courant number, the number of Krylov vectors M, the average simulated years per real day (SYPD), the relative mean velocity error in the $L^2_{h^{\mathrm{ref}}}$ norm, and the SSH RMS error in the L^∞ norm. Reference values are computed with using RK4 with $\Delta t = (1/4)\Delta t_{\mathrm{RK4}}$.

Method	$\Delta t/\Delta t_{\mathrm{RK4}}$	$C(\Delta t)$	M	SYPD	Mean-flow ($L^2_{h^{\mathrm{ref}}}$ rel.)	SSH RMS (L^∞)
RK4	3/4	2.12	–	0.911	0.0542	0.0541
ETD2wave	10	28.3	45	2.527	0.0598	0.0510
ETD2wave	15	42.4	63	2.771	0.0437	0.0603
B-ETD2wave	5	14.1	28	4.051	0.0462	0.0794
B-ETD2wave	7	19.5	35	4.609	0.0370	0.0714
B-ETD3wave	10	28.3	48	3.583	0.1372	0.0864
B-ETD3wave	12.5	35.4	56	4.204	0.0206	0.0716

every two weeks. The velocity and vorticity of the top layer mean flow, and the SSH RMS are shown in Fig. 8, which are computed from the RK4 simulation. Concretely, we compute the relative reference-thickness weighted L^2 error of the mean flow and the L^∞ error of the SSH RMS, which are given in Table 1. Here, we define the reference-thickness weighted $L^2_{h^{\mathrm{ref}}}$ norm for any velocity profile u by the square-root of $\sum_{k=1,\dots,L} \int_\Omega h^{\mathrm{ref}}_k u^2$, which roughly corresponds to the physical L^2 norm of the underlying three-dimensional velocity field.

Comparing the error in these quantities for each method, using the small time step RK4 solution as a base-line, we find that the results of the methods differ very little. We observe that all methods (including the RK4 simulation close to the CFL) reproduce the mean flow up to a similar tolerance of around 2% to 12%. This suggests that the bulk of the error is caused by replacing the true mean value by a sample average of an effectively random trajectory on a finite interval and not by the employed time discretization method. Concerning the maximum error in the SSH RMS, which appears to be a more sensitive criterion, the methods are reproduce the reference value up to ca. 5 to 9 cm, and thus range from ca. 20% to 35% relative accuracy. However, the accuracy is only slightly affected by the larger step-sizes, which suggests that all methods reproduce the chosen statistical quantities similarly accurate in this test. Moreover, the error does not necessarily increase with larger time steps for each method, which indicates that much of it could be attributed to statistical effects caused by the effectively random trajectories and sampling error introduced by approximating mean and variance by a finite number of samples. With regards to performance, Table 1 also shows the simulated years per real day of the various ETD methods and RK4. In all cases, ETD outperforms RK4 in terms of wall times. Moreover, the B-ETD2wave method outperforms the ETD2wave method, although it is forced to use a smaller time step, which is due to the cheaper φ-function computations. Moreover, we note that the barotropic method with three stages is able to take time steps of similar magnitude as ETD2wave, in contrast to the barotropic second order version. We attribute this to the fact that B-ETD2wave is based on RK2, which is not stable on the imaginary axis, but still has to resolve the baroclinic waves contained in the remainder. On the other hand, B-ETD3wave is based on RK3, which includes an imaginary interval in its stability region. This, combined with the fact that the cost of additional stages decreases for the barotropic method in configuration with more than three layers, will likely make it (or even higher order methods) favorable for configurations with more layers.

7. Conclusion

In this paper, we have developed ETD methods that can take big time steps for the multilayer shallow water equations and deliver sufficiently accurate solutions at a reduced cost compared to explicit methods. We have based this development specifically on the spatial TRiSK-scheme, but it applies directly to any scheme based on a Hamiltonian framework. More generally, it should be applicable to any scheme that conserves a discrete energy. This also includes classical finite difference/finite volume schemes on structured quad-meshes, used in ocean modeling.

In the following, we address the further steps that are needed to use these methods in order to improve current ocean models running on massively parallel architectures. Most currently employed ocean models use a splitting of the dynamics into a fast free-surface equation and a slow remainder, which usually speeds up simulations by an order of magnitude. This is achieved by solving the free-surface equation either with explicit or implicit time stepping methods. ETD methods are an attractive substitute, since they restrict the fast dynamics to be linear, and allow for different approximation methods, such as polynomial or rational Krylov methods. Moreover, methods of high order are available. In this paper we have only considered problems with up to three layers, such that the potential for computational speed-up exploiting the structure of the barotropic method was limited. Clearly, if more layers are added, the computational effort for the barotropic methods is going to decrease relative to RK4, since the reduction in size of the reduced linear operator is increased, and the amount of work to compute the φ_s functions remains independent of the number of layers. In such situations, larger time steps, higher order methods, and methods incorporating also the first baroclinic mode into the linear operator, which we did not consider in our numerics, may become increasingly competitive.

In order to reliably consider situations with more layers, it will be essential to make the model more realistic. For instance, the out-cropping of internal layers (layer heights going to zero) can no longer be avoided for thin layers. This can be addressed by leaving the isopycnal reference frame and considering the primitive equations in an arbitrary Lagrangian vertical coordinate system together with tracer equations (e.g., for temperature and salinity) and an equation of state. In the future, we aim to extend the methods to this case. Certainly, the development of monolithic ETD methods for the combined set of equations is desirable, but preliminary versions can be based upon an operator splitting into isopycnal dynamics and separate tracer plus ALE updates, which allow a more direct use of the developed methods. Moreover, the ETD methods should include an appropriate treatment of the tracer equations, and be able to scale to solve a large number of such equations efficiently. Here, exponential methods can also enable larger time steps, and a pre-computation of φ-functions for vertical transport may enable further efficiency.

Additional challenges arise due to the necessity of implementing these methods in massively parallel environments. Concerning the ETD methods proposed here, we first must note that Krylov methods require a “reduce-all” communication step in every iteration, which can be inefficient on certain parallel architectures. Here, other iterative approaches such as an approximation of the matrix functions using Chebyshev polynomials may be used instead, to avoid global communication. An additional problem of the iterative solution of the fast equation, which also plagues split-explicit methods, are the frequent communications with small message size and only a small number of floating point operations in between. This makes overlapping domain decomposition methods a promising alternative, due to the finite speed of propagation of the free surface waves. In this context, exponential methods can additionally exploit the linearity of the propagation matrices and the recursive relations between exponentials associated to time intervals of different length. In terms of incorporating ETD methods in existing computational ocean models, one may also consider them as a drop-in replacement for the single-layer barotropic solver. Here, the speed-up can not come from the layer reduction, but instead must come from the computational advantages of matrix exponentials of the linear operator over the existing implicit or explicit solution procedure.

Finally, global ocean models are expected to use increasingly nonuniform meshes of higher resolution near coastal boundaries, in order to more accurately resolve local features and interface with coastal/estuary models. As long as the bathymetry is sufficiently deep such that linear waves are still sufficiently faster than the advection, ETD methods using a global time step may still be effective. However, as the CFL requirements of the fast waves become less restrictive and the advective CFL becomes more so near the coast, smaller explicit time steps may be more beneficial. Here, the development of ETD methods

Table 2
The discrete operators given concretely in terms of geometrical quantities. d_e and l_e denote the distances between the cell centers and cell vertices, respectively. A_v and A_i are the triangle and cell areas, respectively. $R_{i,v}$ are the kite-areas, the intersection of a primal and dual grid cell divided by the cell area. $w_{e,e'}$ are the edge-weights from (A.1). The index sets in the summation correspond to geometrical connectivity arrays [12].

Divergence:	$(\nabla \cdot_{\mathsf{E}\to\mathsf{I}}\, \boldsymbol{y})_e = (1/A_i)\sum_{e\in\mathsf{EoI}(i)} n_{e,i} l_e \boldsymbol{y}_e$
Gradient:	$(\nabla_{\mathsf{I}\to\mathsf{E}}\, \boldsymbol{y})_i = (1/d_e)\sum_{i\in\mathsf{IoE}(e)} -n_{e,i} \boldsymbol{y}_i$
Curl:	$((\hat{k}\cdot\nabla\times)_{\mathsf{E}\to\mathsf{V}}\, \boldsymbol{y})_v = (1/A_v)\sum_{e\in\mathsf{EoV}(v)} t_{e,v} d_e \boldsymbol{y}_e$
Perpendicular Gradient:	$(\nabla^{\perp}{}_{\mathsf{V}\to\mathsf{E}}\, \boldsymbol{y})_e = (1/l_e)\sum_{v\in\mathsf{VoE}(e)} t_{e,v} \boldsymbol{y}_v$
Perpendicular Flux:	$(\hat{k}\times_{\mathsf{E}\to\mathsf{E}}\, \boldsymbol{y})_e = (1/d_e)\sum_{e'\in\mathsf{EoE}(e)} w_{e,e'} l_e \boldsymbol{y}_{e'}$
Cell to Vertex interpolation:	$\{\boldsymbol{y}\}_{\mathsf{V},v} = (1/A_v)\sum_{i\in\mathsf{CoV}(v)} R_{i,v} A_i \boldsymbol{y}_i$
Vertex to Edge interpolation:	$\{\boldsymbol{y}\}_{\mathsf{E},e} = \sum_{v\in\mathsf{VoE}(e)} \boldsymbol{y}_v/2$
Edge to Cell interpolation:	$\{\boldsymbol{y}\}_{\mathsf{I},i} = (1/A_i)\sum_{e\in\mathsf{EoI}(i)} \boldsymbol{y}_e A_e/2$
Cell to Edge interpolation:	$\{\boldsymbol{y}\}_{\mathsf{E},e} = \sum_{i\in\mathsf{IoE}(e)} \boldsymbol{y}_i/2$

that can take different time steps in different parts of the domain may provide a natural way of realizing that, since they degrade smoothly to explicit methods for smaller time steps.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge funding by the U.S. Department of Energy Office of Science grant DE-SC0016591.

Appendix A. TRiSK operators

In order to implement the TRiSK scheme several discrete operators are required for the differential operators and for averaging quantities from one grid location to another. In total there are eight operators required for the scheme that are built using the connectivity relations defined in [3, p. 6, table 2]. The operators consist of the divergence $\nabla\cdot_{\mathsf{E}\to\mathsf{I}}$, the gradient $\nabla_{\mathsf{I}\to\mathsf{E}}$, the gradient in the perpendicular direction $\nabla^{\perp}{}_{\mathsf{V}\to\mathsf{E}}$, the scalar curl $(\hat{k}\cdot\nabla\times)_{\mathsf{E}\to\mathsf{V}}$, the (perpendicular) flux reconstruction operator $\hat{k}\times_{\mathsf{E}\to\mathsf{E}}$, and the interpolation operators (see Table 2).

The definition of flux reconstruction operator $\hat{k}\times_{\mathsf{E}\to\mathsf{E}}$ given in [17], is a necessary condition for geostrophic balance. The ensures that Coriolis force and pressure gradient balance each other to maintain divergent free flows under the correct conditions. For a given Delaunay triangulation (and its CVT) on the sphere, along with the normals $n_{e,i}$ (elements of the matrix $\boldsymbol{N}$), tangents $t_{e,v}$ (elements of the matrix $\boldsymbol{T}$), kite areas (the intersections between triangles) $R_{i,v}A_i$ (elements of the matrix $\mathbf{R}$), the weights on edge array $w_{e,e'}$ (elements of the matrix $\boldsymbol{W}$) are then defined as quantities satisfying the relation

$$-\boldsymbol{T}\boldsymbol{W} = \boldsymbol{R}\boldsymbol{N}. \tag{A.1}$$

We note that the presented identities are only valid for a domain which has no boundary, e.g., the full sphere.

In order to obtain discrete operators on $\Omega \subset \mathbb{S}^2$, we follow the procedure used in the MPAS-O software and restrict a spherical mesh to a subset of the cells, which eliminates the “dry” cells of zero layer height. Note, that this results only in a first-order accurate resolution of the boundary. We obtain a discretization of the model on the bounded domain by fixing all velocity variables stored in edges adjacent to at least one “dry” cell to zero, which conveniently incorporates the no flux boundary conditions. However, since the edge-tangential velocity is reconstructed from the edge-normal velocity, and this reconstruction takes into account the zero edge velocities in boundary and “dry” edges, this implementation effectively introduces a full no-slip condition for the velocity. Thus, it is essential to employ additional diffusion terms such as (bi-)harmonic closure to obtain a mathematically meaningful model, since the shallow water equations are over-specified with no-slip conditions.

A.1. Choices of the forcing term

In wind-driven circulation, energy is injected at the ocean surface by a source term in the momentum equation. Concretely, the forcing term can be implemented as

$$\boldsymbol{f}_{\text{wind}}[\boldsymbol{h}] = \chi_{k=1}\, \boldsymbol{\tau}_{\text{wind}}/\{\rho_1 \boldsymbol{h}_1\}_{\mathsf{E}},$$

where $\boldsymbol{\tau}_{\mathrm{wind},e} \approx n_e \cdot \tau_{\mathrm{wind}}(x_e)$ is an edgewise approximation to the continuous wind profile, and the characteristic function $\chi_{k=1} \in \{0, 1\}$ ensures that wind forcing is only applied in the top layer.

Energy is typically extracted in the bottom layer, by a drag term that represents interaction of the flow with the (rough) bottom topography. A classical choice for this term is

$$\boldsymbol{f}_{\mathrm{drag}}[\boldsymbol{h}, \boldsymbol{u}] = -\chi_{k=l_{\mathrm{bot}}}\, c_{\mathrm{drag}} \frac{\{\sqrt{2\boldsymbol{K}[\boldsymbol{u}_{l_{\mathrm{bot}}}]}\}_{\mathsf{E}}}{\rho_{l_{\mathrm{bot}}} \{\boldsymbol{h}_{l_{\mathrm{bot}}}\}_{\mathsf{E}}} * \boldsymbol{u}_{l_{\mathrm{bot}}},$$

where l_{bot} is the bottom layer index. This corresponds to a quadratic drag term $-c_{\mathrm{drag}} \left| u_{l_{\mathrm{bot}}} \right| u_{l_{\mathrm{bot}}} / h_{l_{\mathrm{bot}}}$ in the continuous equation.

Due to the massive length scales relevant for global ocean modeling and the relatively coarse discretization, physical viscosities in the momentum equation are usually negligible. However, in order to account for the energy dissipated in scales below the grid resolution (due to turbulence), and to prevent a build-up of vorticity in finest grid scales, numerical dissipation terms have to be introduced to the discrete equation. Here, we employ a classical biharmonic viscosity, which is modified to be energetically consistent. Concretely, we choose

$$\boldsymbol{D}_{\mathrm{bihar}}[\boldsymbol{h}, \boldsymbol{u}] = -\frac{1}{\{\boldsymbol{h}\}_{\mathsf{E}}} * \widetilde{\boldsymbol{\Delta}}_{\mathsf{E}\to\mathsf{E}} \left(\boldsymbol{\nu}_{\mathrm{h}} * \{\boldsymbol{h}\}_{\mathsf{E}} * \left(\widetilde{\boldsymbol{\Delta}}_{\mathsf{E}\to\mathsf{E}} \boldsymbol{u} \right) \right),$$

where $\widetilde{\boldsymbol{\Delta}}_{\mathsf{E}\to\mathsf{E}} = \boldsymbol{\nabla}_{\mathsf{I}\to\mathsf{E}}\, \boldsymbol{\nabla}\boldsymbol{\cdot}_{\mathsf{E}\to\mathsf{I}} - \sqrt{3}\, \boldsymbol{\nabla}^{\perp}{}_{\mathsf{V}\to\mathsf{E}} (\hat{k} \cdot \boldsymbol{\nabla} \boldsymbol{\times})_{\mathsf{E}\to\mathsf{V}}$ is a discrete approximation to an anisotropic vectorial Laplace-Beltrami operator (see [17]).

The appearance of $\boldsymbol{h}$ is motivated by the form of physical viscosities in the shallow water equation (see, e.g., [35]), and the fact that the concrete form given above leads to consistent energy dissipation in the discrete equation. In fact, combining these terms by a choice of

$$\boldsymbol{G}[\boldsymbol{h}, \boldsymbol{u}] = \boldsymbol{D}_{\mathrm{bihar}}[\boldsymbol{h}, \boldsymbol{u}] + \boldsymbol{f}_{\mathrm{drag}}[\boldsymbol{h}, \boldsymbol{u}] + \boldsymbol{f}_{\mathrm{wind}}[\boldsymbol{h}],$$

we obtain for (3.1) the energy equality

$$\frac{\mathrm{d}\boldsymbol{H}}{\mathrm{d}t}(\boldsymbol{V}) = -\sum_{k=1}^{L} \rho_k (\boldsymbol{\nu}_{\mathrm{h}} * \{\boldsymbol{h}_k\}_{\mathsf{E}} * \widetilde{\boldsymbol{\Delta}}_{\mathsf{E}\to\mathsf{E}} \boldsymbol{u}_k, \widetilde{\boldsymbol{\Delta}}_{\mathsf{E}\to\mathsf{E}} \boldsymbol{u}_k)_{\mathsf{E}} - c_{\mathrm{drag}} (\{\sqrt{2\boldsymbol{K}[\boldsymbol{u}_{l_{\mathrm{bot}}}]}\}_{\mathsf{E}} * \boldsymbol{u}_{l_{\mathrm{bot}}}, \boldsymbol{u}_{l_{\mathrm{bot}}})_{\mathsf{E}} + (\boldsymbol{u}_k, \boldsymbol{\tau}_{\mathrm{wind}})_{\mathsf{E}},$$

which shows that the smoothing and damping terms are energy dissipating. The horizontal viscosity $\boldsymbol{\nu}_{\mathrm{h}} \in X_{\mathsf{I}}$ is usually chosen in a grid dependent fashion. However, since we only employ quasi-uniform grids, we set it to a constant in computations.

Additionally, in the multilayer case a vertical smoothing can be introduced in the momentum equation in the form of a vertical Laplacian. This can be based on a mimetic discretization of a vertical gradient and divergence. Since the vertical mesh size is given by the layer thicknesses $\boldsymbol{h}_k$, the vertical Laplacian will depend non-linearly on the variable $\boldsymbol{h}$. However, for the sake of brevity, we omit a detailed presentation. We only note that in this case, the drag term given above can also be interpreted as a Robin-like boundary condition for the vertical Laplacian.

Appendix B. Linearized operators

For convenience, we give the explicit form of the differential operators defined in section 3.2. The second variation of the Hamiltonian (3.5) can be computed as the linearization of (3.3) as

$$\delta^2 \boldsymbol{H}[\boldsymbol{V}]\boldsymbol{W} = \delta \boldsymbol{H}'[\boldsymbol{V}; \boldsymbol{W}] = \begin{pmatrix} g \left(R \boldsymbol{w}^h \right)_k + \rho_k \{\boldsymbol{u}_k * \boldsymbol{w}_k^u\}_{\mathsf{I}} \\ \rho_k \boldsymbol{u}_k * \{\boldsymbol{w}_k^h\}_{\mathsf{E}} + \rho_k \{\boldsymbol{h}_k\}_{\mathsf{E}} * \boldsymbol{w}_k^u \end{pmatrix}_{k=1,2,\ldots,L},$$

for all $\boldsymbol{W} = (\boldsymbol{w}^h, \boldsymbol{w}^u)$. In the case that $\boldsymbol{V} = \boldsymbol{V}^{\mathrm{ref}} = (\boldsymbol{h}^{\mathrm{ref}}, \boldsymbol{0})$, it can be represented by the block-diagonal matrix

$$\delta^2 \boldsymbol{H}[\boldsymbol{V}^{\mathrm{ref}}] = \begin{pmatrix} gR & 0 \\ 0 & \mathrm{diag}_{k=1,2,\ldots,L}(\rho_k \{\boldsymbol{h}_k^{\mathrm{ref}}\}_{\mathsf{E}}) \end{pmatrix}.$$

The linearization of $\boldsymbol{J}$ from (3.4) is given for all $\boldsymbol{W} = (\boldsymbol{w}^h, \boldsymbol{w}^u)$ as

$$\boldsymbol{J}'[\boldsymbol{V}; \boldsymbol{W}] = \mathrm{diag}_{k=1,2,\ldots,L} \frac{1}{\rho_k} \begin{pmatrix} 0 & 0 \\ 0 & \boldsymbol{Q}'[\boldsymbol{V}_k; \boldsymbol{W}_k] \end{pmatrix},$$

where

$$\boldsymbol{Q}'[\boldsymbol{V}_k; \boldsymbol{W}_k]\boldsymbol{y} = \frac{1}{2} \left(\{\boldsymbol{q}'[\boldsymbol{V}_k; \boldsymbol{W}_k]\}_{\mathsf{E}} * \left(\underset{\mathsf{E}\to\mathsf{E}}{\hat{k} \boldsymbol{\times}} \boldsymbol{y} \right) + \underset{\mathsf{E}\to\mathsf{E}}{\hat{k} \boldsymbol{\times}} \left(\{\boldsymbol{q}'[\boldsymbol{V}_k; \boldsymbol{W}_k]\}_{\mathsf{E}} * \boldsymbol{y} \right) \right)$$

and $\boldsymbol{q}'[\boldsymbol{V}_k; \boldsymbol{W}_k] = ((\hat{k}\cdot\nabla\times)_{\mathsf{E}\to\mathsf{V}}\, \boldsymbol{w}^u_k)/\{\boldsymbol{h}_k\}_{\mathsf{V}} - ((\hat{k}\cdot\nabla\times)_{\mathsf{E}\to\mathsf{V}}\,\boldsymbol{u}_k + \boldsymbol{f}) * \{\boldsymbol{w}^h_k\}_{\mathsf{V}}/\{\boldsymbol{h}_k\}_{\mathsf{V}}^2$. We remark that due the sparsity-pattern of $\hat{k}\times_{\mathsf{E}\to\mathsf{E}}$, which has the most entries of any of the discrete operators considered (apart from the biharmonic smoothing term), this term is expensive to evaluate in practice.

Appendix C. Exponential Runge-Kutta schemes

Exponential integrators can be given in terms of their Butcher tableau, which contains the intermediate time points c_i, the coefficients for the internal stages $a_{i,j}$, and the final coefficients b_j in the form

$$\begin{array}{c|c} c_i & a_{i,j} \\ \hline & b_j \end{array} \quad ,$$

which represents the method in terms of the remainder as

$$\boldsymbol{v}_{n,i} = \exp(c_i \Delta t \boldsymbol{A}_n)\boldsymbol{V}_n + \Delta t \sum_{j=1,\dots,i-1} a_{i,j}(\Delta t \boldsymbol{A}_n)\boldsymbol{r}_n(\boldsymbol{v}_{n,j}),$$

$$\boldsymbol{V}_{n+1} = \exp(\Delta t \boldsymbol{A}_n)\boldsymbol{V}_n + \Delta t \sum_{j=1,\dots,S} b_j(\Delta t \boldsymbol{A}_n)\boldsymbol{r}_n(\boldsymbol{v}_{n,j}),$$

where $c_1 = 0$ and $a_{i,j} = 0$ for $j \geq i$ for explicit methods, which implies $\boldsymbol{v}_{n,1} = \boldsymbol{V}_n$. Under the simplifying assumptions $\sum_{j=1,\dots,S} b_j = \varphi_1$ and $\sum_{j=1,\dots,i-1} a_{i,j} = \varphi_1(c_i\cdot)$, these methods can be equivalently rewritten in terms of the residual as:

$$\boldsymbol{v}_{n,i} = \boldsymbol{V}_n + \varphi_1(c_i \Delta t \boldsymbol{A}_n)\boldsymbol{F}(\boldsymbol{V}_n) + \Delta t \sum_{j=2,\dots,i-1} a_{i,j}(\Delta t \boldsymbol{A}_n)\boldsymbol{R}_n(\boldsymbol{v}_{n,j}),$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \varphi_1(\Delta t \boldsymbol{A}_n)\boldsymbol{F}(\boldsymbol{V}_n) + \Delta t \sum_{j=2,\dots,S} b_j(\Delta t \boldsymbol{A}_n)\boldsymbol{R}_n(\boldsymbol{v}_{n,j}).$$

The two stage method from section 4.1 taken from [13] is characterized by the Butcher tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & c_2\varphi_1(c_2\cdot) & 0 \\ \hline & \varphi_1 - (1/c_2)\varphi_2 & (1/c_2)\varphi_2 \end{array} \quad ,$$

i.e. $a_{2,1}(\cdot) = c_2\varphi_1(c_2\cdot)$, $b_1 = \varphi_1 - (1/c_2)\varphi_2$, and $b_2 = (1/c_2)\varphi_2$, where $c_2 \in (0,1)$. A third-order three stage method (also taken from [13]) is given by the Butcher tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & c_2\varphi_1(c_2\cdot) & 0 & 0 \\ c_3 & c_3\varphi_1(c_3\cdot) - a_{3,2} & \gamma c_2\varphi_2(c_2\cdot) + (c_3^2/c_2)\varphi_2(c_3\cdot) & 0 \\ \hline & \varphi_1 - b_2 - b_3 & (\gamma/(\gamma c_2 + c_3))\,\varphi_2 & (1/(\gamma c_2 + c_3))\,\varphi_2 \end{array} \quad ,$$

where either $c_3 = 2/3$, $c_2 \in (0,1)$, and $\gamma = 0$ or $c_2, c_3 \in (0,1) \setminus \{2/3\}$ and $\gamma = -(3c_3^2 - 2c_3)/(3c_2^2 - 2c_2)$.

References

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

[2] R.L. Higdon, A two-level time-stepping method for layered ocean circulation models: further development and testing, J. Comput. Phys. 206 (2005) 463–504.

[3] T. Ringler, M. Petersen, R.L. Higdon, D. Jacobsen, P.W. Jones, M. Maltrud, A multi-resolution approach to global ocean modeling, Ocean Model. 69 (2013) 211–232.

[4] R.V. Madala, Efficient time integration schemes for atmosphere and ocean models, in: D.L. Book (Ed.), Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations, Springer-Verlag, New York, Berlin, 1981, pp. 56–74.

[5] R.L. Higdon, Numerical modelling of ocean circulation, Acta Numer. 15 (2006) 385–470.

[6] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.

[7] R. Archibald, K.J. Evans, J. Drake, J.B. White, Multiwavelet discontinuous Galerkin-accelerated Exact Linear Part (ELP) method for the shallow-water equations on the cubed sphere, Mon. Weather Rev. 139 (2011) 457–473.

[8] C. Clancy, J.A. Pudykiewicz, On the use of exponential time integration methods in atmospheric models, Tellus A 65 (2013).

[9] S. Gaudreault, J.A. Pudykiewicz, An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere, J. Comput. Phys. 322 (2016) 827–848.

[10] V.T. Luan, J.A. Pudykiewicz, D.R. Reynolds, Further development of efficient and accurate time integration schemes for meteorological models, J. Comput. Phys. 376 (2019) 817–837.

[11] J. Thuburn, T. Ringler, W. Skamarock, J. Klemp, Numerical representation of geostrophic modes on arbitrarily structured C-grids, J. Comput. Phys. 228 (2009) 8321–8335.

[12] T.D. Ringler, J. Thuburn, J.B. Klemp, W.C. Skamarock, A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids, J. Comput. Phys. 229 (2010) 3065–3090.

[13] M. Hochbruck, A. Ostermann, Explicit exponential Runge–Kutta methods for semilinear parabolic problems, SIAM J. Numer. Anal. 43 (2005) 1069–1090.

[14] A.L. Stewart, P.J. Dellar, An energy and potential enstrophy conserving numerical scheme for the multi-layer shallow water equations with complete Coriolis force, J. Comput. Phys. 313 (2016) 99–120.

[15] J.K. Dukowicz, Structure of the barotropic mode in layered ocean models, Ocean Model. 11 (2006) 49–68.

[16] D.B. Chelton, R.A. deSzoeke, M.G. Schlax, K.E. Naggar, N. Siwertz, Geographical variability of the first baroclinic Rossby radius of deformation, J. Phys. Oceanogr. 28 (1998) 433–460.

[17] J. Thuburn, C.J. Cotter, A framework for mimetic discretization of the rotating shallow-water equations on arbitrary polygonal grids, SIAM J. Sci. Comput. 34 (2012) B203–B225.

[18] J. Niesen, W.M. Wright, Algorithm 919: a Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators, ACM Trans. Math. Softw. 38 (2012) 22.

[19] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (2003) 3–49.

[20] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 29 (1992) 209–228.

[21] L. Bergamaschi, M. Caliari, M. Vianello, The ReLPM exponential integrator for FE discretizations of advection-diffusion equations, in: International Conference on Computational Science, Springer, 2004, pp. 434–442.

[22] A.Y. Suhov, An accurate polynomial approximation of exponential integrators, J. Sci. Comput. 60 (2014) 684–698.

[23] T.S. Haut, T. Babb, P.G. Martinsson, B.A. Wingate, A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator, IMA J. Numer. Anal. 36 (2015) 688–716.

[24] R.B. Sidje, Expokit: a software package for computing matrix exponentials, ACM Trans. Math. Softw. 24 (1998) 130–156.

[25] V. Faber, T. Manteuffel, Necessary and sufficient conditions for the existence of a conjugate gradient method, SIAM J. Numer. Anal. 21 (1984) 352–362.

[26] C. Greif, J. Varah, Iterative solution of skew-symmetric linear systems, SIAM J. Matrix Anal. Appl. 31 (2009) 584–601.

[27] H.D. Vo, R.B. Sidje, Approximating the large sparse matrix exponential using incomplete orthogonalization and Krylov subspaces of variable dimension, Numer. Linear Algebra Appl. 24 (2017).

[28] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 34 (1997) 1911–1925.

[29] Q. Chen, M. Gunzburger, T. Ringler, A scale-invariant formulation of the anticipated potential vorticity method, Mon. Weather Rev. 139 (2011) 2614–2629.

[30] P.J. Wolfram, T.D. Ringler, M.E. Maltrud, D.W. Jacobsen, M.R. Petersen, Diagnosing isopycnal diffusivity in an eddying, idealized midlatitude ocean basin via Lagrangian, in situ, global, high-performance particle tracking (LIGHT), J. Phys. Oceanogr. 45 (2015) 2114–2133.

[31] T. Ringler, L. Ju, M. Gunzburger, A multiresolution method for climate system modeling: application of spherical centroidal Voronoi tessellations, Ocean Dyn. 58 (2008) 475–498.

[32] M.A. Botchev, Krylov subspace exponential time domain solution of Maxwell's equations in photonic crystal modeling, J. Comput. Appl. Math. 293 (2016) 20–34.

[33] E. Celledoni, I. Moret, A Krylov projection method for systems of ODEs, Appl. Numer. Math. 24 (1997) 365–378.

[34] V. Druskin, A. Greenbaum, L. Knizhnerman, Using nonorthogonal Lanczos vectors in the computation of matrix functions, SIAM J. Sci. Comput. 19 (1998) 38–54.

[35] D. Bresch, Chapter 1 – Shallow-water equations and related topics, in: C. Dafermos, M. Pokorný (Eds.), Handbook of Differential Equations: Evolutionary Equations, vol. 5, in: Handbook of Differential Equations, North-Holland, 2009, pp. 1–104.