Exponential time differencing for mimetic multilayer ocean models

Konstantin Pieper\textsuperscript{1}, K. Chad Sockwell\textsuperscript{1,2}, and Max Gunzburger\textsuperscript{1}

\textsuperscript{1}Florida State University, Dept. Scientific Computing, 400 Dirac Science Library, Tallahassee FL 32306
(kpieper@fsu.edu, kcs12j@my.fsu.edu, mgunzburger@fsu.edu)
\textsuperscript{2}Los Alamos National Laboratory, P.O. Box 1663, T-3, MS-B216, Los Alamos, NM 87545
(sockwell@lanl.gov)

14 Dec 2018

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 describing global or regional oceanic circulation remains challenging. This is due to the fact that the underlying partial differential 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 hyperbolic problems – 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. [10]. 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 [15, 24] 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 [20] and [16, Section 5].

Recently, exponential integrators (see, e.g., [19]), also called exponential time differencing methods (ETD), have gained attention in the context of circulation models [7, 12]. 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 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. Concretely, we work with the TRISSK scheme [31, 25], 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., [19, 18]), developed for semi-linear equations of the form

$$\partial_t V = F(V) = AV + r(V)$$

with an appropriate choice of the linear operator $A$. Here, we prefer an approximation to the Jacobian of the forcing term over the full Jacobian $F'$ (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 $A$ 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 $\varphi$-functions, which are required for the implementation of an ETD method. Moreover, the matrix exponential $\exp(\Delta t A)$ maintains the linearized energy of the RSWE for all $\Delta t$, which improves numerical stability for large time-steps. Since $A$ 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 $A$. In a typical configuration of an global ocean model, the difference in the free-surface speed and speed of internal gravity waves is greater than 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 built-up of turbulent energy in the smallest (grid-level) scales. Here, we employ a variant of the classical bi-harmonic 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 $r$. However, since $A$ does not take into account the dissipation, this can lead to a spurious built-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 $\varphi$-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 $A$.

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 [24]. For the sake of readability, we first explain the single layer model, and then the multi-layer 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 $S^2$ the two-sphere with outward oriented surface-orthogonal unit vector $\hat{k}$, and by $\Omega$ 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 $h: \mathbb{R} \times \Omega \rightarrow \mathbb{R}$ and the velocity $u: \mathbb{R} \times \Omega \rightarrow \mathbb{R}^3$ in the vector-invariant form as

\begin{equation}
\begin{aligned}
\partial_t h &= -\nabla \cdot (hu) \quad \text{in } \Omega, \\
\partial_t u &= -\nabla \left( K[u] + g(h + b) \right) - q[h, u] (hu^\perp) + G[h, u] \quad \text{in } \Omega,
\end{aligned}
\end{equation}

(2.1)

together with the constraint that the velocity should be tangential to the surface $u \cdot \hat{k} = 0$ in $\Omega$, 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, $u^\perp = \hat{k} \times u$ the perpendicular velocity, $q[h, u] = (\hat{k} \cdot \nabla \times u + f)/h$ is the potential vorticity (PV) with $f$ the Coriolis parameter, $b < 0$ is the bathymetry, which encodes the bottom topography. The differential operators are defined in the canonical way on $S^2$. The term $G(h, u)$ 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 $\hat{k} \cdot G(h, u) = 0$, 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 $G \equiv 0$). Consider the total energy over the domain as given by the Hamiltonian

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

(2.2)

Furthermore, introduce a skew-symmetric operator $J$ given formally by

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

(2.3)

In the following, we abbreviate the solution variables by $V = (h,u)$. Furthermore, we endow the solution space $L^2(\Omega) \times L^2(\Omega, \mathbb{R}^3)$ by its canonical Hilbertian structure. The shallow water equations can then be formed using the functional derivative of $H$ given as

$$\frac{\delta H}{\delta V} = \frac{\delta^2 H}{\delta V^2}.$$ 

(2.4)

which given mathematically as the Hilbert space Gâteaux derivative of the energy functional. Note that we abbreviate the functional derivative by $\delta H[V]$, 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 $\delta^2 H[V]$ where $\delta^2 H[V] W = (\delta H)[V; W]$.

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

$$\partial_t V = J[V] \delta^2 H[V] + \left( \begin{array}{c} 0 \\ G[h,u] \end{array} \right).$$

(2.5)

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.

### 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 multi-layer system, each layer’s density is set a priori and is considered the average density of the layer. The density increasing with water depth in order to produce a stable configuration. Knowing the density $\rho$ of each layer a priori gives a convenient set of Lagrangian coordinates, namely the isopycnal contours that separate the layers with different densities. The isopycnal lines on the top and bottom of the layer can then be used to define the midpoint of the layer, which changes in time due to the dynamical behavior of the isopycnal lines. 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 $\rho_k$, with $\rho_1$ being the density for the top layer and $\rho_{k-1} < \rho_k$, $k \leq L$. The velocities $u_k$ and $h_k$ are now defined for each layer, and the corresponding layer coordinates $\eta_k$ are defined by

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

A visualization is given in Figure 1. In the following, we denote by $u$, $h$ and $\eta$ the vectors containing all layer variables, and by $V = (h, u)$ and $V_k = (h_k, u_k)$ the combined solution variable. The multi-layer rotating shallow water equations for $k$ layers are now given by

$$\begin{cases}
\partial_t h_k = -\nabla \cdot (h_k u_k) & \text{in } \Omega, \\
\partial_t u_k = -\nabla \left( K[u_k] + (g/\rho_k) p_k[h] \right) - q[h_h, u_k](h_k h_k^\perp) + G_k(h, u) & \text{in } \Omega,
\end{cases} \tag{2.6}$$

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

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

The Hamiltonian framework can be extended for the multi-layer case (see, e.g., in [28]). The multi-layer Hamiltonian $\mathcal{H}$ can be defined as

$$\mathcal{H}(h, u) = \sum_{k=1}^{L} \rho_k \int_{\Omega} (h_k K[u_k] + g h_k (\eta_{k+1}[h] + h_k/2)) \, . \tag{2.7}$$

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 with diagonal entries

$$\mathcal{J}_k[V] = \frac{1}{\rho_k} \mathcal{J}[h_k, u_k], \quad k = 1, \ldots, L,$$
of the scaled single layer operator on the diagonal. Concretely, in a weak formulation this corresponds to
\[
(W, J[V]Y) = \sum_{k=1}^{L} \frac{1}{\rho_k} (W_k, J_k[V_k]V_k)
\]
\[
= -\sum_{k=1}^{L} \frac{1}{\rho_k} \int_{\Omega} \left( w_k^h \nabla \cdot y_k^u + w_k^u \cdot \nabla y_k^h + q[h_k, u_k] w_k^u \cdot \hat{k} \times y_k^h \right),
\]
where \( W = (w_k^h, w_k^u)_{k=1,...,L} \) and \( Y = (y_k^h, y_k^u)_{k=1,...,L} \). The concrete form of (2.6) can now be derived from (2.5), 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
\[
\tilde{H}(h, u) = \sum_{k=1}^{L} \int_{\Omega} \left( \rho_k h_k K[u_k] + \frac{g}{2} \Delta \rho_k (\eta_k[h])^2 \right),
\]
where \( \Delta \rho_k = \rho_k - \rho_{k-1} \) and the zeroth density is defined as \( \rho_0 = 0 \) for convenience. Further structure can be exposed by introducing the summation matrix \( T \) with entries
\[
T_{k,j} = \begin{cases} 1 & \text{if } k \leq j, \\ 0 & \text{else}, \end{cases}
\]
which 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 (2.8) as
\[
\frac{g}{2} \sum_{k=1}^{L} \Delta \rho_k (\eta_k[h])^2 = \frac{g}{2} \| b + Th \|_{2\Delta \rho}^2,
\]
where \( \| \eta \|_{2\Delta \rho}^2 = \sum_{k=1}^{L} \Delta \rho_k \eta_k^2 \) corresponds to a weighted Euclidean norm. Taking the functional derivative of this term, we obtain \( g T^\top \text{diag}(\Delta \rho)(Th + b) = g \tilde{p}_k[h] \), and the corresponding pressure can be rewritten as:
\[
\tilde{p}_k[h] = \rho_1 \eta_1[h] + \sum_{l=2}^{k} (\rho_l - \rho_{l-1}) \eta_l[h] \\
= \rho_k \eta_k[h] + \sum_{l=1}^{k-1} \rho_l h_l.
\]
Thus, both Hamiltonians lead to the same pressure and it holds \( \delta H = \delta \tilde{H} \). Consequently, both Hamiltonians are equal up to a constant value (i.e. \( \tilde{H} = H + \text{const} \)).

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 perturbation of the internal layers. This gives rise to the well-known fast barotropic mode, and the remaining baroclinic mode(s); see, e.g., [9]. We give an independent exposition, which is relevant for the paper, in the next section.
2.3 Linearization of the model and modes

We perform a linearized stability analysis of (2.6) for $G = 0$; cf. also [9]. Based on the Hamiltonian formalism, the linearized equation for a perturbation $V = V_{\text{ref}} + W$ can be written as

$$\partial_t W = \mathcal{J}_{\text{ref}}[V_{\text{ref}}; W] \delta \mathcal{H}[V_{\text{ref}}] + \mathcal{J}[V_{\text{ref}}] \delta^2 \mathcal{H}[V_{\text{ref}}] W, \quad W(0) = W_0$$

(2.9)

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 $V$, which in turn contains the derivatives of the potential vorticity $q[h, u]$ with respect to the solution variables. We note that for any $V_{\text{ref}} = (h_{\text{ref}}, u_{\text{ref}})$ with $u_{\text{ref}} = 0$, it holds

$$\partial_t W = \mathcal{J}[V_{\text{ref}}] \delta^2 \mathcal{H}[V_{\text{ref}}] W, \quad \text{for } V_{\text{ref}} = (h_{\text{ref}}, 0),$$

(2.10)

since the first term is zero. This system has again a Hamiltonian structure, with a fixed $\mathcal{J}$-operator and a quadratic approximation to the energy:

$$\mathcal{J}_{\text{ref}} = \mathcal{J}[V_{\text{ref}}] = \text{diag}_{k=1,\ldots,L} \frac{1}{\rho_k} \begin{pmatrix} 0 & -\nabla \cdot \hat{h}^\text{ref} \\ -\nabla \cdot (f/h^\text{ref}) \hat{k} \times \end{pmatrix},$$

$$\mathcal{H}_{\text{ref}}(W) = \frac{1}{2} (W, \delta^2 \mathcal{H}[V_{\text{ref}}] W) = \frac{1}{2} \sum_{k=1}^L \int_{\Omega} \left( \rho_k h_k^\text{ref} |u_k^h|^2 + g \Delta \rho_k (T w^h)^2 \right),$$

where $W = (w^h, w^u)$. Thus, (2.10) reads concretely as

$$\left\{ \begin{array}{l}
\partial_t u_k^h = -\nabla \cdot (h_k^\text{ref} w_k^u) \\
\partial_t u_k^u = -(g/\rho_k) \nabla \left( T^T \text{diag}(\Delta \rho) T w^h \right)_k - f \hat{k} \times w_k^u
\end{array} \right. \quad \text{in } \Omega ,$$

(2.11)

Under the simplifying assumption that the Coriolis term and the bathymetry $b$ are flat, i.e. $f \equiv \text{const}, b \equiv \text{const}$, the eigenmodes of (2.11) can be easily computed for the stable reference configuration $h^\text{ref} = h^0$. 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 Figure 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 $\lambda = \pm \| \lambda \|$ of $\mathcal{A}_{\text{ref}} = \mathcal{J}_{\text{ref}} \mathcal{H}_{\text{ref}}$ 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 $\nabla \cdot$ and $\hat{k} \cdot \nabla \times$ of the momentum equation in (2.11) and algebraic manipulations) to solve

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

where $R = T^T \text{diag}(\Delta \rho) T$ 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

$$|\lambda|^2 - f^2 = \mu_{\text{vert}}^\mu_{\text{horiz}} > 0, \quad h^\lambda = h_{\mu,\text{vert}} h_{\mu,\text{horiz}}^\mu, \quad \text{with} \quad g \text{diag}(h^0/\rho) R h_{\mu,\text{vert}} = \mu_{\text{vert}} h_{\mu,\text{vert}}^\mu \text{ and } -\Delta h_{\mu,\text{horiz}} = \mu_{\text{horiz}} h_{\mu,\text{horiz}}^\mu,$$
where \( h^{\mu, \text{horiz}} \in L^2(\Omega) \) with \( \int_\Omega h^{\mu, \text{horiz}} = 0 \) and \( h^{\mu, \text{vert}} \in \mathbb{R}^L \). Due to the fact that variations of the density \( \Delta \rho \) are much smaller than density itself, the layer matrix

\[
\text{diag}(1/\rho) R = \text{diag}(1/\rho) T^\top \text{diag}(\Delta \rho) T = \begin{pmatrix}
1 & 1 & \cdots & 1 \\
\rho_1/\rho_2 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
\rho_1/\rho_L & \rho_2/\rho_L & \cdots & 1 
\end{pmatrix}
\]

(2.12)

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

\[
h^{\max, \text{vert}}_k \approx \frac{h^0_k}{b}, \quad u^{\max, \text{vert}}_k = (1/\rho_k)(R h^{\max, \text{vert}})_k \approx 1.
\]

This leads to the well-known (fast) barotropic free-surface mode with wave-speed \( \sqrt{\mu^{\max, \text{vert}}} \approx \sqrt{-bg} \), 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 \( \eta \)-mode \( \eta^{\mu, \text{horiz}} = T h^{\mu, \text{horiz}} \) as

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

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

\[
T^{-1}_{k,j} = D_{k,j} = \begin{cases} 
1 & \text{if } k = j, \\
-1 & \text{if } k = j - 1, \\
0 & \text{else}.
\end{cases}
\]

This formulation relates the vertical layer-modes to discrete approximations of solutions to the generalized eigenvalue problem from, e.g., \([5, 9]\).

3 Discretization by the TRiSK scheme

The rotating shallow water equations (2.1) and the multilayer version (2.6) 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 \([31, 25]\) (see also \([30]\)).

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 \( e \in \mathcal{E} \) (associated to the point of intersection of primal and dual grid edge \( x_e \)), primal cell grids will be denoted by
\( i \in I \) (associated to cell centers \( x_i \)), and the primal cell vertices will be denoted with \( v \in V \) (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 spaces \( X_I = \mathbb{R}^{N_I} \), \( X_V = \mathbb{R}^{N_V} \), and \( X_E = \mathbb{R}^{N_E} \), 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 \( y_e \approx n_e \cdot y(x_e) \), where \( n_e \) is the geodesic normal to the primal edge \( e \). Corresponding inner products on these spaces are given by

\[
(y, \varphi) = \sum_{i \in I} A_i y_i \varphi_i, \quad (y, \varphi)_V = \sum_{v \in V} A_v y_v \varphi_v, \quad \text{and} \quad (y, \varphi)_E = \sum_{e \in E} A_e y_e \varphi_e,
\]

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 \((l_e \text{ and } d_e)\). 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 or tangential).

The differential operators are built upon the fundamental relations that for any discrete variable \( y \in X_E \), and test functions \( \varphi \in X_I \) and \( \psi \in X_V \) we have

\[
(\varphi, \nabla \cdot y)_I = \sum_{i \in I} \sum_{e \in E_{\text{adj}}(i)} \varphi_{n_{e,i}} l_e y_e, \quad (\psi, (\hat{k} \cdot \nabla \times) y)_V = \sum_{v \in V} \sum_{e \in E_{\text{adj}}(v)} \psi_{t_{e,v}} d_e y_e,
\]

where \( E_{\text{adj}} \) and \( E_{\text{adj}} \) 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 \( y_e \). Concretely, \( n_{e,i} \) is positive if the edge \( e \) is oriented to point out of the cell \( i \) (thus decreasing the divergence by the integrated flow \( l_e y_e \)), and \( t_{e,v} \) is positive if the edge \( e \) is oriented on counterclockwise direction around the dual cell associated to vertex \( v \) (thus increasing the circulation around \( v \) by the integrated flow \( d_e y_e \)). We note that in [30], the schemes are built upon the integrated quantities \( Y_e = l_e y_e \) and \( Y_e = d_e y_e \), whereas we follow the convention used in [25].

Moreover, a discrete gradient is defined for a cell-wise quantity on the primal grid (across a primal edge): For each \( \varphi \in X_I \) and test function \( y \in X_E \) we set

\[
(\nabla \varphi, y)_E = - (\varphi, \nabla \cdot y)_I,
\]

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

\[
\{ \varphi \}_E \in X_E, \quad \{ \psi \}_E \in X_E,
\]

for \( \varphi \in X_I \) and function \( y \in X_E \), which average the values of the two adjacent primal and dual cells to the corresponding edge (with factor \( 1/2 \) each). Interpolation operators from edges to cells are defined by transposition as

\[
(\{y\}_E, \varphi)_I = (y, \{\varphi\}_E), \quad (\{y\}_V, \psi)_V = (y, \{\psi\}_E).
\]

We refer to [25] 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

\[
\hat{k} \times : X_E \to X_E, \quad \text{with} \quad (\hat{k} \times y)_e \sim t_e \cdot y(x_e)
\]

9
for any \( y \in X_E \) representing a continuous vector field \( y \), with \( n_e = \hat{k} \times n_e \) the tangent to the primal edge \( e \). Concretely, it is chosen as a skew-symmetric operator which fulfills the identity

\[
- (\hat{k} \cdot \nabla \times) \hat{k} \times y = \{ \nabla \cdot y \}_V \quad \text{for all} \quad y \in X_E
\]

where \( \{ \cdot \}_V \) is a corresponding interpolation operator from dual do primal cells. We refer to [31] for a derivation and the concrete expressions.

### 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 \( h_k \in X_1 \) and the velocities \( u_k \in X_E \), 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:

\[
K[u] = (1/2) \{ u \ast u \}_1 = \{ u^2 \}_1, \quad \text{(kinetic energy in } X_1) \\
q[h, u] = \left( \hat{k} \cdot \nabla \times u + f \right)/\{ h \}_V, \quad \text{(potential vorticity in } X_V).
\]

Here, \( \ast \) denotes the point- or entry-wise product (Hadamard product), and \( / \) the point-wise division and \( f \in X_V \) is an interpolant of the Coriolis parameter.

The discrete equations are now given as:

\[
\begin{cases}
\hat{h}_k = - \nabla \cdot \{ h_k \}_E \ast u_k & \text{in } X_1, \\
\hat{u}_k = Q[h_k, u_k] \{ h_k \}_E \ast u_k - \nabla \cdot (K[u_k] + (g/\rho_k) \rho_k h_k) + G_k[h, u] & \text{in } X_E. 
\end{cases}
\]

where \( b \in X_1 \) is a cell-wise interpolant of the bathymetry. The operator \( Q[\cdot, \cdot] : X_E \to X_E \) is defined as the skew-symmetric version

\[
Q[h_k, u_k]y = \frac{1}{2} \left( \{ q[h_k, u_k] \}_E \ast \left( \hat{k} \times \left( \{ q[h_k, u_k] \}_E \ast y \right) \right) \right),
\]

where \( y \in X_E \) is a discrete flux. Note that another form of this operator, namely the simpler \( Q[h_k, u_k]y = \{ q[h_k, u_k] \}_E \ast \left( \hat{k} \times \left( \{ q[h_k, u_k] \}_E \ast y \right) \right) \), leads to discrete conservation of potential vorticity, whereas the skew-symmetric form employed above yields energy conservation.

Energy conservation follows directly by introducing a discrete Hamiltonian framework for (3.1). We define the combined solution variable as \( V = (h, u) \in X^L = X^L_1 \times X^L_E \) analogous to the continuous case. It is endowed with the discrete inner product

\[
(V, W)_X = \sum_{k=1}^L (h_k, w^h_k)_1 + (u_k, w^u_k)_E,
\]
where \( W = (w_k^h, w_k^u)_{k=1,\ldots,L} \). The discrete Hamiltonian has the form
\[
H[V] = \frac{1}{2} \sum_{k=1}^{L} (\rho_k(h_k; K[u_k]) + g \Delta \rho_k(\eta_k[h], \eta_k[h])).
\] (3.2)

The functional derivative of the Hamiltonian fulfills the identity
\[
(W, \delta H[V])_X = H'[V; W] = \sum_{k=1}^{L} \left( (w_k^h, \rho_k K[u_k] + g p_k[h])_1 + \rho_k(w_k^u, h_k)_{E* u_k} \right),
\]
using the identity \( (h_k, K'[u_k; w_k^u])_1 = (w_k^u, h_k)_{E* u_k} \) and the concrete form of \( \eta \). Thus, we can write
\[
\delta H[V] = \left( \begin{array}{c}
\rho_k[h_k] + g p_k[h] \\
\rho_k(h_k)_{E* u_k}
\end{array} \right)_{k=1,2,\ldots,L}.
\] (3.3)

From the concrete form of the equations as given above, one can infer the discrete analogue of the operator \( J \), which is given by
\[
J(V) = \text{diag}_{k=1,2,\ldots,L} \frac{1}{\rho_k} \left( \begin{array}{cc}
0 & -\nabla^*_{E-1} \\
-\nabla_{1-\rightarrow E} & Q[h_k, u_k]
\end{array} \right),
\] (3.4)

Using the discrete identities for \( \nabla_{1-\rightarrow E} \) and \( \nabla^*_{E-1} \), the concrete form of \( Q \), and the skew-symmetry of \( \hat{k} \times_{E-\rightarrow E} \), the skew symmetry of \( J \) can be verified by considering a discrete weak formulation. Together, this shows that (3.1) can be described by
\[
\partial_t V = J[V] \delta H(V) + \begin{pmatrix} 0 \\ G[V] \end{pmatrix},
\] (3.5)

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

4 Exponential time integration

Exponential integrators or exponential time differencing methods (ETD) are a special class of time integration methods; see [19] 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, \( G \equiv 0 \). 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 \( V_n \approx V(t_n) \) the current solution at time \( t_n \), \( n = 0, 1, 2, \ldots \), and write
\[
\partial_t V = F[V] = A_n V + r_n[V] = F[V_n] + A_n(V - V_n) + R_n[V],
\] (4.1)
with the remainder and the residual are defined by

\[ r_n[V] = F[V] - A_nV, \]
\[ R_n[V] = F[V] - F[V_n] - A(V - V_n) = r_n[V] - r_n[V_n]. \]

Thus, the forcing term \( F \) is split into an affine linear part, where \( A \) should capture the stiff components of \( F \), and the nonlinear residual \( R_n \). In Rosenbrock-ETD methods, \( A_n \) is chosen to be the Jacobian of \( F \), which corresponds to a Taylor approximation of the forcing term. In the concrete setting with \( G \equiv 0 \) we have

\[ F[V] = J[V] \delta H[V], \]
\[ F'[V_n; W] = J'[V_n; W] \delta H[V] + J[V] \delta^2 H[V_n]W; \quad (4.2) \]

cf. section 2.3. The concrete expressions for the Jacobians are given in appendix B. Thus, a Rosenbrock-ETD method corresponds to a choice of \( A_n = F'[V_n] \), which yields more accurate methods with fewer internal stages. Unfortunately, this is associated to somewhat costly Jacobian evaluations, which are further complicated by the product structure of the forcing term as given above. Instead, we will mainly consider choices of linear operator with zero velocities and heights, which are frozen at a reference configuration \( V_{\text{ref}} = (h_{\text{ref}}, 0) \), since the first term in (4.2) is zero. This leads to a choice of \( A_n = A_{n}^{\text{ref}} = F'[V_{n}^{\text{ref}}] \), which leads to a structurally simpler and more efficiently computable linear operator, at the cost of an increased linearization error. We note that (4.3) has again a Hamiltonian structure. The reference point can be chosen differently in each time-step, in order to take updated height variables into account. Additional approximations of \( A_{n}^{\text{ref}} \), 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.

**4.1.1 Approximation of the residual**

To obtain a concrete 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

\[ V(t_{n+1}) = V(t_n) + \int_0^{\Delta t} \exp((\Delta t - \tau)A_n) (F[V(t_n)] + R_n[V(t_n + \tau)]) \, d\tau. \quad (4.4) \]

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. Concretely, for the class of exponential integrators of interest in this work, we replace the residual by an approximate Taylor expansion,

\[ R_n[V(t_n + \tau)] \approx \sum_{s=2}^{S} \frac{\tau^{s-1}}{(s-1)!} R_{n,s}, \]

with \( R_{n,s} \) determined by the internal stages of the underlying exponential Runge-Kutta method. Consequently, by inserting the above approximation into the solution formula (4.4) we obtain one time-step of the underlying method as:

\[ V(t_{n+1}) \approx V_{n+1} = V_n + \Delta t \left( \varphi_1(\Delta t A_n) F_n + \sum_{s=2}^{S} \varphi_s(\Delta t A_n) R_{n,s} \right), \quad (4.5) \]
where the \( \varphi \)-functions are defined as
\[
\varphi_s(z) = \int_0^1 \exp \left((1 - \sigma)z\right) \frac{\sigma^{s-1}}{(s-1)!} \, d\sigma = \sum_{k=0}^{\infty} \frac{z^k}{(k+s)!} \quad \text{for } z \in \mathbb{C}, \ s = 1, 2, \ldots . \tag{4.6}
\]
In the case \( s = 0 \), we set \( \varphi_0(\cdot) = \exp(\cdot) \). Note that the above definition of \( \varphi_s \) 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 \( \varphi \)-functions and in-homogeneous linear equations; cf., e.g., [22]:

**Proposition 4.1.** The value \( w(\Delta t) = \sum_{s=0}^{S} \varphi_s(\Delta t A) b_s \) for arbitrary \( b_s, s = 0, 1, \ldots, S \), is the terminal value of the solution to the linear problem

\[
\begin{align*}
\partial_\tau w(\tau) &= A w(\tau) + \sum_{s=1}^{S} \frac{\tau^{s-1}}{\Delta t^{s-1}(s-1)!} b_s, \quad (0 < \tau < \Delta t) \\
\tau = 0 &\quad w(0) = b_0.
\end{align*}
\tag{4.7}
\]

In the following, we will briefly present the concrete exponential integrators employed in this work, which are taken from [18, 19].

### 4.1.2 Exponential Euler

In the simplest case, the residual is simply neglected, and the exponential Euler method is obtained as
\[
V_{n+1} = V_n + \Delta t \varphi_1(\Delta t A_n) F[V_n]. \tag{4.8}
\]

In the case that \( A_n \) is only an approximation to the Jacobian, this method is only first order accurate in \( \Delta t \), and not attractive for practical computations. However, in the case that \( A_n = F'[V_n] \) (Rosenbrock exponential Euler), this method is second order accurate (see, e.g., [19]). This is due to the fact that also the linear term vanishes in the Taylor expansion of the residual. Thus, the construction of optimal order methods with minimal number of stages is generally different for Rosenbrock methods and methods with approximate Jacobian. We focus on the case of approximate Jacobians in the following, since approximate Jacobians are more efficient to realize in our setting.

### 4.1.3 Two stage methods

For the case of approximate Jacobians, e.g., \( A_n = A_n^{\text{ref}} \), a family of two-stage, second order in time methods is given by the one-parameter family

\[
\begin{align*}
v_{n,2} &= V_n + c_2 \Delta t \varphi_1(c_2 \Delta t A_n) F[V_n], \\
v_n &= V_n + \Delta t \varphi_1(\Delta t A_n) F[V_n] + (\Delta t/c_2) \varphi_2(\Delta t A_n) R_n[v_{n,2}],
\end{align*} \tag{4.9}
\]

for the parameter \( c_2 \in (0, 1) \); see [18]. 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 are given in appendix C. A general discussion of higher order methods can be found in, e.g., [18, 19].
4.2 Approximation of the matrix functions

As previously mentioned, the most challenging aspect of ETD methods is efficiently evaluating the matrix functions $\phi_s$. This challenge made ETD methods computational infeasible for many years after their discovery due to a lack of efficient matrix function evaluation methods \[21\]. However, in recent years more efficient ways to approximate $\phi_s$ have been found such as Krylov subspace projections \[26\], which can be combined with sub-stepping algorithms \[22\], or instead found with Leja-point interpolation \[1\], or Chebychev polynomial approximations \[29\]. 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 \[14\]. 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 $A \in \mathbb{R}^{N \times N}$ and a vector $b \in \mathbb{R}^N$ is defined as

$$
K_M(A, b) = \text{span}\{b, Ab, \ldots, A^{M-1}b\} = \{p(A)b \mid p \in \mathcal{P}_{M-1}\}
$$

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 $x = (\text{Id} + \Delta tA)^{-1}b$; for ETD the expressions $\phi_s(\Delta tA)b$ are approximated. For this purpose, an orthonormal basis of $K_M(A, b)$ is constructed, which is typically done by the Arnoldi process.

In this work, we will mostly employ linear operators of the form \eqref{eq:linear_operator}, which have the product structure

$$
A = J \delta^2 H,
$$

where additionally the matrix $J$ and $\delta^2 H$ have symmetry properties with respect to the inner product induced by the symmetric $M_X$, the mass matrix of the solution space $X^L = X^L_1 \times X^L_2$. Concretely we have

$$
M_X J = -J^\top M_X, \quad M_X \delta^2 H = \delta^2 H^\top M_X.
$$

This yields the skew-symmetry of $A$ with respect to the inner product induced by the symmetric matrix induced by the second variation of the Hamiltonian.

**Proposition 4.2.** The operator $A$ given in \eqref{eq:linear_operator} with \eqref{eq:hamiltonian_matrix} fulfills

$$
M_H A = -A^\top M_H \quad \text{where} \quad M_H = M_X \delta^2 H.
$$

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

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

14
induced by another symmetric matrix $M$, which can be chosen as either $M_X$ (corresponding to
the physical $L^2$-norm) or $M_H$ (corresponding to a norm induced by a quadratic approximation of
the Hamiltonian).

The orthonormal basis vectors $v_m$, $m = 1, \ldots, M$ can be found through the iterative Arnoldi
process:

$$
\tilde{v}_{m+1} = Av_m - \sum_{j=1}^{m} h_{j,m} v_j, \quad h_{j,m} = \langle v_j, Av_m \rangle_M, \quad j = 1, 2, \ldots, m,
$$

$$
v_{m+1} = \frac{1}{h_{m+1,m}} \tilde{v}_{m+1}, \quad h_{m+1,m} = \| \tilde{v}_{m+1} \|_M,
$$

where $v_1 = b$. The Arnoldi process can be collectively given by the Arnoldi decomposition

$$
A V_M = V_M H_M + h_{M+1,M} v_{M+1} e_M^T = V_{M+1} \tilde{H}_M,
$$

where $V_M \in \mathbb{R}^{N \times M}$ contains the $M$ orthogonal basis vectors, i.e., $V_M^T V_M = I_M$, $e_M$ is the
$M$-th canonical basis vector with entries $e_M,i = \delta_{M,i}$, and $H_M \in \mathbb{R}^{M \times M}$ is the Hessenberg matrix
given by $H_M = V_M^T A V_M$. Finally, the Krylov approximation of a matrix function $\varphi_s(\Delta t A)$
(see, e.g., [19, Section 4.2]) is given by

$$
\varphi_s(\Delta t A) b \approx V_M \varphi_s(V_M^T \Delta t A V_M) V_M^T M b = \| b \|_M V_M \varphi_s(\Delta t H_M) e_1,
$$

where $e_1$ is the first canonical basis vector. Here, $\varphi_s(\Delta t H_M) e_1$ can be computed using a dense
Padé approximation or an exponential of an augmented matrix (see, e.g., [27]).

For an operator $A$ that is skew-symmetric with respect to $M$, there exists a more efficient
method known as the skew-Lanczos process (see, e.g., [11, 13]). 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:

$$
\tilde{v}_{m+1} = Av_m - h_{m-1,m} v_{m-1}, \quad h_{m-1,m} = -h_{m,m-1},
$$

$$
v_{m+1} = \frac{1}{h_{m+1,m}} \tilde{v}_{m+1}, \quad h_{m+1,m} = \| \tilde{v}_{m+1} \|_M,
$$

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

Thus, the skew-Lanczos process avoids most of the reorthogonalization steps, which reduces the
computational cost of the Arnoldi-method from $O(M^2 N)$ to $O(M N)$. 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 [12, 32], which performs orthogonalization only with respect to
the last $p$ Arnoldi vectors, while maintaining an exponential asymptotic convergence rate towards
the exact solution; see [32].

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 $\varphi$-
function holds; see the overview in [19, Section 4.2]. For instance, in the skew-symmetric case [17,
Theorem 4], after a minimum of $M \geq \Delta t |A|$ iterations, where $|A|$ is the spectral radius of $A$,
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 $A$ 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 (4.3). In fact, the dissipation terms contained in $G$ 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., [6]. 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 buildup 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 $A_n = A = \delta^2 H$ is skew-symmetric with respect to the $\delta^2 H$ inner product. We note that this implies that $A$ has purely imaginary spectrum, and the associated eigen-vectors are $\delta^2 H$ orthogonal. Concretely, 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

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

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

$$
\partial_t V_\gamma = F(V_\gamma) - (\Delta t^{2p-1}/\gamma^2)^p (-A^2)^p V_\gamma, \quad V_\gamma(0) = V_0,
$$

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 $A$, it follows that $-A^2 = J^\top \delta^2 H J \delta^2 H$ is a $\delta^2 H$-symmetric positive operator, and thus the appearance of its $p$-th power in (4.14) dissipates the quadratic energy induced by $\delta^2 H$.

**Remark 1.** Additionally, in the concrete case of $A$ derived as (4.3) from (3.5) it can be further verified that $-A^2$ 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 bi-harmonic diffusion with coefficient proportional to $\Delta t^3/\gamma^4$.

Concerning the numerical implementation, a direct application of the Krylov method to (4.13) would lead to a significant increase in computation times if the Krylov space is constructed for $\Delta t A_n = \Delta t A - (\Delta t/\gamma^2)^p (-A^2)^p$, since this requires additional multiplications by $A$. Instead, we build the Krylov space as before for $A$, and apply the modified $\varphi$-function, i.e.,

$$
\varphi_{s,\gamma}(c, \Delta t A)b \approx \|b\|_M \varphi_{s,\gamma}(c, \Delta t M_e) e_1
$$
where $H_M$ 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 $H_M$, 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 TRIK-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 (2.6) is layer-wise volume conserving, i.e.,

$$\langle 1, h_k(t) \rangle_I = \text{const} \quad \text{for all } k = 1, 2, \ldots, L,$$

where $1 \in X_I$ denotes the constant one cell-vector.

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

$$\frac{d}{dt} \langle 1, h_k(t) \rangle_I = \langle 1, \partial_t h_k(t) \rangle_I = \langle 1, \nabla \cdot (\{h_k\}_E * u_k) \rangle_I = -\langle \nabla I_{\rightarrow E}, \{h_k\}_E * u_k \rangle_E = 0,$$

using the discrete adjoint relation between $\nabla \cdot E_{\rightarrow I}$ and $\nabla I_{\rightarrow E}$.

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

$$m[h] = \sum_{k=1}^L \rho_k \langle 1, h_k(t) \rangle_I,$$

(4.15)

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

$$\frac{d}{dt} \langle l, V(t) \rangle_X = \langle l, \partial_t V(t) \rangle_X = \langle l, F[V(t)] \rangle_X = 0,$$

(4.16)

where $\langle l, \cdot \rangle_X$ is a linear functional, represented by testing with the vector $l$. For instance, in the case of (4.15), we choose $l = (l^h, l^u) = (\rho, 0)$, where $\rho_k = \rho_k 1$.

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

**Theorem 4.5.** Assume that $\langle l, F(V) \rangle_X = 0$ for all $V \in X$ (which implies the linear invariant (4.16)). Assume further that

$$\langle l, A_n V \rangle_X = 0 \quad \text{for all } V \in X.$$

Then, the ETD methods presented in this section preserve the same linear invariant.
Proof. We use the explicit formula for the final stage (4.5) to obtain

\[(l, V_{n+1} - V_n)_X = \sum_{s=1}^{S} (l, \varphi_s(A_n)b_s)_X,
\]

where \(b_1 = F(V_n))\) and \(b_s\) for \(s > 1\) is a linear combination of the residuals \(R_n(v^i_n)\) evaluated at the internal stages \(v^i_n\). Now, we directly obtain \((l, b_s)_X = 0\) using the properties of \(F\) and \(A_n\). Then, testing the linear equation (4.7) which corresponds to the expressions involving the \(\varphi\)-functions by \(l\) reveals that also \((l, \varphi_s(A_n)b_s)_X = 0\) for all \(s\), which yields

\[(l, V_{n+1})_X = (l, V_n)_X,
\]

which is the desired property.

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 (4.3) and let \(J = J[V_{\text{ref}}]\) and \(\delta^2 H = \delta^2 H[V_{\text{ref}}]\) for some reference configuration \(V_{\text{ref}} = (h_{\text{ref}}, 0) \in X\).

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 with higher frequency (in contrast to the horizontal modes, which are increasing).

We introduce a reduced-layer space

\[X_{\tilde{L}} = X_{\tilde{L}}^h \times X_{\tilde{L}}^E\]

for \(1 \leq \tilde{L} \ll L\), which is supposed to parametrize the fastest vertical modes of the discrete linear operator \(A = J \delta^2 H\) as in (4.3). We define a corresponding ansatz by the matrix \(\Psi: X_{\tilde{L}} \rightarrow X_L\) with the structure

\[\Psi = \begin{pmatrix} \Psi_h & 0 \\ 0 & \Psi_u \end{pmatrix},\]

Here, the matrix \(\Psi_h\) (and similarly, \(\Psi_u\)) is defined by

\[(\Psi_h V)_i = \sum_{j=1}^{\tilde{L}} \Psi_{h,i}^j V_{k,i}, \quad \text{for all } i \in I,\]
where the vector $\Psi_h^{ji} \in \mathbb{R}^L$ should roughly correspond to the $j$-th fastest height mode, which can be different in each cell $i \in I$.

Remark 2. For the simplified case of a constant Coriolis term and bathymetry, i.e. $f \equiv \text{const}$, $b \equiv \text{const}$, let $\mu_j > \mu_{j+1} > 0$, $j = 1, 2, \ldots, L$ be the eigenvalues of the matrix $A^{\text{vert}} = g \text{diag}(h^0/\rho) R$ with $R$ defined as in (2.12), at the constant reference heights $h^0$ corresponding to the eigenvectors $w_j^{\text{vert}}, \forall j \in \mathbb{R}^L$. These modes correspond to the height variable, whereas the corresponding modes for the velocity are given as $w_j^{u,\text{vert}} = \text{diag}(1/\rho) R w_j^{h,\text{vert}}$. In this case, we can choose $\Psi_h^{ji} = w_j^{h,\text{vert}}$ and $\Psi_u^{j,e} = w_j^{u,\text{vert}}$ for all cells and edges. In this particular case, the following arguments would greatly simplify. However, since the bottom topography is generally not constant, the horizontal modes can only be defined in an approximate sense, separately in each cell or edge.

The corresponding reduced (linearized) Hamiltonian matrix is defined as

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

We note that the above projection can be computed separately for each cell- and edge-stack, since

$$\Psi^\top \text{ where } \Psi^\top \text{ would greatly simplify. However, since the bottom topography is generally not constant, the horizontal modes can only be defined in an approximate sense, separately in each cell or edge.}

In order to provide an energetically consistent projection of the linearized equation to the reduced space, we introduce the orthogonal projection

$$P = \Psi \Psi^\dagger,$$

where $\Psi^\dagger$ is given by the (generalized) Moore-Penrose pseudo-inverse

$$\Psi^\dagger: X^L \rightarrow X^\hat{L}, \quad \Psi^\dagger = (\delta^2 \tilde{H})^{-1} \Psi^\top \delta^2 H.$$

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

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

$$\hat{V} = \arg \min_{\hat{V} \in X^{\hat{L}}} \| V - \Psi \hat{V} \|_{\delta^2 H} = \arg \min_{\hat{V} \in X^{\hat{L}}} \frac{1}{2} (V - \Psi \hat{V}, \delta^2 H (V - \Psi \hat{V}))_X. \quad (5.1)$$

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

$$\Psi^\top M_X \delta^2 H \Psi \hat{V} = \Psi^\top M_X \delta^2 H V.$$

Now, we use additionally the fact that the mass matrix $M_X$ of the space $X^L$ is diagonal and commutes with $\Psi^\top$ in the sense that $\Psi^\top M_X = M_X \Psi^\top$. The last property follows from the fact that $\Psi^\top$ operates only on the layer indices for each cell and edge stack, and that both mass matrices consist of multiple copies of the single layer mass-matrix.

Thus, the projection $PV = \Psi \hat{V}$ is simply the one that minimizes the projection error in the canonical linearized energy norm. Based on this choice of the projection, a reduced linear operator $A_P$ can be defined for $A = A^{\text{ref}} = J \delta^2 H$ as

$$A_P = PA_P = \Psi \hat{A} \Psi^\dagger,$$

where $\hat{A} = \Psi^\dagger A \Psi.$  \hspace{1cm} (5.2)
Using the concrete derivation of $\Psi^\dagger$, the reduced operator $\hat{A}$ again has Hamiltonian structure.

**Proposition 5.2.** Define the skew-adjoint operator

$$\hat{J} = \Psi^\dagger J (\Psi^\dagger)^\top : X^{\hat{L}} \to X^{\hat{L}}.$$

(5.3)

Then, it holds $\hat{A} = \hat{J} \delta^2 \hat{H}$.

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

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

Thereby, the structural properties important for stability of the solutions are preserved. In particular, it also holds

$$A_P = J_P \delta^2 H_P = P J P \delta^2 H_P,$$

which shows Hamiltonian structure also for $A_P$. Thus, we can easily employ the derived linear operator in the context of an ETD method, by simply replacing the operator $A$ by $A_P$. By the construction the adaptation is straightforward. Moreover, the computation of the matrix $\varphi$-functions can be reduced to a smaller-size problem, based on the following observation.

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

$$\varphi_s(A_P) = \frac{1}{s!} (\text{Id} - P) + \Psi \varphi_s(\hat{A}) \Psi^\dagger.$$

**Proof.** Due to (4.6), we have

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

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

Thereby, the computation of $\varphi_s(\Delta t A_P) b_s$ in an ETD method can be reduced to the computation of $\varphi_s(\Delta t \hat{A}) \Psi^\dagger b_s$ and simple applications of $\Psi$ and $\Psi^\dagger$. In the context of a ETD-Krylov method, this significantly reduces the required computational work.

Finally, for illustrative purposes, we explicitly write the exponential Euler method using the operator $A_P$, which is given by

$$V_{n+1} = V_n + \Delta t (\text{Id} - P) F[V_n] + \Delta t \Psi \varphi_1(\Delta t \hat{A}) \Psi^\dagger F[V_n].$$

Thus, the method performs an explicit Euler step with the orthogonal complement $(\text{Id} - P) F(V_n)$ of the forcing term, whereas the part of the forcing term in the space spanned by $\Psi$ is treated with the matrix exponential associated to $\hat{A}$. Similarly, the higher order ETD methods can be rewritten to a form which is more suitable for implementation purposes.

### 5.2 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 $\Psi$, which exploits the simple 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.
Here, a suitable method arises from a direct application of an exponential integrator with a particular choice of \( G \), which we refer to as barotropic ETD (B-ETD) method.

For a reference configuration \( h_{\text{ref}} \in X_L \) define the corresponding total height and average density as

\[
\tilde{h} = \sum_{k=1}^{L} h_{k}^{\text{ref}}, \quad \tilde{\rho} = \sum_{k=1}^{L} \frac{\rho_k h_{k}^{\text{ref}}}{h} \in X_I.
\]

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

\[
\Psi = \begin{pmatrix}
y & 0 \\
0 & 1
\end{pmatrix}, \quad \text{where } y \equiv 1, \ y_k = h_k^{\text{ref}} / \tilde{h}.
\] (5.4)

Additionally, due to the fact that the average density is given as

\[
\rho = \rho^\top y, \quad \text{the concrete form of the reduced Hamiltonian is readily derived as}
\]

\[
\delta^2 \tilde{H} = \begin{pmatrix}
g \text{diag}(\tilde{\rho}) & 0 \\
0 & \text{diag} \left( \langle \tilde{\rho} h \rangle_E \right)
\end{pmatrix}, \quad \text{where } \tilde{\rho} = y^\top R y \in X_I.
\]

and \( R \) is the matrix \( R = T^\top \text{diag}((\Delta \rho)T = (\rho_{\text{min}}{k,l})_{k,l} \). A simple computation now yields the concrete form of \( \Psi^\dagger \) as

\[
\Psi^\dagger = \begin{pmatrix}
(1/\tilde{\rho})^\top y^\top R & 0 \\
0 & \langle \rho \ast h^{\text{ref}} \rangle_E / \langle \tilde{h} \ast \tilde{\rho} \rangle_E
\end{pmatrix}.
\]

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

Remark 3. In the case where \( \rho_k = \rho^{\text{ref}} \) for \( k = 1, 2, \ldots, L \), we obtain that

\[
\Psi^\dagger = \begin{pmatrix}
y^\top & 0 \\
0 & \langle h^{\text{ref}} \rangle_E / \langle \tilde{h} \rangle_E
\end{pmatrix}.
\]

Thus, the roles of the test-functions in \( \Psi^\dagger \) and ansatz-functions in \( \Psi \) 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; see [15].

### 5.3 Total mass conservation

One drawback of the outlined approach is that the form of the test functions in \( \Psi^\dagger \) can not be controlled directly, rather they arise from the choice of \( \Psi \) 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

\[
\delta^2 \tilde{H} = \begin{pmatrix}
g \text{diag}(\rho \tilde{h}^\top / \tilde{\rho}) & 0 \\
0 & \text{diag} \left( \rho \ast \langle h \rangle_E \right)
\end{pmatrix},
\]

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

\[
\delta^2 \tilde{H} = \begin{pmatrix}
g \text{diag}(\tilde{\rho}) & 0 \\
0 & \text{diag} \left( \langle \tilde{\rho} h \rangle_E \right)
\end{pmatrix} \quad \text{and } \Psi^\dagger = \begin{pmatrix}
\rho^\top / \tilde{\rho} & 0 \\
0 & \langle \rho \ast h^{\text{ref}} \rangle_E / \langle \tilde{h} \ast \tilde{\rho} \rangle_E
\end{pmatrix}.
\]
The resulting linear operator from this choice leads to global mass conservation.

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

\[
M[h_{n+1}] = M[h_n], \quad \text{where} \quad M[h] = \sum_{k=1}^{L} \rho_k (1, h_k),
\]

and \((h_n, u_n) = V_n\) are the time-steps (4.5).

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

\[
(l, \tilde{A}_P V)_X = (\rho, \Psi_h \Psi_h^\dagger f_h[V])_{X_L^I} = 0 \quad \text{for all} \quad V \in X,
\]

where \( f_h[V] \in X_L^I \) is the first component (corresponding to height variables) of \( f[V] = J \delta^2 \tilde{H} PV \). We note that \( \Psi_h \Psi_h^\dagger = y \rho^\top / \tilde{\rho} \), from the concrete form of \( \Psi^\dagger \). Thus, it follows that \( \rho^\top \Psi_h \Psi_h^\dagger = \rho^\top y \rho^\top / \tilde{\rho} = \rho^\top, \) since \( \tilde{\rho} = \rho^\top y \). Therefore, it indeed follows \( (\rho, \Psi_h \Psi_h^\dagger f_h[V])_{X_L^I} = (\rho, f_h[V])_{X_L^I} = 0 \), due to the properties of \( J \).

**Remark 4.** Similarly, replacing the matrix \( R \) by the constant rank-one matrix \( (R_i)_{k,l} = \hat{\rho}_i \) in every cell \( i \in I \) results in a ETD method which exactly conserves the total layer volume, defined as \( \sum_k (1, h_k) \).

We note that this method incurs an additional approximation error. However, since the approximation by \( R \) 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 [33]. The computational domain is given by a circular basin on the surface of the sphere of radius 6371.22 km, centered at \((35^\circ N, 0^\circ W)\) 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 [33, Appendix A]; see also Figure 1. We consider a single-layer and a three-layer configuration.

#### 6.1 Algorithmic details

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

##### 6.1.1 Spatial mesh

A quasi-uniform mesh is constructed from a centroidal Voronoi tessellation [23], where the distance of the cell centers \( d_c \) is ca. 16 km resolution. In order to obtain an initial condition for the initial layer configuration, we interpolate the initial heights as in Figure 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

$$\Delta t_C = 1/|A_0|, \quad C(\Delta t) = \Delta t/|A_0| = \Delta t/\Delta t_C$$

where $|\cdot| = \sigma_{\text{max}}(\cdot)$ denotes the largest magnitude eigenvalue, and $A_0 = F'(V^0)$ is the linearized operator at the stable reference configuration. We note that $\Delta t_C$ is determined (up to a constant factor) by the largest quotient of the the local mesh-width and the local free-surface wave-speed $\sqrt{g b(x)}$. Concretely, we obtain $\Delta t_C \approx 37.9$ [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. More specifically, stability of RK4 (for the linearized equation $\partial_t V = A_0 V$) is given for Courant numbers $C(\Delta t) \leq \sqrt{8}$. Thus, the maximal RK4 stepsize is given as $\Delta t_{\text{RK4}} = \sqrt{8} \Delta t_C \approx 107.2$ [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 $A_n = A_{\text{ref}}^n$, where the reference state is chosen to be either the resting state or the current height with zero velocities $(h(t_n), 0)$. Since $A_{\text{ref}}^n$ corresponds to a first order wave-operator, this class of methods will be called ETD$\text{Swave}$, where $S$ refers to the number of internal stages. Rosenbrock-ETD methods are not considered due to the cost of assembling the Jacobian. In practice we found that, despite the higher accuracy for fewer stages, the assembly cost associated with the Jacobian made Rosenbrock-ETD methods noncompetitive to RK4 in our setting, even for very large time-step configurations. The second class of methods are based on section 5. Within this class, we will focus on the methods where $A_n$ is the linear operator that is projected onto the barotropic mode. For this reason we will refer to these methods as B-ETD$\text{Swave}$.

### 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 $\varphi_s$ 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.

Additionally, we comment on the number of Krylov iterations per evaluation of a matrix $\varphi$-function. The theoretical estimates (see, e.g., [19, 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 employ the adaptive a posteriori error criterion suggested in [2], based on [4, 8]. However, in the numerical experiments, we found that the
convergence behavior suggested by the theory was sharp: e.g., a error tolerance of $10^{-6}$ was usually met after $M \geq a_1 C(\Delta t) + a_2$ iterations, where appropriate constants $a_1 \approx 1.25$, $a_2 \approx 15$ were determined empirically. Moreover, using less than $C(\Delta t)$ Krylov iterations usually lead to completely inaccurate solutions and even unstable simulations. This can be contrasted with an approximation of the matrix $\varphi$-function based on RK4 time stepping using (4.7), which requires at a minimum a number of $4/\sqrt{8} = \sqrt{2} \approx 1.41$ matrix multiplications per unit timestep (with $C(\Delta t) = 1$), just to obtain basic stability, and then converges at fourth order.

6.2 Discussion of results

In the first two test-case, obtained using the single-layer configuration, the order of convergence and energy conservation of the ETDSwave methods are investigated. To ensure energy conservation in this scenario, all forcing and smoothing terms are omitted. The third and final test-case uses 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 bi-harmonic smoothing terms.

6.2.1 Single-layer scenario

The first test scenario is used to verify the order of convergence and to investigate the energy conservation properties of the ETDSwave method. For simplicity, this scenario is implemented using the single-layer configuration. No additional smoothing and forcing terms are considered, i.e. $G \equiv 0$, which ensures energy conservation of the time-continuous equations. 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 $\eta_0 = \bar{\eta} \exp \left(- \frac{(x - x_{\text{center}})^2}{2\sigma^2}\right)$, where the radius is $\sigma = 200$ km, the total perturbation height is $\bar{\eta} = 2$ m and $x_{\text{center}}$ is the location at the center of the domain.

In the first case, this is complemented with an initially zero 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 $u_0 = \left(\frac{g}{f(x_{\text{center}})}\right) \hat{k} \times \nabla \eta_0$. 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 Figure 2.

Convergence test The errors from various ETDSwave methods are computed using a reference solution computed with RK4 using a time-step size of $\Delta t = (1/4) \Delta t_C$, and the time-step sizes for the ETD methods are chosen with $\Delta t = 2^j \Delta t_C$, $j = 0, 1, \ldots, 7$. 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 method ETD3wave (detailed in appendix C), where the coefficients are chosen to be $(c_2, c_3) = (1/2, 3/4)$. We choose $A_n = A^{\text{ref}}$, linearized either at the stable reference configuration or at the current height $h_n$ in each time step.

To compare the methods, in Figure 3 we show the relative solution error in the discrete linearized energy norm induced by the matrix $M_H = M_X \delta^2 H^{\text{ref}}$ (cf. section 4.2.1) as a function the Courant number. First, we note that the methods exhibit the expected convergence order. Only the ETD2wave method with $c = 2/3$ is noteworthy, since it appears to have third order
Figure 2: Snapshots of the initial height $h_0$, the initial velocity $u_0$ and the velocity after ten days of simulation time.

Figure 3: The relative error in the solution at the final time in the (linearized) energy norm for various methods and Courant numbers. Left: for initial condition $(h_0,0)$ after six hours. Right: for the initial condition $(h_0,u_0)$ after ten days.

convergence for a large regime of time-step sizes. In the first “fast” 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 $c = 2/3$ 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 second “slow” test-case, we observe an appreciable improvement due to incorporating the current reference height $h_n$ into $A$. We note that this improvement is also present for large Courant numbers.

**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 $\gamma$. Again, we use the initial condition from Figure 2, which is in geostrophic balance.
First, the evolution of the energy from a simulation using RK4 close to the maximal time-step
\[ \Delta t_{\text{RK4}} = \sqrt{8} \Delta t_C, \]
is compared to the energy obtained from using ETD with various \( \gamma \) values. The time-step size for ETD2wave is chosen as \( \Delta t = 10 \Delta t_{\text{RK4}} \) (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.

![Figure 4: Energy for RK4, with Courant number \( C(\Delta t) = (3/4)\sqrt{8} \approx 2.1 \), and ETD, with \( C(\Delta t) = 10\sqrt{8} \approx 28.3 \) for varying \( \gamma \) values over a time horizon of fifteen days. Left, without bi-harmonic smoothing. Right, with bi-harmonic smoothing.](image)

In Figure 4 we plot the evolution of the energy for all methods. We observe that the ETD methods are affected by a larger time-discretization error than RK4, in particular close to the initial time. This is not surprising, due to the much larger time-step employed by these methods. Concerning the influence of \( \gamma \), we observe that for the largest value of \( \gamma = 30 \) the energy is barely affected, which is explained by the fact that \( \gamma \) is bigger than the Courant number \( C(\Delta t) \approx 28.3 \). For smaller \( \gamma \), 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 \( \gamma \), the effect of the artificial dissipation is noticeably larger than the time-discretization error.

Secondly, we repeat the same test, but add a biharmonic smoothing to the model (see Appendix A.1) with horizontal viscosity \( \nu_h = 2 \times 10^{10} \). 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 [33]. The results, given in Figure 4, show that the energy dissipating effect of the biharmonic viscosity is stronger than either the time-discretization error or the artificial numerical diffusion.

6.2.2 Multi-layer 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 [33, Appendix A]. The wind stress \( \tau_\lambda \) 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 $\nu_h = 2 \times 10^{10}$ and $\nu_v = 10^{-4}$, respectively (which are the values given in [33] for the 16 km grid). The three layer configuration for this scenario has initial layer interfaces located at $\eta_0^1 = 0$, $\eta_0^2 = -25/3$, and $\eta_0^3 = -50/3$ [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 densities on some part of the domain), which could lead to a breakdown of the simulation. The layer densities are set to $\rho = (1025, 1027, 1028)$ [kg 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 $\Delta t = (3/4)\Delta t_{RK4}$. The SSH and top layer velocity of the resulting spun-up initial condition are shown in Figure 5. 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 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 and B-ETD2wave (using the reference heights and $c_1 = 1$) with time-steps increased above the maximal RK4 time-step $\Delta t_{RK4} = \sqrt{8}\Delta t_c \approx 107.2$ [s] for the 16 km grid. For ETD2wave the time step is increased 10 and 15 times, and for B-ETD2wave 5 and 7 times over $\Delta t_{RK4}$. 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 depict the evolution of global mass and total
energy for all methods in Figure 6. We note that global mass is conserved for all methods, as predicted by the theory. However, it is evident by the global energy evolution that the solutions differ significantly after the first year of simulation time.

Now, 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 velocity and vorticity of the top layer mean flow, and the SSH RMS are shown in Figure 7, which are computed from the RK4 simulation. Comparing the error in these quantities for each method, using the small time-step RK4 solution as a base-line, we find that they differ very little. Concretely, we compute the relative reference-thickness weighted $L^2$ error of the mean flow and the $L^\infty$ error of the SSH RMS, which are given in Table 1. Here, we define the reference-thickness weighted norm by $\|u\|^2_{L^2_{\text{ht}}} = \sum_{k=1}^{L} \int_{\Omega} h_k^{\text{ref}} u^2$, which roughly corresponds to the physical $L^2$ norm of the underlying three-dimensional velocity field. We observe that all methods (including the RK4 simulation close to the CFL) reproduce the mean flow up to a similar tolerance of $\sim 5\%$. This suggests that the bulk of the error is caused by replacing the true mean value by a sample average of a effectively random trajectory on a finite interval and not by the time-discretization.
Concerning the error in the SSH RMS, which is a more sensitive criterion, no method achieves more than 20% 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. 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, even though it is forced to used a smaller time step. It should be noted that the speed up seen in the B-ETD2wave method is expected to grow as the number of layers in the models grows. This is because the size of reduced linear operator, and hence the amount of work to compute the \( \varphi_s \) functions is independent of the number of layers.

### Acknowledgments

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

### 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 [24, p. 6, table 2]. The operators consist of the divergence \( \nabla \cdot E \rightarrow I \), the gradient \( \nabla I \rightarrow E \), the gradient in the perpendicular direction \( \nabla_{\perp} V \rightarrow E \), the scalar curl \( (\hat{k} \cdot \nabla \times) E \rightarrow V \), the (perpendicular) flux reconstruction operator \( \hat{k} \times E \rightarrow E \), and the interpolation operators.

The definition of flux reconstruction operator \( \hat{k} \times E \rightarrow E \) given in [30], 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 \( N \)), tangents \( t_{e,v} \) (elements of the matrix \( T \)), kite areas (the intersections between triangles) \( R_{e,v}A_i \) (elements of the matrix \( R \)), the weights on edge array \( w_{e,e'} \) (elements of the matrix \( W \)) are then defined as quantities satisfying the relation

\[
-TW = RN. \tag{A.1}
\]

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

Table 1: Comparison of the ETD methods against the base-line RK4 method: the number of Krylov vectors \( M \), the average simulated years per real day (SYPD), the relative mean velocity error in the \( L^2_{h_{ref}} \) norm, and the SSH RMS error in the \( L^\infty \) norm.
We obtain a discretization of the model on the bounded domain by fixing all velocity variables where 
\[ \tau \]
and to prevent a build-up of vorticity in finest grid scales, numerical dissipation terms have to be order to account for the energy dissipated in scales below the grid resolution (due to turbulence), discretization, physical viscosities in the momentum equation are usually negligible. However, 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

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

\[ f_{\text{wind}}[h] = \chi_{k=1} \tau_{\text{wind}} / \{ \rho I h_1 \} _E, \]

where \( \tau_{\text{wind}} \sim n e \cdot \tau_{\text{wind}}(x_e) \) is a 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

\[ f_{\text{drag}}[h, u] = -\chi_{l=\text{bot}} c_{\text{drag}} \rho_{\text{bot}} [h_{\text{bot}}] _E \star [u_{\text{bot}}] _E, \]

where \( l_{\text{bot}} \) is the bottom layer index. This corresponds to a quadratic drag term \(- c_{\text{drag}} [u_{\text{bot}}] _E \star [u_{\text{bot}}] _E / h_{\text{bot}} \) 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

| Divergence: | \((\nabla \cdot \mathbf{E}) \mathbf{y} \) \(e\) = \( (1/\mathbf{A}_i) \sum_{e \in \mathbf{E}_i} n_{e,i} l_e \mathbf{y}_{e} \) |
| Gradient: | \((\nabla \mathbf{E}) \mathbf{y} \) \(i\) = \( (1/d_e) \sum_{e \in \mathbf{E}_i} -n_{e,i} l_i \) |
| Curl: | \((\mathbf{k} \cdot \nabla \times \mathbf{y}) \) \(v\) = \( (1/A_e) \sum_{e \in \mathbf{E}_v} \ell_{e,v} d_e \mathbf{y}_{e} \) |
| Perpendicular Gradient: | \((\nabla \times \mathbf{V} \cdot \mathbf{y}) \) \(e\) = \( (1/\mathbf{A}_e) \sum_{e \in \mathbf{E}_v} \ell_{e,v} d_e \mathbf{y}_{e} \) |
| Perpendicular Flux: | \((\mathbf{k} \times \mathbf{E} \cdot \mathbf{y}) \) \(e\) = \( (1/d_e) \sum_{e' \in \mathbf{E}_e} w_{e,e'} l_e \mathbf{y}_{e'} \) |
| Cell to Vertex interpolation: | \\{ \mathbf{y} \} \(V \) \(v\) = \( (1/A_e) \sum_{e \in \mathbf{E}_v} R_{i,v} A_i \mathbf{y}_i \) |
| Vertex to Edge interpolation: | \\{ \mathbf{y} \} \(E \) \(e\) = \( \sum_{v \in \mathbf{E}_e} \mathbf{y}_v / 2 \) |
| Edge to Cell interpolation: | \\{ \mathbf{y} \} \(I \) \(i\) = \( (1/A_i) \sum_{e \in \mathbf{E}_i} \mathbf{y}_e A_e / 2 \) |
| Cell to Edge interpolation: | \\{ \mathbf{y} \} \(E \) \(e\) = \( \sum_i \rho E E (\mathbf{y}_i l_i / 2) \) |

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. \( R_{i,v} \) and \( A_i \) 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 [25].
introduced to the discrete equation. Here, we employ a classical bi-harmonic viscosity, which is modified to be energetically consistent. Concretely, we choose

\[ D_{\text{bihar}}[h, u] = -\frac{1}{\{h\}_E} \hat{\Delta}_{E\to E} \left( \nu_h \ast \{h\}_E \ast \left( \Delta_{E\to E} u \right) \right), \]

where \( \hat{\Delta}_{E\to E} = \nabla_{l\to E} \nabla_{E\to l} - \sqrt{3} \nabla_{E\to E} \left( k \cdot \nabla \times \right)_{E\to E} \) is a discrete approximation to an anisotropic vectorial Laplace-Beltrami operator (see [30]).

The appearance of \( h \) is motivated by the form of physical viscosities in the shallow water equation (see, e.g., [3]), 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 \( G[h, u] = D_{\text{bihar}}[h, u] + f_{\text{drag}}[h, u] + f_{\text{wind}}[h] \), we obtain for (3.1) the energy equality

\[ \frac{dH}{dt}(V) = -\sum_{k=1}^{L} \rho_k (\nu_h \ast \{h_k\}_E \ast \Delta_{E\to E} u_k, \Delta_{E\to E} u_k)_E - c_{\text{drag}}(\{\sqrt{2K[u_{\text{bot}}]}\}_E \ast u_{\text{bot}}, u_{\text{bot}})_E + (u_k, \tau_{\text{wind}})_E, \]

which shows that the smoothing and damping terms are energy dissipating. The horizontal viscosity \( \nu_h \in X_l \) 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 multi-layer 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 \( h_k \), the vertical Laplacian will depend non-linearly on the variable \( h \). 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.

**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

\[ \delta^2 H[V][W] = (\delta H[V])'(W) = \left( g \left( R w^h_k \right)_k + \rho_k (u_k \ast w^u_k)_k \right), \]

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

\[ \delta^2 H[V^{\text{ref}}] = \begin{pmatrix} gR & 0 \\ 0 & \text{diag}(\rho \{h^{\text{ref}}\}_E) \end{pmatrix}. \]

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

\[ J'[V; W] = \text{diag}_{k=1,2,...,L} \frac{1}{\rho_k} \begin{pmatrix} 0 & 0 \\ 0 & Q'[V; W] \end{pmatrix}, \]
where

\[ Q'[V; W]y = \frac{1}{2} \left( \{q'[V; W]\}_E \ast \left( \hat{k} \times y \right) + \hat{k} \times \left( \{q'[V; W]\}_E \ast y \right) \right) \]

and \( q'[V; W] = (\langle \hat{k} \cdot \nabla x \rangle_{E \rightarrow V} w^n) / \{h\}_V - (\langle \hat{k} \cdot \nabla x \rangle_{E \rightarrow E} u + f) \ast \{w^n\}_V / \{h\}_V^2 \). We remark that due the sparsity-pattern of \( k \times E \rightarrow E \), which has the most entries of any of the discrete operators considered, this term is by far the most expensive to evaluate in practice.

### 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

\[
\begin{array}{c|c}
0 & a_{i,j} \\
c_i & b_j \\
\end{array}
\]

which represents the method in terms of the remainder as

\[
v_{n,i} = \exp(c_i \Delta t A_n) V_n + \Delta t \sum_{j=1,\ldots,i-1} a_{i,j} (\Delta t A_n) \tau_n(v_{n,j}),
\]

\[
V_{n+1} = \exp(\Delta t A_n) V_n + \Delta t \sum_{j=1,\ldots,S} b_j (\Delta t A_n) \tau_n(v_{n,j}),
\]

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

\[
v_{n,i} = V_n + \varphi_1(c_i \Delta t A_n) F(V_n) + \Delta t \sum_{j=2,\ldots,i-1} a_{i,j} (\Delta t A_n) R_n(v_{n,j}),
\]

\[
V_{n+1} = V_n + \varphi_1(\Delta t A_n) F(V_n) + \Delta t \sum_{j=2,\ldots,S} b_j (\Delta t A_n) R_n(v_{n,j}).
\]

The two stage method from section 4.1 taken from [18] is characterized by the Butcher tableau

\[
\begin{array}{c|ccc}
0 & 0 & 0 \\
c_2 & 0 & \varphi_1(c_2) \\
\varphi_1 - (1/c_2) \varphi_2 & (1/c_2) \varphi_2 \\
\end{array}
\]

i.e. \( a_{2,1}(\cdot) = c_2 \varphi_1(c_2), b_1 = \varphi_1 - (1/c_2) \varphi_2 \), and \( b_2 = (1/c_2) \varphi_2 \), where \( c_2 \in (0,1) \). A third-order three stage method (also taken from [18]) is given by the Butcher tableau

\[
\begin{array}{c|cccc}
0 & 0 & 0 & 0 \\
c_2 & 0 & 0 & 0 \\
c_3 & \varphi_1(c_3) - a_{3,2} & \gamma c_2 \varphi_2(c_2) + (c_3^2/c_2) \varphi_2(c_3) & 0 \\
\varphi_1 - b_2 - b_3 & (\gamma/(\gamma c_2 + c_3)) \varphi_2 & (1/(\gamma c_2 + c_3)) \varphi_2 \\
\end{array}
\]

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 \( \gamma = -(3c_3^2 - 2c_3)/(3c_3^2 - 2c_2) \).

32
References

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

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

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

[4] E. Celledoni and I. Moret, *A Krylov projection method for systems of ODEs*, Applied Numerical Mathematics, 24 (1997), pp. 365–378.

[5] D. B. Chelton, R. A. deSzoeke, M. G. Schlax, K. E. Naggar, and N. Siwertz, *Geographical variability of the first baroclinic Rossby radius of deformation*, Journal of Physical Oceanography, 28 (1998), pp. 433–460.

[6] Q. Chen, M. Gunzburger, and T. Ringler, *A scale-invariant formulation of the anticipated potential vorticity method*, Monthly Weather Review, 139 (2011), pp. 2614–2629.

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

[8] V. Druskin, A. Greenbaum, and L. Knizhnerman, *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM Journal on Scientific Computing, 19 (1998), pp. 38–54.

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

[10] J. K. Dukowicz and R. D. Smith, *Implicit free-surface method for the Bryan-Cox-Semtner ocean model*, Journal of Geophysical Research, 99 (1994), pp. 7991–8014.

[11] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM Journal on Numerical Analysis, 21 (1984), pp. 352–362.

[12] S. Gaudreault and J. A. Pudykiewicz, *An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere*, Journal of Computational Physics, 322 (2016), pp. 827–848.

[13] C. Greif and J. Varah, *Iterative solution of skew-symmetric linear systems*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 584–601.

[14] T. S. Haut, T. Babb, P. G. Martinsson, and 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 Journal of Numerical Analysis, 36 (2015), pp. 688–716.

[15] R. L. Higdon, *A two-level time-stepping method for layered ocean circulation models: further development and testing*, Journal of Computational Physics, 206 (2005), pp. 463–504.
[16] R. L. Higdon, *Numerical modelling of ocean circulation*, Acta Numerica, 15 (2006), pp. 385–470.

[17] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1911–1925.

[18] M. Hochbruck and A. Ostermann, *Explicit exponential Runge–Kutta methods for semilinear parabolic problems*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 1069–1090.

[19] ———, *Exponential integrators*, Acta Numerica, 19 (2010), pp. 209–286.

[20] R. V. Madala, *Efficient time integration schemes for atmosphere and ocean models*, in Finite-difference techniques for vectorized fluid dynamics calculations, D. L. Book, ed., Springer-Verlag, New York, Berlin, 1981, ch. 4, pp. 56–74.

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

[22] J. Niesen and W. M. Wright, *Algorithm 919: A Krylov subspace algorithm for evaluating the $\varphi$-functions appearing in exponential integrators*, ACM Transactions on Mathematical Software (TOMS), 38 (2012), p. 22.

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

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

[25] T. D. Ringler, J. Thuburn, J. B. Klemp, and W. C. Skamarock, *A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids*, Journal of Computational Physics, 229 (2010), pp. 3065–3090.

[26] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 209–228.

[27] R. B. Sidje, *Expokit: A Software Package for Computing Matrix Exponentials*, ACM Trans. Math. Softw, 24 (1998), pp. 130–156.

[28] A. L. Stewart and P. J. Dellar, *An energy and potential enstrophy conserving numerical scheme for the multi-layer shallow water equations with complete Coriolis force*, Journal of Computational Physics, 313 (2016), pp. 99–120.

[29] A. Y. Suhov, *An Accurate Polynomial Approximation of Exponential Integrators*, Journal of Scientific Computing, 60 (2014), pp. 684–698.

[30] J. Thuburn and C. J. Cotter, *A framework for mimetic discretization of the rotating shallow-water equations on arbitrary polygonal grids*, SIAM Journal on Scientific Computing, 34 (2012), pp. B203–B225.

[31] J. Thuburn, T. Ringler, W. Skamarock, and J. Klemp, *Numerical representation of geostrophic modes on arbitrarily structured c-grids*, Journal of Computational Physics, 228 (2009), pp. 8321–8335.
[32] H. D. Vo and R. B. Sidje, Approximating the large sparse matrix exponential using incomplete orthogonalization and Krylov subspaces of variable dimension, Numerical Linear Algebra with Applications, 24 (2017).

[33] P. J. Wolfram, T. D. Ringler, M. E. Maltrud, D. W. Jacobsen, and M. R. Petersen, Diagnosing Isopycnal Diffusivity in an Eddying, Idealized Midlatitude Ocean Basin via Lagrangian, in Situ, Global, High-Performance Particle Tracking (LIGHT), Journal of Physical Oceanography, 45 (2015), pp. 2114–2133.