RESEARCH ARTICLE

Time-parallel integration and phase averaging for the nonlinear shallow-water equations on the sphere

Hiroe Yamazaki[1] | Colin J. Cotter[1] | Beth A. Wingate[2]

[1]Department of Mathematics, Imperial College London, London, UK

[2]Department of Mathematics, University of Exeter, Exeter, UK

Correspondence
Hiroe Yamazaki, Department of Mathematics, Imperial College London, London, SW7 2AZ, UK.
Email: h.yamazaki@imperial.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/R029628/1

We describe a proof-of-concept development and application of a phase-averaging technique to the nonlinear rotating shallow-water equations on the sphere, discretised using compatible finite-element methods. Phase averaging consists of averaging the nonlinearity over phase shifts in the exponential of the linear wave operator. Phase averaging aims to capture the slow dynamics in a solution that is smoother in time (in transformed variables), so that larger timesteps may be taken. We overcome the two key technical challenges that stand in the way of studying the phase averaging and advancing its implementation: (1) we have developed a stable matrix exponential specific to finite elements and (2) we have developed a parallel finite averaging procedure. Following recent studies, we consider finite-width phase-averaging windows, since the equations have a finite timescale separation. In our numerical implementation, the averaging integral is replaced by a Riemann sum, where each term can be evaluated in parallel. This creates an opportunity for parallelism in the timestepping method, which we use here to compute our solutions. Here, we focus on the stability and accuracy of the numerical solution. We confirm that there is an optimal averaging window, in agreement with theory. Critically, we observe that the combined time discretisation and averaging error is much smaller than the time discretisation error in a semi-implicit method applied to the same spatial discretisation. An evaluation of the parallel aspects will follow in later work.

KEYWORDS
mixed finite elements, numerical weather prediction, parallel in time, phase averaging

1 | INTRODUCTION

Phase averaging is a technique for approximating highly oscillatory partial differential equations (PDEs), such as the equations that govern the dynamics of models of large-scale geophysical fluid dynamics. Examples of phase averaging include the solution of ordinary differential equations (ODEs; Sanders *et al.*, 2007) and the analysis of fast singular limits (Klainerman and Majda, 1981; Majda and Embid, 1998) in simple geometries.

Phase averaging is an important concept used in understanding the influence of oscillations on mean flows, with connections to Lagrangian averaging. In fact, theoretical and numerical work on phase averaging suggest that nonlinear phase averaging provides more accurate leading-order dynamics (e.g., Wagner and Young, 2015; Kafiabad *et al.*, 2021). Therefore, the tools developed in this work could contribute to further understanding this challenging topic in planetary fluid dynamics. In addition, phase averaging has been proposed as a way to introduce more parallelization into numerical models (Haut and Wingate, 2014; Ariel *et al.*, 2016; Peddle *et al.*, 2019). However, phase averaging in the spherical geometries required for large-scale geophysical fluid dynamics applications has two key technical barriers that stand in the way of studying the phase average as a potentially interesting physical quantity as well as a potential basis for advancing parallel computing: (1) the development of a stable matrix exponential required for the mapping and (2) the development of a parallel finite-interval averaging procedure. This article overcomes these two challenges, providing a proof-of-concept model of phase-averaged timestepping in spherical geometries, as well as a study of the method's accuracy.

We focus on the rotating shallow-water equations in the low Rossby number regime. The phase-averaging technique averages the nonlinearity over all phases of the fast waves to obtain an approximation of the slow dynamics with no fast oscillations present. Since the magnitude of time discretisation errors is governed by the magnitude of time derivatives in the equation, this means that these errors can be reduced for phase-averaged models, allowing larger timesteps to be taken in an accurate numerical integration. However, there is an additional error introduced by the phase averaging itself, which we shall investigate in this work.

For the rotating shallow-water equations, Majda and Embid (1998) showed that taking the low Rossby number limit in the phase-averaged equations leads to the quasigeostrophic equations. However, the quasigeostrophic equations are not uniformly valid, which is one reason why their 3D counterpart is not used for operational weather forecasting. Haut and Wingate (2014) proposed using a phase average of the nonlinearity over a finite-width averaging window T. For small T the original equations are recovered, and for large T the full phase averaging is recovered, which filters all fast dynamics. They proposed to perform the phase averaging numerically, replacing the phase integral with a numerical quadrature rule. The idea is to use parallel computation to implement the averaging: each term in the quadrature rule can be evaluated independently and hence in parallel. Peddle *et al.* (2019) showed that, given a chosen timestepping scheme and

timestep size, there is an optimum averaging window T that minimises the total error (combining numerical timestepping error and averaging error). Below the optimum the timestepping error dominates (and the timestepping scheme may become unstable), and above the optimum the averaging error dominates. For larger timesteps, the optimal T is larger, requiring more quadrature points in the average and consequently requiring more computational cores.

The finite-window phase-averaged model trades computational cores for larger timesteps at the expense of accuracy (because of the additional error from phase averaging). If the level of accuracy is insufficient, a time-parallel predictor–corrector approach might be used to increase accuracy. Haut and Wingate (2014) proposed to use the averaged model as the coarse propagator in a highly efficient Parareal iteration, demonstrating parallel speedups of a factor of 100 in a rotating shallow-water test case. Peddle *et al.* (2019) proved convergence of the iterative Parareal procedure for highly oscillatory PDEs with quadratic nonlinearity, making use of the optimal averaging window T at finite Rossby number. Bauer *et al.* (2022) created a hierarchy of higher-order averaged models that increase accuracy through increasing the number of variables; this type of hierarchy is ideal for predictor–corrector approaches such as revisionist integral deferred correction methods (Ong *et al.*, 2016) and the parallel full approximation scheme in space and time algorithm (Minion, 2011) that compute more accurate correction steps in parallel as new predictor steps are being taken. These are motivations for the work in this article, but here we focus on the impact of the averaging on the rotating shallow-water solution.

This article addresses the challenge of producing a proof of concept of the numerical averaging technique applied to the rotating shallow-water equations on the sphere. This required us to find a performant way of implementing the necessary matrix exponentials, and to find a discretisation approach that avoids instabilities in those exponentials from spurious eigenvalues. It also required us to produce a parallel implementation of the numerical averaging. Our proof of concept allows us to evaluate, for the first time, the solution quality arising from the averaging technique against a test case that is used in dynamical core development.

The rest of this work is organised as follows. In Section 2, we describe the phase-averaging procedure and how it can be applied to the rotating shallow-water equations. We also describe our approach to timestepping these equations. In Section 3, we present our numerical results, examining the impact of averaging window T and timestep Δt on the errors associated with time integration. Finally, in Section 4 we provide a summary.

2 | DESCRIPTION OF THE METHOD

2.1 | Shallow-water equations

In this section, we describe the shallow-water equations and link them to the general notation framework for phase averaging that will be used in subsequent sections.

We begin with the nonlinear shallow-water equations on a two-dimensional surface that is embedded in three dimensions,

$$\boldsymbol{u}_t + f\boldsymbol{u}^\perp + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + g\nabla\eta = 0, \quad (1)$$

$$\eta_t + H\nabla \cdot \boldsymbol{u} + \nabla \cdot [\boldsymbol{u}(\eta - b)] = 0, \quad (2)$$

where $\boldsymbol{u}$ is the horizontal velocity, f is the Coriolis parameter, $\boldsymbol{u}^\perp = \boldsymbol{k} \times \boldsymbol{u}$ where $\boldsymbol{k}$ is the normal to the surface, and g is the gravitational acceleration; η is the free surface elevation, H is the mean layer thickness, and b is the height of the lower boundary, where the layer depth $h = H + \eta - b$; ∇ and $\nabla\cdot$ are appropriate invariant gradient and divergence operators defined on the surface. Here we will concentrate on the case of the equations being solved on the surface of the sphere, so there are no boundary conditions to consider.

Then we rewrite the equations as

$$\boldsymbol{U}_t = \mathcal{L}\boldsymbol{U} + \mathcal{N}(\boldsymbol{U}), \quad (3)$$

where the vector of unknowns $\boldsymbol{U}(t) = (\boldsymbol{u}, \eta)$. The matrix $\mathcal{L}$ represents a linear operator and $\mathcal{N}(\cdot)$ is a nonlinear operator which satisfies

$$\mathcal{L}\boldsymbol{U} = \begin{pmatrix} -f(\cdot)^\perp & -g\nabla \\ -H\nabla\cdot & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \eta \end{pmatrix} = \begin{pmatrix} -f(\boldsymbol{u})^\perp & -g\nabla\eta \\ -H\nabla \cdot \boldsymbol{u} & 0 \end{pmatrix}, \quad (4)$$

$$\mathcal{N}(\boldsymbol{U}) = \begin{pmatrix} -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \\ -\nabla \cdot [\boldsymbol{u}(\eta - b)] \end{pmatrix}. \quad (5)$$

2.2 | Phase averaging

Now we consider an approximation to Equation 3 by averaging the nonlinearity over the fast oscillations. First we introduce a coordinate transformation,

$$\boldsymbol{V}(t) = e^{-\mathcal{L}t}\boldsymbol{U}(t), \quad (6)$$

$$\frac{\partial \boldsymbol{V}}{\partial t}(t) = e^{-\mathcal{L}t}\mathcal{N}\left(e^{\mathcal{L}t}\boldsymbol{V}(t)\right), \quad (7)$$

where $e^{\mathcal{L}t}\boldsymbol{V}$ is the solution at time t to the linear part of the equation $\boldsymbol{U}_t = \mathcal{L}\boldsymbol{U}$ with an initial condition of $\boldsymbol{U}(0) = \boldsymbol{V}$.

To allow averaging the model over a finite time interval s around time t, we follow Bauer *et al.* (2022) and extend Equations 6 and 7 by introducing a phase variable s,

$$\boldsymbol{V}(t, s) = e^{-\mathcal{L}(t+s)}\boldsymbol{U}(t, s), \quad (8)$$

$$\frac{\partial \boldsymbol{V}}{\partial t}(t, s) = e^{-\mathcal{L}(t+s)}\mathcal{N}\left(e^{\mathcal{L}(t+s)}\boldsymbol{V}(t, s)\right). \quad (9)$$

Of course, we are only interested in the solution at $s = 0$, but the averaging approximation below corresponds to the assumption that the solution $V(t, s)$ is insensitive to changes in s. Bauer *et al.* (2022) introduced higher order approximations that parameterise the sensitivity to s near $s = 0$.

An averaging approximation to Equation 9 over the averaging window T with respect to a weight function ρ can be written as

$$\frac{\partial \overline{\boldsymbol{V}}}{\partial t}(t) = \frac{1}{2T} \int_{-T}^{T} \rho\left(\frac{s}{T}\right) e^{-\mathcal{L}(t+s)}\mathcal{N}\left(e^{\mathcal{L}(t+s)}\overline{\boldsymbol{V}}(t)\right) ds, \quad (10)$$

where $\overline{\boldsymbol{V}}$ denotes the averaged $\boldsymbol{V}$. In our computations, ρ is given by

$$\rho(s) = \begin{cases} \rho_0 \exp(-1/((s - 0.5)(s + 0.5))), & -0.5 < s < 0.5, \\ 0, & |s| \geq 0.5, \end{cases} \quad (11)$$

with an appropriate weighting factor, but other weight functions may be used provided that they integrate to 1.

It is very important to note that this averaging integral is *not* integrating along the history of $\overline{V}$ (which would be an integral in the time variable t), but is instead integrating over the phase shift variable s in the exponential operators. Equation 10 is similar to the phase averaging in Peddle *et al.* (2019). The main difference is that our phase shift is defined in the mapping in Equation 8, whereas Peddle *et al.* (2019) introduce the phase shift parameter, s, in the nonlinear term.

The unaveraged Equation 9 has oscillatory nonautonomous contributions coming from the exponentials (corresponding to fast waves in the untransformed Equation 3), and the averaging approximation filters out the components of these contributions with time period below T. This approximation is best understood in various limits. In the limit as $T \to 0$, $\rho(s/T)/(2T) \to \delta(s)$, so we recover the original unaveraged Equation 7. In the limit $T \to \infty$, the asymptotic approximation $U(t) = \exp(-t/\epsilon\mathcal{L}_0)\overline{V}(t) + \mathcal{O}(\epsilon)$ is obtained, where $\mathcal{L} = \epsilon\mathcal{L}_0$ (Majda and Embid, 1998). This asymptotic

approximation describes a slowly evolving solution $\overline{V}(t)$ with superposed fast linear waves: the classic "slow manifold" picture. For intermediate values of T, we can select which nonautonomous contributions we want to remove and which contributions we want to retain. This can be important if the interaction between the two exponentials inside and outside $\mathcal{N}$ in Equation 9 (e.g., through triad interactions in the case of quadratic nonlinearity) leads to oscillations with frequencies approaching those of the dynamics of $\overline{V}$; in that case we would lose long time accuracy if we increased T to remove those oscillations.

For an implementable method, we replace the integral with a Riemann sum to obtain

$$\frac{\partial \overline{V}}{\partial t} \simeq \sum_{k=-N}^{N} w_k e^{-\mathcal{L}(t+s_k)} \mathcal{N}\left(e^{\mathcal{L}(t+s_k)}\overline{V}(t)\right) \quad (12)$$

$$:= \left\langle e^{-\mathcal{L}(t+s)} \mathcal{N}\left(e^{\mathcal{L}(t+s)}\overline{V}(t)\right)\right\rangle_s, \quad (13)$$

where w_k are appropriate weight coefficients (obtained from the product quadrature rule weights and the value of the weight function ρ evaluated at the quadrature points) and $s_k = kT/N$. This defines the angle bracket notation in Equation 13, used in the following section. Since this is an oscillatory integral, there is little reduction in the quadrature error until the oscillations are resolved on the quadrature points, after which the error collapses quickly. In this work we used equispaced quadrature points, with four points per time period of the fastest frequency of $\mathcal{L}$; the resulting dynamics was very insensitive to increasing this number beyond four.

Whilst Equation 13 looks complicated, each of the terms in the sum is independent and so they can be evaluated in parallel if computational resources are available.

2.3 | Time discretisation

In this section, we describe our time integration approach. The general summary is that we use an averaged version of a Lawson exponential integrator; see Hochbruck and Ostermann (2010) for a review. This means that we apply a standard time integration method to Equation 13, and then transform from V back to U to restrict exponentiation to time intervals of $\mathcal{O}(\Delta t)$. These shorter exponentiations are less expensive to compute numerically.

For the classical fourth-order Runge–Kutta scheme, we obtain

$$\overline{U}_1 = \overline{U}^n + \frac{\Delta t}{2}\left\langle e^{-\mathcal{L}s}\mathcal{N}(e^{\mathcal{L}s}\overline{U}^n)\right\rangle_s, \quad (14)$$

$$\overline{U}_2 = e^{\mathcal{L}\frac{\Delta t}{2}}\overline{U}^n + \frac{\Delta t}{2}\left\langle e^{-\mathcal{L}s}\mathcal{N}(e^{\mathcal{L}\frac{\Delta t}{2}}e^{\mathcal{L}s}\overline{U}_1)\right\rangle_s, \quad (15)$$

$$\overline{U}_3 = e^{\mathcal{L}\frac{\Delta t}{2}}\overline{U}^n + \Delta t\left\langle e^{-\mathcal{L}s}\mathcal{N}\left(e^{\mathcal{L}s}\overline{U}_2\right)\right\rangle_s, \quad (16)$$

$$\overline{U}^{n+1} = e^{\mathcal{L}\Delta t}\overline{U}^n + \frac{\Delta t}{6}\left[e^{\mathcal{L}\Delta t}\left\langle e^{-\mathcal{L}s}\mathcal{N}\left(e^{\mathcal{L}s}\overline{U}^n\right)\right\rangle_s \right.$$
$$+ 2e^{\mathcal{L}\frac{\Delta t}{2}}\left\langle e^{-\mathcal{L}s}\mathcal{N}\left(e^{\mathcal{L}\frac{\Delta t}{2}}e^{\mathcal{L}s}\overline{U}_1\right)\right\rangle_s$$
$$+ 2e^{\mathcal{L}\frac{\Delta t}{2}}\left\langle e^{-\mathcal{L}s}\mathcal{N}\left(e^{\mathcal{L}s}\overline{U}_2\right)\right\rangle_s$$
$$\left. + \left\langle e^{-\mathcal{L}s}\mathcal{N}\left(e^{\mathcal{L}\frac{\Delta t}{2}}e^{\mathcal{L}s}\overline{U}_3\right)\right\rangle_s\right]. \quad (17)$$

2.4 | Chebyshev exponentiation

Implementing exponentials $e^{\mathcal{L}t}$ of grid-based discretisations on the sphere is challenging, because the efficiency and parallel scalability of these discretisations relies upon matrix sparsity, and $e^{\mathcal{L}t}$ is not sparse. Instead we need to consider scalable algorithms that construct $e^{\mathcal{L}t}$ using only sparse matrix applications and local operations. This is done for the first time in this work in the context of numerical averaging techniques applied to the rotating shallow-water equations on the sphere.

To implement the exponential operator $e^{\mathcal{L}t}$, we use a Chebyshev approximation,

$$e^{\mathcal{L}t}\overline{U} \approx \sum_{k=0}^{N} a_k P_k(\mathcal{L}t)\overline{U}, \quad (18)$$

where N is the number of polynomials, a_k are polynomial coefficients, and P_k are modified Chebyshev polynomials. The modification is a change of coordinates transforming the imaginary interval

$$\{z = iy : y \in [-L, L]\} \quad (19)$$

to the unit interval $[-1, 1]$, where $L = |\lambda_{\max}|t_{\max}$, $\lambda_{\max}$ is the eigenvalue of $\mathcal{L}$ with maximum magnitude, and the approximation is valid for times $|t| < t_{\max}$.

In general, Krylov subspace methods (of which the Chebyshev approach is one) for oscillatory problems require a number of iterations proportional to the Courant number (Hochbruck and Ostermann, 2010; Pieper *et al.* 2019). The advantage with Chebyshev polynomials is that there is a three-term recurrence, so we do not need to store and compute with the entire Krylov basis. This method starts with the three-term recurrence for Chebyshev polynomials, modified with the above coordinate transformation to get

$$P_0(l) = 1, \quad (20)$$

$$P_1(l) = \frac{-il}{L}, \quad (21)$$

$$P_{k+1}(l) = \frac{2lP_k(l)}{iL} - P_{k-1}(l). \quad (22)$$

Therefore the $P_k(\mathcal{L}t)\overline{\boldsymbol{U}}$ in Equation 18 are obtained recursively as

$$P_0(\mathcal{L}t)\overline{\boldsymbol{U}} = \overline{\boldsymbol{U}}, \quad (23)$$

$$P_1(\mathcal{L}t)\overline{\boldsymbol{U}} = \frac{-it\mathcal{L}\overline{\boldsymbol{U}}}{L}, \quad (24)$$

$$P_{k+1}(\mathcal{L}t)\overline{\boldsymbol{U}} = \frac{2tP_k(\mathcal{L}t)\mathcal{L}\overline{\boldsymbol{U}}}{iL} - P_{k-1}(\mathcal{L}t)\overline{\boldsymbol{U}}. \quad (25)$$

This avoids explicitly forming polynomials of matrices, by instead just forming the action of polynomials of matrices on vectors recursively, by repeated application of $\mathcal{L}$. Further, this application is performed matrix-free by writing the action of the matrix on a vector equivalently as a set of integrals (in the usual finite-element manner), although a mass matrix solve is required in this formulation.

As usual for Chebyshev polynomials, the coefficients a_k are computed by using a fast Fourier transform (Trefethen, 2019). Following Gander and Güttel (2013), we then discard coefficients starting at the highest degree and going downwards until the total magnitude of discarded coefficients exceeds some threshold (1.0×10^{-6} for our results).

We note that, for larger averaging windows, higher degree Chebyshev approximations are needed, meaning that the application of the approximated exponential operator takes longer. This is illustrated in Table 1. We see that the growth is approximately linear in both mesh resolution and time.

For a fully performant method, in the future we will incorporate the rational approximation of exponential integrators technique of Haut *et al.* (2016), which approximates the exponential along the imaginary axis by a sum of rational polynomials. This method can be parallelised over the sum; each parallel term requires the solution of a complex-valued elliptic problem of the form $aI + b\mathcal{L}$, where a and b are rational coefficients. Some more details of the implementation and examination of parallel performance are provided in Schreiber

et al. (2018). When extending to three dimensions, it may also be useful to exploit the vertical horizontal tensor product structure in the exponential, as discussed in Croci and Muñoz-Matute (2022).

However, here we are focussed on the error behaviour of the averaging technique, so the Chebyshev approximation suffices for this purpose. The goal here is to use parallelism as a way of understanding the impact of the averaging technique on the solution of the PDE.

2.5 | Spatial discretisation

In this study, we used the compatible finite-element discretisation for the nonlinear rotating shallow-water equations on the sphere given in Gibson *et al.* (2019). This was chosen because it leads to a discretised $\mathcal{L}$ that still has purely imaginary eigenvalues, and has a discrete Helmholtz decomposition that correctly separates the fast inertia gravity waves and the slow balanced motion. This is critical to addressing the challenge of applying numerical averaging to the rotating shallow-water equations on the sphere. Any similar approach to the discretisation with these properties (e.g., a spectral discretisation or C-grid finite-difference method) is expected to produce similar results.

The compatible finite-element discretisation is built around a pair of spaces $\mathbb{V}_1 \subset H(\text{div})$ and $\mathbb{V}_2 \subset L_2$, selecting $\boldsymbol{u} \in \mathbb{V}_1$ and $\eta \in \mathbb{V}_2$. In these examples we chose BDM2 for $\mathbb{V}_1$ and P1$_{DG}$ for $\mathbb{V}_2$, producing a second-order scheme in space.

The discrete linear operator $\mathcal{L} : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{V}_1 \times \mathbb{V}_2$ is then defined by $\mathcal{L}(\boldsymbol{u}, \eta) = (\boldsymbol{u}_1, \eta_1)$, where

$$\int_\Omega \boldsymbol{w} \cdot \boldsymbol{u}_1 \, dx$$
$$= -\int_\Omega \boldsymbol{w} \cdot (f\boldsymbol{u}^\perp) \, dx + \int_\Omega (\nabla \cdot \boldsymbol{w}) g\eta \, dx, \quad \forall \boldsymbol{w} \in \mathbb{V}_1, \quad (26)$$

$$\int_\Omega \phi \, \eta_1 \, dx = -H \int_\Omega \phi \nabla \cdot \boldsymbol{u} \, dx, \quad \forall \phi \in \mathbb{V}_2. \quad (27)$$

Implementing this requires the solution of a block diagonal system for the basis coefficients of η_1 and a sparse (but

TABLE 1 Table showing the number of Chebyshev iterations required to provide a Chebyshev approximation of the exponential that is truncated once the total magnitude of the coefficients of the remainder is less than 1.0×10^{-6}, for different mesh refinements and times.

	Mesh refinement level	$\Delta s = 15$ min	$\Delta s = 30$ min	$\Delta s = 1$ hr
Number of iterations	3	10	15	24
Number of iterations	4	16	25	41
Number of iterations	5	26	42	73

globally coupled) system for the basis coefficients of $\boldsymbol{u}_1$. For the latter we observe a mesh-independent number of iterations when solving using a scalable iterative method (described below).

The discrete nonlinear operator $\mathcal{N} : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{V}_1 \times \mathbb{V}_2$ is then similarly defined by $\mathcal{N}(\boldsymbol{u}, \eta) = (\boldsymbol{u}_2, \eta_2)$, where

$$
\begin{aligned}
\int_\Omega \boldsymbol{w} \cdot \boldsymbol{u}_2 \, dx = & \int_\Omega \boldsymbol{u} \cdot \nabla^\perp(\boldsymbol{u}^\perp \cdot \boldsymbol{w}) \, dx \\
& - \int_\Gamma \left[\left[\boldsymbol{n}^\perp(\boldsymbol{u}^\perp \cdot \boldsymbol{w})\right]\right] \cdot \tilde{\boldsymbol{u}} \, dS \\
& + \int_\Omega \nabla \cdot \boldsymbol{w} \left(\frac{1}{2}|\boldsymbol{u}|^2\right) \, dx, \quad \forall \boldsymbol{w} \in \mathbb{V}_1, \quad (28)
\end{aligned}
$$

$$
\begin{aligned}
\int_\Omega \phi \, \eta_2 \, dx = & \int_\Omega \nabla \phi \, \boldsymbol{u}(\eta - b) \, dx \\
& + \int_\Gamma [[\phi \boldsymbol{u}]] \, (\eta - b) \, dS, \quad \forall \phi \in \mathbb{V}_2, \quad (29)
\end{aligned}
$$

where $\boldsymbol{n}$ is the outward-pointing unit normal vector to the boundary $\partial \Omega$ of Ω, Γ denotes the set of interior facets in the mesh, with the two sides of each facet arbitrarily labeled by $+$ and $-$, the jump operator is defined by

$$
[[q]] = q^+ \boldsymbol{n}^+ + q^- \boldsymbol{n}^-, \quad (30)
$$

$$
[[\boldsymbol{v}]] = \boldsymbol{v}^+ \cdot \boldsymbol{n}^+ + \boldsymbol{v}^- \cdot \boldsymbol{n}^-, \quad (31)
$$

for any scalar q and vector $\boldsymbol{v}$, and $\tilde{\boldsymbol{u}}$ is evaluated on the upwind side as

$$
\tilde{\boldsymbol{u}} = \begin{cases} \boldsymbol{u}^+ & \text{if } \boldsymbol{u} \cdot \boldsymbol{n}^+ < 0, \\ \boldsymbol{u}^- & \text{otherwise.} \end{cases} \quad (32)
$$

Implementing this requires the solution of the same systems for $\boldsymbol{u}_2$ and η_2 as $\boldsymbol{u}_1$ and η_1, respectively. This upwind stabilisation of the advection terms is the only dissipative term and there are no explicit dissipation terms in the model.

Our code implementation was written using Firedrake (Rathgeber *et al.*, 2016), an automated system for the solution of partial differential equations using the finite-element method, with the resulting matrix systems being solved using Portable, Extensible Toolkit for Scientific Computation libraries (Balay *et al.*, 2021). A direct solver was used for the block diagonal systems for η and the conjugate gradient method preconditioned by incomplete Cholesky factorisation was used to solve the sparse systems for $\boldsymbol{u}$. The terms of the average are computed in parallel using the "ensemble parallelism" capability of Firedrake, which was implemented for this project. This implementation provides message passing interface

subcommunicators for the distribution of the terms of the averaging sum, with the sum being formed by reduction over the subcommunicators.

3 | NUMERICAL EXPERIMENTS

In this section, we show numerical results from a standard test case on the sphere described by Williamson *et al.* (1992). Here we use their test case number 5 (flow over a mountain), where the model is initialised with the layer depth and velocity fields that are in geostrophic balance:

$$
h = H - \left(R\Omega u_0 + \frac{u_0^2}{2}\right) \frac{z^2}{gR^2}, \quad (33)
$$

$$
\boldsymbol{u} = \frac{u_0}{R}(-y, x, 0), \quad (34)
$$

where $R = 6.37122 \times 10^6 \, \text{m}$ is the radius of the Earth, $\Omega = 7.292 \times 10^5 \, \text{s}^{-1}$ is the rotation rate of the Earth, (x, y, z) are the 3D Cartesian coordinates, the maximum zonal wind speed $u_0 = 20 \, \text{m}$, $g = 9.8 \, \text{m} \cdot \text{s}^{-2}$, and $H = 5960 \, \text{m}$. An isolated mountain is placed with its centre at latitude $\phi = \pi/6$ and longitude $\lambda = -\pi/2$. The height of the mountain is described as

$$
b = b_0 \left(1 - \frac{(\min[R_0^2, (\phi - \phi_\text{c})^2 + (\lambda - \lambda_\text{c})^2])^{1/2}}{R_0}\right), \quad (35)
$$

where $b_0 = 2000 \, \text{m}$ and $R_0 = \pi/9$. The sudden appearance of this mountain disturbing the balanced flow creates significant fast unbalanced inertia–gravity waves as well as triggering slow balanced vortex motion.

Icosahedral grids with a piecewise cubic approximation to the sphere are used in the model. The number of cells is $N_\text{C} = 20,480$, the maximum cell centre to cell centre distance is $263 \, \text{km}$, and the minimum distance is $171 \, \text{km}$. A timestep of $900 \, \text{s}$ is used in the averaged model unless stated otherwise. As there is no analytical solution for this problem, the model output is compared with a reference solution generated from a semi-implicit nonlinear shallow-water code provided by Gibson *et al.* (2019) (which we refer to as the "standard model"), using the same spatial resolution of $N_\text{C} = 20,480$ and the same spatial discretisation. A much smaller timestep of $22.5 \, \text{s}$ is used to generate the reference solution, to get as close as possible to the exact solution to the time-continuous space-discrete system we are trying to approximate with the averaging technique.

Figure 1 shows the field of the free surface elevation η at day 15 from the averaged model plotted in a
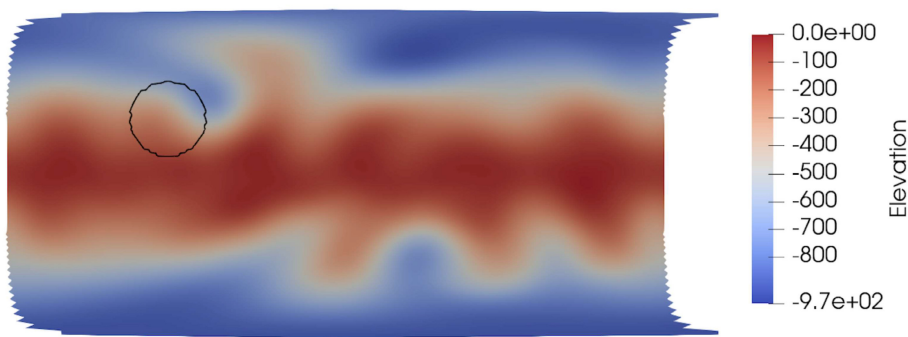
FIGURE 1 Free surface elevation η at day 15 from the averaged model. The timestep is $\Delta t = 900$ s and the averaging window is $T = 1$ hr. The solid line indicates the position of the mountain. [Colour figure can be viewed at wileyonlinelibrary.com]
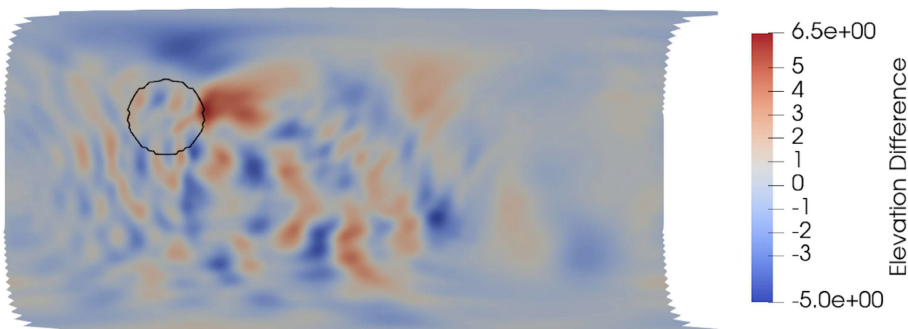

FIGURE 2 Errors in the elevation η at day 15 compared with the reference solution. The solid line indicates the position of the mountain. [Colour figure can be viewed at wileyonlinelibrary.com]
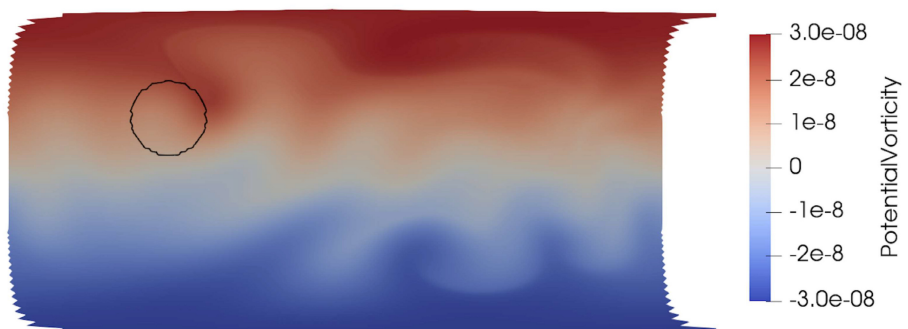
FIGURE 3 Potential vorticity at (top) day 15 and (bottom) day 50 from the averaged model. The timestep is $\Delta t = 900$ s and the averaging window is $T = 1$ hr. The solid lines indicate the position of the mountain. [Colour figure can be viewed at wileyonlinelibrary.com]
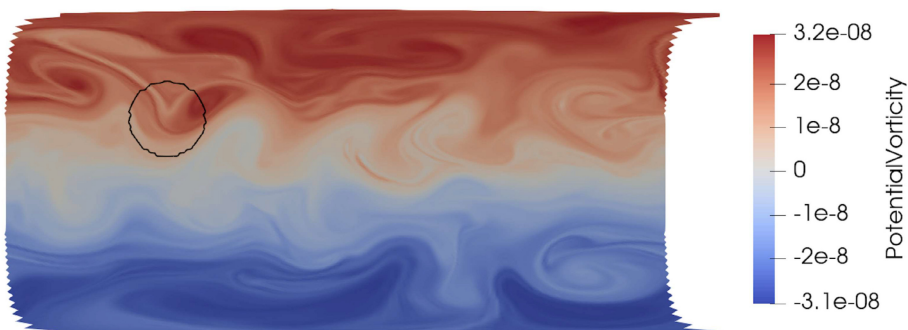
latitude–longitude space. The averaging window is $T = 1$ hr in this plot. The model successfully reproduces waves that travel around the globe as a result of the zonal flow interacting with the mountain. Figure 2 shows the errors in η at day 15 compared with the reference solution. We can see that errors are sufficiently small and not dominated by errors due to grid imprinting.

Figure 3 shows the fields of the potential vorticity at days 15 and 50 from the averaged model. The flow is only weakly nonlinear at day 15, and fine-scale structure has been generated at day 50 as the flow becomes more nonlinear. These results are consistent with the numerical results by Thuburn *et al.* (2014) and Shipton *et al.* (2018).
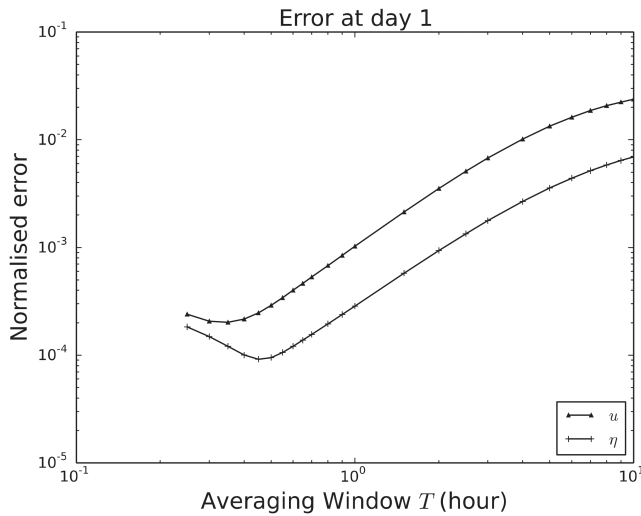
FIGURE 4 Normalised errors in the averaged model at day 1 versus the averaging window T. The solid curve with triangles shows the $H(\text{div})$ norm of $\boldsymbol{u}$, and the solid curve with crosses shows the L_2 norm of η, normalised by the norms of the reference solution. The timestep $\Delta t = 900$ s is fixed in all simulations. Note the clear existence of optimal averaging windows for each variable.

Now, we examine the impact of the averaging window T on the accuracy of the averaged model. Figure 4 shows the $H(\text{div})$ norm of $\boldsymbol{u}$ and the L_2 norm of η at day 1, normalised by the norms of the reference solution, plotted over various averaging windows. The spatial resolution of $N = 20,480$ and the timestep of $\Delta t = 900$ s were kept the same as in Figure 2, whereas a range of values between 0.25 and 10 hr were used for the the averaging window T. The result reveals the clear existence of

optimal averaging windows at around $T = 0.35$ hr for $\boldsymbol{u}$ and $T = 0.45$ hr for η, respectively; this will vary depending on the choice of norm. This result demonstrates that the behavior in the averaged model is consistent with the error bounds shown in Peddle et al. (2019). Figure 5 shows the time evolution of the errors in $\boldsymbol{u}$ and η up to day 5. For both variables, the minima in the error curves move to the left at day 2. From day 3, the errors at averaging windows $T \leq 0.3$ grow rapidly, showing that the model is slowly blowing up in those small averaging windows. This is because the fixed timestep is not resolving the unfiltered fast oscillations at these small averaging window widths.

Finally, we examine the accuracy of the averaged model when using different timestep size Δt. The curves with markers in Figure 6 show the $H(\text{div})$ norm of $\boldsymbol{u}$ and the L_2 norm of η at day 1 in the averaged model, normalised by the norms of the reference solution, when using three different timesteps: $\Delta t = 450$, 900, and 1350 s. The averaging window T was changed between $T_{\min} \leq T \leq 1$ hour, where $T_{\min} = 0.1$, 0.25, and 0.375 hr for the results using $\Delta t = 450$, 900, and 1350 s, respectively. When using T smaller than $T_{\min}$ that is corresponding to each timestep size, the model blows up within 1 day due to the timestepping errors. The results show that, for both variables, the minimum error as well as the optimal averaging window size decreases as the timestep is reduced. As the averaging window increases, the amplitudes of the error become almost identical to each other regardless of the timestep size. This result confirms that the averaged model is more accurate with a smaller timestep when the chosen averaging window is smaller or similar to Δt, and that the size of Δt does not affect the accuracy
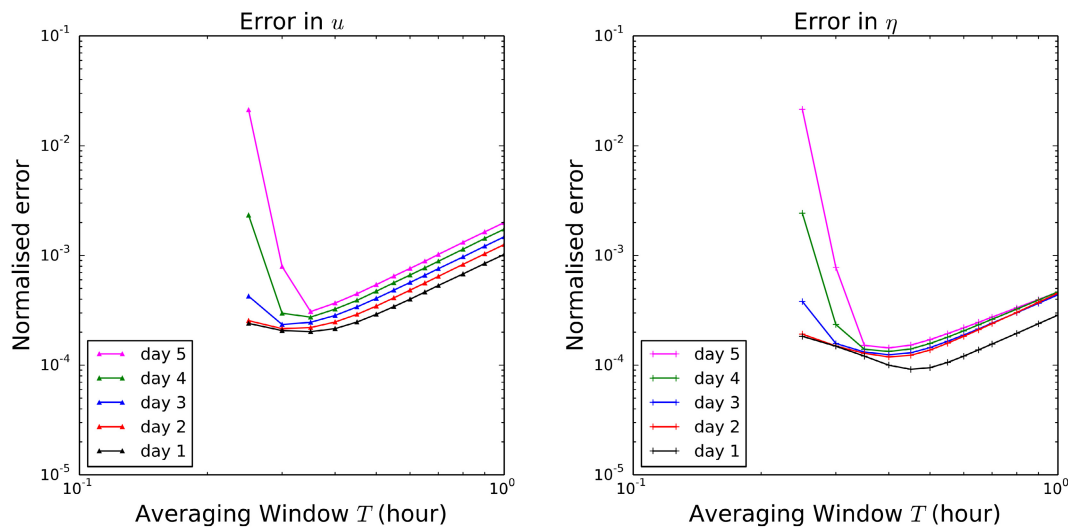

FIGURE 5 Time evolution of the errors up to day 5: (left) the $H(\text{div})$ norm of $\boldsymbol{u}$ and (right) the L_2 norm of η, normalised by the norms of the reference solution. The five curves in each panel show the corresponding errors at days 1, 2, 3, 4, and 5 from the bottom to the top, respectively. The timestep $\Delta t = 900$ s is fixed in all the simulations. [Colour figure can be viewed at wileyonlinelibrary.com]
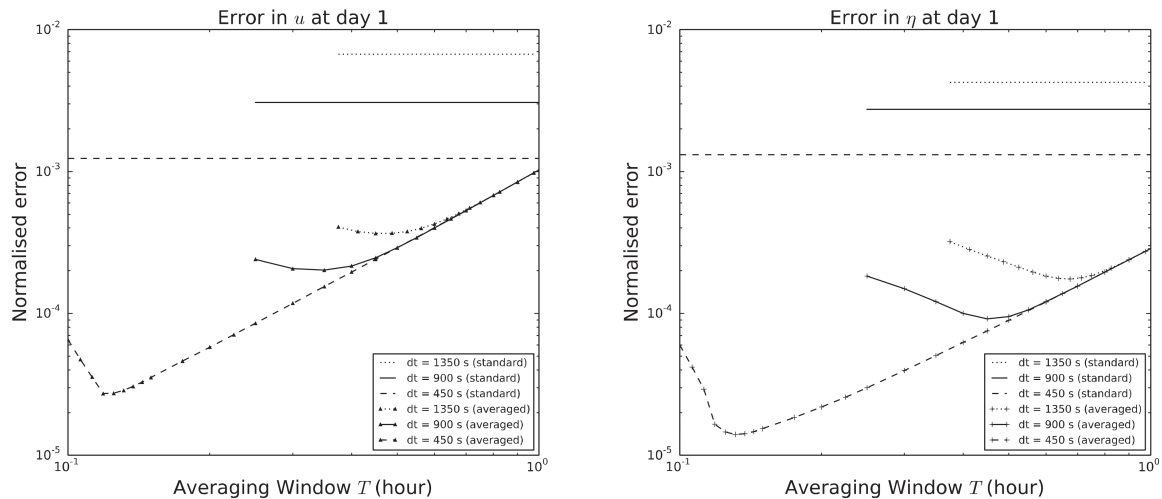
FIGURE 6 Normalised errors in the averaged model at day 1 using different timestep sizes: (left) the $H(\mathrm{div})$ norm of $\boldsymbol{u}$ and (right) the L_2 norm of η, normalised by the norms of the reference solution. Dashed, solid, and dotted curves show the errors from the averaged model using $\Delta t = 450$, 900, and 1350 s, respectively. Dashed, solid, and dotted straight lines show the errors from the standard semi-implicit model (Gibson *et al.*, 2019) using $\Delta t = 450$, 900, and 1350 s, respectively.

with larger averaging windows where the averaging error dominates.

Also shown in Figure 6 as lines without markers are the same errors at day 1 in the standard semi-implicit model (Gibson *et al.*, 2019), also normalised by the norms of the reference solution, when using the same three different timesteps: $\Delta t = 450$, 900, and 1350 s. As the standard model does not have averaging windows, the errors are shown as straight lines regardless of the size of the averaging window. It is clear that, for all three timesteps used in this test, the solution from the averaged model is more accurate than that from the standard model using the same timestep size, for the range of the averaging windows shown in Figure 6. In other words, the averaged model would allow us to use a larger timestep to achieve the same level of accuracy of the standard model when the averaging windows are chosen near the optimum.

4 | SUMMARY AND OUTLOOK

In this work, we presented a phase-averaging framework for the rotating shallow-water equations and a time-integration methodology for it. The new framework includes overcoming two key technical challenges for finite-element methods on the sphere: the development of a stable numerical matrix exponential used for the mappings and a parallel phase-averaging procedure. We presented proof-of-concept results from the rotating shallow-water equations and analysed their errors, which confirm that there is an optimal averaging window value for a given timestep size Δt. Naturally, the

optimal averaging window for both height and velocity fields combined depends on the choice of norm. Critically, we observe that the combined time discretisation and averaging error for the averaged model is much smaller than the time discretisation error in a semi-implicit method applied to the same semidiscretisation, illustrating the benefits of the approach. This is a very significant result, because it suggests that phase averaging could in itself be used as a time-parallel algorithm for the rotating shallow-water equations on the sphere (and perhaps three-dimensional models), without necessarily needing corrections through the Parareal algorithm, as proposed in Haut and Wingate (2014).

In future work, we will explore the combination of phase-averaging methods with implicit or split timestep methods that allow us to take even larger timesteps, we will incorporate parallel rational approximation techniques to speed up the exponential evaluations (Haut *et al.*, 2016), and we will undertake parallel performance benchmarks.

AUTHOR CONTRIBUTIONS

Hiroe Yamazaki: investigation; methodology; software; writing – original draft; writing – review and editing. **Colin J. Cotter:** conceptualization; funding acquisition; investigation; methodology; project administration; software; validation; writing – original draft; writing – review and editing. **Beth A. Wingate:** conceptualization; funding acquisition; investigation; methodology; software; supervision; writing – original draft; writing – review and editing.

ORCID

Colin J. Cotter https://orcid.org/0000-0001-7962-8324

REFERENCES

Ariel, G., Kim, S.J. and Tsai, R. (2016) Parareal methods for highly oscillatory dynamical systems. *SIAM Journal on Scientific Computing*, 38(6), A3540–A3564.

Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P. and Buschelman, K. (2021) PETSc users manual: revision 3.15.

Bauer, W., Cotter, C. and Wingate, B. (2022). Higher order phase averaging for highly oscillatory systems. *Multiscale Modeling & Simulation*, 20(3), 936–956.

Cotter, C.J. and Yamazaki, H. (2023). colinjcotter/krylov_exp: YamazakiCotterWingate2023 (Version YamazakiCotterWingate2023) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.8086143

Croci, M. and Muñoz-Matute, J. (2022) Exploiting Kronecker structure in exponential integrators: fast approximation of the action of φ-functions of matrices via quadrature. arXiv preprint arXiv:221100696.

Firedrake-Zenodo. (2023). Software used in 'Time parallel integration and phase averaging for the nonlinear shallow water equations on the sphere' (Version Firedrake_20230627.0) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.8086698

Gander, M.J. and Güttel, S. (2013) PARAEXP: a parallel integrator for linear initial-value problems. *SIAM Journal on Scientific Computing*, 35(2), C123–C142.

Gibson, T.H., McRae, A.T., Cotter, C.J., Mitchell, L. and Ham, D.A. (2019) *Compatible Finite Element Methods for Geophysical Flows: Automation and Implementation Using Firedrake*. Cham: Springer Nature.

Haut, T. and Wingate, B. (2014) An asymptotic parallel-in-time method for highly oscillatory PDEs. *SIAM Journal on Scientific Computing*, 36(2), A693–A713.

Haut, T.S., Babb, T., Martinsson, P. and Wingate, B. (2016) 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(2), 688–716.

Hochbruck, M. and Ostermann, A. (2010) Exponential integrators. *Acta Numerica*, 19, 209–286.

Kafiabad, H.A., Vanneste, J. and Young, W.R. (2021) Wave-averaged balance: a simple example. *Journal of Fluid Mechanics*, 911, R1.

Klainerman, S. and Majda, A.J. (1981) Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communications in Pure and Applied Mathematics*, 34(4), 481–524.

Majda, A.J. and Embid, P. (1998) Averaging over fast gravity waves for geophysical flows with unbalanced initial data. *Theoretical and Computational Fluid Dynamics*, 11(3), 155–169.

Minion, M. (2011) A hybrid parareal spectral deferred corrections method. *Communications in Applied Mathematics and Computational Science*, 5(2), 265–301.

Ong, B.W., Haynes, R.D. and Ladd, K. (2016) Algorithm 965: RIDC methods: a family of parallel time integrators. *ACM Transactions on Mathematical Software (TOMS)*, 43(1), 1–13.

Peddle, A., Haut, T. and Wingate, B. (2019) Parareal convergence for oscillatory PDEs with finite time-scale separation. *SIAM Journal on Scientific Computing*, 41, A3476–A3497.

Pieper, K., Sockwell, K.C. and Gunzburger, M. (2019) Exponential time differencing for mimetic multilayer ocean models. *Journal of Computational Physics*, 398, 108900.

Rathgeber, F., Ham, D.A., Mitchell, L., Lange, M., Luporini, F., McRae, A.T., Bercea, G.-T., Markall, G.R. and Kelly, P.H.J. (2016) Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3), 1–27.

Sanders, J.A., Verhulst, F. and Murdock, J. (2007) *Averaging Methods in Nonlinear Dynamical Systems*, 2nd edition. New York, NY: Springer.

Schreiber, M., Peixoto, P.S., Haut, T. and Wingate, B. (2018) Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. *The International Journal of High Performance Computing Applications*, 32(6), 913–933.

Shipton, J., Gibson, T.H. and Cotter, C.J. (2018) Higher-order compatible finite element schemes for the nonlinear rotating shallow water equations on the sphere. *Journal of Computational Physics*, 375, 1121–1137.

Thuburn, J., Cotter, C. and Dubos, T. (2014) A mimetic, semi-implicit, forward-in-time, finite volume shallow water model: comparison of hexagonal–icosahedral and cubed-sphere grids. *Geoscientific Model Development*, 7(3), 909–929.

Trefethen, L.N. (2019) *Approximation Theory and Approximation Practice*, Extended edition. Philadelphia, PA: SIAM.

Wagner, G.L. and Young, W.R. (2015) Available potential vorticity and wave-averaged quasi-geostrophic flow. *Journal of Fluid Mechanics*, 785, 401–424.

Williamson, D.L., Drake, J.B., Hack, J.J., Jakob, R. and Swarztrauber, P.N. (1992) A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, 102(1), 211–224.