

A finite-volume module for simulating global all-scale atmospheric flows

Piotr K. Smolarkiewicz^a, Willem Deconinck^a, Mats Hamrud^a,
Christian Kühnlein^a, George Mozdzynski^a, Joanna Szmelter^b,
Nils P. Wedi^a

^a*European Centre For Medium-Range Weather Forecasts, Reading, RG2 9AX, UK*

^b*Loughborough University, Leicestershire LE11 3TU, UK*

Abstract

The paper documents the development of a global nonhydrostatic finite-volume module designed to enhance an established spectral-transform based numerical weather prediction (NWP) model. The module adheres to NWP standards, with formulation of the governing equations based on the classical meteorological latitude-longitude spherical framework. In the horizontal, a bespoke unstructured mesh with finite-volumes built about the reduced Gaussian grid of the existing NWP model circumvents the notorious stiffness in the polar regions of the spherical framework. All dependent variables are co-located, accommodating both spectral-transform and grid-point solutions at the same physical locations. In the vertical, a uniform finite-difference discretisation facilitates the solution of intricate elliptic problems in thin spherical shells, while the pliancy of the physical vertical coordinate is delegated to generalised continuous transformations between computational and physical space. The newly developed module assumes the compressible Euler equations as default, but includes reduced soundproof PDEs as an option. Furthermore, it employs semi-implicit forward-in-time integrators of the governing PDE systems, akin to but more general than those used in the NWP model. The module shares the equal regions parallelisation scheme with the NWP model, with multiple layers of parallelism hybridising MPI tasks and OpenMP threads. The efficacy of the developed nonhydrostatic module is illustrated with benchmarks of idealised global weather.

Key words: atmospheric models, hierarchical modelling, non-oscillatory forward-in-time schemes, gravity waves, numerical weather prediction

PACS:

* Corresponding Author.

Email address: `smolar@ecmwf.int` (Piotr K. Smolarkiewicz).

1 INTRODUCTION

Numerical weather prediction (NWP) has achieved high proficiency over the past 30 years. This owes much to advancements in computer hardware, observational networks and data assimilation techniques as well as numerical methods for integrating hydrostatic primitive equations (HPE). One particular numerical approach embraced widely by NWP combines semi-implicit time stepping with semi-Lagrangian advection (SISL) and with spectral-transform spatial discretisation of the governing HPE [46]. The SISL time stepping enables integrations with Courant numbers of the fluid flow and wave motions much larger than unity, whereas the spectral-transform discretisation facilitates the efficient solution of elliptic equations induced by the SISL approach. Moreover, it circumvents the computational expense of the latitude-longitude (lat-lon) coordinate framework — where meridians converge towards the poles — as spectral transforms can operate on a reduced Gaussian grid with quasi-uniform distribution of nodes on the surface of a sphere [44,45].

The advance of massively parallel computing in the nineteen-nineties and beyond encouraged finer grid intervals in NWP models. This has improved resolution of weather systems and enhanced the accuracy of forecasts, while stimulating development of global nonhydrostatic models.¹ In effect, many operational NWP models nowadays include nonhydrostatic options either for regional predictions or research [24,4,25,1,43,23,47]. However, to date no NWP model runs globally in operations at nonhydrostatic resolutions. Such high resolutions are still computationally unaffordable and too inefficient to meet demands of the limited time window for distributing global forecasts to the end users.

The computational affordability of global nonhydrostatic forecasts, and thus their future, is a twofold concern. On one hand, with communications overwhelming the efficacy of spectral transforms on large grids [44,45], emerging supercomputers with simply scaled up numbers of cores become unaffordable, not least due to the excessive energy bill. On the other hand, the necessity of replacing HPE that have been central to the success of weather prediction and climate projections exacerbates the efficiency problem. In particular, with the simulated vertical extent of the atmosphere relatively thin compared to its horizontal extent, the vertically propagating sound waves supported by the nonhydrostatic Euler equations from which HPE derive, impose severe restrictions on the numerical algorithms. The hydrostatic balance assumption underlying HPE conveniently filters vertically propagating sound waves, therefore permitting large time steps in the numerical integration. Moreover, HPE

¹ The progressive refinement of NWP grids invalidates HPE as horizontal resolution approaches $\mathcal{O}(10^3)$ m.

imply the separability of horizontal and vertical discretisation, thus facilitating the design of effective SISL flow solvers.

Altogether, NWP is at a crossroads. Although massively parallel computer technology promises continued advances in forecast quality, the latter cannot be achieved with the existing apparatus of NWP models by straightforwardly applying it to larger grids. This is well appreciated, and the community vigorously seeks new avenues to advance the status quo of NWP to meet growing demands and expectations of the society. Among a broad range of activities pursued worldwide (see Introduction in [35] for a succinct overview), trends emerge to start afresh and supersede the vested legacy codes with modern developments. Recognising the predictive skills of the existing NWP legacy codes, we seek to mitigate their shortcomings by supplying a complementary nonhydrostatic module with capabilities of cloud-resolving models. The first step towards this paradigm is the development of a global finite-volume nonhydrostatic module capable of working on the reduced Gaussian grid of the Integrated Forecast System (IFS) of the ECMWF [3,40,44].

The IFS is among the world leading global medium-range NWP models, with spectral-transform based spatial discretisation and SISL integrator for the governing PDEs. At present, IFS runs operationally with HPE, at the spectral resolution equivalent to uniform coverage of the Earth surface with a $\lesssim 16$ km grid spacing. In the vertical, the atmosphere is discretised using a hybrid terrain-following pressure (viz. mass) coordinate resolving a ≈ 85 km deep atmosphere with 137 levels, and equivalent height increments increasing (non-uniformly) from about 10 m near the ground to a few kilometres at the very top. The uniform time step 600 s used in operations results in local Courant numbers $\approx 4 - 5$ in jet stream regions and, in particular, in the stratosphere where flow velocities approach 100 m s^{-1} . The efficiency standard is rigid for the IFS: in order to be useful, a 10 day global forecast must be produced in no more than 1 hour of wall-clock time. With 10 prognosed dependent variables in 3D, this puts heavy demand on the efficiency of numerical schemes and their scalability on massively-parallel computers. Furthermore, spatial differentiation is conducted in spectral space, and there are no practical means of calculating derivatives locally in physical space. Moreover, the increased complexity of the nonhydrostatic IFS option significantly increases the number of required spectral transforms, nearly doubling the cost compared to the hydrostatic forecast, which is already projected to be too slow on existing supercomputers at nonhydrostatic scales within the operational time window of 1 hour [44].

The concept of supplementing NWP models with complementary nonhydrostatic add-on modules has been considered earlier [11]. However the approach proposed in our paper is distinct. Rather than adding optional contributions from nonhydrostatic perturbation terms to the governing HPEs, we supply an

autonomous nonhydrostatic module that can operate on the nodes of the IFS grid and, in principle, on any horizontal grid. In general terms, such an approach enables numerical procedures unavailable in SISL spectral models with minimal disruption to their highly optimised codes. In particular, it can replace global communication and computation with local equivalents, providing an effective testing ground for assessing the utility of innovations from outside the realm of spectral methods in the context of real weather. Furthermore, the autonomy of the module enables a variety of implementations, from as simple as advecting selected critical fields of SISL NWP models in a conservative finite-volume fashion, to as complex as replacing the whole dynamical core with a cloud-resolving model while driving the latter with a smoother coarse solution of a spectral transform HPE code. Herein, we document the development of the autonomous hybrid all-scale finite-volume module (hereafter FVM, for brevity).

The FVM is a hybrid descendant of the interdisciplinary all-scale Eulerian/semi-Lagrangian research model EULAG [22,33,35,14] and the generalisation of the non-oscillatory forward-in-time (NFT) atmospheric models to unstructured meshes [28,37,34]. The default governing PDEs of the FVM assume the fully compressible Euler equations under gravity on a rotating sphere, and reduced soundproof PDEs are included as an option. The standard lat-lon spherical framework (sometimes referred to as anholonomic; section 7.2 in [7]) forms the basis of the analytic model formulation, similar in IFS and EULAG [42]. Circumventing the notorious stiffness of the global lat-lon system is delegated to the flexibility of the finite-volume discretisation on an unstructured mesh, with control volumes built around the reduced Gaussian grid of the IFS. In 3D, the mesh is prismatic; i.e., the horizontal discretisation is common for all vertical levels of the computational space [37]. A uniform finite-difference discretisation in the vertical is chosen to facilitate the solution of intricate elliptic problems in thin spherical shells. Continuous curvilinear mappings are employed to mimic a pliant height coordinate in physical space [21,41,33]. The FVM employs established semi-implicit non-oscillatory forward-in-time integrators of the governing PDE systems [35] conceptually similar to, but more general than, those used in the IFS. Because FVM operates at the nodes of the IFS grid, it seamlessly inherits the equal regions domain decomposition parallelisation scheme of the IFS, with multiple layers of parallelism hybridising MPI tasks and OpenMP threads; [18,19] and references therein.

The conceptual building blocks of the FVM were established in the earlier developments of EULAG and the akin generalisations to unstructured meshes, and they are well documented in the literature. Nonetheless, some technical aspects had to be customised for the parallel implementation on the unstructured mesh constrained by the IFS grid. In particular, the edge-based formulation pursued in the earlier works [37,38,32,34] had to be supplemented with elements of a nodal formulation to facilitate massive parallelism. In this paper

we will only briefly review the building blocks, thoroughly discussed in earlier publications, and focus on new elements. For simplicity and conciseness, the presentation is focused on the formulation of the FVM dynamical core; i.e., it is restricted to dry motions and accounts only for idealised heat sources/sinks and momentum dissipation.² For the numerical simulation demonstrating nonhydrostatic capabilities, the quasi-two-dimensional orographic flow with linear vertical shear is adopted after [42]. For the numerical simulation epitomising midlatitude weather systems, the global baroclinic instability benchmark [9] is adapted after [35].

The paper is organised as follows. In section 2 we review the nonhydrostatic governing equations available in FVM. These equations were discussed extensively in [35], passing gradually from physically intuitive Lagrangian form in Cartesian geometry toward the conservation laws consistent with the problem geometry and the solution procedure. Consequently, here we go directly to the latter form, to devote more attention to the technical aspects of the numerical procedures in sections 3 and 4. In particular, section 3 summarizes the outer layer of the FVM numerical apparatus, comprising building blocks of the semi-implicit integration schemes such as NFT template algorithms, semi-implicit time stepping, and the associated elliptic boundary value problems. Section 4 outlines the inner layer of the FVM machinery, including a description of the spatial discretisation, the datastructure underlying the FVM code, and of the hybrid MPI/OpenMP parallelisation scheme. Section 6 substantiates preceding technical developments, demonstrating the efficacy of the FVM module for two idealised flow problems relevant to weather. Section 7 concludes the paper.

2 GOVERNING EQUATIONS

The analytic PDEs solved in the dynamical core of FVM can be written in a compact symbolic form, consistent with the FVM code, as

$$\frac{\partial \mathcal{G}_\varrho}{\partial t} + \nabla \cdot (\mathcal{G}_\varrho \mathbf{v}) = 0 , \quad (1a)$$

$$\frac{\partial \mathcal{G}_\varrho \theta'}{\partial t} + \nabla \cdot (\mathcal{G}_\varrho \mathbf{v} \theta') = -\mathcal{G}_\varrho \left(\widetilde{\mathbf{G}}^T \mathbf{u} \cdot \nabla \theta_a - \mathcal{H} \right) , \quad (1b)$$

² Technically, physical parametrisations of the IFS can be readily used with the FVM, because they are computed in vertical columns at each node of the reduced Gaussian grid.

$$\begin{aligned} \frac{\partial \mathcal{G} \varrho \mathbf{u}}{\partial t} + \nabla \cdot (\mathcal{G} \varrho \mathbf{v} \otimes \mathbf{u}) = & \quad (1c) \\ -\mathcal{G} \varrho \left(\Theta \tilde{\mathbf{G}} \nabla \varphi + \mathbf{g} \Upsilon_B \frac{\theta'}{\theta_b} + \mathbf{f} \times (\mathbf{u} - \Upsilon_C \mathbf{u}_a) - \mathcal{M}'(\mathbf{u}, \mathbf{u}_a, \Upsilon_C) - \mathbf{D} \right). \end{aligned}$$

The system (1) encapsulates three distinct sets of the governing equations: the compressible Euler equations under gravity on a rotating sphere, and their two reduced soundproof forms, the pseudo-incompressible equations of Durran [6] and the anelastic equations of Lipps-Hemler [16,17]. The distinction between the three sets is encrypted in definitions of the generalised density ϱ and pressure variable φ , together with the corresponding dimensionless coefficients Θ , Υ_B , and Υ_C that depend on various states of the potential temperature θ . For ideal gas, θ amounts to specific entropy via $ds = c_p d \ln \theta$, with c_p denoting the specific heat at constant pressure. For the [*compressible, pseudo-incompressible, anelastic*] PDEs, ϱ and φ are defined respectively as

$$\varrho := [\rho(\mathbf{x}, t), \frac{\rho_b(z)\theta_b(z)}{\theta(\mathbf{x}, t)}, \rho_b(z)], \quad \varphi := [c_p \theta_0 \pi', c_p \theta_0 \pi', c_p \theta_b \pi'], \quad (2)$$

where ρ marks air density, while subscripts b , 0 and a refer, respectively, to a horizontally homogeneous and hydrostatically balanced base state, a constant reference value and the ambient state discussed later.³ The generalised pressure φ depends on the Exner pressure $\pi \equiv (p/p_0)^{R_d/c_p}$, where R_d is the gas constant for dry air. Throughout the paper primes denote perturbations with respect to the ambient state; e.g., $\pi' = \pi - \pi_a$, $\theta' = \theta - \theta_a$, and so forth. The coefficients Θ , Υ_B , and Υ_C corresponding to generalised variables in (2) are

$$\Theta := \left[\frac{\theta}{\theta_0}, \frac{\theta}{\theta_0}, 1 \right], \quad \Upsilon_B := \left[\frac{\theta_b(z)}{\theta_a(\mathbf{x})}, \frac{\theta_b(z)}{\theta_a(\mathbf{x})}, 1 \right], \quad \Upsilon_C := \left[\frac{\theta}{\theta_a(\mathbf{x})}, \frac{\theta}{\theta_a(\mathbf{x})}, 1 \right]. \quad (3)$$

Noteworthy, the soundproof equations contained in (1) do not necessitate the provision of constitutive laws for their solution, because their respective pressure perturbations are determined from the elliptic equations that follow from constraining the velocity solutions to satisfy mass continuity. In other words, their constitutive laws were analytically accounted for while deriving the reduced equations, and afterwards are not required for the solution of the dynamical core. This is not the case with fully compressible equations where the ideal gas law

$$\varphi = c_p \theta_0 \left[\left(\frac{R_d}{p_0} \varrho \theta \right)^{R_d/c_p} - \pi_a \right]. \quad (4)$$

³ For adiabatic dynamics $\mathcal{H} \equiv 0$ in (1b), and the pseudo-incompressible density $\rho_b \theta_b / \theta$ can be replaced with $\rho_b \theta_b / \theta_0$, in the spirit of the anelastic PDEs [32,35].

explicitly relates the thermodynamic pressure perturbations to the distribution of temperature and mass in the fluid.

For the three sets of PDEs encapsulated in (1), the mathematical symbolism of (1) already accounts for the generalised time-dependent curvilinear coordinates enabling the representation of orography or dynamic mesh adaptivity [21,41,13]. In particular, (\mathbf{x}, t) refers already to the coordinates of the generalised time-dependent frame, and $\mathcal{G}(\mathbf{x}, t)$ denotes the Jacobian — so, \mathcal{G}^2 is the determinant of the metric tensor that defines the fundamental metric in a space of interest where the problem is solved [21]. Furthermore, $\nabla \cdot (\dots)$ denotes the scalar product of spatial partial derivatives with a vector, so the total derivative underlying conservation form (1), $d/dt = \partial/\partial t + \mathbf{v} \cdot \nabla$, takes the velocity $\mathbf{v} = \dot{\mathbf{x}}$ not necessarily equal to the physical velocity \mathbf{u} for which equations are solved. The $\widetilde{\mathbf{G}}\nabla\varphi$ in the momentum equation symbolises the product of a known matrix of metric coefficients and the vector of partial derivatives, whereas $\widetilde{\mathbf{G}}^T\mathbf{u} = \mathbf{v} - \mathbf{v}^g$ on the rhs of the entropy equation accounts for the mesh velocity \mathbf{v}^g , set to zero in the remainder of this paper. For the readers convenience, various metric coefficients are exemplified for the spherical frame in the appendix A.

The remaining symbols are specific to applications addressed in the paper. In the entropy equation (1b), \mathcal{H} symbolises a heat source/sink. In the momentum equation (1c), the Coriolis parameter is given as $\mathbf{f} \equiv 2\boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ denotes a constant angular velocity of the rotating sphere. The gravitational acceleration $\mathbf{g} = (0, 0, -g)$ enters the buoyancy term on the rhs of (1c). The term $\mathcal{M}'(\mathbf{u}, \mathbf{u}_a, \Upsilon_C) = \mathcal{M}(\mathbf{u}) - \Upsilon_C\mathcal{M}(\mathbf{u}_a)$ symbolises metric forcings in the spherical domain (cf. appendix A), whereas \mathcal{D} denotes a momentum sink.

To facilitate the presentation of the numerical solution procedures, we introduce an auxiliary ambient state $(\mathbf{u}_a, \phi_a, \theta_a)$ assumed to be a known particular solution of the governing PDEs, typically different for each set [35]. The primary role of ambient states is to simplify the design of the initial and boundary conditions as well as to enhance the accuracy of calculations in finite-precision arithmetic. Generally, ambient states can be time-dependent; e.g., prescribing oceanic tidal motions [39]. In this paper, only stationary ambient states are considered, e.g., geostrophically balanced large-scale flows

$$0 = -c_p\theta_a\widetilde{\mathbf{G}}\nabla(\pi_a - \pi_b) - \mathbf{g}\frac{\theta_a - \theta_b}{\theta_b} - \mathbf{f} \times \mathbf{u}_a + \mathcal{M}(\mathbf{u}_a) , \quad (5)$$

for either compressible or pseudo-incompressible momentum equations, together with identically satisfied ambient mass continuity and entropy equations; cf. [35] for a discussion.

3 NUMERICAL APPROXIMATIONS

3.1 Non-oscillatory forward-in-time template

All prognostic equations comprising symbolic system (1) can be written in a compact form of the generalised transport equation for an arbitrary scalar variable Ψ ,

$$\frac{\partial G\Psi}{\partial t} + \nabla \cdot (\mathbf{V}\Psi) = G\mathcal{R} , \quad (6)$$

in which vector field \mathbf{V} as well as scalar fields G and \mathcal{R} are assumed to be known functions of time and space. Depending on the definitions of G , Ψ , \mathbf{V} and \mathcal{R} , (6) expresses either compressible or soundproof PDEs of atmospheric dynamics; see [35] for a comprehensive discussion. For the entropy and the momentum equations in (1), Ψ represents, respectively, the potential temperature perturbations θ' and components of the physical velocity vector \mathbf{u} ; whereas the density ρ is absorbed in $G \equiv \mathcal{G}\rho$. Then \mathcal{R} represents the corresponding parenthetical terms on the rhs in (1), and $\mathbf{V} \equiv G\mathbf{v} = \mathcal{G}\rho\mathbf{v}$. For the mass continuity equation $\mathcal{R} \equiv 0$, but the interpretation of Ψ and \mathbf{V} depends on the system addressed. For soundproof PDEs, the density ρ is prescribed, and (1a) implies the transportive momenta $\mathbf{V} = \mathcal{G}\rho\mathbf{v}$. For the compressible system, $\Psi \equiv \rho$, $G \equiv \mathcal{G}$, so $\mathbf{V} = \mathcal{G}\mathbf{v}$ in the continuity equation. Consequently, the cumulative mass flux $\mathbf{V}\Psi = \mathcal{G}\rho\mathbf{v}$ in the prognostic mass continuity equation amounts to transportive momenta in the entropy and the momentum equations. Formulating numerical integrators of (1) according to the procedures adopted for (6) assures the compatibility of conservative advection of θ' and components of \mathbf{u} with the mass continuity [35].

A key building block for semi-implicit integrators of the PDE systems encapsulated in (1) is a second-order-accurate non-oscillatory forward-in-time (NFT) template algorithm for (6)

$$\Psi_{\mathbf{i}}^{n+1} = \mathcal{A}_{\mathbf{i}} \left(\tilde{\Psi}^n, \mathbf{V}^{n+1/2}, G^n, G^{n+1} \right) + 0.5\delta t \mathcal{R}_{\mathbf{i}}^{n+1} , \quad \tilde{\Psi}^n \equiv \Psi^n + 0.5\delta t \mathcal{R}^n . \quad (7)$$

Here, \mathcal{A} is a shorthand for the NFT advection transport operator MPDATA (for multidimensional positive definite advection transport algorithm) [35]. Furthermore, the index \mathbf{i} symbolises position on the computational grid, δt is a time interval between two consecutive time levels denoted by n and $n+1$, and the ‘‘advecting’’ $\mathbf{V}^{n+1/2}$ is an $\mathcal{O}(\delta t^2)$ estimate of \mathbf{V} at the intermediate time level. The template (7) is congruent with the trapezoidal-rule trajectory integral of the ODE underlying (6) [27,31]. Its respective Euler-forward,

$$\Psi_{\mathbf{i}}^{n+1} = \mathcal{A}_{\mathbf{i}} \left(\Psi^n + \delta t \mathcal{R}^n, \mathbf{V}^{n+1/2}, G^n, G^{n+1} \right) , \quad (8)$$

and Euler-backward,

$$\Psi_{\mathbf{i}}^{n+1} = \mathcal{A}_{\mathbf{i}} \left(\Psi^n, \mathbf{V}^{n+1/2}, G^n, G^{n+1} \right) + \delta t \mathcal{R}_{\mathbf{i}}^{n+1}, \quad (9)$$

forms as well as their combinations for selected counterparts of \mathcal{R} will be also employed in the subsequent discussions.

3.2 Semi-implicit integrators

In the system (1), only the mass continuity equation (1a) is homogeneous, whereas the entropy and momentum equations have non-vanishing right-hand-sides, dependent on the prognosed model variables. In consequence, the entire model algorithm can be reduced to two conceptually distinct steps.

The first step provides the advectors for the first term on the rhs of (7). For soundproof PDEs, $\mathbf{V}^{n+1/2} = (\mathcal{G}\rho\mathbf{v})^{n+1/2}$ is typically evaluated by linear extrapolation from t^{n-1} and t^n and used consistently for all prognostic variables. For compressible PDEs the linearly extrapolated $\mathbf{V}^{n+1/2} = (\mathcal{G}\mathbf{v})^{n+1/2}$ is used only in the density advection,

$$\varrho_{\mathbf{i}}^{n+1} = \mathcal{A}_{\mathbf{i}} \left(\varrho^n, (\mathcal{G}\mathbf{v})^{n+1/2}, \mathcal{G}^n, \mathcal{G}^{n+1} \right) \implies \mathbf{V}^{n+1/2} = \overline{(\mathcal{G}\rho\mathbf{v})}^{n+1/2}, \quad (10)$$

while updating the density and concomitantly evaluating the advectors as cumulative directional mass fluxes for advection of θ' and all components of \mathbf{u} ; see [35] for an exposition.

The second step integrates the entropy and momentum equations in (1). To account for the nonlinearity of the pressure gradient force and/or the metric forces on the rhs of the momentum equation the template algorithm (7) is executed iteratively, lagging nonlinear terms behind:

$$\begin{aligned} \theta'_{\mathbf{i}}{}^\nu &= \widehat{\theta}'_{\mathbf{i}} - 0.5\delta t \left(\widetilde{\mathbf{G}}^T \mathbf{u}^\nu \cdot \nabla \theta_a \right)_{\mathbf{i}} \\ \mathbf{u}_{\mathbf{i}}{}^\nu &= \widehat{\mathbf{u}}_{\mathbf{i}} - 0.5\delta t \left(\Theta^{\nu-1} \widetilde{\mathbf{G}} \nabla \varphi^\nu + \mathbf{g} \Upsilon_B \frac{\theta'^\nu}{\theta_b} \right)_{\mathbf{i}} \\ &\quad - 0.5\delta t \left(\mathbf{f} \times (\mathbf{u}^\nu - \Upsilon_C^{\nu-1} \mathbf{u}_a) - \mathcal{M}'(\mathbf{u}, \mathbf{u}_a, \Upsilon_C)^{\nu-1} \right)_{\mathbf{i}}. \end{aligned} \quad (11)$$

Here, $\widehat{\theta}'_{\mathbf{i}}$ and $\widehat{\mathbf{u}}_{\mathbf{i}}$ are the shorthands for the transport operator \mathcal{A} applied to $\widetilde{\theta}'$ and $\widetilde{\mathbf{u}}$ in (7), subsumed with the respective contributions from forcings \mathcal{H} and \mathcal{D} estimated to $\mathcal{O}(\delta t)$ at t^{n+1} , given as

$$\begin{aligned}\widehat{\theta}'_{\mathbf{i}} &= \mathcal{A}_{\mathbf{i}} \left(\widetilde{\theta}' + 0.5\delta t \mathcal{H}^{n+1}, \mathbf{V}^{n+1/2}, \varrho^{*n}, \varrho^{*n+1} \right), \\ \widehat{\mathbf{u}}_{\mathbf{i}} &= \mathcal{A}_{\mathbf{i}} \left(\widetilde{\mathbf{u}} + 0.5\delta t \mathcal{D}^{n+1}, \mathbf{V}^{n+1/2}, \varrho^{*n}, \varrho^{*n+1} \right),\end{aligned}\tag{12}$$

with $\mathbf{V}^{n+1/2}$ provided by the preceding step, and the effective densities, $\varrho^* := \mathcal{G}\varrho$, specified as $\varrho^{*n} = \mathcal{G}^n \varrho^n$ and $\varrho^{*n+1} = \mathcal{G}^{n+1} \varrho^{n+1}$. Furthermore,

$$\theta'_{\mathbf{i}} = \left(\widehat{\theta}' - 0.5\delta t \widetilde{\mathbf{G}}^T \mathbf{u}^\nu \cdot \nabla \theta_a + \theta_a \right)_{\mathbf{i}}.\tag{13}$$

Throughout (11)-(13), the index $\nu = 1, \dots, N_\nu$ numbers the iterations, with the first guess $\theta'_{\mathbf{i}} = \widehat{\theta}'_{\mathbf{i}}$ generated by advecting full θ as

$$\theta'_{\mathbf{i}} = \mathcal{A}_{\mathbf{i}} \left(\theta^n + \delta t \mathcal{H}^n, \mathbf{V}^{n+1/2}, \varrho^{*n}, \varrho^{*n+1} \right),\tag{14}$$

and $\mathbf{u}'_{\mathbf{i}}$ is obtained by linear extrapolation from t^{n-1} and t^n to t^{n+1} . With this design, the solution is fully second order accurate even for $N_\nu = 1$, and $N_\nu = 2$ gives already close approximation to the trapezoidal integral [31].

The scheme outlined in (11)-(13) contains fully implicit trapezoidal integrals of buoyancy and Coriolis terms; whereas metric forcings, and coefficients depending on full potential temperature are integrated explicitly. For soundproof PDEs pressure gradient terms are always integrated implicitly with the trapezoidal rule, whereas for the compressible Euler equations they can be integrated explicitly or implicitly leading, respectively, to the acoustic option of the solver in the spirit of gas dynamics for high speed flows [31], or the semi-implicit option suitable for large time step simulations of low Mach number flow. Regardless of the selected option, derivation of the closed-form expression for the velocity update is common to all three sets of PDEs. The potential temperature perturbation is substituted in the buoyancy term of the momentum equation with the rhs of the entropy scheme and all terms depending on \mathbf{u}^ν are gathered on the lhs of the momentum scheme, while dropping the spatial grid index \mathbf{i} everywhere, as all dependent variables, coefficients and terms are co-located in (11)-(13). This results in

$$\begin{aligned}\mathbf{u}^\nu + 0.5\delta t \mathbf{f} \times \mathbf{u}^\nu - (0.5\delta t)^2 \mathbf{g} \Upsilon_B \frac{1}{\theta_b} \widetilde{\mathbf{G}}^T \mathbf{u}^\nu \cdot \nabla \theta_a = \\ \widehat{\mathbf{u}} - 0.5\delta t \left(\mathbf{g} \Upsilon_B \frac{\widehat{\theta}'}{\theta_b} - \mathbf{f} \times \Upsilon_C^{\nu-1} \mathbf{u}_a - \mathcal{M}'(\mathbf{u}, \mathbf{u}_a, \Upsilon_C)^{\nu-1} \right) \\ - 0.5\delta t \Theta^{\nu-1} \widetilde{\mathbf{G}} \nabla \varphi^\nu \equiv \widehat{\mathbf{u}} - 0.5\delta t \Theta^{\nu-1} \widetilde{\mathbf{G}} \nabla \varphi^\nu,\end{aligned}\tag{15}$$

which symbolises a system of three linear algebraic equations with three unknown components of the velocity vector \mathbf{u}^ν at each point of the co-located grid. Viewing the lhs of (15) as a linear operator acting on the velocity vector,

$\mathbf{L}\mathbf{u}^\nu$, the closed-form expression for the velocity update may be symbolised as

$$\mathbf{u}^\nu = \check{\mathbf{u}} - \mathbf{C}\nabla\varphi^\nu, \quad (16)$$

where $\check{\mathbf{u}} = \mathbf{L}^{-1}\hat{\mathbf{u}}$ and $\mathbf{C} = \mathbf{L}^{-1}0.5\delta t\Theta^{\nu-1}\widetilde{\mathbf{G}}$ denotes a 3×3 matrix of known coefficients; cf. [21] and [32,34] for expanded expressions in tensorial and explicit component notations.

Completion of the solution requires provision of the pressure perturbation in each iteration ν of the velocity update in (16). For the acoustic option of the compressible solver, the thermodynamic pressure

$$\varphi_{\mathbf{i}}^\nu = c_p\theta_0 \left[\left(\frac{R_d}{p_0} \varrho^{n+1}\theta^{\nu-1} \right)^{R_d/c_v} - \pi_a \right]_{\mathbf{i}}, \quad (17)$$

is used in each iteration ν , and the total potential temperature (required in the coefficients Θ and Υ_C) gets updated according to (13). The potential temperature perturbation θ' is updated according to (11), upon completion of the velocity update for $\nu = N_\nu$. For the compressible and soundproof solvers with implicit integrals of the pressure gradient terms, the thermodynamic pressure evaluation from (17) is superseded with the solutions of the elliptic pressure equations discussed next.

3.3 Elliptic pressure equations

Computational stability of the acoustic solver critically depends on the speed of sound; therefore, this is not a practical option for NWP. However, the availability of the acoustic solver is important because: a) it provides a reference solution for verification of large time step results [35,14]; b) it provides a suitable approach for simulation of extreme events like explosions in natural environments or volcanic eruptions [31]; and c) it guides the development of large time step schemes [35]. In the latter, the evolutionary form of the equation of state, derived by taking d/dt of (4), couples all thermodynamic variables and the flow field (through $d\varrho/dt$ and the mass continuity equation), thus providing a constraint for pressure and velocity in (16). Taking $d/dt(4)$, expressing the result in the conservation law form, and manipulating the terms [35,14] leads to the PDE

$$\frac{\partial \varrho^* \varphi}{\partial t} + \nabla \cdot (\varrho^* \mathbf{v} \varphi) = \varrho^* R^\varphi \quad (18)$$

with the rhs forcing

$$R^\varphi = -\xi\phi \frac{1}{\mathcal{G}} \nabla \cdot (\mathcal{G}\mathbf{v}) - \frac{1}{\varrho^*} \nabla \cdot (\varrho^* \mathbf{v} \phi_a) + \phi_a \frac{1}{\varrho^*} \nabla \cdot (\varrho^* \mathbf{v}) + \xi\phi\Pi, \quad (19)$$

where $\phi_a \equiv c_p \theta_0 \pi_a$, $\phi \equiv \varphi + \phi_a$, $\xi \equiv R_d / (c_p - R_d)$, and the explicit forcing Π due to heat sources/sinks in the entropy equation is $\Pi \equiv \mathcal{H} / \theta$. The PDE (18) extends the equation (56) in [35] to account for the heat forcing \mathcal{H} , according to the derivation detailed in Appendix A of [14]. Interpreting (18) as an archetype PDE (6), and integrating it with a mix of the first-order NFT templates (8)-(9)

$$\varphi_i^{n+1} = \mathcal{A}_i(\tilde{\varphi}, \mathbf{V}^{n+1/2}, \varrho^{*n}, \varrho^{*n+1}) + \delta t \tilde{R}^\varphi|_i^{n+1} \equiv \hat{\varphi} + \delta t \tilde{R}^\varphi|_i^{n+1}, \quad (20)$$

where $\tilde{\varphi} = (\varphi + \delta t \xi \phi \Pi)^n$ and $\tilde{R}^\varphi \equiv R^\varphi - \xi \phi \Pi$, provides a discrete implicit constraint for (16)

$$0 = -\frac{1}{\mathcal{G}} \nabla \cdot (\mathcal{G} \mathbf{v}^\nu) - \frac{1}{\xi} \frac{\phi_a}{\phi^{\nu-1}} \left(\frac{1}{\varrho^* \phi_a} \nabla \cdot (\varrho^* \phi_a \mathbf{v}^\nu) - \frac{1}{\varrho^*} \nabla \cdot (\varrho^* \mathbf{v}^\nu) \right) - \frac{1}{\delta t \xi \phi^{\nu-1}} (\varphi^\nu - \hat{\varphi}). \quad (21)$$

Recalling from section 2 that in stationary coordinates $\mathbf{v} = \tilde{\mathbf{G}}^T \mathbf{u}$, (16) implies

$$\mathbf{v}^\nu = \check{\mathbf{v}} - \tilde{\mathbf{G}}^T \mathbf{C} \nabla \varphi^\nu, \quad (22)$$

and when inserted in (21) leads to an elaborate Helmholtz problem for pressure (referred to in [35] as of the second-kind) composed of the three soundproof like Poisson operators

$$\frac{1}{\zeta} \nabla \cdot \zeta (\check{\mathbf{v}} - \tilde{\mathbf{G}}^T \mathbf{C} \nabla \varphi), \quad (23)$$

where ζ denotes a generalised density.

The Helmholtz problem (21) was widely discussed in [35]. A significant new element of (21) is that it now accounts for diabatic effects, via modification of the $\tilde{\varphi}$ argument of the transport operator \mathcal{A} in (20) and, thus, the modification of the explicit counterpart $\hat{\varphi}$ in the Helmholtz term in (21). Noteworthy, it can be shown that neglecting the $\partial \varphi / \partial t$ on the lhs of (18) leads to the Poisson problem

$$\nabla \cdot \varrho^* \theta (\check{\mathbf{v}} - \tilde{\mathbf{G}}^T \mathbf{C} \nabla \varphi) = \varrho^* \mathcal{H}, \quad (24)$$

which upon approximating the density ϱ^* with $\mathcal{G} \rho_b \theta_b / \theta$, becomes the elliptic constraint of the pseudo-incompressible system of Durran [6]; recall (2) and footnote 3 in section 2. Conversely, neglecting $\partial \varrho / \partial t$ while taking $d/dt(4)$, reveals the residual relation

$$\nabla \cdot \varrho^* (\check{\mathbf{v}} - \tilde{\mathbf{G}}^T \mathbf{C} \nabla \varphi) = 0, \quad (25)$$

which upon the linearisation $q^* \rightarrow \mathcal{G}\rho_b$ amounts to the elliptic constraint of the anelastic system [16]. These observations are not meant to replace rigorous derivations of the reduced soundproof systems, but merely to point out connections of the Helmholtz problem (21) to the Poisson problems for the soundproof PDEs. In the FVM code, the customised Krylov-subspace solver⁴ does not rely on such connections but builds the linear operator associated with each PDE by combining/modifying accordingly the elementary Poisson (23) operators and their preconditioners.

4 DISCRETE APPARATUS

4.1 Spatial discretisation

The mathematical formalism adopted in sections 2 and 3 together with the co-located arrangement of dependent variables simplifies the presentation of the discrete spatial differential operators. The finite-difference (FD) and finite-volume (FV) discretisations in the vertical and horizontal, respectively, combine two standard approaches. In particular, generic second-order-accurate centred finite differences are used in the vertical. The FV discretisation and differentiation on spherical surfaces located at $z = (k - 1)\delta z$ for $k = 1, N$ follows the median-dual approach described in [37]. It is briefly summarized below.

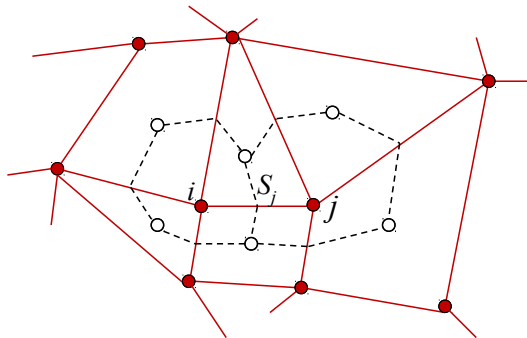


Fig. 1. The edge-based, median-dual approach in 2D. The edge connecting nodes i and j of the primary polygonal mesh pierces, precisely in the edge centre, the face S_j shared by computational dual cells surrounding nodes i and j ; open circles represent geometrical barycentres of the primary mesh, solid lines mark the primary mesh, and dashed lines outline the dual cells with control volumes \mathcal{V}_i and \mathcal{V}_j .

A schematic of an arbitrary mesh on a 2D plane is shown in Fig. 1. The

⁴ The adopted nonsymmetric preconditioned Generalised Conjugate Residual (GCR) approach has been widely discussed in the literature of EULAG; see [32] for a recent overview and a comprehensive list of references.

median-dual FV approach constructs the control volume containing the node i by joining the barycentres of polygonal mesh cells encompassing the node i with the midpoints of the edges originating in the node i . Having defined the mesh in planar geometry, all geometric elements such as cell volume, cell face area, and normals are evaluated from vector calculus in the computational space; i.e., in terms of \mathbf{x} coordinates on a zonally periodic 2D plane [37]. All dependent variables are co-located in the nodes.

The derivations and details of median-dual unstructured mesh discretisation of differential operators entering MPDATA schemes symbolised with \mathcal{A} in (7), are provided in [28] and [36], and a summary of MPDATA with composite FD/FV discretisation in the vertical/horizontal is provided in the following section 4.2. Here, we highlight a key tool for designing discrete differential operators for control-volume schemes; facilitating, in particular, the presentation in section 4.2. For a differentiable vector field \mathbf{A} , the Gauss divergence theorem — $\int_{\Omega} \nabla \cdot \mathbf{A} = \int_{\partial\Omega} \mathbf{A} \cdot \mathbf{n}$ — applied over the control volume \mathcal{V}_i surrounding node i leads to

$$\nabla_i \cdot \mathbf{A} = \frac{1}{\mathcal{V}_i} \sum_{j=1}^{l(i)} A_j^\perp S_j . \quad (26)$$

Hereafter, $l(i)$ numbers edges connecting node i with its neighbours j , and S_j refers both to the face per se and its surface area.⁵ Equation (26) is exact given $\nabla_i \cdot \mathbf{A}$ is interpreted as the mean value of $\nabla \cdot \mathbf{A}$ within the volume \mathcal{V}_i , while A_j^\perp is interpreted as the mean normal component of the vector \mathbf{A} at the cell face S_j . The approximation begins with specifying A_j^\perp in terms of data available on the mesh; i.e., in terms of mean values of the field within the control volumes \mathcal{V}_i and \mathcal{V}_j . One elementary example is

$$A_j^\perp = 0.5 \mathbf{n}_j \cdot [\mathbf{A}_i + \mathbf{A}_j] , \quad (27)$$

where \mathbf{n}_j is a mean outward unit normal to the face S_j . Partial derivatives $\partial_x \Psi$ and $\partial_y \Psi$ of a scalar field Ψ can also be interpreted in terms of the Gauss theorem by representing the two derivatives as the divergence of the augmented vector field $(\Psi, 0)$ and $(0, \Psi)$ as, e.g.,

$$\left. \frac{\partial \Psi}{\partial x} \right|_i = \frac{1}{\mathcal{V}_i} \sum_{j=1}^{l(i)} 0.5(\Psi_i + \Psi_j) S_j^x \quad (28)$$

where S_j^x denotes the x component of the oriented surface element $\mathbf{S}_j = S_j \mathbf{n}_j$ of the face at the j th edge. Evaluation of $\partial_y \Psi|_i$ proceeds analogously, but using the S_j^y component.

⁵ In the FVM code, S_j and \mathcal{V}_i have dimensions of length and area, yet the actual face areas and volumes of prismatic cells are, respectively, $S_j \delta z$ and $\mathcal{V}_i \delta z$, in computational space; consequently, their ratios are always S_j/\mathcal{V}_i regardless of δz .

To illustrate a FD/FV differentiation, consider the 3D operator of the velocity divergence in the first term on the rhs of (21)

$$\begin{aligned} \left[\frac{1}{\mathcal{G}} \nabla \cdot (\mathcal{G}\mathbf{v}) \right]_{k,i} &= \frac{1}{\mathcal{G}_{k,i} \mathcal{V}_i} \sum_{j=1}^{l(i)} \left(\overline{\mathcal{G}_k u_k}^{i,j} S_j^x + \overline{\mathcal{G}_k v_k}^{i,j} S_j^y \right) \\ &+ \frac{1}{\mathcal{G}_{k,i} \delta z} \left(\mathcal{G}w|_{k+1/2,i} - \mathcal{G}w|_{k-1/2,i} \right) , \end{aligned} \quad (29)$$

where

$$\begin{aligned} \overline{\mathcal{G}_k u_k}^{i,j} &= 0.5 (\mathcal{G}_{k,i} u_{k,i} + \mathcal{G}_{k,j} u_{k,j}) , \\ \overline{\mathcal{G}_k v_k}^{i,j} &= 0.5 (\mathcal{G}_{k,i} v_{k,i} + \mathcal{G}_{k,j} v_{k,j}) , \\ \mathcal{G}w|_{k+1/2,i} &= 0.5 (\mathcal{G}_{k,i} w_{k,i} + \mathcal{G}_{k+1,i} w_{k+1,i}) . \end{aligned} \quad (30)$$

The x and y components (see Appendix A) of the oriented surface element \mathbf{S}_j are evaluated from the coordinates of the barycentres defining the face S_j . In particular,

$$\begin{aligned} S_j^x &= y_l - y_r , \\ S_j^y &= x_r - x_l , \end{aligned} \quad (31)$$

where subscripts l and r refer respectively to the barycentres to the left and right of the \vec{ij} edge piercing the S_j face. It is easy to check that for the special case of rectangular mesh with uniform grid increments δx and δy (so $\forall_i \mathcal{V}_i = \delta x \delta y$), the first term on the rhs of (29) becomes a sum of standard centred finite-difference derivatives in x and y , analogous to the z derivative in the second term on the rhs of (29).

The actual execution of (29) and (30) in the FVM code commences with evaluating (30) in a loop over the edges for all vertical levels k . Then the normal velocity to the face S_j (the parenthetic term under the sum in the first term of 29) is evaluated in the analogous loop. Only then, the horizontal divergence is calculated in a loop over nodes, as a sum over $l(i)$, for all vertical levels k and stored as an auxiliary field. Subsequently, the second term on the rhs of (29) is evaluated in a loop over nodes, and combined with the horizontal divergence in a separate loop over nodes and vertical levels. This is different from purely edge-based codes that even in 2D (i.e., FD/FV aspects aside) evaluate both the normal velocity to the face and the divergence in loops over edges and, thus do not require storing connectivity of the nodes with its neighbouring edges. This distinction makes the FVM also a hybrid in terms of edge- and node-based programming driven by the efficiency of mixed MPI/OpenMP parallelisation discussed in section 5.2.

Having established the means of evaluating elementary differential operators the building blocks of section 3 can be combined into a complete apparatus for integrating the governing PDEs presented in section 2. The only nontrivial elements still missing are specifications related to the composite FD/FV MPDATA, and the periodic and polar boundary conditions on each horizontal surface of the computational domain. The latter are recapitulated in section 5.2, after [37], as they are intimately linked with the parallelisation scheme; whereas the former are discussed next.

4.2 Composite FD/FV MPDATA

MPDATA is an iterative application of the first-order accurate upwind scheme, where the first iteration uses physical velocities while all subsequent iterations compensate the leading truncation error of the preceding iteration using analytically derived pseudo-velocities and most recent iterates of the transported field. The MPDATA approach encompasses a number of schemes.⁶ However, even the simplest (basic) algorithm with one corrective iteration is conservative, sign-preserving and at least second-order accurate.

Using the notation of section 3.1, the integral (7) of the homogeneous (viz. $\mathcal{R} \equiv 0$) generalised transport equation (6) can be represented, see [35], as

$$\begin{aligned} \Psi_{\mathbf{i}} &= \mathcal{A}_{\mathbf{i}} \left(\Psi^n, \mathbf{V}^{n+1/2}, G^n, G^{n+1} \right) \\ &= \chi_{\mathbf{i}}^{n+1/2} \mathcal{A}_{\mathbf{i}}^h \left(\Psi^n, \mathbf{V}^{n+1/2}, G^n, \chi^{n+1/2}, \dots \right) \\ &= \chi_{\mathbf{i}}^{n+1/2} \Psi_{\mathbf{i}}^{\mathcal{N}_\eta} . \end{aligned} \tag{32}$$

Here $\mathbf{i} = (k, i)$ is the vector index with k marking the grid point position in the vertical and i a mesh node in the horizontal; the “...” is a shorthand for the usual dependence on the grid/mesh parameters δt , δz , \mathcal{V} and S already used in section 4.1; $\chi^{n+1/2} \equiv G^n/G^{n+1}$; and \mathcal{A}^h refers to the basic version of a composite MPDATA that combines the established FD [29,13] and FV [28,36] forms, formulated for G independent of time in [27,37]. Namely, \mathcal{A}^h iterates for $\eta = 1, \mathcal{N}_\eta$ the discrete form

⁶ The reader interested in a hands-on experience with MPDATA is referred to [10], a recently released library of solvers for systems of transport equations (6).

$$\begin{aligned}
\Psi_{k,i}^{(\eta)} &= \Psi_{k,i}^{(\eta-1)} - \frac{\delta t}{G_{k,i} \mathcal{V}_i} \sum_{j=1}^{l(i)} F_{k,j}^{\perp} \left(\Psi_{k,i}^{(\eta-1)}, \Psi_{k,j}^{(\eta-1)}, V_{k,j}^{\perp(\eta)} \right) S_j \\
&\quad - \frac{\delta t}{G_{k,i} \delta z} \left\{ F_{k+1/2,i}^z \left(\Psi_{k,i}^{(\eta-1)}, \Psi_{k+1,i}^{(\eta-1)}, V_{k+1/2,i}^z(\eta) \right) \right. \\
&\quad \left. - F_{k-1/2,i}^z \left(\Psi_{k-1,i}^{(\eta-1)}, \Psi_{k,i}^{(\eta-1)}, V_{k-1/2,i}^z(\eta) \right) \right\} , \tag{33}
\end{aligned}$$

together with

$$\Psi^{(0)} \equiv \Psi^n , \quad \Psi^{(\mathcal{N}_\eta)} \equiv \Psi^{n+1} , \quad \check{\Psi} \equiv \chi^{n+1/2} \Psi , \tag{34}$$

$$V^{\perp(\eta+1)} = V^{\perp} \left(\mathbf{V}^{(\eta)}, \check{\Psi}^{(\eta)}, \nabla \check{\Psi}^{(\eta)}, G^n, \dots \right) , \quad V^{\perp(1)} \equiv V^{\perp}|^{n+1/2} , \tag{35}$$

$$V^{z(\eta+1)} = V^z \left(\mathbf{V}^{(\eta)}, \check{\Psi}^{(\eta)}, \nabla \check{\Psi}^{(\eta)}, G^n, \dots \right) ; \quad V^{z(1)} \equiv V^z|^{n+1/2} . \tag{36}$$

In (33), the second term on the rhs — reminiscent of (26) — represents the horizontal divergence of advective flux normal to the S_j face of the dual cell surrounding node i . In MPDATA this flux always assumes the functional form of the generic upwind flux; e.g.,

$$F_{k,j}^{\perp} \left(\Psi_{k,i}^{(\eta-1)}, \Psi_{k,j}^{(\eta-1)}, V_{k,j}^{\perp(\eta)} \right) = \left[V_{k,j}^{\perp(\eta)} \right]^+ \Psi_{k,i}^{(\eta-1)} + \left[V_{k,j}^{\perp(\eta)} \right]^- \Psi_{k,j}^{(\eta-1)} , \tag{37}$$

where the nonnegative/nonpositive parts

$$\left[V^{\perp} \right]^+ \equiv \max \left[0, V^{\perp} \right] , \quad \left[V^{\perp} \right]^- \equiv \min \left[0, V^{\perp} \right] \tag{38}$$

always coincide with outflow/inflow from the i th cell. The third term on the rhs — reminiscent of the second term on the rhs of (29) — represents the vertical divergence of the advective flux through the horizontal faces of the prismatic cells. Its explicit functional form is analogous to (37),

$$\begin{aligned}
F_{k+1/2,i}^z \left(\Psi_{k,i}^{(\eta-1)}, \Psi_{k+1,i}^{(\eta-1)}, V_{k+1/2,i}^z(\eta) \right) &= \\
& \left[V_{k+1/2,i}^z(\eta) \right]^+ \Psi_{k,i}^{(\eta-1)} + \left[V_{k+1/2,i}^z(\eta) \right]^- \Psi_{k+1,i}^{(\eta-1)} . \tag{39}
\end{aligned}$$

However, unlike in (37), here $[V^z]^+$ and $[V^z]^-$ always take Ψ from the nodes below and above, respectively, so there is no fixed correspondence between the sign of V^z and the cell inflow/outflow. Negative/positive V^z at the face above the node k correspond to inflow/outflow, and vice-versa at the face below. This is a consequence of the fixed bottom-top orientation of the vertical edges in the FD formulation.

For $\mathcal{N}_\eta = 1$, the algorithm in (33)-(40) is simply a composite FV/FD first-order upwind advection scheme. The crux of the MPDATA is in the subsequent

iterations for $\eta > 1$, in which it relies on expressions specifying the functional dependences (35) and (36) of the $V^{\perp(\eta+1)}$ and $V^{z(\eta+1)}$ advector iterates on the vector field \mathbf{V} , field Ψ and its spatial derivatives $\nabla\Psi$ at the preceding iteration. These expressions can be intricate and their specific form depends on the various MPDATA options. Because they are broadly documented in the literature, we refer the reader to [13,35,10] for technical expositions and novelties pertinent to structured grids; whereas corresponding technicalities for unstructured meshes can be found in [28,36,37].

5 DATASTRUCTURE AND PARALLELISATION

The FVM module is built on a newly developed framework called Atlas. The Atlas framework provides parallel distributed, flexible object-oriented datastructures for both structured grids and unstructured meshes on the sphere. It separates concerns of mathematical model formulation and numerical solutions from the cumbersome management of unstructured meshes, distributed memory parallelism, and input/output of data. While FVM is developed in Fortran 2003, it is recognised that handling flexible structures and carefully controlled memory-management is not easily achieved with the Fortran language. Hence, the language of choice for Atlas is C++, a highly performant language providing excellent object-oriented programming support, and building upon C's memory management proficiency. A Fortran 2003 interface exports all of Atlas' functionality to the FVM.

5.1 Flexible object-oriented datastructure

The FVM instructs Atlas to generate a *Mesh*.⁷ This object stores the horizontal coordinates of every node and requires connectivity tables between the nodes via elements such as triangles, quadrilaterals and lines, as is necessary for unstructured meshes. As the coordinates and connectivity tables can have a large memory footprint for large meshes, the *Mesh* is a distributed object, meaning that the mesh is subdivided in partitions and each parallel task is responsible for one partition. The memory distribution scheme will be detailed in section 5.2.

Using the *Mesh*, *FunctionSpace* objects can be created on demand. A *FunctionSpace* describes in which manner *Fields* are discretised on the mesh. A straightforward *FunctionSpace* is the one where fields are discretised in the

⁷ Herein, italicised terms refer to Atlas classes and their objects.

nodes of the elements. Other *FunctionSpace* objects could describe fields discretised in cell-centres of triangles and quadrilaterals, or in edge-centres of these elements. Another type of *FunctionSpace* describes spectral fields in terms of spherical harmonics. *Fields* are objects that store the actual data contiguously in memory as a one-dimensional array and can be mapped to an arbitrary indexing mechanism to cater for, e.g., the current FVM memory layout, or a different memory layout that proves beneficial on emerging computer hardware. A *Field*, with which a *Metadata* object is associated, can be addressed by a user-defined name. *Metadata* stores basic information like the units of the field, or a time-stamp. It is this flexibility and object-oriented design that leads to more maintainable and future-proof code. Figure 2 illustrates this design with the relevant classes.

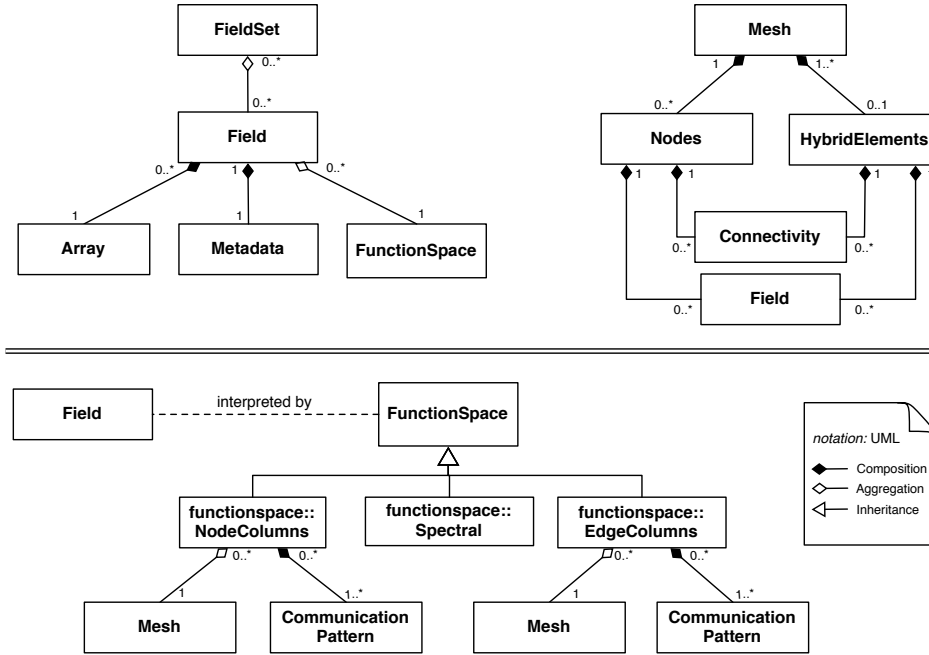


Fig. 2. Atlas datastructure’s aggregations, compositions and class inheritance. A *Field* is composed of *Array* (containing data) and *Metadata* (containing data descriptions), and references an abstract *FunctionSpace*. Multiple *Field* objects can be aggregated into a *FieldSet*. A *Mesh* is composed of *Nodes* and *HybridElements* that store mesh-specific *Fields* (e.g. node coordinates) and (*Connectivity*) tables relating elements and nodes. Concrete *FunctionSpace* objects, such as *functionspace::NodeColumns*, describe how a particular field is discretised and parallelised on the distributed mesh. The characters by the ends of the composition/aggregation arrows quantify the relationships between objects; e.g., 1-1 refers to one-to-one relation, whereas 1-0..* to one-to-any relation.

Currently, the FVM makes use of two particular *FunctionSpaces*: one for fields defined in the nodes of the primary mesh, and one for fields defined in edge-centres of the primary mesh. Example fields stored in edge-centres are advective velocities and fluxes, and vectors normal to the dual mesh faces (S_j in Fig. 1). These normal vectors can be stored in edge-centres of the primary

mesh as every edge corresponds to one face of its dual mesh. The volume \mathcal{V}_i of the dual cell (recall the discussion following eq. 26) is stored in the nodes.

In principle, FVM and Atlas can accommodate nearly any form of horizontal meshing. However, in the envisaged context of being able to use spectral transforms on the nodes of the FV mesh, the nodes have to satisfy certain constraints. In meridional direction, they must coincide with the roots of the ordinary Legendre polynomials, or roots of other suitable quadratures. Furthermore, longitudes on each latitude must be distributed uniformly to satisfy the requirements of fast Fourier transforms (FFTs), while reducing in number going from the equator towards the poles for efficiency. Given these fixed node locations, Atlas is capable of generating a mixed triangular/quadrilateral mesh. The IFS's set of reduced Gaussian grids satisfy these requirements. For illustration, Fig. 3a shows the meshing of a coarse N24 reduced Gaussian grid with an approximate resolution of 3.75° (415 km; N24 indicates 24 latitudes between pole and equator). A second example is the mesh based on triangulating an octahedron with approximately the same resolution, still satisfying the reduced Gaussian grid requirements, Fig. 3b. Both meshes have the same resolution in the vicinity of the poles, but the more uniform triangulation of the octahedral mesh (Fig. 3b) has a locally more homogeneous dual resolution, which provides superior results as discussed in section 6.

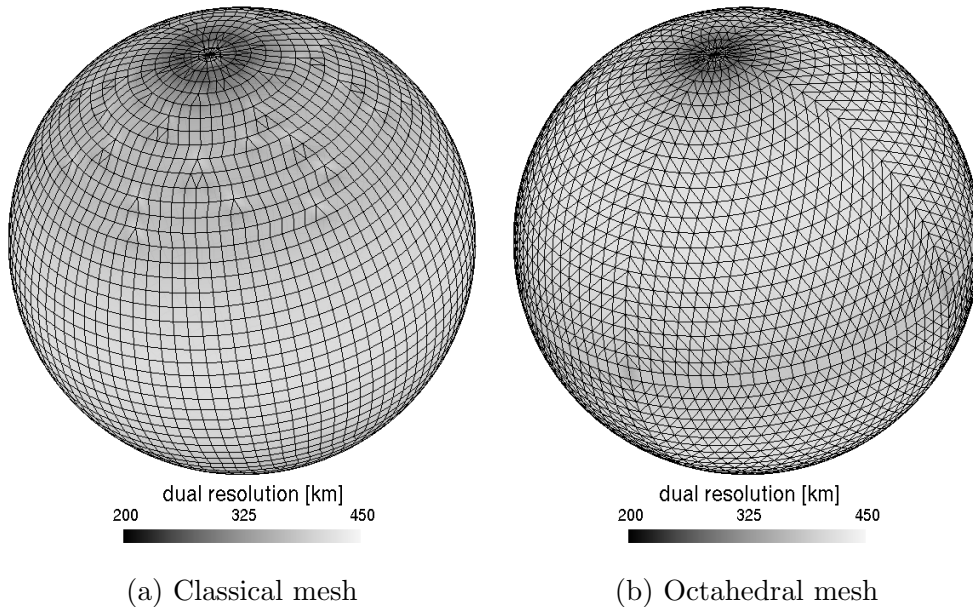


Fig. 3. Primary meshes generated around N24 reduced Gaussian grid points with approximate resolution of 3.75° (415 km). The shading represents the dual resolution, computed as the square root of the local dual volume. The octahedral mesh shows a locally more homogeneous dual resolution than the classical mesh.

5.2 Parallelisation scheme

The FVM is designed to run at High Performance Computing (HPC) centres which provide a grouping of computational cores in tasks. Computational cores in each task have access to shared memory among themselves, whereas tasks don't share memory and are connected by message passing. The FVM hybridises two standards — Message Passing Interface (MPI) and Open Multi-Processing (OpenMP) — for parallelisation, each with a different purpose. MPI is a standard for handling the communication of data between tasks; i.e. the data is distributed between tasks, and sending and receiving of information is required at regular intervals to advance a simulation. OpenMP is a standard for easily managing multiple computational threads working with shared memory; e.g. several iterations in one *do* loop can be executed simultaneously within an algorithm. Following the IFS experience, the hybrid MPI/OpenMP parallelisation delivers about 20% better performance than MPI only, which is related to an improved load balance and memory scalability for large grids.

5.2.1 Distributed memory parallelisation

The parallelisation of FVM relies on the distribution of the computational mesh with its unstructured horizontal index, so that structured vertical columns are always preserved in memory as a contiguous entity. The Atlas framework, upon which the FVM is built, is responsible for generating a distributed mesh, and provides communication patterns using MPI to exchange information between the different partitions of the mesh. To minimise the cost of sending and receiving data, the distribution of the mesh is based on a *equal regions* domain decomposition algorithm optimal for a quasi-uniform node distribution on the globe [18,15]. The equal regions domain decomposition divides the globe in bands oriented in zonal direction, and subdivides each band in a number of regions so that globally each region has the same number of nodes. Noteworthy is that the bands covering the poles are not subdivided, communication-wise forming two polar caps; see [18,15] for details. Figure 4 illustrates the domain decomposition of the mesh displayed in Fig. 3a with 32 partitions.

For evaluating FV differential operators, the FVM requires only nearest neighbour communication by means of a halo between neighbouring partitions, with halo thickness limited to one element of the primary mesh, as shown in Fig. 4. Nearest neighbour communication available in FVM contrasts with global communication required for spectral transforms. It is essential with the trend of increasing computing power by adding more and more cores to the HPC cluster, up to the point where global communication will become a severe bottleneck in achieving the required scalability. The underlying Atlas framework is used to create the internal halos and to manage communication

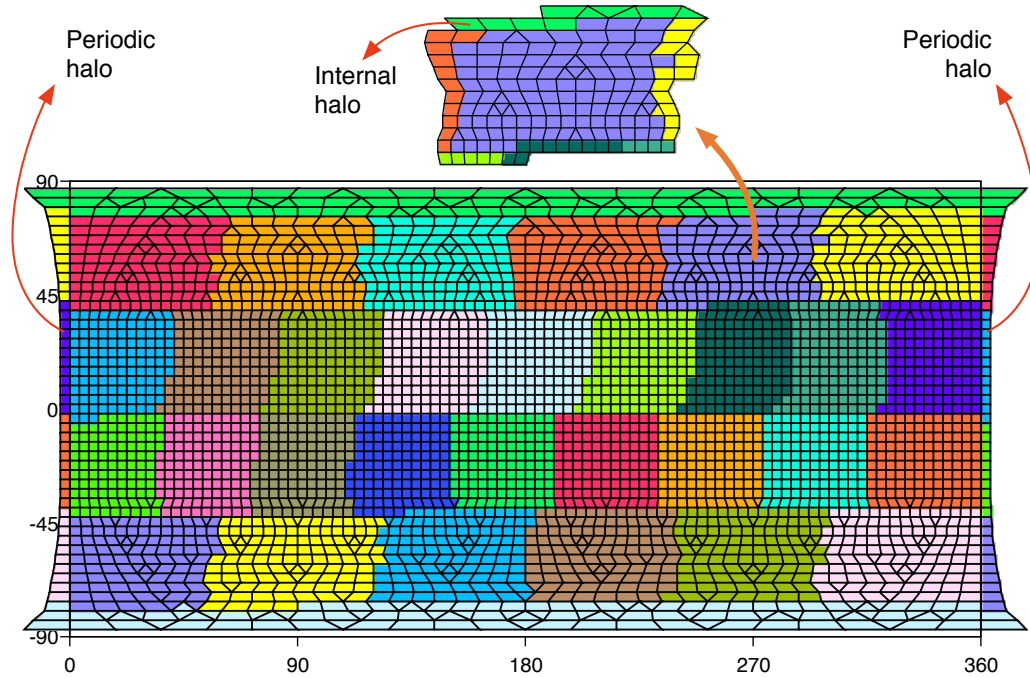


Fig. 4. Equal regions domain decomposition (32 partitions) of mesh displayed in figure 3a. Also shown is the internal halo of one partition, and the periodic halo responsible for the periodic boundary condition.

patterns to exchange data in the halos, thus separating this concern from the FVM.

Because the computational mesh used in the FVM is defined on a 2D periodic longitude-latitude plane, the FVM needs periodic boundary conditions at, say, the Greenwich meridian. To avoid having to implement periodic boundaries explicitly, a periodic halo is created as illustrated in Fig. 4. Consequently, the periodic boundary is treated exactly like any internal boundary between different partitions. Note that the partitions involving the poles are periodic with themselves.

5.2.2 Memory layout and shared memory parallelisation

In memory, a field is stored as one large contiguous 1D array that can be reinterpreted as a multi-dimensional array by the FVM. The FVM chooses the memory layout of a full 3D field with the horizontal unstructured index as the slowest moving index, followed by the structured vertical index, and the fastest index being the number of variables a field contains (e.g. scalar=1, vector=3, tensor=9). Ignoring the index for the variables (i.e., assuming for scalar field), the memory layout is sketched in Fig. 5.

The advantage of this memory layout is twofold. First, it makes the verti-

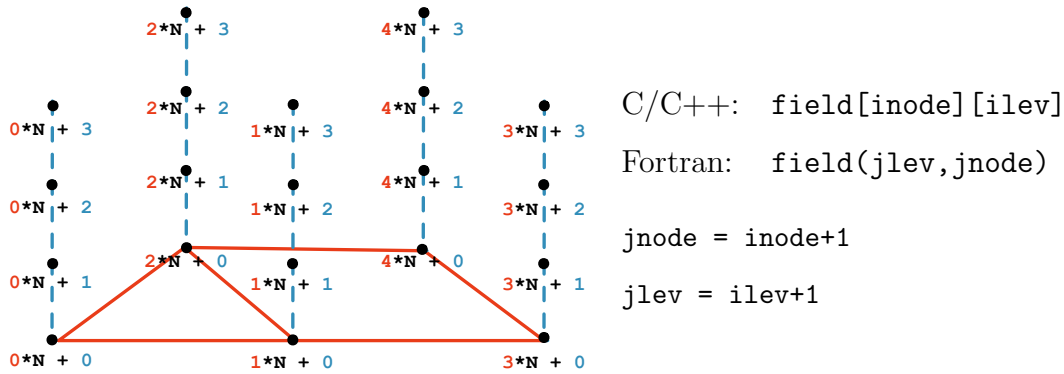


Fig. 5. Memory layout of a scalar field in 3D domain: solid lines show the underlying unstructured mesh; dashed lines mark the structured vertical columns with N denoting the number of vertical levels; and dots, numbered with memory offsets from the first index of the 1D array, represent the field values.

cal columns contiguous in memory, so that a halo-exchange involves contiguous chunks of memory and makes the packing and unpacking of send/receive buffers more efficient. Second, it favours the outer loop to be over the columns and the inner loop to be over the levels within each column, with corresponding indices $jnode$ and $jlev$, Fig. 5. Due to the unstructured nature of the horizontal $jnode$ index, indirect addressing is required to access neighbouring column data. The horizontal index for the outer loop then reduces the cost of this lookup by reusing the node specific computations for the entire column in the inner loop, giving the compiler the opportunity to optimise the inner loop in the vertical direction further with vector instructions, provided that computations for each level are independent of each other. By using shared-memory parallelisation with OpenMP over the outer horizontal index further need for distributing the mesh is avoided, which would otherwise require a larger memory footprint due to having relatively more halo nodes.

6 RESULTS

In the following, we discuss solutions generated with the compressible option of FVM suitable for large time step simulations of low Mach number flow. In the course of the module development the acoustic and two soundproof options encapsulated in (1) were also extensively used to benchmark and verify the module. Their solutions are not shown here, as they corroborate the conclusions of [35] regarding relative comparability of compressible and soundproof results. The two physical problems considered in the current section serve different purposes and are distinct in nature. The sheared orographic flow on a small planet attests the nonhydrostatic quality of the solver, whereas the baroclinic instability targets essentially hydrostatic midlatitude weather systems. The two problems complement each other in demonstrating diverse aspects of

the FVM solutions.

6.1 Quasi two-dimensional orographic flow with linear vertical shear

As pointed out in [42], this classical problem constitutes a particularly discriminating test for a nonhydrostatic model, because in the presence of shear the nonhydrostatic and hydrostatic equations predict a fundamentally different propagation of orographically-forced gravity waves. While hydrostatic models produce a vertically propagating gravity wave, the correct solution is that of a trapped, horizontally propagating gravity wave [12].

The setup of the numerical experiment matches that in [42]. A three-dimensional hill with elliptic horizontal cross section and the classical “witch of Agnesi” vertical profile is centred at the equator

$$h(\phi, \lambda) = h_0 \left((1 + (l_\lambda/L_\lambda)^2) + (l_\phi/L_\phi)^2 \right)^{-1} \quad (40)$$

with $l_\lambda = a \cos^{-1}[\sin^2 \phi_c + \cos^2 \phi_c \cos(\lambda - \lambda_c)]$ and $l_\phi = a \cos^{-1}[\sin \phi_c \sin \phi + \cos \phi_c \cos \phi]$, where the mountain half-width is $L_\lambda = 2.5$ km, and the meridional extent of the ellipse is defined by $L_\phi = |L_\lambda^2 - L_f^2|^{1/2}$, the centre position of the mountain $(\lambda_c, \phi_c) = (3\pi/2, 0)$, and the focus point distance $L_f = a \cos^{-1}[\sin \phi_d \sin \phi_c + \cos \phi_d \cos \phi_c \cos(\lambda_d - \lambda_c)]$ with $(\lambda_d, \phi_d) = (3\pi/2, \pi/3)$; mountain height is $h_0 = 500$ m. All distances and formulae are expressed following great circles on the sphere. Ambient conditions consist of the linearly sheared wind profile $u_a(\phi, z) = U_0(1 + Cz) \cos(\phi)$ below the tropopause located at 10.5 km, and constant aloft; $U_0 = 10 \text{ ms}^{-1}$ and $C = 2.5 \times 10^{-4} \text{ m}^{-1}$; ($v_a = 0, w_a = 0$) and the Brunt-Väisälä frequency $N = 0.01 \text{ s}^{-1}$. The Richardson number of the flow in the troposphere is $Ri \equiv N^2/(U_0 C)^2 = 16$ and in the stratosphere $Ri = \infty$. A constant stability, $S = N^2/g$, ambient atmosphere is isothermal with $T_0 = g^2/(c_p N^2)$, and the thermodynamic profiles of the ambient state are assumed to coincide with the base state, $(\phi_a, \theta_a) = (\phi_b, \theta_b)$. The model is set for the shallow atmosphere approximation and the planetary rotation is set to zero; see [42] for a discussion.

The FVM domain, a spherical shell with the inner radius $a = 20.3718$ km, is discretised in the horizontal using a finite-volume mesh generated around a $N128$ octahedral reduced Gaussian grid (cf. Fig. 3), with horizontal mesh spacing about 250 m at the equator. In the vertical, the model domain is resolved with 137 levels stretched smoothly (as a hyperbolic tangent), such that the vertical spacing changes from 70 m near the ground up to 1400 m near the top. A rigid lid upper boundary at 85 km was chosen, and the inverse time scale of the Rayleigh-damping absorber profile $\alpha = \tau^{-1} \max\{0, (Z - Z_{thres})/(Z_{top} - Z_{thres})\}$ was applied with $Z_{thres} = 55$ km and $\tau = 300$ s. Given the solution characteristics of the trapped wave in the nonhydrostatic model simulations, vertical

absorbers have little effect on the lower tropospheric solution. The integration time is 2 h with a time step $\delta t = 3$ s.

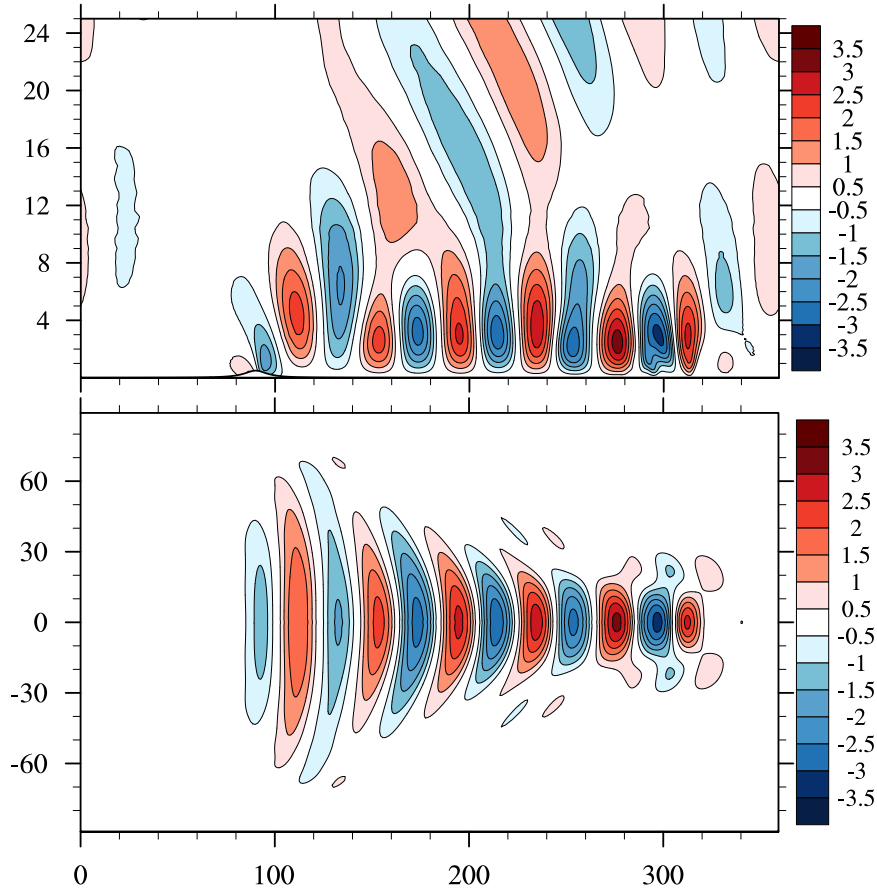


Fig. 6. Gravity wave response after 2h of the simulated quasi-2D sheared orographic flow on a small planet; isolines of vertical velocity [ms^{-1}] are shown in vertical cross-section along the equator (top) and in the horizontal cross-section at $z = 3000$ m (bottom). The abscissa marks the longitude [deg], whereas ordinates mark the altitude [km] and latitude [deg] for the top and bottom panels, respectively.

Figure 6 shows isolines of the vertical velocity after two hours of wave evolution simulated with the FVM. The top panel displays the solution in the vertical cross section at the equator, whereas the bottom panel provides the corresponding display on the spherical surface at $z = 3000$ m above the bottom surface of the shell. The solution is consistent with the numerical results and linear analysis discussed in [42]. Noteworthy is the trapped lee wave behind the mountain with the dominant horizontal wavelength of ≈ 14 km, corresponding to 40° of longitude on the small planet. The wave energy leakage through the tropopause excites a weaker stratospheric wave with the dominant wavelength supported by the ambient conditions about twice that of the trapped wave below. Due to the domain periodicity, the upstream influence of the mountain can already be seen after 2h at the downstream end of the wave packet.

6.2 Baroclinic instability

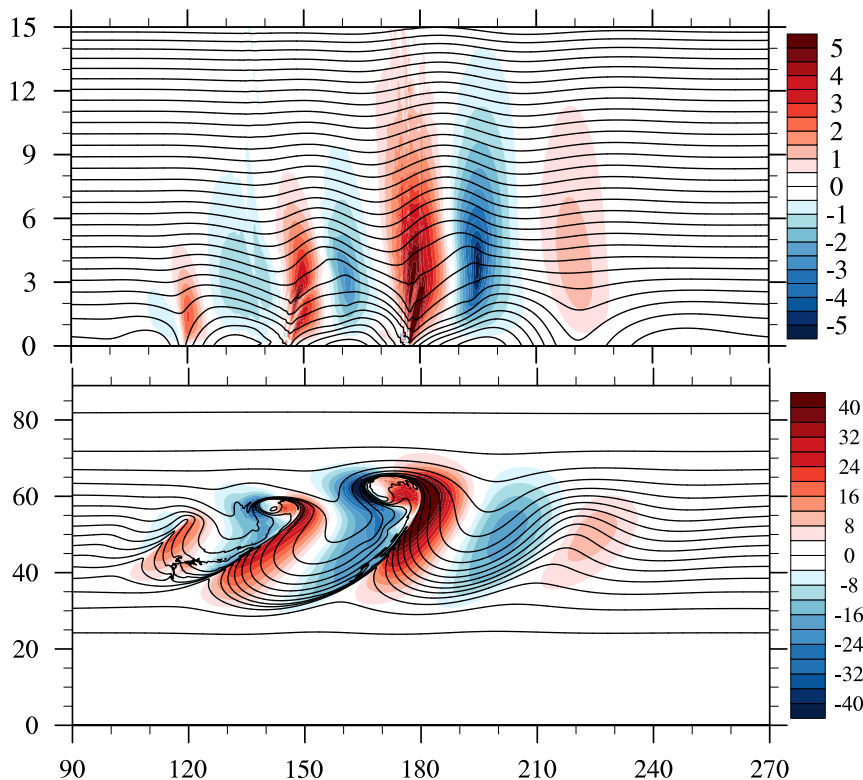


Fig. 7. Baroclinic instability, day 8: (top) color map of vertical velocity [cm/s] overlaid with isentropes, in the vertical cross section at the 53° N latitude; (bottom) color map of surface meridional velocity [m/s] and isentropes. In both panels, isentropes are displayed with the contour interval of 5K. Similarly like in Fig. 6, the abscissa marks longitude [deg], and ordinates mark altitude [km] and latitude [deg], respectively for the top and bottom panels.

In contrast to the section 6.1 that addressed nonhydrostatic performance of the FVM, here we illustrate its capability for simulating essentially hydrostatic motions. For this purpose we repeat the baroclinic instability benchmark adopted in [35] after [9]. In [35], the authors discussed a series of experiments with soundproof and compressible equations using a coarse horizontal resolution of 128×64 nodes of the regular longitude-latitude grid. Here we show the corresponding compressible result generated with FVM using an $N800$ octahedral reduced Gaussian grid, with horizontal grid spacing of 12.5 km corresponding to state-of-the-art operational NWP. The model depth ≈ 24 km is resolved with 61 stretched vertical levels, with smoothly varying resolution of $\delta z = 50$ m near the ground, through $\delta z = 150$ m at 2 km altitude, up to $\delta z = 850$ m near the model top. The employed time step $\delta t = 80$ s resulted in the maximal Courant number $\gtrsim 0.5$ at the time of the frontal collapse.

Figure 7 displays the baroclinic wave train after 8 days evolution from a weakly perturbed unstable-equilibrium initial state consisting of the two planetary

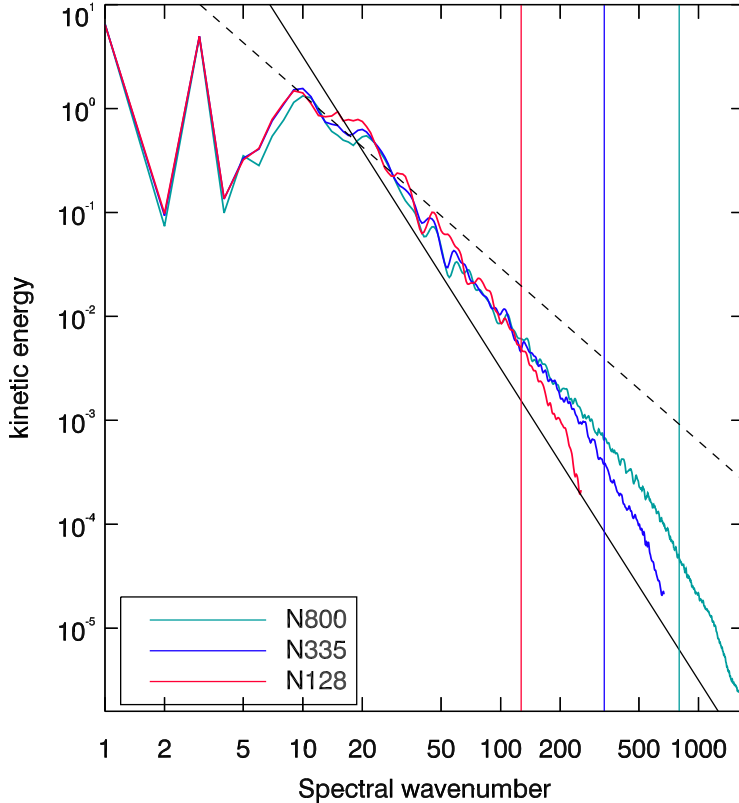


Fig. 8. Surface kinetic energy spectra at day 9 of the baroclinic instability evolution, simulated with various horizontal resolutions. Vertical lines on the right indicate four grid intervals for each corresponding resolution. For reference, the -3 and $-5/3$ slopes are shown with solid and dashed lines, respectively.

jets in midlatitudes. The upper panel shows isentropes overlaid with contours of the vertical velocity, in the vertical cross section through the centre of the northerly jet. Lower panel shows isentropes together with the surface meridional velocity. Together these figures highlight a 3D structure of idealised frontogenesis and formation of weather systems in midlatitudes. These results are consistent with those familiar from the literature. Notably, at the relatively high resolution employed, the simulation begins to capture mesoscale gravity waves radiated at collapsing front [20] that can be seen in the upper panel of Fig. 7 as fine-scale irregularities at $\sim 180^\circ$. In inviscid simulations the associated grid-scale features are intermittent. Generally they are filtered out with an aid of artificial viscosity or subgrid-scale parametrisations. Here they are controlled by the model non-oscillatory numerics providing an implicit subgrid-scale model with documented implicit large-eddy-simulation (ILES) properties [5,30]. The latter is substantiated in Fig. 8 depicting surface kinetic energy spectra at various resolutions. The spectra follow each other closely at the resolved scales of motion and tend to separate at the grid-scale, where implicit dissipation is effective.

We end this section with a comment on the potential efficiency of the proposed

approach — a definitive conclusion is yet to be established as the current FVM code has not been optimised. The presented simulations were conducted on Cray XC30 at the ECMWF. The expense of calculations in section 6.1 is insignificant. The baroclinic instability simulation used 768 MPI tasks and 6 OMP threads (i.e., 4608 cores). The 9720 time steps used to resolve 9 days of the instability evolution took $\lesssim 7$ h of the wall-clock time. An equivalent simulation (not shown) with the 30 km resolution and three times larger time step $\delta t = 240$ s took 20 min of the wall-clock time of the 4608 cores. This is commensurate with 17-fold reduction of the computational work due to the coarser temporal and spatial resolution. Noteworthy, in both calculations average number of iterations in the Helmholtz solver is about 10 per time step, thanks to the physically motivated stopping criterion relating the exit condition to the actual magnitude of the flow divergence [26].

7 CONCLUSIONS

From the perspective of numerical methods, this paper synthesised finite-difference [21] and flexible finite-volume [37] discretisations of the computational space underlying curvilinear coordinate representations of the compressible and soundproof nonhydrostatic PDEs governing global atmospheric dynamics [35]. The work combines the two modes of discretisation with continuous mappings to supplement large 3D atmospheric models. A fully unstructured discretisation of the computational domain in the horizontal allows to circumvent the efficiency issues due to meridians’ convergence towards the poles by resolving the spherical surface with uniform control volumes in the physical space [37]. Here, the regularity of the discretisation is compromised to build the control volumes about the predetermined node locations suitable for spherical harmonics based quadrature on the sphere. This grid arrangement need not lead to a compromise in the solution accuracy, because the degradation of the mesh quality is found to decrease with increasing model resolution, and it is found to be insignificant for natural flows. In return the constrained flexible discretisation facilitates interactions with established NWP models, as illustrated by adopting the efficient parallelisation scheme of the IFS and the use of its spectral transforms (Fig. 8). Finite-difference discretisation in the vertical empowers direct preconditioning of complex elliptic boundary value problems (section 3.3) imperative for thin spherical shells such as the Earth’s atmosphere. The benefits of continuous mappings include the analytic representation of spherical shells with irregular lower boundary (section 6.1), mesh adaptivity [21,13], pliancy of physical vertical coordinate [41,37], and a minimal overhead to the parallelisation.

From the perspective of NWP applications, this paper demonstrates that the FVM on its own can provide solutions representative of elements of real

weather at disparate scales, involving entirely different computation and communication patterns compared to spectral-transform based contemporary global NWP. These elements can be identified in Fig. 9 that depicts a snapshot of the FVM simulation of a global circulation on the N256 reduced Gaussian grid (cf. left panel of Fig. 3) using realistic orography (cf. section 2.5 in [2]), together with an idealised frictional/diabatic forcing established for intercomparison of the dynamical cores of the atmospheric general circulation models [8]. Figure 9 illustrates the importance of the selected benchmarks for the broader NWP context, where these representative elements are embedded in the global atmospheric flow.

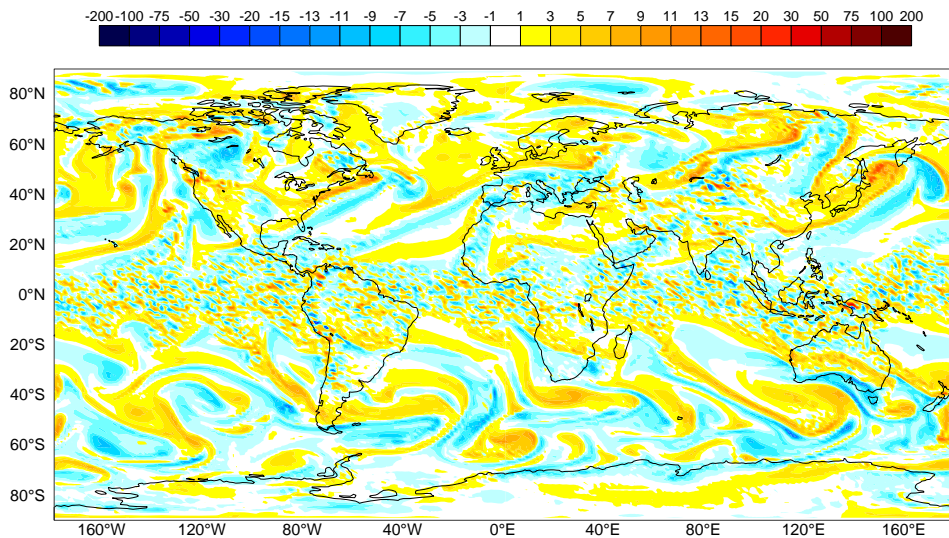


Fig. 9. N256 simulation of a global circulation using the Earth’s orography [2] and idealised diabatic/frictional forcing [8]. Vertical component of instantaneous relative vorticity($\times 10^6$) is displayed at about 4 km above the surface. Noteworthy are baroclinic eddies in midlatitudes of both hemispheres and fine-scale features in the equatorial area and mountainous regions indicative of convection and gravity-waves.

Altogether, the reported development and supporting simulations foreshadow new opportunities for blending the strengths of cloud-resolving and NWP models. The FVM together with the associated infrastructure provides a toolbox of methods previously inaccessible to spectral-transform based models. On the other hand, the IFS provides the required advanced environment to explore alternative discretisation techniques in the context of operational weather forecasting.

Acknowledgements: Comments from two anonymous referees helped to improve the presentation. This work was supported in part by funding received from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2012/ERC Grant agreement no. 320375). The graphics in Fig. 9 used the *Metview* meteorological visualisation software;

Appendix A. Specifications of the spherical frame

In the spherical curvilinear framework of [21], the vector \mathbf{u} represents the physical velocity with components aligned at every point of the spherical shell with axes of a local Cartesian frame (subsequently marked as $_c$) tangent to the lower surface ($r = a$) of the shell; r is the radial component of the vector radius, and a is the radius of the sphere, cf. Fig. 7.7, section 7.2 in [7]. Consequently, $dx_c = r \cos \phi d\lambda$, $dy_c = r d\phi$ and $z_c = r - a$; where λ and ϕ denote longitude and latitude angles, respectively. Then, in the formalism of sections 2 and 3 and in the absence of coordinate stretching, $x = a\lambda$, $y = a\phi$, and $z = z_c$; thereby effectively employing longitude-latitude coordinates standard in many global atmospheric models [37]. Furthermore, the coefficient matrix $\tilde{\mathbf{G}}$ consists of zero off-diagonal entries, whereas $\tilde{G}_1^1 = [\Gamma \cos(y/a)]^{-1}$, $\tilde{G}_2^2 = \Gamma^{-1}$, and $\tilde{G}_3^3 = 1$. Here, $\Gamma = 1 + z/a$, and indices 1, 2, and 3 correspond to x , y , and z components. Consequently, the Jacobian is $\mathcal{G} = \Gamma^2 \cos(y/a)$.

In the momentum equation, the components of the Coriolis acceleration are

$$-\mathbf{f} \times \mathbf{u} = \begin{bmatrix} v f_0 \sin(y/a) - w f_0 \cos(y/a), \\ -u f_0 \sin(y/a), \\ u f_0 \cos(y/a) \end{bmatrix}, \quad (41)$$

where $\mathbf{u} = [u, v, w]$ and $f_0 = 2|\boldsymbol{\Omega}|$. Furthermore, the metric forcings (viz., component-wise Christoffel terms associated with the convective derivative of the physical velocity) are,

$$\mathcal{M}(\mathbf{u}) = (\Gamma a)^{-1} \begin{bmatrix} \tan(y/a) u v - u w, \\ -\tan(y/a) u u - v w, \\ u u + v v \end{bmatrix}. \quad (42)$$

References

- [1] M. Baldauf, A. Seifert, J. Förstner, D. Majewski, M. Raschendorfer, Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities, *Month. Weather Rev.* 139 (2011), 3887–3905.

- [2] G. Balsamo, A. Agusti-Panareda, C. Albergel, A. Beljaars, S. Bousssetta, E. Dutra, T. Komori, S. Lang, J. Munoz-Sabater, F. Pappenberger, P. de Rosnay, I. Sandu, N. Wedi, A. Weisheimer, F. Wetterhall, E. Zsoter, Representing the Earth surfaces in the Integrated Forecasting System: Recent advances and future challenges, ECMWF Tech. Memorandum 729 (2014), Reading UK.
- [3] S.R.M. Barros, D. Dent, L. Isaksen, G. Robinson, G. Mozdzynski, F. Wollenweber, The IFS model: A parallel production weather code *Parallel Comput.* 21 (1995), 1621-1638.
- [4] P. Bénard, J. Vivoda, J. Mašek, P. Smoliková, K. Yessad, Ch. Smith, R. Brožkovaá, J.F. Geleyn, Dynamical kernel of the ALADIN — NH spectral limited-area model: Revised formulation and sensitivity experiments, *Q.J.R. Meteorol. Soc.* 136 (2010) 155–169.
- [5] J.A. Domaradzki, Z. Xiao, P.K. Smolarkiewicz, Effective eddy viscosities in implicit large eddy simulations of turbulent flows, *Phys. Fluids* 15 (2003) 3890-3893.
- [6] D.R. Durran, Improving the anelastic approximation. *J. Atmos. Sci.* 46 (1989) 1453–1461.
- [7] J.A. Dutton, *The Ceaseless Wind*. Dover Publications, 1986, pp. 617.
- [8] I.M. Held, M.J. Suarez, 1994: A proposal for the intercomparison of the dynamical cores of the atmospheric general circulation models. *Bull. Amer. Met. Soc.* 75 (1994) 1825–1830.
- [9] C. Jablonowski, D.L. Williamson, A baroclinic instability test case for atmospheric model dynamical cores, *Q.J.R. Meteorol. Soc.* 132 (2006) 2943–2975.
- [10] A. Jaruga, S. Arabas, D. Jarecka, H. Pawlowska, P.K. Smolarkiewicz, M. Waruszewski, libmpdata++ 1.0: a library of parallel MPDATA solvers for systems of generalised transport equations, *Geosci. Model. Dev.* 8 (2015) 1005–1032.
- [11] Z.I. Janjic, A nonhydrostatic model based on a new approach, *Meteorol. Atmos. Phys.* 82 (2003) 271–285.
- [12] T.L. Keller, Implications of the hydrostatic assumption on atmospheric gravity-waves, *J. Atmos. Sci.* 51 (1994) 1915–1929.
- [13] C. Kühnlein, P.K. Smolarkiewicz, A. Dörnbrack, Modelling atmospheric flows with adaptive moving meshes, *J. Comput. Phys.* 231 (2012) 2741–2763.
- [14] M. Kurowski, W.W. Grabowski, and P.K. Smolarkiewicz, Anelastic and compressible simulations of moist deep convection, *J. Atmos. Sci.* 71 (2014) 3767–3787.
- [15] P. Leopardi, A Partitioning of the Unit Sphere of Equal Area and Small Diameter, *Electronic Transactions on Numerical Analysis* 25 (2006) 309–327.

- [16] F.B. Lipps, R.S. Hemler, A scale analysis of deep moist convection and some related numerical calculations, *J. Atmos. Sci.* 39 (1982) 2192–2210.
- [17] F.B. Lipps, On the anelastic approximation for deep convection, *J. Atmos. Sci.* 47 (1990) 1794–1798.
- [18] G. Mozdzyński, A new partitioning approach for ECMWF’s integrated forecasting system (IFS). p 148-166 in *Proceedings of the Twelfth ECMWF Workshop: Use of High Performance Computing in Meteorology, 30 October - 3 November, 2006, Reading, UK*, World Scientific (2007) 273 pp.
- [19] G. Mozdzyński, M. Hamrud, N.P. Wedi (2015), A Partitioned Global Address Space implementation of the ECMWF Integrated Forecasting System (IFS), *Int. J. High Perform. C.* 29 (2015) 261–273.
- [20] R. Plougonven R., F. Zhang, Internal gravity waves from atmospheric jets and fronts, *Rev. Geophys.*, 52 (2013) 33-76.
- [21] J.M. Prusa, P.K. Smolarkiewicz, An all-scale anelastic model for geophysical flows: dynamic grid deformation, *J. Comput. Phys.* 190 (2003) 601–622.
- [22] J.M. Prusa, P.K. Smolarkiewicz, A.A Wyszogrodzki, EULAG, a computational model for multiscale flows, *Comput. Fluids* 37 (2008) 1193-1207
- [23] A. Qaddouri, V. Lee, The Canadian Global Environmental Multiscale model on the Yin-Yang grid system, *Q.J. Roy. Meteorol. Soc.*, 137 (2011) 1913–1926.
- [24] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno, S. Iga, Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations, *J. Comput. Phys.* 227 (2008) 3486–3514.
- [25] Y. Seity, P. Brousseau, S. Malardel, G. Hello, P. Bénard, F. Bouttier, C. Lac, V Masson, The AROME-France convective-scale operational model, *Month. Weather Rev.* 139 (2011) 976–991.
- [26] W.C. Skamarock, P.K. Smolarkiewicz, J.B. Klemp, Preconditioned conjugate-residual solvers for Helmholtz equations in nonhydrostatic models, *Month. Weather. Rev.* 125 (1997) 587–599.
- [27] P.K. Smolarkiewicz, L.G. Margolin, On forward-in-time differencing for fluids: Extension to a curvilinear framework, *Month. Weather Rev.* 121 (1993) 1847–1859.
- [28] P.K. Smolarkiewicz, J. Szmelter, MPDATA: An edge-based unstructured-grid formulation, *J. Comput. Phys.* 206 (2005) 624–649.
- [29] P.K. Smolarkiewicz, Multidimensional positive definite advection transport algorithm: an overview, *Int. J. Numer. Meth. Fluids* 50 (2006) 1123–1144.
- [30] P.K. Smolarkiewicz, L.G. Margolin, Studies in geophysics. In *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics* (ed. F. F. Grinstein, L. Margolin, W. Rider). Cambridge University Press (2007), pp 546.

- [31] P.K. Smolarkiewicz, J. Szmelter, Iterated upwind schemes for gas dynamics, *J. Comput. Phys.* 228 (2009) 33–54.
- [32] P.K. Smolarkiewicz, J. Szmelter, A nonhydrostatic unstructured-mesh soundproof model for simulation of internal gravity waves, *Acta Geophysica* 59 (2011) 1109–1134.
- [33] P.K. Smolarkiewicz, P. Charbonneau, EULAG, a computational model for multiscale flows: An MHD extension, *J. Comput. Phys.* 236 (2013) 608–623.
- [34] P.K. Smolarkiewicz, J. Szmelter, A.A. Wyszogrodzki, An unstructured-mesh atmospheric model for nonhydrostatic dynamics, *J. Comput. Phys.* 254 (2013) 184–199.
- [35] P.K. Smolarkiewicz, C. Kühnlein, N.P. Wedi, A consistent framework for discrete integrations of soundproof and compressible PDEs of atmospheric dynamics, *J. Comput. Phys.* 263 (2014) 185–205
- [36] J. Szmelter, P.K. Smolarkiewicz, MPDATA error estimator for mesh adaptivity. *Int. J. Numer. Meth. Fluids* 50 (2006) 1269–1293.
- [37] J. Szmelter, P.K. Smolarkiewicz, An edge-based unstructured mesh discretisation in geospherical framework, *J. Comput. Phys.* 229 (2010) 4980–4995.
- [38] J. Szmelter, P.K. Smolarkiewicz, An edge-based unstructured mesh framework for atmospheric flows, *Comput. Fluids* 46 (2011) 455–460.
- [39] A. Warn-Varnas, J. Hawkins, P.K. Smolarkiewicz, S.A. Chin-Bing, D. King, Z. Hallock, Solitary wave effects north of Strait of Messina, *Ocean Modelling* 18 (2007) 97–121.
- [40] C. Temperton, M. Hortal, A. Simmons, A two-time-level semi-Lagrangian global spectral model, *Q.J. Roy. Meteorol. Soc.*, 127 (2001) 111–127.
- [41] N.P. Wedi, P.K. Smolarkiewicz, Extending Gal-Chen and Somerville terrain-following coordinate transformation on time dependent curvilinear boundaries, *J. Comput. Phys.* 193 (2004) 1–20.
- [42] N.P. Wedi, P.K. Smolarkiewicz, A framework for testing global nonhydrostatic models, *Q.J. Roy. Meteorol. Soc.*, 135 (2009) 469–484.
- [43] N.P. Wedi, M. Hamrud, G. Mozdzyński, G. Austad, S. Curic, J. Bidlot, Global, non-hydrostatic, convection-permitting, medium-range forecasts: progress and challenges. *ECMWF Newsletter No. 133* (2012) 17–22.
- [44] N.P. Wedi, M. Hamrud, G. Mozdzyński, A fast spherical harmonics transform for global NWP and climate models. *Month. Weather Rev.* 141 (2013) 3450–3461.
- [45] N.P. Wedi, Increasing horizontal resolution in numerical weather prediction and climate simulations: illusion or panacea? *Phil. Trans. Royal Soc. A*, 372 (2014) 20130289.

- [46] D.L. Williamson, The evolution of dynamical cores for global atmospheric models. *J. Meteor. Soc. Japan*, 85B (2007) 241-269.
- [47] N. Wood, A. Staniforth, A. White, T. Allen, M. Diamantakis, M. Gross, T. Melvin, C. Smith, S. Vosper, M. Zerroukat, J. Thuburn, An inherently mass-conserving semi-implicit semi-Lagrangian discretisation of the deep-atmosphere global non-hydrostatic equations, *Q.J.R. Meteorol. Soc.* 140 (2014) 1505-1520.