Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 206 (2005) 624–649
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

MPDATA: An edge-based unstructured-grid formulation

Piotr K. Smolarkiewicz [a,*], Joanna Szmelter [b]

[a] *Mesoscale and Microscale Meteorology Division, National Center for Atmospheric Research, P.O. BOX 3000, Boulder, CO 83307-3000, USA*
[b] *Royal Military College of Science, Cranfield University, Shrivenham, Swindon SN6 8LA, UK*

Received 5 August 2005; received in revised form 9 December 2005; accepted 10 December 2005
Available online 9 February 2005

Abstract

We present an advancement in the evolution of MPDATA (multidimensional positive definite advection transport algorithm). Over the last two decades, MPDATA has proven successful in applications using single-block structured cuboidal meshes (viz. Cartesian meshes), while employing homeomorphic mappings to accommodate time-dependent curvilinear domains. Motivated by the strengths of the Cartesian-mesh MPDATA, we develop a new formulation in an arbitrary finite-volume framework with a fully unstructured polyhedral hybrid mesh. In MPDATA, as in any Taylor-series based integration method for PDE, the choice of data structure has a pronounced impact on the technical details of the algorithm. Aiming at a broad range of applications with a large number of control-volume cells, we select a general, compact and computationally efficient, edge-based data structure. This facilitates the use of MPDATA for problems involving complex geometries and/or inhomogeneous anisotropic flows where mesh adaptivity is advantageous. In this paper, we describe the theory and implementation of the basic finite-volume MPDATA, and document extensions important for applications: a fully monotone scheme, diffusion scheme, and generalization to complete flow solvers. Theoretical discussions are illustrated with benchmark results in two and three spatial dimensions.

Keywords: Nonoscillatory advection schemes; Finite volume methods; Unstructured meshes

1. Introduction

MPDATA (multidimensional positive definite advection transport algorithm [22,23]) is a family of finite-difference approximations to the advective terms in the conservative (flux) formulation of fluid equations. In general, MPDATA is akin to the dissipative Lax–Wendroff schemes (also known as Taylor–Galerkin in

* Corresponding author.
E-mail addresses: smolar@ucar.edu (P.K. Smolarkiewicz), j.m.szmelter@cranfield.ac.uk (J. Szmelter).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.12.021

the finite-volume/finite-element literature) that can be derived from the first-order-accurate upwind algorithm (alias donor-cell) by adding on the r.h.s. the negative of the space-centered representation of the first-order error. In MPDATA, however, the compensation of the leading truncation error of the upwind scheme is nonlinear. It is achieved through the iterative application of the upwind differencing, where in the second and following iterations the leading truncation error terms (of the upwind scheme) are cast into the form of advective fluxes, defined as products of the current solution iterate and a suitable velocity field. The resulting algorithm is second-order accurate, conservative, fully multidimensional (i.e., free of the splitting errors), and computationally efficient; yet it maintains the signature properties of upwind differencing such as the strict preservation of sign of the transported field and relatively small phase-error.

The theoretical foundation of MPDATA – the modified equation approach – facilitates generalizations of the scheme to transport problems beyond elementary advection. In the early eighties, the algorithm was invented as an inexpensive alternative to flux-limited schemes for evaluating the advection of nonnegative thermodynamic variables (such as water substance) in atmospheric cloud models. Since then, a variety of options have been documented that extend MPDATA to full monotonicity preservation, to third-order accuracy, and to variable sign fields (such as momentum). MPDATA was generalized to a complete fluid solver in the early nineties [26]. In analyzing the truncation error of approximations to the momentum equation, one finds error terms that depend on the interaction of advection with the forcing terms, including the pressure gradient terms. Many implementations of nonoscillatory algorithms treat advection separately from the forcings, leaving this error uncompensated, thereby reducing the order of accuracy of the solution and potentially leading to oscillations and even instability [28]. In MPDATA, this error is compensated by effectively integrating the forcing terms along a flow trajectory rather than at a grid point. A comprehensive review of MPDATA, including both the underlying concepts and the details of implementation, can be found in [30]. The overview of MPDATA as a general, nonoscillatory approach for simulating geophysical flows – viz. high Reynolds number and low Mach number flows – on micro-to-planetary scales has been presented in [33].

Recently, MPDATA has attracted attention in the context of monotonically integrated large eddy simulations (MILES), as a high Reynolds number fluid solver with implicit turbulence modeling capability [12,14,4]. Unlike most nonoscillatory methods, MPDATA is based directly on the *convexity* of upwind advection – i.e., numerical solutions remain bounded by surrounding local values from the preceding time step, given a solenoidal advecting flow and adequately limited temporal increment[1] – rather than on the idea of flux limiting. Iterative application of upwinding in MPDATA greatly simplifies the task of designing higher-order schemes without the necessity of knowing the details of the resulting truncation error. Since the errors are cast in the form of the advective fluxes, the convexity of upwinding warrants their compensation while preserving sign of the transported field. In consequence, the linear computational stability of the first donor-cell step implies the nonlinear stability of the entire MPDATA advection, a property essential for simulations of turbulent flows. Since the upwind scheme filters high frequencies on the grid, and each subsequent step reverses the dissipative error of the preceding step, MPDATA bears an analogy to generalized similarity models, where an estimate of the full unfiltered Navier–Stokes velocity (which enters the subgrid-scale stress tensor) is obtained by an approximate inversion of the filtering operation, i.e., deconvolution; cf. [5].

In this paper, we report on a new development in MPDATA schemes. We derive, discuss, and test the finite-volume formulation of MPDATA with an edge-based data structure and arbitrary hybrid mesh. Although MPDATA has proven successful in simulations of geophysical flows using single block, structured, topologically rectangular meshes employing a continuous time-dependent curvilinear mapping approach [17,36], its potential for finite-volume approximations has been realized only recently. Bacon et al. [2] implemented basic MPDATA advection in the multiscale environmental model OMEGA for operational forecast of weather and pollutant dispersion. Independently, Margolin and Shashkov [13] drew

[1] For arbitrary flow a weaker condition of sign preservation can be assured.

inspiration from the MPDATA approach to develop a second-order sign-preserving conservative interpolation for two-dimensional ALE grids. Our goal is to assist the reader in developing MPDATA methods for fluids on arbitrary meshes. In MPDATA, as in any Taylor-series based integration method for differential equations, the choice of data structure has a pronounced impact on the details of the algorithm. Since our aim is a broad range of applications involving complex geometries and inhomogeneous flows, we have elected a general, compact and computationally efficient edge-based data structure. This contrasts with the approach adopted in [2], where focus on meteorological applications dictated an unstructured-mesh discretization only in the horizontal, with cell-centered and face-centered control-volume staggering of scalar and vector dependent variables, respectively. A discussion of the computational implications of various data structures can be found in [16].

The remainder of the paper is organized as follows. In Section 2, we derive the basic sign-preserving finite-volume MPDATA in abstraction from any particular data structure, while pointing out the differences and parallels to the standard finite-difference formulation. The latter is worthwhile because our derivation of the finite-volume formulation reveals new aspects of MPDATA important (potentially) for its finite-difference counterpart. In Section 3, we specify the algorithm using the edge-based median-dual finite-volume approach, optimal in light of the preceding analysis. The computational stability and accuracy of the basic scheme are evaluated in Section 4. In Section 5, we develop options of MPDATA important for applications: (i) a fully monotone scheme, in the spirit of the flux corrected transport (FCT); (ii) the diffusion scheme; and (iii) an extension of the algorithm to systems of inhomogeneous transport equations (viz., complete flow solvers). Section 6 concludes the paper. Proofs of stability and convexity for finite-volume upwinding are supplemented in Appendix A.

2. Finite-volume MPDATA; basic scheme

Here, we are concerned with the elementary advection equation:

$$\frac{\partial \Psi}{\partial t} = -\nabla \cdot (\mathbf{v}\Psi), \tag{1}$$

where $\Psi = \Psi(\mathbf{x},t)$ is a nondiffusive scalar field assumed nonnegative at $t = 0$, and the $\mathbf{v} = \mathbf{v}(\mathbf{x})$ is a prescribed flow assumed stationary. The adopted assumptions merely simplify the presentation; extensions to nonnegative scalar fields and unsteady flows – discussed later in this paper – directly follow the developments for finite-difference MPDATA, cf. [30]. Similarly, we shall disregard the variability of an unstructured grid in time – an assumption easy to relax following our finite-difference experience [33]. Integrating (1) over the volume of an arbitrary cell – while employing the Gauss divergence theorem $\int_\Omega \nabla \cdot (\mathbf{v}\Psi) = \int_{\partial\Omega} \Psi\, \mathbf{v} \cdot \mathbf{n}$ – results in

$$\Psi_i^{n+1} = \Psi_i^n - \frac{\delta t}{\mathscr{V}_i} \sum_{j=1}^{l(i)} F_j^{\perp} S_j. \tag{2}$$

Fig. 1 shows a face of an arbitrary computational cell containing vertex i, together with the normal and the *edge* connecting vertex i with one of its immediate neighbors j; there are $l(i)$ edges connecting the vertex i with its neighbors. S_j refers both to the face per se and its surface area. Eq. (2) is exact, given Ψ_i^{n+1} and Ψ_i^n are interpreted as the mean values of Ψ within the volume $\mathscr{V}_i$ of the cell containing vertex i; while $F_j^{\perp}$ is interpreted as the mean normal flux through the cell face S_j averaged over temporal increment δt. The approximation begins with specifying fluxes $F_j^{\perp}$ in terms of data available on the grid. The finite-volume first-order-accurate upwind scheme assumes the form

$$F_j^{\perp} = [v_j^{\perp}]^+ \Psi_i^n + [v_j^{\perp}]^- \Psi_j^n, \tag{3}$$

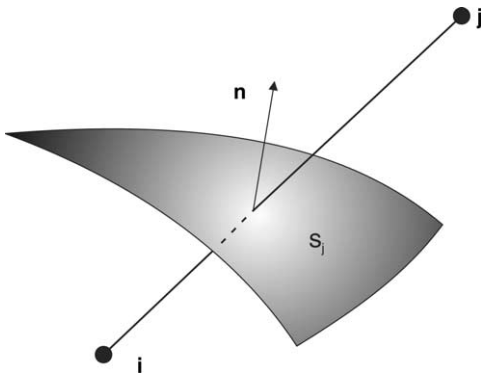

Fig. 1. Schematic of an edge piercing an arbitrary cell-face.

where

$$[v]^+ := 0.5(v + |v|), \quad [v]^- := 0.5(v - |v|), \tag{4}$$

and normal velocity $v^\perp := \mathbf{v} \cdot \mathbf{n}$ is evaluated at the face S_j.[2] The nonnegative/nonpositive parts of $v_j^\perp$ always coincide with outflow/inflow from the ith cell.

It is easy to show (see Appendix A) that

$$\forall_i \quad \sum_{j=1}^{l(i)} [U_j]^+ \leqslant 1, \quad U_j := \frac{v_j^\perp S_j \delta t}{\mathscr{V}_i} \tag{5}$$

suffices for the "positivity" of the solutions – i.e., consistent with the properties of the governing PDE (1), numerical advection in (2), (3) preserves the nonnegative/nonpositive character of the transported fields. The positivity of flux-form algorithms implies *nonlinear stability*, whereas for solenoidal flows

$$\frac{1}{\mathscr{V}_i} \sum_{j=1}^{l(i)} v_j^\perp S_j = 0, \tag{6}$$

(5) also assures the convexity of the solutions, i.e.,

$$\min_{j=1,l(i)} (\Psi_i^n, \Psi_j^n) \leqslant \Psi_i^{n+1} \leqslant \max_{j=1,l(i)} (\Psi_i^n, \Psi_j^n). \tag{7}$$

These apparent strengths of the scheme are offset, however, by the notorious numerical viscosity.

The key idea of MPDATA [22,23] is to compensate for the truncation error of the upwind scheme by reusing the same upwind algorithm but with a pseudo velocity defined based on the leading (dissipative) truncation error of the first step. By construction, this leads to a finite-difference scheme that is second-order accurate, yet sign-preserving for arbitrary flows.

In order to assess the leading truncation error of the finite-volume upwind differencing, we shall adapt the modified-equation procedure familiar from finite-difference MPDATA. The idea is to expand all discrete data into a Taylor series in time and space, and then to represent higher-order temporal derivatives in terms of spatial derivatives, to derive a continuous equation (approximated effectively by the numerical scheme at hand) for improving the approximation. We begin by expanding Ψ_i and Ψ_j that enter the definition of the upwind flux in (3) about a point 's_j' along the edge connecting vertices i and j, where the edge intersects the cell face S_j:

[2] We use := to mean *defined as*, to distinguish from ≡ (*identically*).

$$\begin{aligned}\Psi_i &= \Psi|_{s_j} + \left.\frac{\partial\Psi}{\partial r}\right|_{s_j}(r_i - r_{s_j}) + \mathcal{O}(\delta r^2),\\ \Psi_j &= \Psi|_{s_j} + \left.\frac{\partial\Psi}{\partial r}\right|_{s_j}(r_j - r_{s_j}) + \mathcal{O}(\delta r^2)\end{aligned} \tag{8}$$

with r referring to the parametric description of the edge $r(\lambda) = r_i + \lambda(r_j - r_i)$; $\lambda \in [0,1]$. Now, implementing (8) in the definition of the upwind flux (3), and rearranging the terms, results in

$$F_j^{\perp} = v_j^{\perp}\Psi|_{s_j}^{n} + 0.5|v_j^{\perp}|\left.\frac{\partial\Psi}{\partial r}\right|_{s_j}^{n}(r_i - r_j) + 0.5v_j^{\perp}\left.\frac{\partial\Psi}{\partial r}\right|_{s_j}^{n}(r_i - 2r_{s_j} + r_j) + \mathcal{O}(\delta r^2) \tag{9}$$

which reveals the explicit form of the $\mathcal{O}(\delta r)$ error in the approximation (2), (3). In order to see the $\mathcal{O}(\delta t)$ error, it suffices to expand Ψ^n about $t^{n+1/2}$

$$\Psi|_{s_j}^{n} = \Psi|_{s_j}^{n+1/2} - 0.5\left.\frac{\partial\Psi}{\partial t}\right|_{s_j}^{n+1/2}\delta t + \mathcal{O}(\delta t^2), \tag{10}$$

only in the first term on the r.h.s. of (9). Expanding spatial derivatives (about $t^{n+1/2}$) in the remaining terms is unnecessary, as it would lead to $\mathcal{O}(\delta t\delta r) \sim \mathcal{O}(\delta r^2)$ errors.

For preserving the explicitly forward-in-time (as opposed to centered-in-time) character of the approximation (2), (3), it is important to express the temporal derivative in (10) in terms of spatial differencing to $\mathcal{O}(\delta t, \delta r)$ at least. Although written in integral form, (2), (3) approximates (1) to $\mathcal{O}(\delta r)$ essentially by design.[3] Expanding (1) as

$$\frac{\partial\Psi}{\partial t} = -\mathbf{v}\cdot\nabla\Psi - \Psi\nabla\cdot\mathbf{v}, \tag{11}$$

and replacing the temporal derivative in (10) with the r.h.s. of (11), and then substituting the resulting relation in (9) for $\Psi|_{s_j}^{n}$ in the first term on the r.h.s. results in

$$\begin{aligned}F_j^{\perp} &= v_j^{\perp}\Psi|_{s_j}^{n+1/2} + Error,\\ Error &= -0.5|v_j^{\perp}|\left.\frac{\partial\Psi}{\partial r}\right|_{s_j}^{*}(r_j - r_i) + 0.5v_j^{\perp}\left.\frac{\partial\Psi}{\partial r}\right|_{s_j}^{*}(r_i - 2r_{s_j} + r_j) + 0.5\delta t v_j^{\perp}(\mathbf{v}\nabla\Psi)|_{s_j}^{*}\\ &\quad + 0.5\delta t v_j^{\perp}(\Psi\nabla\cdot\mathbf{v})|_{s_j}^{*} + \mathcal{O}(\delta r^2, \delta t^2, \delta t\delta r).\end{aligned} \tag{12}$$

The asterisk in lieu of the temporal level in the *Error* symbolizes either n, $n + 1/2$, or $n + 1$, as any of these temporal positions can be considered without affecting the form or the order of *Error*. The result in (12) states that the upwind flux in (3) can be decomposed into a time-centered flux through the face and the $\mathcal{O}(\delta r, \delta t)$ flux of a predominantly Fickian character.

As with finite differences, finite-volume MPDATA consists of two upwind (2), (3) iterations. In the first iteration, the input field and flow velocity, Ψ and $\mathbf{v}$, are those from the preceding time step t^n. In the second (corrective) iteration, the input field Ψ is the result of the preceding upwind iteration and the pseudo velocity $\tilde{v} := -\frac{1}{\Psi}Error$. In particular,

$$\widetilde{v}_j^{\perp} = 0.5|v_j^{\perp}|\left(\frac{1}{\Psi}\frac{\partial\Psi}{\partial r}\right)\Big|_{s_j}^{*}(r_j - r_i) - 0.5v_j^{\perp}\left(\frac{1}{\Psi}\frac{\partial\Psi}{\partial r}\right)\Big|_{s_j}^{*}(r_i - 2r_{s_j} + r_j) - 0.5\delta t v_j^{\perp}\left(\mathbf{v}\frac{1}{\Psi}\nabla\Psi\right)\Big|_{s_j}^{*} - 0.5\delta t v_j^{\perp}(\nabla\cdot\mathbf{v})|_{s_j}^{*} \tag{13}$$

[3] Because the linear size of the control volume is $\mathcal{O}(\delta r)$, and because the mean is bounded by the extrema, there exists a point within the control volume where the field value represents the mean; whereupon, the mean value equals the field value at any point within the control volume to $\mathcal{O}(\delta r)$, given singly differentiable fields.

with the asterisk now indicating the first-order estimate of the $n+1$ solution from the preceding upwind iteration. In principle, the entire process of estimating the residual error and compensating it can be continued, iteration after iteration, reducing the magnitude of the truncation error;[4] yet in practice one corrective iteration suffices for recovering the overall accuracy of time-space centered schemes.

The outlined procedure conveys the essence of the finite-volume MPDATA in abstraction from any particular data distribution and details of the discrete representation of differential operators. Although derived for the edges, the general form of the pseudo velocity (13) applies to many finite volume schemes with various cell arrangements. Here, (13) deserves a comment because it offers much guidance on how to design effective implementations, and because it involves a few subtleties compared to the forms previously published in the context of finite differences.

The first term on the r.h.s. of (13) is well known from finite-difference theory, and is straightforward to approximate on unstructured meshes. The second term is new. It depends on the mesh skewness, and vanishes when the cell face is at the midpoint of the edge. Note that as r_{s_j} approaches r_j or r_i, the second and third terms on the r.h.s. of (9) combine into a single Fickian flux $\sim [v_j^{\perp}]^{\pm}$, respectively. This is intuitively correct, as in these limits the corresponding donor-cell flux in (3) is already precise with respect to the location of Ψ. This 'mesh-skewness' term can be set to zero with adequate discretization – e.g. using the median dual finite volume approach, discussed in the next section. The third term $\sim \mathbf{v}\Psi^{-1}\,\nabla\Psi$ in (13) is the most cumbersome. On an orthogonal mesh it becomes naturally decomposed into convective derivatives normal and tangential to cell faces. In a general finite-volume framework, such a decomposition requires an additional effort, unnecessary for the schemes with single corrective iteration. Following Bacon et al. [2], here we shall evaluate the entire convective derivative in terms of Cartesian components. For the sake of computational stability, we shall extrapolate from the experience with finite-difference formulations and assure that the denominators and numerators in approximations to all $\sim\Psi^{-1}\partial\Psi$ factors in (13) use the same elements. This allows for exploiting the elementary boundedness property $|\sum \pm\Psi/\sum\Psi| \leqslant 1$ of positive-definite scalar fields. Finally, approximating the last term on the r.h.s. of (13) is straightforward, because of its independence on Ψ derivatives. In the solenoidal flows, this last term vanishes identically in the first corrective iteration of MPDATA, whereas in subsequent iterations it enters as $\mathcal{O}(\delta t^2)$ and is thus negligible [23]. In arbitrary flows, it may still vanish identically given Ψ represents a mixing ratio (a fluid property per unit of mass) as oppose to a density type variable. Then, $\mathscr{V}_i$ in (2) acquires the sense of a control mass, and the velocities in the subsequent formulae become momenta. In consequence, the fluid density ρ enters explicitly in pseudo-velocity formulae, but the $\sim\nabla\cdot\mathbf{v}$ term becomes multiplicative of the mass-continuity equation (i.e., zero). These results are important for anelastic systems popular in geophysical and astrophysical applications. Since they are independent of spatial discretization, all related finite-difference developments (cf. [33] for a discussion) translate to a finite-volume framework.

3. Implementation

The analysis of the preceding section indicates that some specifications of control volumes may be optimal for MPDATA because they simplify the leading truncation error of the upwind scheme and reduce the computational effort associated with evaluating the anti-truncation-error pseudo velocity (13). In particular, the median-dual finite-volume approach (see [3] for a comprehensive discussion) constructs the control volume associated with the vertex i by joining the centers of polyhedra cells (encompassed, in 3D, by consecutive edges originating in i) and midpoints of edges surrounding the vertex i – thereby allowing for a mean curvature of the face and canceling out the mesh skewness error in (13). Fig. 2 shows a schematic of the resulting cell face in 3D, and a more complete display of the mesh in 2D is shown in Fig. 3.

[4] Reducing the order of the error requires third-order analysis to begin with, cf. [11].

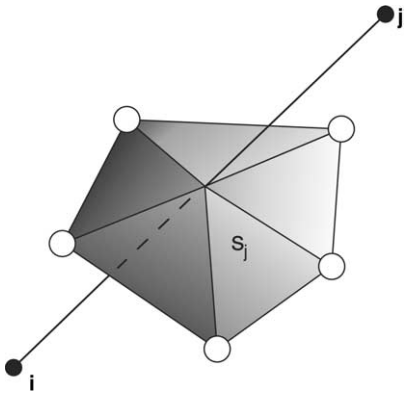

Fig. 2. The edge-based median-dual approach. The edge connecting vertices i and j pierces the face S_j of a 3D computational cell surrounding vertex i. Open circles represent centers of polyhedra cells referred to in the text.

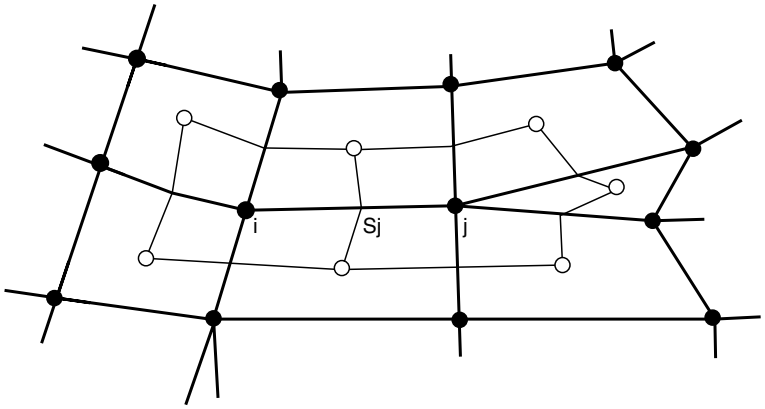

Fig. 3. The edge-based, median-dual approach in 2D. Symbolic and labeling conventions are the same as in Fig. 2.

Having defined the mesh, all geometric elements such as cell volumes, cell face areas, and normals are evaluated from elementary vector calculus. Hereafter, $\mathbf{S}_j \equiv S_j \mathbf{n}_j$ symbolizes the oriented surface element with $\mathbf{n}_j$ denoting the normal. Since normal fluxes in (2) are always proportional to normal velocities $\mathbf{v}_j \cdot \mathbf{n}_j$, the face surface area $S_j = \|\mathbf{S}_j\|$ is absorbed in the definition of the normal velocity $v_j^{\perp} := \mathbf{v}_j \cdot \mathbf{S}_j$ while setting $S_j \equiv 1$ in all relevant formulae. Consequently, the surface-area-weighted advective velocities normal to the cell face S_j are evaluated at cell faces as

$$v_j^{\perp} = \mathbf{S}_j \cdot 0.5[\mathbf{v}_i + \mathbf{v}_j], \tag{14}$$

and a flow chart of the basic MPDATA can be summarized as follows.

- Step 1: Calculate the upwind fluxes normal to the cell face S_j using formulae (3), (4) with advective velocities defined in (14).
- Step 2: Update Ψ, according to (2), using the upwind fluxes evaluated in the preceding step

$$\Psi_i^* = \Psi_i^n - \frac{\delta t}{\mathcal{V}_i} \cdot \sum_{j=1}^{l(i)} F_j^{\perp}. \tag{15}$$

- Step 3: Calculate the surface-weighted antidiffusive pseudo velocity at the face according to (13), using the field value updated in the preceding step and $v_j^\perp$ from (14)

$$\hat{v}_j^\perp = |v_j^\perp| \frac{|\Psi_j^*| - |\Psi_i^*|}{|\Psi_j^*| + |\Psi_i^*| + \varepsilon} - \frac{\delta t}{2} v_j^\perp \left(\mathbf{v} \cdot \frac{\nabla |\Psi^*|}{\overline{|\Psi^*|}} + \nabla \cdot \mathbf{v} \right)_{S_j}, \tag{16}$$

where ε denotes a small constant, e.g. 10^{-10}, to assure that the denominator does not vanish where $\Psi_j^* = \Psi_i^* = 0$.[5] The factor in brackets inside the second term on the r.h.s. of (16) has been written symbolically for conciseness. Its approximation follows the discussion at the end of Section 2.

In finite-volume framework, evaluation of partial derivatives $\partial\Phi/\partial x^I$ – where superscript I refers to Cartesian coordinates x^I – can always be interpreted in terms of Gauss' theorem, by representing the derivative as the divergence of the augmented vector $\Phi\nabla x^I$. Depending upon the specification of an auxiliary control volume which surrounds the edge midpoint, a number of approximations can be found of various degrees of complexity and associated computational effort. Keeping in mind the efficacy of the edge-based formulation, here we consider for the auxiliary control volume the sum of the control volumes of the dual mesh surrounding vertices i and j, see Fig. 3, such that

$$\left(\frac{\partial \Phi}{\partial x^I}\right)_j = \frac{1}{\overline{\mathscr{V}}_j} \left(\sum_{m=1}^{l(i)} \overline{\Phi}^{j,m} S_m^I + \sum_{m'=1}^{l(j)} \overline{\Phi}^{j,m'} S_{m'}^I \right), \quad \overline{\mathscr{V}}_j \equiv \mathscr{V}_i + \mathscr{V}_j, \tag{17}$$

where $\Phi \equiv |\Psi^*|$, $\overline{\Phi}^{j,m} \equiv 0.5(\Phi_i + \Phi_m)$ and S_m^I denotes the Ith area component of the oriented surface element at the mth edge.[6] The associated $\overline{\Phi}$ – a denominator in the second term on the r.h.s. of (16) – is evaluated as a surface-area weighted average from the same control volume

$$\begin{aligned} \overline{\Phi}_j &= \frac{1}{\overline{\mathscr{S}}_j} \left(\sum_{m=1}^{l(i)} \overline{\Phi}^{j,m} |S_m^I| + \sum_{m'=1}^{l(j)} \overline{\Phi}^{j,m'} |S_{m'}^I| + \varepsilon \right), \\ \overline{\mathscr{S}}_j &\equiv \sum_{m=1}^{l(i)} |S_m^I| + \sum_{m'=1}^{l(j)} |S_{m'}^I|. \end{aligned} \tag{18}$$

Similarly, when required, the flow divergence appearing inside the bracket in (16) is evaluated as

$$(\nabla \cdot \mathbf{v})_j = \frac{1}{\overline{\mathscr{V}}_j} \left(\sum_{m=1}^{l(i)} v_m^\perp + \sum_{m'=1}^{l(j)} v_{m'}^\perp \right). \tag{19}$$

- Step 4: Calculate the normal (to the cell face) upwind fluxes as in step 1, but using Ψ^* and $\hat{v}_j^\perp$ from steps 2 and 3, respectively,

$$F_j^* = [\hat{v}_j^\perp]^+ \Psi_i^* + [\hat{v}_j^\perp]^- \Psi_j^*. \tag{20}$$

- Step 5: Updating the field by reusing the upwind scheme

$$\Psi_i^{n+1} = \Psi_i^* - \frac{\delta t}{\mathscr{V}_i} \sum_{j=1}^{l(i)} F_j^* \tag{21}$$

completes the basic MPDATA scheme.

[5] The use of absolute values in (16) extends validity of the scheme to variable-sign fields. It relies on $\Psi^{-1}\partial\Psi/\partial\xi \equiv |\Psi|^{-1}\partial|\Psi|/\partial\xi$, an identity valid for arbitrary Ψ with accuracy to topological sets of measure zero; cf. [30] for a discussion.

[6] We have also tested an alternate, potentially more accurate scheme with a smaller stencil but found no significant differences.

4. Consistency, stability, and accuracy of finite-volume MPDATA

The finite-volume algorithm described in the preceding section is constructed from the classical upwind scheme that is consistent, conditionally stable, and first-order accurate. These properties, together with the algorithm's design, predetermine the consistency, stability, and accuracy of MPDATA [23]. In particular, since the antidiffusive velocity (16) tends to zero as δr, $\delta t \to 0$, the consistency of MPDATA is implied by that of the finite-volume upwind. Similarly, since the corrective upwind Step 5 compensates the first-order leading error of the preceding upwind Step 2, with accuracy to the first order at least, the uncompensated portion of the upwind error remains at second order. The latter suffices for rising the formal accuracy order of finite-difference upwinding – the second- and third-order asymptotic convergence rates of various MPDATA options have been shown for Cartesian meshes, cf. [30] and references therein. For arbitrary meshes, however, this is not necessarily the case, since the formal accuracy of the centered scheme – a target of the MPDATA derivation; cf. Eq. (12) – is not mesh independent [18]. Notwithstanding, Bacon et al. [2] demonstrated the second-order convergence of their cell-centered unstructured MPDATA using a standard "rotating-cone" benchmark while invoking intensive local mesh refinement. Later in this section we shall employ similar tests to illustrate the present discussion. In particular, we shall demonstrate the second- and a near second-order convergence rates of finite-volume MPDATA on, respectively, regular and strongly skewed unstructured grids.

The stability of MPDATA also follows that of the upwind (discussed thoroughly in Appendix A) but it deserves an elaboration, since there are a few subtleties involved. In 1D, (16) together with $|U_j| \leqslant 1$ – viz. the stability and positivity of the original upwind scheme – imply

$$|\hat{U}_j| \leqslant |U_j| - (U_j)^2 \leqslant |U_j|, \tag{22}$$

whereupon the stability of upwind warrants the stability of MPDATA. A similar occurs for multidimensional problems, but the presence of cross-derivatives resulting from the second term on the r.h.s. of (16) makes formal proof difficult. In the standard Cartesian-mesh formulation, this task is facilitated somewhat by the accompanying decomposition of the cross-derivative term into the sum of convective derivatives normal and tangential to cell faces. In [23], the stability of upwind implying that of MPDATA has been proven, but with a caveat that the time step used is smaller than that allowed for the upwind alone; correspondingly, by the factor $2^{-1/2}$ and 2^{-1}, in 2 and 3 spatial dimensions. This particular time step requirement was a result of assuming the worst case scenario where the velocity components flip sign across the cell. Since the latter is a rare event in CFD applications, the heuristic limit recommended for all structured-mesh MPDATA extensions has been that valid for the upwind scheme. For the unstructured-mesh formulation the problem appears more complicated, and a formal proof eludes our effort. However, consider the following argument.

Multiplying (16) by $\delta t/\mathscr{V}_i$ – recall that $\hat{v}_j^{\perp}$ already includes S_j – leads to the antidiffusive local Courant number

$$\sum_{j=1}^{l(i)} \left[\hat{U}_j\right]^+ = \sum_{j=1}^{l(i)} |U_j| \left[\mathscr{A}_j - \frac{1}{2}\operatorname{sgn}(U_j)\mathbf{V}_j \cdot \mathbf{B}_j\right]^+, \tag{23}$$

where $\mathbf{V}$ is the vector of local Courant numbers with components $\delta t v_j^l \overline{\mathscr{S}}_j / \overline{\mathscr{V}}_j$, and $|\mathscr{A}_j|$ as well as $|B_j^l|$ are all bounded by unity. Since the expression in the square brackets on the r.h.s. of (23) is $\mathscr{O}(1)$, one can always find a sufficiently small δt such that the stability of the original upwind implies the stability of the entire scheme. Semi-heuristic considerations suggest half of the time step allowed for the upwind scheme, however standard conditions (5) or (A.10) appear to suffice in practice. In our experience, the results of MPDATA

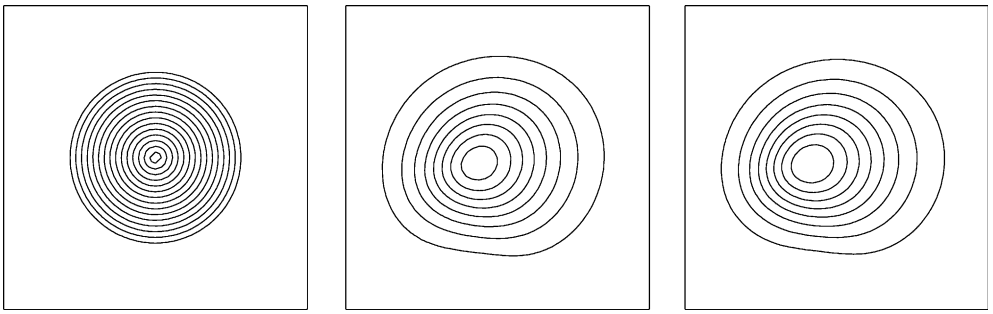

Fig. 4. Isolines of cone advected through six rotations around the center of the lower frame (only a quarter of the domain is shown). The contour interval is 0.25, and the zero contour line is not shown. Left plate, the analytic solution; center plate, the FV MPDATA on a regular square-mesh; right plate, as in the center but for a triangular mesh.

advection (for both structured and unstructured meshes) are only weakly sensitive to the fine details of the cross-derivative term approximations. This suggests that it might be worthwhile to seek an approximation optimal for the stability while facilitating a proof.

Herein, we corroborate the antecedent arguments, and illustrate the performance of the finite-volume MPDATA, using a standard solid-body rotation test [22,23,11,30]. A cone of base radius 15 and height 4, centered initially at $(75, 50)$, is rotating counter clockwise around the center of a $[0, 100] \times [0, 100]$ domain with the angular velocity $\omega = 0.1$. Fig. 4 displays the isolines of the exact result and two finite-volume (FV) MPDATA solutions after 6 rotations. The solution in the central plate uses 10^4 squared cells, whereas the solution in the right plate uses a triangular grid with a similar distribution of points.[7] Except for the time step for triangular meshes, all parameters of the test and of the display are identical to those in Fig. 1 of [30] to facilitate further comparisons. The accuracy of the results displayed is quantified in Table 1, where the corresponding values for the classical upwind and centered-in-time-and-space (leapfrog) schemes are included for reference. The corresponding finite-difference (FD) results are also included.

We want to call the reader's attention to a few items in Table 1. First, for the regular square mesh our edge-based unstructured-mesh finite-volume MPDATA reproduces the finite-difference Cartesian-mesh result, thereby documenting the reflexivity of the approach. Second, MPDATA strictly preserves the sign of the transported field, in clear contrast to the leapfrog scheme with dispersive undershoots reaching $\sim$20% of the solution's amplitude. Third, while the amplitude of the MPDATA results is about 2/3 of the nondissipative leapfrog solutions, the overall accuracy is about 20% better. Fourth, for the triangular mesh the results are presented for two different time steps. This is to illustrate that the precise equivalence of the test for Cartesian and automatically generated unstructured meshes is not possible. The "MPDATA FV triangles $\delta t = 0.1$" calculation matches the time step from the benchmark test in [30], and produces results similar to squares. Although this calculation remained stable, $\delta t = 0.1$ violates the stability condition (5) in a few cells near the domain boundaries – the maximal Courant number based on the advective velocity (14) was 1.64, still twice as large as the maximal Courant number for the corrective iteration 0.81. The choice of $\delta t = 0.061$ satisfies (5) rigorously, giving the maximal Courant number of the original upwind iteration 0.999, while resulting in a maximal Courant number 0.481 for the corrective iteration. A small sensitivity to the change of the time step (e.g., 4% reduction of Max in response to 40% time step reduction), common to a majority of advective solvers, would be further reduced in a third-order type MPDATA scheme [11,30]. All subsequent calculations for triangular meshes in Table 1 were conducted with $\delta t = 0.061$.

[7] The triangular grid has been obtained from the advancing-front mesh generator, using the background spacing = 1, cf. [10].

Table 1
Error norms for solid-body rotation test using finite-volume and finite-difference versions of MPDATA; the classical upwind and leapfrog schemes are included for reference

Scheme	Max	Min	L_∞	L_2
MPDATA FD	2.18	0.	2.00	0.47×10^{-3}
MPDATA FV squares	2.18	0.	2.00	0.47×10^{-3}
MPDATA FV triangles, $\delta t = 0.1$	2.19	0.	1.92	0.47×10^{-3}
MPDATA FV triangles, $\delta t = 0.061$	2.08	0.	2.09	0.52×10^{-3}
Upwind FD	0.27	0.	3.76	1.21×10^{-3}
Upwind FV squares	0.28	0.	3.76	1.04×10^{-3}
Upwind FV triangles	0.25	0.	3.68	1.06×10^{-3}
Leapfrog FD	3.16	−0.62	1.68	0.62×10^{-3}
Leapfrog FV squares	3.11	−0.67	1.71	0.64×10^{-3}
Leapfrog FV triangles	3.11	−0.69	1.74	0.65×10^{-3}

The first column identifies the scheme used, and the remaining columns provide accuracy characteristics of the numerical solutions after 6 rotations of the cone. Second and third columns list the solution extrema (4 and 0 for the exact result). The fourth column provides the values of the maximal solution departure from the exact result, while the fifth column gives the rms errors.

Fig. 5 illustrates the uniform boundedness of corrective Courant numbers by those of the original upwind iteration. The MPDATA solution is displayed on an unstructured triangular grid (third entry in Table 1) together with superimposed contour lines of the local Courant numbers of the original and corrective upwind iterations – defined in (5) and (23), respectively. To optimize the readability of the display, all Courant-number values were premultiplied by a factor of 100. Because the values in the corrective iteration are uniformly smaller (a few times), they are displayed with a quarter of the contour interval used for the original-upwind Courant numbers. The actual extrema of the fields defined in (5) and (23) are 1.64 and 0.81, respectively; while on the average the respective Courant numbers are 0.48 and 0.046 with corresponding standard deviations of 0.19 and 0.07. Note that the uniform boundedness of the corrective Courant numbers holds, even though the sufficient stability condition for upwind has been locally violated.

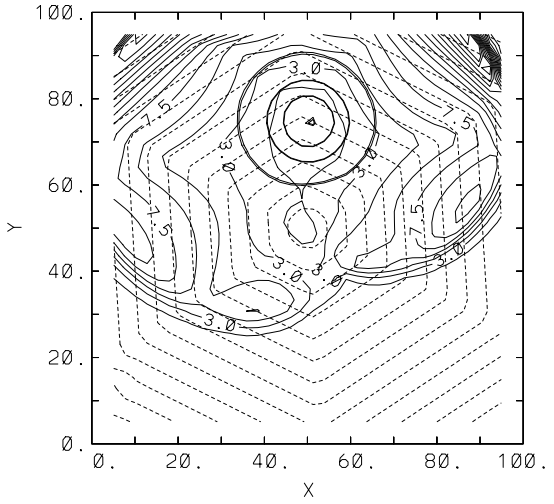

Fig. 5. Isolines of a cone advected through one rotation (heavy solid lines) with contour increment 1, starting at the value 0.25. Superimposed are contours of the local Courant numbers (×100) from the original (dashed lines) and corrective (solid lines) upwind iterations. Original Courant numbers start at the value 6 in the center of the domain and increase outward, whereas corrective Courant numbers (labeled) start at 1.5. Both fields are displayed with contour intervals equal to their starting values.

A comprehensive study of the asymptotic convergence – in the spirit of [25] and [11] – investigates the convergence rate in function of both grid resolution and the Courant number. For unstructured meshes, it requires a 2D generalization and a forbidable computational effort. To illustrate the convergence property of finite-volume MPDATA, here we consider the following modification of the rotating-cone benchmark. The unstructured triangular mesh used in the examples above (the background spacing 1) is regenerated assuming the background spacing 2, 0.5, and 0.25. The cone is replaced by the axially symmetric Gaussian hill $\Psi(\mathbf{x}, t = 0) = A \exp(-0.5(\mathbf{x} - \mathbf{x}_o)/R_o)^2$, where $A = 4$, $R_o = 5$, and $\mathbf{x}_o = (48, 48)$; and the velocity field is constant $\mathbf{v} = (2.4, 2.4)$. The solution is advected over the distance $4\sqrt{2}$ on all meshes using, respectively, 2, 4, 8, and 16 time steps $\delta t = 2$, 1, 0.5, and $0.25\delta t_1$, to maintain a constant (approximately) Courant number.

Table 2 lists the standard norms of the solutions' departures from the analytic result. Both L_2 and L_1 are normalized to reflect the rms and the absolute value of the truncation error per cell and the unit of time [25,11]. Since all norms are equivalent in $\mathbf{R}^n$ (Corollary VII.1.2 in [15]), the asymptotic convergence rates should not depend on the norm selection. The latter as well as the second-order convergence are illustrated in the table.

In order to relate our finite-volume formulation to the finite-difference MPDATA benchmarks reported over years, we considered so far a fairly regular unstructured grid. In general, state-of-the-art mesh generators are designed to maximize the mesh quality (viz. maximize smoothness and minimize skewness). To substantiate further the theory of Section 2, we repeat the above-discussed convergence test on a purposely designed, low-quality, skewed mesh, Fig. 6.

Table 2
Convergence of finite-volume MPDATA on unstructured mesh

Background spacing	L_∞	L_2	L_1
2.00	2.95×10^{-2}	8.73×10^{-3}	1.83×10^{-3}
1.00	7.69×10^{-3}	2.22×10^{-3}	4.73×10^{-4}
0.50	1.93×10^{-3}	5.57×10^{-4}	1.19×10^{-4}
0.25	4.86×10^{-4}	1.43×10^{-4}	3.07×10^{-5}

The first column lists the background spacing of the mesh, the three remaining columns provide norms of the solutions' departures from the analytic result.

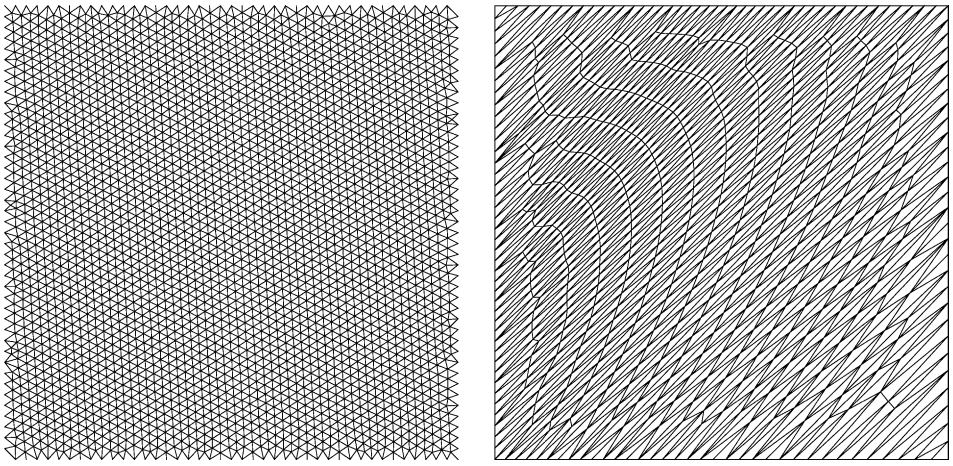

Fig. 6. Regular (left) and skewed (right) grids used in the convergence study. The coarsest meshes are shown.

Table 3
Convergence of finite-volume MPDATA on unstructured skewed mesh

δ_E	L_∞	L_2	L_1
3.99	3.58×10^{-1}	1.38×10^{-1}	2.86×10^{-2}
2.05	2.01×10^{-1}	6.68×10^{-2}	1.38×10^{-2}
1.03	8.51×10^{-2}	2.34×10^{-2}	4.78×10^{-3}
0.51	1.86×10^{-2}	5.14×10^{-3}	9.41×10^{-4}
0.26	3.23×10^{-2}	1.48×10^{-3}	2.52×10^{-4}

The first column lists the equivalent spacing measure, the three remaining columns provide norms of the solutions' departures from the analytic result.

Table 3 lists the error norms for the sequence of skewed meshes, characterized by the equivalent spacing measure $\delta_E := 100(N_{cells})^{-1/2}$. Each mesh is generated by halving the background spacing prescribed at every point of the background mesh used for generation of the coarser mesh. The starting skewed mesh is twice coarser than the regular mesh in Table 2. In effect, the asymptotic second-order convergence rate does not reveal until the third refinement. Without the mesh-quality control, at the fourth refinement the mesh becomes nearly singular in a number of points, whereupon the solution becomes linearly unstable. In consequence, the convergence degrades.

5. Extensions

5.1. Nonoscillatory option

The algorithm described in Steps 1–5 in Section 3 preserves the sign but not the monotonicity of the transported variables [23–25] and, in general, the solutions are not free of spurious extrema, Fig. 7. This is because the antidiffusive velocity (16) is not necessarily solenoidal, even for a solenoidal physical flow (6). In many studies of natural flows preservation of sign is adequate [27]. However, when required, MPDATA can be made fully monotone [25] by adapting the FCT formalism [37] to limit the pseudo veloc-

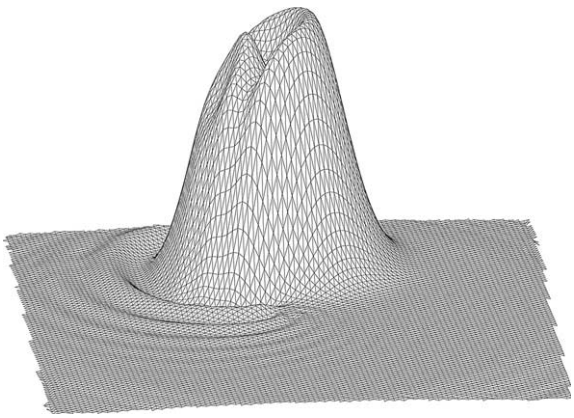

Fig. 7. MPDATA solution for the solid-body rotation from Section 4, but with the cone replaced by the cylinder placed on a large constant background.

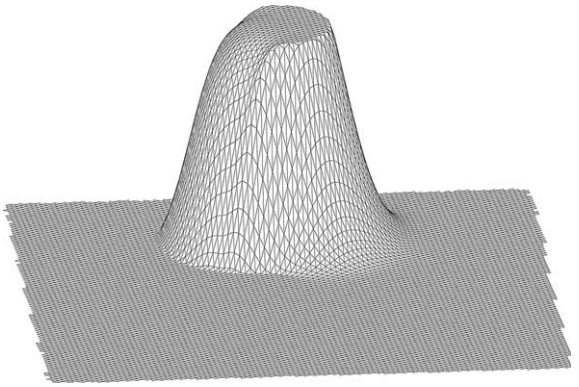

Fig. 8. As in Fig. 7, but for the nonoscillatory-option MPDATA solution.

ities in (16); cf. Fig. 8. It has been argued [25] that MPDATA is particularly well suited for this for several reasons. Firstly, the initial MPDATA iteration is the upwind scheme – i.e., a low-order monotone scheme commonly used as the reference in the FCT design. Secondly, assuring monotonicity of the subsequent iterations provides a higher-order accurate reference solution for the next iteration with the effect of improving the overall accuracy of the resulting FCT scheme. Thirdly, since all MPDATA iterations have similar low phase errors characteristic of the upwind scheme [24], the FCT procedure mixes solutions with consistent phase errors. This benefits the overall accuracy of the resulting FCT scheme (see Fig. 5 in [25] and the accompanying discussion).

The FCT extension for Cartesian-mesh MPDATA has been presented in [25] together with an algebraic theory of FCT limiting.[8] Since the entire article [25] has been devoted to a thorough exposition, here we only present a summary of final formulae, adapted for the edge-based finite-volume algorithm. Using the notation of Section 3, and referring to Fig. 3 for the geometry, the FCT-limited antidiffusive velocity (16) can be written compactly as

$$\begin{aligned}\widehat{\hat{v}_j^{\perp}} = {} & \left[\hat{v}_j^{\perp}\right]^{+}\left(\min(1,\beta_i^{\downarrow},\beta_j^{\uparrow})\left[\operatorname{sgn}(\Psi_i^{*})\right]^{+} + \min(1,\beta_i^{\uparrow},\beta_j^{\downarrow})\left[\operatorname{sgn}(-\Psi_i^{*})\right]^{+}\right) \\ & + \left[\hat{v}_j^{\perp}\right]^{-}\left(\min(1,\beta_i^{\uparrow},\beta_j^{\downarrow})\left[\operatorname{sgn}(\Psi_j^{*})\right]^{+} + \min(1,\beta_i^{\downarrow},\beta_j^{\uparrow})\left[\operatorname{sgn}(-\Psi_j^{*})\right]^{+}\right),\end{aligned} \tag{24}$$

where, for all i, the limiting coefficients $\beta_i^{\uparrow}$ and $\beta_i^{\downarrow}$ are

$$\beta_i^{\uparrow} = \frac{\Psi_i^{\mathrm{MAX}} - \Psi_i^{*}}{\frac{\delta t}{\mathscr{V}_i}\sum_{j=1}^{l(i)}[F_j^{*}]^{-} + \varepsilon}, \quad \beta_i^{\downarrow} = \frac{\Psi_i^{*} - \Psi_i^{\mathrm{MIN}}}{\frac{\delta t}{\mathscr{V}_i}\sum_{j=1}^{l(i)}[F_j^{*}]^{+} + \varepsilon}, \tag{25}$$

and the limiters Ψ_i^{MAX} and Ψ_i^{MIN} assume the form

$$\Psi_i^{\mathrm{MAX}} = \max_{j=1,l(i)}(\Psi_i^{n}, \Psi_j^{n}, \Psi_i^{*}, \Psi_j^{*}), \quad \Psi_i^{\mathrm{MIN}} = \min_{j=1,l(i)}(\Psi_i^{n}, \Psi_j^{n}, \Psi_i^{*}, \Psi_j^{*}). \tag{26}$$

Formulae (25) and (26) conform, respectively, to the formulae (19a,b) and (20a,b) from [25]. Furthermore, (24) extends the corresponding Eq. (18) of [25] – written explicitly only for nonnegative scalars – to fields of

[8] The algebraic formalism has proven useful for synchronous FCT where physical bounds imposed on functions of transported fields alter standard limiters for the individual fields [7,20].

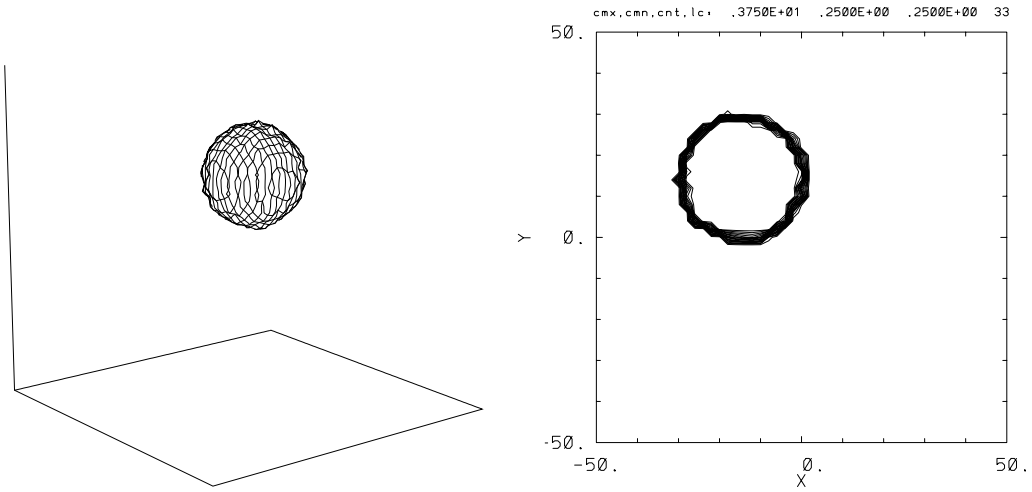

Fig. 9. Initial condition for a unstructured-mesh advection. The $\Psi = 1$ isosurface and contour plot in xy cross-section through (approximately) the center of the sphere are shown in left and right panel, respectively.

arbitrary sign, consistent with the convention of Section 3. Note that $[\mathrm{sgn}(\pm\ldots)]^+$ factors in (24) are merely logical switches, or *if* statements.[9]

Incorporating the nonoscillatory option in the algorithm in Steps 1–5 proceeds in two stages. Firstly, preceding Step 1, the n time-level contributions to the limiters (26) are evaluated and stored, to be completed after Step 2. Secondly, immediately after Step 4 limiting coefficients (25) are evaluated, then the antidiffusive velocities (16) are limited according to (24), and the corrective upwind fluxes are reevaluated as in Step 4 but with $\widehat{\hat{v}_j^{\perp}}$ in lieu of $\hat{v}_j^{\perp}$. Then, the solution of Step 5 is warranted to be locally bounded by the limiters, i.e., $\Psi_i^{\mathrm{MIN}} \leqslant \tilde{\Psi}_i^{n+1} \leqslant \Psi_i^{\mathrm{MAX}}$.

To substantiate the theoretical development, we extend the solid-body rotation benchmark to three spatial dimensions, and perform a series of tests with representative advection schemes.

A sphere with radius 15 and constant density 4, placed initially at $\mathbf{x}_o = (25, 75, 75)$ Fig. 9, rotates with angular velocity $\omega = 0.1 \times 3^{-1/2}(1,1,1)$ around a diagonal of the cuboidal domain $[0,100] \times [0,100] \times [0,100]$. The numerical solution uses an unstructured grid with 204,107 cells, and 1,486,871 edges. Similar to the 2D tests, the 3D grid has been obtained from the advancing-front mesh generator, but using *twice larger background spacing* = 2. Although the resulting average edge length (=2.1) is close to the specified background spacing, there are substantial departures from the grid homogeneity, as measured by the standard deviation of the edge length 0.37, and the extrema 3.8 and 0.8. With the time step $\delta t = 0.018 \cdot 2\pi$, the resulting maximal Courant number on the r.h.s. of (A.10) is 1.83 . For the uniform flow $\mathbf{v}_o = (1,1,1)$, the discrete velocity divergence (19) vanishes to the round-off error. Since the actual rotational flow is prescribed analytically, the maximal Courant-number divergence is 1.6×10^{-3}, only three orders of magnitude smaller than the Courant number itself. In order to generate graphic displays, the field values on the unstructured grid are interpolated linearly to Cartesian mesh with 51^3 grid points. The accurate values of relative accuracy and efficiency measures evaluated on the unstructured grid are listed in Table 4. All solutions are evaluated after $T = 10 \cdot 2\pi \approx 556\delta t$, i.e., one revolution of the sphere around the domain diagonal.

[9] The special cases of $\Psi_i^* = 0$ or $\Psi_j^* = 0$ are nonproblematic, since the corresponding upwind fluxes vanish, thus having no contribution to the formation of spurious extrema.

Table 4
Relative accuracy and efficiency measures for 3D finite-volume advection on unstructured grid

Scheme	Max	Min	L_∞	L_2	L_1	CPU
Upwind	1.5	0.0	3.4	6.1×10^{-3}	1.3×10^{-3}	1
Leapfrog	7.1	−2.6	4.4	5.3×10^{-3}	2.5×10^{-3}	1.1
Leapfrog-Trp.	7.1	−2.5	4.4	5.3×10^{-3}	2.5×10^{-3}	0.8
MPDATA$_\infty$	6.2	−1.2	3.0	3.7×10^{-3}	8.0×10^{-4}	1.9
Leapfrog-Trp. FCT	4.4	0.0	3.0	3.5×10^{-3}	5.4×10^{-4}	2.
MPDATA basic	4.8	0.0	2.9	4.1×10^{-3}	7.8×10^{-4}	4.3
MPDATA FCT	4.3	0.0	2.9	4.1×10^{-3}	7.8×10^{-4}	4.9
MPDATA$_\infty$ FCT	4.4	0.0	2.8	3.2×10^{-3}	4.1×10^{-4}	3.9

The first column identifies the scheme used, the following five columns provide norms of the solutions' departures from the analytic result, and the last column is the upwind-normalized approximate execution time.

Similar to Tables 2 and 3, Table 4 lists three standard norms of the solution error, and both L_2 and L_1 norms are normalized to reflect the rms and absolute value of the truncation error per cell and the unit of time. Inasmuch as L_2 better reflects the overall accuracy, the remaining measures tend to emphasize the amplitude and phase errors (viz. behavioral errors [19]). The approximate efficiency measure, with estimated error ±0.2, is the total CPU time normalized by the total CPU time of the upwind solution.

The first four schemes listed in Table 4 are all linear, included merely for reference. Although sign preserving and computationally efficient, the upwind advection is not of interest because of its low overall accuracy, whereas the leapfrog schemes (both standard and trapezoidal, [37]) are unattractive because of the dispersive oscillations; see Fig. 10 for illustration. The latter also concerns the "infinite-gage" variant of MPDATA – denoted as MPDATA$_\infty$ – the oscillatory Lax–Wendroff limit resulting from the linearization of the basic MPDATA around a large constant background; cf. Section 3.2 in [30] and references therein. The remaining four schemes are nonlinear. The three FCT schemes are formally monotone,[10] whereas the basic MPDATA is only sign-preserving, viz. monotone near "zeros".

The spread of the computational efficiency measure aside, there is a clear demarcation between the execution times of the linear and nonlinear algorithms. The nonlinear algorithms are roughly three to four times as computationally expensive as the linear schemes. Although the nonlinear schemes are more accurate overall, preserving the monotonicity or sign of transported variables appears cost ineffective for pure advection. However, in numerical simulation of fluids, the monotone or sign-preserving advection may not be an option but a necessary prerequisite of the solution's realizability (e.g., for reactive or multi-phase flows). While arguments can be made that the basic and FCT MPDATA are of superior, or at least of comparable, accuracy to the classical leapfrog-trapezoidal FCT scheme (Fig. 11), the monotone MPDATA$_\infty$ (Fig. 12) is the most accurate in all norms. The judgement of which scheme should be the method of choice in not straightforward, as in practice there are factors other than the standard norms and the execution time of the sole advection. Here we only note the superior symmetry of the standard MPDATA (Fig. 13), while postponing further discussion until Section 6.

5.2. Diffusion

The underlying idea of MPDATA is the formal equivalence between the diffusion and advection on a discrete mesh; see Section 3.2 in [24] for extended discussion. The simulation of diffusive transport as an advection problem serves a double purpose. First, it documents the flexibility of the MPDATA approach.

[10] The overshoots with respect to the initial amplitude 4 are cumulative effect due to the residual flow divergence ($\sim\Psi\nabla\cdot\mathbf{v}$), and dissapear for strictly incompressible flows (e.g., for constant $\mathbf{v}$) – consistent with the FCT formalism that defines monotonicity relatively to a smooth low-order solution (here upwind).

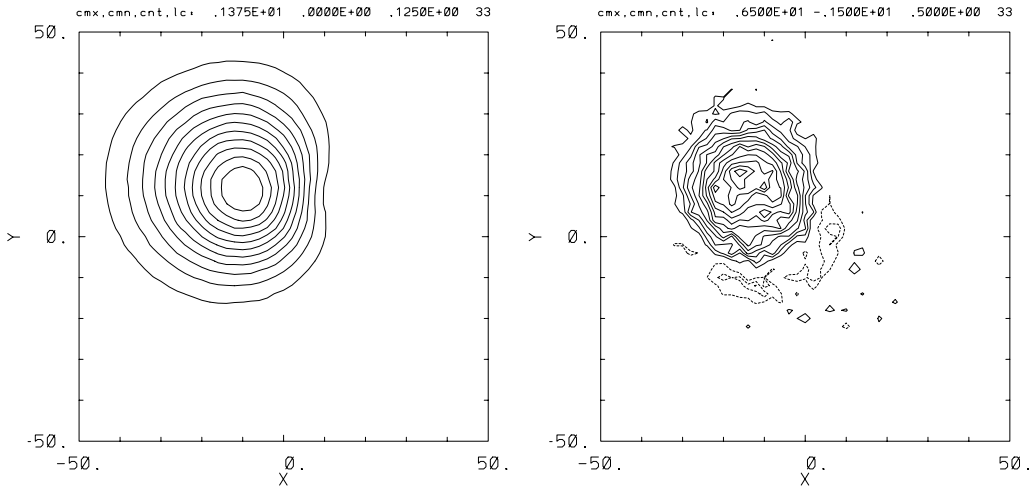

Fig. 10. Contour plots as in Fig. 9 but for the upwind and leapfrog solutions after one revolution of the sphere around a diagonal of the domain. Positive and negative values are displayed with solid and dashed lines, respectively. Contour extrema and interval are indicated above the frame, and zero contours are not shown.

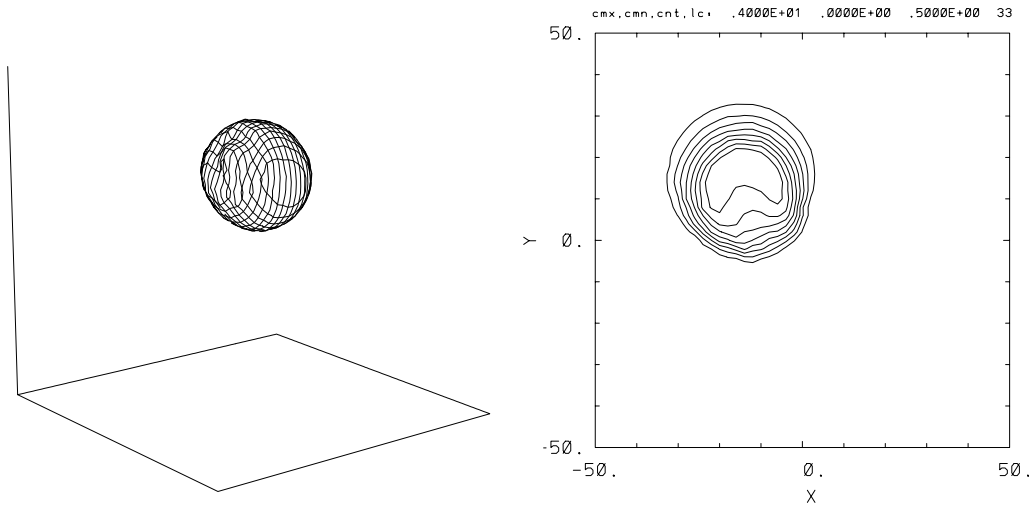

Fig. 11. As in Fig. 9 but for the leapfrog-trapezoidal FCT solution after one revolution of the sphere.

Second, it exposes alternative means of designing advection–diffusion schemes, practical for large eddy simulations of high Reynolds number flows.

Consider, in lieu of the advection Eq. (1), the elementary diffusion problem

$$\frac{\partial \Psi}{\partial t} = -\nabla \cdot (-K \nabla \Psi), \tag{27}$$

where K is a diffusion coefficient. The Fickian flux under the divergence on the r.h.s. of (27) may be formally written in the form of advective flux $-K\nabla\Psi \equiv \boldsymbol{\Omega}\Psi$, where $\boldsymbol{\Omega} = -\frac{K}{\Psi}\nabla\Psi$ (if $\Psi \neq 0$; $\boldsymbol{\Omega} = 0$ otherwise), thereby transforming the diffusion Eq. (27) into the advection equation

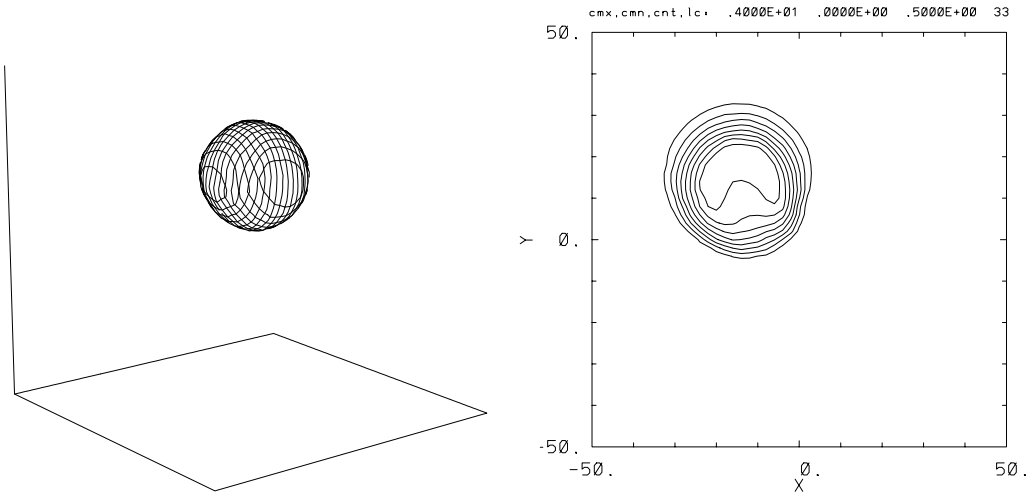

Fig. 12. As in Fig. 11 but for the MPDATA$_\infty$ FCT solution.

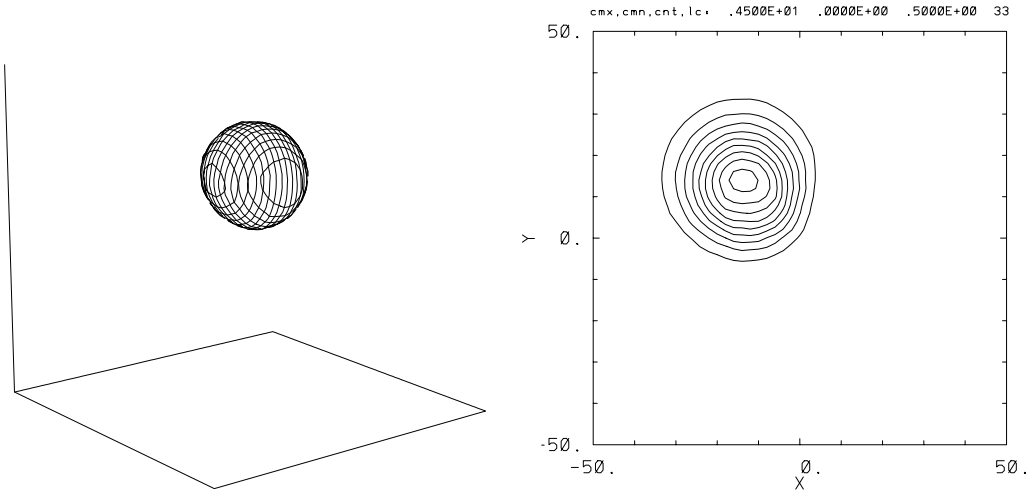

Fig. 13. As in Fig. 11 but for the basic MPDATA solution.

$$\frac{\partial \Psi}{\partial t} = -\nabla \cdot (\boldsymbol{\Omega} \Psi). \tag{28}$$

To verify the discrete advection–diffusion equivalence for 3D unstructured grids, we simulate the external scalar diffusion problem for a sphere of radius R – an archetype of water drop evaporation with known analytic solution, Section 2.2 in [21] – following the Cartesian mesh calculations of [24]. Assuming constant K, spherical symmetry, and initial/boundary conditions

$$\Psi = 0 \quad \text{for } r > R \text{ and } t = 0, \quad \Psi = \Psi_0 \quad \text{for } r \leqslant R \text{ and } t \geqslant 0, \tag{29}$$

the analytic solution takes the form

$$\Psi(r,t) = \frac{R}{r}\Psi_0\left[1 - \operatorname{erf}\left(\frac{r-R}{2\sqrt{Kt}}\right)\right]_{r\geqslant R}, \tag{30}$$

where

$$\operatorname{erf}(y) = \frac{2}{\sqrt{\pi}}\int_0^y \mathrm{e}^{-z^2}\,\mathrm{d}z. \tag{31}$$

The numerical values are selected such as to augment the 3D calculations of the preceding section. Consequently, $R = 15$, $\Psi_0 = 4$, and $K = 1$. The cuboidal domain is assumed, with the linear extend $L = 100$, whereupon $r \in [0, 50\sqrt{3}]$.

Fig. 14 shows the initial condition and the basic MPDATA solution of (28) after 3τ; where $\tau = \frac{R^2}{\pi K}$ is the characteristic time of the external diffusion process, defined as the time after which the surface-integrated flux of Ψ at $r = R$ equals twice its asymptotic value at $t \nearrow \infty$ [21]. The numerical solution uses the same unstructured grid employed in 3D advection tests in Section 5.1, and the same postprocessing procedure is used to generate the graphic display in Fig. 14. The figure conveys the sense of symmetry of the solution, in spite of the mesh irregularity, whereas the quantitative measures of the solution accuracy are listed in Table 5.

The error measures in Table 5 show that the advective solution to the diffusion equation is slightly more accurate then the solution obtained with the standard diffusion scheme. Nonetheless, this relative accuracy improvement is not the crux. Both solutions are dominated by the first order errors due to: (a) a primitive approximation of the internal boundary condition posed in (29), and (b) the generality of the effective

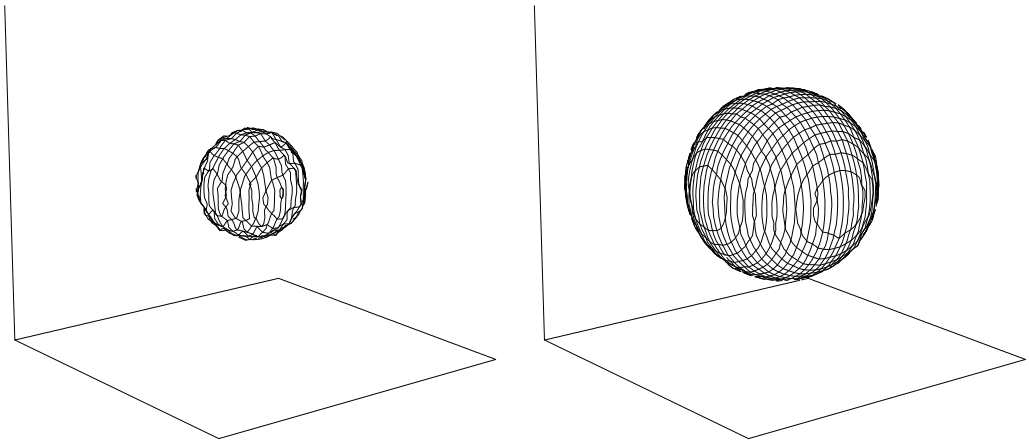

Fig. 14. External diffusion on the sphere; $\Psi = 1$ isosurfaces are shown for $t = 0$ (left plate) and after 3 characteristic times (right plate). Numerical solution in the right plate uses an MPDATA advection scheme.

Table 5
Accuracy measures of the MPDATA solution for the external diffusion problem

Scheme	L_1	L_2	L_∞
MPDATA	2.6×10^{-4}	4.8×10^{-4}	1.5
FTCS	2.8×10^{-4}	5.4×10^{-4}	1.8

The results using classical forward-in-time and centered-in-space diffusion (FTCS) scheme are included for reference. The first column identifies the scheme used and the remaining columns provide norms of the solutions' departures from the exact solution.

diffusion velocity $\boldsymbol{\Omega}$, expanding beyond the simplifying assumption of a steady flow, adopted in (1). These two aspects are important for complete flow solvers, and are addressed in the next section. The key point of the highlighted results is the validity of the discrete advection–diffusion equivalence per se. Given sign-preserving approximations, a parabolic diffusion equation can be solved to a $\delta t\mathcal{O}(\delta r)$ accuracy as a hyperbolic advection equation. This opens avenues for designing a class of new schemes. In particular, in flow problems where solution monotonicity or sign-preservation is a necessary prerequisite of accurate results, incorporating diffusion into a nonoscillatory advection scheme may improve the computational economy of the simulation, while often simplifying the programming effort. Moreover, the advective formulation allows for stable integrations of diffusion problems with $K < 0$, potentially useful for representing backscatter in large-eddy-simulations of high-Reynolds number flows [9].

5.3. Flow solvers

When applying two-time-level time-uncentered advection schemes to practical flow problems, one needs to account for time dependence of the transporting velocities as well as arbitrary forcing on the r.h.s.. In particular, in order to achieve second-order accuracy and to avoid spurious solution modes, it is important to provide at least first-order accurate estimates of the advective velocities at $n + 1/2$ [24], and an approximation of the forcing time-centered at $n + 1/2$ as well as a means of compensating the $\mathcal{O}(\delta t)$ error proportional to convective flux of the r.h.s. [28]. The latter is a consequence of ignoring the r.h.s. while deriving second-order-accurate advection schemes in the spirit of the tensor-viscosity approach [6]; i.e., via the Taylor-series expansion of lower-order algorithms, as in Section 2.

A class of methods for integrating the fluid dynamics equations with nonoscillatory two-time-level schemes – termed NFT, for the "nonoscillatory forward-in-time" – has been broadly documented in the literature. The interested reader is referred to [30] for a review. Since NFT solvers do not depend on the details of spatial discretization – but only assume a second-order-accurate nonoscillatory advection scheme as a building block – the flow chart of the solver is common to flux-form finite-difference and unstructured-mesh finite-volume discretizations. Below, we highlight[11] the NFT solver proven successful for geophysical flows [33], but adapt it to a high-Mach-number flow in a multiply connected domain, an application beyond the reach of the standard, single-block Cartesian-mesh version of MPDATA.

We consider here the conservative adiabatic Euler equations in the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0, \tag{32}$$

$$\frac{\partial \Theta}{\partial t} + \nabla \cdot \Theta \mathbf{v} = 0, \tag{33}$$

$$\frac{\partial q^I}{\partial t} + \nabla \cdot q^I \mathbf{v} = -\frac{\partial p}{\partial x^I}, \tag{34}$$

where ρ and p denote fluid density and pressure, respectively, and the momentum vector $\mathbf{q} = \rho\mathbf{v}$ with q^I denoting its components for $I = 1, 2, 3$. $\Theta := \rho\theta$ is the density weighted potential temperature, with θ denoting the potential temperature defined as

$$\theta := T(p/p_0)^{-R/c_p}, \tag{35}$$

[11] For an extended discussion directed towards aerodynamic applications, the interested reader is referred to [35].

where T is the temperature, R the gas constant, and c_p the specific heat at constant pressure. The constant p_0 in (35) is a reference (free stream) pressure. The system (32)–(34) is supplemented with the ideal-gas constitutive relation

$$p = \rho RT \equiv \mathscr{C}\Theta^{\gamma}, \tag{36}$$

where $\gamma \equiv (1 - R/c_p)^{-1}$.

The governing Euler equations can be written in a symbolic form

$$\frac{\partial \Psi}{\partial t} + \nabla \cdot \Psi \mathbf{v} = F^{\Psi}, \tag{37}$$

where Ψ denotes any of the dependent variables in (32)–(34), and F^{Ψ} is the associated right-hand side – in general, a functional of all dependent variables. Regardless of its complexity, an NFT algorithm for integrating (37) over time step δt can be compactly written as

$$\Psi_i^{n+1} = \mathscr{A}_{\mathbf{i}}(\tilde{\Psi}, \mathbf{v}^{n+1/2}) + 0.5\delta t F^{\Psi}\big|_i^{n+1}. \tag{38}$$

In (38), $\mathscr{A}$ is a shorthand for a NFT advection operator (here MPDATA), $\tilde{\Psi}$ is an auxiliary dependent variable $\widetilde{\Psi}_i = \Psi_i^n + 0.5\delta t F^{\Psi}|_i^n$, and $\mathbf{v}^{n+1/2}$ is an $\mathscr{O}(\delta t^2)$ estimate of the velocity field at $t + \delta t/2$. The overall algorithm in (38) provides fully second-order-accurate solutions Ψ_i^{n+1}, given a second-order accurate advection scheme $\mathscr{A}$ for an inhomogeneous yet time-independent flow $\mathbf{v}$, and an $\mathscr{O}(\delta t^2)$ accurate estimation of the forcing $F^{\Psi}|_i^{n+1}$. Note that this particular NFT form is reminiscent of the Strang splitting [34]; cf. [28] for the underlying theory, alternative forms, and discussion.

Given the general algorithm (38), a solution procedure for (32)–(36) readily follows. First, the advective velocity $\mathbf{v}_i^{n+1/2}$ is extrapolated linearly from the current n, and the preceding $n-1$ time levels. Second, the auxiliary variables $\widetilde{\Psi}_i$ are advected with MPDATA. Since neither (32) nor (33) are forced, this step also provides the solutions ρ^{n+1} and Θ^{n+1}. Then, the updated pressure follows from (36), and momenta are updated according to (38). Evaluating the advective velocities at $n+1$ by dividing the newly updated momenta by the updated density completes the model algorithm cycle.

The performance of the outlined NFT algorithm is illustrated for a transonic flow over an airfoil, AGARD test case 04 for NACA0012 airfoil at Mach number $M = 0.8$ with incidence angle $\alpha = 1.25°$ [1]. The triangular mesh employed, consisting of 16,101 computational points with 323 points along the airfoil, is shown in Fig. 15. The MPDATA solution has been benchmarked against solution 9 from the AGARD

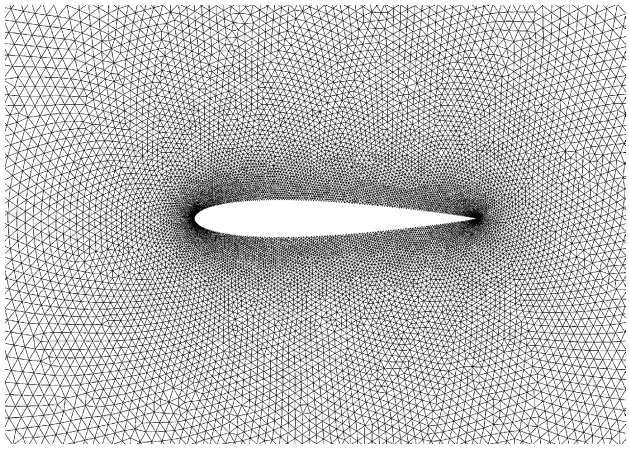

Fig. 15. Computational mesh for the NACA0012 test case.

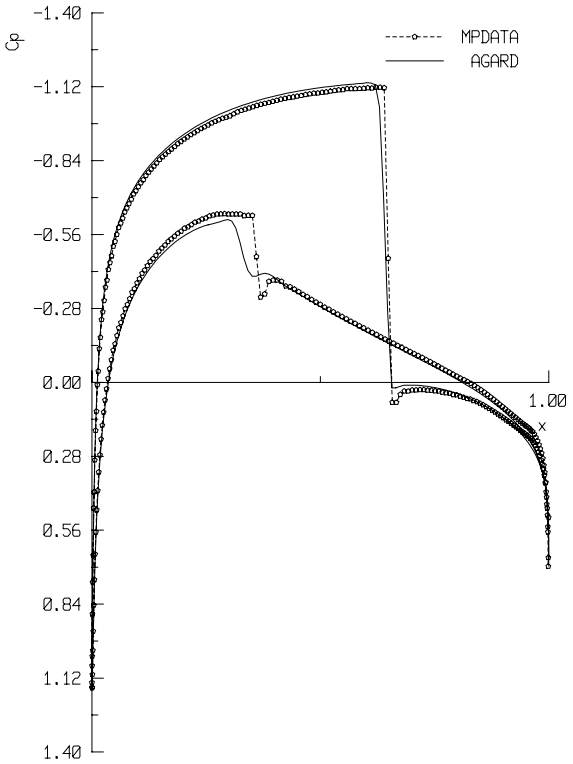

Fig. 16. Surface pressure coefficient for NACA0012 airfoil; $M = 0.8$, $\alpha = 1.25°$. A comparison of the results using MPDATA and AGARD results.

report – contributed by Schmidt and Jameson [1, pp. 6.21–6.23] – obtained by a Runge–Kutta central difference code with a blend of the second- and fourth-order artificial dissipation terms [8], while using a 320×64 structured mesh. In the far field, placed at the distance of 20 chords, boundary conditions are specified from the free-stream values or extrapolated from the Riemann invariants, depending on the character of the flow, super- or subsonic, and inflow versus outflow. At the airfoil, the free-slip condition is assumed.

Fig. 16 shows the standard display of the surface pressure coefficient C_p for the FCT-MPDATA and the AGARD reference solution 9. The MPDATA solution compares favorably with the benchmark result. In the MPDATA solution, both the upper- and lower-surface shocks are captured with one point only, whereas in the AGARD solution the two shocks are captured with 3 and 4 points, respectively. Furthermore, MPDATA gives a much sharper pressure jump on the weaker lower-surface shock (located in the region of higher wall curvature) that, in contrast, is heavily diffused in the AGARD data.

6. Concluding remarks

We developed the theory of basic MPDATA in an arbitrary finite-volume framework with a fully unstructured polyhedral hybrid mesh. We showed that our finite-volume formulation preserves the signature benefits of the standard Cartesian-mesh MPDATA scheme – i.e., sign-preservation, a full multidimensionality, and the overall accuracy exceeding the accuracy of the centered in time and space approximation. Furthermore, we developed extensions important for applications, such as a fully monotone option, the

"infinite gage" MPDATA (a linearized variant proven successful in Cartesian-mesh simulations of turbulent flows), and a diffusion scheme. We generalized the approach to a fully compressible flow solver, thereby documenting that the new framework is well suited for adapting the accomplishments of Cartesian mesh MPDATA in the unstructured mesh environment. The example of flow past a 2D airfoil in Section 5.2 is perhaps the first documented application of an MPDATA-based flow solver in multiply connected domains. The ease of modeling multiply connected domains using unstructured meshes promises new avenues for MPDATA in the area of atmospheric/oceanic flows, for which the scheme was developed originally, and for which MPDATA is known the best.

The presented framework allows extending the applicability of MPDATA to engineering problems. The result for the transonic flow around NACA0012 airfoil shows the promise of the method as it is substantially less diffusive than the Runge–Kutta Jameson schemes routinely used in the aeronautical design. Moreover, unlike many numerical schemes, MPDATA appears not to enlarge numerical diffusion on unstructured meshes, evincing solutions close to analogous solutions on Cartesian meshes. The problem of numerical overdiffusivity is known to be particularly severe for distorted meshes. This highlights the potential importance of the 'mesh-skewness-term' – a new aspect relevant, in general, to both structured- and unstructured-mesh formulations – revealed in the derivation in Section 2, and consequently the importance of a careful choice of discretization. We intend to study this further.

Over the last two decades, finite-difference MPDATA has been frequently compared with other transport schemes, primarily in the context of passive scalar advection. The assessments of MPDATA's relative strengths and weaknesses reported in the literature depend very much on the schemes included in comparisons, choice of test problems, MPDATA's options, and details of implementation. The most common complaints are that the basic MPDATA is too diffusive, while enhanced MPDATAs are too expensive. The most often acknowledged virtues are MPDATA's multidimensionality, robustness, and its underlying conceptual simplicity. These advantages carry over to the finite-volume unstructured-mesh formulation, whereas the relative efficiency of advection becomes less important with increasing complexity of the fluid models. In particular, the accuracy and efficiency measures of various oscillatory and nonoscillatory schemes listed in Table 4 in Section 5.1 stimulate a query "which scheme is optimal". In our experience, there is no straightforward answer. Consider, for instance, that for implicit flow solvers the efficacy of a fluid model depends critically on the efficiency of the associated elliptic solver. The latter, in turn, depends on the spectral composition of the r.h.s. depending, in turn, on the spectral properties of the advective transport [29]. The finite-difference experience with MPDATA indicates that various options have merit for an accurate simulation of complex high-Reynolds-number flows, cf. [31].

Acknowledgments

Personal review by Robert Sharman is gratefully acknowledged. Comments from Anna Crowley and Len Margolin on an earlier version of this paper helped to improve the presentation. This work was supported in part by the US Department of Energy "Climate Change Prediction Program" (CCPP) research initiative.

Appendix A. Stability considerations

A.1. Positivity of finite-volume upwinding

In order to show the sufficiency of (5) for preserving the nonnegative character of the fields transported with the upwind scheme (2), (3), we write the latter in the closed form

$$\Psi_i^{n+1} = \Psi_i^n - \sum_{j=1}^{l(i)} \left\{ [U_j]^+ \Psi_i^n + [U_j]^- \Psi_j^n \right\}, \tag{A.1}$$

where U_j is a contribution to the local Courant number defined in (5). Rearranging (A.1) as

$$\Psi_i^{n+1} = \left(1 - \sum_{j=1}^{l(i)} [U_j]^+ \right) \Psi_i^n - \sum_{j=1}^{l(i)} [U_j]^- \Psi_j^n \tag{A.2}$$

implies

$$\Psi_i^{n+1} \geqslant \left(1 - \sum_{j=1}^{l(i)} [U_j]^+ \right) \Psi_i^n, \tag{A.3}$$

given $\Psi_j^n \geqslant 0\ \forall_j$. Consequently, $1 - \sum_{j=1}^{l(i)} [U_j]^+ \geqslant 0$ suffices for $\Psi_j^{n+1} \geqslant 0\ \forall_j$. □

For nonpositive fields $\Psi_j^n \leqslant 0 \forall_j$, this also holds – since multiplying (A.1) by (-1) recovers the result above for $\widetilde{\Psi}_j^{n+1} \geqslant 0\ \forall_j$, where $\widetilde{\Psi} \equiv -\Psi$. Extending the proof for fields of variable sign involves: (i) decomposing Ψ into the nonnegative and nonpositive parts $[\Psi]^+$ and $[\Psi]^-$, as in (4); and (ii) noting that the Lagrangian counterpart of the governing advection problem in (1), $\mathrm{D}\Psi/\mathrm{D}t = 0$, implies $\mathrm{D}[\Psi]^+/\mathrm{D}t = 0$ and $\mathrm{D}[\Psi]^-/\mathrm{D}t = 0$ (due to the disjoined supports of $[\Psi]^+$ and $[\Psi]^-$ along flow trajectories). In consequence, the Eulerian form (1) must be satisfied for both parts separately, cf. [32] for a discussion.

For finite volume schemes, the sign preservation ensures nonlinear stability, because the finite-volume (2) enforces conservation by design; whereas the sign-preservation enforces conservation of the $|\Psi|$, viz. the solution boundedness in the $\sum|\,|$ norm. From the norm equivalence in $\mathbf{R}^n$ (cf. Corollary VII.1.2 in [15]) the latter ensures the solution boundedness in the L_2 norm; cf. [32] for a discussion.

A.2. Convexity of finite-volume upwinding

In order to show the sufficiency of (5), (6) for the solution convexity in (7), we note first that (6) implies

$$\sum_{j=1}^{l(i)} U_j = 0 \Rightarrow \sum_{j=1}^{l(i)} [U_j]^+ = -\sum_{j=1}^{l(i)} [U_j]^- = \sum_{j=1}^{l(i)} \left| [U_j]^- \right|. \tag{A.4}$$

Since

$$\sum_{j=1}^{l(i)} [U_j]^+ \equiv \sum_{j=1}^{l(i)} \left| [U_j]^+ \right|, \tag{A.5}$$

the positivity/stability condition (5) together with (A.4) ensure

$$\sum_{j=1}^{l(i)} \left| [U_j]^+ \right| = \sum_{j=1}^{l(i)} \left| [U_j]^- \right| \leqslant 1. \tag{A.6}$$

Employing in (A.2) the identity (A.5), equalities (A.4), and the condition (A.6) results in

$$\begin{aligned} \Psi_i^{n+1} &= \Psi_i^n \left(1 - \sum_{j=1}^{l(i)} \left| [U_j]^+ \right| \right) + \sum_{j=1}^{l(i)} \left| [U_j]^+ \right| \Psi_j^n \leqslant \Psi_i^n \left(1 - \sum_{j=1}^{l(i)} \left| [U_j]^+ \right| \right) + \max_{j=1,l(i)} \Psi_j^n \sum_{j=1}^{l(i)} \left| [U_j]^+ \right| \\ &\leqslant \max_{j=1,l(i)} (\Psi_i^n, \Psi_j^n). \end{aligned} \tag{A.7}$$

On the other hand,

$$\Psi_i^{n+1} = \Psi_i^n\left(1-\sum_{j=1}^{l(i)}\left|[U_j]^+\right|\right)+\sum_{j=1}^{l(i)}\left|[U_j]^+\right|\Psi_j^n \geqslant \Psi_i^n\left(1-\sum_{j=1}^{l(i)}\left|[U_j]^+\right|\right)+\min_{j=1,l(i)}\Psi_j^n\sum_{j=1}^{l(i)}\left|[U_j]^+\right|$$
$$\geqslant \min_{j=1,l(i)}(\Psi_i^n,\Psi_j^n)\left(1-\sum_{j=1}^{l(i)}\left|[U_j]^+\right|+\sum_{j=1}^{l(i)}\left|[U_j]^+\right|\right)=\min_{j=1,l(i)}(\Psi_i^n,\Psi_j^n), \tag{A.8}$$

and the union of (A.7) and (A.8) completes the proof.

Note that, (A.6) together with identities in (A.4) and (A.5) gives

$$\sum_{j=1}^{l(i)}[U_j]^+ \leqslant 1, \quad \sum_{j=1}^{l(i)}-[U_j]^- \leqslant 1. \tag{A.9}$$

Since $|U| = [U]^+ - [U]^-$ from the definitions in (4), adding both inequalities in (A.9) results in

$$\sum_{j=1}^{l(i)}|U_j| \leqslant 2, \tag{A.10}$$

a stability condition equivalent to the one familiar from finite-difference literature; e.g. (4) in [23].

References

[1] Test cases for inviscid flow field methods, AGARD Advisory Report 1985, No 211 AGARD-AR-211.
[2] D.P. Bacon, et al., A dynamically adapting weather and dispersion model: the operational environment model with grid adaptivity (OMEGA), Mon. Weather Rev. 128 (2000) 2044–2076.
[3] T.J. Barth. Aspects of unstructured grids and finite volume solvers for the Euler and Navier–Stokes equations, in: Special Course on Unstructured Grid Methods for Advection Dominated Flows, AGARD Report 787, 1992, pp. 6.1–6.61.
[4] 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.
[5] J.A. Domaradzki, N.A. Adams, Direct modelling of subgrid scales of turbulence in large eddy simulation, J. Turbul. 3 (2002) 1–19.
[6] J.K. Dukowicz, J.D. Ramshaw, Tensor viscosity method for convection in numerical fluid dynamics, J. Comput. Phys. 32 (1979) 3890–3893.
[7] W.W. Grabowski, P.K. Smolarkiewicz, Monotone finite-difference approximations to the advection-condensation problem, Mon. Weather Rev. 118 (1990) 2082–2097.
[8] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes, AIAA June 1981, Paper 81-1259.
[9] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, Phys. Fluids A 4 (1992) 633–635.
[10] R. Lohner, K. Morgan, O.C. Zienkiewicz, An adaptive finite element procedure for high speed flows, Comput. Meth. Appl. Mech. Eng. 51 (1985) 441–465.
[11] L.G. Margolin, P.K. Smolarkiewicz, Antidiffusive velocities for multipass donor cell advection, SIAM J. Sci. Comput. 20 (3) (1998) 907–929.
[12] L.G. Margolin, W.J. Rider, A rationale for implicit turbulence modeling, Int. J. Numer. Meth. Fluids 39 (2002) 799–819.
[13] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (2003) 266–298.
[14] L.G. Margolin, P.K. Smolarkiewicz, A.A. Wyszogrodzki, Implicit turbulence modeling for high Reynolds number flows, J. Fluids Eng. 124 (2002) 862–867.
[15] K. Maurin, Analysis Part I: Elements, Reidel, Boston, 1976, 430pp.
[16] D.J. Mavriplis, Three dimensional unstructured multigrid for the Euler equations, AIAA J. 30 (7) (1992) 1753–1761.
[17] J.M. Prusa, P.K. Smolarkiewicz, An all-scale anelastic model for geophysical flows: dynamic grid deformation, J. Comput. Phys. 190 (2003) 601–622.

[18] P.L. Roe, Error estimates for cell-vertex solutions of the compressible Euler equations, ICASE Report No. 87-6, 1987, 40pp.
[19] P.J. Roache, Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1972, 446 pp.
[20] C. Schär, P.K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations, J. Comput. Phys. 128 (1996) 101–120.
[21] Y.S. Sedunov, (P. Greenberg (Ed.)), Physics of Drop Formation in the Atmosphere, 1974, 234 pp. (Israel Program for Sci. Transl.).
[22] P.K. Smolarkiewicz, A simple positive definite advection scheme with small implicit diffusion, Mon. Weather Rev. 111 (1983) 479–486.
[23] P.K. Smolarkiewicz, A fully multidimensional positive definite advection transport algorithm with small implicit diffusion, J. Comput. Phys. 54 (1984) 325–362.
[24] P.K. Smolarkiewicz, T.L. Clark, The multidimensional positive definite advection transport algorithm: further development and applications, J. Comput. Phys. 67 (1986) 396–438.
[25] P.K. Smolarkiewicz, W.W. Grabowski, The multidimensional positive definite advection transport algorithm: nonoscillatory option, J. Comput. Phys. 86 (1990) 355–375.
[26] P.K. Smolarkiewicz, On forward-in-time differencing for fluids, Mon. Weather. Rev. 119 (1991) 2505–2510.
[27] P.K. Smolarkiewicz, Nonoscillatory advection schemes, in: Numerical Methods in Atmospheric Models, Proceedings of Seminar on Numerical Methods in Atmospheric Models, ECMWF, Reading, UK, 9–13 September, 1991, p. 235.
[28] P.K. Smolarkiewicz, L.G. Margolin, On forward-in-time differencing for fluids: extension to a curvilinear framework, Mon. Weather Rev. 121 (1993) 1847–1859.
[29] P.K. Smolarkiewicz, V. Grubišić, L.G. Margolin, On forward-in-time differencing for fluids: stopping criteria for iterative solutions of anelastic pressure equations, Mon. Weather Rev. 125 (1997) 647–654.
[30] P.K. Smolarkiewicz, L.G. Margolin, MPDATA: a finite difference solver for geophysical flows, J. Comput. Phys. 140 (1998) 459–480.
[31] P.K. Smolarkiewicz, L.G. Margolin, A.A. Wyszogrodzki, A class of nonhydrostatic global models, J. Atmos. Sci. 58 (2001) 349–364.
[32] P.K. Smolarkiewicz, J.M. Prusa, VLES modeling of geophysical fluids with nonoscillatory forward-in-time schemes, Int. J. Numer. Meth. Fluids 39 (2002) 799–819.
[33] P.K. Smolarkiewicz, J.M. Prusa, Forward-in-time differencing for fluids: simulation of geophysical turbulence, in: D. Drikakis, B.J. Guertz (Eds.), Turbulent Flow Computation, Kluwer Academic Publishers, Dordrecht, 2002, pp. 207–240.
[34] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[35] J. Szmelter, P.K. Smolarkiewicz, A low-implicit-diffusion flow solver for unstructured meshes, AIAA-2005-0321, January, 2005.
[36] 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.
[37] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.