ELSEVIER

A discrete operator calculus for finite difference approximations

Len G. Margolin [a,*], Mikhail Shashkov [a,1], Piotr K. Smolarkiewicz [b,2]

[a] *Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[b] *National Center for Atmospheric Research, Boulder, CO 80307, USA*

Abstract

In this article we describe two areas of recent progress in the construction of accurate and robust finite difference algorithms for continuum dynamics. The support operators method (SOM) provides a conceptual framework for deriving a discrete operator calculus, based on mimicking selected properties of the differential operators. In this paper, we choose to preserve the fundamental conservation laws of a continuum in the discretization. A strength of SOM is its applicability to irregular unstructured meshes. We describe the construction of an operator calculus suitable for gas dynamics and for solid dynamics, derive general formulae for the operators, and exhibit their realization in 2D cylindrical coordinates. The multidimensional positive definite advection transport algorithm (MPDATA) provides a framework for constructing accurate nonoscillatory advection schemes. In particular, the nonoscillatory property is important in the remapping stage of arbitrary-Lagrangian–Eulerian (ALE) programs. MPDATA is based on the sign-preserving property of upstream differencing, and is fully multidimensional. We describe the basic second-order-accurate method, and review its generalizations. We show examples of the application of MPDATA to an advection problem, and also to a complex fluid flow. We also provide an example to demonstrate the blending of the SOM and MPDATA approaches. © 2000 Elsevier Science S.A. All rights reserved.

1. Introduction

In this article, we review two areas of progress in the construction of accurate and robust finite difference schemes. The first area is the development of consistent spatial difference operators. The second area is the development of multidimensional nonoscillatory advection schemes, which can be used in Eulerian and continuous rezone (ALE) programs. The techniques we will describe were first developed outside the field of numerical solid dynamics, but should have useful applications there as well.

The underlying idea of the support operators method (SOM) is to develop a discrete operator calculus – i.e., finite difference approximations to first-order spatial difference operators like divergence, gradient, and curl – that faithfully reproduces selected properties of the analytic calculus. We will show how to construct such a discrete calculus that is suitable for continuum dynamics. One important advantage of the support operator theory is that it may be applied with equal facility to regular, irregular, and unstructured meshes encountered in both Eulerian and Lagrangian simulations.

The multidimensional positive definite advection transport algorithm (MPDATA) provides a conceptual framework for nonoscillatory advection. MPDATA was first developed for geophysical applications where

---

the preservation of the sign of fields, which on physical grounds must remain nonnegative, is of paramount importance. The application of MPDATA to the advection equation with variable coefficients (in particular, to the momentum equation) guarantees the nonlinear stability when the timestep is suitably restricted. MPDATA is not based on any flux-limiting procedure, and is fully multidimensional. The absence of spatial (directional) splitting errors is especially important on irregular grids. Recent enhancements allow us to use MPDATA as a full fluid solver, meaning that we treat the forcing terms (such as the pressure gradient) consistently to the second-order in time.

The structure of this paper is as follows. In Section 2 we give a brief review of the method of support operators, as applied to the equations of gas dynamics, and then describe the extensions to construct the operators required by the equations of solid dynamics. A formal exposition of SOM, with applications to elliptic and parabolic PDEs, as well as to electromagnetic theory, can be found in [15,20–23,29,31–35].

In Section 3, we outline the concept of basic MPDATA as applied to a model equation for continuum dynamics, and then review its generalizations. A full review (and technical details) of MPDATA in the context of geophysical applications can be found in [44]. In Section 4, we explore the possibility of combining the two approaches, and demonstrate potential advantages in the example of shallow fluid flow on a rotating sphere.

2. Method of support operators

2.1. Basic idea

Most PDEs of mathematical physics can be formulated in terms of only a few fundamental differential operators, such as divergence, gradient, and curl. SOM provides a systematic approach to spatial differencing by constructing discrete analogs of these operators. The differential operators satisfy relationships that closely relate to conservation laws and other physically important principles. Experience has shown that the best results usually are obtained when the numerical approximations of the fundamental operators reproduce those properties of the analytic operators. In this sense, SOM is principally concerned with developing a discrete analog of the differential operator calculus.

As a practical matter, the development of this calculus proceeds in two steps. First one chooses a discrete form for one of the fundamental operators, which we term the *prime* operator. Then, based on the subset of analytic properties that one chooses to maintain, one constructs the other fundamental operators, which we term the *derived* operators. The choice of the prime operator depends on the application and details of the discretization.

In this section, we illustrate the procedure with three examples of relevance to solid dynamics. In Section 2.2 we show how to derive a discrete divergence operator (prime operator) by requiring consistency of the divergence of the velocity field with the time rate of change of volume of a finite difference cell. In Section 2.3 we show how to construct the discrete gradient operator, based on requiring the exact conservation of total energy, which implies the adjointness of the discrete gradient and divergence. In Section 2.4 we consider the approximation of the divergence of a tensor, and the gradient of a vector, both required by the equations of solid dynamics. Finally in Section 2.5, we present an example that illustrates the advantages of the consistent approximation of the divergence operator.

Throughout Section 2 we will focus on the usual staggered mesh, where thermodynamic quantities are stored at cell centers and velocities are stored at cell vertices. The staggered mesh has proven very successful for high-speed flow simulations and is well suited for SOM. However, SOM can be applied to any data structure. A typical staggered mesh grid cell, including the data structure, is shown in Fig. 1.

In this paper, we distinguish between the differential operators grad and div, and the discrete operators GRAD and DIV. In addition, in Section 2.4, we will have need for the discrete approximations for the divergence of a tensor and the gradient of a vector, which we will denote as $\mathscr{D}\mathscr{I}\mathscr{V}$ and $\mathscr{G}\mathscr{R}\mathscr{A}\mathscr{D}$, respectively.

2.2. The divergence operator

Our first task is to define a prime operator. For the Lagrangian staggered grid shown in Fig. 1, the natural choice for a prime operator is DIV, for reasons that will become clear shortly. In a continuum, the
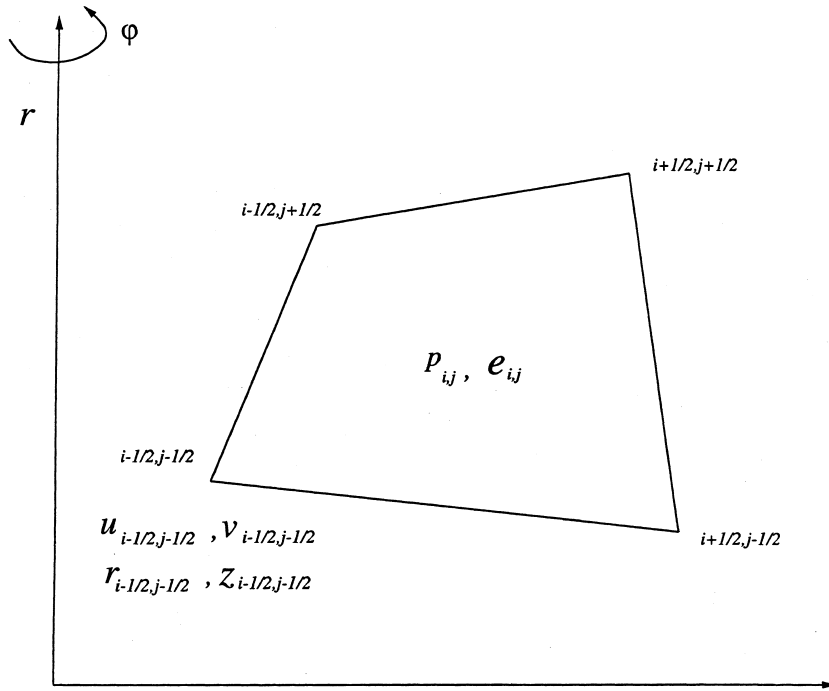
Fig. 1. A typical staggered mesh grid cell, coordinates and components of the velocity vector are located in the nodes, and pressure and internal energy are located in the cell.

time rate of change of a Lagrangian volume element, $V(t)$, is related to the divergence of velocity, $\vec{u}$, as follows:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \int_{V(t)} \operatorname{div} \vec{u} \, \mathrm{d}V', \tag{2.1}$$

(cf. [26,34]). In the discrete case, a natural choice of $V(t)$ is the computational cell itself. For a computational cell (i, j, k), the volume is a function of the coordinates of cell vertices and does not depend explicitly on time

$$V_{i,j,k}(t) = V(x_a^\alpha(t)), \quad \alpha = 1, 2, 3 \text{ and } a \in A, \tag{2.2}$$

where A is the set of all the vertices of the cell (i, j, k). Therefore, we can write

$$\frac{\mathrm{d}V_{i,j,k}}{\mathrm{d}t} = \sum_{\alpha, a \in A} \frac{\partial V_{i,j,k}}{\partial x_a^\alpha} \frac{\mathrm{d}x_a^\alpha}{\mathrm{d}t} = \sum_{\alpha, a \in A} \frac{\partial V_{i,j,k}}{\partial x_a^\alpha} u_a^\alpha, \tag{2.3}$$

where u_a^α denotes the α-component of $\vec{u}$ at vertex a. The integral on the r.h.s. of (2.1) can be expressed (from the mean value theorem) as

$$\int_{V(t)} \operatorname{div} \vec{u} \, \mathrm{d}V' = V \operatorname{div} \vec{u}|_{\vec{x}^*}, \tag{2.4}$$

where $\vec{x}^*$ is some point in the cell. Using (2.3) and (2.4) in (2.1), we define DIV via

$$\left(\operatorname{DIV} \vec{u} \right)_{i,j,k} := \frac{1}{V_{i,j,k}} \sum_{\alpha, a \in A} \frac{\partial V_{i,j,k}}{\partial x_a^\alpha} u_a^\alpha, \tag{2.5}$$

Eq. (2.5) is a very convenient way to write a discrete divergence, because it is coordinate invariant, and can be used for Cartesian, cylindrical, spherical, etc., coordinate systems. It can also be used when the edges of a

grid cell are not segments of straight lines, but are some more general curves. All that is needed to implement (2.3) is an expression for the volume of a cell as a function of the coordinates of its vertices, and perhaps those of neighboring vertices. Note that the definition (2.5) proceeds from the particular physical principle (2.1); however the coefficients of velocity in definition (2.5) depend only on geometry of the grid, and so (2.5) is generally applicable to any vector field.

In order to illustrate the application of (2.5), we consider the case of 2D cylindrical coordinates (r, z), which has great practical importance. The volume of a cell – an example of (2.2) – is

$$
\begin{aligned}
V_{i,j} = \frac{1}{3}\left(r_{i-1/2,j-1/2} + r_{i+1/2,j-1/2} + r_{i-1/2,j+1/2}\right)\frac{1}{2}\left(r_{i-1/2,j-1/2}(z_{i+1/2,j-1/2} - z_{i-i/2,j+1/2})\right.\\
+ r_{i+1/2,j-1/2}(z_{i-i/2,j+1/2} - z_{i-i/2,j-1/2}) + r_{i-i/2,j+1/2}(z_{i-i/2,j-1/2} - z_{i+1/2,j-1/2}))\\
+ \frac{1}{3}\left(r_{i+1/2,j-1/2} + r_{i+1/2,j+1/2} + r_{i-i/2,j+1/2}\right)\frac{1}{2}\left(r_{i+1/2,j-1/2}(z_{i+1/2,j+1/2} - z_{i-i/2,j+1/2})\right.\\
\left. + r_{i+1/2,j+1/2}(z_{i-i/2,j+1/2} - z_{i+1/2,j-1/2}) + r_{i-i/2,j+1/2}(z_{i+1/2,j-1/2} - z_{i+1/2,j+1/2})\right).
\end{aligned}
\tag{2.6}
$$

Corresponding examples of the r and z derivatives of this volume are:

$$
\begin{aligned}
\frac{\partial V_{i,j}}{\partial r_{i+1/2,j+1/2}} = \frac{1}{6}\left(r_{i+1/2,j-1/2}(z_{i+1/2,j+1/2} - z_{i+1/2,j-1/2}) + 2r_{i+1/2,j+1/2}(z_{i-i/2,j+1/2} - z_{i+1/2,j-1/2})\right.\\
\left. + r_{i-i/2,j+1/2}(z_{i-i/2,j+1/2} - z_{i+1/2,j+1/2})\right).
\end{aligned}
\tag{2.7}
$$

and

$$
\frac{\partial V_{i,j}}{\partial z_{i+1/2,j+1/2}} = \frac{1}{6}\left((r_{i+1/2,j-1/2} + r_{i+1/2,j+1/2} + r_{i-i/2,j+1/2})(r_{i+1/2,j-1/2} - r_{i-i/2,j+1/2})\right).
\tag{2.8}
$$

The other derivatives are easily found by cyclic permutation. These expressions are written more compactly in a local notation in [26], where the consistency of these expressions with their Cartesian counterparts for cells far from the axis of symmetry is also demonstrated.

We offer the following interpretation of the discrete divergence in cylindrical coordinates using the formulae (2.7), (2.8), and their permutations. Analytically, this divergence is usually written as

$$
\operatorname{div}\vec{u} = \frac{1}{r}\frac{\partial(ru)}{\partial r} + \frac{\partial v}{\partial z},
$$

where $\vec{u} = (u, v)$. A direct discretization based on this form does not usually preserve the connection between volume and velocity (2.1). Our approach corresponds to discretizing the expression for divergence in *conservative* form

$$
\operatorname{div}\vec{u} = \frac{1}{r}\left(\frac{\partial(ru)}{\partial r} + \frac{\partial(rv)}{\partial z}\right).
$$

The consistency between the change of volume and definitions of operators on the discrete level is particularly important in solid dynamics, where the velocities are used to estimate strain rate tensor, see Section 2.4.

2.3. The gradient operator

Our next task is to derive the discrete gradient GRAD. To proceed we will continue in the framework of continuum mechanics. In particular we will consider the equations of gas dynamics and employ the conservation of total energy.

The Lagrangian form of the equations of gas dynamics is:

$$
\frac{d\rho}{dt} = -\rho\operatorname{div}\vec{u},
\tag{2.9}
$$

$$\rho \frac{\mathrm{d}\vec{u}}{\mathrm{d}t} = -\mathrm{grad}\,p, \tag{2.10}$$

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = -p\,\mathrm{div}\,\vec{u}, \tag{2.11}$$

where ρ is the density, p the pressure, and e is the internal energy.

The conservation law for total energy is derived from the momentum equations and the internal energy equation, by taking the scalar product of (2.10) with the velocity vector, adding the result to the equation for internal energy and integrating over a volume $V(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\int_{V(t)}\left(\frac{|\vec{u}|^2}{2}+e\right)\rho\,\mathrm{d}V'\right] = -\left[\int_{V(t)}(\mathrm{grad}\,p,\vec{u})\,\mathrm{d}V' + \int_{V(t)} p\,\mathrm{div}\,\vec{u}\,\mathrm{d}V'\right] = \oint_{\partial V} p(\vec{u},\vec{n})\,\mathrm{d}S, \tag{2.12}$$

where $\vec{n}$ is the unit outward normal vector to ∂V. This equation states that in the absence of external forces, the change in total energy within the volume $V(t)$ is equal to the work done by surface forces. The last equality in (2.12) is an integral identity that holds for any scalar function p and vector field $\vec{u}$, and expresses the adjointness of grad and div operators. If DIV and GRAD are constructed independently, they may not satisfy a discrete analog of (2.12), whereupon the energy will be only conserved to the order of the truncation error. However, if we define GRAD from a discrete analog to (2.12), then the resulting numerical scheme will conserve energy to roundoff error.

To actually construct the discrete gradient operator in cylindrical coordinates, here we employ the simplest discrete analog of the integral identity relating divergence and gradient and assuming a zero boundary integral

$$\sum_{\text{cells}} p_{i,j}(\mathrm{DIV}\,\vec{u})_{i,j}V_{i,j} + \sum_{\text{nodes}}(G^r_{i+1/2,j+1/2}u_{i+1/2,j+1/2} + G^z_{i+1/2,j+1/2}v_{i+1/2,j+1/2})V_{i+1/2,j+1/2} = 0, \tag{2.13}$$

where $G^r_{i+1/2,j+1/2}$, and $G^z_{i+1/2,j+1/2}$ are, respectively, r and z components of GRADp, at a node $(i+1/2, j+1/2)$. The $V_{i+1/2,j+1/2}$ is a volume associated with a node such that

$$\sum_{\text{nodes}} V_{i+1/2,j+1/2} = \sum_{\text{cells}} V_{i,j} = \text{total volume}.$$

By comparing the coefficients of $u_{i+1/2,j+1/2}$ in the first and second sums, we obtain an explicit expression for $G^r_{i+1/2,j+1/2}$, and similarly for $G^z_{i+1/2,j+1/2}$:

$$G^r_{i+1/2,j+1/2} = -\frac{1}{V_{i+1/2,j+1/2}}\sum_{k,l=0,1}\frac{\partial V_{i+k,j+l}}{\partial r_{i+1/2,j+1/2}}p_{i+k,j+l}, \tag{2.14a}$$

$$G^z_{i+1/2,j+1/2} = -\frac{1}{V_{i+1/2,j+1/2}}\sum_{k,l=0,1}\frac{\partial V_{i+k,j+l}}{\partial z_{i+1/2,j+1/2}}p_{i+k,j+l}. \tag{2.14b}$$

The coefficients in these expressions are the same derivatives of the cell volumes with respect to nodal coordinates that appear in DIV in (2.5). As in the case of the divergence, the coefficients of pressure in (2.14a) and (2.14b) depend only on geometry of the grid, and so (2.14a) and (2.14b) are generally applicable to any scalar field.

When external pressure is applied to the boundary, the discrete analog of the surface integral term should be included in (2.13) (cf. Section 3.3.2 in [34]). This leads to a consistent definition of GRAD at the boundary. [3]

---

[3] Note the advantage for preserving the symmetry of elliptic pressure operators in implicit formulations, important for the convergence of conjugate-gradient methods (cf. [5]).

2.4. Extensions to solid dynamics

The equations of mechanics of continuum media in Lagrangian form can be written as follows:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho \operatorname{div}\vec{u} = 0, \tag{2.15a}$$

$$\rho\frac{\mathrm{d}\vec{u}}{\mathrm{d}t} = \operatorname{div}\hat{\sigma}, \tag{2.15b}$$

$$\rho\frac{\mathrm{d}e}{\mathrm{d}t} = \hat{\sigma}:\hat{\dot{\epsilon}}, \tag{2.15c}$$

where $\hat{\sigma}$ is the stress tensor and $\hat{\dot{\epsilon}} = 0.5[\nabla\vec{u} + (\nabla\vec{u})^*]$ is the strain rate tensor. Note that these equations contain two new differential operators related to the tensors – the divergence of a tensor, $\operatorname{div}\hat{\sigma}$, and the gradient of a vector, $\nabla\vec{u} = \operatorname{grad}\vec{u}$. In discretizing these equations on the staggered grid, the discrete divergence, $\mathcal{DIV}$, uses cell-centered quantities to produce a vertex quantity. Likewise the discrete gradient, $\mathcal{GRAD}$, uses vertex quantities to produce a cell-centered quantity.

Before introducing the discrete operators, it is necessary to identify which properties of the differential operators imply the conservation laws associated with (2.15a)–(2.15c). For the equations of solid dynamics, the conservation laws for mass and volume have the same form as for gas dynamics, relying on the adjointness of the divergence of a vector and the gradient of a scalar. The conservation law for momentum follows from 2.15a)–(2.15c) and the divergence property of $\operatorname{div}\hat{\sigma}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_V \rho\vec{u}\,\mathrm{d}V\right) = \int_V \operatorname{div}\hat{\sigma}\,\mathrm{d}V = \oint_{\partial V} \hat{\sigma}\cdot\vec{n}\,\mathrm{d}S. \tag{2.16}$$

The conservation of total energy follows from (2.15a)–(2.15c) and the integral identity

$$\int_V \nabla\vec{A}:\hat{\sigma}^*\,\mathrm{d}V + \int_V (\vec{A},\operatorname{div}\hat{\sigma})\mathrm{d}V = \oint_S \left(\vec{n},(\hat{\sigma}\cdot\vec{A})\right)\mathrm{d}S, \tag{2.17}$$

as follows

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left\{\int_V\left[\rho\left(e + \frac{\vec{u}^2}{2}\right)\right]\mathrm{d}V\right\} &= \int_V (\operatorname{div}\hat{\sigma},\vec{u})\,\mathrm{d}V + \int_V \hat{\sigma}:\hat{\dot{\epsilon}}\,\mathrm{d}V \\
&= \int_V (\operatorname{div}\hat{\sigma},\vec{u})\mathrm{d}V + \int_V \nabla\vec{u}:\hat{\sigma}\,\mathrm{d}V \\
&= \oint_{\partial V}\left(\vec{n},(\hat{\sigma}\cdot\vec{u})\right)\mathrm{d}S.
\end{aligned} \tag{2.18}$$

The identity (2.17) expresses the adjointness of the divergence of a tensor and the gradient of a vector. Let us note that if $\hat{\sigma} = -p\hat{I}$ then $\operatorname{div}\hat{\sigma} = -\operatorname{grad}p$.

To design a finite difference approximation for (2.15a)–(2.15c), we can use (2.5) for $\mathrm{DIV}\vec{u}$, but need to construct $\mathcal{DIV}\,\hat{\sigma}$ and $\mathcal{GRAD}\,\vec{u}$. The analysis above implies that $\mathcal{DIV}$ must satisfy the discrete analog of the Stokes' theorem (2.16), and that $\mathcal{DIV}$ and $\mathcal{GRAD}$ have to be adjoint to each other (2.17). Finally, it is natural to require that in the case where $\hat{\sigma} = -p\hat{I}$ the finite difference scheme has to coincide with the finite difference scheme for gas dynamics, that is we require that $\mathcal{DIV}(p\hat{I}) = \mathrm{GRAD}p$.

To proceed we need two additional integral identities

$$\int_V \left(\operatorname{grad}(\varphi\psi),\vec{A}\right)\mathrm{d}V = \int_V \varphi(\vec{A},\operatorname{grad}\psi)\,\mathrm{d}V + \int_V \psi(\vec{A},\operatorname{grad}\varphi)\mathrm{d}V, \tag{2.19}$$

$$\int_V \nabla\vec{A}:\psi\hat{\sigma}^*\,\mathrm{d}V + \int_V (\vec{A},\psi\operatorname{div}\hat{\sigma})\mathrm{d}V + \int_V (\vec{A},(\operatorname{grad}\psi\cdot\hat{\sigma}))\mathrm{d}V = \oint_S (\psi\vec{n},(\hat{\sigma}\cdot\vec{A}))\mathrm{d}S. \tag{2.20}$$

We summarize the procedure of discretization in the following sequence of steps:

(1) We choose DIV as the prime operator and construct it exactly as in the case of gas dynamics. We also construct the operator GRAD as we did in the case of gas dynamics. Recall that GRAD uses cell-centered quantities to produce a vertex quantity.

(2) Next we use the discrete analog of the identity (2.19) to construct another discrete analog of the gradient that uses vertex quantities to produce a cell-centered quantity. We denote this operator as $\overline{\text{GRAD}}$. The special choice of $\vec{A}$ as a unit coordinate vector allows us to identify individual components of $\overline{\text{GRAD}}$. In the discrete case we choose ϕ to be a scalar function in the cell centers and ψ to be a scalar function in the vertices. The discrete analog of the volume integral on the l.h.s. in (2.19) contains the product of ϕ and ψ. Since ϕ and ψ are defined at different locations, some averaging is required [31].

(3) Next we construct $\mathscr{DIV}$ using the discrete analog of identity (2.20) and $\overline{\text{GRAD}}$. This identity defines the projection of vector $\mathscr{DIV}\,\hat{\sigma}$ onto directions defined by the vector $\vec{A}$. The derivation follows a similar logic as in step two with one new feature; we have to discretize $\nabla\vec{A} \equiv \text{grad}\,\vec{A}$. However, this discretization is uniquely determined by the natural requirement that $\mathscr{DIV}(p\hat{I}) = \text{GRAD}\,p$.

(4) Finally, we construct $\mathscr{GRAD}$ using the discrete analog of the integral identity represented by the last equality in (2.18), which guarantees the conservation of total energy in the discrete case. Note, that this identity is a special case of (2.20).

The divergence property of operator $\mathscr{DIV}$, which is needed to enforce conservation of momentum in the discrete model, can be verified from the fact that operator $\overline{\text{GRAD}}$ applied to a constant function vanishes. This latter property follows from step two in our procedure.

Here we present the final formulae for $\mathscr{DIV}\,\hat{\sigma}$ and discrete analog of strain rate tensor $\hat{\dot{\epsilon}} = 0.5[\mathscr{GRAD}\,\vec{u} + (\mathscr{GRAD}\,\vec{u})^*]$ in 2D cylindrical coordinates. A detailed derivation is presented in [31].

In 2D cylindrical coordinates, the stress tensor has the form

$$\hat{\sigma} = \begin{pmatrix} \sigma^{rr} & 0 & \sigma^{zr} \\ 0 & \sigma^{\phi\phi} & 0 \\ \sigma^{zr} & 0 & \sigma^{zz} \end{pmatrix},$$

and the strain rate tensor has the same structure.

The formulae for the components of $\mathscr{DIV}\,\hat{\sigma} = (D^r, D^z)$ are:

$$
\begin{aligned}
D^r_{i+1/2,j+1/2} = & -\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \frac{\partial V_{i+k,j+l}}{\partial r_{i+1/2,j+1/2}} \sigma^{rr}_{i+k,j+l} \\
& +\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \left(\frac{\sigma^{rr}_{i+k,j+l}}{V_{i+k,j+l}} \sum_{s,t=\pm 1/2} \frac{\partial V_{i+k,j+l}}{\partial r_{i+k+s,j+l+t}} \right) \\
& -\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \frac{\partial V_{i+k,j+l}}{\partial z_{i+1/2,j+1/2}} \sigma^{zr}_{i+k,j+l} \\
& -\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \left(\frac{\sigma^{\phi\phi}_{i+k,j+l}}{V_{i+k,j+l}} \sum_{s,t=\pm 1/2} \frac{\partial V_{i+k,j+l}}{\partial r_{i+k+s,j+l+t}} \right),
\end{aligned}
\tag{2.21}
$$

$$
\begin{aligned}
D^z_{i+1/2,j+1/2} = & -\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \frac{\partial V_{i+k,j+l}}{\partial r_{i+1/2,j+1/2}} \sigma^{zr}_{i+k,j+l} \\
& +\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \left(\frac{\sigma^{zr}_{i+k,j+l}}{V_{i+k,j+l}} \sum_{s,t=\pm 1/2} \frac{\partial V_{i+k,j+l}}{\partial r_{i+k+s,j+l+t}} \right) \\
& -\frac{1}{V_{i+1/2,j+1/2}} \sum_{k,l=0,1} \frac{\partial V_{i+k,j+l}}{\partial z_{i+1/2,j+1/2}} \sigma^{zz}_{i+k,j+l}.
\end{aligned}
\tag{2.22}
$$

The formulae for the components of strain rate tensor are:

$$\hat{\epsilon}_{i,j}^{rr} = \frac{1}{V_{i,j}} \sum_{k,l=\pm 1/2} u_{i+k,j+l} \frac{\partial V_{i,j}}{\partial r_{i+k,j+l}} - \frac{1}{4} \left(\sum_{k,l=\pm 1/2} u_{i+k,j+l} \right) \left(\frac{1}{V_{i,j}} \sum_{k,l=\pm 1/2} \frac{\partial V_{i,j}}{\partial r_{i+k,j+l}} \right), \tag{2.23}$$

$$\hat{\epsilon}_{i,j}^{zr} = \frac{1}{2V_{i,j}} \sum_{k,l=\pm 1/2} \left(u_{i+k,j+l} \frac{\partial V_{i,j}}{\partial r_{i+k,j+l}} + v_{i+k,j+l} \frac{\partial V_{i,j}}{\partial z_{i+k,j+l}} \right)$$
$$- \frac{1}{8} \left(\sum_{k,l=\pm 1/2} v_{i+k,j+l} \right) \left(\frac{1}{V_{i,j}} \sum_{k,l=\pm 1/2} \frac{\partial V_{i,j}}{\partial r_{i+k,j+l}} \right), \tag{2.24}$$

$$\hat{\epsilon}_{i,j}^{zz} = \frac{1}{V_{i,j}} \sum_{k,l=\pm 1/2} v_{i+k,j+l} \frac{\partial V_{i,j}}{\partial z_{i+k,j+l}}, \tag{2.25}$$

$$\hat{\epsilon}_{i,j}^{\phi\phi} = \frac{1}{4} \left(\sum_{k,l=\pm 1/2} u_{i+k,j+l} \right) \left(\frac{1}{V_{i,j}} \sum_{k,l=\pm 1/2} \frac{\partial V_{i,j}}{\partial r_{i+k,j+l}} \right). \tag{2.26}$$

2.5. Numerical example: simulating a cratering event

The velocity gradients play a significant role in solid dynamic simulations, where the constitutive law is usually history dependent and is given as a differential relation between the stress rate and the strain rate tensors. When the relation between the time rate of change of volume and the trace of the velocity gradient tensor is not preserved in the discrete model, the situation may arise where a particular cell has been compressed – i.e., its density increases – while the trace of its stress tensor is tensile. In this case, the internal stresses will try to further compress the cell rather than to resist compression by its neighbors. More generally, this situation is not restricted to the trace, but may arise for any component of the stress tensor. The lack of agreement between the geometrical state of the cell and the stress tensor can lead to oscillations and even instability in a calculation.

To illustrate the importance of using compatible approximations for the strain rate tensor, we show results from two calculations first published in [26]. In that period many popular solid dynamic codes used the Cartesian form of the velocity gradients for both plane (x, y) and cylindrical (r, z) coordinates. See, for example [4,47] (in particular [47, p. 83, Eqs. (4.39) and (4.41)]). The two calculations were performed on the SHALE code [9], and differ only in that the first uses a Cartesian form for the strain rates, while the second uses the expressions presented in (2.23)–(2.26).

The physical problem that is simulated is a cratering event in the floor of an oil shale mine. A cylindrical borehole is drilled vertically into the floor, partially filled with a chemical explosive, and the top is tamped. The explosive is detonated from the bottom, generating strong stress waves in the rock. The rock close to the explosive is fractured. The ultimate goal of the calculation is to estimate the size and shape of the fractured region and to predict the size distribution of the rubble that is formed.

The geometry of the simulation is shown in Fig. 2, where the angle striped area represents the chemical explosive region, and all other cells are rock. The right and bottom boundaries are transmitting; the top boundary is a free surface and the left boundary is an axis of symmetry. For the purposes of comparison, we monitor the volumetric strain and strain rate in a rock cell on the cylindrical axis of symmetry, and above the explosive. The choice is motivated first by the fact that the difference between two sets of strain rates is of the order of $\delta r/r$ and hence is the largest near the axis, and second by the fact that the stress wave history is more complex above the explosive.

The "estimated" bulk strain rate (from the Cartesian velocity gradients) is compared with the "exact" bulk strain rate (from the actual change in volume of the cell) in Fig. 3. The chosen cell sees a strong vertical compression; it also compresses more strongly on the axis of symmetry and so sees shear. With the Car-
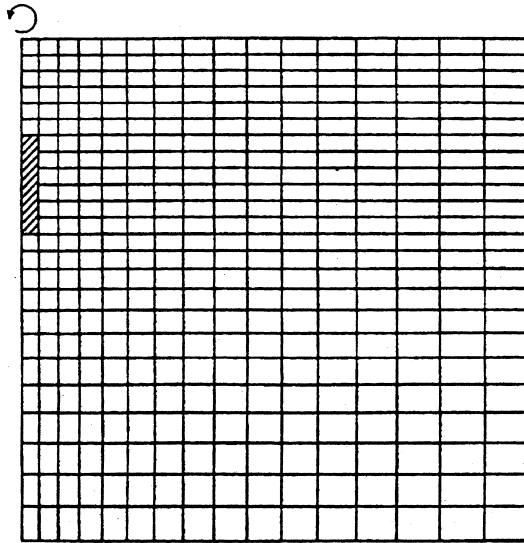
Fig. 2. The initial mesh for the sample problem described in Section 2.5. The cross-hatched region represents a chemical explosive, while all other cells in the mesh are rock. The top of the mesh is a free surface, and the left side of the mesh is a cylindrical axis of symmetry. The right and bottom sides are transmitting boundaries.
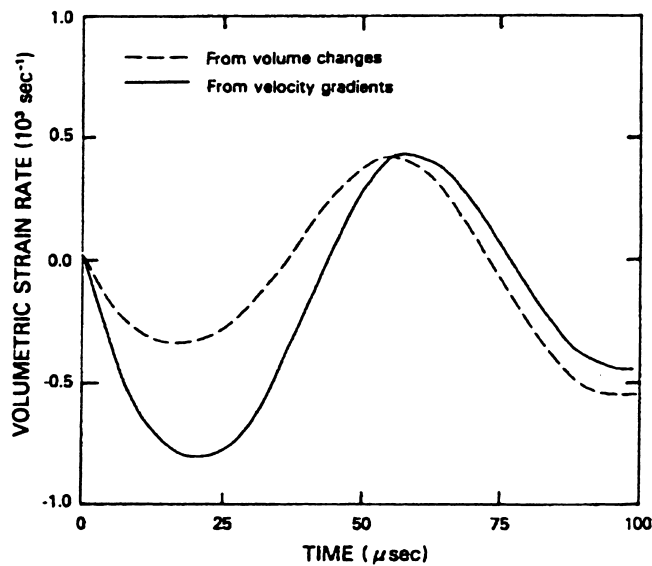
Fig. 3. A comparison of the estimated and the exact volumetric strain rates using the Cartesian form of the velocity gradients. Both curves represent a time history for a cell above the explosive and on the cylindrical axis of symmetry. Note the large discrepancy between the curves at peak compression.

tesian velocity gradients, the estimated peak strain rate is more than twice as large as the exact peak. The estimated and exact strains for the Cartesian calculation are compared in Fig. 4 (these curves are the areas under the curves of Fig. 3). There is a large and systematic discrepancy apparent here. The corresponding comparisons for a calculation using the new strain rates (2.23)–(2.26) are shown in Figs. 5 and 6. In each case these curves, comparing estimated and exact strains and strain rates, lie essentially on the top of each other.
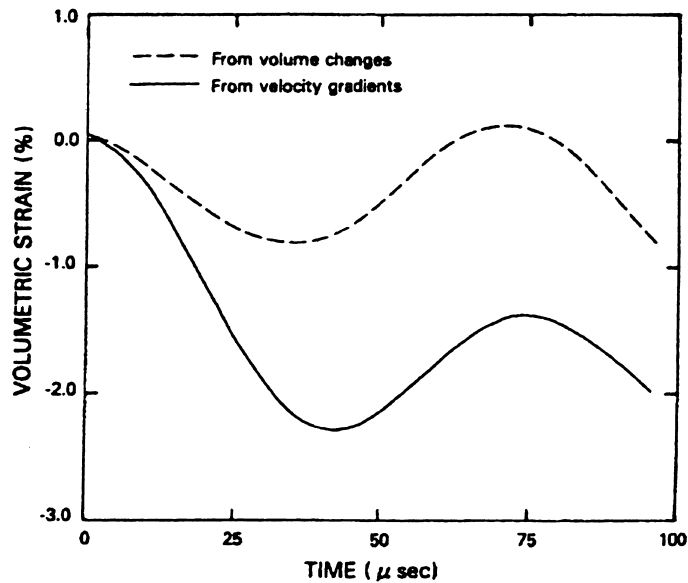
Fig. 4. As in Fig. 3, but comparing the estimated and exact volumetric strains. There is a large and systematic difference between the curves.
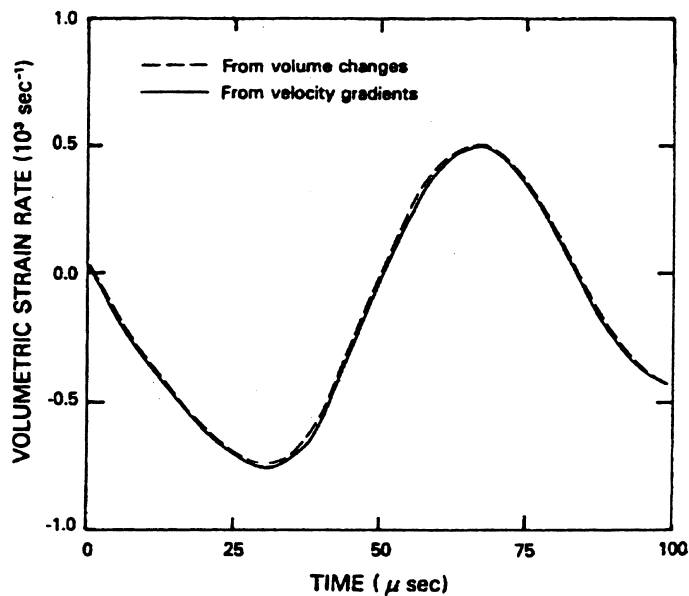

Fig. 5. As in Fig. 3, but with numerical data taken from a simulation using the compatible volumetric strain rates. The two strain rate histories are nearly identical.

2.6. General discussion

We close Section 2 with two summary remarks and references to related work. First, we note that SOM can be applied to a grid of any structure – e.g., logically rectangular, triangular, the Voronoi mesh, etc. In particular, the application of SOM to discretizations for free-Lagrange methods, using Voronoi tesselations to define the cells, is described in [28].

Second, we note that the adjointness of the discrete operators DIV and GRAD allows us to draw conclusions about the linear stability of the method on a general grid. For the semi-discrete equations
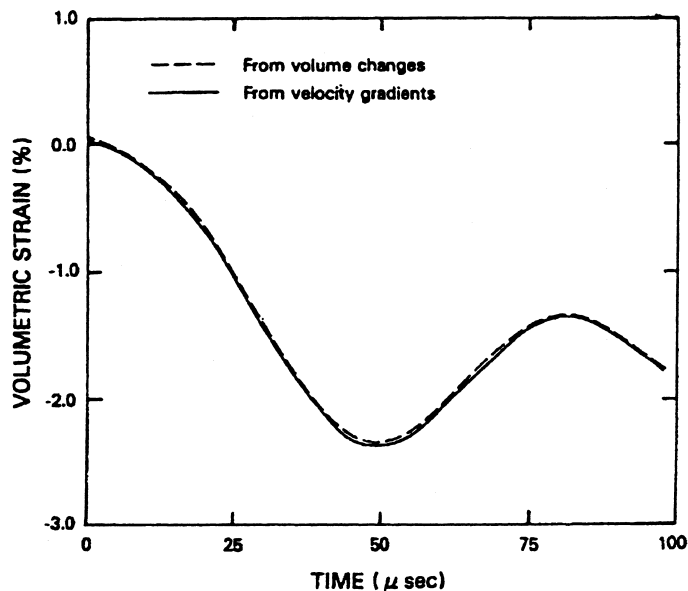
Fig. 6. As in Fig. 3, but comparing the estimated and exact volumetric strains. The two total strain histories are also nearly identical.

(continuous in time), one can use the same energy estimates as apply in the continuous case [1,2,11,12,30]. In addition, when using implicit methods, the adjointness of the discrete operators guarantees that the system of linear equations will be symmetric and positive definite.

An alternate approach, using the conservation of total energy, but choosing GRAD as the prime operator has been described in a series of papers [6–8].

Finally, the material presented above should be viewed as part of a more general effort to create a discrete analog of vector and tensor analysis [16–19].

3. MPDATA

3.1. Introductory remarks

MPDATA is a framework for approximating the advective terms in continuum equations. In general terms, it belongs to the class of nonoscillatory Lax–Wendroff schemes that includes such classical algorithms as FCT [49], TVD [46], and ENO [13]. However, MPDATA is qualitatively different from these other methods. MPDATA's focus is on sign-preserving multidimensional advection rather than on monotone solutions of hyperbolic conservation laws in one spatial dimension. Unlike TVD and ENO schemes, which employ 1D constructions to limit the scalar flux component, MPDATA effectively limits the magnitude of the vector velocity and so is naturally unsplit.

MPDATA is second-order accurate (for sufficiently smooth grids), sign-preserving, conservative, and computationally efficient. It is iterative in nature. The first pass is a simple donor cell approximation, sometimes called upstream differencing, i.e., sign-preserving but only first-order accurate. The second pass increases the accuracy of the calculation by estimating and compensating the (second-order) truncation error of the first pass. Additional passes can be used to estimate the residual error of the previous pass and approximately compensate it. This step may be repeated an arbitrary number of times, leading to successively more accurate solutions of the advection equation.

Originally MPDATA was designed as a simple scheme for handling the transport of nonnegative thermodynamic variables (such as liquid water or water vapor) in atmospheric models [36–38]. Over years, the theory underlying MPDATA has been extended to advection–diffusion equations and to arbitrary curvilinear frameworks [39], to third-order-accurate approximations [27], as well as to a fully monotone

scheme (in the sense of FCT) [43]. More recently, MPDATA has been generalized for systems of equations with arbitrary forcing [41,42]. The utility of MPDATA as a general solver for complex fluid problems has been demonstrated in the context of atmospheric dynamics for both compressible- and incompressible-type formulations of the equations of motion [41,42]. MPDATA has also been used as an interpolator [40] in a class of semi-Lagrangian fluid models, as well as a remapper in arbitrary Lagrangian–Eulerian (ALE) simulations of high-speed flows [3,24].

3.2. A prototype fluid problem

A systematic derivation of general MPDATA beginning with the simplest 1D advection equation can be found in [44]. Here, we summarize the principal results in the context of a generalized transport equation

$$\frac{\partial \Psi}{\partial t} + \operatorname{div}(\vec{u}\Psi) = R, \tag{3.1}$$

where both $\vec{u} = \vec{u}(\vec{x}, t)$ and $R = R(\vec{x}, t)$ are assumed to be the known functions. In fluid dynamics applications, R may represent the pressure gradient forcing or diffusion terms, and $\vec{u}$ is the fluid velocity. In the remapping phase of an ALE method, $\vec{u}$ represents the relative motion of the grid and $R \equiv 0$.

In order to design a fully second-order MPDATA scheme for (3.1), we assume a temporal discretization in the form

$$\frac{\Psi^{n+1} - \Psi^n}{\delta t} + \operatorname{div}(\vec{u}^{n+1/2}\Psi^n) = R^{n+1/2}, \tag{3.2}$$

where δt is the computational timestep and n labels the time level. Expanding (3.2) into a second-order Taylor series about $t = n\delta t$, expressing the resulting temporal derivatives (with accuracy to $\mathcal{O}(\delta t)$] in terms of spatial derivatives (by using (3.2)), and regrouping the terms that do not cancel, leads to the modified equation

$$\frac{\partial \Psi}{\partial t} + \operatorname{div}(\vec{u}\Psi) = R - \operatorname{div}\left[\frac{1}{2}\delta t \vec{u}(\vec{u}, \operatorname{grad}\Psi) + \frac{1}{2}\delta t \vec{u}\Psi \operatorname{div}\vec{u}\right] + \operatorname{div}\left(\frac{1}{2}\delta t \vec{u} R\right) + \mathcal{O}(\delta t^2), \tag{3.3}$$

where all $\mathcal{O}(\delta t)$ errors due to the uncentered time differencing in (3.2) are now expressed by spatial derivatives. Note that, assuming the time levels of both the advective velocity and forcing term are $n + 1/2$ in (3.2) eliminates $\mathcal{O}(\delta t)$ truncation errors proportional to their temporal derivatives in (3.3). Any $\mathcal{O}(\delta t^2)$ approximations to $\vec{u}^{n+1/2}$ and $R^{n+1/2}$ would suffice for second-order accuracy in (3.2); particular approximations are discussed in Section 3.4 of [44].

The $\mathcal{O}(\delta t)$ truncation errors on the r.h.s. of (3.3) have two distinct components. The first is solely due to advection and depends linearly on Ψ. The second is solely due to the forcing and its dependence on Ψ is, in general, unknown. In the following section, we present basic MPDATA for homogeneous transport, while in Section 3.3 we describe how to compensate the errors due to the nonvanishing forcing.

3.3. Homogeneous transport, $R \equiv 0$

The basic idea of MPDATA is to use the sign-preservation property of donor cell schemes to compensate the residual truncation error on the r.h.s. of (3.3). Thus MPDATA consists of a sequence of donor cell steps. In the first pass, the velocity is the physical velocity. In the second and subsequent passes, the velocity is calculated from the field that is being advected and has no physical significance. These velocities are termed antidiffusive, or equivalently pseudo velocities. The pseudo velocities are defined, in part, by representing the expression under the divergence operator in the second term on the r.h.s. of (3.3) as $\vec{u}_{\text{pseudo}}\Psi$. In addition, they must contain similar terms compensating the spatial first-order error of the donor cell approximation itself. An example of the resulting algorithm for a uniform regular grid is described in the following equations:

$$\Psi_{\mathbf{i}}^{(k)} = \Psi_{\mathbf{i}}^{(k-1)} - \sum_{I=1}^{3} \left[F^I(\Psi_{\mathbf{i}}^{(k-1)}, \Psi_{\mathbf{i}+\mathbf{e}_I}^{(k-1)}, V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)}) - F^I(\Psi_{\mathbf{i}-\mathbf{e}_I}^{(k-1)}, \Psi_{\mathbf{i}}^{(k-1)}, V_{\mathbf{i}-1/2\mathbf{e}_I}^{I(k)}) \right], \tag{3.4}$$

where $\mathbf{i} := (i, j, k)$ denotes a location on the grid; $\mathbf{e}_I$ is the unit vector in the Ith spatial direction; F is the donor cell flux function defined below in (3.7) with V^I denoting the normalized advective pseudo velocity in the Ith direction; and $k = 1, \ldots, IORD$ numbers MPDATA iterations such that

$$\Psi^{(0)} := \Psi^n, \tag{3.5a}$$

$$\Psi^{(IORD)} := \Psi^{n+1}, \tag{3.5b}$$

$$V^{I(k+1)} = V^I\left(\vec{V}^{(k)}, \Psi^{(k)}\right), \tag{3.6a}$$

$$V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(1)} := u^I|_{\mathbf{i}+1/2\mathbf{e}_I}^{n+1/2} \frac{\delta t}{\delta x^I}. \tag{3.6b}$$

The flux function

$$F(\Psi_L, \Psi_R, U) := [U]^+ \Psi_L + [U]^- \Psi_R \tag{3.7}$$

depends on the nonnegative and nonpositive ($[\ ]^+$ and $[\ ]^-$, respectively) parts of the normalized velocity (a local Courant number) $U = u\delta t/\delta x$ where δx stands for the length of a cell. With this notation, $IORD = 1$ variant of MPDATA is the classical donor cell scheme, and various options of MPDATA differ merely by specifics of the functional form of the pseudo velocity (3.6a). A basic representation of (3.6a) takes the form

$$V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k+1)} = \left[|V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)}| - (V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)})^2\right] \frac{\Psi_{\mathbf{i}+\mathbf{e}_I}^{(k)} - \Psi_{\mathbf{i}}^{(k)}}{\Psi_{\mathbf{i}+\mathbf{e}_I}^{(k)} + \Psi_{\mathbf{i}}^{(k)}}$$
$$- \sum_{J=1;J\neq I}^{3} \frac{1}{2} V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)} \overline{V_{\mathbf{i}+1/2\mathbf{e}_I}^{J(k)}} \frac{\Psi_{\mathbf{i}+\mathbf{e}_I+\mathbf{e}_J}^{(k)} + \Psi_{\mathbf{i}+\mathbf{e}_J}^{(k)} - \Psi_{\mathbf{i}+\mathbf{e}_I-\mathbf{e}_J}^{(k)} - \Psi_{\mathbf{i}-\mathbf{e}_J}^{(k)}}{\Psi_{\mathbf{i}+\mathbf{e}_I+\mathbf{e}_J}^{(k)} + \Psi_{\mathbf{i}+\mathbf{e}_J}^{(k)} + \Psi_{\mathbf{i}+\mathbf{e}_I-\mathbf{e}_J}^{(k)} + \Psi_{\mathbf{i}-\mathbf{e}_J}^{(k)}}$$
$$- \frac{1}{4} V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)} \left(V_{\mathbf{i}+3/2\mathbf{e}_I}^{I(k)} - V_{\mathbf{i}-1/2\mathbf{e}_I}^{I(k)}\right)$$
$$- \frac{1}{4} V_{\mathbf{i}+1/2\mathbf{e}_I}^{I(k)} \sum_{J=1;J\neq I}^{3} \left(V_{\mathbf{i}+\mathbf{e}_I+1/2\mathbf{e}_J}^{J(k)} + V_{\mathbf{i}+1/2\mathbf{e}_J}^{J(k)} - V_{\mathbf{i}+\mathbf{e}_I-1/2\mathbf{e}_J}^{J(k)} - V_{\mathbf{i}-1/2\mathbf{e}_J}^{J(k)}\right), \tag{3.8a}$$

where

$$\overline{V_{\mathbf{i}+1/2\mathbf{e}_I}^{J(k)}} \equiv \frac{1}{4} \left(V_{\mathbf{i}+\mathbf{e}_I+1/2\mathbf{e}_J}^{J(k)} + V_{\mathbf{i}+1/2\mathbf{e}_J}^{J(k)} + V_{\mathbf{i}+\mathbf{e}_I-1/2\mathbf{e}_J}^{J(k)} + V_{\mathbf{i}-1/2\mathbf{e}_J}^{J(k)}\right). \tag{3.8b}$$

These expressions are valid only for grids that are uniform in each direction. To extend these formulae for irregular grids several strategies are possible. The general hexahedral cell can be mapped onto a cube. This procedure introduces metric coefficients in formulae (3.1)–(3.4), (3.5a), (3.5b), (3.6a), (3.6b), (3.7), and (3.8a), (3.8b), see [44]. When the grid is sufficiently smooth the overall algorithm remains second-order-accurate. For nonsmooth grids, a more general truncation error analysis is required, leading to generalized formulae for the pseudo velocities.

So far, we have assumed that the transport field Ψ has a constant sign. This assumption is important in proving that the pseudo velocities are bounded, and hence that MPDATA is stable. Note that the pseudo velocity includes terms like $\delta\Psi/\sum\Psi$ which are used to approximate the $(1/\Psi)(\partial\Psi/\partial x^I)$ ratios. These terms are bounded when Ψ has a constant sign. Otherwise, they are unbounded leading to arbitrarily large pseudo velocities and unstable schemes. MPDATA can be extended to the transport of variable-sign fields in a number of ways. The simplest and most common way is to replace each Ψ in (3.8a) and (3.8b) by $|\Psi|$, using the relationship $\frac{1}{\Psi}\frac{\partial\Psi}{\partial x^I} = \frac{1}{|\Psi|}\frac{\partial|\Psi|}{\partial x^I}$, a.e. Other possibilities are described in Section 3.2.4 of [44].

The basic algorithm preserves sign but not monotonicity of the transported variables and, in general, the solutions are not free of unphysical extrema. In most cases preservation of sign is adequate. When required, MPDATA can be made fully monotone by employing FCT formalism [49] to limit the pseudo velocities. In fact, MPDATA is very well suited for this for a number of reasons. First, the initial MPDATA iteration is the donor cell scheme – a low-order monotone scheme commonly used as the reference in the FCT design. Second, assuring monotonicity of 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. Third, since all MPDATA iterations have similar low phase errors characteristic of the donor cell scheme, the FCT procedure mixes solutions with consistent phase errors. This benefits significantly the overall accuracy of the resulting FCT scheme (see Fig. 5 in [43] and the accompanying discussion).

3.4. Inhomogeneous transport, $R \neq 0$

If MPDATA is to be used for a fully Eulerian simulation, it is necessary to compensate the $\mathcal{O}(\delta t)$ truncation error in (3.2) dependent on the advective fluxes of the source term; in such simulations R may represent either the gradient of pressure, or the divergence of the stress tensor. This particular error term appears in all approximations to (3.1) that simply combine a time-uncentered advection scheme for homogeneous transport with a second-order approximation of R. Ignoring this error leads to spurious $\sim \mathcal{O}(\delta t)$ sinks/sources of energy and, eventually, to nonlinear instability [41]). Compensating this error to $\mathcal{O}(\delta t^2)$ only requires subtracting a first-order-accurate approximation from the r.h.s. of (3.2).

As an example assume that $R^{n+1/2}$ can be written as $0.5(R^n + R^{n+1})$ where R^{n+1} is an $\mathcal{O}(\delta t^2)$ accurate approximation to R at $t = (n+1)\delta t$. Then a simple, efficient, and fully second-order-accurate MPDATA realization of (3.2) can be compactly written as

$$\Psi_{\mathbf{i}}^{n+1} = \text{MPDATA}_{\mathbf{i}}(\Psi^n + 0.5\delta t R^n, \vec{V}^{n+1/2}) + 0.5\delta t R_{\mathbf{i}}^{n+1}. \tag{3.9}$$

In the above equation, MPDATA symbolizes the homogeneous-transport algorithm discussed in Section 3.2. Advecting the auxiliary field $\Psi^n + 0.5\delta t R^n$ not only compensates the truncation error due to the source term but also has the physical interpretation of integrating the forces along a parcel trajectory rather than at the grid point. This makes (3.9) equivalent (to the second-order) to ALE methods, where the equations of motion are integrated along Lagrangian trajectories [14,25]. The particular compensation (3.9) may be viewed as an application of Strang splitting [45]. Other possibilities are described in [44] and the references therein.

3.5. Numerical example: transport of the rotating cone

For illustration, Fig. 7 displays the results of a standard solid-body rotation test (cf. [44]) using selected variants of MPDATA. The 2D rotation test employs a square mesh of 101 by 101 points. The angular velocity $\omega = 0.1$ and the velocity components are $(u, v) = -\omega(y - y_0, x - x_0)$. The center of rotation (x_0, y_0) is the center of the mesh $(50\delta x, 50\delta x)$. The maximum Courant number $(|u\delta t/\delta x| + |v\delta t/\delta x|)$ is 0.99, and one full rotation requires 628 timesteps. The initial condition is a cone centered at the point $(75\delta x, 50\delta y)$ and has a base diameter of 30 and a height of 4. Fig. 7(a)–(c) show the analytic solution, the first-order donor cell solution, and the second-order solution using basic two-pass MPDATA, all after six rotations. Fig. 7(d) displays the state-of-the-art third-order-accurate MPDATA option with the analytic summation of the infinite series of corrective iterations [27]. The schemes in Figs. 7(c) and (d) use approximately 2.5 and 6.0 times as much computer time as the donor cell algorithm.

4. Combining SOM and MPDATA in Eulerian simulations

4.1. Compatible differencing in 1D

The presence of the flux terms in Eulerian simulations imposes additional requirements if the discrete equations are to exactly conserve energy. These requirements can be expressed as relationships between the
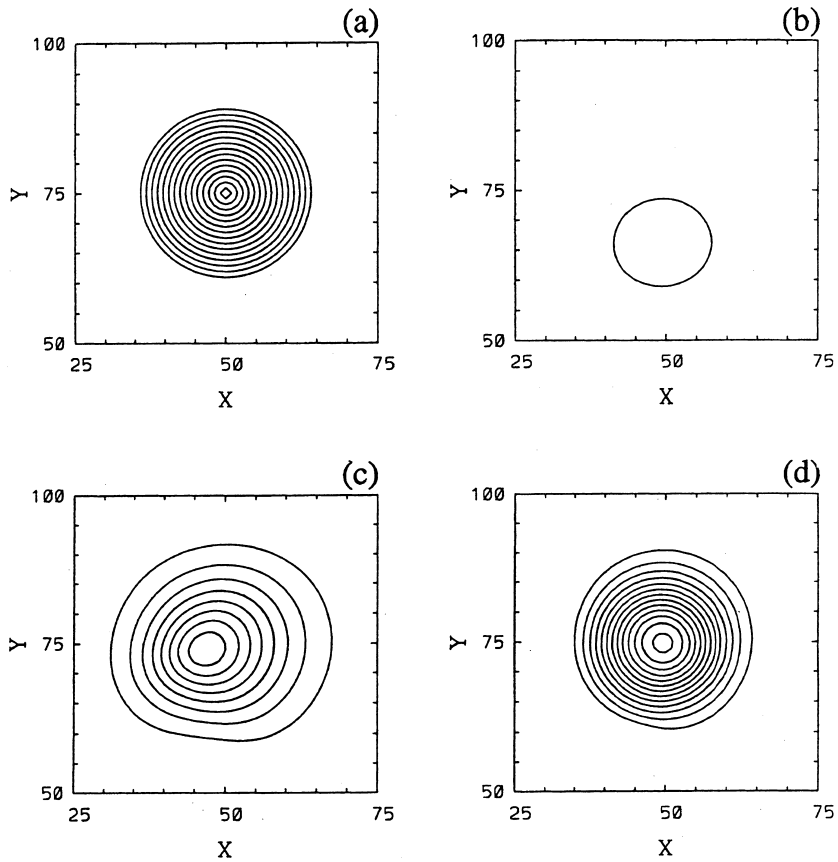
Fig. 7. Isolines of cone advected through six rotations (3768 timesteps) using different variants of MPDATA. The contour interval is 0.25, and the zero contour line is not shown. Plate (a) shows the analytic solution (identical to the initial condition), plate (b) the result using the donor cell scheme, plate (c) the basic MPDATA scheme, and plate (d) the most accurate MPDATA option discussed in [27].

various fluxes (mass, momentum and energy). The goal of this section is to develop a combined Eulerian algorithm utilizing both SOM and MPDATA in the context of shallow fluid flows. In these flows our combined method uses MPDATA to compute a depth flux and then uses SOM to define the momentum flux in terms of the depth flux.

To motivate our algorithm in 2D, we consider the 1D shallow water equations with flat bottom

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0, \tag{4.1}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2) = -gh\frac{\partial h}{\partial x}, \tag{4.2}$$

where h is the fluid depth, u is the fluid velocity, and g is the acceleration of gravity. If no work is done at the boundaries, these equations conserve the total energy E:

$$E = \int_V \left(h\frac{u^2}{2} + g\frac{h^2}{2} \right) dx. \tag{4.3}$$

We discretize Eqs. (4.1) and (4.2) in space, assuming that both h and u are defined at the cell centers, and keep time continuous

$$\frac{\partial h_i}{\partial t} + \frac{F^h_{i+1/2}(u_{i+1/2}) - F^h_{i-1/2}(u_{i-1/2})}{\delta x} = 0, \tag{4.4}$$

$$\frac{\partial (hu)_i}{\partial t} + \frac{F^{hu}_{i+1/2}(u_{i+1/2}) - F^{hu}_{i-1/2}(u_{i-1/2})}{\delta x} = -gh_i(\text{GRAD}\,h)_i \tag{4.5}$$

where $F^h_{i+1/2}(u_{i+1/2})$ is a generic depth flux and $F^{hu}_{i+1/2}(u_{i+1/2})$ a generic momentum flux. Here $u_{i+1/2} = 0.5(u_{i+1} + u_i)$ is the velocity of the node. The GRAD at this point is also a generic form of the discrete gradient.

The discrete total energy E_d can be defined

$$E_d = \sum_{\text{cells}} \left(h_i \frac{u_i^2}{2} + g \frac{h_i^2}{2} \right) \delta x. \tag{4.6}$$

From (4.4) and (4.5) one can derive the time rate of change of the discrete energy

$$\frac{\partial E_d}{\partial t} = \sum_{\text{cells}} \left\{ \left[\frac{1}{2} u_i^2 \left(\frac{F^h_{i+1/2}(u_{i+1/2}) - F^h_{i-1/2}(u_{i-1/2})}{\delta x} \right) - u_i \left(\frac{F^{hu}_{i+1/2}(u_{i+1/2}) - F^{hu}_{i-1/2}(u_{i-1/2})}{\delta x} \right) \right] \right.$$
$$\left. - \left[gh_i \frac{F^h_{i+1/2}(u_{i+1/2}) - F^h_{i-1/2}(u_{i-1/2})}{\delta x} + gu_i h_i(\text{GRAD}\,h)_i \right] \right\}. \tag{4.7}$$

When the discrete depth and momentum fluxes are compatible in the sense

$$F^{hu}_{i+1/2}(u_{i+1/2}) = u_{i+1/2} F^h_{i+1/2}(u_{i+1/2})$$

the expression in the first square brackets in (4.7) has conservative form – i.e., the sum over all cells reduces to boundary terms. When a specific form (such as MPDATA) is used for both F^h and F^{hu}, the two fluxes are not compatible and that expression does not have conservative form, which implies that discrete energy will not be conserved exactly (it will be conserved to the order of the truncation error).

Let us now analyze the expression in the second square brackets in (4.7). Its analytical analog is the total derivative $\partial(h^2 u)/\partial x$, whose integral depends only on the boundary conditions. In principle, the requirement that the expression in the second square brackets be in conservative form constitutes a definition of GRAD. For fluxes linear in the velocity – e.g., donor cell – this leads to an explicit form of h GRAD, which is the form required in (4.2). We know from Godunov's theorem that all second-order nonoscillatory advection schemes must be nonlinear in the velocity [10]; however because of the consistency of the approximation, a linearized form (in velocity) of the fluxes should be sufficient to derive an explicit form of h GRAD.

In the calculations that follow we have extended some of these ideas to 2D shallow fluid flow on the sphere [44]. In particular, we define the momentum flux in terms of the depth flux. However, we continue to use a central difference form for GRAD. Although this compromise will not yield exact conservation, it will improve the conservation properties. Moreover, there are additional advantages: first, because we use MPDATA only for transport of depth, we avoid the problem of transporting fields of variable sign (components of the momentum), (see Section 3.3); second, because we apply MPDATA only once, as compared with three applications (flux of depth and two components of the momentum) in traditional MPDATA, the new algorithm is much more efficient.

4.2. Numerical example: 2D shallow fluid flow on the sphere

For illustration, we repeat simulations of the Rossby–Haurwitz wave evolution [48] on the sphere discussed in [44]. The equations expressing conservation of mass and momentum in a shallow fluid flow on a rotating sphere each has the form of the generalized transport Eq. (3.1). The forcing term is 0 in the mass continuity equation. The momentum equation, in addition to the standard pressure gradient term, also

includes Coriolis acceleration (due to Earth's rotation) and the inertial metric force (due to representing momenta in terms of velocities measured in a local tangent Cartesian framework aligned with standard geographical coordinates; see [44] for details). The uniform, unstaggered mesh consists of 128 points in longitude, and 64 points in latitude. The timestep $\delta t = 40$ s is limited by the propagation speed of the gravity wave. Fig. 8(a) shows the initial condition. Over several days, we expect this initial pattern to move from west to east with little change of shape (except for a slight steepening of the wave in mid-latitudes) and angular velocity $\sim 2.5 \times 10^{-6}$ rad s^{-1}.

Fig. 8(b) shows the numerical solution after 15 days of integration using MPDATA for mass continuity equation and a compatible scheme for momentum (in the spirit of Section 4.1). The corresponding solution generated with MPDATA used for all variables (not shown) is indistinguishable in the figure. Insofar as the overall accuracy is concerned, a quantitative analysis (Table 1) corroborates that the differences between the two solutions are small; for the sake of reference, the solution generated with the first-order-accurate MPDATA (i.e., the upwind scheme) is also included. The compatible scheme has slightly lesser numerical dissipation, as judged by the velocity amplitude, and noticeably smaller conservation errors of energy and enstrophy. The striking difference is in the computational economy, with the compatible approach being about twice cheaper. Although the upwind solution is even less expensive, it is essentially useless as the wave dissipates into a zonal flow [Fig. 8(c)].
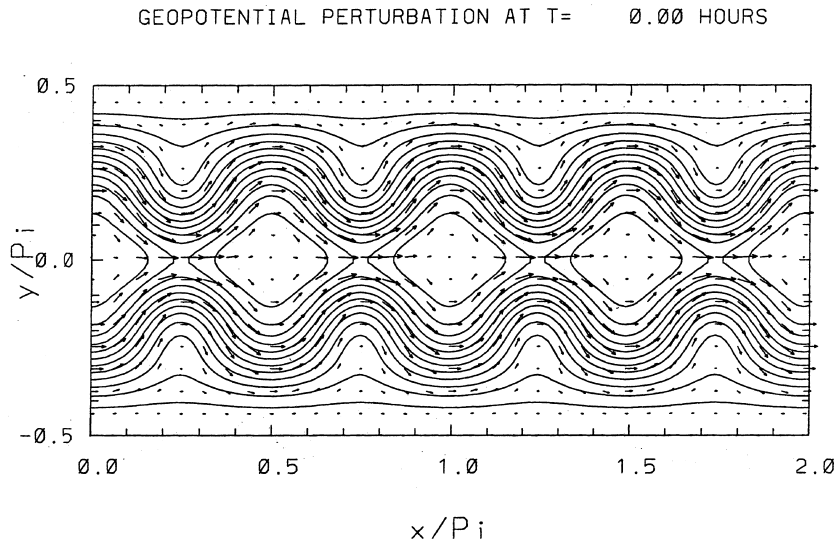

Fig. 8. Free surface height perturbation ($H/\bar{H} - 1$, where $\bar{H} = 8$ km) and the flow vectors for the Rossby–Haurwitz wave test problem: (a) the initial condition; (b) the numerical solution after 15 days using a second-order-accurate MPDATA scheme for mass and a compatible scheme for momentum; (c) the equivalent solution using a first-order-accurate upwind scheme. The contour interval is 0.05.

Table 1
Comparison of the two schemes after simulated five days of the evolution of the Rossby–Haurwitz wave on the sphere[a]

TABLES run	$\|\mathbf{V}\|_\infty$	Mass	Energy	Enstrophy	CPU
MPDATA	97.45	$\sim 10^{-12}$	-5.6×10^{-5}	-1.1×10^{-3}	2633
COMPT.	99.37	$\sim 10^{-12}$	-1.0×10^{-5}	6.3×10^{-4}	1407
UPWIND	52.21	$\sim 10^{-12}$	-1.8×10^{-2}	-1.3×10^{-1}	704

[a] The first row is for the run using MPDATA for all variables, whereas the second row is for the run using MPDATA for mass and the compatible scheme for momentum. For reference, the third row contains data from the run using a first-order-accurate donor cell scheme. The first column characterizes the scheme employed. The second column lists maximal speeds in m s^{-1}. Columns three to five list relative conservation errors of mass, total energy, and enstrophy, respectively. The last column contains CPU times in seconds.

Acknowledgements

References

[1] N.V. Ardelyan, The Convergence of difference Schemes for two-dimensional equations of acoustics and Maxwell's equations, USSR Comput. Math. Math. Phys. 23 (1983) 93–99.

[2] N.V. Ardelyan, S.V. Chernigovskii, Convergence of difference schemes for two-dimensional gas-dynamic equations in acoustics approximations with gravitation taken into account, Diff. Eqns. 28 (1988) 57–63.

[3] C.W. Beason, L.G. Margolin, DPDC: a second-order monotone scheme for advection, Report UCRL-99731, Lawrence Livermore National Laboratory, 1988.

[4] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Eng. 99 (1991) 235–394. See in particular p. 250 (Eq. 2.4.2.6).

[5] P. Bernardet, The pressure term in the anelastic model: a symmetric elliptic solver for a C-Arakawa grid in generalized coordinates, Mon. Wea. Rev. 123 (1995) 2474.

[6] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.

[7] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulation of artificial viscosity for multi-dimensional shock wave problems, J. Comput. Phys. 144 (1998) 70–97.

[8] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998), 227–262.

[9] R.B. Demuth, L.G. Margolin, B.D. Nichols, T.F. Adams, and B.W. Smith, SHALE: a computer program for solid dynamics, Report LA-10236, Los Alamos National Laboratory, 1985.

[10] S.K. Godunov, A difference scheme for numerical computation of discontinuous solutions of equations in fluid dynamics, Math. Sbornik 47 (1959) 271–306.

[11] V.M. Goloviznin, V.K. Korshunov, A. Sabitova, E.A. Samarskaya, Stability of variational-difference schemes in gas-dynamics, Diff. Eqns. 20 (1984) 852–858.

[12] B. Gustafsson, H.O. Kreiss, J. Oliger, Time Dependent Problems and Difference Methods, Chapter 11, Wiley/Interscience, New York, 1995, pp. 445–495.

[13] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high-order accurate essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231–303.

[14] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) 227.

[15] J. Hyman, M. Shashkov, S. Steinberg, The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials, J. Comput. Phys. 132 (1997) 130–148.

[16] J.M. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient, and curl on logically rectangular grids, Comput. Math. Appl. 33 (1997) 81–104.

[17] J.M. Hyman, M. Shashkov, The adjoint operators for natural discretizations for the divergence gradient and curl on logically rectangular grids, IMACS J. Appl. Numer. Math. 25 (1997) 413–442.

[18] J.M. Hyman and M. Shashkov, The orthogonal decomposition theorems for mimetic finite difference methods, SIAM J. Numer. Anal. 36 (1999) 788–818.

[19] J.M. Hyman, M. Shashkov, The approximation of boundary conditions for mimetic finite difference methods, Comput. Math. Appl. 36 (1998) 79–99.

[20] J.M. Hyman, M. Shashkov, Mimetic discretizations for Maxwell's equations and equations of magnetic diffusion, J. Comput. Phys. 151 (1999) 881–909.

[21] J.M. Hyman, M. Shashkov, Mimetic discretizations for Maxwell's equations and equations of magnetic diffusion, in: J.A. DeSanto (Ed.), Mathematical and and Numerical Aspects of Wave Propagations, Proceedings of the Fourth International Conference on Mathematical and Numerical Aspects of Wave Propagation, Golden, Colorado, SIAM, Philadelphia, 1–5 June 1998, pp. 561–563.

[22] T. Korshiya, V. Tishkin, A. Favorskii, M. Shashkov, Variational approach to the construction of finite-difference schemes for the diffusion equations for magnetic field, Diff. Eqns. 18 (1982) 863–872.

[23] T. Korshiya, V. Tishkin, A. Favorskii, M. Shashkov, A variational approach to the construction of difference schemes on curvilinear meshes for the heat-conduction equation, USSR Comput. Maths. Math. Phys. 20 (1980) 135–155.

[24] C.E. Leith, L.G. Margolin, Turbulence modeling in the SHALE code, Report UCRL-ID-103271, Lawrence Livermore National Laboratory, 1990.

[25] L.G. Margolin, Introduction to an arbitrary Lagrangian–Eulerian computing methods for all flow speeds, J. Comput. Phys. 135 (1997) 198–202.

[26] L.G. Margolin, T.F. Adams, Spatial differencing for finite difference codes, Report LA-10249, Los Alamos National Laboratory, 1985.

[27] L.G. Margolin, P.K. Smolarkiewicz, Antidiffusive velocities for multipass donor cell advection, Report UCID-21866, Lawrence Livermore National Laboratory 1989, SIAM J. Sci. Comput., to appear.

[28] N. Mikhailova, V. Tishkin, N. Turina, A. Favorskii, M. Shashkov, Numerical modeling of two-dimensional gas-dynamic flows on a variable-structure mesh, USSR Comput. Maths. Math. Phys. 26 (1986) 74–84.

[29] J.E. Morel, R.M. Roberts, M. Shashkov, A Local Support-operator diffusion discretization scheme for quadrilateral r–z meshes, J. Comput. Phys. 144 (1998) 17–51.

[30] A. Sabitova, E.A. Samarskaya, Stability of variational-difference schemes for the problems of gas-dynamics with heat conduction, Diff. Eqns. 21 (1985) 861–864.

[31] A. Samarskii, V. Tishkin, A. Favorskii, M. Shashkov, Employment of the reference-operator method in the construction of finite-difference analogs of tensor operations, Diff. Eqns. 18 (1982) 881–885.

[32] A. Samarskii, V. Tishkin, A. Favorskii, M. Shashkov, Operational finite-difference schemes, Diff. Eqns. 17 (1981) 854–862.

[33] M. Shashkov, S. Steinberg, Support-operators finite-difference algorithms for general elliptic problems, J. Comput. Phys. 118 (1995) 131–151.

[34] M. Shashkov, Conservative Finite-Difference Methods on General Grids, CRC Press, Boca Raton, FL, 1995.

[35] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1996) 383–405.

[36] P.K. Smolarkiewicz, A simple positive definite advection scheme with small implicit diffusion, Mon. Wea. Rev. 111 (1983) 479–486.

[37] P.K. Smolarkiewicz, A fully multidimensional positive definite advection transport algorithm with small implicit diffusion, J. Comput. Phys. 54 (1984) 325–362.

[38] P.S. Smolarkiewicz, T.L. Clark, The multidimensional positive definite advection transport algorithm: further development and applications, J. Comput. Phys. 67 (1986) 396–438.

[39] P.K. Smolarkiewicz, On forward-in-time differencing for fluids, Mon. Wea. Rev. 119 (1991) 2505–2510.

[40] P.K. Smolarkiewicz, G.A. Grell, A class of monotone interpolation schemes, J. Comput. Phys. 101 (1992) 431–440.

[41] P.K. Smolarkiewicz, L.G. Margolin, On forward-in-time differencing for fluids: extension to a curvilinear framework, Mon. Wea. Rev. 121 (1993) 1847–1859.

[42] P.K. Smolarkiewicz, L.G. Margolin, On forward-in-time differencing for fluids: an Eulerian/semi-Lagrangian nonhydrostatic model for stratified flows, Atmos-Ocean Special 35 (1997) 127–157.

[43] P.K. Smolarkiewicz, W.W. Grabowski, The multidimensional positive definite advection transport algorithm: nonoscillatory option, J. Comput. Phys. 86 (1990) 355.

[44] P.K. Smolarkiewicz, L.G. Margolin, MPDATA: a finite-difference solver for geophysical flows, J. Comput. Phys. 140 (1998) 459.

[45] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506.

[46] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (1984) 995–1011.

[47] S.L. Thompson, CSQ – a two-dimensional hydrodynamic program with energy flow and material strength, Report SAND74-0122, Sandia National Laboratories, 1974.

[48] D.L. Williamson, J.B. Drake, J.J. Hack, R. Jakob, P.N. Swarztrauber, A standard test set for numerical approximations to the shallow water equations on the sphere, J. Comput. Phys. 102 (1992) 211–224.

[49] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.