Quasi-Monotone Advection Schemes Based on Explicit Locally Adaptive Dissipation

ALEXANDER F. SHCHEPETKIN AND JAMES C. MCWILLIAMS

Institute of Geophysics and Planetary Physics, University of California, Los Angeles, Los Angeles, California

(Manuscript received 25 February 1997, in final form 17 June 1997)

ABSTRACT

The authors develop and test computational methods for advection of a scalar field that also include a minimal dissipation of its variance in order to preclude the formation of false extrema. Both of these properties are desirable for advectively dominated geophysical flows, where the relevant scalars are both potential vorticity and material concentrations. These methods are based upon the sequential application of two types of operators: 1) a conservative and nondissipative (i.e., preserving first and second spatial moments of the scalar field), directionally symmetric advection operator with a relatively high order of spatial accuracy; and 2) a locally adaptive correction operator of lower spatial accuracy that eliminates false extrema and causes dissipation. During this correction phase the provisional distribution of the advected quantity is checked against the previous distribution, in order to detect places where the previous values were overshot, and thus to compute the excess. Then an iterative diffusion procedure is applied to the excess field in order to achieve approximate monotone behavior of the solution.

In addition to the traditional simple flow tests, we have made long-term simulations of freely evolving two-dimensional turbulent flow in order to compare the performance of the proposed technique with that of previously known algorithms, such as UTOPIA and FCT. This is done for both advection of vorticity and passive scalar. Unlike the simple test flows, the turbulent flow provides nonlinear cascades of quadratic moments of the advected quantities toward small scales, which eventually cannot be resolved on the fixed grid and therefore must be dissipated. Thus, not only the ability of the schemes to produce accurate shape-preserving advection, but also their ability to simulate subgrid-scale dissipation are being compared. It is demonstrated that locally adaptive algorithms designed to avoid oscillatory behavior in the vicinity of steep gradients of the advected scalars may result in overall less dissipation, yet give a locally accurate and physically meaningful solution, whereas algorithms with built-in hyperdiffusion (i.e., those traditionally used for direct simulation of turbulent flows) tend to produce a locally unsufficient and, at the same time, globally excessive amount of dissipation. Finally, the authors assess the practial trade-offs required for large models among the competing attributes of accuracy, extrema preservation, minimal dissipation (e.g., appropriate to large Reynolds numbers), and computational cost.

1. Introduction

The most energetic geophysical fluid motions occur at large spatial scales, hence have a large Reynolds number, and thus often have advection as their dominant evolutionary process. Furthermore, because of the various constraints of planetary rotation, stable density stratification, and the mean shear and magnetic fields, these motions are often anisotropic in the sense that the horizontal advection (parallel to the earth's surface) dominates the vertical. Finally, fluid trajectories typically span less than a density scale height, so that the mass field is approximately incompressible. Therefore, skillful numerical methods for calculating advection by an approximately two-dimensional, nondivergent velocity are of great geophysical importance.

Corresponding author address: Dr. Alexander F. Shchepetkin, Institute of Geophysics and Planetary Physics, University of California, Los Angeles, 3845 Slichter Hall, Box 951567, Los Angeles, CA 90095-1567.
E-mail: alex@atmos.ucla.edu

a. Advection and dissipation

Advection schemes that conserve energy (essentially a quadratic integral functional of the solution field), enstrophy (mean-square vorticity or potential vorticity), and quadratic variance of passive scalar have been used widely in geophysical computational fluid dynamics (Lilly 1965; Arakawa 1966; Arakawa and Lamb 1981; Arakawa and Hsu 1990; Sadourny 1975). These methods usually have had only second-order spatial accuracy, perhaps because of the difficulty of proving their conservation properties. The conservation properties guarantee stability of the algorithms suitable for long-term simulations. However, due both to dispersive computational errors (i.e., where different Fourier components in the advected field propagate differentially, giving rise to oscillatory patterns) and to advectively induced turbulent cascades, these methods allow accumulations of variance at the smallest resolved scales (i.e., the grid scale). This variance must be controlled through smoothing of the fields, hence dissipation of the advectively conserved quantities, in order to avoid a physically false evolution toward equipartition states (Bennett and Haidvogel 1983). This is typically accom-

plished through additional Laplacian or higher-order diffusion operators or low-pass spatial filters. Grid-scale dissipation is a necessary artifice in lieu of the continuing cascades of variance to the actual dissipation by molecular collisions at scales much finer than are computationally feasible. Because of this, it is plausible that minimizing the computational dissipation, while avoiding an equipartition catastrophe, improves the geophysical realism of the simulation.

In contrast, the advection schemes developed for engineering practices have focused primarily on accuracy, shape preservation of the profile of the advected quantity, and handling of discontinuities, while conservation of quadratic invariants has not been a design goal (Boris and Book 1973; Zalesak 1979; van Leer 1979; van Leer 1985; Colella and Woodward 1984). In these schemes, the necessary dissipation is accomplished implicitly through the form of their discretization errors. Some recent geophysical applications of these schemes have been shown to produce smaller overall dissipation and greater accuracy than quadratic-conservative schemes combined with explicitly diffusive dissipation [e.g., Bennet and Cummins (1988); Carpenter et al. (1990); Farrow and Stevens (1995); Hecht et al. (1995); Leonard et al. (1995b)].

A nondissipative, finite-difference advection scheme is an unachievable goal—even in the absence of physically induced dissipation—because the introduction of a fixed grid violates Galilean invariance, resulting in false oscillations associated with the dispersive errors. As the result, the dissipative mechanisms designed to control these nonphysical effects are also allowed to violate Galilean invariance (Boris 1989). These mechanisms include viscosity proportional to the local velocity (Hyman 1979); nonlinear viscosity, which switches on and off, depending on the behavior of the solution (Forester 1977); hyperdiffusion inherent for upstream-biased schemes (Leonard 1979; Davies 1980; Tremback et al. 1987; Rasch 1994); and nonlinear, gradient-limiting, and flux-limiting algorithms (Zalesak 1979; Colella and Woodward 1984; Sweby 1985; Bott 1989; Colella 1990; Odstrvcil 1990; Leonard and Niknafs 1991; LeVeque 1992; Chlond 1994; Thuburn 1996).

All dissipative algorithms may be subdivided into two major groups:

1) *Scale-selective* or *weak* dissipation relies on an even-order derivative as the dominant term in either the explicit smoothing operator or the implicit truncation error. Disregarding a first-order upstream scheme and Laplacian diffusion (the classical fluid-dynamical representation of molecular dissipation), both of which are too dissipative for simulation of convectively dominated flows, the lowest-order example is a third-order scheme with a fourth-order derivative providing the route to dissipation. The great majority of semi-Lagrangian schemes have dissipation of this kind [e.g., McCalpin (1988); Raymond (1994)], and higher-order schemes are seldom used. The term ''weak'' is used because the dissipation formally vanishes when grid size goes to 0 for a fixed degree of smoothness of the solution.

2) *Location-selective*, *locally adaptive*, or *strong* dissipation relies on locally imposed dissipation sufficient enough to suppress dispersive overshoots in regions of strong curvature in the advected quantity. This requires the capability of the algorithm to locally lower its local order of accuracy to the level of first-order upstream advection, although the overall amount of dissipation is much less than with a uniformly applied first-order upstream scheme. These algorithms are often referred to as flux limiting, but in general it may be in done in many different ways: for example, local explicit diffusion (Forester 1977); gradient limiting (van Leer 1979); imposing total variance diminishing (TVD) constraints, (Sweby 1985; LeVeque 1992); constraining the interpolation polynomials for monotonicity preservation, (Colella and Woodward 1984); use of exponential splines that can capture sharp change of gradient without causing oscillations (Chlond 1994; Böttcher 1996). The term ''strong'' is chosen to emphasize that this kind of dissipation is allowed to be as strong as necessary, even at the expense of a local loss of analyticity required to estimate a formal order of accuracy.

Historically, virtually all advection algorithms that have strong dissipation also have weak dissipation in their truncation error. [An exception is the three-time-level version of the flux-corrected transport (FCT) algorithm (Zalesak 1979).] This is because the majority of these algorithms have forward-in-time, forward two-time-level schemes, which require dissipation for computational stability [e.g., LeVeque (1992)]. Leonard (1979) formulated the following minimal requirements for advective schemes, whether or not flux limiting is employed: the leading-order advective truncation error should (i) have a derivative of higher order than the highest physical derivative in the equation to be approximated and (ii) be (weak) dissipation dominant.

Below we demonstrate that it is important to maintain high-order accuracy to minimize the phase errors in an advective scheme. Once an adaptive damping mechanism is used to prevent buildup of dispersive errors, the requirement for the dissipative dominance is no longer necessary; however, we keep the requirement that the advective scheme be numerically stable when this mechanism is turned off.

b. Principle of monotonicity

The physically motivated principle of monotonicity for an incompressible flow is that the concentration of an advected scalar on a Lagrangian particle may be only (i) conserved and (ii) diffused, satisfying the conservation principle for advection, the maximum principle for the diffusion equation, and the superposition principle for both. Thus, the dissipative processes cannot amplify existing extrema nor steepen gradients beyond limits to which they are steepened by the flow without diffusion.

A difficulty with this principle is that it cannot be trans-

lated in a simple way to the discrete level. We repeat the basic statements of the theory of monotone methods (LeVeque 1992; Harten 1983), in order to understand the consequences for the design of a numerical algorithm.

An equivalent continuous definition for the principle of monotonicity states that for any two initial conditions ${}^{(1)}q(\mathbf{x}, t_0)$ and ${}^{(2)}q(\mathbf{x}, t_0)$, such that

$$ {}^{(1)}q(\mathbf{x}, t_0) \geq {}^{(2)}q(\mathbf{x}, t_0), $$

for all $\mathbf{x}$, this inequality remains valid at any later time,

$$ {}^{(1)}q(\mathbf{x}, t) \geq {}^{(2)}q(\mathbf{x}, t), \qquad t \geq t_0. \tag{1} $$

As a consequence, if an additional positive amount of the advected quantity is injected somewhere into the flow, this injection cannot cause the altered concentration to be smaller than the original one anywhere later in time.

A discrete two-time-level algorithm $q_i^{n+1} = \mathcal{H}(q^n; i)$ is called *monotone* if it guarantees that for any pair of two discrete fields ${}^{(1)}q_i^n$ and ${}^{(2)}q_i^n$ such that,[1]

$$ {}^{(1)}q_i^n \geq {}^{(2)}q_i^n \qquad \forall i, $$

the same property holds at the new time step,

$$ {}^{(1)}q_i^{n+1} \geq {}^{(2)}q_i^{n+1} \qquad \forall i. \tag{2} $$

If an algorithm is monotone, it has the TVD, positive-definiteness, monotonicity [e.g., LeVeque (1992)] and constancy-preserving (Leonard et al. 1996) properties. On the other hand, none of these properties is sufficient to guarantee that the algorithm is monotone. To prove that an algorithm is monotone, it is sufficient to check that it is *convex*, that is,

$$ \frac{\partial}{\partial q_i^n} \mathcal{H}(q_j^n; j) \geq 0 \qquad \forall i, j, q^n. \tag{3} $$

This limitation is severe, because any monotone (convex) algorithm is at most first-order accurate in space and time (Godunov 1959).

Any linear monotone algorithm,

$$ \mathcal{H}_i(q_j; j) = \sum_j c_{ij} q_j, \tag{4} $$

with matrix coefficients c_{ij} that do not depend on q_j, is first-order accurate globally in space. A natural means of confining the inaccuracy to be only local is to allow the c_{ij} to depend on the solution q_j. This dependency means that even if the continuous problem of advection of a passive scalar is linear, the discrete algorithm is inherently nonlinear and violates the principle of superposition. All locally adaptive algorithms inherently have this feature.

Any locally adaptive procedure is essentially comprised of three components: a high-order accurate *basic* advection scheme; a *constraining* algorithm, expressing the desired property at a discrete level; and an *enforcement* mechanism, which allows the final solution to be as accurate as possible without violation of the constraint, unavoidably resulting in dissipation.

Typically the constraining algorithm requires that the new time step values at grid point $\mathbf{i}$ lie within *permissible bounds*,

$$ q_{\mathbf{i}}^{(\min)} \leq q_{\mathbf{i}}^{n+1} \leq q_{\mathbf{i}}^{(\max)}, \tag{5} $$

where $q_{\mathbf{i}}^{(\min)}$ and $q_{\mathbf{i}}^{(\max)}$ are usually determined from the values of of field q at the previous time step at location $\mathbf{i}$ and its upstream neighbors. Algorithms with this type of constraint have been devised in one, two, and three dimensions.

Though the requirement for the value of the advected quantity at the grid point $\mathbf{i}$ to lie within the triange/tetrahedron built on upstream points is a natural consequence from the principle of monotonicity, no constraining algorithm is likely to fully assure this principle. This is because the discrete algorithms involve unreliable assumptions about the behavior of the field structure between grid points. In a flux form of discrete conservation law the gridpoint values are naturally interpreted as the *average* of the advected quantity within the control volume—the grid cell. On the other hand, the constraining algorithm treats them as just sampling values, which implies that for the purpose of calculating the permissible bounds the field is treated as it is constant within the cell.

In one dimension this limiter guarantees TVD property of the method; however, the constraint Eq. (5) is somewhat more restrictive than standard TVD limiters in the sense that it not only prohibits amplification of the existent and creation of new extrema in field q, but also restricts changes in it from one time step to the next at grid points which are not extrema, thus restricting possible growth of gradients per time step. The constraint Eq. (5) does not prevent, however, artificial steepening of the slopes induced by the basic scheme over many time steps, because it has an interval of insensitivity, in which the behavior of the basic scheme remains uncontrolled. Since the permissible bounds are recomputed at every time step, it make take several time steps before an overshot is detected. This may result in the evolution of smooth profiles toward a series of steps (''staircasing''). {Alternatively, one can designate slope limiters [e.g., van Leer (1979)], but the generalization to multiple dimensions is not straightforward.}

A suggestion about how to prevent staircasing is to make the basic scheme be at least linearly stable. Although this principle is not entirely new (van Leer 1979; Zalesak 1979; Leonard and Niknafs 1991), it is far from being universally accepted. If their flux-limiting functions are set to unity, the family of classical second-order TVD flux-limited schemes described in LeVeque (1992) become equivalent to the forward-in-time, second-order centered-in-space scheme, which is linearly unstable. In contrast, if the antidiffusive fluxes are applied at full weight in Zalesak's FCT algorithm, it becomes equivalent to its basic scheme, which is centered in space and leapfrog or predictor–corrector in time, which is stable (Zalesak 1979).

[1] Although both Harten and LeVeque formulate only one-dimensional version of the definition of monotone algorithm, generalization to the multidimensional case is self-evident.

However, the ''general FCT procedure,'' as interpreted by Smolarkiewicz, has a two-time-level, centered-in-space and forward-in-time basic scheme, hence it is linearly unstable (Smolarkiewicz and Grabovski 1990). (Since these kinds of FCT algorithms are becoming widely used in geophysical modeling, we further analyze their behavior in appendix A.)

A different kind of deficiency is amplitude ''clipping'' of narrow extrema, and it can also occur with constraining limiters that appear to be too restrictive in the vicinity of the extrema. Although this phenomenon is well studied (Leonard and Niknafs 1991), a reliable algorithm to discriminate physical extrema from dispersive overshoots in multiple dimensions has not yet been designed; this is further discussed in appendix A. If such a discriminator is used, the resultant solution no longer has the discrete TVD property.

In summary, the excessive dissipation and low-order accuracy of a monotone discrete algorithm and the lack of any assurance of consistency with the monotonicity principle by higher-order, locally adaptive algorithms based on particular constraining algorithms limit the direct applicability of this principle; however, this does not preclude it as a guideline in designing numerical schemes.

c. Approach

The goal of the algorithms studied in the present paper is to achieve an optimal balance of minimal dissipation and physically accurate behavior of the solution on the grid scale without spurious oscillations. The advection and dissipation stages are separated. This allows the use of a high-order accurate, centered-in-space, nearly quadratic conservative basic advection scheme. Its cost is offset by the fact that it is used only once, and that it is not accompanied by any alternative low-order, upstream scheme. Moreover, in comparison with high-order upstream techniques, the centered scheme is simpler and less expensive to compute because no upstream switching is involved. After the advection step is performed, a correction stage to suppress possible dispersive oscillations is applied. This procedure is purely dissipative; it uses the principle of monotonicity as the basis for its design and, unlike most dissipative operators built into the truncation error of an upstream advection scheme, it may be formulated in a physically meaningful way.

However, these algorithms are not stictly monotone and do not have either TVD or monotonicity preservation properies, for the two following reasons:

1) The dissipation stage is a nonlinear filtering procedure, which involves explicit computation and iterative diffusion of the dispersive overshoots; hence the desirable property (the values of the advected quantity at the new time step lie within permissible bounds) is achieved only asymptotically.

2) We provide an option in which the constraint imposed by Eq. (5) is relaxed or even turned off in a controllable manner in the vicinity of a physically acceptable extremum; hence, the solution does not satisfy TVD property even if a strict enforcement mechanism is used.

Unlike previous adaptive nonlinear filters (e.g., Forester 1977; Engquist et al. 1989; Shyy et al. 1991), the proposed techniques are fully multidimensional for both advective and corrective phases, and we argue that this approach is more suitable for incompressible high–Reynolds number flows, where the contributions due to fluxes along different directions are strongly coupled and therefore cannot be limited separately.

d. Organization

This paper is organized as follows. In section 2 we introduce explicit locally adaptive dissipation and discuss measures to minimize dispersive errors, for uniform, one-dimensional advection. In section 3 we generalize to two dimensions and perform traditional numerical tests for rotational flow. In section 4 we describe numerical experiments with two-dimensional turbulence to compare the performance of the new algorithm with previously proposed schemes, such as UTOPIA and various versions of the FCT algorithm. In section 5 we discuss how adaptive and nonadaptive numerical schemes simulate subgrid-scale dissipation when turbulent cascades create strong gradients of the advected field on the grid scale. In section 6 we discuss numerical biases arising during long time integrations of turbulence. In section 7 we report the computational costs for different schemes. In section 8 we discuss the trade-offs associated with different degrees of strictness in enforcement of monotonicity, higher accuracy, and smaller dissipation.

2. One-dimensional advection

Consider first the one-dimensional hyperbolic problem:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}(cq) = 0, \tag{6}$$

where q is the advected quantity and c is the advection speed. For one-dimensional incompressible flows, c must be a spatial constant, and we assume $c > 0$.

a. Basic schemes

The general, symmetric scheme to approximate $\partial q/\partial x$ at the rth order of accuracy may be represented as

$$\left.\frac{\partial q}{\partial x}\right|_{x_i} \approx \frac{1}{\Delta x}\sum_{m=1}^{r/2} a_m(q_{i+m} - q_{i-m}), \tag{7}$$

where the coefficients $\{a_m | m = 1, r/2\}$ are defined for each r as

$$
\begin{aligned}
r = 2 \qquad & a_1 = \frac{1}{2} \\
r = 4 \qquad & \{a_1, a_2\} = \left\{\frac{2}{3}, -\frac{1}{12}\right\} \\
r = 6 \qquad & \{a_m\} = \left\{\frac{3}{4}, -\frac{3}{20}, \frac{1}{60}\right\} \\
r = 8 \qquad & \{a_m\} = \left\{\frac{4}{5}, -\frac{1}{5}, \frac{4}{105}, -\frac{1}{280}\right\} \\
r = 10 \qquad & \{a_m\} = \left\{\frac{5}{6}, -\frac{5}{21}, \frac{5}{84}, -\frac{5}{504}, \frac{1}{1260}\right\}. \quad (8)
\end{aligned}
$$

We use a leapfrog (LF) time step to advance in time the discrete approximation for Eq. (6). The computational mode of LF time step is suppressed by applying trapezoidal rule (TR) correction every 20 time steps.

As test problems we chose the three different initial conditions shown on the top left of Fig. 1. They are a top-hat profile on the left,

$$
q(x)|_{t=0} = \begin{cases} 0, & x < 3/32 \\ 1, & 3/32 \le x \le 9/32 \\ 0, & x > 9/32, \end{cases} \qquad (9)
$$

and two relatively smooth profiles defined by

$$
q(x)|_{t=0} = \begin{cases} \cos^4\left(\pi \dfrac{x - x_0}{2\sigma}\right), & x_0 - \sigma \le x \le x_0 + \sigma \\ 0, & \text{everywhere else,} \end{cases} \qquad (10)
$$

where $(x_0, \sigma) = (81/128, 3/64)$ for the narrower peak in the middle and $(113/128, 3/32)$ for the wider peak on the right. The grid resolution is 256 for the unit periodic domain. Hence the square profile is 48 grid points wide, while the narrow and wide peaks are approximately 6 and 12 grid points wide, measured on their half-height. Only about one-third of the domain is plotted for each of the three cases.

The five panels on the left side of Fig. 1 below the initial conditions show nondissipative solutions obtained by approximations (7)–(8). The Courant number is $\mathrm{Cu} = c\Delta t/\Delta x = 1/4$, which is the half of the critical value allowed by the stability limit for LF step. The solutions are shown after 1024 time steps, corresponding to one period. In the ideal case the final distribution should coincide with the initial state. The exact solution, shown by the dotted line, is superimposed for reference.

The schemes (7)–(8) with LF time step have a purely dispersive truncation error, since they all conserve exactly both the net content and the quadratic variance of the advected quantity. Except for the wide peak, all of the solutions clearly exhibit spurious oscillations. For the narrow and wide peaks, the solutions converge with increasing order of accuracy. Measured by the negative overshoot values at the trailing edge of the peak, the second-order scheme is less accurate for its wide peak (i.e., with twice as fine grid resolution) than the fourth-order scheme for its narrow peak. For the top hat, both the amplitude and the wavelength of the dispersive oscillations decrease with increasing order of accuracy. There is a drastic improvement when going from second to fourth order, while further changes are moderate.

b. Explicit locally adaptive dissipation

Our proposal to suppress the spurious oscillations is explicit locally adaptive dissipation (ELAD), as follows. At every time step, just after the advective step is made as above, the newly computed field $\hat{q}^{n+1}$ is checked for overshoots.

Step 1: Determine the minimum and maximum permissible values for q_i^{n+1} at every grid point x_i based on the values of q^n at x_i and the adjacent upstream point x_{i-1},

$$
\begin{aligned}
q_i^{(\min)} &= \min(q_i^n, q_{i-1}^n) \\
q_i^{(\max)} &= \max(q_i^n, q_{i-1}^n). \qquad (11)
\end{aligned}
$$

Step 2: Compute the excess field for $\hat{q}^{n+1}$,

$$
\begin{aligned}
\epsilon_i = \max(0, \hat{q}_i^{n+1} - q_i^{(\max)}) \\
+ \min(0, \hat{q}_i^{n+1} - q_i^{(\min)}), \qquad (12)
\end{aligned}
$$

which measures the overshot magnitude. Obviously, $\epsilon_i = 0$ if the computed value of $\hat{q}_i^{n+1}$ is within the permissible range, $q_i^{(\min)} < \hat{q}_i^{n+1} < q_i^{(\max)}$.

Step 3: Diffuse the excess field by applying a discrete diffusive operator at the maximum weight allowed by the well-known stability criterion for the diffusion equation,

$$
q_i^{n+1} = \hat{q}^{n+1} + \frac{1}{2}(\epsilon_{i+1} - 2\epsilon_i + \epsilon_{i-1}). \qquad (13)
$$

Iterate step 2 and step 3 until overshoots are eliminated or reduced below the acceptable level.

The solutions obtained by this corrected scheme are shown in the right column of Fig. 1, again for successive orders of accuracy in the basic advection scheme. Eight diffusive iterations were used for the second-order scheme, and four for all others. Except for the second-order scheme, the oscillations are almost completely eliminated, but this occurs at the expense of some clipping of the narrow peak.

We emphasize that one should not just subtract the excess field ϵ_i from $\hat{q}_i^{n+1}$ in place of step 3, Eq. (13). Doing so would violate the conservation principle for the total content of q. The proposed approach provides a mechanism for redistribution of the excessive amounts of q among the grid points in order to eliminate the dispersive wiggles while preserving total content and causing some dissipation of variance.

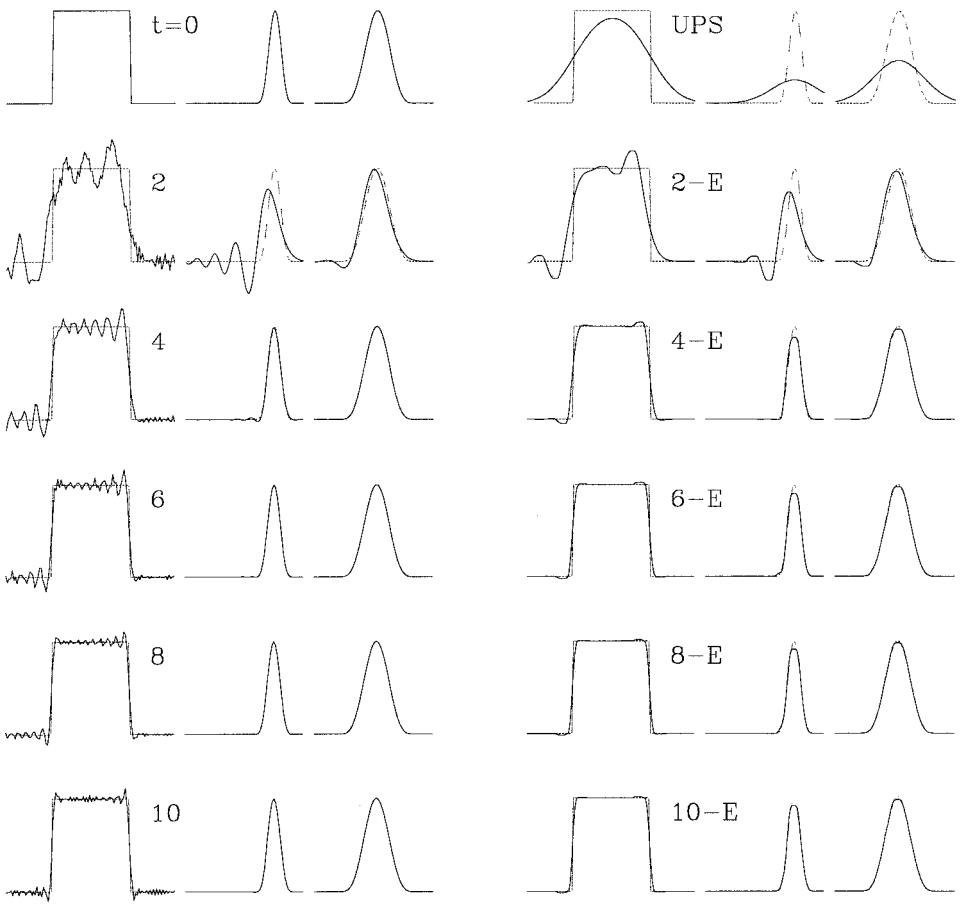

FIG. 1. One-dimensional advection of sharp and smooth passive scalar profiles. (Left) Top pattern labeled $t = 0$ shows three different initial conditions. Only approximately one-third of the whole periodic domain of 256 grid points is shown in each case. The rectangle is 48 grid points wide, while the smooth profiles are 6 and 12 grid points wide, measured at a level of one half of their peak. The functions specifying the three scalar profiles are equal to 0 outside the regions of 48, 12, and 24 grid points. For each numerical scheme described below, these three cases were run independently from each other for one period, which corresponds to 1024 time steps at Courant number of 1/4. Labels 2, 4, 6, 8, and 10 indicate order of accuracy of spatial differencing, while suffix -E inicates cases when the explicit adaptive dissipation was applied. Solutions shown on the left half are nondissipative. Note how the solution for both smooth profiles converges with the increase of order of accuracy of the scheme. (Right) Top pattern shows first-order-accurate upstream solution.

c. Sources of numerical dispersion

The results in Fig. 1 show the advantage of higher-order schemes, despite the fact that the solutions are not smooth and one cannot rely on convergence in terms of Taylor series. In fact, what is more important is that the higher-order schemes tend to generate dispersive oscillations at a shorter wavelength and, therefore, require fewer diffusive iterations to remove them and cause smaller dissipation. To explain this behavior, consider a single Fourier component,

$$q_k(x_j) = e^{ikx_j} = e^{ik\Delta xj}.$$

When the discrete operator (7) is applied to $q_k(x_j)$, the result is

$$i\mathcal{K}(k)q_k(x_j).$$

For the continuous derivative, $i\mathcal{K}(k) = ik$, but the discrete derivative yields

$$i\mathcal{K}_r(k) = \frac{2i}{\Delta x}\sum_{m=1}^{r/2} a_m \sin(mk\Delta x), \tag{14}$$

where $0 < k < \pi/\Delta x$. The functions $\mathcal{K}_r(k)$, $r = 2, 4, \ldots, 10$, are plotted in Fig. 2a. Since $\partial\mathcal{K}_r(k)/\partial k$ corresponds to the group velocity for this hyperbolic problem, the critical value of k at which $\mathcal{K}_r(k)$ reaches its maximum can be interpreted as the point of dispersive breakdown of the scheme. Beyond this point the scheme cannot propagate Fourier components at the right speed, so this part of the spectrum must be suppressed to avoid spurious oscillations. This critical k increases with the order of accuracy. However, increasing order is not the only way to move the dispersive breakdown to larger k. The dispersive break-

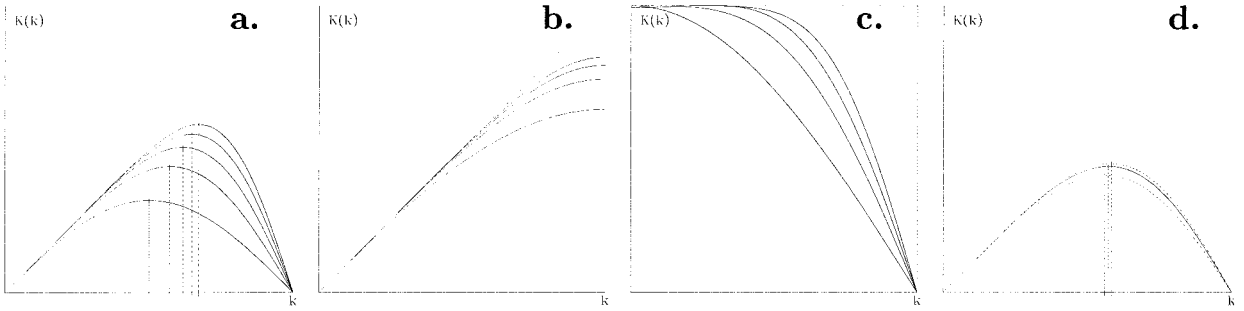

FIG. 2. Spectral characteristics of discrete operators. (a) *Nonstaggered* finite difference approximations for the first derivative from the second (the lowest curve) to the tenth- (top) orders of accuracy; vertical dashed line for each curve indicates point of dispersive breakdown. (b) The same as (a), but for *staggered* finite differences of the second (the lowest curve) to the eighth (top) orders of accuracy. (c) Interpolation to midpoint of the second (the lowest curve) to the eighth (top) orders of accuracy. (d) Comparison of spectral characteristics of fourth-order *nonstaggered* discrete difference operator (solid curve); operator obtained by combination of fourth-order interpolation to midpoints and fourth-order *staggered* difference (dashed curve above solid), and combination of fourth-order interpolation and second-order staggered difference (dashed curve below solid).

down of staggered finite differences always happens at the highest resolved wavenumber regardless of the order of accuracy, unlike for the nonstaggered finite differences above; these are defined by

$$\left.\frac{\partial q}{\partial x}\right|_{x_{i+1/2}} \approx \frac{1}{\Delta x}\sum_{m=1}^{r/2} b_m(q_{i+1+(m-1)} - q_{i-(m-1)}), \quad (15)$$

where the coefficients $\{b_m\}$ are

$$\begin{aligned} r &= 2 \qquad b_1 = 1 \\ r &= 4 \qquad \{b_1, b_2\} = \left\{\frac{9}{8}, -\frac{1}{24}\right\} \\ r &= 6 \qquad \{b_m\} = \left\{\frac{75}{64}, -\frac{25}{384}, \frac{3}{640}\right\} \\ r &= 8 \qquad \{b_m\} = \left\{\frac{1225}{1024}, -\frac{245}{3072}, \frac{49}{5120}, -\frac{5}{7168}\right\} \end{aligned} \quad (16)$$

and the spectral functions are

$$i\mathcal{K}_r(k) = \frac{2i}{\Delta x}\sum_{m=1}^{r/2} b_m \sin\left(\frac{2m-1}{2}k\Delta x\right) \quad (17)$$

(see Fig. 2b).

For advection one needs to compute the derivative at the same points as the field, and some kind of nonstaggered finite-difference approximation is required. This scheme may be constructed with a staggered derivative applied to the advected quantity after interpolation to midpoints. Because the dispersive error of staggered differencing is smaller than that due to the interpolation procedure, it makes sense to use higher-order interpolation in combination with lower-order staggered differencing. For example, let

$$\overline{q}^{\,x}_{i+1/2} = \sum_{m=1}^{r/2} d_m(q_{i+1+(m-1)} - q_{i-(m-1)}), \quad (18)$$

along with the sets of coefficients

$$\begin{aligned} r &= 2 \qquad d_1 = \frac{1}{2} \\ r &= 4 \qquad \{d_1, d_2\} = \left\{\frac{9}{16}, -\frac{1}{16}\right\} \\ r &= 6 \qquad \{d_m\} = \left\{\frac{75}{128}, -\frac{25}{256}, \frac{3}{256}\right\} \\ r &= 8 \qquad \{d_m\} = \left\{\frac{1225}{2048}, -\frac{245}{2048}, \frac{49}{2048}, -\frac{5}{2048}\right\}, \end{aligned} \quad (19)$$

be the interpolation scheme of order r. Then the discrete approximation to Eq. (6) may be written as

$$\left.\frac{\partial q}{\partial t}\right|_{x_i} = -c\left(\frac{\overline{q}^{\,x}_{i+1/2} - \overline{q}^{\,x}_{i-1/2}}{\Delta x}\right). \quad (20)$$

As an alternative to (19), one can use slightly different sets of coefficients to compute $\overline{q}_{i+1/2}$,

$$\begin{aligned} r &= 4 \qquad \{d_1', d_2'\} = \left\{\frac{7}{12}, -\frac{1}{12}\right\} \\ r &= 6 \qquad \{d_m'\} = \left\{\frac{37}{60}, -\frac{2}{15}, \frac{1}{60}\right\} \\ r &= 8 \qquad \{d_m'\} = \left\{\frac{533}{840}, -\frac{139}{840}, \frac{29}{840}, -\frac{1}{280}\right\}, \end{aligned} \quad (21)$$

so that (20) yields a scheme of order of overall accuracy r that is equivalent to Eqs. (7)–(8) when c is constant.

Although the combination of (20), (18), and (19) or

(21) is formally only second-order accurate in the general case of nonconstant c, these schemes perform almost as well as the uniformly accurate schemes of order r, obtained by subsequent use of rth-order interpolation (19) and rth-order staggered differencing (15), see Figs. 2c,d. This is because the advecting velocity field is typically smoother than the advected scalar. This property is useful for designing schemes of minimal complexity for general domain geometry, where expansion of the computational stencil is not desirable.

Discretization of the time derivative causes additional error. A time-stepping algorithm for a Fourier component q_k may be written as

$$\hat{q}_k^{n+1} = \sigma(k)\hat{q}_k^n, \tag{22}$$

where the amplification factor $\sigma(k)$ can be represented as

$$\sigma(k) = e^{-i\omega_k^{\text{num}}\Delta t}e^{-\delta\Delta t}, \tag{23}$$

where $\omega_k^{\text{num}}\,\Delta t$ and $\delta\Delta t$ are the phase increment and the amplitude decay per time step. It is natural to form two nondimensional parameters,

$$\frac{c_k^{\text{num}}}{c} = -\frac{\omega_k^{\text{num}}\Delta t}{k\Delta x(c\Delta t/\Delta x)}, \tag{24}$$

which is the ratio of numerical to analytical phase speeds, and

$$\frac{\delta\Delta t}{(c\Delta t/\Delta x)} = \frac{\delta\Delta x}{c}, \tag{25}$$

which is the decay rate per one-gridpoint travel, based on analytical phase speed.

In the case of LF time step there is no amplitude decay and,

$$\frac{c_k^{\text{num}}}{c} = \frac{\arcsin[(c\Delta t/\Delta x)\mathcal{K}(k)\Delta x]}{(c\Delta t/\Delta x)k\Delta x}. \tag{26}$$

If the argument of arcsin exceeds 1 in absolute value, ω_k^{num} becomes complex, which indicates numerical instability. This function is plotted in Fig. 3, for Courant numbers Cu $= c\Delta t/\Delta x = 0$ (a) and for Cu $= 0.375$ (d) and 0.75 (h). In case d there is a significant phase lead for the intermediate wavenumbers, while case h is unstable for all orders of accuracy exceeding 2.

The Crank–Nicholson (CN, implicit) time step is also nondissipative, and provides a phase speed ratio similar to (26), except that arcsin is replaced by arctan. This is shown in Figs. 3e and 3i. Unlike LF, it causes phase delay additional to the delay due to the spatial discretization error. Though CN is unconditionally stable, its use for Cu > 1 brings additional phase error, which is equivalent to the reduction of the effective spatial resolution by a factor of 1/Cu.

Two other often used time-stepping algorithms are the LF–trapezoidal rule and LF–Adams–Multon (both predictor–corrector) time steps. Unlike LF and CN, they are not reversible in time, therefore some numerical dissipation is the result (Figs. 3f,g,j,k). The dissipation rates, however, are vanishingly small in comparison with the dissipation introduced by upstream-biased semi-Lagrangian interpolation schemes (Figs. 3b,c).

The last two cases are interesting, because one can show that for the case of semi-Lagrangian schemes the phase errors of time and space differencing partially comensate each other (if Cu $= 1/2$ the phase errors cancel exactly). This advantage, however, is lost due to the large dissipation. Comparison between Figs. 3b and 3g (as well as Figs. 3c and 3k) suggests that the high-order *centered*-in-space schemes provide a wider range of Fourier components that propagate with accurate phase speed, and, at the same time, are not suppressed by the numerical dissipation. We therefore use high-order centered-in-space advection schemes as the basic scheme for adaptive advection algorithms like FCT and ELAD.

Comparison between different time-stepping algorithms shows that even near the CFL limits of stability, the phase error is dominated by error due to spatial discretization. The LF–TR and AM predictor–corrector methods are more accurate, but they also bring additional computational cost. If the FCT or ELAD procedure is used, the additional cost is offset by the larger time step allowed due to extended range of accuracy and CFL stability because the flux limiting is done only once per time step, not during both predictor and corrector stages. Thus, the AM time step is the choice for FCT algorithm; see appendix A.

The upstream-biased schemes retain numerical stability as long as the departure poins lie within the stencil. Thus, the high-order versions may be used with Cu $>$ 1. In this regime, however, the phase behavior is unattractive because of the phase ''folding'' associated with aliasing. It occurs first when Cu exceeds 1/2. The Fourier components with $N/(4\text{ Cu}) < k < N/2$ propagate in the opposite direction. This effect is not noticeable in practice up to Cu $= 1$, because this portion of the spectrum is suppressed anyway, however, further increase of the Courant number is not desirable because it is equivalent to the decrease of effective resolution inversely proportional to the Courant number.

3. Tests in uniformly rotating flow

Rotating cone and grooved-cylinder tests have become standard for evaluating the performance of advection schemes (Crowley 1968; Orszag and Jayne 1974; Zalesak 1979; Smolarkiewicz 1982, 1983; Tremback et al. 1987; Carpenter et al. 1990; Chlond 1994). Consider the two-dimensional passive-scalar advection problem,

$$\frac{\partial q}{\partial t} + \boldsymbol{\nabla}(\mathbf{u}q) = 0, \tag{27}$$

where the velocity field $\mathbf{u} = (u, v)$ corresponds to solid-body rotation,

$$u = -\omega y, \qquad v = +\omega x. \tag{28}$$

For $\omega = 2\pi$, the period of rotation is equal to 1, and

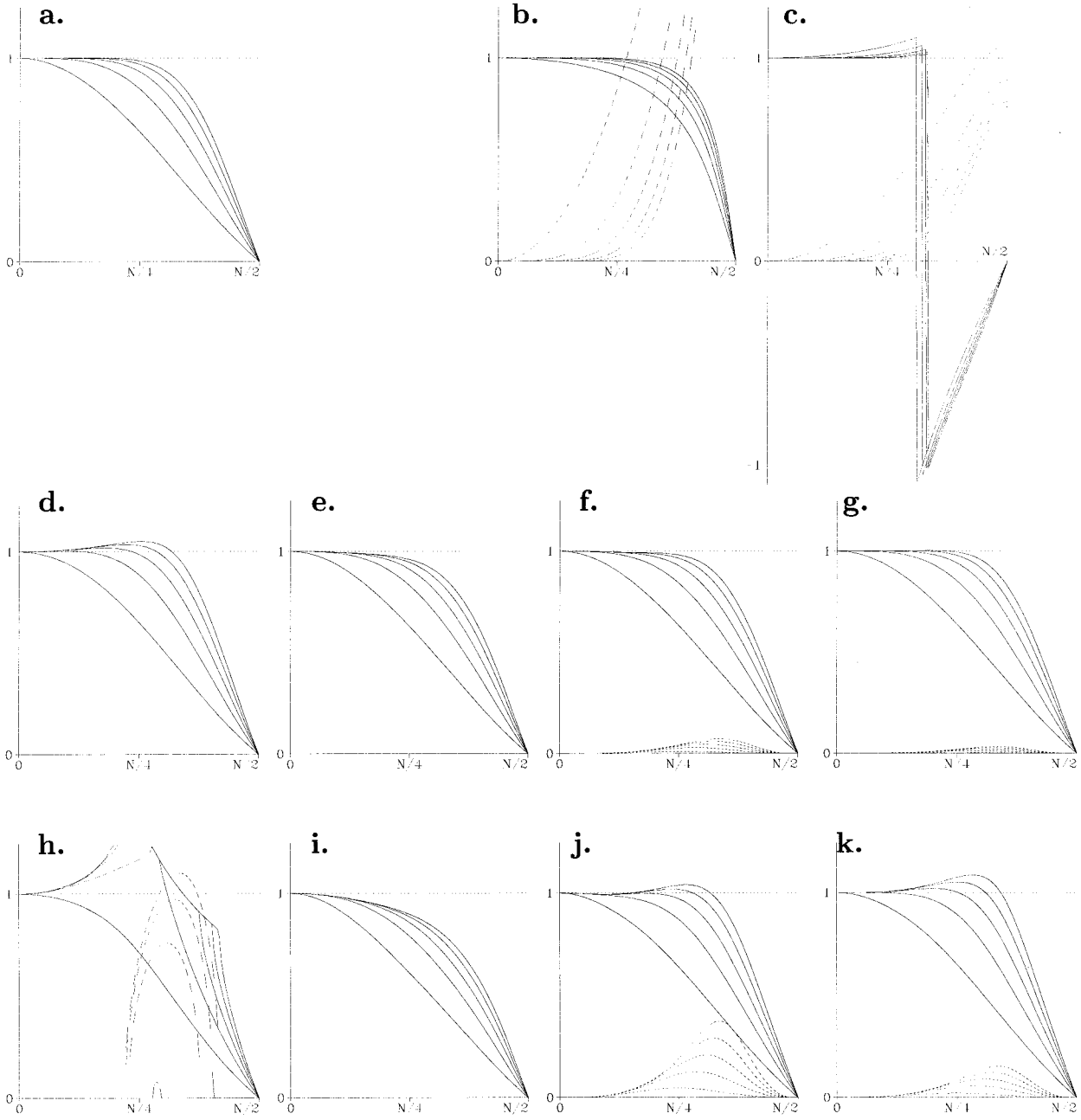

FIG. 3. Normalized phase speed (solid curves) and amplitide decay per gridpoint travel (if any, dashed curves) as functions of wavenumber. (a) Centered in space for the second (the lowest curve) to tenth (the top) orders of accuracy and infinitely small Courant number; (b) semi-Lagrangian upstream interpolations from first (the lowest phase speed, the strongest decay) to ninth orders of accuracy for Cu = 0.375; (c) the same as (b), but for Cu = 0.75; (d) to (g) same as (a), but for Cu = 0.375 and different time steps: (d) LF; (e) CN (implicit); (f) LF–trapezoidal (predictor–corrector); (g) Adams–Multon; (h) to (k) the same as (d) to (g), respectively, but for Cu = 0.75. (Case h is linearly unstable for all orders of accuracy greater than second.)

after one period the exact solution coincides with the initial distribution. The passive scalar q is defined within the domain $-1/2 < x, y < 1/2$ with cyclic boundary conditions in both directions. The initial distribution $q|_{t=0}$ (Fig. 4a or Fig. 5a) is equal to 0 everywhere except in a circle of radius $r = 1/8$ centered at $x_0 = -1/4$, $y_0 = 0$, which is offset from the center of rotation. Inside the circle it either linearly decays from its central maximum value to 0 at the edge (Fig. 4a) or has a piecewise-constant shape (Fig. 5a).

The discrete, conservative, nondissipative advection step is

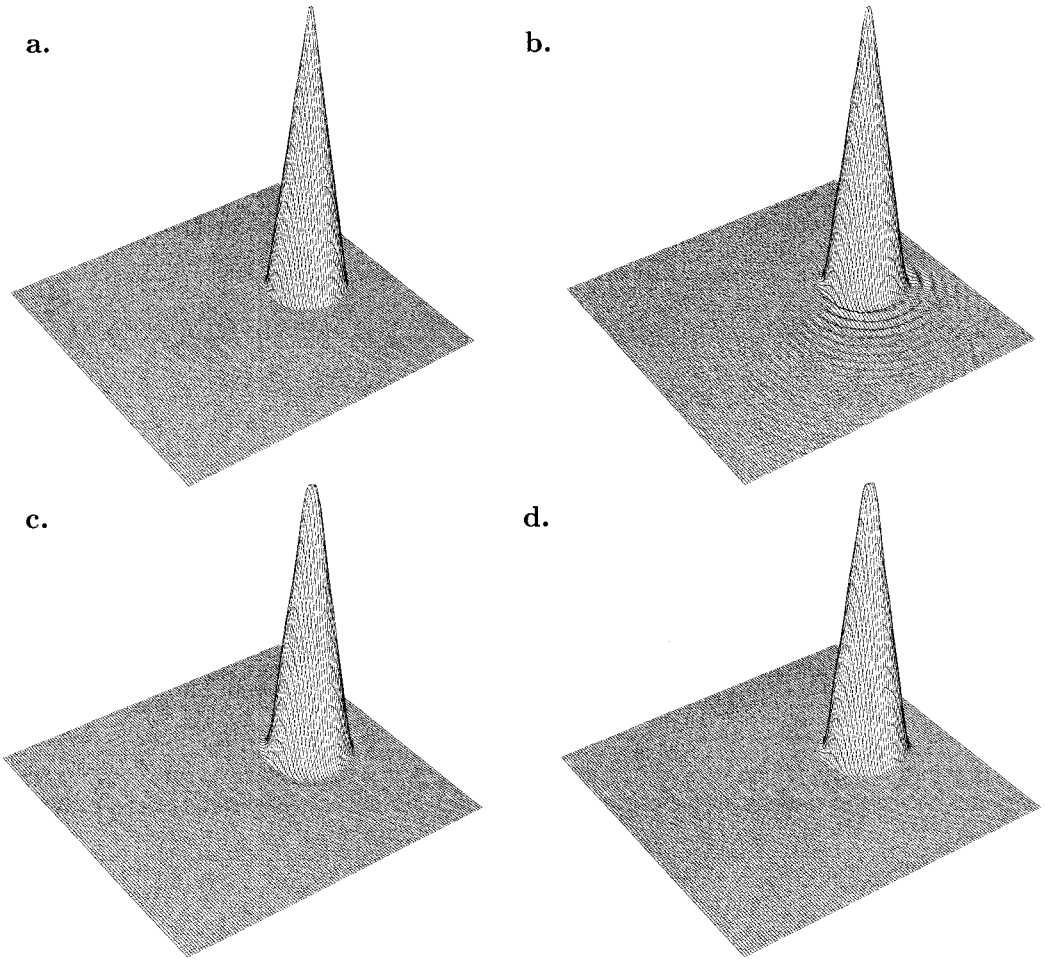

FIG. 4. Rotating cone problem. Grid resolution 128 × 128. (a) Initial state/exact solution after one period. (b) Solution after one period obtained by nondissipative sixth-order advection scheme [Eq. (25)]; (c) solution after one period obtained by fourth-order advection scheme corrected by ELAD. (d) The same as (c), but the sixth-order advection scheme is used.

$$\hat{q}^{n+1} = q^{n-1} - 2\Delta t[\delta_x(u\overline{q}^x) + \delta_y(v\overline{q}^y)], \quad (29)$$

where δ_x and δ_y are staggered, rth-order finite differences $r = 2, \ldots, 8$, Eq. (15), and $\overline{q}^x$, $\overline{q}^y$ are interpolations to midpoint locations in the x and y directions, Eq. (18). As in the one-dimensional case, overshooting will result from (29). To correct this, we apply the following ELAD procedure to $\hat{q}^{n+1}$ after the advection step.

Step 1: Determine the values at two upstream points,

$$q^{(\mathrm{sx})}_{i+1/2,j} = \begin{cases} q^n_{i,j}, & \text{if } u_{i+1/2,j} > 0 \\ q^n_{i+1,j}, & \text{if } u_{i+1/2,j} < 0 \end{cases}$$

$$q^{(\mathrm{sy})}_{i,j+1/2} = \begin{cases} q^n_{i,j}, & \text{if } v_{i,j+1/2} > 0 \\ q^n_{i,j+1}, & \text{if } v_{i,j+1/2} < 0, \end{cases} \quad (30)$$

then define permissible bounds for $q^{n+1}_{i,j}$,

$$q^{(\min)}_{i,j} = \min(q^{(\mathrm{sx})}_{i-1/2,j}, q^{(\mathrm{sx})}_{i+1/2,j}, q^{(\mathrm{sy})}_{i,j-1/2}, q^{(\mathrm{sy})}_{i,j+1/2})$$
$$q^{(\max)}_{i,j} = \max(q^{(\mathrm{sx})}_{i-1/2,j}, q^{(\mathrm{sx})}_{i+1/2,j}, q^{(\mathrm{sy})}_{i,j-1/2}, q^{(\mathrm{sy})}_{i,j+1/2}). \quad (31)$$

Step 2: Compute the excess field,

$$\epsilon_{i,j} = \min(0, \hat{q}^{n+1}_{i,j} - q^{\min}_{i,j}) + \max(0, \hat{q}^{n+1}_{i,j} - q^{\max}_{i,j}). \quad (32)$$

Step 3: Diffuse the excess field using a nine-point discrete Laplacian operator, applied with the maximum weight allowed by the diffusive stability criterion (which is somewhat less restrictive for a nine-point Laplacian than for the usual five-point one):

$$q^{n+1}_{i,j} = \hat{q}^{n+1}_{i,j} - \epsilon_{i,j} + \frac{1}{6}(\epsilon_{i-1,j} + \epsilon_{i+1,j} + \epsilon_{i,j-1} + \epsilon_{i,j+1}) + \frac{1}{12}(\epsilon_{i-1,j-1} + \epsilon_{i+1,j-1} + \epsilon_{i-1,j+1} + \epsilon_{i+1,j+1}). \quad (33)$$

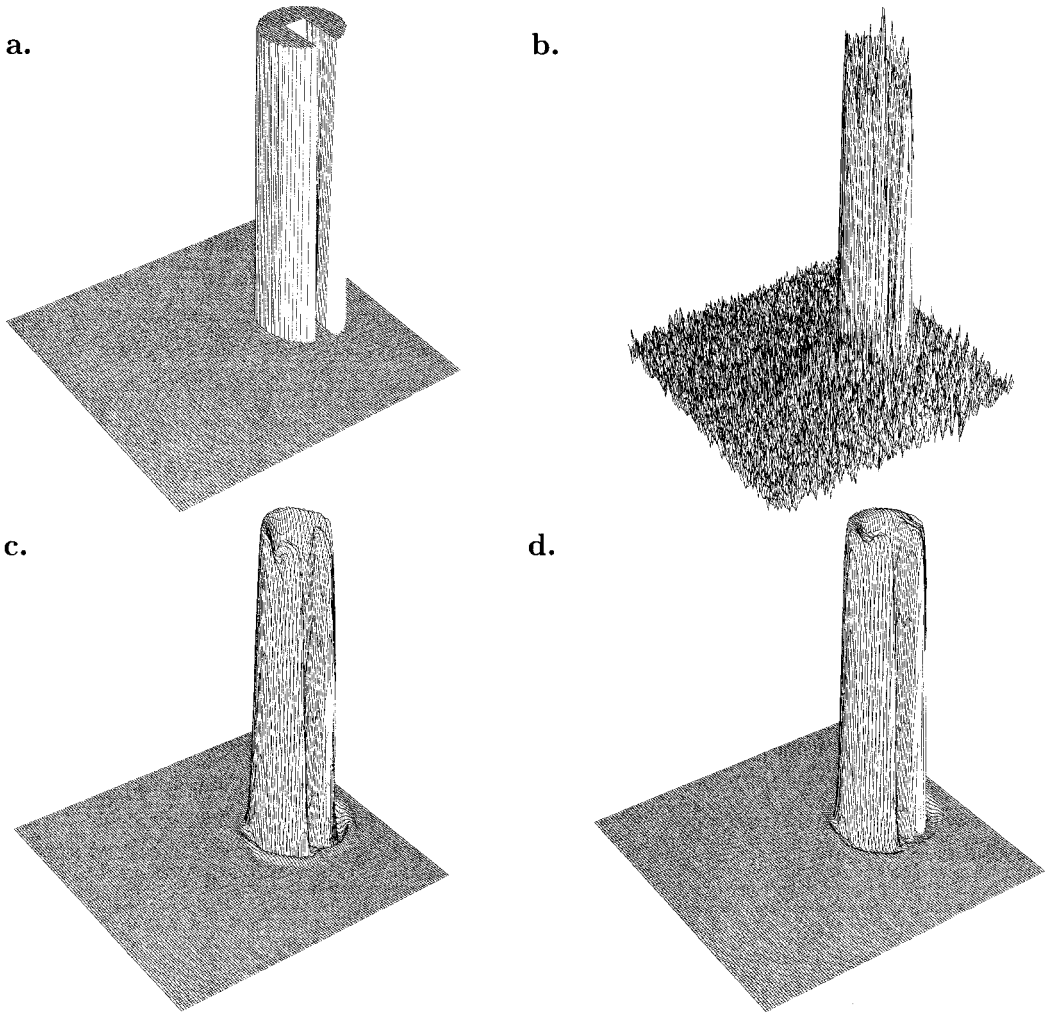

FIG. 5. Grooved cylinder rotation test. Grid resolution 128 × 128. (a) Initial state/exact solution after one period. (b) Solution after one period obtained by *nondissipative* sixth-order advection scheme; (c) solution after one period obtained by fourth-order advection scheme with explicit adaptive dissipation. (d) The same as (c), but the sixth-order advection scheme is used.

Iteratively repeat step 2 and step 3 several times.

Numerical solutions obtained by different schemes are shown in Figs. 4–6. Sixth-order, nondissipative solutions are shown in Fig. 4b and Fig. 5b: this scheme generates mild oscillations around the rotating cone and stronger ones around the grooved cylinder. In all other solutions, the ELAD procedure was applied, with four diffusive iterations at every time step. The results are largely with oscillations but do exhibit some degree of smoothing of the initial profiles. For the rotating cone, when the initial function is continuous, the fourth- and sixth-order solutions are barely distinguishable (Figs. 4c,d). However, for the grooved cylinder an improvement in accuracy is still evident going from fourth to sixth order, (Figs. 5c,d), but further progress is slight when going to eighth and tenth orders (Figs. 6a,b).

For comparison in Fig. 7 we also show a grooved-cylinder solution obtained by a sixth-order FCT algorithm with strictly monotone behavior (Zalesak 1979). Notice that its shape is similar to the ELAD result in Fig. 6d, except that the edges are somewhat smoother, which is indicative of its greater dissipation. This comparison is further discussed below.

Note that the ELAD procedure yields monotone solutions only if the the diffusive iterations converge. In practice we have chosen the number of iterations such that the computational cost of the whole corrective procedure is approximately equal to that of the basic advective scheme at fourth order. Thus, some remaining nonmonotonicity is seen in Figs. 5c,d. Alternatively, one might use a lower-order advection scheme in combination with a higher number of iterations and achieve acceptably nonmonotone behavior; however, our experience shows that this is both

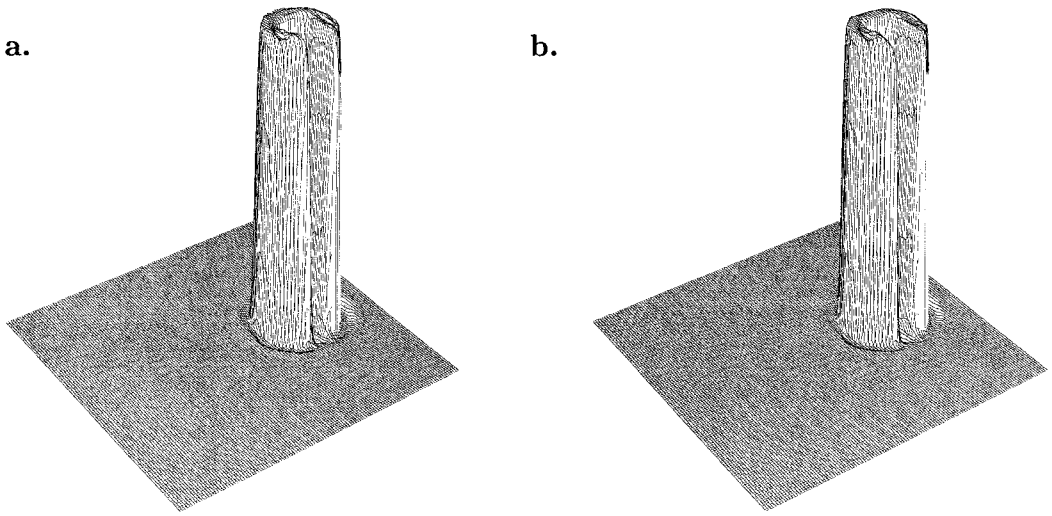

FIG. 6. Similar to Figs. 5c and 5d, except that (a) eighth-order advection scheme is used. (b) Tenth-order advection scheme is used.

more dissipative and computationally more expensive than a ''cost balanced'' combination, with at least fourth-order advection and fewer diffusive iterations.

4. Monotonicity of extrema and dissipation rate in two-dimensional turbulence

The numerical tests presented above are not sufficient to evaluate the geophysical utility of an advection scheme, because the velocity fields are too simple and do not include the physically necessary dissipation due to turbulent cascades. Also, their velocities are one-dimensionally nondivergent,

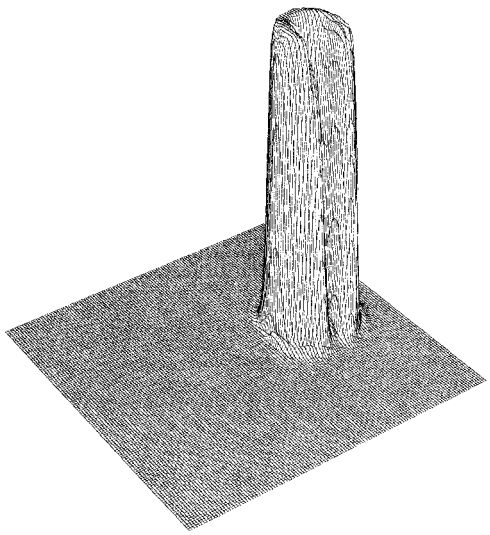

FIG. 7. Solution after one rotation obtained by sixth-order version of the FCT algorithm.

$$\frac{\partial u}{\partial x} = 0, \qquad \frac{\partial v}{\partial y} = 0, \tag{34}$$

which favors algorithms for adaptive control of nonmonotone behavior by split-directional schemes (e.g., Carpenter et al. 1990; Chlond 1994; Böttcher 1996).

In the case of deformational flow, relations (34) are no longer valid, so that if a split-directional method is used, one faces a dilemma of either chosing conservative discretization for the advection equation and treating the flow as divergent during the one-directional substeps, or using advective discretization and abandoning the exact global conservation of the content of advected quantity. Satisfying the property of exact global conservation favors the first choice, while shape- and constancy-preservation favors the second (Leonard et al. 1996). Typically split-directional methods cannot have both properties simultaneously, unless special measures are introduced (e.g., Easter 1993). These measures involve the introduction of additional terms into the conservative discrete scheme, or solving an additional auxiliary equation to cancel the effects of one-directional divergence. As the result, the computational cost of the algorithm nearly doubles and becomes close to that of a fully multidimensional approach, while the operator-splitting error still remains. This consideration suggests the advantage of the use of a fully multidimensional method for the simulation of nondivergent or nearly nondivergent flows.

A deformational flow test has also been proposed for steady, two-dimensional flow (Smolarkiewicz 1982), where a nondissipative, analytical solution is available for comparison (Staniforth et al. 1987); however, this test is also quite special with respect to turbulent cascades.

Hence, we choose freely developing, two-dimension-

al turbulence as our next test. The physics of this problem includes strong dissipation of both enstrophy (vorticity variance) and passive scalar variance, cascades of energy to large scales and of enstrophy and passive-scalar variance to small scales, and spontaneous formation of coherent vortices that shelter the advected scalars in their cores. The latter occurs because of the alignment of scalar isolines with streamlines, which suppresses the advective tendency (e.g., McWilliams 1984; Babiano et al. 1987).

For this problem, we initialize vorticity ζ and passive scalar concentration q as two uncorrelated random fields, similar to McWilliams (1990). For each field we specify a narrowband spectrum with a given shape and random phase,

$$\zeta\hat{\mathbf{k}} = C\left(\frac{k^{\alpha}}{k_0^{2\alpha} + k^{2\alpha}}\right)^{1/2} e^{2\pi i \,\mathrm{rand}(\mathbf{k})}, \tag{35}$$

and Fourier transform it back to physical space. In Eq. (35) $\mathbf{k} = (k_x, k_y)$ is the two-dimensional wavenumber, $k = (k_x^2 + k_y^2)^{1/2}$; rand( ) is random value, uniformly distributed within the range of [0, 1]; and k_0 and α are parameters. For all computations presented here $k_0 = 20$ and $\alpha = 10$. The normalization coefficient C is chosen to make the total energy of the system to be unity.

Starting from these initial condition, the fields evolve according to governing equations,

$$\frac{\partial \zeta}{\partial t} = -\mathcal{J}(\psi, \zeta) + \mathcal{D}_{\zeta} \tag{36}$$

$$\zeta = \nabla^2\psi \tag{37}$$

$$\frac{\partial q}{\partial t} = -\mathcal{J}(\psi, q) + \mathcal{D}_{q}, \tag{38}$$

where ψ is the streamfunction related to the velocity components u, v by

$$u = -\frac{\partial \psi}{\partial y}, \qquad v = +\frac{\partial \psi}{\partial x}, \tag{39}$$

the Jacobian operator is

$$\mathcal{J}(\psi, \zeta) = \frac{\partial \psi}{\partial x}\frac{\partial \zeta}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial \zeta}{\partial x}, \tag{40}$$

and ∇^2 denotes the Laplacian operator. The dissipative terms $\mathcal{D}_{\zeta}$ and $\mathcal{D}_{q}$ are defined only at the discrete level here. Doubly periodic boundary conditions are assumed.

In discrete approximation for Eq. (40), the velocity components u and v, computed from ψ by Eq. (39), are staggered relatively to the scalar grid points,

$$u = -\delta_y\overline{\psi}^{xy}, \qquad v = +\delta_x\overline{\psi}^{xy},$$
$$u' = -\delta_{y'}\psi, \qquad v' = +\delta_{x'}\psi. \tag{41}$$

Here u' and v' are auxiliary velocity components in diagonal directions x' and y', respectively; δ_x, δ_y, $\delta_{x'}$, and $\delta_{y'}$ are the staggered second-, fourth-, or sixth-order finite differences [Eq. (15)]; an overbar with a single coordinate superscript denotes the second-, fourth-, or sixth-order interpolation to the midpoint in the indicated direction [Eq. (18)]; and $\overline{\psi}^{y} = (\overline{\psi}^{x})^{y} = (\overline{\psi}^{y})^{x}$ is a two-dimensional interpolation of ψ to the vertices of its grid cells. Then the Jacobian operator is discretized similarly to Eq. (29),

$$\mathcal{J}(\psi, \zeta) = \frac{1}{2}\{\delta_x(u\overline{\zeta}^{x}) + \delta_y(v\overline{\zeta}^{y}) + \delta_{x'}(u'\overline{\zeta}^{xy}) + \delta_{y'}(v'\overline{\zeta}^{xy})\}, \tag{42}$$

which has the same order of accuracy as the elementary differencing and interpolation schemes.

The first two terms in curly braces have a structure similar to the energy-conserving versions of Arakawa's Jacobians, while the second pair corresponds to the enstrophy-conserving version. Each of these pairs constitute advection schemes in flux divergent form, and, in principle, can be used separately. Unlike the second-order Arakawa Jacobian (Arakawa 1966), which can be represented in both advective and flux-divergent form, higher-order, discrete versions of (42) cannot be written in advective form and also do not retain exactly the symmetry property $\mathcal{J}(\psi, \zeta) = -\mathcal{J}(\zeta, \psi)$.

From knowledge of ψ and ζ at time levels $n - 1$ and n, we can make the LF time step obtain the first guess at time step $n + 1$:

$$\hat{\zeta}^{n+1} = \zeta^{n-1} - 2\Delta t\mathcal{J}(\psi^n, \zeta^n). \tag{43}$$

Next, the new field is examined for the presence of overshoots, and the excess field is computed and diffused in the same way as in the ELAD procedure, Eqs. (31)–(33) in the previous section. Finally, from ζ^{n+1}, a multigrid elliptic solver is used to compute the new ψ. In order to suppress LF mode decoupling, a corrective step is used occasionally, just after the inviscid advective step but before the ELAD procedure. The advection and dissipation algorithm for the passive scalar q is the same as for vorticity ζ.

a. Alternative schemes for comparison

There is a huge variety of advection schemes, but we restrict our comparisons to multidimensional schemes that are extendable to high orders of accuracy. For example, we do not consider MPDATA (Smolarkiewicz 1984; Smolarkiewicz and Clark 1986) because it is not extendable; nor do we consider PPM (Colella and Woodward 1984) because it uses directional splitting. We will compare the following four classes of alternative schemes.

1) STD: The standard (STD) geophysical modeling practice is to use a low-order basic scheme with an explicit operator for weak dissipation. A representative of this is the second-order Arakawa Jacobian with an iterated Laplacian operator to provide dissipation (i.e., hyperdiffusion), for example, $\mathcal{D}_{\zeta} = -\nu_4\nabla^4\zeta$. Here the hyperdiffusion coefficient ν_4 is adjusted experimentally

to ensure adequate smoothness of the solution without excessive dissipation. In practice, this is achieved with a grid-size Reynolds number, $\mathrm{Re}_g = [|u|(\Delta x)^3]/\nu_4$, in the approximate range 20–50 for turbulent flows.

2) FCT: We use a version of the flux-corrected transport algorithm whose basic scheme is symmetric and sixth-order in space and third-order Adams–Multon in time. The flux limiting for FCT is similar to Zalesak (1979); see also appendix A. If the flux-limiting is turned off, the scheme remains stable and becomes roughly nondissipative. (The only remaining source of dissipation is due to time stepping, which is negligible in comparison with, say, the dissipation of an upstream-biased scheme.) Once the flux-limiting procedure is applied, it guarantees that the new values lie within the permissible bounds, $q_{i,j}^{(\min)}$ and $q_{i,j}^{(\max)}$, for example, as in Eq. (31). This algorithm produces strictly monotone behavior in its discrete solutions, but, as shown below, this occurs at the expense of excessive dissipation and ''clipping'' of narrow extrema.

To avoid clipping, Zalesak proposed an *extremum discriminator,* which assumes that extrema may occur between grid points; hence the permissible bounds may exceed the values at the surrounding grid points. Although this discriminator yields excellent results for advection in a simple flow such as solid-body rotation, we have found that its use in a turbulent flow may be dangerous because of confusion between dispersion-induced oscillations and legitimate physical extreme. Therefore, we have designed a more restrictive discriminator which accepts an extremum only if the advected quantity has its second derivative of only one sign over four consequent grid points in each coordinate direction, which insures that the accepted extremum is properly resolved (see appendix A). Since ELAD also exhibits a tendency toward clipping, albeit a milder one than with FCT (see below), we will also use a discriminator in a variant form of ELAD.

Whether or not the discriminator is used in this version of the FCT algorithm, virtually all the dissipation is due to the flux-limiting procedure; hence it is of the strong type. This property makes it qualitatively different from the class of forward-in-time, upstream-biased schemes, such as UTOPIA (see below) or semi-Lagrangian schemes, that inherently contain dissipation of the weak type.

3) UTOPIA: Leonard (1979), Leonard et al. (1993), Rasch (1994) and Leonard et al. (1995a,b) proposed the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA) as a two-dimensional generalization of his earlier algorithm, QUICKEST (Leonard 1979). Similarly to semi-Lagrangian schemes, UTOPIA provides partial compensation of phase errors caused by time and space differencing truncation errors due to mutually dependent time and space discretization (e.g., Davies 1980). UTOPIA has a (weak-) dissipation-dominant truncation error, proportional to the fourth derivative of the advected field (as in hyperdiffusion) and the local advecting velocity (cf. Leonard 1979). At the same time, unlike semi-Lagrangian schemes, UTOPIA is formulated in fully multidimensional flux-divergent form, which guarantees global conservation. UTOPIA can also be used as a basic scheme in a flux-limiting algorithm, for example, one that enforces monotonicity of the field in the vicinity of steep gradients (Thuburn 1995). Although developed independently, this algorithm can also be viewed as a higher-order generalization of the second-order scheme of Colella (1990) and also as a generalization of van Leer (1985). We use a particular version of UTOPIA developed by Rasch (1994), as described in appendix B.

4) γ-scheme: We have also devised an alternative high-order discretization for advection and dissipation of the weak type:

$$\begin{aligned}\mathcal{J}(\psi, \zeta) - \mathcal{D}_\zeta &= \frac{1}{2\Delta h}\Bigg[\mathrm{FX}_{i+1/2,j} - \mathrm{FX}_{i-1/2,j} + \mathrm{FY}_{i,j+1/2} - \mathrm{FY}_{i,j-1/2} \\ &\quad + \frac{1}{\sqrt{2}}(\mathrm{FX}'_{i+1/2,j+1/2} - \mathrm{FX}'_{i-1/2,j-1/2} \\ &\quad + \mathrm{FY}'_{i-1/2,j+1/2} - \mathrm{FY}'_{i+1/2,j-1/2})\Bigg], \quad (44)\end{aligned}$$

where $\Delta h = \Delta x = \Delta y$ is the horizontal grid spacing; FX and FY are the fluxes of advected quantity ζ in the x and y directions, respectively; and FX′ and FY′ are the fluxes in the diagonal directions x' and y':

$$\mathrm{FX}_{i+1/2,j} = u_{i+1/2,j}\overline{\zeta}^{x}_{i+1/2,j} + \gamma|u_{i+1/2,j}|\left(\frac{1}{3}\zeta_{i+2,j} - \zeta_{i+1,j} + \zeta_{i,j} - \frac{1}{3}\zeta_{i-1,j}\right)$$

$$\mathrm{FY}_{i,j+1/2} = v_{i,j+1/2}\overline{\zeta}^{y}_{i,j+1/2} + \gamma|v_{i,j+1/2}|\left(\frac{1}{3}\zeta_{i,j+2} - \zeta_{i,j+1} + \zeta_{i,j} - \frac{1}{3}\zeta_{i,j-1}\right)$$

$$\mathrm{FX}'_{i+1/2,j+1/2} = u'_{i+1/2,j+1/2}\overline{\zeta}^{x'}_{i+1/2,j+1/2} + \gamma|u'_{i+1/2,j+1/2}|\left(\frac{1}{3}\zeta_{i+2,j+2} - \zeta_{i+1,j+1} + \zeta_{i,j} - \frac{1}{3}\zeta_{i-1,j-1}\right)$$

$$\mathrm{FY}'_{i-1/2,j+1/2} = v'_{i-1/2,j+1/2}\overline{\zeta}^{y'}_{i-1/2,j+1/2} + \gamma|v'_{i-1/2,j+1/2}|\left(\frac{1}{3}\zeta_{i-2,j+2} - \zeta_{i-1,j+1} + \zeta_{i,j} - \frac{1}{3}\zeta_{i+1,j-1}\right),$$

where the velocity components are as in (41). The overbar with superscript denotes interpolation by (18)–(21) with $r = 4$ in the indicated direction.

The terms with coefficient γ represent a velocity-dependent hyperdiffusion. Because these terms at leading order are proportional to the third derivative of ζ and the third power of grid spacing Δh, they vanish when $\Delta h \to 0$, and thus they do not correspond to any physical term in the continuous equation. Their presence, however, ensures dissipative dominance of the truncation error, making the scheme third-order accurate with respect to the derivatives of advected scalar. This scheme has only weak dissipation and thus does not ensure monotonic behavior. The coefficient γ is an adjustable nondimensional parameter whose value controls the strength of the dissipation. The choice $\gamma = 1/4$ is equivalent to the use of three-point, asymmetric (upstream biased), quadratic interpolation for q to compute fluxes, as in Leonard (1979). This establishes a plausible upper limit for γ, but we have found that lower values suffice; below we present results with $\gamma = 0.05$, 0.1, and 0.2.

When $\gamma \neq 0$, the operator (44) cannot be used in combination with LF time-stepping algorithm due to instability. A third-order predictor–corrector step is used instead (Hyman 1979),

$$\begin{aligned}
\zeta^{n+1,*} &= \zeta^{n-1} - 2\Delta t[\mathcal{J}(\psi^n, \zeta^n) - \mathcal{D}_\zeta^n] \\
\zeta^{n+1} &= \frac{1}{5}\zeta^{n+1,*} \\
&\quad + \frac{4}{5}\left\{\zeta^n + \frac{\Delta t}{2}[\mathcal{J}(\psi^n, \zeta^n) - \mathcal{D}_\zeta^n \right. \\
&\quad \left. + \mathcal{J}(\psi^{n+1,*}, \zeta^{n+1,*}) - \mathcal{D}_\zeta^{n+1,*}]\right\},
\end{aligned} \tag{45}$$

where $n + 1,*$ indicates the preliminary value of the new time-step field, and the elliptic solver for ψ is used after both the predictor and corrector stages. Unlike UTOPIA, the γ-scheme has independent discretization of space and time. Consequently, it does not require transverse terms in the advective flux computations to overcome the computational instability associated with flux-splitting in multidimensions [see Leonard et al. (1996)].

b. Results

Our criterion for evaluating the performance of an algorithm is its ability to produce physically plausible solutions with as little dissipation of quadratic invariants as possible and essentially monotone behavior. Thus, we monitor energy, enstrophy, and quadratic variance, as well as the time history of gridpoint extrema in ζ and q (using a tracking algorithm described in appendix C).

Figure 8 shows the ζ and q fields soon after the coherent vortices begin to form. From the numerical point of view, this is the most difficult period because the structures are small and subject to strong deformation and straining. Despite the formation of quite strong gradients, no oscillatory behavior is apparent. Except within the vortex cores, the passive scalar is subject to strong mixing in the filaments roughly aligned with the vorticity filaments. This solution is calculated with the sixth-order advection scheme with ELAD. The grid resolution is 256×256, so a typical vortex core at this time is only 6–10 grid points in diameter, and filaments are only 3–4 grid points wide.

Figure 9 shows the history of vorticity and passive-scalar extrema for sixth- and fourth-order schemes with ELAD. The relevant velocity scale is set by the initial normalization of the nondimensional kinetic energy,

$$\begin{aligned}
\epsilon_{\text{kin}}|_{t=0} &= \frac{1}{2}\int\int |\nabla\psi|^2 \, dx \, dy \\
&\approx \sum_{i,j\in\mathcal{D}} (\psi_{i+1,j} - \psi_{i,j})^2 + (\psi_{i,j+1} - \psi_{i,j})^2 = 1.
\end{aligned} \tag{46}$$

The nondimensional timescale is based on this velocity scale and the domain size, 2π. There are many extrema in the random initial scalar fields, but most of them rapidly decay before $t \approx 20$ when the coherent vortices are formed. Afterward, there is a noticeable difference between the dominant extrema of vorticity in the coherent vortices, and the smaller extrema associated with the filaments in the background. The former remain virtually constant and the latter decay strongly. Virtually no weak ζ extrema can endure. On average, the weak q extrema decay faster than those of ζ, as indicated by the lower bottom envelope in Fig. 9b compared to Fig. 9a. The occasional sudden drop of the strong extrema at late times corresponds to the destruction of a vortex by deformation in the strong strain field due to other vortices nearby. The slowly decaying q extrema in Fig. 9b correspond to the concentrations that ''paint'' the cores of coherent vortices. Unlike the vorticity field, where the enduring extrema are the dominant ones, the passive scalar concentration trapped within a coherent vortex may be of any value established by the initial conditions. All q extrema not trapped within coherent vortices rapidly decay. One can see in Figs. 9a and 9b that the number of enduring extrema is exactly the same for ζ and q. There is no significant growth of any of enduring extremal values with this advection scheme, indicating a satisfactorily monotone behavior. Figures 9c and 9d show ζ extrema for fourth- and second-order schemes with ELAD. Notice that the degree of monotone behavior degrades as the order of the basic scheme is reduced, with the biggest change occurring between fourth and second orders.

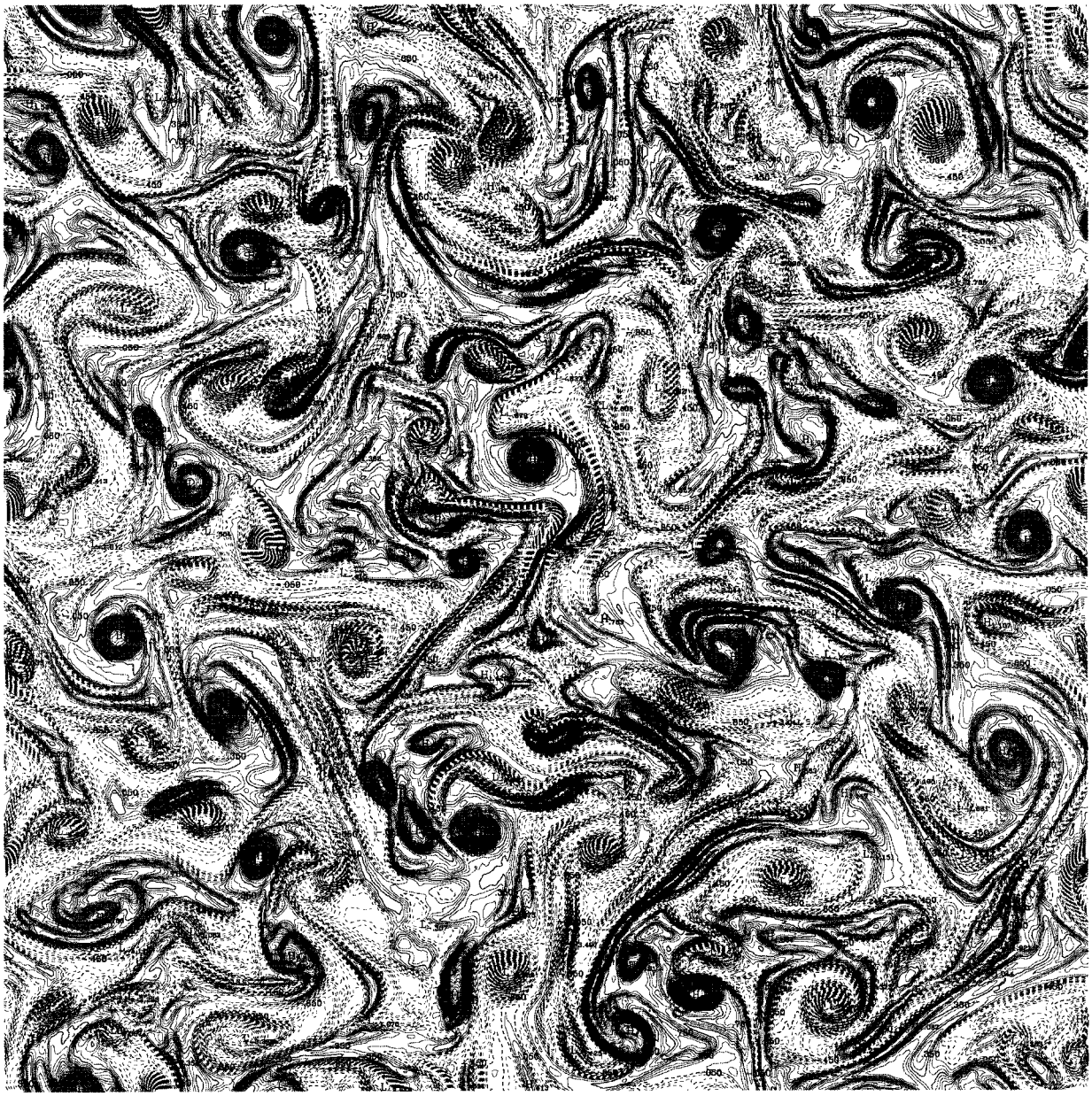

FIG. 8a. Snapshot of vorticity field at $t \approx 25.9$ for turbulent flow simulation computed with sixth-order accurate advection scheme with ELAD. Grid resolution is 256×256. The whole computational domain is shown. Minimum and maximum values are ± 10, contour interval is 0.1, so that there are approximately 200 contour levels on this plot. Negative values are contoured in dashed lines.

Figure 10 shows extremum histories for the FCT algorithm. Unlike the previous scheme, FCT without a discriminator (panel a for ζ and b for q) yields a solution that is strictly monotone in the discrete sense. Notice, however, that this is accomplished with considerably greater extremum decay than seen in Fig. 9, and this FCT algorithm yields one of the most dissipative solutions we have obtained. Figure 10c is for a version of FCT algorithm with the extremum discriminator proposed by Zalesak (1979). Unlike in Fig. 10a, the decay of the dominant extrema is virtually eliminated. The fast oscillations in the dominant extrema sampled on the discrete grid have an amplitude of about 1% of the extremum. They are associated with the motion of narrow vortices peaks along the discrete grid ($\approx$6–8 grid points wide), since the location of an extremum, as diagnosed by the discriminator, may coincide with a grid point (which gives the maximum values in Fig. 10c) or fall in between. Averaged over these oscillations, there is little tendency for systematic growth of extrema, although nothing in principle precludes the possibility. In the vicinity of an extremum, this version of the FCT

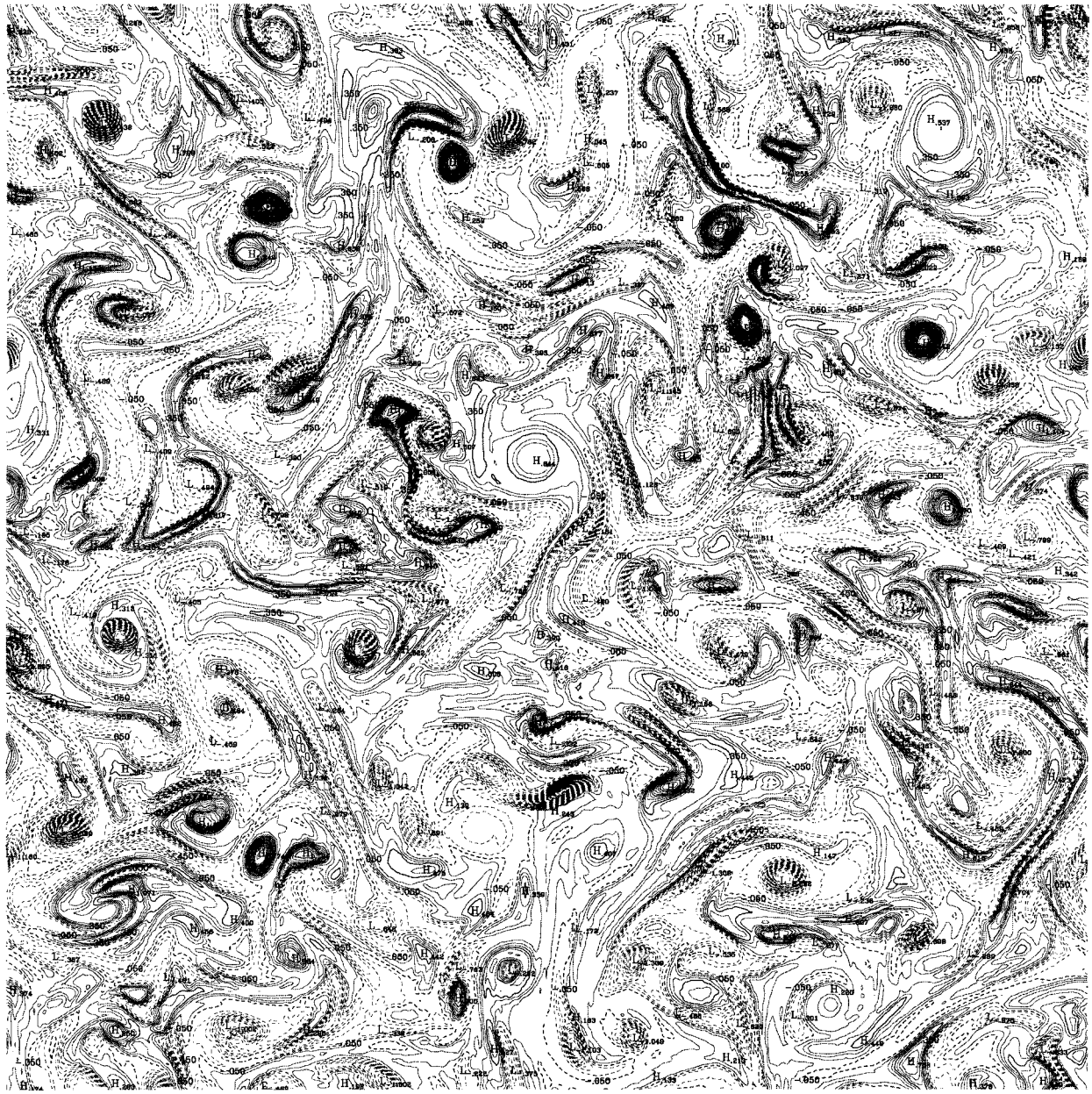

FIG. 8b. Snapshot of passive scalar field from the same flow simulation as in Fig. 8a. Time at which this snapshot is taken and contour interval are kept the same.

algorithm completely relies on the accuracy of its basic scheme. However, the behavior of the background extrema in this solution is unacceptably noisy, especially in the passive scalar (not shown), as evident in the large amplitude of the bottom envelope in Fig. 10c and the almost complete loss of monotone behavior after a new extremum arises. Figure 10d shows the effect of the more restrictive discriminator (appendix A). In comparison with Fig. 10c, its behavior is more physically meaningful for background extrema (almost as well as in Figs. 9a, 9c, and 10a), while in fact, looking more similar to Figs. 10c,d and preserving the dominant extrema better than do FCT and ELAD without discriminators.

To better preserve the dominant extrema with ELAD, we replace the constraint based on only gridpoint values with the same discriminator used in Fig. 10d—see Figs. 11a,b. The results are generally similar with FCT and ELAD, although the latter shows slightly less monotone behavior and dissipation of the background field. These disadvantages may be repaired by performing an additional two-dimensional convexity check in the ex-

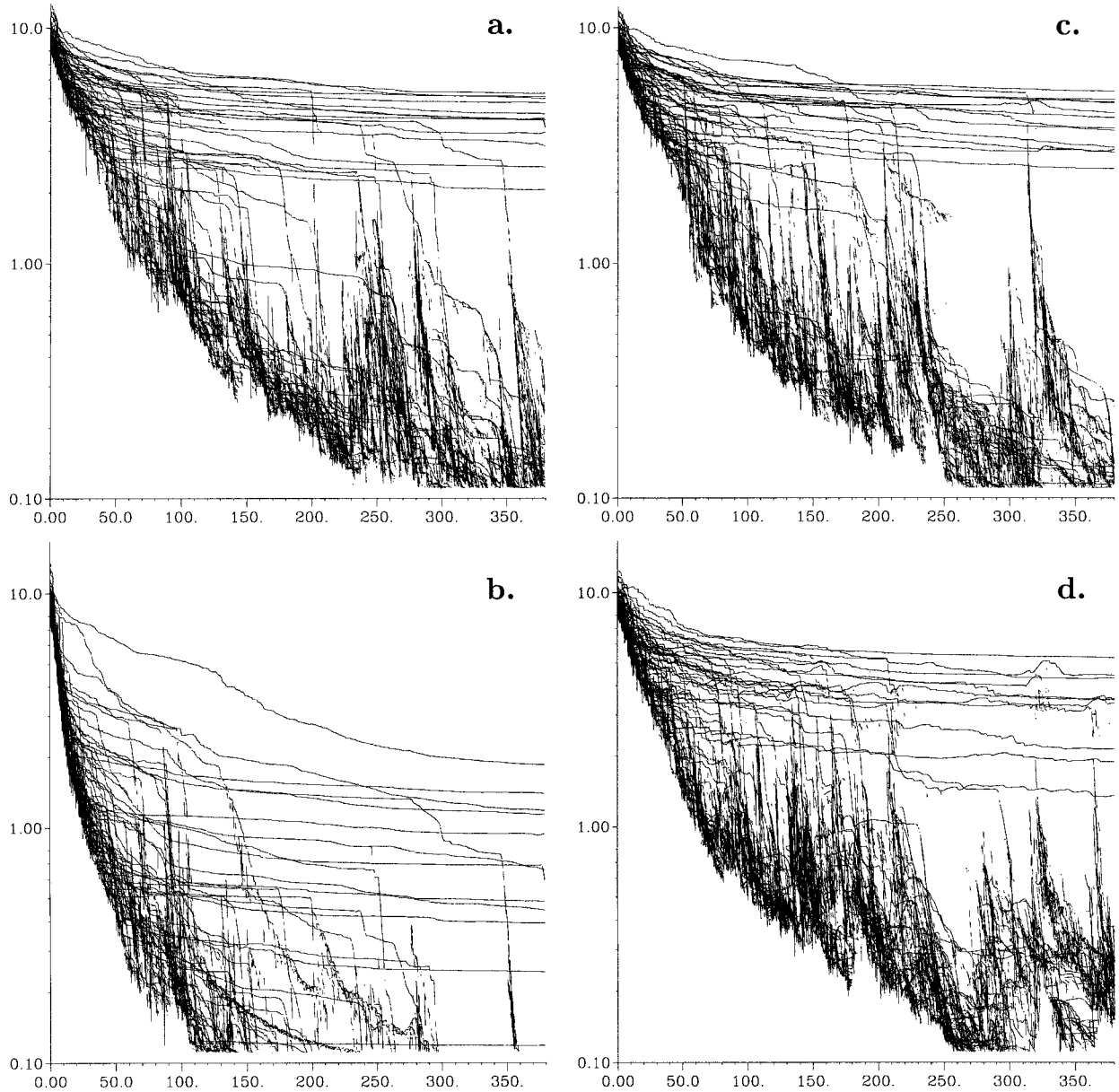

FIG. 9. Time history of extrema of vorticity and passive scalar for two-dimensional turbulent flow. (a) Absolute values of individual vorticity extrema as functions of time for the solution obtained by 6-ELAD. (b) The same as (a), but for the extrema of passive scalar. (c) The same as (a) (extrema of vorticity), but for 4-ELAD. (d) The same as (a) but for 2-ELAD; 64 largest minima and 64 largest maxima are traced in each case.

tremum discriminator. This involves replacing Eq. (A18), in appendix A. We compute

$$\begin{aligned} \mathrm{qxx}_{i,j} &= q^n_{i-1,j} - 2q^n_{i,j} + q^n_{i+1,j} \\ \mathrm{qyy}_{i,j} &= q^n_{i,j-1} - 2q^n_{i,j} + q^n_{i,j+1}, \end{aligned} \tag{47}$$

then check that the derivatives are of the same sign, and, if not, reset them to 0,

$$\text{if} \quad \mathrm{qxx}_{i,j}\mathrm{qyy}_{i,j} < 0 \Rightarrow \begin{cases} \mathrm{qxx}_{i,j} = 0 \\ \mathrm{qyy}_{i,j} = 0. \end{cases} \tag{48}$$

Otherwise the discriminator algorithm is unchanged. [Because operations (47) and (48) are local with respect to i, j, they are combined for execution efficiency.] The solution obtained with this discriminator is shown in Figs. 11c and 11d. Now there is little nonmonotone behavior (less than in Figs. 11a and 11b), and both almost as rapid decay of background extrema and less decay of dominant extrema than in ELAD without a discriminator (cf. Figs. 9a,b).

In Fig. 12, ζ extremum histories are shown for several algorithms with weak-type dissipation. The STD algo-

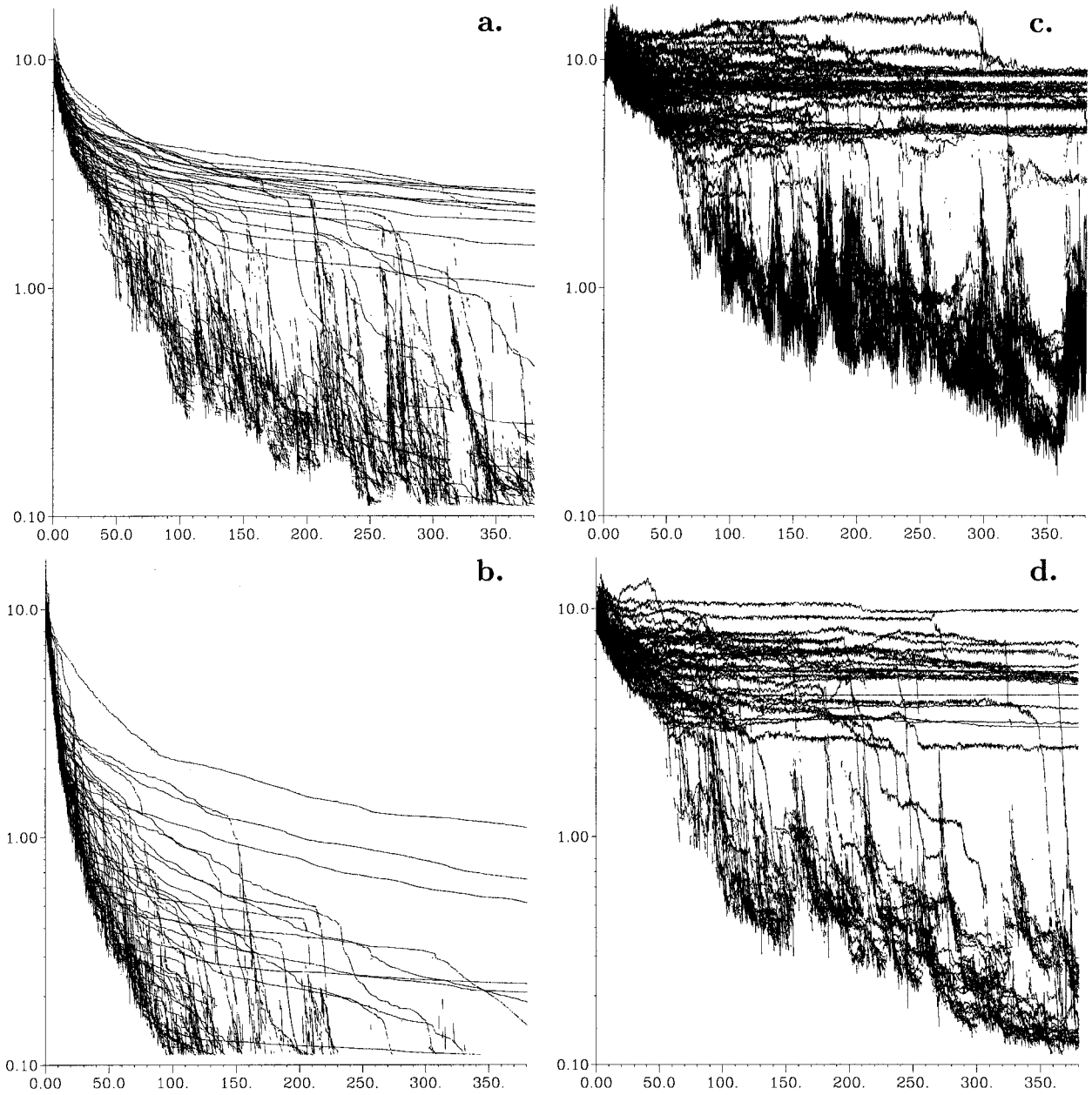

FIG. 10. The same as Fig. 9, but for the the different options of FCT algorithm. (a) and (b) History of extrema of vorticity and passive scalar, FCT without extrema discriminator. (c) The same as (a) (vorticity only), but the extrema discriminator from Zalesak (1979), is used to avoid clipping. (d) The same as (c), but with the use of sixth-point convexity check discriminator in each direction (see appendix A).

rithm (Fig. 12a), even with a hyperdiffusion coefficient chosen as a compromise between grid-scale smoothness and small dissipation, is still deficient by both measures compared with ELAD (with or without the discriminator; Figs. 9a and 11d) as well as with the γ-scheme (Figs. 12b,c with $\gamma = 0.05$ and 0.2) and UTOPIA (Fig. 12d). The scheme with $\gamma = 0.05$ does exhibit excessive nonmonotone behavior and background noise, but these gradually disappear with increasing γ in association with increasing dissipation; even $\gamma = 0.1$ (not shown) is satisfactory by these measures. With $\gamma = 0.2$ the extrema behave very similar to those with UTOPIA, except for a small increase in an enduring extremum immediately before its collapse during a destructive vortex interaction (associated with insufficient dissipation with the γ-scheme during strong deformation, as further discussed in section 5).

Although flux limiting was not used in the version of UTOPIA used here, and therefore monotone behavior is not formally guaranteed, there are virtually no oc-

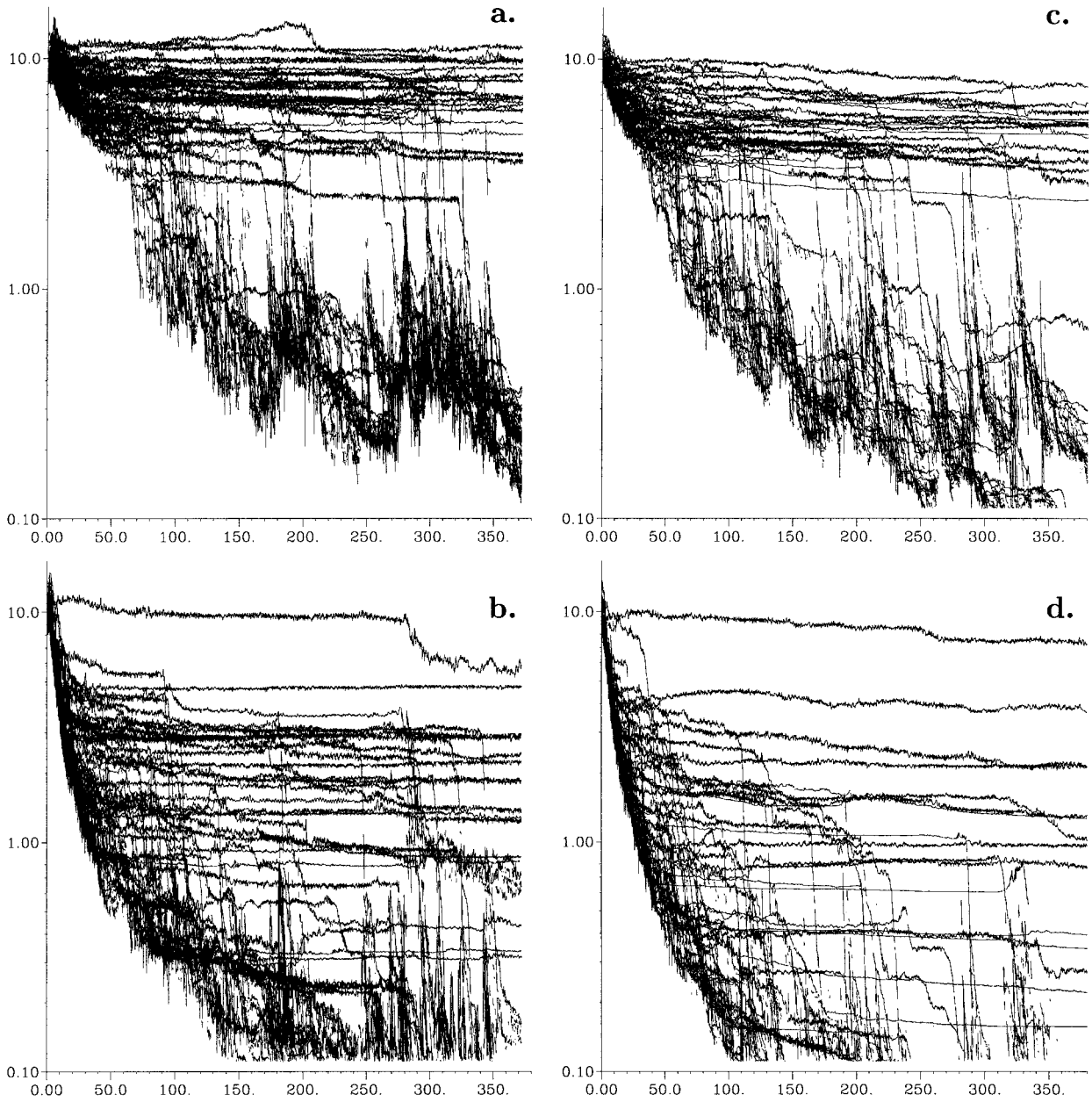

FIG. 11. The same as Fig. 9, but for 6-ELAD with two different versions of extrema discriminators: Histories of vorticity (a) and (c) and passive scalar (c) and (d) extrema are shown. (a) and (b) Modified extrema discriminator from appendix A (the same as in FCT algorithm in Fig. 10d) applied in each direction; (c) and (d) Same as (a) and (b), but for discriminator with two-dimensional convexity check.

casions of artificial extremum growth. This scheme produces a smooth, physically meaningful solution. The similarity between Figs. 12c,d is not arbitrary. One can show that the truncation errors (based on Taylor series analysis) of UTOPIA and γ-scheme [Eq. (44) with $\gamma = 0.2$], are very similar (both are third-order accurate). This is further confirmed by the histories of dissipation, Fig. 13. One can see however, that unlike the γ scheme, UTOPIA produces less artificial rises in extremal values before their drop in the case of vortex destruction. This is because of the generally smaller dispersive error for the highest resolved Fourier components due to phase error compensation of an algorithm with dependent discretization of time and space. Recall that in the case of constant advecting velocity UTOPIA becomes equivalent to semi-Lagrangian upstream interpolation algorithm (cf. Fig. 3).

Time series of energy, enstrophy, and passive-scalar variance are shown in Figs. 13a–c. Algorithms with ELAD are, in general, less dissipative than both algo-

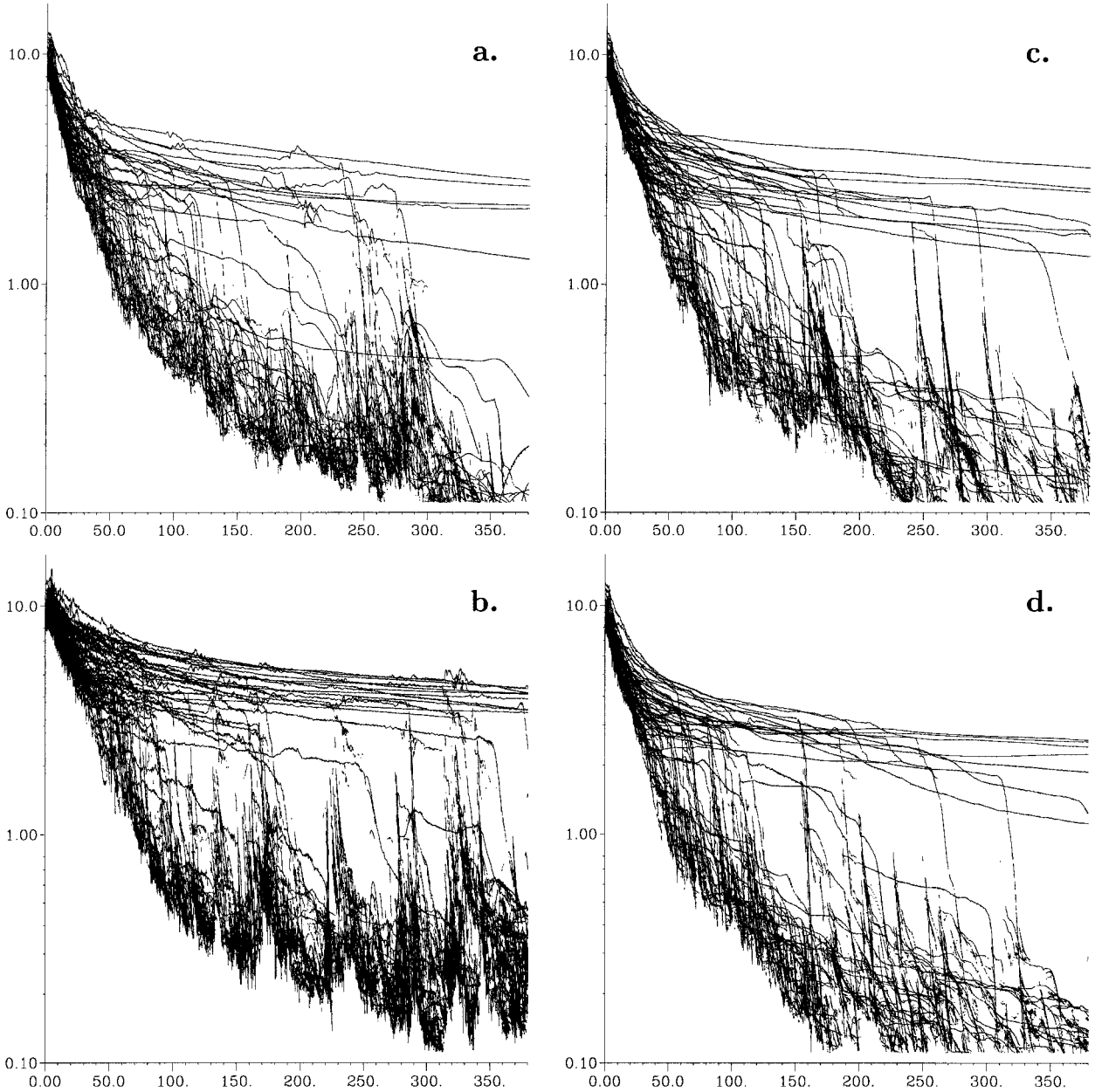

FIG. 12. (a) The same as Fig. 9a (vorticity extrema only) but for the ''standard'' second-order-accurate Arakawa Jacobian with explicit hyperdiffusion $\mathcal{D}_\zeta = -\nu_4\nabla^4\zeta$; (b) $\gamma = 0.05$ scheme; (c) $\gamma = 0.2$ scheme. (d) UTOPIA: In case (a) the hyperdiffusivity coefficient ν_4 is chosen to make $\nu_4/[|u_{\max}|(\Delta x)^3] = 0.0125$, where $|u_{\max}|$ is the maximum absolute value of velocity. Although the overall dissipation of this simulation is comparable to that of UTOPIA as well as γ-scheme Eq. (42) with $\gamma = 0.2$, this solution appears to be much noisier.

rithms with FCT and schemes with weak-type dissipation. The use of our new discriminator reduces the dissipation in both ELAD and FCT algorithms. Algorithms based on a second-order scheme are relatively dissipative, but fourth- and six-order schemes have similar dissipation rates. Passive-scalar dissipation is much greater than enstrophy dissipation, because the enduring vortex cores typically contain larger ζ than q values. The energy dissipation is the least among these quadratic measures, as expected from its inverse-cascade tendency in two-dimensional turbulence, and it occurs almost entirely during the early period before the coherent vortices emerger. The total energy losses over the integration period in Fig. 13a vary significantly among the different algorithms by as much as a factor of 5. The relative ordering among the algorithms is generally similar in all three dissipation measures, but FCT is anomalously dissipative in energy even with the use

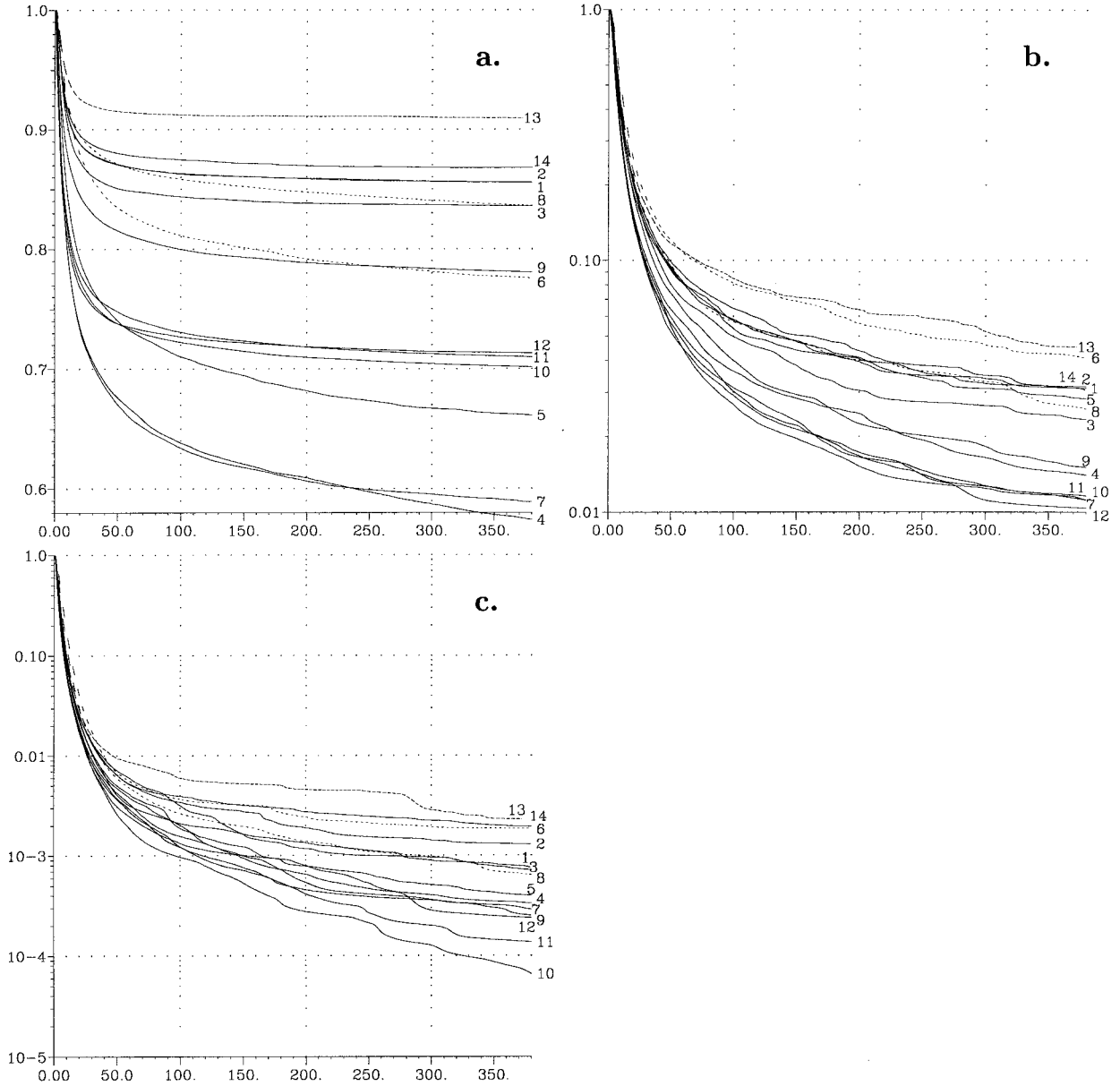

FIG. 13. Time history of resolved quadratic moments for turbulent flow similation. (a) Energy, normalized by its initial value; (b) enstrophy; (c) quadratic variance of passive scalar. Labels 1, 2, and 3 correspond, respectively, to *sixth-*, *fourth-* and *second*-order centered advection schemes combined with ELAD; 4, 5, 6 to FCT algorithm with sixth-order accurate in space, Adams–Multon time step basic scheme, and 4, no extrema discriminator; 5, new discriminator, see appendix A; 6, discriminator from Zalesak (1979); 7, the same as 4, but the basic scheme is now second-order accurate in space; 8, 9, and 10, algorithm (42) with $\gamma = 0.05$, 0.1, and 0.2, respectively; 11, UTOPIA; 12, second-order Arakawa Jacobian with hyperdiffusivity; 13, sixth-order ELAD with extrema discriminator similar to that used in 5; 14, the same as 13, but with two-dimensional convexity check in the extrema discriminator.

of an extrema discriminator. Causes of this effect are discussed in appendix A.

5. ELAD as a minimal sufficient model for subgrid-scale mixing: How does it work?

Our hypothesis about the physical mechanism of dissipation in turbulent flows with high Reynolds number is the following. Straining creates regions with steep gradients of vorticity and passive scalar. In these regions the molecular diffusive fluxes become strong enough to limit further steepening. In terms of the dissipation rate the details of the molecular dissipation, such as diffusion coefficients, are not important. The molecular diffusion simply ''absorbs'' the nonlinear cascade and, therefore, is entirely controlled by it. Decrease (or increase) of

molecular transport coefficients would result in subsequent increase (or decrease) of gradients of the advected quantity, so that the net effect of molecular fluxes remains virtually unchanged. Conversely, the molecular dissipation plays no significant role, unless it is enhanced by the nonlinear cascade.

By their design, adaptively dissipative mechanisms have a purely numerical nature: advective flow with strong deformation creates steep changes in gradients of the advected quantity, which can no longer be resolved by the discrete advection operator. Centered advective schemes generate dispersive overshoots, which are detected and diffused by the correction procedure. In this section we show that this numerical mechanism also creates physically interpretable dissipation in a situation when dissipation is both physically necessary and numerically unavoidable. Of course, the corrections triggered by violation of a given constraint may underestimate the totality of the cascade routes to dissipation, but we here examine the hypothesis that the minimal dissipation provided by adaptive schemes is sufficient.

We have selected an event corresponding to a sudden drop of a ζ extremum at $t = 200$ on Fig. 9a. During this event a vortex is strained by a stronger nearby vortex, and a partial merger ensues. The initial state is chosen at $t = 199$ from the simulation obtained by a sixth-order scheme with ELAD. It is shown on the leftmost panel of the third row (labeled as 6-ELAD) in Fig. 14, where only a 51×51 gridpoint portion of the whole computational domain of 256×256 is plotted. The small rectangle drawn in the bottom-left corner of this panel represents an 8×8 fragment of the actual grid used in this simulation. The vortices are approximately 8–10 grid points across, which suggests that they are only marginally resolved. We expect that their evolution will generate even smaller scales, which unavoidably are underresolved and, therefore, must be suppressed by the dissipative mechanism of the numerical algorithm.

This marginally resolved dynamics is compared with solutions produced on a finer grid. For these we interpolate the initial vorticity field to a twice-as-fine grid using a ''convex and monotone'' procedure:

$\mathcal{P}[\zeta_{\{c\}} \rightarrow \zeta_{\{f\}}]$:

$$\zeta_{i+1/4,j+1/4} = \frac{9}{16}\zeta_{i,j} + \frac{3}{16}\zeta_{i+1,j} + \frac{3}{16}\zeta_{i,j+1} + \frac{1}{16}\zeta_{i+1,j+1}$$

$$\left.\begin{aligned}\zeta_{i+3/4,j+1/4} &= \\ \zeta_{i+1/4,j+3/4} &= \\ \zeta_{i+3/4,j+3/4} &=\end{aligned}\right\} \begin{cases}\text{similar bilinear interpolation} \\ \text{based on } \zeta_{i,j},\ \zeta_{i+1,j},\ \zeta_{i,j+1},\ \zeta_{i+1,j+1}.\end{cases} \tag{49}$$

Since all weighting coefficients in (49) are positive, no extremum can be created that is not present on the coarser grid. The procedure is applied to the interpolated field to double its resolution once again. The refined fields are used as the initial conditions for solutions on grids 512×512 and 1024×1024, shown in the top and second from the top rows in Fig. 14. The results using 6-ELAD at the three different resolutions are in the top three rows in Fig. 14. In each row the times shown are $t \approx 199$, 207, 215, 223, and 231.

Because the effective dissipation of an ELAD scheme depends on the grid resolution, the three different solutions are for three physically different problems. Thus, we do not expect convergence in a classical sense with refinement of the grid. Instead, we observe the appearance of finer and finer structures, while the behavior of the features resolved by all of the grids retain similarity. After one of the vortices is strained, the filament (see middle pattern for each sequence) is only three to four points wide for each resolution. When axisymmetrization of the surviving vortex occurs, the gradients of vorticity are adjusted to approximately three grid points wide. (For high-resolution cases it may take a longer time, since the scale change is greater and the dissipation is reduced.) The radial vorticity profile has a distinct central core with approximately the same peak value of vorticity as in the larger vortex of the initial condition. Around this core there is a peripheral ''skirt'' of smaller vorticity, which was produced by a mixing of fluid from the strained vortex with the ambient fluid. After axisymmetrization, this profile will persist until the next strong vortex interaction.

4-ELAD produces a result very similar to 6-ELAD. The only visible differences are a slightly larger area of negative vorticity (this is the overshoot seen in Fig. 14 in dashed lines) and an insignificant difference in orientation of the vortices on the second panel. During the transition from the first to the second panel, the line connecting the centers of the vortices turns slightly more than 180° in the counterclockwise direction, and the lower-order basic scheme results in a small phase delay, hence a different orientation.

Use of a second-order basic scheme (in 2-ELAD) results in a catastrophic loss of both accuracy and monotone behavior. The orientation of the vortex pair is significantly off its true position and the magnitude of dispersive overshoots is comparable to the value of vorticity of the strained core. The fundamental explanation of this failure is the relatively low wavenumber of dispersive breakdown in the second-order scheme (Fig. 2). Because the ELAD procedure can efficiently suppress oscillations only on the smallest resolved scale, the combination of the second-order scheme with ELAD results in an algorithm that has an intermediate range of spatial scales which are neither accurately advected by the basic scheme nor effectively suppressed by ELAD. In 2-ELAD in Fig. 14, dispersive error causes an amplification of vorticity of the strained core and, at the same time, creates a spurious region of negative vorticity just behind the strained core. The region and the core form a dipole, which tends to propagate away from the larger vortex, and, as a result, merger is inhibited and virtually no vorticity from the strained vortex attaches to the larger vortex.

The FCT algorithm with a sixth-order basic scheme

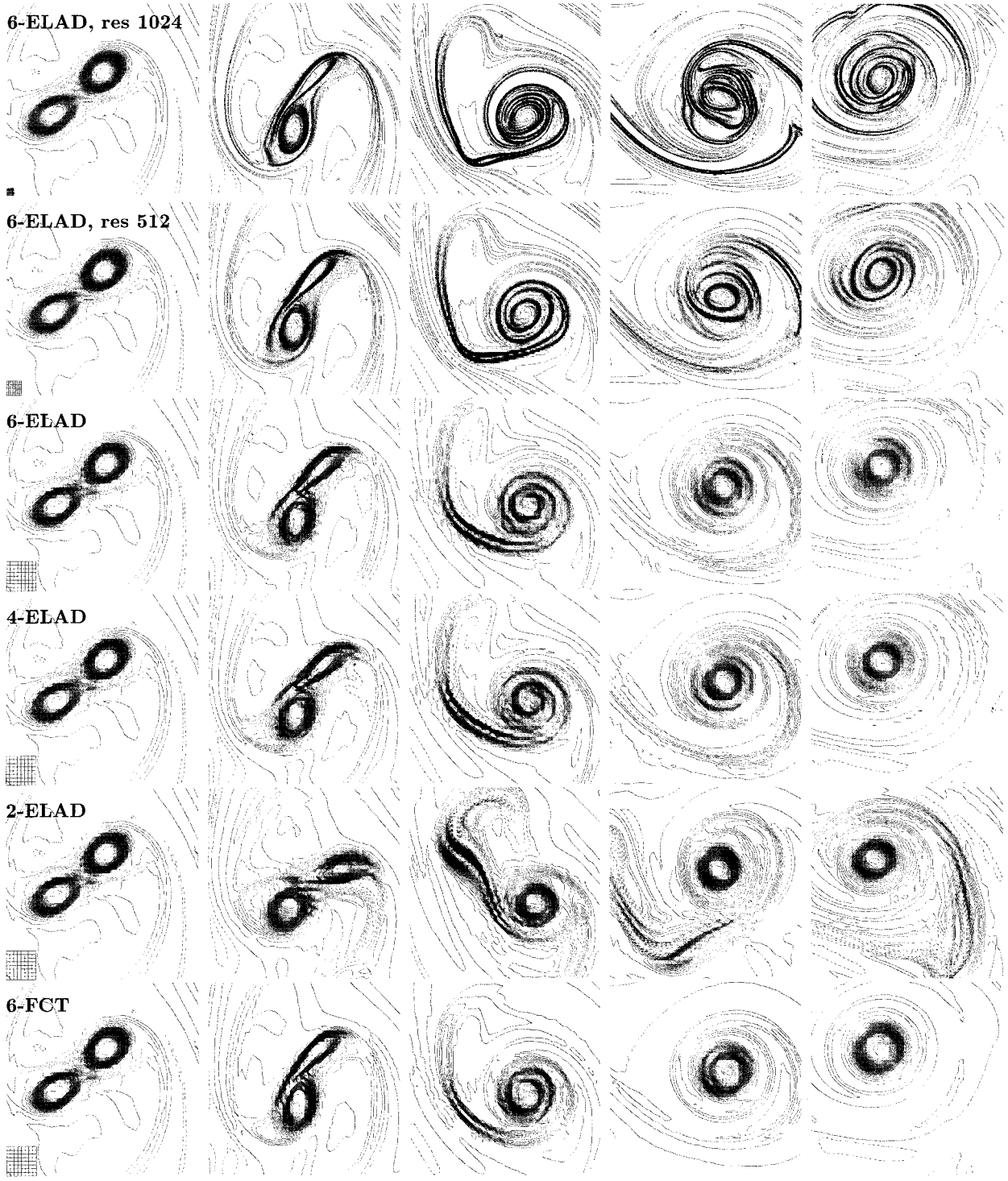

FIG. 14. Vortex straining event simulated by different schemes with dissipation of strong type.

produces results similar to 6- and 4-ELAD in Fig. 14, with a slightly smaller amplitude of filaments in the background and a somewhat larger surviving vortex with a flatter profile in the core. Unlike ELAD, this version of FCT has a strictly monotone behavior on the discrete level, thus no negative vorticity (dashed lines) is present in this solution.

Figure 15 represents the weak dissipation class of algorithms. Although the built-in dissipation in both UTOPIA and γ-scheme (44) depends on the local ve-

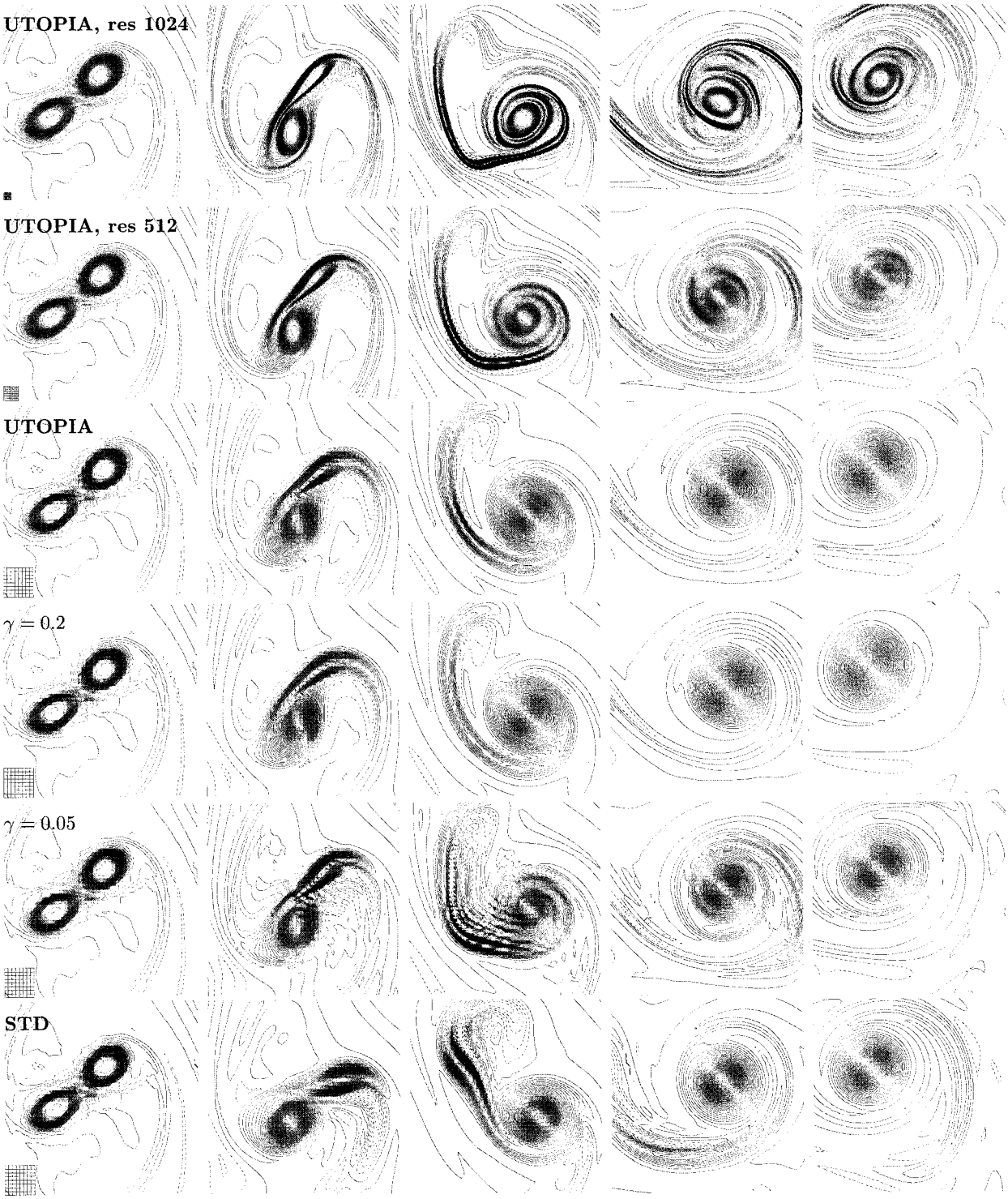

FIG. 15. The same as Fig. 14, but for schemes with dissipation of weak type.

locity magnitude, and therefore has some adaptive aspects, these schemes do not make any explicit attempt to enforce monotonicity. They provide physically acceptable solutions, although some generation of artificial extrema can be seen near the tail of the filament and between the filament and the surviving vortex (see three middle times on Fig. 15). UTOPIA and the $\gamma = 0.2$ scheme produce very similar results. A choice of too

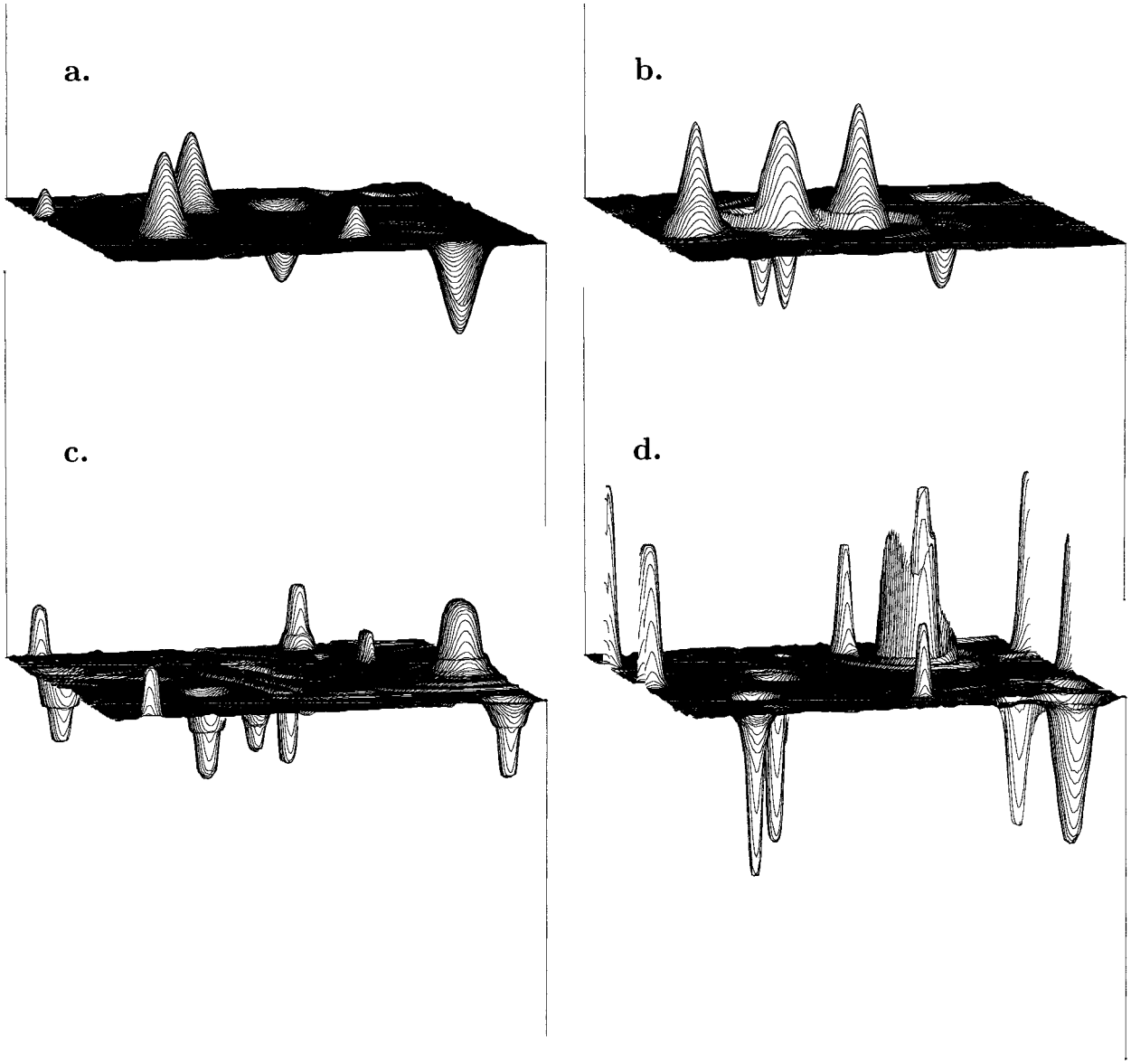

FIG. 16. Three-dimensional plots of vorticity field at $t \approx 400$ for turbulent flow simulations produced by different numerical schemes. (a) UTOPIA; (b) $\gamma = 0.1$ scheme; (c) 6-FCT; (d) 6-ELAD. Grid resolution is 256×256. Vertical spikes at rear-left and front-right corners indicate the largest vorticity values of the initial state.

small a γ value results in spurious oscillations (see $\gamma = 0.05$ middle panel), without returning a significant reduction of the overall dissipation during the event. The artificial extrema eventually go away, but clearly the scheme does not provide enough dissipation to prevent their creation. At the same time, the overall smoothing of the ζ profile in these three cases is obviously larger than that for adaptive schemes, Fig. 14.

As with 6-ELAD in Fig. 14, we have calculated solutions with three different resolutions using UTOPIA. The convergence tendencies with the grid refinement for the UTOPIA scheme are similar to that of ELAD schemes, except for providing a somewhat lower effective resolution. Thus, the final pattern of the vortex straining event experiment for UTOPIA on 1024×1024 grid is similar to 6-ELAD on 512×512, rather than on the finest grid. If the 1024×1024 UTOPIA calculations were carried forward, the vortex profile would evolve into a smooth shape, similar to that obtained on the coarser grids, even without further strong vortex interactions (unlike with ELAD). The final row in Fig. 15 is based on the STD algorithm. As expected, its behavior is less accurate, produces more oscillatory behavior, and causes greater vortex profile smoothing than all previously described algorithms.

The primary flaw of scale-selective dissipation—ei-

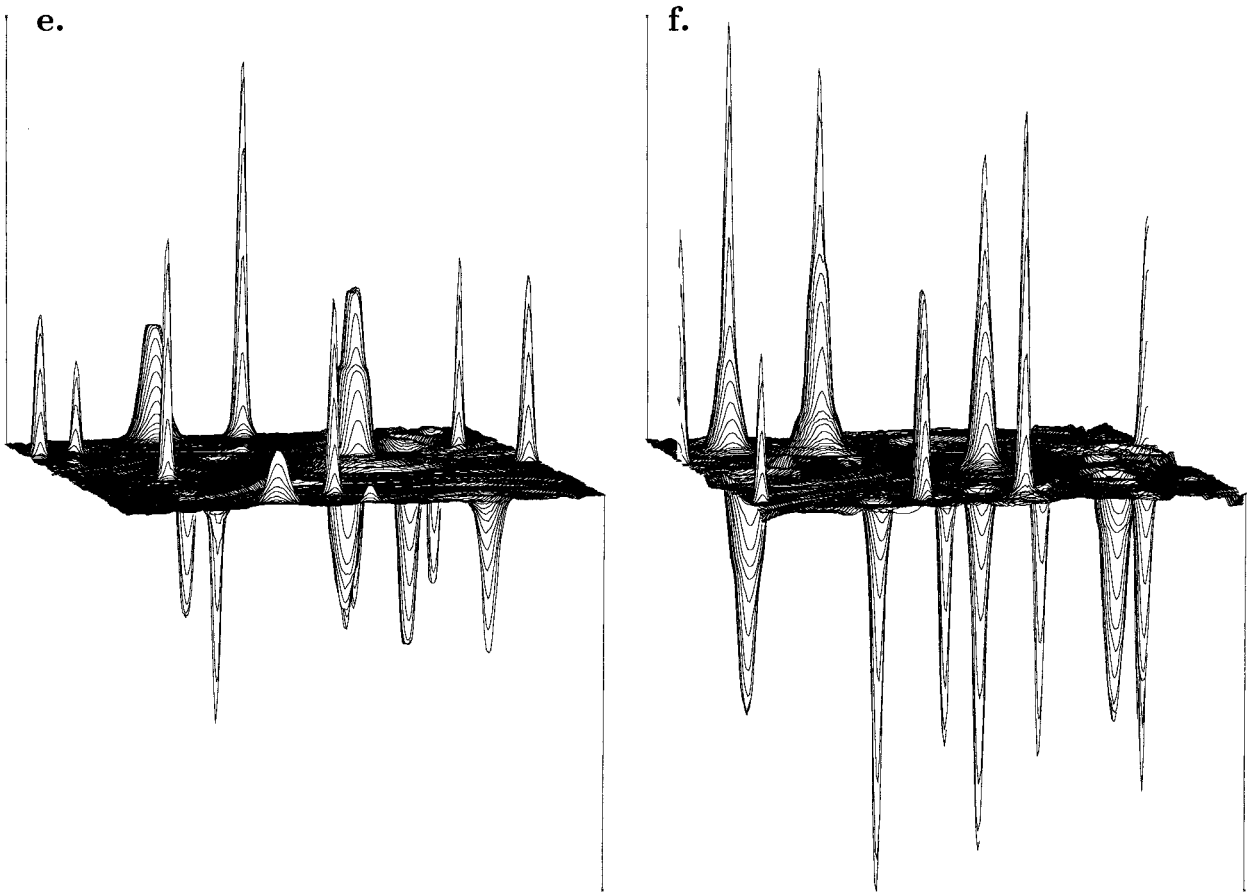

FIG. 16. *(Continued)* (e) 6-FCT with extrema discriminator; (f) 6-ELAD with extrema discriminator. The same as (a), (d), and (f) but for grid resolution 512 × 512. (g) UTOPIA; (h) 6-ELAD; (i) 6-ELAD with extrema discriminator.

ther by hyperdiffusion or through the truncation terms of weak-type dissipation—stems from its primary hypothesis about the turbulent cascade. In two-dimensional turbulence, there is greater net dissipation of enstrophy, which cascades toward small scales, than of energy, which cascades toward large scales. Hyperdiffusion is a simple operator that produces such effects. However, these properties of the nonlinear cascade are meaningful only in a statistical or domain-integrated sense, while hyperdiffision imposes this tendency locally at every grid point, without regard to the particular flow configuration or dynamical behavior. As a consequence, scale-selective dissipation is often a compromise that is both locally insufficient and globally excessive. The design philosophy behind FCT and ELAD (as well as any other algorithm with dissipation of strong type) is the principle of monotonicity. It tries to provide a physically adequate treatment of all scales supported by the grid in a deterministic, rather than statistical, sense while suppressing only the features that cannot be properly treated by the basic advection scheme. There is no explicit assurance that adaptive dissipation will ensure the desirable statistical properties of the turbulent cascade, but these nevertheless occur simply as consequences of the nonlinear local fluid dynamics.[2]

6. Numerical biases imposed on long-term dynamics and statistics of turbulent flow

The vortex straining event discussed in the previous section has a duration of 30 time units near $t = 200$. Given that a typical value of peak vorticity is approximately 10, this interval is about 10 parcel recirculation times within

[2] It should be mentioned that UTOPIA may be combined with a flux-limiting procedure similar to FCT (Thuburn 1995). The resultant algorithm guarantees monotone behavior, at the expense of increasing the overall dissipation of the algorithm, which is already more dissipative than any of the algorithms presented in Fig. 14. A potential remedy is to generalize UTOPIA to higher (odd) orders of accuracy, thus maintaining the dissipative dominance of the truncation error. This measure will decrease the built-in dissipation, making the subsequent flux-limiting the primary dissipative mechanism [e.g., NIRVANA in one dimension and MACHO in multidimensions (Leonard and Niknafs 1991; Leonard et al. 1995a; Leonard et al. 1996]. This path thus has some similarities with that of FCT and ELAD.

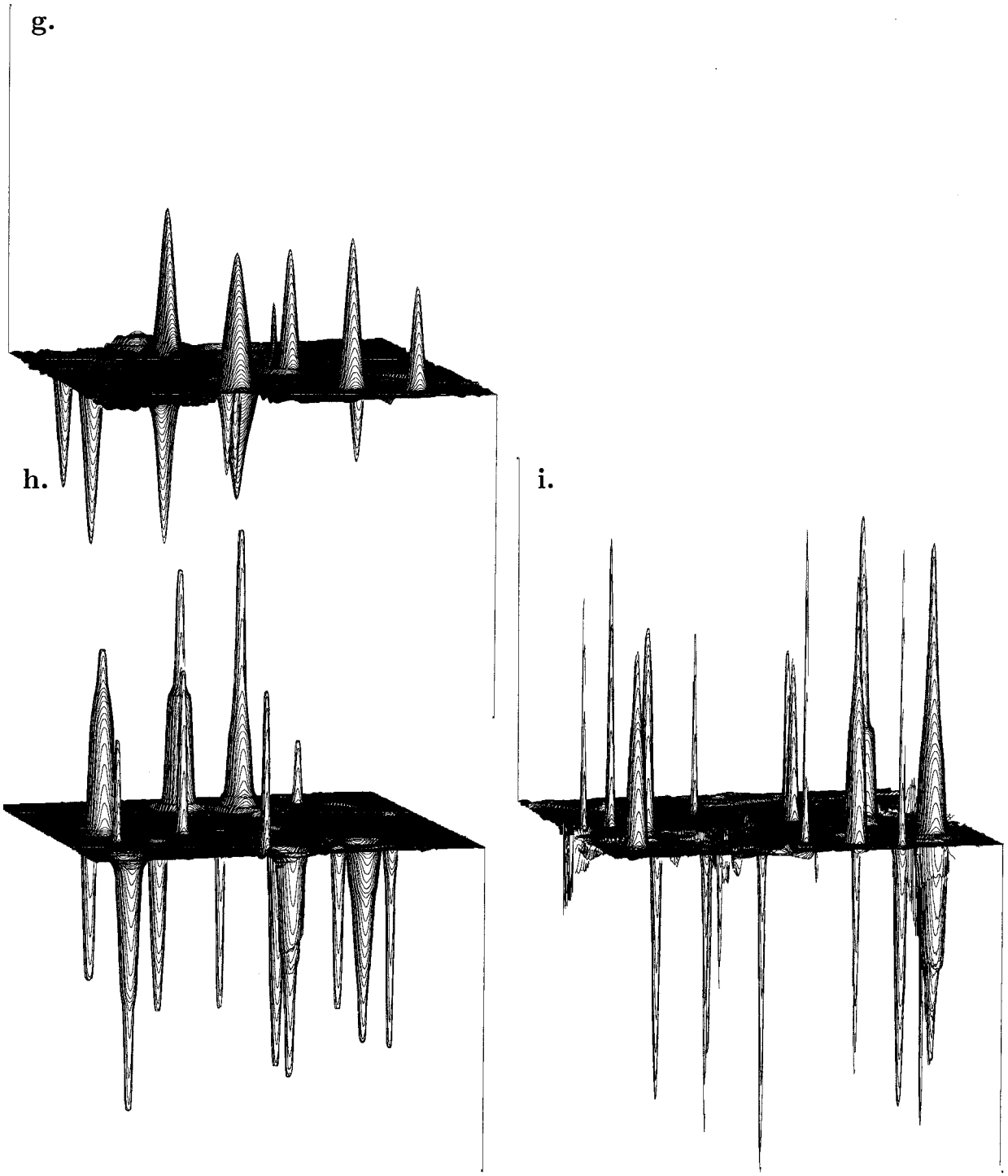

FIG. 16. *(Continued)*

a coherent vortex, which is only 7% of the integration period for the calculated solutions.

Over an integration spanning many vortex straining events, biases due to the numerical algorithm may accumulate and contaminate the statistics of the turbulent flow. One such effect, mentioned above, is the *numerical aging* associated with hyperdiffusion, either as an explicit operator or as implicit in the γ-scheme and UTOPIA. Hyperdiffusion tends to smooth vorticity gradients indiscriminantly both in the background vorticity filaments (en

route to being physically dissipated because of their continuing stretching) and on the edges of the vortex cores (where, once an axisymmetric, stable, stationary state occurs and filamentation ceases, there is no physical mechanism for continued dissipation). Maintaining the vorticity gradient on the vortex edge is conducive to resisting further vortex filamentation in straining flow. When hyperdiffusion smooths a vortex profile, it both broadens the vortex radius, which expands its critical merger distance and increases its interaction cross section with other vortices, and it creates a broader peripheral skirt of weak vorticity that can be swept away in filaments by straining, thereby weakening the vortex circulation. These effects will act more strongly on smaller vortices and thereby make them relatively less persistent than larger vortices, beyond whatever physical tendencies there might be.

Vorticity profiles at the late time of $t \approx 400$ are plotted in Fig. 16 for the whole computational domain of 256×256 grid points. Only a few large vortices remain in solutions with dissipation of weak type: UTOPIA (Fig. 16a) and $\gamma = 0.1$ scheme (Fig. 16b). In both solutions, albeit to an apparently lesser degree in Fig. 16b with this smallish value of γ, the sizes, shapes, and vorticity gradients of all surviving vortices appear to have undergone significant numerical aging over the individual vortex histories. This interpretation is based on the expectation that extrema within surviving vortices should be preserved (Carnevale et al. 1991; Weiss and McWilliams 1993).

In contrast, the algorithms with dissipation of strong type, ELAD (Figs. 16d,f) and FCT (Figs. 16c,e), tend to produce vortices with much stronger edge gradients in ζ. The resultant vortices also preserve their shape except when they are involved in vortex interactions. After emergence, the vortices remain somewhat more compact and occupy a smaller fraction of the domain, compared to solutions with hyperdiffusion. Hence, the probability of inelastic interactions for each individual vortex is also smaller, so the vortex number remains larger because of more surviving small vortices. Thus, there appears to be less numerical aging here. One apparent bias is in the vortex core shapes, where both algorithms without the use of discriminators (especially FCT; Figs. 16c,d) tend to produce overly flat core distributions, suggesting that vorticity mixing takes place in a region where it is expected to be small. Although its rate is slower than that caused by hyperdiffusion, this results in growth of the vortex radius and reduction of the extremum amplitude. Both of these effects decrease the survival probability for an individual vortex and act preferentially on the smaller vortices.

This bias may be countered by using extremum discriminators: see Fig. 16e for FCT with the discriminator algorithm in appendix A and Fig. 16f for ELAD with this algorithm augmented by the convexity check in Eqs. (47)–(48). Now the vortices have much larger extrema and are somewhat narrower. Small vortices of six to eight grid points in diameter may survive during the whole period of time without significant decay. The vortex profiles now more evidently depend on their histories of interaction; one can see multiple peripheral skirts for some vortices in Figs. 16e,f indicating incremental growth through mergers with weaker vortices.

To verify these findings, we repeated the numerical calculations with the ELAD and UTOPIA algorithms on a finer grid of 512×512. The initial condition was obtained by the interpolation described in the previous section. Because now there are twice as many points per vortex extremum in the initial state, the vortices are better resolved and we do not expect that extremum discriminator to make as much difference as in the 256×256 solutions. The results are shown in Figs. 16h,i. Now the decay of extrema in the absence of a discriminator is significantly smaller, and although the flat-core shape bias is still evident, it occurs at a smaller scale relative to the typical vortex radius. Use of the discriminator still brings some change, but it is not as strong as on the coarser grid. Furthermore, solutions at different grid resolutions using ELAD with a discriminator (Figs. 16f and 16i) exhibit relatively modest differences, although there are still more vortices and steeper edges on the finer grid. A similar resolution comparison for solutions with UTOPIA can be made between Figs. 16a and 16g. The resolution tendencies are the same as with ELAD: more vortices remain and gradients are steeper. However, the biases toward lower extremum amplitude and smoother profile shape remain. In terms of the number of remaining vortices and sharpness of vorticity gradients, the 512×512 UTOPIA solution is more like that of the 256×256 FCT and ELAD solutions with extremum discriminators, rather than their 512×512 counterpart.

7. Convergence of dissipative properties of turbulent flow

Despite the clear indication that adaptive techniques are generally less dissipative, the fundamental question—Do they produce the *optimum* amount of dissipation?—remains to be answered. This question is nontrivial, because reliable fully converged solutions for turbulent flows are not available. No discrete model can resolve all possible scales, especially in the situation when there are nonlinear cascades toward small scales. Some dissipative mechanism must be employed to prevent accumulation of excessive variance in the small scales. If achieving high Reynolds number is the goal, so that the Reynolds number based on grid size, physical velocity, and molecular viscosity is much larger than 1,[3] viscous dissipation can no longer accomplish this

[3] In practice, for the turbulent flow simulations with a reasonably high-order accurate centered (nondissipative) finite-difference advection scheme the actual threshold is roughly 10–20. This also sets a limit to the maximum physical Reynolds number that can be achieved on a given grid for a simulation that resolves viscous processes directly, making it impractical to use such solution on a fine grid for the purpose of verification of a subgrid-scale mixing model run on a relatively coarse grid.

goal, and apparently no other physical mechanism is available. Moreover, because the threshold between scales that *are* and *are not* correctly represented is set by the *discretized* advection scheme, introduction of an additional artificial term into the continuous equation immediately introduces the dilemma of double counting—for some scales the subgrid dissipation model adds to the physically resolved processes. The opposite situation may also occur, where the basic advection scheme cannot properly handle strain-induced gradients and the resultant discretization errors are not suppressed by the subgrid model. The former situation results in overdissipated and the latter in underdissipated solutions. Underdissipation leads to an unphysical solution because of aliasing contaminating the resolved scales. Overdissipation is, on the other hand, somewhat less harmful because it can be simply viewed as underutilization of the grid resolution, and in principle, be compensated for by the use of a finer grid. In practice, however, every doubling of resolution for a two-dimensional problem causes an increase of computational effort by a factor of 8 (2 of which is due to a finer time step required to satisfy the CFL criterion), which strongly limits the practically achievable Reynolds number.

It is instructive to compare histories of decay of enstrophy and quadratic variance of passive scalar produced by the same scheme at different resolutions. In the best case, the subgrid-scale model would have the property that scales resolved by both fine and coarse grids behave in a similar way, while scales that can be resolved only on the fine grid are simulated directly on that grid, and if the same simulation is carried out on the coarse grid they are suppressed without causing feedback to the resolved scales. Of course, this is applicable only for a quantity that is cascaded toward small scales. This is not valid, for example, for energy in 2D turbulent flow, because energy is cascaded toward large scales. In fact, we have found that the time history of energy converges with the refinement of the grid toward nearly zero dissipation, and for the finest resolution we have used (1024 × 1024 grid, 6-ELAD with discriminator), the system loses less than 1% of its initial energy.

We interpolate the initial vorticity and passive scalar fields from 256 × 256 to 512 × 512 grid using the ''monotone'' interpolation procedure (49) described in the previous section. Then the interpolated field is interpolated again to 1024 × 1024 grid by the same procedure. Starting from these three fields, the simulations were carried out independently from each other using both UTOPIA and 6-ELAD with discriminator algorithms.

Given a fine-grid solution, we compute quadratic variance (both for vorticity and passive scalar) on that grid as well as the portion of the quadratic variance *resolved* by the coarser grid. To do so, first we apply a smoothing operator to the fine-grid solution,

$$\mathcal{S}[\zeta_{\{f\}} \to \zeta'_{\{f\}}]:$$

$$\begin{aligned}\zeta'_{i+1/4,j+1/4} = {} & \frac{1}{4}\zeta_{i+1/4,j+1/4} \\ & + \frac{1}{8}(\zeta_{i-1/4,j+1/4} + \zeta_{i+3/4,j+1/4} \\ & \quad + \zeta_{i+1/4,j-1/4} + \zeta_{i+1/4,j+3/4}) \\ & + \frac{1}{16}(\zeta_{i-1/4,j-1/4} + \zeta_{i+3/4,j-1/4} \\ & \quad + \zeta_{i-1/4,j+3/4} + \zeta_{i+3/4,j+3/4}), \end{aligned} \quad (50)$$

where indices $i, j \in \{f\}$ are half-integers (i, j = 1, 3/2, 2, 5/2, . . .) because they belong to the refined grid $\{f\}$. After that the smoothed field is projected to coarse grid by the cell-centered restriction operator,

$$\mathcal{R}[\zeta_{\{f\}} \to \zeta_{\{c\}}]:$$

$$\zeta_{i,j} = \frac{1}{4}(\zeta'_{i-1/4,j-1/4} + \zeta'_{i+1/4,j-1/4} + \zeta'_{i-1/4,j+1/4} + \zeta'_{i+1/4,j+1/4}), \quad (51)$$

which is adjoint to operator $\mathcal{P}$ defined by Eq. (49). In Eq. (51) the indices $i, j \in \{c\}$ are integer numbers because they belong to the coarse grid $\{c\}$. The quadratic variance is computed again, but now from field $\zeta_{\{c\}}$, consequently the portion of quadratic variance of field $\zeta_{\{f\}}$ is resolved by grid $\{c\}$. The smoothing step Eq. (50) removes fine structures that exist on grid $\{f\}$ but cannot be resolved on grid $\{c\}$. Not doing so will cause aliasing error. In principle, this smoothing may be applied several times, but we found that the results reported here are not very sensitive to it. There is also inherent smoothing in both the interpolation $\mathcal{P}$ and the restiction $\mathcal{R}$ operators themselves. In fact, one can verify that for any field $\zeta_{\{c\}}$ on grid $\{c\}$

$$\mathcal{R}[\mathcal{P}[\zeta_{\{c\}}]] = S[\zeta_{\{c\}}]. \quad (52)$$

The whole procedure may be repeated again in order to compute the portion of quadratic variance resolved by an even coarser grid. These quadratic variances are compared with variances computed from the solutions obtained directly on those grids.[4]

The results are presented in Fig. 17. Labels such as

[4] In the present study we avoid using discrete Fourier transforms as a diagnostic tool. The fine scales filtered out by the technique described above are associated with filamental vorticity and passive scalar in the background field outside the coherent structures. When successive coarsening is applied, this background field becames more and more flat, while the steepness of vorticity gradients on the edges of the coherent structures remains unchanged, if it can be resolved by the coarse grid, or restricted consistently with the resolution of that grid, if it cannot. If, as an alternative analysis, one tries to apply spectral filtering by eliminating Fourier components above some cutoff wavenumber, a spurious wavy pattern appears in the background if the field is transformed back to physical space. This wavy pattern is associated with spectral ''leakage'' from the steep gradients on the edges of coherent structures.

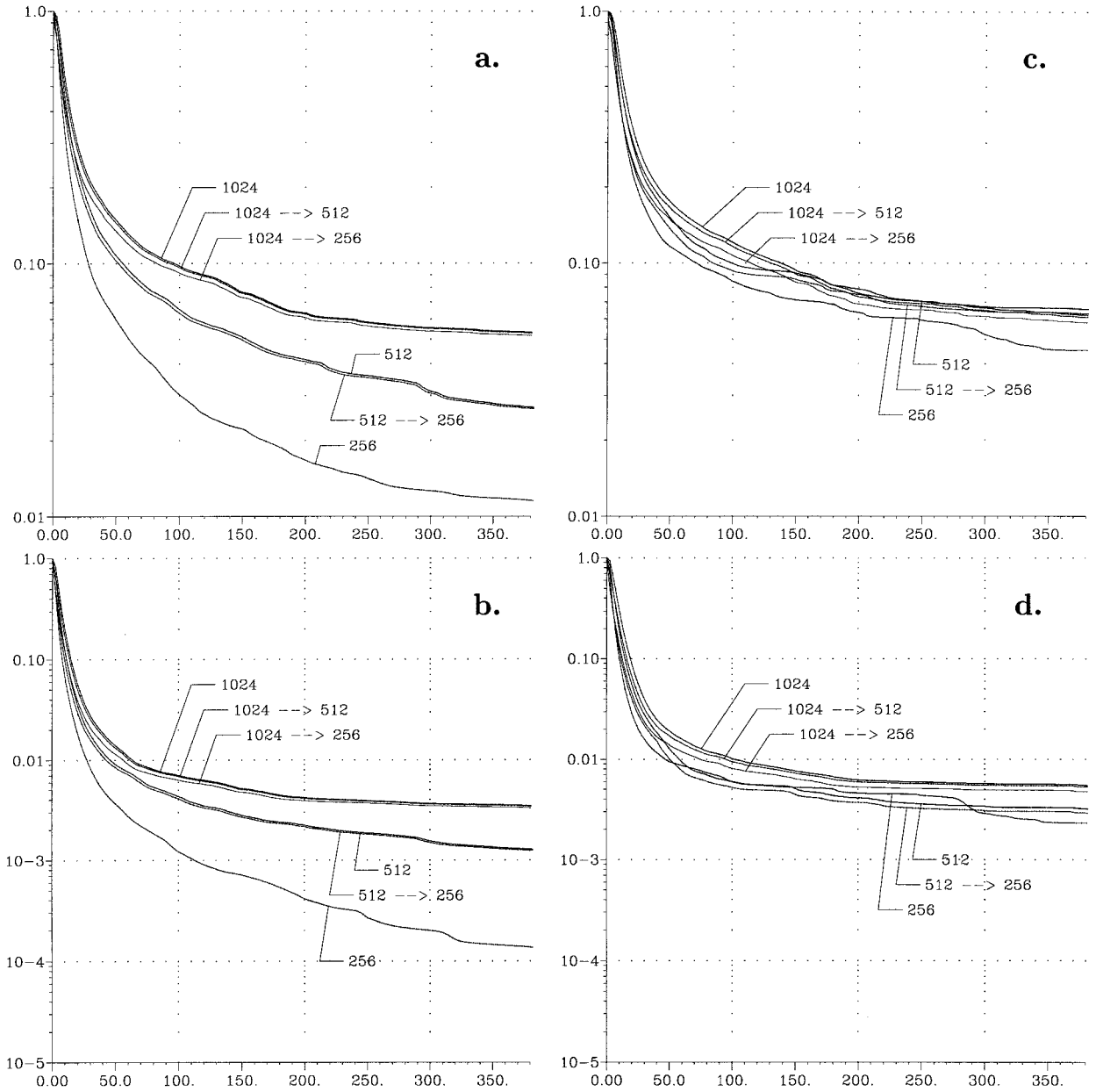

FIG. 17. Time history of resolved quadratic moments for turbulent flow similations for different grid resolutions. (a) Enstrophy, UTOPIA; (b) passive scalar variance, UTOPIA; (c) enstrophy, 6-ELAD with discriminator; (d) passive scalar variance, 6-ELAD with discriminator; The first number of 256, 512, and 1024 indicates resolution of the grid on which the simulation is performed; the second number, $\rightarrow$ 512 or $\rightarrow$ 256, if any, resolution of the grid to which the solution was coarsened before the second moment is computed.

$1024 \rightarrow 256$ indicate that the simulation is performed on grid 1024×1024, then fields are coarsened to grid 256×256 (in this particular case twice) in order to compute the variance. If the curve is labeled by a single number, the variance is computed on the same grid as the solution. The results in Fig. 17 suggest that for the solution produced by 6-ELAD scheme with a discriminator the behavior of the coarsened enstrophy and quadratic variance of passive scalar is reasonably close to that computed directly on the same grid as the solution. It should be stressed that for this class of problems there is no convergence in a *deterministic* sense. The initial states for all these problems are 100% correlated across all three grid resolutions; however, we observe that they become almost completely uncorrelated after a time of roughly 20–35, which is less than 10% of the duration of the whole simulation. There is no indication that the coarsened variance falls below that computed from the coarser-grid solution, except, perhaps, for a few brief fluctuations of enstrophy and quadratic variance from

the solution on the 256 × 256 grid associated with discrete vortex merger events. This verifies that despite smaller overall dissipation relative to the other methods, 6-ELAD does not produce underdissipated solutions. At the same time, the closeness of the curves labeled 1024 → 512 and 512 (especially for enstrophy dissipation) suggests that the solutions on these two grids are reasonably well converged in a sense of similarity of *statistical* behavior. This convergence is somewhat less successful for the passive scalar, which exhibits a larger overall dissipation.

UTOPIA, Figs. 17a,b, produces an excessive amount of dissipation. The variances of fine-grid solutions are less sensitive to coarsening than those produced by ELAD, simply because they are smoother on the grid scale. At the same time, the differences between different grid resolutions are much larger, and there is no indication that the rate of dissipation is close to the converged state.

We conclude that algorithms with adaptive dissipation, and expecially those with discriminators, help resist numerical aging and increase the effective resolution on a given computational grid. They also give at least an indication of solution convergence in terms of long-time vortex statistical behavior to a greater degree than the solutions with dissipation of weak type. Without being able to calculate solutions with even more initial vortices on even finer grids, we are not yet prepared to discuss the asymptotically true vortex population statistics. Nevertheless, any further tests of the theoretical predictions of Carnevale et al. (1991), Weiss and McWilliams (1993), and Bartello and Warn (1996) should be made with adequate attention to possible numerical aging biases.

8. Comparison of computational costs

All computations presented in this paper were performed on an SGI Power Challenge with a 90-MHz R8000 processor. This is a pipelined processor, which requires codes similar to those for vector machines. The available software allows direct estimation of the computational cost for every DO loop in the code, thus not only is timing possible for the complete algorithm but also for the distribution throughout the different steps. It is natural to express computational costs in CPU clock cycles per grid point. All codes were manually tuned to the level of at least 90% of maximum theoretical performance, and a special effort was made to ensure optimal pipelining of all logical switches.

The computational costs of Jacobians of different orders Eq. (42) are the following: second order, 21 clock cycles, where 12 are spent to compute terms associated with normal and 9 with diagonal fluxes; fourth, 36 = 20 + 16, respectively; sixth, 48 = 27 + 21. These schemes are computed with uniform orders of accuracy, that is, both interpolations and differencing are done with the same order, Eqs. (19)–(16). Increasing the order of accuracy causes only a moderate increase of cost, despite the use of a larger grid stencil. Thus, the fourth-order Jacobian uses a $7 \times 7 = 49$ point stencil with all ψ and ζ points involved, while the sixth-order scheme has a $9 \times 9 = 81$ point stencil. It is only slightly more than twice as expensive in comparison with the simple second-order Arakawa Jacobian, which has only a nine-point stencil. With only a small degradation of numerical performance, the higher-order schemes can be simplified by using second-order, staggered differencing for interpolation along with Eq. (21), and the computational cost will be decreased by 10% and 15%.

The ELAD costs are the following: step 1 + iter(step 2 + step 3) = 14 + iter(7/2 + 5), where iter is the number of diffusive iterations. In all computations presented in this paper we chose iter = 4, resulting in 48 clock cycles for the whole procedure. Thus, sixth-, fourth-, and second-order ELAD schemes require 96, 84, and 69 clock cycles, respectively.

The FCT algorithm with simplified permissible bounds requires 83 cycles, in which 25 are spent to compute both low- and high-order fluxes and their divergences and 58 are spent for the flux-limiting procedure. The extremum discriminator from Zalesak (1979) needs an extra 49 cycles, which results in an overall cost of 118 clock cycles. The new extremum discriminator with the six-point convexity check in each direction requires 22 clock cycles.

Rasch's UTOPIA algorithm costs 49 clock cycles per grid point, while the γ scheme (44) needs only 42. These costs should be doubled because of the use of a predictor–corrector step.[5,6]

We can compare these costs with that of the needed multigrid Poisson solver, which is approximately 90 clock cycles per grid point. Thus, all of these advection–dissipation operators have a similar cost.

In general, adaptive algorithms are more expensive and the flux-limiting procedure itself is the largest contributor to the cost. This encourages the use of the highest possible order of accuracy for the basic advection scheme, since the use of a second- instead of a fourth-order scheme brings only minor savings while dramatically degrading performance.

The traditional strategy for improving the numerical quality of a solution merely by grid refinement is not cost-effective compared to an algorithmic improvement. Each doubling of the grid resolution for a two-dimensional problem increases the computational effort by a factor of 8 (4 because of the increased number of grid points and 2 because of the time-step restriction of the

[5] In the case of passive-scalar advection where velocities are known, there is no need for a predictor–corrector step for UTOPIA. However, our experience is that it is needed for advection of vorticity.

[6] A useful simplification for the γ-scheme is to drop γ terms during the LF predictor step, while doubling their weights during the corrector. This results in a 69 clock-cycle scheme instead of 84.

CFL criterion for an explicit scheme). In contrast, the increase in computational cost per time step and grid point due to replacing the STD algorithm (with the necessary addition of explicit hyperdiffusion) with any of the algorithms studied in this paper is only about a factor of 3. Thus, it is generally more attractive to choose a more sophisticated technique instead of increasing the grid resolution.

9. Discussion

Many computational algorithms commonly used for advection and dissipation have solutions with either excessive nonmonotone behavior (numerical dispersion error) or excessive smoothing and dissipation, depending upon the choice of the diffusion or hyperdiffusion coefficient. We have developed a quasi-monotone advection algorithm, which consists of a high-order, integrally conservative and nondissipative, advection scheme followed by a locally adaptive corrective procedure, ELAD. This procedure detects grid points where the values of the advected scalar were overshot during the advection step, calculates the overshooting excess, and iteratively diffuses it away. Although the combination of low-order dissipation with high-order advection has been used previously [e.g., Forester (1977); Hyman (1979); also see Woodward and Colella (1984) for a review], our algorithm is unlike its predecessors because the correction is applied only to the excess field, not the total advected scalar. The proposed procedure may also be viewed as an inversion of the FCT design strategy. The multidimensional, antidiffusive flux-limiting procedure of the FCT algorithm is performed in a safe mode: discrete monotonicity is guaranteed during every step, while the high-order solution is approached ''from below'' by correcting a low-order predction. In contrast, ELAD starts from a high-order, nonmonotone solution and iteratively approaches monotone behavior ''from above.''

We have tested ELAD in one and two space dimensions. The tests include scalar advection leading to shape-invariant, spatially uniform translation and rotation, as well as cascading, dissipative two-dimensional turbulence. In these tests we have also included representatives from several classes of alternative algorithms that also seek a balance among small dissipation, shape preservation, and monotone behavior.

Existing algorithms provide two essentially different mechanisms for numerical dissipation: *weak* type, which occurs through a hyperdiffusively dominant truncation error, usually proportional to fourth- or higher even-order spatial derivative of the advected quantity; and *strong* type, which locally corrects nonmonotone behavior in a high-order advective prediction with a flux-limiting procedure. Semi-Lagrangian and all forward-in-time, upstream-biased schemes unavoidably have weak-type dissipation, with the option of a subsequent strong-type correction; we have used UTOPIA as a representative of this class. Alternatively, symmetric-in-space, three-time-level methods may be used as basic schemes to construct algorithms with dissipation of purely strong type; FCT and ELAD are in this class. The overall dissipation of FCT and ELAD depends on the dispersive properties of the basic scheme, which encourages the use of high-order advection. We have also included explicit hyperdiffusion and the γ-scheme as other algorithms with weak-type dissipation.

In all the tests, ELAD performs at least as well as the alternative algorithms. Compared to ELAD, the algorithms with weak-type dissipation escape less well the problem of compromising unsatisfactorily between low dissipation and montone behavior. The behavior of ELAD in the vicinity of strong gradients in the advected quantity is closest to that of FCT; however, it produces significantly less dissipation than FCT, especially in energy. They both appear to introduce a long-time shape bias toward flat-top extrema, hence excessive decay of extremum amplitudes, but this bias can be countered by use of extremum discriminators based upon a refined definition of overshooting. We have demonstrated a gain in effective resolution for methods with strong-type dissipation in comparison with weak-type dissipation. Although the resultant algorithms are more sophisticated and more expensive per grid point and time step, the extra computational cost is offset by a gain in solution quality that cannot be achieved at a similar cost simply by refinement of the grid resolution.

Our experience with turbulent flows shows that algorithms with dissipation of only strong type can absorb the nonlinear cascade near the grid scale, where advective dynamics can no longer be accurately calculated, and thus provide an adequate, albeit minimal, subgrid-scale model for maintaining smoothness of the solution. This supports the views advocated by Boris (1989) (Boris et al. 1992; Beets and Koren 1996):

1) Adaptive monotone algorithms themselves can provide a sufficient parametrization for the subgrid-scale transfer and dissipation processes.
2) Any such subgrid-scale model should be constructed by considering not only the unresolved, continuously differentiable physical processes, but also the discretized model properties.

These two principles are justified even if the discrete dissipative mechanism cannot be expressed as representing any particular continuous operator, always remembering that the important standard of solution convergence is to molecular dissipation at some large but finite Reynolds number.

Although we restricted ourselves to two-dimensional flows, the generalization of ELAD is straightforward to three-dimensional, incompressible flows for scalars conserved on Largrangian parcels. This makes it potentially quite useful for oceanic and atmospheric models, although further consideration is needed for the appropriate boundary conditions. If the advective equation involves more than Lagrangian conservation, then the simpler γ-scheme may still provide benefits compared to alternative methods.

Acknowledgments. This research was supported by the National Science Foundation Grant ECS 9217394 and Office of Naval Research Grant ONR-N00014-95-1-0316.

APPENDIX A

FCT Algorithm

We follow the methodology of the original FCT algorithm (Zalesak 1979). For brevity of notation we restrict our discussion to the one-dimensional case. Generalization to multidimensions is self-evident, provided that the components of velocity and fluxes are placed on fully staggered grids.

Consider the three-time-level FCT algorithm. The low-order upstream fluxes, high-order fluxes, and antidiffusive fluxes are defined as

$$\mathrm{FLX}^{n-1}_{i+1/2} = \begin{cases} q^{n-1}_i u_{i+1/2} & u_{i+1/2} > 0 \\ q^{n-1}_{i+1} u_{i+1/2} & u_{i+1/2} < 0, \end{cases} \tag{A1}$$

$$\mathrm{FHX}^{n}_{i+1/2} = \overline{q}^{\,xn}_{i+1/2} u_{i+1/2}, \tag{A2}$$

$$\mathrm{AX}_{i+1/2} = \mathrm{FHX}^{n}_{i+1/2} - \mathrm{FLX}^{n-1}_{i+1/2}, \tag{A3}$$

where $\overline{q}^{x}$ is computed using one of the interpolation formulas (18)–(21).

Then, using the low-order fluxes, compute preliminary values for q^{n+1},

$$q^*_i = q^{n-1}_i - \frac{2\Delta t}{\Delta x}(\mathrm{FLX}^{n-1}_{i+1/2} - \mathrm{FLX}^{n-1}_{i-1/2}), \tag{A4}$$

and, after the limiting functions CX are computed, the antidiffusive fluxes are added to this preliminary field to complete the computation of the field q at the new time step,

$$q^{n+1}_i = q^*_i - \frac{2\Delta t}{\Delta x}(\mathrm{CX}_{i+1/2}\mathrm{AX}_{i+1/2} - \mathrm{CX}_{i-1/2}\mathrm{AX}_{i-1/2}). \tag{A5}$$

Flux-limiting functions CX, $0 \le \mathrm{CX}_{i+1/2} \le 1$ are computed by Eqs. (7)–(13) from Zalesak (1979) [Eqs. (7′)–(13′) are to be used in multidimensions], so there is no need to repeat these formulas here. Unlike in the original algorithm, we do not reset the antidiffusive fluxes according to Eqs. (14) or (14′), and we use a slightly different definition for the permissible bounds for q^{n+1}_i, which will be discussed below.

If all CX in Eq. (A5) are set to 1, the algorithm becomes equivalent to its basic scheme, which is leapfrog (LF) time step. As long as the high-order fluxes are computed by a symmetric interpolation scheme, this basic scheme is stable and nondissipative.

The FCT algorithm (A1)–(A5) behaves reasonably well, however its major disadvantage is the time lag in computation of the low-order fluxes, which may result in some distortion of the shape of advected profiles (see Fig. A1, cases with prefix LF). Also, in our experience, it is sensitive to the LF computational mode, which requires the use of a corrective step every few time steps.

As an alternative to Eqs. (A1), (A3), (A4), and (A5) one can choose

$$\mathrm{FLX}^{n}_{i+1/2} = \begin{cases} q^{n}_i u_{i+1/2}, & u_{i+1/2} > 0 \\ q^{n}_{i+1} u_{i+1/2}, & u_{i+1/2} < 0, \end{cases} \tag{A6}$$

$$\mathrm{AX}_{i+1/2} = \mathrm{FHX}^{n}_{i+1/2} - \mathrm{FLX}^{n}_{i+1/2}, \tag{A7}$$

$$q^*_i = q^n_i - \frac{\Delta t}{\Delta x}(\mathrm{FLX}^{n}_{i+1/2} - \mathrm{FLX}^{n}_{i-1/2}), \tag{A8}$$

$$q^{n+1}_i = q^*_i - \frac{\Delta t}{\Delta x}(\mathrm{CX}_{i+1/2}\mathrm{AX}_{i+1/2} - \mathrm{CX}_{i-1/2}\mathrm{AX}_{i-1/2}), \tag{A9}$$

which makes the basic scheme unstable when all flux-limiting functions are set to 1. Though the presence of flux-limiting restricts the growth of instability, and the resultant algorithm has TVD, ENO, and positive-definiteness properties, it exhibits strong ''staircasing'' tendency, which diminishes its practical usefulness (see Fig. A1, cases labeled with prefix FW-).

A sensible choice is to use predictor–corrector time stepping (LF–trapezoidal rule), where both stages are FCT procedures. In practice, however, one can simplify the algorithm, avoiding the repetition of the flux-limiting procedure, thus, allowing only a mild increase of the computational cost in comparison with FCT algorithm (A1)–(A5).

At first, using the unlimited high-order fluxes and LF step, predict the new time step values,

$$q^{**}_i = q^{n-1}_i - \frac{2\Delta t}{\Delta x}(\mathrm{FHX}^{n}_{i+1/2} - \mathrm{FHX}^{n}_{i-1/2}), \tag{A10}$$

then interpolate q_i in time, recompute the high-order fluxes and compute the antidiffusive fluxes,

$$q^{***}_i = \left(\frac{1}{2} - \gamma\right) q^{**}_i + \left(\frac{1}{2} + 2\gamma\right) q^{n}_i - \gamma q^{n-1}_i, \tag{A11}$$

$$\mathrm{FHX}^{***}_{i+1/2} = \overline{q}^{\,x***}_{i+1/2}, \tag{A12}$$

$$\mathrm{AX}_{i+1/2} = \mathrm{FHX}^{***}_{i+1/2} - \mathrm{FLX}^{n}_{i+1/2}. \tag{A13}$$

Because q^{**}_i and q^{***}_i are needed exclusively to compute the high-order and antidiffusive fluxes, their values may be overshot. This preliminary step does not have to be an FCT procedure, and generally speaking, it even does not have to be conservative. Equations (A10) and (A11) may be combined into

$$q^{***}_i = \left(\frac{1}{2} + 2\gamma\right) q^{n}_i + \left(\frac{1}{2} - 2\gamma\right) q^{n-1}_i - \left(\frac{1}{2} - \gamma\right)\frac{2\Delta t}{\Delta x}(\mathrm{FHX}^{n}_{i+1/2} - \mathrm{FHX}^{n}_{i-1/2}), \tag{A14}$$

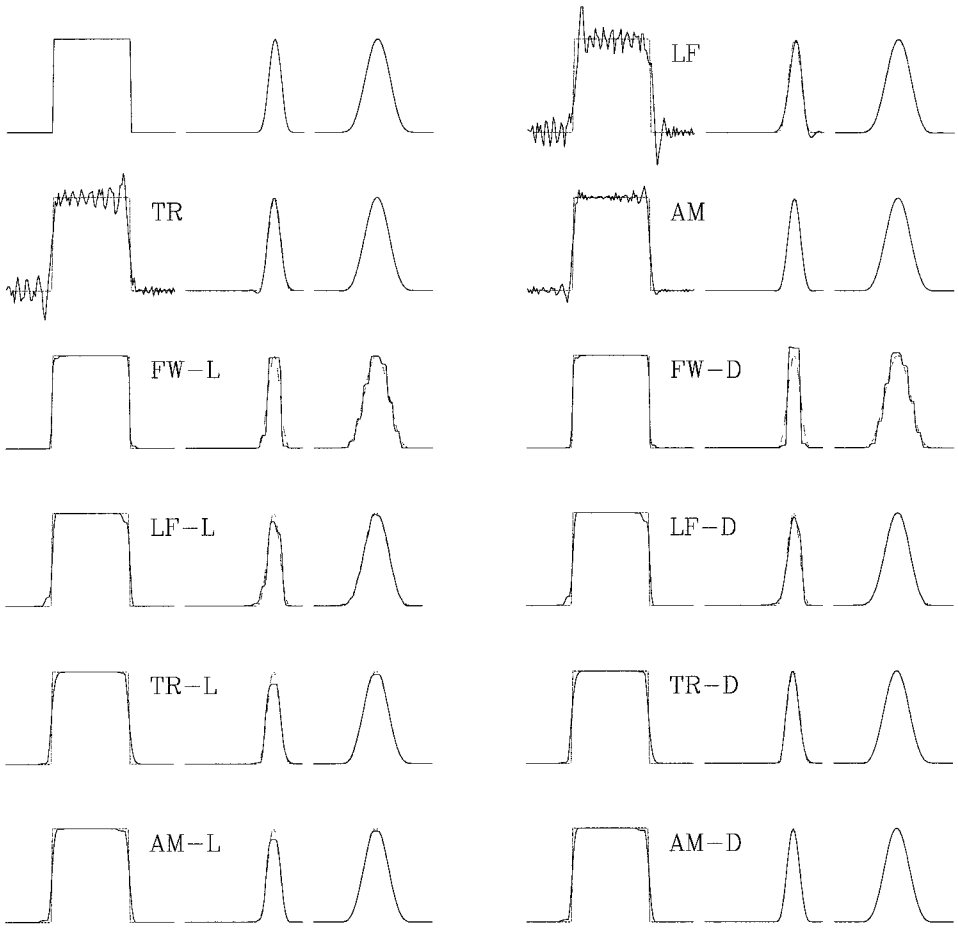

FIG. A1. Comparison of various options of FCT algorithm. The initial conditions, top left, are the same as in Fig. 1. Labels FW, LF, TR, and AM correspond to forward Euler, LF, LF–trapezoidal rule and Adams–Multon time steps for the basic scheme, which is sixth-order centered in space for the all cases presented here. To make the differences between the different time steps more transparrent, the Courant number is set to 1/3. Suffix -L labels cases when flux limiting is enabled, while -D indicates activation of both flux-limiting and extrema discrimination. Note the location of the dominant dispersive spikes near the step in the three cases when flux-limiting is turned off. Forward scheme is ustable and, therefore, was not tested in this regime. When flux limiting is enabled it produces the sharpest resolution of the step function (due to interplay between instability and flux limiting); however, smooth profiles are heavily damaged by staircasing. Also note distortion of shape of the narrow peak profile for the LF-L and LF-D cases, caused by time lag of the diffusive fluxes.

where there is no need to keep q^{n-1} after q^{***} is computed. (In practice fields q^{n-1}, q^{n+1}, q^{*}, q^{**}, and q^{***} share the same storage array.)

Once the antidiffusive fluxes are computed, FCT procedure given by Eqs. (A8)–(A9) is used to compute q_i^{n+1}. The choice of $\gamma = 1/12$ makes the basic scheme equivalent centered in space, third-order in time Adams–Multon step, which is stable and has excellent dispersive properties (see Fig. A1, cases labeled with AM). Setting $\gamma = 0$ makes the corrective step equivalent to trapezoidal rule. It is also stable, but it causes mild phase delay for the high-wavenumber Fourier components (see patterns labeled TR in Fig. 17).

a. Limiter and discriminator

Let $\hat{q}_{i+1/2}$ be the value of q at nearest point upstream from velocity point $u_{i+1/2}$,

$$\hat{q}_{i+1/2} = \begin{cases} q_i^n, & \text{if } u_{i+1/2} > 0 \\ q_{i+1}^n, & \text{if } u_{i+1/2} < 0, \end{cases} \tag{A15}$$

(in practice this operation is performed at virtually no cost because it is combined with computation of the upstream fluxes). After this step, the permissible bounds for q_i^{n+1} are defined as

$$\min(\hat{q}_{i-1/2}, \hat{q}_{i+1/2}) \le q_i^{n+1} \le \max(\hat{q}_{i-1/2}, \hat{q}_{i+1/2}), \tag{A16}$$

which is more restrictive than in Zalesak (1979) and Smolarkiewicz and Grabovski (1990), where the bounds are defined simply as the minimum and the maximum values among all nearest surrounding points, including downstream points. Now the downstream points are excluded. In the two-dimensional case, similar to (A15), both $\hat{q}_{i+1/2,j}$ and $\hat{q}_{i,j+1/2}$ are defined at the same points as the velocity components $u_{i+1/2,j}$ and $v_{i,j+1/2}$. Then Eq. (A16) becomes

$$\min(\hat{q}_{i-1/2,j}, \hat{q}_{i+1/2,j}, \hat{q}_{i,j-1/2}, \hat{q}_{i,j+1/2}) \le q_{i,j}^{n+1} \le \max(\hat{q}_{i-1/2,j}, \hat{q}_{i+1/2,j}, \hat{q}_{i,j-1/2}, \hat{q}_{i,j+1/2}). \quad \text{(A17)}$$

Because the flow is nondivergent, at least one of four surronding velocity components is directed away from point i, j; therefore $q_{i,j}^n$ is automatically included into the arguments of min and max.

The permissible bounds described above do not allow a possibility of existence of an extrema between grid points, thus ''clipping'' of narrow extrema is the result. To avoid clipping, we design an extremum discriminator, which is described below.

As a preliminary step, compute the discrete analog of second derivative of field q,

$$\text{qxx}_i = q_{i-1}^n - 2q_i^n + q_{i+1}^n, \quad \text{(A18)}$$

then, the criterion of the existence of an extremum between points i and $i + 1$, without specifying its type (minimum or maximum) is formulated as follows,

$$\left.\begin{array}{l}\text{if } (q_i^n - q_{i-1}^n)(q_{i+2}^n - q_{i+1}^n) < 0 \\ \text{and } \text{qxx}_i \text{qxx}_{i+1} > 0\end{array}\right\} \Rightarrow \begin{cases}\text{there is an extre-} \\ \text{mum between } i \\ \text{and } i + 1,\end{cases}$$

and the location of the extremum and the extremal value may be computed, for example, as the point of intersection of straight lines going trough pairs of points q_{i-1}, q_i and q_{i+1}, q_{i+2} [see Fig. 8 in Zalesak (1979)]:

$$x^{(\text{peak})} = \Delta x\left(i + \frac{1}{2} - \frac{1}{2}\frac{\text{qxx}_{i+1} - \text{qxx}_i}{\text{qxx}_{i+1} + \text{qxx}_i}\right),$$

$$q_{i+1/2}^{(\text{peak})} = \frac{1}{2}(q_{i+1}^n + q_i^n) + \frac{\frac{1}{2}(q_{i+1}^n - q_i^n)(\text{qxx}_{i+1} - \text{qxx}_i) - \text{qxx}_{i+1}\text{qxx}_i}{\text{qxx}_{i+1} + \text{qxx}_i}. \quad \text{(A19)}$$

The intersection point $x^{(\text{peak})}$ is within x_i and x_{i+1} as long as qxx_i and qxx_{i+1} are of the same sign; this point is an extremum if the first derivative changes sign; thus the logical conditions above are the *necessary* conditions for the existence of an extremum between x_i and x_{i+1}.

In practice neither of Eqs. (A19) is used. Instead, the permissible bounds are reset to $\pm\infty$ if the extremum is detected. It turned out to be more important to impose a more restrictive criterion for acceptance of the extremum in order to exclude a possibility of treatment of dispersion induced oscillations as legitimate physical extrema. Thus, an extremum is accepted if the second derivative retains its sign in at least four consequent grid points around the point $i + 1/2$. To do so, we assign

$$q_{i+1/2}^{(\min)} = \hat{q}_{i+1/2} \qquad q_{i+1/2}^{(\max)} = \hat{q}_{i+1/2} \qquad \forall i, \quad \text{(A20)}$$

and then reset these values, if an extremum is detected,

$$\left.\begin{array}{ll}\text{if} & (q_i^n - q_{i-1}^n)(q_{i+2}^n - q_{i+1}^n) < 0 \\ \text{and} & \text{qxx}_i \text{qxx}_{i+1} > 0 \\ \text{and} & \text{qxx}_{i-1}\text{qxx}_i > 0 \\ \text{and} & \text{qxx}_{i+1}\text{qxx}_{i+2} > 0\end{array}\right\} \Rightarrow \begin{cases} q_{i+1/2}^{(\min)} = -\infty \\ q_{i+1/2}^{(\max)} = +\infty, \end{cases} \quad \text{(A21)}$$

finally, as an alternative to Eq. (A16),

$$\min(q_{i-1/2}^{(\min)}, q_{i+1/2}^{(\min)}) \le q_i^{n+1} \le \max(q_{i-1/2}^{(\max)}, q_{i+1/2}^{(\max)}). \quad \text{(A22)}$$

Similarly, in two dimensions,

$$\min(q_{i-1/2,j}^{(\min)}, q_{i+1/2,j}^{(\min)}, q_{i,j-1/2}^{(\min)}, q_{i,j+1/2}^{(\min)}) \le q_{i,j}^{n+1} \le \max(q_{i-1/2,j}^{(\max)}, q_{i+1/2,j}^{(\max)}, q_{i,j-1/2}^{(\max)}, q_{i,j+1/2}^{(\max)}), \quad \text{(A23)}$$

where $q_{i+1/2,j}^{(\min)}$, $q_{i+1/2,j}^{(\max)}$, $q_{i,j+1/2}^{(\min)}$, and $q_{i,j+1/2}^{(\max)}$ are defined in a manner similar to Eqs. (A20)–(A21), provided that the discrete analog of second derivatives $\text{qyy}_{i,j}$ is computed as a preliminary step similar to Eq. (A18). By its design, the discriminator (A21) is a purely logical construction. Unlike in Leonard and Niknafs (1991), no prescribed threshold criterion is involved. At the same time, (A21) is more restrictive than discriminators used in Zalesak (1979) and Leonard and Niknafs (1991) because it requires satisfaction of the convexity check in a wider region around the extremum.

Equations (A15)–(A16) are used in cases labeled with suffix -L (limiter only) in Fig. A1, while Eqs. (A15), (A18), (A20), (A21), and (A22) are used in cases with -D (limiter and discriminator).

b. Uncompensated residual dissipation

Although by its design the FCT algorithm applies the antidiffusive fluxes with the maximum possible ''safe'' weight, there are causes for systematic underestimation of the antidiffusive fluxes. One such cause is due to Eq. (13) [or (13′)] in Zalesak (1979), where the limiting functions are chosen as mimima between the two adjacent grid points. This happens in both in one dimension and in multidimensions. The other, perhaps more important, cause occurs in multidimensions only. In the FCT flux-limiting algorithm the collective effect of antidiffusive fluxes at every grid cell is decomposed into sums of incoming and outgoing fluxes [Eqs. (7′) and (10′)]. In the case where there is a large velocity component in the direction perpendicular to the gradient of advected quantity, the antidiffusive fluxes associated with this velocity component are the dominant contributors to

both sums. If the sums are combined, these fluxes approximately cancel each other, so that the absolute value of each sum may be much larger than that of the combined effect. On the other hand, because the value of advected quantity is smooth along the direction of this component, these fluxes are not the major sources of the dispersive error, compared to the antidiffusive fluxes along the gradient. The latter ones will be restricted by the weighting coefficients governed predominantly by the stronger fluxes, which will result in underestimation of these coefficients.

More quantitative analysis of this effect is complicated because of highly anisotropic, nonlinear, and nonanalytical nature of the flux-limiting algorithm. Also, there is no obvious way to improve the flux-limiting part of the FCT algorithm by somehow relaxing the logical constraints in any of Eqs. (7′)–(13′) without causing overshoots. A potential way to eliminate the underestimation is to construct an *iterative* FCT procedure, similar to approach used in MPDATA (Smolarkiewicz and Grabovski 1990). This path, however, does not seem attractive because it implies the iterative repetition of the flux-limiting procedure, which is already the most computationally expensive part of the FCT algorithm. Because the residual dissipation is effectively low order (like Newtonian diffusion), it is mostly seen in the overall decay of energy, rather than enstrophy or quadratic variance of the passive scalar.

APPENDIX B

Details of Rasch's Algorithm

In this technical description we focus on implementation of the two-dimensional upstream switch, which, we believe, is the most efficient method on a pipelined or vector processor machine. Inquires about philosophy of the scheme and derivation should be addressed to Rasch (1994) and Leonard et al. (1995b). The algorithm implies that

$$q^{n+1}_{i,j} = q^n_{i,j} - \mathrm{FX}_{i+1/2,j} + \mathrm{FX}_{i-1/2,j} - \mathrm{FY}_{i,j+1/2} + \mathrm{FY}_{i,j-1/2}, \tag{B1}$$

where, in order to compute fluxes we compute first the spatial derivatives of the advected quantity,

$$\begin{aligned}
\mathrm{qx}_{i+1/2,j+1/2} &= q^n_{i+1,j+1} - q^n_{i,j+1} + q^n_{i+1,j} - q^n_{i,j}\\
\mathrm{qy}_{i+1/2,j+1/2} &= q^n_{i+1,j+1} - q^n_{i+1,j} + q^n_{i,j+1} - q^n_{i,j}\\
\mathrm{qxx}_{i,j} &= q^n_{i+1,j} - 2q^n_{i,j} + q^n_{i-1,j}\\
\mathrm{qyy}_{i,j} &= q^n_{i,j+1} - 2q^n_{i,j} + q^n_{i,j-1}\\
\mathrm{qxy}_{i+1/2,j+1/2} &= q^n_{i+1,j+1} - q^n_{i+1,j} - q^n_{i,j+1} - q^n_{i,j}.
\end{aligned} \tag{B2}$$

Define Courant numbers in both directions as

$$\hat{\mathrm{u}} = u\frac{\Delta t}{\Delta h}, \qquad \hat{\mathrm{v}} = v\frac{\Delta t}{\Delta h}, \tag{B3}$$

where $\Delta h = \Delta x = \Delta y$ is the grid spacing and Δt is the time step. If computed from streamfunction field ψ, the natural placements of the velocity components are $u_{i,j+1/2}$ and $v_{i+1/2,j}$, so that in order to compute fluxes, one needs to reinterpolate them to the locations $u_{i+1/2,j}$ and $v_{i,j+1/2}$. In practice we use a fourth-order accuracy interpolation scheme. In fact, all four of these fields are needed to compute fluxes FX and FY.

To compute flux component FX, first, at location $i + 1/2$, j we define intermediate quantities,

$$\begin{aligned}
\mathrm{CURVX}_{i+1/2,j} &= \begin{cases} \mathrm{qxx}_{i,j}, & \text{if } \hat{\mathrm{u}}_{i+1/2,j} > 0 \\ \mathrm{qxx}_{i+1,j}, & \text{if } \hat{\mathrm{u}}_{i+1/2,j} < 0 \end{cases}\\
\mathrm{CURVY}_{i+1/2,j} &= \begin{cases} \mathrm{qyy}_{i,j}, & \text{if } \hat{\mathrm{u}}_{i+1/2,j} > 0 \\ \mathrm{qyy}_{i+1,j}, & \text{if } \hat{\mathrm{u}}_{i+1/2,j} < 0 \end{cases}\\
\mathrm{GRADY}_{i+1/2,j} &= \begin{cases} \mathrm{qy}_{i+1/2,j-1/2}, & \text{if } \hat{\mathrm{v}}_{i+1/2,j} > 0 \\ \mathrm{qy}_{i+1/2,j+1/2}, & \text{if } \hat{\mathrm{v}}_{i+1/2,j} < 0 \end{cases}\\
\mathrm{TWIST}_{i+1/2,j} &= \begin{cases} \mathrm{qxy}_{i+1/2,j-1/2}, & \text{if } \hat{\mathrm{v}}_{i+1/2,j} > 0 \\ \mathrm{qxy}_{i+1/2,j+1/2}, & \text{if } \hat{\mathrm{v}}_{i+1/2,j} < 0, \end{cases}
\end{aligned} \tag{B4}$$

then compute the flux component itself,

$$\mathrm{FX}_{i+1/2,j} = \hat{\mathrm{u}}_{i+1/2,j}\left[\frac{q^n_{i,j} + q^n_{i+1,j}}{2} - \hat{\mathrm{u}}_{i+1/2,j}\frac{q^n_{i+1,j} - q^n_{i,j}}{2} - \frac{1 - \hat{\mathrm{u}}^2_{i+1/2,j}}{6}\mathrm{CURVX}_{i+1/2,j} - \frac{1}{4}\hat{\mathrm{v}}_{i+1/2,j}(\mathrm{GRADY}_{i+1/2,j} + (1 - \hat{\mathrm{v}}_{i+1/2,j})\mathrm{CURVY}_{i+1/2,j} - \hat{\mathrm{u}}_{i+1/2,j}\mathrm{TWIST}_{i+1/2,j})\right]. \tag{B5}$$

There is no need to store CURVX, CURVY, GRADY, and TWIST in scratch workspace arrays. However, to eliminate operations inside the upstream switches (B4) one needs to precompute and store in scratch arrays intermediate fields (B2). This feature of the algorithm is important for the efficiency when the vector or pipelined processor is used, because these kind of processors execute both versions inside the logical switch, then

accept one of the results and discard the other, thus doubling the cost of operations inside the switch. For the same reason we have rearranged operations (cf. Rasch 1994) in order to eliminate the use of nested switches. One can easily recognize CRAY implementation of the upstream switches [Eq. (B4)] via CRAY conditional vector merger functions CVMGM and CVMGP.

Flux component FY is computed in a similar way,

$$\mathrm{CURVX}_{i,j+1/2} = \begin{cases} \mathrm{qxx}_{i,j}, & \text{if } \hat{\mathrm{v}}_{i,j+1/2} > 0 \\ \mathrm{qxx}_{i,j+1}, & \text{if } \hat{\mathrm{v}}_{i,j+1/2} < 0 \end{cases}$$

$$\mathrm{CURVY}_{i,j+1/2} = \begin{cases} \mathrm{qyy}_{i,j}, & \text{if } \hat{\mathrm{v}}_{i,j+1/2} > 0 \\ \mathrm{qyy}_{i,j+1}, & \text{if } \hat{\mathrm{v}}_{i,j+1/2} < 0 \end{cases}$$

$$\mathrm{GRADX}_{i,j+1/2} = \begin{cases} \mathrm{qx}_{i-1/2,j+1/2}, & \text{if } \hat{\mathrm{u}}_{i,j+1/2} > 0 \\ \mathrm{qx}_{i+1/2,j+1/2}, & \text{if } \hat{\mathrm{u}}_{i,j+1/2} < 0 \end{cases}$$

$$\mathrm{TWIST}_{i,j+1/2} = \begin{cases} \mathrm{qxy}_{i-1/2,j+1/2}, & \text{if } \hat{\mathrm{u}}_{i,j+1/2} > 0 \\ \mathrm{qxy}_{i+1/2,j+1/2}, & \text{if } \hat{\mathrm{u}}_{i,j+1/2} < 0 \end{cases} \quad \text{(B6)}$$

and

$$\mathrm{FY}_{i,j+1/2} = \hat{\mathrm{v}}_{i,j+1/2}\left[\frac{q^n_{i,j} + q^n_{i,j+1}}{2} - \hat{\mathrm{v}}_{i,j+1/2}\frac{q^n_{i,j+1} - q^n_{i,j}}{2} - \frac{1 - \hat{\mathrm{v}}^2_{i,j+1/2}}{6}\mathrm{CURVY}_{i,j+1/2} - \frac{1}{4}\hat{\mathrm{u}}_{i,j+1/2}(\mathrm{GRADX}_{i,j+1/2} + (1 - \hat{\mathrm{u}}_{i,j+1/2})\mathrm{CURVX}_{i,j+1/2} - \hat{\mathrm{v}}_{i,j+1/2}\mathrm{TWIST}_{i,j+1/2})\right]. \quad \text{(B7)}$$

To ensure second-order accuracy in time, a predictor–corrector time step is used. At first, the streamfunction ψ is extrapolated to time level $n + 1/2$,

$$\psi^{n+1/2} = \frac{3}{2}\psi^n - \frac{1}{2}\psi^{n-1}. \quad \text{(B8)}$$

Then velocity components are computed and the vorticity field is advanced in time according to Eq. (B1). After the new time step streamfunction ψ^{n+1} is obtained by resolving the elliptic problem, it is interpolated between the time steps n and $n + 1$, and velocities at half time step $n + 1/2$ and vorticity at new time step $n + 1$ are recomputed again. Finally, computation of the new time streamfunction completes the time step.

APPENDIX C

Algorithm for Identification of Extrema

Suppose fields ζ, q, and ψ are known at two moments of time, typically 10–20 time steps apart. First, we analyze both old and new time fields ζ and q for the presence of extrema. A value at point $\times$ is accepted as an extremum if it dominates over its 20 neighbors $+$,

$$\begin{array}{ccccc} & + & + & + & \\ + & + & + & + & + \\ + & + & \times & + & +. \\ + & + & + & + & + \\ & + & + & + & \end{array} \quad \text{(C1)}$$

For each field, minima and maxima are sorted by amplitude and stored along with indices i, j. At the old time at every extremum we initialize a Lagrangian particle and trace it to the new time step. To do so, first determine to which grid box $(i, i + 1) \times (j, j + 1)$ the particle belongs and interpolate the streamfunction between times in the surrounding 16 points. Then we determine fractional displacements of the particle from the center of the box, h_x, h_y, so that $-1/2 < h_x, h_y < 1/2$, $h_x, h_y = 0$ if the particle is located exactly in the center of the box and $h_x, h_y = \pm 1/2$ on the side. Using cubic interpolation and off-centered fourth-order differencing,

$$\left.\frac{\partial\psi}{\partial x}\right|_{i+1/2+h} = \alpha\psi_{i+2} + (\beta - \alpha)\psi_{i+1} + (\gamma - \beta)\psi_i - \gamma\psi_{i-1}, \quad \text{(C2)}$$

where

$$\begin{aligned} \alpha &= -\frac{1}{24} + \frac{h}{2} + \frac{h^2}{2} \\ \beta - \alpha &= +\frac{27}{24} - \frac{h}{2} - \frac{3h^2}{2} \\ \gamma - \beta &= -\frac{27}{24} - \frac{h}{2} + \frac{3h^2}{2} \\ -\gamma &= +\frac{1}{24} + \frac{h}{2} - \frac{h^2}{2}, \end{aligned} \quad \text{(C3)}$$

compute the velocitiy components at particle. Finally, advance the coordinate of the particle in time. Because in the vicinity of a strong extremum of vorticity Lagrangian particles tend to spin around, we choose an iterative backward Euler time step for this procedure, which ''locks'' the particle on the extremum.

After the location of the Lagrangian particle is determined at the new time, the new time list of extrema is searched for an extremum located no further than a

prescribed critical radius, which is chosen to be 1.5 of the grid size. If an extremum is found (positive identification), a line connecting its value at the old and new time is plotted.

In practice we use a 20-time-step interval between examining the extrema and the same time step as in the main model for the tracking the Lagrangian particles.

REFERENCES

Arakawa, A., 1966: Computational design for long term numerical integration of equations of fluid motion: Two-dimensional incompressible flow. Part. 1. *J. Comput. Phys.,* **1,** 119–143.

——, and V. R. Lamb, 1981: A potential enstrophy and energy conserving schemes for the shallow water equations. *Mon. Wea. Rev.,* **109,** 18–36.

——, and Y.-J. G. Hsu, 1990: Energy conserving and potential-enstrophy dissipating schemes for the shallow water equations. *Mon. Wea. Rev.,* **118,** 1960–1969.

Babiano, A., C. Basdevant, B. Legras, and R. Sadourny, 1987: Vorticity and passive-scalar dynamics in two-dimensional turbulence. *J. Fluid Mech.,* **183,** 379–397.

Bartello, P., and T. Warn, 1996: Self-similarity of decaying two-dimensional turbulence. *J. Fluid Mech.,* **326,** 357–372.

Beets, C., and B. Koren, 1996: Large-eddy simulation with accurate implicit subgrid-scale diffusion. Rep. NM-R9601, Centrum voor Wiskunde en Informatica, IMAU, Utrecht, the Netherlands, 23 pp.

Bennett, A. F., and D. Haidvogel, 1983: Low-resolution numerical simulation of decaying two-dimensional turbulence. *J. Atmos. Sci.,* **40,** 738–748.

——, and P. F. Cummins, 1988: Tracking fronts in solutions of the shallow water equations. *J. Geophys. Res.,* **93,** 1293–1301.

Boris, J. P., 1989: On large eddy simulation using subgrid turbulence models. Comment 1. *Lect. Notes,* **357,** 344–351.

——, and D. L. Book, 1973: Flux corrected transport. I: SHASTA, a fluid transport algorithm that works. *J. Comput. Phys.,* **11,** 38–39.

——, F. F. Grinstein, E. S. Oran, and R. L. Kolbe, 1992: New insights into large eddy simulation. *Fluid Dyn. Res.,* **10,** 199–228.

Bott, A., 1989: A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. *Mon. Wea. Rev.,* **117,** 1006–1015.

Böttcher, M., 1996: A semi-Lagrangian advection scheme with modified exponential splines. *Mon. Wea. Rev.,* **124,** 716–729.

Carnevale, G. F., J. C. McWilliams, Y. Pomeau, J. B. Weiss, and W. R. Young, 1991: Evolution of vortex statistics in two-dimensional turbulence. *Phys. Rev. Lett.,* **66,** 2735–2737.

Carpenter, R. L., K. K. Droegemeier, P. R. Woodward, and C. E. Hane, 1990: Application of the piecewise parabolic method (PPM) to meterological modeling. *Mon. Wea. Rev.,* **118,** 586–612.

Chlond, A., 1994: Locally modified version of Bott's advection scheme. *Mon. Wea. Rev.,* **122,** 111–125.

Colella, P., 1990: Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.,* **87,** 171–200.

——, and P. Woodward, 1984: The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.,* **54,** 174–201.

Crowley, W. P., 1968: Numerical advection experiments. *Mon. Wea. Rev.,* **96,** 1–12.

Davies, H. C., 1980: A pseudo-upstream differencing scheme for advection. *J. Comput. Phys.,* **37,** 280–286.

Easter, R. C., 1993: Two modified versions of Bott's positive definite numerical advection scheme. *Mon. Wea. Rev.,* **121,** 297–304.

Engquist, B., P. Lötstedt, and B. Sjögreen, 1989: Nonlinear filters for efficient schock computation, *J. Comput. Phys.,* **52,** 509–537.

Farrow, D. E., and D. P. Stevens, 1995: A new tracer advection scheme for Bryan and Cox type ocean general circulation models. *J. Phys. Oceanogr.,* **25,** 1731–1741.

Forester, C. K., 1977: Higher order monotonic convective difference schemes. *J. Comput. Phys.,* **23,** 1–22.

Godunov, S. K., 1959: Finite-difference method for the numerical computations of equations of gas dynamics. *Mat. Sb. (N. S.),* **7,** 271–290.

Harten, A., 1983: High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.,* **49,** 357–393.

Hecht, M., W. Holland, and P. Rasch, 1995: Upwind-weighted advection schemes for ocean tracer transport: An evaluation in a passive tracer context. *J. Geophys. Res.,* **100,** 20 763–20 778.

Hyman, J. M., 1979: A method of lines approach to the numerical solution of conservation laws. *Advances in Computational Methods for Partial Differential Equations,* R. Vichnevetsky and R. S. Stepleman, Eds., Vol. III, IMACS, 313–321.

Leonard, B. P., 1979: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Mech. Eng.,* **19,** 59–98.

——, and H. S. Niknafs, 1991: Sharp monotonic resolution of discontinuities without clipping of narrow extrema. *Comput. Fluids,* **19,** 141–154.

——, M. K. McVean, and A. P. Lock: 1993, Positivity preserving numerical schemes for multidimensional advection. NASA Tech. Memo. 106055 KOMP 93-05, Institute for Computational Mechanics in Propulsion, 62 pp.

——, A. P. Lock, and M. K. McVean, 1995a: The NIRVANA scheme applied to one-dimensional advection. *Int. J. Numer. Methods Heat Fluid Flow,* **5,** 341–377.

——, M. K. McVean, and A. P. Lock, 1995b: The flux integral method for multidimensional convection and diffusion. *Appl. Math. Model.,* **19,** 333–342.

——, A. P. Lock, and M. K. McVean, 1996: Conservative explicit unrestricted-time-step constancy-preserving advection schemes. *Mon. Wea. Rev.,* **124,** 2588–2606.

LeVeque, R. J., 1992: *Numerical Methods for Conservation Laws,* Birkhäuser, 214 pp.

Lilly, D. K., 1965: On the computational stability of the time-dependent non-linear geophysical fluid dynamics problem. *Mon. Wea. Rev.,* **93,** 11–26.

McCalpin, J. D., 1988: A quantitative analysis of the dissipation inherent in semi-Lagrangian advection. *Mon. Wea. Rev.,* **116,** 2330–2336.

McWilliams, J. C., 1984: The emergence of isolated, coherent vortices in turbulent flow. *J. Fluid Mech.,* **146,** 21–43.

——, 1990: The vortices of two-dimensional turbulence. *J. Fluid Mech.,* **219,** 361–385.

Odstrvcil, D., 1990: A new optimized FCT algorithm for shock wave problems. *J. Comput. Phys.,* **91,** 71–93.

Orszag, S. A., and L. W. Jayne, 1974: Local errors of difference approximations to hyperbolic equations. *J. Comput. Phys.,* **14,** 93–103.

Rasch, P. J., 1994: Conservative shape-preserving two-dimensional transport on a spherical reduced grid. *Mon. Wea. Rev.,* **122,** 1337–1350.

Raymond, W. H., 1994: Diffusion and numerical filters. *Mon. Wea. Rev.,* **122,** 757–761.

Sadourny, R., 1975: The dynamics of finite-difference models of the shallow-water equations. *J. Atmos. Sci.,* **32,** 680–689.

Shyy, W., M.-H. Chen, R Mittal, and H. S. Udaykumar, 1991: On the suppression of numerical oscillations using a nonlinear filter. *J. Comput. Phys.,* **102,** 49–62.

Smolarkiewicz, P. K., 1982: The multidimensional Crowley advection schemes. *Mon. Wea. Rev.,* **110,** 1967–1983.

——, 1983: A simple positive definite advection scheme with small implicit diffusion. *Mon. Wea. Rev.,* **111,** 479–486.

——, 1984: A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. *J. Comput. Phys.,* **54,** 325–362.

——, and T. L. Clark, 1986: The multidimensional positive definite

advection transport algorithm: Further development and applications. *J. Comput. Phys.*, **56,** 396–438.

——, and W. Grabovski, 1990: The multidimensional positive definite advection transport algorithm: Nonoscillatory option. *J. Comput. Phys.*, **86,** 355–375.

Staniforth, A., J. Côté, and J. Pudykiewicz, 1987: Comments on ''Smolarkiewicz's deformational flow.'' *Mon. Wea. Rev.*, **115,** 894–900.

Sweby, P. K., 1985: High resolution TVD schemes using flux limiters. *Lectures in Applied Mathematics*, Part 2, C. B. Vreugdenhil and B. Koren, Eds., Vieweg, 289–309.

Thuburn, J., 1995: Dissipation and cascade to small scales in numerical models using a shape-preserving advection scheme. *Mon. Wea. Rev.*, **123,** 1888–1903.

——, 1996: Multidimensional flux-limited advection scheme. *J. Comput. Phys.*, **123,** 74–83.

Tremback, C. J., J. Powell, W. R. Cotton, and R. A. Pielke, 1987: The forward-in-time upstream advection scheme: Extensions to higher orders. *Mon. Wea. Rev.*, **115,** 540–555.

van Leer, B., 1979: Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov's method. *J. Comput. Phys.*, **32,** 101–136.

——, 1985: Upwind-difference methods for aerodynamic problems governed by the Euler equations. *Lectures in Applied Mathematics*, Part 2, C. B. Vreugdenhil and B. Koren, Eds., Vieweg, 117–138.

Weiss, J. B., and J. C. McWilliams, 1993: Temporal scaling behavior of decaying two-dimensional turbulence. *Phys. Fluids A*, **5,** 608–621.

Weiyan, T., 1992: *Shallow Water Hydrodynamics*. Elsevier Science Publishers, 1992.

Woodward, P., and P. Colella, 1984: The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, **54,** 115–173.

Zalesak, S. T., 1979: Fully multidimensional flux corrected transport for fluids. *J. Comput. Phys.*, **31,** 335–362.