Journal of Computational Physics 259 (2014) 617–635

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A level set two-way wave equation approach for Eulerian interface tracking

Henri Samuel [a,b,c,*]

[a] CNRS, IRAP, 14, avenue Édouard Belin, F-31400 Toulouse, France
[b] Université de Toulouse, UPS-OMP, IRAP, Toulouse, France
[c] Bayerisches Geoinstitut, Universität Bayreuth, Germany

ARTICLE INFO

Article history:
Received 11 June 2013
Received in revised form 16 October 2013
Accepted 23 November 2013
Available online 1 December 2013

Keywords:
Advection
Two-way wave equation
Level set
Interface tracking

ABSTRACT

We present a new approach to perform Eulerian level set interface tracking. It consists in advecting the level set function using a second-order two-way wave equation instead of the standard one-way wave advection equation. The resulting numerical schemes are simple to implement, more stable, and significantly less prone to dissipation errors than popular (*e.g.*, WENO) discretizations of the one-way advection equation of higher order, in particular for long advection times. While both the two-way wave advection and the associated numerical schemes were derived previously, these approaches have never been combined with level set advection. Since the level set function to advect is smooth by construction, this ensures the stability of the solution when using the two-way advection equation discretized using centered finite difference schemes. Our numerical tests show that the two-way wave equation approach yields more accurate results than the standard one-way wave equation for the level set advection, at a considerably smaller computational cost.

1. Introduction

High-accuracy modeling of the advection of a scalar field, C, in an arbitrary external velocity field $\mathbf{u}$ is a fundamental requirement and a recurrent problem in computational fluid dynamics (*e.g.*, [19,15,22] and references therein). This process can be described in its simplest form by the following linear hyperbolic equation:

$$\partial_t C + \mathbf{u} \cdot \nabla C = 0, \tag{1}$$

where t is the time. The above equation is also known as the *one-way wave equation*. A special case of advection problems is the tracking of an interface Ω that marks the boundary between two fluid regions of arbitrary shape and size. The location of the interface may be defined according to the value of C (*e.g.*, $\Omega \equiv \mathbf{x}_c$ such that $C(\mathbf{x}_c) = c$). If C is discontinuous, solving blindly for Eq. (1) using Eulerian approaches can often be challenging because the presence of sharp variations in C yields the indetermination of partial derivatives appearing in the advection term across the discontinuities. A way around these difficulties is the level set approach that consists in replacing in Eq. (1) the sharply varying field C by a continuous and smoothly varying field ϕ [16]:

$$\partial_t \phi + \mathbf{u} \cdot \nabla \phi = 0. \tag{2}$$

* Correspondence to: CNRS, IRAP, 14, avenue Édouard Belin, F-31400 Toulouse, France
E-mail address: henri.samuel@irap.omp.eu.

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.11.027

The level set function, ϕ, is usually defined as the signed distance function to the interface that satisfies the Eikonal equation:

$$|\nabla\phi| = 1. \tag{3}$$

In general, the external velocity field $\mathbf{u}$ will not maintain ϕ as a signed distance function, therefore the above Eikonal requirement must be continuously enforced. Failing in doing so may result in the development of non-smooth variations in ϕ, yielding poorly accurate results when solving for Eq. (2). The Eikonal requirement can be enforced by solving for Eq. (3) using a fast marching method [19] or by solving a non-linear evolutionary equation to steady state [21]:

$$\partial_\tau\phi = S(\phi_0)\big(1 - |\nabla\phi|\big), \tag{4}$$

where ϕ_0 is the solution of Eq. (2) prior to reinitialization, τ represents a fictitious time, and $S(\phi_0)$ is a smoothed signed function:

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \varepsilon^2}}, \tag{5}$$

where ε is taken as the grid spacing (assuming a constant grid spacing). Alternatively, one may construct an *extension velocity* field $\mathbf{u}_{\text{ext}}$ that replaces $\mathbf{u}$ in Eq. (1) such that the Eikonal requirement is preserved [1,19]. However, the construction of $\mathbf{u}_{\text{ext}}$ is performed via the fast marching method, which reduces to low order (second at most) compared to the use of higher order (*e.g.*, fifth-order WENO) schemes to discretize the spatial derivatives in Eq. (4). Higher order fast marching methods exist [2] but are more difficult to implement. Since the level set function is smooth, the numerical errors associated with the advection step tend to be smaller than in the case of discontinuous/sharply varying quantities. Nevertheless, even when using high-order schemes such as fifth-order WENO reconstruction, dissipation errors remain present and constitute a main source of inaccuracy. Increasing the order of the advection schemes or using Lagrangian particles to correct for erroneous values of the level set [4,5] considerably reduces the numerical errors. However, such strategies are generally more complicated to implement and to parallelize efficiently. In addition, they yield a substantial extra cost, in particular when the interface spans a significant fraction of the computational domain [18]. It is therefore desirable to devise a level set method that improves the accuracy of the Eulerian advection step and its associated computational cost.

In this paper, we propose to use a second order, *two-way* wave equation instead of Eq. (1) to perform the advection step of the level set method. If the initial conditions are appropriately specified, the solutions of the standard one-way wave equation (Eq. (1)) and the two-way advection equation are identical. However, the discretization of the two-way advection equation yields significantly smaller dissipation and dispersion errors, provided that the function to advect is smooth. The paper is organized as follows: In Section 2 we summarize a few popular approaches to discretize the one-way advection equation (Eq. (1)). In Section 3 we present the two-way advection equation as an alternative to the one-way advection equation. In Section 4 we compare the accuracy of the solution of the one-way advection equation, and the two-way advection equation via theoretical analysis and simple linear advection tests. Section 5 shows the application of the two-way advection equation for level set interface tracking and the comparison with the results obtained using standard level set approach based on the one-way advection equation.

2. The one-way advection equation

In the following we briefly review a few popular approaches to solve the standard advection equation in Eulerian domains. We focus on the one-dimensional version of Eq. (1) along the rectilinear x-direction:

$$\partial_t C + u\partial_x C = 0. \tag{6}$$

Extension to multidimensional space is straightforward and will be presented in Section 2.2.

2.1. Spatial discretization

Assuming a divergence-free velocity field, Eq. (6) can be written in semi-discrete conservative flux difference form:

$$\frac{dC_j}{dt} + \frac{1}{\Delta x}(F_{j+1/2} - F_{j-1/2}) = 0, \tag{7}$$

where Δx is the grid spacing/cell length (assumed to be constant), j is the grid index along the x-direction that refers to the cell-center coordinate, $x_j = \Delta x j$, and $x_{j\pm1/2} = x_j \pm \Delta x/2$. $F_{j\pm1/2}$ is the numerical convective flux through the interface located between cell j and cell $j \pm 1$ at time t. For stability, the flux $F_{j\pm1/2}$ is upwind-biased according to the method of characteristics:

$$F_{j\pm1/2} = -\min\big[\text{sign}(u_{j\pm1/2}), 0\big]F^-_{j\pm1/2} + \max\big[\text{sign}(u_{j\pm1/2}), 0\big]F^+_{j\pm1/2}. \tag{8}$$

Within this frame, a variety of choices for the negative (left-going) $F^-_{j\pm1/2}$ and positive (right-going) fluxes $F^+_{j\pm1/2}$ exists (in the following, we adopt $+$ and $-$ exponents to refer to the right-going and left-going fluxes, respectively). The simplest stable choice is the first-order, one-sided flux:

$$F^-_{j+1/2} = u_{j+1/2} C_{j+1}, \qquad F^+_{j+1/2} = u_{j+1/2} C_j. \tag{9}$$

While monotone, the above first-order fluxes are very dissipative and significant accuracy can be gained by using higher order fluxes such as [3]:

$$F^\pm_{j+1/2} = \frac{u_{j+1/2}}{2}(C_{j+1} + C_j) - \frac{1}{2}\left[\left(1 - \varphi\left(\theta^\pm_{j+1/2}\right)\right)|u_{j+1/2}| + \frac{u^2_{j+1/2}\Delta t}{\Delta x}\varphi\left(\theta^\pm_{j+1/2}\right)\right](C_{j+1} - C_j), \tag{10}$$

where the flux limiter $\varphi_{j+1/2}$ is a function of $\theta^\pm_{j+1/2}$, the ratio of the upwind-biased slopes of C across the cell interface $j+1/2$:

$$\theta^-_{j+1/2} = \frac{C_{j+2} - C_{j+1}}{C_{j+1} - C_j}, \qquad \theta^+_{j+1/2} = \frac{C_j - C_{j-1}}{C_{j+1} - C_j}. \tag{11}$$

Among the various possibilities, popular choices of limiter functions are:

$$\varphi(\theta) = \begin{cases} \max[0, \min(1, \theta)] & \textit{Minmod}, \\ \max[0, \min(2\theta, 1), \min(\theta, 2)] & \textit{Superbee}, \\ \frac{\theta + |\theta|}{1 + |\theta|} & \textit{Van Leer}. \end{cases}$$

If C is smooth Eq. (10) is equivalent to a second-order *Lax–Wendroff* flux. Otherwise, the above expression reverts to the low-order monotone upwind flux ensuring a Total Variation Diminishing (TVD) scheme [8].

An even more accurate upwind-biased numerical flux can be obtained through the use of a third-order or higher ($L \geqslant 2$) Weighted Essentially Non-Oscillatory (WENO) reconstruction [13,17]:

$$F^\pm_{j+1/2} = u_{j+1/2} \sum_{l=1}^{L} w^\pm_l p^\pm_l, \tag{12}$$

which is a convex combination of the L possible approximations $p_{l=1,\dots,L}$ of C. The latter are determined by Newton polynomial interpolation of order L constructed using L different stencils: $I^+_{l=1,\dots,L} = \{x_{j+l-L}, x_{j+l-L+1}, \dots, x_{j+l-1}\}$, $I^-_{l=1,\dots,L} = \{x_{j+l-L+1}, x_{j+l-L+2}, \dots, x_{j+l}\}$. For the third-order reconstruction ($L = 2$) this yields the two following approximations:

$$p^\pm_1 = -\frac{\delta^\pm_{j-1}}{2} + \frac{3\delta^\pm_j}{2}, \qquad p^\pm_2 = +\frac{\delta^\pm_j}{2} + \frac{\delta^\pm_{j+1}}{2}, \tag{13}$$

where $\delta^\pm$ are upwind-biased C values taken at different grid locations:

$$\delta^-_{j+l, l=-L+1,\dots,L-1} = C_{j-l+1}, \qquad \delta^+_{j+l, l=-L+1,\dots,L-1} = C_{j+l}. \tag{14}$$

In the WENO approach the definition of the non-linear weights $w^\pm_l$ is based on the measure of the smoothness $S^\pm_l$ of each stencil $I^\pm_l$ defined here following [9] as the sum of the L_2-norms of all the derivatives of the interpolation polynomial $p^\pm_l(x)$ constructed for each stencil over the grid cell j:

$$S^\pm_l = \sum_{k=1}^{L-1} \Delta x^{2k-1} \int_{x_{j-1/2}}^{x_{j+1/2}} \left(\frac{d^k}{dx^k} p^\pm_l\right)^2 dx. \tag{15}$$

For the third-order WENO reconstruction ($L = 2$) the smoothness indicators for each of the two stencils $I^\pm_l$ are:

$$S^\pm_1 = \left(\delta^\pm_j - \delta^\pm_{j-1}\right)^2, \qquad S^\pm_2 = \left(\delta^\pm_{j+1} - \delta^\pm_j\right)^2. \tag{16}$$

The non-linear weights are:

$$w^\pm_{l=1,L} = \frac{\gamma_l}{(\epsilon^\pm + S^\pm_l)^2} \frac{1}{\sum_{k=1}^{L} \gamma_k (\epsilon^\pm + S^\pm_k)^{-2}}, \tag{17}$$

where, γ_l, the optimal linear weights for $L = 2$ are:

$$\gamma_1 = \frac{1}{3}, \qquad \gamma_2 = \frac{2}{3}, \tag{18}$$

and ϵ is a small number designed to avoid a zero division, and is defined as [6]:

$$\epsilon^{\pm} = 10^{-6}\max\left\{\left(\delta^{\pm}_{j-L+1}\right)^2, \ldots, \left(\delta^{\pm}_{j+L-1}\right)^2\right\} + 10^{-99}. \tag{19}$$

The non-linear weights are designed to favor the use of the smoother Lth-order stencils. This prevents interpolation across regions of sharp variations that tend to produce overshoots in the solution. If C is smooth then $w_l = \gamma_l$, yielding a third-order accurate scheme [9]. Although the WENO-based spatial discretization such as Eq. (12) is not TVD, it can yield a Total Variation Bounded (TVB) scheme [20].

Since $x_{j-1/2} = x_{j+1/2} - \Delta x$, $F^{\pm}_{j-1/2}$ can be computed by recalculating the weights w_l and the polynomials p_l by shifting down the space index j by one unit.

2.2. Extension to multidimensional space

The one-dimensional approaches above can be applied to multi-dimensional advection problems by combining flux differences in each spatial direction, for instance in a 2D (x, y) domain (using k as discrete index along the y-direction):

$$\frac{dC_{j,k}}{dt} + \frac{1}{\Delta x}(F_{j+1/2} - F_{j-1/2}) + \frac{1}{\Delta y}(F_{k+1/2} - F_{k-1/2}) = 0. \tag{20}$$

Another possibility is the dimensional splitting of the Partial Differential Equation (1) into several initial value problems. In the 2D (x, y) space domain:

$$C^{n+2} = \left[\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}\right]C^n, \tag{21}$$

where $\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}$ and $\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}$ are respectively the numerical approximations to the advection operators along the x- and y-directions over one time step Δt.

The above splitting scheme yields a solution that is second-order accurate in time every other time step, for a computational cost similar to that of a first-order splitting scheme [23] or to the use of Eq. (20).

Similarly, such a splitting can also be extended to 3D (*e.g.*, (x, y, z)) space:

$$C^{n+2} = \left[\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}\tilde{\mathcal{A}}^{\Delta t}_{z,u_z}\tilde{\mathcal{A}}^{\Delta t}_{z,u_z}\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}\right]C^n, \tag{22}$$

where $\tilde{\mathcal{A}}^{\Delta t}_{z,u_z}$ is the numerical approximation to the advection operator along the z-direction over one time step.

2.3. Time discretization

The numerical time integration of Eq. (7) can be achieved through a first-order TVD Euler explicit step:

$$C^{n+1} = C^n + \Delta t\mathbf{u}^n \cdot \nabla C^n = \tilde{\mathcal{A}}^{\Delta t}_{\mathbf{u}} C^n, \tag{23}$$

where $\tilde{\mathcal{A}}^{\Delta t}_{\mathbf{u}}$ is the numerical approximation to the advection operator $\mathcal{A}_{\mathbf{u}} = \partial_t + \mathbf{u}\cdot\nabla$ over one time step. A second-order accurate TVD *Runge–Kutta* solution can be achieved by performing an additional Euler step followed by averaging [20]:

$$C^{n+1} = \frac{1}{2}\left(C^n + \tilde{\mathcal{A}}^{\Delta t}_{\mathbf{u}}\tilde{\mathcal{A}}^{\Delta t}_{\mathbf{u}} C^n\right). \tag{24}$$

3. The two-way advection equation

As an alternative to the resolution of the advection equation (1) Wu [24] proposed to model advective transport through the resolution of a second-order Wave Equation Model (WEM). The latter is obtained by applying the advection operator $\mathcal{A}_{\mathbf{u}} = \partial_t + \mathbf{u}\cdot\nabla$ to C in two opposite directions of the velocity field, hence $\mathcal{A}_{-\mathbf{u}}\mathcal{A}_{+\mathbf{u}}C = 0$, leading to:

$$\partial_{tt}C = \mathbf{u}\cdot\nabla(\mathbf{u}\cdot\nabla C) - \partial_t\mathbf{u}\cdot\nabla C. \tag{25}$$

If the initial conditions on the time derivative of C are appropriately specified, the solutions of the second-order two-way equation and the first-order one-way advection equation are identical. These initial conditions need to be consistent with the one-way advection equation (Eq. (1)):

$$\partial_t C(\mathbf{x}, t=0) = -\mathbf{u}(\mathbf{x}, t=0)\cdot\nabla C(\mathbf{x}, t=0), \tag{26}$$

where $\mathbf{x}$ is the position vector. The equivalence between the two-way and the one-way advection equation can easily be seen for a 1D unbounded domain with constant velocity u. In this case, the solution of Eq. (6) is simply a translation along the x-axis in the direction of u, *i.e.*, $C(x, t) = C(x - ut, t = 0)$. Then, Eq. (25) simplifies to the classical acoustic wave equation:

$$\partial_{tt}C = u^2\partial_{xx}C, \tag{27}$$

with the following initial conditions consistent with the advection equation (6):

$$C(x,t=0) = f(x), \qquad \partial_t C(x,t=0) = -u\partial_x C(x,t=0) = f'(x) = g(x). \tag{28}$$

The general d'*Alembert* solution of Eqs. (27)–(28) is the sum of two waves traveling in opposite directions:

$$C(x,t) = \frac{1}{2}\big[f(x+ut) + f(x-ut)\big] + \frac{1}{2u}\int_{x-ut}^{x+ut} g(X)\,dX. \tag{29}$$

Using Eq. (28) the above expression yields: $C(x,t) = f(x-ut)$, *i.e.*, a translation of the initial shape of C, which is identical to the solution of the one-way wave equation. The two symmetric characteristics revealed in Eq. (29) suggest that contrary to the one-way advection equation, symmetric centered schemes are best suited for discretizing the spatial derivatives in Eq. (27).

3.1. Spatial discretization

As suggested previously, the partial derivatives in Eq. (27) can be approximated by centered Finite Differences (FD) stencils of even order $2L$:

$$\left(\frac{\partial^2 C}{\partial h^2}\right)_j \cong \sum_{l=-L}^{L} s_l C_{j+l}, \tag{30}$$

where the stencil weights s_l can be determined using the following recurrence formulas [7,14]:

$$s_l = \begin{cases} \frac{(-1)^{l+1}}{(\Delta h l)^2}\prod_{1\leqslant m\leqslant L, m\neq l}\left|\frac{m^2}{m^2-l^2}\right| & (l=1,2,\ldots,L), \\ 2\sum_{m=1}^{L} s_m & (l=0). \end{cases} \tag{31}$$

The lowest possible value $L=1$ yields the second-order semi-discrete FD scheme:

$$\frac{d^2C_j}{dt^2} = \frac{u^2}{\Delta x^2}(C_{j-1} - 2C_j + C_{j+1}). \tag{32}$$

As for the one-way advection equation, the one-dimensional approach above (Eq. (27)) can be extended to multi-dimensional advection problems by performing a dimensional splitting according to Eq. (21).

3.2. Time discretization

The time derivatives in Eq. (27) can also be approximated by the same centered FD stencils (Eq. (30)). With $L=1$, Eq. (32) can be approximated by the following explicit FD scheme:

$$C_j^{n+1} = \alpha^2\left[C_{j-1}^n + 2\left(\frac{1}{\alpha^2} - 1\right)C_j^n + C_{j+1}^n\right] - C_j^{n-1}, \tag{33}$$

where $\alpha = u\Delta t/\Delta x$ is the *Courant* number.

Note that the initial conditions (Eq. (28)) also require the evaluation of the first-order time derivatives of C, or the value $C_j^{n-1} = C(x_j, t_0 - \Delta t)$ in the above equation at the first time step. For purely 1D linear advection if $C(x,t=t_0) = f(x)$ is known analytically, therefore $C(x,t-\Delta t) = f(x-u\Delta t)$ can straightforwardly be determined. However, if $f(x)$ has no analytical expression, $\partial_t C$ must be accurately determined. Even small errors can have dramatic impact on the solution, by producing unphysical solutions such as waves traveling upstream [10]. In this case, the method of characteristics is a suitable approach to evaluate $\partial_t C|_{t=t_0} = -u\partial_x C(x,t_0)$ [24]. However, the situation gets worse in the more general multi-dimensional case if $\mathbf{u} = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))$ with $f_1 \neq f_2 \neq f_3$ and if Eq. (27) is used to perform dimensional splitting. In this case, $\partial_t C$ (or C_j^{n-1} in Eq. (33)) must be evaluated at every splitting step. Since the focus here is on Eulerian advection, using the method of characteristics at every grid point and every time step to evaluate $\partial_t C$ would make little sense as it would be equivalent to solving the one-way advection equation (Eq. (1)) in the whole computational domain with Lagrangian methods. Instead, one can use an asymmetric stencil that incorporates information about the characteristics in order to approximate the second-order time derivatives:

$$\left(\frac{\partial^2 C}{\partial t^2}\right)_j^n \cong \frac{1}{\delta t_1 + \delta t_2}\left[\left(\frac{\partial C}{\partial t}\right)_{t+\delta t_1} - \left(\frac{\partial C}{\partial t}\right)_{t-\delta t_2}\right]. \tag{34}$$

By setting $\delta t_1 = \Delta t/2$ and $\delta t_2 = \Delta x/(2u)$ and discretizing the first-order time derivatives with a second-order centered FD stencil Eq. (34) becomes:

$$\left(\frac{\partial^2 C}{\partial t^2}\right)_{x,t} \cong \frac{1}{2(\Delta x/u + \Delta t)}\left[\frac{C(x,t) - C(x,t-\Delta x/u)}{\Delta x/u} - \frac{C(x,t+\Delta t) - C(x,t)}{\Delta t}\right]. \tag{35}$$

Along the characteristics, Eq. (6) reduces to the ordinary differential equation: $d_t C = 0$, *i.e.*, C remains constant. This enables one to write (assuming that $u > 0$):

$$C(x, t - \Delta x/u) = C(x + \Delta x, t). \tag{36}$$

Using the above exact relationship and Eq. (35) finally yields an expression for the second-order time derivatives that incorporates the information about the characteristics:

$$\left(\frac{\partial^2 C}{\partial t^2}\right)_j^n \cong \frac{2\alpha}{(\alpha+1)\Delta t^2}\left[\alpha C_{j+1}^n - (\alpha+1)C_j^n + C_j^{n+1}\right]. \tag{37}$$

Komatsu et al. [10] used the above asymmetric stencil combined with a symmetric stencil for spatial derivatives embedded in the following implicit multi-level scheme to discretize Eq. (27):

$$(1-\xi)(\partial_{tt}C)_{j-1}^n + \xi(\partial_{tt}C)_j^n = \alpha^2\left[\theta\left(C_{j-1}^{n+1} - 2C_j^{n+1} + C_{j+1}^{n+1}\right) + (1-\theta)\left(C_{j-1}^n - 2C_j^n + C_{j+1}^n\right)\right]. \tag{38}$$

By selecting semi-empirical optimum values for $\xi = 0.5155\alpha + 0.9688$ and $\theta = -0.1691\alpha - 0.3152$ such that the largest truncation error terms are minimized, Komatsu et al. [10] obtained the following Second-Order Wave equation for Method for Advective Calculation (SOWMAC) implicit scheme:

$$c_1 C_{j-1}^{n+1} + c_2 C_j^{n+1} + c_3 C_{j+1}^{n+1} = c_4 C_{j-1}^n + c_5 C_j^n + c_6 C_{j+1}^n, \tag{39}$$

where coefficients $c_1, c_2, \ldots, c_6$, expressed for a velocity u of arbitrary sign, are:

$$\begin{aligned}
c_1 &= 0.3776\alpha_{0+} + 0.3152\alpha_{0-} - 0.5467\alpha_{1+} + 0.4843\alpha_{1-} + 0.1691\alpha^2,\\
c_2 &= 1.3072\alpha_{0+} + 0.0624|\alpha| - 0.3382\alpha^2,\\
c_3 &= 0.3152\alpha_{0+} + 0.3776\alpha_{0-} + 0.4843\alpha_{1+} - 0.5467\alpha_{1-} + 0.1691\alpha^2,\\
c_4 &= 0.3776\alpha_{0+} + 0.3152\alpha_{0-} + 0.5157\alpha_{1+} - 0.4533\alpha_{1-} + 0.1381\alpha^2,\\
c_5 &= 1.3072 - 0.0624|\alpha| - 0.2762\alpha^2,\\
c_6 &= 0.3152\alpha_{0+} + 0.3776\alpha_{0-} - 0.4533\alpha_{1+} + 0.5157\alpha_{1-} + 0.1381\alpha^2,\\
\alpha &= u_j\frac{\Delta t}{\Delta x},\\
\alpha_{0+} &= \mathrm{AINT}\left(\frac{\alpha+1}{|\alpha|+1}\right), \qquad \alpha_{0-} = \mathrm{AINT}\left(\frac{1-\alpha}{1+|\alpha|}\right),\\
\alpha_{1+} &= \frac{|\alpha|+\alpha}{2}, \qquad \alpha_{1-} = \frac{|\alpha - |\alpha||}{2},
\end{aligned} \tag{40}$$

where AINT is a generic function that truncates the decimals of its argument. The above scheme is second-order in time and space, and it yields a tri-diagonal system that can be efficiently solved using the Thomas algorithm.

The above one-dimensional approach (Eq. (27)) can be generalized to non-constant velocity field $u(x,t)$ and for variable spatial dimensions by discretizing Eq. (25) [24,25]. In this case, however, the corresponding equation contains spatial cross-derivatives which, upon testing, yield strongly anisotropic dispersive errors that can be difficult to handle. For these reasons, in the case of multidimensional space, we will apply the simple one-dimensional approach above (Eq. (27)) combined with dimensional splitting (see Section 2.2) to cases where u varies in space and time. Although such a simplification would seem rather crude *a priori*, published numerical tests (as well as the tests presented in this manuscript) show that the associated errors are rather small in practice, even in the case of sharply varying velocity field [10]. Indeed, for multi-dimensional cases the SOWMAC approach combined with dimensional splitting yields more accurate results than the discretization of Eq. (25).

4. Accuracy of the one-way vs. the two-way advection solutions

We compare the two approaches presented in Sections 2 and 3 to model advective transport, focusing on the spatial discretization.

4.1. Fourier analysis

We conduct standard Fourier analysis to reveal the differences in accuracy between the numerical solutions of the one-way advection equation discretized using the schemes described in Section 2 and the discretization of the two-way advection equation (Eq. (3)). We consider the linear advection with constant velocity u in a 1D unbounded domain, for which the solution is the superposition of individual modes, characterized by their wave number k and frequency ω. Given the initial conditions specified in Eq. (28), a particular mode satisfies both Eqs. (6) and (27): $C(x,t) = e^{i(kx-\omega t)}$, where i is the complex unit such that $i^2 = -1$. Similarly, the numerical solution to the advection equation (6) is:

$$C_j(t) = e^{i(kj\Delta x - \tilde{\omega}t)}. \tag{41}$$

The approximation of the spatial derivatives in Eqs. (6) and (27) yields a dispersion relation different from that of the true solution: $\omega = uk$. Such a difference is indicative of the level of truncation induced by the discretization. The latter produces dispersion errors that involve differences in group and phase velocities of the solution, and dissipation/amplitude errors. A measure of these two types of errors is:

$$E_\omega = \frac{\mathbf{Re}(\tilde{\omega}) - \omega}{\omega} = \frac{\mathbf{Re}(\tilde{\omega})}{uk} - 1, \tag{42a}$$

$$E_a = 1 - \exp\left[ut\mathbf{Im}(\tilde{\omega})\right]. \tag{42b}$$

The spatial derivatives in the one-way advection equation (7) can be discretized using a simple centered symmetric second-order (CS2) scheme. In this case, substituting C_j with Eq. (41) in the resulting FD equation yields the expression for $\tilde{\omega}_{\text{one-way, CS2}}$, the frequency associated with the second-order centered discretization of the spatial derivatives in Eq. (7). This results in the following frequency and amplitude errors:

$$E_\omega^{\text{one-way, CS2}} = \frac{1}{6k\Delta x}\left[8\sin(k\Delta x) - \sin(2k\Delta x)\right] - 1, \tag{43a}$$

$$E_a^{\text{one-way, CS2}} = 0. \tag{43b}$$

Similarly, using a symmetric second-order FD stencil to discretize the spatial derivatives present in the two-way advection equation (Eq. (27)) yields no dissipation errors ($E_a^{\text{two-way, CS2}} = 0$) and results in the following expression for the frequency error:

$$E_\omega^{\text{two-way, CS2}} = \frac{\sqrt{2 - 2\cos k\Delta x}}{k\Delta x} - 1. \tag{44}$$

By comparing Eq. (43) and Eq. (44) one can see that for relatively well resolved wavelengths (*i.e.*, if the smallest half-wavelength is represented with 4 points or less) $E_\omega^{\text{two-way, CS2}} < E_\omega^{\text{one-way, CS2}}$ the two-way advection equation solution is less prone to development of spurious dispersive oscillations than the one-way advection equation solution discretized with centered schemes of identical order (compare also plain light blue and dashed green curves in Fig. 1(a)).

High frequency dispersive errors can be damped by using asymmetric upwind-biased stencils such as the WENO reconstruction presented in Section 2.1. However, this standard fix is known to introduce additional dissipation errors that progressively damp the solution and reduce the accuracy with time. For instance, using a third-order WENO scheme (see Section 2), the corresponding expressions for the frequency and amplitude errors associated with the discretization of Eq. (7) are in the best case (*i.e.*, assuming that C is perfectly smooth, in the sense of Eq. (15)):

$$E_\omega^{\text{one-way, WENO3}} = \frac{1}{6k\Delta x}\left[8\sin(k\Delta x) - \sin(2k\Delta x)\right] - 1, \tag{45a}$$

$$E_a^{\text{one-way, WENO3}} = 1 - \exp\left\{\frac{ut}{\Delta x}\left[4\cos(k\Delta x) - 3 - \cos(2k\Delta x)\right]\right\}. \tag{45b}$$

The comparison of Eq. (43) with Eq. (45) reveals that not only the frequency error has not improved over the one-way second-order centered scheme, but the use of a higher order asymmetric stencil has introduced amplitude errors, otherwise absent in symmetric discretizations. Consequently, when advecting a sufficiently smooth field the use of the two-way advection equation discretized using a symmetric stencil is recommended over the standard one-way advection equation.

The above results can be generalized for higher order WENO and FD discretizations of the one-way and the two-way advection equations, respectively. Indeed, the frequency associated with the discretization of the spatial derivatives of arbitrary order $2L$ in Eq. (27) is:

$$\tilde{\omega}_{\text{two-way}} = u\left(-\sum_{l=-L}^{L} s_l e^{ikl\Delta x}\right)^{1/2} = u\left(-\sum_{l=-L}^{-1} s_l e^{ikl\Delta x} - \sum_{l=1}^{l=L} s_l e^{ikl\Delta x} - s_0\right)^{1/2}. \tag{46}$$

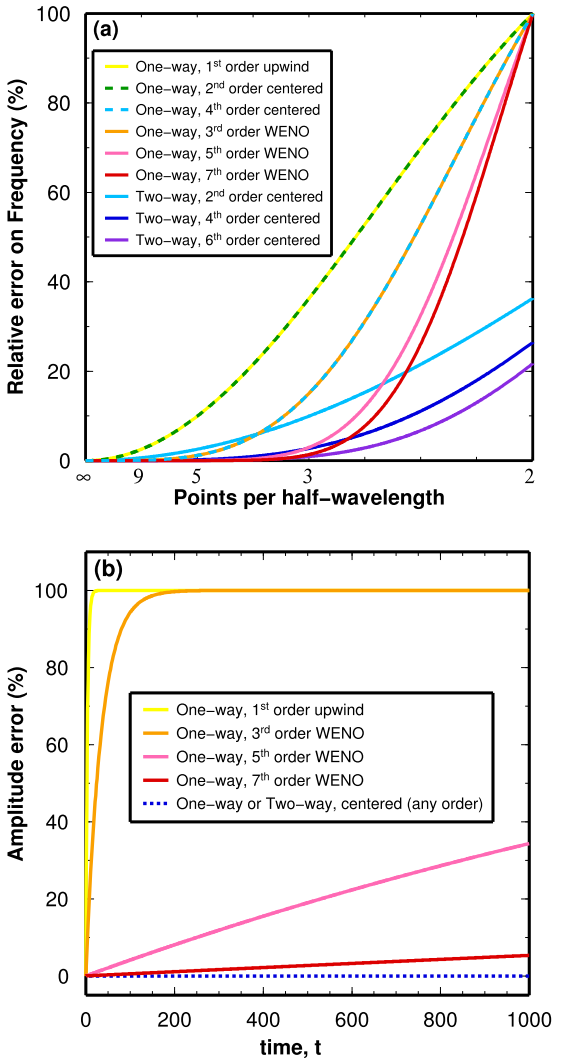

Fig. 1. Results of the Fourier analysis of the spatial discretization of the one-way advection equation and the two-way advection equation. (a) Relative frequency error and (b) amplitude error for various spatial discretizations. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

If the FD stencil weights s_l are chosen according to centered symmetric formula (Eq. (31)) then $s_l = s_{-l}$ leading to:

$$\tilde{\omega}_{\text{two-way, symmetric}} = u\left[-s_0 - \sum_{l=1}^{L} s_l\left(e^{ikl\Delta x} + e^{-ikl\Delta x}\right)\right]^{1/2} = u\left[-2\sum_{l=1}^{L} s_l - \sum_{l=1}^{L} s_l 2\cos(kl\Delta x)\right]^{1/2}. \tag{47}$$

Since the stencil coefficients in Eq. (31) are all real, $\mathbf{Im}(s_{l=-L,\ldots,L}) = 0$, and we obtain $\mathbf{Im}(\tilde{\omega}_{\text{two-way, symmetric}}) = 0$ for all L.

Similarly, one can express the discretization of the spatial derivatives for the one-way advection equation (Eq. (6)) with a stencil s_l of arbitrary width $2L$, leading to the associated frequency:

$$\tilde{\omega}_{\text{one-way}} = -ui\left(\sum_{l=-L}^{-1} s_l e^{ikl\Delta x} + \sum_{l=1}^{l=L} s_l e^{ikl\Delta x} + s_0\right). \tag{48}$$

If one uses centered schemes (*i.e.*, $F_{j\pm1/2} = (F^{-}_{j\pm1/2} + F^{+}_{j\pm1/2})/2$) the associated stencil values are symmetric: $s_l = -s_{-l}$ and $s_0 = 0$. Therefore the corresponding frequency is:

$$\tilde{\omega}_{\text{one-way, symmetric}} = u\sum_{l=1}^{L} -i\left(e^{ikl\Delta x} - e^{-ikl\Delta x}\right)s_l = u\sum_{l=1}^{L} 2\sin(kl\Delta x)s_l. \tag{49}$$

The terms in the above equation are real, therefore $E_a^{\text{one-way, symmetric}} = 0$ for all L.

If, however, one uses upwind-biased schemes (Eq. (8)), this yields asymmetric stencils $s_l \neq -s_{-l}$ and results in an imbalance of the summation terms present in Eq. (48). Consequently, $E_a^{\text{one-way, asymmetric}} = \mathbf{Im}(\tilde{\omega}_{\text{one-way, WENO}}) \neq 0$ for all L.

The above results are illustrated in Fig. 1 whose top panel displays the frequency error associated with the spatial discretization of Eq. (6) with several symmetric FD schemes or asymmetric (WENO) schemes, and the errors associated with the spatial discretization of Eq. (27) with second-, fourth- and sixth-order centered FD schemes for different resolution wavelengths. Cross overs between the one-way and two-way curves indicate that lower-order discretizations of the two-way advection equation can yield smaller dispersive errors than higher-order discretization of the one-way advection equation. Moreover, the amplitude errors displayed in Fig. 1(b) illustrate the advantage of the two-way advection equation discretized using spatially centered schemes (free of amplitude errors) over the asymmetric discretizations of the one-way advection equation (using WENO schemes), whose errors continuously increase with time. Therefore, even for 1D linear advection over a long time the two-way advection equation is clearly superior to the one-way advection equation (WENO discretization), as long as the smallest half-wavelength to be advected is correctly resolved with a minimum of three grid points.

4.2. Numerical tests

The Fourier analysis presented above is supplemented by the following 1D, 2D and 3D linear advection tests.

4.2.1. 1D translation test

We consider the 1D advection with constant velocity $u = 1$ in a periodic domain of length 100 discretized using N cells of equal size Δx. The field to advect is the sum of three Gaussians:

$$C(x, t = 0) = \frac{1}{2.15}\left[e^{-0.1A^2} + (1 + A)e^{-0.05A^2} - \frac{A}{2}e^{-0.015A^2}\right],$$

with $A = x - 20$. Fig. 2 shows the results obtained at $t = 300$ corresponding to 1500 equal time steps $\Delta t = 0.2\Delta x/u$. The time discretization of the one-way advection equation is performed via a second-order TVD Runge–Kutta scheme (Eq. (24)), while the two-way advection equation (Eq. (27)) is discretized in time using a centered scheme that is also second-order accurate. Several spatial discretizations of the one-way advection equation are considered: symmetric schemes (Fig. 2(a)–(d)) as well as asymmetric schemes: first-order upwind (Fig. 2(e)), second-order *Lax–Wendroff* with *Van Leer*, *Minmod* or *Superbee* limiters (Fig. 2(f)–(h)), 3rd-, 5th- and 7th-order WENO schemes (Fig. 2(d)–(f)). The spatial derivatives for the two-way advection equation are discretized using symmetric centered FD stencils of 2nd, 4th and 6th order (Fig. 2(m)–(o)). The second-order SOWMAC scheme is also shown (Fig. 2(p)). As predicted by the Fourier analysis (Section 4.1), the solutions of the one-way advection equation discretized using centered schemes are affected by dispersive errors, which are clearly visible up to the 8th-order scheme (*i.e.*, spurious oscillations in Fig. 2(a)–(d)). Similar dispersive errors can be seen for the solutions of the two-way advection equation, however these are invisible to the naked eye for fourth-order schemes and higher. Therefore, in agreement with the Fourier analysis, the solution of the two-way advection equation is more stable and more accurate than the solution of the one-way advection equation discretized using centered schemes of similar or higher order. This is particularly true for the SOWMAC solution of the two-way advection equation that is more accurate than the one-way advection equation discretized using an 8th-order centered scheme.

In contrast to centered schemes, the solutions obtained with the one-way advection equation discretized using asymmetric schemes (Fig. 2(e)–(l)) are visibly affected by numerical dissipation, with amplitude errors decreasing with the order of the spatial discretization, as predicted by the Fourier analysis (Section 4.1).

The comparison of the solutions of the one-way advection equation displayed in Fig. 2 suggests that the scheme that yields a reasonably accurate solution at a lowest computational cost is the 7th-order WENO scheme. This represents a much higher order (and associated computational cost) than the use of the two-way advection equation discretized using fourth-order centered or only second-order SOWMAC schemes that yield solutions of comparable or of higher accuracy. Despite the presence of dissipation, WENO discretizations seem to perform the best among the schemes that were tested here for the one-way advection equation. For this reason, in the following we will restrict our comparison between the one-way and two-way advection by considering only WENO discretizations of the one-way advection equation and centered discretizations of the two-way advection equation.

Fig. 3 displays the L_1 and L_2 errors associated with the discretizations of the one-way advection equation using WENO schemes and the two-way advection equation using centered FD schemes as a function of the grid resolution Δx, at an early elapsed time $t = 10^{-3}$ and at a final elapsed time $t = 300$.

$$L_1 = \frac{1}{N}\sum_{j=1}^{N}\left|C_j - C_j^{\text{true}}\right|, \qquad L_2 = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(C_j - C_j^{\text{true}}\right)^2}.$$

The errors measured at the early elapsed time should be mostly dominated by the truncation terms in the spatial discretization, while the errors measured at the final time result from both spatial and temporal discretizations. The slopes

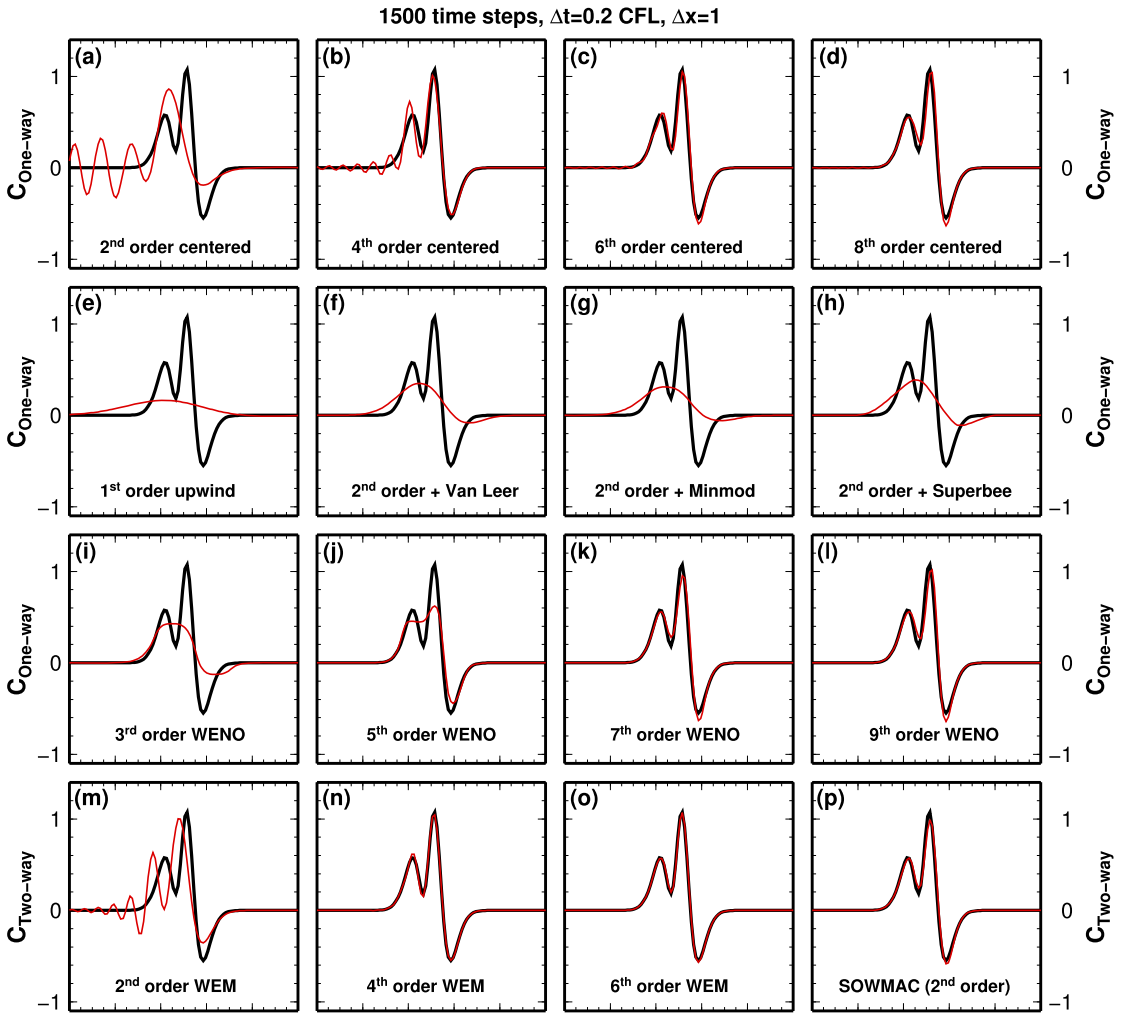

Fig. 2. Numerical results of the one-dimensional linear advection test in a periodic domain during 1500 time steps of equal length $\Delta t = 0.2\Delta x/u$, with $u = 1$. ((a)–(l)) Results obtained using the one-way advection equation. ((m)–(p)) Results obtained using the two-way advection equation discretized with FD schemes of various order in space. Black: true solution. Red: numerical solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$L_1/\Delta x$ and $L_2/\Delta x$ are consistent with the order of the spatial discretization of each corresponding scheme. However, one can see that at moderate resolution the discretizations of the two-way advection equation are significantly more accurate than that of the one-way advection equation discretized using WENO schemes of higher orders. For instance, for $\Delta x < 0.33$ the two-way advection equation discretized with a centered second-order FD stencil is more accurate that the one-way advection equation discretized with WENO schemes of order $\leqslant 9$. At the final time, the two-way advection equation spatially discretized with fourth-order and sixth-order FD stencils yields more accurate results than any discretization of the one-way advection equation shown in Fig. 3. The slopes $L_1/\Delta x$ and $L_2/\Delta x$ for all discretizations (except for the third-order WENO reconstruction) are close to 2, which may be indicative of the domination of the errors due to the time discretization.

Table 1 lists the effective spatial order of accuracy estimated using the slopes of the L_2 error curves (similar results would have been obtained using the L_1 errors instead), at small times and at large times. At small times the effective order of accuracy of the spatial discretization of the two-way advection equation is very close to the order of the FD stencil used. On the other hand, the effective order of accuracy for the spatial discretizations of the one-way advection equation can be smaller than that of the corresponding WENO schemes. This is due to the fact that the field to advect is not sufficiently smooth (according to Eq. (15)) so that the linear and the non-linear WENO weights differ significantly, therefore reducing the order of accuracy of the scheme. At larger times, the error is dominated by the time discretization, resulting in effective order of accuracy close or lower than two, for most discretizations of the one-way and two-way advection equations.

Overall, the results of the advection tests displayed in Fig. 2 and Fig. 3 and listed in Table 1 confirm the Fourier analysis, and show that the use of the two-way advection equation yields more accurate results than higher order discretizations of the one-way advection equation. This is also in good agreement with earlier work [24,25,10].

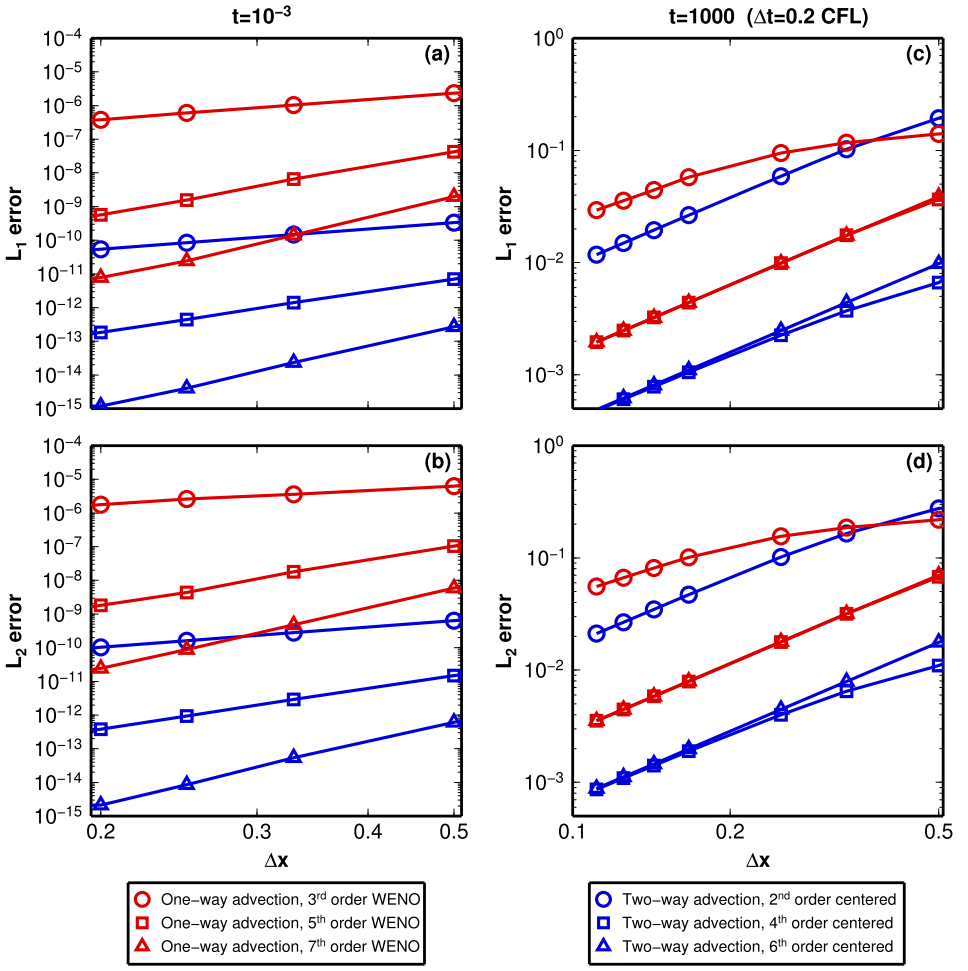

Fig. 3. Numerical results of the one-dimensional linear advection test with constant velocity $u = 1$ in a periodic domain L_1 and L_2 errors obtained using the one-way (red symbols) and the two-way (blue symbols) advection equation as a function of the number of grid points used to discretize the spatial domain. ((a)–(b)) Errors after one small time step $\Delta t = 0.001$. ((c)–(d)) Errors after $t = 1000$ (5000 time steps $\Delta t = 0.2\Delta x/u$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Effective spatial accuracy of the one-way advection equation and the two-way advection equation discretized using different schemes.

Advection equation	Scheme	Effective convergence power	
		$t \ll 1$	$t \gg 1$
One-way	3rd-order WENO	1.96	–
One-way	5th-order WENO	4.75	1.98
One-way	7th-order WENO	6.10	1.99
Two-way	2nd-order centered	1.99	1.75
Two-way	4th-order centered	3.98	1.72
Two-way	6th-order centered	5.99	1.99
Two-way	SOWMAC	1.95	1.93

4.2.2. 2D solid rotation test

We extend the comparison between the solutions of the one-way advection equation and the two-way advection equation to a 2D square domain (x, y) of unit length discretized using square cells of size $\Delta x = \Delta y = \Delta h$. The field to advect is a circular disk of radius 0.15 whose center is located at $x = x_0 = 0.5$ and $y = y_0 = 0.75$. The disk is smoothed in order to avoid the development of dispersion errors when solving for the two-way advection equation (as seen for instance in Fig. 2 and predicted by the Fourier analysis):

$$C(x, y) = \frac{1}{2}\left\{1 - \tanh\left[50\left(\sqrt{(x - x_0)^2 + (y - y_0)^2} - 0.15\right)\right]\right\}.$$

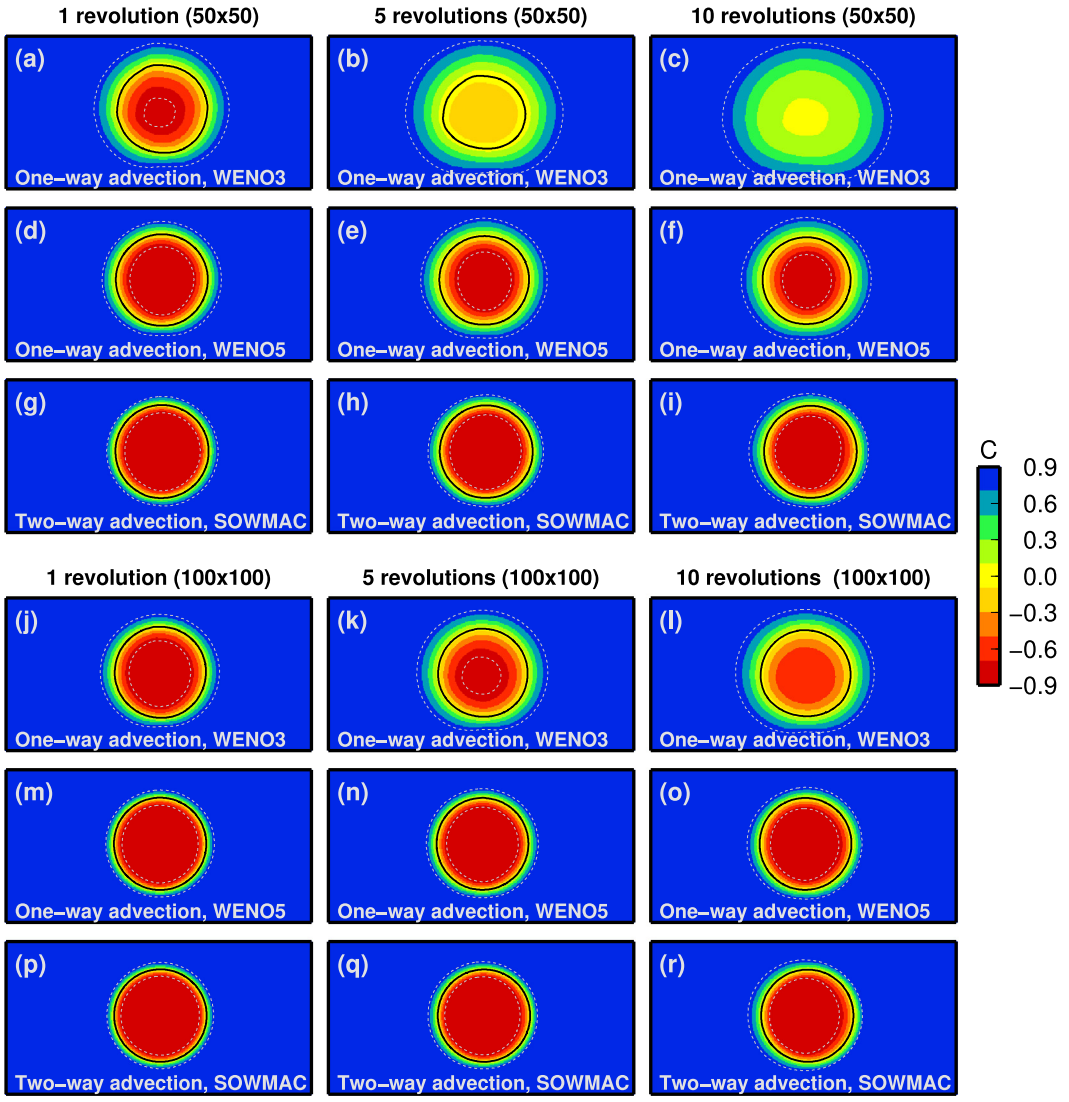

Fig. 4. 2D advection of a smooth disk in a solid rotation velocity field using the one-way advection equation (with fifth-order WENO and second-order TVD-RK schemes for space and time discretizations) or the two-way advection equation (with the second-order SOWMAC scheme (Eq. (39))). Results obtained after 1, 5 and 10 revolutions, using 50×50 grid cells ((a)–(g)) or using 100×100 grid cells ((h)–(m)).

A solid rotation velocity field is prescribed:

$$\mathbf{u} = \begin{cases} u_x = +x - 0.5, \\ u_y = -y + 0.5. \end{cases} \tag{50}$$

The one-way advection equation was integrated in time using an explicit second-order TVD Runge–Kutta scheme (Eq. (24)), and the spatial terms were discretized using a fifth-order WENO scheme. The two-way advection equation (Eq. (27)) was solved using the SOWMAC scheme (Eq. (39)).

The multidimensional advection is performed via Eq. (20) for the one-way advection equation, while dimensional splitting (Eq. (21)) is used for the two-way advection equation with the one-dimensional advection operators $\tilde{\mathcal{A}}^{\Delta t}_{x,u_x}$ and $\tilde{\mathcal{A}}^{\Delta t}_{y,u_y}$ being the numerical approximation of the two-way advection equation along the x- and y-directions.

Fig. 4 shows the results obtained on 50×50 and 100×100 grid cells after 1, 5 and 10 revolutions. The artificial smearing of the disk is apparent on all results obtained with the one-way advection equation and increases with time and/or increasing Δh. After five revolutions the shape strongly deviates from its initial circular form (Fig. 4(e)–(g) and Fig. 4(k)–(m)). These artifacts are much less apparent for the lower-order discretization of the two-way advection equation (Fig. 4(a)–(c) and Fig. 4(h)–(j)).

These results along with other published work [24,25,10] confirm the findings of the 1D advection tests (Section 4.2.1) and suggest that the use of the two-way advection equation is more suitable to model the multi-dimensional advection

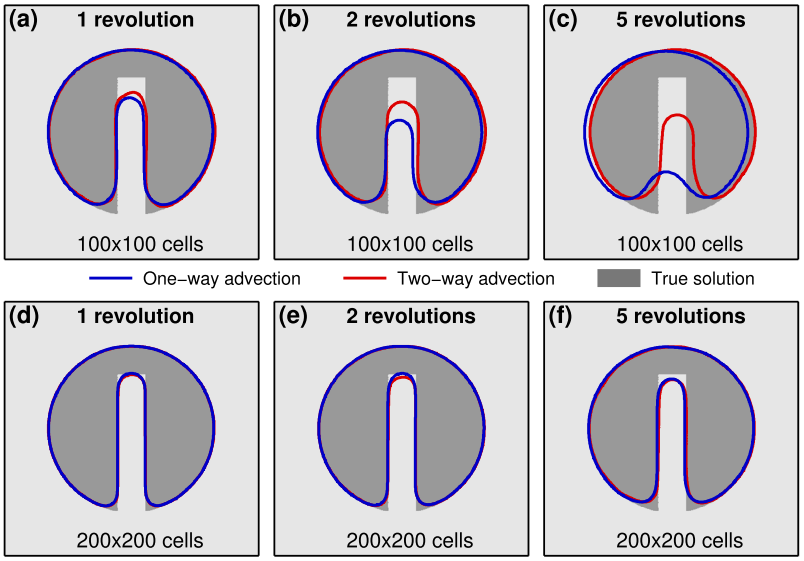

Fig. 5. 2D Zalesak disk test performed using the one-way level set method (with a fifth-order WENO and second-order TVD-RK for space and time discretizations, blue contours) or the two-way level set method (with the second-order SOWMAC scheme (Eq. (39)), red contours). Results for 1, 2 and 5 revolutions are shown on either a 100×100 cell grid ((a)–(c)) or on a 200×200 cell grid ((d)–(f)). The true solution is displayed in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

during long periods of time. Additionally, it confirms that the SOWMAC scheme originally designed for constant velocity remains accurate for variable velocity fields.

5. Application to level set interface tracking

Despite its higher accuracy, a major drawback of the two-way advection equation discretized using spatially centered schemes is that it can yield unstable dispersive errors in the presence of sharply varying quantities [24,25]. This restriction naturally suggests applying the two-way advection equation to track the advection of an interface via the level set method [16,19,15], because in this case the field to advect is smooth by construction, regardless the complexity of the interface's topology.

In the following, we evaluate the performance of this new level set method by testing it against two typical cases. In these tests, the Eikonal requirement (Eq. (3)) is implicitly enforced through the resolution of the reinitialization equation (4), whose spatial terms are discretized using a fifth-order WENO reconstruction and a second-order TVD Runge–Kutta scheme (Eq. (24)) for time integration. As noted by [21,19], a too frequent use of Eq. (4) can artificially move the interface. However in practice, a few iterations of Eq. (4) per time step are necessary to reach convergence. A constant number of iterations per time step was applied (typically between 2 and 4) to solve for (4). We investigated also a threshold criterion as in [21], which did not yield significant differences.

Eq. (3) is explicitly solved only once to initialize ϕ using a second-order fast marching method.

As in Section 4.2.2, the multidimensional advection is performed via Eq. (20) for the one-way advection equation, while dimensional splitting (Eq. (21)) is used for two-way advection equation.

The one-way advection equation was integrated in time using an explicit second-order TVD Runge–Kutta scheme (Eq. (24)), and the spatial terms were discretized using a fifth-order WENO scheme. The two-way advection equation (Eq. (27)) was solved using the SOWMAC scheme (Eq. (39)). The time step was set to $\Delta t = 0.25\Delta h / \max(|u_x|, |v_y|, |u_z|)$ to ensure the stability of both schemes.

5.1. Solid rotation of Zalesak's disk

This test consists in advecting a slotted disk of radius 0.15 initially located at $x = 0.5$ and $y = 0.75$ [26]. The slot width and length are respectively 0.05 and 0.25 and the same velocity field used in Section 4.2.2 (Eq. (50)) is specified.

Fig. 5 displays the results obtained on a 100×100 cell grid and on a 200×200 cell grid after 1, 2 and 5 revolutions. As for the previous tests presented in Section 4.2, the two-way advection equation performs better than the one-way advection equation discretized using higher-order schemes. For instance, one can notice that the slot has almost disappeared after five revolutions on the 100×100 cell grid with the one-way advection equation, while it is much more pronounced for the two-way advection equation solution. Although less apparent, the differences between the one-way advection equation and the two-way advection equation solutions remain on the finer grid (Fig. 5(d)–(f)) and increase with time.

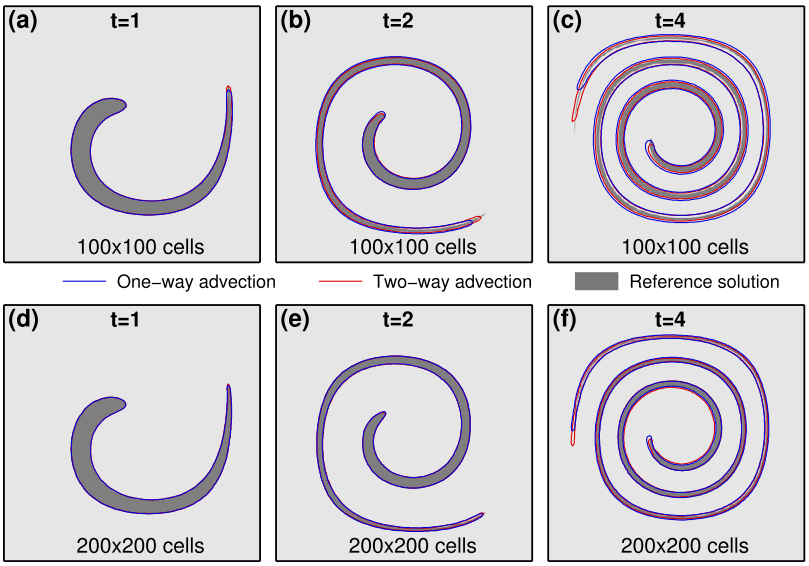

Fig. 6. 2D vortex test performed using the one-way level set method (with a fifth-order WENO and second-order TVD-RK for space and time discretizations, blue contours) or the two-way level set method (with the second-order SOWMAC scheme (Eq. (39)), red contours). Results for $T = 1, 2, 4$ are shown on either a 100×100 cell grid ((a)–(c)) or a 200×200 cell grid ((d)–(f)). A highly accurate Lagrangian reference solution is displayed in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Error on the area for the vortex test at different times and on two different grids.

Advection equation/Scheme	$t = 1$	$t = 2$	$t = 4$
	100×100 grid		
One-way/RK2-WENO3	2.37%	37.76%	194.25%
One-way/RK2-WENO5	1.18%	15.36%	117.19%
One-way/RK2-WENO7	0.87%	8.73%	89.06%
Two-way/SOWMAC	0.20%	0.81%	57.48%
	200×200 grid		
One-way/RK2-WENO3	0.59%	6.19%	72.31%
One-way/RK2-WENO5	0.43%	2.81%	28.99%
One-way/RK2-WENO7	0.29%	1.85%	16.59%
Two-way/SOWMAC	0.09%	0.94%	12.25%

5.2. The vortex test

We consider the advection of a disk of radius 0.15 located initially at $x = 0.5$ and $y = 0.75$. The velocity field $\mathbf{u} = (\partial_y \psi, -\partial_x \psi)$ is specified via the following streamfunction, ψ:

$$\psi(x, y) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y).$$

Such a velocity field yields a strong shearing of the disk, which would have led to strong dispersion errors in the two-way advection equation solution. However, since the level set is constantly maintained as a smooth field the two-way advection equation solution can be applied to track the interface $\phi = 0$. Fig. 6 shows snapshots at three different times ($t = 1, 2, 4$) calculated on two different grids. A highly accurate Lagrangian method was used to calculate the reference solution displayed in dark gray in Fig. 6. The results show that the two-way advection equation solution is visibly closer to the true solution than the one-way advection equation solution. As in the previous tests, differences between the solutions of one-way advection equation and the two-way advection equation increase with time and with decreasing the grid resolution. Different tests carried out using smaller time steps showed comparable results, indicating that most of the differences observed are due to the accuracy of the spatial discretization.

Table 2 shows the area error corresponding to each of the six frames displayed in Fig. 6. The magnitude of these errors confirms the qualitative interpretation of Fig. 6. Indeed, the area error associated with the two-way advection equation solution is 5–10 times smaller than higher order discretizations of the one-way advection equation.

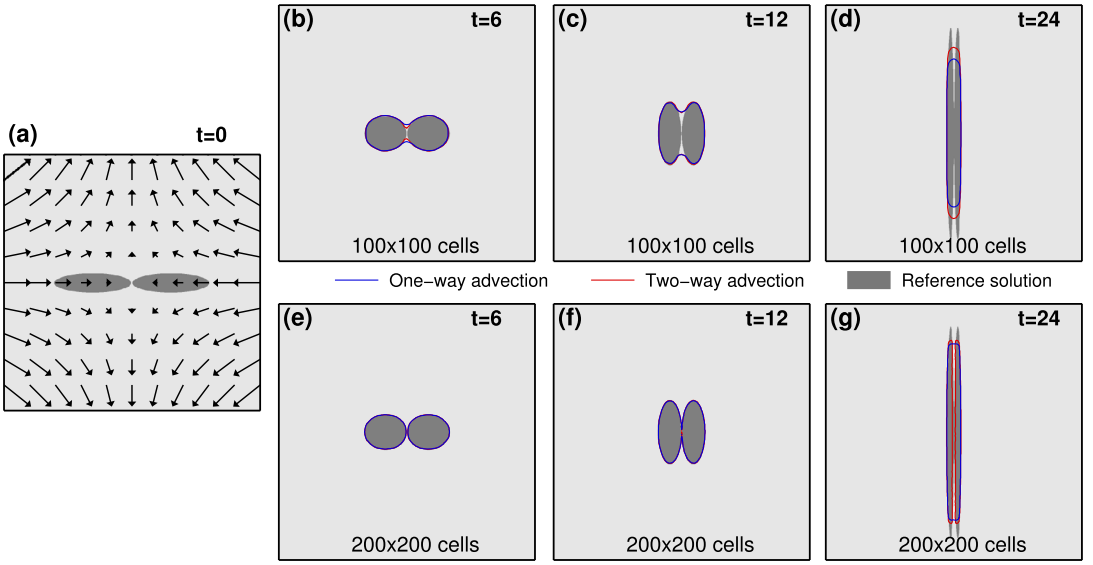

Fig. 7. 2D merging test performed using the one-way level set method (with a fifth-order WENO and second-order TVD-RK for space and time discretizations, blue contours) or the two-way level set method (with the second-order SOWMAC scheme (Eq. (39)), red contours). (a) Imposed velocity field and initial shape of the interfaces. Results for $T = 6, 12, 24$ are shown on either a 100×100 cell grid ((b)–(d)) or a 200×200 cell grid ((e)–(g)). A highly accurate Lagrangian reference solution is displayed in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.3. Merging test

While the level set function is constructed to be smooth, there are instances where the level set smoothness can be reduced, as in the case of interface merging or breakup [11]. We therefore compared the performances of the two-way and the one-way level set advection for a 2D stagnation point flow in a 1×1 domain:

$$\mathbf{u} = \begin{cases} u_x = +0.1(x - 0.5), \\ u_y = -0.1(z - 0.5). \end{cases}$$

At $t = 0$ the interface forms two identical ellipses of vertical and horizontal semi-axis lengths and 0.0375 and 0.15, respectively (Fig. 7(a)). The ellipses' centers are located at $\{x = 0.3475, y = 0.5\}$ and $\{x = 0.6525, y = 0.5\}$. The domain is discretized using either 100×100 or 200×200 square cells, such that at $t = 0$, the two ellipses are separated by a distance δ_e of at least one grid cell. Given the imposed stagnation point flow, the two ellipses will first converge towards the center of the domain, and will subsequently stretch along the vertical direction. Consequently, δ_e will first decrease, then increase. This could lead to merging and separation of the interface depending on the ratio between δ_e and the grid spacing and on the accuracy of the level set advection. Fig. 7 displays three snapshots in time ($t = 6, 12, 24$) on two different grids. A highly accurate Lagrangian method used to calculate the reference solution is also displayed in dark gray in Fig. 7. The solution obtained on the coarser grid shows artificial merging of the two interfaces, at $t = 6$ and $t = 12$. However, one sees that the solution of the two-way advection equation is more accurate than the solution of the one-way advection equation. At final time $t = 24$ both the one-way advection equation and the two-way advection equation solutions show a single ellipse which differs from the more complicated topography of the reference solution (Fig. 7(d)). As expected, better results are obtained on the finer grid (Fig. 7(e)–(g)), where the one-way advection equation and the two-way advection equation solutions are both very close to the reference solution. However, at final time $t = 24$ the solution of the two-way advection equation is much closer to the reference solution than the solution of the one-way advection equation. The latter has remained comparable to the coarser grid solution. Consequently, the results obtained in this test show that even when the level set smoothness is reduced the two-way advection equation remains more accurate than solution of the one-way advection equation.

5.4. Three-dimensional deformation test

We illustrate the use of the two-way level set advection in three-dimensional, time-dependent flows, as proposed by [12], in a $1 \times 1 \times 1$ domain discretized using 100 cubic cells in each spatial direction:

$$\mathbf{u} = \begin{cases} u_x = 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/3), \\ u_y = -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/3), \\ u_z = -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/3). \end{cases}$$

Table 3
Error on the volume for the 3D deformation test at final time $t = 3$, using a 100×100 grid.

Advection equation/Scheme	Volume error
One-way/RK2-WENO3	−53.10%
One-way/RK2-WENO5	−26.15%
One-way/RK2-WENO7	−14.22%
Two-way/SOWMAC	−2.63%

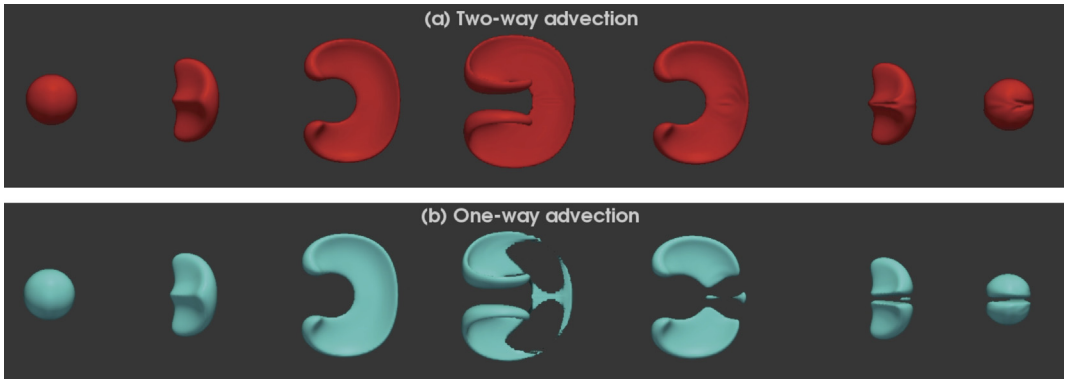

Fig. 8. 3D deformation test performed using the one-way level set method (with a fifth-order WENO and second-order TVD-RK for space and time discretizations, blue isosurfaces) or the two-way level set method (with the second-order SOWMAC scheme (Eq. (39)), red isosurfaces). Results for $t = 0, 0.375, 0.75, 1.5, 2.25, 2.625, 3$ are shown on a $100 \times 100 \times 100$ cell grid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

At $t = 0$, the interface is a sphere of radius 0.15 located at $x = y = z = 0.35$. The experiment is conducted until $t = 3$. The imposed time periodicity of $\mathbf{u}$ implies a symmetry of the true solution with respect to $t = 1.5$, at which the deformation is maximum. As previously, we use Eq. (22) to solve for the two-way advection equation, and a 3D version of Eq. (20) for the one-way advection equation. Fig. 8 shows snapshots in time for the solutions of the one-way advection equation and the two-way advection equation. Dissipation errors are considerably more pronounced for the solution of the one-way advection equation where significant volume loss can be observed. The volume errors at final time $t = 3$ are shown in Table 3 for the solution of the one-way advection equation discretized using WENO schemes of various orders and the two-way advection equation solution discretized using the SOWMAC scheme. The volume loss for the solution of the two-way advection equation is only 2.6%, which compares well with accurate Eulerian–Lagrangian approaches. Indeed, Enright et al. [4] reported an almost identical volume loss for the same test using a particle level set method known to be less prone to dissipation errors than classical Eulerian approaches. On the other hand, the volume loss associated with the solution of the one-way advection equation is five to twenty times larger. Note that there are a number of corrections that can be applied to further improve or to enforce volume conservation.

Therefore, the straightforward extension of the two-way advection equation to three-dimensional space also yields considerably more accurate results than the solution of the one-way advection equation, with volume errors that are roughly one order of magnitude smaller for the two-way advection equation.

5.5. Computational cost

The associated computational cost to the one-way advection equation and two-way advection equation approach in 1D or in multidimensional space is expected to show a linear increase with the number of unknowns. This assumption can be checked in Fig. 9 that represents the execution time necessary to compute 1000 advection time steps, t_e, as a function of the number of unknowns N in 1D or in 3D domains. In the latter case, we consider a domain of aspect ratio 1 discretized using cubic cells. WENO schemes of different order were chosen for the spatial discretization of the one-way advection equation, while the two-way advection equation is discretized using an implicit SOWMAC scheme. To remain consistent with the second-order temporal accuracy of the two-way advection equation, the one-way advection equation was integrated in time using a second-order TVD explicit Runge–Kutta scheme. The calculations were performed on a standard desktop station using 2.4 GHz Xeon CPUs. Constants involved in each numerical scheme were precomputed is order to further reduce the execution time and standards optimizing options were used for compilation.

As expected, for $N > 10^4$ the execution time increases quasi-linearly with N. The average ratios $t_e^{\text{one-way}}/t_e^{\text{two-way}}$, listed in Table 4 for 1D and 3D domains, show that performing a two-way advection equation is at least 2.7 to 6.6 times faster than solving for the one-way advection equation with a scheme of comparable effective accuracy (*e.g.*, fifth-order WENO, and higher for longer time integration). Note that in the multidimensional case, the computational cost associated with the

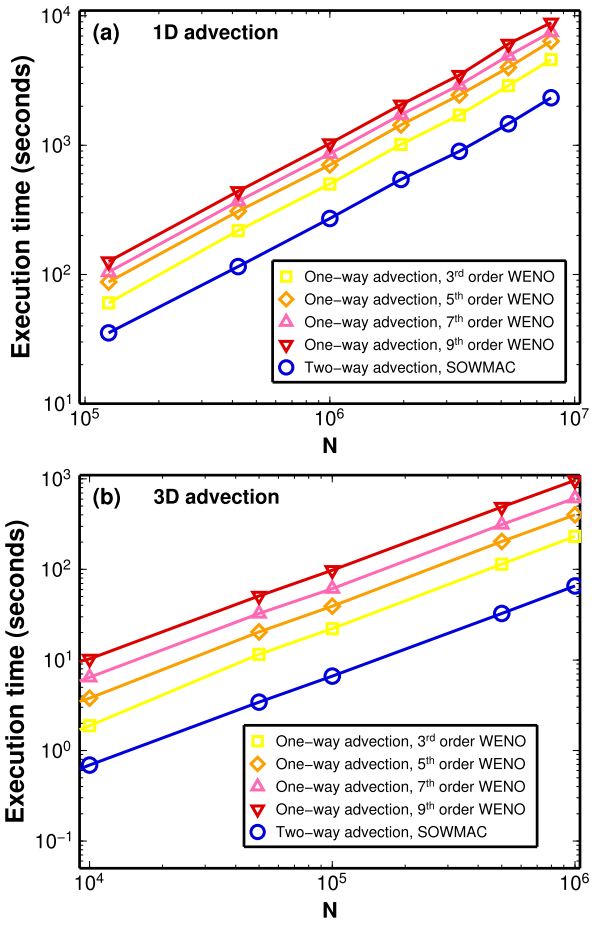

Fig. 9. Execution time t_e as a function of the number of unknowns N to perform 1000 time steps by solving the one-way advection equation or the two-way advection equation. The one-way advection equation is integrated with a second-order explicit TVD Runge–Kutta scheme and is discretized spatially using different WENO schemes. The two-way advection equation is discretized using a second-order SOWMAC scheme [10] leading to a tri-diagonal implicit scheme. The slopes t_e/N are close to unity.

Table 4
Average ratio $t_e^{\text{one-way}}/t_e^{\text{two-way}}$ of the execution time required to solve the one-way advection equation with different spatial discretizations to the execution time required to solve the two-way advection equation with a SOWMAC scheme of comparable temporal truncation error.

Scheme	$t_e^{\text{one-way}}/t_e^{\text{two-way}}$ (1D)	$t_e^{\text{one-way}}/t_e^{\text{two-way}}$ (3D)
RK2-WENO3	3.4	1.9
RK2-WENO5	6.6	2.7
RK2-WENO7	11.0	3.2
RK2-WENO9	18.0	3.8

two-way advection equation discretization will improve over the one-way advection equation discretizations with increasing the domain aspect ratio. In such cases, the ratio $t_e^{\text{one-way}}/t_e^{\text{two-way}}$ is expected to converge towards the value measured for 1D domains.

In addition, it should be noted that the two-way advection equation discretized using the SOWMAC scheme yields narrower stencils, which would improve the parallelization and the corresponding efficiency compared to the one-way advection equation discretized using WENO schemes that involve wider stencils.

6. Conclusion

We have considered the use of a two-way advection equation as an alternative to the one-way advection equation for level set interface tracking. Both theoretical considerations and tests have shown that the use of the two-way advection equation yields more accurate results than the discretization of the one-way advection equation using centered FD stencils or more stable WENO schemes of much higher order. Therefore, if the quantity to be advected is sufficiently smooth, the

two-way advection equation is a better alternative to the one-way advection equation, both for accuracy and computational cost. This is particularly true when the advection time is long because the two-way advection equation can be discretized using schemes that are less prone to dissipation errors than popular asymmetric schemes. Moreover, the requirement of advecting a smooth quantity to ensure stability naturally suggests the use of the two-way advection equation embedded in the level set method for Eulerian interface tracking. In this case, the quantity to advect remains smooth by construction, which prevents the development of dispersion errors. While both the two-way wave advection [24,25] and the associated numerical schemes (*i.e.*, the SOWMAC [10]) were derived previously, these approaches have never been combined with level set advection.

We have conducted advection tests in spatially varying velocity fields, and compared the performances of the level set advection using the standard one-way advection equation discretized using popular fifth-order WENO reconstruction with the two-way equation discretized using a second-order scheme. The results show that the use of the two-way advection equation yields significantly more accurate results for the level set advection, at a considerably smaller computational cost.

A possible improvement of the present the level set approach would be to use the two-way advection equation to enforce the Eikonal requirement of the level set, which would reduce the associated numerical errors and the computational cost. Since the evolutionary PDE for the level set reinitialization equation (4) can be recast as an advection equation, it may be possible to solve it using the two-way advection equation instead. This could be investigated in the future.

Although the present study restricts the use of the two-way advection equation to smooth level sets, future effort will be spent on extending the use of the two-way advection equation to advection of arbitrary scalar fields including those containing discontinuities. Possible ways of achieving this generalization will be the use of flux corrections, and/or a dynamical switch between the one-way advection equation and the two-way advection equation depending on the local smoothness of the quantity to advect.

Acknowledgements

I thank three anonymous reviewers for their thorough comments that led to a significant improvement of this work. I also acknowledge the funds from the *Stifterverband für die Deutsche Wissenschaft*, from the *Centre de Coopération Universitaire Franco–Bavarois*, and from the *Institut National des Sciences de l'Univers* (*CNRS*) via the *Observatoire Midi-Pyrénées* (project ADVECT). Figures were made with the Generic Mapping Tools (P. Wessel and W.H.F Smith, EOS, Trans. AGU 76 (1995) 329), and with the ParaView software.

Appendix A. Errors associated with the spatial discretization of the one-way wave advection equation

A.1. Fifth-order WENO discretization

The three possible third-order polynomials are:

$$p_1^{\pm} = +\frac{\delta_{j-2}^{\pm}}{3} - \frac{7\delta_{j-1}^{\pm}}{6} + \frac{11\delta_j^{\pm}}{6},$$
$$p_2^{\pm} = -\frac{\delta_{j-1}^{\pm}}{6} + \frac{5\delta_j^{\pm}}{6} + \frac{\delta_{j+1}^{\pm}}{3},$$
$$p_3^{\pm} = +\frac{\delta_j^{\pm}}{3} + \frac{5\delta_{j+1}^{\pm}}{6} - \frac{\delta_{j+2}^{\pm}}{6}$$

and linear weights are:

$$\gamma_1 = \frac{1}{10}, \qquad \gamma_2 = \frac{6}{10}, \qquad \gamma_3 = \frac{3}{10}.$$

Therefore, the associated frequency is:

$$\tilde{\omega}_{\mathrm{WENO5}} = \frac{1}{420\Delta x}\left[28e^{-3ik\Delta x} - 126e^{-2ik\Delta x} + 420e^{-ik\Delta x} - 105 - 252e^{ik\Delta x} + 42e^{2ik\Delta x} - 4e^{3ik\Delta x} - 3e^{-4ik\Delta x}\right].$$

Using Eqs. (42) together with the above expression yields the corresponding expressions for the frequency and amplitude errors:

$$E_{\omega}^{\mathrm{WENO5}} = \frac{1}{420k\Delta x}\left[672\sin(k\Delta x) - 168\sin(2k\Delta x) + 32\sin(3k\Delta x) - 3\sin(4k\Delta x)\right] - 1,$$
$$E_{a}^{\mathrm{WENO5}} = 1 - \exp\left\{\frac{ut}{140\Delta x}\left[-35 + 56\cos(k\Delta x) - 28\cos(2k\Delta x) + 8\cos(3k\Delta x) - \cos(4k\Delta x)\right]\right\}.$$

A.2. Seventh-order WENO discretization

The four possible fourth-order interpolations are:

$$p_1^{\pm} = -\frac{1}{4}\delta_{j-3}^{\pm} + \frac{13}{12}\delta_{j-2}^{\pm} - \frac{23}{12}\delta_{j-1}^{\pm} + \frac{25}{12}\delta_j^{\pm},$$
$$p_2^{\pm} = +\frac{1}{12}\delta_{j-2}^{\pm} - \frac{5}{12}\delta_{j-1}^{\pm} + \frac{13}{12}\delta_j^{\pm} + \frac{1}{4}\delta_{j+1}^{\pm},$$
$$p_3^{\pm} = -\frac{1}{12}\delta_{j-1}^{\pm} + \frac{7}{12}\delta_j^{\pm} + \frac{7}{12}\delta_{j+1}^{\pm} - \frac{1}{12}\delta_{j+2}^{\pm},$$
$$p_4^{\pm} = +\frac{1}{4}\delta_j^{\pm} + \frac{13}{12}\delta_{j+1}^{\pm} - \frac{5}{12}\delta_{j+2}^{\pm} + \frac{1}{12}\delta_{j+3}^{\pm}.$$

The linear weights are:

$$\gamma_1 = \frac{1}{35}, \qquad \gamma_2 = \frac{12}{35}, \qquad \gamma_3 = \frac{18}{35}, \qquad \gamma_4 = \frac{4}{35}.$$

Therefore, the associated frequency is:

$$\tilde{\omega}_{\mathrm{WENO7}} = \frac{1}{2520\Delta x} i\big[-45e^{-4ik\Delta x} + 240e^{-3ik\Delta x} - 840e^{-2ik\Delta x} + 2520e^{-ik\Delta x} - 504 - 1680e^{ik\Delta x} + 360e^{2ik\Delta x} - 60e^{3ik\Delta x} + 5e^{4ik\Delta x} + 4e^{-5ik\Delta x}\big].$$

The corresponding expressions for the frequency and amplitude errors are:

$$E_{\omega}^{\mathrm{WENO7}} = \frac{1}{2520k\Delta x}\big[4\sin(5k\Delta x) + 4200\sin(k\Delta x) - 50\sin(4k\Delta x) - 1200\sin(2k\Delta x) + 300\sin(3k\Delta x)\big],$$
$$E_{a}^{\mathrm{WENO7}} = 1 - \exp\left\{\frac{ut}{630\Delta x}\big[-10\cos(4k\Delta x) + 45\cos(3k\Delta x) - 120\cos(2k\Delta x) + 210\cos(k\Delta x) - 126 + \cos(5k\Delta x)\big]\right\}.$$

References

[1] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (1999) 2–22.
[2] D.L. Chopp, Some improvements of the fast marching method, SIAM J. Sci. Comput. 23 (2002) 230–244.
[3] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer, New York, USA, 1999.
[4] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (1) (2002) 83–116.
[5] D. Enright, F. Lossaso, R. Fedkiw, A fast and accurate semi-Lagrangian particle level set method, Colloids Surf. 83 (2005) 479–490.
[6] R. Fedkiw, B. Merriman, S. Osher, Simplified upwind discretizations of systems of hyperbolic conservation laws containing advection equations, J. Comput. Phys. 157 (2000) 302–326.
[7] B. Fornberg, Calculation of weights in finite difference formulas, SIAM Rev. 40 (1998) 685–691.
[8] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.
[9] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[10] T. Komatsu, K. Ohgushi, K. Asai, Refined numerical scheme for advective transport in diffusion simulation, J. Hydraul. Eng. 123 (1997) 41–50.
[11] K.Y. Lervåg, Å. Ervik, Curvature calculations for the level-set method, in: Proceedings, ENU-MATH 2011, Springer, 2013.
[12] R.J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627–665.
[13] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[14] Y. Liu, M.K. Sen, An implicit staggered-grid finite-difference method for seismic modelling, Geophys. J. Int. 179 (2009) 459–474.
[15] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, USA, 2003.
[16] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[17] J.-M. Qiu, C.-W. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, J. Comput. Phys. 230 (2011) 863–889.
[18] H. Samuel, M. Evonuk, Modeling advection in geophysical flows with particle level sets, Geochem. Geophys. Geosyst. 11 (2010) Q08020, http://dx.doi.org/10.1029/2010GC003081.
[19] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
[20] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[21] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[22] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas–Liquid Multiphase Flows, Cambridge University Press, Cambridge, 2011.
[23] R.F. Warming, R.M. Beam, Upwind second-order differences schemes and applications in aerodynamic flows, AIAA J. 14 (1976) 1241–1249.
[24] J. Wu, Wave equation model for solving advection-diffusion equation, Int. J. Numer. Methods Fluids 37 (1994) 2717–2733.
[25] J. Wu, A wave equation model to solve the multidimensional transport equation, Int. J. Numer. Methods Fluids 24 (1997) 423–439.
[26] S. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.