Journal of Computational Physics 272 (2014) 1–22

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An improved bounded semi-Lagrangian scheme for the turbulent transport of passive scalars

Siddhartha Verma[a,*], Y. Xuan[a], G. Blanquart[b]

[a] Graduate Aerospace Laboratories, California Institute of Technology, United States
[b] Department of Mechanical Engineering, California Institute of Technology, United States

ARTICLE INFO

Article history:
Received 26 June 2013
Received in revised form 18 March 2014
Accepted 31 March 2014
Available online 16 April 2014

Keywords:
Semi-Lagrangian
Bounded
Oscillationless
Cubic Hermite
Monotone
Turbulent mixing
Passive scalar
Mixing layer

ABSTRACT

An improved bounded semi-Lagrangian scalar transport scheme based on cubic Hermite polynomial reconstruction is proposed in this paper. Boundedness of the scalar being transported is ensured by applying derivative limiting techniques. Single sub-cell extrema are allowed to exist as they are often physical, and help minimize numerical dissipation. This treatment is distinct from enforcing strict monotonicity as done by D.L. Williamson and P.J. Rasch [5], and allows better preservation of small scale structures in turbulent simulations. The proposed bounding algorithm, although a seemingly subtle difference from strict monotonicity enforcement, is shown to result in significant performance gain in laminar cases, and in three-dimensional turbulent mixing layers. The scheme satisfies several important properties, including boundedness, low numerical diffusion, and high accuracy. Performance gain in the turbulent case is assessed by comparing scalar energy and dissipation spectra produced by several bounded and unbounded schemes. The results indicate that the proposed scheme is capable of furnishing extremely accurate results, with less severe resolution requirements than all the other bounded schemes tested. Additional simulations in homogeneous isotropic turbulence, with scalar timestep size unconstrained by the CFL number, show good agreement with spectral scheme results available in the literature. Detailed analytical examination of gain and phase error characteristics of the original cubic Hermite polynomial is also included, and points to dissipation and dispersion characteristics comparable to, or better than, those of a fifth order upwind Eulerian scheme.

1. Introduction

The need for high accuracy at small scales makes turbulent simulations extremely sensitive to numerical errors. This presents considerable challenges when conducting practical simulations of important turbulent phenomena such as weather modelling [1–3], and tracking of pollutants in the atmosphere and oceans. Turbulent transport of scalars is especially susceptible to numerical errors, and imposes stringent grid resolution requirements [4]. To alleviate the computational cost involved in turbulent scalar transport, we propose a novel bounding algorithm for cubic Hermite polynomial-based semi-Lagrangian schemes [5,6]. Numerical performance of the scheme is compared to that of several bounded and unbounded schemes, with regard to the following scheme properties

[*] Corresponding author.

http://dx.doi.org/10.1016/j.jcp.2014.03.062
0021-9991/© 2014 Elsevier Inc. All rights reserved.

- Boundedness
- Low numerical diffusion
- High accuracy
- Conservation

Of the four properties listed, scalar boundedness is a crucial property that must be respected when required by physical constraints. Almost all transported scalars, both passive and active, have physical bounds that must be maintained during transport. Violating boundedness in certain cases can lead to severe detrimental effects. Some examples include irreversible spurious precipitation in weather simulations [5], and the appearance of negative absolute temperature and unphysical species mass fractions (*i.e.*, outside the bounds $[0, 1]$) in reacting flows.

Due to the critical nature of scalar boundedness in such simulations, there has been significant effort in the past to construct bounded finite-volume (*e.g.*, WENO3 [7], MPWENO5 [8], OSMP7 [9], BQUICK [10]) schemes based on the Eulerian treatment [11] of the scalar transport equation. Unfortunately, these transport schemes exhibit significant numerical diffusion, which leads to a decrease in simulation accuracy. Higher order schemes coupled with a Flux Corrected Transport (FCT) algorithm [12,13] can mitigate the issue of numerical diffusion. However, the resulting increase in accuracy comes at a significant increase in computational cost, owing to the need for a larger computational stencil. The use of high order finite-volume schemes is also accompanied with increasingly restrictive stability criteria (*i.e.*, a smaller limiting Courant–Friedrichs–Lewy (CFL) number) [1], which leads to further increase in computational cost. In addition, high-order schemes give rise to larger and more frequent oscillations (Runge's phenomenon), which makes ensuring boundedness difficult. The proposed Bounded Cubic Hermite polynomial (BCH) transport scheme, which is based on the semi-Lagrangian treatment of the advection–diffusion equation [1], addresses several of these numerical issues.

As an alternative to Eulerian schemes, semi-Lagrangian (SL) schemes are known to be stable at large CFL values, but suffer from the inability to conserve mass inherently [1]. This may not be an issue of major concern in certain scenarios [5]. The ability of SL schemes to maintain stability for $\text{CFL} > 1$ makes them especially suitable for applications involving long integration times. This has led to widespread adoption of SL schemes in the atmospheric community, and a vast body of work exists in the literature [1,5,6,14–19]. An exhaustive review in this regard can be found in Ref. [1]. In addition to decreased restriction on time-step size, SL schemes can yield enhanced spatial accuracy, which makes them suitable for use in interface tracking in multiphase flows [16], and for turbulent simulations, which is the primary motivation for the current work.

As already discussed, simulations often require that the scalar being transported respects physical bounds, and the use of high-order interpolation can violate this constraint (Godunov's theorem [20]). Algorithms that restore boundedness can introduce numerical dissipation, and minimizing this is one of the main goals of the proposed BCH scheme. Most existing SL schemes that make an effort to maintain scalar boundedness are based on the use of either Lagrange polynomials [14,15] or Hermite polynomials [5,6] for interpolation. The form of the polynomial used, in addition to the bounding mechanism used, can have a measurable impact on the accuracy of the simulation [6]. This will become evident from the results presented in Section 5.

The objective of this paper is to provide a description of the BCH scheme, and to test its performance with regard to the four properties listed earlier. Performance evaluation of certain existing semi-Lagrangian schemes [6,14,15] that make an effort to maintain scalar boundedness is also included, to assess the improvement that the proposed scheme offers. A brief discussion of the basic workings of semi-Lagrangian schemes is outlined in Section 2. The formulation of the proposed bounding algorithm is presented in Section 3. Section 4 presents an in-depth analysis of gain and phase error characteristics of the original cubic Hermite polynomial, for uniform one dimensional (1D) advection of a passive scalar. Section 5 provides configuration description of the numerical tests used, which include three laminar cases and two turbulent cases. One of the turbulent test cases is used to assess numerical performance in comparison to spectral schemes, when working with large timesteps unconstrained by the CFL number. Section 5 also involves discussion of scheme performance with regard to the scheme properties listed earlier, in addition to comparison with several other commonly used Eulerian and semi-Lagrangian schemes.

2. Semi-Lagrangian schemes

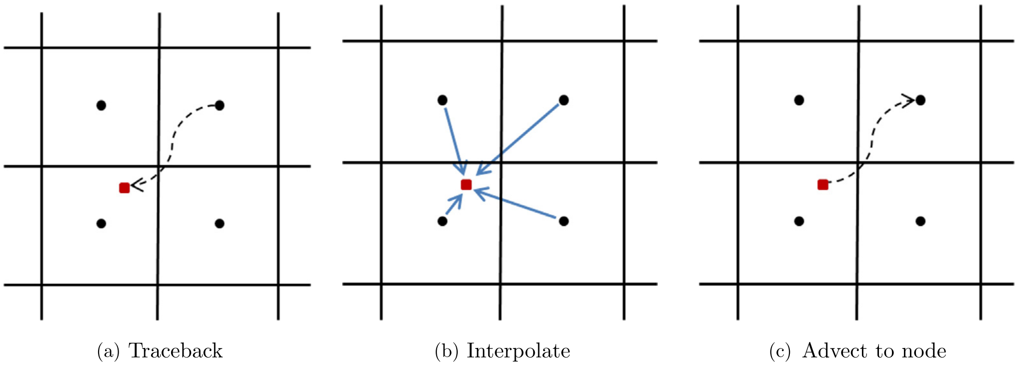
This section provides a brief outline of the numerical implementation of semi-Lagrangian schemes. Readers interested in the proposed bounding algorithm may proceed directly to Section 3.

2.1. Numerical implementation

The transport of passive scalars is governed by the advection–diffusion equation

$$\frac{Dz}{Dt} = \frac{\partial z}{\partial t} + \boldsymbol{u} \cdot \nabla z = \mathcal{D} \nabla^2 z \tag{1}$$

where the transported scalar quantity is denoted by 'z', $\mathcal{D}$ is the molecular diffusivity, and the $\frac{D}{Dt}$ operator represents the material derivative. All numerical tests used in this paper assume the incompressibility condition ($\nabla \cdot \boldsymbol{u} = 0$). To solve the


(a) Traceback

(b) Interpolate

(c) Advect to node

Fig. 1. Basic steps involved in semi-Lagrangian transport. Location $\boldsymbol{x}^n$ in Eq. (4) corresponds to the red square shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

discretized form of Eq. (1), we employ a fractional step method (Godunov splitting [11]), whereby the solution procedure is split into two steps: pure advection followed by diffusion. Using the superscript ($\star$) to signify an intermediate solution associated with the advection step only, we get

$$\frac{Dz^\star}{Dt} = 0 \tag{2}$$

Eq. (2) implies that under pure advection, the scalar value associated with a Lagrangian particle remains unchanged. In the absence of source terms, and for non-negative molecular diffusivity, this prohibits the formation of new maxima and minima, as pointed out by Williamson and Rasch [5].

The advection step denoted by Eq. (2) is accomplished in semi-Lagrangian schemes via a sequence of three steps: traceback, interpolation, and advection to node (Fig. 1). Semi-Lagrangian schemes are different from Eulerian and Lagrangian schemes in that they combine particle tracking (as done in Lagrangian schemes) with a static computational grid (generally used with Eulerian schemes). Tracking scalar particles in a fully Lagrangian fashion is not very efficient computationally, especially when using parallelized algorithms. The semi-Lagrangian framework adopts a workable compromise by employing a fixed grid, and tracking particles associated with the grid nodes.

The scalar value at any grid node 'i' is updated by first tracking the relevant material particle backward in time (Fig. 1(a)). Once this location, $\boldsymbol{x}^n$, has been determined, the scalar value at that location ($z_i^\star = z^n(\boldsymbol{x}^n)$) is computed using a suitable interpolation procedure (Fig. 1(b)). This intermediate scalar value is then assigned 'as-is' to the grid node in accordance with Eq. (2) (Fig. 1(c)). At this point, the advection step is complete. The diffusion term (as well as any other source terms present) may be accounted for as follows

$$z^{n+1} = z^\star + \Delta t\big(\mathcal{D}\nabla^2 z^\star\big) \tag{3}$$

to obtain the final value z_i^{n+1}. We now discuss each of the three basic steps of semi-Lagrangian transport in detail.

2.2. Traceback

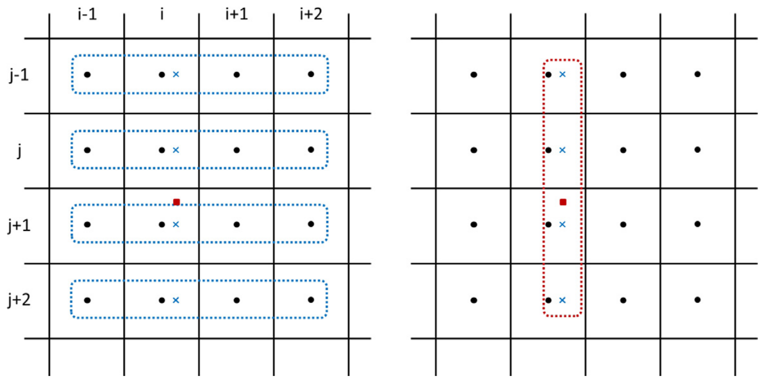
For the traceback step depicted in Fig. 1(a), we track the location of a material particle that coincides with a grid node at time t^{n+1}, to its location $\boldsymbol{x}^n$ at the previous timestep t^n

$$\boldsymbol{x}^n = \boldsymbol{x}^{n+1} - \int\limits_n^{n+1} \boldsymbol{u}\big(\boldsymbol{x}(t), t\big)\, dt \tag{4}$$

This step is equivalent to the computation of curved characteristic lines emanating from the grid nodes. An appropriate numerical integrator must be used for solving Eq. (4) since the velocity field $\boldsymbol{u}$ can vary temporally as well as spatially. For the current work, a second order Runge–Kutta (RK2) integrator was used. Intermediate velocity values were computed using linear interpolation of cell face velocity components. We point out that the traceback step is completely independent of the passive scalar being transported. Thus, the $\boldsymbol{x}^n$ values need to be computed just once per timestep, even if transporting several distinct scalars simultaneously.

2.3. Interpolation

The interpolating polynomial used for the step shown in Fig. 1(b), along with any boundedness preserving corrections, forms the core of any semi-Lagrangian transport scheme. For comparison purposes, we will discuss three interpolation procedures, namely, the proposed Bounded Cubic Hermite polynomial (BCH), the fully Monotonic Cubic Hermite polynomial



(a) Step 1: 4 interpolations in x

(b) Step 2: 1 interpolation in y

Fig. 2. Dimensional splitting to decompose 2D interpolation into a cascade of 1D interpolations.

(MCH) [6], and a bounded cubic Lagrange polynomial (SL3) [14,15]. 3D interpolation in all cases is done via a successive 'cascade' of 1D interpolations, based on the 'tensor product' approach [5]. Interpolation in this manner is much more efficient computationally when compared to constructing interpolating surfaces. The cascading procedure is illustrated for a 2D case in Fig. 2. One of the directions, for instance x, is selected for the first set of 1D interpolations. The scalar is interpolated at the locations marked by the blue '×'s for each of the four rows $j-1$ through $j+2$ (Fig. 2(a)). The second step uses these four interpolated values ($z_{\times_{j-1}}, z_{\times_j}, z_{\times_{j+1}}$, and $z_{\times_{j+2}}$) to compute $z_i^\star$ (Fig. 2(b)) by interpolating in the orthogonal direction (y).

One known issue of interpolating with polynomials of order greater than 1 is that these give rise to spurious oscillations (Godunov's theorem). Large over/undershoots are present close to sharp gradients due to the inability of the limited resolution grid to resolve high wavenumber modes (Gibb's phenomenon). This can be especially problematic in simulations that involve abrupt transitions or rapid fluctuations in the scalar field, for instance at fuel-oxidizer interfaces in non-premixed combustion, and with passive transport of low diffusivity scalars. The oscillations may cause the transported scalar to breach appropriate bounds, resulting in unphysical simulation data. High order scalar transport schemes must, therefore, encompass some means of detecting and correcting these unphysical oscillations. Details of such correction mechanisms in the bounded 1D interpolation algorithms for both SL3 and BCH are given in Section 3.

2.4. *Incorporating the diffusive fluxes and other source terms*

The algorithm discussed up until now accounts for just the advection step of Godunov splitting, *i.e.*, determining $z^\star$ (Fig. 1(c)). The diffusive fluxes, in addition to any source or sink terms in the material transport equation (Eq. (1)), are incorporated into the final update of the scalar value using Eq. (3). Although the diffusive term is implemented in a fully explicit fashion in this work, it is possible to use implicit formulations of Eq. (3) involving z^{n+1}.

In the current implementation, a second order accurate finite difference formula is used for the discretization of the Laplacian operator (∇^2) in Eq. (3). A fourth order accurate discretization was ruled out since it gave rise to unphysical oscillations close to velocity stagnation points and in regions with sharp scalar gradients. To maintain second order accuracy on non-uniform grids, the discrete operator is implemented as a combination of the divergence operator acting on a gradient operator ($\nabla^2 z = \nabla \cdot (\nabla z)$).

$$(\nabla z_{i+\frac{1}{2}})_x = \frac{z_{i+1} - z_i}{x_{i+1} - x_i} \tag{5}$$

$$\left(\nabla^2 z_i\right)_x = \frac{(\nabla z_{i+\frac{1}{2}})_x - (\nabla z_{i-\frac{1}{2}})_x}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \tag{6}$$

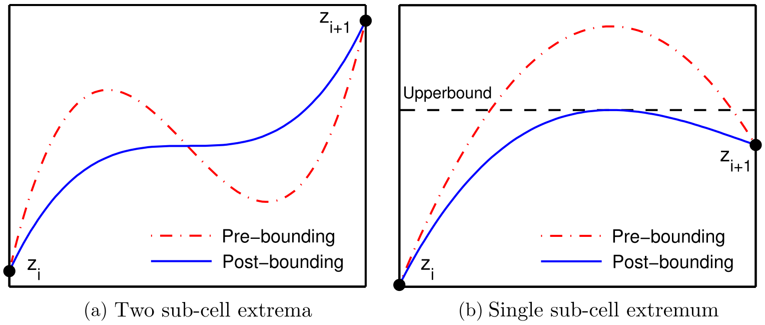
$$\nabla^2 z = \left(\nabla^2 z_i\right)_x + \left(\nabla^2 z_i\right)_y + \left(\nabla^2 z_i\right)_z \tag{7}$$

Eq. (6) reduces to the more familiar form $(\nabla^2 z_i)_x = \frac{z_{i+1} - 2z_i + z_{i-1}}{\Delta x^2}$ when working with uniform grids. Incorporation of the source terms, as outlined here, completes the scalar transport step for semi-Lagrangian schemes.

3. Bounded interpolating polynomials

3.1. *Bounded Cubic Hermite (BCH) polynomial*

The BCH scheme makes use of Hermite polynomials, which differ from Lagrange polynomials in that they require knowledge of scalar derivatives (z_i', z_{i+1}') in addition to scalar values (z_i, z_{i+1}) at the interpolating nodes. The use of derivatives



(a) Two sub-cell extrema (b) Single sub-cell extremum

Fig. 3. Initial and final profiles for bounding in (a) Case 1; (b) Case 2.

as constraints results in an enhanced ability to resolve sub-cell shape [5,16], which in turn leads to a marked increase in interpolation accuracy. In the current formulation, scalar derivatives are computed using 2nd order central difference (CD2), which is equivalent to the 'arithmetic mean' derivative estimation mentioned in Ref. [6]. The appropriate interpolation coefficients are given below, using Fig. 2 as reference for the spatial indices. Subscript '$i+\frac{1}{2}$' indicates the cell face located in the middle of node locations 'i' and '$i+1$'.

$$w_i = 1 - 3\left(\frac{x_\times - x_i}{\Delta x}\right)^2 + 2\left(\frac{x_\times - x_i}{\Delta x}\right)^3 \tag{8}$$

$$w_{i+1} = 1 - w_i \tag{9}$$

$$w'_i = (x_\times - x_i)\left(1 - 2\frac{x_\times - x_i}{\Delta x} + \left(\frac{x_\times - x_i}{\Delta x}\right)^2\right) \tag{10}$$

$$w'_{i+1} = \frac{(x_\times - x_i)^2}{\Delta x}\left(-1 + \frac{x_\times - x_i}{\Delta x}\right) \tag{11}$$

$$z'_i = \frac{z_i - z_{i-1}}{x_i - x_{i-1}} \cdot \frac{x_{i+\frac{1}{2}} - x_i}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} + \frac{z_{i+1} - z_i}{x_{i+1} - x_i} \cdot \frac{x_i - x_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \tag{12}$$

$$z'_{i+1} = \frac{z_{i+1} - z_i}{x_{i+1} - x_i} \cdot \frac{x_{i+\frac{3}{2}} - x_{i+1}}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}} + \frac{z_{i+2} - z_{i+1}}{x_{i+2} - x_{i+1}} \cdot \frac{x_{i+1} - x_{i+\frac{1}{2}}}{x_{i+\frac{3}{2}} - x_{i+\frac{1}{2}}} \tag{13}$$

$$z_\times = z_i \cdot w_i + z_{i+1} \cdot w_{i+1} + z'_i \cdot w'_i + z'_{i+1} \cdot w'_{i+1} \tag{14}$$

For uniform grids, the derivatives reduce to the familiar CD2 form $z'_i = \frac{z_{i+1} - z_{i-1}}{2\Delta x}$.

Boundedness (a term which we take to mean that the interpolated scalar value respects physical bounds, as opposed to requiring strict sub-cell *monotonicity*) is enforced in all of the cascading 1D interpolations by modifying the scalar derivatives at the interpolating nodes (z'_i and z'_{i+1}). For certain configurations (*e.g.*, the laminar and turbulent simulations discussed in Sections 5.3 and 5.4), the physical bounds are known with certainty, and lower and upper bounds can be specified accordingly. However, for other simulations, the concept of physical scalar bounds may not be applicable (*e.g.*, the homogeneous isotropic turbulent simulations discussed in Section 5.5), in which case the bounding operation associated with single-extremum cases (described below) is limited to the advection step, to prevent the generation of spurious mathematical extrema.

The cubic polynomial can give rise to two distinct scenarios, depending on the sign of the interpolating derivatives. Either a single extremum, or two extrema may exist within the interpolating cell (Fig. 3), and each case is treated differently as explained below.

3.1.1. Case 1: Two sub-cell extrema

If the two interpolating derivatives have the same sign ($z'_i z'_{i+1} > 0$), there may exist two sub-cell extrema in between nodes i and $i+1$ (Fig. 3(a)). The high wavenumber modes involved in this case cannot be resolved by the computational grid (Nyquist criterion), and their existence is caused by the presence of artificial numerical oscillations. To avoid the formation of two extrema, the interpolating cubic polynomial is forced to be monotone in this case, irrespective of whether or not the interpolated value breaches physical bounds.

To enforce monotonicity in the cubic Hermite polynomial, we use a derivative limiting procedure that is well-documented in the literature [21–23]. Constructing a monotonic cubic interpolant requires that both z'_i and z'_{i+1} have the same sign as the limiting derivative, z'_{limit}, and be smaller in magnitude than the same.

$$z'_{limit} = 3\frac{(z_{i+1} - z_i)}{\Delta x} \tag{15}$$

The interpolating derivatives are modified to enforce compliance with these conditions using a 'minmod' limiter, so as to minimize the change engendered.

$$z'_i = \mathbf{minmod}(z'_i, z'_{limit}) \tag{16}$$

$$z'_{i+1} = \mathbf{minmod}(z'_{i+1}, z'_{limit}) \tag{17}$$

The minmod limiter is defined conventionally as follows [23]

$$\mathbf{minmod}(a,b) = \begin{cases} \mathbf{sign}(a) \cdot \mathbf{min}(\mathbf{abs}(a,b)) & \text{if } ab > 0 \\ 0 & \text{if } ab < 0 \end{cases} \tag{18}$$

The interpolation profiles before and after enforcing monotonicity for the case of two sub-cell extrema are shown in Fig. 3(a).

3.1.2. Case 2: Single sub-cell extremum

In case the two interpolating derivatives have opposite signs, there exists a single extremum in between nodes i and $i+1$ (Fig. 3(b)). Allowing sub-cell extrema to exist is not only *physically* plausible (they may have been missed by the grid upon discretization), but is *essential* for minimizing numerical dissipation, as will be shown in Sections 5.3 and 5.4. The treatment of these local extrema is also one of the major differences with the strictly monotonic formulation proposed in Ref. [5]. Sub-cell extrema are especially important in turbulent simulations of low molecular diffusivity scalars, where small scale fluctuations play a vital role in overall transport. Schemes that enforce strict monotonicity in the interpolating polynomial [5,14] destroy these extrema altogether, resulting in loss of accuracy and significant numerical dissipation. This observation becomes evident from the discussion in Section 5.4.2, even for turbulent flows with relatively high molecular diffusivity scalar.

With a single extremum, if the interpolated scalar value breaches physical bounds, the extremum of the interpolating polynomial is also guaranteed to be unbounded (Fig. 3(b)). The objective then is to modify both z'_i and z'_{i+1} such that the *location* of the extremum remains unchanged, and the extremum *value* itself is equal to the appropriate bound. The modified interpolation profile is constructed as follows:

$$z_\times(x) = z_i \cdot w_i(x) + z_{i+1} \cdot w_{i+1}(x) + a \cdot z'_i \cdot w'_i(x) + b \cdot z'_{i+1} \cdot w'_{i+1}(x) \tag{19}$$

The extremum location (x_{ext}) is computed as the appropriate root of the derivative of Eq. (14), and 'a' and 'b' are determined using the following constraints:

$$z_\times(x_{ext}) = bound \tag{20}$$

$$\left.\frac{dz_\times}{dx}\right|_{x_{ext}} = 0 \tag{21}$$

To ensure that the nature of the extremum (*i.e.*, curvature sign) is not changed, both a and b are constrained to be positive. If either a or b is negative, strict monotonicity is enforced using the minmod limiter (Eqs. (15) through (18)).

3.2. Summary of the BCH scheme

Using the details of the bounding procedure for both Cases 1 and 2, the entire bounded interpolation algorithm for the BCH scheme can be summarized as follows:

1. Compute z'_i and z'_{i+1} using Eqs. (12) and (13).
2. If $z'_i \cdot z'_{i+1} \geq 0$, force monotonicity via Eqs. (15) through (18).
3. Compute $z_\times$ (Eq. (14)).
4. If $z_\times \notin [lowerbound, upperbound]$ (which implies $z'_i \cdot z'_{i+1} < 0$), modify the profile using Eqs. (19) through (21). If either a or $b < 0$, force monotonicity via Eqs. (15) through (18).

The algorithm described in Sections 3.1 and 3.2, specifically as it pertains to a single sub-cell extremum, is a key contribution of this paper. An alternative to using the procedure described above is to enforce strict monotonicity (Eqs. (15) through (18)) for *all* 1D polynomials in the interpolating cascade. Doing so gives us a scheme similar to one discussed by Rasch and Williamson in Ref. [6], which we will refer to as the MCH (Monotone Cubic Hermite) interpolation scheme for convenience. As will become evident from results presented in Section 5, using the BCH scheme leads to significant improvement over MCH in turbulent simulations, with regard to numerical dissipation.

3.3. Bounded Lagrange polynomial (SL3)

Bounded forms of the Lagrange polynomial are generally used by the climate modelling community for transporting moisture in air [5]. We use one such algorithm from Ref. [15] to compare performance of the BCH scheme with other bounded semi-Lagrangian schemes. The algorithm selected attempts to mirror, as closely as possible, the *modus operandi* of bounding in the BCH scheme.

Cubic Lagrange polynomials are constructed using a cascade (as shown for a 2D case in Fig. 2) of 1D 4-point stencils:

$$w_i = \prod_{\substack{-1 \le k \le 2 \\ k \ne i}} \frac{(x_\times - x_k)}{(x_i - x_k)} \tag{22}$$

$$z_\times = \sum_{i=-1}^{2} w_i z_i \tag{23}$$

As discussed earlier, this cubic polynomial does not ensure scalar boundedness. Upon breaching bounds, the interpolating Lagrange polynomial can be modified to ensure boundedness either by switching locally to first-order linear interpolation, or by 'clipping' to the closest bound. For scheme comparison in this paper, we use the clipping procedure introduced by Bermejo et al. in [14], and improved upon by Sun et al. in [15]. This procedure subjects the interpolated scalar value to the following constraints:

$$f^- = \min(z_i, z_{i+1}), \qquad f^+ = \max(z_i, z_{i+1}) \tag{24}$$

$$z_\times = \begin{cases} f^-: & \text{if } z_\times < f^- \\ f^+: & \text{if } z_\times > f^+ \end{cases} \tag{25}$$

To minimize numerical dissipation, clipping is turned off if a single sub-cell extremum is detected via the following condition

$$(z_{i+2} - z_{i+1}) \cdot (z_i - z_{i-1}) < 0 \tag{26}$$

If such single sub-cell extremum breaches physical bounds, the interpolated value is clipped to the global maximum or minimum, as might be appropriate. When Eq. (26) does not hold true, a single cell contains two extrema. As discussed in Section 3.1, such a profile cannot be resolved by the computational grid (Nyquist criterion). In this situation, the interpolated scalar is clipped using Eq. (25).

We note that one of the advantages of using Hermite polynomials in BCH over Lagrange polynomials is that we can detect and correct for *all* spurious high wavenumbers (two sub-cell extrema), which may not be the case when using Lagrange polynomials. For instance, clipping via Eq. (25) will not be triggered if the computed scalar value at the interpolating location lies within the interval $[f^-, f^+]$, even if there exist two sub-cell extrema.

3.4. *Additional remarks*

3.4.1. *Inspiration*

The formulation of the BCH scheme was inspired by attempts to use the 'Constrained Interpolation Profile' scheme (CIP) [16] for turbulent mixing simulations. CIP involves transporting the scalar derivative values (in all three directions) as passive scalars themselves. Unfortunately, the scalar and its derivatives rapidly become inconsistent with one another in turbulent tests. This makes it necessary to recompute the derivatives using the scalar field every few timesteps. The derivative transport process was eventually discarded altogether in favor of computing the derivatives at each timestep using CD2, as discussed in Section 3.1. This resulted in considerably lower memory requirement of the algorithm, as well as an appreciable decrease in numerical complexity of the scheme.

3.4.2. *Derivative estimation*

Of the several finite-difference formulae used for estimating the scalar derivatives during the formulation of the BCH scheme (*e.g.*, 4th order finite difference (FD4), high order upwind differences, *etc.*), 2nd order central difference was seen to yield the best results. FD4 was observed to be too oscillatory, especially in regions close to velocity stagnation points and large scalar gradients, whereas the various upwind difference formulae introduced large dispersion errors.

3.4.3. $\mathcal{C}^0$ *continuity*

The bounding procedure in BCH makes the interpolating polynomial $\mathcal{C}^0$ continuous, since the value of the derivative used at a particular grid node may differ for interpolation in adjacent cells. $\mathcal{C}^1$ continuous forms of the monotone Hermite cubic polynomial were not examined since they are known to suffer from severe dispersion errors [5,6]. Furthermore, using $\mathcal{C}^1$ forms of the interpolating polynomial can cause minor local corrections (necessary to restore boundedness) to 'ripple out' instantaneously throughout the domain. This sort of behavior is usually undesirable in hyperbolic systems.

3.4.4. Cascade interpolation

We close this discussion by pointing out that the cascade procedure discussed in Section 2.3 is not commutative, when used with the bounded form of SL3 or BCH (the unbounded form of SL3, however, does commute). Nevertheless, neither the laminar tests nor the turbulent tests exposed any discernible differences when the ordering of the cascading direction was altered (*i.e.*, x–y–z, y–x–z, z–x–y, *etc.*).

4. Order of accuracy, gain, and phase error

Theoretical analysis of the BCH scheme involves determining the order of accuracy, and analyzing gain and phase error properties of the scheme. Analytical examination requires that we make certain simplifications, such as assuming uniform one-dimensional flow, and analyzing the reconstructed polynomial without bounding enforced. In general, there are relatively few instances of bounds being breached in the computational domain ($\lesssim 0.02\%$ in the turbulent mixing layer, Section 5.4), which makes the latter a reasonable simplification. Furthermore, we consider only the error associated with the interpolation step, and neglect any error inherent in the traceback step.

4.1. Order of accuracy

The order of accuracy of the cubic Hermite polynomial is determined by expressing $z^\star$ explicitly in terms of the grid node scalar values, using one of the 4-point 1D stencils shown in Fig. 2. A uniform grid with a constant 1D velocity field u is assumed, and all instances of $x_\times - x_i$ shown in Section 3.1 are replaced by their dependence on the constant CFL number σ ($\because\ x_\times - x_i = -u\Delta t = \sigma\Delta x,\ \ \sigma = |u|\Delta t/\Delta x$). With these simplifications, and the assumption of zero molecular diffusivity, the interpolating polynomial takes the following form:

$$z_i^{n+1} = z_i^\star = z_{i-1}^n w_{i-1} + z_i^n w_i + z_{i+1}^n w_{i+1} + z_{i+2}^n w_{i+2} \tag{27}$$

with

$$\begin{aligned} w_{i-1} &= \frac{-\sigma(\sigma-1)^2}{2} \qquad w_i = \frac{(\sigma-1)(3\sigma^2-2\sigma-2)}{2} \\ w_{i+1} &= \frac{\sigma(-3\sigma^2+4\sigma+1)}{2} \qquad w_{i+2} = \frac{\sigma^2(\sigma-1)}{2} \end{aligned} \tag{28}$$

and z^* being the intermediate solution associated with the advection step as described in Section 2.1 (Eq. (2)). The discretization error can be evaluated from Eq. (27) by expanding the scalar values at time step 'n' in their Taylor-series around $x_\times$ (*e.g.*, $z_{i+1} = z_i^\star + z_i^{\star\prime}(1-\sigma)\Delta x + z_i^{\star\prime\prime}(1-\sigma)^2\Delta x^2/2! + \cdots$). We must bear in mind that this type of analysis has limited range of applicability since a Taylor-series expansion assumes a sufficiently smooth function, which may not necessarily be valid for physically realistic cases [6]. Taking $z_i^\star$ to be the exact value at the interpolating location, the residual in Eq. (27) (after the relevant expansions) gives the discretization error at every time step:

$$\epsilon = \sigma(\sigma-1)(2\sigma-1)\frac{\Delta x^3}{3!} z_i^{\star\prime\prime\prime} - 9\sigma^2(\sigma-1)^2\frac{\Delta x^4}{4!} z_i^{\star\prime\prime\prime\prime} + \cdots \tag{29}$$

The total error accumulated after simulation time 'T' is computed as follows:

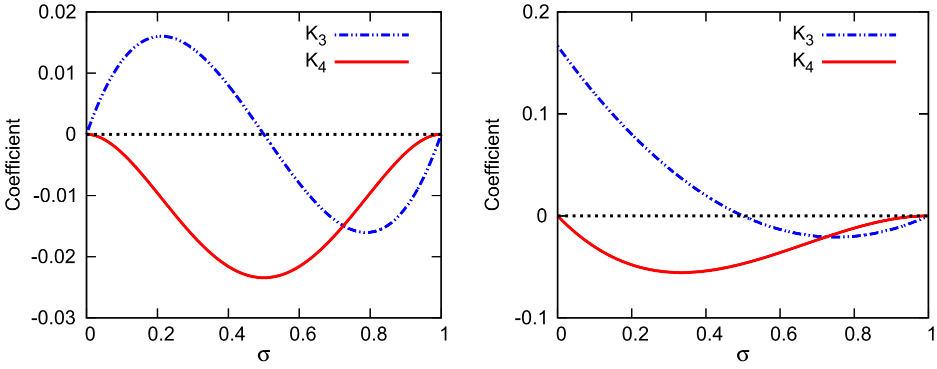
$$\epsilon_{tot} = \frac{T}{\Delta t}\cdot\epsilon = \frac{T\cdot|u|}{\sigma\Delta x}\cdot\epsilon \tag{30}$$

$$= T\cdot|u|\left[(\sigma-1)(2\sigma-1)\frac{\Delta x^2}{3!} z_i^{\star\prime\prime\prime} - 9\sigma(\sigma-1)^2\frac{\Delta x^4}{4!} z_i^{\star\prime\prime\prime\prime} + \cdots\right] \tag{31}$$

The leading term in Eq. (31) is second order accurate in space, and determines the formal order of accuracy of the scheme. Second order spatial accuracy is confirmed via grid convergence analysis in Section 5.2. Switching to the bounded form of the interpolant (Section 3.1) is expected to reduce the order of accuracy to first order locally in regions where bounds are enforced.

We emphasize the absence of a second order derivative in both Eqs. (29) and (31), which is highly desirable, since it is the predominant source of numerical dissipation when present. Even order derivatives in the truncation error give rise to dissipation errors, whereas odd order derivatives give rise to dispersion errors. The combined effect of these errors is termed "numerical diffusion" [11].

The dependence of the truncation error on σ suggests that the CFL number affects not only temporal accuracy, but also spatial accuracy for semi-Lagrangian schemes. This is unlike the behavior of Eulerian schemes, where the CFL number is related primarily to temporal accuracy. The variation of the coefficients for the first two terms in Eqs. (29) and (31) with σ is plotted in Fig. 4. Inspecting the single step error (Fig. 4(a)) is convenient for analytical examination, however, it is necessary to be mindful of the cumulative errors (Fig. 4(b)) that actually dictate the accuracy of the final solution. Both graphs indicate zero discretization error for $\sigma = 1$, which is not surprising since the interpolation becomes exact for



(a) Single step error (Eq. 29)

(b) Cumulative error (Eq. 31)

Fig. 4. Dependence of truncation error terms on σ. (a) Single step coefficients: $K_3 = \sigma(\sigma - 1)(2\sigma - 1)/6$, $K_4 = -9\sigma^2(\sigma - 1)^2/24$. (b) Cumulative error coefficients: $K_3 = (\sigma - 1)(2\sigma - 1)/6$, $K_4 = -9\sigma(\sigma - 1)^2/24$.

this case. Unfortunately, when working with non-uniform (*e.g.*, turbulent) velocity fields, the local CFL number fluctuates spatially, which makes it difficult to determine the discretization error with certainty. The coefficient of the third order derivative in Fig. 4(a) (associated with numerical dispersion, or, phase error) vanishes for $\sigma = 0.5$, whereas the fourth order derivative coefficient (associated with numerical dissipation, or gain error) attains its maximum at this point. It is interesting to note the change of sign of the K_3 coefficient at $\sigma = 0.5$. This will translate into phase speeds that are smaller (if $\sigma < 0.5$) and larger (if $\sigma > 0.5$) than the theoretical values. The combined effect of the third and fourth order derivatives can be minimized by keeping σ as close to 1 as possible. The observations discussed here are corroborated via Fourier analysis in the following section.

4.2. Gain and phase error

Gain and phase error analysis of Eulerian transport schemes is generally done via the commonly used von-Neumann and modified-equation methods [11]. Casting the transport equation in its Lagrangian form (Eq. (2)) requires that we use a slightly different technique for the BCH scheme (without bounding enforced), such as the Fourier analysis method described by Utsumi et al. [24]. Assuming periodic boundary conditions, and using the relations given by Eqs. (2) and (27), the Fourier coefficients ($\hat{z}^{n+1}(\kappa)$) of the scalar after the advection step are given as:

$$\hat{z}^{n+1}(\kappa) = \frac{1}{L}\sum_{i=0}^{N-1} z_i^{\star} e^{-j\kappa i\Delta x}\Delta x \tag{32}$$

$$\Rightarrow \quad \hat{z}^{n+1}(\kappa,\sigma) = \frac{1}{L}\sum_{i=0}^{N-1}\left[z_{i-1}^n w_{i-1}(\sigma) + z_i^n w_i(\sigma) + z_{i+1}^n w_{i+1}(\sigma) + z_{i+2}^n w_{i+2}(\sigma)\right] e^{-j\kappa i\Delta x}\Delta x \tag{33}$$

where L is the domain size. We use the periodic boundary conditions to identify that

$$\frac{1}{L}\sum_{i=0}^{N-1} z_{i-1}^n e^{-j\kappa i\Delta x}\Delta x = e^{-j\kappa\Delta x}\hat{z}^n(\kappa) \tag{34}$$

Eq. (33) thus simplifies to

$$\hat{z}^{n+1}(\kappa,\sigma) = \hat{z}^n(\kappa)\left[e^{-j\kappa\Delta x} w_{i-1}(\sigma) + w_i(\sigma) + e^{j\kappa\Delta x} w_{i+1}(\sigma) + e^{2j\kappa\Delta x} w_{i+2}(\sigma)\right] \tag{35}$$

The complex gain factor can now be computed as follows:

$$g(\kappa,\sigma) = \frac{\hat{z}^{n+1}}{\hat{z}^n} \tag{36}$$

$$= e^{-j\kappa\Delta x} w_{i-1}(\sigma) + w_i(\sigma) + e^{j\kappa\Delta x} w_{i+1}(\sigma) + e^{2j\kappa\Delta x} w_{i+2}(\sigma) \tag{37}$$

The amplitude of $g(\kappa,\sigma)$ can be used to determine the gain error, and its argument to determine the numerical phase speed error.

For the analytical solution of the advection step ($z^{n+1}(x) = z^n(x - u\Delta t)$), the amplitude and phase angle of the gain factor are 1 and $\kappa\sigma\Delta x$, respectively:

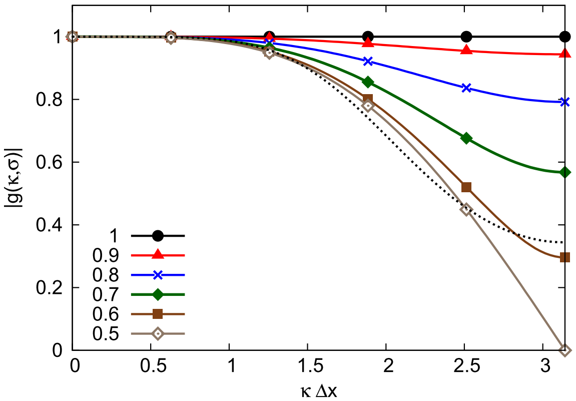


Fig. 5. Amplitude of the gain factor (Eq. (37)) for various CFL numbers. The gain is exact for $\sigma = 1$. Curves for $\sigma \in [0, 0.5)$ are not shown since $|g(\kappa, \sigma)| = |g(\kappa, 1 - \sigma)|$. The broken line corresponds to the curve for HOUC5.

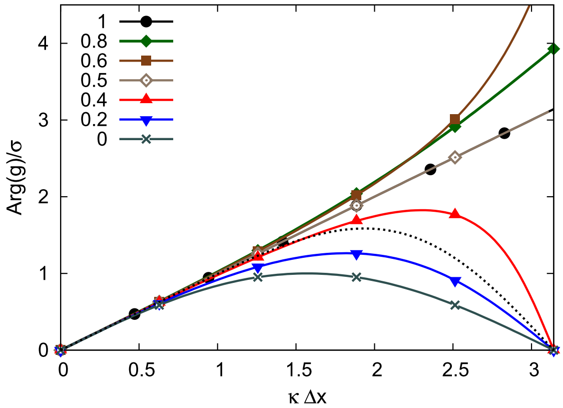


Fig. 6. Numerical phase speed (Arg(g)/σ: Eq. (37)) as a function of the true wave speed ($\kappa\Delta x$). The phase speed is exact for $\sigma = 0.5$ and 1. The broken line corresponds to the curve for HOUC5.

$$\hat{z}^{n+1}(\kappa) = \frac{1}{L}\int_0^L z^n(x - u\Delta t)e^{-j\kappa x}dx \tag{38}$$

$$= e^{j\kappa\sigma\Delta x} \cdot \hat{z}^n(\kappa) \tag{39}$$

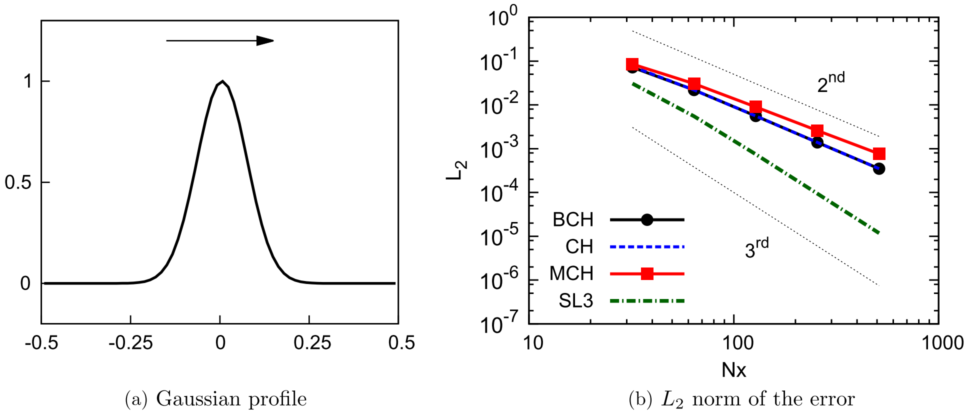
Using these values as reference, the dependence of the gain amplitude and the numerical phase speed on $\kappa\Delta x$ (the analytical phase speed) is plotted in Figs. 5 and 6. We can surmise that the BCH scheme is stable for $\sigma \in [0, 1]$ since the gain amplitude (Fig. 5) never gets larger than 1. In fact, the stability property can be extended to $\sigma > 1$, since the integer part of σ results only in a phase shift of $\hat{z}^{n+1}$, leading to the same gain amplitude as for $\sigma - [\sigma]$:

$$\left|g(\sigma)\right| = \left|g\big(\sigma - [\sigma]\big)\right| \tag{40}$$

Stability for $\sigma > 1$ was confirmed in numerical tests discussed in Section 5.3.3.

The gain error is worst for $\sigma = 0.5$, indicating large numerical dissipation for high wavenumber modes at this CFL number. This result is consistent with curve K_4 shown in Fig. 4(a). The numerical phase speed (Arg(g)/σ) becomes exact for $\sigma = 0.5$ and 1, as shown in Fig. 6. Interestingly, the phase speed is lower than the theoretical value for $\sigma < 0.5$, and greater for $\sigma > 0.5$. These results are consistent with curve K_3 in Fig. 4(b). We note that the numerical phase speed matches the exact phase speed relatively well for $\sigma \geq 0.8$, but large lead and lag errors manifest at lower CFL values (except for $\sigma = 0.5$). Considering the need to minimize both gain and phase errors in the advection step, we infer that the CFL number must be kept as close to 1 as possible.

The gain and phase error curves for a fifth order upwind convective Eulerian scheme (HOUC5) [25] are also included in Figs. 5 and 6 for comparison, and the relevant expressions have been derived in Appendix A. The curves suggest that the performance of BCH is similar to, or better than, that of the 5th order accurate HOUC5 scheme. The gain error in the worst case scenario for BCH ($\sigma = 0.5$) is comparable to that of the HOUC5 scheme, which points to superior performance with respect to numerical dissipation. The phase error is considerably smaller for $\sigma \geq 0.4$, indicating improved numerical dispersion characteristics.



(a) Gaussian profile

(b) L_2 norm of the error

Fig. 7. 1D advection of a Gaussian scalar profile, run for 5 rotations with CFL = 0.9.

5. Numerical tests and results

We now examine results obtained from simulations using five bounded (WENO3 [7], BQUICK [10], SL3, BCH, MCH) and two unbounded (QUICK [26] and HOUC5 [25]) schemes. Of these, QUICK, BQUICK, WENO3, and HOUC5 are Finite Volume based Eulerian schemes, which use the divergence form of the scalar transport equation (Eq. (41)).

$$\frac{\partial z}{\partial t} + \nabla \cdot (\boldsymbol{u} z) = \mathcal{D}\nabla^2 z \tag{41}$$

The objective is to assess and compare scheme performance with regard to boundedness, numerical diffusion, accuracy, and conservation. Both laminar and turbulent test configurations are considered. Laminar tests are essential for examining certain qualitative as well as quantitative aspects of scalar schemes. These qualities are difficult to assess in turbulent cases owing to the non-deterministic nature of turbulent velocity fields. Results from the laminar tests are discussed first, followed by those from the turbulent tests.

5.1. Numerical solver

Numerical tests for both laminar and turbulent cases were carried out using a low-Mach number Navier–Stokes solver (NGA [27]). The code uses a staggered variable arrangement, with scalar values stored at the cell centers, and velocity components stored at the appropriate cell faces. The velocity field was solved using a spatially second-order accurate, energy conserving finite-difference scheme. Time integration for velocity was done using a 2nd order semi-implicit Crank–Nicolson method.

Whenever possible, multiple scalars were transported simultaneously in the test cases, with each scalar being assigned a distinct Eulerian or semi-Lagrangian scheme. The simultaneous transport was preferable for consistent comparison of results from the various scalar schemes, especially due to the non-deterministic nature of turbulent velocity fields. Time integration of the scalar transport equation was done using a 2nd order semi-implicit Crank–Nicolson method for the Eulerian schemes. The integration of the diffusive term in the semi-Lagrangian schemes was treated using the forward Euler method (Eq. (3)).

5.2. Formal order of accuracy

The formal order of accuracy for BCH without bounds enforced (or simply, CH) was determined analytically in Section 4.1. We now inspect numerically, the order of accuracy of all the semi-Lagrangian schemes discussed in this paper. Fig. 7(b) shows the L_2 norm of the error for uniform advection of a 1D Gaussian scalar profile (Fig. 7(a)), run for 5 rotations with a CFL of 0.9. The curve suggests 2nd order spatial convergence ($O(\Delta x^2)$) for CH, which is consistent with the analytical results obtained in Section 4.1. We note that the results for both CH and BCH are virtually identical, which happens since the need to enforce bounding arises very rarely in the C^∞ continuous Gaussian profile. The result for MCH, however, is comparatively degraded, since MCH suppresses the extremum in the Gaussian profile even when physical bounds are not breached. SL3 exhibits $O(\Delta x^3)$ convergence, which is the best possible order of accuracy achievable using a cubic polynomial. The Hermite cubic polynomials are limited to $O(\Delta x^2)$ or lower, due to the fact that the relevant derivatives were computed using a 2nd order accurate finite-difference formula (Section 3.4).

5.3. Laminar tests

Two laminar tests, consisting of a 1D case and a 2D case, were used for comparing scheme performance. For the 1D case, the composite profile shown in Fig. 8(a) was advected using a uniform velocity field. The simulation was run for 5

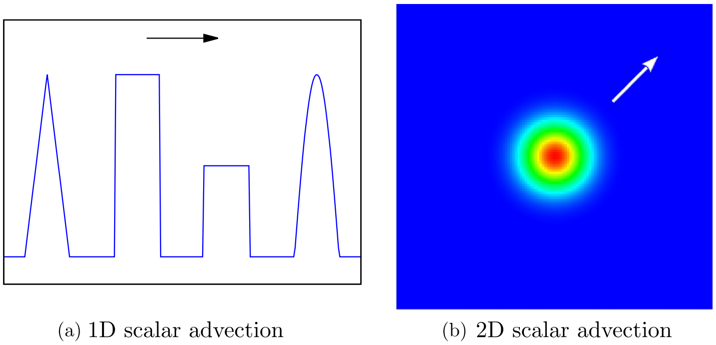

(a) 1D scalar advection (b) 2D scalar advection

Fig. 8. 1D and 2D scalar advection in uniform velocity fields.

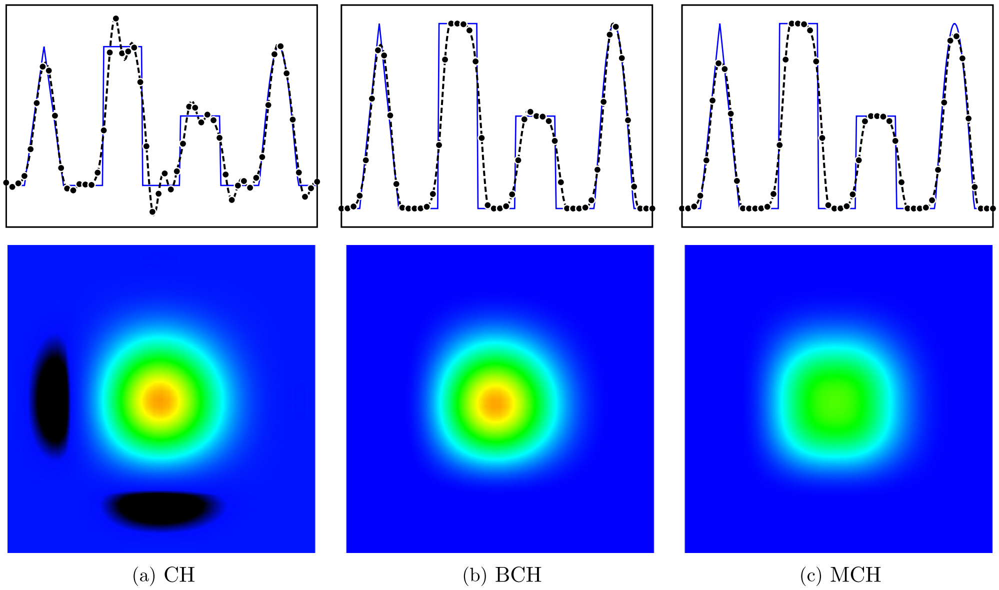

(a) CH (b) BCH (c) MCH

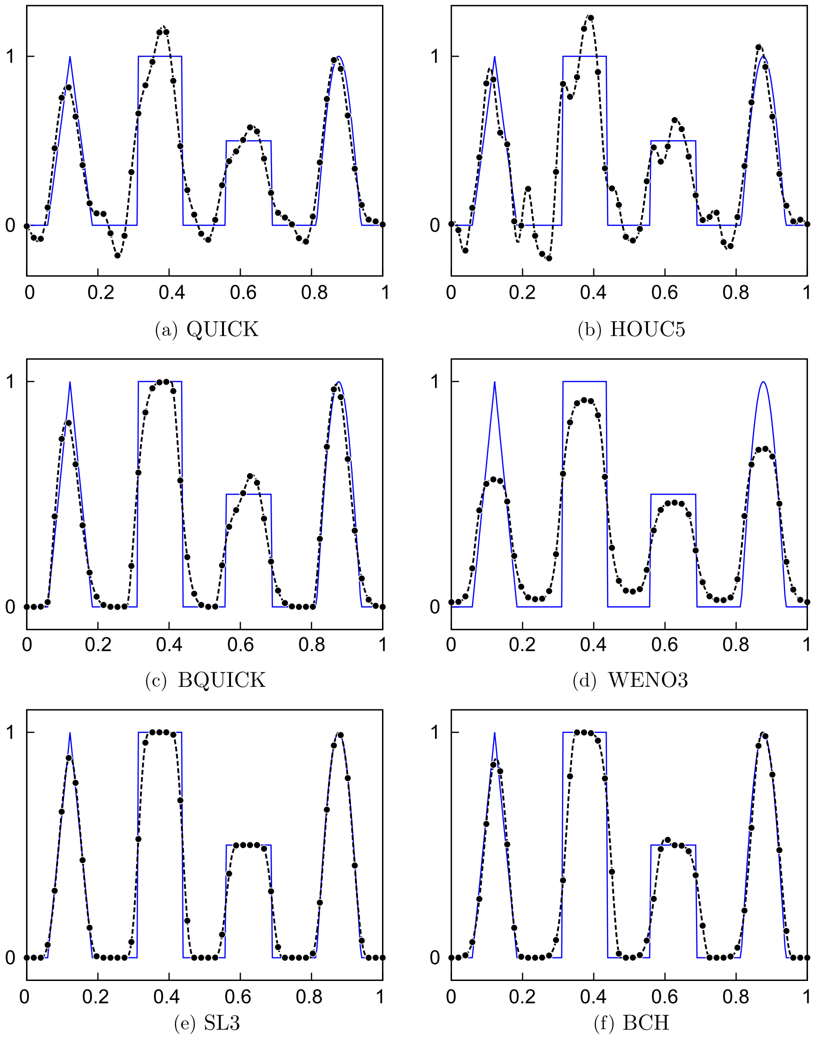
Fig. 9. Comparison of unbounded, bounded, and monotone cubic Hermite interpolation. Results from 1D simulations are shown on top, and those from 2D simulations are shown on the bottom. Regions colored in black for the 2D case indicate scalar values outside the interval [0, 1].

complete rotations before comparing results from the various scalar transport schemes. The 1D profile shape introduces sharp gradients in the scalar field, which are generally the locations where maximum occurrences of numerical errors like dispersion and overshoot happen. The 2D laminar test (Fig. 8(b)) involves uniform advection of a 2D Gaussian scalar profile along the 45-degree diagonal. Results for this simulation were compared after ten rotations. Periodic boundary conditions and zero molecular diffusivity were used for both laminar tests.

5.3.1. Bounding vs. monotonicity

Laminar tests provide a good qualitative indication of boundedness. The effects of enforcing boundedness are best observed by comparing the same cubic Hermite interpolating polynomial with three different strategies: standard Cubic Hermite (CH), Bounded Cubic Hermite (BCH), and Monotone Cubic Hermite (MCH). Fig. 9 indicates that the bounding operation is crucial for suppressing oscillations, and plays an important role in preventing the propagation of numerical errors (as found in the unbounded CH scheme). BCH is not as effective as MCH in preventing oscillations at the half-step in the 1D profile, since these oscillations are still within the specified physical bounds. Enforcing strict monotonicity using MCH, however, introduces significant numerical dissipation, as is evident from a marked decrease in scalar maxima in Fig. 9(c).

Doing a qualitative comparison with the QUICK, BQUICK, HOUC5, WENO3, and SL3 schemes, we see the presence of large oscillations for the two unbounded schemes (QUICK and HOUC5) in Figs. 10 and 11. The four bounded schemes (BQUICK, WENO3, SL3, BCH), however, adhere strictly to the physical scalar bounds (0 and 1 in this case). Close inspection of the half-step in the 1D profile reveals that BQUICK (Fig. 10(c)) gives rise to oscillations similar to those caused by QUICK (Fig. 10(a)), if physical bounds are not breached. However, BCH (Fig. 9(b)) is able to suppress spurious oscillations at the half-step (which do manifest in CH, Fig. 9(a)) to a certain extent, even though the oscillations do not breach physical bounds.



(a) QUICK

(b) HOUC5

(c) BQUICK

(d) WENO3

(e) SL3

(f) BCH

Fig. 10. Comparison of scalar transport schemes for the 1D uniform advection case (5 rotations with CFL = 0.8 on a 256 cell grid). Physical bounds of 0 and 1 were used in BCH, SL3, and BQUICK.

5.3.2. Numerical dispersion and dissipation

The effects of numerical dispersion can be observed qualitatively through asymmetric deformation of the scalar profiles. We note that of all the results shown in Figs. 10 and 11, only SL3 preserves the symmetry of the initial profile. The upwinded schemes (QUICK, BQUICK, and HOUC5) show a great deal of asymmetric deformation, especially for the 2D advection case. BCH exhibits some dispersion, but these errors can be disposed of altogether by setting $\sigma = 0.5$ or $\sigma = 1$ in these laminar cases, as discussed in Section 4.

Numerical dissipation, which is caused by the presence of even order derivatives in the truncation error (Section 4), manifests as a smoothing of sharp edges in the 1D advection case and a drop in the maximum scalar value for the 2D case. Any dissipative effects noticeable in the laminar tests are purely numerical since they were run with zero molecular diffusivity. The maxima of the scalar profiles for the 1D and 2D cases are listed in Table 1. Both SL3 and BCH perform equally well with regard to numerical dissipation in uniform advection tests (Table 1, Figs. 10 and 11), and are superior to the other bounded schemes tested.

Before further discussion, we would like to emphasize that although the performance of SL3 may seem to be slightly better than that of BCH in the uniform advection cases, BCH performs significantly better in the turbulent cases (as will be shown in Section 5.4). This observation should lead to the important realization that the suitability of numerical schemes for use in turbulent flows should not be judged based on performance in simple uniform advection cases. These idealized tests often fail to account for chaotic behavior that may affect scheme performance in unpredictable ways.

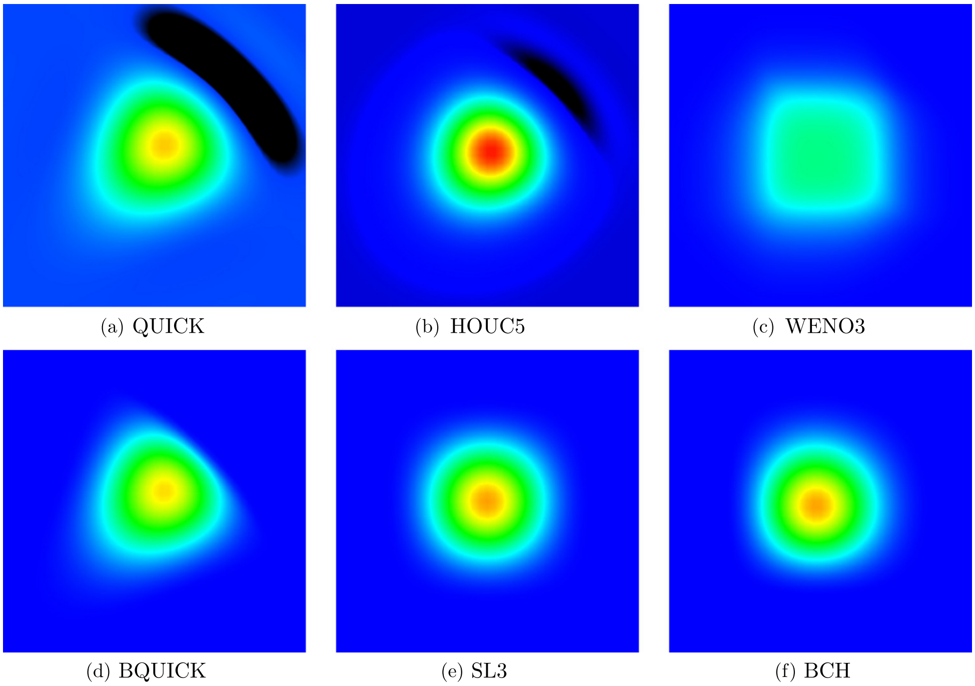

(a) QUICK (b) HOUC5 (c) WENO3
(d) BQUICK (e) SL3 (f) BCH

Fig. 11. Uniform advection of a Gaussian profile along the 45° diagonal. Results shown after 5 rotations with $\mathrm{CFL} = 0.4$ on a 64×64 grid. Red indicates maximum for the scalar ($z = 1$), and blue indicates minimum ($z = 0$). Regions colored in black depict scalar values less than 0 for QUICK and HOUC5. An explicit solver was used for all cases, and the HOUC5 case was run with $\mathrm{CFL} = 0.2$ due to stability issues. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of scheme results for scalar maxima in the laminar tests, mean scalar dissipation rate in the turbulent mixing layer at the beginning of the self-similar regime (center plane only: $\chi = 2\mathcal{D}\langle(\partial z/\partial x)^2 + (\partial z/\partial z)^2\rangle$, partial derivatives of scalar 'z' with respect to the co-ordinate axes 'x' and 'z'), and relative computational cost with respect to BCH.

Scheme	$\max_z$ – laminar tests		χ – turbulent tests		Cost
	1D	2D	$\kappa\eta = 1.5$	$\kappa\eta = 3$	
SL3	1.0000	0.8360	1.16e−3	1.56e−3	1.18
CH	1.2081	0.8398	1.76e−3	1.75e−3	0.83
BCH	1.0000	0.8399	1.48e−3	1.70e−3	1.00
MCH	0.9999	0.5712	9.54e−4	1.43e−3	0.92
WENO3	0.9179	0.3673	6.58e−4	1.16e−3	1.01
QUICK	1.1782	0.7872	1.14e−3	1.54e−3	0.45
BQUICK	0.9986	0.7834	1.13e−3	1.54e−3	1.02
HOUC5	1.2413	0.9804	1.43e−3	–	0.68

5.3.3. CFL > 1, and non-uniform velocity field

Limitations on time step size for certain simulations (*e.g.*, for weather prediction purposes) can often be imposed by numerical stability considerations, rather than by accuracy concerns [1]. Semi-Lagrangian schemes can reduce computational cost in such applications by permitting the use of much larger CFL values than would be allowable for Eulerian schemes.

We repeat some of the incompressible 1D tests discussed previously, with a spatially varying laminar velocity profile ($u = 1 + 0.5\cos(2\pi x)$, $v = 1 + \pi y \sin(2\pi x)$) and a maximum CFL > 1. Spatial variation in the velocity causes the CFL to fluctuate locally and span a range of values ($|u|_{\min} \leq \mathrm{CFL} \cdot \Delta x/\Delta t \leq |u|_{\max}$). The same grid convergence results for the Gaussian profile from Fig. 7 are shown in Fig. 12 with non-uniform velocity. Both CH and BCH maintain second order accuracy, and SL3 maintains third order accuracy. The 1D experiments shown in Fig. 10 were also repeated with the non-uniform velocity field for $\mathrm{CFL}_{\max} > 1$. The resulting profiles after 5 rotations are shown in Fig. 13. The proposed bounding algorithm (BCH) results in lower numerical dissipation than enforcing strict monotonicity (MCH), which reinforces similar observations from the uniform velocity test.

Although the proposed bounding algorithm appears to function well for the case when $\mathrm{CFL}_{\max} > 1$, maintaining sufficient accuracy in turbulent simulations often requires that the time step size be smaller than physical time scales [28]. Therefore, the capability to run at CFL > 1 plays a somewhat less crucial role, compared to the high spatial accuracy inherent in the bounded cubic Hermite polynomial. Nonetheless, results obtained by running turbulent simulations with $\mathrm{CFL}_{\max} > 1$ are discussed in Section 5.5. Additionally, it was observed that turbulent simulations with high order accurate Eulerian schemes

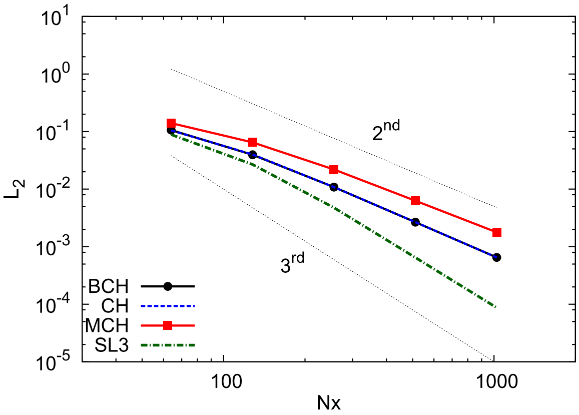


Fig. 12. Grid convergence for Gaussian scalar profile advected for 5 rotations with $\mathrm{CFL}_{\max} = 1.9$ in a non-uniform, divergence free velocity field: $u = 1 + 0.5\cos(2\pi x)$, $v = 1 + \pi y \sin(2\pi x)$.

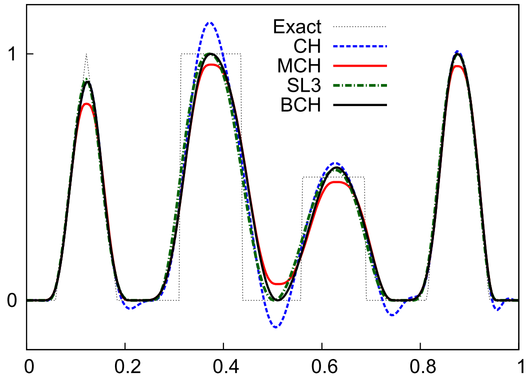


Fig. 13. 1D profile advection with non-uniform velocity field. 5 rotations with $\mathrm{CFL}_{\max} = 1.8$ on a 256 cell grid.

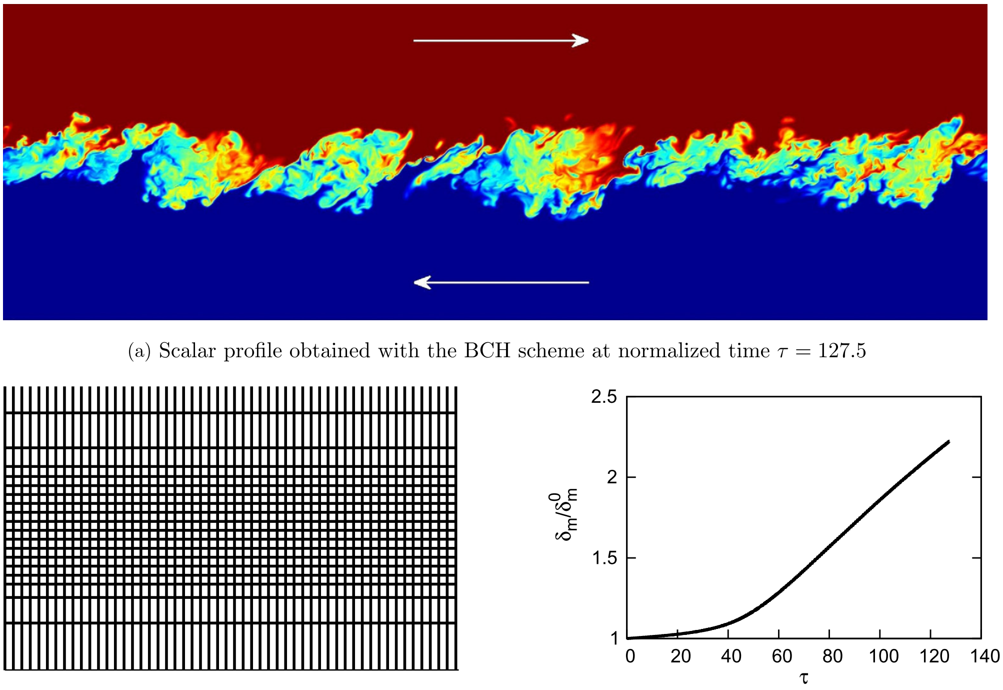
required unnecessarily restrictive time step sizes even for CFL < 1, *i.e.*, the largest useable time step size was restricted by numerical stability concerns. This reflects the well-known cost reducing potential of SL schemes in this regard, even when using CFL < 1.

5.4. Turbulent mixing layer

5.4.1. Configuration setup

In addition to the laminar tests described above, numerical tests were conducted in a 3D turbulent mixing layer configuration. This configuration follows very closely the work of Rogers and Moser [29]. It consists of two turbulent streams parallel to one another and going in opposite directions, separated by an initial interface at the mid-plane (Fig. 14(a)). The velocity field was initialized with two de-correlated velocity fields obtained from Direct Numerical Simulation (DNS) of a turbulent boundary layer [29]. The initial scalar distribution was represented by a Heaviside function, with the top half of the domain consisting of scalar values set to 1, and the bottom half set to 0 (Fig. 14(a)). The scalar gradient in the y-direction acts as a source of scalar variance for the turbulent cascade of scalar energy, and obviates the need for forcing the scalar explicitly. The molecular diffusivity of the scalar was kept equal to the kinematic viscosity of the fluid (*i.e.*, unity Schmidt number). Periodic boundary conditions were imposed in the streamwise (x) and spanwise (z) directions with periods $125\delta_m^0$ and $(125/4)\delta_m^0$, respectively [29], with δ_m^0 being the initial momentum thickness of the mixing layer as defined by Rogers and Moser [29]. Based on these values, the initial Reynolds number was computed to be $Re_m^0 = 960$ ($Re_m^0 = \delta_m^0 \Delta U/\nu$). A Neumann boundary condition was used in the cross-stream (y) direction, which spans $81\delta_m^0$.

In order to reduce the computational cost of the turbulent test case, a stretched mesh (Fig. 14(b)) was used in the cross-stream direction, whereas the grid cells in the periodic directions were uniformly sized. The grid cells in the stretched direction were clustered close to the center plane so as to capture accurately the unsteady development of the core region (approximately $32\delta_m^0$ in height) of the mixing layer. A grid size of $1280 \times 444 \times 320$ cells was selected for adequate resolution of the velocity field by ensuring that $\kappa_{\max}\eta \geq 1.5$. $\kappa_{\max}$ here is the largest wavenumber resolvable by the grid and η is the Kolmogorov scale on the center plane ($\eta = (\nu^3/\epsilon)^{1/4}$, where ν is the kinematic viscosity, and $\epsilon = \langle 2\nu S_{ij} S_{ij}\rangle$ is the average kinetic energy dissipation rate on the plane). The computational cells used were of unity aspect ratio in the core region, and were stretched in the y-direction outside the core (stretching factor ≈ 3%), as depicted in Fig. 14(b).

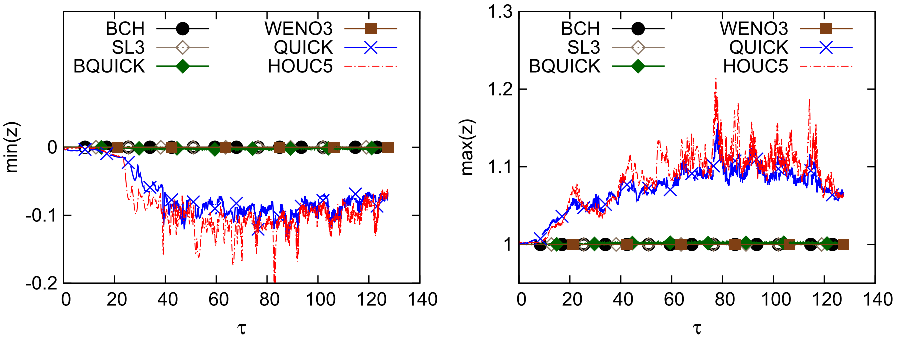


(a) Scalar profile obtained with the BCH scheme at normalized time $\tau = 127.5$

(b) Computational mesh

(c) Time evolution of momentum thickness

Fig. 14. Turbulent mixing layer configuration.



(a) Scalar minimum

(b) Scalar maximum

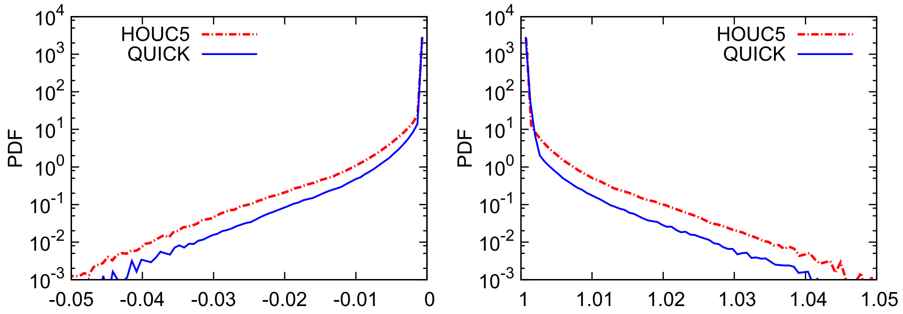
Fig. 15. Time evolution of the scalar maximum and minimum in the turbulent mixing layer.

The simulation was run with a maximum CFL of 0.6, keeping the time step size for both the velocity and scalar solvers the same. We point out that in a turbulent case, the spatially and temporally varying local CFL value may span the entire range of $[0, \mathrm{CFL}_{\mathrm{max}}]$. Results from the various scalar transport schemes were compared once the core of the mixing layer attained a self-similar state, as indicated by the linear growth of the momentum thickness. The time evolution of the momentum thickness normalized by the initial momentum thickness is shown in Fig. 14(c). After a transient growth, the mixing layer enters a self-similar regime around $\tau = 60$. $\tau = t\Delta U/\delta_m^0$ is the normalized time defined by Rogers and Moser [29]. The growth rate in the self-similar regime, *i.e.*, the slope of the linear portion of the curve in Fig. 14(c) is similar to the value observed by Rogers and Moser.

5.4.2. Results

Turbulent simulations provide a more rigorous indication of the practical usability of numerical schemes in physically realistic scenarios, when compared to laminar simulations. The properties already discussed in Section 5.3 are, therefore, re-examined in the turbulent mixing layer.

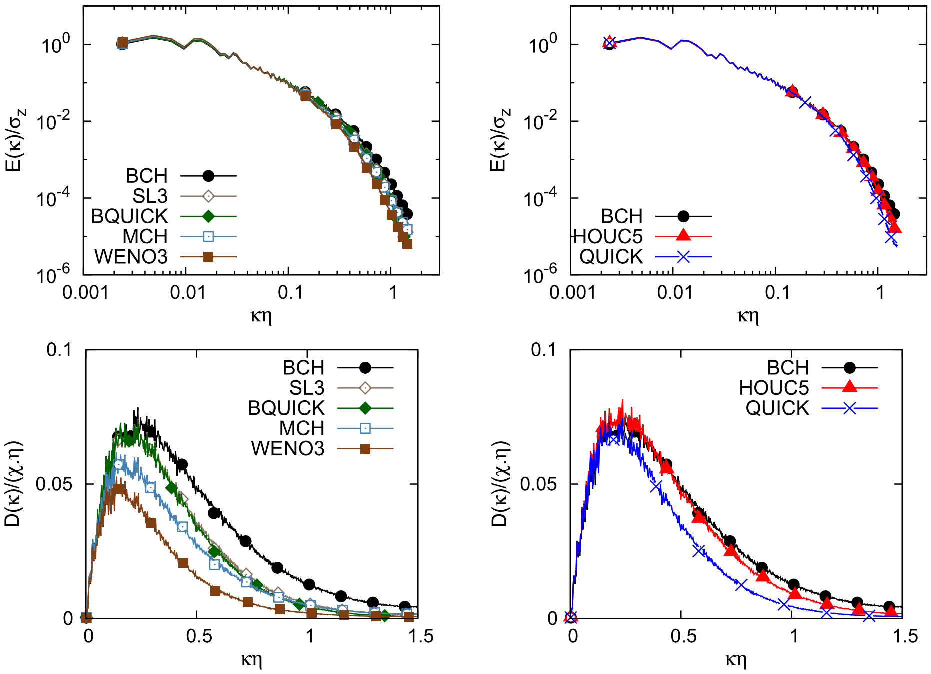
Boundedness was inspected using the time evolution of the maximum and the minimum of the scalar in the computational domain (Fig. 15). It is evident that the unbounded schemes breach the global maximum and minimum, but the bounded schemes do not. In fact, both the QUICK and HOUC5 schemes can over/undershoot by more than 10%. Probability Density Functions (PDF) of scalar values breaching physical bounds for both these schemes are shown in Fig. 16, and confirm



(a) Lower bound breached

(b) Upper bound breached

Fig. 16. Probability density function of the scalar value breaching bounds. Data shown at $\tau = 120$.



(a) BCH *vs.* bounded schemes

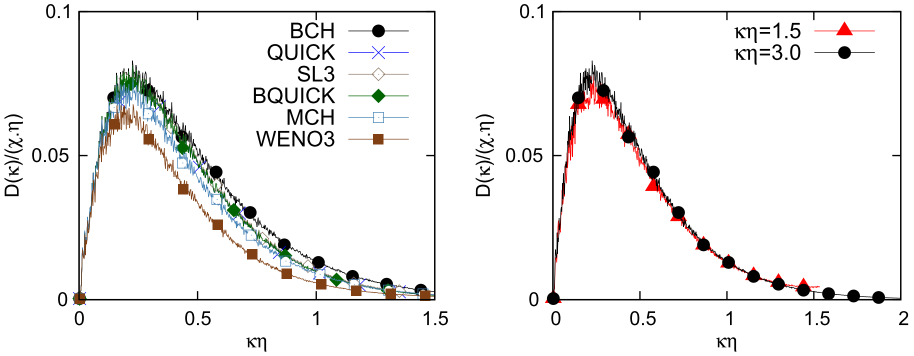
(b) BCH *vs.* unbounded schemes

Fig. 17. 1D scalar energy (top) and dissipation (bottom) spectra in the streamwise direction, computed on the center plane of the mixing layer ($\kappa\eta = 1.5$). The dissipation spectra have been normalized by the instantaneous χ value for BCH at resolution $\kappa\eta = 3$, and then averaged over 11 separate snapshots in the self-similar state ($\tau = 106.25$ to 127.5, Fig. 14(c)). The scalar energy spectra have been averaged similarly, after normalization by the area under the individual curves (*i.e.*, the scalar variance σ_z).

the occurrence of more severe oscillations for HOUC5 than for QUICK (Runge's phenomenon). The PDFs indicate that the majority of the breached values lie close to the bounds, however, there is a non-negligible distribution corresponding to relatively large deviations.

Numerical dispersion was not examined in the mixing layer as it is difficult to distinguish between deformation caused by numerically-induced dispersion and that caused by straining from the non-uniform velocity field.

Numerical dissipation in turbulence computations poses one of the most significant hurdles, as it leads to degradation of accuracy at the dynamically important small length scales. This in turn can have a negative impact at the large scales. The effects of numerical dissipation are difficult to assess directly, and are instead evaluated by examining the scalar energy and dissipation spectra at the center plane of the mixing layer (Fig. 17). The scalar energy spectrum is defined as $E(\kappa) = \langle|\hat{z}(\boldsymbol{\kappa})|^2\rangle$, where $\hat{z}(\boldsymbol{\kappa})$ are the Fourier coefficients of the scalar field. The dissipation spectrum is defined as $D(\kappa) = 2\mathcal{D}\kappa^2 E(\kappa)$, and indicates the intensity of *molecular* dissipation as a function of the wavenumber. Discriminating between scalar energy spectra at large wavenumbers (*i.e.*, the small length scales) becomes difficult (Fig. 17), for which reason we will use the analogous but more informative dissipation spectra for comparing scheme performance. We observe that molecular dissipation peaks close to $\kappa\eta \approx 0.2$, and drops off to zero as $\kappa\eta$ approaches 1.5. There is reasonable agreement



(a) Dissipation spectra at increased grid resolution ($\kappa\eta = 3$)

(b) Coarse grid and fine grid spectra produced by BCH

Fig. 18. 1D center plane dissipation spectra for the turbulent mixing layer with increased grid resolution ($\kappa\eta = 3$). All spectra normalized by the χ value for BCH, and then averaged over 11 separate snapshots.

among the scheme results for low wavenumbers ($\kappa\eta \leq 0.1$), but large disparity can be observed at intermediate and high wavenumbers.

As is evident from Fig. 17, numerical dissipation results in a marked degradation of the dissipation spectra at high wavenumber modes for QUICK, BQUICK, WENO3, SL3, and MCH. The magnitudes of the dissipation spectra for BCH and HOUC5 are noticeably higher, which is suggestive of comparatively smaller artificial dissipative damping. A more quantitative comparison is shown in Table 1, where higher values of the average scalar dissipation rate ($\chi = \langle 2\mathcal{D}|\nabla Z|^2\rangle$, averaged on the center plane) indicate lower numerical dissipation. The dissipation spectra as well as the mean scalar dissipation rates suggest that numerical dissipation in BCH is lower than in QUICK, and is comparable to that in HOUC5. This is especially remarkable since both QUICK and HOUC5 are unbounded schemes, and we expect numerical damping in bounded schemes to be slightly higher than that in unbounded schemes. The extremely well behaved nature of the BCH scheme may be attributed to the sub-cell shape resolving capability of Hermite polynomials. Low numerical dissipation in BCH can be linked to the absence of a second order derivative in Eqs. (29) and (31), which in turn implies that the excessive dissipation inherent in MCH results entirely from strict monotonicity enforcement.

Accuracy of the results furnished by BCH in the turbulent mixing layer is gauged by examining scalar dissipation spectra from a simulation run at twice the grid resolution (Fig. 18) of the base simulation. Upon grid refinement, spectra from all the schemes (barring WENO3) collapse onto a single curve (Fig. 18(a)), which is almost identical to the spectrum produced by BCH at the lower resolution (Fig. 18(b)). This collapse, in conjunction with the grid independent behavior of BCH, may be taken as evidence that the result furnished by the BCH scheme is indeed the ‘true’ solution of the simulation. Results for HOUC5 at the higher resolution could not be obtained owing to stability issues. We note that numerical dissipation in all of the schemes can be reduced greatly by increasing grid resolution (as indicated by the spectra, and the χ values listed in Table 1), but at a steep accompanying rise in computational cost.

5.4.3. Mass conservation

Loss in conservation of the transported quantity is a central issue of concern when using semi-Lagrangian schemes. The interpolation procedure outlined in Section 2.3 is generally non-conservative, owing to the fact that the sum of all interpolation weights associated with a particular grid node (for a given time step) may not be equal to 1. This happens primarily in non-uniform velocity fields, where traceback causes certain grid nodes to be weighted more (or less) heavily than others. In addition to that, bounding corrections also introduce conservation errors into the scheme. There have been various efforts in the past to make semi-Lagrangian schemes conservative. However, these methods either introduce certain anomalous effects, such as failing to maintain physical bounds (as can happen with the use of so-called ‘mass-fixers’ [17,18]), or are so prohibitively expensive that they are suitable for practical use only in 2D simulations (remapping techniques [30,19]). As Williamson and Rasch [5] point out, conservation may only be important in a relative sense, especially when the conservation errors are small in comparison to source terms that may be present. That is to say, it may be possible to tolerate loss of conservation to a certain degree, in the same way that we tolerate discretization errors.

Turbulent simulations run using BCH show deviations in the total scalar mass amounting to approximately 0.5% in the mixing layer (Fig. 19). The curves shown in Fig. 19 were evaluated as the total mass percentage deviation in the domain normalized by the effective core thickness. Both the mass deviation and the core thickness evolve in time. Normalization by the core thickness ratio δ_m/Ly (which varies from 0.012 to 0.027) was done to account for the fact that regions outside the core, where the scalar value is effectively constant, experience near perfect interpolation and do not suffer from mass deviation.

It is important to note that minor mass fluctuations seem to have no measurable impact on the ability of the scheme to predict transport characteristics, as already confirmed in Section 5.2. On the contrary, results from the majority of the Eulerian schemes (Fig. 17), which do in fact conserve the transported scalar perfectly, were seen to be largely unusable if

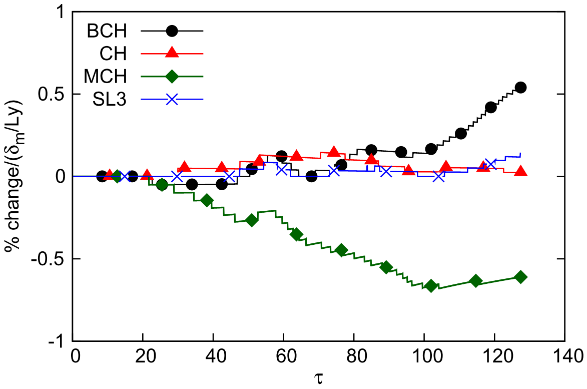


Fig. 19. Time evolution of scalar mass change in the turbulent mixing layer. The y-axis has been normalized by δ_m/Ly, which is a measure of effective core thickness in time.

high accuracy is desired in turbulent cases. Thus, depending on the demands of the physical problem, it might be preferable to tolerate a minute and inconsequential loss in conservation (as done in BCH), rather than to suffer from substantial numerical errors inherent in the transport scheme.

5.4.4. Cost

Apart from well behaved numerics, computational efficiency plays a decisive role in determining the usability of numerical algorithms. Tests used to measure computational performance of the BCH scheme demonstrate that improved accuracy of the scheme is not necessarily associated with increased computational cost. In fact, turbulent runs indicate that the computational cost of the BCH scheme is comparable to that of BQUICK and WENO3 (Table 1).

Raw grind time, however, does not serve as a truly reliable measure of computational cost. For instance, scalar dissipation spectra in Fig. 18 establish that all bounded schemes require at least a twofold increase in grid resolution to be able to produce results of quality comparable to BCH. At a constant CFL number, this amounts to a 16× increase in computational cost. Needless to say, this is a prohibitively large increase, and makes the BCH scheme immediately attractive for use in computationally intensive simulations.

5.5. Homogeneous isotropic turbulence

The turbulent mixing layer configuration discussed in Section 5.4 was selected specifically since it provides an ideal environment for testing scalar boundedness and assessing numerical diffusion. We now use an alternate turbulent configuration, namely, homogeneous isotropic turbulence, to compare the numerical performance of BCH against spectral schemes, which are frequently used in turbulence computations involving simple geometries. The configuration utilizes a uniform mesh on a cubic domain of length 2π, with periodic boundary conditions in all three directions. The velocity was forced spectrally in a low wavenumber shell [31].

Due to the absence of a natural means of sustaining scalar fluctuations in homogeneous isotropic turbulence, a mean scalar gradient ($\boldsymbol{G}$) was imposed as follows [32]:

$$\frac{\partial z}{\partial t} + \boldsymbol{u} \cdot \nabla z = \mathcal{D}\nabla^2 z + \boldsymbol{G} \cdot \boldsymbol{u} \tag{42}$$

$$\boldsymbol{G} = (1, 0, 0) \tag{43}$$

The presence of the mean scalar gradient term implies that it is no longer appropriate to consider the scalar as being bounded by its initial maximum and minimum values. Under these conditions, the probability density function of scalar fluctuations is well represented by a Gaussian distribution [33].

The time steps for velocity and scalar transport were decoupled to enable multiple velocity iterations for a single scalar step. Since semi-Lagrangian schemes are stable for CFL > 1, this allows us to take larger timesteps for the scalar, constrained only by the physically relevant Kolmogorov time scale ($\Delta t_z < \tau_\eta = (\nu/\epsilon)^{1/2}$), instead of being constrained by the numerical stability of the velocity scheme. This can result in substantial computational savings, especially when using different grid resolutions for the velocity and scalar, determined by the Kolmogorov and Batchelor length scales ($\eta_B = \eta/\sqrt{Sc}$), respectively [33,34].

The physical parameters for the homogeneous simulation were selected ($Re_\lambda = 140$, $Sc = 1$) to allow for direct comparison with spectral scheme results available in Ref. [32]. A grid size of 512^3 was used to ensure that $\kappa_{\max}\eta = 1.5$. At this grid resolution, the time step associated with stable velocity iterations was approximately $0.001\tau_\eta$. To assess the performance of BCH for $\mathrm{CFL}_{\max} > 1$, Δt_z was taken to be larger than the velocity timestep, and the simulations were run until the scalar statistics became stationary. The relevant scalar energy and dissipation spectra are plotted in Fig. 20, and indicate good

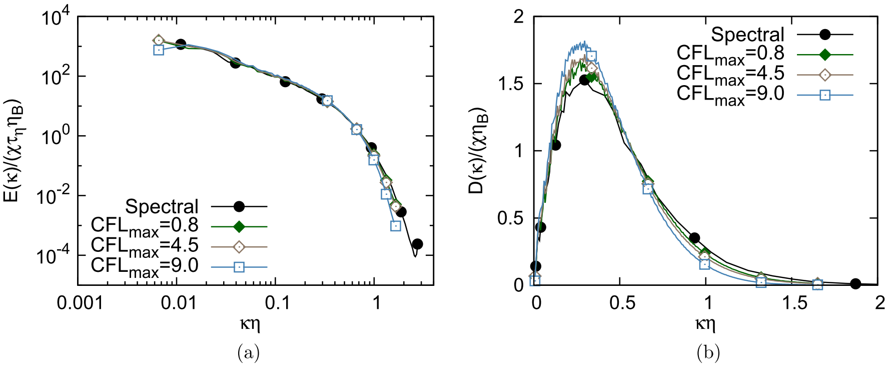


Fig. 20. 3D energy and dissipation spectra for homogeneous isotropic turbulence simulations. Results shown for spectral scheme (data taken from Ref. [32]), and BCH with three different values of CFL_{max}.

agreement between results produced by BCH and the spectral scheme. Simulations run with CFL_{max} values much larger than 1 suffer from the presence of time-splitting errors, which may be related to the use of the forward Euler method in Eq. (3). The maximum deviation from mass conservation, defined as (total change in scalar mass in domain)/(initial mass), was observed to be approximately 0.1%, 0.2%, and 0.5%, for the CFL_{max} values of 0.8, 4.5, and 9, respectively. As in the mixing layer case, turbulent statistics in the homogeneous configuration remain unaffected by the minor loss in conservation.

6. Summary

In this paper, an improved semi-Lagrangian transport scheme for passive scalar mixing has been proposed and tested in both laminar and turbulent flows. This scheme, named BCH, is designed specifically for the turbulent transport of scalars that must respect physical bounds. The scheme is based on the semi-Lagrangian treatment of the advection–diffusion equation, and makes use of cubic Hermite polynomials for interpolation. Boundedness of the scalar field is ensured by applying derivative limiting techniques to the Hermite polynomial. The novelty of the proposed algorithm lies in the way that sub-cell extrema are treated to restore boundedness. The shape-preserving ability of Hermite polynomials makes them especially useful for resolving fine scale scalar structures in turbulent simulations.

Performance of the proposed transport scheme, along with that of several other commonly used schemes (*e.g.*, BQUICK, WENO3, HOUC5, *etc.*), is tested in two laminar cases, and in a three dimensional turbulent mixing layer. The BCH scheme outperforms all other bounded schemes tested, with regard to the ability to resolve small scale turbulent structures. This is confirmed by examining scalar energy and dissipation spectra on the center plane of the mixing layer. Simulations run with the other bounded schemes require at least twice the spatial resolution to obtain spectra of comparable quality. In addition to cost savings on account of a coarser grid, comparatively lenient stability conditions for semi-Lagrangian schemes allow a further drop in computational cost for BCH, in comparison to Eulerian schemes. Simulations in homogeneous isotropic turbulence indicate that results produced by BCH are comparable to those produced by spectral schemes, with a slight loss of accuracy caused by time splitting errors, when using very large timesteps unconstrained by the CFL number. Although semi-Lagrangian schemes are known to suffer from mass conservation issues, the turbulent test cases indicate minimal deviation in total scalar mass for the BCH scheme. The minimal loss in conservation has no measurable impact on predicting the correct turbulent characteristics.

Overall, the BCH scheme offers an attractive alternative to Eulerian schemes for turbulent simulations of passive scalar mixing, on account of reduced numerical errors and computational cost, and relative ease of implementation.

Acknowledgements

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. The authors also gratefully acknowledge funding from the U.S. Department of Energy–Basic Energy Sciences (DE-SC0006591).

Appendix A. Gain and phase error for the HOUC5 scheme

The curves for HOUC5 shown in Figs. 5 and 6 were derived using the procedure described in Ref. [35]. Beginning with Eq. (2), we can write the analytical solution for a 1D periodic problem with uniform advection as follows

$$z(x,t) = f(t)e^{j\kappa x} \tag{A.1}$$

Denoting volume averages stored at the cell centers with an overbar $(\bar{\cdot})$, we have

$$\bar{z}_i = \frac{1}{\Delta x} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} z(x)dx \tag{A.2}$$

$$= \frac{f(t)}{j\kappa\Delta x}\left(e^{j\kappa x_{i+\frac{1}{2}}} - e^{j\kappa x_{i-\frac{1}{2}}}\right) \tag{A.3}$$

This expression can be generalized to the following form

$$\bar{z}_{i+m} = \frac{f(t)e^{j\kappa x_{i+\frac{1}{2}}}}{j\kappa\Delta x}\left(e^{j\kappa m\Delta x} - e^{j\kappa(m-1)\Delta x}\right) \tag{A.4}$$

In the finite-volume formulation, cell face scalar flux values are used to discretize Eq. (2)

$$\frac{\partial \bar{z}_i}{\partial t} = -\frac{u}{\Delta x}(\bar{z}_{i+\frac{1}{2}} - \bar{z}_{i-\frac{1}{2}}) \tag{A.5}$$

For HOUC5, the cell face scalar values are computed as follows

$$\bar{z}_{i+\frac{1}{2}} = -\frac{6}{120}\bar{z}_{i-1} + \frac{54}{120}\bar{z}_i + \frac{94}{120}\bar{z}_{i+1} - \frac{26}{120}\bar{z}_{i+2} + \frac{4}{120}\bar{z}_{i+3} \tag{A.6}$$

$$\bar{z}_{i-\frac{1}{2}} = -\frac{6}{120}\bar{z}_{i-2} + \frac{54}{120}\bar{z}_{i-i} + \frac{94}{120}\bar{z}_i - \frac{26}{120}\bar{z}_{i+1} + \frac{4}{120}\bar{z}_{i+2} \tag{A.7}$$

Expanding the right hand sides of these two expressions using Eq. (A.4), and substituting into Eq. (A.5) we have

$$\frac{\partial \bar{z}_i}{\partial t} = -\frac{uf(t)e^{j\kappa x_{i+\frac{1}{2}}}}{60j\kappa\Delta x^2}\left[-3e^{-j\kappa 3\Delta x} + 33e^{-j\kappa 2\Delta x} - 10e^{-j\kappa\Delta x} - 80 + 75e^{j\kappa\Delta x} - 17e^{j\kappa 2\Delta x} + 2e^{j\kappa 3\Delta x}\right] \tag{A.8}$$

To compare this expression to the analytical solution, we differentiate Eq. (A.3) with respect to time, and use the fact that $df/dt = -uj\kappa f(t)$ (from Eqs. (2) and (A.1)) to get

$$\frac{\partial \bar{z}_i}{\partial t} = -\frac{uf(t)e^{j\kappa x_{i+\frac{1}{2}}}}{\Delta x}\left(1 - e^{-j\kappa\Delta x}\right) \tag{A.9}$$

Comparing Eq. (A.8) and Eq. (A.9), we get the expression for the effective wavenumber (κ_{eff})

$$\kappa_{eff}\Delta x = \left[-3e^{-j\kappa 3\Delta x} + 33e^{-j\kappa 2\Delta x} - 10e^{-j\kappa\Delta x} - 80 + 75e^{j\kappa\Delta x} - 17e^{j\kappa 2\Delta x} + 2e^{j\kappa 3\Delta x}\right]/60j\left(1 - e^{-j\kappa\Delta x}\right) \tag{A.10}$$

The numerical phase speed for HOUC5 is equal to $\Re(\kappa_{eff}\Delta x)$ (Fig. 6), and the gain amplitude is equal to $e^{-\Im(\kappa_{eff}\Delta x)}$ (Fig. 5).

References

[1] A. Staniforth, J. Côté, Semi-Lagrangian integration schemes for atmospheric models – a review, Mon. Weather Rev. 119 (1991) 2206–2223.

[2] D.L. Williamson, J.G. Olson, Climate simulations with a semi-Lagrangian version of the NCAR community climate model, Mon. Weather Rev. 122 (1994) 1594–1610.

[3] M. Chen, R.B. Rood, L.L. Takacs, Impact of a semi-Lagrangian and an Eulerian dynamical core on climate simulations, J. Climate 10 (1997) 2374–2389.

[4] D.A. Donzis, P.K. Yeung, Resolution effects and scaling in numerical simulations of passive scalar mixing in turbulence, Physica D 239 (2010) 1278–1287.

[5] D.L. Williamson, P.J. Rasch, Two-dimensional semi-Lagrangian transport with shape-preserving interpolation, Mon. Weather Rev. 117 (1989) 102–129.

[6] P.J. Rasch, D.L. Williamson, On shape-preserving interpolation and semi-Lagrangian transport, SIAM J. Sci. Stat. Comput. 11 (1990) 656–687.

[7] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[8] D.S. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2000) 405–452.

[9] V. Daru, C. Tenaud, High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations, J. Comput. Phys. 193 (2004) 563–594.

[10] M. Herrmann, G. Blanquart, V. Raman, Flux corrected finite volume scheme for preserving scalar boundedness in reacting large-eddy simulations, AIAA J. 44 (2006) 2879–2886.

[11] R.J. Leveque, Finite-Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[12] J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1973) 38–69.

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

[14] R. Bermejo, A. Staniforth, The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes, Mon. Weather Rev. 120 (1992) 2622–2632.

[15] W.-Y. Sun, K.-S. Yeh, R.-Y. Sun, A simple semi-Lagrangian scheme for advection equations, Q. J. R. Meteorol. Soc. 122 (1996) 1211–1226.

[16] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys. 169 (2001) 556–593.

[17] A. Priestley, A quasi-conservative version of the semi-Lagrangian advection scheme, Mon. Weather Rev. 121 (1993) 621–629.

[18] M. Lentine, J.T. Grétarsson, R. Fedkiw, An unconditionally stable fully conservative semi-Lagrangian method, J. Comput. Phys. 230 (2011) 2857–2879.

[19] F. Xiao, T. Yabe, X. Peng, H. Kobayashi, Conservative and oscillation-less atmospheric transport schemes based on rational functions, J. Geophys. Res. 107 (2002) 1–11.

[20] S.K. Godunov, A finite-difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–306.

[21] C. de Boor, B. Swartz, Piecewise monotone interpolation, J. Approx. Theory 21 (1977) 411–416.

[22] F.N. Fritsch, R.E. Carlson, Monotone piecewise cubic interpolation, SIAM J. Numer. Anal. 17 (1980) 238–246.

[23] H.T. Huynh, Accurate monotone cubic interpolation, SIAM J. Numer. Anal. 30 (1993) 57–100.

[24] T. Utsumi, T. Kunugi, T. Aoki, Stability and accuracy of the cubic interpolated propagation scheme, Comput. Phys. Commun. 101 (1997) 9–20.

[25] R.R. Nourgaliev, T.G. Theofanous, High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set, J. Comput. Phys. 224 (2007) 836–866.

[26] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Methods Appl. Mech. Eng. 19 (1979) 59–98.

[27] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comput. Phys. 227 (2008) 7125–7159.

[28] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, J. Comput. Phys. 113 (1994) 1–4.

[29] M.M. Rogers, R.D. Moser, Direct simulation of a self-similar turbulent mixing layer, Phys. Fluids 6 (1994) 903–923.

[30] P.H. Lauritzen, A stability analysis of finite-volume advection schemes permitting long time steps, Mon. Weather Rev. 135 (2007) 2658–2673.

[31] K. Alvelius, Random forcing of three-dimensional homogeneous turbulence, Phys. Fluids 11 (1999) 1880–1889.

[32] D.A. Donzis, K.R. Sreenivasan, P.K. Yeung, The Batchelor spectrum for mixing of passive scalars in isotropic turbulence, Flow Turbul. Combust. 85 (2010) 549–566.

[33] T. Gotoh, S. Hatanaka, H. Miura, Spectral compact difference hybrid computation of passive scalar in isotropic turbulence, J. Comput. Phys. 231 (2012) 7398–7414.

[34] J.-B. Lagaert, G. Balarac, G.-H. Cottet, Hybrid spectral-particle method for the turbulent transport of a passive scalar, J. Comput. Phys. 260 (2014) 127–142.

[35] X. Nogueira, I. Colominas, L. Cueto-Felgueroso, S. Khelladi, On the simulation of wave propagation with a higher-order finite volume scheme based on reproducing kernel methods, Comput. Methods Appl. Mech. Eng. 199 (2010) 1471–1490.