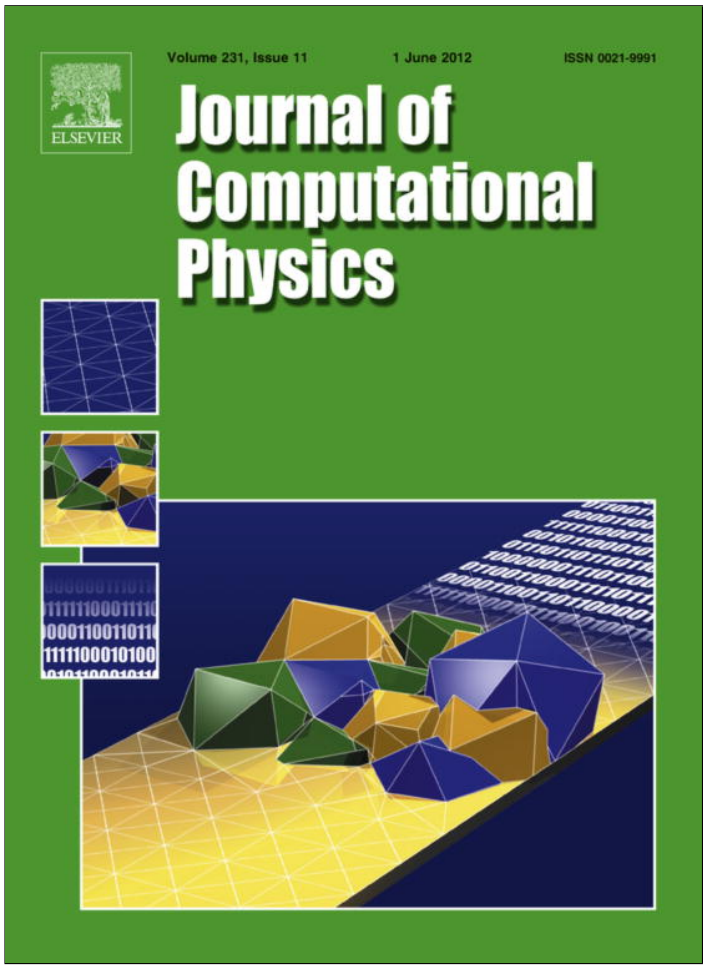


Journal of Computational Physics 231 (2012) 4028–4052

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An Eulerian–Lagrangian WENO finite volume scheme for advection problems

Chieh-Sen Huang [a,*,1], Todd Arbogast [b,c,2], Jianxian Qiu [d,3]

[a] Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan, ROC
[b] The University of Texas at Austin, Department of Mathematics, 1 University Station C1200, Austin, TX 78712, USA
[c] The University of Texas at Austin, Institute for Computational Engineering and Sciences, 1 University Station C0200, Austin, TX 78712, USA
[d] School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, PR China

ARTICLE INFO

Article history:
Received 19 August 2011
Received in revised form 4 January 2012
Accepted 20 January 2012
Available online 14 February 2012

Keywords:
Eulerian–Lagrangian
Semi-Lagrangian
WENO reconstruction
Finite volume
Locally mass conservative
Characteristics
Hyperbolic
Strang splitting

ABSTRACT

We develop a locally conservative Eulerian–Lagrangian finite volume scheme with the weighted essentially non-oscillatory property (EL–WENO) in one-space dimension. This method has the advantages of both WENO and Eulerian–Lagrangian schemes. It is formally high-order accurate in space (we present the fifth order version) and essentially non-oscillatory. Moreover, it is free of a CFL time step stability restriction and has small time truncation error. The scheme requires a new integral-based WENO reconstruction to handle trace-back integration. A Strang splitting algorithm is presented for higher-dimensional problems, using both the new integral-based and pointwise-based WENO reconstructions. We show formally that it maintains the fifth order accuracy. It is also locally mass conservative. Numerical results are provided to illustrate the performance of the scheme and verify its formal accuracy.

1. Introduction

Given $a(x,t)$, consider the one (and later, two) space dimensional initial value problem for a hyperbolic advection equation

$$\frac{\partial u}{\partial t}+\frac{\partial (au)}{\partial x}=\nabla_{t,x}\cdot\begin{pmatrix}u(x,t)\\ a(x,t)u(x,t)\end{pmatrix}=0,\quad x\in\mathbb{R},\ t>0, \tag{1.1}$$

$$u(x,0)=u_0(x),\quad x\in\mathbb{R}. \tag{1.2}$$

The object of this paper is to develop an *Eulerian–Lagrangian weighted essentially non-oscillatory* (EL–WENO) finite volume scheme for this one space dimensional problem, as well as to extend it to multiple dimensions.

Both *essentially non-oscillatory* (ENO) and *weighted essentially non-oscillatory* (WENO) methods [12–14,16,17,20] have proven to be very successful schemes with high-order accuracy when handling hyperbolic equations in many applications. ENO/WENO schemes use the idea of adaptive stencils in the reconstruction procedure based on the local smoothness of the

* Corresponding author.
E-mail addresses: huangcs@math.nsysu.edu.tw (C.-S. Huang), arbogast@ices.utexas.edu (T. Arbogast), jxqiu@xmu.edu.cn (J. Qiu).

[1] This author was supported in part under Taiwan National Science Council Grant 99-2115-M-110-006-MY3.
[2] This author was supported in part by US National Science Foundation Grants DMS-0713815 and DMS-0835745.
[3] This author was supported in part by National Science Foundation of China Grant No. 10931004 and ISTCP of China Grant No. 2010DFR00700.

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2012.01.030

numerical solution to automatically achieve high-order accuracy and a nonoscillatory property near discontinuities. However, both ENO and WENO (or any scheme with an Eulerian approach) are explicit schemes designed on a fixed grid, so they suffer from a CFL time step stability restriction.

Schemes using an *Eulerian–Lagrangian* or *semi-Lagrangian* approach [1,2,4,6,8–11,19,26,27] look to characteristic analysis to aid in solving the problem. They have in common the fact that the advection is treated by a characteristic tracing algorithm (a Lagrangian frame of reference) from a fixed Eulerian grid over each time step. These methods have the significant advantage that CFL number restrictions of purely Eulerian methods are alleviated because of the Lagrangian nature of the advection. Furthermore, because the spatial and temporal dimensions are coupled through the characteristic tracing, the influence of time truncation error is greatly reduced.

Recently, Qiu, Christlieb, and Shu [21–23] derived semi-Lagrangian (i.e., Eulerian–Lagrangian) WENO finite difference schemes for the advection equation. Those schemes do not suffer the CFL time step restriction. However, the schemes in [21,23] could only handle the constant convection case, i.e., $a(x,t) = a$. The scheme in [22] removes this restriction. In their work, the integral form of (1.1) is taken over a triangular region, and this is used to reconstruct a high-order flux in a conservative scheme. However, they did not develop theoretically a fifth order method.

Carrillo and Vecil [5, Section 3.5] also proposed two semi-Lagrangian schemes for a variable advection equation. Their first scheme requires knowledge of the derivative of the characteristic curves with respect to the initial point, which is not normally available. This scheme is also not conservative. Their second scheme is a flux balance method. It was developed for the constant velocity case, and its extension to variable velocity was mentioned in passing, devoid of details regarding its definition and implementation. The implication seems to be that one should reconstruct the solution as in the constant velocity case and simply integrate it over the trace-back region (so, in particular, no integral-based reconstruction was provided). Moreover, no extension to multi-dimensions was given.

The new scheme we develop follows the Eulerian–Lagrangian framework of the Characteristics-Mixed Method [1,2,4], introduced by Arbogast, Chilakapati, and Wheeler, and the Finite Difference Locally Conservative Eulerian–Lagrangian Method, introduced by Douglas and Huang [9]. In this framework, we trace along the characteristics each computational Eulerian grid cell or element E backward in time over the time step to its Lagrangian trace-back region $\check{E}$. Average mass is simply transported from the trace-back region $\check{E}$ to the grid cell E. This is the most natural way to apply the Eulerian–Lagrangian approach, the scheme conserves mass locally, and it is simple and efficient to compute in one-dimension. However, it leaves us with a low order finite volume method [3,4,9].

We will combine this framework with a WENO reconstruction for high-order approximation in the spatial variable at the previous time level. Our locally conservative finite volume scheme achieves the advantages of both WENO and Eulerian–Lagrangian schemes. That is, our scheme is not only high-order accurate and essentially non-oscillatory in space, but it is also CFL time step stability limit free and has small time truncation error.

To be a bit more precise, in the Eulerian–Lagrangian framework, as one traces E to $\check{E}$, one sweeps out a space–time region. The integral form of (1.1) is applied over this space–time region. Since fluid flux across the two space–time side-lateral boundaries vanish, the integral of the mass over E at the advanced time level is equal to the integral of the mass over the non-grid cell $\check{E}$ at the earlier time level. The WENO reconstruction is applied at the earlier time level from grid cell average values. The novelty of our approach is that we devise a WENO reconstruction that targets high-order approximation of these trace-back integrals. In [18], a detailed study of WENO reconstruction and interpolation was given, including treatment of high-order integration given point values. The specific result we need is high-order integration given cell-average values, which we develop in this paper. We require a possibly different reconstruction for each subinterval arising from decomposing $\check{E}$ into the original Eulerian grid. In fact, we provide only the fifth-order reconstruction, since this was missing from the finite difference approach taken in [22].

Once it is understood how to approximate the one-dimensional problem (1.1), the higher dimensional problem can be handled by Strang splitting [7,25]. Now our grid cells are tensor-products of intervals, such as $E \times F$, and, for example, a Strang split solution in x would involve the trace-back region $\check{E} \times F$. An integral over the $\check{E}$ part of this region can be computed using our one-dimensional integral-based WENO reconstruction. However, the integration over F requires quadrature, and so a special treatment is needed to maintain local mass conservation. To maintain accuracy, we need also a traditional, high order, pointwise-based WENO reconstruction. We show formally that our two-dimensional, finite volume scheme is fifth-order convergent in space and locally mass conservative. Numerical examples also bear this out.

The paper is organized as follows. We provide the local mass conservation relation in Section 2, which is the foundation of all Eulerian–Lagrangian type schemes. In Section 3, we derive the linear and nonlinear weight functions for the WENO reconstruction procedure and define the reconstructed polynomials for fifth order accurate integration and interpolation. We define our locally conservative, finite volume scheme for one-dimensional problems in Section 4. A two space dimensional version is given in Section 5, using a Strang splitting technique. We also show the formal fifth order spatial accuracy of this two-dimensional, finite volume method and its local mass conservation. Sections 6 and 7 demonstrate the numerical performance of the proposed one and two space dimensional schemes. We conclude the paper in Section 8.

2. The local conservation relation

For $h > 0$, let the spatial grid be defined, respectively, by midpoints and grid points

$$x_i := ih \quad \text{and} \quad x_{i\pm1/2} := (i \pm 1/2)h$$

and then the grid cells or elements are

$$E_i := [x_{i-1/2}, x_{i+1/2}].$$

For $\Delta t > 0$ and any function $v(x,t)$, let the time levels be

$$t^n := n\Delta t \quad \text{and} \quad v_i^n := v(x_i, t^n).$$

Let $z(t;x)$ be the solution of the final value problem given by

$$\frac{dz}{dt} = a(z,t), \quad z(t^{n+1};x) = x \tag{2.1}$$

and set

$$\check{x}^n := \check{x}^n(x) := z(t^n;x). \tag{2.2}$$

We call $\check{x}^n$ the *trace-back* or *predecessor point* of x. Then the *trace-back* or *predecessor set* corresponding to E_i at time level t^n is defined as

$$\check{E}_i^n := \left[\check{x}_{i-1/2}^n, \check{x}_{i+1/2}^n\right] := \left[\check{x}^n(x_{i-1/2}), \check{x}^n(x_{i+1/2})\right]. \tag{2.3}$$

Define the space–time region $\mathcal{E}_i^{n+1} \subset \mathbb{R} \times (t^n, t^{n+1})$ to be the set contained between $E_i \times \{t^{n+1}\}$, $\check{E}_i^n \times \{t^n\}$, and the two integral curves $z(t;x_{i\pm1/2})$, $t^n < t < t^{n+1}$. If the lateral boundary of $\mathcal{E}_i^{n+1}$ is $\mathcal{S}_i^{n+1}$, so that $\partial\mathcal{E}_i^{n+1} = E_i \cup \check{E}_i^n \cup \mathcal{S}_i^{n+1}$, we note that the normal to $\partial\mathcal{E}_i^{n+1}$ along $\mathcal{S}_i^{n+1}$ is orthogonal to the vector $(u, a(x,t)u)$. Thus (1.1) and (1.2) and the (space–time) divergence theorem imply that

$$\int_{\mathcal{E}_i^{n+1}} \nabla_{t,x} \cdot \begin{pmatrix} u(x,t) \\ a(x,t)u(x,t) \end{pmatrix} dxdt = \int_{E_i} u^{n+1}(x)dx - \int_{\check{E}_i^n} u^n(x)dx = 0. \tag{2.4}$$

The approximation of the advection will be based on this relation expressing local mass conservation.

3. The WENO reconstructions

We will assume in this section that the trace-back points (and therefore the trace-back sets $\check{E}_i^n$) are found exactly. We will revisit this assumption later in Section 4 when we describe the time discretization. Let $\bar{u}_i^{n+1}$ be the finite volume numerical approximation

$$\bar{u}_i^{n+1} \approx \frac{1}{h}\int_{E_i} u^{n+1}(x)dx = \frac{1}{h}\int_{\check{E}_i^n} u^n(x)dx$$

by (2.4). Therefore we define $\bar{u}_i^{n+1}$ by

$$\bar{u}_i^{n+1} := \frac{1}{h}\int_{\check{E}_i^n} R^n(x;\bar{u}^n)dx, \tag{3.1}$$

where $R^n(x;\bar{u}^n)$ is a piecewise-polynomial reconstruction of $\left\{\bar{u}_j^n\right\}_j$ with local in space truncation error accuracy of $\mathcal{O}(h^{s+1})$ for smooth solutions. That is, we require

$$\int_{\check{E}_i^n} R^n(x;\bar{u}^n)dx = \int_{\check{E}_i^n} u^n(x)dx + \mathcal{O}(h^{s+1}) \tag{3.2}$$

assuming that the solution is correct up to time t^n. The main difficulty lies in determining how to define such a reconstruction $R^n(x;\bar{u}^n)$, and this is described below in Sections 3.1 and 3.2 for $s = 5$.

Once we have the reconstruction, by (2.4), we then have that

$$\sum_i \left|\frac{1}{h}\int_{E_i} u(x,t^{n+1})dx - \bar{u}_i^{n+1}\right| h = \mathcal{O}(h^s). \tag{3.3}$$

Moreover, having $\{\bar{u}_i^{n+1}\}_i$, we could apply a standard WENO reconstruction or postprocessing to define a piecewise-polynomial $q(x,t^{n+1})$ so that

$$q_i^{n+1} := q(\xi_i, t^{n+1}) = u^{n+1}(\xi_i) + \mathcal{O}(h^s)$$

for some points $\xi_i \in E_i$ (see Section 3.3), and then we have the pointwise estimate

$$\sup_i \left|u^{n+1}(\xi_i) - q_i^{n+1}\right| = \mathcal{O}(h^s). \tag{3.4}$$

Note that $q(x,t^{n+1})$ is not needed during the time stepping process, so this reconstruction can be omitted or computed only at time steps when it is otherwise needed.

3.1. Liner reconstruction for integration

In order to be precise, we provide a sixth order WENO reconstruction, giving a fifth order scheme, i.e., $s = 5$ in (3.2) above. Other order WENO reconstructions could be derived accordingly, see, e.g., [22].

For each cell $E_i = [x_{i-1/2}, x_{i+1/2}]$, we begin by constructing three polynomials, each of degree two, $P_{i-1}(x)$, $P_i(x)$, and $P_{i+1}(x)$. These are defined, for $k = i-1, i, i+1$, by posing the interpolation requirements

$$\frac{1}{h}\int_{E_j} P_k(x)dx = \bar{u}_j^n, \quad j = k-1, k, k+1, \tag{3.5}$$

where $\bar{u}_j^n$ are the known cell-average values. To be precise, let

$$L_{\pm1}(x) := \frac{1}{2}\left[\left(\frac{x}{h} \pm \frac{1}{2}\right)^2 - \frac{1}{3}\right], \quad L_0(x) := \frac{13}{12} - \left(\frac{x}{h}\right)^2$$

and then

$$P_k(x) := \sum_{j=-1}^{1} \bar{u}_{k+j}^n L_j(x - x_k), \quad k = i-1, i, i+1. \tag{3.6}$$

The full reconstruction is a convex combination of these three polynomials.

For our new method, we need to integrate a reconstructed polynomial over a typical trace-back set $\check{E}_j^n = \left[\check{x}_{j-1/2}^n, \check{x}_{j+1/2}^n\right]$. Therefore, we need to decompose $\check{E}_j^n$ into the grid cells $E_i = [x_{i-1/2}, x_{i+1/2}]$. This leads us to decompose $\check{E}_j^n$ into a union of the following four types of subintervals, where we write $x_\gamma = \gamma h$ for any $\gamma \in \mathbb{R}$:

1. $[x_{i-1/2}, x_{i+1/2}]$, when the whole E_i is contained in $\check{E}_j^n$;
2. $[x_{i-1/2}, x_{i-1/2+\alpha}]$, $0 < \alpha < 1$, when $\check{x}_{j+1/2}^n$ intersects E_i but not $\check{x}_{j-1/2}^n$;
3. $[x_{i-1/2+\beta}, x_{i+1/2}]$, $0 < \beta < 1$, when $\check{x}_{j-1/2}^n$ intersects E_i but not $\check{x}_{j+1/2}^n$;
4. $[x_{i-1/2+\beta}, x_{i-1/2+\alpha}]$, $0 < \beta < \alpha < 1$, when $\check{E}_j^n$ is contained in E_i.

Our goal is to find the corresponding linear weights for each subinterval of $\check{E}_j^n$ so that the higher order accuracy of the cell-average (3.2) is maintained.

We begin with a Type 2 subinterval $[x_{i-1/2}, x_{i-1/2+\alpha}]$. We will find the linear weight functions $C_k^i(\alpha)$, $k = i-1, i, i+1$, so that

$$\int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} P_i^{0,\alpha}(x)dx = \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} u(x,t^n)dx + \mathcal{O}(h^6), \tag{3.7}$$

where $P_i^{0,\alpha}$ is a second degree polynomial defined by

$$P_i^{0,\alpha}(x) := \sum_{k=i-1}^{i+1} C_k^i(\alpha)P_k(x) = \sum_{k=i-1}^{i+1} C_k^i(\alpha)\sum_{j=-1}^{1} \bar{u}_{k+j}^n L_j(x - x_k) = \sum_{\ell=i-2}^{i+2}\left(\sum_{k=\max(i,\ell)-1}^{\min(i,\ell)+1} C_k^i(\alpha)L_{\ell-k}(x - x_k)\right)\bar{u}_\ell^n. \tag{3.8}$$

To determine the $C_k^i(\alpha)$, we need an auxiliary construction. Let $U_5(x)$ be the polynomial of degree five so that

$$U_5(x_{i+k+1/2}) = \int_{x_{i-\frac{5}{2}}}^{x_{i+k+1/2}} u(x,t^n)dx, \quad k = -3, -2, -1, 0, 1, 2.$$

That is, with the standard Lagrange basis, scaled by h,

$$\mathcal{L}_j(x) := h\prod_{\substack{k=-3\\k\neq j}}^{2} \frac{x - (k+\frac{1}{2})h}{(j-k)h}, \quad j = -3, -2, -1, 0, 1, 2,$$

we have

$$U_5(x) = \sum_{j=-2}^{2} \mathcal{L}_j(x - x_i)\sum_{\ell=i-2}^{i+j} \bar{u}_\ell^n = \sum_{\ell=i-2}^{i+2}\left(\sum_{j=\ell-i}^{2} \mathcal{L}_j(x - x_i)\right)\bar{u}_\ell^n. \tag{3.9}$$

Since $U_5(x)$ interpolates six points of the integral of u, i.e., sums of the five cell-average values $\{\bar{u}_{i-2}, \bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, \bar{u}_{i+2}\}$ times h, we note that

$$U_5(x_{i-1/2+\alpha}) - U_5(x_{i-1/2}) = \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} u(x,t^n)dx + \mathcal{O}(h^6) = \sum_{\ell=i-2}^{i+2}\left(\sum_{j=\ell-i}^{2}\left[\mathcal{L}_j\left(\left(\alpha-\frac{1}{2}\right)h\right) - \mathcal{L}_j\left(-\frac{1}{2}h\right)\right]\right)\bar{u}_\ell^n. \tag{3.10}$$

It remains only to match the coefficients of $\bar{u}_\ell^n$ above with those from the integral of (3.8) over $[x_{i-1/2}, x_{i-1/2+\alpha}]$ to determine the $C_k^i(\alpha)$. Although tedious to calculate, the five equations are consistent, and the resulting three coefficients are

$$C_{i-1}^i(\alpha) := \frac{(2-\alpha)(3-\alpha)}{20}, \quad C_i^i(\alpha) := \frac{(2+\alpha)(3-\alpha)}{10}, \quad C_{i+1}^i(\alpha) := \frac{(1+\alpha)(2+\alpha)}{20}. \tag{3.11}$$

For a Type 1 subinterval $[x_{i-1/2}, x_{i+1/2}]$, any constant convex combination of linear weights that sum to one will work, since for $P_i^{0,1}(x) = \sum_{k=i-1}^{i+1} C_k^i(1)P_k(x)$,

$$\int_{x_{i-1/2}}^{x_{i+1/2}} P_i^{0,1}(x)dx = h\bar{u}_i^n = \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t^n)dx \tag{3.12}$$

by the definition of $P_i(x)$ (3.5). We shall choose

$$\left\{C_{i-1}^i(1), C_i^i(1), C_{i+1}^i(1)\right\} := \{1/10, 3/5, 3/10\},$$

which is consistent with Type 2 intervals, i.e., formulas (3.8) and (3.11) with $\alpha = 1$.

For a Type 3 subinterval $[x_{i-1/2+\beta}, x_{i+1/2}]$, by symmetry, we again use the same formulas (3.8) and (3.11) to define

$$P_i^{\beta,1}(x) := P_i^{0,\beta}(x) = \sum_{k=i-1}^{i+1} C_k^i(\beta)P_k(x). \tag{3.13}$$

Then we have, by (3.7) and (3.12),

$$\begin{aligned}\int_{x_{i-1/2+\beta}}^{x_{i+1/2}} P_i^{\beta,1}(x)dx &= \int_{x_{i-1/2}}^{x_{i+1/2}} P_i^{0,\beta}(x)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\beta}(x)dx = \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t^n)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} u(x,t^n)dx + \mathcal{O}(h^6)\\ &= \int_{x_{x_{i-1/2+\beta}}}^{x_{i+1/2}} u(x,t^n)dx + \mathcal{O}(h^6).\end{aligned} \tag{3.14}$$

Note that for $\alpha = \beta = 1/2$ in Types 2 and 3, our linear weight functions give the same set of constant linear weights {3/16, 5/8, 3/16} as in CWENO [16,20].

We finally turn to a Type 4 subinterval $[x_{i-1/2+\beta}, x_{i-1/2+\alpha}]$. Intuitively, one might wish to find similar linear weight functions $C_k^i(\beta,\alpha)$, $k = i-1, i, i+1$. But in fact, for arbitrary (β,α) it is not always possible do so, since $C_{i-1}^i(\beta,\alpha)$ or $C_{i+1}^i(\beta,\alpha)$ could be singular for certain values of β and α. Fig. 3.1 gives the plot of $C_{i-1}^i(\beta,\alpha)$ and $C_{i+1}^i(\beta,\alpha)$ where these are singular. From the plot it is easy to see that for any given $\beta > 0$ there is an $\alpha_1(\beta)$ so that $C_{i-1}^i(\beta,\alpha_1) = \infty$. Similarly there is an $\alpha_2(\beta)$ giving $C_{i+1}^i(\beta,\alpha_2) = \infty$.

In fact, Fig. 3.2 gives the feasible region of (α,β) so that $0 < C_k^i(\beta,\alpha) < 1$, $k = i-1, i, i+1$. One can read the plot as follows. Given a starting subinterval end point defined by β, the x-cross section is the feasible interval. For example, if $\beta = 0$, then the feasible interval is all of $[0,1]$, so any α is allowed. However, if $\beta = 0.5$, the feasible interval for α is only about $[0.65,1]$, and if $\beta = 0.2$, then the feasible interval for α is a union of two subintervals, i.e., about $[0.2,0.68]$ and $[0.9,1]$.

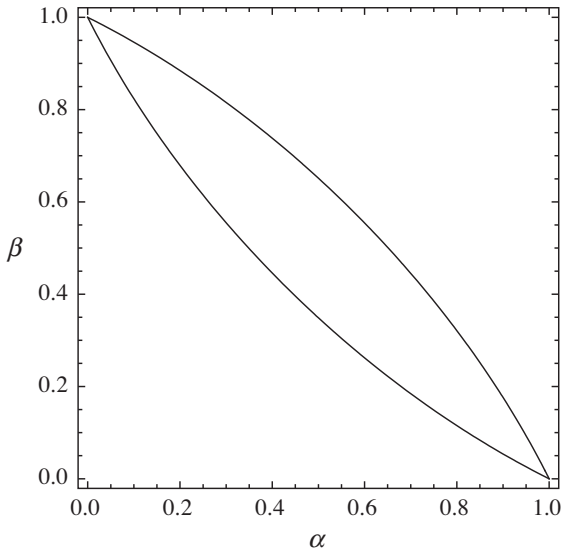


Fig. 3.1. The singularity plots for both $C_{i-1}^i(\beta,\alpha)$ (top) and $C_{i+1}^i(\beta,\alpha)$ (bottom).

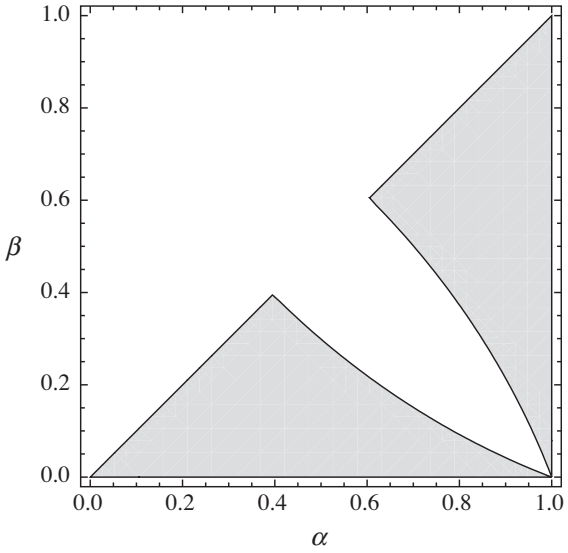


Fig. 3.2. The feasible region for $0 \leqslant C_k^i 417(\beta,\alpha)$, $k = i-1, 1, i+1$, when $\beta < \alpha$.

The point is that for an arbitrary subinterval $[x_{i-1/2+\beta}, x_{i-1/2+\alpha}]$, we cannot find feasible linear weights to define $P_i^{\beta,\alpha}(x)$ so that integration over the subinterval is preserved to fifth order. We can do this, as shown above, only when one of the interval's end points is a grid point. We present below another way to define the second order polynomial $P_i^{\beta,\alpha}(x)$ on $[x_{i-1/2+\beta}, x_{i-1/2+\alpha}]$ so that the integration remains high-order approximated, i.e.,

$$\int_{x_{i-1/2+\beta}}^{x_{i-1/2+\alpha}} P_i^{\beta,\alpha}(x)dx = \int_{x_{i-1/2+\beta}}^{x_{i-1/2+\alpha}} u(x,t^n)dx + \mathcal{O}(h^6). \tag{3.15}$$

From (3.7), we have

$$\begin{aligned}\int_{x_{i-1/2+\beta}}^{x_{i-1/2+\alpha}} u(x,t^n)dx &= \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} u(x,t^n)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} u(x,t^n)dx = \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} P_i^{0,\alpha}(x)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\beta}(x)dx + \mathcal{O}(h^6)\\ &= \int_{x_{i-1/2+\beta}}^{x_{i-1/2+\alpha}} P_i^{0,\alpha}(x)dx + \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} (P_i^{0,\alpha}(x) - P_i^{0,\beta}(x))dx + \mathcal{O}(h^6),\end{aligned} \tag{3.16}$$

where, again, $P_i^{0,\alpha}(x)$ and $P_i^{0,\beta}(x)$ are defined in (3.8). Therefore, we define

$$P_i^{\beta,\alpha}(x) := P_i^{0,\alpha}(x) + \mathcal{A} - \mathcal{B}, \tag{3.17}$$

where

$$\mathcal{A} := \frac{1}{(\alpha-\beta)h}\int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\alpha}(x)dx, \tag{3.18}$$

$$\mathcal{B} := \frac{1}{(\alpha-\beta)h}\int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\beta}(x)dx. \tag{3.19}$$

Then we have (3.15) as desired. Moreover, our definition does not create extra computational costs, since

$$\int_{x_{i-1/2+\beta}}^{x_{i-1/2+\alpha}} P_i^{\beta,\alpha}(x)dx = \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} P_i^{0,\alpha}(x)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\beta}(x)dx. \tag{3.20}$$

To compute the left-hand side above, we only need to compute the middle term, since the last term would have been computed when we handled the previous grid cell.

Although everything is defined implicitly above, to complete the construction, we define the full reconstruction piecewise-polynomial $P^n(x;\bar{u}^n)$ on $\check{E}_j^n$. First, let the left and right endpoints of $E_j = [x_{j-1/2}, x_{j+1/2}]$ be traced back to $\check{E}_j^n = \left[\check{x}_{j-1/2}^n, \check{x}_{j+1/2}^n\right]$, and identify the grid indices i_L and i_R, where

$$x_{i_L-1/2} \leqslant \check{x}_{j-1/2}^n < x_{i_L+1/2} \quad \text{and} \quad x_{i_R-1/2} < \check{x}_{j+1/2}^n \leqslant x_{i_R+1/2}.$$

Then define α_k and β_k so that $0 \leqslant \beta_k < \alpha_k \leqslant 1$ and

$$\check{E}_j^n = \bigcup_{k=0}^{i_R-i_L} [x_{i_L+k-1/2+\beta_k}, x_{i_L+k-1/2+\alpha_k}]. \tag{3.21}$$

Finally, $P^n(x;\bar{u}^n)$ is defined as

$$P^n(x;\bar{u}^n) := \sum_{k=0}^{i_R-i_L} P_{i_L+k}^{\beta_k,\alpha_k}(x)\chi_{[x_{i_L+k-1/2+\beta_k}, x_{i_L+k-1/2+\alpha_k}]}(x), \quad x \in \check{E}_j^n, \tag{3.22}$$

where $\chi_S(x)$ is the characteristic function of S and the $P_i^{\beta_k,\alpha_k}(x)$ are defined for Types 1–3 subintervals ($\beta_k = 0$ and/or $\alpha_k = 1$) by (3.8), (3.11) and (3.13), and for Type 4 subintervals ($0 < \beta_k < \alpha_k < 1$) by (3.17)–(3.19). Moreover, we have

$$\int_{\check{E}_j^n} P^n(x;\bar{u}^n)dx = \int_{\check{E}_j^n} u(x,t^n)dx + \mathcal{O}(h^6). \tag{3.23}$$

We emphasize that our construction gives a piecewise-polynomial, so we may have more than one set of linear weights and reconstruction polynomials in a single grid cell E_i. This is very different from the traditional WENO reconstruction. However, it does not pose any difficulty or inconsistency, since we only want to find the correct approximation of the integral of the reconstruction over a particular subinterval, not the approximation of the reconstruction itself at points within the grid cell. This is the key to the success of the scheme. Traditional WENO requires that one compute the left and right fluxes from the reconstruction, and therefore consistency is required for all the reconstruction polynomials in a single grid cell.

Lemma 3.1. *The number of integration computations needed is only at most one plus the number of computational grid cells.*

Proof. This result is due to two simple facts. Firstly, we note that there is no need to compute the integration of $P^n(x;\bar{u}^n)$ on a Type 1 subinterval E_i, since it is equal to $h\bar{u}_i$. Secondly, integration of a Type 3 subinterval can be obtained by

$$\int_{x_{i-1/2+\beta}}^{x_{i+1/2}} P_i^{\beta,1}(x)dx = \int_{x_{i-1/2}}^{x_{i+1/2}} P_i^{\beta,1}(x)dx - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{\beta,1}(x)dx = h\bar{u}_i^n - \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} P_i^{0,\beta}(x)dx. \tag{3.24}$$

Thus, except the very first cell, the main computation has already been computed for the previous cell. Therefore, the scheme is very efficient. □

Lemma 3.2. *Mass is conserved locally by the linear reconstruction; that is,*

$$\int_{E_i} P^n(x;\bar{u}^n)dx = h\bar{u}_i^n \quad \textit{for all } i.$$

Proof. Consider a grid cell E_i. Let

$$E_i \cap \left\{\check{x}_{j-1/2}^n\right\}_j = \{x_{i-1/2+\alpha_k}\}_{k=0}^N$$

be the set of trace-back grid points that intersect E_i, and the endpoints of E_i, where we order the set so that $0 = \alpha_0 < \alpha_1 < \cdots < \alpha_N = 1$. Then

$$\int_{E_i} P^n(x;\bar{u}^n)dx = \sum_{k=0}^{N-1} \int_{x_{i-1/2+\alpha_k}}^{x_{i-1/2+\alpha_{k+1}}} P_i^{\alpha_k,\alpha_{k+1}}(x,t^n)dx.$$

If $N = 1$, E_i is a Type 1 subinterval, and the result was noted earlier. Otherwise, (3.20) shows that the sum collapses for any Type 4 subintervals that we may have, and (3.24) shows that the initial Type 2 and final Type 3 subintervals combine to give the claim. □

3.2. WENO reconstruction for integration

The WENO reconstruction is achieved by first computing the usual smoothness indicator

$$IS_k^i := \sum_{l=1}^{2} \int_{x_{i-1/2}}^{x_{i+1/2}} h^{2l-1}\left(\frac{\partial^l P_k(x)}{\partial x^l}\right)^2 dx, \quad k = i-1, i, i+1.$$

An explicit integration yields

$$IS_{i-1}^i = \frac{13}{12}(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i)^2 + \frac{1}{4}(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i)^2, \tag{3.25}$$

$$IS_i^i = \frac{13}{12}(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1})^2 + \frac{1}{4}(\bar{u}_{i-1} - \bar{u}_{i+1})^2, \tag{3.26}$$

$$IS_{i+1}^i = \frac{13}{12}(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2})^2 + \frac{1}{4}(3\bar{u}_i - 4\bar{u}_{i+1} + \bar{u}_{i+2})^2. \tag{3.27}$$

The nonlinear weights $\omega_k^i(\alpha)$ are

$$\omega_k^i(\alpha) := \frac{\gamma_k^i(\alpha)}{\gamma_{i-1}^i(\alpha) + \gamma_i^i(\alpha) + \gamma_{i+1}^i(\alpha)}, \quad k = i-1, i, i+1, \tag{3.28}$$

where

$$\gamma_k^i(\alpha) := \frac{C_k^i(\alpha)}{\left(\epsilon + IS_k^i\right)^2}, \quad k = i-1, i, i+1 \tag{3.29}$$

and $\epsilon > 0$ is small, taken to be 10^{-6} in our numerical tests. Although we may have more than one set of WENO weights in a single interval E_i, the smoothness indicator is uniquely defined, since it represents the smoothness of the input cell-average values $\bar{u}_k$.

Now for a Type 1 or 2 subinterval $[x_{i-1/2}, x_{i-1/2+\alpha}]$, we define the piecewise-polynomial reconstruction $R_i^{0,\alpha}(x)$ by

$$R_i^{0,\alpha}(x) := \sum_{k=i-1}^{i+1} \omega_k^i(\alpha) P_k(x). \tag{3.30}$$

We have

$$\int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} R_i^{0,\alpha}(x)dx = \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} u(x,t^n)dx + \mathcal{O}(h^6), \tag{3.31}$$

because

$$\begin{aligned}\int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} \left(R_i^{0,\alpha}(x) - P_i^{0,\alpha}(x)\right)dx &= \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} \sum_{k=i-1}^{i+1} \left(\omega_k^i(\alpha) - C_k^i(\alpha)\right)P_k(x)dx \\ &= \int_{x_{i-1/2}}^{x_{i-1/2+\alpha}} \sum_{k=i-1}^{i+1} \left(\omega_k^i(\alpha) - C_k^i(\alpha)\right)(P_k(x) - u(x,t^n))dx = \mathcal{O}(h^6)\end{aligned} \tag{3.32}$$

and, for $k = i-1, i, i+1$, $\omega_k^i(\alpha) = C_k^i(\alpha) + \mathcal{O}(h^2)$ and each $P_k(x)$ is a parabolic reconstruction that is fourth order accurate in the integration of $u(x,t^n)$ over our subinterval. Similarly, for a Type 3 subinterval $[x_{i-1/2+\beta}, x_{i+1/2}]$, we define

$$R_i^{\beta,1}(x) := \sum_{k=i-1}^{i+1} \omega_k^i(\beta) P_k(x) = R_i^{0,\beta}(x) \tag{3.33}$$

and note that

$$\int_{x_{i-1/2+\beta}}^{x_{i+1/2}} R_i^{\beta,1}(x)dx = \int_{x_{i-1/2+\beta}}^{x_{i+1/2}} u(x,t^n)dx + \mathcal{O}(h^6). \tag{3.34}$$

Finally, for a Type 4 subinterval $[x_{i-1/2+\beta}, x_{i-1/2+\alpha}]$, we define

$$R_i^{\beta,\alpha}(x) := R_i^{0,\alpha}(x) + \mathcal{A}_w - \mathcal{B}_w, \tag{3.35}$$

$$\mathcal{A}_w := \frac{1}{(\alpha-\beta)h} \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} R_i^{0,\alpha}(x)dx, \tag{3.36}$$

$$\mathcal{B}_w := \frac{1}{(\alpha-\beta)h} \int_{x_{i-1/2}}^{x_{i-1/2+\beta}} R_i^{0,\beta}(x)dx \tag{3.37}$$

and then

$$\int_{i-1/2+\beta}^{i-1/2+\alpha} R_i^{\beta,\alpha}(x)dx = \int_{i-1/2+\beta}^{i-1/2+\alpha} u(x,t^n)dx + \mathcal{O}(h^6). \tag{3.38}$$

This result could be obtained similarly as (3.16).

Finally, we define $R^n(x;\bar{u}^n)$ similar to $P^n(x;\bar{u}^n)$ in (3.22). Using the decomposition (3.21) of $\check{E}_j^n$, let

$$R^n(x;\bar{u}^n) := \sum_{k=0}^{i_R-i_L} R_{i_L+k}^{\beta_k,\alpha_k}(x)\chi_{\left[x_{i_L+k-1/2+\beta_k}, x_{i_L+k-1/2+\alpha_k}\right]}(x), \quad x \in \check{E}_j^n. \tag{3.39}$$

In summary, for the scheme (3.1), we have obtained (3.2), and therefore (3.3), i.e., we have shown the local truncation error estimate for smooth solutions. Moreover, by reasoning similar to the proof of Lemma 3.2, we see that mass is conserved locally. We have proved the following result.

Lemma 3.3. *The local truncation error of the reconstruction is $\mathcal{O}(h^5)$, so also for the method; that is, (3.3) holds with $s = 5$. Moreover, mass is conserved locally by the reconstruction; that is,*

$$\int_{E_i} R^n(x;\bar{u}^n)dx = h\bar{u}_i^n \quad \textit{for all } i.$$

3.3. Optional WENO postprocessing for pointwise values

Optionally, one may postprocess the cell-average values $\bar{u}_i^{n+1}$ to obtain a high order pointwise approximation $q(x,t^{n+1})$ of $u(x,t^{n+1})$ at certain points using a standard WENO reconstruction, similar to that done above. For completeness, we briefly review the process. However, the reader should realize that this step is optional, since higher order pointwise values do not enter into the time stepping computation.

For a fixed time level, the standard WENO reconstruction starts with the cell average values $\bar{u}_i^{n+1}$ on E_i, and produces (in our case) a fifth order accurate approximation of $u(x,t^{n+1})$ at the predetermined points $\xi_i \in E_i$ for all i. Normally $\xi_i = x_{i\pm1/2}$ for WENO and $\xi_i = x_i$ for CWENO.

For example, we will take $\xi_i = x_{i+1/2}$. We use $P_k(x)$, $k = i-1, i, i+1$, defined in (3.6) and $U_5(x)$ constructed above in (3.9) with $\bar{u}_i^{n+1}$ in place of $\bar{u}_i^n$. We find the linear weights C_k^i so that

$$\sum_{k=i-1}^{i+1} C_k^i P_k(\xi_i) = U_5'(\xi),$$

which in the case $\xi_i = x_{i+1/2}$ is $\left\{C_{i-1}^i, C_i^i, C_{i+1}^i\right\} = \{1/10, 3/5, 3/10\}$. We then modify the weights using the nonlinear weight procedure in Section 3.2 above. We achieve a fifth order accurate approximation $q(x,t^{n+1})$ at each ξ_i.

If we choose $\xi_i = x_i$ instead, we would have the linear weights $\left\{C_{i-1}^i, C_i^i, C_{i+1}^i\right\} = \{-9/80, 49/40, -9/80\}$. Because some weights are negative, a splitting technique [24] needs to be applied in this case.

Because the $\bar{u}_i^{n+1}$ are not exact cell-averages, it is not clear that our postprocessed reconstruction is $\mathcal{O}(h^5)$, as claimed. What we have shown from (3.1) and (3.2) is that, locally,

$$\int_{E_i} \left(\bar{u}_i^{n+1} - u(x,t^{n+1})\right)dx = \mathcal{O}(h^6);$$

that is, $h\bar{u}_i^{n+1}$ is sixth order accurate. Let $V_5(x)$ be the polynomial analogous to $U_5(x)$ that is defined by (3.9) with exact cell averages $\frac{1}{h}\int_{E_i} u(x,t^{n+1})dx$ in place of $\bar{u}_i^n$. Therefore on E_i,

$$|U_5(x) - V_5(x)| = \sum_{\ell=i-2}^{i+2}\left(\sum_{j=\ell-i}^{2}\frac{1}{h}\mathcal{L}_j(x-x_i)\right)\left(h\bar{u}_\ell^n - \int_{E_\ell} u(x,t^{n+1})dx\right)$$

maintains $\mathcal{O}(h^6)$ accuracy, since

$$\lambda = \max_{E_i}\sum_{j=-2}^{2}\left|\frac{1}{h}\mathcal{L}_j(x-x_i)\right|$$

is bounded. This means that although we use perturbed data for our interpolation, we still maintain the desired order of accuracy for U_5, and also its derivative. This justifies the accuracy of our standard WENO postprocessing reconstruction.

4. The finite volume procedure in one space dimension

We define the full finite volume procedure in this section. In general, the trace-back points $\check{x}_i^n$ can not be found analytically. We need to use some approximate ODE solver to solve (2.1). While any reasonable solver should work well, we chose to use a fourth order Runge–Kutta method. Note that we can use micro-stepping to solve (2.1) over the time step $[t^n, t^{n+1}]$ if $\check{x}_i^n$ are not accurate enough, since $a(x,t)$ is independent of the solution. With the approximated trace-back points $\tilde{X}_{i+1/2}^n$, and therefore the approximated trace-back regions $\widetilde{E}_i^n = \left[\tilde{X}_{i-1/2}^n, \tilde{X}_{i+1/2}^n\right]$, the WENO reconstruction procedure given in Section 3 can be computed.

Assume that the cell-averages $\{\overline{U}_i^0\}_i$ are obtained from the initial condition $u_0(x)$ and $\{\overline{U}_i^n\}_i$ have been derived up to time level n. For each $E_i = [x_{i-1/2}, x_{i+1/2}]$, we find the approximated trace-back set $\widetilde{E}_i^n = \left[\tilde{X}_{i-1/2}^n, \tilde{X}_{i+1/2}^n\right]$, and define $\overline{U}_i^{n+1}$ by

$$\overline{U}_i^{n+1} := \frac{1}{h}\int_{\widetilde{E}_i^n} \widetilde{R}^n(x;\overline{U}^n)dx, \tag{4.1}$$

where $\widetilde{R}^n(x;\overline{U}^n)$ is the WENO piecewise-polynomial reconstructed from $\{\overline{U}_i^n\}_i$ for integration as described in Section 3, but using the approximated trace-back sets $\widetilde{E}_i^n$. After $\overline{U}_i^{n+1}$ is obtained, we may apply the WENO postprocessing reconstruction for pointwise values, if desired. This completes our scheme.

Note that the only additional error is from approximating the trace-back regions. An algorithm as in [2] should be used to adjust the approximate trace-back set volume. Also, an analogue of the argument in [9] and/or [3] should provide a rigorous error analysis for the overall scheme.

Theorem 4.1. *The method has formal $\mathcal{O}(h^5)$ accuracy, and it is locally mass conservative.*

Proof. The accuracy follows directly from the truncation error part of Lemma 3.3 (which continues to hold for $\widetilde{R}^n$ in place of R^n, assuming a sufficiently accurate characteristic tracing). The local mass conservation follows in a Lagrangian setting directly from the mass conservation part of Lemma 3.3, since the reconstruction is locally conservative and all mass is accounted for within the approximate space–time regions $\widetilde{\mathcal{E}}_i^{n+1}$ (see (2.4)).

We can also demonstrate the conservation in an Eulerian setting as follows, again using Lemma 3.3. The scheme is

$$\begin{aligned} h\overline{U}_i^{n+1} &= \int_{\widetilde{E}_i^n} \widetilde{R}^n(x;\overline{U}^n)dx = \int_{E_i} \widetilde{R}^n(x;\overline{U}^n)dx + \int_{\widetilde{E}_i^n} \widetilde{R}^n(x;\overline{U}^n)dx - \int_{E_i} \widetilde{R}^n(x;\overline{U}^n)dx \\ &= h\overline{U}_i^n + \int_{\check{X}_{i-1/2}^n}^{\check{X}_{i+1/2}^n} \widetilde{R}^n(x;\overline{U}^n)dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \widetilde{R}^n(x;\overline{U}^n)dx = h\overline{U}_i^n + \int_{\hat{X}_{i-1/2}^n}^{x_{i-1/2}} \widetilde{R}^n(x;\overline{U}^n)dx - \int_{\hat{X}_{i+1/2}^n}^{x_{i+1/2}} \widetilde{R}^n(x;\overline{U}^n)dx, \end{aligned} \tag{4.2}$$

which is in conservative form. □

5. A method for two-dimensional problems

In this section, we extend the one-dimensional method to multiple dimensions. For simplicity, we extend only to two-dimensions, since three and higher higher dimensional cases follow easily from the two-dimensional case. That is, we approximate the unknown function $u(x,y,t)$ satisfying the equation

$$\frac{\partial u}{\partial t} + \frac{\partial(a_1 u)}{\partial x} + \frac{\partial(a_2 u)}{\partial y} = 0, \quad (x,y) \in \mathbb{R}^2, t > 0, \tag{5.1}$$

$$u(x,y,0) = u_0(x,y), \quad (x,y) \in \mathbb{R}^2, \tag{5.2}$$

where $a_1(x,y,t)$, $a_2(x,y,t)$, and $u_0(x,y)$ are given. We use a Strang splitting technique [7,25] in space to decouple the problem into two one dimensional problems.

For a first order in time splitting, over the time interval $[t^n, t^{n+1}]$, we approximate in two steps. First, for each fixed $y \in \mathbb{R}$, from the current value of the approximate solution $u^n(x,y)$ as the initial condition at time t^n, we approximate the x-sweep

$$\frac{\partial u}{\partial t} + \frac{\partial(a_1 u)}{\partial x} = 0, \quad u(x,y,t^n) = u^n(x,y), \quad x \in \mathbb{R}, \ t \in (t^n, t^{n+1}] \tag{5.3}$$

for $\tilde{u}^{n+1}(x,y)$. Second, for each fixed $x \in \mathbb{R}$, from $\tilde{u}^{n+1}(x,y)$ as initial condition at time t^n, we approximate the y-sweep

$$\frac{\partial u}{\partial t} + \frac{\partial(a_2 u)}{\partial y} = 0, \quad u(x,y,t^n) = \tilde{u}^{n+1}(x,y), \quad y \in \mathbb{R}, t \in (t^n, t^{n+1}]. \tag{5.4}$$

The result is the final approximation $u^{n+1}(x,y)$.

A second order Strang splitting requires three steps. First, one solves the x sweep (5.3) only over the time interval $(t^n, t^{n+1/2}]$, where $t^{n+1/2} := t^n + \Delta t/2$, for $\tilde{u}^{n+1/2}(x,y)$. The second step is the same as the y-sweep (5.4) above, but starting from the initial condition $\tilde{u}^{n+1/2}(x,y)$ and resulting in $\tilde{\tilde{u}}^{n+1}(x,y)$. The third step is to solve the rest of the x-sweep (5.3) over the time interval $(t^{n+1/2}, t^{n+1}]$, starting from the initial condition $\tilde{\tilde{u}}^{n+1}(x,y)$, for the final solution $u^{n+1}(x,y)$.

Thus, we need only provide a way to solve the one-dimensional x- and y-sweep problems (5.3) and (5.4), each of which are the same as (1.1), and also to limit the number of y points required for the x-sweep(s) and x points required for the y-sweep. We describe only the approximation of the x-sweep, since the y sweep is then defined by symmetry. Moreover, we define only the full x-sweep, since the half sweep needed for the second order splitting is then clear.

5.1. The two-dimensional scheme

The x-grid has been defined in Section 2, and for simplicity, we define the y-grid similarly using the same $h > 0$ and the y-grid cells or elements

$$F_j := [y_{j-1/2}, y_{j+1/2}].$$

As with any finite volume scheme, we approximate the average of u,

$$\bar{u}_{i,j}^n := \frac{1}{h^2}\int_{F_j}\int_{E_i} u(x,y,t^n)dxdy$$

by $\overline{U}_{i,j}^n$, defined below.

For any fixed y, the final value problem

$$\frac{dz}{dt} = a_1(z,y,t), \quad z(t^{n+1};x,y) = x \tag{5.5}$$

can be used to define the exact trace-back point of x to time t^n as $\check{x}^n(x,y) := z(t^n;x,y)$. The trace-back set corresponding to E_i at time level t^n is then

$$\check{E}_i^n(y) := [\check{x}^n(x_{i-1/2},y).\check{x}^n(x_{i+1/2},y)]. \tag{5.6}$$

Similarly, we can approximate the solution to (5.5) as $\tilde{z}$ and define the approximate trace-back points $\check{X}^n(x;y) := \tilde{z}(t^n;x,y)$ and the approximate trace-back set

$$\widetilde{E}_i^n(y) := [\check{X}^n(x_{i-1/2},y),\check{X}^n(x_{i+1/2},y)] \tag{5.7}$$

possibly adjusted for volume conservation as described in [2] and as noted above in Section 4.

Analogous to (2.4), we have that

$$\int_{E_i} u^{n+1}(x,y)dx - \int_{\widetilde{E}_i^n(y)} u^n(x,y)dx = 0. \tag{5.8}$$

This equation suggests that we would like to define approximate cell averages by

$$\overline{U}_{i,j}^{n+1} \approx \frac{1}{h^2}\int_{F_j}\int_{\widetilde{E}_i^n(y)} u^n(x,y)dxdy.$$

Three problems arise. First, we cannot compute the integral involving every y, so we will use a quadrature formula. Second, in place of u^n we need to use a reconstruction of the approximate solution at time t^n, and we will use one similar to that presented above for the one-dimensional case. Third, our reconstruction must be made from the two-dimensional cell averages. To maintain high order accuracy, we will need to include a second, y-reconstruction technique in the method.

Since we aim for $\mathcal{O}(h^5)$ accuracy, we use a three point Gauss rule to approximate the integral in y over F_j. Let the Gauss points and corresponding weights be $y_k^j \in F_j$ and w_k^j, $k = -1, 0, 1$. Analogous to (4.1), we can state the x-sweep as

$$\overline{U}_{i,j}^{n+1} := \frac{1}{h^2}\sum_k w_k^j \int_{\widetilde{E}_i^n(y_k^j)} \widetilde{R}^{n,j}\left(x, y_k^j; \overline{U}^n\right)dx, \tag{5.9}$$

where it remains to define the reconstruction $\widetilde{R}^{n,j}\left(x, y_k^j; \overline{U}^n\right)$.

If desired, at any time level, a pointwise WENO postprocessing reconstruction can be applied to obtain higher order pointwise values. This is done below in Section 7 to define a discrete maximum norm.

5.2. The two-dimensional reconstruction for integration

We begin with a reconstruction targeting high order pointwise approximation. Fix a y-grid cell F_j and a Gauss point y_k^j. Let $\mathcal{R}_k^j(y;\bar{v})$ be a one-dimensional, piecewise-polynomial, WENO reconstruction in the y-direction of the values $\{\bar{v}_\ell\}_\ell$ which is $\mathcal{O}(h^5)$ accurate at the Gauss point y_k^j.

To be specific, on the interval $F_j = [y_{j-1/2}, y_{j+1/2}]$, let $y_{\pm1}^j = y_j \pm h\sqrt{3/5}$ and $y_0^j = y^j$ be the three Gaussian quadrature points. The linear reconstruction is given basically in (3.5) and (3.6) as

$$\mathcal{R}_{\text{linear},k}^j(y;\bar{v}) = \sum_{\ell=j-1}^{j+1} r_{\ell-j}\sum_{m=-1}^{1}\bar{v}_{\ell+m}L_m(y-y_\ell), \tag{5.10}$$

wherein the linear weights $r_{\ell-j}$ for $y_{\pm1}^j$ are

$$r_{-1} = \frac{\mp9+22\sqrt{15}}{40(\pm2+3\sqrt{15})}, \quad r_0 = 1-r_{-1}-r_1, \quad \text{and} \quad r_1 = \frac{\pm9+22\sqrt{15}}{40(\mp2+3\sqrt{15})}$$

and for y_0^j, $r_{\pm1} = -9/80$ and $r_0 = 98/80$. The weights for $y_{\pm1}^j$ are all between 0 and 1, and so a nonlinear WENO modification can be applied to define $\mathcal{R}_k^j(y;\bar{v})$. The weights for y_0^j includes negative values, but [24] explains how to treat such negative weights in a WENO reconstruction. However, we do not use $\mathcal{R}_0^j(y;\bar{v})$.

For each fixed x-index i, we reconstruct in y as follows. For each Gauss point index $k = \pm1$, we set

$$V_i^{n,j,\pm1} := \mathcal{R}^j_{\pm1}\left(y^j_{\pm1}; \overline{U}^n_{i,(\cdot)}\right). \tag{5.11}$$

We will show later in (5.17) that these are $\mathcal{O}(h^5)$ approximations of the average mass in the interval E_i, for the given point $y = y^j_k$. For index $k = 0$, however, we use the definition

$$V_i^{n,j,0} = \frac{1}{w^j_0}\left[h\overline{U}^n_{i,j} - w^j_{-1}V_i^{n,j,-1} - w^j_1 V_i^{n,j,1}\right], \tag{5.12}$$

so that mass is conserved locally under the Gauss quadrature rule; that is,

$$\sum_k w^j_k V_i^{n,j,k} = \overline{U}^n_{i,j}h. \tag{5.13}$$

We now complete the description of the x-sweep (5.9) by defining

$$\widetilde{R}^{n,j}\left(x, y^j_k; \overline{U}^n\right) := \widetilde{R}^n(x; V^{n,j,k}), \tag{5.14}$$

where the one-dimensional reconstruction for integration $\widetilde{R}^n$ is the one used in (4.1).

We remark that one could use (5.11) to define $V_i^{n,j,0}$ in place of (5.12). As one can see from the next subsection, formal $\mathcal{O}(h^5)$ accuracy would be obtained, but the mass balance would only be $\mathcal{O}(h^5)$ accurate. Moreover, if the linear reconstruction $\mathcal{R}^j_{\text{linear},k}(y;\bar{v})$ in (5.10) were to be used, then the use of (5.11) to define $V_i^{n,j,0}$ would result in both formal $\mathcal{O}(h^5)$ accuracy and mass conservation. However, the scheme presented here is designed to both reduce oscillation and maintain mass conservation. It is also relatively efficient, as the following lemma records.

Lemma 5.1. *For a full x-sweep on an $m \times n$ grid, the main computations needed are (1) 2mn full WENO reconstructions in y given by (5.11) and mn simple reconstructions in y given by (5.12), (2) using these values as in (5.14), 3n solutions of the one-dimensional method on a grid with m elements, and finally (3) n compilation of one-dimensional results in (5.9).*

5.3. Formal accuracy

We have the following result.

Theorem 5.2. *The two-dimensional EL–WENO finite volume scheme (5.9) is formally $\mathcal{O}(h^5)$ accurate in space. Moreover, it is locally mass conservative.*

Proof. In light of (5.8), we note that

$$h^2\bar{u}^{n+1}_{i,j} := \int_{F_j}\int_{E_i} u^{n+1}(x,y)dxdy = \int_{F_j}\int_{\tilde{E}^n_i(y)} u^n(x,y)dxdy. \tag{5.15}$$

We claim that

$$\int_{F_j}\int_{\tilde{E}^n_i(y)} u^n(x,y)dxdy = \sum_k w^j_k \int_{\widetilde{E^n_i}\left(y^j_k\right)} \widetilde{R}^{n,j}\left(x, y^j_k; \bar{u}^n\right)dx + \mathcal{O}(h^7), \tag{5.16}$$

which will prove the local truncation error or formal accuracy of our scheme (5.9) is $\mathcal{O}(h^5)$. The quadrature and approximate trace-back error (assuming sufficiently accurate characteristic tracing) is simple to account for:

$$\int_{F_j}\int_{\tilde{E}^n_i(y)} u^n(x,y)dxdy = \sum_k w^j_k \int_{\widetilde{E^n_i}\left(y^j_k\right)} u^n\left(x, y^j_k\right)dx + \mathcal{O}(h^7).$$

Consider next the reconstruction in y. For each index j, let

$$\bar{v}^n_j(x) := \frac{1}{h}\int_{F_j} u^n(x,y)dy$$

be the average mass in F_j for the given value of x. Our one-dimensional reconstruction $\mathcal{R}^j_k(y;\bar{v}^n(x))$ is accurate in y for such average values at the Gauss point y^j_k, i.e.,

$$\mathcal{R}^j_k\left(y^j_k; \bar{v}^n(x)\right) = u^n\left(x, y^j_k\right) + \mathcal{O}(h^5).$$

However, by definition (see (5.10)), the reconstruction operator does not depend explicitly on x, and it is linear in the $\bar{v}^n(x)$, so in fact

$$\frac{1}{h}\int_{E_i}\mathcal{R}^j_k\left(y^j_k; \bar{v}^n(x)\right)dx = \mathcal{R}^j_k\left(y^j_k; \frac{1}{h}\int_{E_i}\bar{v}^n(x)dx\right) = \mathcal{R}^j_k\left(y^j_k; \bar{u}^n_{i,(\cdot)}\right),$$

since

$$\frac{1}{h}\int_{E_i}\bar{v}_j^n(x)dx=\bar{u}_{i,j}^n=\frac{1}{h^2}\int_{F_j}\int_{E_i}u(x,y,t^n)dxdy.$$

Therefore, the y reconstruction accurately approximates the x-grid cell averages, i.e.,

$$\mathcal{R}_k^j\left(y_k^j;\bar{u}_{i,(\cdot)}^n\right)=\tilde{v}^n\left(y_k^j\right)+\mathcal{O}(h^5), \tag{5.17}$$

where we define

$$\tilde{v}_i^n(y):=\frac{1}{h}\int_{E_i}u^n(x,y)dx.$$

So, if $v_i^{n,j,k}$ is defined from (5.11) and (5.12) using $\bar{u}_{i,(\cdot)}^n$ in place of $\overline{U}_{i,(\cdot)}^n$, then

$$\tilde{v}_i^n\left(y_k^j\right)=v_i^{n,j,k}+\mathcal{O}(h^5),\quad k=\pm1.$$

We need to treat the case $k=0$ specially. We compute

$$\begin{aligned}v_i^{n,j,0}-\tilde{v}_i^n(y_k^0)&=\frac{1}{w_0^j}\left[h\overline{U}_{i,j}^n-w_{-1}^j v_i^{n,j,-1}-w_1^j v_i^{n,j,1}-w_0^j\tilde{v}_i^n(y_k^0)\right]=\frac{1}{w_0^j}\left[h\overline{U}_{i,j}^n-\sum_k w_k^j\tilde{v}_i^n(y_k^k)\right]+\mathcal{O}(h^5)\\&=\frac{1}{w_0^j}\left[h\overline{U}_{i,j}^n-\int_{F_j}\tilde{v}_i^n(y)dy\right]+\mathcal{O}(h^5)=\mathcal{O}(h^5)\end{aligned}$$

using that the Gauss weights are $\mathcal{O}(h)$ and noting that the last quantity in brackets above vanishes identically. Thus, for all k,

$$\tilde{v}_i^n\left(y_k^j\right)=v_i^{n,j,k}+\mathcal{O}(h^5).$$

Finally, the one-dimensional reconstruction for integration in x, $\widetilde{R}^n$, was seen earlier to be formally accurate for each fixed y, so we have at the Gauss points that

$$\int_{\widetilde{E}_i^n(y_k^j)}u^n(x,y_k^j)dx=\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^n(x;\tilde{v}^n\left(y_k^j\right))dx+\mathcal{O}(h^6)=\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^n(x;v^{n,j,k})dx+\mathcal{O}(h^6)=\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^{n,j}(x,y_k^j;\bar{u}^n)dx+\mathcal{O}(h^6)$$

using (5.14) in the last equality. Since the Gauss weights are $\mathcal{O}(h)$, the claim (5.16), and therefore the $\mathcal{O}(h^5)$ formal accuracy of the method, follows.

The proof of the local mass conservation is based on three facts. First, during the x-sweep, no mass crosses the y-faces; that is, mass is constrained locally to the strip $\mathbb{R}\times F_j$ for each j. Second, all mass is accounted for in the quadrature rule over F_j by the $V_i^{n,j,k}$, as we saw in (5.13). Finally, each one-dimensional transport in the x-direction is locally mass conservative by Theorem 4.1. We can also express the scheme (5.9) conservatively in an Eulerian setting. Using (5.14) and (5.13), we have

$$h^2\overline{U}_{i,j}^{n+1}=\sum_k w_k^j\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^n(x;V^{n,j,k})dx=h^2\overline{U}_{i,j}^n+\sum_k w_k^j\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^n(x;V^{n,j,k})dx-h\sum_k w_k^j V_i^{n,j,k}.$$

Now Lemma 3.3 (for the approximate characteristics, i.e., $\widetilde{R}^n$ in place of R^n) shows that for $k=-1,0,1$,

$$\int_{E_i}\widetilde{R}^n(x;V^{n,j,k})dx=hV_i^{n,j,k}$$

and thus

$$\begin{aligned}h^2\overline{U}_{i,j}^{n+1}&=h^2\overline{U}_{i,j}^n+\sum_k w_k^j\left\{\int_{\widetilde{E}_i^n(y_k^j)}\widetilde{R}^n(x;V^{n,j,k})dx-\int_{E_i}\widetilde{R}^n(x;V^{n,j,k})dx\right\}\\&=h^2\overline{U}_{i,j}^n+\sum_k w_k^j\left\{\int_{\check{X}_{i-1/2}^n(y_k^j)}^{x_{i-1/2}}\widetilde{R}^n(x;V^{n,j,k})dx-\int_{\check{X}_{i+1/2}^n(y_k^j)}^{x_{i+1/2}}\widetilde{R}^n(x;V^{n,j,k})dx\right\}.\end{aligned} \tag{5.18}$$

The proof is complete. □

6. Some numerical results in one space dimension

All the examples herein use a periodic boundary condition. In all our one-dimensional results, m is the number of grid cells used. Corresponding to the estimate (3.3), we report errors measured in the discrete L_h^1 norm

$$\|u^n - \bar{u}^n\|_{1,h} := \sum_i \left| \frac{1}{h} \int_{E_i} u(x, t^n) dx - \bar{u}_i^n \right| h, \tag{6.1}$$

and, corresponding to (3.4), errors measured in the discrete L_h^∞ norm

$$\|u^n - q^n\|_{\infty,h} := \sup_i |u_i^n - q_i^n| \tag{6.2}$$

using the points $\xi_i = x_i$.

6.1. Example 1

We first test our scheme in the simple case of $a(x,t) = 1/3$, with initial condition $u_0(x) = 0.75 + 0.25 \sin(\pi x)$ over $[0,2]$. The exact solution is $u(x,t) = u_0(x - t/3)$. We present two tests. First, we use a time step $\Delta t = 40h$, which is about 13.3 times the CFL time limit, i.e., $\mathrm{CFL}_{\Delta t} = \max|a|\Delta t/\mathrm{h} = 13.3$. The final time is at $T = 10$. The L_h^1 and L_h^∞ errors as defined in (6.1) and (6.2) are reported in Table 6.1, where m is the number of grid cells. A fifth order convergence is observed, as expected from Theorem 4.1.

For the second test, we choose Δt so that grid points trace back to grid points, i.e., Δt is a multiple of $3h$. The L_h^1 error is within round off error for all m, and a superconvergence of the sixth order in the L_h^∞ error is obtained. This seems reasonable, since the error in (3.2) remains zero for all iterations if there is no initial error and the trace-back regions are exact for all times. The test with $\Delta t = 45h$ and the final time at $T = 9$, using long double precision, is given in Table 6.2.

6.2. Example 2

We test our scheme on a case with $a(x,t) = \sin(x)$ over $[0,2\pi]$, with exact solution

$$u(x,t) = \frac{\sin(2\arctan(e^{-t}\tan(x/2)))}{\sin(x)}.$$

Note that Type 4 subintervals occur in this example. We report the L_h^1 and L_h^∞ errors in Table 6.3. The initial time is at 0.1 and the final time is at $T = 1$. We use 10 steps, so $\Delta t = 0.09$. A fifth order convergence is observed.

In Table 6.4, we report the errors using only 5 time steps. In fact, we can use only one time step, and the results are given in Table 6.5. It can be seen that the errors are slightly reduced when the number of time steps is reduced, since less numerical diffusion builds up. However, the order of convergence remains basically the same fifth order, since it reflects the spatial convergence order.

In the above tests, we used a fixed point iteration to find the exact trace-back points. In the final test, we use an approximate Runge–Kutta solver for (2.1), and within each time step defined by (4.1), we use $m/10$ micro-time steps for the Runge–Kutta solver. The results are reported in Table 6.6. Comparing Tables 6.3 and 6.6, we see that the Runge–Kutta solver is in fact providing a good approximation to the trace-back points.

Table 6.1
Example 1. Error and convergence order at $T = 10$ with $\Delta t = 40h$.

m	L_h^1 error	Order	L_h^∞ error	Order
20	3.40206E−05	–	2.91819E−05	–
40	7.90078E−07	5.42827	7.35534E−07	5.31014
80	2.34044E−08	5.07714	2.39566E−08	4.94030
160	7.07622E−10	5.04766	7.57568E−10	4.98290
320	2.18519E−11	5.01715	2.21733E−11	5.09448
640	6.88475E−13	4.98821	6.08624E−13	5.18713

Table 6.2
Example 1. Error and convergence order at $T = 9$ with $\Delta t = 45h$.

m	L_h^1 error	L_h^∞ error	Order
20	3.28371E−33	3.20158E−06	–
40	1.10645E−32	7.22244E−08	5.47016
80	1.50295E−32	1.31427E−09	5.78015
160	2.36420E−32	2.19594E−11	5.90328
320	4.98864E−32	3.49684E−13	5.97264
640	8.96463E−32	5.35670E−15	6.02857

Table 6.3
Example 2. Error and convergence order at $T = 1$ with 10 steps.

m	L_h^1 error	Order	L_h^∞ error	Order
20	1.11043E−02	–	8.65266E−03	–
40	4.47664E−04	4.63256	5.66291E−04	3.93353
80	1.31403E−05	5.09035	1.81398E−05	4.96431
160	2.78878E−07	5.55822	5.36271E−07	5.08005
320	5.56687E−09	5.64663	1.25553E−08	5.41659
640	1.24537E−10	5.48221	3.25145E−10	5.27107

Table 6.4
Example 2. Errors and convergence order at $T = 1$ with 5 steps.

m	L_h^1 error	Order	L_h^∞ error	Order
20	8.74024E−03	–	6.89093E−03	–
40	2.32960E−04	5.22952	4.15458E−04	4.05192
80	7.34720E−06	4.98674	1.43913E−05	4.85143
160	1.55773E−07	5.55968	3.72114E−07	5.27331
320	3.42601E−09	5.50678	9.70153E−09	5.26139
640	1.09435E−10	4.96838	3.00589E−10	5.01235

Table 6.5
Example 2. Errors and convergence order at $T = 1$ with one step.

m	L_h^1 error	Order	L_h^∞ error	Order
20	2.81444E−03	–	3.01100E−03	–
40	1.98279E−04	3.82724	1.92974E−04	3.96377
80	6.04810E−06	5.03491	8.59603E−06	4.48859
160	1.29589E−07	5.54447	2.18847E−07	5.29568
320	2.33589E−09	5.79383	4.19884E−09	5.70378
640	3.92587E−11	5.89481	7.51634E−11	5.80382

Table 6.6
Example 2. Error and convergence order at $T = 1$ with 10 steps and $m/10$ micro-time steps for the trace-back Runge–Kutta solver.

m	L_h^1 error	Order	L_h^∞ error	Order
20	1.11044E−02	–	8.65267E−03	–
40	4.47667E−04	4.63256	5.66291E−04	3.93353
80	1.31403E−05	5.09036	1.81398E−05	4.96431
160	2.78878E−07	5.55822	5.36263E−07	5.08007
320	5.56624E−09	5.64679	1.25548E−08	5.41663
640	1.16465E−10	5.57873	3.25641E−10	5.26882

6.3. Example 3

In this example, we test a case with $a(x,t) = \sin(t)$ on $[0,2]$, for which the exact solution is $u(x,t) = u_0(x + 1 + \cos(t))$, where $u_0(x)$ is chosen as in Example 1. We use a fourth order Runge–Kutta method to approximate the trace-back points for system (2.1). We take the time step $\Delta t = h^{4/5}$, and a fifth order convergence is obtained at the final time $T = 4$, see Table 6.7.

We also used m micro-time steps in the Runge–Kutta method within each time step defined by (4.1). The results are reported in Table 6.8. Finally, since the exact trace-back point could be found analytically, we test it with 5 time steps and report the results in Table 6.9. We seem to obtain a sixth order superconvergence for both of these tests, but it is not clear why, since the grid points are not traced back to grid points.

6.4. Example 4

In the final example, we take a standard test case, called *Shu's linear test* [14]. In this case, $a(x,t) = 1$ and the initial data $u_0(x)$ is defined as

Table 6.7
Example 3. Errors and convergence order at $T = 4$ with $\Delta t = h^{4/5}$, using RK4 for the trace-back points.

m	L_h^1 error	Order	L_h^∞ error	Order
20	5.61267E−04	–	4.86282E−04	–
40	1.94068E−05	4.85405	1.87017E−05	4.70055
80	6.51947E−07	4.89567	6.49074E−07	4.84865
160	2.14898E−08	4.92303	2.09586E−08	4.95277
320	6.98781E−10	4.94266	6.64485E−10	4.97916
640	2.24770E−11	4.95832	2.03981E−11	5.02573

Table 6.8
Example 3. Errors and convergence order at $T = 4$, with 5 steps, m micro-time RK4 steps.

m	L_h^1 error	Order	L_h^∞ error	Order
20	3.96103E−05	–	3.30529E−05	–
40	4.95198E−07	6.32172	4.54642E−07	6.18390
80	8.20460E−09	5.91543	7.91174E−09	5.84459
160	1.33948E−10	5.93669	1.22479E−10	6.01339
320	2.18257E−12	5.93949	1.83076E−12	6.06395
640	1.45575E−13	3.90620	1.69054E−12	0.11496

Table 6.9
Example 3. Errors and convergence order at $T = 4$, with 5 steps and exact trace-back points.

m	L_h^1 error	Order	L_h^∞ error	Order
20	3.96105E−05	–	3.30536E−05	–
40	4.95196E−07	6.32174	4.54625E−07	6.18399
80	8.20452E−09	5.91544	7.91116E−09	5.84464
160	1.33942E−10	5.93674	1.22519E−10	6.01281
320	2.19103E−12	5.93385	1.81899E−12	6.07372
640	1.40254E−13	3.96549	2.10054E−13	3.11430

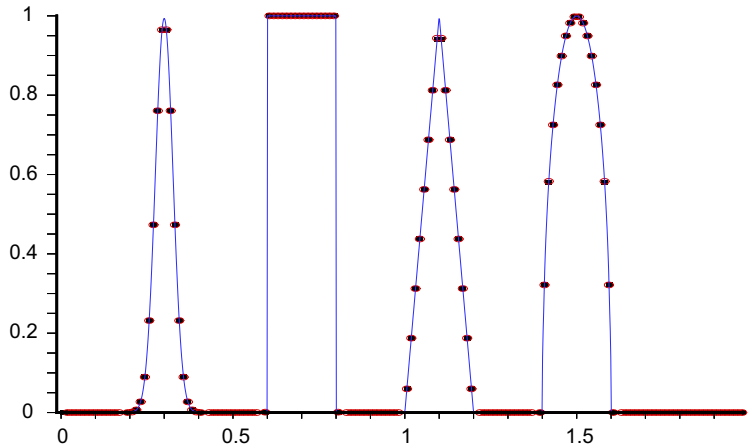
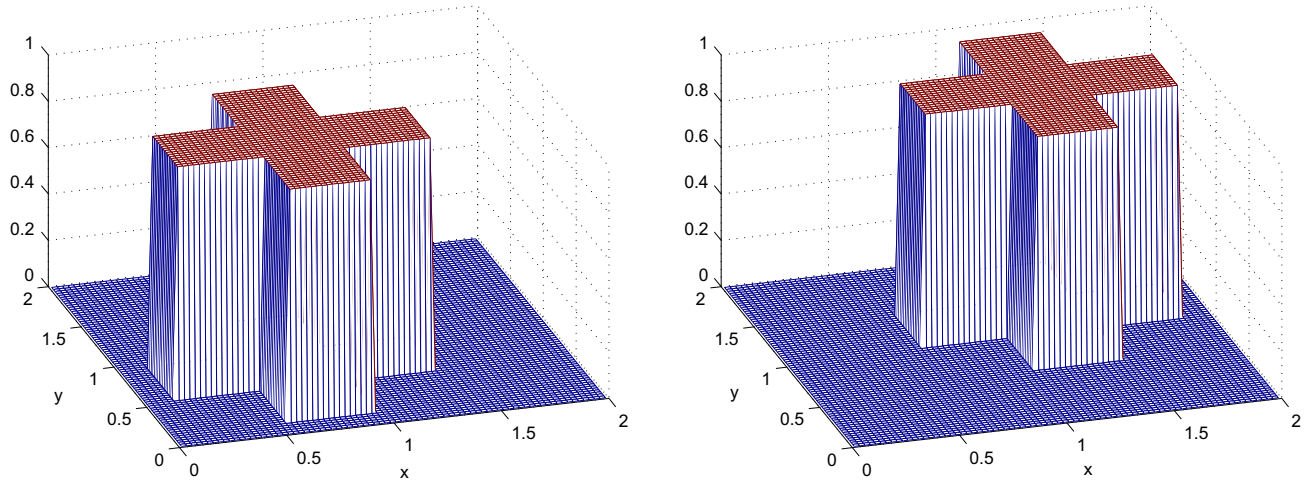


Fig. 6.1. Example 4. Shu's linear test with $m = 160$. The blue curve shows the exact solution, while the solid black squares are the exact solution at the center of each cell. The red circles are the computed solution plotted at the center of each cell. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$u_0(x) = \begin{cases} \frac{1}{6}(G(x,\beta,z-\delta) + G(x,\beta,z+\delta) + 4G(x,\beta,z)), & 0.2 \leqslant x \leqslant 0.4, \\ 1, & 0.6 \leqslant x \leqslant 0.8, \\ 1 - |10(x-1.1)|, & 1 \leqslant x \leqslant 1.2, \\ \frac{1}{6}(F(x,\alpha,a-\delta) + F(x,\alpha,a+\delta) + 4F(x,\alpha,a)), & 1.4 \leqslant x \leqslant 1.6, \\ 0, & \text{otherwise}, \end{cases}$$

Table 7.1
Example 5 with true solution $u(x,y,t) = \sin(\pi x + \pi y - 2\pi t)$. Errors and convergence order at $t = 20$, for $\Delta t = 2.5h$ ($\mathrm{CFL}_{\Delta t} = 2.5$).

m	L_h^1 error	Order	L_h^∞ error	Order
20	1.28182E−02	–	4.99453E−03	–
40	4.16771E−04	4.94280	1.85201E−04	4.75319
80	1.30725E−05	4.99465	6.15524E−06	4.91113
160	4.08569E−07	4.99981	1.91862E−07	5.00367
320	1.27688E−08	4.99989	5.88077E−09	5.02792



(a) Initial condition. (b) $t = 0.5$ with $\Delta t = 10h$.

Fig. 7.1. Example 5 with an initial cross pattern. Plot of cell average values at $t = 0$ and $t = 0.5$, using $\Delta t = 10h$. The time step allows perfect reproduction of the cross pattern on the grid.

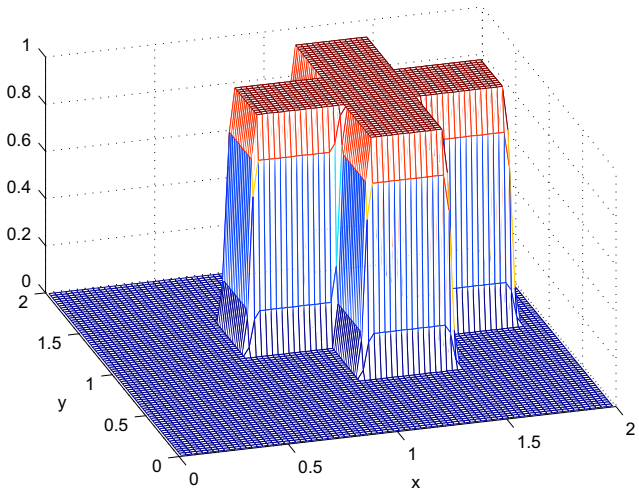


Fig. 7.2. Example 5 with an initial cross pattern. Plot of cell average values at $t = 0.5$ using $\Delta t = 10.5h$. The time step does not allows reproduction of the cross pattern on the grid.

where

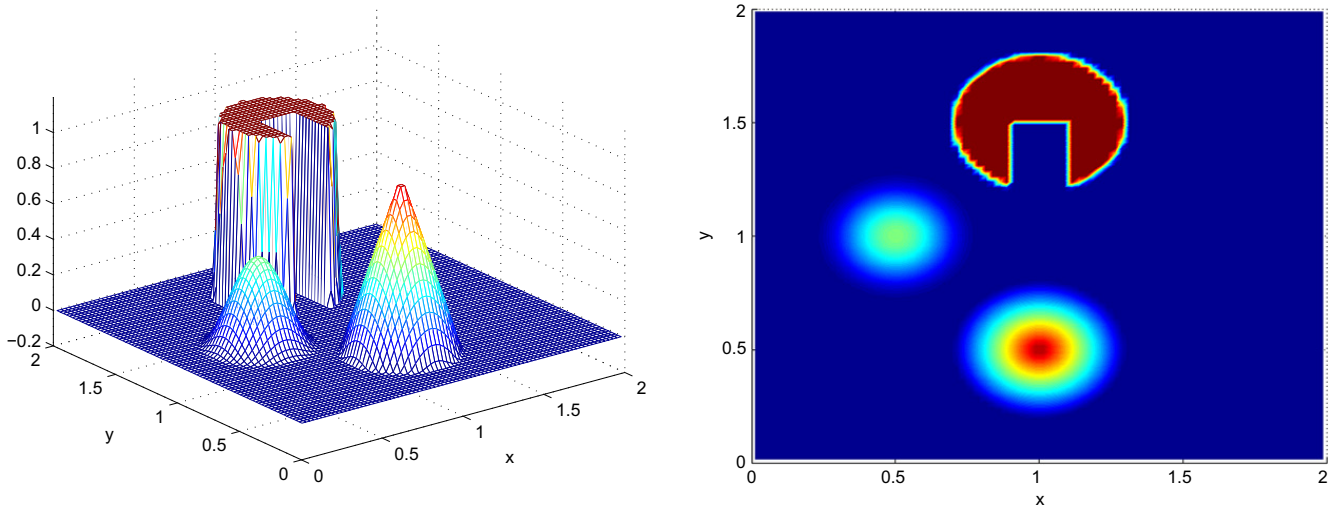
$$G(x,\beta,z) = e^{-\beta(x-z)^2} \quad \text{and} \quad F(x,\alpha,a) = \sqrt{\max(1-\alpha^2(x-a)^2, 0)}.$$

The constants are set to $a = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \log 2/(36\delta^2)$.

We compute the solution up to time $T = 2$ with $m = 160$ grid cells. The resulting solution is shown in Fig. 6.1. The blue curve stands for the exact solution, the solid black squares are the exact solution at the center of each cell. The red circles

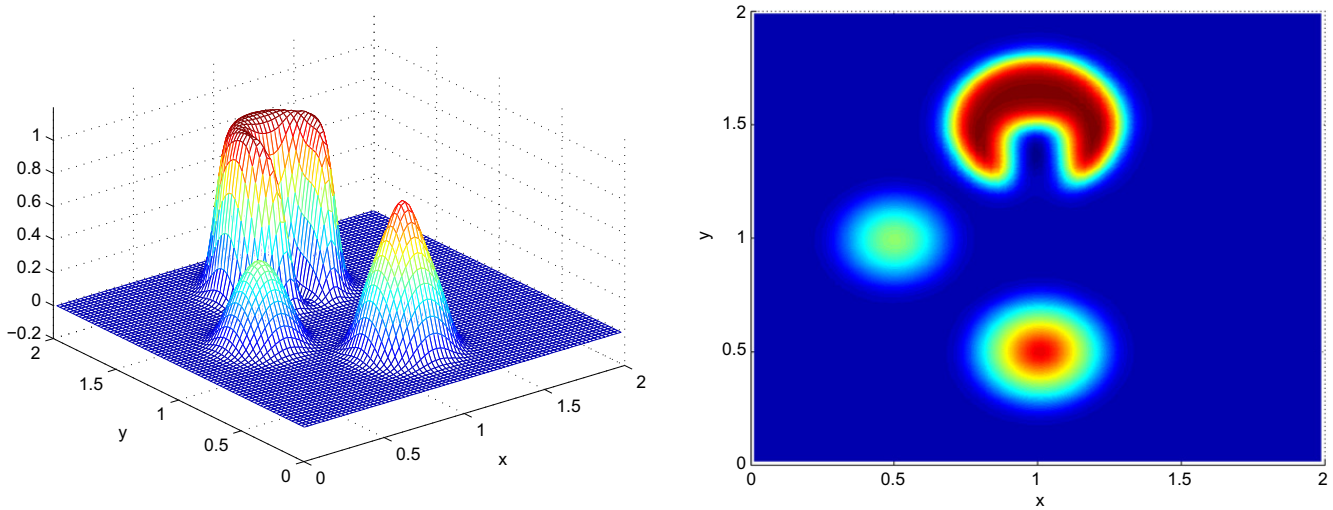
Table 7.2
Example 6. Errors, convergence order, and mass error at $t = 2\pi$, using $\Delta t = 2h$ ($CFL_{\Delta t} = 2$).

m	L_h^1 error	Order	L_h^∞ error	Order	Mass error
20	9.17146E−04	–	1.00826E−03	–	2.80779E−15
40	9.79220E−05	3.22745	1.48453E−04	2.76379	1.28105E−14
80	4.83232E−06	4.34084	9.68675E−06	3.93785	4.61530E−14
160	1.23696E−07	5.28785	2.87469E−07	5.07454	1.05292E−13
320	3.85239E−09	5.00490	5.89530E−09	5.60769	4.89608E−14



(a) Cell average values. (b) Contour plot.

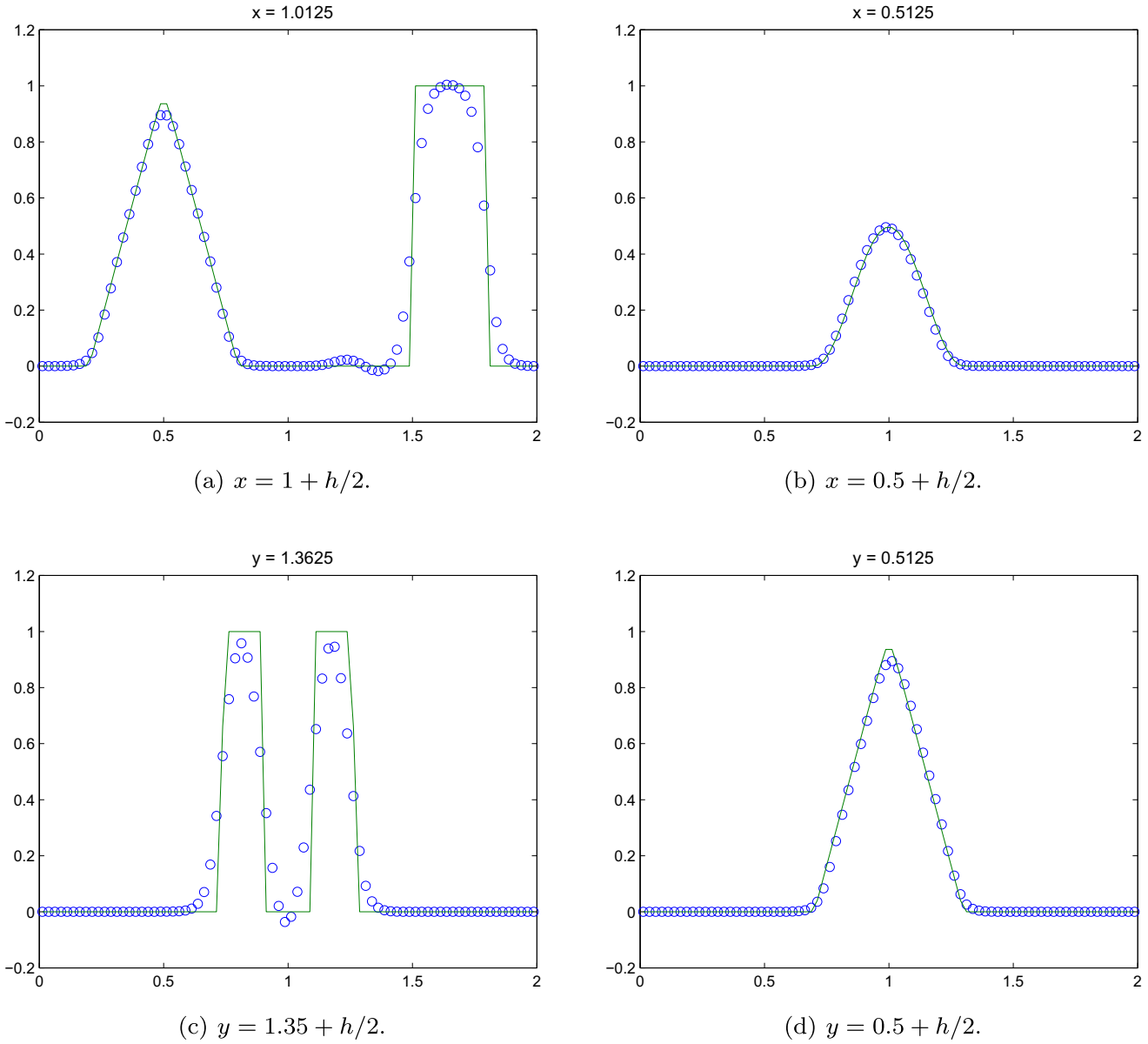
Fig. 7.3. Example 7. Initial condition, shown using (a) cell average values and (b) a graphically smoothed contour plot.



(a) Cell average values. (b) Contour plot.

Fig. 7.4. Example 7. Solution at $t = 12\pi$, using $\Delta t = 4h$ ($CFL_{\Delta t} = 4$), shown using (a) cell average values and (b) a graphically smoothed contour plot.

are the computed solution at the center of each cell. It is easy to see that the numerical solution maintains the same resolution as the initial condition, since all the trace-back points are exact and at grid points. The results shown use $\Delta t = 2/10$, i.e., 10 steps, but in fact, we could use one step, $\Delta t = 2$. No numerical diffusion builds up. We could run the test with a larger final time, and the results would remain the same as long as the final time was a multiple of 2.



(a) $x = 1 + h/2$. (b) $x = 0.5 + h/2$.

(c) $y = 1.35 + h/2$. (d) $y = 0.5 + h/2$.

Fig. 7.5. Example 7. Cross-sections of the numerical solution at (a) $x = 1 + h/2$, (b) $x = 0.5 + h/2$, (c) $y = 1.35 + h/2$, and (d) $y = 0.5 + h/2$. The true solution is also shown as a solid line.

7. Some numerical results in two space dimensions

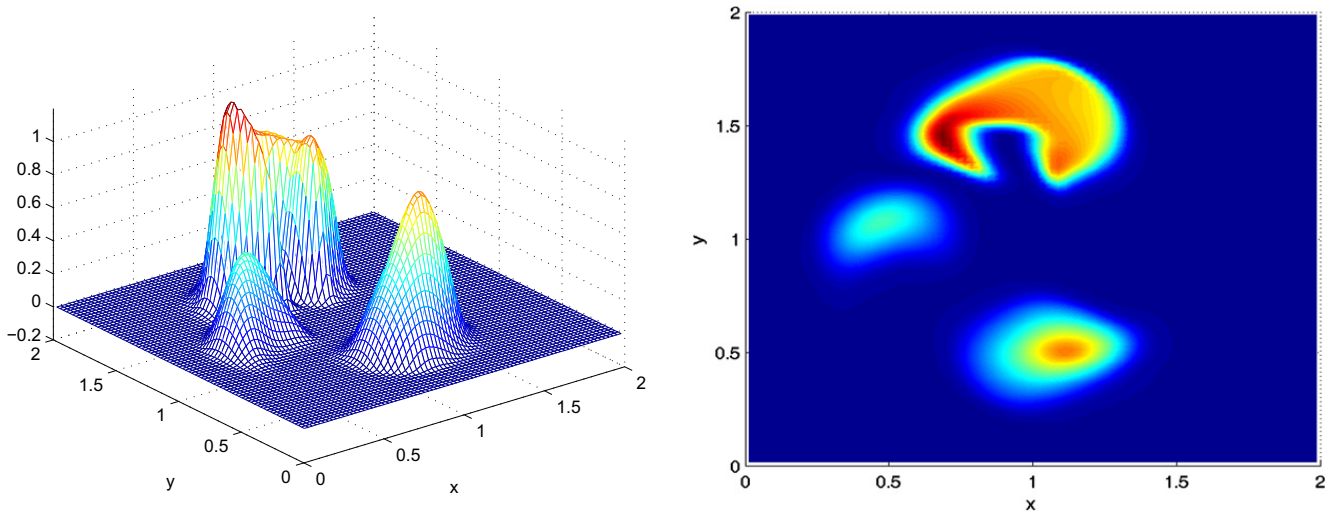
All the examples herein use a periodic boundary condition; however, mass is restricted to the interior of the domain. Most of the examples are taken from LeVeque [15] and Qiu and Shu [22]. For our two-dimensional results, m is the number of grid cells used in each direction. We report errors measured in the discrete L_h^1 norm

$$\|u^n - \bar{u}^n\|_{1,h} := \sum_{i,j} \left| \frac{1}{h^2} \int_{F_j} \int_{E_i} u(x, t^n) dx dy - \bar{u}_{i,j}^n \right| h^2 \tag{7.1}$$

and a discrete L_h^∞ norm. In all cases, the global mass balance error is negligible, as reported in Table 7.2, so we do not otherwise comment on it.

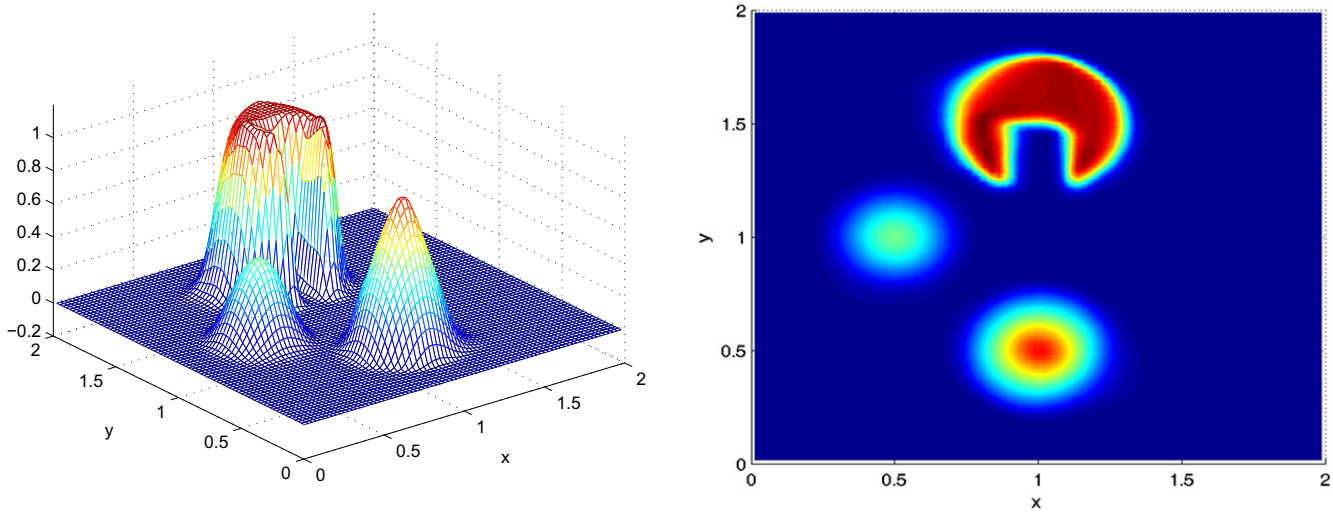
Our L_h^∞ norm is based on a pointwise postprocessing of the two-dimensional discrete solution. We chose to target the cell centers (x_i, y_j). As in the y-sweep described in Section 5.2, we first perform high order WENO reconstruction of the solution in the y-direction targeting the point y_j for constant x_i. This results in a high order approximation of the average mass in the interval $E_i \times \{y_j\}$, as noted in (5.17). Then in the x-direction, for each y_j, we apply the optional postprocessing described in Section 3.3. The result is $q_{i,j}^n$, a high order (i.e., fifth order in our case) pointwise approximation of $u_{i,j}^n$. Thus we define

$$\|u^n - q^n\|_{\infty,h} := \sup_{i,j} \left| u_{i,j}^n - q_{i,j}^n \right|. \tag{7.2}$$



(a) Cell average values. (b) Contour plot.

Fig. 7.6. Example 8. First order Strang splitting solution at $T = 1.5$ using $\Delta t = 2h$ ($\mathrm{CFL}_{\Delta t} = 4$).



(a) Cell average values. (b) Contour plot.

Fig. 7.7. Example 8. Second order Strang splitting solution at $T = 1.5$ using $\Delta t = 4h$ ($\mathrm{CFL}_{\Delta t} = 8$).

Of course, there is no reason to begin with the y-direction. We could also reconstruct in the x-direction first and then reconstruct in the y-direction. In fact there is very little difference between these two possibilities, and we report the worse of the two.

All two-dimensional figures in this section show the results of using an 80×80 grid. In our Strang-split, two-dimensional simulations, the CFL limited time step would be

$$\Delta t_{\mathrm{CFL}} = \frac{h}{\max\left(\max_{x,y}|a_1(x,y)|, \max_{x,y}|a_2(x,y)|\right)}$$

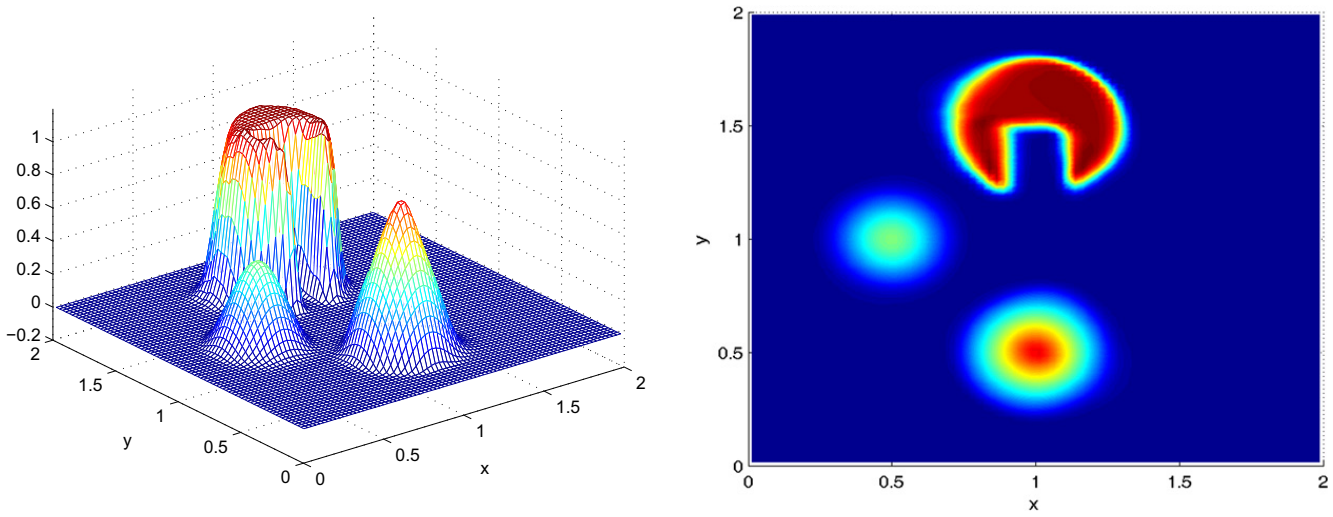
and so, for a given Δt, the CFL number is

$$\mathrm{CFL}_{\Delta t} = \max\left(\max_{x,y}|a_1(x,y)|, \max_{x,y}|a_2(x,y)|\right)\frac{\Delta t}{h}.$$

We use a first order Strang splitting unless otherwise noted.

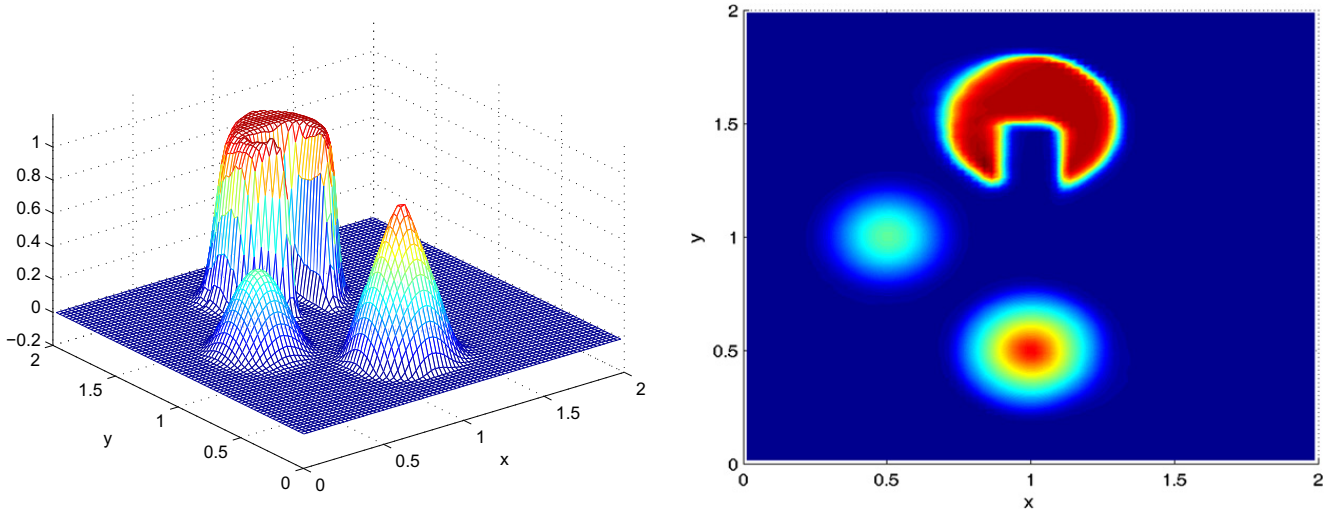
7.1. Example 5

This example is a two dimensional linear transport. The governing equation is



(a) Cell average values. (b) Contour plot.

Fig. 7.8. Example 8. Second order Strang splitting solution at $T = 1.5$ using $\Delta t = 8h$ ($\mathrm{CFL}_{\Delta t} = 16$).



(a) Cell average values. (b) Contour plot.

Fig. 7.9. Example 8. Second order Strang splitting solution at $T = 1.5$ using $\Delta t = 12h$ ($\mathrm{CFL}_{\Delta t} = 24$).

$$u_t + u_x + u_y = 0, \quad x \in (0,2), \quad y \in (0,2), \ t > 0. \tag{7.3}$$

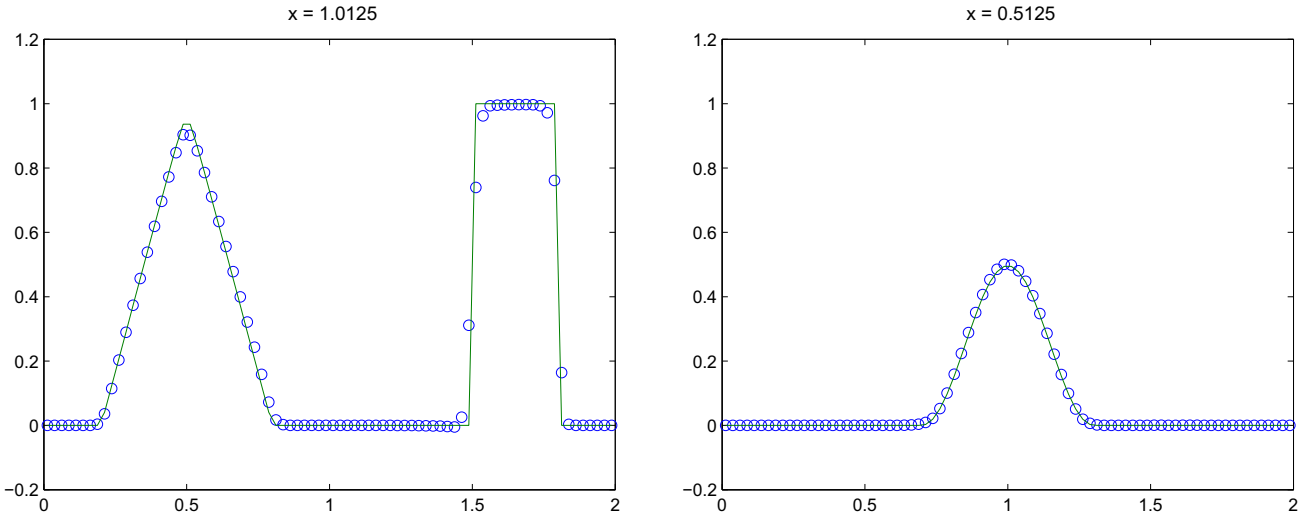
The equation is split into two one-dimensional equations, each of which is evolved by the proposed EL–WENO finite volume method. Note that there is no dimensional splitting error in time and the spatial error is the dominant error. Moreover, In this case $a_1 \equiv a_2 \equiv 1$, so $\Delta t_{\mathrm{CFL}} = h$.

Table 7.1 gives the L_h^1 and L_h^∞ errors and the corresponding orders of convergence for the smooth solution $u(x,y,t) = \sin(\pi x + \pi y - 2\pi t)$, using $\Delta t = 2.5h$. Very clean fifth order convergence is observed for our fifth order reconstruction, as expected from the formal convergence theory Theorem 5.2.

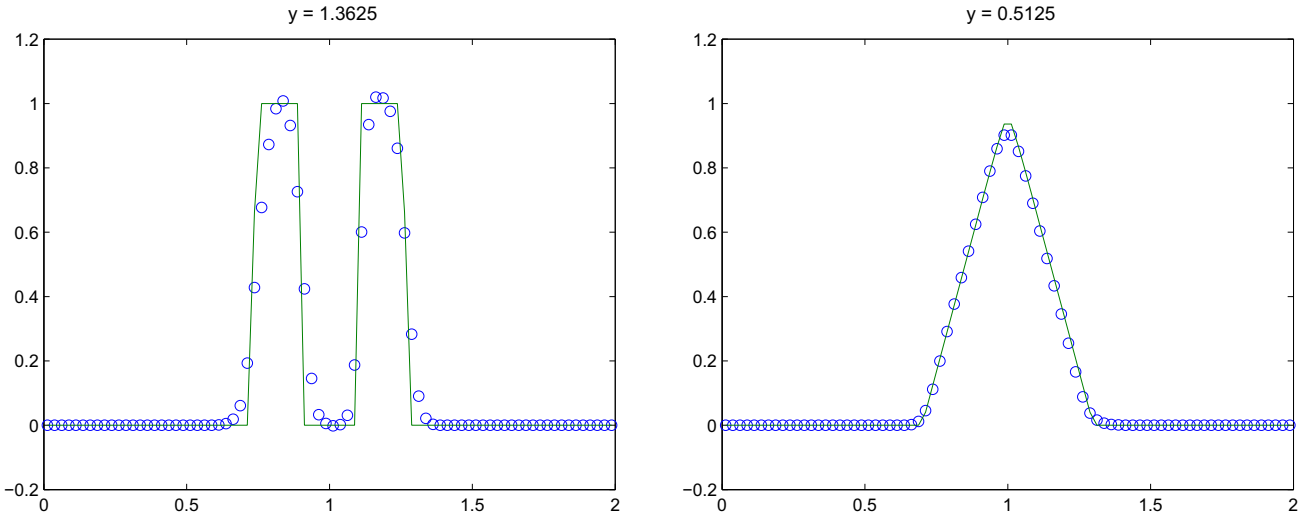
We next advect a cross pattern, as shown in Figs. 7.1 and 7.2. It is clear that we have less numerical diffusion error when Δt is chosen so that grid points are traced back to grid points, as in Fig. 7.1(b), where $\Delta t = 10h$, since then the cross pattern can be represented exactly on the grid, as opposed to Fig. 7.2, where $\Delta t = 10.5h$ and we are one-half grid cell off each time step.

7.2. Example 6

The next example is a two dimensional rigid body rotation. The governing equation is



(a) $x = 1 + h/2$. (b) $x = 0.5 + h/2$.



(c) $y = 1.35 + h/2$. (d) $y = 0.5 + h/2$.

Fig. 7.10. Example 8. Cross-sections of the numerical solution at (a) $x = 1 + h/2$, (b) $x = 0.5 + h/2$, (c) $y = 1.35 + h/2$, and (d) $y = 0.5 + h/2$. The true solution is also shown as a solid line.

$$u_t - ((y-1)u)_x + ((x-1)u)_y = 0, \quad x \in [0,2], \quad y \in [0,2], \ t > 0. \tag{7.4}$$

The initial condition is a smooth, radial bump function, defined as

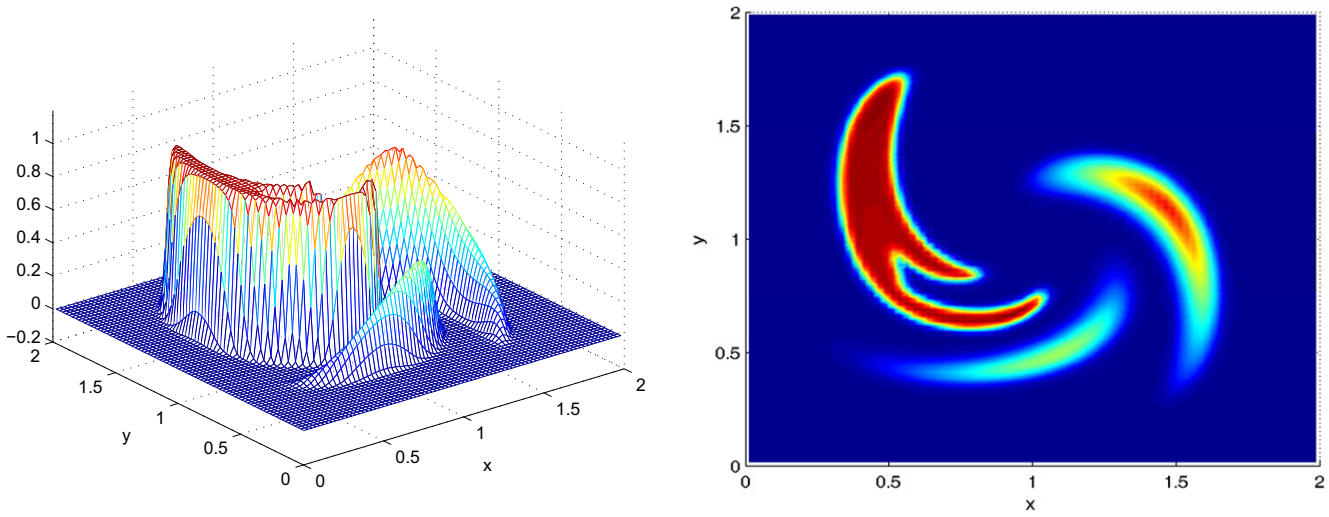
$$u(x,y,0) = \frac{2}{5}[\psi(1 + r(x,y))\psi(1 - r(x,y)) + 1],$$
$$r(x,y) = \sqrt{(x-1)^2 + (y-1)^2},$$

where $\psi(s) = e^{-1/s^2}$ for $s > 0$ and $\psi(s) = 0$ otherwise.

Table 7.2 gives the L_h^1 and L_h^∞ errors, the corresponding orders of convergence, and the global mass balance error, where $\Delta t = 2h$ ($\mathrm{CFL}_{\Delta t} = 2$). Very clean fifth order convergence is observed, as expected, and the mass balance error is nonzero due only to rounding error. For this example, we used a second order in time Strang splitting.

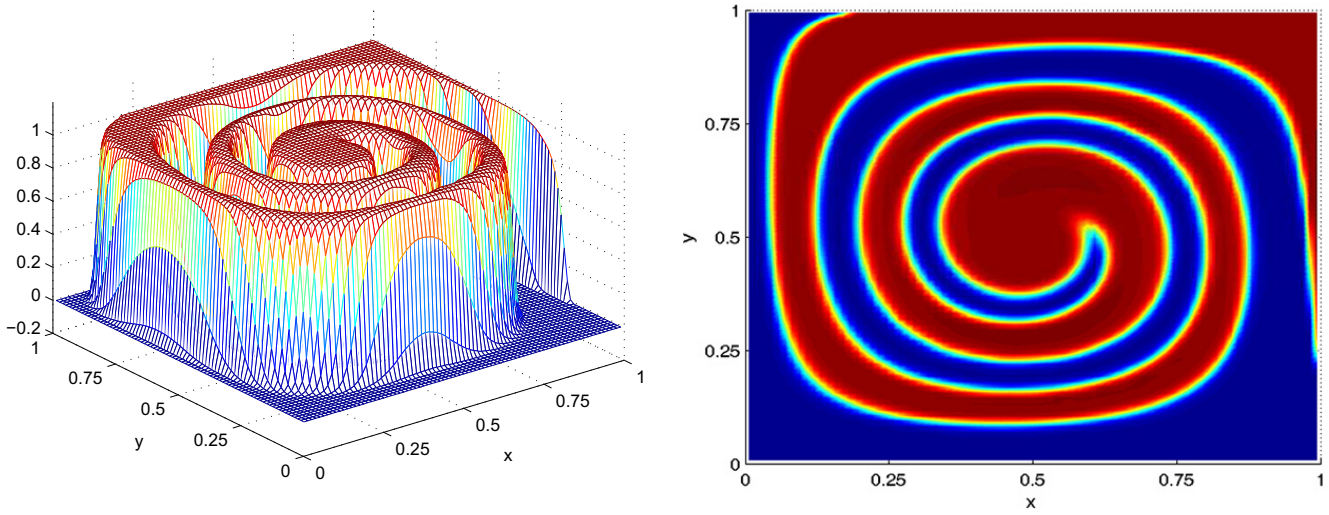
7.3. Example 7

This next example is again a two-dimensional rigid body rotation as in (7.4) of Example 6. However, the initial condition used now includes a slotted disk, a cone, and a "smooth" hump, similar to that used by LeVeque [15]. The initial condition is shown in Fig. 7.3.



(a) Cell average values. (b) Contour plot.

Fig. 7.11. Example 8. Second order Strang splitting solution at time $t = 0.75 = T/2$ using $\Delta t = 8h$ ($\mathrm{CFL}_{\Delta t} = 16$).



(a) Cell average values. (b) Contour plot.

Fig. 7.12. Example 9. Solution at time $t = 2.5$ using $\Delta t = 8h$ ($\mathrm{CFL}_{\Delta t} = 8$).

The numerical solution after six full revolutions of the scheme using $\Delta t = 4h$ ($\mathrm{CFL}_{\Delta t} = 4$) are shown in Fig. 7.4. One-dimensional cross-sections showing the solution are given in Fig. 7.5, benchmarked with the exact solution. Some oscillation in the solution is observed. However, it is not due to the difference of integrals needed for the handling of Type 4 subintervals, since none arose during the computation. We also computed the solution using time steps that were non-integral multiples of h, e.g., $4.5h$, but the results show no significant difference to those shown here.

7.4. Example 8

A more severe test is obtained by using a swirling deformation flow. Following [15], we take the velocity in the form

$$a_1(x,y) = \sin^2\left(\frac{\pi x}{2}\right)\sin(\pi y)g(t), \quad a_2(x,y) = -\sin^2\left(\frac{\pi y}{2}\right)\sin(\pi x)g(t). \tag{7.5}$$

This flow satisfies $a_1 = a_2 = 0$ on the boundaries of our domain $(0,2)\times(0,2)$. The function $g(t)$ is used to introduce time dependence in the flow field, and we use

$$g(t) = 2\cos(\pi t/T)$$

on the time interval $0 \leqslant t \leqslant T$. The flow slows down and reverses direction in such a way that the initial data is recovered at time T. We use $T = 1.5$, and the same initial condition as in the previous example, Example 7.

We remark that in [15], Leveque worked over the unit square domain. Compared to his work, we have rescaled both the velocity field and $g(t)$. Moreover, Qiu and Shu [22] used a similar scaling over the domain $(-\pi, \pi)^2$.

The results using a first order Strang splitting are shown in Fig. 7.6. They are not very good, because the splitting error is large.

The second order Strang splitting improves the result significantly, as shown in Figs. 7.7–7.9 for three different time steps, $\Delta t = 4$ h, 8 h, and 12 h. The results are remarkably good, especially for the largest time step, which has a CFL number of $\mathrm{CFL}_{\Delta t} = 24$ and therefore is the least numerically diffusive. We also report cross-sectional slices for the $\Delta t = 8h$ case in Fig. 7.10, benchmarked with exact solution. Again some oscillation is observed. In this example, Type 4 subintervals do arise.

At time $T/2$ the initial data is quite deformed. Fig. 7.11 shows the solution at this time using $\Delta t = 8h$. The solution is very similar to that given in [15,22].

7.5. Example 9

As a final example, we use the swirling flow (7.5) of the previous example, but we scale it to the unit square and take $g(t) \equiv 1$. The initial condition is

$$u(x, y, 0) = \begin{cases} 1 & \text{if } (x-1)^2 + (y-1)^2 < 0.8^2, \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 7.12 shows the computed solution at time $t = 2.5$ using $\Delta t = 8h$ ($\mathrm{CFL}_{\Delta t} = 8$). The level of numerical diffusion on this 80×80 grid is extremely low, since it takes only 25 steps to reach the final time.

8. Conclusions

We defined a locally conservative Eulerian–Lagrangian finite volume method with a WENO reconstruction (EL–WENO). It is a generalization of the Finite Difference Locally Conservative Eulerian–Lagrangian Method [9] and the Characteristics-Mixed Method [1,2,4]. The fifth order version was worked out in detail. For each time step, the grid points are first traced backward in time, perhaps using a Runge–Kutta solver. For each grid cell, these trace-back points define the trace-back set. The mass in the trace-back set advects forward to the new time level over the time step. It is therefore only necessary to integrate accurately the mass over the trace-back set.

An integration-based, piecewise-polynomial WENO reconstruction was developed to obtain a fifth order representation of the fluid mass at the previous time level, although other order methods could presumably be developed. To define the reconstruction, the trace-back set must be decomposed into the fixed Eulerian grid. The definition of the linear weights depends on this decomposition and varies locally. Rules for computing the integrated mass were developed that are fifth order accurate. Moreover, the scheme was shown to be computationally efficient and locally mass conservative.

Numerical results showed that the optimal fifth order accuracy is obtained for both constant and variable coefficient cases. Large time steps, greatly exceeding the CFL limit, could be taken by our scheme; in fact, we could take a fixed number of time steps regardless of the number of cells in the computation, as long as the trace-back points are approximated accurately enough.

Strang splitting was used to handle multidimensional problems. Even though the scheme is based on finite volumes, we were able to show fifth order spatial accuracy as well as local mass conservation. Numerical results bore these facts out.

Our scheme inherits the high-order accuracy and nonoscillatory property from WENO schemes, and the CFL time step limit free property and small time truncation error from Eulerian–Lagrangian methods.

References

[1] T. Arbogast, A. Chilakapati, M.F. Wheeler, A characteristic-mixed method for contaminant transport and miscible displacement, in: T.F. Russell et al. (Eds.), Computational Methods in Water Resources IX, Numerical Methods in Water Resources, vol. 1, Computational Mechanics Publications, Southampton, UK, 1992, pp. 77–84.
[2] T. Arbogast, C. Huang, A fully mass and volume conserving implementation of a characteristic method for transport problems, SIAM J. Sci. Comput. 28 (2006) 2001–2022.
[3] T. Arbogast, W.-H. Wang, Stability, monotonicity, maximum and minimum principles, and implementation of the volume corrected characteristic method, SIAM J. Sci. Comput. 33 (2011) 1549–1573.
[4] T. Arbogast, M.F. Wheeler, A characteristics-mixed finite element method for advection dominated transport problems, SIAM J. Numer. Anal. 32 (1995) 404–424.
[5] J. Carrillo, F. Vecil, Nonoscillatory interpolation methods applied to Vlasov-based models, SIAM J. Sci. Comput. 29 (2007) 1179–1206.
[6] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, An Eulerian–Lagrangian localized adjoint method for the advection–diffusion equation, Adv. Water Res. 13 (1990) 187–206.
[7] C. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys. 22 (1976) 330–351.
[8] H.K. Dahle, R.E. Ewing, T.F. Russell, Eulerian–Lagrangian localized adjoint methods for a nonlinear advection–diffusion equation, Comput. Methods Appl. Mech. Eng. 122 (1995) 223–250.
[9] J. Douglas Jr., C.-S. Huang, The convergence of a locally conservative Eulerian–Lagrangian finite difference method for a semilinear parabolic equation, BIT 41 (2001) 480–489.

[10] J. Douglas Jr., F. Pereira, L.-M. Yeh, A locally conservative Eulerian–Lagrangian numerical method and its application to nonlinear transport in porous media, Comput. Geosci. 4 (2000) 1–40.
[11] J. Douglas Jr., T.F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 19 (1982) 871–885.
[12] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes III, J. Comput. Phys. 71 (1987) 231–303.
[13] A. Harten, S. Osher, Uniformly high-order accurate nonoscillatory schemes I, SIAM J. Numer. Anal. 24 (1987) 279–309.
[14] J.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[15] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627–665.
[16] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, Math. Model. Numer. Anal. 33 (1999) 547–571.
[17] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[18] Y.-Y. Liu, C.-W. Shu, M. Zhang, On the positivity of linear weights in WENO approximations, Acta Math. Appl. Sin. 25 (2009) 503–538.
[19] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier–Stokes equations, Numer. Math. 38 (1981/82) 309–332.
[20] J. Qiu, C.-W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, J. Comput. Phys. 183 (2002) 187–209.
[21] J.-M. Qiu, A. Christlieb, A conservative high order semi-Lagrangian WENO method for the Vlasov equation, J. Comput. Phys. 229 (2010) 1130–1149.
[22] 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.
[23] J.-M. Qiu, C.-W. Shu, Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation, Commun. Comput. Phys. 10 (2011) 979–1000.
[24] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (2002) 108–127.
[25] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[26] H. Wang, M. Al-Lawatia, A locally conservative Eulerian–Lagrangian control-volume method for transient advection–diffusion equations, Numer. Methods Partial Differ. Equat. 22 (2006) 577–599.
[27] H. Wang, D. Liang, R.E. Ewing, S.L. Lyons, G. Qin, An ELLAM approximation for highly compressible multicomponent flows in porous media. Locally conservative numerical methods for flow in porous media, Comput. Geosci. 6 (2002) 227–251.