SIAM J. SCI. COMPUT.
Vol. 22, No. 4, pp. 1461–1488

A THIRD-ORDER SEMIDISCRETE CENTRAL SCHEME FOR CONSERVATION LAWS AND CONVECTION-DIFFUSION EQUATIONS*

ALEXANDER KURGANOV† AND DORON LEVY‡

Abstract. We present a new third-order, semidiscrete, central method for approximating solutions to multidimensional systems of hyperbolic conservation laws, convection-diffusion equations, and related problems. Our method is a high-order extension of the recently proposed second-order, semidiscrete method in [A. Kurgonov and E. Tadmor, *J. Comput Phys.*, 160 (2000) pp. 241–282].

The method is derived independently of the specific piecewise polynomial reconstruction which is based on the previously computed cell-averages. We demonstrate our results by focusing on the new third-order central weighted essentially nonoscillatory (CWENO) reconstruction presented in [D. Levy, G. Puppo, and G. Russo, *SIAM J. Sci. Comput.*, 21 (1999), pp. 294–322]. The numerical results we present show the desired accuracy, high resolution, and robustness of our method.

Key words. hyperbolic systems, convection-diffusion equations, central difference schemes, high-order accuracy, nonoscillatory schemes, weighted essentially nonoscillatory reconstruction.

AMS subject classifications. Primary 65M10; Secondary 65M05

PII. S1064827599360236

1. Introduction.

Numerical methods for approximating solutions of hyperbolic conservation laws,

$$\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}f(u(x,t))=0, \tag{1.1}$$

and of the related convection-diffusion equations,

$$\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}f(u(x,t))=\frac{\partial}{\partial x}Q[u(x,t),u_x(x,t)], \tag{1.2}$$

have attracted much attention in recent years (see, e.g., [6, 35] and the references therein). Here, $u(x,t)$ is a conserved quantity, $f(u)$ is a nonlinear convection flux, and $Q(u,u_x)$ is a dissipation flux satisfying the (weak) parabolicity condition, $\frac{\partial}{\partial s}Q(u,s)\geq 0\ \forall u,s$. In the most general case $u=(u_1,\ldots,u_n)$ is an n-vector in a $d+1$-dimensional space, $(x,t)=(x_1,\ldots,x_d,t)$, and f and Q are vector-functions.

In this paper, we focus on the class of *central schemes*, all of which can be viewed as an extension of the well-known Lax–Friedrichs (LxF) scheme [5]. The first-order LxF method enjoys the major advantage of simplicity over the upwind schemes (e.g., the Godunov scheme [7]): no (approximate) Riemann solvers or characteristic decompositions are involved in its construction, and therefore, its realization for complicated

*Received by the editors June 15, 1999; accepted for publication (in revised form) November 15, 1999; published electronically November 2, 2000.

http://www.siam.org/journals/sisc/22-4/36023.html

†Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 (kurganov@math.lsa.umich.edu). The work of this author was supported in part by an NSF Group Infrastructure grant. Part of this work was done while the author was visiting the Lawrence Berkeley National Lab.

‡Department of Mathematics, University of California, Berkeley, CA 94720, and Lawrence Berkeley National Lab, Berkeley, CA 94720 (dlevy@math.berkeley.edu). The work of this author was supported in part by the Applied Mathematical Sciences subprogram of the Office of Science, U.S. Department of Energy, under contract DE–AC03–76–SF00098.

multidimensional systems is rather simple. At the same time, the LxF scheme suffers from excessive numerical dissipation, which causes a poor (smeared) resolution of discontinuities and rarefaction waves.

A second-order, nonoscillatory central scheme was first introduced by Nessyahu and Tadmor in [30]. Since then, the Nessyahu–Tadmor (NT) scheme was further extended to higher orders of accuracy [28] (also see [3, 19]), as well as to the multidimensional systems (1.1), in [1] and [12] (also see [18, 20, 21, 22]).

The main ingredient in the construction of the NT method is a second-order, nonoscillatory, monotonic upstream schemes for conservation laws (MUSCL)-type [17], piecewise linear interpolant (instead of the piecewise constant one employed in the LxF scheme) in combination with the exact solver for the time evolution. This approach allows us to significantly improve the resolution of nonsmooth solutions to hyperbolic conservation laws, (1.1), while retaining the main advantage of the LxF scheme—*simplicity*.

Unfortunately, applying the fully discrete NT scheme (or its higher-order extensions) to the second-order convection-diffusion equations, (1.2), does not provide the desired resolution of discontinuities (see, e.g., [14, 15, 16]). This loss of resolution occurs due to the accumulation of excessive numerical dissipation, which is typical of fully discrete central schemes with small time-steps $\Delta t \sim (\Delta x)^2$ (see [16] for details).

To circumvent this difficulty, a second-order *semidiscrete* central scheme was introduced by Kurganov and Tadmor in [16]. This scheme has smaller dissipation than the NT scheme and, unlike the fully discrete central schemes, it can be efficiently used with time-steps as small as required by the CFL stability restriction due to the diffusion term.

The basic idea in the construction of the second-order semidiscrete scheme was to use more accurate information about the *local* speed of propagation of the discontinuities. One was then able to derive a nonstaggered semidiscrete central method, by first integrating over nonequally spaced control volumes, out of which a new piecewise linear interpolant was reconstructed and finally projected on its cell-averages (without evolving in time). The final step was first introduced in [10], in a somewhat different context of transforming staggered methods into nonstaggered methods.

In this paper we extend the results of [16] by introducing a new *third-order, semidiscrete, central scheme*. Our new scheme is derived in a general form which is independent of the reconstruction step, as long as the reconstructed interpolant is sufficiently accurate and nonoscillatory. In particular, we use the new third-order central weighted essentially nonoscillatory (CWENO) reconstruction proposed in [21]. This reconstruction provides a third-order accurate interpolant which is built from the given cell-averages such that it is nonoscillatory in the essentially nonoscillatory (ENO) sense (see [9, 32]). This interpolant is written as a convex combination of two one-sided linear functions and one centered parabola. In smooth regions this convex combination guarantees the desired third-order accuracy. It automatically switches to a second-order, one-sided, linear reconstruction in the presence of large gradients. Such weighted essentially non-oscillatory (WENO) reconstructions were first introduced in the upwind framework, [11, 27], after which they were extended to the central framework [19, 20, 21, 22]. A different approach to creating a high-order (ENO-type) method which does not require Riemann solvers was suggested in [26].

This paper is organized as follows. We start in section 2 with a brief overview of central schemes for conservation laws. In particular we focus in section 2.1 on the CWENO reconstruction which we use as the building block for our third-order

method below.

We then proceed to construct our new third-order scheme. First, we deal with the fully discrete, one-dimensional (1D) setup in section 3. This new fully discrete scheme is sketched in (3.7). We give only the required details that are necessary to fulfill our final goal, namely, to derive the semidiscrete scheme.

With the fully discrete scheme, (3.7), we are ready to approach the semidiscrete limit in section 4.1. Our new third-order, one-dimensional, semidiscrete scheme is then summarized in (4.5). This scheme is written in a general form which is independent of the reconstruction step and can also be combined with any appropriate ODE solver for carrying out the time evolution. In section 4.2 we then extend our semidiscrete scheme to multidimensional hyperbolic and (degenerate) parabolic problems.

We end by presenting several numerical examples in section 5, in which we approximate solutions to hyperbolic conservation laws as well as to convection-diffusion equations. Our new method is shown to enjoy the expected high accuracy as well as the robustness and the simplicity of the entire family of central schemes.

2. Central schemes for conservation laws. We briefly overview the framework of central schemes for conservation laws. Consider the 1D system (1.1). To approximate its solutions, we introduce a spatial scale, Δx, and integrate over the cell $I(x) := \{\xi \mid |\xi - x| \le \Delta x/2\}$,

$$\bar{u}_t + \frac{1}{\Delta x}\left[f\left(\left(x+\frac{\Delta x}{2},t\right)\right) + f\left(u\left(x-\frac{\Delta x}{2},t\right)\right)\right] = 0. \tag{2.1}$$

Here and below, $\bar{u}$ denotes the average of u over I,

$$\bar{u}(x,t) := \frac{1}{\Delta x}\int_{I(x)} u(\xi,t)d\xi.$$

Introducing a time scale, Δt, integrating in time from t to $t+\Delta t$, and sampling (2.1) at the cells $[x_j, x_{j+1}]$, we obtain

$$\bar{u}^{n+1}_{j+1/2} = \bar{u}^n_{j+1/2} - \frac{1}{\Delta x}\int_{\tau=t^n}^{t^{n+1}} \left[f(u(x_{j+1},\tau)) - f(u(x_j,\tau))\right] d\tau, \tag{2.2}$$

where $x_j := j\Delta x$, $t^n := n\Delta t$, $u^n_j := u(x_j,t^n)$, and $\bar{u}^n_j := \bar{u}(x_j,t^n)$. Assuming that at time $t = t^n$ we have computed the cell-averages of the approximate solution, $\{\bar{u}^n_j\}$, we would like to utilize (2.2) to compute the cell-averages at the next time level, $t^{n+1} = t^n + \Delta t$. To that extent, we introduce a piecewise-polynomial reconstruction,

$$u(x,t^n) \approx \sum_j P_j(x)\chi_j(x), \tag{2.3}$$

where $\chi_j(x)$ is the characteristic function of the cell $I_j := I(x_j)$, and $P_j(x)$ is a polynomial which is reconstructed from the computed cell-averages, $\{\bar{u}^n_j\}$. The degree of the polynomial depends on the desired order of accuracy of the method. Having such an approximation to $u(x,t^n)$, (2.3), we can easily compute the right-hand side (RHS) of (2.2). The first term, $\bar{u}^n_{j+1/2}$, equals

$$\bar{u}^n_{j+1/2} = \int_{x_j}^{x_{j+1/2}} P_j(x)dx + \int_{x_{j+1/2}}^{x_{j+1}} P_{j+1}(x)dx.$$

For a sufficiently small time-step, Δt, the solution of (1.1) subject to the initial data (2.3), prescribed at time $t = t^n$, will remain smooth at some neighborhood of the grid points x_j for $t \in [t^n, t^{n+1}]$. Hence, the integrals on the RHS of (2.2) can be approximated using a sufficiently accurate quadrature, which is determined by the overall desired accuracy of the method. The values at the intermediate times which will be required in the quadrature can be predicted either by a Taylor expansion or using a Runge–Kutta method (consult [3, 19, 28, 30]).

For example, a piecewise-constant reconstruction, $P_j(x) = \bar{u}_j^n$, and a first-order quadrature,

$$\int_{t^n}^{t^{n+1}} f(u(t))dt \sim \Delta t f(\bar{u}^n),$$

will result in the staggered-LxF scheme (with $\lambda := \Delta t/\Delta x$ denoting the mesh ratio),

$$\bar{u}_{j+1/2}^{n+1} = \frac{\bar{u}_{j+1}^n + \bar{u}_j^n}{2} - \lambda(f(\bar{u}_{j+1}^n) - f(\bar{u}_j^n)).$$

A piecewise linear reconstruction, $P_j(x) = \bar{u}_j^n + (u_x)_j^n(x - x_j)$, with a second-order quadrature in time (such as the midpoint rule), results in the NT scheme. Applying nonlinear limiters on the discrete slopes, $(u_x)_j^n$, will prevent oscillations (for details, see [30]).

To obtain a third-order central scheme, one should use a third-order, piecewise parabolic reconstruction together with a more accurate quadrature in time, e.g., Simpson's quadrature rule (see [28] for details).

Remarks.

1. *Robustness.* In order to reconstruct a nonoscillatory interpolant, one typically is required to use nonlinear limiters. These limiters decrease the order of accuracy of the method at extrema and by that they play a stabilizing role (e.g., see [17, 28, 30, 35]).

2. *Numerical dissipation and time step.* When using fully discrete central schemes to approximate solutions of convection-diffusion equations, (1.2), the stability restriction enforces small time-steps, $\Delta t \sim (\Delta x)^2$. That is why the numerical dissipation is accumulated and we do not obtain high resolution of discontinuities (see [16] for details).

This problem can be avoided by using semidiscrete schemes instead of the fully-discrete schemes. Such a second-order, central, semidiscrete scheme was introduced in [16]. In this paper we develop a third-order, central, semidiscrete scheme with small numerical dissipation, which can be used efficiently with the small time-steps required due to the second-order operators.

3. *Upwind schemes.* Sampling (2.1) at the cells I_j will result in upwind schemes. Here, one remains with the discontinuities along the interfaces and is bound to solve the Riemann problems there, or at least to approximate their solutions. In the scalar, 1D case this can be easily accomplished, but the Riemann problem has no known solution in the general case of systems and/or several space dimensions.

This is the reason why central schemes can be considered as universal methods for solving hyperbolic conservation laws: Riemann solvers are not involved in their construction and, moreover, since (2.2) can be carried out componentwise, no characteristic decomposition is required.

2.1. CWENO reconstruction. The first 1D, third-order central scheme in [28] implemented the nonoscillatory piecewise parabolic reconstruction proposed by Liu

and Osher in [25]. Since then, a variety of simpler reconstructions has appeared in the literature. Among these, we would like to mention the central-ENO reconstruction in [3] and the central-WENO (CWENO) reconstruction in [19] and [21], which was extended to the two-dimensional (2D) setup in [20] and [22].

Our new third-order semidiscrete method which we develop in section 3 and section 4 below can be integrated with any third-order, nonoscillatory reconstruction. In our numerical simulations presented in section 5, we will use the method recently presented in [21], of which we will now give a brief overview.

In each cell I_j we reconstruct a quadratic polynomial as a convex combination of three polynomials $P_{\rm L}(x), P_{\rm R}(x)$, and $P_{\rm C}(x)$,

$$P_j(x) = w_{\rm L}P_{\rm L}(x) + w_{\rm R}P_{\rm R}(x) + w_{\rm C}P_{\rm C}(x), \tag{2.4}$$

with positive weights $w_i \geq 0 \forall i \in \{{\rm C}, {\rm R}, {\rm L}\}$ and $\sum_i w_i = 1$. The polynomials $P_{\rm L}(x), P_{\rm R}(x)$ correspond to left and right one-sided linear reconstructions, respectively, while $P_{\rm C}(x)$ is a parabola, centered around x_j.

The linear functions, $P_{\rm R}(x)$ and $P_{\rm L}(x)$, are uniquely determined by requiring them to conserve the one-sided cell-averages ($\bar{u}^n_j, \bar{u}^n_{j+1}$ and $\bar{u}^n_j, \bar{u}^n_{j-1}$, respectively) as

$$P_{\rm R}(x) = \bar{u}^n_j + \frac{\bar{u}^n_{j+1} - \bar{u}^n_j}{\Delta x}(x - x_j), \quad P_{\rm L}(x) = \bar{u}^n_j + \frac{\bar{u}^n_j - \bar{u}^n_{j-1}}{\Delta x}(x - x_j). \tag{2.5}$$

The centered parabola, $P_{\rm C}(x)$, is chosen so as to satisfy

$$P_{\rm EXACT}(x) = c_{\rm L}P_{\rm L}(x) + c_{\rm R}P_{\rm R}(x) + (1 - c_{\rm L} - c_{\rm R})P_{\rm C}(x) \tag{2.6}$$

with constants c_i's. Here, $P_{\rm EXACT}(x)$ is the unique parabola that conserves the three-cell-averages, $\bar{u}^n_{j-1}, \bar{u}^n_j$, and $\bar{u}^n_{j+1}$, which is given by

$$P_{\rm EXACT}(x) = u^n_j + u'_j(x - x_j) + \frac{1}{2}u''_j(x - xj)^2. \tag{2.7}$$

The approximations to the point-values of $u(x_j, t^n), u_x(x_j, t^n)$, and $u_{xx}(x_j, t^n)$ are denoted by u^n_j, u'_j, u''_j and are given by

$$u^n_j = \bar{u}^n_j - \frac{1}{24}(\bar{u}^n_{j+1} - 2\bar{u}^n_j + \bar{u}^n_{j-1}),$$
$$u'_j = \frac{\bar{u}^n_{j+1} - \bar{u}^n_{j-1}}{2\Delta x}, \qquad u''_j = \frac{\bar{u}^n_{j-1} - 2\bar{u}^n_j + \bar{u}^n_{j+1}}{\Delta x^2}.$$

In [21] it was shown that every symmetric selection of the constants c_i's in (2.6) will provide the desired third-order accuracy. For example, by taking $c_{\rm L} = c_{\rm R} = 1/4$, (2.5)–(2.7) yield

$$P_{\rm C}(x) = \bar{u}^n_j - \frac{1}{12}(\bar{u}^n_{j+1} - 2\bar{u}^n_j + \bar{u}^n_{j-1}) + \frac{\bar{u}^n_{j+1} - \bar{u}^n_{j-1}}{2\Delta x}(x - x_j) + \frac{\bar{u}^n_{j+1} - 2\bar{u}^n_j + \bar{u}^n_{j-1}}{\Delta x^2}(x - x_j)^2.$$

In smooth regions, the coefficients w_i of the convex combination in (2.4) are chosen to guarantee the maximum order of accuracy (in this particular case, order three), but in the presence of a discontinuity they are automatically switched to the best

one-sided stencil (which generates the least oscillatory reconstruction). The weights are taken as

$$w_i = \frac{\alpha_i}{\sum\limits_m \alpha_m}, \quad \text{where} \quad \alpha_i = \frac{c_i}{(\epsilon + IS_i)^p}, \qquad i, m \in \{\text{C}, \text{R}, \text{L}\}, \tag{2.8}$$
$$c_{\text{L}} = c_{\text{R}} = 1/4, \qquad c_{\text{C}} = 1/2.$$

The constant ϵ guarantees that the denominator does not vanish and is taken as $\epsilon = 10^{-6}$. The value of p may be chosen to provide the highest accuracy in smooth areas and to ensure the nonoscillatory nature of the solution near the discontinuities (consult [11]; see also [19, 21]). In [11] the value $p = 2$ was empirically selected, and here we use the same p in most of the examples presented below. Finally, the smoothness indicators, IS_i, are defined as

$$IS_i = \sum_{l=1}^{2} \int_{x_{j-1/2}}^{x_{j+1/2}} (\Delta x)^{2l-1} (P_i^{(l)}(x))^2 dx.$$

A direct computation then results in

$$IS_{\text{L}} = (\bar{u}_j^n - \bar{u}_{j-1}^n)^2, \qquad IS_{\text{R}} = (\bar{u}_{j+1}^n - \bar{u}_j^n)^2, \tag{2.9}$$
$$IS_{\text{C}} = \frac{13}{3}(\bar{u}_{j+1}^n - 2\bar{u}_j^n + \bar{u}_{j-1}^n)^2 + \frac{1}{4}(\bar{u}_{j+1}^n - \bar{u}_{j-1}^n)^2.$$

It is easy to see that in the presence of large gradients, this reconstruction switches to one of the second-order one-sided linear reconstructions, P_{R} or P_{L}. For more details we refer to [21].

3. The fully discrete one-dimensional construction. In this section we present the new third-order method in the fully discrete framework. Since we are mainly interested in deriving the semidiscrete scheme, we will concentrate only on the details which are required for that task. The scheme we derive here is a third-order extension of the fully discrete second-order scheme presented in [16].

Following [16], we would like to augment the integration over the Riemann fans by more accurate information about the *local* speed of wave propagation. We start by assuming that in every cell, I_j, we have reconstructed a piecewise polynomial interpolant, $P_j(x, t^n)$, from the previously computed cell-averages, $\{\bar{u}_j^n\}$. Then, an upper bound on the speed of propagation of discontinuities at the cell boundaries, $x_{j+1/2}$, is given by

$$a_{j+1/2}^n = \max_{u \in \mathcal{C}(u_{j+1/2}^-, u_{j+1/2}^+)} \rho\Big(\frac{\partial f}{\partial u}(u)\Big), \tag{3.1}$$

where $\rho(A)$ denotes the spectral radius of a matrix A, i.e., $\rho(A) := \max_i |\lambda_i(A)|$ with $\lambda_i(A)$ being its eigenvalues. We denote by $u_{j+1/2}^+$ and $u_{j+1/2}^-$ the left and right intermediate values of $u(x, t^n)$ at $x_{j+1/2}$, i.e.,

$$u_{j+1/2}^+ := P_{j+1}(x_{j+1/2}, t^n), \quad u_{j+1/2}^- := P_j(x_{j+1/2}, t^n),$$

and by $\mathcal{C}(u_{j+1/2}^-, u_{j+1/2}^+)$ a curve in phase space that connects $u_{j+1/2}^-$ and $u_{j+1/2}^+$ via the Riemann fan.

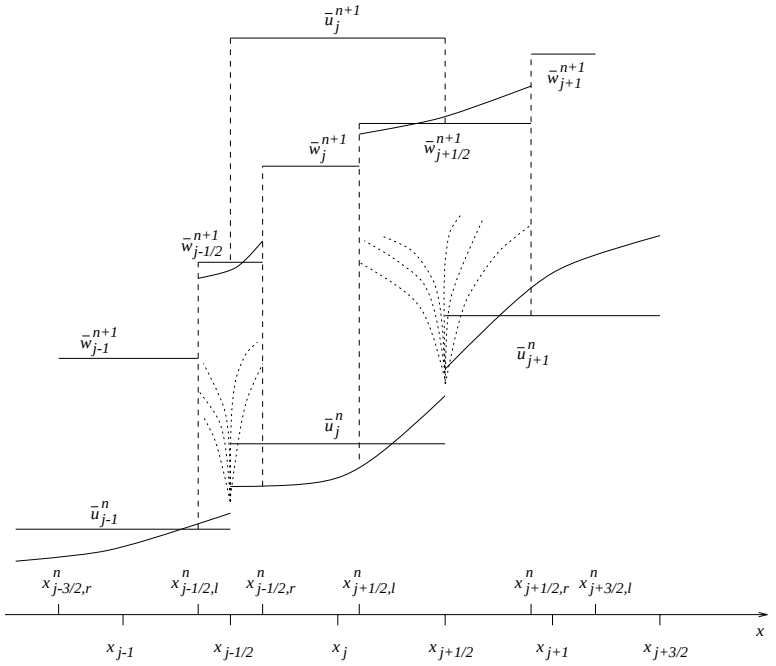

FIG. 3.1. *Modified central differencing.*

Remark. In most practical applications, these local maximal speeds can be easily evaluated. For example, in the genuinely nonlinear or linearly degenerate case one finds that (3.1) reduces to

$$a_{j+1/2}^n := \max\left\{\rho\Big(\frac{\partial f}{\partial u}(u_{j+1/2}^-)\Big), \rho\Big(\frac{\partial f}{\partial u}(u_{j+1/2}^+)\Big)\right\}. \tag{3.2}$$

Given the piecewise polynomial interpolant at time t^n, $\{P_j(x,t^n)\}$, and the local speeds of propagation, $\{a_{j+1/2}^n\}$, we construct the fully discrete, central method in two steps, which are schematically described in Figure 3.1. First, we integrate over the control volumes, $[x_{j-1/2,l}^n, x_{j-1/2,r}^n]\times[t^n,t^{n+1}]$, $[x_{j-1/2,r}^n, x_{j+1/2,l}^n]\times[t^n,t^{n+1}]$, and $[x_{j+1/2,l}^n, x_{j+1/2,r}^n]\times[t^n,t^{n+1}]$, obtaining $\bar{w}_{j-1/2}^{n+1}$, $\bar{w}_j^{n+1}$, and $\bar{w}_{j+1/2}^{n+1}$, respectively. Due to the finite speed of propagation, the points $x_{j+1/2,l}^n$ and $x_{j+1/2,r}^n$,

$$x_{j+1/2,l}^n := x_{j+1/2} - a_{j+1/2}^n\Delta t, \quad x_{j+1/2,r}^n := x_{j+1/2} + a_{j+1/2}^n\Delta t,$$

separate between smooth and nonsmooth regions. That is, the solution of (1.1) subject to the piecewise polynomial initial data prescribed at time $t = t^n$ may be nonsmooth only inside the intervals $[x_{j+1/2,l}^n\,,\, x_{j+1/2,r}^n]$ for $t\in[t^n,t^{n+1})$.

In the second step, we repeat the nonoscillatory reconstruction (this time on a nonuniformly spaced grid) and project the obtained reconstruction on the original, uniform grid, ending up with the cell-averages at the next time level t^{n+1}, $\{\bar{u}_j^{n+1}\}$. This last step does not involve time integration and was introduced in the context of changing staggered methods into nonstaggered methods in [10].

We now turn to the detailed description of this algorithm. Assume that the

piecewise polynomial reconstruction in cell I_j at time t^n is of the form

$$P_j(x,t^n)=A_j+B_j(x-x_j)+\frac{1}{2}C_j(x-x_j)^2. \tag{3.3}$$

Then a direct computation of the integrals over the control volumes, $[x^n_{j+1/2,l},x^n_{j+1/2,r}]\times[t^n,t^{n+1}]$ and $[x^n_{j-1/2,r},x^n_{j+1/2,l}]\times[t^n,t^{n+1}]$, yields

$$\begin{aligned}\bar w^{n+1}_{j+1/2}&=\frac{A_j+A_{j+1}}{2}+\frac{\Delta x-a^n_{j+1/2}\Delta t}{4}(B_j-B_{j+1})\\&+\left(\frac{\Delta x^2}{16}-\frac{a^n_{j+1/2}\Delta t\Delta x}{8}+\frac{(a^n_{j+1/2}\Delta t)^2}{12}\right)(C_j+C_{j+1})\\&-\frac{1}{2a^n_{j+1/2}\Delta t}\left\{\int_{t^n}^{t^{n+1}}\Big[f(u(x^n_{j+1/2,r},t))dt-f(u(x^n_{j+1/2,l},t))\Big]\,dt\right\},\end{aligned} \tag{3.4}$$

and

$$\begin{aligned}\bar w^{n+1}_{j}&=A_j+\frac{\Delta t}{2}(a^n_{j-1/2}-a^n_{j+1/2})B_j\\&+\left[\frac{(\Delta x)^2}{24}-\frac{\Delta t\Delta x}{12}(a^n_{j-1/2}+a^n_{j+1/2})\right.\\&\qquad\left.+\frac{(\Delta t)^2}{6}\Big((a^n_{j-1/2})^2-a^n_{j-1/2}a^n_{j+1/2}+(a^n_{j+1/2})^2\Big)\right]C_j\\&-\frac{1}{\Delta x-\Delta t(a^n_{j-1/2}+a^n_{j+1/2})}\left\{\int_{t^n}^{t^{n+1}}\Big[f(u(x^n_{j+1/2,l},t))dt\right.\\&\qquad\left.-f(u(x^n_{j-1/2,r},t))\Big]\,dt\right\},\end{aligned} \tag{3.5}$$

respectively. To complete these computations, one should approximate the flux integrals on the RHS of (3.4) and (3.5) using, e.g., Simpson's quadrature as described in section 2.

At this stage, the approximate cell-averages, $\{\bar w^{n+1}_{j+\frac12},\,\bar w^{n+1}_j\}$, realize the solution at $t=t^{n+1}$ over a nonuniform grid, which is oversampled by twice the number of the original cells at $t=t^n$. To convert these nonuniform averages back into the original grid, we proceed along the lines of [10].

First, from the cell-averages, $\bar w^{n+1}_{j+\frac12}$, $\bar w^{n+1}_j$, given by (3.4)–(3.5), we reconstruct a third-order, piecewise polynomial, nonoscillatory interpolant (e.g., the CWENO interpolant described in section 2.1), which we will denote by $\tilde w^{n+1}_{j+1/2}(x)$ and $\tilde w^{n+1}_j(x)$, respectively. In fact, we do not need any high-order reconstruction $\tilde w^{n+1}_j(x)$ since it will be averaged out (consult Figure 3.1).

We note in passing that even for a nonuniform grid data, the CWENO interpolant can be written explicitly (in the spirit of section 2.1), but these details are irrelevant for the semidiscrete scheme, which will be described in section 4. At that point, all we need is to assume that such a reconstruction exists and that for all j it takes the form

$$\begin{aligned}\tilde w^{n+1}_{j+1/2}(x)&=\tilde A_{j+1/2}+\tilde B_{j+1/2}(x-x_{j+1/2})+\frac{1}{2}\tilde C_{j+1/2}(x-x_{j+1/2})^2,\\ \tilde w^{n+1}_{j}(x)&=\bar w^{n+1}_j,\end{aligned} \tag{3.6}$$

in the nonsmooth region $(x^n_{j+1/2,l}, x^n_{j+1/2,r})$ and in the smooth region $(x^n_{j-1/2,r}, x^n_{j+1/2,l})$, respectively. Given (3.4), (3.5), and (3.6), we conclude by computing the new cell-averages at time t^{n+1} according to

$$\begin{aligned}\bar{u}^{n+1}_j &= \frac{1}{\Delta x}\left[\int_{x_{j-1/2}}^{x^n_{j-1/2,r}} \tilde{w}^{n+1}_{j-1/2}(x)dx + \int_{x^n_{j-1/2,r}}^{x^n_{j+1/2,l}} \tilde{w}^{n+1}_{j}(x)dx + \int_{x^n_{j+1/2,l}}^{x_{j+1/2}} \tilde{w}^{n+1}_{j+1/2}(x)dx\right]\\ &= \lambda a^n_{j-1/2}\tilde{A}_{j-1/2} + \left[1-\lambda(a^n_{j-1/2}+a^n_{j+1/2})\right]\bar{w}^{n+1}_j + \lambda a^n_{j+1/2}\tilde{A}_{j+1/2}\\ &\quad + \frac{\lambda\Delta t}{2}\left((a^n_{j-1/2})^2\tilde{B}_{j-1/2} - (a^n_{j+1/2})^2\tilde{B}_{j+1/2}\right)\\ &\quad + \frac{\lambda(\Delta t)^2}{6}\left((a^n_{j-1/2})^3\tilde{C}_{j-1/2} + (a^n_{j+1/2})^3\tilde{C}_{j+1/2}\right).\end{aligned} \tag{3.7}$$

Remark. The third-order reconstruction (3.6) is necessary in order to guarantee the overall third-order accuracy, since simple averaging over $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ (without reconstruction) reduces the order of the resulting scheme (see [10]).

4. The semidiscrete scheme. We are now ready to derive our main result, which is the new third-order, semidiscrete, central scheme. First, we describe our ideas in the 1D framework and then we extend them to multidimensional problems.

4.1. 1D problems. We start with the derivation of the third-order semidiscrete scheme for 1D (systems of) hyperbolic conservation laws. Using the fully discrete scheme obtained in section 3, the semidiscrete approximation can be directly written as the limit

$$\frac{d}{dt}\bar{u}_j(t) = \lim_{\Delta t\to 0}\frac{\bar{u}^{n+1}_j - \bar{u}^n_j}{\Delta t}. \tag{4.1}$$

Substituting (3.7) into (4.1) results in

$$\begin{aligned}\frac{d\bar{u}_j}{dt} = \lim_{\Delta t\to 0}\Bigg\{&\frac{1}{\Delta x}a^n_{j-1/2}\tilde{A}_{j-1/2} - \frac{1}{\Delta x}(a^n_{j-1/2}+a^n_{j+1/2})\bar{w}^{n+1}_j\\ &+\frac{1}{\Delta x}a^n_{j+1/2}\tilde{A}_{j+1/2} + \frac{1}{\Delta t}(\bar{w}^{n+1}_j - \bar{u}^n_j)\Bigg\}.\end{aligned} \tag{4.2}$$

In the limit as $\Delta t\to 0$, all the Riemann fans have zero widths and therefore,

$$\tilde{A}_{j+1/2} = \bar{w}^{n+1}_{j+1/2}, \qquad \tilde{A}_{j-1/2} = \bar{w}^{n+1}_{j-1/2}. \tag{4.3}$$

Using (3.3) we can also obtain

$$\begin{aligned}u(x^n_{j+1/2,r},t) &\to P_{j+1}(x_{j+1/2},t)\\ &= A_{j+1} - \frac{\Delta x}{2}B_{j+1} + \frac{(\Delta x)^2}{8}C_{j+1} =: u^+_{j+1/2}(t),\\ u(x^n_{j+1/2,l},t) &\to P_j(x_{j+1/2},t)\\ &= A_j + \frac{\Delta x}{2}B_j + \frac{(\Delta x)^2}{8}C_j =: u^-_{j+1/2}(t).\end{aligned} \tag{4.4}$$

Finally, plugging (3.4), (3.5), and (4.3) into (4.2) we compute the time limit explicitly, ending up with our new semidiscrete scheme,

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{2\Delta x}\left[f(u^+_{j+1/2}(t)) + f(u^-_{j+1/2}(t)) - f(u^+_{j-1/2}(t)) - f(u^-_{j-1/2}(t))\right]$$

$$+\frac{a_{j+1/2}(t)}{2\Delta x}\left[u^+_{j+1/2}(t)-u^-_{j+1/2}(t)\right]-\frac{a_{j-1/2}(t)}{2\Delta x}\left[u^+_{j-1/2}(t)-u^-_{j-1/2}(t)\right], \tag{4.5}$$

with local speeds $a_{j+1/2}(t)$, e.g.,

$$a_{j+1/2}(t):=\max\left\{\rho\Big(\frac{\partial f}{\partial u}(u^-_{j+1/2}(t))\Big),\rho\Big(\frac{\partial f}{\partial u}(u^+_{j+1/2}(t))\Big)\right\}.$$

Remarks.

1. We would like to emphasize that the scheme (4.5) was derived independently of any specific piecewise-quadratic reconstruction. If one wants, e.g., to use the CWENO reconstruction described in section 2.1, then the values of A_j, B_j, and C_j in (4.4) are

$$\begin{aligned}
A_j&=\bar u^n_j-\frac{w_{\rm C}}{12}(\bar u^n_{j+1}-2\bar u^n_j+\bar u^n_{j-1}),\\
B_j&=\frac{1}{\Delta x}\left[w_{\rm R}(\bar u^n_{j+1}-\bar u^n_j)+w_{\rm C}\frac{\bar u^n_{j+1}-\bar u^n_{j-1}}{2}+w_{\rm L}(\bar u^n_j-\bar u^n_{j-1})\right],\\
C_j&=2w_{\rm C}\frac{\bar u^n_{j-1}-2\bar u^n_j+\bar u^n_{j+1}}{\Delta x^2},
\end{aligned}$$

where $w_{\rm L}, w_{\rm C}$, and $w_{\rm R}$ are defined in (2.8).

2. Our third-order scheme, (4.5), admits the conservative form

$$\frac{d\bar u_j}{dt}=-\frac{H_{j+1/2}(t)-H_{j-1/2}(t)}{\Delta x}, \tag{4.6}$$

with the numerical flux

$$\begin{aligned}
H_{j+1/2}(t):=&\frac{f(u^+_{j+1/2}(t))+f(u^-_{j+1/2}(t))}{2}\\
&-\frac{a_{j+1/2}(t)}{2}\left[u^+_{j+1/2}(t)-u^-_{j+1/2}(t)\right].
\end{aligned} \tag{4.7}$$

This scheme is a natural generalization of the second-order semidiscrete scheme from [16]. Moreover, the second-order scheme has exactly the same form, (4.6)–(4.7); the only difference is in the more accurate computation of the intermediate value $u^+_{j+1/2}(t)$ and $u^-_{j+1/2}(t)$. It is interesting to note that also the fully discrete, staggered, second- and third-order central schemes have the same structure (see [28]).

3. Similar to the case of the second-order scheme [16], the nonoscillatory property of the piecewise parabolic reconstruction, (3.3), will guarantee the nonoscillatory nature of our semidiscrete scheme. But unlike the piecewise linear reconstruction utilized in the second-order method, a piecewise parabolic reconstruction can be only essentially nonoscillatory. This means that, in principle, such a reconstruction may increase the total variation of the computed piecewise constant solution. Our numerical examples, however, demonstrate that the growth of the total variation is always bounded. Such desirable behavior of bounded total variation in the context of central-WENO schemes was already observed in [23].

4. We would like to stress once again the simplicity of our new method, which does not require any (approximate) Riemann solver or any use of the characteristic variables—the reconstruction of piecewise polynomial interpolant, (3.3), is carried

out *componentwise*. In particular, unlike the standard central schemes, but similar to the second-order semidiscrete method in [16], our method is based on one grid (and not on staggering between two grids). This can be a big advantage (compared with the traditional central schemes) when dealing with boundary conditions and complex geometries.

5. Finally, similar to the second-order semidiscrete scheme [16], the third-order scheme, (4.6)–(4.7), boils down in the scalar linear case to an upwind scheme. In the nonlinear, *scalar*, case, this is a finite volume scheme based on a local LxF (Rusanov) monotone flux. Since we derived our method as a central Godunov-type scheme, our result also naturally holds for systems (including multi-dimensional systems as shown below).

Next, let us consider the general convection-diffusion equation, (1.2). Similar to the case of the second-order semidiscrete scheme [16], operator splitting is not needed. We can apply our third-order semidiscrete scheme, (4.6)–(4.7), to the (degenerate) parabolic equation, (1.2), in a straightforward manner. This results in the scheme

$$\frac{d\bar{u}_j}{dt} = -\frac{H_{j+1/2}(t) - H_{j-1/2}(t)}{\Delta x} + Q_j(t). \tag{4.8}$$

Here, $H_{j+1/2}(t)$ is our numerical convection flux, (4.7), and $Q_j(t)$ is a high-order approximation to the diffusion term, $Q(u, u_x)_x$. In the examples below we use the fourth-order central differencing of the form

$$\begin{aligned} Q_j(t) = \frac{1}{12\Delta x}\Big[&- Q(u_{j+2}(t), (u_x)_{j+2,j}) + 8Q(u_{j+1}(t), (u_x)_{j+1,j}) \\ &-8Q(u_{j-1}(t), (u_x)_{j-1,j}) + Q(u_{j-2}(t), (u_x)_{j-2,j})\Big], \end{aligned} \tag{4.9}$$

where

$$\begin{aligned} (u_x)_{j+2,j} &:= \frac{1}{12\Delta x}\Big[25u_{j+2}(t) - 48u_{j+1}(t) + 36u_j - 16u_{j-1}(t) + 3u_{j-2}(t)\Big], \\ (u_x)_{j+1,j} &:= \frac{1}{12\Delta x}\Big[3u_{j+2}(t) + 10u_{j+1}(t) - 18u_j + 6u_{j-1}(t) - u_{j-2}(t)\Big], \\ (u_x)_{j-1,j} &:= \frac{1}{12\Delta x}\Big[u_{j+2}(t) - 6u_{j+1}(t) + 18u_j - 10u_{j-1}(t) - 3u_{j-2}(t)\Big], \\ (u_x)_{j-2,j} &:= \frac{1}{12\Delta x}\Big[-3u_{j+2}(t) + 16u_{j+1}(t) - 36u_j + 48u_{j-1}(t) - 25u_{j-2}(t)\Big], \end{aligned} \tag{4.10}$$

and $\{u_j(t)\}$ are point-values of the reconstructed polynomials, (3.3), i.e., $u_j(t) = P_j(x_j, t)$.

4.2. Multidimensional extensions. Without loss of generality, let us consider the 2D (system of) convection-diffusion equations,

$$u_t + f(u)_x + g(u)_y = Q^x(u, u_x, u_y)_x + Q^y(u, u_x, u_y)_y, \tag{4.11}$$

where the case $Q^x \equiv Q^y \equiv 0$ corresponds to the 2D pure hyperbolic problem.

Suppose that we have computed an approximate solution to (4.11) at some time t and have reconstructed a 2D piecewise polynomial, third-order, ENO interpolant over the uniform spatial grid, $(x_j, y_k) = (j\Delta x, k\Delta y)$.

Following [16], the 2D extension of our third-order semidiscrete scheme, (4.8),(4.7), can be written in the following form:

$$\frac{d\bar{u}_{j,k}}{dt} = -\frac{H^x_{j+1/2,k}(t) - H^x_{j-1/2,k}(t)}{\Delta x} - \frac{H^y_{j,k+1/2}(t) - H^y_{j,k-1/2}(t)}{\Delta y} + Q^x_{j,k}(t) + Q^y_{j,k}(t). \tag{4.12}$$

Here, $H^x_{j+1/2,k}(t)$ and $H^y_{j,k+1/2}(t)$ are x- and y-numerical convection fluxes, respectively (they can be viewed as a generalization of the 1D flux, (4.7)),

$$\begin{aligned} H^x_{j+1/2,k}(t) &:= \frac{f(u^+_{j+1/2,k}(t)) + f(u^-_{j+1/2,k}(t))}{2} \\ &\quad - \frac{a^x_{j+1/2,k}(t)}{2}\left[u^+_{j+1/2,k}(t) - u^-_{j+1/2,k}(t)\right], \\ H^y_{j,k+1/2}(t) &:= \frac{g(u^+_{j,k+1/2}(t)) + g(u^-_{j,k+1/2}(t))}{2} \\ &\quad - \frac{a^y_{j,k+1/2}(t)}{2}\left[u^+_{j,k+1/2}(t) - u^-_{j,k+1/2}(t)\right]. \end{aligned} \tag{4.13}$$

The numerical fluxes, (4.13), are expressed in terms of the intermediate values, $u^{\pm}_{j+1/2,k}(t)$, $u^{\pm}_{j,k+1/2}(t)$, which are obtained from the piecewise polynomial reconstruction. The local speeds, $a^x_{j+1/2,k}(t)$ and $a^y_{j,k+1/2}(t)$, are computed, e.g., by

$$\begin{aligned} a^x_{j+1/2,k}(t) &:= \max\left\{\rho\Big(\frac{\partial f}{\partial u}(u^-_{j+1/2,k}(t))\Big), \rho\Big(\frac{\partial f}{\partial u}(u^+_{j+1/2,k}(t))\Big)\right\}, \\ a^y_{j,k+1/2}(t) &:= \max\left\{\rho\Big(\frac{\partial g}{\partial u}(u^-_{j,k+1/2}(t))\Big), \rho\Big(\frac{\partial g}{\partial u}(u^+_{j,k+1/2}(t))\Big)\right\}. \end{aligned} \tag{4.14}$$

Finally, $Q^x_{j,k}(t)$ and $Q^y_{j,k}(t)$ are high-order, central differencing approximations to the diffusion terms $Q^x(u, u_x, u_y)_x$ and $Q^y(u, u_x, u_y)_y$.

Remarks.

1. We would like to emphasize that the problem of constructing a 2D, third-order, nonoscillatory interpolant is highly nontrivial. Several essentially 2D reconstructions were proposed in [20, 21, 22]. Alternatively, one can use 1D CWENO reconstruction, direction by direction, in order to compute the intermediate values $u^{\pm}_{j+1/2,k}(t)$ and $u^{\pm}_{j,k+1/2}(t)$.

Following is the recipe for the computation of $u^-_{j+1/2,k}$ (the computation of other intermediate values can be carried out in a similar way):

$$u^-_{j+1/2,k} = w_{\rm L}P^k_{\rm L}(x_{j+1/2}) + w_{\rm R}P^k_{\rm R}(x_{j+1/2}) + w_{\rm C}P^k_{\rm C}(x_{j+1/2}), \tag{4.15}$$

where the P's are the polynomials introduced in section 2.1,

$$\begin{aligned} P^k_{\rm R}(x) &= \bar{u}_{j,k} + \frac{\bar{u}_{j+1,k} - \bar{u}_{j,k}}{\Delta x}(x - x_j), \\ P^k_{\rm L}(x) &= \bar{u}_{j,k} + \frac{\bar{u}_{j,k} - \bar{u}_{j-1,k}}{\Delta x}(x - x_j), \end{aligned}$$

(4.16)
$$P_{\rm C}^k(x) = \bar{u}_{j,k} - \frac{1}{12}(\bar{u}_{j+1,k} - 2\bar{u}_{j,k} + \bar{u}_{j-1,k}) - \frac{1}{12}(\bar{u}_{j,k+1} - 2\bar{u}_{j,k} + \bar{u}_{j,k-1}) + \frac{\bar{u}_{j+1,k} - \bar{u}_{j-1,k}}{2\Delta x}(x - x_j) + \frac{\bar{u}_{j+1,k} - 2\bar{u}_{j,k} + \bar{u}_{j-1,k}}{\Delta x^2}(x - x_j)^2.$$

The weights, $w_{\rm L}, w_{\rm R}, w_{\rm C}$, which are given by (2.8), are based on the smoothness indicators in (2.9).

Note that the only difference between this reconstruction and the 1D reconstruction, (2.4)–(2.9), is an additional term in $P_{\rm C}^k(x)$, $-\frac{1}{12}(\bar{u}_{j,k+1} - 2\bar{u}_{j,k} + \bar{u}_{j,k-1})$, which corresponds to the second derivative in the y direction and guarantees the third-order accuracy of the computed intermediate values. This "dimension by dimension" approach was implemented in Example 5 below.

2. It is straightforward to extend the 2D scheme, (4.12), to more space dimensions. In particular, the dimension-by-dimension approach is a very simple and promising approach for multidimensional problems.

5. Numerical examples. We conclude the paper with a number of numerical examples. Here, in order to retain the overall high accuracy, the semidiscrete scheme is combined with a high-order, stable ODE solver to complete the spatio-temporal discretization. Numerically, we observed that a variety of explicit methods provides satisfactory results in the context of our semidiscrete scheme.

For the inviscid problems (Examples 1, 2, 3, and 5), we used the third-order total variation diminishing (TVD) Runge–Kutta-type method introduced by Shu and Osher in [33]. However, if we apply this time-integration method or any other standard Runge–Kutta-type method to (degenerate) parabolic problems, the time-step can be very small due to their strict stability restrictions.

To overcome this difficulty, we used (in Examples 4 and 5) the third-order ODE solver (called DUMKA3) by Medovikov [29]. This explicit method has larger stability domains (compared with the standard Runge–Kutta methods), which allow larger time-steps. In practice, DUMKA3 works as fast as implicit methods (see [29] for details).

We abbreviate our third-order semidiscrete scheme by SD3, which will be combined with the third-order TVD Runge–Kutta-type method (RK3) or with DUMKA3.

Example 1: Linear accuracy test. Consider the scalar linear hyperbolic equation

$$u_t + u_x = 0, \qquad x \in [0, 2\pi], \tag{5.1}$$

augmented with the smooth initial data, $u(x,0) = \sin x$, and periodic boundary conditions. This simple problem admits a global classical solution, which was computed at time $T = 1$ with a varying number of grid points, N.

In Table 5.1 we check the accuracy of our third-order semidiscrete scheme, SD3, coupled with the RK3 ODE solver. Clearly, this is a high-order method. The asymptotic convergence rate seems to be better than three, which is similar to the superconvergence observed in [27].

The error is measured in terms of the pointwise values,

$$\|\tilde{u} - u\|_{L^1} := \Delta x \sum_j |\tilde{u}_j(T) - u(x_j, T)|, \quad \|\tilde{u} - u\|_{L^\infty} := \max_j |\tilde{u}_j(T) - u(x_j, T)|.$$

TABLE 5.1
Accuracy test for the linear advection problem (5.1)*; the errors at* $T = 1$.

N	L^1-error	Rate	L^∞-error	Rate
40	4.492e-02	–	2.822e-02	–
80	1.092e-02	2.04	1.065e-02	1.41
160	2.162e-03	2.34	3.426e-03	1.64
320	1.811e-04	3.58	4.705e-04	2.86
640	9.267e-06	4.29	2.267e-05	4.38
1280	5.409e-07	4.10	1.171e-06	4.27

TABLE 5.2
Accuracy test for the Burgers equation, (5.2)*; the pre-shock errors.*

N	L^1-error	Rate	L^∞-error	Rate
40	2.370e-02	–	2.225e-02	–
80	5.759e-03	2.04	9.053e-03	1.30
160	1.161e-03	2.31	2.921e-03	1.63
320	9.541e-05	3.61	3.926e-04	2.90
640	4.882e-06	4.29	1.778e-05	4.46
1280	3.044e-07	4.00	5.732e-07	4.96

Here, $\tilde{u}$ is an approximate solution, which is realized by its values at the grid points, x_j,

$$\tilde{u}_j(T) = P_j(x_j, T),$$

where the P_j's are the piecewise parabolic interpolants, (3.3), constructed at the final time $t = T$.

Example 2: The Burgers equation. In this example we approximate solutions to the inviscid Burgers equation,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \qquad x \in [0, 2\pi], \tag{5.2}$$

augmented with the smooth initial data, $u(x, 0) = 0.5 + \sin x$, and periodic boundary conditions.

The unique entropy solution of (5.2) develops a shock discontinuity at the critical time $T_c = 1$. Table 5.2 shows the L^1- and L^∞-norms of the errors at the preshock time $T = 0.5$, when the solution is still smooth. Once again, our results indicate that the method is a high-order method also when implemented for nonlinear problems.

In Figures 5.1 and 5.2 we present the approximate solutions at the postshock time $T = 2$, when the shock is well developed. The essentially nonoscillatory nature of our scheme can be clearly observed.

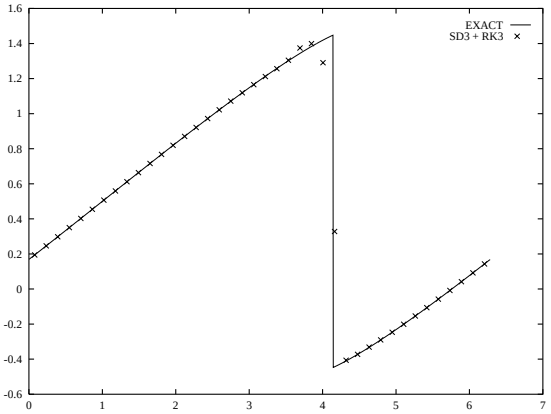

FIG. 5.1. *The Burgers equation,* (5.2)*;* $T = 2$*,* $N = 40$.

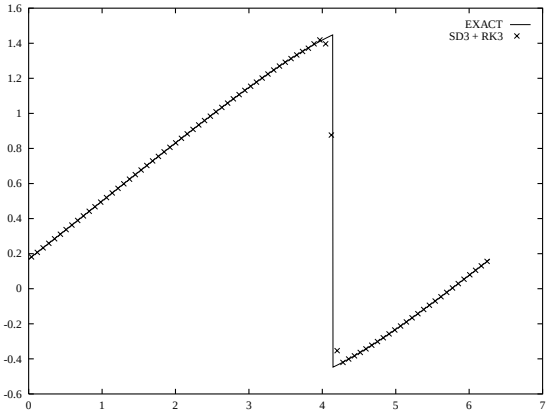

FIG. 5.2. *The Burgers equation,* (5.2)*;* $T = 2$*,* $N = 80$.

Example 3: Euler equations of gas dynamics. Let us consider the 1D Euler system

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}m\\ \rho u^2+p\\ u(E+p)\end{bmatrix}=0,\qquad p=(\gamma-1)\cdot\left(E-\frac{\rho}{2}u^2\right),$$

where ρ, u, $m=\rho u$, p, and E are the density, velocity, momentum, pressure, and total energy, respectively. We solve this system subject to Sod's Riemann initial data, proposed in [34],

$$\vec{u}(x,0)=\begin{cases}\vec{u}_L=(1,0,2.5)^T, & x<0,\\ \vec{u}_R=(0.125,0,0.25)^T, & x>0.\end{cases}$$

The approximations to the density, velocity, and pressure obtained by the SD3 scheme with the RK3 time discretization are presented in Figures 5.3–5.8. The coefficient p in the smoothness indicator, (2.8)–(2.9), was taken as 0.6, which seems to be the optimal value in this specific example. WENO-type schemes do require a parameter tuning in order to reduce the amplitude of the oscillations. Formally, this does not affect the order of accuracy of the method (in certain ranges of the parameters).

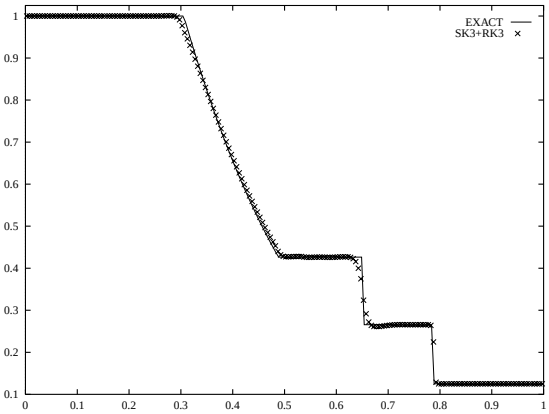

FIG. 5.3. *Sod problem*—density. $N = 200$, $T = 0.1644$.

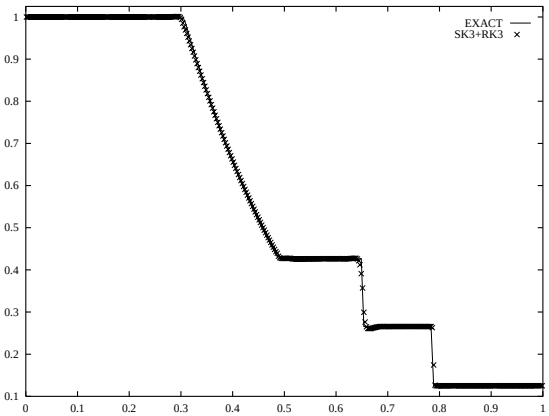

FIG. 5.4. *Sod problem*—density. $N = 400$, $T = 0.1644$.

We would like to stress again that our SD3 scheme does not employ the characteristic decomposition. To improve the resolution of the contact discontinuity, which is always smeared while the solution to the system of Euler equations is computed by the central method, we implemented the artificial compression method (ACM) by Harten [8]. In the context of central schemes, the ACM can be implemented as a corrector step to the componentwise approach (see [30] for details).

Example 4: Convection-diffusion equations—the Buckley–Leverett model. In this example we solve the 1D Buckley–Leverett equation,

$$u_t + f(u)_x = \varepsilon(\nu(u)u_x)_x, \quad \varepsilon\nu(u) \geq 0, \tag{5.3}$$

which can be viewed as a prototype model for the two-phase flow in oil reservoirs. Typically, $\nu(u)$ vanishes at some values of u, and thus (5.3) is a degenerate parabolic equation. Specifically, we take

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}, \quad \nu(u) = 4u(1-u), \quad \varepsilon = 0.01,$$

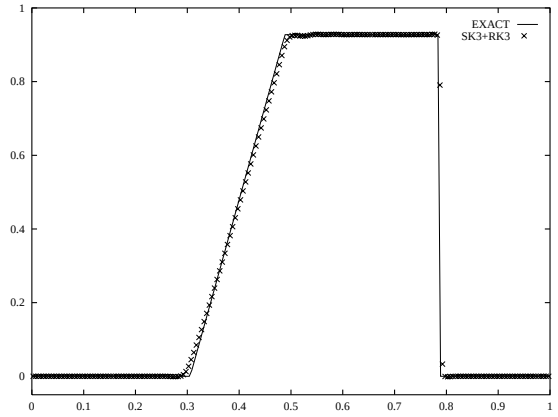

FIG. 5.5. *Sod problem*—velocity. $N = 200$, $T = 0.1644$.

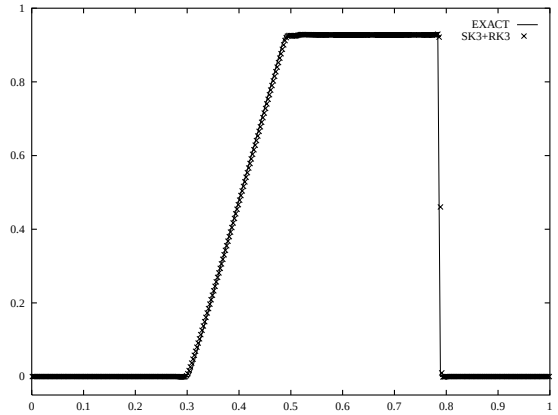

FIG. 5.6. *Sod problem*—velocity. $N = 400$, $T = 0.1644$.

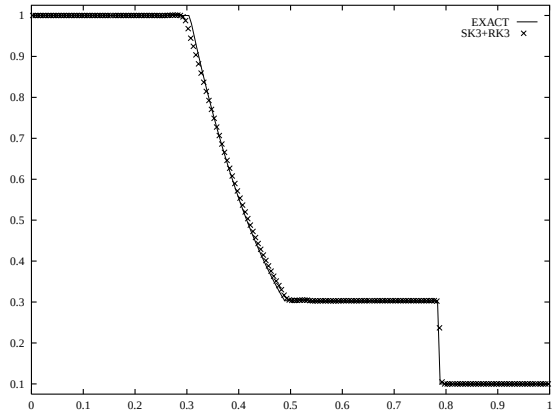

FIG. 5.7. *Sod problem*—pressure. $N = 200$, $T = 0.1644$.

and consider the initial value problem with the Riemann initial data,

$$u(x,0) = \begin{cases} 0, & 0 \le x < 1 - \frac{1}{\sqrt{2}}, \\ 1, & 1 - \frac{1}{\sqrt{2}} \le x \le 1. \end{cases} \tag{5.4}$$

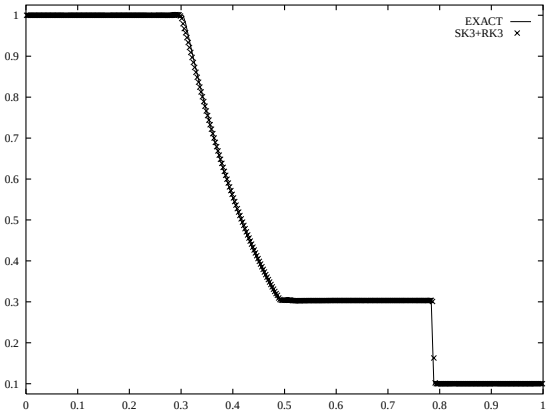

FIG. 5.8. *Sod problem*—pressure. $N = 400$, $T = 0.1644$.

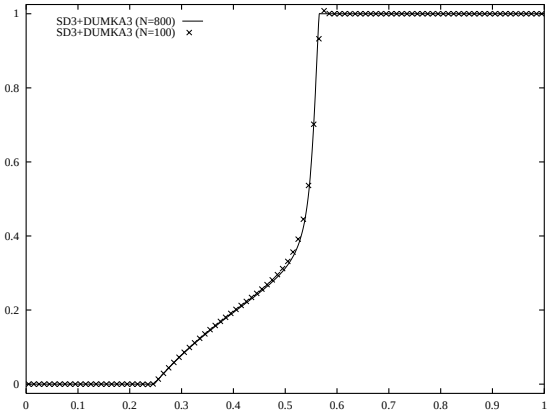

FIG. 5.9. *Buckley–Leverett model,* (5.3)–(5.4). $T = 0.2$.

The numerical solution to this problem, obtained by the SD3 scheme augmented with the DUMKA3 ODE solver, is presented in Figure 5.9.

The model, (5.3), becomes more complicated by adding the effects of gravitation. These effects can be obtained, e.g., by taking

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}(1 - 5(1-u)^2), \tag{5.5}$$

which is nonmonotone on the interval $u \in [0, 1]$.

The numerical solution to this initial value problem is shown in Figure 5.10. Note that the exact solution to problem (5.3)–(5.4) is not available, but our solutions seem to converge to the physically relevant solutions in both cases—with gravitation and without it.

Example 5: Incompressible Euler and Navier–Stokes equations. In this example we consider 2D viscous and inviscid incompressible flow governed by the Navier–Stokes ($\nu > 0$) and Euler ($\nu = 0$) equations,

$$\vec{u}_t + (\vec{u} \cdot \nabla)\vec{u} + \nabla p = \nu \Delta \vec{u}. \tag{5.6}$$

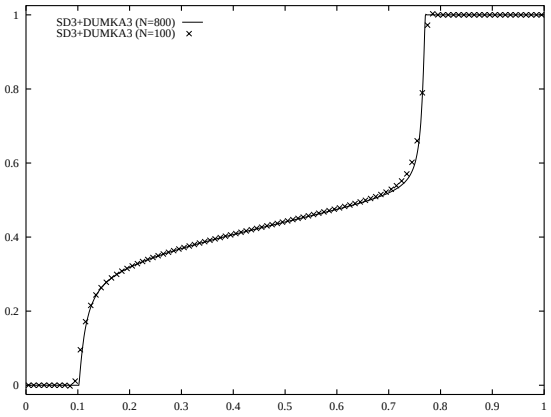

FIG. 5.10. *Buckley–Leverett model,* (5.3)–(5.4)*, including the gravitational effect,* (5.5)*.* $T = 0.2$.

Here, p denotes the pressure, and $\vec{u} = (u, v)$ is the two-component divergence-free velocity field satisfying

$$u_x + v_y = 0. \tag{5.7}$$

In the 2D case (5.6) admits an equivalent scalar formulation in terms of the vorticity,

$$\omega_t + (u\omega)_x + (v\omega)_y = \nu\Delta\omega, \tag{5.8}$$

where $\omega := v_x - u_y$. The incompressibility, (5.7), implies that (5.8) can be written in an equivalent conservative form,

$$\omega_t + f(\omega)_x + g(\omega)_y = \nu\Delta\omega, \tag{5.9}$$

with a *global* convection flux, $(f, g) := (u\omega, v\omega)$. A second-order, fully discrete, staggered, central scheme was used to solve the 2D vorticity equations in [24]. This scheme was proved to satisfy a maximum principle for the vorticity. (For an equivalent scheme in the velocity formulation, see [13].)

When applied to (5.9), our 2D, third-order, semidiscrete scheme, (4.12)–(4.14), takes the form

$$\begin{aligned}\frac{d\bar{\omega}_{j,k}}{dt} = &-\frac{H^x_{j+1/2,k}(t) - H^x_{j-1/2,k}(t)}{\Delta x}\\ &-\frac{H^y_{j,k+1/2}(t) - H^y_{j,k-1/2}(t)}{\Delta y} + \nu Q_{j,k}(t),\end{aligned} \tag{5.10}$$

with the numerical convection fluxes,

$$\begin{aligned}H^x_{j+1/2,k}(t) = &\frac{u_{j+1/2,k}(t)}{2}\left[\omega^+_{j+1/2,k}(t) + \omega^-_{j+1/2,k}(t)\right]\\ &-\frac{a^x_{j+1/2,k}(t)}{2}\left[\omega^+_{j+1/2,k}(t) - \omega^-_{j+1/2,k}(t)\right],\\ H^y_{j,k+1/2}(t) = &\frac{v_{j,k+1/2}(t)}{2}\left[\omega^+_{j,k+1/2}(t) + \omega^-_{j,k+1/2}(t)\right]\\ &-\frac{a^y_{j,k+1/2}(t)}{2}\left[\omega^+_{j,k+1/2}(t) - \omega^-_{j,k+1/2}(t)\right],\end{aligned} \tag{5.11}$$

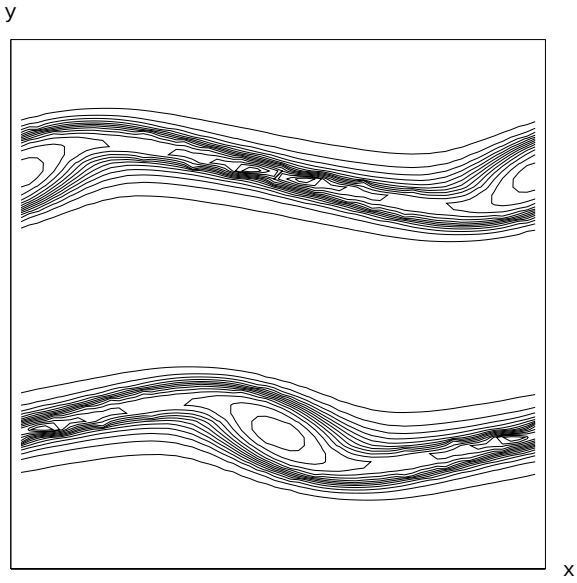

FIG. 5.11. *Incompressible Euler equations; third-order method;* $T = 4$, 64×64 *grid.*

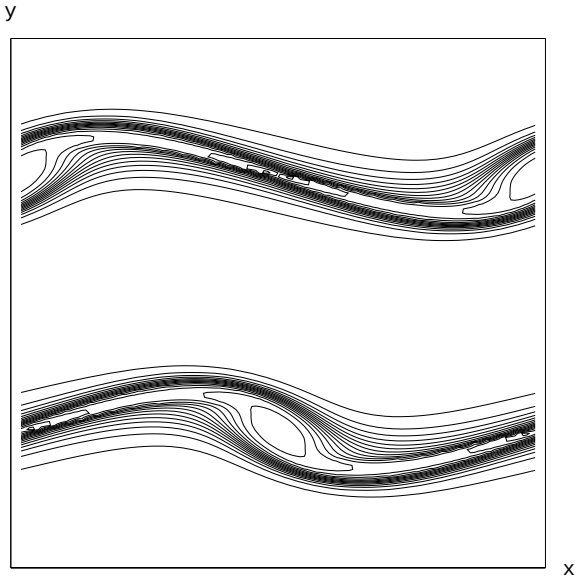

FIG. 5.12. *Incompressible Euler equations; third-order method;* $T = 4$, 128×128 *grid.*

and the local speeds,

$$a^x_{j+1/2,k}(t) := |u_{j+1/2,k}(t)|, \qquad a^y_{j,k+1/2}(t) := |v_{j,k+1/2}(t)|. \tag{5.12}$$

To approximate the linear viscosity, $\Delta\omega$, we used the fourth-order central differencing,

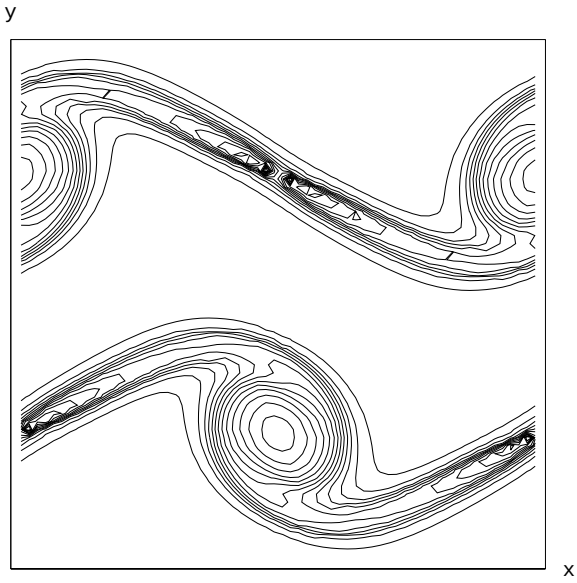

FIG. 5.13. *Incompressible Euler equations; third-order method;* $T = 6$, 64×64 *grid.*

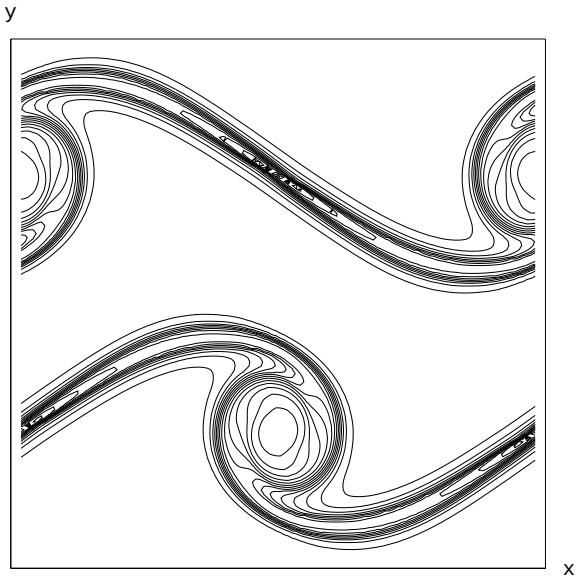

FIG. 5.14. *Incompressible Euler equations; third-order method;* $T = 6$, 128×128 *grid.*

$$Q_{j,k}(t) = \frac{-\omega_{j+2,k}(t) + 16\omega_{j+1,k}(t) - 30\omega_{j,k}(t) + 16\omega_{j-1,k}(t) - \omega_{j-2,k}(t)}{12\Delta x^2} + \frac{-\omega_{j,k+2}(t) + 16\omega_{j,k+1}(t) - 30\omega_{j,k}(t) + 16\omega_{j,k-1}(t) - \omega_{j,k-2}(t)}{12\Delta y^2}. \tag{5.13}$$

To compute the intermediate values of the vorticity, we use the "dimension by dimension" approach described in section 4.2: we reconstruct the corresponding CWENO

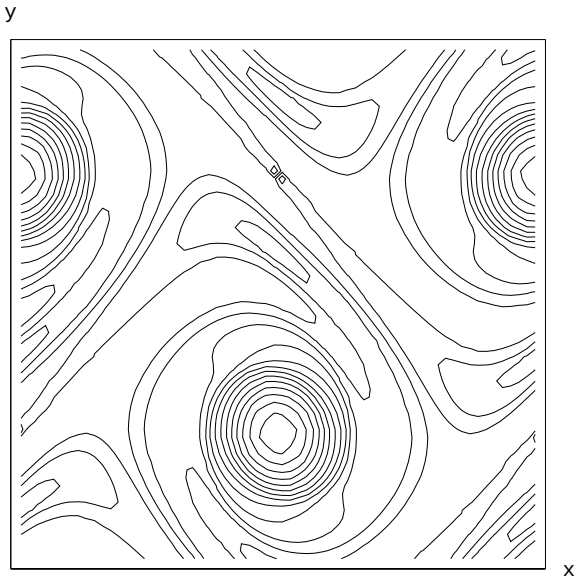

FIG. 5.15. *Incompressible Euler equations; third-order method;* $T = 10$, 64×64 *grid.*

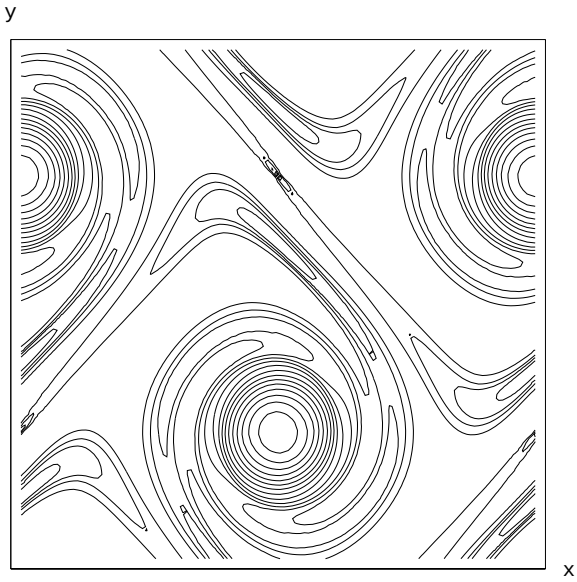

FIG. 5.16. *Incompressible Euler equations; third-order method;* $T = 10$, 128×128 *grid.*

interpolants in the x- and y-directions to obtain the values of $\omega^{\pm}_{j+1/2,k}$ and $\omega^{\pm}_{j,k+1/2}$.

Another important point in the incompressible computations is that in every time-step one has to recover the velocities, $\{u_{j,k}, v_{j,k}\}$, from the known values of the vorticity, $\{\omega_{j,k}\}$. This can be done in many different ways (consult, e.g., [24] and the references therein). Here we have used a stream-function, ψ, such that $\triangle\psi = -\omega$, which is obtained by solving the nine-points Laplacian, $\triangle\psi_{j,k} = -\omega_{j,k}(t)$. This provides the values of the stream-function with fourth-order accuracy. Its gradient,

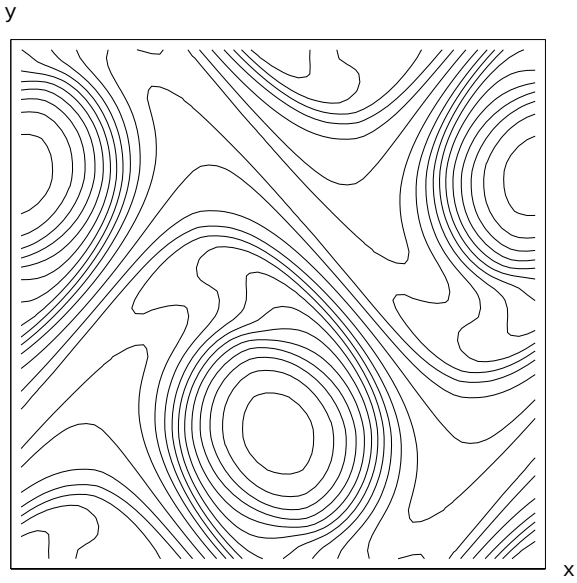

FIG. 5.17. *Incompressible Euler equations; second-order method;* $T = 10$, 64×64 *grid.*

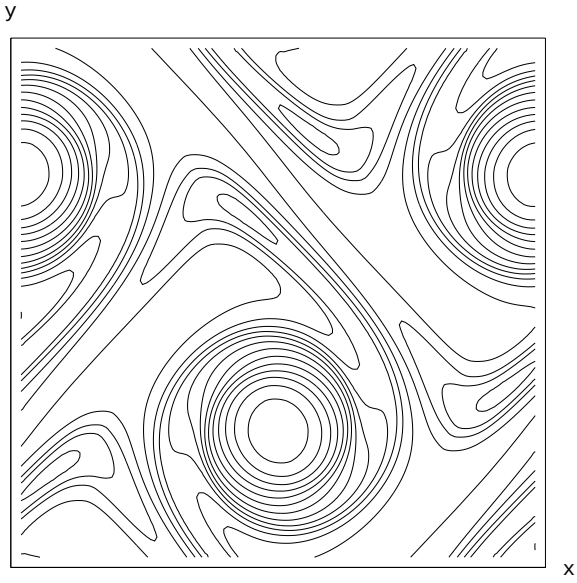

FIG. 5.18. *Incompressible Euler equations; second-order method;* $T = 10$, 128×128 *grid.*

$\nabla\psi$, then recovers the velocity field,

$$u_{j,k}(t) = \frac{-\psi_{j,k+2} + 8\psi_{j,k+1} - 8\psi_{j,k-1} + \psi_{j,k-2}}{12\Delta y},$$
$$v_{j,k}(t) = \frac{\psi_{j+2,k} - 8\psi_{j+1,k} + 8\psi_{j-1,k} - \psi_{j-2,k}}{12\Delta x}. \tag{5.14}$$

Remarks.

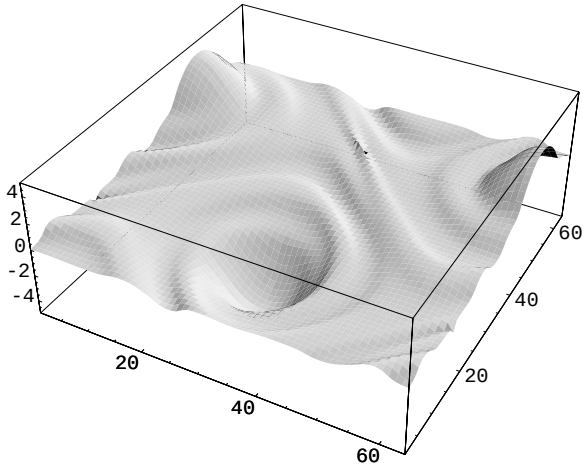

FIG. 5.19. *Incompressible Euler equations; third-order method;* $T = 10$, 64×64 *grid.*

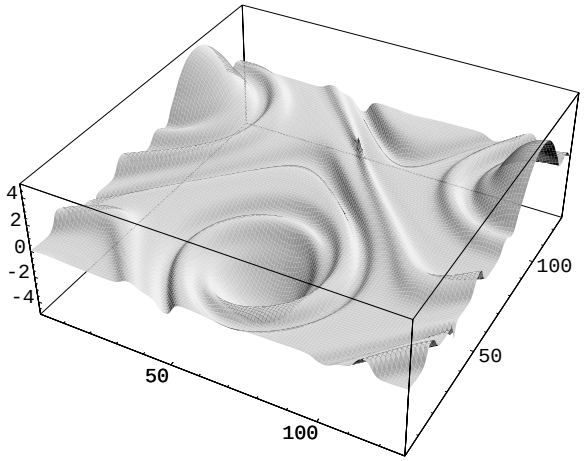

FIG. 5.20. *Incompressible Euler equations; third-order method;* $T = 10$, 128×128 *grid.*

1. Observe that in this way we retain the discrete incompressibility, namely, the discrete velocities computed in (5.14) satisfy

$$\frac{-u_{j+2,k} + 8u_{j+1,k} - 8u_{j-1,k} + u_{j-2,k}}{12\Delta x} + \frac{-v_{j,k+2} + 8v_{j,k+1} - 8v_{j,k-1} + v_{j,k-2}}{12\Delta y} = 0.$$

2. The point-values of the vorticity, which are required for using the nine-points Laplacian, were computed from its cell-averages using the "dimension by dimension" recipe, (4.15)–(4.16).

Finally, the intermediate values of velocities can be computed, e.g., using fourth-order averaging,

$$\begin{aligned} u_{j+1/2,k}(t) &= \frac{-u_{j+2,k}(t) + 9u_{j+1,k}(t) + 9u_{j,k}(t) - u_{j-1,k}(t)}{16}, \\ v_{j,k+1/2}(t) &= \frac{-v_{j,k+2}(t) + 9v_{j,k+1}(t) + 9v_{j,k}(t) - v_{j,k-1}(t)}{16}. \end{aligned} \tag{5.15}$$

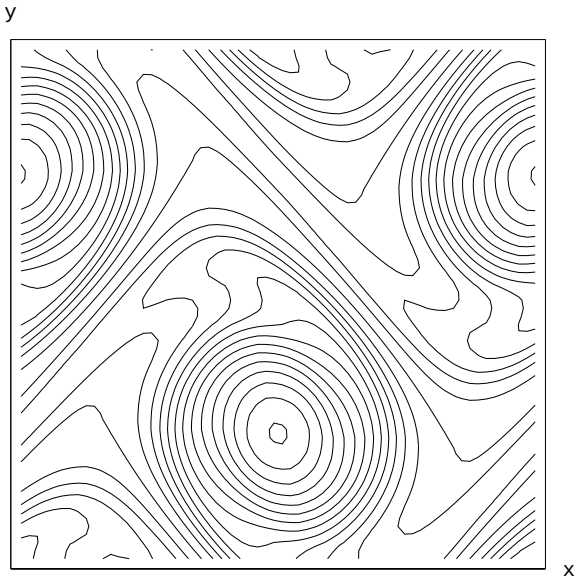

FIG. 5.21. *Incompressible Navier–Stokes equations; third-order method;* $T = 10$, 64×64 *grid.*

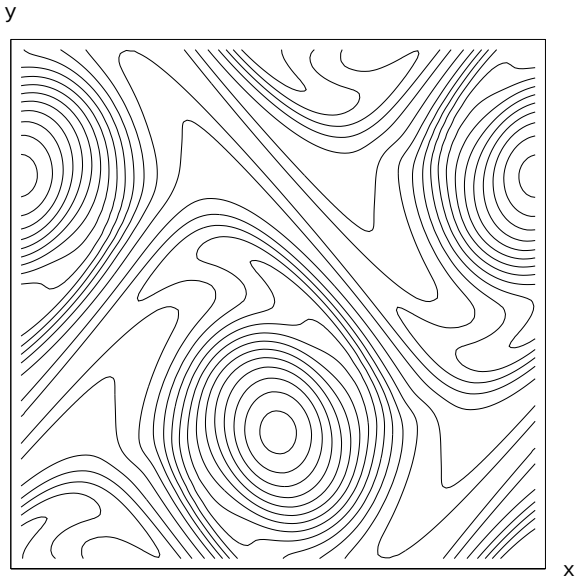

FIG. 5.22. *Incompressible Navier–Stokes equations; third-order method;* $T = 10$, 128×128 *grid.*

We start our numerical experiments by checking the accuracy of our scheme, (5.10)–(5.15), augmented with the DUMKA3 time discretization. We consider the Navier–Stokes equations, (5.6)–(5.7) with $\nu = 0.05$, subject to the smooth periodic initial data (taken from [4]),

$$u(x, y, 0) = -\cos(x)\sin(y), \qquad v(x, y, 0) = \sin(x)\cos(y). \tag{5.16}$$

The exact solution to this problem is simply an exponential decay of the initial data,

TABLE 5.3
Accuracy test for the Navier–Stokes equations. (5.6)–(5.7), (5.16), $\nu = 0.05$. *Errors at* $T = 2$.

$Nx \times Ny$	L^∞-error	Rate	L^1-error	Rate	L^2-error	Rate
32×32	2.429e-02	–	1.791e-01	–	4.559e-02	–
64×64	4.571e-03	2.41	2.814e-02	2.67	7.635e-03	2.58
128×128	8.342e-04	2.45	3.869e-03	2.86	1.146e-03	2.74
256×256	1.208e-04	2.79	4.966e-04	2.96	1.502e-04	2.93

given by

$$u(x,y,t) = -\cos(x)\sin(y)e^{-2\nu t}, \qquad v(x,y,t) = \sin(x)\cos(y)e^{-2\nu t}.$$

The approximate solution with a different number of grid points was computed at time $t = 2$. The errors, measured in terms of vorticity in the L^∞-, L^1-, and L^2-norms are shown in Table 5.3.

Next, the third-order semidiscrete scheme, (5.10)–(5.15), was implemented for the periodic double shear-layer model problem taken from [2]. First, we solve the Euler equations, (5.6)–(5.7) with $\nu = 0$, subject to the $(2\pi, 2\pi)$-periodic initial data,

$$u(x,y,0) = \begin{cases} \tanh(\frac{1}{\rho}(y - \pi/2)), & y \le \pi, \\ \tanh(\frac{1}{\rho}(3\pi/2 - y)), & y > \pi, \end{cases} \qquad v(x,y,0) = \delta \cdot \sin(x). \tag{5.17}$$

Here, the "thick" shear-layer width parameter, ρ, is taken as $\frac{\pi}{15}$ and the perturbation parameter $\delta = 0.05$.

The numerical results at times $T = 4, 6, 10$ with $N = 64 \times 64$ and $N = 128 \times 128$ grid points are presented in Figures 5.11 through 5.16 and 5.19 through 5.20. In order to compare the quality of the results obtained with our new method to previous results, we plot in Figures 5.17 and 5.18 the results obtained for the same double shear-layer problem with the second-order central scheme proposed in [24]. Compared with the second-order method, the new third-order method can resolve the large gradients better. Since we are using only an essentially nonoscillatory reconstruction, some oscillations are created with the third-order method (and not with the "fully" nonoscillatory second-order method).

Finally, we solve the Navier–Stokes equations, (5.6)–(5.7) with $\nu = 0.01$, augmented with the "thick" shear-layer periodic initial data, (5.17).

The numerical results at time $T = 10$ with $N = 64 \times 64$ and $N = 128 \times 128$ grid points are presented in Figures 5.21–5.24.

Acknowledgment. The authors would like to thank Professors S. Karni and R. S. Krasny for helpful comments.

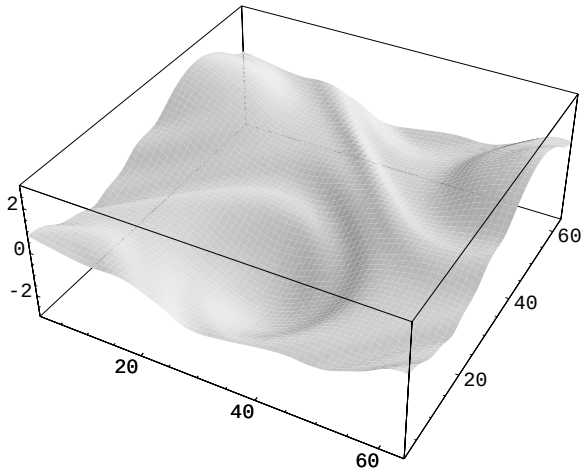

FIG. 5.23. *Incompressible Navier–Stokes equations; third-order method;* $T = 10$, 64×64 *grid.*

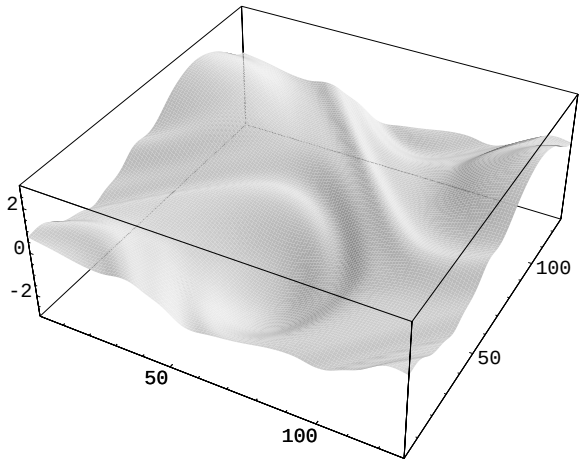

FIG. 5.24. *Incompressible Navier–Stokes equations; third-order method;* $T = 10$, 128×128 *grid.*

REFERENCES

[1] P. ARMINJON AND M. C. VIALLON, *Généralisation du Schéma de Nessyahu-Tadmor pour Une Équation Hyperbolique à Deux Dimensions D'espace*, C.R. Acad. Sci. Paris, Ser. I Math., 320 (1995), pp. 85–88.

[2] J. B. BELL, P. COLELLA, AND H. M. GLAZ, *A second-order projection method for the incompressible Navier–Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.

[3] F. BIANCO, G. PUPPO, AND G. RUSSO, *High-order central schemes for hyperbolic systems of conservation laws*, SIAM J. Sci. Comput., 21 (1999), pp. 294–322.

[4] A. J. CHORIN, *Numerical solution of the Navier–Stokes equations*, Math. Comp., 22 (1968), pp. 745–762.

[5] K. O. FRIEDRICHS AND P. D. LAX, *Systems of conservation equations with a convex extension*, Proc. Nat. Acad. Sci. U.S.A, 68 (1971), pp. 1686–1688.

[6] E. GODLEWSKI AND P. A. RAVIART, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer, New York, 1996.

[7] S. K. GODUNOV, *A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics*, Mat. Sb., 47 (1959), pp. 271–290.

[8] A. Harten, *The artificial compression method for computation of shocks and contact discontinuities,* III. *Self-adjusting hybrid schemes*, Math. Comp., 32 (1978), pp. 363–389.
[9] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes* III, J. Comput. Phys., 71 (1987), pp. 231–303.
[10] G.-S. Jiang, D. Levy, C.-T. Lin, S. Osher, and E. Tadmor, *High-resolution nonoscillatory central schemes with nonstaggered grids for hyperbolic conservation laws*, SIAM J. Numer. Anal., 35 (1998), pp. 2147–2168.
[11] G. S. Jiang and C. W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
[12] G.-S. Jiang and E. Tadmor, *Nonoscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., 19 (1998), pp. 1892–1917.
[13] R. Kupferman and E. Tadmor, *A fast high-resolution second-order central scheme for incompressible flows*, Proc. Nat. Acad. Sci. U.S.A, 94 (1997), pp. 4848-4852.
[14] A. Kurganov, D. Levy, and P. Rosenau, *On Burgers-type equations with nonmonotonic dissipative fluxes*, Comm. Pure Appl. Math., 51 (1998), pp. 443–473.
[15] A. Kurganov and P. Rosenau, *Effects of a saturating dissipation in Burgers-type equations*, Comm. Pure Appl. Math., 50 (1997), pp. 753–771.
[16] A. Kurganov and E. Tadmor, *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282.
[17] B. van Leer, *Towards the ultimate conservative difference scheme,* V. *A second order sequel to Godunov's method*, J. Comput. Phys., 32 (1979), pp. 101–136.
[18] D. Levy, *A third-order* 2D *central scheme for conservation laws*, in Systém hyperboliques: Nouveaux schémas et nouvelles applications, Vol. 1, Lecture Notes of Écoles CEA-EDF-INRIA, INRIA Rochquencourt, France, 1998, pp. 489–504.
[19] D. Levy, G. Puppo, and G. Russo, *Central WENO schemes for hyperbolic systems of conservation laws*, Math. Model. Numer. Anal., 33 (1999), pp. 547–571.
[20] D. Levy, G. Puppo, and G. Russo, *A third order central WENO scheme for* 2D *conservation laws*, Appl. Numer. Math., 33 (2000), pp. 407–414.
[21] D. Levy, G. Puppo, and G. Russo, *Compact central WENO schemes for multidimensional conservation laws*, SIAM J. Sci. Comput., 22 (2000), pp. 656–672.
[22] D. Levy, G. Puppo, and G. Russo, *Central WENO schemes for multi-dimensional hyperbolic systems of conservation laws*, submitted.
[23] D. Levy, G. Puppo, and G. Russo, *On the behavior of the total variation in CWENO methods for conservation laws*, Appl. Numer. Math., 33 (2000), pp. 415–421.
[24] D. Levy and E. Tadmor, *Nonoscillatory central schemes for the incompressible* 2*-D Euler equations*, Math. Res. Lett., 4 (1997), pp. 1–20.
[25] X.-D. Liu and S. Osher, *Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes* I, SIAM J. Numer. Anal., 33 (1996), pp. 760–779.
[26] X. D. Liu and S. Osher, *Convex ENO high order multi-dimensional schemes without field by field decomposition or staggered grids*, J. Comput. Phys., 142 (1998) pp. 304–330.
[27] X. D. Liu, S. Osher, and T. Chan, *Weighted essentially nonoscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
[28] X. D. Liu and E. Tadmor, *Third order nonoscillatory central scheme for hyperbolic conservation laws*, Numer. Math., 79 (1998), pp. 397–425.
[29] A. A. Medovikov, *High order explicit methods for parabolic equations*, BIT, 38 (1998), pp. 372–390.
[30] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 408–463.
[31] P. L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, J. Comput. Phys., 43 (1981), pp. 357–372.
[32] C. W. Shu, *Numerical experiments on the accuracy of ENO and modified ENO schemes*, J. Sci. Comput., 5 (1990), pp. 127–149.
[33] C. W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[34] G. Sod, *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*, J. Comput. Phys., 22 (1978), pp. 1–31.
[35] E. Tadmor, *Approximate solutions of nonlinear conservation laws*, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Math. 1697, A. Quarteroni, ed., Springer, Berlin, 1998, pp. 1–149.