SMAI-JCM
SMAI Journal of Computational Mathematics

Maximum-principle-satisfying second-order Intrusive Polynomial Moment scheme

Jonas Kusch, Graham W. Alldredge & Martin Frank

Volume 5 (2019), p. 23-51.
<http://smai-jcm.cedram.org/item?id=SMAI-JCM_2019__5__23_0>

© Société de Mathématiques Appliquées et Industrielles, 2019
Certains droits réservés.

cedram
Article mis en ligne dans le cadre du
Centre de diffusion des revues académiques de mathématiques
http://www.cedram.org/

SMAI Journal of Computational Mathematics
Vol. 5, 23-51 (2019)

Maximum-principle-satisfying second-order Intrusive Polynomial Moment scheme

Jonas Kusch [1]
Graham W. Alldredge [2]
Martin Frank [3]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
E-mail address: jonas.kusch@kit.edu
[2] FU Berlin, Berlin, Germany
E-mail address: graham.alldredge@fu-berlin.de
[3] Karlsruhe Institute of Technology, Karlsruhe, Germany
E-mail address: martin.frank@kit.edu.

Abstract. Using standard intrusive techniques when solving hyperbolic conservation laws with uncertainties can lead to oscillatory solutions as well as nonhyperbolic moment systems. The Intrusive Polynomial Moment (IPM) method ensures hyperbolicity of the moment system while restricting oscillatory over- and undershoots to specified bounds. In this contribution, we derive a second-order discretization of the IPM moment system which fulfills the maximum principle. This task is carried out by investigating violations of the specified bounds due to the errors from the numerical optimization required by the scheme. This analysis gives weaker conditions on the entropy that is used, allowing the choice of an entropy which enables choosing the exact minimal and maximal value of the initial condition as bounds. Solutions calculated with the derived scheme are nonoscillatory while fulfilling the maximum principle. The second-order accuracy of our scheme leads to significantly reduced numerical costs.

2010 Mathematics Subject Classification. 35L65, 35R60, 65M08.

Keywords. uncertainty quantification, conservation laws, maximum principle, moment system, hyperbolic, oscillations.

1. Introduction

Hyperbolic conservation laws play an important role in modeling various physical and engineering problems. Examples include the shallow water equations in hydrology as well as the Euler equations in gas dynamics. Finite-volume schemes, which are perhaps the most popular numerical methods for hyperbolic problems, are initially designed for the scalar hyperbolic conservation law,

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

because the solution theory of this problem is well established. The conservation law (1.1) is generally supplemented with initial conditions

$$u(t=0,x) = u_0(x) \tag{1.2}$$

as well as boundary conditions, though the latter do not play a role in this work.

We wish to determine the solution u which depends on the spatial variable x $D = \mathrm{R}$ and time t R^+. The function $f : \mathrm{R} \to \mathrm{R}$ is the system flux. Since u can become discontinuous even for smooth initial conditions u_0, the solution must be seen in the weak sense. To ensure uniqueness, an entropy condition is imposed to pick the physically meaningful weak solution [19, Chapter 3.8.1]. An important

This work was supported by the German Research Foundation (DFG). Jonas Kusch and Martin Frank were supported under grant FR 2841/6-1 and Graham Alldredge under AL 2030/1-1.

property of such an entropy solution is the maximum principle (see [17, Chapter 2.4]), which states that

$$\min_{x \in D} u_0(x) \leq u(t,x) \leq \max_{x \in D} u_0(x)$$

for all t and x. Finite-volume schemes are carefully constructed to satisfy this property on a discrete level, see for example [3, 7, 23, 33, 14].

In many practical applications the model parameters and initial conditions are not deterministic, and classical finite-volume methods do not take this into account. One popular approach for uncertain partial differential equations is the stochastic-Galerkin (SG) method [11]. It is based on polynomial chaos [30] and promises pseudo-spectral convergence for smooth data [4]. The key idea is to parameterize the uncertainty with the help of a random variable $\xi \in \Theta \subset \mathbb{R}^P$ and span the solution with the help of orthonormal polynomials $\phi_i : \Theta \to \mathbb{R}$. In the following, we assume a scalar random variable, i.e. $P = 1$ and $\xi \in \Theta \subset \mathbb{R}$. The solution is then approximated by the closure $U_{SG} : \mathbb{R}^{N+1} \to \mathbb{P}(\Theta)$ given by

$$u(t,x,\xi) \approx U_{SG}(u(t,x))(\xi) = \sum_{i=0}^{N} u_i(t,x)\phi_i(\xi).$$

The stochastic-Galerkin ansatz leads to a coupled deterministic system of equations for the expansion coefficients $u \in \mathbb{R}^{N+1}$ with $u = (u_0, \dots, u_N)^T$ (which also correspond to moments of the solution)[1]. Simple applications, such as the steady diffusion equation [31] or the advection equation [12] show the expected spectral convergence. However, the solutions to hyperbolic problems are generally non-smooth, and thus the SG method converges slowly and exhibits the oscillations of Gibbs phenomenon. In addition to that, the stochastic-Galerkin solution can violate the maximum principle, leading to unphysical solutions. In the case of systems the SG equations may not be hyperbolic, making it impossible to solve with standard methods [8].

The Intrusive Polynomial Moment (IPM) method [25, 26, 8] is designed to preserve hyperbolicity and is constructed to bound oscillations. The IPM approach is to replace the stochastic-Galerkin ansatz with one derived from a minimum-entropy principle. The IPM ansatz has the form

$$u(t,x,\xi) \approx (s')^{-1}\left(\sum_{i=0}^{N} \lambda_i(u(t,x))\phi_i(\xi)\right),$$

where $(s')^{-1}$ is the inverse function of the derivative of a strictly convex entropy density $s : \mathbb{R} \to \mathbb{R}$ and λ_i are the expansion coefficients which need to be chosen to match moment constraints. Unlike the SG method, in an IPM method the expansion coefficients of the ansatz do not correspond to the moments of the ansatz, which above we have collected into the vector u. Minimum-entropy methods have been used in kinetic theory, where they are sometimes called M_N methods, see for example [21, 9, 15, 16]. The IPM method has a few key advantages. First, for a scalar conservation law such as (1.1), the entropy density can be chosen such that the solution only takes values in (u_-, u_+). The interval (u_-, u_+) can be chosen by the user to, e.g., enforce a maximum principle for the discrete solution. These bounds on the solution also restrict the under- and overshoots of oscillations. Other attractive properties possessed by the IPM system are hyperbolicity and entropy dissipation. However, these nice properties of the IPM method do come at certain costs and challenges. The main cost is computational, since an optimization problem must be numerically solved in every spatial cell at each time step. In

[1] To clarify notation, we use bold letters to indicate vectors as well as functions, which map onto vectors. Note that from now on, all vectors will have dimension $N + 1$. Scalar products between two vectors $x, y \in \mathbb{R}^{N+1}$ will be denoted by $x^T y := \sum_{i=0}^{N} x_i y_i$.

order to reduce these costs, one should take advantage of both the parallelizability of the method [10] and high-order numerical schemes for the resulting moment equations.

One of the main challenges facing the IPM method is that the design of a high-order numerical scheme is more complicated. Unlike the SG method, the moments u of the numerical solution must stay within a certain set, called the realizable set, to ensure that the IPM ansatz can be reconstructed.

Another challenge with IPM methods is that while they successfully dampen oscillations near the bounds u_- and u_+, the solutions can still oscillate heavily between these bounds.

We tackle these two challenges in this paper. After reviewing the derivation of the IPM method in section 2, we give a naïve, out-of-the-box numerical method for the IPM equations in section 3 to demonstrate the problem of maintaining realizability. Next, in section 4, we begin to address the problem of numerically maintaining realizability with a first-order scheme through either a time-step restriction or modification of the numerical method. In section 5 we extend these results to a second-order scheme. In section 6, we discuss properties of the minimum-entropy approximation, and study an entropy which leads to smaller oscillations in the solution. Section 7 presents numerical results for the uncertain Burgers' and advection equations. Finally in section 8, we summarize our findings and give an outlook on future work.

2. Stochastic Galerkin and IPM

In this section, we recall the derivations of the stochastic-Galerkin and Intrusive Polynomial Moment systems. Our derivation is carried out for the scalar hyperbolic equation with uncertain initial condition

$$\partial_t u(t,x,\omega) + \partial_x f(u(t,x,\omega)) = 0, \tag{2.1a}$$

$$u(0,x,\omega) = u_0(x,\omega). \tag{2.1b}$$

Here, $x \in D = \mathbb{R}$ is the spatial domain, $t \in \mathbb{R}^+$ is time and $f : \mathbb{R} \to \mathbb{R}$ is the flux[2].

The stochastic variable is $\omega \in \Omega$, where Ω is the set of all possible outcomes of a random experiment. The probability measure $dP(\omega)$ with $\int_\Omega dP(\omega) = 1$ imposes a weighting of different events ω. Assuming the solution is a second-order random field, we can make use of the generalized polynomial chaos (gPC) approach, which lets us represent the random solution with the help of a random variable $\xi \in \Theta$, which has the probability distribution function $f_\Xi \in L^1(\Theta)$. In the following we abuse notation by writing $u = u(t,x,\xi(\omega))$, and we drop the dependency on ω. From the theory of scalar, hyperbolic problems, we know that the solution satisfies a maximum principle [17, Chapter 2.4]

$$\min_{x \in D, \xi \in \Theta} u_0(x,\xi) \le u(t,x,\xi) \le \max_{x \in D, \xi \in \Theta} u_0(x,\xi)$$

for all $t \in \mathbb{R}^+$. Furthermore, for a fixed ξ, any strictly convex function $s : \mathbb{R} \to \mathbb{R}$ is an entropy to this problem, meaning that there exists an entropy flux $h : \mathbb{R} \to \mathbb{R}$ satisfying $h'(u) = s'(u) f'(u)$ such that

$$\partial_t s(u) + \partial_x h(u) = 0$$

for strong solutions. We wish to determine how the uncertainty of the initial condition propagates through the solution over time. In order to derive methods such as stochastic Galerkin, we multiply (2.1) with the basis function ϕ_i and the probability distribution function f_Ξ and integrate with respect to ξ over Θ. The basis functions are chosen to be orthonormal with respect to f_Ξ. To simplify the notation, we introduce the bracket operator

$$\langle g \rangle := \int_\Theta g(\xi) f_\Xi(\xi) d\xi.$$

[2] The flux f may also depend directly on ω, but for now we suppress this from the notation for clarity

The resulting system is then

$$\partial_t \, u(t,x,\cdot)\phi_i \; + \partial_x \; f(u(t,x,\cdot))\phi_i \; = 0, \tag{2.2a}$$

$$u(0,x,\cdot)\phi_i \; = \; u_0(x,\cdot)\phi_i \; . \tag{2.2b}$$

Since the basis functions are orthonormal, the moments $u\phi_i$ can be interpreted as Fourier coefficients. Provided the solution is sufficiently smooth, these coefficients fall to zero rapidly for increasing order i, so the first moments should suffice for a good approximation. Furthermore, the lower-order moments give the most familiar quantities such as the mean and variance. This motivates using only the first $N+1$ moments to define a discretization of the true solution u,

$$u_i(t,x) := \; u(t,x,\cdot)\phi_i \quad \text{for} \quad i = 0, \cdots, N;$$

we collect these moments and the basis functions into the vectors $u = (u_0, \cdots, u_N)^T$ and $\phi = (\phi_0, \cdots, \phi_N)^T$, respectively. The main problem is to find a good ansatz $U : \mathbb{R}^{N+1} \to L^1(\Theta)$ approximating the solution[3] u, which allows us to write (2.2) as a closed system of equations for the moments u. For the stochastic-Galerkin method, the ansatz is given by

$$U_{SG}(u(t,x))(\xi) = \sum_{i=0}^{N} u_i(t,x)\phi_i(\xi) = u(t,x)^T \phi(\xi).$$

(Note that in kinetic theory, this corresponds to the well-known P_N closure, see for example [6, 5, 22, 28].) In the vector notation the resulting stochastic-Galerkin system is written as

$$\partial_t u + \partial_x \; f(u^T\phi)\phi \; = 0,$$

$$u(0,x) = \; u_0(x,\cdot)\phi \; .$$

The SG system is attractive because it is relatively cheap to simulate and U_{SG} converges pseudo-spectrally to the correct solution u for smooth problems. However, the main drawback is that it exhibits the Gibbs phenomenon, i.e., the solution oscillates heavily for nonsmooth problems. Also, for systems of hyperbolic equations, the resulting SG system might no longer be hyperbolic, and thus ill-posed. An example of a classical SG solution for a hyperbolic problem can be found in Figure 2.1.

The IPM method, which was introduced in [25], is constructed to overcome these problems. The idea of the IPM method is to choose the ansatz $U_{ME} : \mathbb{R}^{N+1} \to L^1(\Theta)$ that minimizes the convex entropy $s(u)$ under the moment constraints $u = \; u\phi$, i.e.,

$$U_{ME}(u) = \underset{u \; L^1(\Theta)}{\arg\min} \; s(u) \quad \text{subject to } u = \; u\phi \; . \tag{2.4}$$

This problem has the unconstrained finite-dimensional dual problem

$$\hat{\lambda}(u) := \underset{\lambda \; \mathbb{R}^{N+1}}{\arg\min} \; \big[\; s \; (\lambda^T\phi) \; - \lambda^T u \; \big], \tag{2.5}$$

where $s \; : \mathbb{R} \to \mathbb{R}$ is the Legendre transformation of s, and $\hat{\lambda} : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ are called the dual variables. The solution to the primal problem (2.4) is given by

$$U_{ME}(u) = \big[s \; \big]^{-1} (\hat{\lambda}(u)^T\phi) = s \; (\hat{\lambda}(u)^T\phi). \tag{2.6}$$

We also use the notation

$$U_{ME}(u) =: u_{ME}(\hat{\Lambda}(u)) \tag{2.7}$$

[3] We assume that Θ is compact, in which case the ansatz must lie in $L^1(\Theta)$ to ensure the existence of a moment vector.

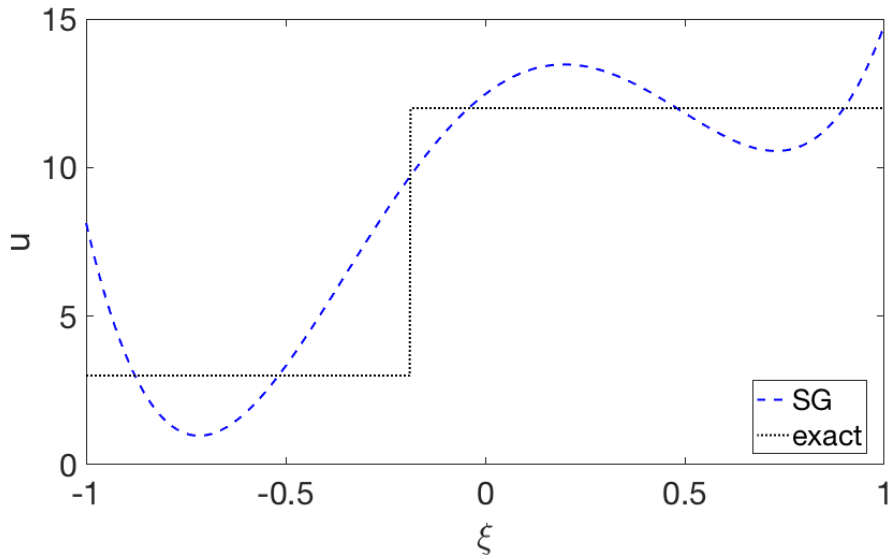


Figure 2.1. Approximation result for fixed x and t using Burgers' equation (see section 7.1). The SG approximation oscillates heavily and violates the maximum principle.

for the ansatz, where $u_{ME} := (s')^{-1}$, and $\hat{\Lambda} : \mathbb{R}^{N+1} \to \mathbb{P}(\Theta)$ with $\hat{\Lambda}(u) = \hat{\lambda}(u)^T \phi$ is what we call the dual state. Inserting this into (2.2) leads to the closed moment system

$$\partial_t u + \partial_x \langle f(U_{ME}(u))\phi \rangle = 0, \tag{2.8a}$$

$$u(0,x) = \langle u_0(x,\cdot)\phi \rangle . \tag{2.8b}$$

This is the system of equations of the IPM method.

The IPM system (2.8) has nice features. First, it generalizes the stochastic-Galerkin method, in that the SG method can be recovered with the quadratic entropy $s(u) = \frac{1}{2}u^2$. Second, it is hyperbolic for any strictly convex s. Its solutions also satisfy the entropy-dissipation law

$$\frac{d}{dt}S(t) := \frac{d}{dt}\int_D s(U_{ME}(u(t,x)))\, dx \le 0,$$

see for example [25, 18]. Also, with the IPM method one can design the entropy so that the entropy ansatz $U_{ME}(u) = u_{ME}(\hat{\Lambda}(u))$ only takes values within a specified interval. This gives a guaranteed bound on the magnitude of oscillations. In [25] the log-barrier entropy density

$$s(u) = -\ln(u - u_-) - \ln(u_+ - u) \tag{2.9}$$

is used, where the scalars u_- and u_+, $u_- < u_+$, are user-specified parameters. Clearly this entropy does not allow an ansatz which takes on values outside the interval (u_-, u_+). Since we know that the solution should be bounded by

$$u_{\min} := \min_{x,\xi} u_0(x,\xi) \quad \text{and} \quad u_{\max} := \max_{x,\xi} u_0(x,\xi),$$

one can take $u_+ := u_{\max} + \Delta u$ and $u_- := u_{\min} - \Delta u$ with $\Delta u \quad [0, \infty)$.

When the solution to the primal problem (2.4) can only take values in (u_-, u_+), the problem is only feasible if the moment vector u lies in the set

$$R := \{ u \quad \mathbb{R}^{N+1} \mid u : \Theta \to (u_-, u_+) \text{ such that } u = \langle u\phi \rangle \}. \tag{2.10}$$

We call $R \subset \mathbb{R}^{N+1}$ the realizable set. This is important to keep in mind when designing numerical methods, because when the numerical solution leaves the realizable set R, the ansatz u_{ME} is undefined, and so the IPM method crashes. We consider this in the next two sections.

3. Discretization of the IPM system

The IPM system (2.8) can be rewritten as

$$\partial_t u + \partial_x F(u) = 0$$

with the flux $F : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$, $F(u) = \int f(u_{ME}(\hat{\Lambda}(u)))\phi$ depending on the dual state

$$\hat{\Lambda}(u) = \hat{\lambda}(u)^T \phi.$$

For efficiency of exposition, we sometimes omit the dependence on u. The IPM system is hyperbolic, so it is naturally solved by a finite-volume method. First we discretize the spatial domain into cells. The discrete unknowns are chosen to be the spatial averages over each cell at time t_n, given by

$$u_{ij}^n \approx \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u_i(t_n, x) dx.$$

If a moment vector in cell j at time t_n is denoted as $u_j^n = (u_{0j}^n, \cdots, u_{Nj}^n)^T \in \mathbb{R}^{N+1}$, the finite-volume scheme can be written in conservative form with the numerical flux $G : \mathbb{R}^{N+1} \times \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ as

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left(G(u_j^n, u_{j+1}^n) - G(u_{j-1}^n, u_j^n) \right) \tag{3.1}$$

for $j = 1, \cdots, N_x$ and $n = 0, \cdots, N_t$, where N_x is the number of spatial cells and N_t is the number of time steps. The numerical flux is assumed to be consistent, i.e., that $G(u, u) = F(u)$. To ensure stability, a CFL condition has to be derived by investigating the eigenvalues of $\partial_u F$.

When a consistent numerical flux $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $g = g(u_l, u_r)$ is available for the deterministic problem (2.1), then for the IPM system we can simply take

$$G(u_j^n, u_{j+1}^n) = \int g(u_{ME}(\hat{\Lambda}(u_j^n)), u_{ME}(\hat{\Lambda}(u_{j+1}^n)))\phi \, .$$

This choice of the numerical flux is a common choice in kinetic theory and is called kinetic flux. The time update of the moment vector now becomes

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left(\int g(u_{ME}(\hat{\Lambda}_j^n), u_{ME}(\hat{\Lambda}_{j+1}^n))\phi - \int g(u_{ME}(\hat{\Lambda}_{j-1}^n), u_{ME}(\hat{\Lambda}_j^n))\phi \right), \tag{3.2}$$

where $\hat{\Lambda}_j^n := \hat{\Lambda}(u_j^n)$ for all j. Note that the computation of $\hat{\Lambda}_j^n$ requires solving the dual problem (2.5) for the moment vector u_j^n.

Unfortunately (3.2) cannot be implemented because the dual problem cannot be solved exactly. [4] Instead, it must be solved numerically, for example with Newton's method. The stopping criterion for the numerical optimizer ensures that the approximate multiplier vector it returns, which we denote $\bar{\lambda}_j^n \in \mathbb{R}^{N+1}$ for the moment vector u_j^n, satisfies the stopping criterion

$$\left| u_j^n - \int u_{ME}(\bar{\lambda}_j^n)^T \phi \, \phi \right| < \tau. \tag{3.3}$$

This is derived from the first-order necessary conditions for the dual problem. Once the numerical optimizer finds such a $\bar{\lambda}_j^n$, the corresponding dual state $\bar{\Lambda}_j^n := \bar{\lambda}_j^{nT} \phi \in P(\Theta)$ can be used in (3.2) for the unknown $\hat{\Lambda}_j^n$. This gives Algorithm 1.

[4] Equation (3.2) also includes integral evaluations which cannot be computed in closed form. Their approximation by numerical quadrature, however, does not play a role in the realizability problems we discuss below.

Algorithm 1 IPM for Uncertainty Quantification

for $j = 0$ to $N_x + 1$ **do**
  $u_j^0 = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u_0(x, \cdot)\boldsymbol{\phi}\, dx$
end for
for $n = 0$ to N_t **do**
  for $j = 0$ to $N_x + 1$ **do**
    $\bar{\boldsymbol{\lambda}}_j^n \approx \arg\min_{\boldsymbol{\lambda}} \left(s\left(\boldsymbol{\lambda}^T \boldsymbol{\phi}\right) - \boldsymbol{\lambda}^T \boldsymbol{u}_j^n \right)$ such that (3.3) holds
    $\bar{\Lambda}_j^n = \left(\bar{\boldsymbol{\lambda}}_j^n\right)^T \boldsymbol{\phi}$
  end for
  for $j = 1$ to N_x **do**
    $\boldsymbol{u}_j^{n+1} = \boldsymbol{u}_j^n - \frac{\Delta t}{\Delta x} \left(g(u_{ME}(\bar{\Lambda}_j^n), u_{ME}(\bar{\Lambda}_{j+1}^n))\boldsymbol{\phi} - g(u_{ME}(\bar{\Lambda}_{j-1}^n), u_{ME}(\bar{\Lambda}_j^n))\boldsymbol{\phi} \right)$
  end for
end for

For most test cases in this paper, Dirichlet boundary conditions are used, i.e. ghost cells with moment vectors $\boldsymbol{u}_0^n = \boldsymbol{u}_L \boldsymbol{\phi}$ and $\boldsymbol{u}_{N_x+1}^n = \boldsymbol{u}_R \boldsymbol{\phi}$ are implemented. Algorithm 1 crashes when the numerical optimizer cannot find a moment vector $\bar{\boldsymbol{\lambda}}_j^n$ satisfying the stopping criterion. This can only [5] happen when the moment vector $\boldsymbol{u}_j^n$ is not realizable. We tested an implementation of Algorithm 1 on the uncertain Burgers' equation as described in section 7.1. We chose the initial condition given in (7.2), and ran simulations with different values of the optimization tolerance τ and the solution-bound parameter Δu. We chose the time step Δt according to the classical time-step restriction

$$\frac{\Delta t}{\Delta x} \max_{u \in [u_-, u_+]} |f'(u)| \leq 1. \tag{3.4}$$

The solution bounds u_- and u_+, which parametrize the entropy (2.9), are important parameters in the implementation. Thus one would like to choose Δu as small as possible. Furthermore, since the maximum velocity $\max\{f'(u)\}$ is determined over the interval $u \in [u_-, u_+]$, the larger we take Δu, the larger the maximum velocity may be. A larger maximum velocity would lead the CFL condition to impose a tighter time-step restriction and add numerical viscosity. In [25] the authors chose $u_+ = u_{\max} + \Delta u$ and $u_- = u_{\min} - \Delta u$ with $\Delta u = 0.5$. Consequently, over- and undershoots as large as 0.5 are allowed, and we test a few values here. We chose all other parameters in the experiments as given in section 7.1.

In Table 3.1, for different values of the optimization tolerance τ and the entropy parameter Δu we report how long Algorithm 1 ran until it crashed due to loss of realizability. The results indicate that this is more likely for smaller values of Δu, while decreasing the optimization tolerance seems to help slightly. It is clear, then, that this direct insertion of the numerical optimizer in Algorithm 1 gives a method which does not preserve realizability.

In addition to an increased chance of crashes, smaller values of Δu can also lead to more oscillatory solutions. We consider this aspect later in Section 6. First, we treat the problem of realizability.

4. Modified scheme to preserve realizability

To understand the reason for the loss of realizability in our tests, we analyze the effects of not being able to solve the optimization problem exactly. It turns out that the optimization error can destroy

[5] Except for some realizable cases where the problem is so poorly conditioned that the numerical optimizer fails to find the minimizer even though it exists. See, e.g., [1].

Table 3.1. Number of time steps until the dual problem cannot be solved. Check marks indicate successful calculations for all time steps.

Δu \ τ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
10^{-1}	✓	✓	✓	✓	✓
10^{-3}	3	3	12	✓	✓
10^{-5}	✓	9	6	8	19

the monotonicity properties that would otherwise be inherited from the underlying scheme for the original PDE (2.1) and would guarantee bounds on the discrete solution.

4.1. **Monotonicity and the optimization error**

The main step in Algorithm 1 is

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left[g(u_{ME}(\bar{\Lambda}_j^n), u_{ME}(\bar{\Lambda}_{j+1}^n))\boldsymbol{\phi} \; - \; g(u_{ME}(\bar{\Lambda}_{j-1}^n), u_{ME}(\bar{\Lambda}_j^n))\boldsymbol{\phi}\right]. \tag{4.1}$$

We can analyze the right-hand side as a function of the point values of the dual states $\bar{\Lambda}_j^n$ by defining $H : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as

$$\begin{aligned} H(\Lambda\ , \Lambda_c, \Lambda_r; \Delta\Lambda) := u_{ME}(\Lambda_c + \Delta\Lambda) \\ - \frac{\Delta t}{\Delta x}\left(g(u_{ME}(\Lambda_c), u_{ME}(\Lambda_r)) - g(u_{ME}(\Lambda\ ), u_{ME}(\Lambda_c))\right), \end{aligned} \tag{4.2}$$

where $\Delta\Lambda$ is used for a point value of the optimization error in Λ. (We typically view $\Delta\Lambda$ as a fixed parameter and are more interested in the behavior of H as a function of its first three arguments.) With

$$\Delta\Lambda_j^n = \Delta\Lambda_j^n(\xi) := \hat{\Lambda}_j^n(\xi) - \bar{\Lambda}_j^n(\xi),$$

(4.1) can now be written as

$$u_j^{n+1} = \left[H(\bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n; \Delta\Lambda_j^n)\boldsymbol{\phi}\right]. \tag{4.3}$$

Since $u_{ME}(\hat{\Lambda}_j^n) = u_{ME}(\bar{\Lambda}_j^n + \Delta\Lambda_j^n)$ fulfills the moment constraint in (2.4) exactly, the equality $u_j^n = u_{ME}(\bar{\Lambda}_j^n + \Delta\Lambda_j^n)\boldsymbol{\phi}$ holds. Therefore, multiplying the first term in $H(\bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n; \Delta\Lambda_j^n)$ with $\boldsymbol{\phi}$ and integrating with respect to ξ yields the moment vector u_j^n in (4.1).

In (4.3), we have written u_j^{n+1} simply as the moments of the update function H, so the realizability of u_j^{n+1} can be established by considering whether H lies in (u_-, u_+). This leads directly to the concept of monotone schemes for scalar conservation laws, because monotone schemes give numerical solutions which satisfy a maximum principle. Thus monotonicity can be used to ensure realizability.

Proposition 4.1. *Assume H is monotonically increasing in each of its first three arguments. Then if the entropy ansatz u_{ME} only takes values in (u_-, u_+), the moment vector u_j^{n+1} computed according to* (4.3) *(or equivalently* (4.1)*) is realizable for any dual states $\bar{\Lambda}_{j-1}^n$, $\bar{\Lambda}_j^n$, and $\bar{\Lambda}_{j+1}^n$.*

Proof. We must show that u_j^{n+1} lies in the realizable set R, which we defined in (2.10). Due to (4.3) it suffices to show that $H(\bar{\Lambda}_{j-1}^n(\xi), \bar{\Lambda}_j^n(\xi), \bar{\Lambda}_{j+1}^n(\xi); \Delta\Lambda_j^n(\xi)) \quad (u_-, u_+)$ for all $\xi \quad \Theta$. For an arbitrary but fixed ξ, let us define

$$\bar{\Lambda}_{j,\max}^n(\xi) := \max\left[\bar{\Lambda}_{j-1}^n(\xi), \bar{\Lambda}_j^n(\xi), \bar{\Lambda}_{j+1}^n(\xi)\right].$$

By monotonicity we have for this ξ

$$\begin{aligned} H(\bar{\Lambda}^n_{j-1}(\xi), \bar{\Lambda}^n_j(\xi), \bar{\Lambda}^n_{j+1}(\xi); \Delta\Lambda^n_j(\xi)) &\le H(\bar{\Lambda}^n_{j,\max}(\xi), \bar{\Lambda}^n_{j,\max}(\xi), \bar{\Lambda}^n_{j,\max}(\xi); \Delta\Lambda^n_j(\xi)) \\ &= u_{ME}(\bar{\Lambda}^n_{j,\max}(\xi) + \Delta\Lambda^n_j(\xi)) < u_+. \end{aligned}$$

Since ξ was arbitrary, we have $H < u_+$ for every ξ. The other direction, $H > u_-$ can be shown analogously. Finally, since $H \quad (u_-, u_+)$ for every ξ, then $u^{n+1}_j = H(\bar{\Lambda}^n_{j-1}, \bar{\Lambda}^n_j, \bar{\Lambda}^n_{j+1}; \Delta\Lambda^n_j)\boldsymbol{\phi}$ is realizable. ∎

Now, monotonicity of H depends on the monotonicity of the scheme defined by the numerical flux $g = g(u, u_r)$ for the original PDE (2.1). We assume that under the standard CFL condition,

$$\frac{\Delta t}{\Delta x} \max_{u \in [u_-, u_+]} |f'(u)| \le 1, \tag{4.4}$$

$g(u, u_r)$ gives a monotone scheme for the underlying equation, i.e., that the function $h : \mathbb{R}\times\mathbb{R}\times\mathbb{R} \to \mathbb{R}$ defined by

$$h(u, v, w) = v - \frac{\Delta t}{\Delta x}(g(v, w) - g(u, v)) \tag{4.5}$$

is monotonically increasing in each argument. [6] This implies

$$\frac{\partial g}{\partial u} \ge 0, \tag{4.6a}$$

$$1 - \frac{\Delta t}{\Delta x}\left[\frac{\partial g}{\partial u} - \frac{\partial g}{\partial u_r}\right] \ge 0, \tag{4.6b}$$

$$\frac{\partial g}{\partial u_r} \le 0. \tag{4.6c}$$

Using this along with properties of the entropy ansatz, we can immediately show that H is monotone in the first and third arguments, since

$$\frac{\partial H}{\partial \Lambda} = \frac{\Delta t}{\Delta x}\frac{\partial g}{\partial u} u'_{ME}(\Lambda) = \frac{\Delta t}{\Delta x} \underbrace{\frac{\partial g}{\partial u}}_{\substack{\ge 0 \\ \text{by (4.6a)}}} \underbrace{\frac{1}{s'(u_{ME}(\Lambda))}}_{\substack{\ge 0 \\ \text{by convexity}}} \ge 0, \tag{4.7a}$$

$$\frac{\partial H}{\partial \Lambda_r} = -\frac{\Delta t}{\Delta x} \underbrace{\frac{\partial g}{\partial u_r}}_{\substack{\le 0 \\ \text{by (4.6c)}}} \underbrace{\frac{1}{s'(u_{ME}(\Lambda_r))}}_{\substack{\ge 0 \\ \text{by convexity}}} \ge 0, \tag{4.7b}$$

where we used that $u'_{ME}(\Lambda) = [(s')^{-1}(\Lambda)]' = 1/s'(u_{ME}(\Lambda))$. The properties in (4.7) hold for *any* value of $\Delta\Lambda^n_j$. But in the second argument, the optimization error $\Delta\Lambda^n_j$ can destroy monotonicity:

$$\frac{\partial H}{\partial \Lambda_c} = u_{ME}(\Lambda_c + \Delta\Lambda^n_j) - \frac{\Delta t}{\Delta x}\left[\frac{\partial g}{\partial u} u_{ME}(\Lambda_c) - \frac{\partial g}{\partial u_r} u_{ME}(\Lambda_c)\right] \tag{4.8a}$$

$$= u_{ME}(\Lambda_c + \Delta\Lambda^n_j)\left[1 - \frac{u_{ME}(\Lambda_c)}{u_{ME}(\Lambda_c + \Delta\Lambda^n_j)}\frac{\Delta t}{\Delta x}\left[\frac{\partial g}{\partial u} - \frac{\partial g}{\partial u_r}\right]\right]. \tag{4.8b}$$

[6] The update functions H and h are simply related by

$$h(u_{ME}(\Lambda), u_{ME}(\Lambda_c), u_{ME}(\Lambda_r)) = H(\Lambda, \Lambda_c, \Lambda_r; 0).$$

The u_{ME} factor in front is again nonnegative by convexity of s, but since the ratio $u_{ME}(\Lambda_c)/u_{ME}(\Lambda_c + \Delta\Lambda_j^n)$ can certainly be larger than one, the standard CFL condition (4.4) cannot be applied to show nonnegativity of the second factor in (4.8b).

There are now two ways to achieve monotonicity despite the optimization error.

4.2. **Modifying the CFL condition**

The more precisely the numerical optimizer solves the optimization problem, the smaller the ratio $u_{ME}(\Lambda_c)/u_{ME}(\Lambda_c + \Delta\Lambda_j^n)$ becomes. This suggests using it as a stopping criterion and then incorporating it into a modified CFL condition. Summing up the findings from subsection 4.1, we obtain the following theorem:

Theorem 4.2. *Assume that the entropy ansatz only takes values in (u_-, u_+) and that g gives a monotone scheme. Then when the numerical optimizer enforces the stopping criterion*

$$\max_{\xi \in \Theta} \left| \max_{\Lambda \in [\bar{\Lambda}_{j,min}^n, \bar{\Lambda}_{j,max}^n]} \frac{u_{ME}(\Lambda(\xi))}{u_{ME}(\Lambda(\xi) + \Delta\Lambda_j^n(\xi))} \right| \leq \gamma, \tag{4.9}$$

where

$$\bar{\Lambda}_{j,min}^n(\xi) := \min \left\{ \bar{\Lambda}_{j-1}^n(\xi), \bar{\Lambda}_j^n(\xi), \bar{\Lambda}_{j+1}^n(\xi) \right\} \quad \text{and} \quad \bar{\Lambda}_{j,max}^n(\xi) := \max \left\{ \bar{\Lambda}_{j-1}^n(\xi), \bar{\Lambda}_j^n(\xi), \bar{\Lambda}_{j+1}^n(\xi) \right\},$$

the new moment vector $\boldsymbol{u}_j^{n+1}$ computed by (4.3) (i.e., (4.1)) is realizable under the modified CFL condition

$$\gamma \frac{\Delta t}{\Delta x} \max_{u \in [u_-, u_+]} |f'(u)| \leq 1. \tag{4.10}$$

The condition (4.9) can be used instead of or in addition to (3.3). The user chooses the parameter γ. Larger values of γ make the condition easier to fulfill, i.e., require fewer optimization iterations, but come at the cost of requiring smaller time steps and leading to more diffusive solutions.

But a stopping criterion based on (4.9) cannot be implemented directly because $\Delta\Lambda_j^n$ is of course unknown. An approximation of $\Delta\Lambda_j^n = (\hat{\boldsymbol{\lambda}}_j^n - \bar{\boldsymbol{\lambda}}_j^n)^T \boldsymbol{\phi}$ can be constructed using the Newton step. If we let $\bar{\boldsymbol{\lambda}} \in \mathbb{R}^{N+1}$ denote an iterate in the optimization algorithm, $H \in \mathbb{R}^{N+1 \times N+1}$ the Hessian and $g \in \mathbb{R}^{N+1}$ the gradient of the dual problem (2.5), we approximate $\hat{\boldsymbol{\lambda}}$ by

$$\hat{\boldsymbol{\lambda}} \approx \bar{\boldsymbol{\lambda}} - \zeta H^{-1}(\bar{\boldsymbol{\lambda}}) g(\bar{\boldsymbol{\lambda}}). \tag{4.11}$$

A safety parameter ζ is used to prevent underestimating the distance between $\boldsymbol{\lambda}$ and $\hat{\boldsymbol{\lambda}}$.

Even with this approximation, a stopping criterion based on (4.9) faces the problem that, since it includes $\bar{\Lambda}_{j\pm1}^n$, the numerical solutions at neighboring cells are coupled, thus destroying parallelizability. We avoid this by simply taking $\bar{\Lambda}_{j,\min}^n = \bar{\Lambda}_{j,\max}^n = \bar{\Lambda}_j^n$ and assuming that the safety parameter ζ can account for the error this introduces.

There are potential drawbacks of using Algorithm 1 with the modified CFL condition (4.10). First, it further restricts the time step, which introduces numerical diffusion. Second, the stopping criterion is difficult to implement, and it's not immediately clear if we can practically satisfy it for a reasonably small value of γ. Third, the choice $\Delta u = 0$ is prohibited if the initial condition takes on values of $\min u_0$ or $\max u_0$ on a nonzero measure. This is because in this case, the correct dual state $\hat{\Lambda}$ goes to infinity, leading to an infinite value of γ no matter how precisely the numerical optimizer solves the dual problem. We explore these potential problems in our numerical results in section 7.

Remark 4.3. The issue of realizability is also an issue for minimum-entropy methods in kinetic theory. A realizability-preserving modified CFL condition similar to the one presented in Theorem 4.2

was derived in [1]. But in kinetic theory, one only needs to ensure the nonnegativity of the underlying update, whereas we need to enforce both upper and lower bounds. Because of this difference the modified CFL condition from [1] is not enough to ensure realizability for the IPM method.

4.3. **Modifying the scheme**

Another way to prevent the optimization error from destroying the monotonicity properties of the underlying scheme is to remove the optimization error completely from our application of the underlying scheme, so that the ratio $u_{ME}(\Lambda_c)/u_{ME}(\Lambda_c + \Delta\Lambda_j^n)$ doesn't even appear in (4.8). This is the case if $\Delta\Lambda_j^n = 0$, i.e. if the dual state belonging to the first term of H equals the inexact dual state used in the numerical fluxes. Since the exact dual state cannot be computed, this means using the dual states $\bar{\Lambda}_j^n$ also in the first term of H.

More specifically, let us define the modified update function $\bar{H}(\Lambda, \Lambda_c, \Lambda_r) = H(\Lambda, \Lambda_c, \Lambda_r; 0)$, i.e.,

$$\bar{H}(\Lambda, \Lambda_c, \Lambda_r) := u_{ME}(\Lambda_c) - \frac{\Delta t}{\Delta x}\left(g(u_{ME}(\Lambda_c), u_{ME}(\Lambda_r)) - g(u_{ME}(\Lambda), u_{ME}(\Lambda_c))\right).$$

Now $\bar{H}$ immediately inherits the monotonicity properties of h in (4.5) under the original CFL condition (4.4), no matter how big or small the optimization error is. The algorithm when using $\bar{H}$ instead of H as underlying function can be written in the original form as

$$u_j^{n+1} = \bar{H}(\bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n)\boldsymbol{\phi} \tag{4.12a}$$

$$= \bar{u}_j^n - \frac{\Delta t}{\Delta x}\left(g(u_{ME}(\bar{\Lambda}_j^n), u_{ME}(\bar{\Lambda}_{j+1}^n))\boldsymbol{\phi} - g(u_{ME}(\bar{\Lambda}_{j-1}^n), u_{ME}(\bar{\Lambda}_j^n))\boldsymbol{\phi}\right), \tag{4.12b}$$

where $\bar{u}_j^n := u_{ME}(\bar{\Lambda}_j^n)\boldsymbol{\phi} \in \mathbb{R}^{N+1}$, and we present it in Algorithm 2.

Algorithm 2 Modified IPM algorithm

for $j = 0$ to $N_x + 1$ **do**
  $u_j^0 = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u_0(x, \cdot)\boldsymbol{\phi}\, dx$
end for
for $n = 0$ to N_t **do**
  for $j = 0$ to $N_x + 1$ **do**
    $\bar{\boldsymbol{\lambda}}_j^n \approx \arg\min_{\boldsymbol{\lambda}} s(\boldsymbol{\lambda}^T\boldsymbol{\phi}) - \boldsymbol{\lambda}^T u_j^n$ such that (3.3) holds
    $\bar{\Lambda}_j^n = \bar{\boldsymbol{\lambda}}_j^{nT}\boldsymbol{\phi}$
    $\bar{u}_j^n = u_{ME}(\bar{\Lambda}_j^n)\boldsymbol{\phi}$
  end for
  for $j = 1$ to N_x **do**
    $u_j^{n+1} = \bar{u}_j^n - \frac{\Delta t}{\Delta x}\left(g(u_{ME}(\bar{\Lambda}_j^n), u_{ME}(\bar{\Lambda}_{j+1}^n))\boldsymbol{\phi} - g(u_{ME}(\bar{\Lambda}_{j-1}^n), u_{ME}(\bar{\Lambda}_j^n))\boldsymbol{\phi}\right)$
  end for
end for

But of course, one cannot simply use any value of the optimization tolerance τ and expect to end up with accurate results. However, when the numerical flux G is Lipschitz continuous in each argument with constant K, the error between the update of Algorithm 2 and the exact update of (3.2) is simply

$O(\tau)$. Indeed, let $c := \Delta t/\Delta x$; then we have

$$\left| \left(u_j^n - \frac{\Delta t}{\Delta x} \left(G(u_j^n, u_{j+1}^n)) - G(u_{j-1}^n, u_j^n) \right) \right.$$
$$\left. - \left(\bar{u}_j^n - \frac{\Delta t}{\Delta x} \left(G(\bar{u}_j^n, \bar{u}_{j+1}^n) - G(\bar{u}_{j-1}^n, \bar{u}_j^n) \right) \right) \right| \leq (1 + 4cK)\,\tau.$$

Therefore we simply need to choose τ with the order of accuracy of the one-step error, which in this case is $O(\Delta t \Delta x) = O(\Delta x^2)$. Then the results computed by Algorithm 2 have the same order of accuracy as those computed by the exact method.

The main drawback to Algorithm 2 is that it is no longer in conservative form. However, when we take $\tau = O(\Delta x^2)$, the nonconservative part vanishes as the grid is refined. Furthermore, in our numerical results below we did not observe any large increases in error compared to the solutions computed using the method presented in section 4.2 with small values of γ.

Remark 4.4. Proposition 4.1 shows realizability if integrals are evaluated exactly, by showing that $H \quad (u_-, u_+)$ for all ξ. When using quadrature rules to approximate integrals, it suffices to show $H \quad (u_-, u_+)$ for all quadrature points ξ_k, hence the requirements of Proposition 4.1 ensure realizability when using quadrature rules. For more details on the realizable set for quadrature rules, see [2].

5. Extending the scheme to higher order

The main computational expense of minimum-entropy methods comes from the repeated numerical solution of the dual problem, which needs to be solved for every spatial cell. With a high-order method, fewer spatial cells can achieve a desired level of accuracy. In this section we show how to construct a realizability-preserving second-order method.

5.1. **Second-order spatial reconstruction**

First we give a stable second-order method for the original PDE (2.1) and then plug the entropy ansatz u_{ME} into this method and integrate the equations against the basis functions ϕ to get a second-order method for the IPM system.

We start by defining a linear spatial reconstruction of the solution in each cell j by $p_j^n(x) = u_j^n + (x - x_j)\sigma_j^n$. Here $\sigma_j^n := \sigma(u_{j-1}^n, u_j^n, u_{j+1}^n)$ is the slope of the reconstruction in cell j at time step t_n. We use the second-order stable *minmod* slope $\sigma : \mathrm{R} \times \mathrm{R} \times \mathrm{R} \to \mathrm{R}$, which is given by

$$\sigma(u, v, w) = \frac{1}{\Delta x} \mathrm{minmod}(w - v, v - u)$$

with the minmod function

$$\mathrm{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b|, ab > 0 \\ b & \text{if } |b| < |a|, ab > 0 \\ 0 & \text{else} \end{cases}.$$

The reconstructions give cell edge values

$$u_{j\pm1/2}^{n,} := u_j^n \pm \sigma_j^n \frac{\Delta x}{2}, \tag{5.1}$$

which are inserted into the numerical flux to give the time update:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(g(u_{j+1/2}^{n,-}, u_{j+1/2}^{n,+}) - g(u_{j-1/2}^{n,-}, u_{j-1/2}^{n,+})). \tag{5.2}$$

When we use the slopes given by the minmod limiter, the reconstructions further have the property that the edge values are bounded by the values of the cell means. This property is crucial for realizability. [7]

For the IPM method, we apply this numerical scheme point-wise in ξ using the entropy ansätze computed by the numerical optimizer. That is, for every ξ we compute the slope

$$\tilde{\sigma}_j^n = \sigma(u_{ME}(\bar{\Lambda}_{j-1}^n), u_{ME}(\bar{\Lambda}_j^n), u_{ME}(\bar{\Lambda}_{j+1}^n)). \tag{5.3}$$

This gives the edge values

$$u_{j\pm1/2}^{n,}(\bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n) := u_{ME}(\bar{\Lambda}_j^n) \pm \frac{\Delta x}{2}\tilde{\sigma}_j^n.$$

Now we want to consider the monotonicity of the time update (5.2) with respect to the dual states of the cell average and both sides of the neighboring edges. For this we need to define the dual states of the edges,

$$\bar{\Lambda}_{j\pm1/2}^{n,} := s\,(u_{j\pm1/2}^{n,}(\bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n)), \tag{5.4}$$

so that we can write (5.2) applied to IPM with

$$\begin{aligned} H_2(\Lambda_c, \Lambda_r^-, \Lambda_r^+, \Lambda^-, \Lambda^+; \Delta\Lambda) &:= u_{ME}(\Lambda_c + \Delta\Lambda) \\ &\quad - \frac{\Delta t}{\Delta x}\Big[g(u_{ME}(\Lambda_r^-), u_{ME}(\Lambda_r^+)) - g(u_{ME}(\Lambda^-), u_{ME}(\Lambda^+))\Big], \end{aligned} \tag{5.5a}$$

as

$$u_j^{n+1} = H_2(\bar{\Lambda}_j^n, \bar{\Lambda}_{j+1/2}^{n,-}, \bar{\Lambda}_{j+1/2}^{n,+}, \bar{\Lambda}_{j-1/2}^{n,-}, \bar{\Lambda}_{j-1/2}^{n,+}; \Delta\Lambda_j^n)\boldsymbol{\phi}\ . \tag{5.5b}$$

After having derived this underlying scheme we can find a time-step restriction which ensures realizability.

Theorem 5.1. *Assume that the entropy ansatz only takes values in (u_-, u_+) and that g gives a monotone scheme. Then the time-updated moment vector u_j^{n+1} from the second-order in space scheme (5.5) is realizable under the time-step restriction*

$$\bar{\gamma}\max_{u\in[u_-,u_+]}|f'(u)|\frac{\Delta t}{\Delta x} \le \frac{1}{2} \tag{5.6}$$

where $\bar{\gamma}$ satisfies

$$\max_{\xi\in\Theta}\Big|\max_{\Lambda\in[\bar{\Lambda}_{j,min}^n, \bar{\Lambda}_{j,max}^n]}\frac{u_{ME}'(\Lambda)}{u_{ME}'(\Lambda + \Delta\Lambda_{j\pm1/2}^{n,})}\Big| \le \bar{\gamma}, \tag{5.7}$$

with

$$\bar{\Lambda}_{j,min}^n(\xi) := \min_{j-2\le i\le j+2}\bar{\Lambda}_i^n(\xi) \quad \textit{and} \quad \bar{\Lambda}_{j,max}^n(\xi) := \max_{j-2\le i\le j+2}\bar{\Lambda}_i^n(\xi).$$

Proof. As in Proposition 4.1, we show that $H_2(\bar{\Lambda}_j^n, \bar{\Lambda}_{j+1/2}^{n,-}, \bar{\Lambda}_{j+1/2}^{n,+}, \bar{\Lambda}_{j-1/2}^{n,-}, \bar{\Lambda}_{j-1/2}^{n,+}; \Delta\Lambda_j^n)$ increases monotonically in its first five arguments.

We show monotonicity by adopting the technique of writing H_2 as a convex combination of evaluations of the first-order scheme of (4.3) [24]. We write

$$u_{ME}(\bar{\Lambda}_j^n + \Delta\Lambda_j^n) = u_{ME}(\hat{\Lambda}_j^n) = \frac{1}{2}\Big[u_{ME}(\hat{\Lambda}_{j-1/2}^{n,+}) + u_{ME}(\hat{\Lambda}_{j+1/2}^{n,-})\Big], \tag{5.8}$$

[7] For other slopes which do not have this property, one would have to implement a bound-preserving limiter, see e.g. [24].

where $\hat{\Lambda}^{n,}_{j\pm1/2}$ are defined as in (5.4) but with $(\bar{\Lambda}^n_{j-1}, \bar{\Lambda}^n_j, \bar{\Lambda}^n_{j+1})$ replaced by $(\hat{\Lambda}^n_{j-1}, \hat{\Lambda}^n_j, \hat{\Lambda}^n_{j+1})$, [8] and insert this into (5.5a), so that after adding and subtracting

$$\frac{\Delta t}{\Delta x} g\left(u_{ME}\left(\bar{\Lambda}^{n,+}_{j-1/2}\right), u_{ME}\left(\bar{\Lambda}^{n,-}_{j+1/2}\right)\right)$$

we can write H_2 as

$$\begin{aligned} &H_2(\bar{\Lambda}^n_j, \bar{\Lambda}^{n,-}_{j+1/2}, \bar{\Lambda}^{n,+}_{j+1/2}, \bar{\Lambda}^{n,-}_{j-1/2}, \bar{\Lambda}^{n,+}_{j-1/2}; \Delta\Lambda^n_j) \\ &\quad = \frac{1}{2}\left(H_1(\bar{\Lambda}^{n,+}_{j-1/2}, \bar{\Lambda}^{n,-}_{j+1/2}, \bar{\Lambda}^{n,+}_{j+1/2}; \Delta\Lambda^-_{j+1/2}) + H_1(\bar{\Lambda}^{n,-}_{j-1/2}, \bar{\Lambda}^{n,+}_{j-1/2}, \bar{\Lambda}^{n,-}_{j+1/2}; \Delta\Lambda^+_{j-1/2})\right), \end{aligned} \tag{5.9}$$

where

$$H_1(\Lambda\ , \Lambda_c, \Lambda_r; \Delta\Lambda) := u_{ME}(\Lambda_c + \Delta\Lambda) - 2\frac{\Delta t}{\Delta x}\left(g(u_{ME}(\Lambda_c), u_{ME}(\Lambda_r)) - g(u_{ME}(\Lambda\ ), u_{ME}(\Lambda_c))\right)$$

and

$$\Delta\Lambda^{n,}_{j\pm1/2} := \hat{\Lambda}^{n,}_{j\pm1/2} - \bar{\Lambda}^{n,}_{j\pm1/2}. \tag{5.10}$$

The function H_1 is similar to the first-order update function (4.2), so that one readily recognizes that each H_1 term in (5.9) is monotone in the relevant arguments under the conditions

$$2\frac{u_{ME}(\Lambda)}{u_{ME}\left(\Lambda + \Delta\Lambda^{n,}_{j\pm1/2}\right)} u \max_{[u_-,u_+]} |f'(u)|\frac{\Delta t}{\Delta x} \le 1, \tag{5.11}$$

for $\Lambda \in \left[\bar{\Lambda}^n_{j,\min}, \bar{\Lambda}^n_{j,\max}\right]$ respectively.

Thus under (5.6) H_2 is monotone in its first five arguments, and the realizability of u^{n+1}_j follows. ■

Unfortunately a stopping criterion based on (5.7) leads to an even stronger coupling of the numerical optimization. We avoid this by adopting the same strategy as in Section 4.3: that is, we replace H_2 with $\bar{H}_2(\Lambda_c, \Lambda_r^-, \Lambda_r^+, \Lambda^-, \Lambda^+) := H_2(\Lambda_c, \Lambda_r^-, \Lambda_r^+, \Lambda^-, \Lambda^+; 0)$. Thus we do not have to consider the optimization error when checking monotonicity, and we get monotonicity under the condition

$$u \max_{[u_-,u_+]} |f'(u)|\frac{\Delta t}{\Delta x} \le \frac{1}{2}. \tag{5.12}$$

In order to maintain accuracy, we use the stopping criterion (3.3) with $\tau = O(\Delta x^3)$.

Remark 5.2. When replacing moments as proposed in Section 4.3, the optimization error no longer affects realizability. In this case, the IPM solution inherits the bounds guaranteed by the underlying scheme $\bar{H}$. This can be used to further increase the order of the spatial discretization: A scheme of arbitrarily high order guaranteeing bounds on the solution can be constructed with DG or WENO methods using bound-preserving limiters. Choosing such a method as underlying scheme yields a realizable moment update of arbitrarily high order. Furthermore, since bound-preserving methods exist for systems, this strategy can also be used to construct realizability preserving methods if the original problem is a system of equations. It is however important to point out the need to control the non-conservative error, which arises when replacing moments.

5.2. **Second-order time integration**

For time integration we use strong stability-preserving (SSP) methods. These are the standard choice for hyperbolic equations and allow us to build on our analysis of forward Euler steps, since SSP

[8] In words, $\hat{\Lambda}^{n,}_{j\pm1/2}$ are derived from the pointwise linear reconstruction using the values of the exact entropy ansatz instead of the approximate entropy ansatz returned by the numerical optimizer.

methods can be written as convex combinations of forward Euler steps. We rewrite a forward Euler step in the form

$$u_j^{n+1} = u_j^n + \Delta t L_j(\bar{\Lambda}_{j-2}^n, \bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n, \bar{\Lambda}_{j+2}^n), \tag{5.13}$$

where

$$L_j(\bar{\Lambda}_{j-2}^n, \bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n, \bar{\Lambda}_{j+2}^n) := -\frac{1}{\Delta x}\Big[g(u_{ME}(\bar{\Lambda}_{j+1/2}^-), u_{ME}(\bar{\Lambda}_{j+1/2}^+))\phi \Big] - \Big[g(u_{ME}(\bar{\Lambda}_{j-1/2}^-), u_{ME}(\bar{\Lambda}_{j-1/2}^+))\phi \Big].$$

In particular, we use multistep SSP methods [29]. With multistep methods, we are able to re-use the evaluations of L_j from previous time steps, in contrast to single-step (i.e., multistage Runge–Kutta) methods, which require multiple evaluations of L_j for each time step. The time update for a general multistep SSP method has the form

$$u_j^{n+1} = \sum_{i=1}^{m} \alpha_i u_j^{n+1-i} + \Delta t \beta_i L_j(\bar{\Lambda}_{j-2}^n, \bar{\Lambda}_{j-1}^n, \bar{\Lambda}_j^n, \bar{\Lambda}_{j+1}^n, \bar{\Lambda}_{j+2}^n),$$

where m is the number of past steps used to compute the $(n+1)$-th time step. When a forward Euler step remains realizable under time step Δt_{FE}, then the multistep SSP method remains realizable under time step $c\Delta t_{\mathrm{FE}}$, where

$$c := \min_{\{i:\beta_i>0\}} \frac{\alpha_i}{|\beta_i|}.$$

We use the four-step second-order method found in [13]:

$$\alpha = \Big[\frac{8}{9}, 0, 0, \frac{1}{9}\Big]^T, \quad \beta = \Big[\frac{4}{3}, 0, 0, 0\Big]^T, \quad c = \frac{2}{3}. \tag{5.14}$$

With this multistep SSP method, the new CFL condition is given by

$$\max_{u\ [u_-,u_+]} |f'(u)| \frac{\Delta t}{\Delta x} \le \frac{1}{3}. \tag{5.15}$$

6. Choosing the Entropy

While the log-barrier does the job of enforcing bounds on the oscillations around $\min u_0$ and $\max u_0$, it is not the only choice which achieves such bounds. If we look at the form of the entropy ansatz $u_{ME}(\Lambda) = (s')^{-1}(\Lambda)$ in (2.6), we see that it is sufficient that the derivative s' maps the open interval (u_-, u_+) to the entire real line. I.e., it suffices that

$$\lim_{u \to u_+} s'(u) \to \infty \quad \text{and} \quad \lim_{u \to u_-} s'(u) \to -\infty \tag{6.1}$$

to achieve the desired bounds on the entropy ansatz. We can use this to find a new entropy with better properties.

In choosing an entropy, our goals are to satisfy the original maximum principle as closely as possible and to obtain a solution which oscillates as little as possible. The first step is ensured by condition (6.1), as long as we take $\Delta u = u_+ - \max u_0 = \min u_0 - u_-$ as small as possible. In fact, ideally we would like to just choose $\Delta u = 0$.

The log-barrier entropy (2.9) achieves condition (6.1) indirectly: by ruling out values outside of (u_-, u_+) using barriers in s' itself. There exist, however, moment vectors for which the ansatz must take on the value $\max u_0$ or $\min u_0$ on sets of nonzero measure. The moment vectors of the initial

condition $u(0,x) = u_0(x,\cdot)\phi$ take on such values when, for example, u_0 attains its maximum or minimum (over all x D and ξ Θ) at some point in space with certainty (i.e., constant in ξ). These moments lie on the boundary of the set of realizability when $\Delta u = 0$, but since the realizable set is open, here the optimization problem has no solution. In the limit as a sequence of moment vectors approaches such a nonrealizable moment, the corresponding limit of entropy ansätze does converge, but the entropy value $s(U_{ME}(u)$ goes to infinity. In this sense, the log-barrier entropy does not always recover the certain case gracefully.

But we can fulfill condition (6.1) without forcing s itself to take infinite values. An entropy which achieves this is

$$s(u) = (u - u_-)\ln(u - u_-) + (u_+ - u)\ln(u_+ - u). \tag{6.2}$$

Note that a similar version of this entropy has also been used in [27]. With $u_- = 0$ and $u_+ = 1$, this is the entropy for particles with Fermi–Dirac statistics. In the following, this entropy is called bounded-barrier (BB) entropy. It satisfies condition (6.1) but is finite on the interval $[u_-, u_+]$. We compare the two entropy functions in Figure 6.1a.

When one also compares the entropy ansätze resulting from the log-barrier and BB entropies in Figure 6.1b, an interesting difference sticks out: Here the BB entropy not only gives a much better solution, but in contrast to the solution using the log-barrier entropy it is not oscillatory around the value $u = (u_+ + u_-)/2 =: u_M$. Further consideration of the shapes of the entropy functions offers a possible explanation. In Figure 6.1a we notice that the log-barrier entropy is much flatter than the BB entropy around their minimum at $u = u_M$. Thus the log-barrier entropy does not distinguish among these values very well, and as a result the oscillations in its entropy ansatz seen in Figure 6.1b are allowed because they have only a small effect on the value of the entropy. Correspondingly, values near the boundaries of the domain u_- and u_+ are strongly punished by the log-barrier entropy; this is in contrast to the bounded-barrier entropy, which simply takes finite values even at the end points.

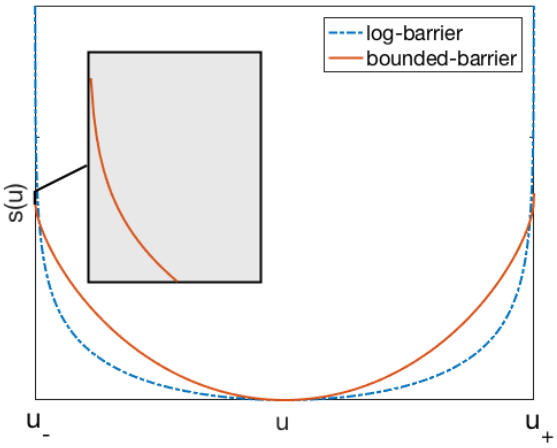
We tested this hypothesis by modifying the values of the slope around u_M using the family of entropies

$$s_k(u) = \left(\frac{s(u) - s\left(\frac{1}{2}(u_- + u_+)\right)}{s(u_{\max}) - s\left(\frac{1}{2}(u_- + u_+)\right)} \right)^k,$$

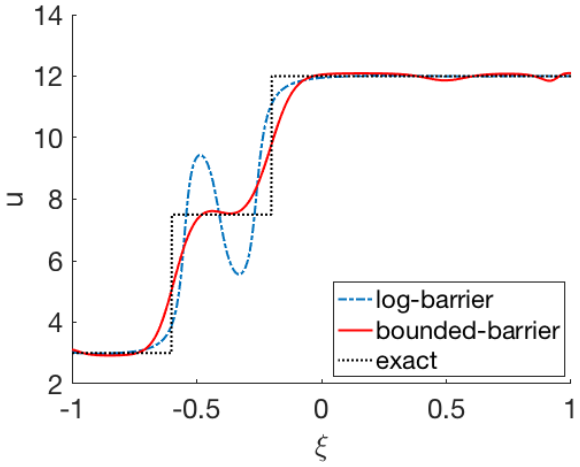
where s is the bounded-barrier entropy. As we show in Figure 6.2a, the higher k is, the flatter the entropy is around u_M, so for higher values of k, we expect the entropy ansatz to be more oscillatory. This is then exactly what we observe in Figure 6.2b.

Another difference between the log- and bounded-barrier entropies is the dependence of the oscillations on the choice of Δu. In numerical experiments, we noticed that with the log-barrier entropy, smaller values of Δu are disadvantageous because the solutions are more oscillatory for smaller values of Δu. We show an example of this behavior in Figure 6.3. Here, we reconstruct a shock from u_M to u_{max}. The bounded-barrier entropy with $\Delta u = 0$ again gives the best result. As we will see in the numerical results in the next section, the bounded-barrier entropy's more gentle behavior near the bounding values u_- and u_+ allows us to choose $\Delta u = 0$ in all our numerical tests, thus exactly enforcing the original maximum principle.

Remark 6.1. For scalar hyperbolic equations, every convex function is an entropy. This is not the case for systems of equations, meaning that the bounded-barrier entropy cannot be used in such a setting. However, the study shows that given a set of admissible entropies for such a system, one should choose an entropy which sufficiently distinguishes between different solution values (provided that such an entropy exists).



(A)

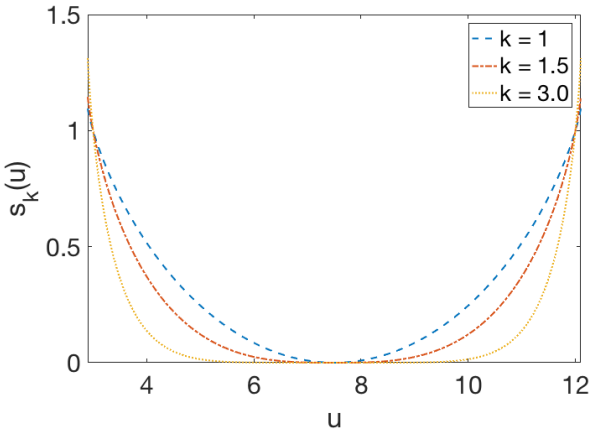


(B)

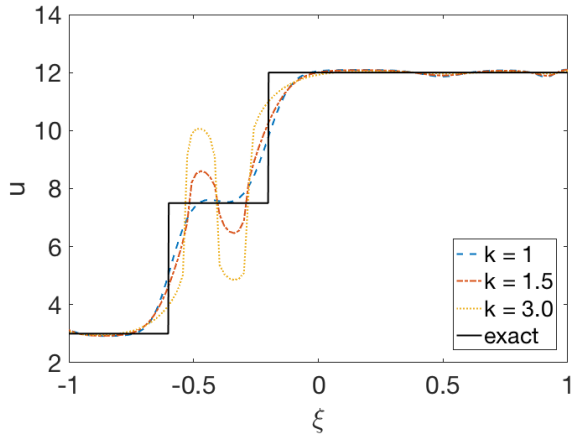
FIGURE 6.1. (A) Comparison of entropies and (B) resulting approximation with $\Delta u = 0.1, N = 10$.

7. Numerical Results

In the following, we first compare the log-barrier and the bounded-barrier entropy in different test cases before turning to investigating the effectiveness of the two strategies to impose realizability. The exact solutions of all problems can be determined with the help of characteristics, see for example [20, Chapter 3]. Furthermore, we use the upwind numerical flux in all test cases.



(A)



(B)

FIGURE 6.2. (A) Family of entropies and (B) corresponding reconstruction for $\Delta u = 0.1, N = 10$.

7.1. **Comparing different entropies**

We start by comparing results when making use of the log- and bounded-barrier entropies. Following [25], we solve the uncertain Burgers' equation

$$\partial_t u(t,x,\xi) + \partial_x \frac{u(t,x,\xi)^2}{2} = 0, \tag{7.1a}$$

$$u(t=0,x,\xi) = u_0(x,\xi), \tag{7.1b}$$

with the first-order method in Algorithm 2. As in [25], we choose the random initial condition

$$u_0(x,\xi) := \begin{cases} u_L, & \text{if } x < x_0 + \sigma\xi \\ u_L + \frac{u_R - u_L}{x_0 - x_1}(x_0 + \sigma\xi - x), & \text{if } x \quad [x_0 + \sigma\xi, x_1 + \sigma\xi] \\ u_R, & \text{else} \end{cases} \tag{7.2}$$

which is a forming shock with a linear connection from x_0 to x_1. In our case, ξ is uniformly distributed on the interval $[-1, 1]$. Due to the fact that we recalculate moments to ensure realizability, we can use

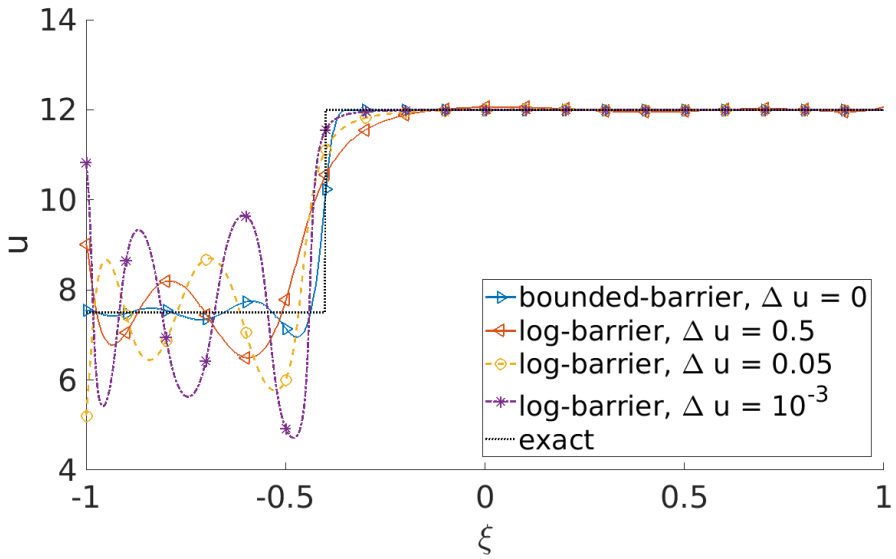


FIGURE 6.3. Approximation behavior for different values of Δu with $N = 10$.

the original CFL condition (4.4). We use the following parameter values:

$[a, b] = [0, 3]$	range of spatial domain
$N_x = 160$	number of spatial cells
$t_{end} = 0.15$	end time
$x_0 = 0.5, x_1 = 1.5, u_L = 12, u_R = 3, \sigma = 0.2$	parameters of initial condition (7.2)
$N + 1 = 5$	number of moments
$\tau = 10^{-7}$	gradient tolerance (3.3)
$\Delta u \quad \{0, 0.001, 0.5\}$	distance u_0 to IPM bounds

Additionally, we computed all integrals in ξ using a forty-point Gauss-Legendre quadrature.

Since the log-barrier entropy is infinite at u_+ and u_-, we need to choose $\Delta u > 0$. We choose $\Delta u = 0.5$ as in [25] as well as $\Delta u = 0.001$ to demonstrate the effects when the solutions lie close to the minimal and maximal value of the exact solution. Note that the maximal velocity of the equation is $u_+ = u_L + \Delta u$, so consequently the CFL condition of the deterministic problem (where velocities are bounded by u_L) cannot be used. The bounded-barrier entropy shows good approximation results for small values of Δu, so we set this parameter to zero, allowing the use of the deterministic CFL condition. Plotting the solutions at fixed values for ξ in Figure 7.2 shows the expected poor approximation behavior of the log-barrier entropy for small values of Δu. The choice $\Delta u = 0.5$ leads to over- and undershoots when using the log-barrier entropy, whereas the bounded-barrier entropy nicely approximates the solution. Furthermore, the solution obtained with the bounded-barrier entropy fulfills the original maximum principle. Looking at the dependency on ξ for a fixed spatial cell in Figure 7.1, one observes that the log-barrier entropy has oscillations whereas the bounded-barrier entropy gives a nonoscillatory solution.

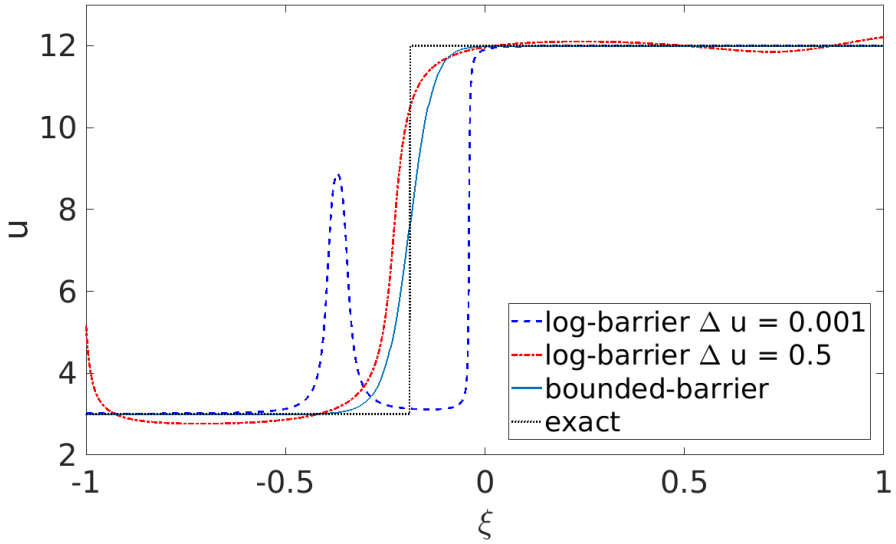


FIGURE 7.1. Solutions for log-barrier and bounded-barrier entropies at fixed spatial position x.

Let us now turn to a new initial condition for the uncertain Burgers' equation in order to investigate the oscillations arising at a noncritical state u_M:

$$u_0(x,\xi) := \begin{cases} u_L, & \text{if } x \le x_0 + \sigma\xi \\ u_L + (u_M - u_L)\cdot \frac{x_0+\sigma\xi - x}{x_0 - x_1}, & \text{if } x \quad (x_0+\sigma\xi, x_1+\sigma\xi] \\ u_M, & \text{if } x \quad (x_1+\sigma\xi, x_2+\sigma\xi] \\ u_M + (u_R - u_M)\cdot \frac{x_3+\sigma\xi - x}{x_3 - x_2}, & \text{if } x \quad (x_2+\sigma\xi, x_3+\sigma\xi] \\ u_R, & \text{if } x > x_3 + \sigma\xi, \end{cases} \tag{7.3}$$

This initial condition describes two forming shocks that connect the three states u_L, u_M, and u_R. All parameters which have been modified can be found in the following table:

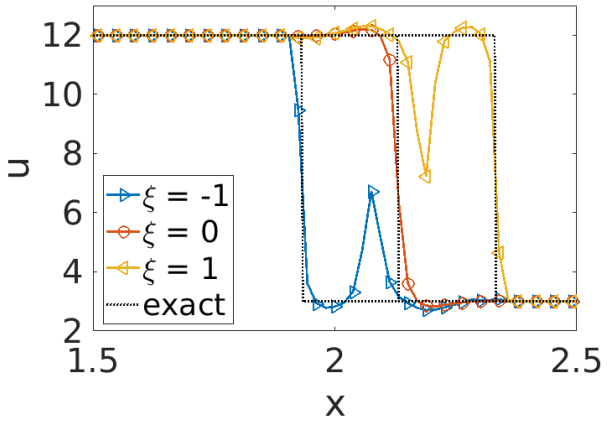
$t_{end} = 0.04$	end time
$x_0 = 0.8, x_1 = 0.98, x_2 = 1.32, x_3 = 1.5, \sigma = 0.5$	parameters of initial condition (7.3)
$N+1 = 16$	number of moments

The results for this problem can be seen in Figure 7.3. One observes that the solution using the log-barrier entropy is oscillatory, whereas with the bounded-barrier entropy the solution shows only small oscillations. While the IPM scheme with the bounded-barrier entropy fulfills the maximum principle, the solution of the log-barrier entropy has over- and undershoots as large as Δu.

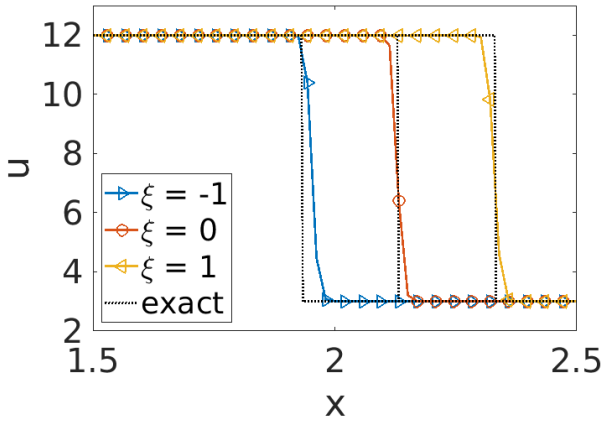
7.2. **Comparison of entropies in two-dimensional Random Space**

To compare both entropies in a two-dimensional random domain (i.e., $P = 2$), the initial condition of the Burgers' test case is changed to

$$u_0(x) := \begin{cases} u_L + \sigma_0\xi_0, & \text{if } x < x_0, \\ u_M + \sigma_1\xi_1, & \text{if } x \quad [x_0, x_1], \\ u_R, & \text{else,} \end{cases} \tag{7.4}$$



(A) log-barrier entropy, $\Delta u = 0.5$.



(B) bounded-barrier entropy, $\Delta u = 0$.

FIGURE 7.2. Solution for different entropies evaluated at $\xi \quad \{-1, 0, 1\}$.

where ξ_0 and ξ_1 are both uniformly distributed in $[-1, 1]$. This test case represents an uncertain multiple-shock flow, which is studied in compressible fluid mechanics, see [25]. Realizability is again preserved by recalculating moments, meaning that the original CFL condition (4.4) can be used. As in [25], a tensorized Clenshaw-Curtis quadrature rule of level 3 and an increased number of $N_x = 6000$ spatial grid points is used. In contrast to the other test cases, we need to choose a fine resolution of the spatial grid to minimize the effects of numerical diffusion, which significantly affects the solution in this test case.

$[a, b] = [0, 1]$	range of spatial domain
$N_x = 6000$	number of spatial cells
$t_{end} = 0.01115$	end time
$x_0 = 0.3, x_1 = 1.6, \sigma_0 = 0.2, \sigma_1 = 0.2,$	parameters of initial condition (7.4)
$u_L = 12, u_M = 6, u_R = 1$	
$N + 1 = 5$	number of moments

The results are given in Figure 7.4. IPM again fulfills the maximum principle when the bounded-barrier entropy is used. The solution has only small oscillations around the intermediate state u_M and shows good agreement with the exact solution. When trying to approach a maximum principle by

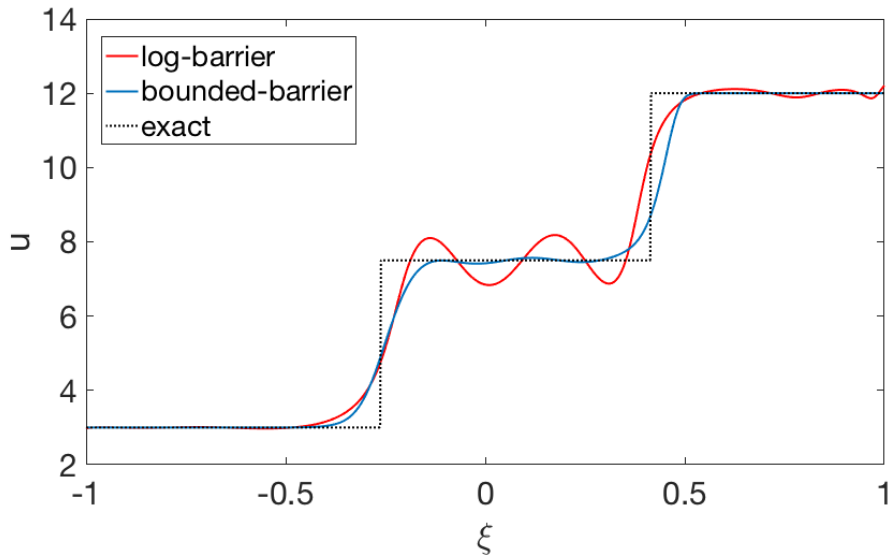


FIGURE 7.3. Solutions for log-barrier and bounded-barrier entropies at $x = 2.1$.

choosing a small value of Δu with the log-barrier entropy, the solution starts to oscillate heavily at the intermediate state. The solution resembles the one-dimensional result for a small value of Δu depicted in Figure 7.1. Choosing the IPM bounds further away from the exact solution bounds (as in [25]), we obtain a more accurate solution. However the maximum principle is not fulfilled since the solution takes on values bigger than 12.34 (off the color scale) while showing oscillations at the intermediate state. This is also in agreement with the one-dimensional results shown before, where the maximum principle is violated by the log-barrier entropy.

7.3. **Convergence of different schemes**

Due to its advantages compared to the log-barrier entropy, the following results have been obtained using the bounded-barrier entropy with $\Delta u = 0$. To investigate the convergence properties of the proposed first- and second-order schemes, we look at the advection equation with uncertain initial data

$$\begin{aligned} \partial_t u(t,x,\xi) + \partial_x u(t,x,\xi) &= 0, \\ u(t=0,x,\xi) &= \sin(x + 0.05\pi\xi), \end{aligned}$$

where $x \in [0,2]$ and $t_{end} = 0.1$. We use periodic boundary conditions at the boundaries of the spatial domain. The number of moments we calculate is 3. We study the L^1 error of the expected value for different numbers of spatial discretization points. Let u_h denote a numerical solution. For first-order methods, it is constant across space in each spatial cell, and for second-order methods, it is defined according to the linear reconstructions given in Section 5. Then we compute the L^1 error for each moment component by

$$e := \int_0^2 |u_h(t_{end},x) - u(t_{end},x)|\,dx,$$

where $u(t_{end},x)$ is the exact solution to the system of IPM moment equations (2.8) at the final time t_{end}, and the absolute value and integral are taken component-wise. In the following convergence results, we plot only the results for the zero-th component of e. The resulting convergence plot is given

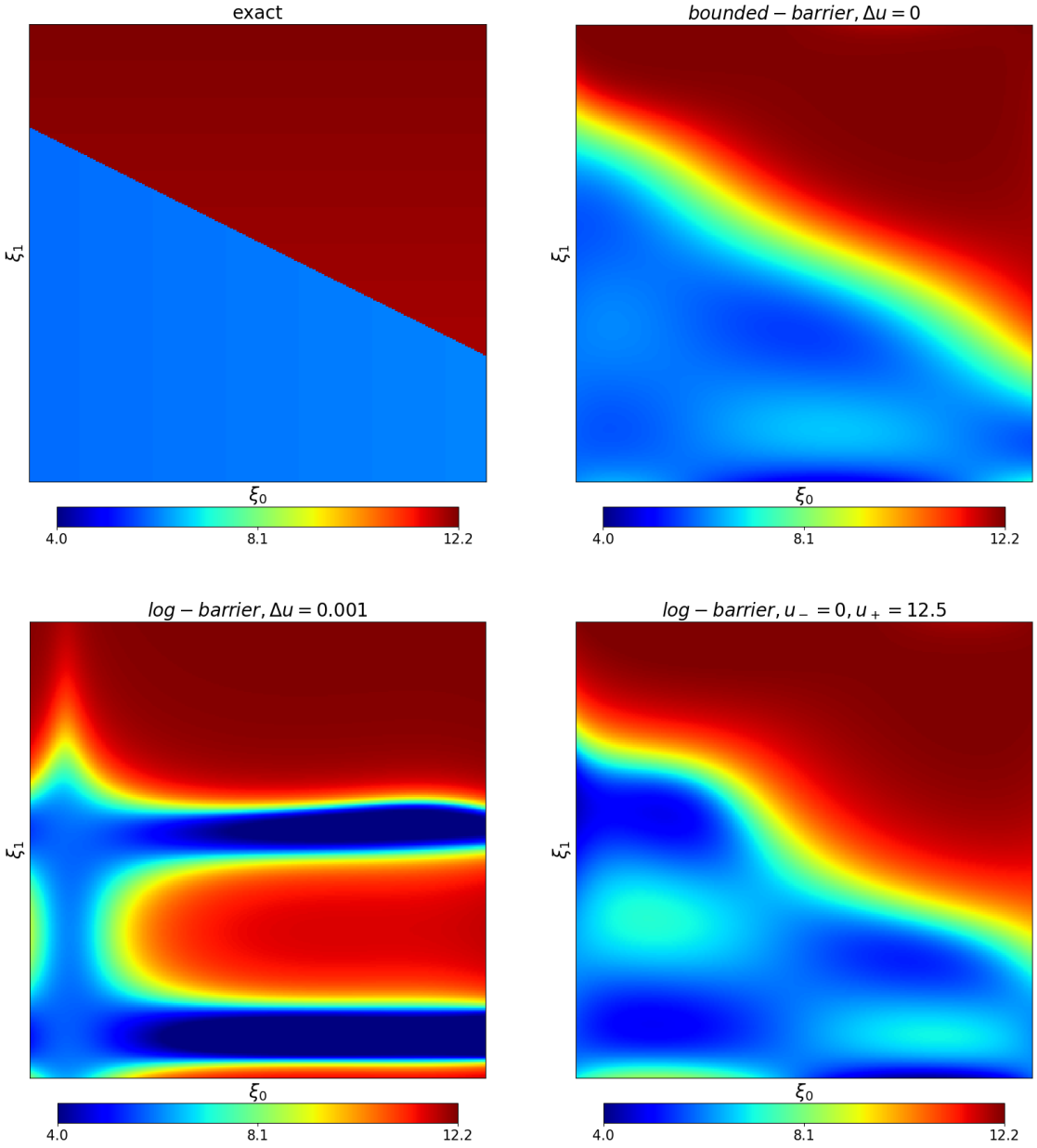
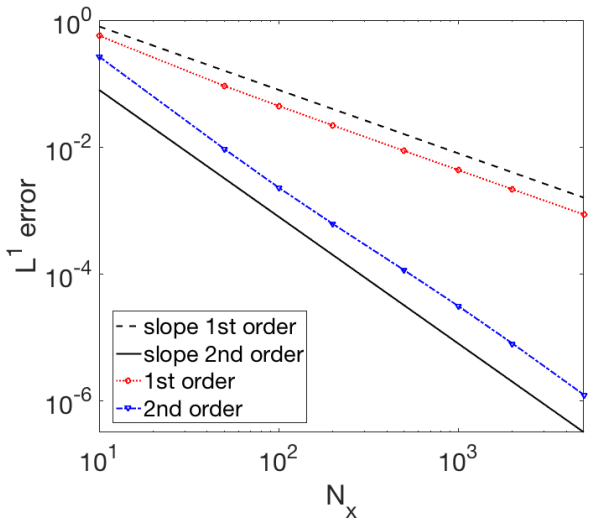


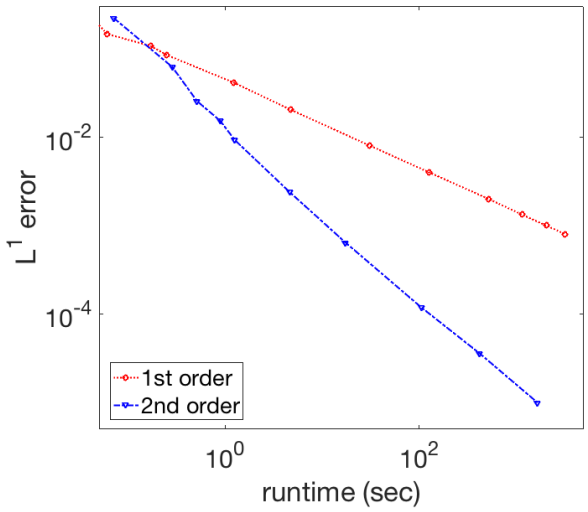
Figure 7.4. Solution at $x = 0.4$ with different entropies.

in Figure 7.5a. Both methods recalculate moments with the inaccurate dual states, meaning that in order to preserve the expected convergence rate $p \in \{1, 2\}$, the stopping criterion of the optimization method needs to be set to $\tau = \Delta x^{p+1}$. For the time discretization of the second-order scheme, the four-step SSP scheme (5.14) has been used. Heun's method is used to calculate the first three time steps. That the different schemes show the expected convergence.

The efficiency of the two methods shown in Figure 7.5b demonstrates that the second-order scheme reaches most levels of accuracy with less computing time than the first-order scheme.



(A)



(B)

FIGURE 7.5. (A) Convergence of different IPM discretizations and (B) efficiency when using first- and second-order methods.

7.4. **Comparison of strategies to preserve realizability**

Two strategies to ensure realizability have been presented in section 4.2 and section 4.3, namely using a modified CFL condition or modifying moments. To compare these two strategies, we look at the uncertain advection equation given by

$$\partial_t u(t,x,\xi) + a(\xi)\partial_x u(t,x,\xi) = 0,$$
$$u(t=0,x) = u_0(x).$$

We choose $a(\xi) := 11 + \xi$, where ξ is uniformly distributed on $[-1,1]$, so the velocity is uniformly distributed in the interval $[10,12]$. What is interesting about this equation is that the velocity of the system is not known, which means that our CFL condition adds artificial viscosity to smaller velocities,

while high velocities are well resolved. We use the deterministic initial condition

$$u_0(x) := \begin{cases} u_L, & \text{if } x < x_0, \\ u_L + \frac{u_R - u_L}{x_0 - x_1}(x_0 - x), & \text{if } x \quad [x_0, x_1], \\ u_R, & \text{else.} \end{cases} \tag{7.5}$$

Parameters of the calculation can be found in the following table:

$N_x = 80$	number of spatial cells
$t_{end} = 0.19$	end time
$x_0 = 0.5, x_1 = 0.55, u_L = 12, u_R = 3$	parameters of initial condition (7.5)
$N + 1 = 10$	number of moments
$\gamma \quad \{1.5, 1.1, 1 + 10^{-7}\}, \zeta = 5$	CFL modification
$\zeta = 5$	safety factor in estimation of $\hat{\lambda}$ (4.11)
$\Delta u \quad \{0, 10^{-7}\}$	distance u_0 to IPM bounds

When modifying moments, we perform the computation using a first-order scheme as well as with second-order spatial reconstructions using the minmod limiter. Since the second-order time discretization adds artificial viscosity without improving the accuracy, we use the explicit Euler method. We use $\Delta u = 10^{-7}$ so that we can achieve the stopping criterion (4.10) on every quadrature point. Figure 7.6

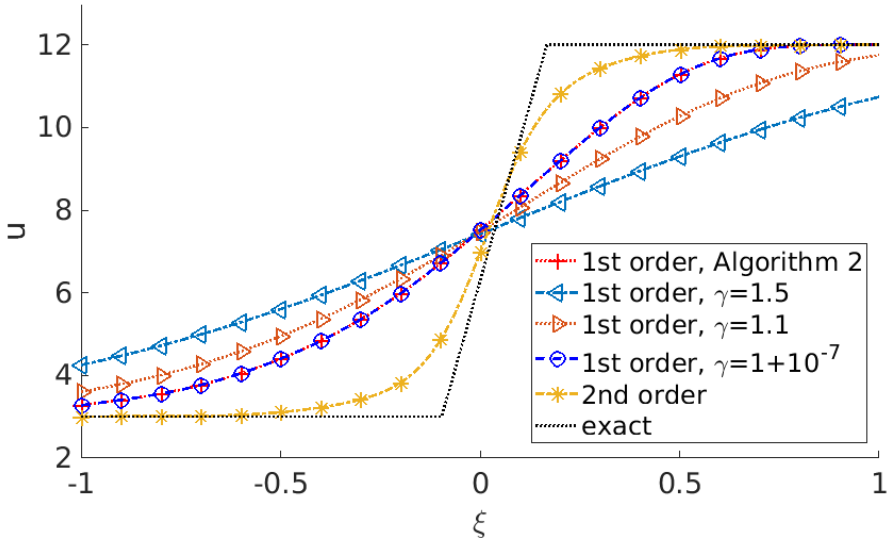


Figure 7.6. Solutions at $x \quad = 2.6$ with and without using a spatial limiter.

shows the solution at a fixed position x . Using Algorithm 2 to ensure realizability allows the use of the deterministic CFL condition (4.4). We compare this solution to those obtained with a modified CFL conditions according to section 4.2. As expected, modifying the CFL condition by $\gamma = 1.5$ leads to a heavily smeared-out solution. However, we found that for this problem, it was possible to set γ as small as $1 + 10^{-7}$. The solution calculated with this value of γ is essentially identical to the solution using Algorithm 2; they differ only on the order of 10^{-3} in the L^∞ norm. One can conclude that both realizability-preserving strategies for first-order methods are satisfactory.

Figure 7.6 also shows that the second-order spatial reconstructions give much better results. This improvement is also seen in the expected value and standard deviation in Figure 7.7. The standard deviation is particularly improved by going to second-order.

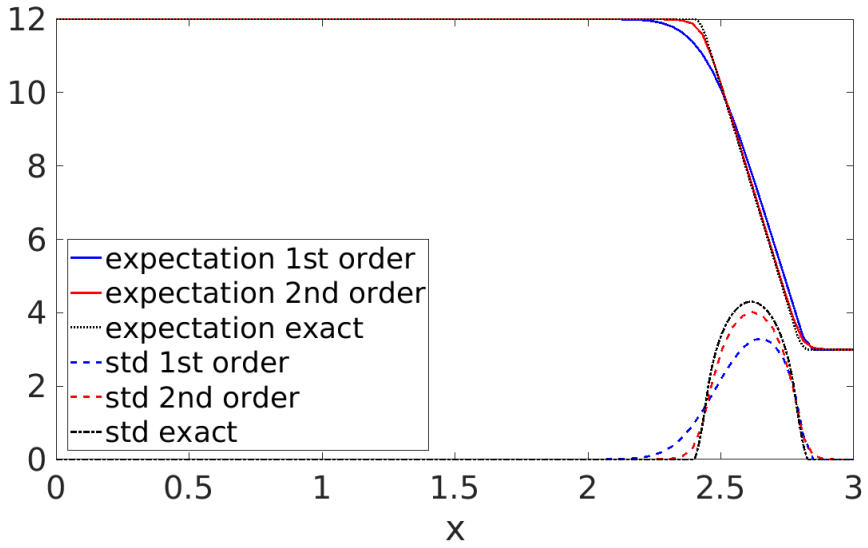


Figure 7.7. Expected value and standard deviation with and without limiters.

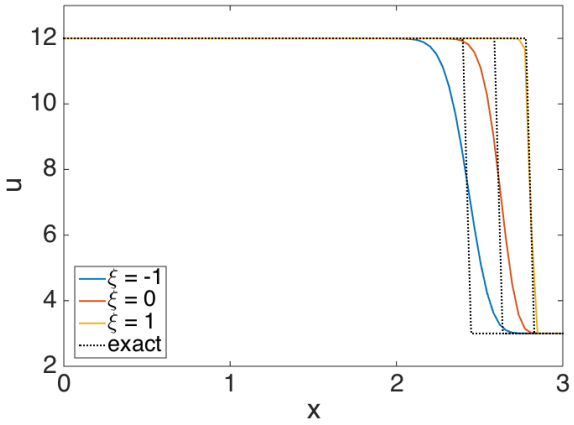
To underline the effects of artificial viscosity, we plot the solution when recalculating moments for first- and second-order spatial reconstructions for $\xi \quad \{-1, 0, 1\}$. Figure 7.8a shows that the solution is well resolved if $\xi = 1$. In the case of $\xi = -1$, the solution is smeared out, since the CFL condition does not allow the scheme to sharply capture shocks. In Figure 7.8b, we see that this effect is smaller when using second-order reconstructions.

8. Conclusion and Outlook

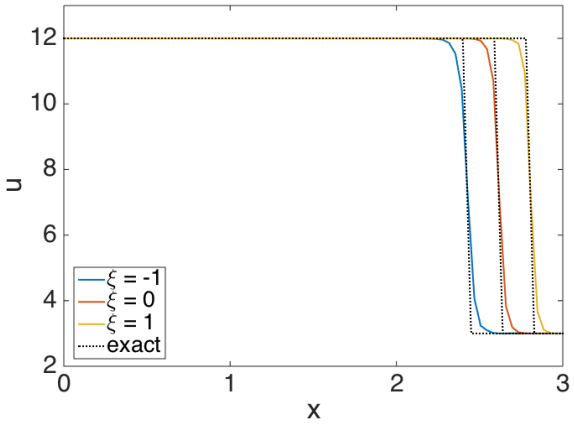
In this paper, we have investigated robust implementations of the IPM method for uncertain scalar hyperbolic conservation laws. The standard discretization of the IPM moment system can easily lead to nonrealizable moments and these nonrealizable moments cause the numerical solver to crash because in these cases the ansatz is undefined. This is especially true when the IPM bounds u_- and u_+ are chosen very close to the bounds of the true solution. In order to construct a second-order discretization of the IPM scheme that prevents such realizability problems, we investigated the numerical scheme in terms of the monotonicity of the underlying scheme for the original PDE. We derived two first-order schemes which preserve realizability: The first scheme makes use of a modified CFL condition and the second scheme recalculates moments from the inexact dual state. We also extended this second scheme to second order.

We also investigated the approximation properties of the IPM scheme using different entropies. By considering the entropy ansatz directly, we showed that the solution is not bounded due to properties of the entropy density s itself but rather its derivative s . This allowed us to use an entropy, which we called the bounded-barrier entropy, that takes finite values at the bounds u_- and u_+. The bounded-barrier entropy behaves more gracefully near the boundary values u_- and u_+, which we showed also leads to better solutions at intermediate values. This allowed us to take the IPM bounds to be the minimal and maximal value of the true solution, thus allowing the method to fulfill the exact maximum principle of the underlying PDE.

We applied our numerical schemes to the uncertain Burgers' equation as well as the uncertain advection equation. We observed that in contrast to solutions using the log-barrier entropy the solutions



(A) First order.



(B) Second order.

Figure 7.8. Solution evaluated at $\xi \quad \{-1, 0, 1\}$ with and without limiters.

calculated using the bounded-barrier entropy fulfill the maximum principle and are nonoscillatory, particularly at intermediate states.

We consider the IPM method a promising tool to treat uncertain hyperbolic equations which is a clear improvement over the stochastic-Galerkin method. In order to compete with the faster computation times of the stochastic-Galerkin method one should focus on accelerating the process of solving the dual problem, taking advantage of parallelizability (see for example [25, 10]), as well as higher-order schemes. An extension to higher-order schemes should be straightforward with bound-preserving limiters [24, 32].

References

[1] G. Alldredge, C. D Hauck, and A. L. Tits. High-order entropy-based closures for linear transport in slab geometry II: A computational study of the optimization problem. *SIAM Journal on Scientific Computing*, 34(4):B361–B391, 2012.

[2] G. W. Alldredge, C. D. Hauck, D. P. OĹeary, and A. L. Tits. Adaptive change of basis in entropy-based moment closures for linear kinetic equations. *Journal of Computational Physics*, 258:489–508, 2014.

[3] J. B. Bell, C. N. Dawson, and G. R. Shubin. An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74(1):1–24, 1988.

[4] C. Canuto and A. Quarteroni. Approximation results for orthogonal polynomials in Sobolev spaces. *Mathematics of Computation*, 38(157):67–86, 1982.

[5] K. M Case and P. F. Zweifel. *Linear transport theory*. Addison-Wesley Pub. Co., 1967.

[6] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of modern physics*, 15(1):1–89, 1943.

[7] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1):171–200, 1990.

[8] B. Després, G. Poëtte, and D. Lucor. *Robust Uncertainty Propagation in Systems of Conservation Laws with the Entropy Closure Method*, pages 105–149. Springer International Publishing, 2013.

[9] B. Dubroca and A. Klar. Half-moment closure for radiative transfer equations. *Journal of Computational Physics*, 180(2):584–596, 2002.

[10] C. K. Garrett, C. Hauck, and J. Hill. Optimization and large scale computation of an entropy-based moment closure. *Journal of Computational Physics*, 302:573–590, 2015.

[11] R. G Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover, 2003.

[12] D. Gottlieb and D. Xiu. Galerkin method for wave equations with uncertain coefficients. *Commun. Comput. Phys*, 3(2):505–518, 2008.

[13] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

[14] J.-L. Guermond, M. Nazarov, B. Popov, and Y. Yang. A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations. *SIAM Journal on Numerical Analysis*, 52(4):2163–2182, 2014.

[15] C. Hauck and R. McClarren. Positive P_N closures. *SIAM Journal on Scientific Computing*, 32(5):2603–2626, 2010.

[16] C. D. Hauck. High-order entropy-based closures for linear transport in slab geometry. *Commun. Math. Sci*, 9(1):187–205, 2011.

[17] H. Holden and N. H. Risebro. *Front tracking for hyperbolic conservation laws*, volume 152. Springer, 2015.

[18] J. Kusch. Uncertainty quantification for hyperbolic equations. *RWTH Aachen University*, pages 1–23, 2015.

[19] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag Basel, 1992.

[20] R. J. LeVeque. Nonlinear conservation laws and finite volume methods. In *Computational methods for astrophysical fluid flow*, pages 1–159. Springer, 1998.

[21] C. D. Levermore. Moment closure hierarchies for kinetic theories. *Journal of Statistical Physics*, 83(5-6):1021–1065, 1996.

[22] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. John Wiley and Sons, Inc., New York, NY, 1984.

[23] X.-D. Liu. A maximum principle satisfying modification of triangle based adapative stencils for the solution of scalar hyperbolic conservation laws. *SIAM journal on numerical analysis*, 30(3):701–716, 1993.

[24] X.-D. Liu and S. Osher. Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I. *SIAM Journal on Numerical Analysis*, 33(2):760–779, 1996.

[25] G. Poëtte, B. Després, and D. Lucor. Uncertainty quantification for systems of conservation laws. *Journal of Computational Physics*, 228(7):2443–2467, 2009.

[26] G. Poëtte, B. Després, and D. Lucor. Treatment of uncertain material interfaces in compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 200(1):284–308, 2011.

[27] G. Poëtte, B. Després, and D. Lucor. Uncertainty propagation for systems of conservation laws, high order stochastic spectral methods. In *Spectral and High Order Methods for Partial Differential Equations*, pages 293–305. Springer, 2011.

[28] G. C. Pomraning. *The Equations of Radiation Hydrodynamics*. Oxford, 1973.

[29] C.-W. Shu. Total-variation-diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084, 1988.

[30] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[31] D. Xiu and G. Em Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4927–4948, 2002.

[32] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.

[33] X. Zhang and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 2011.