# Entropy–Based Methods for Uncertainty Quantification of Hyperbolic Conservation Laws

**Martin Frank, Jonas Kusch, and Jannick Wolters**

**Abstract** Using standard intrusive techniques when solving hyperbolic conservation laws with uncertainties can lead to oscillatory solutions as well as non-hyperbolic moment systems. Entropy-based Stochastic Galerkin methods, on the other hand, guarantee hyperbolicity and entropy decay. A key challenge facing these methods is computational cost, since they require repeatedly solving a non-linear optimization problem. Furthermore, the spatial and temporal discretization needs to preserve realizability, meaning that the existence of a unique solution to the optimization problem must be ensured. We review strategies to guarantee realizability, which use a special choice of the numerical flux while considering errors from the optimization solve. Most importantly, we indicate how intrusive entropy-based closures can be made competitive. We show several numerical test cases and discuss the advantages and disadvantages of several uncertainty propagation methods.

## 1 Introduction

Hyperbolic equations play an important role in various research as well as industrial areas. The most common equations of this kind model the behavior of liquids, gases and plasmas and are thus widely used in the automotive and aerospace industry. Because of this popularity, many highly efficient and robust implementations for these models are available. The respective codes have shown to, for example,

M. Frank (✉) · J. Kusch · J. Wolters
Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344
Eggenstein-Leopoldshafen, Germany
e-mail: martin.frank@kit.edu

J. Kusch
e-mail: jonas.kusch@kit.edu

J. Wolters
e-mail: jannick.wolters@kit.edu

simulate the airflow around an airfoil very precisely, but only if the provided input data is identical or at least extremely close to the experimental setup. Any arising uncertainties in the input parameters, originating from e.g. measurement tolerances, imperfect information or modeling assumptions cannot be represented and thus lead to differences in the results of experiments and simulations. Therefore, propagating these uncertainties through complex partial differential equations has become an important topic in the last decades.

We consider a parameterized system of hyperbolic equations of the form

$$\partial_t \boldsymbol{u}(t, \boldsymbol{x}, \boldsymbol{\xi}) + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}(t, \boldsymbol{x}, \boldsymbol{\xi})) = \boldsymbol{0} \ \text{ in } D \ , \tag{1a}$$

$$\boldsymbol{u}(t = 0, \boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{u}_{\text{IC}}(\boldsymbol{x}, \boldsymbol{\xi}) \tag{1b}$$

with state variable $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^m$ depending on time $t \in \mathbb{R}^+$, spatial position $\boldsymbol{x} \in D \subset \mathbb{R}^d$ and uncertain parameter $\boldsymbol{\xi} \in \Theta \subseteq \mathbb{R}^p$. The physical flux is given by $\boldsymbol{f} : \mathcal{D} \to \mathbb{R}^{d \times m}$. Note, that for ease of notation, the uncertainties $\boldsymbol{\xi}$ here only enter through the initial condition, i.e. only the initial data is subject to randomness. Boundary conditions are omitted for now as they are specific to the studied problem and will be supplied for the individual test cases in the later sections. We assume that all random parameters are independent with a joint probability density function $f_{\Xi} = \prod_{i=1}^p f_{\Xi,i}(\xi_i)$.

As the solution of (1a) is now subject to randomness, one is often interested in determining the statistical moments of the solution, where the first and second order moments, i.e. the mean and variance of $\boldsymbol{u}$, given by

$$\mathrm{E}[\boldsymbol{u}] = \langle \boldsymbol{u} \rangle, \qquad \mathrm{Var}[\boldsymbol{u}] = \langle (\boldsymbol{u} - \mathrm{E}[\boldsymbol{u}])^2 \rangle$$

are usually most interesting. We define the bracket operator above as

$$\langle \cdot \rangle := \int_{\Theta} \cdot f_{\Xi}(\boldsymbol{\xi}) \, d\xi_1 \dots d\xi_p.$$

Generally, the methods of Uncertainty Quantification (UQ) can be divided into two groups, intrusive and non-intrusive, meaning the methods either require an intrusive change of an existing deterministic solver, or the existing code can be repurposed in a black-box manner. Several textbooks on UQ have appeared in recent years [5, 10, 27, 28, 35, 40], and we refer the reader to these textbooks for a general overview and references to the original works. In this paper, we nevertheless want to shed some light on how these methods can be expected to perform when applied to hyperbolic conservation laws. We will also make some general statements that we believe are not well-known in some part of the literature. As it has been shown [33], the stochastic Galerkin method cannot be applied directly to conservation laws because it is prone to yield oscillatory solutions that might result in the loss of hyperbolicity and e.g. negative densities. We then discuss in detail the Intrusive Polynomial Moment (IPM) method [33], which can be seen as a generalization to the stochastic Galerkin method.

This method is based on entropy minimization, and choosing a suitable entropy guarantees hyperbolicity. On the other hand, the method comes at the cost of solving an optimization problem at every point in time for every spatial cell.

The IPM method has been inspired by entropy-based closures in kinetic theory. In fact, the moment closure problem for a kinetic equation in velocity space is very similar to treating uncertain parameters [20]. In the case of kinetic equations, the techniques to put entropy-based closures into practice have been refined in a series of papers [1–3, 9, 14, 29], and the application to UQ draws from this experience. In this paper, we discuss realizability-preserving spatial discretizations, and acceleration techniques to solve the IPM system more efficiently. This review is based on the papers [19, 20, 23].

## 2 Why Galerkin-Type Intrusive Methods?

For the following discussion, we assume that we want to approximate the expected value

$$\mathrm{E}[\boldsymbol{u}] = \langle \boldsymbol{u} \rangle = \int_{\Theta} \boldsymbol{u} f_{\Xi} \, d\xi_1 \ldots d\xi_p$$

of the solution $\boldsymbol{u}$ at a given time $t$ as a function of $\boldsymbol{x}$. Since the expected value is an integral of the solution against the probability density function, all non-intrusive UQ methods can be understood and analyzed as numerical quadrature rules

$$\mathrm{E}[\boldsymbol{u}](t, x) \approx \sum_{k=1}^{N} w^k \boldsymbol{u}(t, \boldsymbol{x}, \boldsymbol{\xi}^k) \ .$$

- Monte-Carlo (MC) methods sample $\boldsymbol{\xi}^k$ from the probability density function $f_{\Xi}$ and take $w^k = \frac{1}{N}$.
- Number-theoretic/Quasi-Monte Carlo (QMC) methods for uniformly distributed random variables use a low-discrepancy sequence for $\boldsymbol{\xi}^k$ and again $w^k = \frac{1}{N}$.
- Tensorized quadrature rules take one-dimensional (e.g. Gaussian) quadrature rules for each random input $\xi_i$. The grid for the $\boldsymbol{\xi}^k$ is defined as a Cartesian product of the one-dimensional grids and the weights are products of the one-dimensional weights.
- Sparse grid quadrature rules use nodes $\boldsymbol{\xi}^k$ on a (e.g. Smolyak) sparse grid and weights that come from nested quadrature rules (e.g. Clenshaw-Curtis).

In the UQ literature, these methods are discussed based on their error formulas, and the so-called *curse of dimensionality* is often mentioned. If $\boldsymbol{u}$ denotes the true expected value and $\boldsymbol{u}_N$ its approximation with $N$ nodes/samples, then the methods behave in the following way:

- The MC error is determined by the root mean square error $E[(\boldsymbol{u} - \boldsymbol{u}_N)^2]^{1/2} = V_{MC}(\boldsymbol{u})N^{-1/2}$.

- Multi-level Monte Carlo (MLMC) methods use control variates to make the constant $V(\boldsymbol{u})$ smaller [11].
- The QMC error typically behaves like $|\boldsymbol{u} - \boldsymbol{u}_N| \leq V_{QMC}(\boldsymbol{u})(\log N)^p N^{-1}$ [6].
- The tensorized grid error has the form $|\boldsymbol{u} - \boldsymbol{u}_N| \leq V_{\text{tens}}(\boldsymbol{u}) N^{-\alpha/p}$.
- The sparse grid quadrature error behaves like $|\boldsymbol{u} - \boldsymbol{u}_N| \leq V_{\text{sparse}}(\boldsymbol{u})(\log N)^p N^{-\beta}$.

In the latter two cases, $\alpha$ and $\beta$ are related to the differentiability of $\boldsymbol{u}$ with respect to $\boldsymbol{\xi}$. All error constants $V$ depend on $\boldsymbol{u}$ and certain of its derivatives. For instance, in the case of sparse grids, $\boldsymbol{u}$ has to be in a certain Sobolev space with mixed higher-order derivatives. For an overview and a critical discussion of the assumptions on the solution we refer the reader to the excellent paper [38]. In the mathematical UQ literature, the *curse of dimensionality* is usually defined as an effective decay of the convergence rate when it is measured in the total number of nodes $N$ and when the dimension $p$ is increased. This is clearly the case for tensorized grids. But one should also note that for both QMC and sparse grid quadrature the decaying term dominates the log term only if $N \gg 2^p$ ($N \gg 2^{p/\beta}$ respectively). These methods therefore only mitigate the curse. Finally, one often finds the statement that MC methods do not suffer from the curse, because the convergence rate is independent of the dimension $p$. However, MC methods might be impractical in high dimensions. This can be seen from the simple example (that every reader can easily try) of approximating the volume of the unit sphere in $p$ dimensions: Draw a uniform sample in $[-1, 1]^p$ and determine if the sampled node is inside the unit sphere. Then the ratio of the points inside the sphere to the total number of samples converges to the volume divided by $2^p$. For $p = 20$, MC with 100 million samples will not produce any significant digit. The reason is that the volume of the sphere becomes so small that it becomes almost impossible to draw a sample within the sphere. In other words, the error constant increases rapidly with dimension $p$. It should be noted, however, that all of these methods are embarrassingly parallel because uncoupled problems need to be solved.

Intrusive methods on the other hand do not rely on any form of quadrature, but rather derive a system of equations that describes the time evolution of the moments directly. The resulting system can then be solved with classical numerical methods for deterministic equations. In contrast to non-intrusive methods this does not decouple the problem. In the stochastic Galerkin (SG) method, the solution $\boldsymbol{u}$ is expanded in a series of polynomial basis functions $\varphi_i : \Theta \to \mathbb{R}$, such that for the multi-index $i = (i_1, \cdots, i_p)$ we have $|i| := \sum_{k=1}^{p} |i_k| \leq N$. The usual choice for these functions $\varphi_i$ are orthonormal polynomials with respect to the probability distribution function, i.e. $\langle \varphi_i \varphi_j \rangle = \prod_{n=1}^{p} \delta_{i_n j_n}$. This yields the so called generalized polynomial chaos (gPC) expansion

$$\mathcal{U}(\hat{\boldsymbol{u}}; \boldsymbol{\xi}) := \sum_{|i| \leq N} \hat{\boldsymbol{u}}_i \varphi_i(\boldsymbol{\xi}) = \hat{\boldsymbol{u}}^T \boldsymbol{\varphi}(\boldsymbol{\xi}) . \tag{2}$$

The unknown, but deterministic expansion coefficients $\hat{\boldsymbol{u}}_i \in \mathbb{R}^m$ are called moments. For a more compact notation, we collect these moments in the moment matrix $\hat{\boldsymbol{u}}$. This matrix holds all moments for which $|i| \leq N$ holds. Therefore, $\hat{\boldsymbol{u}}$ is defined

as $\hat{\boldsymbol{u}} := (\hat{\boldsymbol{u}}_i)_{|i| \leq N} \in \mathbb{R}^{M \times m}$ with corresponding basis functions $\boldsymbol{\varphi} := (\varphi_i)_{|i| \leq N} \in \mathbb{R}^M$. The total number of basis functions for which $|i| \leq N$ holds is

$$M := \binom{N + p}{p} .$$

When the gPC approximation (2) is known, statistical quantities of interest can be computed as

$$\mathrm{E}[\mathcal{U}(\hat{\boldsymbol{u}})] = \hat{\boldsymbol{u}}_0 , \quad \mathrm{Var}[\mathcal{U}(\hat{\boldsymbol{u}})] = \mathrm{E}[\mathcal{U}(\hat{\boldsymbol{u}})^2] - \mathrm{E}[\mathcal{U}(\hat{\boldsymbol{u}})]^2 = \left( \sum_{i=1}^{N} \hat{u}_{\ell i}^2 \right)_{\ell = 1, \cdots, m} .$$

The SG moment system is obtained by plugging the gPC ansatz (2) into the stochastic problem (1a) and projecting the resulting residual to zero (Galerkin projection), which yields the system

$$\partial_t \hat{\boldsymbol{u}}_i(t, \boldsymbol{x}) + \nabla \cdot \langle \boldsymbol{f}(\mathcal{U}(\hat{\boldsymbol{u}}))\varphi_i \rangle = \boldsymbol{0} , \tag{3a}$$

$$\hat{\boldsymbol{u}}_i(t = 0, \boldsymbol{x}) = \langle \boldsymbol{u}_{\mathrm{IC}}(\boldsymbol{x})\varphi_i \rangle . \tag{3b}$$

As mentioned previously, the main caveat of the method is that the moment system is not necessarily hyperbolic and thus not applicable to every problem. We will investigate this thoroughly in the following sections, but at this point we want to discuss why one should be interested in an intrusive method like SG at all. Putting SG into practice requires working with the model and new code. Additionally, the trivial parallelism of non-intrusive methods is lost. A statement one often finds in the UQ literature is that one should use SG because it has spectral convergence. This means that the convergence rate of the method only depends on the smoothness of the function. Moreover, if this smoothness is large or even infinite (i.e. the solution possesses derivatives of orders up to infinity) then the curse of dimensionality can be overcome. However, both Gauss and Clenshaw-Curtis quadrature also show spectral convergence [39], so if a function is smooth enough those methods can be used as well.

On the plus side, because sparse grids rely on nested quadrature rules, and similar to modal versus nodal DG methods, SG reaches the same formal accuracy with fewer unknowns. Furthermore, in many cases the expected value of a solution of a hyperbolic system is more smooth and does not have shocks (examples can be found in Sect. 6). Although this is not true in general [37], one often does not have to use a high-resolution shock-capturing scheme. Two further advantages will be utilized in this paper: Whereas collocation methods use a global grid in the uncertain parameters, for intrusive methods one can enrich the discretization of the parameter space adaptively. Furthermore, especially for the IPM method one can iterate faster into steady state. Given these potential advantages we argue that although intrusive methods have shortcomings it is worthwhile to study them, especially in the context of hyperbolic conservation laws.

# 3 Hyperbolic Conservation Laws and the IPM Method

After introducing hyperbolic conservation laws and the concept of entropy, this section is focused on the derivation of the IPM method.

## 3.1 Hyperbolic Conservation Laws and Entropy Variables

Our numerical discretization of the random space should preserve certain properties of hyperbolic equations. Although they are well-known, we briefly summarize them in the following to fix notation. Ignoring uncertainties for the time being, to characterize hyperbolicity we put the conservative form (1a) into its *quasi-conservative form*. Defining the flux Jacobians $A_j := \nabla_u f_j \in \mathbb{R}^{m \times m}$, the system (1a) can be rewritten as

$$\partial_t u + \sum_{j=1}^{d} A_j(u) \partial_{x_j} u = 0 .$$ (4)

Denoting the flux Jacobian into direction $w \in \mathbb{R}^d$ by

$$A(u, w) := \sum_{j=1}^{d} A_j w_j ,$$

we call a system *hyperbolic*, if the flux Jacobian $A(u, w)$ has only real eigenvalues $\lambda_k(u, w)$ for $k = 1, \cdots, m$ with a complete family of eigenvectors $r_k(u, w)$ for all states $u \in \mathcal{D}$ and every direction $w \in \mathbb{R}^d$ with $\|w\| = 1$. Note that for one spatial dimension, i.e. $d = 1$, the direction is $w = 1$ and therefore hyperbolicity holds if the flux Jacobian $\nabla_u f$ is diagonalizable with real eigenvalues.

Hyperbolic problems tend to form shocks, in which case the original system can no longer be solved. Therefore, the concept of *weak solutions* has been introduced, which tests the original problem against smooth basis functions with compact support and then moves derivatives from the solution onto these basis functions [24, Chapter 3.4]. Unfortunately, weak solutions are not unique and can show non-physical behavior. Hence, one is left with having to pick physical meaningful solutions from possible weak solution candidates. This motivates a further concept, called the entropy solution [24, Chapter 3.8.1]. Note that in the case of scalar equations, the entropy solution is actually unique under certain smallness assumptions [16, Chapter 2.4]. Let us first introduce the entropy:

**Definition 1** Let $\mathcal{D}$ be convex. Then a convex function $s : \mathcal{D} \to \mathbb{R}$ is called an *entropy* for the conservation Eqs. (1a) if there exist $d$ functions $\tilde{F}_j : \mathcal{D} \to \mathbb{R}$, called *entropy fluxes*, which fulfill the *integrability condition*

$$\nabla_u s(u) \nabla_u f_j(u) = \nabla_u \tilde{F}_j(u) , \quad j = 1, \cdots, d .$$ (5)

For classical solutions, the integrability condition ensures conservation of entropy: By multiplying $\nabla_{\boldsymbol{u}} s(\boldsymbol{u})$ from the left with the original Eq. (1a), we get with the integrability condition (5) as

$$\partial_t s(\boldsymbol{u}) + \sum_{j=0}^d \partial_{x_j} \tilde{F}_j(\boldsymbol{u}) = 0 . \tag{6}$$

Since (6) is again in conservation form, the entropy is conserved at smooth solutions and the functions $\tilde{F}_j$ are the flux functions of the entropy balance law (6).

If $\boldsymbol{u}$ is a weak solution, which fulfills

$$\partial_t s(\boldsymbol{u}) + \sum_{j=0}^d \partial_{x_j} \tilde{F}_j(\boldsymbol{u}) \leq 0 \tag{7}$$

in a weak sense for all admissible entropies $s$, then $\boldsymbol{u}$ is called an entropy solution. Note that opposed to the entropy used in thermodynamics, the mathematical entropy $s$ is dissipated in time: By integrating (7) over the spatial domain, while assuming that the entropy fluxes are zero at the boundary, we obtain

$$\frac{d}{dt} \int_D s(\boldsymbol{u}) \, d\boldsymbol{x} \leq 0 . \tag{8}$$

The notion of entropy is closely related to hyperbolicity, which can be shown with the help of the *entropy variables*

$$\boldsymbol{v} = \nabla_{\boldsymbol{u}} s(\boldsymbol{u})^T \in \mathbb{R}^m . \tag{9}$$

If $s$ is strictly convex, the mapping $\boldsymbol{v}(\boldsymbol{u})$ is one-to-one and the solution $\boldsymbol{u}$ can be represented in terms of entropy variables as $\boldsymbol{u} : \mathbb{R}^m \to \mathbb{R}^m$ with $\boldsymbol{u}(\boldsymbol{v}) = (\nabla_{\boldsymbol{u}} s)^{-1}(\boldsymbol{v})$.[1] A change from the conserved quantities $\boldsymbol{u}$ to their corresponding entropy variables can be performed to put (1a) in its *symmetric form*

$$\partial_t \boldsymbol{u}(\boldsymbol{v}) + \sum_{j=1}^d \partial_{x_j} \boldsymbol{g}_j(\boldsymbol{v}) = \boldsymbol{0} , \tag{10}$$

where the flux with respect to the entropy variables has been denoted by $\boldsymbol{g}_j$, i.e.

$$\boldsymbol{g}_j(\boldsymbol{v}) := \boldsymbol{f}_j(\boldsymbol{u}(\boldsymbol{v})) \quad \text{with } j = 1, \cdots, d . \tag{11}$$

Our goal is to check hyperbolicity, i.e. (10) needs to be brought into its quasi-conservative form (4). Applying the chain rule results in

---

[1] Note that we have prescribed $\boldsymbol{u}$ to be in $\mathbb{R}^m$, i.e. strictly speaking we have $\boldsymbol{u}(\boldsymbol{v}) = (\nabla_{\boldsymbol{u}} s)^{-T}(\boldsymbol{v})$.

$$\boldsymbol{H}(\boldsymbol{v})\partial_t \boldsymbol{v} + \sum_{j=1}^{d} \boldsymbol{B}_j(\boldsymbol{v})\partial_{x_j}\boldsymbol{v} = \boldsymbol{0} , \qquad (12)$$

with

$$\boldsymbol{H}(\boldsymbol{v}) = \nabla_{\boldsymbol{v}}\boldsymbol{u}(\boldsymbol{v}) \qquad \text{and} \qquad \boldsymbol{B}_j(\boldsymbol{v}) = \nabla_{\boldsymbol{v}}\boldsymbol{g}_j(\boldsymbol{v}) . \qquad (13)$$

Note that $\boldsymbol{H}(\boldsymbol{v}) = \left(\nabla_{\boldsymbol{u}}^2 s(\boldsymbol{u})\right)^{-1}$, which can be checked by differentiating $\nabla_{\boldsymbol{u}} s(\boldsymbol{u}(\boldsymbol{v})) = \boldsymbol{v}$ with respect to $\boldsymbol{v}$. Therefore, $\boldsymbol{H}(\boldsymbol{v})$ is symmetric positive definite and can therefore be rewritten as $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$, where $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ is orthonormal and $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times m}$ is a diagonal matrix with positive entries. Consequently, the regular, symmetric matrix $\boldsymbol{H}^{1/2} := \boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T$ exists. Multiplying (12) with $\boldsymbol{H}^{-1} = \boldsymbol{H}^{-1/2}\boldsymbol{H}^{-1/2}$ from the left results in the system

$$\partial_t \boldsymbol{v} + \sum_{j=1}^{d} \boldsymbol{H}^{-1/2}\boldsymbol{H}^{-1/2}\nabla_{\boldsymbol{v}}\boldsymbol{g}_j(\boldsymbol{v})\boldsymbol{H}^{-1/2}\boldsymbol{H}^{1/2}\partial_{x_j}\boldsymbol{v} = \boldsymbol{0} . \qquad (14)$$

It remains to check under which conditions the flux Jacobian of this system is diagonalizable with real eigenvalues. Since $\boldsymbol{H}^{-1/2}$ is symmetric, symmetry of $\boldsymbol{B}_j$ suffices to show symmetry of $\boldsymbol{H}^{-1/2}\nabla_{\boldsymbol{v}}\boldsymbol{g}_j(\boldsymbol{v})\boldsymbol{H}^{-1/2}$. Multiplying this matrix with $\boldsymbol{H}^{-1/2}$ from the left and $\boldsymbol{H}^{1/2}$ from the right is a similarity transformation and therefore does not change eigenvalues. Hence when $\boldsymbol{B}_j$ is symmetric, the system (14) is diagonalizable with real eigenvalues and therefore hyperbolic. This can be ensured via the concept of entropy.

**Theorem 1** *The matrices $\boldsymbol{B}_j$ are symmetric iff the integrability condition* (5) *holds.*

*Proof* See e.g. [36]. □

In the case of scalar equations, all convex functions can be used as entropies. In particular, a family of entropies, which is also called the Kružkov entropy [18], given by

$$s(u) = |u - k| \qquad \text{for all } k \in \mathbb{R}$$

fulfills the integrability condition for the entropy flux

$$\tilde{F}_j(u) = \text{sgn}(u - k)(f_j(u) - f_j(k)) .$$

This family of entropies can be employed to derive several solution properties for scalar equations. One of these properties is the *maximum–principle*

$$\|u(t, \cdot)\|_{L^\infty(D)} \le \|u_{\text{IC}}\|_{L^\infty(D)} , \qquad (15)$$

which guarantees bounds on the solution imposed by its initial condition.

## *3.2 The Intrusive Polynomial Moment Method*

Let us now consider a system of hyperbolic conservation laws of the form (1a) and move back to the discretization of the random domain. As discussed earlier, the polynomial ansatz of stochastic-Galerkin does not necessarily preserve hyperbolicity. A generalization of stochastic-Galerkin, which ensures hyperbolicity is the Intrusive Polynomial Moment (IPM) method [33], which in the field of kinetic theory is known as an entropy closure, see e.g. [25]. Instead of expanding the conserved variables $\boldsymbol{u}$ with polynomials as done by SG, the IPM method performs such an expansion on the entropy variables (9). Hence, substituting the entropy variables $\boldsymbol{v} = \nabla_{\boldsymbol{u}} s(\boldsymbol{u})^T$ into (1a) yields

$$\partial_t \boldsymbol{u}(\boldsymbol{v}(t, \boldsymbol{x}, \boldsymbol{\xi})) + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{v}(t, \boldsymbol{x}, \boldsymbol{\xi}))) = \boldsymbol{0} \,, \tag{16}$$

where again $\boldsymbol{u} : \mathbb{R}^m \to \mathbb{R}^m$ with $\boldsymbol{u}(\boldsymbol{v}) = \nabla_{\boldsymbol{u}} s(\boldsymbol{u})$. Now, a finite dimensional representation of the entropy variables is obtained by an expansion in terms of gPC polynomials, i.e.

$$\boldsymbol{v}(t, \boldsymbol{x}, \boldsymbol{\xi}) \approx \boldsymbol{v}_N(t, \boldsymbol{x}, \boldsymbol{\xi}) := \sum_{|i| \le N} \hat{\boldsymbol{v}}_i(t, \boldsymbol{x}) \varphi_i(\boldsymbol{\xi}) = \hat{\boldsymbol{v}}(t, \boldsymbol{x})^T \boldsymbol{\varphi}(\boldsymbol{\xi}) \,, \tag{17}$$

where the *entropic expansion coefficients* (also called *dual variables*) $\hat{\boldsymbol{v}}_i \in \mathbb{R}^m$ are collected in the matrix $\hat{\boldsymbol{v}} := (\hat{\boldsymbol{v}}_i)_{i \le |N|} \in \mathbb{R}^{M \times m}$. Replacing the exact entropy variables inside the original problem (16) by this expansion, we obtain

$$\partial_t \boldsymbol{u}\left(\hat{\boldsymbol{v}}(t, \boldsymbol{x})^T \boldsymbol{\varphi}(\boldsymbol{\xi})\right) + \nabla \cdot \boldsymbol{f}\left(\boldsymbol{u}\left(\hat{\boldsymbol{v}}(t, \boldsymbol{x})^T \boldsymbol{\varphi}(\boldsymbol{\xi})\right)\right) = \tilde{\boldsymbol{r}}(t, \boldsymbol{x}, \boldsymbol{\xi}) \,. \tag{18}$$

Similar to stochastic-Galerkin, the residual $\tilde{\boldsymbol{r}}$ is again projected to zero, yielding

$$\partial_t \left\langle \boldsymbol{u}\left(\hat{\boldsymbol{v}}(t, \boldsymbol{x})^T \boldsymbol{\varphi}\right) \varphi_i \right\rangle + \nabla \cdot \left\langle \boldsymbol{f}\left(\boldsymbol{u}\left(\hat{\boldsymbol{v}}(t, \boldsymbol{x})^T \boldsymbol{\varphi}\right)\right) \varphi_i \right\rangle = \boldsymbol{0} \tag{19}$$

for $|i| \le N$. The moments belonging to the dual variables $\hat{\boldsymbol{v}}$ are now given by

$$\hat{\boldsymbol{u}}_i(\hat{\boldsymbol{v}}) = \left\langle \boldsymbol{u}\left(\hat{\boldsymbol{v}}^T \boldsymbol{\varphi}\right) \varphi_i \right\rangle \qquad \text{for } |i| \le N \,. \tag{20}$$

This mapping, i.e. $\hat{\boldsymbol{u}} : \mathbb{R}^{M \times m} \to \mathcal{R} \subset \mathbb{R}^{M \times m}$ is one-to-one, meaning that similar to $\boldsymbol{v}(\boldsymbol{u})$, we can define a function $\hat{\boldsymbol{v}}(\hat{\boldsymbol{u}})$ with $\hat{\boldsymbol{v}} : \mathcal{R} \to \mathbb{R}^{M \times m}$. Making use of this mapping as well as the definition of the moments in (19) yields the IPM system

$$\partial_t \hat{\boldsymbol{u}}_i + \nabla \cdot \left\langle \boldsymbol{f}\left(\boldsymbol{u}\left(\hat{\boldsymbol{v}}(\hat{\boldsymbol{u}})^T \boldsymbol{\varphi}\right)\right) \varphi_i \right\rangle = \boldsymbol{0} \,, \quad \text{for } |i| \le N \,. \tag{21}$$

The IPM system posses several desirable properties. Especially, if the entropy $s(\boldsymbol{u})$ fulfills the integrability condition (5), the IPM system is hyperbolic:

**Theorem 2** *The IPM system can be brought into its symmetric form with symmetric positive definite temporal Jacobian and symmetric spatial Jacobian, if the entropy $s(\boldsymbol{u})$ fulfills the integrability condition* (5).

*Proof* In its symmetric form, the IPM system (19) reads

$$\hat{\boldsymbol{H}}(\hat{\boldsymbol{v}})\partial_t \hat{\boldsymbol{v}} + \sum_{j=1}^d \hat{\boldsymbol{B}}_j(\hat{\boldsymbol{v}})\partial_{x_j}\hat{\boldsymbol{v}} = \boldsymbol{0} \;, \tag{22a}$$

$$\text{with} \qquad \hat{\boldsymbol{H}}(\hat{\boldsymbol{v}}) := \left\langle \nabla_{\boldsymbol{v}} \boldsymbol{u}(\boldsymbol{v}_N) \otimes \boldsymbol{\varphi}\boldsymbol{\varphi}^T \right\rangle \;, \tag{22b}$$

$$\hat{\boldsymbol{B}}_j(\hat{\boldsymbol{v}}) := \left\langle \nabla_{\boldsymbol{u}} \boldsymbol{f}_j(\boldsymbol{u}(\boldsymbol{v}_N))\nabla_{\boldsymbol{v}} \boldsymbol{u}(\boldsymbol{v}_N) \otimes \boldsymbol{\varphi}\boldsymbol{\varphi}^T \right\rangle \;. \tag{22c}$$

Here, we abuse notation by defining the multiplication of $\hat{\boldsymbol{H}} \in \mathbb{R}^{m \cdot M \times m \cdot M}$ with $\boldsymbol{y} \in \mathbb{R}^{M \times m}$ by

$$\left(\hat{\boldsymbol{H}} \cdot \boldsymbol{y}\right)_{li} := \sum_{l'=1}^m \sum_{i'=1}^M \hat{H}_{(l-1)m+i,(l'-1)m+i'}\, y_{l'i'} \;.$$

The same holds for the multiplication with $\hat{\boldsymbol{B}}_j$. As done for (12), if we can ensure $\hat{\boldsymbol{H}}$ being symmetric positive definite and $\hat{\boldsymbol{B}}_j$ symmetric, we know that the IPM system is hyperbolic. Obviously, $\hat{\boldsymbol{H}}$ is symmetric. Multiplication with $\hat{\boldsymbol{v}} \in \mathbb{R}^{M \times m}$ from both sides gives

$$\hat{\boldsymbol{v}}^T \hat{\boldsymbol{H}} \hat{\boldsymbol{v}} = \left\langle \boldsymbol{v}_N^T \nabla_{\boldsymbol{v}} \boldsymbol{u}(\boldsymbol{v}_N)\boldsymbol{v}_N \right\rangle > 0 \;,$$

where we use that $\nabla_{\boldsymbol{v}} \boldsymbol{u} = \boldsymbol{H}$ is symmetric positive definite as done in (12). It remains to show symmetry of $\hat{\boldsymbol{B}}_j$ for all $j = 1, \cdots, d$. Using the definition of $\boldsymbol{B}_j$ from (13), we can rewrite (22c) as

$$\hat{\boldsymbol{B}}_j(\hat{\boldsymbol{v}}) := \left\langle \boldsymbol{B}_j(\boldsymbol{v}_N) \otimes \boldsymbol{\varphi}\boldsymbol{\varphi}^T \right\rangle \;.$$

By Theorem 1, we know that $\boldsymbol{B}_j$ is symmetric, from which we can conclude symmetry of $\hat{\boldsymbol{B}}_j$. □

Recall that solving the IPM system requires the mapping $\hat{\boldsymbol{v}}(\hat{\boldsymbol{u}})$, i.e. a mapping from the moments to the dual variables. This mapping can be defined by inverting the dual variables to moments map (20). The inverse exists, since the Jacobian of $\hat{\boldsymbol{u}}(\hat{\boldsymbol{v}})$ is $\nabla_{\hat{\boldsymbol{v}}} \hat{\boldsymbol{u}}(\hat{\boldsymbol{v}}) = \hat{\boldsymbol{H}}(\hat{\boldsymbol{v}})$ which is positive definite, i.e. the dual variables to moments map is strictly monotonically increasing. Unfortunately, the inversion can generally not be performed analytically. In this case one needs to determine $\hat{\boldsymbol{v}}$ by solving the non-linear system of equations

$$\left\langle \boldsymbol{u} \left( \hat{\boldsymbol{v}}^T \boldsymbol{\varphi} \right) \boldsymbol{\varphi}^T \right\rangle^T = \hat{\boldsymbol{u}} \tag{23}$$

for a given moment vector $\hat{\boldsymbol{u}}$ numerically. This task is commonly performed by reformulating (23) as a root-finding problem

$$\mathcal{G}(\hat{\boldsymbol{v}}; \hat{\boldsymbol{u}}) \overset{!}{=} \boldsymbol{0}$$

with

$$\mathcal{G}(\boldsymbol{w}; \hat{\boldsymbol{u}}) := \left\langle \boldsymbol{u}\left(\boldsymbol{w}^T \boldsymbol{\varphi}\right) \boldsymbol{\varphi}^T \right\rangle^T - \hat{\boldsymbol{u}} \ . \tag{24}$$

Here, one often uses Newton's method to determine the root of $\mathcal{G}$. Then, with $\nabla_{\boldsymbol{w}} \mathcal{G}(\boldsymbol{w}; \hat{\boldsymbol{u}}) = \hat{\boldsymbol{H}}(\boldsymbol{w})$ a Newton update takes the form $\boldsymbol{d} : \mathbb{R}^{M \times m} \times \mathbb{R}^{M \times m} \to \mathbb{R}^{M \times m}$ with

$$\boldsymbol{d}(\boldsymbol{w}, \hat{\boldsymbol{u}}) := \boldsymbol{w} - \hat{\boldsymbol{H}}(\boldsymbol{w})^{-1} \cdot \mathcal{G}(\boldsymbol{w}; \hat{\boldsymbol{u}}) \ . \tag{25}$$

The function $\boldsymbol{d}$ will in the following be called dual iteration function. Now, the Newton iteration for an input moment vector $\hat{\boldsymbol{u}}$ is given by

$$\boldsymbol{w}^{(l+1)} = \boldsymbol{d}(\boldsymbol{w}^{(l)}, \hat{\boldsymbol{u}}) \ . \tag{26}$$

The exact dual state is then obtained by computing the fixed point of $\boldsymbol{d}$, meaning that one converges the iteration (26), i.e. $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}}(\hat{\boldsymbol{u}}) = \lim_{l \to \infty} \boldsymbol{d}(\boldsymbol{w}^{(l)}, \hat{\boldsymbol{u}})$. To obtain a finite number of iterations, a stopping criterion

$$\sum_{i=0}^{m} \left\| \mathcal{G}(\boldsymbol{w}^{(l)}; \hat{\boldsymbol{u}}) \right\| < \tau \tag{27}$$

is used, where $\tau > 0$ is a user determined parameter.

It remains to discuss the discretization of the spatial and time domain, which for ease of presentation, we perform for a scalar problem as well as a one dimensional spatial domain. When dividing the spatial domain into cells $[x_{j-1/2}, x_{j+1/2}]$ with $j = 1, \cdots, N_x$ and using discrete times $t_n$ with $n = 1, \cdots, N_t$, we can approximate the $i$-th order moment by

$$\hat{\boldsymbol{u}}_{ij}^n \simeq \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \hat{\boldsymbol{u}}_i(t_n, x) dx \ .$$

The full moment vector in cell $j$ at time $t_n$ is denoted by $\hat{\boldsymbol{u}}_j^n = (\hat{\boldsymbol{u}}_{0j}^n, \cdots, \hat{\boldsymbol{u}}_{Nj}^n)^T \in \mathbb{R}^{M \times m}$. Furthermore, the corresponding dual variables are denoted by

$$\hat{\boldsymbol{v}}_j^n := \hat{\boldsymbol{v}}\left(\hat{\boldsymbol{u}}_j^n\right) \ . \tag{28}$$

Then, a finite-volume scheme for the IPM system (21) can be written as

$$\hat{\boldsymbol{u}}_j^{n+1} = \hat{\boldsymbol{u}}_j^n - \frac{\Delta t}{\Delta x} \left( \boldsymbol{G}^*(\hat{\boldsymbol{v}}_j, \hat{\boldsymbol{v}}_{j+1}) - \boldsymbol{G}^*(\hat{\boldsymbol{v}}_{j-1}, \hat{\boldsymbol{v}}_j) \right) \ , \tag{29}$$

where $\boldsymbol{G}^* : \mathbb{R}^{M \times m} \times \mathbb{R}^{M \times m} \to \mathbb{R}^{M \times m}$ is the numerical flux which needs to be consistent with the physical flux of the IPM system. i.e. we must have $\boldsymbol{G}^*(\hat{\boldsymbol{v}}, \hat{\boldsymbol{v}}) = \left\langle \boldsymbol{f}\left(\boldsymbol{u}\left(\hat{\boldsymbol{v}}^T \boldsymbol{\varphi}\right)\right) \boldsymbol{\varphi}^T \right\rangle$. In order to evaluate the moments to dual variables map (28), we

can use the defined Newton iteration for the input moments $\hat{\boldsymbol{u}}_j^n$. Altogether, this yields the scheme from Algorithm 1.

---

**Algorithm 1** IPM algorithm

---
1: **for** $j = 0$ to $N_x + 1$ **do**
2:     $\hat{\boldsymbol{u}}_j^0 = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \langle u_{\text{IC}}(x, \cdot) \boldsymbol{\varphi} \rangle dx$
3: **for** $n = 0$ to $N_t$ **do**
4:     **for** $j = 0$ to $N_x + 1$ **do**
5:        **while** $\|\mathcal{G}(\boldsymbol{v}_j^{(m)}; \hat{\boldsymbol{u}}_j^n)\| > \tau$ **do**
6:           $\boldsymbol{v}_j^{(m+1)} \leftarrow \boldsymbol{d}(\boldsymbol{v}_j^{(m)}; \hat{\boldsymbol{u}}_j^n)$
7:           $m \leftarrow m + 1$
8:        $\hat{\boldsymbol{v}}_j^n \leftarrow \boldsymbol{v}_j^{(m+1)}$
9:     **for** $j = 1$ to $N_x$ **do**
10:       $\hat{\boldsymbol{u}}_j^{n+1} \leftarrow \hat{\boldsymbol{u}}_j^n - \frac{\Delta t}{\Delta x} \left[ \boldsymbol{G}^*(\hat{\boldsymbol{v}}_j, \hat{\boldsymbol{v}}_{j+1}) - \boldsymbol{G}^*(\hat{\boldsymbol{v}}_{j-1}, \hat{\boldsymbol{v}}_j) \right]$

---

## 4 Realizability-Preserving Spatial Discretization

In this section we further describe the concept of realizability and present a realizability-preserving discretization and improved version of Algorithm 1.

### 4.1 Realizability

As previously discussed, the IPM method and minimal entropy closures in general face several challenges. Besides increased computational costs, the IPM method cannot invert the mapping $\hat{\boldsymbol{u}} : \mathbb{R}^{M \times m} \to \mathcal{R} \subset \mathbb{R}^{M \times m}$ when the moment vector $\hat{\boldsymbol{u}}$ leaves the so-called realizable set $\mathcal{R}$, which results in a failure of the method [19]. To discuss this issue, we consider a scalar, one-dimensional conservation law of the form

$$\partial_t u(t, x, \xi) + \partial_x f(u(t, x, \xi)) = 0 , \tag{30a}$$

$$u(0, x, \xi) = u_{\text{IC}}(x, \xi) , \tag{30b}$$

i.e. $m$, $p$ and $d$ are equal to one. The following discussion is however valid for arbitrary dimensions and we make this simplification for ease of exposition. For scalar problems of the form (30), the solution fulfills the maximum–principle [16, Chapter 2.4]

$$\min_{x \in D, \xi \in \Theta} u_{\text{IC}}(x, \xi) \leq u(t, x, \xi) \leq \max_{x \in D, \xi \in \Theta} u_{\text{IC}}(x, \xi) ,$$

which ideally should be preserved by the discretization of the random domain. The IPM method at least enables one to impose a user-defined lower bound $u_-$ and upper bound $u_+$ on the solution by choosing an entropy $s(u)$, which takes infinite values when $u \notin (u_-, u_+)$. One such entropy is the log-barrier entropy [33]

$$s(u) = -\ln(u - u_-) - \ln(u_+ - u) . \tag{31}$$

Then, by the entropy dissipation property (8), the IPM solution $u(v_N) \equiv (s')^{-1}(v_N)$ will remain inside the interval $(u_-, u_+)$. Similarly, for systems such as the Euler equations, certain solution quantities such as positivity of density, energy and pressure can again be achieved by the choice of a suitable entropy. Recall, that the image of the dual variables to moments map (20) has been denoted by $\mathcal{R}$. This set is called *realizable set*. For entropies imposing solution bounds $u_-$ and $u_+$ it is given by

$$\mathcal{R} := \left\{ \hat{\boldsymbol{u}} \in \mathbb{R}^{N+1} \,\middle|\, \exists u : \Theta \to (u_-, u_+) \text{ such that } \hat{\boldsymbol{u}} = \langle u\boldsymbol{\varphi} \rangle \right\} . \tag{32}$$

When proposing numerical methods to solve the IPM system (21), it is crucial to prevent the moments generated by this method from leaving this set, since then, the dual variables to moments map cannot be inverted, i.e. the system (23) has no solution. In the following, we propose an algorithm which keeps the moments inside $\mathcal{R}$ and we will refer to this property as preserving realizability.

## *4.2 Realizability-Preserving Discretization*

The presented general Algorithm 1 will not necessarily preserve realizability, i.e. it generates moments $\hat{\boldsymbol{u}} \notin \mathcal{R}$. The two sources for this are the choice of the numerical flux as well as the fact that the system (23) cannot be solved exactly, i.e. the moments to dual variables map has errors. Let us first write down the realizability preserving algorithm presented in [19] and then discuss why it maintains $\hat{\boldsymbol{u}} \notin \mathcal{R}$. When again using $u : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ with $u(v) = (s')^{-1}(v)$, the chosen numerical flux is the *kinetic flux*

$$\boldsymbol{G}^*(\hat{\boldsymbol{v}}_\ell, \hat{\boldsymbol{v}}_r) = \left\langle f^*(u(\hat{\boldsymbol{v}}_\ell^T \boldsymbol{\varphi}), u(\hat{\boldsymbol{v}}_r^T \boldsymbol{\varphi}))\boldsymbol{\varphi} \right\rangle , \tag{33}$$

where $f^*(u_\ell, u_r)$ is a monotone flux for the underlying deterministic problem. Note that this choice of $\boldsymbol{G}^*$ is common in the field of kinetic theory, see e.g. [8, 13, 31, 32]. We assume that the original, deterministic scheme

$$H(u, v, w) = v - \frac{\Delta t}{\Delta x} \left( f^*(v, w) - f^*(u, v) \right) \tag{34}$$

keeps the solution inside the bounds $u_-$ and $u_+$, i.e. $H(u_{j-1}^n, u_j^n, u_{j+1}^n) \in [u_-, u_+]$ if all inputs are bounded by $u_-, u_+$. This can for example be achieved with monotone schemes or, for high order methods, with bound preserving limiters [4, 7, 12, 26,

41]. Note that since the integral in (33) can generally not be solved analytically, a quadrature rule must be employed to approximate the kinetic flux. It can however be shown that the resulting quadrature error will not influence realizability of the numerical scheme. Then, a realizability preserving implementation is given by

---

**Algorithm 2** Modified IPM algorithm

---

1: **for** $j = 0$ to $N_x + 1$ **do**
2:     $\hat{\boldsymbol{u}}_j^0 = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \langle u_{\text{IC}}(x, \cdot) \boldsymbol{\varphi} \rangle dx$
3: **for** $n = 0$ to $N_t$ **do**
4:     **for** $j = 0$ to $N_x + 1$ **do**
5:         **while** $\|\mathcal{G}(\boldsymbol{v}_j^{(m)}; \hat{\boldsymbol{u}}_j^n)\| > \tau$ **do**
6:             $\boldsymbol{v}_j^{(m+1)} \leftarrow \boldsymbol{d}(\boldsymbol{v}_j^{(m)}; \hat{\boldsymbol{u}}_j^n)$
7:             $m \leftarrow m + 1$
8:         $\bar{\boldsymbol{v}}_j^n \leftarrow \boldsymbol{v}_j^{(m+1)}$
9:         $\bar{\boldsymbol{u}}_j^n \leftarrow \left\langle u\left(\left(\bar{\boldsymbol{v}}_j^n\right)^T \boldsymbol{\varphi}\right) \boldsymbol{\varphi} \right\rangle$
10:    **for** $j = 1$ to $N_x$ **do**
11:        $\hat{\boldsymbol{u}}_j^{n+1} \leftarrow \bar{\boldsymbol{u}}_j^n - \frac{\Delta t}{\Delta x}\left[\boldsymbol{G}^*(\bar{\boldsymbol{v}}_j, \bar{\boldsymbol{v}}_{j+1}) - \boldsymbol{G}^*(\bar{\boldsymbol{v}}_{j-1}, \bar{\boldsymbol{v}}_j)\right]$

---

Here, we use $\bar{\boldsymbol{v}}$ and $\bar{\boldsymbol{u}}$ instead of $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{u}}$ to stress that these quantities are affected by the inexact Newton iteration. The main difference to Algorithm 1, besides the choice of the numerical flux, is the recalculation of the moment vector from the inexact dual variables $\bar{\boldsymbol{v}}$ in line 9. It can be shown that Algorithm 2 preserves realizability: To simplify notation, let us define the exact and inexact dual states $\Lambda := v_N = \hat{\boldsymbol{v}}^T \boldsymbol{\varphi}$ and $\bar{\Lambda} := \bar{\boldsymbol{v}}^T \boldsymbol{\varphi}$. Then, the inexact moments are given by

$$\bar{\boldsymbol{u}}_j^n = \left\langle u\left(\bar{\Lambda}_j^n\right) \boldsymbol{\varphi} \right\rangle \tag{35}$$

and the moment update becomes

$$\hat{\boldsymbol{u}}_j^{n+1} = \bar{\boldsymbol{u}}_j^n - \frac{\Delta t}{\Delta x}\left[\left\langle f^*\left(u\left(\bar{\Lambda}_{j-1}^n\right), u\left(\bar{\Lambda}_j^n\right)\right) \boldsymbol{\varphi} \right\rangle - \left\langle f^*\left(u\left(\bar{\Lambda}_j^n\right), u\left(\bar{\Lambda}_{j+1}^n\right)\right) \boldsymbol{\varphi} \right\rangle\right] . \tag{36}$$

Plugging the definition of the inexact moments (35) into the moment update (36) yields

$$\hat{\boldsymbol{u}}_j^{n+1} = \left\langle \left(u\left(\bar{\Lambda}_j^n\right) - \frac{\Delta t}{\Delta x}\left[f^*\left(u\left(\bar{\Lambda}_j^n\right), u\left(\bar{\Lambda}_{j+1}^n\right)\right) - f^*\left(u\left(\bar{\Lambda}_{j-1}^n\right), u\left(\bar{\Lambda}_j^n\right)\right)\right]\right) \boldsymbol{\varphi} \right\rangle$$
$$= \left\langle H(u(\bar{\Lambda}_{j-1}^n), u(\bar{\Lambda}_j^n), u(\bar{\Lambda}_{j+1}^n)) \boldsymbol{\varphi} \right\rangle .$$

Now, since the ansatz $u(\Lambda) = (s')^{-1}(\Lambda)$ only takes values in $[u_-, u_+]$, we have

$$H(u(\bar{\Lambda}_{j-1}^n), u(\bar{\Lambda}_j^n), u(\bar{\Lambda}_{j+1}^n)) \in [u_-, u_+]$$

for all $\xi \in \Theta$. Therefore, the time updated moments belong to an underlying function which is bounded by $u_-$ and $u_+$, i.e. we have $\hat{\boldsymbol{u}}_j^{n+1} \in \mathcal{R}$. The construction of the presented algorithm also guarantees that the CFL condition of the original scheme ensures stability. Note that the presented IPM scheme can be extended to systems, multi-dimensional problems and higher order methods, whenever a bound-preserving scheme (34) exists. Furthermore, when replacing all integrals by quadrature rules, the time updated moments will remain realizable, since they belong to a function which fulfills the prescribed bounds on the quadrature points. The main drawback of this strategy to preserve realizability is that it introduces a non-conservative error by recalculating moments. However, when using a Lipschitz continuous numerical flux $\boldsymbol{G}^*$, the error by recalculating moments is of order $\mathcal{O}(\tau)$ [19]. I.e. by choosing a sufficiently small stopping criterion for Newton's method the error from the recalculation step becomes negligibly small.

A second strategy, which does not require recomputing moments and therefore does not add such an error is choosing a modified CFL condition. Here, the main idea is to account for effects the error in $\Lambda$ has on the scheme by choosing a smaller time step size. Denoting this error by

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

a more restrictive CFL condition, which ensures realizability is given by

**Theorem 3** *Let us assume that the entropy ansatz only takes values in $(u_-, u_+)$ and the underlying numerical flux $f^*$ is monotone. If furthermore, the numerical optimizer enforces the stopping criterion*

$$\max_{\xi \in \Theta} \left\{ \max_{\Lambda \in \left[ \bar{\Lambda}_{j,min}^n, \bar{\Lambda}_{j,max}^n \right]} \frac{u'(\Lambda(\xi))}{u'(\Lambda(\xi) + \Delta \Lambda_j^n(\xi))} \right\} \le \gamma \, , \tag{37}$$

*with*

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

*the time updated moment vector $\hat{\boldsymbol{u}}_j^{n+1}$ is realizable under the modified CFL condition*

$$\gamma \frac{\Delta t}{\Delta x} \max_{u \in [u_-, u_+]} |f'(u)| \le 1 \, . \tag{38}$$

The proof of this theorem as well as an implementation strategy can be found in [19]. Due to its simplicity and efficient implementation of different discretization schemes, we will use the strategy of recalculating moments, i.e. Algorithm 2 in the following.

# 5 Accelerating the IPM Solution

In this section we describe two acceleration techniques, as presented in [23], which reduce the numerical effort of the IPM method.

## 5.1 Adaptivity

As previously mentioned, in contrast to non-intrusive methods, intrusive methods allow a more fine-grained control over the solution, as the uncertainties or more precisely their respective moments, are directly propagated through time and the corresponding quantities of interest (e.g. mean or variance) are not collected in a secondary step as in e.g. SC or MC methods. Using adaptivity, we try to avoid using high-order moment representations and corresponding high order quadrature rules in portions of the domain, where the quantities of interest are well-represented with low-order moments. As these lower-order moment bases result in non-linear system (23) that are easier to solve, this approach can significantly reduce overall runtimes. We use the discontinuity sensor described in [30] in the UQ context. To do this, the polynomial approximation at refinement level $\ell$ is defined as

$$\tilde{\boldsymbol{u}}_\ell := \sum_{|i| \leq M_\ell} \boldsymbol{u}_i \varphi_i \ .$$

We further define an indicator for a moment vector at level $\ell$ as

$$\boldsymbol{S}_\ell := \frac{\langle \left( \tilde{\boldsymbol{u}}_\ell - \tilde{\boldsymbol{u}}_{\ell-1} \right)^2 \rangle}{\langle \tilde{\boldsymbol{u}}_\ell^2 \rangle} \ . \tag{39}$$

Note, that a similar indicator has been used in [17] for intrusive methods in UQ. We use the first element in $\boldsymbol{S}_\ell$, i.e. the density $\rho$, to determine the refinement level. This regularity indicator is therefore computed for every cell at every timestep and the current refinement level is kept if the indicator lies in the interval $I_\delta := [\delta_-, \delta_+]$. If its value falls below $\delta_-$, the refinement level is decreased to the next lower refinement level and vice versa if the value exceeds $\delta_+$. See [23] for more details on the method.

## 5.2 One-Shot IPM

The second method is limited to steady state problems. In this case, we are interested in solving

$$\nabla \cdot \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\xi})) = \boldsymbol{0} \ \text{ in } D \tag{40}$$

with adequate boundary conditions. Then, the IPM moment system reads

$$\nabla \cdot \langle \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{v}_N(\boldsymbol{x}, \boldsymbol{\xi}))) \boldsymbol{\varphi}^T \rangle^T = \boldsymbol{0} \quad \text{in } D . \tag{41}$$

Steady state problems are usually solved by introducing a pseudo-time and iterating the solution until the condition

$$\sum_{j=1}^{N_x} \Delta x_j \| \hat{\boldsymbol{u}}_j^n - \hat{\boldsymbol{u}}_j^{n-1} \| \leq \varepsilon , \tag{42}$$

with convergence tolerance $\varepsilon$, is fulfilled. To obtain a more compact notation, let us define the pseudo-time update of the moments by

$$\boldsymbol{c}(\boldsymbol{w}_\ell, \boldsymbol{w}_c, \boldsymbol{w}_r) := \langle \boldsymbol{u}(\boldsymbol{w}_c^T \boldsymbol{\varphi}) \boldsymbol{\varphi}^T \rangle^T \tag{43}$$
$$- \frac{\Delta t}{\Delta x} \left( \langle \boldsymbol{g}(\boldsymbol{u}(\boldsymbol{w}_c^T \boldsymbol{\varphi}), \boldsymbol{u}(\boldsymbol{w}_r^T \boldsymbol{\varphi})) \boldsymbol{\varphi}^T \rangle^T - \langle \boldsymbol{g}(\boldsymbol{u}(\boldsymbol{w}_\ell^T \boldsymbol{\varphi}), \boldsymbol{u}(\boldsymbol{w}_c^T \boldsymbol{\varphi})) \boldsymbol{\varphi}^T \rangle^T \right) .$$

Note that the first term of this update recalculates moments from inexact dual variables, i.e. we perform a recalculation step according to Algorithm 2. To indicate the usage of inexact dual states, we again use the notation $\bar{\boldsymbol{v}}_j^n$. Then, the moment iteration of cell $j$, which is performed until (42) is fulfilled reads

$$\hat{\boldsymbol{u}}_j^{n+1} = \boldsymbol{c} \left( \bar{\boldsymbol{v}}_{j-1}^n, \bar{\boldsymbol{v}}_j^n, \bar{\boldsymbol{v}}_{j+1}^n \right) . \tag{44}$$

During each iteration, the dual variables $\bar{\boldsymbol{v}}_j^n$ are again obtained by iterating

$$\boldsymbol{v}_j^{(l+1)} = \boldsymbol{d}(\boldsymbol{v}_j^{(l)}; \hat{\boldsymbol{u}}_j^n)$$

until the stopping criterion (27) is fulfilled. A schematic of this method is given in Fig. 1. In the following, we refer to updating the dual variables as the inner loop and the iteration of the moments as the outer loop. The key idea of the *One-Shot IPM* (osIPM) method is to break up the inner loop and iterate moments and dual variables to their steady state simultaneously. This method is motivated by the One-Shot method in shape optimization [15], which proposes to perform only a single iteration for the primal, dual and design updates. The osIPM method now reads

$$\boldsymbol{v}_j^{n+1} = \boldsymbol{d}(\boldsymbol{v}_j^n, \hat{\boldsymbol{u}}_j^n) \quad \text{for all j} , \tag{45a}$$
$$\hat{\boldsymbol{u}}_j^{n+1} = \bar{\boldsymbol{u}}_j^n - \frac{\Delta t}{\Delta x} \left[ \boldsymbol{G}^*(\boldsymbol{v}_{j-1}, \boldsymbol{v}_j) - \boldsymbol{G}^*(\boldsymbol{v}_j, \boldsymbol{v}_{j+1}) \right] \quad \text{for all j} . \tag{45b}$$

Note that the dual variables from the One-Shot iteration are written without a bar to indicate that they are not intended to be a solution of the dual problem. It can be
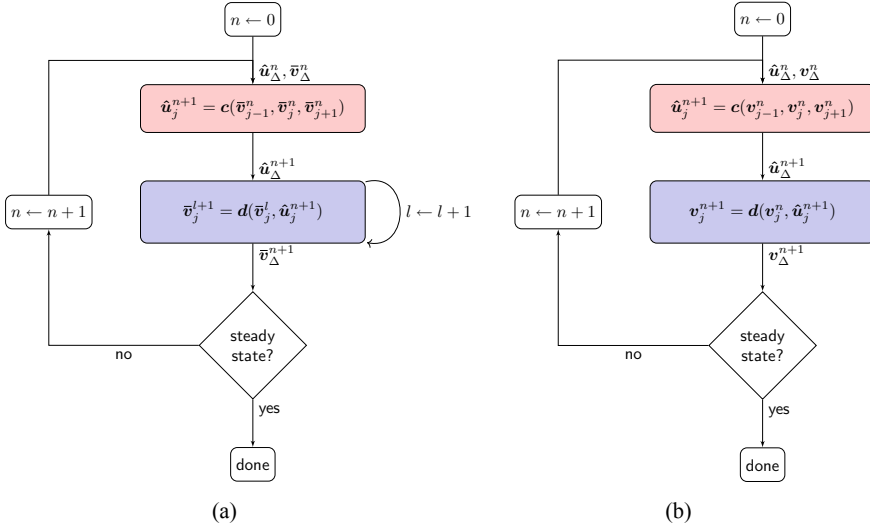
**Fig. 1** *Left:* IPM method for steady state problems. *Right:* osIPM method. The use of $\Delta$ indicates that all spatial cells of the corresponding quantity are collected in a vector.

shown that the osIPM method converges locally [23]. Numerical studies show that the One-Shot IPM method requires more iterations of the outer loop compared to the general IPM method, but as these iterations are significantly cheaper in terms of computational effort, the method yields a significant boost in the performance [23].

## 6 Results

In order to demonstrate the properties of the presented IPM method and the related acceleration techniques, in this section we show four different test cases for the Burgers and the Euler equations, each highlighting different aspects.

### 6.1 Burgers' Equation

In the following, we investigate Burgers' forming shock testcase from [33], which has also been investigated in [19–22]. The stochastic Burgers equation for a one-dimensional spatial domain is given by

$$\partial_t u(t, x, \xi) + \partial_x \frac{u(t, x, \xi)^2}{2} = 0 \,,$$
$$u(t = 0, x, \xi) = u_{\text{IC}}(x, \xi) \,.$$

The initial condition is now randomly distributed, i.e. we have

$$u_{\text{IC}}(x, \xi) := \begin{cases} u_L, & \text{if } x < x_0 + \sigma\xi \\ u_L + \frac{u_R - u_L}{x_0 - x_1}(x_0 + \sigma\xi - x), & \text{if } x \in [x_0 + \sigma\xi, x_1 + \sigma\xi] \\ u_R, & \text{else} \end{cases} . \quad (46)$$

The initial condition describes a forming shock with a linear connection from $x_0 + \sigma\xi$ to $x_1 + \sigma\xi$, i.e. the random variable $\xi$ shifts the forming shock structure to the left and right. We assume a uniformly distributed random variable in the interval $[-1, 1]$, i.e. $\xi \sim U([-1, 1])$. Furthermore, we use the following parameter values:

| | |
|---|---|
| $D = [a, b] = [0, 3]$ | Range of spatial domain |
| $N_x = 2000$ | Number of spatial cells |
| $t_{\text{end}} = 0.0909$ | End time |
| $x_0 = 0.5, x_1 = 1.5, u_L = 12, u_R = 1, \sigma = 0.2$ | Parameters of initial condition (46) |
| $N + 1 = 6$ | Number of moments |
| $\varepsilon = 10^{-7}$ | Accuracy of Newton's method |

We compare the solution in $\xi$ at a fixed spatial position $x^*$ for time $t_{\text{end}}$ for stochastic-Galerkin and IPM in Fig. 2. The IPM method uses the *bounded–barrier* entropy

$$s(u) = (u - u_-) \ln(u - u_-) + (u_+ - u) \ln(u_+ - u) ,$$

which, in contrast to the log–barrier entropy (31) takes finite values at $u_-$ and $u_+$. Indeed, as seen in Sect. 4.2, it suffices that the ansatz $(s')^{-1}$ only takes values in $[u_-, u_+]$ to enforce such bounds. For the bounded–barrier entropy, we can choose the distance to the exact solution to be zero, i.e. we have $\Delta u := u_R - u_- = u_+ - u_L = 0$. We also show results for the log–barrier entropy with $\Delta u = 0.5$. It can be seen that stochastic-Galerkin oscillates heavily while both IPM solutions maintain the overall shock characteristics. However, the bounded–barrier entropy is able to capture the shock more adequately while maintaining the maximum–principle (15). In the following, we focus on the bounded–barrier entropy and investigate its behavior when approximating expectation value and variance. For this, we let the simulation run until an increased end time $t_{\text{end}} = 0.14$ is reached. Expectation value and variance are shown in Fig. 3. While stochastic-Galerkin yields a step-like profile, the IPM method when using the bounded–barrier entropy shows a significantly improved solution. Note, that the log–barrier entropy yields a similar result.
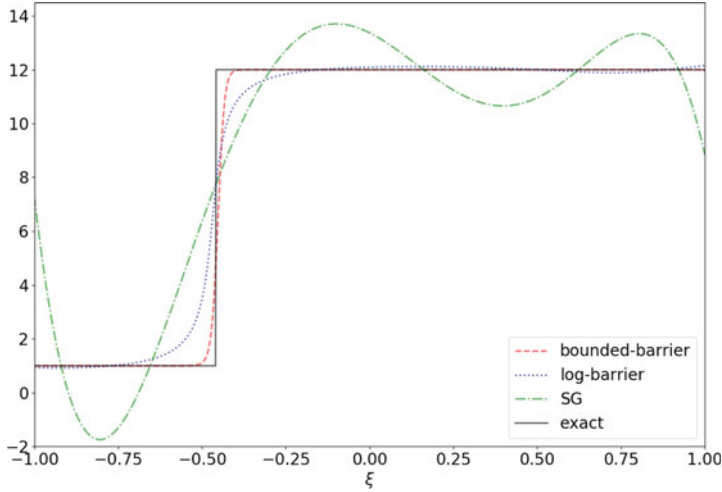
**Fig. 2** Solution SG and IPM with bounded–barrier ($\Delta u = 0$) and log–barrier ($\Delta u = 0.5$) entropies at fixed spatial position $x^* = 1.5$ for time $t_{\text{end}} = 0.0909$.

## 6.2 Euler Equations

A further commonly used test case for intrusive methods is Sod's shock tube with uncertain shock position, see for example [22, 33, 34]. The stochastic Euler equations in one spatial dimension read

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e + p) \end{pmatrix} = \mathbf{0} \,,$$

where, in our test case, we use the initial condition

$$\rho_{\text{IC}} = \begin{cases} \rho_L & \text{if } x < x_{\text{interface}}(\xi) \\ \rho_R & \text{else} \end{cases} \,,$$

$$(\rho u)_{\text{IC}} = 0 \,,$$

$$(\rho e)_{\text{IC}} = \begin{cases} \rho_L e_L & \text{if } x < x_{\text{interface}}(\xi) \\ \rho_R e_R & \text{else} \end{cases} \,.$$

Here, $\rho$ denotes the density, $u$ is the velocity and $e$ is the specific total energy. One can determine the pressure $p$ from

$$p = (\gamma - 1)\rho \left( e - \frac{1}{2}u^2 \right) \,.$$
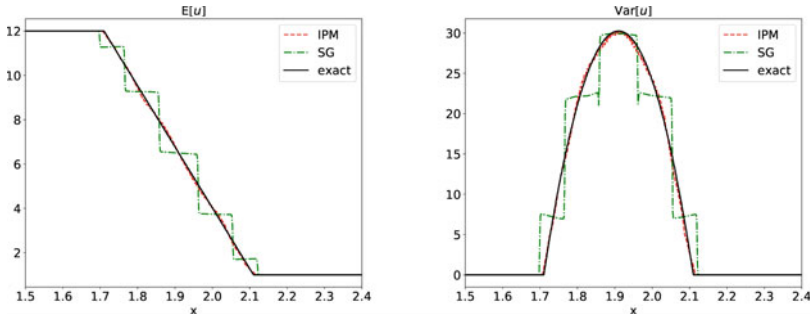
**Fig. 3** Expectation value and variance for SG and IPM at time $t_{\text{end}} = 0.14$.

The heat capacity ratio is $\gamma$ and has a value of 1.4 for air. We use the random interface position $x_{\text{interface}}(\xi) = x_0 + \sigma\xi$, where $\xi$ is again uniformly distributed in the interval $[-1, 1]$. The IPM method again needs to pick a suitable entropy. In this work, we choose the entropy

$$ s(\rho, \rho u, \rho e) = -\rho \ln \left( \rho^{-\gamma} \left( \rho e - \frac{(\rho u)^2}{2\rho} \right) \right) , $$

though more choices are possible. Parameter values which differ from Burgers' test case are

| | |
|---|---|
| $D = [a, b] = [0, 1]$ | Range of spatial domain |
| $N_x = 5000$ | Number of spatial cells |
| $t_{\text{end}} = 0.14$ | End time |
| $x_0 = 0.5, \sigma = 0.05$ | Interface position parameters |
| $\rho_L = 1.0, e_L = 2.5, \rho_R = 0.125, e_R = 0.25$ | Initial states |

When running the simulation, the SG method fails already during the first time update. The reason for this can be seen in Fig. 4. Here, the SG and IPM reconstructions of the gas density $\rho$ are depicted at $t = 0$ s at a fixed spatial cell. While the IPM reconstruction maintains positivity, the Gibbs phenomena that result from the polynomial representation of SG lead to negative density values. A similar behavior can be seen for the energy $e$. Then, the eigenvalues of the Euler equations, which include $v \pm \sqrt{\gamma p / \rho}$ become complex, i.e. the system is no longer hyperbolic.

As discussed, the IPM method maintains hyperbolicity, meaning that one can run the simulation until the desired end time $t_{\text{end}} = 0.14$ s is reached. The resulting expectation values and variances are depicted in Fig. 5. It can be seen that the IPM method yields a satisfactory approximation of the expectation value and variance at the rarefaction wave as well as the contact discontinuity. However, the shock
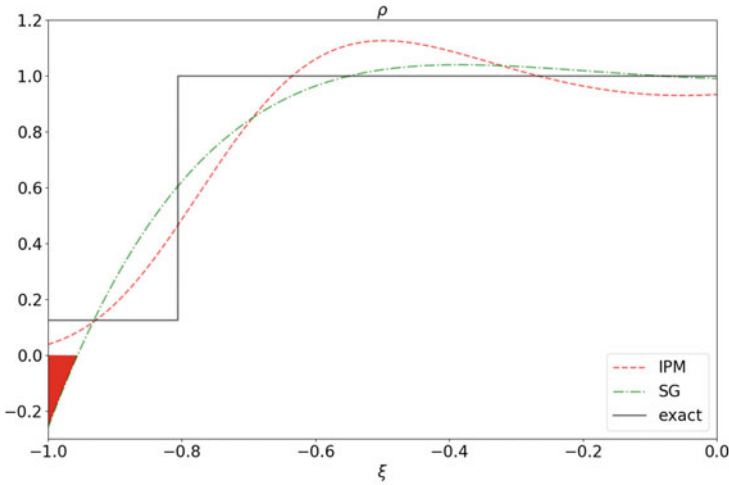
**Fig. 4** Initial density for SG and IPM at fixed spatial position $x^* = 0.46$ when using 6 moments. The view is zoomed to $\xi \in [-1, 0]$ and negative regions are marked in red.
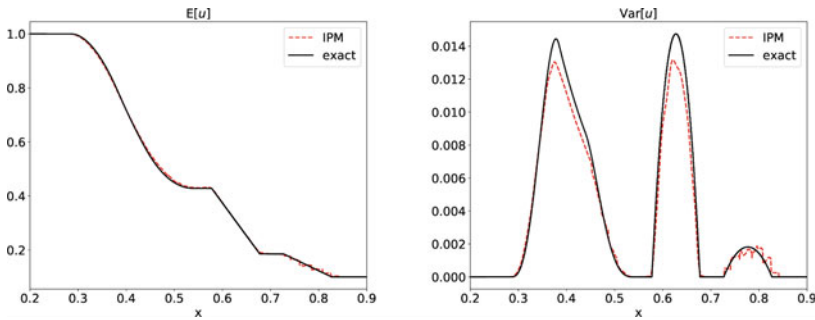


**Fig. 5** Expectation value and variance for SG and IPM at time $t_{\text{end}} = 0.14$ s.

yields discontinuous step-like profiles, similar to the stochastic-Galerkin results for Burgers' equation.

## 6.3 2-D Euler Equations with One-Shot

In order to demonstrate the acceleration impact of the aforementioned One-Shot strategy for the steady-state case, we will quantify the effects of an uncertain angle of attack $\phi \sim U(0.75, 1.75)$ for a NACA0012 airfoil using the Euler equations in two spatial dimensions. This test-case is taken from [23]. Similar to the 1-D case, the stochastic Euler equations in two dimensions are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e \end{pmatrix} + \partial_{x_1} \begin{pmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho e + p) \end{pmatrix} + \partial_{x_2} \begin{pmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho e + p) \end{pmatrix} = \mathbf{0} \,,$$

with the closure term for the pressure as

$$p = (\gamma - 1)\rho \left( e - \frac{1}{2}(u_1^2 + u_2^2) \right) \,.$$

As in the previous test case, the heat capacity ratio is set to $\gamma = 1.4$. For this test case, we apply the Euler slip boundary condition to the airfoil's boundary as $\boldsymbol{v}^T \boldsymbol{n} = 0$, where $\boldsymbol{n}$ denotes the surface normal. At a sufficiently large distance away from the airfoil, we prescribe a far field flow with a given Mach number of $Ma = 0.8$, pressure $p = 101\,325$ Pa and a temperature of 273.15 K. The uncertain angle of attack $\phi$ is uniformly distributed in the interval of $[0.75, 1.75]$ degrees or in other words $\phi(\xi) = 1.25 + 0.5\xi$ with $\xi \sim U(-1, 1)$. The initial condition in the entire domain is equal to the far field boundary values and thus violates the Euler slip boundary condition at the airfoil. Consequently, we iterate in pseudo-time to correct the flow solution until the expectation value of the density fulfills the criterion (42) with $\varepsilon = 6 \cdot 10^{-6}$.

The used computational mesh (see Fig. 6c) consists of 22 361 triangular elements and resembles a circular domain of 40 m diameter, where the airfoil of 1 m length is located at the very center. The mesh is finely resolved close to the airfoil as we are only interested in effects close to the airfoil and becomes coarser the closer to the far field boundary. In order to be able to measure the quality of the obtained solutions with and without the One-Shot acceleration strategy, we compute a reference solution using stochastic-Collocation with 100 Gauss-Legendre quadrature points (see Fig. 6). We will show the L$^2$-error behavior of the discrete quantity $\boldsymbol{e}_\Delta = (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{N_x})^T$, where $\boldsymbol{e}_j$ is the cell average of the quantity $\boldsymbol{e}$ in spatial cell $j$. The discrete L$^2$ norm is denoted by

$$\|\boldsymbol{e}_\Delta\|_\Delta := \sqrt{\sum_{j=1}^{N_x} \Delta x_j \boldsymbol{e}_j^2} \,.$$

Given the SC reference solution $\boldsymbol{u}_\Delta$ and the moments of a compared numerical method $\hat{\boldsymbol{u}}_\Delta$, we investigate the relative error

$$\frac{\|\mathrm{E}[\boldsymbol{u}_\Delta] - \mathrm{E}[\mathcal{U}(\hat{\boldsymbol{u}}_\Delta)]\|_\Delta}{\|\mathrm{E}[\boldsymbol{u}_\Delta]\|_\Delta} \quad \text{and} \quad \frac{\|\mathrm{Var}[\boldsymbol{u}_\Delta] - \mathrm{Var}[\mathcal{U}(\hat{\boldsymbol{u}}_\Delta)]\|_\Delta}{\|\mathrm{Var}[\boldsymbol{u}_\Delta]\|_\Delta} \,.$$

As small fluctuations in the large cells of the coarse far field would dominate this error measure, we only compute the error inside a box of one meter height and 1.1 m length around the airfoil. Figure 7 shows the resulting error with respect to
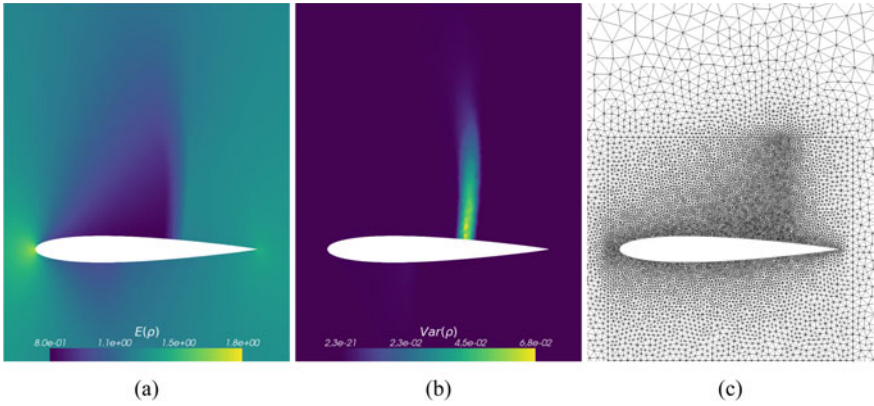
**Fig. 6** Reference solution $E[\rho]$ and $Var[\rho]$ and the mesh close to the airfoil which is used in the computation of all presented methods.
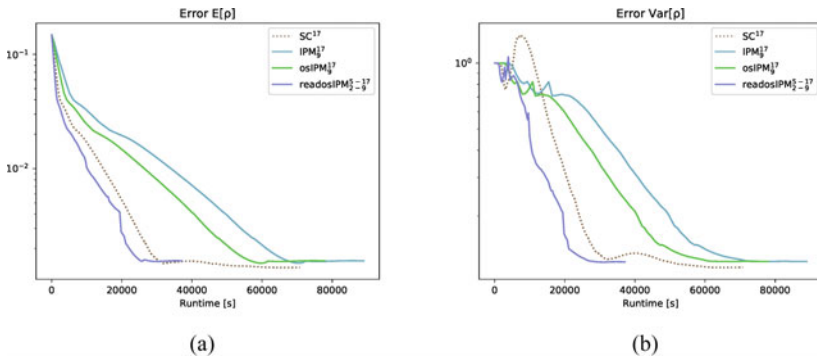


**Fig. 7** Comparison of the relative $L^2$-error of the density for IPM, osIPM, readosIPM and SC. All IPM related methods are converged to a residual of $\varepsilon = 6 \cdot 10^{-6}$, whereas SC is converged to a residual of $\varepsilon = 1 \cdot 10^{-7}$. All computations are performed with 5 MPI threads.

the reference solution of IPM, osIPM and SC. The superscript in the figure denotes the number of used quadrature points, whereas the subscript denotes the moment order. We chose a total polynomial order of 9 for all IPM methods, meaning 10 moments are used in the computation and a Clenshaw-Curtis quadrature rule of order 4, resulting in 17 quadrature points. Based on the same quadrature set, all IPM solutions were also compared to a SC solution in order to get a better understanding of the methods convergence behavior and acceleration properties of osIPM. As it can be seen from the presented results, the osIPM yields the same error and almost identical convergence history as IPM, while being significantly faster. In comparison to SC however, the errors for the mean as well as the variance are comparably small, but the SC method reaches the given error level faster in terms of computational time. Only when the One-Shot approach is combined with adaptivity and refinement

**Table 1** Moment and quadrature setup for the applied refinement levels.

| Refinement level | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Total degree of moments | 1 | 2 | 3 | 4 |
| Number of quadrature points | 3 | 5 | 5 | 9 |

retardation, as it can be seen for the *readosIPM* plot in Fig. 7, the IPM method is yields even faster convergence rates than SC. For more information about refinement retardation and adaptivity, see [23]. Also note that the acceleration of the One-Shot approach becomes more dominant when looking at higher dimensional uncertainties, see [23].

### 6.4   Unsteady 2-D Euler Equations with Adaptivity

To investigate adaptivity, we again use the two dimensional Euler equations. The problem setting is similar to the transient Sod shock tube test case from above. The geometry describes a nozzle similar to a de Laval nozzle (see Fig. 8e), where the initial condition is set to a discontinuity positioned in the middle of the narrow part of the nozzle. The density in the left part is set to 1 kg m$^{-3}$ and the energy is set to $\rho e = p/(\gamma - 1) = 2.5$ J m$^{-3}$ with the pressure $p$ equal to 1 Pa. For the right part of the domain the density is set to $\rho = 0.8$ kg m$^{-3}$ and the pressure $p = 0.125$ Pa. The gas in both sides is at rest. For this testcase we inflict the initial condition with one uncertainty, i.e. the shock's position, which is now modeled as $x_{\text{shock}} \sim U(-0.5, 0.5)$. The used computational mesh consists of 76 696 triangular cells and is refined in the area of the shock and the nozzle opening towards the right side of the domain (see Fig. 8a). The applied boundary conditions are Euler slip conditions for the wall of the nozzle and Dirichlet conditions set to the initial condition for the left and right side of the mesh. The shown results in Fig. 8 resemble a time of 6 s.

As for the previous testcases, the reference solution was computed using stochastic-Collocation (see Figs. 8a, 8b) with a Gauss-Legendre quadrature with 50 quadrature points. As the previously mentioned parameter $\delta_{\pm}$ are user determined, these refinement/coarsening thresholds were set to $\delta_- = 1.5E - 3$ and $\delta_+ = 5E - 4$ for the presented results in Figs. 8c, 8d. The resulting refinement levels are shown in Fig. 8c and the total order of used moments in combination with the associated quadrature points for each refinement level are given in Table 1. For the quadrature a tensorized Clenshaw-Curtis rule is used.

As for the previous test cases, we observe a good agreement between the IPM and the reference solution computed by SC. Due to the lower degree of moments the IPM solution again show a more step-like profile in the emerging shocks. As expected, the presented refinement levels are high in the regions around the shock and lower
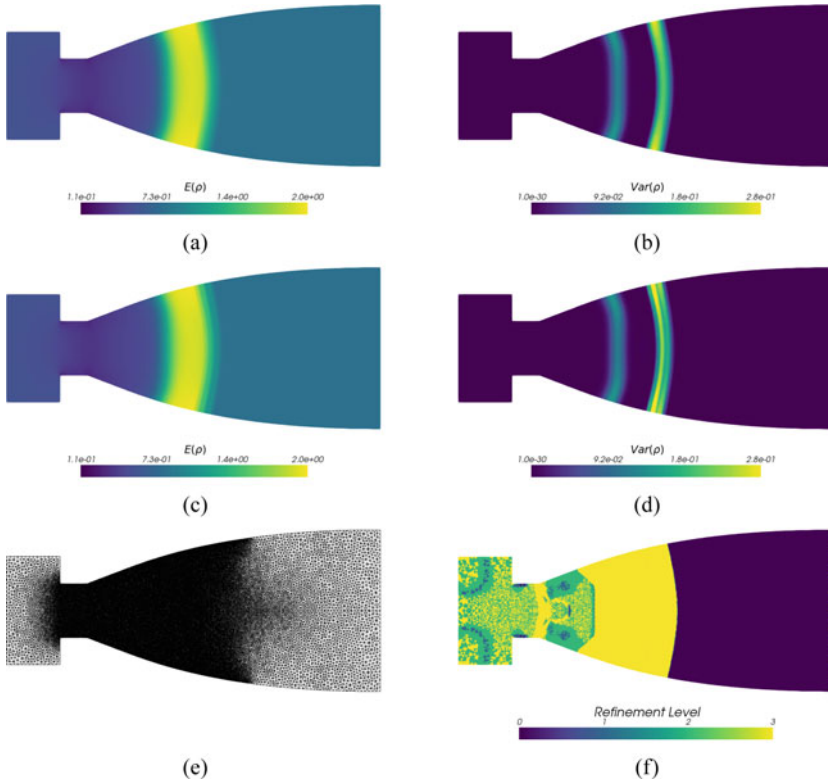
**Fig. 8** Comparison for the mean and variance of the SC reference solution (**a**, **b**) and the adaptive IPM method (**c**, **d**) with the refinement levels in **f**. **e** shows the computational mesh. All computations are performed using 20 MPI threads.

order moments are started to be used further upstream where the flow becomes more and more constant as time progresses. Further upstream of the shock, the method even uses the lowest refinement level as the shock has not yet reached this part of the nozzle and thus the solution is still equal to the initial condition. All in all the results show that the method chooses high levels of refinement in areas where they are required by the complexity of the solution. Thus, the method is computationally much of efficient in the remainder of the domain.

# References

1. Alldredge, G., Hauck, C., Tits, A.: High-order entropy-based closures for linear transport in slab geometry II: a computational study of the optimization problem. SIAM J. Sci. Comput. **34**(4), B361–B391 (2012)
2. Alldredge, G., Hauck, C.D., O'Leary, D.P., Tits, A.L.: Adaptive change of basis in entropy-based moment closures for linear kinetic equations. J. Comput. Phys. **258**, 489–508 (2014)
3. Alldredge, G.W., Frank, M., Hauck, C.D.: A regularized entropy-based moment method for kinetic equations. SIAM J. Appl. Math. **79**(5), 1627–1653 (2019)
4. Bell, J.B., Dawson, C.N., Shubin, G.R.: An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. J. Comput. Phys. **74**(1), 1–24 (1988)
5. Bijl, H., Lucor, D., Mishra, S., Schwab, C.: Uncertainty Quantification in Computational Fluid Dynamics, vol. 92. Springer (2013)
6. Caflisch, R.: Monte Carlo and Quasi-Monte Carlo methods. Acta Numerica (1998)
7. Colella, P.: Multidimensional upwind methods for hyperbolic conservation laws. J. Comput. Phys. **87**(1), 171–200 (1990)
8. Deshpande, S.: Kinetic theory based new upwind methods for inviscid compressible flows. In: 24th Aerospace Sciences Meeting, p. 275 (1986)
9. Kristopher Garrett, C., Hauck, C., Hill, J.: Optimization and large scale computation of an entropy-based moment closure. J. Computat. Phys. **302**, 573–590 (2015)
10. Ghanem, R., Higdon, D., Owhadi, H.: Handbook of Uncertainty Quantification. Springer (2017)
11. Giles, M.B.: Multilevel Monte Carlo methods. Acta Numerica (2018)
12. Guermond, J.-L., Nazarov, M., Popov, B., Yang, Y.: A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations. SIAM J. Numer. Anal. **52**(4), 2163–2182 (2014)
13. Harten, A., Lax, P.D., van Leer, B.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev. **25**(1), 35–61 (1983)
14. Hauck, C.: High-order entropy-based closures for linear transport in slab geometry. Commun. Math. Sci. **9**, 187–205 (2011)
15. Hazra, S.B., Schulz, V., Brezillon, J., Gauger, N.R.: Aerodynamic shape optimization using simultaneous pseudo-timestepping. J. Comput. Phys. **204**(1), 46–64 (2005)
16. Holden, H., Risebro, N.H.: Front Tracking for Hyperbolic Conservation Laws, vol. 152. Springer (2015)
17. Kröker, I., Rohde, C.: Finite volume schemes for hyperbolic balance laws with multiplicative noise. Appl. Numer. Math. **62**(4), 441–456 (2012)
18. Kružkov, S.N.: First order quasilinear equations in several independent variables. Math. USSR-Sbornik **10**(2), 217 (1970)
19. Kusch, J., Alldredge, G.W., Frank, M.: Maximum-principle-satisfying second-order intrusive polynomial moment scheme. SMAI J. Comput. Math. **5**, 23–51 (2019)
20. Kusch, J., Frank, M.: Intrusive methods in uncertainty quantification and their connection to kinetic theory. Int. J. Adv. Eng. Sci. Appl. Math. **10**(1), 54–69 (2018)
21. Kusch, J., Frank, M.: An adaptive quadrature-based moment closure. Int. J. Adv. Eng. Sci. Appl. Math. **11**(3), 174–186 (2019)
22. Kusch, J., McClarren, R.G., Frank, M.: Filtered stochastic Galerkin methods for hyperbolic equations. J. Comput. Phys. **403**, 109073 (2019)
23. Kusch, J., Wolters, J., Frank, M.: Intrusive acceleration strategies for uncertainty quantification for hyperbolic systems of conservation laws (2019)
24. LeVeque, R.J.: Numerical Methods for Conservation Laws. Birkhäuser Verlag Basel (1992)
25. David Levermore, C.: Moment closure hierarchies for kinetic theories. J. Stat. Phys. **83**(5–6), 1021–1065 (1996)
26. Liu, X.-D.: A maximum principle satisfying modification of triangle based adapative stencils for the solution of scalar hyperbolic conservation laws. SIAM J. Numer. Anal. **30**(3), 701–716 (1993)

27. Le Maître, O.P., Knio, O.M.: Spectral Methods for Uncertainty Quantification. Springer (2010)
28. McClarren, R.G.: Uncertainty Quantification and Predictive Computational Science. Springer (2018)
29. Olbrant, E., Hauck, C.D., Frank, M.: A realizability-preserving discontinuous Galerkin method for the M1 model of radiative transfer. J. Comput. Phys. **231**(17), 5612–5639 (2012)
30. Persson, P.-O., Peraire, J.: Sub-cell shock capturing for discontinuous Galerkin methods. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, p. 112 (2006)
31. Perthame, B.: Boltzmann type schemes for gas dynamics and the entropy property. SIAM J. Numer. Anal. **27**(6), 1405–1421 (1990)
32. Perthame, B.: Second-order Boltzmann schemes for compressible Euler equations in one and two space dimensions. SIAM J. Numer. Anal. **29**(1), 1–19 (1992)
33. Poëtte, G., Després, B., Lucor, D.: Uncertainty quantification for systems of conservation laws. J. Comput. Phys. **228**(7), 2443–2467 (2009)
34. Schlachter, L., Schneider, F.: A hyperbolicity-preserving stochastic Galerkin approximation for uncertain hyperbolic systems of equations. J. Comput. Phys. **375**, 80–98 (2018)
35. Smith, R.C.: Uncertainty Quantification: Theory, Implementation, and Applications. SIAM (2013)
36. Tadmor, E.: The numerical viscosity of entropy stable schemes for systems of conservation laws. I. Math. Comput. **49**(179), 91–103 (1987)
37. Tokareva, S., Schwab, C., Mishra, S.: High order SFV and mixed SDG/FV methods for the uncertainty quantification in multidimensional conservation laws. In: Abgrall, R., Beaugendre, H., Congedo, P.M., Dobrzynski, C., Perrier, V., Ricchiuto, M. (eds.) High Order Nonlinear Numerical Schemes for Evolutionary PDEs, pp. 109–133. Springer (2014)
38. Trefethen, L.N.: Cubature, approximation, and isotropy in the hypercube. SIAM Rev. **59** (2017)
39. Xiang, S., Bornemann, F.: On the convergence rates of gauss and Clenshaw-Curtis quadrature for functions of limited regularity. SIAM J. Numer. Anal. **50**(5), 2581–2587 (2012)
40. Xiu, D.: Numerical Methods for Stochastic Computations. Princeton University Press (2010)
41. Zhang, X., Shu, C.-W.: Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 467, pp. 2752–2776. The Royal Society (2011)