On the efficient implementation of PVM methods and simple Riemann solvers. Application to the Roe method for large hyperbolic systems

Ernesto Pimentel-García[1], Carlos Parés[1], Manuel J. Castro[1], Julian Koellermeier[2]

University of Málaga[1], Peking University[2]

August 2, 2019

Abstract

PVM methods can be considered as approximations of the Roe method in which the absolute value of the Roe matrix appearing in the numerical viscosity is replaced by the evaluation of the Roe matrix at a chosen polynomial that approximates the absolute value function. They are in principle cheaper than the Roe method since the computation and the inversion of the eigenvector matrix is not necessary. In this article, an efficient implementation of the PVM based on polynomials that interpolate the absolute value function at some points is presented. This implementation is based on the Newton form of the polynomials. Moreover, many numerical methods based on simple Riemann solvers may be interpreted as PVM methods and thus this implementation can be also applied to them: the close relation between PVM methods and simple Riemann solvers are revisited here and new shorter proofs based on the classical interpolation theory are given. In particular, Roe method can be interpreted both as a SRS and as a PVM method so that the new implementation can be used. This implementation, that avoids the computation and the inversion of the eigenvector matrix, is called Newton Roe method. Newton Roe method yields the same numerical results of the standard Roe method, with less runtime for large PDE systems. Numerical results for two-layer Shallow Water Equations and Quadrature-Based Moment Equations show a significant speedup if the number of equations is large enough.

Keywords: PVM methods, simple Riemann solvers, Roe method, finite volume methods, path-conservative methods, large hyperbolic systems.

Acknowledgments. CP, MC, EP have been partially financed by the State Research Agency (SRA) and European Regional Development Fund (ERDF) through Research project MTM2015-70490-C2-1-R. The last author has been funded by a joint postdoctoral scholarship of Free University Berling and Peking University.

1 Introduction

Roe method (see [21]) is one of the most popular conservative methods to solve systems of conservation laws. Its natural extension to non-conservative systems was given by Toumi (see [25]) and reformulated as a path-conservative method in [20] and [19]. Nevertheless its application to large hyperbolic systems can be costly as it requires the complete knowledge

of the eigenstructure of the system matrix. To overcome this difficulty, different families of methods have been proposed in the literature. Two of them are the Approximate Riemann solvers (ARS), in the sense of Harten et al. [11], and the Polynomial Viscosity matrix (PVM) methods, introduced in [4].

In the expression of Roe method two different terms can be distinguished: a centered numerical flux (or a centered fluctuation in the nonconservative case) plus a viscous term that involves the product of the absolute value of the Roe matrix by the difference of the states at both sides of the inter-cell. In PVM methods, the absolute value of the matrix is replaced by the evaluation of the Roe matrix at a chosen polynomial what avoids the necessity of computing the eigenvalues and eigenvectors. In many cases, the chosen polynomial interpolates the absolute value function at some points (either in the Lagrange or the Hermite sense): we will call interpolatory PVM to these methods for simplicity. A new implementation of interpolatory PVM methods is presented here. This new implementation is based on the Newton form of the polynomials that is known to be the more efficient way to evaluate the interpolation polynomials.

Approximate Riemann solvers on their side are based on the approximation of the solutions of the Riemann problems associated at each inter-cell. In the case of the so-called simple Riemann solvers (SRS) these approximations consist of some constant states linked by jump discontinuities that travel at constant speed. ARS and SRS were generalized for nonconservative hyperbolic systems using the path-conservative framework in [19]. As it has been pointed out in [18], there is a close connection between PVM and SRS methods. In particular, under certain assumptions, a SRS method can be interpreted as an interpolatory PVM. In this case, the efficient implementation of PVM methods introduced here can also be applied to the SRS. For the sake of completeness, the main results of [18] are revisited here: in order to clarify and simplify the results shown there, new shorter proofs based on the classical interpolation theory are given.

Interestingly enough, Roe method can be interpreted itself as a complete SRS and as a PVM, as it was pointed out in [4]. Therefore, the implementation based on the Newton form of the corresponding interpolating polynomial can be used: this new implementation will be called *Newton Roe method* here. The advantage of Newton Roe method compared to the standard implementation is that (a) the eigenvectors have not to be computed and (b) no matrix inversion is required. This new implementation shows a significant speedup if the number of equations of the hyperbolic systems is large enough. In particular we are interested in solving large nonconservative hyperbolic systems like those corresponding to the multi-layer shallow-water system (see for example [5]) or the Quadrature-Based Moment Equations (QBME, see [12] and the references therein) where the number of equations is a parameter of the model, either given in terms of the number of layers in the first case or the number of moments in the second one.

Although the application of the methods to general nonconservative systems is considered here for the sake of generality, all of them reduce to conservative methods for systems of conservation laws so that the new implementations can be used as well in this particular case.

The outline of this paper is as follows: in Section 2 PVM methods are briefly presented. Then, the implementation of interpolatory PVM methods using the Newton form of the interpolating polynomial is introduced together with a study of the complexity. In Section 3, after recalling the definition of SRS, its relation with PVM methods is analyzed. Section 4 is devoted to the description of the Roe method as a PVM solver that allows us to apply the Newton form of the interpolation polynomial to improve the computational

efficiency of its implementation. Several numerical tests to compare both methods are presented in Section 5. Two well known nonconservative systems are considered: the two-layer shallow-water system and the QBME moment models in primitive and partially conservative variables. Following [17], Ferrari's formuls is used to compute the eigenvalues of the two-layer shallow water system. In the case of the QBME model the eigenvalues are explicitly known but the full eigenvector decomposition might be costly to compute. Moreover, the number of equations increases when considering more moments, leading to large nonconservative hyperbolic systems. Finally, in Section 6 we summarize the conclusion of this work.

2 PVM methods

2.1 Definition

We consider a general nonconservative system

$$\partial_t W + \mathcal{A}(W)\partial_x W = 0, \quad x \in \mathbb{R}, t > 0, \tag{2.1}$$

where $W(x,t)$ takes value in an open convex subsetΩ, of $\mathbb{R}^N$, and

$$\begin{array}{c} \mathcal{A} : \Omega \rightarrowtail \mathcal{M}_{N\times N}(\mathbb{R}) \\ W \rightarrowtail \mathcal{A}(W) \end{array},$$

is a smooth locally bounded map. We assume that the system is strictly hyperbolic, i.e., for each $W \in \Omega$ the matrix $\mathcal{A}(W)$ has N real distinct eigenvalues

$$\lambda_1(W) < ... < \lambda_N(W),$$

with associated eigenvector $R_1(W), ..., R_N(W)$. The characteristic fields $R_i(W)$ can be either genuinely nonlinear or linearly degenerated.

In order to define a PVM method for (2.1) it is necessary to introduce the concept of a generalized Roe matrix in the sense of Toumi in [25] based on a family of paths, i.e. a Lipchitz-continuous function

$$\begin{array}{c} \Phi : [0,1] \times \Omega \times \Omega \rightarrowtail \Omega \\ (\xi; W_L, W_R) \rightarrowtail \Phi(\xi; W_L, W_R) \end{array}$$

that satisfies:

$$\Phi(0; W_L, W_R) = W_l, \quad \Phi(1; W_L, W_R) = W_R, \quad \forall W_L, W_R \in \Omega, \tag{2.2}$$

and

$$\Phi(\xi; W, W) = W, \quad \forall \xi \in [0,1], W \in \Omega. \tag{2.3}$$

Given a family of paths Φ, a function $\mathcal{A}_\Phi : \Omega \times \Omega \rightarrowtail \mathcal{M}_{N\times N}(\mathbb{R})$ is called a Roe linearization if it verifies the following properties:

- For any $W_L, W_R \in \Omega$, $\mathcal{A}_\Phi(W_L, W_r)$ has N distinct real eigenvalues.
- For any $W \in \Omega$, $\mathcal{A}_\Phi(W, W) = \mathcal{A}(W)$.

- For any $W_L, W_R \in \Omega$,

$$\mathcal{A}_\Phi(W_L, W_R)(W_R - W_L) = \int_0^1 \mathcal{A}(\Phi(\xi; W_L, W_R)) \frac{\partial \Phi}{\partial \xi}(\xi; W_L, W_R) d\xi. \tag{2.4}$$

Remark 1. If the system is conservative, i.e. if $\mathcal{A}(W)$ is the Jacobian of a flux function F, (2.4) reduces to the usual Roe property

$$\mathcal{A}_\Phi(W_L, W_R)(W_R - W_L) = F(W_R) - F(W_L),$$

and thus the usual notion of Roe matrix is recovered.

We also need to chose a polynomial at every inter-cell:

$$p^{i+\frac{1}{2}}(x) = \sum_{j=0}^{s} \alpha_j^{i+\frac{1}{2}} x^j. \tag{2.5}$$

Definition 1. *The PVM method corresponding to the Roe linearization $\mathcal{A}_\Phi$ and the polynomials $p^{i+\frac{1}{2}}$ is the numerical scheme that writes as follows:*

$$W_i^{n+1} = W_i^n - \frac{\Delta x}{\Delta t}(D^+_{i-\frac{1}{2}} + D^-_{i+\frac{1}{2}}), \tag{2.6}$$

where

$$D^\pm_{i+\frac{1}{2}} = D^\pm(W_i^n, W_{i+1}^n) = \frac{1}{2}\mathcal{A}_{i+\frac{1}{2}}(W_{i+1}^n - W_i^n) \pm \frac{1}{2}\mathcal{Q}_{i+\frac{1}{2}}(W_{i+1}^n - W_i^n), \tag{2.7}$$

with

$$\mathcal{A}_{i+\frac{1}{2}} = \mathcal{A}_\Phi(W_i^n, W_{i+1}^n), \tag{2.8}$$

and the numerical viscosity matrix

$$\mathcal{Q}_{i+\frac{1}{2}} = p_r^{i+\frac{1}{2}}(\mathcal{A}_{i+\frac{1}{2}}) = \sum_{j=0}^{r} \alpha_j^{i+\frac{1}{2}} A^j_{i+\frac{1}{2}}. \tag{2.9}$$

PVM methods are path-conservative in the sense introduced in [19]. The idea behind these methods is the following: if instead of a polynomial, the absolute value is chosen to compute the viscosity matrix, i.e.

$$\mathcal{Q}_{i+\frac{1}{2}} = \left|\mathcal{A}_{i+\frac{1}{2}}\right|, \tag{2.10}$$

the resulting numerical scheme is the standard Roe method. The idea is then to choose a polynomial $p_r^{i+\frac{1}{2}}$ that approximates the absolute value function so that the numerical scheme is expected to be close to the Roe method but computationally less expensive, since the evaluation of the Roe matrix at the polynomial may be cheaper than the computation of its absolute value (that requires the knowledge of the eigenstructure).

In this paper we only consider polynomial approximations of the absolute value function that are based on Lagrange or Hermite interpolation, i.e. $p_r^{i+\frac{1}{2}}$ is the polynomial of degree less or equal than $r-1$ that interpolates the absolute value function at r different points $\sigma_j^{i+1/2}$, $j = 1, \ldots, r$.:

$$p_r^{i+\frac{1}{2}}(\sigma_j^{i+1/2}) = \left|\sigma_j^{i+1/2}\right|, \quad j = 1, \ldots, r,$$

or the polynomial $p_{2r}^{i+\frac{1}{2}}$ of degree $2r-1$ that interpolates the absolute value function and its derivative at r different points $\sigma_j^{i+1/2}$, $j=1,\ldots,r$:

$$p_{2r}^{i+\frac{1}{2}}(\sigma_j^{i+1/2}) = \left|\sigma_j^{i+1/2}\right|, \quad (p_{2r}^{i+\frac{1}{2}})'(\sigma_j^{i+1/2}) = sign(\sigma_j^{i+1/2}), \quad j=1,\ldots,r,$$

where

$$sign(x) = \begin{cases} -1 & x<0, \\ 0 & x=0, \\ 1 & x>0. \end{cases}$$

Well known conservative methods like Rusanov, Lax-Friedrichs, HLL, FORCE, GFORCE, etc. can be interpreted as PVM methods based on interpolating polynomials: see [4].

Remark 2. When one of the interpolation points $\sigma_j^{i+1/2}$ is equal to 0, the interpolated value is taken to be $\epsilon > 0$ instead of 0 as an entropy fix technique to avoid the appearance of 'dog-leg' phenomena.

2.2 Implementation: the Lagrange case

Once the interpolation points and values have been chosen, the most efficient way to evaluate the interpolation polynomials $p_r^{i+1/2}$ is using its Newton form, based on the well-known divided differences. Let us describe an algorithm to compute the product

$$\mathcal{Q}_{i+\frac{1}{2}}(W_{i+1}^n - W_i^n) \tag{2.11}$$

based on this form of the polynomial. For the sake of simplicity, the dependency on the inter-cell will not be explicitly written, so that indexes and super-indexes $i+1/2$ will be dropped. Moreover, W_L and W_R will be used instead of W_i^n and W_{i+1}^n for simplicity. Using this notation, one has that, in the case of Lagrange interpolation, (2.11) writes as follows:

$$\begin{aligned} \mathcal{Q}(W_R - W_L) &= p_r(\mathcal{A})(W_R - W_L) \\ &= [\sigma_1](W_R - W_L) + \sum_{i=2}^{r} [\sigma_1,\ldots,\sigma_i] \prod_{j=1}^{i-1} \left(\mathcal{A}(W_R - W_L) - \sigma_j(W_R - W_L)\right), \end{aligned}$$

where the divided differences are recursively defined as follows:

- $[\sigma_i] = |\sigma_i|, \quad j=1,\ldots,r.$
- Given $k+1$ indexes $\{i_0,...,i_k\} \subset \{1,...,r\}$,

$$[\sigma_{i_0},\sigma_{i_1},...,\sigma_{i_k}] = \frac{[\sigma_{i_1},...,\sigma_{i_k}] - [\sigma_{i_0},...,\sigma_{i_{k-1}}]}{\sigma_{i_k} - \sigma_{i_0}}. \tag{2.12}$$

Once the divided differences have been computed, the following algorithm can be used to compute (2.11) in an optimal way:

- $V_0 = W_R - W_L$,
- For i = 1 to r:

$$V_i = \mathcal{A}V_{i-1} - \sigma_i V_{i-1},$$

and finally,

$$p_r(\mathcal{A})(W_R - W_L) = [\sigma_1]V_0 + [\sigma_1, \sigma_2]V_1 + ... + [\sigma_1, ..., \sigma_r]V_{r-1}. \tag{2.13}$$

The operations needed to compute (2.11) are thus the following:

- $\frac{3}{2}r(r+1)$ operations to compute the divided differences;
- r matrix-vector products;
- $2r$ scalar-vector products;
- $2r$ vector sums.

Therefore, the total number of operations is

$$\frac{3}{2}r(r+1) + r(2N^2 - N) + 4rN.$$

Since in practice r is at most $O(N)$ the complexity of the algorithm is

$$O\left(2rN^2\right). \tag{2.14}$$

The complexity of the computation of $D^{\pm}$, that involves another matrix/vector product is the same.

2.3 Implementation: the Hermite case

In the case of Hermite interpolation, (2.11) writes as follows:

$$\begin{aligned} \mathcal{Q}(W_R - W_L) &= p_{2r}(\mathcal{A})(W_R - W_L) \\ &= [\tilde{\sigma}_1](W_R - W_L) + \sum_{i=2}^{2r}[\tilde{\sigma}_1, \ldots, \tilde{\sigma}_i] \prod_{j=1}^{i-1} \left(\mathcal{A}(W_R - W_L) - \tilde{\sigma}_j(W_R - W_L)\right), \end{aligned}$$

where

$$\tilde{\sigma}_{2j-1} = \tilde{\sigma}_{2j} = \sigma_j, \quad j = 1, \ldots, r,$$

and the divided differences are defined in the same way with the following exceptions:

- $[\tilde{\sigma}_{2j-1}, \tilde{\sigma}_{2j}] = sign(\sigma_j), \quad j = 1, \ldots r.$

The algorithm to compute (2.11) is then the same and its complexity is, in this case:

$$O\left(4rN^2\right). \tag{2.15}$$

Remark 3. This algorithm can be easily adapted to PVM based on a polynomial that interpolates the absolute value and its derivative at some points and only the absolute value at some other points.

3 Simple Riemann solvers

3.1 Definition

According to [19], the generalized definition of simple Riemann solver (SRS) for (2.1) is as follows

Definition 2. *Let us take a family of paths Φ in Ω. We suppose that for every pair of states $W_L, W_R \in \Omega$, a finite number $s \geq 1$ of speeds*

$$\sigma_0 = -\infty < \sigma_1 < ... < \sigma_s < \sigma_{s+1} = +\infty, \tag{3.1}$$

and $s-1$ intermediate states

$$W_0 = W_L, W_1, ..., W_{s-1}, W_s = W_R, \tag{3.2}$$

are chosen. The function $R : \mathbb{R} \times \Omega \times \Omega \rightarrowtail \Omega$ given by

$$R(\sigma; W_L, W_R) = \begin{cases} W_0 = W_L, & if \quad \sigma < \sigma_1, \\ W_1, & if \quad \sigma_1 < \sigma < \sigma_2, \\ \vdots & \\ W_j, & if \quad \sigma_j < \sigma < \sigma_{j+1} \\ \vdots & \\ W_{s-1}, & if \quad \sigma_{s-1} < \sigma < \sigma_s, \\ W_s = W_R, & if \quad \sigma_s < \sigma, \end{cases} \tag{3.3}$$

is said to be a SRS for (2.1) if it satisfies

$$R(\sigma; W, W) = W, \quad \forall W \in \Omega, \tag{3.4}$$

and

$$\sum_{j=1}^{s} \sigma_j (W_j - W_{j-1}) = \int_0^1 \mathcal{A}(\Phi(\xi; W_L, W_R)) \frac{\partial \Phi}{\partial \xi}(\xi; W_L, W_R) d\xi. \tag{3.5}$$

Any SRS for (2.1) leads to a path-conservative numerical method:

$$W_i^{n+1} = W_i^n - \frac{\Delta x}{\Delta t}(D^+(W_{i-1}^n, W_i^n) + D^-(W_i^n, W_{i+1}^n)), \tag{3.6}$$

where

$$D^-(W_L, W_R) = \begin{cases} \displaystyle\int_0^1 \mathcal{A}(\Phi(\xi; W_L, W_R)) \frac{\partial \Phi}{\partial \xi}(\xi; W_L, W_R) d\xi & if \quad \sigma_s < 0, \\ \displaystyle\sum_{\sigma_{j+1} < 0} \sigma_{j+1}(W_{j+1} - W_j) & if \quad \sigma_1 < 0 < \sigma_s, \\ 0 & if \quad \sigma_1 > 0. \end{cases} \tag{3.7}$$

and

$$D^+(W_L, W_R) = \begin{cases} 0 & if \quad \sigma_s < 0, \\ \displaystyle\sum_{\sigma_{j+1} > 0} \sigma_{j+1}(W_{j+1} - W_j) & if \quad \sigma_1 < 0 < \sigma_s, \\ \displaystyle\int_0^1 \mathcal{A}(\Phi(\xi; W_L, W_R)) \frac{\partial \Phi}{\partial \xi}(\xi; W_L, W_R) d\xi & if \quad \sigma_1 > 0. \end{cases} \tag{3.8}$$

3.2 Relation between PVM and SRS methods

In [18] the relation between PVM and SRS methods was studied: the main results shown there are revisited here and new shorter proofs are given. The following result gives a necessary and sufficient condition to have the equivalence between a PVM method and a SRS.

Theorem 1. *Given a PVM and a SRS method based on the same family of paths, the following statements are equivalent:*

1. $D^{\pm}_{PVM}(W_L, W_R) = D^{\pm}_{SRS}(W_L, W_R), \quad \forall W_L, W_R \in \Omega,$
2. *For every* $W_L, W_R \in \Omega$*:*

$$\sum_{i=1}^{r} |\sigma_i|(W_i - W_{i-1}) = p(A_\Phi(W_L, W_R))(W_R - W_L). \tag{3.9}$$

Proof. Taking into account (3.5) and (2.4) we have the following equality:

$$\sum_{j=1}^{r} \sigma_j(W_j - W_{j-1}) = \mathcal{A}_\Phi(W_L, W_R)(W_R - W_L), \tag{3.10}$$

where $\mathcal{A}_\Phi$ is the Roe linearization chosen to define the PVM. Now, given $W_L, W_R \in \Omega$, we have:

$$\begin{aligned}
&D^{+}_{PVM}(W_L, W_R) = D^{+}_{SRS}(W_L, W_R) \\
&\quad \Leftrightarrow \frac{1}{2}\mathcal{A}_\Phi(W_L, W_R)(W_R - W_L) + \frac{1}{2}p(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L) \\
&\qquad = \sum_{\sigma_{j+1}>0} \sigma_{j+1}(W_{j+1} - W_i) \\
&\quad \Leftrightarrow \frac{1}{2}\mathcal{A}_\Phi(W_L, W_R)(W_R - W_L) + \frac{1}{2}p(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L) \\
&\qquad = \frac{1}{2}\sum_{j=1}^{r} \sigma_j(W_j - W_{j-1}) + \frac{1}{2}\sum_{j=1}^{r} |\sigma_j|(W_j - W_{j-1}) \\
&\quad \Leftrightarrow p(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L) = \sum_{j=1}^{r} |\sigma_j|(W_j - W_{j-1}),
\end{aligned}$$

where (3.10) has been used. The proof of the equivalence between

$$D^{-}_{PVM}(W_L, W_R) = D^{-}_{SRS}(W_L, W_R)$$

and (3.9) is similar. □

3.3 PVM based on Lagrange interpolation

We first show that any PVM method based on Lagrange polynomial interpolation can be seen as a SRS. More precisely, the following result holds:

Theorem 2. *Any PVM based on a polynomials* p_r *of degree less or equal than r-1 that interpolates the graph of the absolute value at r different points* σ_i*,* $i = 1, ..., r$ *can be interpreted as a SRS with speeds* σ_i*,* $i = 1, ..., r$*.*

Proof. Let us consider the Lagrange polynomial basis $l_i(\lambda), i = 1, ..., r$, where l_i is the polynomial of degree $r-1$ such that:

$$l_i(\sigma_j) = \delta_{i,j} = \begin{cases} 1 & if \quad i = j, \\ 0 & if \quad i \neq j. \end{cases}$$

Then p_r can be written in its Lagrange form:

$$p(\lambda) = \sum_{i=1}^{r} |\sigma_i| l_i(\lambda).$$

Let us define:

$$V_i = l_i(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L), \quad i = 1, ..., r.$$

The intermediate states W_i, $i = 0, ..., r$ given by

- $W_0 = W_L$;
- $W_i = V_i + W_{i-1}, \quad j = 1, \ldots, r$

together with the speeds σ_i, $i = 1, \ldots, r$ define a SRS. In effect, from the equality

$$\sum_{i=1}^{r} l_i(\lambda) = 1, \quad \forall \lambda \in \mathbb{R}$$

we deduce

$$W_r = (W_r - W_0) + W_0 = \sum_{i=1}^{r} V_i + W_0 = (W_R - W_L) + W_0 = W_R.$$

Now, (3.4) is trivially checked and (3.5) is easily deduced from the Roe property (2.4) and

$$\sum_{i=1}^{r} \sigma_i l_i(\lambda) = \lambda, \quad \forall \lambda \in \mathbb{R}.$$

Finally, we have:

$$\sum_{i=1}^{r} |\sigma_i| V_i = \sum_{i=1}^{r} |\sigma_i| l_i(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L) = p(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L).$$

Therefore, using Theorem 1, the SRS solver coincides with the PVM method. □

Let us prove now a kind of reciprocal:

Theorem 3. *A SRS with r speeds σ_i, $i = 1, ..., r$ such that $V_i = W_i - W_{i-1}$, $i = 1, ..., r$, are linearly independent, can be interpreted as the PVM based on the polynomial p of degree less or equal than $r-1$ that interpolates the points*

$$\{(\sigma_i, |\sigma_i|)\}, \quad i = 1, \cdots, r.$$

Proof. First, if $r < N$, the set of linearly independent vectors $\{V_1, ..., V_r\}$ is completed to obtain a basis $\{V_1, ..., V_N\}$ of $\mathbb{R}_N$ and the set of real numbers $\sigma_1, ..., \sigma_r$ is completed to obtain a family of pairwise different numbers $\sigma_1, ..., \sigma_N$. Then we consider the matrix $\mathcal{A}_\Phi(W_L, W_R)$ whose eigenvalues are $\sigma_1, ..., \sigma_N$ and the corresponding eigenvectors are $\{V_1, ..., V_N\}$. We also consider the polynomial p such that $p(\sigma_i) = |\sigma_i|$, $i = 1, ..., r$. Let us see that the PVM associated with $\mathcal{A}_\Phi$ and p is equivalent to the SRS. Using the property (3.5) we have that:

$$\begin{aligned} \mathcal{A}_\Phi(W_L, W_R)(W_R - W_L) &= \mathcal{A}_\Phi(W_L, W_R)\left(\sum_{i=1}^{r} V_i\right) && (3.11)\\ &= \sum_{i=1}^{r} \sigma_i V_i && (3.12)\\ &= \int_0^1 \mathcal{A}(\Phi(\xi; W_L, W_R))\frac{\partial \Phi}{\partial \xi}(\xi; W_L, W_R)d\xi. && (3.13) \end{aligned}$$

Therefore, $\mathcal{A}_\Phi$ defines a Roe matrix. Moreover:

$$p(\mathcal{A}_\Phi(W_L, W_R))(W_R - W_L) = p(\mathcal{A}_\Phi(W_L, W_R))\sum_{i=1}^{r} V_i = \sum_{i=1}^{r} p(\sigma_i)V_i = \sum_{i=1}^{r} |\sigma_i|V_i,$$

and using again Theorem 1, we have that the PVM is equivalent to the SRS. □

Therefore, when a SRS satisfies the hypothesis of Theorem 3 it can be interpreted as an interpolatory PVM and the implementation described in the previous section can be thus applied.

3.4 PVM based on Hermite interpolation

We show now that any PVM method based on Hermite interpolation can be seen as a SRS solver.

Theorem 4. *Any PVM based on a polynomial p_{2r} of degree less or equal than $2r-1$ such that*

$$p_{2r}(\sigma_i) = |\sigma_i|,\ i = 1, ..., r, \quad p'_{2r}(\sigma_i) = sign(\sigma_i),\ i = 1, ..., r,$$

where $\sigma_1 < \sigma_2 < ... < \sigma_I < 0 < \sigma_{I+1} < ... < \sigma_r$, can be interpreted as a SRS with $r+1$ speeds σ_i, $i = 1, ..., r$, and 0.

Proof. Let us consider the Hermite polynomial basis $h_i(\lambda)$, $i = 1, ..., r$, $k_i(\lambda)$, $i = 1, ..., r$, i.e. the polynomials of degree less or equal than $2r - 1$ such that

$$\begin{aligned} h_i(\sigma_j) = \delta_{i,j},\ h'_i(\sigma_j) = 0, \quad \forall i, j,\\ k_i(\sigma_j) = 0,\ k'_i(\sigma_j) = \delta_{i,j}, \quad \forall i, j. \end{aligned}$$

Using this basis, the interpolating polynomial can be written as follows:

$$p_{2r}(\lambda) = \sum_{i=1}^{r} |\sigma_i| h_i(\lambda) + \sum_{i=1}^{r} sign(\sigma_i) k_i(\lambda), \quad \forall \lambda. \tag{3.14}$$

Let us define:

$$
\begin{aligned}
&V_i^0 = h_i(\mathcal{A}_\Phi)(W_R - W_L), \quad i = 1, \dots, r,\\
&V_i^1 = k_i(\mathcal{A}_\Phi)(W_R - W_L), \quad i = 1, \dots, r,\\
&W_i^0 = W_{i-1}^0 + V_i^0, \quad i = 1, .., r,\\
&W_i^1 = -\frac{1}{\sigma_{i+1} - \sigma_i} V_i^1, \quad i = 1, ..., I-1,\\
&W_I^{1,-} = \frac{1}{\sigma_I} V_I^1,\\
&W_I^{1,+} = -\frac{1}{\sigma_{I+1}} V_{I+1}^1,\\
&W_i^1 = -\frac{1}{\sigma_{i+1} - \sigma_i} V_{i+1}^1, \quad i = I+1, ..., r-1.
\end{aligned}
$$

Let us consider the function:

$$
R(\sigma) = \begin{cases}
W_L & if \quad \sigma < \sigma_1,\\
W_1 = W_1^0 + W_1^1 & if \quad \sigma_1 < \sigma < \sigma_2,\\
\vdots & \\
W_{I-1} = W_{I-1}^0 + W_{I-1}^1 & if \quad \sigma_{I-1} < \sigma < \sigma_I,\\
W_I^- = W_I^0 + W_I^{1,-} & if \quad \sigma_I < \sigma < 0,\\
W_I^+ = W_I^0 + W_I^{1,+} & if \quad 0 < \sigma < \sigma_{I+1},\\
W_{I+1} = W_{I+1}^0 + W_{I+1}^1 & if \quad \sigma_{I+1} < \sigma < \sigma_{I+2}\\
\vdots & \\
W_{r-1} = W_{r-1}^0 + W_{r-1}^1 & if \quad \sigma_{r-1} < \sigma < \sigma_r\\
W_r^0 & if \quad \sigma_r < \sigma.
\end{cases} \tag{3.15}
$$

Let us verify that R is a SRS. First, reasoning like in the proof of Theorem 2 and taking into account the equality

$$
\sum_{i=1}^{r} h_i(\lambda) = 1, \quad \forall \lambda,
$$

we obtain

$$
W_r^0 = W_R.
$$

Next, (3.4) can be trivially checked. Let us check property (3.5):

$$\begin{aligned}
&\sigma_1\left(W_1^0+W_1^1-W_L\right)+\sum_{i=2}^{I-1}\sigma_i\left(W_i^0+W_i^1-W_{i-1}^0-W_{i-1}^1\right)\\
&\quad+\sigma_I\left(W_I^0+W_I^{1,-}-W_{I-1}^0-W_{I-1}^0\right)+\sigma_{I+1}\left(W_{I+1}^0+W_{I+1}^1-W_I^0-W_I^{1,+}\right)\\
&\quad+\sum_{i=I+2}^{r-1}\sigma_i\left(W_i^0+W_i^1-W_{i-1}^0-W_{i-1}^1\right)+\sigma_r\left(W_r^0-W_{r-1}^0-W_{r-1}^1\right)\\
&\quad=\sum_{i=1}^{r}\sigma_i(W_i^0-W_{i-1}^0)+\sum_{i=1}^{I-1}(\sigma_i-\sigma_{i-1})W_i^1+\sigma_I W_I^{1,-}-\sigma_{I+1}W_I^{1,+}\\
&\qquad+\sum_{i=I+2}^{r-1}(\sigma_i-\sigma_{i-1})W_i^1\\
&\quad=\sum_{i=1}^{r}\sigma_i V_i^0+\sum_{i=1}^{r}V_i^1\\
&\quad=\left(\sum_{i=1}^{r}(\sigma_i h_i+k_i)\right)(\mathcal{A}_\Phi)(W_R-W_L)\\
&\quad=\mathcal{A}_\Phi(W_R-W_L)\\
&\quad=\int_0^1\mathcal{A}(\Phi(\xi;W_L,W_R))\frac{\partial\Phi}{\partial\xi}(\xi;W_L,W_R)d\xi.
\end{aligned}$$

where the Roe property and the equality

$$\left(\sum_{i=1}^{r}(\sigma_i h_i+k_i)\right)(\lambda)=\lambda,\quad\forall\lambda$$

have been used. Therefore, R is a SRS. Let us finally check (3.9):

$$\begin{aligned}
&|\sigma_1|\left(W_1^0+W_1^1-W_L\right)+\sum_{i=2}^{I-1}|\sigma_i|\left(W_i^0+W_i^1-W_{i-1}^0-W_{i-1}^1\right)\\
&\quad+|\sigma_I|\left(W_I^0+W_I^{1,-}-W_{I-1}^0-W_{I-1}^0\right)+|\sigma_{I+1}|\left(W_{I+1}^0+W_{I+1}^1-W_I^0-W_I^{1,+}\right)\\
&\quad+\sum_{i=I+2}^{r-1}|\sigma_i|\left(W_i^0+W_i^1-W_{i-1}^0-W_{i-1}^1\right)+|\sigma_r|\left(W_r^0-W_{r-1}^0-W_{r-1}^1\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{r} |\sigma_i|(W_i^0 - W_{i-1}^0) + \sum_{i=1}^{I-1}(|\sigma_i| - |\sigma_{i-1}|)W_i^1 + |\sigma_I|W_I^{1,-} - |\sigma_{I+1}|W_I^{1,+} \\
&\quad + \sum_{i=I+2}^{r-1}(|\sigma_i| - |\sigma_{i-1}|)W_i^1 \\
&= \sum_{i=1}^{r} |\sigma_i|V_i^0 - \sum_{i=1}^{I} V_i^1 + \sum_{i=I+1}^{r} V_i^1 \\
&= \left(\sum_{i=1}^{r}(|\sigma_i|h_i + sign(\sigma_i)k_i)\right)(\mathcal{A}_\Phi)(W_R - W_L) \\
&= p_{2r}(\mathcal{A}_\Phi)(W_R - W_L),
\end{aligned}
$$

and thus the PVM is equivalent to the SRS.

□

Remark 4. The main advantage of having an expression of a PVM method as a SRS is that this form makes easier to prove properties such as the positivity of the method: see [18].

4 Application to the Roe method

4.1 Standard form

As it has been said in Section 2, given a Roe linearization $\mathcal{A}_\Phi$, the standard form of the Roe method is given by (2.6) with:

$$
D^{\pm}_{i+\frac{1}{2}} = D^{\pm}(W_i^n, W_{i+1}^n) = \frac{1}{2}\mathcal{A}_{i+\frac{1}{2}}(W_{i+1}^n - W_i^n) \pm \frac{1}{2}\left|\mathcal{A}_\Phi(W_i^n, W_{i+1}^n)\right|(W_{i+1}^n - W_i^n),
$$

where

$$
|\mathcal{A}_\Phi(W_i^n, W_{i+1}^n)| = R_\Phi(W_i^n, W_{i+1}^n)|\Lambda_\Phi(W_i^n, W_{i+1}^n)|R_\Phi^{-1}(W_i^n, W_{i+1}^n), \tag{4.1}
$$

being $|\Lambda_\Phi(W_i^n, W_{i+1}^n)|$ the diagonal matrix whose coefficients are the absolute value of the eigenvalues of $\mathcal{A}_\Phi(W_i^n, W_{i+1}^n)$, and $R_\Phi(W_i^n, W_{i+1}^n)$ is the matrix whose i-th column is a right-eigenvector associated to the i-th eigenvalue.

4.2 SRS form

The Roe method can be interpreted as the scheme corresponding to the complete SRS

$$
R(\lambda; W_L, W_R) = \begin{cases} W_0 = W_L, & if \quad \lambda < \lambda_1, \\ W_j, & if \quad \lambda_j < \lambda < \lambda_{j+1} \\ W_N = W_R, & if \quad \lambda_N < \lambda, \end{cases} \tag{4.2}
$$

where λ_i, $i = 1, ..., N$, are the eigenvalues of $\mathcal{A}_\Phi$ and the intermediates states W_i verify:

$$
W_i - W_{i-1} = \alpha_i R_i, \quad i = 1, ..., N,
$$

where R_i is the right eigenvector associated to λ_i and α_i is the i-coordinate of the vector $W_R - W_L$ in the $\mathbb{R}^{N+1}$ basis defined by the eigenvectors. The reciprocal is also true: any complete SRS, i.e. any SRS with N speeds, is equivalent to a Roe method.

4.3 PVM form

According to Theorem 3, a Roe method can be written as the PVM method based on the polynomial $p_N^{i+\frac{1}{2}}$ of degree less or equal than $N-1$ that verifies:

$$p_N(\lambda_i) = |\lambda_i|, \quad i = 1, ..., N. \tag{4.3}$$

Therefore, it can be implemented following the algorithm proposed in Section 2.2 what leads to the Newton Roe method. Since $r = N$, in this case the complexity of the algorithm is (see (2.14)):

$$O\left(2N^3\right).$$

Observe that, with this implementation, it is not necessary to compute the eigenvectors what, in many cases, may constitute an important saving of computational time. If the eigenvectors are explicitly known or easy to compute, still the computation of the absolute value of the matrix requires the inversion of the eigenvectors matrix. If, for instance, the inverse is computed by solving N linear systems using the LU factorization, the complexity would be

$$O\left(\frac{3}{2}N^3 + 2N^3\right) = O\left(\frac{7}{2}N^3\right),$$

so that still the Newton Roe implementation is cheaper.

Remark 5. Let us suppose that a Roe matrix $\mathcal{A}_\Phi$ and some approximations $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N$ of the wave speeds are available that do not coincide with the eigenvalues of the Roe matrix. One could then implement the PVM based on the Roe matrix and the polynomial that interpolates the absolute value function at the approximated speeds. The resulting PVM method would be then equivalent to a new Roe method whose matrix $\widetilde{\mathcal{A}}_\Phi$ has the approximated speeds as eigenvalues and the states V_i given in the proof of Theorem 2 as the corresponding eigenvectors.

4.4 Close or double eigenvalues

In some cases, the Roe matrix has eigenvalues that are very close or even identical (if the system is not strictly hyperbolic). In these cases, the divided differences are not well-defined or involves close to zero denominators. In order to fix this difficulty, the following strategy can be used: let us suppose that $\lambda_k \simeq \lambda_{k+1}$ or $\lambda_k = \lambda_{k+1}$, then instead of considering the Lagrange interpolation:

$$p(\lambda_i) = |\lambda_i|, \quad i = 1, ..., N$$

the following interpolation is used:

$$p(\lambda_i) = |\lambda_i|, \ i \neq k+1, \quad p'(\lambda_k) = sign(\lambda_k).$$

This is again an advantage compared to the implementation of the standard form of the Roe method, since in the case of a double eigenvalue the computation of the Jordan form is necessary to compute the absolute value of the matrix.

5 Models and numerical tests

In this section Roe method and Newton Roe method will be applied to the following models:

- the two-layer shallow water equations,
- the Quadrature-Based Moment equations for rarefied gas.

The methods have been implemented in C++ and run on the linux-subsystem of a 64-bit Windows 10 YOGA 720 with Intel Core i7-7200 2.5 GHz machine. The Eigen library [10] has been used to compute all matrix-vectors operations.

5.1 Two-layer shallow water equations

We consider the homogeneous two-layer 1-D shallow water system (see [5]):

$$\begin{cases} \dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_1}{\partial x} = 0, \\ \dfrac{\partial q_1}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_1^2}{h_1} + \dfrac{1}{2}gh_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial x}, \\ \dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_2}{\partial x} = 0, \\ \dfrac{\partial q_2}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_2^2}{h_2} + \dfrac{1}{2}gh_2^2\right) = -\dfrac{\rho_1}{\rho_2}gh_2\dfrac{\partial h_1}{\partial x}. \end{cases} \tag{5.1}$$

Index 1 refers to the upper layer while index 2 refers to the lower layer. This system uses the following notation:

- $h_i = h_i(x,t) \geq 0$ is the thickness of the i-th layer at the section of coordinate x at time t.
- $q_i = q_i(x,t)$ is the discharg of the i-th layer at the section of coordinate x at time t.
- g is the intensity of the gravitational field.
- ρ_i refers to the constant density of the i-th layer.

The bottom is assumed to be flat. Folowing [8], this system can be written in the form

$$\partial_t W + F(W)_x + B(W)W_x = 0, \tag{5.2}$$

where

$$W = \begin{pmatrix} h_1 \\ q_1 \\ h_2 \\ q_2 \end{pmatrix}, \quad F(W) = \begin{pmatrix} q_1 \\ \dfrac{q_1^2}{h_1} + \dfrac{1}{2}gh_1^2 \\ q_2 \\ \dfrac{q_2^2}{h_2} + \dfrac{1}{2}gh_2^2 \end{pmatrix},$$

$$B(W) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & gh_1 & 0 \\ 0 & 0 & 0 & 0 \\ grh_2 & 0 & 0 & 0 \end{pmatrix},$$

with $r = \rho_1/\rho_2$. System (5.1) can be rewritten in the form:

$$\partial_t W + \mathcal{A}(W)\partial_x W = 0,$$

with

$$\mathcal{A}(W) = J(W) + B(W) = \frac{\partial F}{\partial W}(W) + B(W).$$

The eigenvalues of $\mathcal{A}(W)$ are the roots of the characteristic polynomial:

$$p(\lambda) = (\lambda^2 - 2u_1\lambda + u_1^2 - gh_1)(\lambda^2 - 2u_2\lambda + u_2^2 - gh_2) - rgh_1gh_2, \tag{5.3}$$

where $u_i = q_i/h_i$, $i = 1, 2$. When $r \cong 1$, first order approximations of the eigenvalues were given in [22]:

$$\lambda_{ext}^{\pm} = \frac{u_1h_1 + u_2h_2}{h_1 + h_2} \pm \sqrt{g(h_1 + h_2)}, \tag{5.4}$$

$$\lambda_{int}^{\pm} = \frac{u_1h_2 + u_2h_1}{h_1 + h_2} \pm \sqrt{g'\frac{h_1h_2}{h_1 + h_2}\left(1 - \frac{(u_1 - u_2)^2}{g'(h_1 + h_2)}\right)}, \tag{5.5}$$

where $g' = (1 - r)g$.

The exact expression of the eigenvalues can be obtained by using Ferrari's method to find an analytical solution for quartic equations: following [17], this expression is as follows:

$$\lambda_{ext}^{\pm} = \frac{\frac{a}{2} \pm \sqrt{Z} \pm \sqrt{-A - Z \mp \frac{B}{\sqrt{Z}}}}{2}. \tag{5.6}$$

$$\lambda_{int}^{\pm} = \frac{\frac{a}{2} \pm \sqrt{Z} \mp \sqrt{-A - Z \mp \frac{B}{\sqrt{Z}}}}{2}, \tag{5.7}$$

where:

$$Z = \frac{1}{3}\left(2\sqrt{\Delta_0}cos\left(\frac{\theta}{3}\right) - A\right),$$

$$\theta = arccos\left(\frac{\Delta_1}{2\sqrt{\Delta_0^3}}\right),$$

$$A = 2b - \frac{3a^2}{4},$$

$$B = 2c - ab + a^3 4,$$

$$\Delta_0 = b^2 + 12d - 3ac,$$

$$\Delta_1 = 27a^2d - 9abc + 2b^3 - 72bd + 27c^2,$$

with:

$$a = -2(u_1 + u_2),$$

$$b = u_1 - gh_1 + 4u_1u_2 + u_2^2 - gh_2,$$

$$c = -2u_2(u_1^2 - gh_1) - 2u_1(u_2^2 - gh_2),$$

$$d = (u_1^2 - gh_1)(u_2^2 - gh_2) - rgh_1gh_2,$$

being a, b, c, d the coefficients of the polynomial p written in the form:

$$p(\lambda) = \lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d.$$

Given an eigenvalue λ, an associated eigenvector is given by:

$$R_i = \begin{pmatrix} 1 \\ \lambda \\ \mu \\ \lambda\mu \end{pmatrix}, \tag{5.8}$$

where:

$$\mu = \frac{(\lambda - u_1)^2}{gh_1} - 1.$$

A more detailed description of the calculation of the eigenvalues and eigenvectors of this system can be found in [17].

We consider the Roe matrix of the system based on the family of straight segments

$$\Phi(s; W_L, W_R) = W_L + s(W_R - W_L)$$

described in [5].

5.1.1 Test 1: Dam-break problem

We consider the internal dam-break test introduced in [8]: the equations are solved in the space interval $[0, 10]$ with initial conditions:

$$h_1(x, 0) = \begin{cases} 0.2, & if \quad x < 5, \\ 0.8, & if \quad x \geq 5, \end{cases}$$

$$h_2(x, 0) = \begin{cases} 0.8, & if \quad x < 5, \\ 0.2, & if \quad x \geq 5, \end{cases}$$

$$q_1(x, 0) = q_2(x, 0) = 0.$$

Roe and Newton Roe methods are run in the time interval $[0, 10]$ with CFL $= 0.9$. Since both methods are equivalent, the numerical results are identical to machine precision and therefore we don't compare them. In Table 1 the CPU times in (s) corresponding to both implementations using meshes with different number of cells are shown.

#Cells	Standard Roe	Newton Roe
1250	10.09	9.80
2500	28.96	28.73
5000	102.85	100.09
10000	362.65	349.47

Table 1: Test 1: CPU times in (s) for meshes with different number of cells for the standard Roe scheme and the Newton Roe scheme.

It can be seen that here is a small speedup for the Newton Roe method for this system, for which $N = 4$.

5.2 Hyperbolic Moment Models for rarefied gases

An application that involves larger systems of hyperbolic PDEs is given by continuum models for rarefied gases. In addition to the standard conservation laws, the accurate description of non-equilibrium flows in rarefied gases requires further equations [23]. These equations model the evolution of higher order moments of the underlying particle distribution function, see e.g. [24]. The resulting so-called *moment models* based on the work in [9] are not always hyperbolic, but have been modified to yield hyperbolicity using different frameworks, see [3, 6, 13]. There are many applications of these models, e.g. for shock tube computations and 2D flows, see [15, 7]. The modifications lead to nonconservative, hyperbolic PDE systems, such as the QBME system, which was numerically investigated in [16]. The QBME model is very suitable for the application of the Newton Roe solver, because the eigenvalues are analytically known, whereas the full eigenvector decomposition is difficult to compute. In addition, the model contains more equations leading to a large potential speedup.

5.2.1 QBME moment models in primitive variables

In the following, we will briefly describe the QBME model equations, for more details see [12]. The general form of the model equations is already given in the form

$$\partial_t W_M + \mathcal{A}_M(W_M)\,\partial_x W_M = S(W_M) \tag{5.9}$$

where $W_M \in \mathbb{R}^{M+1}$ is the vector of unknown variables given by

$$W_M = (\rho, v, \theta, f_3, \dots, f_M)^T \tag{5.10}$$

with density ρ, velocity v, temperature θ, and the $f_i, i = 0, \dots, M$ with $f_0 = 1, f_1 = 0, f_2 = 0$ are the so-called *moments*, coefficients from the expansion of an underlying distribution function, and the transport matrix $\mathcal{A}_M \in \mathbb{R}^{(M+1)\times(M+1)}$ depends on W_M.

The right-hand side $S \in \mathbb{R}^{M+1}$ denotes the collision term, which will be the same for all models. It will be approximated by the BGK model [1] and reads

$$S(W_M) = -\frac{1}{\tau} B W_M, \tag{5.11}$$

for non-linear relaxation time $\tau = \frac{\mathrm{Kn}}{\rho}$ with Knudsen number Kn and diagonal matrix

$$B = \mathrm{diag}\,(0, 0, 0, 1, \dots, 1) \in \mathbb{R}^{(M+1)\times(M+1)}, \tag{5.12}$$

corresponding to conservation of mass, momentum and energy and a relaxation of the higher order coefficients towards equilibrium.

For the QBME [12, 13] the system matrix can be written as $\mathcal{A}_M =$

$$\begin{pmatrix}
v & \rho & & & & & \\
\frac{\theta}{\rho} & v & 1 & & & & \\
 & 2\theta & v & \frac{6}{\rho} & & & \\
 & 4f_3 & \frac{\rho\theta}{2} & v & 4 & & \\
-\frac{\theta f_3}{\rho} & 5f_4 & \frac{3f_3}{2} & \theta & v & 5 & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
-\frac{\theta f_{M-2}}{\rho} & Mf_{M-1} & \frac{(M-2)f_{M-2}+\theta f_{M-4}}{2} & -\frac{M(M+1)f_M}{2\theta} & -\frac{3f_{M-3}}{\rho} & \theta & v & M \\
-\frac{\theta f_{M-1}}{\rho} & (M+1)f_M & -f_{M-1} + \frac{\theta f_{M-3}}{2} & \frac{3(M+1)f_M}{\rho\theta} & -\frac{3f_{M-2}}{\rho} & & \theta & v
\end{pmatrix}. \tag{5.13}$$

The characteristic polynomial of the system matrix A_{M} reads

$$\chi_M(\lambda) = \theta^{\frac{M+1}{2}} \mathrm{He}_{M+1}\left(\frac{\lambda - v}{\sqrt{\theta}}\right), \tag{5.14}$$

The real eigenvalues λ_j are thus the shifted and scaled roots of the probabilists' Hermite polynomial He_{M+1}

$$\lambda_j = v + \sqrt{\theta} c_j, \ j = 1, \ldots, M+1, \tag{5.15}$$

where the c_j are now the Hermite roots such that $\mathrm{He}_{M+1}(c_j) = 0$. The system is thus globally hyperbolic.

The entries of the eigenvectors v_j can be derived as

$$\begin{aligned}
v_{j,1} &= \rho \mathrm{He}_0(c_j), \\
v_{j,2} &= \theta^{1/2} \mathrm{He}_1(c_j), \\
v_{j,3} &= \theta \mathrm{He}_2(c_j), \\
v_{j,4} &= \frac{1}{3!} \rho \theta^{3/2} \mathrm{He}_3(c_j), \\
v_{j,5} &= \frac{1}{4!} \rho \theta^{4/2} \mathrm{He}_4(c_j) - f_3 \sqrt{\theta} c_j, \\
v_{j,6} &= \frac{1}{5!} \rho \theta^{5/2} \mathrm{He}_5(c_j) + f_3 \theta \frac{1 - c_j^2}{2} - f_4 \sqrt{\theta} c_j, \\
&\vdots \\
v_{j,M} &= \frac{1}{(M-1)!} \rho \theta^{(M-1)/2} \mathrm{He}_{M-1}(c_j) + f_{M-3} \theta \frac{1 - c_j^2}{2} - f_{M-2} \sqrt{\theta} c_j, \\
v_{j,M+1} &= \frac{1}{M!} \rho \theta^{M/2} \mathrm{He}_M(c_j) + f_{M-2} \theta \frac{1 - c_j^2}{2} - f_{M-1} \sqrt{\theta} c_j + (M+1) f_M \frac{c_j^2 - 1}{2}.
\end{aligned}$$

Remark 6. Other hyperbolic moment models such as the ones mentioned in [2, 14] exist, but the models are all given in a similar form with analytical eigenvalues and eigenvectors in the primitive variables. We thus chose the QBME model exemplarily to focus on the numerical method.

5.2.2 QBME model in partially-conservative variables

To ensure exact conservation of mass, momentum and energy, the moment model above needs to be written in terms of the partially-convective/partially-conservative variables $u_M \in \mathbb{R}^{M+1}$ with

$$U_M = \left(\rho, \rho v, (v^2 + \theta)\rho, v(v^2 + 3\theta)\rho + 6 f_3, f_4, \ldots, f_M\right)^T. \tag{5.16}$$

The transformation matrix from primitive to partially-conservative variables is

$$\frac{\partial U}{\partial W} = \begin{pmatrix}
1 & 0 & & & & \\
v & \rho & & & 0 & \\
v^2 + \theta & 2v\rho & \rho & & & \\
v(v^2 + 3\theta) & 3(v^2 + \theta)\rho & 3v\rho & 6 & & \\
 & & & & 1 & \\
 & 0 & & & \ddots & \\
 & & & & & 1
\end{pmatrix}, \tag{5.17}$$

and its inverse

$$\left(\frac{\partial U}{\partial W}\right)^{-1} = \begin{pmatrix} 1 & 0 & & & & & \\ -\frac{v}{\rho} & \frac{1}{\rho} & & & & 0 & \\ \frac{v^2-\theta}{\rho} & -\frac{2v}{\rho} & \frac{1}{\rho} & & & & \\ -\frac{v(v^2-3\theta)}{6} & \frac{v^2-\theta}{2} & -\frac{v}{2} & \frac{1}{6} & & & \\ & & & & 1 & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}. \tag{5.18}$$

Choosing U_M as the vector of unknowns results in a new system of equations that describes the same physical behavior but with different variables, with the benefit of exact conservation of mass, momentum and energy as the first three variables in U_M. Assuming the new variables U_M can be computed from W_M with Jacobian $\frac{\partial U_M}{\partial W_M} \in \mathbb{R}^{(M+1)\times(M+1)}$, which is a lower triangular matrix depending on the entries of W_M, the change of variables can be performed. We use $\left(\frac{\partial U_M}{\partial W_M}\right) BW_M = \widetilde{B}W_M$, with

$$\widetilde{B} = diag\,(0, 0, 0, 6, 1, \ldots, 1) \in \mathbb{R}^{(M+1)\times(M+1)}, \tag{5.19}$$

The resulting system

$$\partial_t U_M + \widetilde{\mathcal{A}_M}\ \partial_x U_M = -\frac{1}{\tau}\widetilde{B}W_M. \tag{5.20}$$

uses a new transformed system matrix

$$\widetilde{\mathcal{A}_M} = \left(\frac{\partial U_M}{\partial W_M}\right) \mathcal{A}_M \left(\frac{\partial U_M}{\partial W_M}\right)^{-1} \tag{5.21}$$

with the same eigenvalues as $\mathcal{A}_M$.

We know from the previous section, that we can write the primitive system matrix using its eigenvalues (5.15) and corresponding eigenvectors as

$$\mathcal{A}_M = V^{-1}\Lambda V, \tag{5.22}$$

where V is the matrix containing the eigenvectors and Λ is a diagonal matrix containing the eigenvalues. The corresponding eigenvectors and eigenvalues are given in section 5.2.1.

The eigenvalues of the partially-conservative models will be the same as those of the primitive variables. However, the eigenvectors change according to the following computation

$$\begin{aligned} \widetilde{\mathcal{A}_M} &= \frac{\partial U}{\partial W}\mathcal{A}\left(\frac{\partial U}{\partial W}\right)^{-1} \\ &= \frac{\partial U}{\partial W}V^{-1}\Lambda V\left(\frac{\partial U}{\partial W}\right)^{-1} \\ &= \widetilde{V}^{-1}\Lambda\widetilde{V} \end{aligned}$$

for $\widetilde{V}$ the corresponding eigenvector matrix of the partially-conservative system matrix. This matrix is thus given by

$$\widetilde{V} = V\left(\frac{\partial U}{\partial W}\right)^{-1}. \tag{5.23}$$

As V and $\left(\frac{\partial U}{\partial W}\right)^{-1}$ are analytically given, the exact eigenvectors can be computed analytically during the simulation, even though already their evaluation might be computationally expensive during a standard Roe scheme.

The QBME model in partially-conservative variables can be computed analytically by following the transformation of the variables. The final system reads

$$\widetilde{\mathcal{A}}_{\text{QBME}} = \widetilde{\mathcal{A}}_{M,1} + \widetilde{\mathcal{A}}_{M,2}. \tag{5.24}$$

with $\widetilde{\mathcal{A}}_{M,1} =$

$$\begin{pmatrix}
0 & 1 & & & & & & \\
0 & 0 & 1 & & & & & \\
0 & 0 & 0 & 1 & & & & \\
-v^4+6v^2\theta-3\theta^2-\frac{24vf_3}{\rho} & 4\left(v^3-3v\theta+\frac{6f_3}{\rho}\right) & -6v^2+6\theta & 4v & 24 & & & \\
\frac{3v^2f_3-5\theta f_3-10vf_4}{2\rho}-\frac{v(v^2-3\theta)\theta}{6} & \frac{\theta}{2}(v^2-\theta)\frac{-3vf_3+5f_4}{\rho} & -\frac{v\theta}{2}+\frac{3f_3}{2\rho} & \frac{\theta}{6} & v & 5 & & \\
b_{1,5} & b_{2,5} & b_{3,5} & b_{4,5} & \theta & v & 6 & \\
\vdots & \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots \\
b_{1,M-1} & b_{2,M-1} & b_{3,M-1} & b_{4,M-1} & & & \theta & v & M \\
b_{1,M} & b_{2,M} & b_{3,M} & b_{4,M} & & & & \theta & v
\end{pmatrix}, \tag{5.25}$$

and the following entries in the first four columns

$$\begin{aligned}
b_{1,i} &= \frac{f_{i-2}\left(u^3-3\theta u\right)+\theta f_{i-3}\left(u^2-\theta\right)+f_{i-1}\left((i-1)u^2-\theta(i+1)\right)-2(i+1)uf_i}{2\rho}, \\
b_{2,i} &= \frac{3f_{i-2}\left(\theta-u^2\right)-2\theta uf_{i-3}-2(i-1)uf_{i-1}+2(i+1)f_i}{2\rho}, \\
b_{3,i} &= \frac{\theta f_{i-3}+3uf_{i-2}+(i-1)f_{i-1}}{2\rho}, \\
b_{4,i} &= -\frac{f_{i-2}}{2\rho},
\end{aligned}$$

Note that for entries $b_{1,i}$, we need to use the coefficient $f_2 = 0$.

The second matrix in (5.24) contains seven additional terms in the last two rows such that the modification is given by

$$\widetilde{\mathcal{A}}_{M,2} = \frac{M\,(M+1)}{2\rho\theta}\begin{pmatrix} & & & & \emptyset \\ \widehat{m}_{M-1,1} & \widehat{m}_{M-1,2} & \widehat{m}_{M-1,3} & & \\ \widehat{m}_{M,1} & \widehat{m}_{M,2} & \widehat{m}_{M,3} & \widehat{m}_{M,4} & \end{pmatrix}, \tag{5.26}$$

The terms in the second but last row are given by

$$\begin{aligned}
\widehat{m}_{M-1,1} &= f_M\left(\theta-u^2\right), \\
\widehat{m}_{M-1,2} &= 2uf_M, \\
\widehat{m}_{M-1,3} &= -f_M.
\end{aligned}$$

Note, that for $M = 4$, the three entries above need to be multiplied by 6, due to the variable transformation that basically scales the fourth equation. This does not affect the other cases with $M \neq 4$.

The additional entries in the last row are defined as

$$\begin{aligned}
\widehat{m}_{M,1} &= \frac{\theta^2 f_{M-1} - u^3 f_M - \theta u(u f_{M-1} - 3 f_M)}{M}, \\
\widehat{m}_{M,2} &= \frac{\left(-3\theta f_M + 3u^2 f_M + 2\theta u f_{M-1}\right)}{M}, \\
\widehat{m}_{M,3} &= -\frac{(\theta f_{M-1} + 3u f_M)}{M}, \\
\widehat{m}_{M,4} &= \frac{f_M}{M}.
\end{aligned}$$

Similarly as in the shallow water model, we use a linear path with one Gauss quadrature point to compute the Roe linearization of the system matrix and for the numerical treatment of the source term $S_{i+\frac{1}{2}}$ we simply evaluate S in $\frac{W_{M_i}+W_{M_{i+1}}}{2}$. For more details on the proper numerical discretization of the nonconservative QBME system, we refer to [16].

5.2.3 Test 2: Shock tube case

The one-dimensional shock tube problem is considered by choosing the initial conditions

$$W_M(0,x) = \begin{cases} W_M^L & \text{if } x < 0, \\ W_M^R & \text{if } x > 0, \end{cases} \tag{5.27}$$

and non-linear relaxation time $\tau = \frac{\mathrm{Kn}}{\rho}$.

We consider the system in primitive and partially-conservative variables. According to the tests in [2], the left and right states are chosen as

$$W_M^L = (7, 0, 1, 0, \ldots, 0)^T, \qquad W_M^R = (1, 0, 1, 0, \ldots, 0)^T, \tag{5.28}$$

or equivalently

$$U_M^L = (7, 0, 7, 0, \ldots, 0)^T, \qquad U_M^R = (1, 0, 1, 0, \ldots, 0)^T, \tag{5.29}$$

corresponding to a jump in density at the discontinuity at $x = 0$.

We consider $\mathrm{Kn} = 0.05$ representing a relatively small Knudsen number close to the continuum flow regime and we will take $x \in [-2, 2]$, $CFL = 0.3$, and $t_{end} = 0.3s$.

Primitive variables

We are going to compare the runtime for $M = 5$ (i.e. the system has 6 equations) and $M = 11$ (i.e. the system has 12 equations) in primitive variables. We show in Figure 1 the numerical solution of the problem (5.27) using both ways of writing the Roe scheme in order to see that both give the same result. A detailed comparison of numerical solutions for this test case can be found in [16]. In Tables 2 and 3 we show the CPU runtimes in (s) for different discretizations of the domain and for $M = 5$ and $M = 11$, respectively, using the standard Roe scheme and the new Newton form. Here we observe a big difference in CPU times between both ways of writing the Roe scheme, due to the larger complexity of the model and the additional number of equations. We see that taking 12 variables ($M = 11$) the difference is getting bigger.

In Table 4 and in Figure 2 we observe that for 1000 cells of the domain, as we increase the number of moments M, the speedup of the new Newton Roe scheme is increasing.

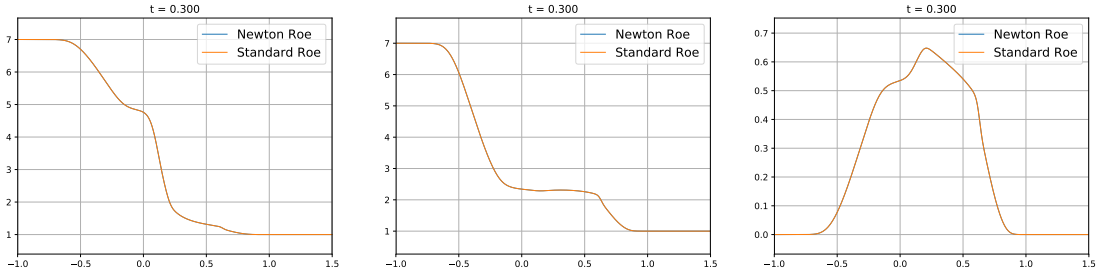

Figure 1: Numerical solution of the problem (5.27) computed in the primitive variables with $M = 5$, 1000 cells, $CFL = 0.3$ at time $t = 0.3$. Starting from the left the density ρ, pressure $p = \rho\theta$ and velocity u are plotted.

#Cells	Standard Roe	Newton Roe	Speedup
125	0.31	0.22	1.40
250	0.62	0.36	1.74
500	1.61	0.63	2.56
1000	4.14	1.49	2.79

Table 2: Test 2: CPU times in (s) for different number of cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with $M = 5$ (average of 5 runs) and using primitives variables.

#Cells	Standard Roe	Newton Roe	Speedup
125	0.77	0.38	2.04
250	2.10	0.73	2.88
500	7.40	2.55	2.91
1000	21.40	5.92	3.61

Table 3: Test 2: CPU times in (s) for different number of cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with $M = 11$ (average of 5 runs) and using primitives variables.

M	Standard Roe	Newton Roe	Speedup
5	4.14	1.44	2.03
7	8.14	2.85	2.86
9	14.78	4.60	3.22
11	21.40	5.92	3.61

Table 4: Test 2: CPU times in (s) for 1000 cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with different number of moments M (average of 5 runs) and using primitives variables.

Partially-conservative variables

We are going to compare the runtime for $M = 5$ (i.e. the system has 6 equations) and $M = 13$ (i.e. the system has 14 equations) in partially-conservative variables. This time we are not going to show the results of the numerical schemes because the standard Roe

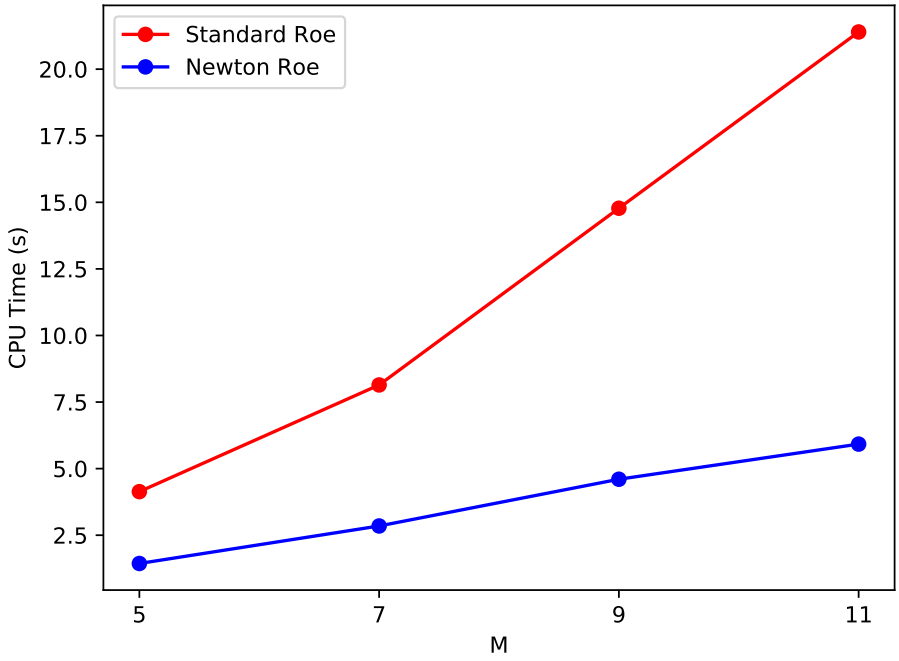

Figure 2: Number of moments M vs CPU time (s) for the standard Roe scheme and its Newton's form using primitive variables.

scheme and the Newton Roe scheme again give the same solution. A detailed comparison of numerical solutions for this test case can be found in [16]. In Tables 5 and 6 we show the CPU runtimes in (s) for different discretizations of the domain and for $M = 5$ and $M = 13$, respectively, using the standard Roe scheme and the new Newton form. Again we observe a big difference in CPU times between both ways of writing the Roe scheme and this difference is even bigger than when we were using primitive variables because with partially-conservative variables we have to add the computation of the eigenvectors as was seen in (5.23). We see that taking 14 variables ($M = 13$) the difference is increasing.

#Cells	Standard Roe	Newton Roe	Speedup
250	0.72	0.30	2.37
500	2.15	0.65	3.29
1000	5.30	1.65	3.21
2000	18.32	5.53	3.32

Table 5: Test 2: CPU times in (s) for different number of cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with $M = 5$ (average of 5 runs) and using partially-conservative variables.

In Table 7 and in Figure 3 we observe that for 1000 cells of the domain, as we increase the number of moments M, the speedup of the new Newton Roe scheme is increasing. As observed before, the differences between both ways of writing the Roe scheme are bigger using the partially-conservative variables. The new Newton Roe solver yields a significant

#Cells	Standard Roe	Newton Roe	Speedup
250	3.70	1.06	3.49
500	12.23	2.87	4.27
1000	40.62	9.09	4.47
2000	166.48	39.10	4.26

Table 6: Test 2: CPU times in (s) for different number of cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with $M = 13$ (average of 5 runs) and using partially-conservative variables.

speedup in all test cases for the QBME model.

M	Standard Roe	Newton Roe	Speedup
5	18.32	5.53	3.32
7	39.10	11.10	3.52
9	71.97	19.23	3.74
11	106.73	25.84	4.13
13	166.48	39.10	4.26

Table 7: Test 2: CPU times in (s) for 2000 number of cells of the domain obtained for the usual Roe scheme and the Newton Roe scheme with different number of moments M (average of 5 runs) and using partially-conservative variables.

Remark 1. *We only present results for odd M here as the computation using even M leads to very different runtimes in our simulation software, due to specifications of the underlying linear algebra libraries. However, the resulting flow solutions are still the same as for the standard Roe scheme.*

6 Conclusions

In this paper, an efficient implementation of the PVM methods that are based on interpolation polynomials has been presented: the Newton form of the polynomial is used to reduce the number of calculations. Next, the relation between SRS and PVM, already studied in [18], has been revisited. In particular, it has been shown that many SRS can be interpreted as PVM methods based on a Lagrange interpolation polynomial, what allows one to use the implementation based on the Newton form of the polynomial. In particular, Roe method can be interpreted in terms of a complete SRS and thus as a PVM method, what allows us to implement it using the Newton form of the polynomial. We have numerically compared the efficiency of the standard implementation of Roe method and the new one for two different models: the two-layer shallow water equations and the Quadrature-Based Moment equations for rarefied gases. According to our results, a small speedup has been obtained using the Newton Roe method compared to the standard one for the two-layer shallow-water system, as the number of equations is not big enough. In the case of the QBME model the speedup increases with the number of moments: Newton Roe method is about 3.5 times faster than the standard Roe method for the 11 moment equations in primitive variables. In the case of the partially-conservative formulation of the QBME model the results are even better: Newton Roe method is 4.1 times faster than

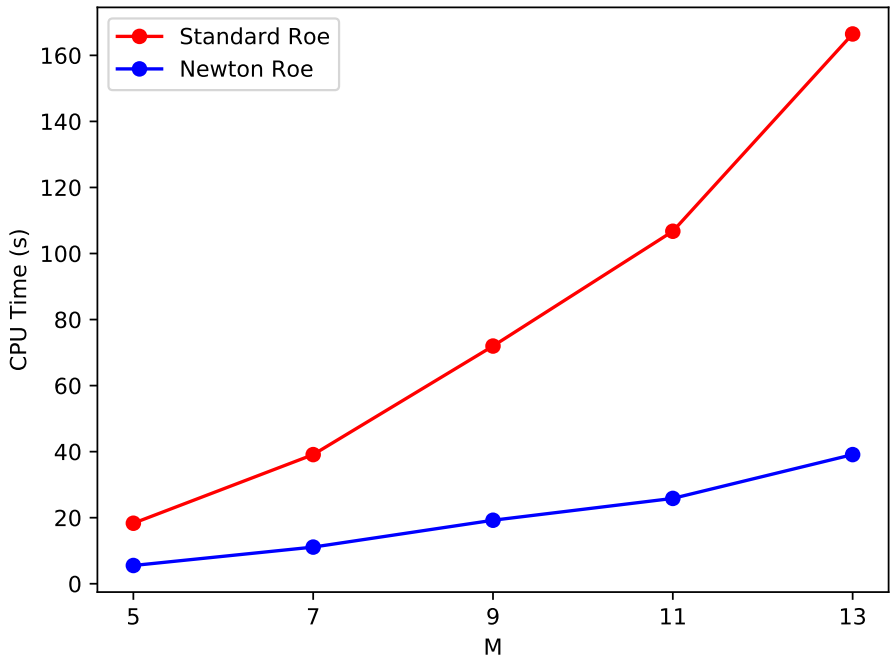

Figure 3: Number of moments M vs CPU time (s) for the standard Roe scheme and its Newton's form using partially-conservative variables.

the standard one, due to the fact that the standard implementation requires the computation of the eigenvectors. Moreover, this factor increases when the number of moments increases. Therefore, we can conclude that the Newton Roe method yields an improvement of the standard Roe scheme for systems with a large number of equations.

References

[1] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. 1. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511–525, 1954. 18

[2] Z. Cai, Y. Fan, and R. Li. Globally hyperbolic regularization of Grad's moment system in one dimensional space. *Commun. Math. Sci.*, 11(2):547–571, 2013. 19, 22

[3] Z. Cai, Y. Fan, and R. Li. Globally hyperbolic regularization of Grad's moment system. *Communications on Pure and Applied Mathematics*, 67(3):464–518, 2014. 18

[4] M.J. Castro and E. Fernández-Nieto. A class of computationally fast first order finite volume solvers: PVM methods. *SIAM Journal on Scientific Computing*, 34(4):A2173–A2196, 2012. 2, 5

[5] M.J. Castro, J. Macías, and C. Parés. A Q-scheme for a class of systems of coupled conservation laws with source term. Application to a two-layer 1-D shallow water

system. *ESAIM: Mathematical Modelling and Numerical Analysis*, 35(1):107–127, 2001. 2, 15, 17

[6] Y. Fan, J. Koellermeier, J. Li, R. Li, and M. Torrilhon. Model reduction of kinetic equations by operator projection. *Journal of Statistical Physics*, 162(2):457–486, 2016. 18

[7] Y. Fan and R. Li. Globally hyperbolic moment system by generalized Hermite expansion. *Scientia Sinica Mathematica*, 45(10)(10):1635–1676, 2015. 18

[8] E. Fernández-Nieto, M.J. Castro, and C. Parés. On an intermediate field capturing riemann solver based on a parabolic viscosity matrix for the two-layer shallow water system. *Journal of Scientific Computing*, 48(1-3):117–140, 2011. 15, 17

[9] H. Grad. On the kinetic theory of rarefied gases. *Communications on Pure and Applied Mathematics*, 2(4):331–407, 1949. 18

[10] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 15

[11] A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM review*, 25(1):35–61, 1983. 2

[12] J. Koellermeier. *Derivation and numerical solution of hyperbolic moment equations for rarefied gas flows.* dissertation, RWTH Aachen University, Aachen, 2017. 2, 18

[13] J. Koellermeier, R. Pascal Schaerer, and M. Torrilhon. A framework for hyperbolic approximation of kinetic equations using quadrature-based projection methods. *Kinetic and Related Models*, 7(3):531–549, 2014. 18

[14] J. Koellermeier and M. Torrilhon. Simplified hyperbolic moment equations. In *Proceedings of the 16th International Conference on Hyperbolic Problems*, 2016. 19

[15] J. Koellermeier and M. Torrilhon. Numerical solution of hyperbolic moment models for the Boltzmann equation. *European Journal of Mechanics - B/Fluids*, 64:41–46, 2017. 18

[16] J. Koellermeier and M. Torrilhon. Numerical study of partially conservative moment equations in kinetic theory. *Communications in Computational Physics*, 21(04)(4):981–1011, 2017. 18, 22, 24

[17] N. Krvavica, M. Tuhtan, and G. Jelenić. Analytical implementation of roe solver for two-layer shallow water equations with accurate treatment for loss of hyperbolicity. *Advances in Water Resources*, 122:187–205, 2018. 3, 16, 17

[18] T. Morales de Luna, M. J. Castro, and C. Parés. Relation between PVM schemes and simple riemann solvers. *Numerical Methods for Partial Differential Equations*, 30(4):1315–1341, 2014. 2, 8, 13, 25

[19] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44(1):300–321, 2006. 1, 2, 4, 7

[20] C. Parés and M.J. Castro. On the well-balance property of Roe's method for nonconservative hyperbolic systems. applications to shallow-water systems. *ESAIM: M2AN*, 38(5):821–852, 2004. 1

[21] P.L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357 – 372, 1981. 1

[22] J.B. Schijf and J.C. Schönfled. Theoretical considerations on the motion of salt and fresh water. IAHR, 1953. 16

[23] H. Struchtrup. *Macroscopic Transport Equations for Rarefied Gas Flows: Approximation Methods in Kinetic Theory*. Interaction of Mechanics and Mathematics. Springer Berlin Heidelberg, 2006. 18

[24] M. Torrilhon. Modeling nonequilibrium gas flow based on moment equations. *Annual Review of Fluid Mechanics*, 48(1):429–458, 2016. 18

[25] I. Toumi. A weak formulation of roe's approximate riemann solver. *Journal of computational physics*, 102(2):360–373, 1992. 1, 3