MOX–Report No. 11/2014

An offline-online Riemann solver for one-dimensional systems of conservation laws

Taddei, T.; Quarteroni, A.; Salsa, S.

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

An offline-online Riemann solver for one-dimensional systems of conservation laws

T. Taddei[♣], A. Quarteroni[♯†], S. Salsa[§],

January 14, 2014

[♣] Mechanical Engineering Department, Massachusetts Institute of Technology,
77 Mass Avenue, Cambridge, MA 02139, USA
ttaddei@mit.edu

[♯] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
alfio.quarteroni@polimi.it

[†] Modelling and Scientific Computing, Institute of Analysis and Scientific Computing,
École Polytechnique Féderale de Lausanne,
Station 8, EPFL, CH-1015 Lausanne, Switzerland
alfio.quarteroni@epfl.ch

[§] Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
sandro.salsa@polimi.it

Keywords: nonlinear conservation laws, model reduction, reduced basis method.

AMS Subject Classification: 65M08, 65M15, 65M25, 35L65, 35L67

Abstract

In this paper we present a new technique based on two different phases, here called *offline and online stages*, for the solution of the Riemann problem for one-dimensional hyperbolic systems of conservation laws. After theoretically motivating our offline-online technique, we prove its effectiveness by means of two numerical examples.

1 Introduction and motivations

Let us consider the one-dimensional system of conservation laws:

$$\mathbf{u}_t + \mathbf{F}(\mathbf{u})_x = 0 \quad (x,t) \in \mathbb{R} \times (0,\infty) \tag{1}$$

where $\mathbf{u} : \mathbb{R} \times (0,\infty) \to \mathbb{R}^m$ and $\mathbf{F} \in C^2(\mathbb{R}^m;\mathbb{R}^m)$ for some integer $m \geq 2$. We assume that the flux $\mathbf{F}$ is strictly hyperbolic, i.e. the Jacobian matrix $D\mathbf{F}$ is diagonizable with distinct real eigenvalues.

For the numerical solution to (1), a wide class of finite volume methods - e.g., Godunov method ([5]) and its high-order extensions - requires the approximation of the solution to Riemann problems at cell interfaces between the neighboring cell averages. Then, in practical computations the Riemann problem is solved an extremely large number of time, representing therefore the single most burdensome task in the numerical method.

This is why since early sixties much effort has been devoted to develop efficient Riemann solvers either exact or approximate. Exact Riemann solvers are based on the solution of a certain number of algebraic nonlinear equations via an iterative procedure; this computational effort depends therefore on the accuracy of the initial guess we provide.

To alleviate the computational load, in this paper we propose and analyze a strategy based on the decomposition of the computational work into two different phases:

- *offline stage*: we build a polynomial initial guess for the nonlinear equations by solving a certain number of Riemann problems with random left and right data;
- *online stage*: for each time step and for each cell interface we solve the Riemann problem by using the previously computed polynomial guess to initialize the iterative solver.

In this work we show that the polynomial initial guess built offline is particularly accurate and determines a significant improvement in the performance of the iterative solver. Furthermore, while the offline stage is carried out only once, the online stage is repeated many times. For these reasons we expect that the additional work required by the offline step can be rewarded by the computational gain obtained during the online stage.

Of course, the idea of decoupling the computational effort into two different stages is well-established in the context of Reduced Order Modelling for PDEs (see, e.g., [8] and [2]). However, at the best of our knowledge this paper represents the first attempt to exploit this technique in the development of Riemann solvers for systems of conservation laws.

The outline of the paper is as follows. In Section 2 we first introduce some notation and the main classical results concerning the Riemann problem. Then we present a new theorem which stands at the base of our strategy. In Section 3 we introduce the offline online decomposition algorithm and finally in Sections 4 and 5 we apply our method to the numerical solution by finite volumes of both the p-system and the Euler equations for ideal gases.

2 Problem setting: some theoretical results

This section deals with the analysis of the one dimensional Riemann problem:

$$\begin{cases} \mathbf{u}_t + \mathbf{F}(\mathbf{u})_x = 0, & (x,t) \in \mathbb{R} \times (0,\infty) \\ \mathbf{u}(x,0) = \mathbf{g}(x) = \begin{cases} \mathbf{u}_l & x < 0 \\ \mathbf{u}_r & x \geq 0 \end{cases} & x \in \mathbb{R} \end{cases} \tag{2}$$

for two given vectors $\mathbf{u}_l, \mathbf{u}_r \in \mathbb{R}^m$. With the purpose of analyzing the way $\mathbf{u}$ depends on $(\mathbf{u}_l, \mathbf{u}_r)$, we need some definitions and notable results. Our analysis is largely inspired by [4].

For all $\mathbf{z} \in \mathbb{R}^m$ and $k = 1, \dots, m$, let us introduce the triplets $(\lambda_k, \mathbf{r}_k, \mathbf{l}_k)$ such that

$$D\mathbf{F}(\mathbf{z})\mathbf{r}_k(\mathbf{z}) = \lambda_k(\mathbf{z})\mathbf{r}_k(\mathbf{z}), \quad D\mathbf{F}(\mathbf{z})^T\mathbf{l}_k(\mathbf{z}) = \lambda_k(\mathbf{z})\mathbf{l}_k(\mathbf{z}),$$

$$\lambda_1(\mathbf{z}) < \ldots < \lambda_m(\mathbf{z}).$$

We suppose that, for each $k = 1, \dots, m$, the eigenpair $(\lambda_k, \mathbf{r}_k)$ satisfies one of the following conditions:

$$\begin{cases} \nabla\lambda_k(\mathbf{z}) \cdot \mathbf{r}_k(\mathbf{z}) \neq 0 \ \forall\, \mathbf{z} \in \mathbb{R}^m & \text{(genuinely nonlinear eigenpair)}, \\ \nabla\lambda_k(\mathbf{z}) \cdot \mathbf{r}_k(\mathbf{z}) = 0 \ \forall\, \mathbf{z} \in \mathbb{R}^m & \text{(linearly degenerate eigenpair)}. \end{cases} \tag{3}$$

For a given $\mathbf{u}_0 \in \mathbb{R}^m$, the *k-th rarefaction curve* $\mathbf{R}_k(\mathbf{u}_0)$ is defined as the path in $\mathbb{R}^m$ of the solution of the ODE $\mathbf{v}'(s) = \mathbf{r}_k(\mathbf{v}(s))$ in a neighborhood of $s = 0$, with initial condition $\mathbf{v}(0) = \mathbf{u}_0$. Further, we define the *shock set*:

$$S(\mathbf{u}_0) = \{\mathbf{z} \in \mathbb{R}^m : \mathbf{F}(\mathbf{z}) - \mathbf{F}(\mathbf{u}_0) = \sigma(\mathbf{z} - \mathbf{u}_0) \, \text{for a constant } \sigma = \sigma(\mathbf{z}, \mathbf{u}_0)\}, \tag{4}$$

which consists of the union of m differentiable curves $\mathbf{S}_k(\mathbf{u}_0)$, $k = 1, \dots, m$, such that each curve $S_k(\mathbf{u}_0)$ passes through $\mathbf{u}_0$, with tangent $\mathbf{r}_k(\mathbf{u}_0)$ (see [4, Th.2, section 11.2.3]). It follows that the curves $\mathbf{R}_k(\mathbf{u}_0)$ and $\mathbf{S}_k(\mathbf{u}_0)$ agree at least to first order at $\mathbf{u}_0$. In the linearly degenerate case these curves in fact coincide.

In order to avoid non-physical solutions of the Riemann problem we introduce the sets

$$\begin{cases} \mathbf{R}_k^+(\mathbf{u}_0) = \{\mathbf{z} \in \mathbf{R}_k(\mathbf{u}_0) : \lambda_k(\mathbf{z}) > \lambda_k(\mathbf{u}_0)\}, \\ \mathbf{R}_k^-(\mathbf{u}_0) = \{\mathbf{z} \in \mathbf{R}_k(\mathbf{u}_0) : \lambda_k(\mathbf{z}) < \lambda_k(\mathbf{u}_0)\}; \\ \mathbf{S}_k^+(\mathbf{u}_0) = \{\mathbf{z} \in \mathbf{S}_k(\mathbf{u}_0) : \lambda_k(\mathbf{u}_0) < \sigma(\mathbf{z}, \mathbf{u}_0) < \lambda_k(\mathbf{z})\}, \\ \mathbf{S}_k^-(\mathbf{u}_0) = \{\mathbf{z} \in \mathbf{S}_k(\mathbf{u}_0) : \lambda_k(\mathbf{z}) < \sigma(\mathbf{z}, \mathbf{u}_0) < \lambda_k(\mathbf{u}_0)\}; \end{cases} \tag{5}$$

and the following curves

$$\mathbf{T}_k(\mathbf{u}_0) := \mathbf{R}_k^+(\mathbf{u}_0) \cup \{\mathbf{u}_0\} \cup \mathbf{S}_k^-(\mathbf{u}_0), \tag{6}$$

for $k = 1, \ldots, m$. Thanks to the previous observations, if $(\lambda_k, \mathbf{r}_k)$ is genuinely nonlinear, $\mathbf{T}_k(\mathbf{u}_0)$ admits a C^1 parametrization $\epsilon \to \mathbf{\Upsilon}_k(\epsilon, \mathbf{u}_0)$ such that $\left\{\mathbf{\Upsilon}_k(\epsilon, \mathbf{u}_0) : \epsilon > 0\right\} = \mathbf{R}_k^+(\mathbf{u}_l)$ and $\left\{\mathbf{\Upsilon}_k(\epsilon, \mathbf{u}_0) : \epsilon < 0\right\} = \mathbf{S}_k^-(\mathbf{u}_l)$. On the other hand, if $(\lambda_k, \mathbf{r}_k)$ is linearly degenerate, $\mathbf{T}_k(\mathbf{u}_0) := \mathbf{R}_k(\mathbf{u}_0) = \mathbf{S}_k(\mathbf{u}_0)$ and it is still possible to parametrize the curve through a regular function $\epsilon \to \mathbf{\Upsilon}_k(\epsilon, \mathbf{u}_0)$. We observe that each $\mathbf{T}_k(\mathbf{u}_0)$ glues together the physically relevant parts of the k-th rarefaction and k-th shock curve.

We now recall the definition of "physical" (entropic) solution to the Riemann problem (2), ([4]).

Definition 2.1. $\mathbf{u} \in L^\infty(\mathbb{R} \times (0, \infty); \mathbb{R}^m)$ *is said to be an integral solution to (2) if it satisfies:*

$$\int_0^\infty \int_{\mathbb{R}} \mathbf{u} \cdot \mathbf{v}_t + \mathbf{F}(\mathbf{u}) \cdot \mathbf{v}_x \, dxdt + \int_{\mathbb{R}} \mathbf{g} \cdot \mathbf{v}|_{t=0} \, dx = 0 \tag{7}$$

for all $\mathbf{v} \in C_0^1\left(\mathbb{R}^2; \mathbb{R}^m\right)$. *Furthermore*[1], $\mathbf{u}$ *is said to be admissible (or entropic) if for all* $(x,t) \in \mathbb{R} \times (0, \infty)$ *the left and right limits* $\mathbf{u}^\pm(x,t) := \lim_{y \to x^\pm} \mathbf{u}(y,t)$ *do exist and, if* $\mathbf{u}^+(x,t) \neq \mathbf{u}^-(x,t)$, *then* $\mathbf{u}^+(x,t) \in S_k^-(\mathbf{u}^-(x,t))$ *for some* $k \in \{1, \ldots, m\}$.

On the ground of the previous properties and definitions, we can introduce three special types of solutions to problem (2), ([4]).

Lemma 2.1. *Let* $(\lambda_k, \mathbf{r}_k)$ *be linearly degenerate and let* $\mathbf{u}_r \in S_k(\mathbf{u}_l)$, *for some* $k \in \{1, \ldots, m\}$. *Then,*

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{u}_l & x < \sigma t \\ \mathbf{u}_r & x > \sigma t \end{cases} \qquad \sigma = \lambda_k(\mathbf{u}_l) = \lambda_k(\mathbf{u}_r) \tag{8}$$

is called k-contact discontinuity *and it represents an admissible integral solution to the Riemann problem (2).*

Lemma 2.2. *Let* $(\lambda_k, \mathbf{r}_k)$ *be genuinely nonlinear and let* $\mathbf{u}_r \in S_k^-(\mathbf{u}_l)$. *Then*

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{u}_l & x < \sigma t \\ \mathbf{u}_r & x > \sigma t \end{cases} \qquad \sigma = \sigma(\mathbf{u}_l, \mathbf{u}_r) \tag{9}$$

is called k-shock wave *and is an admissible integral solution to the Riemann problem (2).*

[1] This entropy condition is well suited for the solution to the Riemann problem. We refer to [4, Section 11.4] for more general entropy criteria.

Lemma 2.3. *Let $(\lambda_k, \mathbf{r}_k)$ be genuinely nonlinear and let $\mathbf{u}_r \in R_k^+(\mathbf{u}_l)$. Then, there exists a continuous admissible integral solution $\mathbf{u}$ of Riemann problem (2), which is a k-*simple wave *constant along lines through the origin. More precisely, the solution is*

$$\mathbf{u}(x,t) = \mathbf{v}\left(\Theta\left(\frac{x}{t}\right)\right) \text{ where } \begin{cases} \mathbf{v}(\Theta_l) = \mathbf{u}_l, \quad \mathbf{v}(\Theta_r) = \mathbf{u}_r; \\ \mathbf{v}'(s) = \mathbf{r}_k(\mathbf{v}(s)); \\ \Theta(s) := \begin{cases} \Theta_l & s < F_k'(\Theta_l), \\ G_k(s) & F_k'(\Theta_l) \leq s < F_k'(\Theta_r), \\ \Theta_r & s \geq F_k'(\Theta_r), \end{cases} \\ F_k(s) = \int_0^s \lambda_k(\mathbf{v}(t))\,dt, \, G_k = \left(F_k'\right)^{-1}. \end{cases} \tag{10}$$

*We refer to this solution as to k-*th rarefaction wave*.*

Let us introduce the vector $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_m) \in \mathbb{R}^m$ and the corresponding states $\mathbf{u}_1, \dots, \mathbf{u}_{m-1}$ such that:

$$\mathbf{u}_1 = \boldsymbol{\Upsilon}_1(\epsilon_1, \mathbf{u}_l), \, \mathbf{u}_2 = \boldsymbol{\Upsilon}_2(\epsilon_2, \mathbf{u}_1), \dots, \mathbf{u}_{m-1} = \boldsymbol{\Upsilon}_{m-1}(\epsilon_{m-1}, \mathbf{u}_{m-2}).$$

Finally, we introduce the map:

$$\boldsymbol{\epsilon} \to \boldsymbol{\Upsilon}(\boldsymbol{\epsilon}, \mathbf{u}_l), \quad \boldsymbol{\Upsilon}(\boldsymbol{\epsilon}, \mathbf{u}_l) = \boldsymbol{\Upsilon}_m(\epsilon_m, \mathbf{u}_{m-1}). \tag{11}$$

Due to the fact that for each $k \in \{1, \dots, m\}$ $\mathbf{T}_k \in C^1$, the map $\boldsymbol{\Upsilon}(\cdot, \mathbf{u}_l)$ is smooth with respect to the first argument. Next theorem guarantees that if $\mathbf{u}_l$ and $\mathbf{u}_r$ are sufficiently close to each other, there exists only one admissible solution to Riemann problem and it consists of a sequence of rarefaction waves, shock waves, and/or contact discontinuities. We refer to [4, Th. 4, section 11.2.4] for the proof.

Theorem 2.1. *For each $k = 1, \dots, m$ assume that the pair $(\lambda_k, \mathbf{r}_k)$ is either genuinely nonlinear or else linearly degenerate. Suppose further the left state $\mathbf{u}_l$ is given.*

Then, there exists a neighborhood $\mathcal{N}(\mathbf{u}_l)$ of the left state $\mathbf{u}_l$ such that if $\mathbf{u}_r \in \mathcal{N}(\mathbf{u}_l)$, there exists a unique admissible (entropic) integral solution $\mathbf{u} \in L^\infty(\mathbb{R} \times (0, \infty); \mathbb{R}^m)$ to the Riemann problem, which is constant on lines through the origin. Furthermore, the solution consists of the (possibly coincident) states $\mathbf{u}_l, \mathbf{u}_1, \dots, \mathbf{u}_{m-1}, \mathbf{u}_r$, which are separated by contact discontinuities, shock waves or rarefaction waves, and there exists $\boldsymbol{\epsilon} \in \mathbb{R}^m$ such that $\mathbf{u}_1 = \boldsymbol{\Upsilon}_1(\epsilon_1, \mathbf{u}_l), \dots, \mathbf{u}_r = \boldsymbol{\Upsilon}_m(\epsilon_m, \mathbf{u}_{m-1})$.

Thanks to Theorem 2.1 and to Lemmas 2.1, 2.2 and 2.3, we can obtain a closed formula of the solution by computing the constant states $\mathbf{u}_1, \dots, \mathbf{u}_{m-1}$. Therefore, the differential problem (2) can be reformulated as the problem of determining $m - 1$ states in $\mathbb{R}^m$. While the solution to (2), as a function of the parameters, the space and time, does not depend smoothly on the parameters (i.e., $\mathbf{u} : \mathcal{D} \times \mathbb{R} \times (0, \infty) \to \mathbb{R}^m$ is not regular with respect to the first argument), the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{m-1} \in \mathbb{R}^m$ depend smoothly on data as stated in the next theorem.

Theorem 2.2. *Assume the system (2) is strictly hyperbolic and let $\mathbf{u}_l, \mathbf{u}_r \in \mathbb{R}^m$. If $\mathbf{u}_l, \mathbf{u}_r$ are sufficiently closed to each other, then there exists a neighborhood $\mathcal{N}(\mathbf{u}_l, \mathbf{u}_r) \subset \mathbb{R}^{2m}$ such that the constant states $\mathbf{u}_j \in C^1(\mathcal{N}(\mathbf{u}_l, \mathbf{u}_r); \mathbb{R}^m)$ for $j = 1, \ldots, m - 1$.*

In view of the proof of Theorem 2.2, we recall the following result.

Theorem 2.3. ***(Dependence of eigenvalues and eigenvectors on parameters)*** *Let $\mathbf{B} \in C^1(\mathbb{R}^p; \mathbb{R}^{m,m})$, $m, p \in \mathbb{N}$. For a given $\mathbf{z}_0 \in \mathbb{R}^p$, let $\mathbf{B}(\mathbf{z}_0)$ be diagonizable with distinct eigenvalues $\lambda_1(\mathbf{z}_0) < \ldots < \lambda_m(\mathbf{z}_0)$. Then there exists a neighborhood $\mathcal{N}(\mathbf{z}_0)$ of $\mathbf{z}_0$ such that the eigenvalues $\lambda_k \in C^1(\mathcal{N}(\mathbf{z}_0); \mathbb{R})$ and the left and right eigenvectors $\mathbf{l}_k, \mathbf{r}_k \in C^1(\mathcal{N}(\mathbf{z}_0); \mathbb{R}^p)$ and satisfy the normalization $|\mathbf{r}_k(\mathbf{z})|, |\mathbf{l}_k(\mathbf{z})| = 1$ for each $k \in \{1, \ldots, m\}$ and for all $\mathbf{z} \in \mathcal{N}(\mathbf{z}_0)$.*

The proof is a straightforward application of the Implicit Function Theorem to the equation $\mathbf{\Phi}(\mathbf{r}, \lambda, \mathbf{z}) = (\mathbf{B}(\mathbf{z})\mathbf{r} - \lambda \mathbf{r}, |\mathbf{r}|^2 - 1) = 0$ (see [4, Th. 2, section 11.1.2]).

The first step consists in verifying the continuous dependence of $\mathbf{\Upsilon}(\epsilon, \mathbf{u}_l)$ with respect to $\mathbf{u}_l$. By definition, it is straightforward that $R_k(\mathbf{u}_l)$ depends smoothly on $\mathbf{u}_l$. On the other hand, the continuous dependence of the shock set is less evident and is addressed by the next Lemma.

Lemma 2.4. *Let us consider the strictly hyperbolic system (1). Then, for each $k = 1, \ldots, m$ there exists $\mathbf{\Psi}_k : I \times \mathbb{R}^m \to \mathbb{R}^m$ such that for each $\mathbf{u}_0 \in \mathbb{R}^m$ $\mathbf{\Psi}_k(\cdot, \mathbf{u}_0) : I \to \mathbb{R}^m$ is a parametrization of $S_k(\mathbf{u}_0)$ and $\mathbf{\Psi}_k(0, \mathbf{u}_0) = \mathbf{u}_0$. Furthermore, there exists a neighborhood $\mathcal{N}(0, \mathbf{u}_0)$ of $(0, \mathbf{u}_0)$ such that $\mathbf{\Psi}_k \in C^1(\mathcal{N}(0, \mathbf{u}_0); \mathbb{R}^m)$, $k = 1, \ldots, m$.*

Proof. Let us define[2]:

$$\mathbf{B}(\mathbf{z}, \mathbf{u}) := \int_0^1 D\mathbf{F}(\mathbf{u} + t(\mathbf{z} - \mathbf{u}))\, dt \quad \mathbf{z}, \mathbf{u} \in \mathbb{R}^m.$$

We first observe that $\mathbf{B}(\mathbf{z}, \mathbf{u})(\mathbf{z} - \mathbf{u}) = \mathbf{F}(\mathbf{z}) - \mathbf{F}(\mathbf{u})$. As a consequence, $\mathbf{z} \in S(\mathbf{u})$ if and only if:

$$(\mathbf{B}(\mathbf{z}, \mathbf{u}) - \sigma I)\,(\mathbf{z} - \mathbf{u}) = \mathbf{0}. \qquad (12)$$

Since the system is strictly hyperbolic, $\mathbf{B}(\mathbf{u}_0, \mathbf{u}_0) = D\mathbf{F}(\mathbf{u}_0)$ has distinct eigenvalues, thus (due to Theorem 2.3) there exist a neighborhood $\mathcal{N}(\mathbf{u}_0, \mathbf{u}_0)$ and smooth

[2] This proof follows the same idea of the proof of [4, Th.2, section 11.2.3].

functions $\lambda_k : \mathcal{N}(\mathbf{u}_0, \mathbf{u}_0) \to \mathbb{R}$, $\mathbf{r}_k : \mathcal{N}(\mathbf{u}_0, \mathbf{u}_0) \to \mathbb{R}^m$ and $\mathbf{l}_k : \mathcal{N}(\mathbf{u}_0, \mathbf{u}_0) \to \mathbb{R}^m$ $(k = 1, \dots, m)$ such that:

$$\begin{cases} \mathbf{B}(\mathbf{z}, \mathbf{u})\mathbf{r}_k(\mathbf{z}, \mathbf{u}) = \lambda_k(\mathbf{z}, \mathbf{u})\mathbf{r}_k(\mathbf{z}, \mathbf{u}), \\ \mathbf{B}^T(\mathbf{z}, \mathbf{u})\mathbf{l}_k(\mathbf{z}, \mathbf{u}) = \lambda_k(\mathbf{z}, \mathbf{u})\mathbf{l}_k(\mathbf{z}, \mathbf{u}), \end{cases}$$

for all $(\mathbf{z}, \mathbf{u}) \in \mathcal{N}(\mathbf{u}_0, \mathbf{u}_0)$. Furthermore, $\{\mathbf{r}_k(\mathbf{z}, \mathbf{u})\}_k$, $\{\mathbf{l}_k(\mathbf{z}, \mathbf{v})\}_k$ are bases of $\mathbb{R}^m$ and $\mathbf{r}_l(\mathbf{z}, \mathbf{v}) \cdot \mathbf{l}_k(\mathbf{z}, \mathbf{v}) = 0$ if $l \neq k$.

If we fix $k \in \{1, \dots, m\}$, equation (12) holds if and only if $(\mathbf{z} - \mathbf{u}) \parallel \mathbf{r}_k(\mathbf{z}, \mathbf{u})$. As a consequence of the previous discussion, this condition can be reformulated as:

$$\begin{cases} \mathbf{l}_l(\mathbf{z}, \mathbf{u}) \cdot (\mathbf{z} - \mathbf{u}) = 0 & \text{if } l \neq k \\ \mathbf{l}_k(\mathbf{z}, \mathbf{u}) \cdot (\mathbf{z} - \mathbf{u}) \neq 0 & \text{if } l = k. \end{cases} \tag{13}$$

In view of the application of the Implicit Function Theorem, we define

$$\boldsymbol{\Phi}_k : \mathbb{R}^{2m} \to \mathbb{R}^{m-1}; \quad \boldsymbol{\Phi}_k(\mathbf{z}, \mathbf{v}) = \begin{bmatrix} \vdots \\ \mathbf{l}_{k-1}(\mathbf{z}, \mathbf{u}) \cdot (\mathbf{z} - \mathbf{u}) \\ \mathbf{l}_{k+1}(\mathbf{z}, \mathbf{u}) \cdot (\mathbf{z} - \mathbf{u}) \\ \vdots \end{bmatrix}$$

$\boldsymbol{\Phi}_k(\mathbf{z}, \mathbf{v}) = \mathbf{0}$ is a system of $m-1$ equations in $2m$ variables. Let $\mathbf{z}' = [z_1, \dots, z_{m-1}]$ and $\tilde{\mathbf{l}}_j = [(\mathbf{l}_j)_1, \dots, (\mathbf{l}_j)_{m-1}]^T$, we observe that:

$$\boldsymbol{\Phi}_k(\mathbf{u}_0, \mathbf{u}_0) = \mathbf{0}, \quad D_{\mathbf{z}'}\boldsymbol{\Phi}_k(\mathbf{z}, \mathbf{u}) = \begin{bmatrix} \vdots \\ \tilde{\mathbf{l}}_{k-1}^T(\mathbf{z}, \mathbf{u}) \\ \tilde{\mathbf{l}}_{k+1}^T(\mathbf{z}, \mathbf{u}) \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ \left(D_{\mathbf{z}'}\mathbf{l}_{k-1}^T(\mathbf{z}, \mathbf{u})(\mathbf{z} - \mathbf{u})\right)^T \\ \left(D_{\mathbf{z}'}\mathbf{l}_{k+1}^T(\mathbf{z}, \mathbf{u})(\mathbf{z} - \mathbf{u})\right)^T \\ \vdots \end{bmatrix}.$$

As $\{\mathbf{l}_k(\mathbf{z}, \mathbf{u})\}_k$ is a basis of $\mathbb{R}^m$, $\operatorname{rank}(D_{\mathbf{z}'}\boldsymbol{\Phi}_k(\mathbf{u}_0, \mathbf{u}_0)) = m - 1$. Therefore, thanks to the Implicit Function Theorem, there exists a neighborhood $\mathcal{N}((\mathbf{u}_0)_m,\, \mathbf{u}_0)$ and a C^1 function $\tilde{\boldsymbol{\Psi}}_k : \mathcal{N}((\mathbf{u}_0)_m,\, \mathbf{u}_0) \to \mathbb{R}^m$ such that:

$$\begin{cases} \boldsymbol{\Phi}_k(\tilde{\boldsymbol{\Psi}}_k(\mathbf{v}), \mathbf{v}) = 0 \Leftrightarrow \tilde{\boldsymbol{\Psi}}_k(\mathbf{v}) \in S_k(\mathbf{u}) \quad \forall\, \mathbf{v} = \begin{bmatrix} s \\ \mathbf{u} \end{bmatrix} \in \mathcal{N}((\mathbf{u}_0)_m,\, \mathbf{u}_0), \\ \tilde{\boldsymbol{\Psi}}_k((\mathbf{u}_0)_m,\, \mathbf{u}_0) = \mathbf{u}_0. \end{cases}$$

The result follows by taking $\boldsymbol{\Psi}_k(t, \mathbf{u}) = \tilde{\boldsymbol{\Psi}}_k((\mathbf{u}_0)_m + t,\, \mathbf{u})$. □

Thanks to the previous Lemma, the curves $\mathbf{T}_k(\mathbf{u}_l) = R_k^+(\mathbf{u}_l) \cup \{\mathbf{u}_l\} \cup S_k^-(\mathbf{u}_l)$ are of class C^1 with respect to $\mathbf{u}_l$.

We are now ready to prove our main result. We limit ourselves to the case $m = 2$. The extension to the general case is not particularly complex.

Proof. (Theorem 2.2) Given $\mathbf{u}_l$ and $\mathbf{u}_r$, we consider the parametrizations $\mathbf{\Upsilon}_k : I \times \mathbb{R}^2 \to \mathbb{R}^2$ for the curves $\mathbf{T}_k(\cdot)$, $k = 1, 2$. Let us introduce the function $\mathbf{\Phi} : \mathbb{R}^6 \to \mathbb{R}^2$ such that: $\mathbf{\Phi}(\boldsymbol{\epsilon}, \mathbf{u}_l, \mathbf{u}_r) = \mathbf{\Upsilon}_2(\epsilon_2, \mathbf{\Upsilon}_1(\epsilon_1, \mathbf{u}_l)) - \mathbf{u}_r$. Due to Lemma 2.4, we have that $\mathbf{\Phi}(\boldsymbol{\epsilon}, \mathbf{u}_l, \mathbf{u}_r)$ is of class C^1. Furthermore,

$$D_{\boldsymbol{\epsilon}}\mathbf{\Phi}(\boldsymbol{\epsilon}, \mathbf{u}_l, \mathbf{u}_r) = [D_{\mathbf{u}}\mathbf{\Upsilon}_2(\epsilon_2, \mathbf{\Upsilon}_1(\epsilon_1, \mathbf{u}_l))\partial_{\epsilon_1}\mathbf{\Upsilon}_1(\epsilon_1, \mathbf{u}_l),\ \partial_{\epsilon_2}\mathbf{\Upsilon}_2(\epsilon_2, \mathbf{\Upsilon}_1(\epsilon_1, \mathbf{u}_l))]$$

$$= [\mathbf{r}_1(\mathbf{u}_l),\ \mathbf{r}_2(\mathbf{u}_l)] + \mathcal{O}(\boldsymbol{\epsilon}),$$

where we used the fact that $D_{\mathbf{u}}\mathbf{\Upsilon}_k(\boldsymbol{\epsilon}, \mathbf{u})|_{\boldsymbol{\epsilon}=\mathbf{0}} = \mathbb{I}$ for $k = 1, 2$.

Since $\{\mathbf{r}_k(\mathbf{u}_l)\}_k$ is a basis of $\mathbb{R}^2$, we have that, for sufficiently small ϵ_1, ϵ_2,

$$\operatorname{rank}\left(D_{\boldsymbol{\epsilon}}\mathbf{\Phi}(\epsilon, \mathbf{u}_l, \mathbf{u}_r)\right) = 2.$$

The thesis follows by applying once more the Implicit Function Theorem. □

Since $\mathbf{u}_j$ is a regular function of $\mathbf{u}_l$, $\mathbf{u}_r$ for $j = 1, \ldots, m-1$, we expect that $\mathbf{u}_j = \mathbf{u}_j(\mathbf{u}_l, \mathbf{u}_r)$ can be well-approximated by a polynomial expansion. Therefore, Theorem 2.2 justifies a possible strategy for the solution of problem (2) based on an offline-online decomposition. In the offline stage the polynomial coefficients are estimated by solving a number of Riemann problems, then in the online stage the polynomial approximation is used to speed-up the solution of the Riemann problem. We refer to Section 3 for the details.

3 An offline-online strategy for the resolution of the Riemann problem

Motivated by the previous mathematical analysis, in this section we focus on the development of an offline-online strategy for the resolution of the Riemann problem (2). Let us define $\boldsymbol{\mu} := [\mathbf{u}_l, \mathbf{u}_r]^T$ and the compact set $\mathcal{D} \subset \mathbb{R}^{2m}$. In order to stress the dependence on the parameters, in the following we refer to Riemann($\boldsymbol{\mu}$) as to the Riemann problem (2) associated with $\boldsymbol{\mu} \in \mathcal{D}$ and we denote by $\mathbf{u}_1(\boldsymbol{\mu}), \ldots, \mathbf{u}_{m-1}(\boldsymbol{\mu})$, the constant states of the solution to Riemann($\boldsymbol{\mu}$).

We assume that $\mathbf{u}_1(\boldsymbol{\mu}), \ldots, \mathbf{u}_{m-1}(\boldsymbol{\mu})$ satisfy the algebraic nonlinear system

$$\boldsymbol{\mathcal{L}}(\mathbf{u}_1(\boldsymbol{\mu}), \ldots, \mathbf{u}_{m-1}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} \tag{14}$$

where $\boldsymbol{\mathcal{L}} : \mathbb{R}^{(m-1)m} \times \mathbb{R}^{2m} \to \mathbb{R}^{(m-1)m}$ is a given function associated with the problem (1). As we will see in Sections 4 and 5 - where (14) corresponds to (19) and (24), respectively-, this function can be defined explicitly for many relevant problems.

We can now detail the procedure briefly sketched in the introduction.

In the offline stage, given the parameter space $\mathcal{D} \subset \mathbb{R}^{2m}$ we first identify a training set $\Xi_{train} := \{\boldsymbol{\mu}^k\}_{k=1}^{n_{train}} \subset \mathcal{D}$ of cardinality n_{train}, which represents a suitable surrogate of $\mathcal{D}$. Then, we compute $\mathbf{u}_1(\boldsymbol{\mu}^k), \ldots, \mathbf{u}_{m-1}(\boldsymbol{\mu}^k)$ yielding the constant-state solutions to the Riemann problem (2) for all $\boldsymbol{\mu} = \boldsymbol{\mu}^k$, $k = 1, \ldots, n_{train}$. Finally, we use these values to compute a piecewise polynomial approximation $\Pi_{N,n_{train}} \mathbf{u}_j : \mathcal{D} \to \mathbb{R}^m$ for $\mathbf{u}_j$, $j = 1, \ldots, m-1$, of order N.

In the online stage, for any new value of the parameter $\boldsymbol{\mu} \in \mathcal{D}$ we solve the nonlinear system (14) by using $\Pi_{N,n_{train}} \mathbf{u}_1(\boldsymbol{\mu}), \ldots, \Pi_{N,n_{train}} \mathbf{u}_{m-1}(\boldsymbol{\mu})$ as initial guesses of the iterative solver.

Algorithm 1 summarizes the offline-online procedure.

Algorithm 1 Offline-online Riemann solver: general framework

```
Offline stage
    Build the training set $\Xi_{train} \subset \mathcal{D}$.
    for $k = 1, \ldots, n_{train}$
        Compute $\mathbf{u}_j^k := \mathbf{u}_j(\boldsymbol{\mu}^k)$ by solving the nonlinear system (14) for $\boldsymbol{\mu} = \boldsymbol{\mu}^k$.
    end for
    Use $\{(\boldsymbol{\mu}^k, \mathbf{u}_j^k),\ k = 1, \ldots, n_{train}\}$ to compute a $N$-order polynomial approximation $\Pi_{N,n_{train}} \mathbf{u}_j : \mathcal{D} \to \mathbb{R}^m$ for $\mathbf{u}_j$, $j = 1, \ldots, m-1$.
Online stage
    Given $\boldsymbol{\mu} = (\mathbf{u}_l, \mathbf{u}_r)$,
    Compute $\mathbf{u}_j(\boldsymbol{\mu})$, $j = 1, \ldots, m-1$, using $\Pi_{N,n_{train}} \mathbf{u}_j(\boldsymbol{\mu})$ as initial guess of the iterative solver for the nonlinear system (14).
```

In this work for the sake of simplicity the training set is generated through a uniform sampling, while we use the Newton method (see, e.g., [7]) as iterative solver for (14).

Some comments are in order.

In many situations, the system (14) can be deeply simplified, thus in practice one requires only a few nonlinear equations. Therefore, we can limit the offline-online strategy here proposed to the independent variables of the nonlinear equations to be solved.

It is well-known that, without a sufficiently accurate initial estimate, it is necessary to combine a Newton-like high order and locally convergent method with a more robust root finding method (e.g., bisection or secant methods). On the other hand, if $\Pi_{N,n_{train}} \mathbf{u}_j(\boldsymbol{\mu})$ is reasonably accurate, we expect that Newton method does not require to be coupled with other globally convergent techniques. As a consequence, the speed-up is potentially relevant.

Before concluding, we recall that the idea of using an offline-online computational strategy to provide an accurate initial guess for the nonlinear Newton-like solver has already been exploited in [3]. However, in our work the guess we provide through the offline stage corresponds with a surrogate of the solution $\mathbf{u}(\boldsymbol{\mu})$ - the constant states $\mathbf{u}_1(\boldsymbol{\mu})$, …, $\mathbf{u}_{m-1}(\boldsymbol{\mu})$ - instead than with the PDE solution itself.

In the next two sections, we apply our technique to two different notable hyperbolic systems arising in Fluid Dynamics. The numerical experiments presented below have been realized in `Matlab`® ([6]).

4 Application to the *p-system*

In this section we apply our method to the *p-system*, which is a special case of (1) characterized by

$$\mathbf{u} := \begin{bmatrix} w \\ v \end{bmatrix}, \quad \mathbf{F}(w,v) = \begin{bmatrix} -v \\ p(w) \end{bmatrix} \tag{15}$$

and representing a model for isentropic gas dynamics in Lagrangian coordinates. Let us introduce $a := -p'$; in the following we assume that $a(w) := -p'(w)$, $a(w) > 0$, $a'(w) < 0$ for all w and $\lim_{w\to 0^+} p(w) = +\infty$. We refer to [1, Section 5.5] for a thorough analysis of the Riemann problem for this system.

This section is organized as follows. First, we introduce the nonlinear equation to be solved that corresponds to (14). Then, we evaluate our technique through a number of numerical results.

4.1 Explicit formula for the p-system

Fixed $\mathbf{u}_l \in \mathbb{R}^2_+$, by simple calculations we can deduce that in the case at hand the sets introduced in Section 2 take the following form:

$$\begin{aligned}
S_1^-(\mathbf{u}_l) &:= \{(w,v) \in \mathbb{R}^2_+ : w < w_l,\ v = \Phi_1(w,\mathbf{u}_l) = v_l - \sqrt{(p(w)-p(w_l))(w_l - w)}\} \\
S_2^-(\mathbf{u}_l) &:= \{(w,v) \in \mathbb{R}^2_+ : w > w_l,\ v = \Phi_2(w,\mathbf{u}_l) = v_l - \sqrt{(p(w)-p(w_l))(w_l - w)}\} \\
R_1^+(\mathbf{u}_l) &:= \{(w,v) \in \mathbb{R}^2_+ : w > w_l,\ v = \Psi_1(w,\mathbf{u}_l) = v_l + \textstyle\int_{w_l}^{w} \sqrt{a(s)}\,ds\} \\
R_2^+(\mathbf{u}_l) &:= \{(w,v) \in \mathbb{R}^2_+ : w < w_l,\ v = \Psi_2(w,\mathbf{u}_l) = v_l - \textstyle\int_{w_l}^{w} \sqrt{a(s)}\,ds\}
\end{aligned} \tag{16}$$

Furthermore, we introduce the curves $\mathbf{T}_k(\mathbf{u}_l) = \mathbf{R}_k^+(\mathbf{u}_l) \cup \{\mathbf{u}_l\} \cup \mathbf{S}_k^-(\mathbf{u}_l)$, $k = 1,2$, and their parametrizations $w \to \boldsymbol{\Upsilon}_k(w,\mathbf{u}_l) = [w, \Upsilon_k(w,\mathbf{u}_l)]$ where

$$\Upsilon_1(w,\mathbf{u}_l) = \begin{cases} \Phi_1(w,\mathbf{u}_l) & w < w_l \\ \Psi_1(w,\mathbf{u}_l) & w \geq w_l \end{cases} \qquad \Upsilon_2(w,\mathbf{u}_l) = \begin{cases} \Psi_2(w,\mathbf{u}_l) & w < w_l \\ \Phi_2(w,\mathbf{u}_l) & w \geq w_l \end{cases} \tag{17}$$

for $k = 1,2$. Finally, we introduce the partition $\{\mathcal{R}_i(\mathbf{u}_l)\}_{i=1}^4$ induced by the curves $\mathbf{T}_k(\mathbf{u}_l)$:

$$\begin{aligned}
\mathcal{R}_1(\mathbf{u}_l) &= \{(w,v) \in \mathbb{R}^2_+ : w > w_l,\ \Upsilon_2(w,\mathbf{u}_l) < v < \Upsilon_1(w,\mathbf{u}_l)\}; \\
\mathcal{R}_2(\mathbf{u}_l) &= \{(w,v) \in \mathbb{R}^2_+ :\ v < \min\{\Upsilon_1(w,\mathbf{u}_l), \Upsilon_2(w,\mathbf{u}_l)\}\}; \\
\mathcal{R}_3(\mathbf{u}_l) &= \{(w,v) \in \mathbb{R}^2_+ : w < w_l,\ \Upsilon_1(w,\mathbf{u}_l) < v < \Upsilon_2(w,\mathbf{u}_l)\}; \\
\mathcal{R}_2(\mathbf{u}_l) &= \{(w,v) \in \mathbb{R}^2_+ :\ v > \max\{\Upsilon_1(w,\mathbf{u}_l), \Upsilon_2(w,\mathbf{u}_l)\}\}.
\end{aligned} \tag{18}$$

Figure 1 shows the partition $\{\mathcal{R}_i(\mathbf{u}_l)\}_{i=1}^4$

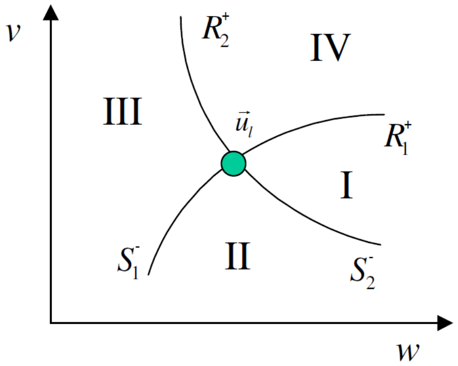

Figure 1: State space partition induced by the curves $\mathbf{T}_k(\mathbf{u}_l)$, $k = 1, 2$.

Thanks to Lemmas 2.1, 2.2 and 2.3, it is possible to derive a closed formula for the solution to the Riemann problem, which is based on the knowledge of the intermediate state $\mathbf{u}_1$. Next Lemma provides the nonlinear equation (see (14)) to be solved for obtaining $\mathbf{u}_1$. We refer to [1] for the proof.

Lemma 4.1. *Should it exists, the solution to the Riemann problem (2) for the p-system (15) consists of three constant (possibly coincident) states, $\mathbf{u}_l, \mathbf{u}_1, \mathbf{u}_r$. Furthermore, if $\mathbf{u}_r \in \mathcal{R}_i(\mathbf{u}_l)$ for some $i \in \{1, \dots, 4\}$, $\mathbf{u}_1 = [w_1, v_1]^T \in \mathbb{R}_+^2$ is such that w_1 is the only solution to the nonlinear equation*

$$f(w, \boldsymbol{\mu}) = v_r - v_l + h(w, w_l) + h(w, w_r) = 0 \quad w \in I_i(\boldsymbol{\mu}), \tag{19a}$$

and $v_1 = v_l - h(w_1, w_l)$, where:

$$h(w, w_K) := \begin{cases} \sqrt{(p(w) - p(w_K))(w_K - w)} & \text{if } w \le w_K, \\ -\displaystyle\int_{w_K}^{w} \sqrt{a(s)}\, ds & \text{if } w > w_K, \end{cases} \qquad K = l, r \tag{19b}$$

and

$$I_i(\boldsymbol{\mu}) := \begin{cases} (w_l, w_r) & \text{if } i = 1, \\ (0, \min\{w_l, w_r\}) & \text{if } i = 2, \\ (w_r, w_l) & \text{if } i = 3, \\ (\max\{w_r, w_l\}, \infty) & \text{if } i = 4. \end{cases} \tag{19c}$$

4.2 Application of the offline-online strategy to the p-system

Thanks to Lemma 4.1, for each Riemann problem Riemann$(\boldsymbol{\mu})$, we need to solve only one scalar nonlinear equation depending on the position of $\mathbf{u}_r$ with respect to $\mathbf{u}_l$. More precisely, we first identify the region $\mathcal{R}_i$ associated with $\boldsymbol{\mu}$ and then we solve the nonlinear equation (19a).

As a result, we just need to provide an approximation of $w_1(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathcal{D}$. In all our numerical simulations we consider $p(w) := 0.75w^{-2.5}$, $\mathcal{D} := [1,3]^4$ and we propose a second order polynomial approximation, one for each region, that we compute using the least squares method.

During the offline stage we consider a two-level iterative method based on bisection and Newton methods ([7]). We set the tolerance $tol_{coarse} = 10^{-2}$ for bisection method and $tol_{fine} = 10^{-8}$ for Newton method. This polyalgorithm will also be used online as benchmark to evaluate our method.

In order to present the results, we introduce the following two sets:

$$\begin{cases} \mathcal{D}^{OFF}_{n_{train},h} := \left\{(\mathbf{u}_l, \mathbf{u}_r) \in [1,3]^4 : |\mathbf{u}_l - \mathbf{u}_r|_\infty \le h,\ 1,\dots,n_{train}\right\}, \\ \mathcal{D}^{ON}_{n'_{train},h} := \left\{(\mathbf{u}_l, \mathbf{u}_r) \in [1,3]^4 : |\mathbf{u}_l - \mathbf{u}_r|_\infty \le h,\ 1,\dots,n'_{train}\right\}, \end{cases} \tag{20}$$

where $h > 0$. $\mathcal{D}^{OFF}_{n_{train},h}$ is used to build the polynomial approximations during the offline stage, while $\mathcal{D}^{ON}_{n'_{train},h}$ is used to evaluate the performance of our method during the online stage. Both sets are built through a uniform random sampling.

In view of the integration of our Riemann solver with the Godunov method, we are also interested in evaluating whether the speed-up of our offline online method depends on h. We expect indeed that, among the Riemann problems to be solved at each time step, only a few of them show large values of the jump $|\mathbf{u}_l - \mathbf{u}_r|$.

We first assess the quality of our polynomial approximation. Table 1 reports the mean error

$$\epsilon^{mean} := \frac{1}{n'_{train}} \sum_{j=1}^{n'_{train}} |w_1(\boldsymbol{\mu}_j) - \Pi_{2,n_{train}} w(\boldsymbol{\mu}_j)| \tag{21}$$

for the different regions $\mathcal{R}_i$, $i = 1,\dots,4$.

Table 1: Mean error ϵ^{mean} for the four polynomial approximations ($n_{train} = 2000$, $n'_{train} = 5000$)

	Region I	Region II	Region III	Region IV
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \le 0.5$	0.0029	0.0041	0.0041	0.0118
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \le 0.9$	0.0061	0.0087	0.0115	0.0526

Note that it depends on h and on which region it is actually evaluated.

We now proceed to quantify how many times the procedure fails; that is, given $\mathcal{D}^{ON}_{n'_{train},h} \subset \mathcal{D}$, for how many $\boldsymbol{\mu} \in \mathcal{D}^{ON}_{n'_{train},h}$ (*failed points*) the Newton algorithm initialized with the precomputed polynomial approximation does not converge. In addition, we compare our results with the ones obtained by using another initial guess, $w_1^{a\,priori}(\boldsymbol{\mu})$, defined as follows:

$$w_1^{a\,priori}(\boldsymbol{\mu}) := \begin{cases} \frac{1}{2}(w_l + w_r) & \text{if } \mathbf{u}_r \in \mathcal{R}_1(\mathbf{u}_l) \cup \mathcal{R}_3(\mathbf{u}_l) \\ \frac{1}{2}\min\{w_l, w_r\} & \text{if } \mathbf{u}_r \in \mathcal{R}_2(\mathbf{u}_l) \\ \max\{w_l, w_r\} & \text{if } \mathbf{u}_r \in \mathcal{R}_4(\mathbf{u}_l) \end{cases} \tag{22}$$

Table 2: Number of failed points ($n_{train} = 2000$, $n'_{train} = 5000$) for different maximum distances h in (20)

	number of failed points	
	$\Pi_{2,n_{train}} w$	$w_1^{a\,priori}$
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.5$	29 (0.5%)	911 (18.22%)
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.6$	19 (0.38%)	916 (18.32%)
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.7$	36 (0.72%)	918 (18.36%)
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.8$	32 (0.64%)	903 (18.06%)
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.9$	21 (0.42%)	907 (18.14%)
$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 1$	32 (0.64%)	869 (17.38%)

Table 2 shows that the number of failed points associated with the offline guess does not depend on h and is significantly smaller than the one associated with $w_1^{a\,priori}(\boldsymbol{\mu})$.

Finally, we make a comparison between the performance of the standard Riemann solver and that of the offline-online Riemann solver for two different values of h. We specify that in the current implementation the standard Riemann solver is called into play whenever the Riemann solver here proposed fails.

Tables 3 gathers the results.

For this particular case study, we have that the speed-up seems not to depend on the parameter h.

Before concluding, we observe that for $h = 0.9$ and $n_{train} = 2000$, the computational time[3] associated with the offline stage is 9.75 seconds, 9.63 seconds for the resolution of the n_{train} nonlinear equations, 0.12 seconds for the allocation of the data structures and the calculation of the polynomial terms. Based on these results, we can draw two observations.

[3] The values here gathered are obtained by averaging the results of four independent simulations.

Table 3: Comparison between the offline-online Riemann solver and the standard one ($n_{train} = 2000$, $n'_{train} = 5000$)

$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.5$	bisection it.	Newton it.	ave. time (speed-up)
Riemann Solver	6.9876	2.9808	0.0064 (1)
Offline-Online Riemann Solver	-	3.0294	0.0029 (2.21)

$\lvert\mathbf{u}_l - \mathbf{u}_r\rvert_\infty \leq 0.9$	bisection it.	Newton it.	ave. time (speed-up)
Riemann Solver	7.0940	2.9836	0.0069 (1)
Offline-Online Riemann Solver	-	3.2720	0.0032 (2.16)

- For our particular simulation the offline-online strategy is convenient if the number of Riemann problems to be solved is larger than 2714 (*break even point*).
- In view of the application of this technique in combination with finite volume methods for the approximation of general nonlinear hyperbolic problems, we think that the main difficulty consists in the *a priori* definition of the parameter set $\mathcal{D}$. Indeed, given $\mathbf{u}_0 : \mathbb{R} \to \mathbb{R}^m$, it is hard to define a set $\mathcal{D} = \mathcal{D}' \times \mathcal{D}'$ such that $\{\mathbf{u}(x,t) : (x,t) \in \mathbb{R} \times (0,\infty)\} \subset \mathcal{D}'$. Since the construction of the polynomial approximation is particularly fast, we expect that it is possible to efficiently integrate the offline and online stages in order to correct the polynomial approximation at run-time. We refer to a future work for further developments.

5 Application to the *Euler* equations for ideal gases

This section addresses the case of the Euler equations for ideal gases, a special case of (1) with

$$\mathbf{U} := \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) := \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{bmatrix}, \quad E = \frac{1}{2}\rho u^2 + \frac{1}{\gamma - 1}p. \tag{23}$$

ρ is the gas density, u is the particle velocity, p is the pressure and E is the total energy per unit volume. We refer to $\mathbf{U}$ as to the vector of the conserved variables. The vector of the primitive variables is $\mathbf{W} := [\rho,\, u,\, p]^T$.

We refer to [9, Chap.4] for a detailed discussion about the solution to the Riemann problem (2)-(23). In the following, we first present the nonlinear equation to be solved to compute the solution to the Riemann's problem, then we present some numerical results.

5.1 Explicit formula for Euler equations

According to the theory developed in Section 2, the solution (for the primitive variables) consists of four constant states $\mathbf{W}_l, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_r$. The first two states, $\mathbf{W}_l, \mathbf{W}_1$, are separated by a 1-shock or by a 1-rarefaction wave, while $\mathbf{W}_1, \mathbf{W}_2$ are connected through a 2-contact discontinuity. Finally, $\mathbf{W}_2, \mathbf{W}_r$ are pieced together through a 3-shock or a 3-rarefaction wave. It is possible to show that $\mathbf{W}_1 = [\rho_{\star,L},\, u_\star,\, p_\star]^T$ and $\mathbf{W}_2 = [\rho_{\star,R},\, u_\star,\, p_\star]^T$ where $\rho_{\star,L}, \rho_{\star,R}, u_\star, p_\star \in \mathbb{R}$. In addition, there exists a closed formula for $\rho_{\star,L}, \rho_{\star,R}, u_\star$ depending only on $p^\star$ (see [9, Chap. 4] for all the details).

In conclusion, the solution to the Riemann problem (2)-(23) depends only on the resolution of a scalar nonlinear equation for the pressure $p_\star$. The following Lemma ([9, Prop.4.2.1, section 4.2]) completes the discussion.

Lemma 5.1. *The pressure component $p_\star$ of the solution to the Riemann problem (2) associated to the Euler equations (23) is given by the root of the algebraic equation*

$$f(p,\boldsymbol{\mu}) := f_l(p,\mathbf{W}_l) + f_l(p,\mathbf{W}_r) + u_r - u_l, \tag{24a}$$

where:

$$f_K(p,\mathbf{W}_K) = \begin{cases} (p-p_K)\left[\dfrac{A_K}{p+B_K}\right]^{\frac{1}{2}} & p > p_K \\ \dfrac{2a_R}{\gamma-1}\left[\left(\dfrac{p}{p_K}\right)^{\frac{\gamma-1}{2\gamma}} - 1\right] & p \le p_K \end{cases} \quad \textit{where } K = l, r \tag{24b}$$

and the data-dependent constants are

$$A_K := \frac{2}{(\gamma+1)\rho_K},\; B_K := \frac{\gamma-1}{\gamma+1}p_K,\; a_K := \sqrt{\frac{\gamma p_K}{\rho_K}}. \tag{24c}$$

Furthermore, if we define $p_{min} := \min\{p_l, p_r\}$ and $p_{max} := \max\{p_l, p_r\}$, we can distinguish the following three cases:

$$\begin{cases} \textit{if } f(p_{min},\boldsymbol{\mu}) > 0, \; \textit{then}\, p^\star \; \textit{lies in } \; (0, p_{min}); \\ \textit{if } f(p_{min},\boldsymbol{\mu}) \le 0, \; f(p_{max},\boldsymbol{\mu}) \ge 0 \; \textit{then}\, p^\star \; \textit{lies in } \; (p_{min}, p_{max}); \\ \textit{if } f(p_{max},\boldsymbol{\mu}) < 0, \; \textit{then}\, p^\star \; \textit{lies in } \; (p_{max}, \infty). \end{cases} \tag{24d}$$

5.2 Application of the offline-online strategy to the Euler equation

Following the same steps of Section 4.2, given the Riemann problem Riemann $(\boldsymbol{\mu})$, we identify the interval (24d) the solution belongs to and then we solve appropriately the nonlinear equation (24) for $p^\star$. As a result, during the offline stage we compute a piecewise second order polynomial approximation of $p^\star : \mathcal{D} \to \mathbb{R}$ based on the domain partition induced by (24d). In our numerical simulations we consider $\gamma = 1.4$ and $\mathcal{D} = ([1,5]\times[1,5]\times[1,5])^2$. Unlike the previous example, during the offline stage we consider the Newton method initialized through (see [9, (4.47) section 4.3]):

$$p_0(\boldsymbol{\mu}) := \max\{\epsilon_{tol}, p_{PV}(\boldsymbol{\mu})\}, \quad p_{PV}(\boldsymbol{\mu}) = \frac{1}{2}(p_l + p_r) - \frac{1}{8}(u_r - u_l)(\rho_r + \rho_l)(a_l + a_r), \tag{25}$$

where a_l, a_r are defined in (24c). In Tables 4 and 5 we compare the accuracy of the piecewise polynomial $\Pi_{2,n_{train}} p(\boldsymbol{\mu})$ with the accuracy of the a priori estimate $p_0(\boldsymbol{\mu})$.

Table 4: Mean error for the three polynomial approximations ($\Pi(\boldsymbol{\mu}) = \Pi_{2,n_{train}} p(\boldsymbol{\mu})$, $n_{train} = 2000$, $n'_{train} = 5000$), comparison with the mean error for the a priori guess (25).

	Region I		Region II		Region III	
	$\Pi(\boldsymbol{\mu})$	$p_0(\boldsymbol{\mu})$	$\Pi(\boldsymbol{\mu})$	$p_0(\boldsymbol{\mu})$	$\Pi(\boldsymbol{\mu})$	$p_0(\boldsymbol{\mu})$
$\lvert\mathbf{W}_l - \mathbf{W}_r\rvert_\infty \le 0.7$	0.0091	0.032	0.0040	0.0056	0.0103	0.0331
$\lvert\mathbf{W}_l - \mathbf{W}_r\rvert_\infty \le 0.9$	0.0128	0.0536	0.0074	0.0099	0.0137	0.0331

Unlike the previous example, our "empirical" guess is comparable with the a priori guess (25). We observe that the choice of (25) is far from being self-evident: as explained in [9] it results from the inspection of a linearised solution. On the other hand, our "empirical" guess is computed automatically and it is influenced by the equation only through the function $p^\star(\boldsymbol{\mu})$ and the parameter space $\mathcal{D}$. This is why we expect that for more involved problems, the derivation of so accurate a priori guesses might be extremely complex. On the other hand, the method here proposed is not influenced by the problem.

Table 5: Mean number of Newton iteration and average time using the precomputed guess ($\Pi(\boldsymbol{\mu}) = \Pi_{2,n_{train}} p(\boldsymbol{\mu})$, $n_{train} = 2000$, $n'_{train} = 5000$), comparison with the Newton method with the a priori guess $p_0(\boldsymbol{\mu})$ (25)

	Newton its		ave. time	
	$\Pi(\boldsymbol{\mu})$	$p_0(\boldsymbol{\mu})$	$\Pi(\boldsymbol{\mu})$	$p_0(\boldsymbol{\mu})$
$\lvert\mathbf{W}_l - \mathbf{W}_r\rvert_\infty \leq 0.7$	2.9938	3.1426	$3.35 \cdot 10^{-4}$	$3.35 \cdot 10^{-4}$
$\lvert\mathbf{W}_l - \mathbf{W}_r\rvert_\infty \leq 0.9$	3.0238	3.2530	$3.40 \cdot 10^{-4}$	$3.40 \cdot 10^{-4}$

6 Conclusions

In this work we have proposed a new exact Riemann solver based on an offline/online computational decomposition. Our approach has been theoretically justified by Theorem 2.2, which guarantees the smooth dependence of the intermediate states $\mathbf{u}_1, \ldots, \mathbf{u}_{m-1}$ on initial left and right data.

The application of the method to the p-system and to the Euler equation for ideal gases showed that the guesses provided by the offline stage are extremely accurate and guarantee the convergence of the Newton method, while the availability of theoretical a priori guesses is limited to a very small number of differential problems.

We refer to a future work the analysis and the numerical evaluation of the application of this technique in combination in thee framework of finite volume approximation of general nonlinear hyperbolic problems.

References

[1] A. Bressan. *Hyperbolic Systems of Conservation Laws-The One-Dimensional Cauchy Problem.* Oxford Univ. Press, Oxford, 2000.

[2] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 242(0):623 – 647, 2013.

[3] K. Carlberg, J. Ray, and B. van Bloemen Waanders. Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting. *Computational Methods in Applied Mechanics and Engineering*, 2012. (submitted).

[4] L. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, II edition, 2010.

[5] S. K. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb*, 47:357 – 393, 1959.

[6] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[7] A. Quarteroni, L. Sacco, and F. Saleri. *Numerical Mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer, II edition, 2007.

[8] G. Rozza, D. Huynh, and A. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Meth. Eng*, 15:229–275, 2008.

[9] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a practical introduction*. Springer, Berlin, 1997.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

11/2014 Taddei, T.; Quarteroni, A.; Salsa, S.
An offline-online Riemann solver for one-dimensional systems of conservation laws

10/2014 Antonietti, P.F.; Dedner, A.; Madhavan, P.; Stangalino, S.; Stinner, B.; Verani, M.
High order discontinuous Galerkin methods on surfaces

09/2014 Chen, P.; Quarteroni, A.
A new algorithm for high-dimensional uncertainty quantification problems based on dimension-adaptive and reduced basis methods

08/2014 Cattaneo, L; Zunino, P.
A computational model of drug delivery through microcirculation to compare different tumor treatments

07/2014 Agasisti, T.; Ieva, F.; Paganoni, A.M.
Heterogeneity, school-effects and achievement gaps across Italian regions: further evidence from statistical modeling

06/2014 Benzi, M.; Deparis, S.; Grandperrin, G.; Quarteroni, A.
Parameter estimates for the relaxed dimensional factorization preconditioner and application to hemodynamics

05/2014 Rozza, G.; Koshakji, A.; Quarteroni, A.
Free Form Deformation Techniques Applied to 3D Shape Optimization Problems

04/2014 Palamara, S.; Vergara, C.; Catanzariti, D.; Faggiano, E.; Centonze, M.; Pangrazzi, C.; Maines, M.; Quarteroni, A.
Patient-specific generation of the Purkinje network driven by clinical measurements: The case of pathological propagations

03/2014 Kashiwabara, T.; Colciago, C.M.; Dede, L.; Quarteroni, A.
Well-posedness, regulariy, and convergence analysis of the Finite Element approximation of a Generalized Robin boundary value problem

02/2014 Antonietti, P.F.; Sarti, M.; Verani, M.
Multigrid algorithms for high order discontinuous Galerkin methods