

A Comparative Study of Iterative Riemann Solvers for the Shallow Water and Euler Equations

Carlos Muñoz Moncayo^{*1}, Manuel Quezada de Luna^{†1}, and David I. Ketcheson^{‡1}

¹King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

September 11, 2023

Abstract

We investigate the achievable efficiency of exact solvers for the Riemann problem for two systems of first-order hyperbolic PDEs: the shallow water equations and the Euler equations of compressible gas dynamics. Many approximate solvers have been developed for these systems; exact solution algorithms have received less attention because the computation of the exact solution typically requires an iterative solution of algebraic equations, which can be expensive or unreliable. We investigate a range of iterative algorithms and initial guesses. In addition to existing algorithms, we propose simple new algorithms that are guaranteed to converge and to remain in the range of physically admissible values at all iterations. We apply the existing and new iterative schemes to an ensemble of test Riemann problems. For the shallow water equations, we find that Newton's method with a simple modification converges quickly and reliably. For the Euler equations we obtain similar results; however, when the required precision is high, a combination of Ostrowski and Newton iterations converges faster. These solvers are slower than standard approximate solvers like Roe and HLLC, but come within a factor of two in speed. We also provide a preliminary comparison of the accuracy of a finite volume discretization using an exact solver versus standard approximate solvers.

1 Background and Motivation

The Riemann problem is central to both the theory and the numerical solution of hyperbolic partial differential equations (PDEs). Most modern numerical methods for hyperbolic problems – including finite difference, finite volume, and finite element discretizations – are based on approximating the solution of the Riemann problem. In the course of a simulation, billions of Riemann problems may

^{*}carlos.munozmoncayo@kaust.edu.sa

[†]manuel.quezada@kaust.edu.sa

[‡]david.ketcheson@kaust.edu.sa

be solved at each step, so fast solvers are essential. However, the exact solution of the Riemann problem typically requires the solution of a nonlinear algebraic system, which may be costly. It is principally for this reason that approximate Riemann solvers have been preferred, and exact Riemann solvers are rarely, if ever, employed. In this work we use the term *exact* to refer to a solver that guarantees a solution accurate to within a prescribed tolerance, such as machine precision.

We focus on what are perhaps the most-studied hyperbolic systems: the shallow water (or Saint-Venant) equations and the compressible Euler equations. These two systems have a similar structure and have played a key role in the development of numerical methods. A wide variety of approximate Riemann solvers is available for each of them; see [24, 25] for an extensive exposition. The exact solution for either of these systems can be determined by solving a scalar nonlinear algebraic equation. The two major perceived drawbacks of doing so are the cost of an iterative algebraic solver and the lack of robustness (i.e., the lack of a guarantee of convergence).

Although approximate Riemann solvers have long since emerged as preferable to exact solvers for numerical computation, there is a small body of work devoted to developing fast exact Riemann solvers for the Euler equations of gas dynamics; the most advanced methods available were developed about 30 years ago [4, 17]. We mention also one very recent project in which the exact solution was obtained by training a neural network [6]; however, evaluating the neural network was significantly slower than applying traditional approximate solvers.

In simulation, exact solvers can be employed adaptively, to be used in regions of more extreme conditions, while approximate solvers are used in other regions [22, 27, 19]. Exact solvers are also of independent interest, since they are required in the random choice method [3] and may be used for other theoretical purposes. Another potential application of the solvers discussed in this work are the ADER methods, which are designed to attain arbitrarily high order of accuracy for finite volume and finite element solution of balance laws, and require an approximation of the exact Riemann solution for the homogeneous system as an essential step; see e.g. [21, 26, 7]. Furthermore, it seems worthwhile to revisit the question of whether exact Riemann solvers may be of practical use, for three reasons. First, relative changes in the computational cost of flops versus memory access over the last two decades may mean that iterative solvers are now more competitive. Second, some new ideas have appeared in the literature recently regarding potential initial guesses and iterative methods. Third, many codes that make use of Riemann solvers also have other expensive computational parts; the cost of the iterative solver may not be dominant relative to these other components. If the cost is not prohibitive, a robust solver that guarantees a solution with a desired accuracy and required properties is attractive.

We survey and test established methods, while also extending and improving them in various ways. Whereas previous studies [4, 17] focused exclusively on the Euler equations, we also study the shallow water equations, which have a similar structure. We consider a wider range of initial guesses and iterative methods. We also address some questions related to convergence criteria that prevented previous approaches from giving accurate results in certain regimes.

The specific contributions of the present work are:

- the introduction and testing of some new initial guesses and iterative solvers;
- the first methods, for both the Euler and shallow water equations, that are provably-convergent for all initial data; these are also proven to yield a physically-admissible state at every iteration (see Section 2.4.1). For the shallow water equations, this is based on a new proven upper bound for the depth (see Section 2.3.1).

- the most extensive comparison of iterative Riemann solvers to date;
- an updated comparison of the relative computational cost of approximate versus iterative solvers, using typical modern computational hardware (see Section 4).

The remainder of this work is organized as follows. In Section 2 we discuss the shallow water equations, reviewing the structure of the Riemann solution. We review many existing initial guesses and iterative techniques for its solution, and present some new ones. A comparison of all of these methods is given in Section 2.6. In Section 3 we provide a similar analysis and evaluation of iterative solvers for the Riemann problem for the Euler equations. In Section 4, we conduct a preliminary comparison of the best approaches found in Sections 2 and 3 with standard approximate Riemann solvers in finite volume simulations.

The code used for this work is available in the reproducibility repository [15].

2 The shallow water equations

In two space dimensions, the shallow water equations take the form

$$\partial_t \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) + \partial_y \mathbf{g}(\mathbf{q}) = \mathbf{0}, \quad (1)$$

where

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{f}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \mathbf{g}(\mathbf{q}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}. \quad (2)$$

In this section we briefly review the exact solution of this problem and then describe several iterative algorithms for its approximation.

2.1 Exact solution of the Riemann problem

For simplicity we consider a Riemann problem with coordinate-aligned initial data. The solution for any other plane-wave Riemann problem can be obtained from what follows by simple rotation. The structure of the Riemann solution is then the same as for the one-dimensional shallow water equations with a passive tracer:

$$\partial_t \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) = \mathbf{0}, \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}^+, \quad (3a)$$

$$\mathbf{q}(x, 0) = \begin{cases} \mathbf{q}_\ell, & \text{if } x < 0, \\ \mathbf{q}_r, & \text{if } x > 0. \end{cases} \quad (3b)$$

Here we review the solution structure; a more detailed discussion can be found in several extensive works [24, 12, 9]. It is convenient to make use of the vector of primitive variables $\mathbf{w} = [h, u, v]^T$. The solution involves the initial states \mathbf{w}_ℓ and \mathbf{w}_r , given by the initial data (3b), as well as two intermediate states $\mathbf{w}_{*\ell}$, \mathbf{w}_{*r} . These states are connected by three waves. We would like to determine the middle states $\mathbf{w}_{*\ell} = (h_*, u_*, v_{*\ell})^T$, $\mathbf{w}_{*r} = (h_*, u_*, v_{*r})^T$ and the structure of the waves connecting them. The characteristic speeds of the system (3a) are given by the eigenvalues of the flux Jacobian $A(\mathbf{q}) = \mathbf{f}'(\mathbf{q})$:

$$\lambda_1 = u - \sqrt{gh} \leq \lambda_2 = u \leq \lambda_3 = u + \sqrt{gh}. \quad (4)$$

We will refer to the wave associated with a given eigenvalue by the corresponding index. For this problem, the 2-wave is a linearly degenerate contact wave (or contact discontinuity). The values of h and u are constant across the 2-wave, so we write simply $h_* = h_{*\ell} = h_{*r}$ and $u_* = u_{*\ell} = u_{*r}$. The 1- and 3-waves are genuinely nonlinear. The 1-wave is a shock if $h_* > h_\ell$, in which case the states on either side of the 1-wave are related by the Rankine-Hugoniot condition. On the other hand, if $h_* < h_\ell$, the 1-wave is a rarefaction, in which case the states on either side are related through the corresponding Riemann invariant. Similar statements apply to the 3-wave and the relation between h_* and h_r . Taking these conditions together, one finds that h_* is given by the unique real root of

$$\phi(h; \mathbf{q}_\ell, \mathbf{q}_r) = f(h; h_\ell) + f(h; h_r) + u_r - u_\ell, \quad (5a)$$

where

$$f(h; h_{\ell/r}) = \begin{cases} f_R(h; h_{\ell/r}) := 2(\sqrt{gh} - \sqrt{gh_{\ell/r}}), & \text{if } h \leq h_{\ell/r} \text{ (rarefaction),} \\ f_S(h; h_{\ell/r}) := (h - h_{\ell/r})\sqrt{\frac{1}{2}g\left(\frac{h+h_{\ell/r}}{hh_{\ell/r}}\right)}, & \text{if } h > h_{\ell/r} \text{ (shock).} \end{cases} \quad (5b)$$

The function ϕ is sometimes referred to as the depth function [24]. Since we are interested in finding a root of ϕ for (fixed but arbitrary) given states \mathbf{q}_ℓ and \mathbf{q}_r , we will write simply $\phi(h)$ for brevity. Once h_* is found, the corresponding velocity is given by

$$u_* = \frac{1}{2}(u_\ell + u_r) + \frac{1}{2}[f(h_*; h_r) - f(h_*; h_\ell)]. \quad (6)$$

In this work we focus on Riemann problems for which the solution satisfies $h > 0$ at all points. Riemann problems in which dry states appear or are initially present can be detected *a priori* and solved by explicit formulas. For practical use, an implementation of the dry-state treatment presented in [24, Sec. 6.3] is provided in the reproducibility repository for this work [15].

2.2 Properties of the depth function

For $h > 0$, the first three derivatives of $f(h, h_{\ell/r})$ are given by

$$f'(h; h_{\ell/r}) = \begin{cases} \frac{\sqrt{g}}{\sqrt{h}}, & \text{if } 0 < h \leq h_{\ell/r}, \\ \frac{g(2h^2 + hh_{\ell/r} + h_{\ell/r}^2)}{2\sqrt{2}h^2\sqrt{g\left(\frac{1}{h} + \frac{1}{h_{\ell/r}}\right)}h_{\ell/r}}, & \text{if } h > h_{\ell/r} > 0, \end{cases} \quad (7a)$$

$$f''(h; h_{\ell/r}) = \begin{cases} \frac{-\sqrt{g}}{2\sqrt{h^3}}, & \text{if } 0 < h \leq h_{\ell/r}, \\ \frac{-g^2(5h + 3h_{\ell/r})}{4\sqrt{2}h^4\sqrt{g^3\left(\frac{1}{h} + \frac{1}{h_{\ell/r}}\right)^3}}, & \text{if } h > h_{\ell/r} > 0, \end{cases} \quad (7b)$$

and

$$f'''(h; h_{\ell/r}) = \begin{cases} \frac{3\sqrt{g}}{4\sqrt{h^5}}, & \text{if } 0 < h \leq h_{\ell/r}, \\ \frac{3gh_{\ell/r}(10h^2 + 13hh_{\ell/r} + 5h_{\ell/r}^2)}{8\sqrt{2}h^4\sqrt{g\left(\frac{1}{h} + \frac{1}{h_{\ell/r}}\right)}(h+h_{\ell/r})^2}, & \text{if } h > h_{\ell/r} > 0, \end{cases} \quad (7c)$$

Observe that (for $h > 0$) $\phi'(h) = f'(h; h_\ell) + f'(h; h_r) > 0$ regardless of whether the nonlinear waves are shocks or rarefactions. Similarly, (for $h > 0$) $\phi''(h) < 0$ and $\phi'''(h) > 0$ [24]. Finally, observe that $\phi(0) \leq 0$ and $\lim_{h \rightarrow \infty} \phi(h) = +\infty$. Together, these properties guarantee that $\phi(h)$ has a unique real root. Also, letting

$$h_{\min} := \min(h_\ell, h_r), \quad (8a)$$

$$h_{\max} := \max(h_\ell, h_r), \quad (8b)$$

we have that if $\phi(h_{\min}) > 0$ both waves are rarefactions, and if $\phi(h_{\max}) < 0$ both waves are shocks.

2.3 Computation of the initial guess

Since we will proceed iteratively to approximate h_* , it is advantageous to start with an accurate initial guess $h^{[0]}$ that can be computed efficiently. Moreover, as we will see, a lower bound $0 < h_- \leq h_*$ will also be required for the iterative part of most algorithms. We describe seven possible initial guesses in the following sections.

2.3.1 Two-rarefaction approximation (RR)

If we assume that the 1- and 3- waves are rarefactions, then the root of ϕ can be computed explicitly:

$$h_{\text{RR}} := \frac{(u_\ell - u_r + 2\sqrt{gh_\ell} + 2\sqrt{gh_r})^2}{16g}. \quad (9)$$

In case one or both of the waves is a shock, this is not the exact solution but can serve as an initial guess. This estimate never underestimates the true depth, as we now prove (see a similar result obtained by Guermond & Popov for the Euler equations [5, Section 4.1]).

Lemma 2.1. $h_* \leq h_{\text{RR}}$.

Proof. Let

$$\phi_{\text{RR}}(h) := 2(\sqrt{gh} - \sqrt{gh_\ell}) + 2(\sqrt{gh} - \sqrt{gh_r}) + u_r - u_\ell = f_R(h, h_\ell) + f_R(h, h_r) + u_r - u_\ell, \quad (10)$$

which is the corresponding $\phi(h)$ function if both waves were rarefactions. Let us first prove that

$$h > h_{\ell r} \quad \Rightarrow \quad f_R(h, h_{\ell r}) < f_S(h, h_{\ell r}). \quad (11)$$

Rewrite $f_R(h, h_{\ell r})$ and $f_S(h, h_{\ell r})$ as

$$f_R = 2\sqrt{gh_{\ell r}} \left(\frac{x-1}{\sqrt{x}+1} \right), \quad f_S = \frac{\sqrt{gh_{\ell r}}}{\sqrt{2}} \left(\frac{(x-1)\sqrt{x+1}}{\sqrt{x}} \right),$$

where $x := \frac{h}{h_{\ell r}}$. We want to show that $f_R(x) < f_S(x)$ when $x > 1$. To see this, rewrite $f_S(x)$ as

$$f_S(x) = \underbrace{\frac{\sqrt{x+1}(\sqrt{x}+1)}{2\sqrt{2}\sqrt{x}}}_{=:g(x)} \left[2\sqrt{gh_{\ell r}} \left(\frac{x-1}{\sqrt{x}+1} \right) \right] = g(x)f_R(x).$$

The first derivative of $g(x)$ is

$$g'(x) = \frac{\sqrt{x^3} - 1}{4\sqrt{2x^3(x+1)}}.$$

Finally, note that $g(1) = 1$ and $g'(x) > 0, \forall x > 1$. Therefore, $g(x) > 1, \forall x > 1 \implies f_R(x) < f_S(x), \forall x > 1$. If both waves are rarefactions there is nothing to prove. If both waves are shocks, then for every $h > h_{\max}$

$$\begin{aligned} \phi(h) &= f_S(h, h_{\min}) + f_S(h, h_{\max}) + u_r - u_\ell \\ &> f_R(h, h_{\min}) + f_R(h, h_{\max}) + u_r - u_\ell \\ &= \phi_{RR}(h). \end{aligned}$$

Therefore $h_* < h_{RR}$. If there is one rarefaction and one shock, we proceed analogously. ■

2.3.2 Average depth (AV)

A simple and inexpensive initial guess is the arithmetic average:

$$h^{[0]} = h_{AV} := \frac{h_\ell + h_r}{2}. \quad (12)$$

2.3.3 Quadratic approximation (QA)

We also consider an initial guess presented in Lemma 3.8 of the recent work by Azerad et. al.[1], given by

$$h^{[0]} = h_{QA} := \begin{cases} h_{RR} & \text{if } 0 \leq \phi(x_0 h_{\min}), \\ \sqrt{h_{\min} h_{\max}} \left(1 + \frac{\sqrt{2}(u_\ell - u_r)}{\sqrt{g h_{\min} + \sqrt{g h_{\max}}}} \right) & \text{if } \phi(x_0 h_{\max}) < 0, \\ \left(-\sqrt{2h_{\min}} + \sqrt{3h_{\min} + 2\sqrt{2h_{\min} h_{\max}} + \sqrt{\frac{2}{g}}(u_\ell - u_r)\sqrt{h_{\min}}} \right)^2 & \text{otherwise,} \end{cases} \quad (13)$$

where $x_0 = (2\sqrt{2} - 1)^2$. It also satisfies the property $h_{QA} \geq h_*$.

2.3.4 Primitive-variable linearized solver (PV)

Any approximate-state Riemann solver can be used to provide an initial guess for an iterative solver. Here we consider the use of the linearized solver introduced by Toro [24]. Linearizing the system around the state $\hat{\mathbf{w}}$ in primitive variables gives

$$\mathbf{w}_t + \mathbf{A}(\hat{\mathbf{w}})\mathbf{w}_x = \mathbf{0}. \quad (14)$$

Solving this system exactly we get

$$\hat{h} = \frac{1}{2}(h_\ell + h_r) + \frac{\frac{1}{2}(u_\ell - u_r)}{\sqrt{2g/(h_\ell + h_r)}}. \quad (15)$$

We consider a modification of this solution, constructed to enforce positivity preservation [24]

$$h^{[0]} = h_{\text{PV}} := \frac{1}{2}(h_\ell + h_r) + \frac{\frac{1}{4}(u_\ell - u_r)(h_\ell + h_r)}{\sqrt{gh_\ell} + \sqrt{gh_r}}. \quad (16)$$

2.3.5 Two-shock approximation (SS)

If we assume that both waves are shocks, then the depth function (5) becomes

$$\phi_{\text{SS}}(h) = (h - h_\ell)\sqrt{\frac{1}{2}g\left(\frac{h + h_\ell}{h h_\ell}\right)} + (h - h_r)\sqrt{\frac{1}{2}g\left(\frac{h + h_r}{h h_r}\right)} + u_r - u_\ell. \quad (17)$$

Since we cannot find a root of this function explicitly, we approximate it linearly as is done by Toro [24]. We set $y_{\ell/r}(h) = \sqrt{\frac{1}{2}g\left(\frac{h + h_{\ell/r}}{h h_{\ell/r}}\right)}$ and assume that we have an estimate of the root, denoted by \bar{h} . Then our problem becomes

$$(h - h_\ell)y_\ell(\bar{h}) + (h - h_r)y_r(\bar{h}) + u_r - u_\ell = 0, \quad (18)$$

and it follows that

$$h^{[0]} = h_{\text{SS}} := \frac{h_\ell y_\ell(\bar{h}) + h_r y_r(\bar{h}) - u_r + u_\ell}{y_\ell(\bar{h}) + y_r(\bar{h})}. \quad (19)$$

Here we use the estimate $\bar{h} = h_{\text{PV}}$.

2.3.6 Convex combination

If $\phi(h_{\min}) > 0$, the solution has two rarefactions and h_* is given explicitly by (9). Otherwise, we define the lower bound

$$h_- = \begin{cases} h_{\max} & \text{if } \phi(h_{\max}) < 0 \\ h_{\min} & \text{if } \phi(h_{\min}) < 0 < \phi(h_{\max}) \end{cases} \quad \begin{array}{l} \text{(1- and 3-shocks),} \\ \text{(1- (3-) shock and 3- (1-) rarefaction),} \end{array} \quad (20)$$

and the upper bound

$$h_+ = \begin{cases} h_{\text{RR}} & \text{if } \phi(h_{\max}) < 0 \\ \min(h_{\max}, h_{\text{RR}}) & \text{if } \phi(h_{\min}) < 0 < \phi(h_{\max}) \end{cases} \quad \begin{array}{l} \text{(1- and 3-shocks),} \\ \text{(1- (3-) shock and 3- (1-) rarefaction),} \end{array} \quad (21)$$

for h_* .

We can compute an initial guess that is a convex combination of these bounds, choosing the coefficients such that the initial guess is biased towards the bound whose residual is closer to 0,

$$h^{[0]} = h_{\text{CC}} := \frac{|\phi(h_+)|}{|\phi(h_-)| + |\phi(h_+)|} h_- + \frac{|\phi(h_-)|}{|\phi(h_-)| + |\phi(h_+)|} h_+ = \frac{\phi(h_+)h_- - \phi(h_-)h_+}{\phi(h_+) - \phi(h_-)}. \quad (22)$$

This initial guess can be further interpreted as the root of an affine function that intersects with ϕ at h_- and h_+ .

2.3.7 HLLE

We also consider the HLLE solver, first introduced by Harten, Lax, and van Leer, and then improved by Einfeldt [2]. It is based on the assumption that the solution consists of three constant states, q_ℓ , q_m , and q_r , separated by two waves. Assuming that the speeds of these waves are s_ℓ and s_r and, by imposing conservation on the approximate solution, we get

$$q_m = \frac{f(q_r) - f(q_\ell) - s_r q_r + s_\ell q_\ell}{s_\ell - s_r}. \quad (23)$$

Here s_ℓ and s_r are chosen in terms of the eigenvalues of f' evaluated at the Roe average [12]. Thus, we can set $h^{[0]} = h_{\text{HLLE}} := h_m$.

2.4 Iterative algorithms

There are several approaches that can be taken for the iterative process. In this section we describe the algorithms tested in this study.

2.4.1 Positive Newton method

Perhaps the most widely used root-finding method is the Newton-Rhapson iteration, also known as Newton's method. For a function $\psi(x)$, this method starts with an initial guess $x^{[0]}$ and proceeds by the iteration

$$x^{[k+1]} = x^{[k]} - \frac{\psi(x^{[k]})}{\psi'(x^{[k]})}. \quad (24)$$

In general, Newton's method is not guaranteed to converge. But for functions sharing some of the properties of $\phi(h)$, convergence is guaranteed by starting from the left.

Lemma 2.2. *Let $\psi(x) : (0, \infty) \rightarrow \mathbb{R}$ be a real-valued, twice continuously-differentiable function such that $\lim_{x \rightarrow 0} \psi(x) < 0$, $\psi'(x) > 0$, and $\psi''(x) < 0$. Then ψ has a unique real root $x_* > 0$. Let $x^{[0]} \leq x_*$. Then the sequence $x^{[k]}$ produced by Newton's method (24) has the properties that $x^{[k]} \in [x_0, x_*]$ for all k and $x^{[k]} \rightarrow x_*$ as $k \rightarrow \infty$.*

Proof. The existence of a unique real root follows from $\lim_{x \rightarrow 0} \psi(x) < 0$ and $\psi'(x) > 0$.

To prove the next property, assume by induction that $x^{[j]} \in [x_0, x_*]$ for $j = 0, \dots, k$. Then the properties of ψ guarantee that $\psi(x^{[k]}) < 0$ and $\psi'(x^{[k]}) \geq \psi'(x_*) > 0$. This implies that $x^{[k]} < x^{[k+1]} < x_*$.

Since the sequence of iterates is strictly increasing and bounded from above, it converges to some value \hat{x} and is a Cauchy sequence, i.e.

$$\lim_{k \rightarrow \infty} (x^{[k+1]} - x^{[k]}) = 0. \quad (25)$$

Since ψ' is continuous, it is bounded in $[x_0, x_*]$. By this, the continuity of ϕ , (25) and (24), we have

$$0 = \lim_{k \rightarrow \infty} \psi'(x^{[k]})(x^{[k+1]} - x^{[k]}) = \lim_{k \rightarrow \infty} \psi(x^{[k]}) = \psi(\hat{x}).$$

But since ψ has a unique real root, we must have $\hat{x} = x_*$. ■

Corollary 2.2.1. *Let left and right states $\mathbf{q}_\ell, \mathbf{q}_r$ be given and consider the application of Newton's method (24) to the depth function (5). If the initial guess satisfies $0 \leq h^{[0]} \leq h_*$, then the iteration converges to h_* .*

Proof. If $h_* = 0$, then $h^{[0]} = h_*$. Otherwise, we can apply Lemma 2.2. ■

In order to define an initial guess that satisfies Corollary 2.2.1, we replace our initial guess $h^{[0]}$ with (see [5], where the analogous quantity is used for the Euler equations)

$$\tilde{h}^{[0]} := \max \left(h_{\min}, h^{[0]} - \phi \left(h^{[0]} \right) / \phi' \left(h^{[0]} \right) \right). \quad (26)$$

Indeed, since ϕ is concave and strictly increasing, we have

$$0 = \phi(h_*) \leq \phi(h^{[0]}) + \phi'(h^{[0]})(h_* - h^{[0]}), \quad (27)$$

$$h^{[0]} - \phi \left(h^{[0]} \right) / \phi' \left(h^{[0]} \right) \leq h_*. \quad (28)$$

Moreover, if $h_{\min} \leq h_*$, then $\tilde{h}^{[0]} \leq h_*$. Thus, by Lemma 2.2, Newton's method not only is guaranteed to converge, but it also guarantees positivity of all $h^{[k]}$. For this reason, we refer to the algorithm given by (26) combined with (24) as *positive Newton*.

2.4.2 Two-step Newton method

This method, introduced by McDougall & Wotherspoon [14], consists of a two-step predictor-corrector iteration that yields a convergence rate of $1 + \sqrt{2}$, with the same number of function evaluations per iteration as the Newton-Raphson method presented above. Its first iteration is given by

$$x^{[\frac{1}{2}]} = x^{[0]}, \quad x^{[1]} = x^{[0]} - \frac{\psi(x^{[0]})}{\psi' \left(\frac{1}{2} [x^{[0]} + x^{[\frac{1}{2}}]] \right)} = x^{[0]} - \frac{\psi(x^{[0]})}{\psi'(x^{[0]})}, \quad (29)$$

while for $k > 1$ we have

$$x^{[k+\frac{1}{2}]} = x^{[k]} - \frac{\psi(x^{[k]})}{\psi' \left(\frac{1}{2} [x^{[k-1]} + x^{[k-\frac{1}{2}}]] \right)}, \quad (30)$$

$$x^{[k+1]} = x^{[k]} - \frac{\psi(x^{[k]})}{\psi' \left(\frac{1}{2} [x^{[k]} + x^{[k+\frac{1}{2}}]] \right)}.$$

2.4.3 Ostrowski's method

Ostrowski's method is cubically convergent and given by the two-step iteration [16]

$$x^{[k+\frac{1}{2}]} = x^{[k]} - \frac{\psi(x^{[k]})}{\psi'(x^{[k]})}, \quad (31a)$$

$$x^{[k+1]} = x^{[k+\frac{1}{2}]} - \frac{\psi \left(x^{[k+\frac{1}{2}} \right)}{\psi'(x^{[k]})} \cdot \frac{\psi(x^{[k]})}{\psi(x^{[k]}) - 2\psi \left(x^{[k+\frac{1}{2}} \right)}. \quad (31b)$$

Although higher-order generalizations of Ostrowski's method have been proposed, the original version (31) has proved to be the most efficient in several contexts. See, e.g [16], [11].

2.4.4 Ostrowski-Newton method

An efficient combination of Ostrowski's and Newton's methods can be obtained by performing a single Ostrowski iteration and then proceeding with Newton iterations (if the convergence criterion is not satisfied already). The Newton iteration is thus initialized using $\tilde{h}^{[0]}$ as defined in (26), but where $h^{[0]}$ is the result of one iteration of Ostrowski's method.

2.4.5 Bounding quadratic polynomials

This iterative method is based on an algorithm proposed by Guermond & Popov [5] in the context of the Euler equations, but is adapted in a straightforward way to the shallow water equations. Assuming we are provided two estimates of x_* , $(x_-^{[0]}, x_+^{[0]})$, which are also (respectively) lower and upper bounds, we consider the Hermite polynomials

$$P_-(x) = \psi(x_-^{[k]}) + \psi[x_-^{[k]}, x_-^{[k]}](x - x_-^{[k]}) + \psi[x_-^{[k]}, x_-^{[k]}, x_+^{[k]}](x - x_-^{[k]})^2, \quad (32a)$$

$$P_+(x) = \psi(x_+^{[k]}) + \psi[x_+^{[k]}, x_+^{[k]}](x - x_+^{[k]}) + \psi[x_-^{[k]}, x_+^{[k]}, x_+^{[k]}](x - x_+^{[k]})^2. \quad (32b)$$

Here

$$\psi[x_-^{[k]}, x_-^{[k]}] = \psi'(x_-^{[k]}), \quad \psi[x_-^{[k]}, x_-^{[k]}, x_+^{[k]}] = \frac{\psi[x_-^{[k]}, x_+^{[k]}] - \psi[x_-^{[k]}, x_-^{[k]}]}{x_+^{[k]} - x_-^{[k]}}, \quad \psi[x_-^{[k]}, x_+^{[k]}] = \frac{\psi(x_+^{[k]}) - \psi(x_-^{[k]})}{x_+^{[k]} - x_-^{[k]}}$$

are divided differences. It can be shown that $P_-(x) \leq \psi(x) \leq P_+(x)$ for $x \in [x_-^k, x_+^k]$. Both P_- and P_+ interpolate $\psi(x)$ at x_-^k and x_+^k and both have one root in the interval $[x_-^k, x_+^k]$. These roots are given by

$$x_{\pm}^{[k+1]} = x_{\pm}^{[k]} - \frac{2\psi(x_{\pm}^{[k]})}{\sqrt{\psi'(x_{\pm}^{[k]})^2 - 4\psi(x_{\pm}^{[k]})\psi[x_-^{[k]}, x_{\pm}^{[k]}, x_+^{[k]}]}}. \quad (33a)$$

Proceeding analogously to Lemma 4.5 of Guermond & Popov [5], we see that $x_-^{[k+1]} < x_* < x_+^{[k+1]}$. Moreover, the convergence of both $x_-^{[k]}$ and $x_+^{[k]}$ is cubic.

As in the case of the positive Newton method, if $h_{\min} \leq h_*$, we can use $h_-^{[0]} := h_{\min}$ as initial lower bound. Furthermore Lemma 2.1 ensures that $h_+^{[0]} := h_{\text{RR}}$ is always an upper bound for h_* .

Remark 1. *We have designed and implemented a similar approach based on cubic polynomials and yielding quartic convergence. However, due to the difficulty of solving the cubic, that method was less efficient and robust than other methods considered here, so we omit its description.*

2.4.6 Single quadratic polynomial

The method of bounding quadratic polynomials was originally introduced to obtain an upper bound for the intermediate pressure in the Riemann problem for the EEs, such that a prescribed error for the maximum wave speed is guaranteed. However, we are interested in approximating the middle state iteratively until certain convergence criterion, that relies uniquely on the current iterate, is met (see Section 2.5). Thus, the computation of two bounds for h_* , $h_-^{[k]}$ and $h_+^{[k]}$, at each step is

not necessary. Instead, we can use a single quadratic polynomial to approximate ϕ from below and whose root, $h_+^{[k]}$, approximates h_* from above. This polynomial is constructed as in (32a), but the lower bound is fixed as $h_-^{[k]} = h_-^{[0]}$.

2.4.7 Single linear polynomial

A further simplification of the previous method is to use a single linear polynomial to approximate ϕ . This polynomial interpolates ϕ at the lower and upper bounds of h_* , $h_-^{[0]}$ and $h_+^{[k]}$; moreover, its root,

$$h_+^{[k+1]} = \frac{\phi(h_+^{[k]})h_-^{[0]} - \phi(h_-^{[0]})h_+^{[k]}}{\phi(h_+^{[k]}) - \phi(h_-^{[0]})}, \quad (34)$$

is an approximation of h_* from above by the concavity of ϕ .

2.5 Termination criteria

An appropriate convergence criterion must be established in order to properly assess and compare the efficiency of the iterative solvers. One standard approach is to terminate if the residual is below a specified tolerance:

$$|\phi(h^{[k]})| < \varepsilon_r. \quad (35)$$

This can be checked at no extra cost for the methods considered here, since the depth function must be evaluated at each step for the next approximation. Note that condition (35) implies the rough bound

$$|h^{[k]} - h_*| \lesssim \frac{\varepsilon_r}{|\phi'(h_*)|}, \quad (36)$$

which is practically the best that one can hope for based on the conditioning of the problem. In particular, in floating-point arithmetic, the best possible solution is accurate to approximately $\varepsilon_m/|\phi'(h_*)|$, where ε_m is the unit roundoff. On the other hand, (36) implies that the requirement (35) can be too difficult to satisfy when $|\phi'(h_*)| \gg 1$, as discussed by Quarteroni et al. [18, Ch. 6.5].

Another widely used convergence criterion is based on stagnation, i.e., stopping when the relative change in the solution from one iteration to the next is below a specified tolerance, as is done by Toro [24]:

$$\frac{|h^{[k]} - h^{[k-1]}|}{\frac{1}{2}|h^{[k]} + h^{[k-1]}|} < \varepsilon_s. \quad (37)$$

From the estimate [18]

$$|h^{[k-1]} - h_*| \approx \left| \frac{h^{[k]} - h^{[k-1]}}{1 - \phi'(h_*)} \right|, \quad (38)$$

we obtain that the bound (37) implies

$$|h^{[k-1]} - h_*| \lesssim \frac{1}{2} |h^{[k]} + h^{[k-1]}| \frac{\varepsilon_s}{|1 - \phi'(h_*)|}. \quad (39)$$

It is clear that when $|\phi'(h_*) - 1| \ll 1$ or $\gg 1$, the stagnation criterion leads to the same difficulties faced with the residual criterion. To obtain a bound on the error of $h^{[k-1]}$, the computation of an extra iterate is required if we desire to use stagnation. For this reason, the results presented in this work were obtained with the residual convergence criterion (35). There exist modifications of the residual criterion, like [8]

$$|\phi(h^{[k]})| < |\phi(h^{[0]})|\varepsilon_{r1} + \varepsilon_{r2}, \quad (40)$$

that ensure (for a fixed initial guess strategy) that the method be invariant under scaling of the problem. Since we want to compare the influence of different initial guesses in the performance of the iterative solvers, we prefer to use (35) in order to avoid a bias that would favor an inaccurate initial guess.

Remark 2. *The bracketing property of the bounding quadratic polynomials from Section 2.4.5, $h_-^{[k]} \leq h_* \leq h_+^{[k]}$, allows the use of an absolute error measure as a termination criterion, $h_+^{[k]} - h_* \leq h_+^{[k]} - h_-^{[k]} < \varepsilon_a$, instead of a relative one. This is particularly useful when $|\phi'(h_*)| \gg 1$ and the residual termination criterion becomes hard to satisfy. However, out of all the methods considered in this work, the absolute error criterion is only suitable for the method of bounding quadratic polynomials.*

2.6 Performance comparison

In this section we study how quickly each method reaches a desired solution accuracy, in wall clock time. In order to do so, we run each method on a fixed set of 10^7 Riemann problems generated pseudo-randomly¹. Following the approach used by Gottlieb & Groth [4], 20% of the problems involve strong waves and 80% involve only weak waves. For the strong wave problems, the initial depth on each side is taken as $h = 10^k$ with k distributed uniformly in $[-4, 4]$ ($k \sim \mathcal{U}([-4, 4])$); the velocities are taken as $u_\ell = 10^k$ and $u_r = -10^k$, with $k \sim \mathcal{U}([-2, 2])$. For the weak waves the depth is taken as $h_\ell, h_r \sim \mathcal{U}([0.1, 1])$, and velocities are set to zero. The rationale behind this approach is that in numerical applications most Riemann problems that arise will involve relatively nearby states.

The Riemann solvers were implemented in Fortran 90, the sets of Riemann problems were provided to the modules compiled with f2py 1.22.3 and GNU Fortran compiler 11.2.0. Similar tests were performed using Intel's Fortran compiler and, noting that the proportion of execution times between different methods was preserved, for the sake of reproducibility we present the results obtained with the GNU compiler.

2.6.1 Comparison of initial guesses

To evaluate the various initial guesses on an equal footing, we test each of them as a starting point for the positive Newton method. As a benchmark for accuracy we use the average relative initial error (ARIE) over the set of problems, defined as $\frac{1}{10^7} \sum_{i=1}^{10^7} |h_{*i} - h^{[0]}_i| / |h_{*i}|$.

In our implementation, we always check for a two-rarefaction solution and use the exact solution (9) in that case. This step ensures that $h_{\min} \leq h_*$ which is crucial to avoid negative depth with the positive Newton, Ostrowski-Newton, and bounding polynomials methods. For the initial guesses

¹using the Mersenne Twister algorithm provided by the *random* library in Python 3.9.7.

Initial guess	Time	Average iterations	Average relative initial error	
			Weak waves	Strong waves
AV (average) (12)	0.900 s	3.0	4.75%	3500%
RR (two-rarefaction) (9)	0.906 s	2.5	0.6%	4200%
QA (quadratic approximation) (13)	1.013 s	2.3	0.6%	1.18%
CC (convex combination) (22)	1.042 s	2.0	0.08%	4.95%
PV (primitive-variables) (16)	0.959 s	3.0	4.75%	4300%
SS (two-shock) (19)	0.906 s	2.2	0.12%	10.95%
HLLE (23)	1.093 s	3.1	7.71%	4100%

Table 1: Comparison of execution time, average iterations, and average relative initial error (before iterating) obtained using the positive Newton iterative solver for the shallow water equations with different initial guesses.

that make use of the lower bound, we can obtain improved accuracy by using h_{\max} if $\phi(h_{\max}) < 0$. It can be shown that h_{\max} is a lower bound for h_* in this case because of the monotonicity of the depth function and the entropy conditions for the shallow water equations; the same estimate is used in [5] for the Euler equations. Since the two-rarefaction solution can be checked *a priori* and written explicitly, we study only problems for which the solution involves at least one shock.

The results are presented in Table 1. In terms of speed, results for all methods are within about 20% of each other. We see that the initial guess achieving the fewest average iterations is the CC initial guess (22), followed by SS (19). The AV initial guess gives a slightly faster run-time because it is cheaper to initialize and because the strong wave problems (for which it yields a large initial error and hence requires more iterations) compose only a small fraction of the test set.

The two most accurate initial guesses are CC (22) for problems with weak waves and QA (13) for problems with strong waves. Both of these have slower than average overall run-time due to requiring three depth function evaluations in the worst case. Depending on the desired accuracy, this suggests the possibility of using an approximate state Riemann solver that may not need to be refined through iteration, as is done in, e.g., [24, Chap. 10.3], [25, Chap. 9]. An adaptive noniterative Riemann solver using the CC and QA initial guesses is described in Appendix A.

The SS approximation is nearly as efficient as the fastest methods and is nearly as accurate as the most accurate initial guesses (CC and QA).

2.6.2 Comparison of root-finding algorithms

We now proceed to compare the various iterative algorithms. Where possible, we use the two-shock approximation initial guess, since it gave the best trade-off between initial accuracy and overall execution time in the previous comparison. We consider two different values for the tolerance: 10^{-6} and 10^{-12} . The results, presented in Table 2, show that all methods, except the ones based on interpolating quadratic and linear polynomials (Sections 2.4.5 - 2.4.7), fall within about 10% of each other. The Ostrowski-Newton method is the fastest for both tolerances. We can attribute this to the fact that Ostrowski’s method attains the convergence criteria in the first half-step for most of the problems, as reflected in the number of iterations, while the positive Newton method avoids nonphysical iterates as described in Section 2.4.1. We also note that the two-step Newton method

Method \ Tolerance	10^{-6}		10^{-12}	
	Time	Iterations	Time	Iterations
Positive Newton-SS	0.732 s	1.4	0.874 s	2.2
Ostrowski-SS	0.724 s	1.0	0.860 s	1.3
Ostrowski-Newton-SS	0.668 s	1.1	0.780 s	1.4
Two-step Newton-SS	0.710 s	1.4	0.867 s	2.2
Bounding quadratic polynomials-RR	1.252 s	2.1	1.672 s	2.6
Single quadratic polynomial-RR	0.918 s	2.1	1.237 s	2.6
Single linear polynomial-RR	0.805 s	2.3	1.154 s	3.3

Table 2: Comparison of execution time for different iterative Riemann Solvers for shallow water equations with their best-performing initial guess

of Section 2.4.2 is slightly faster than the positive Newton method of Section 2.4.1. However, one might still prefer the latter due to its outstanding robustness and provable convergence.

For Riemann problems with extremely large and/or small initial states, two additional challenges arise. For the solvers in which positivity is not guaranteed, the depth may become negative. For all solvers, it may be impossible to reach a prescribed small tolerance for the residual because $|\phi'(h^{[k]})| \gg 1$, as discussed in Section 2.5. This can be alleviated in the case of the bounding quadratic polynomials by terminating the iterative process when either the residual-based or the absolute error (see Remark 2) criteria are satisfied. The positive Newton’s and Ostrowski-Newton’s methods are remarkably less prone than other solvers to stagnate before reaching the prescribed tolerance. Since a very low number of iterations is required for typical Riemann problems, one way to deal with this issue is to fix the maximum number of iterations *a priori* as suggested by Toro [25].

3 The Euler equations

3.1 Exact solution of the Riemann problem

Just as for the shallow water equations, the Riemann problem for the Euler equations of an ideal compressible gas in 2D or 3D reduces essentially to the 1D problem if we consider a planar, coordinate-aligned Riemann problem; see e.g. [12, Section 18.8.1]. In that case we have

$$\partial_x \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) = \mathbf{0}, \quad (41a)$$

$$\mathbf{q}(x, 0) = \begin{cases} \mathbf{q}_\ell & \text{if } x < 0, \\ \mathbf{q}_r & \text{if } x > 0, \end{cases} \quad (41b)$$

with

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}, \quad (42)$$

and

$$E = \rho \left(\frac{1}{2} u^2 + e \right), \quad e = \frac{p}{(\gamma - 1)\rho}, \quad (43)$$

where the heat capacity ratio γ is assumed to be constant.

To describe the exact solution for (41) it is useful to use the primitive variables $\mathbf{w} = (\rho, u, p)^T$. The solution consists of three waves, each one associated with a characteristic speed of the system, given by the eigenvalues of the flux Jacobian $\mathbf{A}(\mathbf{q}) = \mathbf{f}'(\mathbf{q})$:

$$\lambda_1 = u - a, \quad \lambda_2 = u, \quad \lambda_3 = u + a, \quad (44)$$

where $a = \sqrt{\frac{\gamma p}{\rho}}$ is the sound speed. The 1- and 3- waves are genuinely nonlinear, and can be either shocks or rarefactions, while the 2- wave is linearly degenerate and thus a contact discontinuity. They separate four constant states $\mathbf{w}_\ell, \mathbf{w}_{*\ell}, \mathbf{w}_{*r}, \mathbf{w}_r$, where \mathbf{w}_ℓ and \mathbf{w}_r are given by the initial data and the intermediate states are given by $\mathbf{w}_{*\ell} = (\rho_{*\ell}, u_*, p_*)$ and $\mathbf{w}_{*r} = (\rho_{*r}, u_*, p_*)$. Thus the unknown quantities are $p_*, u_*, \rho_{*\ell}, \rho_{*r}$. If $p_* < p_\ell$ the 1-wave is a rarefaction and connects \mathbf{w}_ℓ and $\mathbf{w}_{*\ell}$ through the isentropic relation and the generalised Riemann invariants. If $p > p_\ell$ the 1-wave is a shock, so \mathbf{w}_ℓ and $\mathbf{w}_{*\ell}$ are related by the Rankine-Hugoniot conditions. Analogous statements hold for the 3-wave. Taking all of this into account we obtain that p_* is the root of the pressure function

$$\phi(p; \mathbf{w}_\ell, \mathbf{w}_r) = f(p; \mathbf{w}_\ell) + f(p; \mathbf{w}_r) + u_r - u_\ell, \quad (45)$$

where f is given by

$$f(p; \mathbf{w}_{\ell/r}) = \begin{cases} (p - p_{\ell/r}) \left[\frac{C_{\ell/r}}{p + B_{\ell/r}} \right]^{\frac{1}{2}} & \text{if } p > p_{\ell/r} \text{ (shock)}, \\ \frac{2C_{\ell/r}}{(\gamma-1)} \left[\left(\frac{p}{p_{\ell/r}} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] & \text{if } p \leq p_{\ell/r} \text{ (rarefaction)}, \end{cases} \quad (46)$$

and $C_{\ell/r}, B_{\ell/r}$ are given by

$$C_{\ell/r} = \frac{2}{(\gamma+1)\rho_{\ell/r}}, \quad B_{\ell/r} = \frac{(\gamma-1)}{(\gamma+1)} p_{\ell/r}. \quad (47)$$

Let us assume that we already have found p_* , and therefore the wave structure for our problem. Then, we have closed-form solutions for $u_*, \rho_{*\ell}$, and ρ_{*r} . First, we compute the intermediate velocity

$$u_* = \frac{1}{2} (u_\ell + u_r) + \frac{1}{2} [f(p_*; \mathbf{w}_\ell) + f(p_*; \mathbf{w}_r)]. \quad (48)$$

The values of $\rho_{*\ell}$ and ρ_{*r} will be also determined by the nature of the 1- and 3- waves. If the 1-wave or the 3-wave are shocks, then

$$\rho_{*\ell} = \rho_\ell \left[\frac{\frac{p_*}{p_L} + \frac{\gamma-1}{\gamma+1}}{\frac{\gamma-1}{\gamma+1} \frac{p_*}{p_\ell} + 1} \right] \quad \text{or} \quad \rho_{*r} = \rho_r \left[\frac{\frac{p_*}{p_r} + \frac{\gamma-1}{\gamma+1}}{\frac{\gamma-1}{\gamma+1} \frac{p_*}{p_r} + 1} \right], \quad (49)$$

and if the 1-wave or the 3-wave are rarefactions, then

$$\rho_{*\ell} = \rho_L \left(\frac{p_*}{p_L} \right)^{\frac{1}{\gamma}} \quad \text{and} \quad \rho_{*r} = \rho_R \left(\frac{p_*}{p_R} \right)^{\frac{1}{\gamma}}, \quad (50)$$

respectively.

For an in-depth derivation of the Euler equations and its solution strategy, we refer the reader to, e.g. [25, 12]. An excellent review of exact solver algorithms can be found in the work of Gottlieb & Groth [4].

Since the solution of the Riemann problem (41) can be explicitly computed once p_* is computed, our goal is to determine the root of the pressure function (45) in a computationally efficient way. We remark that in the presence of a vacuum state, i.e. p_ℓ , p_r , or p_* equal to zero, the complete exact solution for the Riemann problem can be obtained explicitly, and therefore we do not consider problems with this behavior here. However, for practical purposes, an implementation of the vacuum-state treatment presented in [25, Sec. 4.6] is available in the reproducibility repository [15].

3.1.1 Properties of the pressure function

The following general properties of the pressure function, which behaves similarly to the depth function for the SWEs, are useful in designing algorithms for its solution. We have

$$f'(p; \mathbf{w}_{\ell/r}) = \begin{cases} \left(\frac{C_{\ell/r}}{B_{\ell/r} + p} \right)^{1/2} \left[1 - \frac{p - p_{\ell/r}}{2(B_{\ell/r} + p)} \right] & \text{if } p > p_{\ell/r} \text{ (shock),} \\ \frac{1}{\rho_{\ell/r} a_{\ell/r}} \left(\frac{p}{p_{\ell/r}} \right)^{-(\gamma+1)/2\gamma} & \text{if } p \leq p_{\ell/r} \text{ (rarefaction),} \end{cases} \quad (51)$$

and

$$f''(p; \mathbf{w}_{\ell/r}) = \begin{cases} -\frac{1}{4} \left(\frac{C_{\ell/r}}{B_{\ell/r} + p} \right)^{1/2} \left[\frac{4B_{\ell/r} + 3p + p_{\ell/r}}{(B_{\ell/r} + p)^2} \right] & \text{if } p > p_{\ell/r} \text{ (shock),} \\ -\frac{(\gamma+1)a_{\ell/r}}{2\gamma^2 p_{\ell/r}^2} \left(\frac{p}{p_{\ell/r}} \right)^{-(3\gamma+1)/2\gamma} & \text{if } p \leq p_{\ell/r} \text{ (rarefaction).} \end{cases} \quad (52)$$

It is straightforward to see that $\phi'(p) = f'(p, \mathbf{w}_\ell) + f'(p, \mathbf{w}_r) > 0$ and $\phi''(p) = f''(p, \mathbf{w}_\ell) + f''(p, \mathbf{w}_r) < 0$, therefore ϕ is monotonically increasing and concave down. Therefore, it follows that $\phi(p_{\min}) > 0$ implies that both waves are rarefactions, and $\phi(p_{\max}) < 0$ implies that both waves are shocks, where $p_{\min} = \min(p_\ell, p_r)$ and $p_{\max} = \max(p_\ell, p_r)$.

3.2 Computation of the initial guess

In order to find the root of ϕ exactly we follow an iterative process. We will proceed as in Section 2.3. We consider a total of six initial guesses, three of which we describe in the following sections. The other three (AV, CC, and HLLC) are the natural analogs of the SW initial guesses described in Sections 2.3.2, 2.3.6 and 2.3.7.

3.2.1 Two-rarefaction approximation (RR)

If the 1- and 3- waves are rarefactions, then

$$\phi(p; \mathbf{w}_\ell, \mathbf{w}_r) = \phi_{\text{RR}}(p) := \frac{2C_\ell}{(\gamma-1)} \left[\left(\frac{p}{p_\ell} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] + \frac{2C_r}{(\gamma-1)} \left[\left(\frac{p}{p_r} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right], \quad (53)$$

and its root is given explicitly by

$$p_{\text{RR}} := \left[\frac{a_\ell + a_r - \frac{1}{2}(\gamma - 1)(u_r - u_\ell)}{a_\ell/p_\ell^{\frac{\gamma-1}{2\gamma}} + a_r/p_r^{\frac{\gamma-1}{2\gamma}}} \right]^{\frac{2\gamma}{\gamma-1}}. \quad (54)$$

For problems whose solution does not include two rarefactions, this value can be used as an initial guess.

3.2.2 Primitive-variable linearized solver (PV)

Any approximate Riemann solver can be used to provide an initial guess for an iterative solver. Here we consider the use of the linearized solver introduced by Toro [22]. Linearizing the system around the state $\hat{\mathbf{w}}$ in primitive variables gives the linearized system (14). Choosing $\hat{\rho}$ and \hat{a} as the arithmetic mean between the left and right values, we get the following solution:

$$p_* = \frac{1}{2}(p_\ell + p_r) - \frac{1}{8}(u_r - u_\ell)(\rho_\ell + \rho_r)(a_\ell + a_r).$$

In case this value is negative, it is common practice to take $p_* = \epsilon \ll 1$. However, since we know that (except in the case of two rarefactions) $\min(p_\ell, p_r)$ is a positive lower bound for p_* , we set

$$p^{[0]} = p_{\text{PV}} := \max\left(\min(p_\ell, p_r), \frac{1}{2}(p_\ell + p_r) - \frac{1}{8}(u_r - u_\ell)(\rho_\ell + \rho_r)(a_\ell + a_r)\right). \quad (55)$$

3.2.3 Two-shock approximation (SS)

Next we consider another initial guess based on an approximate solver that was proposed by Toro [23]. If both waves are shocks, then the pressure function becomes

$$\phi_{\text{SS}}(p) := (p - p_\ell) g_\ell(p) + (p - p_r) g_r(p) + u_r - u_\ell, \quad (56)$$

where

$$g_{\ell/r}(p) = \left[\frac{C_{\ell/r}}{p + B_{\ell/r}} \right]^{1/2}. \quad (57)$$

This can be solved explicitly if we choose an approximation for the argument of $g_{\ell/r}$. We take the approximate value p_{PV} , yielding

$$\phi_{\text{SS}}(p) = (p - p_\ell) g_\ell(p_{\text{PV}}) + (p - p_r) g_r(p_{\text{PV}}) + u_r - u_\ell, \quad (58)$$

from which we get the initial guess

$$p^{[0]} = p_{\text{SS}} := \frac{g_\ell(p_{\text{PV}}) p_\ell + g_r(p_{\text{PV}}) p_r - u_r + u_\ell}{g_\ell(p_{\text{PV}}) + g_r(p_{\text{PV}})}. \quad (59)$$

3.3 Iterative algorithms

For the Euler equations, we will test all of the root-finding iterative algorithms introduced in Section 2.4, along with two more described below. These additional solvers were studied by Gottlieb & Groth [4] and shown to perform better than methods introduced before that. Besides the algorithms discussed here, we also implemented and tested Pike's exact Riemann solver for the Euler equations [17]. However, we were not able to reproduce the results from [17] or to obtain competitive results with that solver, so we omit it here.

3.3.1 Gottlieb & Groth's method (GG)

A novel approach to compute the intermediate state of a Riemann problem exactly was introduced by Gottlieb & Groth [4]. It is a Newton iteration in which the intermediate state velocity (rather than the pressure) is updated directly:

$$u^{[k+1]} = u^{[k]} - \frac{p_\ell^*(u^{[k]}) - p_r^*(u^{[k]})}{p_\ell^{*'}(u^{[k]}) - p_r^{*'}(u^{[k]})}. \quad (60)$$

Here the values of p_ℓ^* , $p_\ell^{*'}$, p_r^* , $p_r^{*'}$ depend on the nature of the 1- and 3- waves given that $u^{[k]}$ is the middle state velocity. For instance, if there is a shock (with $u^{[k]} \leq u_{\ell/r}$),

$$p_{\ell/r}^*(u^{[k]}) = p_{\ell/r} + C_{\ell/r}(u^{[k]} - u_{\ell/r})W_{\ell/r}, \quad (61)$$

$$p_{\ell/r}^{*'}(u^{[k]}) = \frac{2C_{\ell/r}W_{\ell/r}^3}{1 + W_{\ell/r}^2}, \quad (62)$$

$$W_{\ell/r} = \frac{\gamma + 1}{4} \frac{u^{[k]} - u_\ell}{a_{\ell/r}} + \mp \left[1 + \left[\frac{\gamma + 1}{4} \frac{u^{[k]} - u_{\ell/r}}{a_{\ell/r}} \right]^2 \right]^{1/2}. \quad (63)$$

Meanwhile, if there is a rarefaction (with $u^{[k]} > u_{\ell/r}$), we have

$$a_{\ell/r}^* = a_{\ell/r} + \mp \frac{\gamma - 1}{2} (u^{[k]} - u_{\ell/r}), \quad (64)$$

$$p_{\ell/r}^*(u^{[k]}) = p_{\ell/r} \left(\frac{a_{\ell/r}^*}{a_{\ell/r}} \right)^{2\gamma/(\gamma-1)}, \quad (65)$$

$$p_{\ell/r}^{*'}(u^{[k]}) = \mp \gamma p_{\ell/r}^* / a_{\ell/r}^*, \quad (66)$$

where for each \mp we take the minus sign for ℓ and the plus sign for r . The authors use a two-rarefaction approximation as initial guess for this method. Since the iterations are performed on the velocity, the initial guess takes the form

$$u^{[0]} = \frac{(u_\ell + z_1 a_\ell)z_2 + (u_r - z_1 a_r)}{1 + z_2}, \quad (67)$$

where

$$z_1 = \frac{2}{\gamma - 1}, \quad z_2 = \frac{a_r}{a_\ell} \left(\frac{p_\ell}{p_r} \right)^{\frac{\gamma-1}{2\gamma}}. \quad (68)$$

For this method, we employ a different convergence criterion, following [4]. The iteration continues until $|(p_r^* - p_\ell^*)/p_r^*| < \epsilon$, for a specified tolerance ϵ . This method takes advantage of the fact that the sound speed a appears more often than the density ρ in computational simulations of perfect gases. The goal is not only to compute the constant primitive variables of the intermediate state p^* and u^* , but also to provide an estimate of $\rho_{*\ell}$ and ρ_{*r} through $a_{*\ell}$ and a_{*r} in order to completely characterize the waves in the Riemann solution.

3.3.2 van Leer's method (vL)

van Leer introduced a modification of Godunov's second formulation [28, 4] to estimate p_* . This method consists of reducing the difference between two estimates u_ℓ^* and u_r^* for the intermediate velocity. The functions $u_{\ell/r}^*(\cdot)$ are determined by the wave jump equations and can be represented by

$$u_{\ell/r}^*(p^{[k]}) = u_{\ell/r} \mp \frac{p^{[k]} - p_\ell}{A_{\ell/r}(p^{[k]})}, \quad (69)$$

and their derivatives by

$$u_{\ell/r}^{*'}(p^{[k]}) = \begin{cases} \mp(A_{\ell/r}^2 + \gamma p_k \rho_k)/(2A_{\ell/r}^3) & p^{[k]} \geq p_{\ell/r}, \\ \mp(\gamma p_{\ell/r} \rho_{\ell/r})^{-1/2} \left[\frac{p^{[k]}}{p_{\ell/r}} \right]^{-(\gamma+1)/(2\gamma)} & p^{[k]} < p_{\ell/r}. \end{cases} \quad (70)$$

where again for each \mp we take the minus sign for ℓ and the plus sign for r , and

$$A_{\ell/r} = \begin{cases} (\gamma p_{\ell/r} \rho_{\ell/r})^{1/2} \left[\frac{\gamma+1}{2\gamma} \frac{p^{[k]}}{p_{\ell/r}} + \frac{\gamma-1}{2\gamma} \right]^{1/2} & p^{[k]} \geq p_{\ell/r}, \\ \frac{\gamma-1}{2\gamma} (\gamma p_{\ell/r} \rho_{\ell/r})^{1/2} \frac{1 - p^{[k]}/p_{\ell/r}}{1 - (p^{[k]}/p_{\ell/r})^{(\gamma-1)/(2\gamma)}} & p^{[k]} < p_{\ell/r}. \end{cases} \quad (71)$$

The pressure approximation is updated via

$$p^{[k+1]} = p^{[k]} - \frac{u_\ell^*(p^{[k]}) - u_r^*(p^{[k]})}{u_\ell^{*'}(p^{[k]}) - u_r^{*'}(p^{[k]})}. \quad (72)$$

For this method, we use the convergence criterion $|(u_r^* - u_\ell^*)/u_r^*| < \epsilon$.

3.4 Performance comparison

For the performance comparison we proceed analogously to the shallow water equations. We consider 10^7 Riemann problems generated pseudo-randomly, with 20% of them involving strong waves and 80% involving only weak waves. For the problems with strong waves we take each of the initial pressures as $p_\ell, p_r = 10^k$ with $k \sim \mathcal{U}([-4, 4])$; the particle velocities as $u_\ell = 10^k$ and $u_r = -10^k$ with $k \sim \mathcal{U}([-2, 2])$; the densities are taken as $\rho_\ell, \rho_r \sim \mathcal{U}([0.01, 0.9])$. For the problems with weak waves we use $p_\ell, p_r \sim \mathcal{U}([0.1, 1])$, $u_\ell = u_r = 0$, and $\rho_\ell, \rho_r \sim \mathcal{U}([0.1, 0.9])$.

Again, in our implementation we always check first for a two-rarefaction solution and use the exact solution (54) in that case. Therefore, in this section we consider only Riemann problems whose solution includes one or more shocks. This is guaranteed for the strong wave problems since the fluid on both sides is initially flowing toward the origin. The initial conditions for the weak wave problems were set to resemble the behavior of the Sod shock tube problem [20], where the exact Riemann solution consists of a shock, a contact discontinuity, and a rarefaction. Once the two-rarefaction scenario has been discarded, we have the bound $p_{\min} \leq p_*$ and the modification of Newton's method (26) can be used to avoid non-positive pressures.

The solvers of Gottlieb & Groth and of van Leer were both originally developed to allow for different ratios of specific heats on the left and right (see [4], [28]). Notice that both of these solvers compute u_* in the iterative process to obtain p_* . Therefore, to allow a fair timing comparison, u_* is computed along with p_* in every solver.

Initial guess	Time (s)	Iterations	Average relative initial error	
			Weak waves	Strong waves
AV (average pressure) (12)	2.847	3.6	7.05%	51%
RR (two-rarefaction) (54)	2.893	2.5	0.13%	193%
PV (primitive variables linearized) (55)	2.832	3.5	7.05%	33%
SS (two-shock) (59)	2.247	2.3	0.11%	25%
CC (convex combination) (22)	3.317	2.1	0.04%	28%
HLLE (23)	2.957	3.3	4.35%	20%

Table 3: Comparison of execution time of Newton’s iterative solver with different initial guesses and their average relative initial error the Euler equations with different initial guesses

3.4.1 Comparison of initial guesses

We compare the initial guesses by applying each in combination with Newton’s method, as is done in Section 2.6. The results are presented in Table 3. The two-rarefaction (RR) and convex combination (CC) approximations require fewer iterations on average. However, the greater cost of the CC approximation itself outweighs the advantage obtained from the reduced number of iterations required, yielding the slowest method overall. Again we conclude that the two-shock approximation is the best initial guess available for an iterative scheme (assuming that one always checks first for the two-rarefaction case).

In Appendix A we define an adaptive noniterative Riemann solver based on the CC approximation, similar to that proposed for the SWEs. This approximation can be used as a solver for problems with weak waves, and as the initial guess for an iterative algorithm for problems with strong waves. Even though for weak wave problems this solver does not provide an exact solution, the accuracy attained with our ensemble of test problems is competitive (0.04% average relative error). Furthermore, an improvement in execution time of approximately 25% was obtained with respect to the fastest fully iterative scheme considered in this work.

3.4.2 Comparison of root-finding algorithms

Next we assess the performance of each iterative scheme. For most of the schemes, we use the SS (two-shock) approximation, based on its performance in the comparison above. As upper and lower bounds of p_* for the bounding polynomial iterative scheme we use p_{RR} and p_{min} , respectively. Given its previously-demonstrated superior performance [4], we use the RR (two rarefaction) approximation for the flow velocity as starting guess for the GG solver.

The results for two different tolerances are shown in Table 4. For $\epsilon = 10^{-6}$ we obtain again that the positive Newton and two-step Newton methods converge faster (in wall clock time) than the other iterative schemes. The bounding polynomial-based iterative scheme converges in the fewest iterations, but is drastically slower due to the cost per iteration.

For $\epsilon = 10^{-12}$, van Leer’s method and the single linear polynomial-based method fail to converge for some problems, either because a negative pressure is generated or a fixed maximum number of iterations is exceeded. Also, in contrast with the results obtained for the shallow water

Method \ Tolerance	$\epsilon = 10^{-6}$		$\epsilon = 10^{-12}$	
	Time	Iterations	Time	Iterations
Positive Newton-SS	1.654 s	1.5	2.258 s	2.3
Ostrowski-Newton-SS	1.838 s	1.1	2.003 s	1.4
Ostrowski-SS	1.762 s	1.0	2.061 s	1.3
GG-RR	2.860 s	2.4	3.702 s	3.3
Two-step Newton-SS	1.670 s	1.5	2.232 s	2.2
van Leer-SS	1.973 s	2.5	-	-
Bounding quadratic polynomials-RR	4.898 s	1.0	5.541 s	1.2
Single quadratic polynomial-RR	3.203 s	1.1	3.753 s	1.5
Single linear polynomial-RR	-	-	-	-

Table 4: Comparison of execution time for different iterative Riemann Solvers for Euler Equations with their best performing initial guesses

equations, we have that Ostrowski’s and Ostrowski-Newton’s methods are faster than positive Newton’s method at this tighter tolerance. This suggests that as the function ϕ increases in computational complexity and more accuracy is needed, higher order methods may become more efficient. Note that Ostrowski-Newton’s method preserves the robustness of the positive Newton’s method with the fix (26). This property relies on the way the Fortran compiler handles undefined numbers if the first Ostrowski half-step (31) returns a non-positive approximation; therefore, care must be taken in its implementation.

4 Finite volume simulations

Riemann solvers are most widely used as a building block in numerical simulations, and we now compare the performance of the iterative solvers studied in this work with that of the most commonly-used approximate solvers. We use the wave-propagation approach implemented in Clawpack [10, 13]. The Riemann solvers compute the full characteristic structure of the Riemann solution in order to obtain the value along the cell interface. We test the solvers in combination with the first-order Godunov method and then with the standard second-order Clawpack (Lax-Wendroff-Leveque) approach.

We first consider a shallow water shock-wave interaction problem inspired by the Woodward-Collela blast wave problem, with reflecting boundary conditions at $x = \pm 5$ and the following initial data:

$$h(x) = \begin{cases} 30, & \text{if } x \leq -2, \\ 1, & \text{if } -2 < x < 2, \\ 50, & \text{if } 2 \leq x, \end{cases} \quad hu(x) = 0. \quad (73)$$

Grid	L^2 error for h (%)			
	50	150	450	1350
Iterative	2.07	1.61	1.15	0.59
HLLE	1.96	1.48	1.17	0.54
Roe	1.97	1.52	1.19	0.55

Table 5: Accuracy of Godunov’s method for the shallow water equations with different Riemann solvers

Grid	L^2 error for ρ (%)			
	50	150	450	1350
Iterative	42.07	34.96	20.24	9.64
HLLE	42.84	31.86	23.51	12.3
Roe	41.8	34.84	20.22	9.65

Table 6: Accuracy of Godunov’s method for the Euler equations with different Riemann solvers

We compare the accuracy of the solution obtained with the Ostrowski-Newton iterative solver with the Roe and HLLE approximate solvers, at final time $T = 10$ on a range of mesh sizes, as shown in Table 5. For this we use a self-convergence test with the solution given by each solver in a grid of 4050 cells as exact solution, and the errors are measured using the corresponding standard grid function norms. The execution time of each simulation is measured with the same final time and a grid of 4050 volume cells. An important factor for accuracy and timing measurement is the choice of the time step, which is chosen to yield a CFL number of approximately 0.9 at each step. The results can be seen in Table 7, where a grid function norm is used to compute the relative L^2 error in h .

For the Euler equations we use the positive Newton iterative solver. We take the domain $[0, 1]$, wall boundary conditions, and the initial conditions

$$\rho(x) = 0.1, \quad \rho u(x) = 0, \quad E(x) = \begin{cases} 1000/(\gamma - 1), & \text{if } x < 0.1, \\ 1, & \text{if } 0.1 \leq x \leq 0.9, \\ 100/(\gamma - 1), & \text{if } 0.9 < x. \end{cases} \quad (74)$$

Accuracy and execution time are measured analogously to the shallow water equations with a final time $T = 0.5$, and the errors being computed in ρ . The results are shown in Tables 6 and 8.

We obtain similar results for both the shallow water and Euler equations. For both problems, all three Riemann solvers give similar accuracy. We see, on the other hand, that with Godunov’s method the approximate solvers are faster than the iterative solver by a factor of 2 for the shallow water equations and for a factor of 3 for the Euler equations. These relative differences are somewhat smaller when a second order scheme is used. For higher-order methods, with more costly additional operations such as WENO reconstruction, ADER schemes, or for multiphysics simulations, the relative increase in cost will be further diminished.

5 Conclusion

The new positivity-ensuring modifications of iterative solvers presented here, combined with accurate lower bounds for the depth and pressure, yield iterative solvers that are as robust as any approximate Riemann solver and are guaranteed to give physically-admissible outputs. Unsurprisingly, in terms of speed the best iterative methods still lag behind approximate solvers, but by a smaller margin (about a factor of two) than what has been reported in the past. This may be due to relative shifts in the speed of flops compared to memory accesses over recent decades.

Riemann Solver	Execution time	
	First order	Second order
Iterative	4.4 s	4.9 s
HLLE	2.3 s	3.3 s
Roe	2.2 s	3.3 s

Table 7: Execution time of finite volume simulations with different Riemann solvers for the shallow water equations

Riemann Solver	Execution time	
	First order	Second order
Iterative	11.0 s	12.3 s
HLLE	3.3 s	5.0 s
Roe	3.6 s	5.2 s

Table 8: Execution time of finite volume simulations with different Riemann solvers for the Euler equations

The new cubically-convergent (bounding quadratic polynomials) iteration succeeds in reducing the number of iterations required for solution of the Euler Riemann problem and can be coupled with an absolute error-based convergence criterion, but is overall inefficient due to the cost per iteration. Its simplified versions, relying on the computation of a single interpolating polynomial, are somewhat more competitive in terms of efficiency, but also less robust. Furthermore, they are not well-suited for an absolute error-based termination criterion, as the lower bound is fixed from the beginning of the iterative process. Among the iterative methods tested, we recommend the positive Newton method (proposed herein) as a simple, robust, and near-best-efficiency solver.

Acknowledgements

We thank the anonymous referees for their helpful comments, one of which led to the introduction and testing of the single quadratic polynomial method.

A Approximate state Riemann solvers

Given the high accuracy obtained with the CC and QA initial guesses for the SWE Riemann problem, a new non-iterative approximate state Riemann solver is proposed in Algorithm 1. We take advantage of the fact that the criterion for determining the nature of the waves can be reused to compute the CC approximation. To also obtain an approximation of u_* , (6) can still be used.

Similarly, we also propose an adaptive solver for the EEs in 2. However, as discussed in Section 3.4, the CC approximation of p_* might not be accurate enough when the Riemann solution involves strong waves. Therefore, this proposed adaptive solver employs iteration for two-shock solutions.

References

- [1] Pascal Azerad, Jean-Luc Guermond, and Bojan Popov. Well-balanced second-order approximation of the shallow water equation with continuous finite elements. *SIAM Journal on Numerical Analysis*, 55(6):3203–3224, 2017.
- [2] Bernd Einfeldt. On Godunov-type methods for gas dynamics. *SIAM Journal on numerical analysis*, 25(2):294–318, 1988.
- [3] James Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Communications on pure and applied mathematics*, 18(4):697–715, 1965.

Algorithm 1 Adaptive non-iterative Riemann solver for the shallow water equations

Require: $\mathbf{q}_\ell, \mathbf{q}_r$ **Ensure:** \tilde{h}_* ▷ Approximation of h_*
1: $h_{\min} \leftarrow \min(h_\ell, h_r)$
2: **if** $\phi(h_{\min}) > 0$ **then** ▷ Check if solution has two rarefactions
3: $\tilde{h}_* \leftarrow h_{RR}$ ▷ Exact solution
4: **else**
5: $h_{\max} \leftarrow \max(h_\ell, h_r)$
6: **if** $\phi(h_{\max}) < 0$ **then** ▷ Two shocks (Strong waves)
7: $\tilde{h}_* \leftarrow h_{QA}$ ▷ Quadratic approximation
8: **else** ▷ One rarefaction and one shock (Weak waves)
9: $\tilde{h}_* \leftarrow h_{CC}$ ▷ Convex combination
10: **end if**
11: **end if**

Algorithm 2 Adaptive iterative Riemann solver for the Euler equations

Require: $\mathbf{q}_\ell, \mathbf{q}_r$ **Ensure:** \tilde{p}_* ▷ Approximation of p_*
1: $p_{\min} \leftarrow \min(p_\ell, p_r)$
2: **if** $\phi(p_{\min}) > 0$ **then** ▷ Check if solution has two rarefactions
3: $\tilde{p}_* \leftarrow p_{RR}$ ▷ Exact solution
4: **else**
5: $p_{\max} \leftarrow \max(p_\ell, p_r)$
6: Compute $\phi(p_{\max})$
7: $p^{[0]} \leftarrow p_{CC}$
8: **if** $\phi(p_{\max}) > 0$ **then** ▷ One rarefaction and one shock (Weak waves)
9: $\tilde{p}_* \leftarrow p^{[0]}$ ▷ Use convex combination as solution
10: **else** ▷ Two shocks (Strong waves)
11: $p^{[0]} \leftarrow \max(p_{\max}, p^{[0]} - \phi(p^{[0]})/\phi'(p^{[0]}))$ ▷ More accurate Positive Newton modification
12: $\tilde{p}_* \leftarrow \text{Newton}(p^{[0]})$ ▷ Newton iterations
13: **end if**
14: **end if**

- [4] JJ Gottlieb and Clinton PT Groth. Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. *Journal of Computational Physics*, 78(2):437–458, 1988.
- [5] Jean-Luc Guermond and Bojan Popov. Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations. *Journal of Computational Physics*, 321:908–926, 2016.
- [6] Vitaliy Gyrya, M Shashkov, A Skurikhin, and Svetlana Tokareva. Machine learning approaches for the solution of the Riemann problem in fluid dynamics: a case study. *preprint*, 2020.
- [7] Christiane Helzel, David Kerkmann, and Leonardo Scandurra. A new ADER method inspired by the active flux method. *Journal of Scientific Computing*, 80(3):1463–1497, 2019.
- [8] Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.
- [9] David I Ketcheson, Randall J LeVeque, and Mauricio J Del Razo. *Riemann problems and Jupyter solutions*, volume 16. SIAM, 2020.
- [10] David I Ketcheson, Kyle Mandli, Aron J Ahmadi, Amal Alghamdi, Manuel Quezada de Luna, Matteo Parsani, Matthew G Knepley, and Matthew Emmett. Pyclaw: Accessible, extensible, scalable tools for wave propagation problems. *SIAM Journal on Scientific Computing*, 34(4):C210–C231, 2012.
- [11] Richard F King. A family of fourth order methods for nonlinear equations. *SIAM journal on numerical analysis*, 10(5):876–879, 1973.
- [12] Randall J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, 2002.
- [13] Kyle T Mandli, Aron J Ahmadi, Marsha Berger, Donna Calhoun, David L George, Yianis Hadjimichael, David I Ketcheson, Grady I Lemoine, and Randall J LeVeque. Clawpack: building an open source ecosystem for solving hyperbolic pdes. *PeerJ Computer Science*, 2:e68, 2016.
- [14] Trevor J McDougall and Simon J Wotherspoon. A simple modification of Newton’s method to achieve convergence of order $1 + 2$. *Applied Mathematics Letters*, 29:20–25, 2014.
- [15] Carlos Muñoz Moncayo, Manuel Quezada de Luna, and David I. Ketcheson. Reproducibility repository for the paper ”A Comparative Study of Iterative Riemann Solvers for the Shallow Water and Euler Equations”. <https://doi.org/10.5281/zenodo.8335057>, 2023.
- [16] Miodrag S Petković, Beny Neta, Ljiljana D Petković, and Jovana Džunić. Multipoint methods for solving nonlinear equations: A survey. *Applied Mathematics and Computation*, 226:635–660, 2014.
- [17] Jack Pike. Riemann solvers for perfect and near-perfect gases. *AIAA journal*, 31(10):1801–1808, 1993.
- [18] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.
- [19] James J Quirk. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers & fluids*, 23(1):125–142, 1994.
- [20] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of computational physics*, 27(1):1–31, 1978.

- [21] Vladimir A Titarev and Eleuterio F Toro. ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17(1):609–618, 2002.
- [22] Eleuterio F. Toro. A linearized Riemann solver for the time-dependent Euler equations of gas dynamics. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1892):683–693, 1991.
- [23] Eleuterio F. Toro. Direct Riemann solvers for the time-dependent Euler equations. *Shock Waves*, 5(1):75–80, 1995.
- [24] Eleuterio F. Toro. *Shock-capturing methods for free-surface shallow flows*. John Wiley, 2001.
- [25] Eleuterio F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.
- [26] Eleuterio F. Toro. The ADER path to high-order Godunov methods. In *Continuum Mechanics, Applied Mathematics and Scientific Computing: Godunov’s Legacy*, pages 359–366. Springer, 2020.
- [27] Eleuterio F Toro and Cheng-Chiang Chou. A linearized Riemann solver for the steady supersonic Euler equations. *International journal for numerical methods in fluids*, 16(3):173–186, 1993.
- [28] Bram Van Leer. Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov’s method. *Journal of computational Physics*, 32(1):101–136, 1979.