Quasi-Lagrangian Acceleration of Eulerian Methods

Igor Kliakhandler[1] and Alexander Kurganov[2*]

[1] *Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, USA*
[2] *Mathematics Department, Tulane University, New Orleans, LA 70118, USA*

Abstract. We present a simple and efficient strategy for the acceleration of explicit Eulerian methods for multidimensional hyperbolic systems of conservation laws. The strategy is based on the Galilean invariance of dynamic equations and optimization of the reference frame, in which the equations are numerically solved. The optimal reference frame moves (locally in time) with the average characteristic speed of the system, and, in this sense, the resulting method is quasi-Lagrangian. This leads to the acceleration of the numerical computations thanks to the optimal CFL condition and automatic adjustment of the computational domain to the evolving part of the solution. We show that our quasi-Lagrangian acceleration procedure may also reduce the numerical dissipation of the underlying Eulerian method. This leads to a significantly enhanced resolution, especially in the supersonic case. We demonstrate a great potential of the proposed method on a number of numerical examples.

AMS subject classifications: 76M12, 65B99, 35L65

Key words: Eulerian methods, quasi-Lagrangian method, finite volume methods, Galilean invariance.

1 Introduction

We study numerical methods for hyperbolic systems of conservation laws, which, in the one-dimensional (1-D) case, read:

$$\mathbf{u}_t+\mathbf{f}(\mathbf{u})_x=\mathbf{0}, \tag{1.1}$$

where $\mathbf{u}(x,t):=(u^{(1)}(x,t),u^{(2)}(x,t),\dots,u^{(N)}(x,t))^T$ is an N-dimensional vector of unknowns and $\mathbf{f}(\mathbf{u}(x,t)):=(f^{(1)}(\mathbf{u}(x,t)),f^{(2)}(\mathbf{u}(x,t)),\dots,f^{(N)}(\mathbf{u}(x,t)))^T$ is the flux function. We restrict our consideration to initial value problems (IVP) and initial-boundary value problems with periodic boundary conditions.

*Corresponding author. *Email addresses:* igor@mtu.edu (I. Kliakhandler), kurganov@math.tulane.edu (A. Kurganov)

There are two big classes of the numerical methods for (1.1): Eulerian and Lagrangian ones. The main advantage of Eulerian methods is that they employ stationary spatial grids, which makes numerical flux approximation relatively easy. In the Lagrangian framework, the grid moves together with the medium, which typically leads to a very high resolution of contact waves, but makes the computation of numerical fluxes substantially more involved.

Another drawback of Lagrangian methods is a lack of control of the developed grid structure: the grid, which is moving with the fluid, may become highly irregular. This would affect both the efficiency of the method and its accuracy. One of the ways to overcome this difficulty is to use ALE methods (see, e.g., [1,16,19] and references therein), in which the computed solution is projected onto the regular grid after each time step or after every few time steps so that one makes sure that the mesh does not become highly irregular. Another way of fixing the problem of irregular grid formation while enjoying the main advantage of Lagrangian methods—automatic adaptivity of the grid to the structure of the computed solution—is to use the adaptive moving mesh methods (see, e.g., [6,7,15,22] and references therein). In these methods, the mesh is moving not with the fluid, but according to a moving mesh PDE, [6,7,22], or the estimated local errors in the computed solution, [15].

In this paper, we only study Eulerian methods and focus on two specific issues: their efficiency and numerical dissipation. It is well-known that the CFL condition, related to the spectral radius of the Jacobian $\partial\mathbf{f}/\partial\mathbf{u}$, is a fundamental stability restriction on the size of time steps in Eulerian methods. We propose a general strategy for reduction of the CFL number for any given Eulerian method. The main idea is to use the Galilean invariance of the system (1.1), which allows one to choose, at each time step, the reference frame with the least restrictive CFL condition. The entire mesh is then shifted to stay in the selected frame. Notice, however, that unlike the case of Lagrangian or moving mesh methods, the structure of the mesh does not change at all by the proposed mesh shift.

In a nutshell, the strategy works as follows. At each time step, we add a linear advection term $-\sigma\mathbf{u}_x$ to the left-hand side of (1.1) and solve the resulting system

$$\mathbf{u}_t+\mathbf{f}(\mathbf{u})_x-\sigma\mathbf{u}_x=\mathbf{0}, \tag{1.2}$$

where σ is a constant. Obviously, solutions of (1.2) are obtained from the corresponding solutions of (1.1) by the change of variables $x\to x-\sigma t$. However, the constant σ provides us with an additional degree of freedom, and a wise choice of σ may help to improve both efficiency and resolution, achieved by the numerical method applied to (1.2) instead of (1.1). Our approach can be viewed as quasi-Lagrangian since σ is chosen so that the reference frame moves at the average characteristic velocity, as quantified in §2. We note that the proposed method is not a moving mesh method, but rather a "moving framework" one. It resembles a more sophisticated hybrid Eulerian-Lagrangian method from [24]. However, unlike the method from [24], our approach retains the simplicity of Eulerian methods. We would also like to mention that adding the linear convection term $-\sigma\mathbf{u}_x$ as it is done in (1.2) resembles the artificial wind method from [21]. However, we add the

linear convection term to make the largest and the smallest eigenvalues of the Jacobian to be different in sign but close by the amplitude to improve efficiency and reduce numerical dissipation, while in [21] the linear convection term is added to make all eigenvalues to be of the same sign to simplify upwinding.

The proposed quasi-Lagrangian strategy can be straightforwardly extended to any number of space dimensions. The efficiency improvement as well as the enhanced resolution is demonstrated via the numerical viscosity analysis in §3 and on a number of one- and two-dimensional (2-D) examples in §4.

2 Description of the Method

We describe our quasi-Lagrangian approach on an example of finite-volume methods for the 1-D system (1.1). In the finite-volume setting, the computed quantities at time level $t=t^n$ are solution cell averages:

$$\overline{\mathbf{u}}_j^n \approx \int\limits_{x^n_{j-\frac{1}{2}}}^{x^n_{j+\frac{1}{2}}} \mathbf{u}(x,t^n)\,dx.$$

Here, for the sake of simplicity, we consider a uniform spatial grid with $x^n_{j+\frac{1}{2}}-x^n_{j-\frac{1}{2}}\equiv\Delta x$. The numerical solution is evolved to the next time level $t^{n+1}:=t^n+\Delta t^n$ according to the scheme, which can be typically written in the flux form:

$$\overline{\mathbf{u}}_j^{n+1}=\overline{\mathbf{u}}_j^n-\lambda^n\left[\mathbf{H}_{j+\frac{1}{2}}-\mathbf{H}_{j-\frac{1}{2}}\right],\qquad \lambda^n:=\frac{\Delta t^n}{\Delta x}.$$

Here, $\mathbf{H}_{j+\frac{1}{2}}:=\mathbf{H}_{j+\frac{1}{2}}(\ldots,\overline{\mathbf{u}}_j^n,\overline{\mathbf{u}}_{j+1}^n,\ldots)$ are numerical fluxes, which are supposed to be consistent ($\mathbf{H}_{j+\frac{1}{2}}(\ldots,\mathbf{u},\mathbf{u},\ldots)=\mathbf{f}(\mathbf{u})$) and sufficiently accurate. A high order of accuracy can be achieved in several different ways (see, e.g., [2, 3, 8, 14, 23]). We will focus on Godunov-type approach, in which the computed solution is globally approximated by a piecewise polynomial interpolant

$$\widetilde{u}^n(x)=\mathcal{P}_j^n(x)\ \text{ for } x^n_{j-\frac{1}{2}}<x<x^n_{j+\frac{1}{2}},$$

where $\mathcal{P}_j^n$ are polynomials. This reconstruction must be conservative, non-oscillatory and sufficiently accurate since it is used to compute the numerical fluxes, which typically inherit these properties from the interpolant.

For a given Godunov-type scheme, one may establish a time-step stability restriction, which requires Δt^n to be proportional to $\Delta x/a^n$, where a^n is the largest characteristic speed:

$$a^n=\max_x\left\{\rho\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\widetilde{\mathbf{u}}^n(x))\right)\right\}. \tag{2.1}$$

Here, $\rho(\cdot)$ is a spectral radius of the Jacobian and the maximum is taken over the entire computational domain. Note that formula (2.1) can be rewritten as:

$$a^n = \max\{a^n_{\max}, -a^n_{\min}\},$$

where

$$a^n_{\max} := \max_x \left\{ \lambda_N \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\widetilde{u}^n(x)) \right) \right\}, \quad a^n_{\min} := \min_x \left\{ \lambda_1 \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\widetilde{u}^n(x)) \right) \right\},$$

and $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_N$ are the eigenvalues of the Jacobian. The terms $a^n_{\max}$ and $a^n_{\min}$ have a clear meaning: global maximal and minimal characteristic speeds. Globally, the entire wave structure moves at the average characteristic speed

$$\sigma^n = \frac{a^n_{\max} + a^n_{\min}}{2}.$$

If σ^n is not zero, the CFL number can be reduced by switching to the new coordinate system $(t, x - \sigma^n t)$, in which the average characteristic velocity will be zero (locally in time, i.e., for $t \in [t^n, t^n + \Delta t^n)$). In the new coordinates, we numerically solve the modified system

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x - \sigma^n \mathbf{u}_x = \mathbf{0}, \quad t \in [t^n, t^n + \Delta t^n), \tag{2.2}$$

for which the CFL number, compared to the CFL number for the original system (1.1), is reduced by a factor β^n given by

$$\beta^n = \frac{a^n}{a^n - |\sigma^n|}. \tag{2.3}$$

Summary of the quasi-Lagrangian acceleration strategy: we begin at $t = t^n$ with the cell averages $\overline{u}^n_j$, which are evolved to $t = t^{n+1}$ numerically solving (2.2). We then shift all spatial grid points x^n_j to the new locations $x^{n+1}_j = x^n_j + \sigma^n \Delta t^n$ so that the computed solution is translated by $\sigma^n \Delta t^n$. This completes one time step of the resulting method.

Remarks.

1. The proposed acceleration strategy may be implemented with any explicit Eulerian method, not necessarily with a Godunov-type finite-volume scheme.

2. An additional advantage of the proposed method is automatic adjustment of the computational domain to the location of the computed waves. This is important for solving initial value problems, for which the infinite spatial domain is artificially truncated so that the "action" happens inside the computational domain. Our method minimizes its size, which leads to an additional efficiency gain.

3. Our approach is easily extended to a higher number of space dimensions. Consider, for example, a 2-D hyperbolic system of conservation laws:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{0}. \tag{2.4}$$

To apply our method to this system, one will first need to use the solution, computed at time $t=t^n$ to evaluate the largest and the smallest characteristic speeds in both x- and y-directions:

$$a^n_{\max}:=\max_{x,y}\left\{\lambda_N\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\widetilde{u}^n(x,y))\right)\right\},\quad a^n_{\min}:=\min_{x,y}\left\{\lambda_1\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\widetilde{u}^n(x,y))\right)\right\},$$
$$b^n_{\max}:=\max_{x,y}\left\{\lambda_N\left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\widetilde{u}^n(x,y))\right)\right\},\quad b^n_{\min}:=\min_{x,y}\left\{\lambda_1\left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\widetilde{u}^n(x,y))\right)\right\}.$$

Here, $\lambda_1(\cdot)\le\lambda_2(\cdot)\le\ldots\le\lambda_N(\cdot)$ are the eigenvalues of the Jacobians $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}$, $\widetilde{u}^n(x,y)$ is a global interpolant of computed discrete solution, and the maximum and minimum are taken over the entire computational domain. Then, instead of solving (2.4), we numerically solve the modified equation:

$$\mathbf{u}_t+\mathbf{f}(\mathbf{u})_x+\mathbf{g}(\mathbf{u})_y-\sigma^n\mathbf{u}_x-\delta^n\mathbf{u}_y=\mathbf{0},\quad t\in[t^n,t^n+\Delta t^n),$$

where the linear advection coefficients are:

$$\sigma^n=\frac{a^n_{\max}+a^n_{\min}}{2},\quad \delta^n=\frac{b^n_{\max}+b^n_{\min}}{2}.$$

Finally, we shift both the mesh and the computed solution at time level $t=t^{n+1}$ by $(\sigma^n\Delta t^n,\delta^n\Delta t^n)^T$.

3 Numerical Viscosity Analysis

The implementation of our quasi-Lagrangian acceleration strategy clearly changes the amount of numerical dissipation present at an underlying method, which, in turn, may have an influence on the achieved resolution. It is hard (if at all possible) to estimate this change in the case of a general Eulerian method applied to a general system of conservation laws, especially when a high-order method is applied to a nonlinear system. It is instructive, however, to study the change in the dissipation of first-order schemes applied to simple 1-D linear systems.

Recall that the first-order numerical flux can be written in the viscosity form:

$$\mathbf{H}_{j+\frac{1}{2}}=\frac{\mathbf{f}(\overline{\mathbf{u}}^n_j)+\mathbf{f}(\overline{\mathbf{u}}^n_{j+1})}{2}-\frac{1}{2}Q_{j+\frac{1}{2}}(\overline{\mathbf{u}}^n_{j+1}-\overline{\mathbf{u}}^n_j),\tag{3.1}$$

where $Q_{j+\frac{1}{2}}$ is a numerical viscosity matrix (see, e.g., [3]). We consider the simplest (yet nontrivial[†]) case of a linear 2×2 diagonal system with constant coefficients:

$$\mathbf{u}_t+\mathbf{f}(\mathbf{u})_x=\mathbf{0},\quad \mathbf{f}(\mathbf{u})=\Lambda\mathbf{u},\quad \Lambda=\begin{pmatrix}\lambda_2 & 0\\ 0 & \lambda_1\end{pmatrix},\quad \lambda_1<\lambda_2,\tag{3.2}$$

[†]This case is nontrivial since we apply nonlinear numerical methods to the linear system (3.2) and its modification.

where λ_1 and λ_2 are constants. According to our acceleration strategy, we modify (3.2) to obtain:

$$\mathbf{u}_t+\frac{1}{2}\begin{pmatrix}\lambda_2-\lambda_1 & 0\\ 0 & \lambda_1-\lambda_2\end{pmatrix}\mathbf{u}_x=\mathbf{0}. \tag{3.3}$$

We consider two particular examples of the Rusanov [17] and the Harten-Lax-van Leer (HLL) [5] schemes. Our choice is motivated by numerical experiments, presented in §3, where the second-order central-upwind schemes [9–11] have been used (the first-order version of the scheme from [11] is the Rusanov scheme, while the first-order reduction of the scheme from [10] is the HLL scheme).

Example 3.1—the Rusanov Scheme

We begin with the Rusanov scheme, whose numerical flux can be written in the viscosity form (3.1) with

$$Q^{\text{Rus}}_{j+\frac{1}{2}}=a_{j+\frac{1}{2}}I,\quad a_{j+\frac{1}{2}}:=\max\left\{\rho\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\overline{\mathbf{u}}^n_j)\right),\rho\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\overline{\mathbf{u}}^n_{j+1})\right)\right\},$$

where I is the identity matrix. When the Rusanov scheme is applied to the linear system (3.2), the numerical viscosity becomes:

$$Q^{\text{Rus}}_{j+\frac{1}{2}}=a_{j+\frac{1}{2}}I,\quad a_{j+\frac{1}{2}}=\max\{|\lambda_2|,|\lambda_1|\}.$$

Now, when we numerically solve (3.3) instead of the original linear system, the numerical viscosity of the Rusanov scheme clearly reduces to:

$$Q^{\text{Rus}}_{j+\frac{1}{2}}=a_{j+\frac{1}{2}}I,\quad a_{j+\frac{1}{2}}=\frac{\lambda_2-\lambda_1}{2}. \tag{3.4}$$

Notice that the numerical viscosity becomes smaller by a factor of

$$\frac{2\max\{|\lambda_2|,|\lambda_1|\}}{\lambda_2-\lambda_1},$$

which is also the acceleration factor, see (2.3).

Remark. We would like to point out that unless the $\lambda_2=-\lambda_1$, our approach guarantees both efficiency improvement and reduction of numerical dissipation for the Rusanov scheme. The improvement factor, however, is especially large when λ_1 and λ_2 are close to each other and $\text{sgn}(\lambda_1)=\text{sgn}(\lambda_2)$.

Example 3.2—the HLL Scheme

The numerical flux of the HLL scheme is given by:

$$\mathbf{H}^{\text{HLL}}_{j+\frac{1}{2}}=\frac{a^+_{j+\frac{1}{2}}\mathbf{f}(\overline{\mathbf{u}}^n_j)-a^-_{j+\frac{1}{2}}\mathbf{f}(\overline{\mathbf{u}}^n_{j+1})}{a^+_{j+\frac{1}{2}}-a^-_{j+\frac{1}{2}}}+\frac{a^+_{j+\frac{1}{2}}a^-_{j+\frac{1}{2}}}{a^+_{j+\frac{1}{2}}-a^-_{j+\frac{1}{2}}}(\overline{\mathbf{u}}^n_{j+1}-\overline{\mathbf{u}}^n_j), \tag{3.5}$$

where $a^+_{j+\frac{1}{2}}$ and $a^-_{j+\frac{1}{2}}$ are the largest and the smallest local signal speeds, respectively, see [5]. In the case of a linear flux (3.2), the local speeds are independent of j and are equal to:

$$a^+_{j+\frac{1}{2}}=\max\{\lambda_2,0\},\qquad a^-_{j+\frac{1}{2}}=\min\{\lambda_1,0\}.$$

Rewriting (3.5) in the numerical viscosity form (3.1), we obtain (in the linear 2×2 case (3.2)):

$$Q^{\rm HLL}_{j+\frac{1}{2}}=\begin{pmatrix}|\lambda_2| & 0\\ 0 & |\lambda_1|\end{pmatrix}.$$

Notice, that in the case of a symmetric Riemann fan, the HLL scheme reduces to the Rusanov scheme. Therefore, when the HLL scheme is applied to the modified system (3.3), its numerical viscosity is:

$$Q^{\rm HLL}_{j+\frac{1}{2}}=\frac{1}{2}\begin{pmatrix}\lambda_2-\lambda_1 & 0\\ 0 & \lambda_2-\lambda_1\end{pmatrix},$$

which is identical to (3.4).

Unlike the Rusanov scheme, the numerical viscosity of the HLL scheme may or may not reduce by applying the scheme to the modified system. This depends upon the relation between λ_2 and λ_1. If $\lambda_2\geq3\lambda_1>0$ or $\lambda_1\leq3\lambda_2<0$, then the numerical viscosity coefficients will become smaller for both components of the system. Otherwise, one of the viscosity coefficients will increase. This indicates that in the case of the system with both slow and fast moving waves, with the slow waves being "significant" ones, our acceleration method may lead to deterioration in achieved resolution. At the same time, when all the "significant" waves are of about the same speed, our approach would provide both efficiency and resolution improvements.

4 Numerical Examples

In this section, we demonstrate a great potential of the proposed quasi-Lagrangian strategy on a number of 1-D and 2-D gas dynamics test problems. The underlying scheme used in all the examples below, is the second-order semi-discrete central-upwind scheme from [9]. We have implemented it with a piecewise linear generalized minmod reconstruction with the minmod parameter $\theta=1.3$ in the 1-D examples and $\theta=1$ in the 2-D one (see, e.g., [9] for details). The time integration has been performed by the third-order strong stability preserving Runge-Kutta solver [4].

In the following, we will refer to the central-upwind scheme and to its quasi-Lagrangian modification as CU and QLCU, respectively. A time step size, used by the QLCU scheme is typically much larger than the time step used by the CU scheme to solve the same problem. We will refer to the ratio between the average QLCU time step and the average CU time step as the *average CFL acceleration factor*.

Example 4.1—Three Variations of the Sod Problem

We consider the 1-D Euler equations of gas dynamics:

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2+p \\ u(E+p) \end{pmatrix}_x = \mathbf{0}, \tag{4.1}$$

where ρ, u, p, and E are the density, velocity, pressure, and the total energy, respectively. The system is complete through the equation of state for the ideal gas:

$$E=\frac{p}{\gamma-1}+\frac{\rho u^2}{2}, \quad \gamma=1.4. \tag{4.2}$$

We numerically solve three supersonic variations of the Sod shock-tube problem [20]. The first set of initial data is:

$$(\rho(x,0),u(x,0),p(x,0))^T=\begin{cases} (1.000,10,1.0)^T, & x<0.5, \\ (0.125,10,0.1)^T, & x>0.5. \end{cases} \tag{4.3}$$

This data is obtained from the Sod data[‡] by adding 10 to the initial velocity.

Numerical solutions at time $t=0.25$, computed by both the QLCU and CU schemes are presented in Figures 1–2. In both cases, we use a uniform grid with $\Delta x=1/100$ (the reference solution is computed by the QLCU scheme with $\Delta x=1/1000$). One may clearly see the superiority of the resolution achieved by the QLCU scheme. This is also confirmed by Table 1, where we compare the L^1-errors in the density field for four different grids. As expected, both the QLCU and CU schemes are first-order accurate in the presence of the discontinuities, but the QLCU errors are about 3 times smaller than the CU ones. In addition, the QLCU scheme is much faster in this example and the average CFL acceleration factor is 7.

The second set of data is obtained from the Sod data by adding 1 to the initial velocity and multiplying the initial densities by 100:

$$(\rho(x,0),u(x,0),p(x,0))^T=\begin{cases} (100,1,1.0)^T, & x<0.5, \\ (12.5,1,0.1)^T, & x>0.5. \end{cases} \tag{4.4}$$

We apply the QLCU and CU schemes to the IVP (4.1)–(4.2), (4.4), and the solutions, computed at time $t=4$, are shown in Figures 3–4. Once again, we use a uniform grid with $\Delta x=1/100$ and the reference solution computed by the QLCU scheme with $\Delta x=1/1000$. The results are similar to what has been obtained for the IVP (4.1)–(4.3). As expected, application of our acceleration procedure leads to both enhanced resolution and efficiency improvement (the average CFL acceleration factor is about 7.01 here).

[‡]The Sod initial data are: $(\rho(x,0),u(x,0),p(x,0))^T=\begin{cases} (1.000,0,1.0)^T, & x<0.5, \\ (0.125,0,0.1)^T, & x>0.5. \end{cases}$

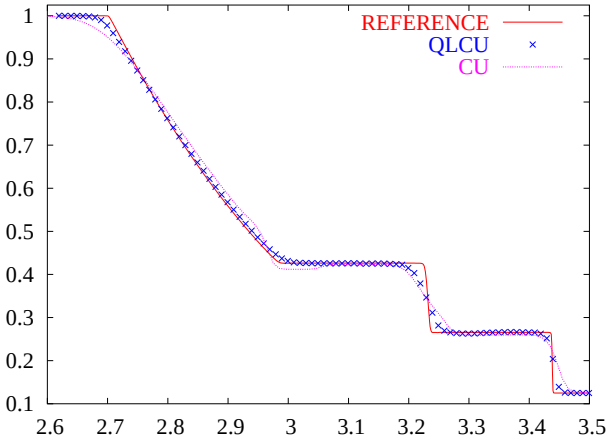

Figure 1: IVP (4.1)–(4.3): density by the QLCU and CU schemes.

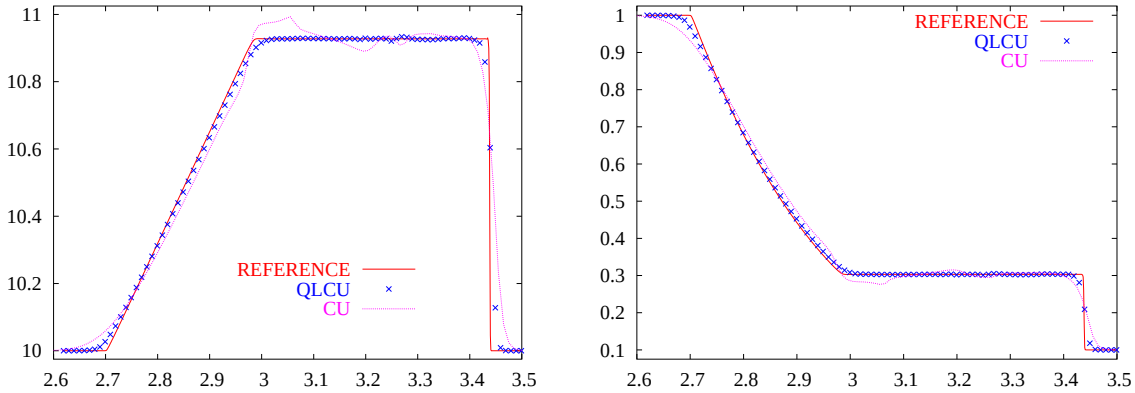

Figure 2: IVP (4.1)–(4.3): velocity (left) and pressure (right) by the QLCU and CU schemes.

The third set of data is obtained from the Sod data by adding a variable function $V(x)$ to the initial velocity:

$$(\rho(x,0),u(x,0),p(x,0))^T = \begin{cases} (1.000,V(x),1.0)^T, & x<0.5, \\ (0.125,V(x),0.1)^T, & x>0.5, \end{cases} \qquad V(x)=10+e^{-50(x-0.5)^2}. \quad (4.5)$$

The solution of the IVP (4.1)–(4.2), (4.5) develops much more complex wave structure than the self-similar solution of the Riemann problem (4.1)–(4.3) studied above. In Figures 5–6, we show the solutions of (4.1)–(4.2), (4.5) at time $t=0.25$ computed by the QLCU and CU schemes on the same uniform grid with $\Delta x = 1/100$ (as before, the reference solution is computed by the QLCU scheme with $\Delta x = 1/1000$). As one can see there, the use of our quasi-Lagrangian acceleration approach significantly enhances the overall resolution. Notice that now the difference between the QLCU and CU results is

Δx	QLCU Scheme		CU Scheme	
	L^1-error	Rate	L^1-error	Rate
1/100	6.54e-03	–	1.54e-02	–
1/200	3.28e-03	1.00	9.94e-03	0.64
1/400	1.75e-03	0.91	5.24e-03	0.92
1/800	9.94e-04	0.82	2.80e-03	0.90
1/1600	5.37e-04	0.89	1.41e-03	0.99

Table 1: L^1-errors and experimental convergence rates.

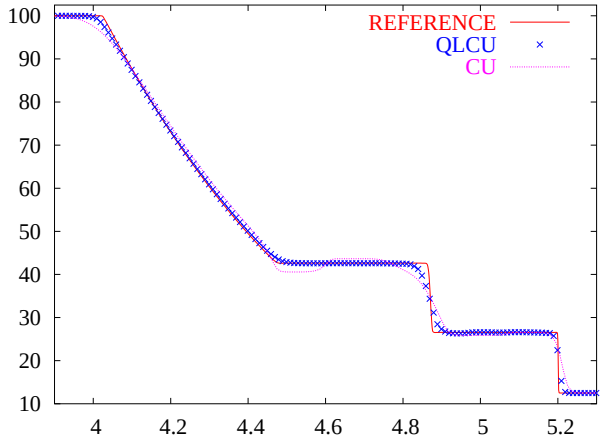

Figure 3: IVP (4.1)–(4.2), (4.4): density by the QLCU and CU schemes.

even more pronounced than in the two previously considered Riemann problems. At the same time, the average CFL acceleration factor is almost as large as before (it is about about 6.22 now).

Example 4.2—Three Discontinuities Traveling to the Right

In this example, we numerically solve the 1-D Euler equations of gas dynamics (4.1) subject to the initial data:

$$(\rho(x,0),u(x,0),p(x,0))^T = \begin{cases} (5.99924,69.5975,460.894)^T, & x<0.5, \\ (5.99924,43.80367,46.095)^T, & x>0.5, \end{cases} \tag{4.6}$$

which is a supersonic modification (50 is added to the initial velocities) of Test 4 from [23][§].

[§]Toro's Test 4 initial data are: $(\rho(x,0),u(x,0),p(x,0))^T = \begin{cases} (5.99924,19.5975,460.894)^T, & x<0.5, \\ (5.99924,-6.19633,46.095)^T, & x>0.5. \end{cases}$

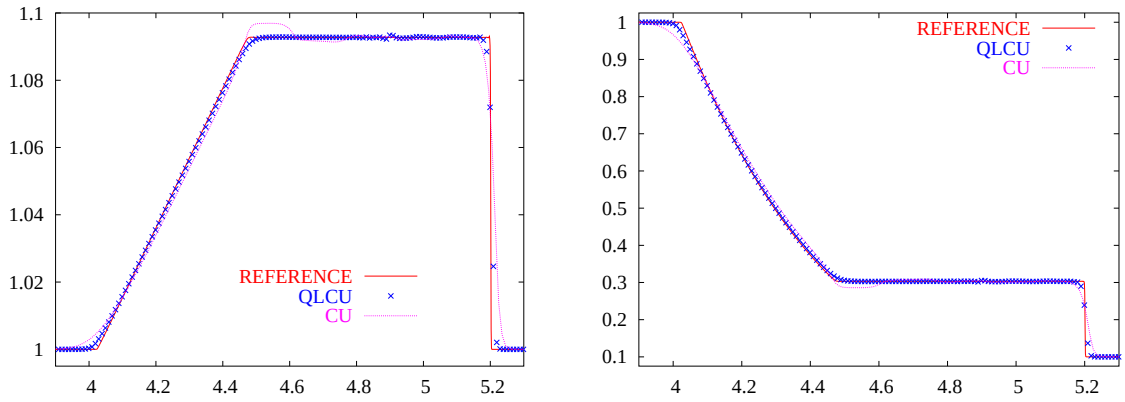

Figure 4: IVP (4.1)–(4.2), (4.4): velocity (left) and pressure (right) by the QLCU and CU schemes.

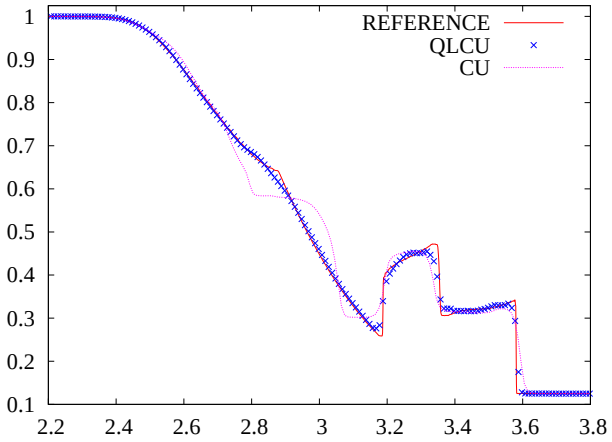

Figure 5: IVP (4.1)–(4.2), (4.5): density by the QLCU and CU schemes.

The solution is computed at time $t=0.035$ by the QLCU and CU schemes on a uniform spatial grid with $\Delta x = 1/100$. In Figures 7–8, we plot these numerical solutions together with the reference solution, obtained by the QLCU scheme on a 10 times finer mesh. The average CFL acceleration factor is about 3.44. It is smaller here than in Example 4.1, but the difference in the achieved resolution is still very substantial, especially in the density field (see Figure 7).

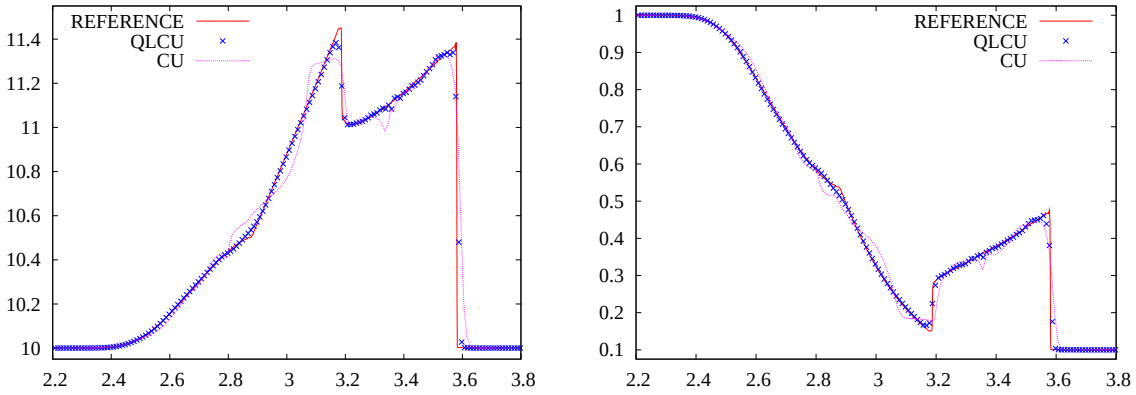

Figure 6: IVP (4.1)–(4.2), (4.5): velocity (left) and pressure (right) by the QLCU and CU schemes.

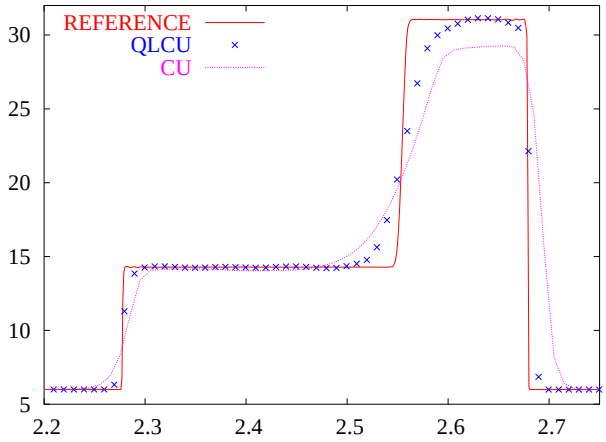

Figure 7: IVP (4.1)–(4.2), (4.6): density by the QLCU and CU schemes.

Example 4.3—Two-Dimensional Riemann Problem

We now consider the 2-D Euler equations of gas dynamics:

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ u(E+p) \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ v(E+p) \end{pmatrix}_y = \mathbf{0}, \qquad E=\frac{p}{\gamma-1}+\frac{\rho}{2}(u^2+v^2),\ \gamma=1.4, \tag{4.7}$$

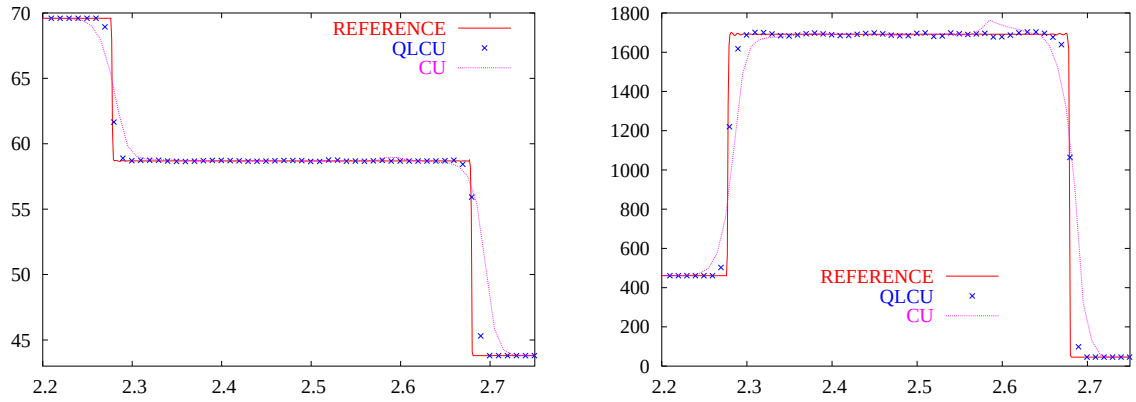

Figure 8: IVP (4.1)–(4.2), (4.6): velocity (left) and pressure (right) by the QLCU and CU schemes.

where v is the y-component of the velocity. We numerically solve the 2-D Riemann problem with the following initial data:

$$(\rho(x,y,0),u(x,y,0),v(x,y,0),p(x,y,0))^T=\begin{cases}(4,5.0,4.5,1)^T, & x>0.5,\ y>0.5,\\ (8,5.0,5.5,1)^T, & x<0.5,\ y>0.5,\\ (4,6.5,5.5,1)^T, & x<0.5,\ y<0.5,\\ (12,6.5,4.5,1)^T, & x>0.5,\ y<0.5.\end{cases}\tag{4.8}$$

This is a supersonic modification of Configuration 5 data taken from [12, 13, 18]: all the initial densities have been increased by a factor of 4 and all the initial velocities u and v have been increased by 5.75 and 5, respectively[¶].

We compute the solution of the IVP (4.7)–(4.8) at time $t=0.2$ using the QLCU and CU schemes. In this problem, the average CFL acceleration factor is about 5.28. In Figure 9, we show the solutions obtained on a uniform grid with $\Delta x=\Delta x=1/400$. The QLCU solution seems to be much sharper resolved (some of the density structures cannot be even seen in Figure 9 (right), where the CU solution is plotted), but a little more oscillatory. Since the exact solution of the IVP (4.7)–(4.8) is not available, we compare the obtained solutions with the corresponding solutions computed using a finer mesh with $\Delta x=\Delta y=1/800$, see Figure 10. One may clearly see that some of the solution features, captured on the coarser grid by the QLCU scheme only, now emerge in the finer mesh

[¶]The original Configuration 5 data are:

$$(\rho(x,y,0),u(x,y,0),v(x,y,0),p(x,y,0))^T=\begin{cases}(1,-0.75,-0.5,1)^T, & x>0.5,\ y>0.5,\\ (2,-0.75,\ 0.5,1)^T, & x<0.5,\ y>0.5,\\ (1,\ 0.75,\ 0.5,1)^T, & x<0.5,\ y<0.5,\\ (3,\ 0.75,-0.5,1)^T, & x>0.5,\ y<0.5.\end{cases}$$

CU computation as well (compare Figure 10 (right) with Figure 9 (left)). However, the coarser QLCU solution is still better resolved than the finer CU one.

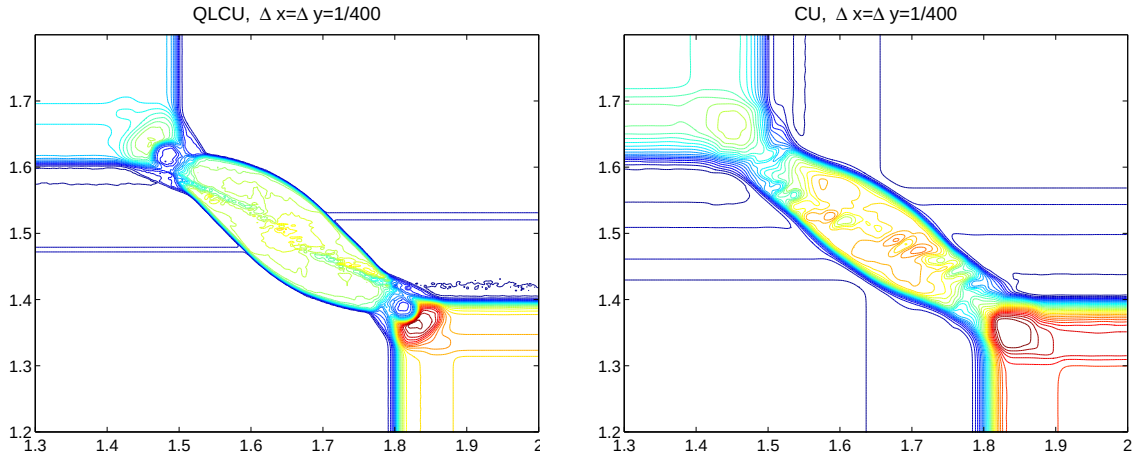

Figure 9: Solution (density) of the IVP (4.7)–(4.8) by the QLCU (left) and CU (right) schemes.

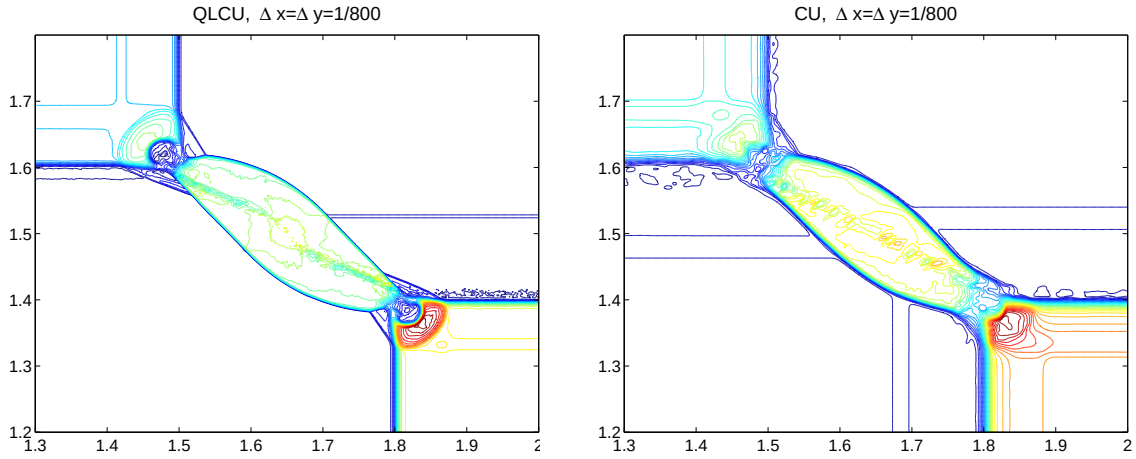

Figure 10: The same as Figure 9, but the computational mesh is refined.

5 Concluding Remarks

We have proposed a very simple quasi-Lagrangian approach for efficiency improvement of Eulerian methods for general multidimensional hyperbolic systems of conservation laws. Our method is based on the moving frame idea: the grid, while remaining structured, is shifted at every time level with the average wave speed. By doing this, the

maximum speed is reduced and the CFL condition, which restricts the size of time steps in explicit Eulerian methods, is weakened.

Our strategy is simple and universal—it allows one to accelerate any Eulerian method without increasing its complexity. The main advantages of the quasi-Lagrangian acceleration approach has been demonstrated using the central-upwind scheme from [9], applied to 1-D and 2-D Euler equations of gas dynamics.

Acknowledgments

The work of A. Kurganov was supported in part by NSF grants DMS-0310585 and DMS-0610430.

References

[1] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Engrg., 99 (1992), 235-394.

[2] B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, CIME Lecture Notes (A. Quarteroni edt.), Lecture Notes in Mathematics 1697, Springer-Verlag, 1997.

[3] E. Godlewski and P.-A. Raviart, Numerical Approximation of Hyperbolic Systems of Conservation Laws, Springer-Verlag New York, 1996.

[4] S. Gottlieb, C.-W. Shu and E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev., 43 (2001), 89-112.

[5] A. Harten, P. Lax and B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev., 25, (1983), 35-61.

[6] R. D. Haynes, W. Huang and R. D. Russell, A moving mesh method for time-dependent problems based on Schwartz waveform relaxation, in Proceedings of Domain Decomposition Methods in Science and Engineering XVII, Lecture Notes in Computational Science and Engineering, Springer-Verlag, Vol. 60, U. Langer, M. Discacciati, D. E. Keyes, O. B. Widlund (Eds.), 229-236, 2008.

[7] W. Huang and X. Zhan, Adaptive moving mesh modeling for two dimensional groundwater flow and transport, AMS Contemporary Mathematics, 383 (2005), 283-296.

[8] D. Kröner, Numerical Schemes for Conservation Laws, Wiley Chichester, 1997.

[9] A. Kurganov and C.-T. Lin, On the reduction of numerical dissipation in central-upwind schemes, Commun. Comput. Phys., 2 (2007), 141-163.

[10] A. Kurganov, S. Noelle and G. Petrova, Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations, SIAM J. Sci. Comput., 23 (2001), 707-740.

[11] A. Kurganov and E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, J. Comput. Phys., 160 (2000), 214-282.

[12] A. Kurganov and E. Tadmor, Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers, Numer. Methods Partial Differential Equations, 18 (2002), 584-608.

[13] P. Lax and X.-D. Liu, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM J. Sci. Comp., 19 (1998), 319-340.

[14] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
[15] K. Lipnikov and M. Shashkov, The error-minimization-based strategy for moving mesh methods, Commun. Comput. Phys., 1 (2006), 53-80.
[16] L. G. Margolin, Introduction to "An arbitrary Lagrangian-Eulerian computing method for all flow speeds", J. Comput. Phys., 135 (1997), 198-202.
[17] V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR, 1 (1961), 267-279.
[18] C. W. Schulz-Rinne, J. P. Collins and H. M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, SIAM J. Sci. Comput., 14 (1993), 1394-1414.
[19] M. Shashkov, Conservative Finite-Difference Methods on General Grids, CRC Press Boca Raton, 1996.
[20] G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys., 22 (1978), 1-31.
[21] I. V. Sokolov, E. V. Timofeev, J-I. Sakai and K. Takayama, Artificial wind—a new framework to construct simple and efficient upwind shock-capturing schemes, J. Comput. Phys., 181 (2002), 354-393.
[22] H. Tang and T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal., 41 (2003), 487-515.
[23] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction, Second edition, Springer-Verlag Berlin, 1999.
[24] H. Trac and U.-L. Pen, A moving frame algorithm for high Mach number hydrodynamics, New Astronomy, 9 (2004), 443-465.