ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

LAGRANGE-FLUX EULERIAN SCHEMES FOR COMPRESSIBLE MULTIMATERIAL FLOWS

Florian De VUYST[2], Thibault GASC[1,2,3], Renaud MOTTE[3], Mathieu PEYBERNES[4] and Raphaël PONCET[5]

[1] CMLA, ENS Cachan Université Paris-Saclay, CNRS,
94235 Cachan, France
e-mail: devuyst@cmla.ens-cachan.fr

[2]Maison de la Simulation USR 3441
CEA Saclay, F-91191 Gif-sur-Yvette
e-mail: thibault.gasc@cea.fr

[3]CEA DAM DIF
F-91297 Arpajon

[4] CEA, DEN, DM2S, STMF
CEA Saclay, F-91191 Gif-sur-Yvette
e-mail: mathieu.peybernes@cea.fr

[5] CGG
27 Avenue Carnot, F-91300 Massy
e-mail: raphael.poncet@cgg.com

Keywords: Compressible fluid, Lagrangian remapping, Lagrange-flux scheme, multimaterial flow, performance model

Abstract. *This paper is devoted to the redesign of the Lagrangian remapping process for compressible Hydrodynamics in order to achieve better computational performance. As an unintended outcome, the analysis has lead us to the discovery of a new family of solvers – the so-called Lagrange-flux schemes – that appear to be very promising for the CFD community. Results obtained with the Lagrange-flux approach are comparable with those obtained with the "legacy" Lagrange-remap solver using staggered variables, but provide a far better scalable computational performance, as proven using performance modeling. We also extend Lagrange-flux scheme to the case of multimaterial flows involving interface capturing advection schemes.*

1 Motivation and introduction

For complex compressible flows involving multiphysics phenomenons like e.g. high-speed elastoplasticity, multimaterial interaction, plasma, gas-particles etc., a Lagrangian description of the flow is generally preferred for easier physical coupling. To ensure robustness, some spatial remapping on a regular mesh may be added. A particular case is the family of the so-called Lagrange+remap schemes [2], also referred to as Lagrangian remapping that apply a remap step on a reference (say Eulerian) mesh after each Lagrangian time advance step. Acknowledged legacy codes implementing remapped Lagrange solvers usually define thermodynamical variables at cell centers and velocity variables at mesh nodes (see figure 1). In Poncet et al. [1],

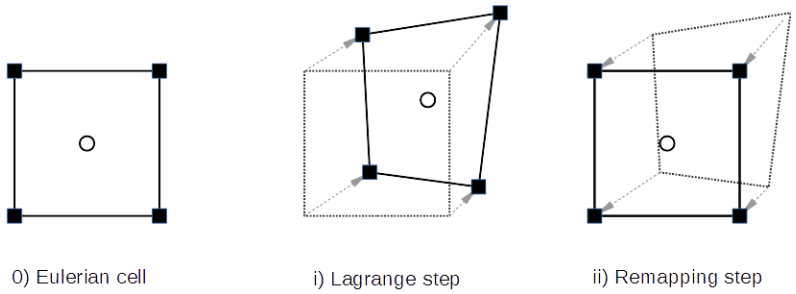


Figure 1: "Legacy" staggered Lagrange-remap scheme: thermodynamical variables are located at cell centers (circles) whereas velocity variables are located at cell nodes (squares).

we have achieved a node-based performance analysis of a reference legacy Lagrange-remap hydrodynamics solver used in industry. By analyzing each kernel of the whole algorithm, using roofline-type models [3] on one side and refined Execution Cache Memory (ECM) models [4], [5] on the other side, we have been able not only to quantitatively predict the performance of the whole algorithm — with relative errors in the single digit range — but also to identify a set of features that limit the whole global performance. This can be roughly summarized into three features:

1. Staggered velocity variables involve a rather big amount of communication to/from CPU caches and memory with low arithmetic intensity, thus lowering the whole performance;

2. Alternating direction (AD) strategies (see the appendix in [6]) or more specifically AD remapping procedures also generate too much communication with a loss of CPU occupancy and a rather poor multicore scalability.

3. For multimaterial hydrodynamics using VOF-based interface reconstruction methods, there is a strong loss of performance due to some array indirections and noncoalescent data in memory. Vectorization of such algorithms is also not trivial.

From these observations and as a result of the analysis, we decided to "rethink" Lagrange-remap schemes, with possibly modifying some aspects of the solver in order to improve node-based performance of the hydrocode solver. We have searched for alternative formulations that lower communication and improve both arithmetic intensity and SIMD property of the algorithm.

In this paper, we describe the process of redesign of Lagrange-remap schemes leading to higher performance solvers. Actually, this redesign methodology has also givn us ideas of

innovative Eulerian solvers. The so-called *Lagrange-flux schemes* appear to be very promising in the Computational Fluid Dynamics community.

The paper is organized as follows. In section 2, we first formulate the requirements for the design of higher-performance Lagrange-remap schemes. In section 3 we give a description of the Lagrange step and formulate it under a Finite Volume form. In section 4 we focus on the remap step which is reformulated as a finite volume scheme with pure convective fluxes. This interpretation is applied in section 5 to build the so-called Lagrange-flux schemes. We also discuss the important issue of achieving second order accuracy (in both space and time). In section 7 we deal with the extension to multimaterial flow with the use of low-diffusive and accurate interface-capturing methods.

2 Requirements

Starting from a "legacy" staggered Lagrange-remap solver and related observed performance measurements, we want to improve the performance by modifying the computational approach under the following constraints and requirements:

1. A Lagrangian solver must be used (for multiphysics coupling issue).
2. To reduce communication, we prefer to use collocated cell-centered variables rather than a staggered scheme.
3. To reduce communication, we prefer use a direct multidimensional remap solver rather than splitted alternating direction AD projections.
4. The method should be simply extended to second-order accuracy (in both space and time).
5. The solver must be able to be naturally extended to multimaterial flows.

Before going further, let us first comment the above requirements. The second requirement should imply the use of a cell-centered Lagrange solver. Fairly recently, Després and Mazeran in [7] and Maire and et al. [8] (with high-order extension in [9]) have proposed pure cell-centered Lagrangian solvers based on the reconstruction of nodal velocities. In our study, we will examine if it is possible to use approximate and simpler Lagrangian solvers in the Lagrange+remap context, in particular for the sake of performance. The fourth assertion requires a full multidimensional remapping step, probably taking into account geometric elements (deformation of cells and edges) if we want to ensure high-order accuracy remapping. We have to find a good trade-off between simplifications-approximations and accuracy (or properties) of the numerical solver.

3 Lagrangian time advance step

As example, let us consider the compressible Euler equations for two-dimensional problems. Denoting ρ, $\boldsymbol{u} = (u_i)_i$, $i \in \{1,2\}$, p and E the density, velocity, pressure and specific total energy respectively, the mass, momentum and energy conservation equations are

$$\partial_t U_\ell + \nabla \cdot (\boldsymbol{u}\, U_\ell) + \nabla \cdot \boldsymbol{\pi}_\ell = 0, \quad \ell = 1, \ldots, 4, \tag{1}$$

where $U = (\rho, (\rho u_i)_i, \rho E)$, $\boldsymbol{\pi}_1 = \vec{0}$, $\boldsymbol{\pi}_2 = (p, 0)^T$, $\boldsymbol{\pi}_3 = (0, p)^T$ and $\boldsymbol{\pi}_4 = p\boldsymbol{u}$. For the sake of simplicity, we will use a perfect gas equation of state $p = (\gamma - 1)\rho(E - \frac{1}{2}|\boldsymbol{u}|^2)$, $\gamma \in (1, 3]$. The speed of sound c is given by $c = \sqrt{\gamma p/\rho}$.

For any volume V_t that is advected by the fluid, from the Reynolds transport theorem we have

$$\frac{d}{dt}\int_{\mathscr{V}_t} U_\ell \, d\boldsymbol{x} = \int_{\partial\mathscr{V}_t} \{\partial_t U_\ell + \nabla\cdot(\boldsymbol{u}\, U_\ell)\}\, d\boldsymbol{x} = -\int_{\partial\mathscr{V}_t} \boldsymbol{\pi}_\ell\cdot\boldsymbol{\nu}\, d\sigma$$

where $\boldsymbol{\nu}$ is the normal unit vector exterior to $\mathscr{V}_t$. This leads to a natural explicit finite volume scheme in the form

$$|K^{n+1,L}|(U_\ell)_K^{n+1,L} = |K|(U_\ell)_K^n - \Delta t^n \sum_{A^{n+\frac{1}{2},L}\subset\partial K^{n+\frac{1}{2},L}} |A^{n+\frac{1}{2},L}|\, \boldsymbol{\pi}_A^{n+\frac{1}{2},L}\cdot\nu_A^{n+\frac{1}{2},L}. \tag{2}$$

In expression (2), the superscript "L" indicates the Lagrange evolution of the quantity. Any Eulerian cell K is deformed into the Lagrangian volume $K^{n+\frac{1}{2},L}$ at time $t^{n+\frac{1}{2}}$, and into the Lagrangian volume $K^{n+1,L}$ at time t^{n+1}. The pressure flux terms through the edges $A^{n+1/2,L}$ are evaluated at time $t^{n+\frac{1}{2}}$ in order to get second-order accuracy in time. Of course, that means that we need a predictor step for the velocity field $\boldsymbol{u}^{n+\frac{1}{2},L}$ at time $t^{n+\frac{1}{2}}$ (not written here for simplicity).

Notations. From now on, we will use the simplified notation $\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{u}^{n+\frac{1}{2},L}$.

4 Rethinking the remap step

The remapping step consists in projecting the fields U_ℓ defined at cell centers $K^{n+1,L}$ onto the initial (reference) Eulerian mesh with cells K. Starting from an interpolated vector-valued field $\mathscr{I}^{n+1,L}U^{n+1,L}$, we project the field on piecewise-constant function on the Eulerian mesh, according to the integral formula

$$U_K^{n+1} = \frac{1}{|K|}\int_K \mathscr{I}^{n+1,L}U^{n+1,L}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{3}$$

Practically, they are many ways to deal with the projection operation (3). One can assemble elementary projection contributions by computing the volume intersections between the reference mesh and the deformed mesh. But this procedure requires the computation of all the geometrical elements. Moreover, the projection needs local tests of projection with conditional branching (think about the very different cases of compression, expansion, pure translation, etc). Thus the procedure is not SIMD and with potentially a loss of performance. The incremental remapping can also interpreted as a transport/advection process, as already emphasized by Dukowicz and Baumgardner [10].

Let us now write a different original formulation of the remapping process. In this step, there is no time evolution of any quantity, and in some sense we have $\partial_t U = 0$, that we rewrite

$$\partial_t U = \partial_t U + \nabla\cdot(-\boldsymbol{v}^{n+\frac{1}{2}}U) \;+\; \nabla\cdot(\boldsymbol{v}^{n+\frac{1}{2}}U) = 0.$$

We decide to split up this equation into two substeps, a backward convection and a forward one:

i) Backward convection:

$$\partial_t U + \nabla\cdot(-\boldsymbol{v}^{n+\frac{1}{2}}U) = 0. \tag{4}$$

ii) Forward convection:

$$\partial_t U + \nabla\cdot(\boldsymbol{v}^{n+\frac{1}{2}}U) = 0. \tag{5}$$

Each convection problem is well-posed on the time interval $[0, \Delta t^n]$ under a standard CFL condition. Let us now focus into these two steps and the way to solve them.

4.1 Backward convection in Lagrangian description

After the Lagrange step, if we solve the backward convection problem (4) over a time interval Δt^n using a Lagrangian description, we have

$$|K|(U_\ell)_K^{n,\star} = |K^{n+1,L}|(U_\ell)_K^{n+1,L}. \tag{6}$$

Actually, from the cell $K^{n+1,L}$ we go back to the original cell K with conservation of the conservative quantities. For $\ell = 1$ (conservation of mass), we have

$$|K|\,\rho_K^{n,\star} = |K^{n+1,L}|\,\rho_K^{n+1,L}$$

showing the variation of density by volume variation. For $\ell = 2, 3, 4$, it is easy to see that both velocity and specific total energy are kept unchanged is this step:

$$\boldsymbol{u}^{n,\star} = \boldsymbol{u}^{n+1,L}, \quad E^{n,\star} = E^{n+1,L}.$$

Thus, this step is clearly computationally inexpensive.

4.2 Forward convection in Eulerian description

From the discrete field $(U_K^{n,\star})_K$ defined on the Eulerian cells K, we then solve the forward convection problem over a time step Δt^n under an Eulerian description. A standard Finite Volume discretization of the problem will lead to the classical time advance scheme

$$U_K^{n+1} = U_K^{n,\star} - \frac{\Delta t^n}{|K|} \sum_{A \subset \partial_K} |A|\, U_A^{n+\frac{1}{2},\star}\, (\boldsymbol{v}_A^{n+\frac{1}{2}} \cdot \nu_A) \tag{7}$$

for some interface values $U_A^{n+\frac{1}{2},\star}$ defined from the local neighbor values $U_K^{n,\star}$. We finally get the expected Eulerian values U_K^{n+1} at time t^{n+1}.

Notice that from (6) and (7) we have also

$$|K|\, U_K^{n+1} = |K^{n+1,L}|\, U_K^{n+1,L} - \Delta t^n \sum_{A \subset \partial K} |A|\, U_A^{n+\frac{1}{2},\star}\, (\boldsymbol{v}_A^{n+\frac{1}{2}} \cdot \nu_A) \tag{8}$$

thus completely defining the remap step under the finite volume scheme form (8). Let us emphasize that we do not need any mesh intersection or geometric consideration to achieve the remapping process. The finite volume form (8) is now suitable for a straightforward vectorized SIMD treatment. From (8) it is easy to achieve second-order accuracy for the remapping step by usual finite volume tools (MUSCL reconstruction + second-order accurate time advance scheme for example).

4.3 Full Lagrange+remap time advance

Let us note that the Lagrange+remap scheme is actually a conservative finite volume scheme: putting (2) into (8) gives for all ℓ:

$$\begin{aligned} (U_\ell)_K^{n+1} = (U_\ell)_K^n \quad &- \quad \frac{\Delta t^n}{|K|} \sum_{A^{n+\frac{1}{2},L} \subset \partial K^{n+\frac{1}{2},L}} |A^{n+\frac{1}{2},L}|\, (\boldsymbol{\pi}_\ell)_A^{n+\frac{1}{2},L} \cdot \nu_A^{n+\frac{1}{2},L} \\ &- \quad \frac{\Delta t^n}{|K|} \sum_{A \subset \partial K} |A|\, (U_\ell)_A^{n+\frac{1}{2},\star}\, (\boldsymbol{v}_A^{n+\frac{1}{2}} \cdot \nu_A) \end{aligned} \tag{9}$$

that can also be written

$$(U_\ell)_K^{n+1} = (U_\ell)_K^n - \frac{\Delta t^n}{|K|} \sum_{A\subset\partial K} |A| \left(\frac{|A^{n+\frac{1}{2},L}|}{|A|} (\boldsymbol{\pi}_\ell)_A^{n+\frac{1}{2},L} \cdot \nu_A^{n+\frac{1}{2},L} \right) - \frac{\Delta t^n}{|K|} \sum_{A\subset\partial K} |A| \left((U_\ell)_A^{n+\frac{1}{2},\star} (\boldsymbol{v}_A^{n+\frac{1}{2}} \cdot \nu_A) \right). \quad (10)$$

We recognize into (10) pressure-related fluxes and convective numerical fluxes.

4.4 Comments

The finite volume formulation (10) is attractive and seems rather simple at first sight. But we should not forget that we have to compute a velocity Lagrange vector field $\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{u}^{n+\frac{1}{2},L}$ where the variables should be located at cell nodes to return a well-posed deformation. Moreover, expression (10) involves geometric elements like the length of the deformed edges $A^{n+\frac{1}{2},L}$. Among rigorous constructions of collocated Lagrangian solvers, let us mention the GLACE scheme by Després-Mazeran [7] and the cell-centered EUCCLHYD solver by Maire et al [8]. Both are rather computationally expensive and their second-order accurate extension is not easy to achieve.

Although it is possible to couple these Lagrangian solvers with the flux-balanced remapping formulation, it is also worth searching for simplified approximate Lagrange formulations. One of the difficulty in the analysis of Lagrange-remap schemes is that, in some sense, space and time are coupled by the deformation process.

Below, we derive a formulation that leads to a proper separation between space and time, in order to determine the order of accuracy in a simple way. The idea is to make the time step tend to zero in the Lagrange-remap scheme (method of lines [11]), then exhibit the instantaneous spatial numerical fluxes through the Eulerian cell edges that will serve for the construction of an explicit finite volume scheme. Because the method needs an approximate Riemann solver in Lagrangian form, we will call it a Lagrange-flux scheme.

5 Derivation of a simple second-order accurate Lagrange-flux scheme

From conclusions of the discussion above, we would like to be free from any "complex" collocated Lagrangian solver involving complex geometric elements. Another difficult point is to define the deformation velocity field $\boldsymbol{v}^{n+\frac{1}{2}}$ at time $t^{n+\frac{1}{2}}$, in an accurate manner.

In what follows, we are trying to deal with time accuracy in a different manner. Let us comme back to the Lagrange+remap formula (10). Let us consider a "small" time step $t > 0$ that fulfils the usual stability CFL condition. We have

$$(U_\ell)_K(t) = (U_\ell)_K^n - \frac{t}{|K|} \sum_{A\subset\partial K} |A| \left(\frac{|A^L(t/2)|}{|A|} (\boldsymbol{\pi}_\ell)_A^L(t/2) \cdot \nu_A^L(t/2) \right) - \frac{t}{|K|} \sum_{A\subset\partial K} |A| \, (U_\ell)_A^\star(t/2) \, \boldsymbol{v}_A(t/2) \cdot \nu_A.$$

By making t tend to zero, ($t > 0$), we have $A^L(t/2) \to A$, $(\boldsymbol{\pi}_\ell)^L(t/2) \to \boldsymbol{\pi}_\ell$, $\boldsymbol{v}(t/2) \to \boldsymbol{u}$, $(U_\ell)^\star \to U_\ell$, then we get a semi-discretization in space of the conservation laws. That can be seen as a method of lines ([11]):

$$\frac{d(U_\ell)_K}{dt} = -\frac{1}{|K|} \sum_{A\subset\partial K} |A| \, ((\boldsymbol{\pi}_\ell)_A \cdot \nu_A) - \frac{1}{|K|} \sum_{A\subset\partial K} |A| \, (U_\ell)_A \, (\boldsymbol{u}_A \cdot \nu_A). \quad (11)$$

We get a classical finite volume method in the form

$$\frac{dU_K}{dt} = -\frac{1}{|K|} \sum_{A \subset \partial K} |A| \, \Phi_A$$

with a numerical flux Φ_A whose components are

$$(\Phi_\ell)_A = (U_\ell)_A \, (\boldsymbol{u}_A \cdot \nu_A) + (\boldsymbol{\pi}_\ell)_A \cdot \nu_A. \tag{12}$$

In (11), pressure fluxes $(p_\ell)_A$ and interface velocities $\boldsymbol{u}_A$ can be computed from an approximate Riemann solver in Lagrangian coordinates (for example the Lagrangian HLL solver, see [12]). Then, the interface states $(U_\ell)_A$ can be computed from a upwind process according to the sign of the normal velocity $(\boldsymbol{u}_A \cdot \nu_A)$. To get higher-order accuracy in space, one can use a standard MUSCL reconstruction + slope limiting process. At this stage, because there is no time discretization, all acts on the Eulerian mesh and fluxes are defined at the edges the the Eulerian cells.

To get high-order accuracy in time, one can then apply a standard high-order time advance scheme (RK2, etc.). For the second-order Heun scheme for example, we have the following algorithm:

1. Compute the time step Δt^n subject to some CFL condition;
2. **Predictor step**. MUSCL reconstruction + slope limitation: from the discrete values U_K^n, compute a discrete gradient for each cell K.
3. Use a Lagrangian approximate Riemann solver to compute pressure fluxes $\boldsymbol{\pi}_A^n$ and interface velocities $\boldsymbol{u}_A^n$
4. Compute the upwind edge values $(U_\ell)_A^n$ according to the sign of $(\boldsymbol{u}_A^n \cdot \nu_A)$;
5. Compute the numerical flux Φ_A^n as defined in (12);
6. Compute the first order predicted states $U_K^{\star,n+1}$:

$$U_K^{\star,n+1} = U_K^n - \frac{\Delta t^n}{|K|} \sum_{A \subset \partial K} |A| \, \Phi_A^n$$

7. **Corrector step**. MUSCL reconstruction + slope limitation: from the discrete values $U_K^{\star,n+1}$, compute a discrete gradient for each cell K.
8. Use a Lagrangian approximate Riemann solver to compute pressure fluxes $\boldsymbol{\pi}_A^{\star,n+1}$ and interface velocities $\boldsymbol{u}_A^{\star,n+1}$
9. Compute the upwind edge values $(U_\ell)_A^{\star,n+1}$ according to the sign of $(\boldsymbol{u}_A^{\star,n+1} \cdot \nu_A)$;
10. Compute the numerical flux $\Phi_A^{\star,n+1}$ as defined in (12);
11. Compute the second-order accurate states U_K^{n+1} at time t^{n+1}:

$$U_K^{n+1} = U_K^n - \frac{\Delta t^n}{|K|} \sum_{A \subset \partial K} |A| \, \frac{\Phi_A^n + \Phi_A^{\star,n+1}}{2}.$$

One can appreciate the simplicity of the numerical solver.

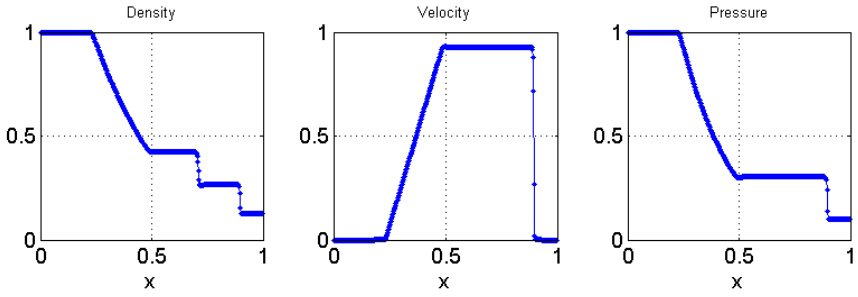


Figure 2: Second-order Lagrange-flux scheme on reference Sod's 1D shock tube problem. Time is $T = 0.23$, 384 mesh points. Here Sweby's slope limiter with coefficient 1.5 is used.

5.1 Details on the Lagrangian HLL approximate solver

A HLL approximate Riemann solver [12] in Lagrangian coordinates can be used to easily compute interface pressure and velocity. For a local Riemann problem made of a left state U_L and a right state U_R, the contact pressure $p^\star$ is given by the formula

$$p^\star = \frac{\rho_R p_L + \rho_L p_R}{\rho_L + \rho_R} - \frac{\rho_L \rho_R}{\rho_L + \rho_R} \max(c_l, c_R)\,(u_R - u_L), \tag{13}$$

and the normal contact velocity $u^\star$ by

$$u^\star = \frac{\rho_L u_L + \rho_R u_R}{\rho_L + \rho_R} - \frac{\rho_L \rho_R}{\rho_L + \rho_R} \frac{p_R - p_L}{\max(c_l, c_R)} \tag{14}$$

leading to very simple operations.

5.2 Numerical experiments

Shock tube problems. An as example, we test the Lagrange-flux scheme presented in section 5 on the reference one-dimensional Sod shock tube problem [13]. We use a Runge-Kutta 2 (RK2) time integrator and a MUSCL reconstruction using the second-order Sweby slope limiter [14]

$$\phi(a,b) = (ab > 0)\ \text{sign}(a)\ \max\big(\min(|a|, \beta|b|),\ \min(\beta|a|, |b|)\big)$$

with coefficient $\beta = 1.5$. We use a uniform grid made of 384 points. The final time is $T = 0.23$ and a CFL number equal to 0.25 . On figure 2, one can observe a good behaviour of the Eulerian solver, with rather sharp discontinuities and low numerical diffusion into rarefaction fans.

The second reference example is a case of two moving-away rarefaction fans under near-vacuum conditions (see Toro [12]). It is known that the Roe scheme breaks down for this case. The related Riemann problem is made of the left state $(\rho, u, p)_L = (1, -2, 0.4)$ and right state $(\rho, u, p)_R = (1, 2, 0.4)$. The final time of $T = 0.16$. We use 8192 mesh points to stress the behaviour on a refined mesh. Numerical results are given in figure 3. One can observe once again a nice behaviour of the method with a rather good accuracy.

Complex flow with high density ratios. In order to validate and compare both staggered Lagrange+remap and collocated Lagrange-flux, we propose the following original two-dimensional monofluid case. The computational domain is the rectangle $[0, L_x] \times [0, L_y]$ with $L_x = 3$ and $L_y = 2$. Boundary conditions are wall slip boundary conditions. The initial solution is a piecewise constant solution being constant into three different regions (see figure 4). Into

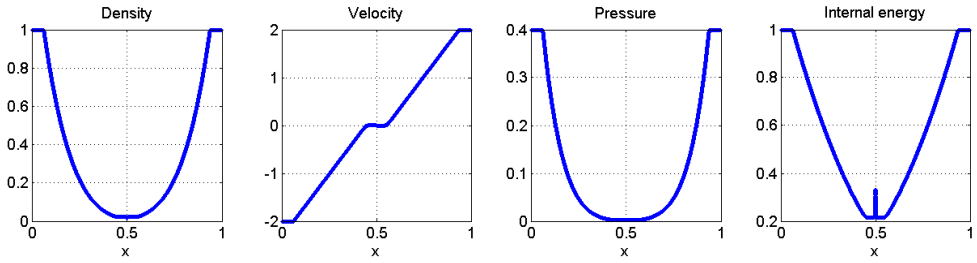


Figure 3: Second-order Lagrange-flux scheme on a double rarefaction near-vacuum case. Final time is $T = 0.16$, 8192 mesh points.

the region number 1 delimited by the rectangle $[0, 1] \times [1.1,\ 2]$, the solution is defined by $(\rho, u, v, p) = (1, 0, 0, 1)$. Into the region number 2 delimited by the rectangle $[0, 2] \times [0.9,\ 1.1]$, the solution is defined by $(\rho, u, v, p) = (100, 0, 0,\ 0.1)$. Into the last region number 3, we impose $(\rho, u, v, p) = (0.125, 0, 0,\ 0.1)$. A perfect gas law with $\gamma = 1.4$ is used. For spatial discretization, we use a rather fine Cartesian mesh made of 1200×400 points.

Into region nb 2, we have a dense cold fluid and a high density ratio between zone 2 and the two other zones. In region nb 1, we have a greater pressure that initiates a flow dynamics. The solution develops a complex structure made of several propagating shock waves, a low moving heavy fluid zone delimited by contact discontinuities, and a vortex of rarefied gas near the right border of the region 2. Numerical results between Lagrange-flux discretization versus staggered

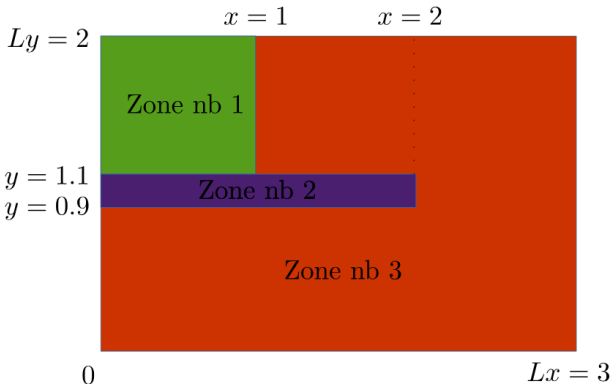
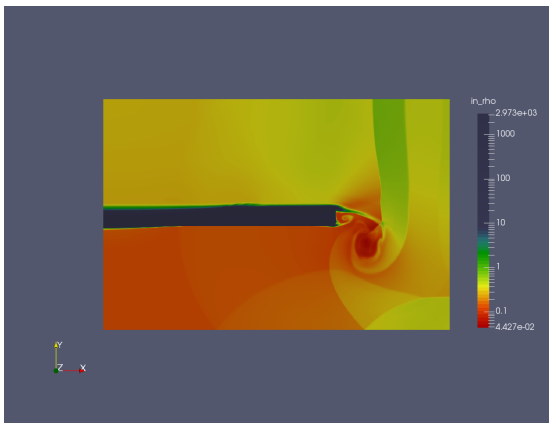


Figure 4: Geometry and initial conditions. All boundary conditions are wall slip conditions.

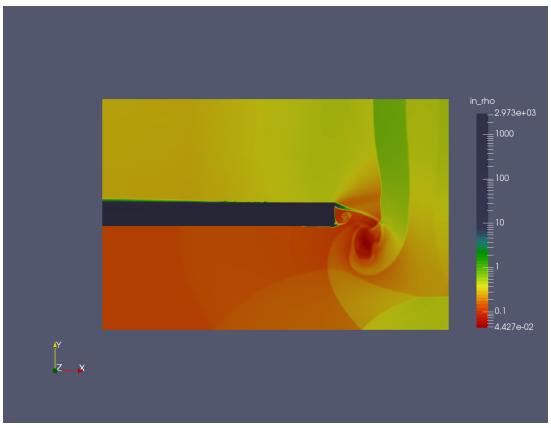
Lagrange+remap with splitted alternating directional remapping are shown in figure 5. One can observe that the density fields at snapshot time $t = 2$ sec are very comparable from the accuracy point of view. One can only notice a slight difference of capture of the contacts.

6 Performance results

In the "performance" companion paper [15], we propose a rigorous performance modeling approach to analyze the performance of both legacy Lagrange-remap solver and the Lagrange-flux scheme. It is indeed shown that performance results are far better using the Lagrange-flux scheme. The reformulated approach is also more suitable for SIMD and multi-core scalability. We refer to the companion paper [15] for interested readers.



(a) Lagrange-flux scheme, final time $t = 2$ sec. Density field, colormap in log scale.



(b) Legacy staggered Lagrange + splitted directional remapping, final time $t = 2$ sec. Density field, colormap in log scale.

Figure 5: Comparison of results between Lagrange-flux discretization vs staggered Lagrange+remap with splitted directional remapping.

7 Dealing with multimaterial flows

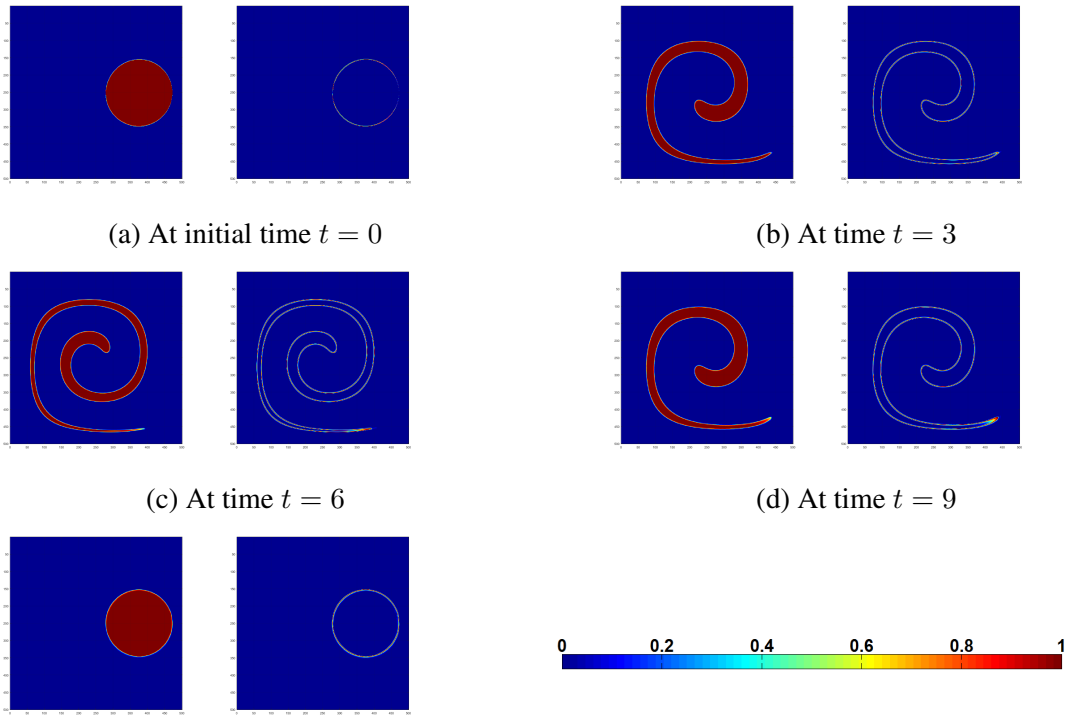
We would like to give an outline of the possible extension of Lagrange-flux schemes to compressible multimaterial/multifluid flows, i.e. flows that composed of different immiscible fluids and separated by free boundaries. Details of the multimaterial extension can be found into [18].

For pure Lagrange+remap schemes, usually VOF-based interface reconstruction (IR) algorithms are used (Young's PLIC, etc.). After the Lagrangian evolution, for the cells that host more than one fluid, fluid interfaces are reconstructed. During the remapping step, one has to evaluate the mass fluxes per material. From the computational point of view and computing performance, this process generally slows down the whole performance because of many array indirections in memory and specific treatment into mixed cells along with the material interfaces.

If the geometry of the Lagrangian cells is not completely known (as in the case of Lagrange-flux schemes), anyway we have to proceed differently. A possibility is to use interface capturing (IC) schemes, e.g. conservative Eulerian schemes that evaluate the convected mass fluxes through Eulerian cell edges. This can be achieved by the use of antidiffusive/low-diffusive advection solvers in the spirit of Després-Lagoutière's limited-downwind approach [16] or VoFire [17]. In a recent work [18], we have analysed the origin of known artefacts and numerical interface instabilities for this type of solvers and concluded that the reconstruction of fully multidimensional gradients with multidimensional gradient limiters was necessary. Thus, we decided to use low-diffusive advection schemes with a Multidimensional Limiting Process (MLP) in the spirit of [20]. The resulting method is quite accurate, shape-preserving and free from any artefact. We show some numerical results in the two next subsections. Let us emphasize that the interface capturing strategy perfectly fits with the Lagrange-flux flow description, and the resulting schemes are really suitable for vectorization (SIMD feature) with data coalescence into memory.

7.1 Interface capturing for pure advection problems

Let us first present numerical results on a pure scalar linear advection problem. The forward-backward advection case proposed by Rider and Kothe [21] is a hat-shaped function which is advected and stretched into a rotating vector field, leading to a filament structure. Then by applying the opposite velocity field, one have to retrieve the initial disk shape. In figure 6, we show the numerical solutions obtained on a grid 500^2 for both the scalar field (of variable $z \in [0,1]$) and the quantity $z(1-z)$ that indicates the spreads of the diffuse interface. One can conclude the very good behaviour of the method, providing both stability, accuracy and shape preservation.



(a) At initial time $t = 0$

(b) At time $t = 3$

(c) At time $t = 6$

(d) At time $t = 9$

(e) At final time $t = 12$

Figure 6: Validating the low-diffusive interface capturing scheme on the Kothe-Rider advection case, mesh 500^2.

7.2 Three-material hydrodynamics problem

We then consider our interface capturing method for multifluid hydrodynamics. Because material mass fractions are advected, that is

$$\partial_t y_k + \boldsymbol{u} \cdot \nabla y_k = 0, \quad k = 1, ..., N,$$

one can use the advection solver of these variables but we prefer a conservative form of the equations

$$\partial_t(\rho y_k) + \nabla \cdot (\rho y_k \boldsymbol{u}) = 0$$

in order to enforce mass conservation (see also [22]). The multimaterial Lagrange-flux scheme is tested on the reference "triple point" test case, found e.g. in Loubère et al. [23]. This problem is a three-state two-material 2D Riemann problem in a rectangular vessel. The simulation

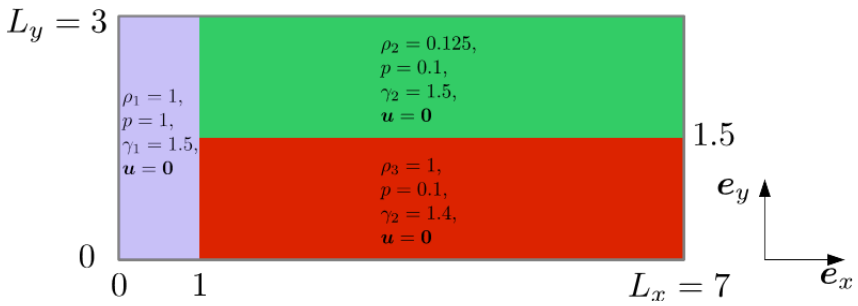
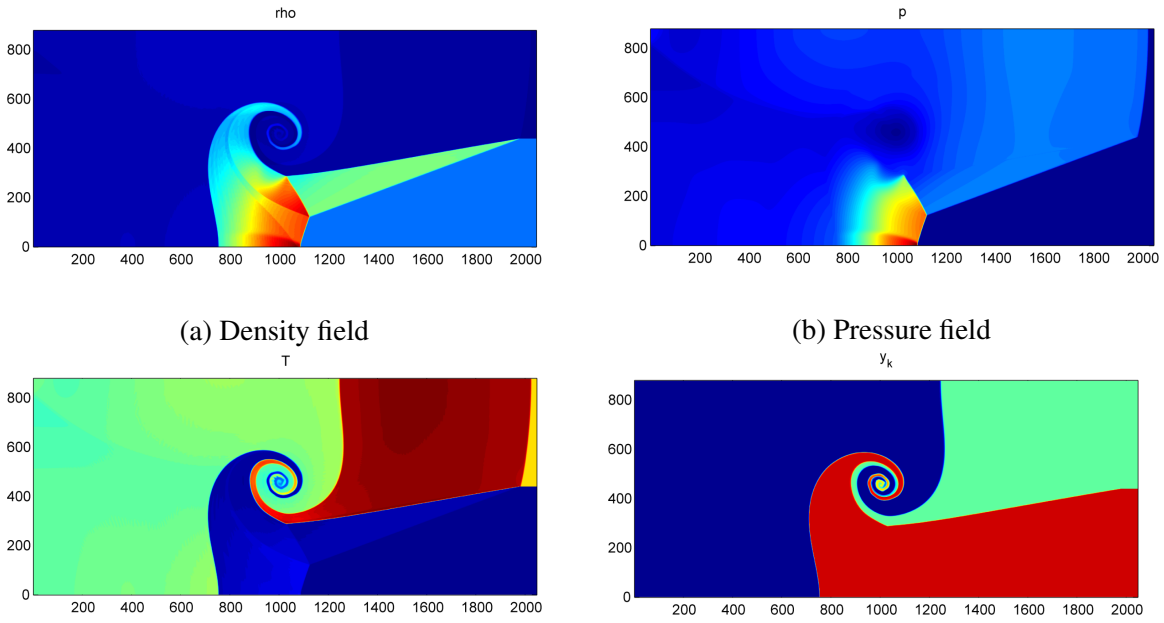


Figure 7: Geometry and initial configuration for the reference triple-point case.



(a) Density field

(b) Pressure field

(c) Temperature field

(d) Colored representation of material indicators

Figure 8: Results on the multimaterial "triple point" case (perfect gases) using a collocated Lagrange+remap solver + low-diffusive interface capturing advection scheme, mesh made of 2048×878 points. Final time is $T = 3.3530$.

domain is $\Omega = (0, 7) \times (0, 3)$ as described in figure 7. The domain is splitted up into three regions Ω_i, $i = 1, 2, 3$ filled with two perfect gases leading to a two-material problem. Perfect gas equations of state are used with $\gamma_1 = \gamma_3 = 1.5$ and $\gamma_2 = 1.4$. Due to the density differences, two shocks in sub-domains Ω_2 and Ω_3 propagate with different speeds. This create a shear along the initial contact discontinuity and the formation of a vorticity. Capturing the vorticity is of course the difficult part to compute. We use a rather fine mesh made of 2048×878 points (about 1.8M cells). One figure 8, we plot the density, pressure, temperature fields respectively and indicate the location of the three material zones. On can observe a nice capture of both shocks and contact discontinuities. The vortex is also captured accurately.

8 Concluding remarks and perspectives

This paper is firstly focused on the redesign of Lagrange-remap Hydrodynamics solvers in order to achieve better HPC node-based performance. We have reformulated the remapping step under a finite volume flux balance, allowing for a full SIMD algorithm. As an unin-

tended outcome, the analysis has lead us to the discovery of a new promising family of Eulerian solvers – the so-called Lagrange-flux solvers – that show simplicity, accuracy, and flexibility with a proven higher performance capability (see the "performance" ECCOMAS companion paper [15]). Interface capturing methods can be easily plugged-in for solving multimaterial flow problems.

REFERENCES

[1] R. Poncet, M. Peybernes, T. Gasc and F. De Vuyst, Performance modeling of a compressible hydrodnamics solver on multicore CPUs, Proceedings of the int. conf. on parallel computing PARCO2015, Edinburgh, 2015 (in press).

[2] C.W. Hirt, A.A. Amsden and J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics, 14:227–253 (1974).

[3] S. Williams, A. Waterman, and D. Patterson: Roofline: An Insightful Visual Performance Model for Multicore Architectures, Commun. ACM, 52, pp 6576 (2009).

[4] J. Treibig and G. Hager, Introducing a Performance Model for Bandwidth-Limited Loop Kernels. Proceedings of the Workshop Memory issues on Multi- and Manycore Platforms at PPAM 2009, Lecture Notes in Computer Science, 6067, pp. 615–624 (2010).

[5] H. Stengel, J. Treibig, G. Hager and G. Wellein, Quantifying performance bottlenecks of stencil computations using the Execution-Cache-Memory model. Proc. ICS15, the 29th Int. Conf. on Supercomputing, 2015, DOI: 10.1145/2751205.2751240.

[6] P. Colella and P.R. Woodward, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys.,54:115–173 (1984).

[7] B. Després and C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Rational Mech. Anal. 178 (2005) 327–372.

[8] P.-H. Maire, R. Abgrall, J. Breil and J. Ovadia, A cell-centered Lagrangian scheme for compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.

[9] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (2009) 2391–2425.

[10] J. K. Dukowicz and J. R. Baumgardner, Incremental remapping as a transport/advection algorithm. J. Comput. Phys., 160, 318335 (2000).

[11] W. E. Schiesser, The Numerical Method of Lines, Academic Press, ISBN 0-12-624130-9 (1991).

[12] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics, 3rd Edition, Springer (2010).

[13] G.A. Sod, A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws" . J. Comput. Phys. 27: 1–31 (1971).

[14] P.K. Sweby, High resolution schemes using flux-limiters for hyperbolic conservation laws, SIAM J. Num. Anal. 21 (5): 995–1011 (1984).

[15] T. Gasc, F. De Vuyst, M. Peybernes, R. Poncet, R. Motte, Building a more efficient Lagrange-remap scheme thanks to performance modeling, this conference, Paper P12210, in Minisymposium "MS 414 - New trends in numerical methods for multi-material compressible fluid flows", Proc. of the Conference ECCOMAS 2016, submitted.

[16] B. Després and F. Lagoutière, Contact discontinuity capturing schemes for linear advection and compressible gas dynamics, J. Sci. Comp., 16(4), 479–524 (2001).

[17] B. Després, F. Lagoutière, E. Labourasse and I. Marmajou, An antidissipative transport scheme on unstructured meshes for multicomponent flows, IJFV, 30–65 (2010).

[18] F. De Vuyst, M. Béchereau, T. Gasc, R. Motte, M. Peybernes and R. Poncet, Stable and accurate low-diffusive interface capturing advection schemes, submitted to IJNMF, Special Issue of the MULTIMAT 2015 Conf. Würsburg, of the (2016).

[19] F. De Vuyst, T. Gasc, R. Motte, M. Peybernes and R. Poncet, Lagrange-flux schemes: reformulating second-order accurate Lagrange-remap schemes for better node-based HPC performance, submitted to Oil & Gas Science and Technologies OGST (2016).

[20] J.S. Park and C. Kim, Multi-dimensional Limiting Process for Discontinuous Galerkin Methods on Unstructured Grids, chapter in Computational Fluid Dynamics 2010, Springer, 179–184 (2011).

[21] A.J. Rider and D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys., 141(2), 112–152 (1998).

[22] A. Bernard-Champmartin and F. De Vuyst, A low diffusive Lagrange-remap scheme for the simulation of violent air-water free-surface flows, J. of Comput. Physics, 274, 19–49 (2014).

[23] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil and S. Galera, ReALE: A reconnection-based arbitrary-Lagrangian-Eulerian method, J. Comp. Phys., 229, 4724–4761 (2010).