



PROJET DE FIN D'ÉTUDES

A scalable Lagrange-Remap scheme for compressible multimaterial Euler equations with sharp interface reconstruction

Author:
Bastien CHAUDET

Supervisors:
Dr. Jean-Philippe BRAEUNIG (CEA)
Pr. Christophe HAZARD (ENSTA)

THIS REPORT IS NOT CONFIDENTIAL.

Internship at the CEA, DAM, DIF, F-91297 Arpajon Cedex.

From June to November 2014

Certificate of non confidentiality

The present report is not confidential. Accordingly, any disclosure of its content in paper and/or electronic format to third parties is allowed.

Acknowledgments

First of all, I would like to thank Stéphane Bernard and Jean-Philippe Perlat for greeting me in their service and laboratory, respectively.

I also thank Patrick Le Tallec and Florian De Vuyst for their pieces of advise, and for giving time to fruitful discussions.

I am particularly grateful to my supervisor Jean-Philippe Braeunig for his unchanging help and involvement, for his availability, and his support. His ideas and the clarity of his explanations have greatly contributed to the work presented in this report.

Thanks to Mathieu Peybernes and Raphaël Poncet for providing such a clear numerical code, and for answering all my questions.

Abstract

This work is in the field of multi-material compressible fluid flows simulation. The proposed scheme is eulerian and related to finite volumes methods, but in a Lagrange-Remap formalism on regular orthogonal meshes. The Lagrangian scheme is staggered and the remap phase is similar to a finite volume advection scheme. The multi-material extension uses classical VOF fluxes for sharp interface reconstruction. The originality of the scheme is in the attempt for a 9 points remap scheme without directional splitting. This strategy should allow to preserve good properties of classical multi-material staggered schemes, while saving parallel communications with the one step remap. Results will be discussed and compared to those from classical Lagrange-Remap schemes on severe benchmarks.

Keywords: Lagrange-Remap, Euler equations, multi-fluid flows, VOF, directional splitting, HPC, hydrodynamics, numerical analysis

Contents

Abstract	1
Introduction	7
1 Lagrangian phase	10
1.1 Lagrangian formalism	10
1.2 Discretization	10
2 Remap phase	15
2.1 Alternate Directions Remap (AD)	16
2.1.1 <i>X</i> -Remap phase	16
2.1.2 <i>Y</i> -Remap phase	19
2.2 Direct Remap	21
2.3 Direct Remap with Corner Fluxes	22
3 Specificities of the Direct Remap with Corner Fluxes	27
3.1 Specific case: linear advection with first order reconstruction for the remap	27
3.2 Accuracy of the geometric representation for non-linear deformations	28
3.3 Impact of the corners on the scheme	31
4 Multi-material extension	38
4.1 Lagrangian phase	38
4.2 Alternate Directions Remap	40
4.3 Direct Remap	42
4.4 Direct Remap with Corner Fluxes	43
5 Numerical results	46
5.1 First observations and convergence	46
5.1.1 Mono-material	46
5.1.2 Multi-material	48
5.2 Robustness	49

Conclusion	55
A Annexe: Étude de la diffusion de la vorticité en hydrodynamique par différents types de schémas eulériens	57
References	78

List of Figures

1	Notations of indices in cell $c = (i, j)$	11
2	Beginning (left) and end (right) of prediction phase.	13
3	Beginning (left) and end (right) of correction phase.	13
4	Lagrangian and Eulerian meshes.	15
5	Lagrangian displacement of cell $c = (i, j)$	16
6	Computation of volume fluxes at X -faces.	17
7	Primary and dual meshes around node p	19
8	Computation of volume fluxes at Y -faces.	20
9	Stencils of 5 points remaps (left) and 9 points remaps (right).	21
10	Computation of volume fluxes at both X and Y -faces.	21
11	Definition of corner volume fluxes.	23
12	Definition of volume fluxes at both X and Y -faces.	24
13	Lagrangian representation of the volume.	25
14	Definition of corner fluxes on the dual mesh.	26
15	Lagrangian displacement of one single node.	29
16	Representation of the Lagrangian volume in the exact (top-left), AD (top-right), Direct (bottom-left) and DirectCF (bottom-right) cases.	30
17	Mass corner flux in the exact (left), Direct (centre) and DirectCF (right) cases.	31
18	Mass face fluxes at X (top) and Y (bottom) remap phases in the AD case.	31
19	Comparison between upwind, average, linear x - y and linear diagonal reconstructions at the corners (from left to right).	36
20	Stencils of both x - y (left) and diagonal (right) linear reconstructions.	37
21	Multi-material sharp interface model.	40
22	X -Remap. Intersection between the interface and the volume fluxes.	41
23	Y -Remap. Intersection between the interface and the volume fluxes.	42
24	Rectangular approximation of Lagrangian cell.	43
25	Placement of Lagrangian interface.	44
26	DirectCF remap. Intersection between the interface and volume fluxes.	44
27	Monomat advection. Direct (left), AD (centre) and DirectCF (right).	46
28	Monomat advection. $\log \ \rho^{final} - \rho^0\ _{L^2}$ in function of $\log(\Delta x)$	47

29	Monomat rotation. Direct (left), AD (centre) and DirectCF (right).	47
30	Monomat rotation. $\log \ \rho^{final} - \rho^0\ _{L^2}$ in function of $\log(\Delta x)$	48
31	Multimat advection. Direct (left), AD (centre) and DirectCF (right).	48
32	Multimat advection. $\log \ k_1^{final} - k_1^0\ _{L^2}$ in function of $\log(\Delta x)$	49
33	Volume fraction at $t = 1$ ms. AD (left) and DirectCF (right).	50
34	Density at initial state, $t = 0$ s. AD top, DirectCF bottom.	50
35	Density at final state, $t = 1$ ms. AD top, DirectCF bottom.	51
36	Zoom, density at final state, $t = 1$ ms. AD left, DirectCF right.	51
37	Volume fraction at $t = 0, 3.5, 4.5$ and 6 ms. AD left, DirectCF right.	52
38	Zoom, volume fraction at final state, $t = 6$ ms. AD left, DirectCF right.	53
39	Schéma de la maille sur laquelle on définit $\text{rot}(\mathbf{u})$ pour les schémas décalé, centré et BBC (de gauche à droite et de haut en bas).	61
40	Schéma du champ de vitesses sur le maillage dans le cas décalé.	63
41	Schéma de la déformation lagrangienne dans le cas décalé pour un vortex ponctuel.	63
42	Schéma du champ de vitesses sur le maillage dans le cas centré.	66
43	Schéma de la déformation lagrangienne dans le cas centré pour un vortex ponctuel.	67
44	Schéma du champ de vitesses sur le maillage pour les schémas BBC.	68
45	Schéma de la déformation lagrangienne dans le cas décalé pour un vortex idéal.	72
46	Schéma de la déformation lagrangienne dans le cas centré pour un vortex idéal.	74

Introduction

Fluid mechanics, especially hydrodynamics, has first been conceptualized by L. Euler in 1755 when he introduced the equation of perfect fluids $\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla P$. Then it has been generalized by C. Navier and G. Stokes by taking into account the viscosity of the fluid in the equation. In the framework of this internship, we got interested in compressible Euler equations, neglecting the viscous term. L. Euler's approach was to study these equations into a fixed referential, unlike J. L. Lagrange, who took the point of view of a referential following the matter during its displacement.

Those points of view can be extended to the numerical resolution of these equations. Indeed, Lagrangian simulation codes are implemented such that the mesh moves with the fluid. They are often preferred to Eulerian simulation codes because they naturally refine the mesh around zones of interest (high compression zones, shocks, etc) and give more faithful retranscriptions of contact discontinuities. However, they are not really robust to vorticity, since the mesh tends to deform with the fluid. An alternative can be found in ALE (**A**rbitrary **L**agrangian **E**ulerian) schemes. The difference between ALE and pure Lagrangian schemes comes from the remap phase. Indeed, in order to fix the problems of stability and robustness of Lagrangian schemes when vortices appear in the simulation, Lagrangian mesh and variables are remapped onto a regularized mesh at each time step. This avoids to get tangled cells, while saving the interesting geometric properties of Lagrangian schemes. The specific case of remapping onto the initial Eulerian mesh at each time step is the Lagrange-Remap formalism. This formalism constitutes the basis of the work presented in this report.

The extension to multi-material models enables to simulate much more complex flows. There are several physical and numerical models for multi-fluid flows, more or less diffusive, depending on the miscibility of the fluids in the physical case, and on the properties of the scheme. A numerical model adapted to the simulation of miscible fluids is the mixing model: mixed cells, i.e. cells containing two species, are represented with one single pressure, and a mass concentration per material. Another numerical multi-material model is the reconstruction of a sharp interface between the two fluids, which of course models non-miscible fluids, and multiphase flows. The (discontinuous) interface is represented as a segment in each mixed-cell, and is placed with volume fractions and the normal vector to this segment.

In the framework of HPC (**H**igh **P**erformance **C**omputing), a trend in numerical analysis is to build numerical schemes adapted to exascale computers constraints, i.e. using a lot of computational cores. Two of the main requirements for a scalable scheme are to include kernels with high arithmetic intensities, and to reduce

as much as possible the number of communications needed when parallelizing with MPI (**M**essage **P**assing **I**nterface). The performances of a code using many cores can be restricted significantly if such features are not taken into account. Still regarding HPC issues, it is often important to reduce the complexity of algorithms. Here, we consider fixed Eulerian regular orthogonal grids.

There exists one type of remap used in most codes: the AD (**A**lternate **D**irections) remap. It is known to be quite accurate and robust but it shows one main drawback: it does not seem very adapted to HPC. To be more specific, in the framework of MPI parallelization, since the remap is processed in two steps, the number of communications is high. An alternative remap - the Direct remap -, using the same fluxes but remapping in one step, has been proposed by Q. Debray and J.-P. Braeunig in 2013, see [4].

The work presented in this report follows on from the latter, and aims at testing a new type of remap which is more adapted to HPC, and incorporating it in a 2D hydrodynamics Lagrange-Remap scheme. The main idea is still to perform the remap in one step, using the same tools as the AD remap, while recovering at least the same robustness and accuracy as the AD remap.

The work done during this internship includes the theoretical study of this new remap (Direct with Corner Fluxes), getting familiar with SHY code, a mono-material Lagrange-Remap research code with an AD remap provided by the CEA (M. Peybernes and R. Poncet), the implementation of Direct and Direct with Corner Fluxes remaps in SHY, and the extension to two-materials for all remaps: theoretical study and implementation of two different multi-fluid models. An additional theoretical study on the diffusion of some hydrodynamics eulerian schemes has been conducted during the first two months, that took place at the LRC-MESO (**L**aboratoire de **R**echerche **C**onventionné **M**odélisation **M**ESOscopique), CMLA (**C**entre de **M**athématiques et **L**eurs **A**pplications), ENS Cachan. A report of this study can be found in the Annexe.

First, the Lagrangian phase will be presented, writing both continuous and discrete systems of equations. Then, the principles of three different types of remap: the AD remap, the Direct remap and the remap studied in this work (the Direct Remap with Corner Fluxes), will be explained. The Direct Remap with Corner Fluxes will be analysed in further details, before coming to the multi-material extension of the three schemes. Finally, several results of convergence and robustness will conclude this report.

1 Lagrangian phase

1.1 Lagrangian formalism

The Lagrangian formalism consists in solving the equations in the matter referential. At this aim, a special derivative has been introduced: the material derivative $d_t = \partial_t + \mathbf{u} \cdot \nabla_x$, where \mathbf{u} denotes the velocity of the fluid particle. Let the classical thermodynamical quantities be: ρ (density), e (specific internal energy) and P (pressure). A term of artificial viscosity Q , called Wilkins pseudo-viscosity, see [12], is added in order to remain entropic. It makes the scheme more stable by spreading shocks and regularizing the solution at its neighbourhoods. This continuity is used to recover local differential equations from integral ones.

With these notations, in this formalism, the equations to be solved on the moving domain $\Omega(t)$ write:

$$\left\{ \begin{array}{l} \frac{d}{dt} \left(\int_{\Omega(t)} \rho \, dx \right) = 0 \\ \frac{d}{dt} \mathbf{x} = \mathbf{u} \\ \rho \frac{d}{dt} \mathbf{u} = -\nabla(P + Q) \\ \frac{d}{dt} e = -(P + Q) \frac{d}{dt} \left(\frac{1}{\rho} \right) \\ P = \mathcal{P}(\rho, e) \end{array} \right. \quad (1)$$

The Wilkins pseudo-viscosity is the following:

$$Q = a_1 \rho c |\Delta \mathbf{u}| + a_2 \rho |\Delta \mathbf{u}|^2$$

where $\Delta \mathbf{u}$ is the velocity gap in the direction of the shock, a_1 , a_2 are two real numbers called respectively the linear and quadratic pseudo-viscosity coefficients, and c is the sound speed.

Let us now proceed to both space and time discretizations of system (1).

1.2 Discretization

A SGPC (**S**taggered **G**rid **P**redictor **C**orrector) scheme is used. Space discretization is based on Wilkins', which is second order in space, see [11]. It is a predictor corrector scheme in time, which is second order as well.

We denote by Δx and Δy the space steps in the two directions, and Δt^n the time step at t^n (iteration n). The prediction phase computes values at $t^{n+1/2} = t^n + \Delta t^n/2$. Cell-centred variables and node-centred variables are written respectively in cell $c = (i, j)$ in $(\Delta x, \Delta y)$ basis, and at node $p = (i \pm 1/2, j \pm 1/2)$, see Figure 1.

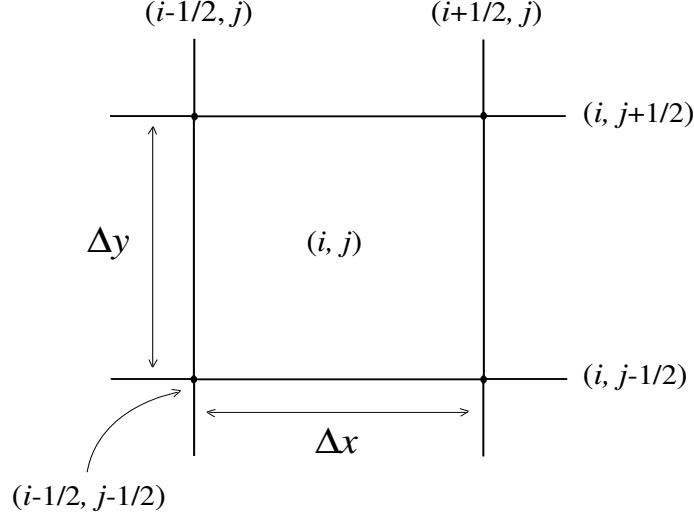


Figure 1: Notations of indices in cell $c = (i, j)$.

The domain $\Omega(t^n)$ is splitted up into rectangular cells like the one represented in Figure 1, i.e. $\Omega(t^n) = \bigcup \Omega_c^n$. The whole mesh is deformed with respect to the movement of the fluid, that is, the matter contained into Ω_c^n at t^n is to be found in $\Omega_c^{n+1, lag}$ at the end of the Lagrangian phase. In particular, the mass in each moving subdomain Ω_c remains constant. The discrete system of equation writes, $\forall p, c$:

Prediction phase

$$\left\{ \begin{array}{l} m_c^{n+1/2} = m_c^n \\ \mathbf{x}_p^{n+1/2} = \mathbf{x}_p^n + \frac{\Delta t^n}{2} \mathbf{u}_p^n \\ Vol_c^{n+1/2} = \mathcal{F}(\mathbf{x}_p^{n+1/2})_{p \in \{c\}} \\ \mathbf{u}_p^{n+1/2} = \mathbf{u}_p^n - \frac{\Delta t^n}{2} (\nabla(P + Q))_p^{n+1/2} / \rho_p^n \\ e_c^{n+1/2} = e_c^n - (P_c^n + Q_c^n) \left(\frac{1}{\rho_c^{n+1/2}} - \frac{1}{\rho_c^n} \right) \\ P_c^{n+1/2} = \mathcal{P}(\rho_c^{n+1/2}, e_c^{n+1/2}) \end{array} \right. \quad (2)$$

$$\text{where } \mathcal{F}((\mathbf{x}_p)_{p \in \{c\}}) = \frac{1}{2} \left\| \Delta \mathbf{x}_{i-1/2, j} \wedge \Delta \mathbf{x}_{i, j-1/2} \right\| \\ + \frac{1}{2} \left\| \Delta \mathbf{x}_{i+1/2, j} \wedge \Delta \mathbf{x}_{i, j+1/2} \right\|$$

$$\text{with } \Delta \mathbf{x}_{i \pm 1/2, j} = \mathbf{x}_{i \pm 1/2, j+1/2} - \mathbf{x}_{i \pm 1/2, j-1/2}$$

$$\text{and } \Delta \mathbf{x}_{i, j \pm 1/2} = \mathbf{x}_{i+1/2, j \pm 1/2} - \mathbf{x}_{i-1/2, j \pm 1/2}$$

Moreover, we have, at node $p = (i - 1/2, j - 1/2)$:

$$\left(\nabla(P + Q) \right)_p^{n+1/2} \cdot \mathbf{e}_x = \frac{1}{2 \Delta x} \left[\left(P_{i, j-1}^{n+1/2} + Q_{i, j-1}^n + P_{i, j}^{n+1/2} + Q_{i, j}^n \right) \right. \\ \left. - \left(P_{i-1, j-1}^{n+1/2} + Q_{i-1, j-1}^n + P_{i-1, j}^{n+1/2} + Q_{i-1, j}^n \right) \right] \\ \left(\nabla(P + Q) \right)_p^{n+1/2} \cdot \mathbf{e}_y = \frac{1}{2 \Delta y} \left[\left(P_{i-1, j}^{n+1/2} + Q_{i-1, j}^n + P_{i, j}^{n+1/2} + Q_{i, j}^n \right) \right. \\ \left. - \left(P_{i-1, j-1}^{n+1/2} + Q_{i-1, j-1}^n + P_{i, j-1}^{n+1/2} + Q_{i, j-1}^n \right) \right]$$

where nodal quantities (mass m_p^n and density ρ_p^n) are defined as follows

$$\rho_p^n = \frac{m_p^n}{\Delta x \Delta y}, \quad \text{with } m_p^n = \frac{1}{4} \sum_{c \in \{p\}} m_c^n.$$

In order to compute the discrete pseudo-viscosity, the value of $\Delta \mathbf{u}$ is approximated by $L \operatorname{div} \mathbf{u}$, $L = \sqrt{\Delta x \Delta y}$ being the characteristic length of a cell, which leads to:

$$Q_c^n = \begin{cases} \rho_c^n a_1 L c_c^n |(\operatorname{div} \mathbf{u})_c^n| + a_2 L^2 |(\operatorname{div} \mathbf{u})_c^n|^2 & \text{if } (\operatorname{div} \mathbf{u})_c^n < 0, \\ 0 & \text{else.} \end{cases}$$

The values of $\mathbf{u}_p^{n+1/2}$ enable to perform the Lagrangian displacement of the mesh, see Figure 2. Then the correction phase uses the values at $t^{n+1/2}$ to express Lagrangian variables, i.e. variables at the centres and nodes of the Lagrangian deformed mesh, see Figure 3. The end of the Lagrangian phase is denoted $t^{n+1, lag} = t^n + \Delta t^n$.

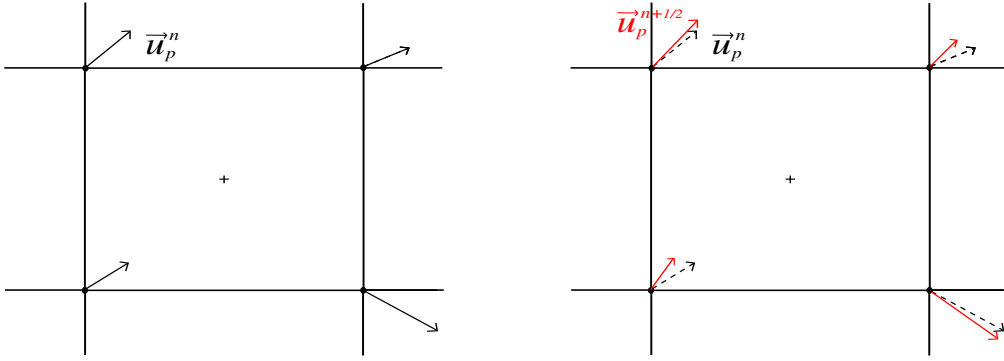


Figure 2: Beginning (left) and end (right) of prediction phase.

Correction phase

$$\left\{ \begin{array}{l} m_c^{n+1, lag} = m_c^n \\ \mathbf{x}_p^{n+1, lag} = \mathbf{x}_p^n + \Delta t^n \mathbf{u}_p^{n+1/2} \\ Vol_c^{n+1, lag} = \mathcal{F}((\mathbf{x}_p^{n+1})_{p \in \{c\}}) \\ \mathbf{u}_p^{n+1, lag} = 2 \mathbf{u}_p^{n+1/2} - \mathbf{u}_p^n \\ e_c^{n+1, lag} = e_c^n - (P_c^{n+1/2} + Q_c^n) \left(\frac{1}{\rho_c^{n+1, lag}} - \frac{1}{\rho_c^n} \right) \\ P_c^{n+1, lag} = \mathcal{P}(\rho_c^{n+1, lag}, e_c^{n+1, lag}) \end{array} \right. \quad (3)$$

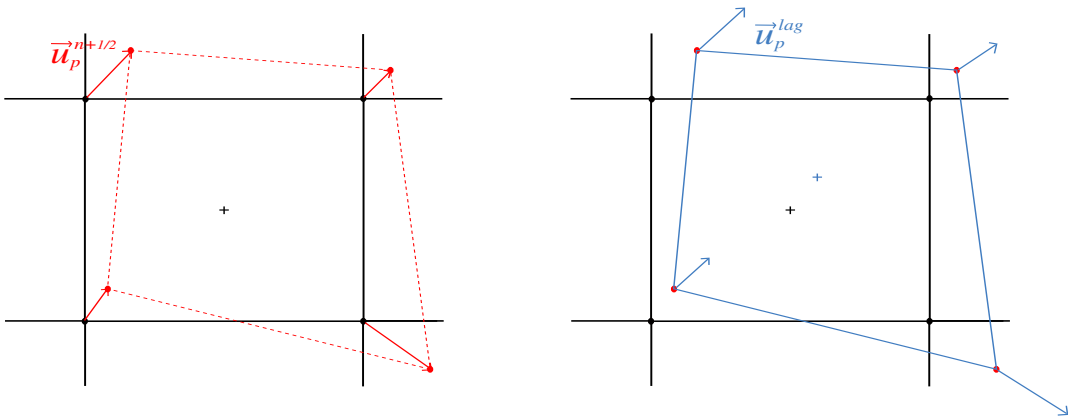


Figure 3: Beginning (left) and end (right) of correction phase.

The two successive phases enable to get the second order in time. Plus, the scheme is entropic (under a CFL type condition) thanks to the pseudo-viscosity. It is also conservative in mass (locally and globally). But regarding total energy or momentum, it is not so clear since one cannot define rigorously those quantities because of the node-centred velocities. However, there exist a way to define both quantities such that the scheme is conservative in momentum, and in total energy as well, but with an error in $O(\Delta t^2)$. For further details about the computations, see [3].

2 Remap phase

The principle is the following: the mesh moves during the Lagrangian phase and the values of Lagrangian variables are computed, then new values of ρ , e and \mathbf{u} on the Eulerian mesh are interpolated from values on the Lagrangian mesh. In Figure 4, it means that one has to interpolate values at black crosses and dots from values at blue crosses and dots. This phase is only a geometric interpolation, all equations ruling physical properties have been solved numerically during the Lagrangian phase.

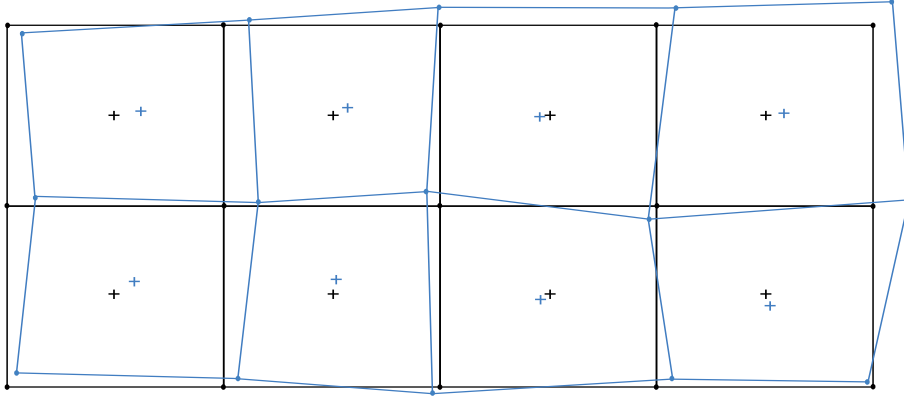


Figure 4: Lagrangian and Eulerian meshes.

Variables are remapped (interpolated) thanks to a Finite-Volume type remap scheme. That is to say, volume fluxes at the faces of the Eulerian mesh are computed from the Lagrangian displacement of the nodes, which enables to get variable fluxes and finally values after the remap phase in a conservative formalism.

Somehow, this second phase can be seen as a pure advection of Lagrangian variables, which explains the Finite-Volume type scheme. The remap phase is often called advection phase.

In the following sections, several types of remap will be presented. The Lagrangian displacement of one reference cell $c = (i, j)$, knowing the velocities at each node, is represented Figure 5. At this step, the Lagrangian phase is over, i.e. the values of variables at $t^{n+1, lag}$ have been computed from those at t^n . From now, the exponent $n+1, lag$ will be replaced by lag in order to simplify notations.

Black lines define the fixed Eulerian mesh, whereas red dotted lines define the faces of the Lagrangian cell.

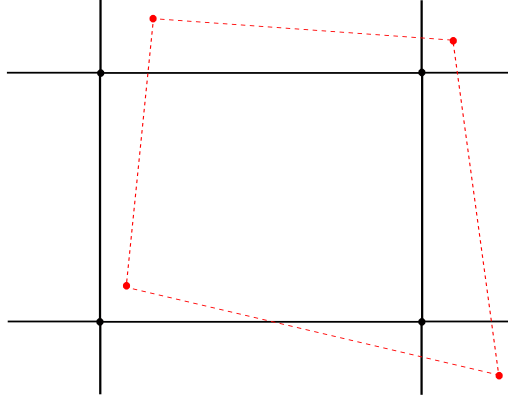


Figure 5: Lagrangian displacement of cell $c = (i, j)$.

2.1 Alternate Directions Remap (AD)

As mentioned before, two successive steps (remaps) are required to process the full AD remap. First, only x components of the velocities are taken into account to compute fluxes at all X -faces (i.e. vertical faces), and remap all variables. Second, only y components of the velocities are taken into account to compute fluxes at all Y -faces (i.e. horizontal faces), and remap all variables once again. The name "Alternate Directions" comes from this X - Y alternating.

2.1.1 X -Remap phase

Let us first consider the Lagrangian displacement along X , computed from horizontal node predicted velocities $\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_x$, according to (3), see Figure 6. Blue dots define the positions of X -Lagrangian nodes. Thus, the volume fluxes considered by this X -remap are the ones delimited by blue dotted lines. They are approximated by the blue rectangles, drawn from the mid-points of those blue dotted segments.

Formally, in cell (i, j) , the respective volume fluxes at left ($l = i - 1/2, j$) and right ($r = i + 1/2, j$) faces write:

$$dVol_{i-1/2,j} = \mathbf{u}_{i-1/2,j}^{n+1/2} \cdot \mathbf{e}_x \Delta t^n \Delta y := \frac{1}{2} \left(\mathbf{u}_{i-1/2,j-1/2}^{n+1/2} + \mathbf{u}_{i-1/2,j+1/2}^{n+1/2} \right) \cdot \mathbf{e}_x \Delta t^n \Delta y$$

$$dVol_{i+1/2,j} = \mathbf{u}_{i+1/2,j}^{n+1/2} \cdot \mathbf{e}_x \Delta t^n \Delta y := \frac{1}{2} \left(\mathbf{u}_{i+1/2,j-1/2}^{n+1/2} + \mathbf{u}_{i+1/2,j+1/2}^{n+1/2} \right) \cdot \mathbf{e}_x \Delta t^n \Delta y$$

Those values enable to compute a 1D representation of the Lagrangian volume and density at this step (denoted t^{lagx}) as follows, which will be the approximation of the X -remap:

$$Vol^{lagx} = Vol^n + dVol_l - dVol_r,$$

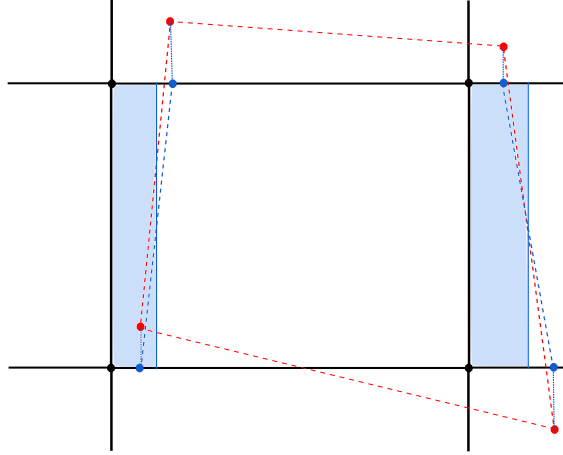


Figure 6: Computation of volume fluxes at X -faces.

$$\rho^{lagx} = \rho^n \frac{Vol^n}{Vol^{lagx}} \quad \text{and} \quad e^{lagx} = e^{lag}.$$

Let us now compute the variable fluxes at each face. Given the volume flux at a face, one only has to define a value of the variable at this face to get the variable flux. Such a definition can be based on first or second order reconstruction (or even higher order but this will not be discussed in the present work).

First order: upwind reconstruction

Let a be the cell-centred variable on the Lagrangian mesh (ρ , e , or \mathbf{u} on the Lagrangian dual mesh) to reconstruct on the Eulerian one. At X -face $l = (i-1/2, j)$, the upwind reconstruction writes:

$$a_{i-1/2,j}^{upw} = a_c \quad \text{where} \quad c = \begin{cases} (i-1, j) & \text{if } dVol_{i-1/2,j} > 0 \\ (i, j) & \text{if } dVol_{i-1/2,j} < 0 \end{cases}$$

Second order: linear reconstruction

In order to build a second order scheme, the remap phase needs to be second order as well. But second order spatial reconstruction implies the computation of a gradient. Since the solution can be discontinuous, the use of slope limiters when computing discrete gradient is necessary, and avoids to get spurious oscillations. In mathematical terms, taking c defined as in the upwind reconstruction, the value of a at X -face $l = (i-1/2, j)$ is:

$$a_{i-1/2,j}^{o2} = a_c + \frac{\delta_x^{o2} a_c}{2} \left(\text{sgn}(dVol_{i-1/2,j}) \Delta x_c^{lag} - \Delta t^n \mathbf{u}_{i-1/2,j}^{n+1/2} \cdot \mathbf{e}_x \right)$$

where $\Delta x_c^{lag} = \Delta x + \Delta t^n (\mathbf{u}_l^{n+1/2} - \mathbf{u}_r^{n+1/2}) \cdot \mathbf{e}_x$ is the width of the Lagrangian cell, and $\delta_x^{o2} a_c$ represents the limited value of the gradient in cell c . For example, if $c = (i, j)$, and x_c^{lag} denotes the x coordinate of the centre of Lagrangian cell c :

$$\delta_x^{o2} a_{i,j} = \frac{1}{2} \left(\Phi_{i,j}^+ \delta_x a_{i+1/2,j} + \Phi_{i,j}^- \delta_x a_{i-1/2,j} \right)$$

$$\text{with } \delta_x a_{i\pm 1/2,j} = \frac{a_{i\pm 1,j} - a_{i,j}}{x_{i\pm 1,j}^{lag} - x_{i,j}^{lag}}$$

$$\text{and } \Phi_{i,j}^+ = \varphi(r), \quad \Phi_{i,j}^- = \varphi(1/r), \quad r = \frac{\delta_x a_{i-1/2,j}}{\delta_x a_{i+1/2,j}}$$

The limiter function φ makes the scheme degenerate to order 1 when strong gradients are detected, see [3] for further details. The choice made here is Van Leer limiter:

$$\varphi(r) = \frac{r + |r|}{1 + r}$$

Now the reconstructed values at X -faces have been determined, variable fluxes can be computed easily, and variables can be remapped. This is the X -remap phase, denoted t^{projx} . According to previous notations, in cell c , the conservation of cell-centred variables leads to:

$$m_c^{projx} = m_c^{lagx} + \rho_l^{lagx o2} dVol_l - \rho_r^{lagx o2} dVol_r$$

$$m_c^{projx} e_c^{projx} = m_c^{lagx} e_c^{lagx} + \rho_l^{lagx o2} e_l^{lagx o2} dVol_l - \rho_r^{lagx o2} e_r^{lagx o2} dVol_r$$

Remark: Instead of reconstructing the quantity ρe at the faces, the variables ρ and e are independently reconstructed, for robustness issues.

Regarding node-centred variables, i.e. x and y components of the velocities, the principle is exactly the same, but on the dual mesh. That is to say, variables are reconstructed (at first or second order) at the faces of the dual mesh, and then remapped. The nodes of the dual mesh are defined as the centres of the primary mesh, see Figure 7. The black lines and dots respectively draw the primary and dual meshes. Here, in the regular orthogonal case, the dual mesh remains regular orthogonal.

Let us use the index p to denote objects related to the dual cell (also called nodal cell) containing node p . For example, the faces of this dual cell p are the f_p . The nodal mass and mass fluxes at its faces write:

$$m_p = \frac{1}{4} \sum_{cell \in \{p\}} m_{cell}$$

$$dm_{f_p} = \frac{1}{4} \sum_{cell \in \{p\}} (dm_f)_{cell}$$

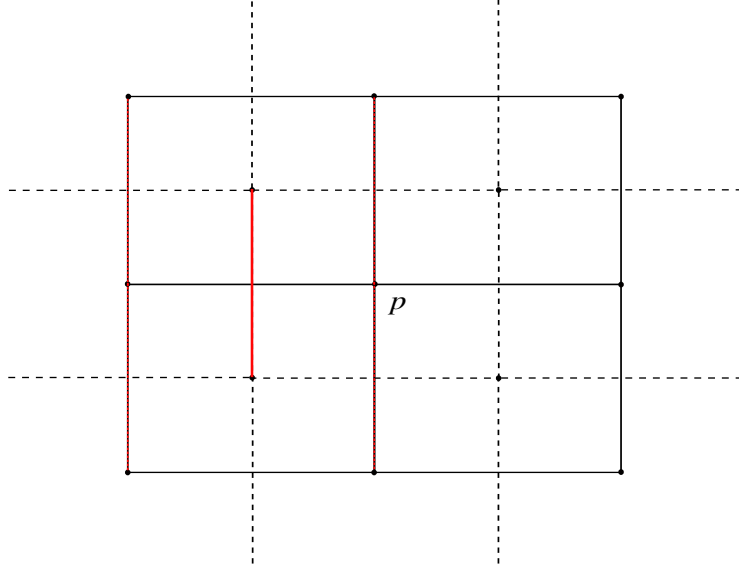


Figure 7: Primary and dual meshes around node p .

For example, the mass flux at the left face of the dual mesh, corresponding to $f_p = l_p$, which is in red in Figure 7, is computed by taking the average algebraic value of the mass fluxes at the left faces of each cell c containing node p , that are all in red as well on the figure.

Thus, when writing momentum conservation at node p , one obtains:

$$m_p^{projx} \mathbf{u}_p^{projx} = m_p^{lagx} \mathbf{u}_p^{lagx} + dm_{l_p} \mathbf{u}_{l_p}^{lagx o2} - dm_{r_p} \mathbf{u}_{r_p}^{lagx o2}$$

which gives the velocities at t^{projx} , and ends the X -remap phase.

2.1.2 Y -Remap phase

Details of the computations will not be written because they are strictly equivalent to those performed in the X -remap phase. The Lagrangian displacement along Y is only computed from vertical node predicted velocities $\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_y$, see Figure 8. The only difference is that, instead of starting computations with values of the variables at t^{lag} like for the X -remap, this phase starts with values at t^{projx} , i.e. values of the variables after the X -remap has been performed.

The computation of volume fluxes at the top $t = (i, j + 1/2)$ and bottom $b = (i, j - 1/2)$ faces enable to write at t^{lagy} , taking into account that $Vol^{projx} = Vol^n$:

$$Vol^{lagy} = Vol^n + dVol_b - dVol_t ,$$

$$\rho^{lagy} = \rho^n \frac{Vol^n}{Vol^{lagy}} \quad \text{and} \quad e^{lagy} = e^{projx} .$$

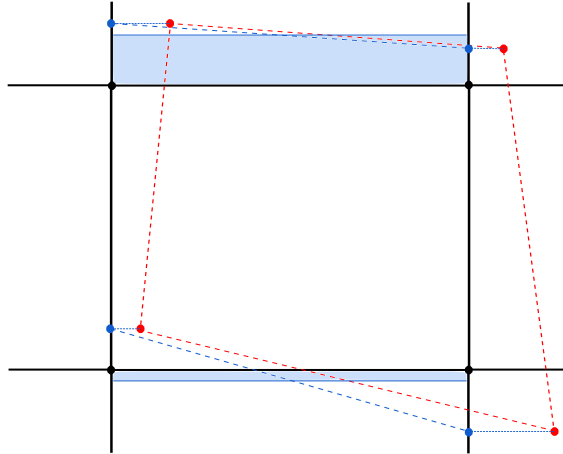


Figure 8: Computation of volume fluxes at Y -faces.

Then Y -remap (after reconstruction of Y -faces values) on both primary and dual meshes provides the values of all variables at $t^{projy} = t^{n+1}$.

Let us recap, at each iteration, once the Lagrangian phase is done, the ADI remap splits in four successive steps:

1. Lagrangian displacement of the mesh along X ($lagx$)
2. Remap of Lagrangian variables ($projx$)
3. Lagrangian displacement of the mesh along Y ($lagy$)
4. Remap of Lagrangian variables ($projy = \text{iteration } n + 1$)

Remark 1: Because of the unidirectional way fluxes are computed and remapped, one can think at first that the AD remap is a 5 points remap. That is to say, the stencil is the cross-shaped group of cells containing the cell at the centre and its four first neighbours, see Figure 9. Actually, thanks to the directional splitting, matter can be exchanged through the corners. For example, from cell (i, j) , it is possible to transfer matter to cell $(i + 1, j)$ through face $(i + 1/2, j)$ during the X -remap phase, and then transfer a part of this matter to cell $(i + 1, j + 1)$ through face $(i + 1, j + 1/2)$ during the Y -remap phase. Somehow, the directional splitting computes implicit corner fluxes. That is why the AD remap must be thought as a 9 points remap (star-shaped group of 5 plus the cells in the 4 diagonal directions, as shown Figure 9).

Remark 2: Let us count the number of MPI communications required during one time step: one synchronization for the Lagrangian phase, one for the X -remap phase, and another one for the Y -remap phase. That is a total of **3 in 2D**.



Figure 9: Stencils of 5 points remaps (left) and 9 points remaps (right).

Remark 3: A fixed order of treating X and Y directions necessarily introduces a loss of symmetry in the geometry, especially in cases where the solution is supposed to be symmetric with respect to $x = y$. In order to diminish this effect, another alternating is processed: X - Y remap at odd iterations and Y - X remap at even iterations. It also produces an almost second order in time remap scheme because it is similar to a Strang splitting when regarding two successive iterations.

2.2 Direct Remap

The name "Direct" means that fluxes at the X and Y -faces are independently computed and remapped, but at the same time. This is a one-step remap, see [4]. The Lagrangian displacement along both directions X and Y and the fluxes are computed respectively from $\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_x$ and $\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_y$, exactly the same way as for the ADI remap, see Figure 10.

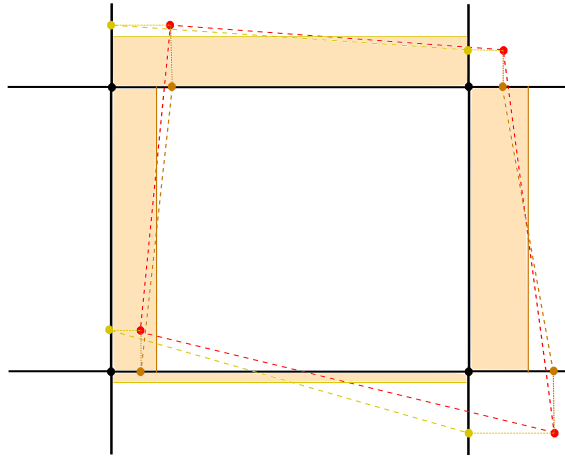


Figure 10: Computation of volume fluxes at both X and Y -faces.

Orange and Yellow dots refer to X and Y displacements, respectively. The values of volume fluxes at each face l, r, t, b lead to the following Lagrangian quantities after the grid has been displaced, at t^{lagxy} :

$$Vol^{lagxy} = Vol^n + dVol_l - dVol_r + dVol_b - dVol_t,$$

$$\rho^{lagxy} = \rho^n \frac{Vol^n}{Vol^{lagxy}} \quad \text{and} \quad e^{lagxy} = e^{lag}.$$

The reconstruction of face variables is also the same as for the AD remap, i.e. variables at X and Y -faces are interpolated using 1D second order limited linear reconstructions in both X and Y directions. When writing mass and energy conservations, one gets:

$$m_c^{proj} = m_c^{lagxy} + \sum_{faces} \rho_f^{lagxy \ o2} dVol_f$$

$$m_c^{proj} e_c^{proj} = m_c^{lagxy} e_c^{lagxy} + \sum_{faces} \rho_f^{lagxy \ o2} e_f^{lagxy \ o2} dVol_f$$

Similarly, momentum conservation on the dual mesh writes:

$$m_p^{projx} \mathbf{u}_p^{projx} = m_p^{lagx} \mathbf{u}_p^{lagx} + \sum_{faces} dm_{f_p} \mathbf{u}_{f_p}^{lagx \ o2}$$

As for the previous type of remap, let us recap the successive steps of the Direct remap, once the Lagrangian phase is over:

1. Lagrangian displacement of the mesh along X and Y ($lagxy$)
2. Remap of Lagrangian variables ($proj = \text{iteration } n + 1$)

Remark 1: Unlike the AD remap, since the remap is in one step, there is no other flux than the explicit fluxes at X and Y -faces. This remap is a strict 5 points remap with a star-shaped stencil, just like a classical Finite-Volume scheme. Since there are no corner fluxes, corner effects will not be taken into account by the scheme. This might lead to an accuracy not quite as good as the AD remap, even in simple cases like the linear advection in the $x = y$ direction.

Remark 2: Thanks to one-step remap, the number of MPI communications reduces to **2 in 2D** (one at each phase), and in fact 2 whatever the dimension.

2.3 Direct Remap with Corner Fluxes

The initial idea is to build a type of remap that combines the advantages of both remaps presented before. That is to say, a one-step remap that takes into account corner effects. Those two requirements together imply necessarily the definition of corner fluxes, since one-step face fluxes cannot report corner effects.

For the corner flux at a given node p to be defined unequivocally, it has to depend only on node quantities. The displacement of the node seems to be the most natural

way to define the volume flux at this node. In the same way as what has been done for face fluxes, a rectangular approximation of the corner flux is computed, setting that the vector $\mathbf{u}_p^{n+1/2} \Delta t^n$ represents the diagonal of the rectangular corner flux, see Figure 11.

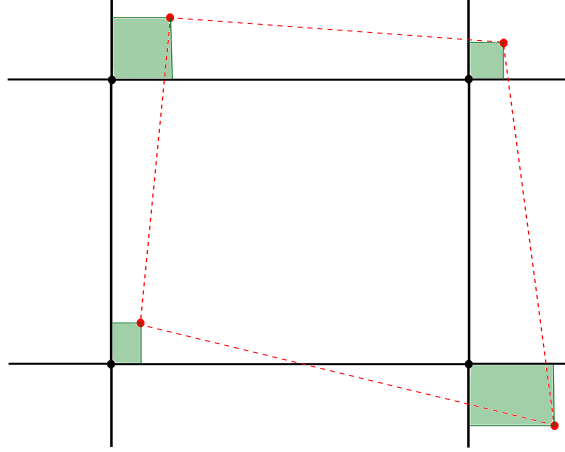


Figure 11: Definition of corner volume fluxes.

Thus, the volume flux at each corner p can be expressed as follows:

$$dVol_p = |\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_x| \Delta t^n \times |\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_y| \Delta t^n > 0$$

Setting $dx_p^{n+1/2} = (\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_x) \Delta t^n$ and $dy_p^{n+1/2} = (\mathbf{u}_p^{n+1/2} \cdot \mathbf{e}_y) \Delta t^n$, one obtains

$$dVol_p = |dx_p^{n+1/2} \times dy_p^{n+1/2}|$$

Remark: Corner fluxes are completely defined from a positive value $dVol_p$, and both signs of $dx_p^{n+1/2}$ and $dy_p^{n+1/2}$. These signs enable to identify the donor cell and the receiving cell.

Now corner fluxes have been well defined, one needs to compute fluxes at the faces of the cell. Here again, according to what is done in other remaps, a natural choice is the rectangular approximation of the trapezoid, see Figure 12. Note that at each face, the volume flux is well defined since it only depends on the velocities at its nodes, and not on the donor cell. This is mandatory for the sake of conservation.

A general methodology can be given for face volume fluxes computation. At each face, one has to determine the coordinates (x_f^-, y_f^-) , (x_f^+, y_f^+) of the vertices of the trapezoid, i.e. the red circles in Figure 12. For example, at any X -face f which extremities are p^- and p^+ , we start computing y_f^- and y_f^+ using this simple formula:

$$y_f^- = y_{p^-}^n + \max(0, dy_{p^-}^{n+1/2}), \quad y_f^+ = y_{p^+}^n + \min(0, dy_{p^+}^{n+1/2})$$

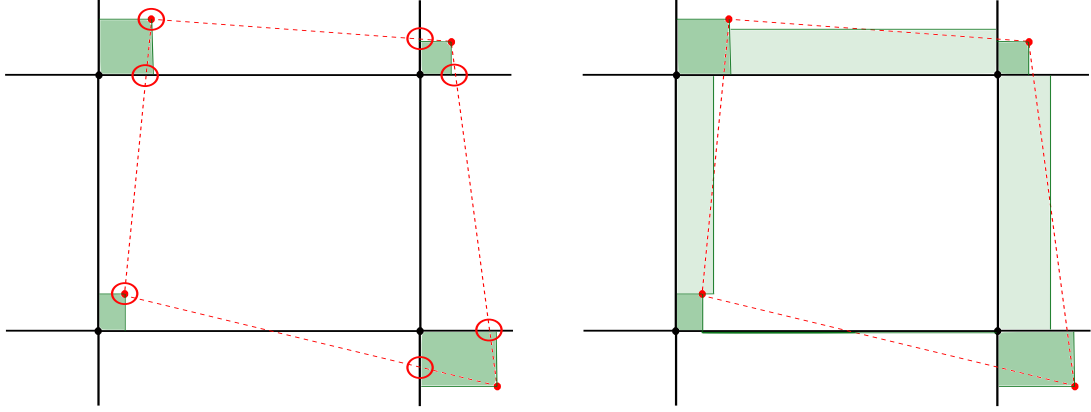


Figure 12: Definition of volume fluxes at both X and Y -faces.

Then x_f^- and x_f^+ are determined thanks to the equation of the line corresponding to Lagrangian face f , computed from $\mathbf{x}_p^{n+1/2}$. This finally leads to the following volume flux at X -face f :

$$dVol_f = \frac{1}{2}(x_f^- + x_f^+) \times (y_f^+ - y_f^-)$$

Remark: In this remap, the volume flux at each face depends on both x and y components of the velocities at its extremities.

Given volume fluxes at the faces and at the corners, one is able to compute Lagrangian geometric quantities such as the volume (see Figure 13) as well as thermodynamical quantities at time t^{lagCF} :

$$Vol^{lagCF} = Vol^n + \sum_{faces} dVol_f + \sum_{corners} dVol_p,$$

$$\rho^{lagCF} = \rho^n \frac{Vol^n}{Vol^{lagCF}} \quad \text{and} \quad e^{lagCF} = e^{lag}.$$

In this case, values have to be defined at faces, but also at corners. As for the faces, a choice has to be made between upwind reconstruction and limited linear reconstruction. The importance of this choice and the way of defining second order values at the corners will be discussed in section 3.2. Here, those reconstructed values at the corners are assumed to be known. The equations of the remap on the primary mesh write in cell c :

$$m_c^{proj} = m_c^{lagCF} + \sum_{faces} \rho_f^{lagCF \ o2} dVol_f + \sum_{corners} \rho_p^{lagCF \ o2} dVol_p$$

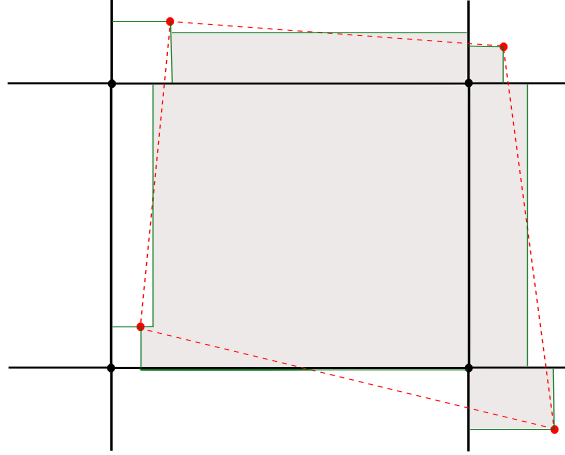


Figure 13: Lagrangian representation of the volume.

$$\begin{aligned}
 m_c^{proj} e_c^{proj} &= m_c^{lagCF} e_c^{lagCF} + \sum_{faces} \rho_f^{lagCF} e_f^{lagCF} dVol_f \\
 &\quad + \sum_{corners} \rho_p^{lagCF} e_p^{lagCF} dVol_p
 \end{aligned}$$

For obvious reasons of compatibility between the primary and dual meshes, it is essential to define fluxes at the corners p'_p of the dual mesh, i.e. at the centres of the cells of the primary mesh $c = p'_p$. There are several ways to define those corner fluxes. Note that as it is the case for the faces, only mass corner fluxes need to be defined to remap x and y components of velocities. In order to stay as consistent as possible with what is done at the faces (i.e. in the AD remap), one must express nodal mass corner fluxes as follows:

$$dm_{p'_p} = \frac{1}{4} \sum_{cell \in \{p\}} (dm_{p'})_{cell}$$

For instance, if p' stands for left-bottom, like in Figure 14, the mass corner flux at the left-bottom node of the dual cell p is defined as the average of all mass corner fluxes at left-bottom nodes of the cells $cell \in \{p\}$.

Remark: In this particular case, the left-bottom mass corner flux in the cell situated at the right-bottom of node p , say cell p_{+-} , equals zero. Indeed, the direction of this corner flux makes it equal zero for this cell, and non zero for the cell situated at the left-bottom of node p , say p_{--} . Thus, this flux will be counted as the flux at the right-bottom node of cell p_{--} when computing the right-bottom mass corner flux in the dual cell.

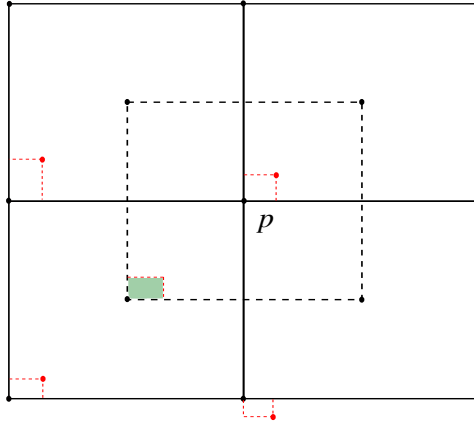


Figure 14: Definition of corner fluxes on the dual mesh.

Regarding mass fluxes at the faces, they are computed similarly to AD or Direct remaps. All mass fluxes being known, let us write momentum conservation equation:

$$m_p^{proj} \mathbf{u}_p^{proj} = m_p^{lagCF} \mathbf{u}_p^{lagCF} + \sum_{faces} dm_{f_p} \mathbf{u}_{f_p}^{lagCF o2} + \sum_{corners} dm_c \mathbf{u}_c^{lagCF o2}$$

Here is a brief recap of the successive steps of the Direct Remap with Corner Fluxes (denoted DirectCF), after the end of the Lagrangian phase:

1. Lagrangian displacement of the mesh along X , Y and diagonal directions ($lagCF$)
2. Remap of Lagrangian variables ($proj = \text{iteration } n + 1$)

Remark 1: The desired properties seem to be recovered. Indeed, in terms of MPI communications, the one-step aspect of this remap implies **2 in 2D**. It also implies a **higher arithmetic intensity** than the AD remap. Plus, the corner effects are obviously taken into account, in a way that appears to be more faithful than the AD remap since corner fluxes are computed directly from the displacement of the nodes.

Remark 2: Geometric computations are more complex than in the AD case.

3 Specificities of the Direct Remap with Corner Fluxes

Before implementing this new remap, it is natural to wonder in what extent could it be beneficial, i.e. whether it is worth performing those complex computations, and how do these corners affect the initial scheme.

3.1 Specific case: linear advection with first order reconstruction for the remap

Let us first compare the DirectCF and AD remaps in the most basic situation: linear advection, which corresponds to the case of a constant velocity field $\mathbf{u} = (u_x, u_y)$. The remaps are performed with a first order, i.e. upwind, reconstruction of variables at corners and faces. Let a be a scalar volume quantity defined on the primary mesh, the goal is to compare $a^{proj, CF}$ and $a^{proj, AD}$. Horizontal and vertical displacements of the nodes are respectively defined as $dx = u_x \Delta t$, $dy = u_y \Delta t$. To simplify computations, dx and dy are assumed to be positive. Two dimensionless variables are introduced: $\varepsilon_x = dx/\Delta x$, $\varepsilon_y = dy/\Delta y$.

Direct remap with Corner Fluxes

At each node p , X -face f_X and Y -face f_Y , volume fluxes write:

$$dVol_p = dx dy, \quad dVol_{f_X} = dx (\Delta y - dy), \quad dVol_{f_Y} = dy (\Delta x - dx)$$

Since the velocity field is constant, the grid is not deformed. In particular, $Vol^{lag} = \Delta x \Delta y$. Moreover, in the pure advection case, $a_{i,j}^{lag} = a_{i,j}$ for all (i, j) . Thus, when writing the equation of mass conservation in cell (i, j) , one obtains:

$$\begin{aligned} \Delta x \Delta y a_{i,j}^{proj} &= \Delta x \Delta y a_{i,j} - dVol_{f_X} (a_{i,j} - a_{i-1,j}) \\ &\quad - dVol_{f_Y} (a_{i,j} - a_{i,j-1}) - dVol_p (a_{i,j} - a_{i-1,j-1}) \end{aligned}$$

$$a_{i,j}^{proj, CF} = a_{i,j} (1 - \varepsilon_x)(1 - \varepsilon_y) + a_{i-1,j} \varepsilon_x (1 - \varepsilon_y) + a_{i-1,j} \varepsilon_y (1 - \varepsilon_x) + a_{i-1,j-1} \varepsilon_x \varepsilon_y$$

Alternate Directions remap

Let us first consider the X -remap. This time, at each X -face f_X , the volume flux writes:

$$dVol_{f_X} = dx \Delta y$$

and for all (i, j) , mass conservation leads to:

$$\Delta x \Delta y a_{i,j}^{projx} = \Delta x \Delta y a_{i,j} - dVol_{f_X}(a_{i,j} - a_{i-1,j})$$

$$a_{i,j}^{projx} = a_{i,j} (1 - \varepsilon_x) + a_{i-1,j} \varepsilon_x$$

so that

$$a_{i,j-1}^{projx} = a_{i,j-1} (1 - \varepsilon_x) + a_{i-1,j-1} \varepsilon_x$$

Then for the Y -remap, at each Y -face f_Y , the volume flux writes:

$$dVol_{f_Y} = dy \Delta x$$

$$\text{thus, } \Delta x \Delta y a_{i,j}^{proj} = \Delta x \Delta y a_{i,j}^{projx} - dVol_{f_Y}(a_{i,j}^{projx} - a_{i,j-1}^{projx})$$

$$a_{i,j}^{proj, AD} = a_{i,j}(1 - \varepsilon_x)(1 - \varepsilon_y) + a_{i-1,j} \varepsilon_x(1 - \varepsilon_y) + a_{i-1,j} \varepsilon_y(1 - \varepsilon_x) + a_{i-1,j-1} \varepsilon_x \varepsilon_y$$

The result obtained is the same for both schemes. In other words, the two remaps are strictly equivalent in this specific case. First, it shows that the DirectCF remap degenerates to the right well-known scheme at order 1 in such a simple situation. Second, this feature underlines the importance of second order reconstruction for the DirectCF remap. The accuracy of the results and the differences with AD remap results will highly depend on this reconstruction.

3.2 Accuracy of the geometric representation for non-linear deformations

Let us now consider a very simple non-linear situation: the displacement of one single node in the direction $x = y$, see Figure 15. The initial density equals 1 in the cell at the centre and 0 in every other cell. It is assumed that $dx = dy$. In order to estimate the impact of the corner fluxes on both geometric representation and effective mass fluxes, for each remap, the Lagrangian volume will be computed and compared to the exact Lagrangian volume, so will the mass flux at the top right corner.

To be more specific, the quotient $Vol^{lag}/(\Delta x)^2$ is computed in function of $\varepsilon = dx/\Delta x$. Geometric representations of the different Lagrangian volumes are compared in Figure 16.

The results of the computations give, stopping at the third order in ε for the Direct remap with Corner Fluxes:

$$\text{Exact: } 1 + \varepsilon$$

$$\text{AD: } 1 + \varepsilon + \varepsilon^2/4$$

$$\text{Direct: } 1 + \varepsilon$$

$$\text{Direct with Corner Fluxes: } 1 + \varepsilon + \varepsilon^3 + o(\varepsilon^3)$$

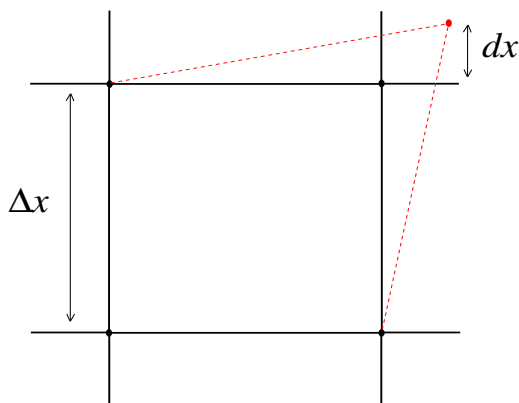


Figure 15: Lagrangian displacement of one single node.

Remark 1: In the AD case, since the effective Lagrangian volume $Vol^{lag, AD}$ does not appear directly, the value is deduced from ρ^{projy} , defining this volume such that $\rho^{projy} Vol^{lag, AD} = m^n$. Besides, in this simple case, this representation is strictly equivalent to a rectangular approximation, which consists in taking the middles of Lagrangian faces and taking the rectangle passing through these points. Let us be a little more specific, starting with denoting $\rho^n = m^n / \Delta x^2$. It is clear that:

$$Vol^{lagx} = \Delta x^2 + \frac{dx}{2} \Delta x, \quad \text{thus } m^{projx} = \frac{m^n}{1 + \varepsilon/2}$$

Then, since $Vol^{lagy} = Vol^{lagx}$, one gets:

$$m^{projy} = \frac{m^{projx}}{1 + \varepsilon/2} = \frac{m^n}{(1 + \varepsilon/2)^2}$$

And finally, according to the definition of $Vol^{lag, AD}$ given below:

$$\frac{Vol^{lag, AD}}{\Delta x^2} = \frac{m^n}{m^{projy}} = (1 + \varepsilon/2)^2 = 1 + \varepsilon + \varepsilon^2/4$$

Remark 2: The Direct gives exactly the right Lagrangian volume. And the DirectCF is one order more accurate than the AD in this particular case.

In order to estimate corner effects, let us determine the mass flux passing through the top-right corner during the remap phase. Mass fluxes are coloured in green in the Figures.

In the Direct and DirectCF cases, this mass corner flux is easily computed knowing ρ^{lag} and $dVol_p$, see Figure 17.

But in the AD case, the computation is less straight-forward because of the directional splitting, see Figure 18. Indeed, the result depends on the displacement of

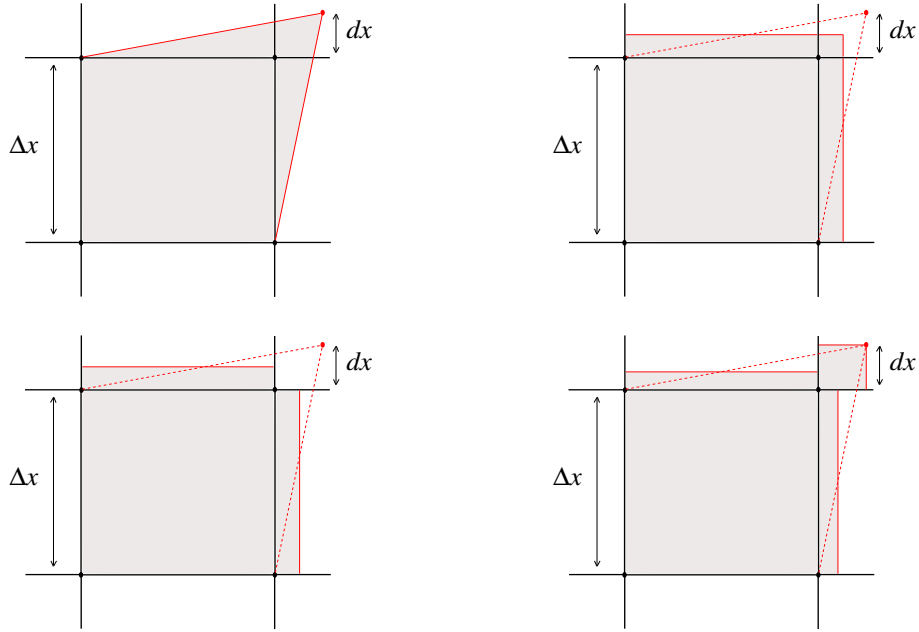


Figure 16: Representation of the Lagrangian volume in the exact (top-left), AD (top-right), Direct (bottom-left) and DirectCF (bottom-right) cases.

the node situated at the right of the top-right node considered, which does not seem really natural. This displacement is denoted dx' .

Similarly to the representation of the volume, the quotient dm_p/m is expressed in function of $\varepsilon = dx/\Delta x$ and $\varepsilon' = dx'/\Delta x$ (for the AD remap), which gives at the third order

$$\text{Exact: } \varepsilon^2 - 2\varepsilon^3$$

$$\text{AD: } \varepsilon^2/4 + \varepsilon\varepsilon'/4 - \frac{1}{4}(\varepsilon^3 + \varepsilon\varepsilon'^2 + \varepsilon'\varepsilon^2)$$

$$\text{Direct: } 0$$

$$\text{Direct with Corner Fluxes: } \varepsilon^2 - \varepsilon^3$$

Remark 1: The results for the Direct and AD remaps are not close to the exact value, they are not even accurate at the highest order (second) in this simple case. The fact that the result in the AD case depends on ε' shows a drawback of this remap, at least formally.

Remark 2: In this example, the DirectCF gives results for Lagrangian volume and mass corner flux that coincide with exact values at the highest order. Thus, it seems that the DirectCF offers a better geometric lagrangian representation than the one in the AD remap.

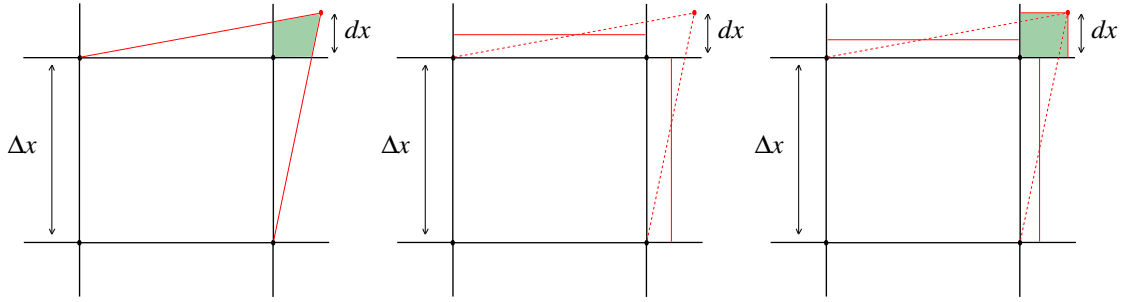


Figure 17: Mass corner flux in the exact (left), Direct (centre) and DirectCF (right) cases.

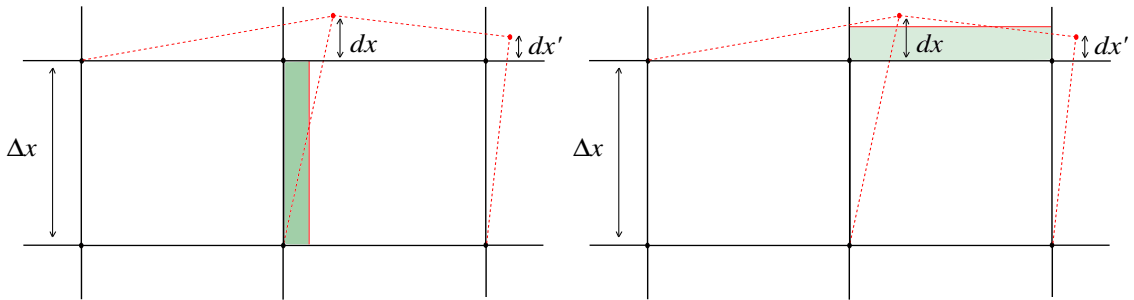


Figure 18: Mass face fluxes at X (top) and Y (bottom) remap phases in the AD case.

3.3 Impact of the corners on the scheme

The first notable impact of adding corners fluxes in the Lagrangian representation is the mandatory addition of corner fluxes on the dual mesh, as seen in section 2.3.

Another consequence, also mentioned in section 2.3, is the definition of reconstructed values of variables at the corners. As for the reconstruction at faces, the order of interpolation is the user's choice, which has to be consistent with the way face values are reconstructed. Contrary to the first order reconstruction that is obvious, it seems that second order reconstruction can be computed in different ways. Three of them are presented in the following paragraphs.

First order: upwind reconstruction

Let a^{lag} be the cell-centred lagrangian variable to reconstruct at the corner. This exponent lag is dropped to simplify notations: $a^{lag} \leftrightarrow a$. At the left-bottom corner

$p = (i - 1/2, j - 1/2)$, the upwind reconstruction writes:

$$a_p^{upw} = a_c \quad \text{where } c = \begin{cases} (i - 1, j - 1) & \text{if } dx_p^{n+1/2} > 0, \quad dy_p^{n+1/2} > 0 \\ (i, j) & \text{if } dx_p^{n+1/2} < 0, \quad dy_p^{n+1/2} < 0 \\ (i - 1, j) & \text{if } dx_p^{n+1/2} > 0, \quad dy_p^{n+1/2} < 0 \\ (i, j - 1) & \text{if } dx_p^{n+1/2} < 0, \quad dy_p^{n+1/2} > 0 \end{cases}$$

Second order: average reconstruction in the diagonal direction

If corner fluxes are taken into account exactly as face fluxes are, the value of each variable at corner $p = (i - 1/2, j - 1/2)$ has to be reconstructed along the diagonal direction $x = y$. The most basic reconstruction writes:

$$\widetilde{a}_p^{o2, avg} = \frac{1}{2} \begin{cases} (a_{i-1, j-1} + a_{i, j}) & \text{if } dx_p^{n+1/2} dy_p^{n+1/2} > 0 \\ (a_{i-1, j} + a_{i, j-1}) & \text{if } dx_p^{n+1/2} dy_p^{n+1/2} < 0 \end{cases}$$

Actually, in order to prevent the creation of negative masses, which can happen if $\rho_p^{o2, avg} > \rho_c$, with c the donor cell defined in the upwind reconstruction, the following formula is preferred:

$$a_p^{o2, avg} = \min(a_c, \widetilde{a}_p^{o2, avg})$$

Second order: linear limited reconstruction in both X and Y directions

Another way of reconstructing corner values comes from the the 2D grid point of view. Second order expressions for face values rely on a writing related to coordinates. For example, at X -face f , denoting by \mathbf{x}_f the middle of this face and \mathbf{x}_c the centroid of the donor cell, the 1D X -reconstruction is based on the following feature:

$$\mathbf{x}_f = \mathbf{x}_c \pm \frac{\Delta x}{2} \mathbf{e}_x$$

Thus, noting that at corner $p = (i - 1/2, j - 1/2)$, one can write

$$\mathbf{x}_p = \mathbf{x}_c \pm \frac{\Delta x}{2} \mathbf{e}_x \pm \frac{\Delta y}{2} \mathbf{e}_y$$

It seems fair to choose at node p a reconstruction based on the one introduced in section 2.1, that is:

$$a_p^{o2, xy} = a_c + \frac{\delta_x^{o2} a_c}{2} \left(\text{sgn}(dVol_{f_x}) \Delta x_c^{lag} - \Delta t^n \mathbf{u}_{f_x}^{n+1/2} \cdot \mathbf{e}_x \right) \\ + \frac{\delta_y^{o2} a_c}{2} \left(\text{sgn}(dVol_{f_y}) \Delta y_c^{lag} - \Delta t^n \mathbf{u}_{f_y}^{n+1/2} \cdot \mathbf{e}_y \right)$$

$$\text{where } f_x = \begin{cases} (i - 1/2, j - 1) & \text{if } dy_p^{n+1/2} > 0 \\ (i - 1/2, j) & \text{if } dy_p^{n+1/2} < 0 \end{cases}$$

$$\text{and } f_y = \begin{cases} (i-1, j-1/2) & \text{if } dx_p^{n+1/2} > 0 \\ (i, j-1/2) & \text{if } dx_p^{n+1/2} < 0 \end{cases}$$

Remark: This formula is quite convenient since it can be expressed as a simple sum or difference of values of a at centres and faces of the cells, which are already known.

Second order: linear limited reconstruction in the diagonal direction

Let us denote by \mathbf{v} and \mathbf{v}_\perp the unit vectors along the diagonal and antidiagonal directions, such that:

$$\mathbf{v} = \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \quad \mathbf{v}_\perp = \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix}$$

Note that $\mathbf{v} \cdot \mathbf{v}_\perp = 0$ only if $\Delta x = \Delta y$. The approach is the same as for the x - y reconstruction, but using the coordinate system $(\mathbf{v}, \mathbf{v}_\perp)$ instead of $(\mathbf{e}_x, \mathbf{e}_y)$. In other words, at corner p , let us write:

$$\begin{aligned} \mathbf{x}_p &= \mathbf{x}_c \pm \frac{1}{2} \sqrt{\Delta x^2 + \Delta y^2} \mathbf{v} \\ \text{or, } \mathbf{x}_p &= \mathbf{x}_c \pm \frac{1}{2} \sqrt{\Delta x^2 + \Delta y^2} \mathbf{v}_\perp \end{aligned}$$

For a quantity a_c in cell c , one is able to compute discrete limited gradients $\delta_{\mathbf{v}}^{o2} a_c$ and $\delta_{\mathbf{v}_\perp}^{o2} a_c$ along the diagonal and antidiagonal directions, with respect to what is done along the X and Y directions in section 2.1.1. In cell $c = (i, j)$, this leads:

$$\delta_{\mathbf{v}}^{o2} a_{i,j} = \frac{1}{2} \left(\Phi_{i,j}^+ \delta_{\mathbf{v}} a_{i+1/2, j+1/2} + \Phi_{i,j}^- \delta_{\mathbf{v}} a_{i-1/2, j-1/2} \right)$$

$$\text{with } \delta_{\mathbf{v}} a_{i\pm 1/2, j\pm 1/2} = \frac{a_{i\pm 1, j\pm 1} - a_{i,j}}{(\mathbf{x}_{i\pm 1, j\pm 1}^{lag} - \mathbf{x}_{i,j}^{lag}) \cdot \mathbf{v}}$$

$$\text{and } \Phi_{i,j}^+ = \varphi(r), \quad \Phi_{i,j}^- = \varphi(1/r), \quad r = \frac{\delta_{\mathbf{v}} a_{i-1/2, j-1/2}}{\delta_{\mathbf{v}} a_{i+1/2, j+1/2}}$$

The expression of $\delta_{\mathbf{v}_\perp}^{o2} a_{i,j}$ is similar. Given those discrete gradients, one can write a second order reconstruction as follows:

$$a_p^{o2, diag} = \begin{cases} a_c + \frac{\delta_{\mathbf{v}}^{o2} a_c}{2} \left(\text{sgn}(dx_p^{n+1/2}) \Delta \mathbf{x}_c^{lag} - \Delta t^n \mathbf{u}_p^{n+1/2} \right) \cdot \mathbf{v} \\ a_c + \frac{\delta_{\mathbf{v}_\perp}^{o2} a_c}{2} \left(\text{sgn}(dx_p^{n+1/2}) \Delta \mathbf{x}_c^{lag} - \Delta t^n \mathbf{u}_p^{n+1/2} \right) \cdot \mathbf{v}_\perp \end{cases}$$

The reconstruction is performed along \mathbf{v} if $dx_p^{n+1/2} dy_p^{n+1/2} > 0$ and along \mathbf{v}_\perp if $dx_p^{n+1/2} dy_p^{n+1/2} < 0$.

Remark: Signs of quantities in the formula depend on the sign of $dx_p^{n+1/2}$ since \mathbf{v} and \mathbf{v}_\perp have been chosen in the $x > 0$ direction.

Second order: multidimensional reconstruction

Instead of computing 1D linear reconstructions along all directions, the idea here is to build a bidimensional patch on the 9-points stencil. In order to be consistent with what is done for the second order in the 1D case, this patch has to be a plane. According to the multidimensional reconstruction model presented in [5], with respect to the notations previously introduced, the analytic expression of the parametrization $\check{a}_{i,j}$ of the plane reconstructed at cell $c = (i, j)$ for the quantity a writes:

$$\check{a}_{i,j}(\mathbf{x}) = a_{i,j} + \mathbf{G}_{i,j} \cdot (\mathbf{x} - \mathbf{x}_c)$$

Such a writing ensures that

$$\frac{1}{|\Omega_{i,j}|} \int_{\Omega_{i,j}} \check{a}_{i,j}(\mathbf{x}) \, d\mathbf{x} = a_{i,j}$$

The difficulty is to find the components of $\mathbf{G}_{i,j} = (G_{i,j}^1, G_{i,j}^2)$ that will provide a reconstruction as close to the real solution as possible. This will be done solving a least squares problem. The criterion chosen is the mean value of $\check{a}_{i,j}$ in each cell of the 9-points stencil. Consequently, the functional $E_{i,j}$ to minimize is the following:

$$E_{i,j} = \frac{1}{2} \sum_{\substack{|k-i| \leq 1 \\ |l-j| \leq 1}} \left(\frac{1}{|\Omega_{k,l}|} \int_{\Omega_{k,l}} \check{a}_{i,j}(\mathbf{x}) \, d\mathbf{x} - a_{k,l} \right)^2$$

The minimum is reached if, for all (k, l)

$$\frac{1}{|\Omega_{k,l}|} \int_{\Omega_{k,l}} \check{a}_{i,j}(\mathbf{x}) \, d\mathbf{x} = a_{k,l}$$

Let us denote:

$$\begin{cases} X_{i,j,k,l}^1 = \frac{1}{|\Omega_{k,l}|} \int_{\Omega_{k,l}} (x - x_c) \, d\mathbf{x} \\ X_{i,j,k,l}^2 = \frac{1}{|\Omega_{k,l}|} \int_{\Omega_{k,l}} (y - y_c) \, d\mathbf{x} \end{cases}$$

This leads to the following system in the matrix form for the minimum:

$$\begin{pmatrix} X_{i,j,i-1,j-1}^1 & X_{i,j,i-1,j-1}^2 \\ \vdots & \vdots \\ \vdots & \vdots \\ X_{i,j,i-1,j}^1 & X_{i,j,i-1,j}^2 \end{pmatrix} \cdot \begin{pmatrix} G_{i,j}^1 \\ G_{i,j}^2 \end{pmatrix} = \begin{pmatrix} a_{i-1,j-1} - a_{i,j} \\ \vdots \\ \vdots \\ a_{i-1,j} - a_{i,j} \end{pmatrix}$$

The overdetermined system to be solved rewrites: $\mathbf{X}_{i,j} \mathbf{G}_{i,j} = \mathbf{A}_{i,j}$, where geometric ($\mathbf{X}_{i,j}$) and physical ($\mathbf{A}_{i,j}$) quantities are dissociated. If the rank of the matrix $\mathbf{X}_{i,j}$ is maximal, then the symmetric matrix ${}^t\mathbf{X}_{i,j} \mathbf{X}_{i,j}$ is invertible and the solution of the least squares problem writes:

$$\mathbf{G}_{i,j} = ({}^t\mathbf{X}_{i,j} \mathbf{X}_{i,j})^{-1} {}^t\mathbf{X}_{i,j} \mathbf{A}_{i,j}$$

The matrix $({}^t\mathbf{X}_{i,j} \mathbf{X}_{i,j})^{-1} {}^t\mathbf{X}_{i,j}$ is called the Moore-Penrose pseudo-inverse of $\mathbf{X}_{i,j}$, see [6]. In this case, note that ${}^t\mathbf{X}_{i,j} \mathbf{X}_{i,j}$ is a 2×2 matrix, thus its inversion is not very costly in terms of calculation time. It can be done directly, using the commatrix formula, or in two steps: first performing a QR decomposition ${}^t\mathbf{X}_{i,j} \mathbf{X}_{i,j} = \mathbf{Q}_{i,j} \mathbf{R}_{i,j}$, then inverting the upper triangular matrix $\mathbf{R}_{i,j}$ and writing

$$\mathbf{G}_{i,j} = \mathbf{R}_{i,j}^{-1} {}^t\mathbf{Q}_{i,j} {}^t\mathbf{X}_{i,j} \mathbf{A}_{i,j}$$

In order to facilitate the possible extension to higher order reconstructions, the second method has been implemented.

The last formula gives the values of $G_{i,j}^1$ and $G_{i,j}^2$, which enable to get the expression of $\check{a}_{i,j}$, the non-limited reconstruction of $a_{i,j}$ on the stencil. Since the main objective is to remain robust, a limitation has to be performed. Formally, the limiter writes as a vector function $\Phi_{i,j}$ directly applied to $\mathbf{G}_{i,j}$, such that the new limited reconstruction gives:

$$\check{a}_{i,j}(\mathbf{x}) = a_{i,j} + \Phi_{i,j}(\mathbf{G}_{i,j}) \cdot (\mathbf{x} - \mathbf{x}_c)$$

The goal of the limitation is to diminish the norm of $\mathbf{G}_{i,j}$ when high gradients appear, but its direction needs to remain the same. Consequently, the form of the limiter function is the following:

$$\Phi_{i,j}(\mathbf{G}_{i,j}) = \phi_{i,j} \mathbf{G}_{i,j}$$

Once again, giving priority to robustness, the choice made for the scalar $\phi_{i,j}$ is:

$$\phi_{i,j} = \frac{1}{\|\mathbf{G}_{i,j}\|} \min \left(|\delta_x^{o2} a_{i,j}|, |\delta_y^{o2} a_{i,j}|, |\delta_{\mathbf{v}}^{o2} a_{i,j}|, |\delta_{\mathbf{v}_\perp}^{o2} a_{i,j}| \right)$$

All $\delta^{o2} a_{i,j}$ are computed using Van Leer slope limitation, as presented before (see section 2.1.1 for x , y , and previous paragraph for \mathbf{v} , \mathbf{v}_\perp). Note that the limitation coefficient takes into account the variations of a in all directions. Besides, the norm

of multidimensional limited gradient $\Phi_{i,j}(\mathbf{G}_{i,j})$ equals the minimum of the norms of limited 1D gradients in all directions. This ensures that the scheme remains stable.

Now the analytic expression of limited $\check{a}_{i,j}$ is known, the reconstructed value of a at any corner or face is obtained by evaluating the function $\check{a}_{i,j}$ at this point. For example, at corner p , if the centroid of $dVol_p$ is denoted \mathbf{x}_{c_p} , one simply gets:

$$a_p^{o2, multid} = \check{a}_{i,j}(\mathbf{x}_{c_p}) = a_{i,j} + \Phi_{i,j}(\mathbf{G}_{i,j}) \cdot (\mathbf{x}_{c_p} - \mathbf{x}_c)$$

Remark: Note that in the multidimensional case, unlike all other second order reconstructions presented before, values at faces are also reconstructed using $\check{a}_{i,j}$. Thus, corners and faces are treated with the same limitation coefficient.

To emphasize the importance of the corners on the scheme, and on the results obtained, a simple test-case has been ran with each one of the four different reconstructions. Let us consider the linear advection of a square ($\rho = 10 \text{ kg.m}^{-3}$) into a field ($\rho = 0.1 \text{ kg.m}^{-3}$) with velocity $\mathbf{u} = (5 ; 5) \text{ m.s}^{-1}$. The mesh contains 200×200 square cells. Fluxes at the faces are computed with a second order reconstruction. Results are gathered in Figure 19.

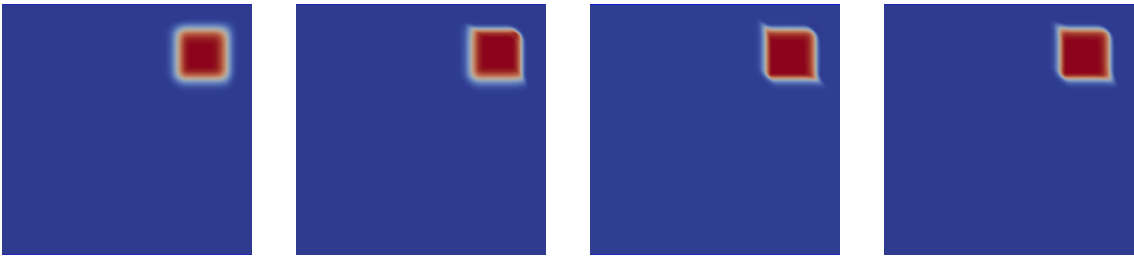


Figure 19: Comparison between upwind, average, linear x - y and linear diagonal reconstructions at the corners (from left to right).

Remark 1: As expected, the upwind reconstruction is the most diffusive, and the non-limited average reconstruction creates oscillations. In the upwind case, the shape of the square is preserved but the diffusion at the edges makes it not very accurate.

Remark 2: The linear x - y reconstruction tends to spread the corners along the $x = -y$ direction (i.e. the direction orthogonal to propagation), like with the Direct remap, see section 5. Here, this could be due to a wrong second order reconstruction at the corners. Indeed, the same reconstruction as the AD remap is used in each direction, and then the results are combined. This means that in each direction, the stencil used to reconstruct values has a width of 5 cells, which gives the 9 cells cross-shaped stencil $(i, [j - 2, j + 2]) \cup ([i - 2, i + 2], j)$. There is an inconsistency between this stencil (used for reconstruction) and the one of the remap, which is 9

cells square-shaped, coloured in blue on Figure 20. For example, it seems wrong to take into account the value in cell $(i + 2, j)$ and not the one in cell $(i + 1, j + 1)$ for the reconstruction of a value at the corner $(i + 1/2, j + 1/2)$.

Remark 3: The linear diagonal reconstruction seems more adapted since it shows the same default, but in a lesser extent. The stencil for this reconstruction is the union of the stencil for the x - y reconstruction and the X -shaped stencil containing the two diagonals, which are 5 cells-long each. The total stencil is 17 cells-wide, see Figure 20, and it includes the 9 points stencil of the remap.

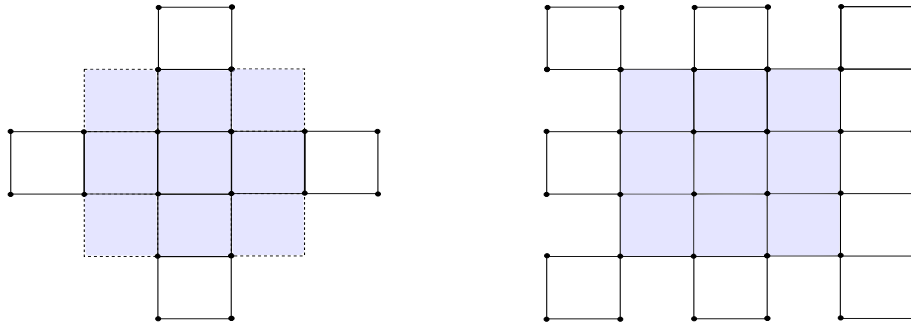


Figure 20: Stencils of both x - y (left) and diagonal (right) linear reconstructions.

Remark 4: The stencil of reconstruction in the multidimensional case is exactly the one of the remap since the patch is reconstructed over these 9 cells.

Remark 5: Even if corner fluxes are very small volumes ($O(\Delta x^2 \Delta t^2)$), their impact on the results is significant.

4 Multi-material extension

The numerical multi-material model displayed in this section is adapted to non-miscible fluids. It draws a sharp interface between two distinct materials. This method is called VOF (**V**olume **O**f **F**luids), see [8]. Its implementation in SHY code will be limited to two fluids.

4.1 Lagrangian phase

The multi-material Lagrangian phase is almost the same as in the mono-material case. The partial quantity of material α in cell c is denoted αc . Let us define the volume fraction of material α in cell c at t^n as:

$$k_{\alpha c}^n = \frac{Vol_{\alpha c}^n}{Vol_c^n}$$

The hypothesis made in this model is called "iso-deformation", that is to say, all materials are assumed to have the same compressibility during Lagrangian phase. In other words, the following equality holds at each iteration n

$$k_{\alpha c}^{n+1, lag} = k_{\alpha c}^{n+1/2} = k_{\alpha c}^n$$

Such an assumption could seem quite rough, but it actually gives good results.

Prediction phase

$$\left\{ \begin{array}{l} m_c^{n+1/2} = m_c^n \\ m_{\alpha c}^{n+1/2} = m_{\alpha c}^n \\ \mathbf{x}_p^{n+1/2} = \mathbf{x}_p^n + \frac{\Delta t^n}{2} \mathbf{u}_p^n \\ Vol_c^{n+1/2} = \mathcal{F}((\mathbf{x}_p^{n+1/2})_{p \in \{c\}}) \\ Vol_{\alpha c}^{n+1/2} = Vol_{\alpha c}^n \frac{Vol_c^{n+1/2}}{Vol_c^n} = k_{\alpha c}^n Vol_c^{n+1/2} \\ \mathbf{u}_p^{n+1/2} = \mathbf{u}_p^n - \frac{\Delta t^n}{2} (\nabla(P + Q))_p^{n+1/2} / \rho_p^n \\ e_{\alpha c}^{n+1/2} = e_{\alpha c}^n - (P_{\alpha c}^n + Q_c^n) \left(\frac{1}{\rho_{\alpha c}^{n+1/2}} - \frac{1}{\rho_{\alpha c}^n} \right) \\ P_{\alpha c}^{n+1/2} = \mathcal{P}(\rho_{\alpha c}^{n+1/2}, e_{\alpha c}^{n+1/2}) \end{array} \right. \quad (4)$$

Moreover, the average pressure in a mixed-cell writes:

$$P_c^{n+1/2} = \sum_{\alpha c} k_{\alpha c}^{n+1/2} P_{\alpha c}^{n+1/2}$$

In the same way, total mass and internal specific energy are naturally defined as:

$$m_c^{n+1/2} = \sum_{\alpha c} m_{\alpha c}^{n+1/2}$$

$$e_c^{n+1/2} = \left(\sum_{\alpha c} m_{\alpha c}^{n+1/2} e_{\alpha c}^{n+1/2} \right) / m_c^{n+1/2}$$

Remark: The equation verified by the velocity is strictly the same as in the mono-material case. Indeed, the deformation of the cell is determined by the average pressure, and not partial pressures. Note that the definition of the pseudo-viscosity Q given in section 1 only depends on \mathbf{u} , so there is only one Q in each cell.

Correction phase

$$\left\{ \begin{array}{l} m_c^{n+1, lag} = m_c^n \\ m_{\alpha c}^{n+1, lag} = m_{\alpha c}^n \\ \mathbf{x}_p^{n+1, lag} = \mathbf{x}_p^n + \Delta t^n \mathbf{u}_p^{n+1/2} \\ Vol_c^{n+1, lag} = \mathcal{F}\left(\left(\mathbf{x}_p^{n+1}\right)_{p \in \{c\}}\right) \\ Vol_{\alpha c}^{n+1, lag} = k_{\alpha c}^n Vol_c^{n+1, lag} \\ \mathbf{u}_p^{n+1, lag} = 2 \mathbf{u}_p^{n+1/2} - \mathbf{u}_p^n \\ e_{\alpha c}^{n+1, lag} = e_{\alpha c}^n - (P_{\alpha c}^{n+1/2} + Q_c^n) \left(\frac{1}{\rho_{\alpha c}^{n+1, lag}} - \frac{1}{\rho_{\alpha c}^n} \right) \\ P_{\alpha c}^{n+1, lag} = \mathcal{P}(\rho_{\alpha c}^{n+1, lag}, e_{\alpha c}^{n+1, lag}) \end{array} \right. \quad (5)$$

Thanks to system (5), all partial and average variables can be recovered at $t^{n+1, lag}$.

Interface positioning

As explained in the introduction, the model of interface in one cell is a segment unequivocally defined from volume fractions and the normal vector to this interface (in the case of two materials). A method to determine the normal vector to the

interface is due to Youngs in [13]. If \mathbf{n}_α and k_α respectively denote the outwards normal vector (in the sense of material α) and the volume fraction:

$$\mathbf{n}_\alpha = \frac{\nabla k_\alpha}{\|\nabla k_\alpha\|}$$

Then, given the volume fraction in each cell of the domain, by computing a discrete gradient (based on a 9 points square-shaped stencil), the normal vector is determined in every mixed-cell of the domain.

The knowledge of normal vectors and volume fractions enables to place the interfaces on each mixed-cell, see Figure 21. Let us now perform the remap.

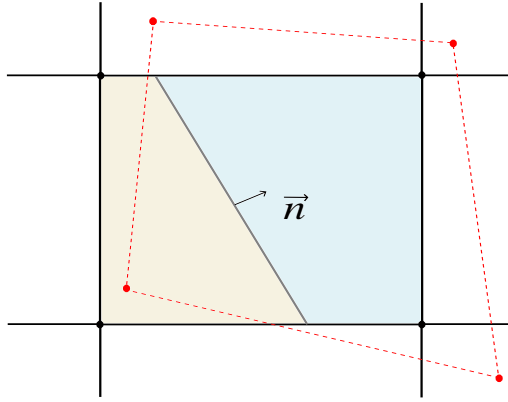


Figure 21: Multi-material sharp interface model.

Remark 1: The interface is not continuous on the domain.

Remark 2: Since k_α remains constant during the whole Lagrangian phase, \mathbf{n}_α is also assumed to be constant.

4.2 Alternate Directions Remap

X-Remap phase

Similarly to what is done in the mono-material case, the goal here is to compute volume fluxes of each material, i.e. the $dVol_{\alpha f}$, at all *X*-faces. The principle remains the same: Lagrangian displacement of the mesh along *X*, then remap. Node velocities give the value of Vol^{lagx} , a geometric representation of this volume, which is a rectangle thanks to the rectangular approximation of volume fluxes.

Taking into consideration the hypothesis exposed in the previous section, $k_{\alpha c}^{lagx} = k_{\alpha c}^n$ and $\mathbf{n}_{\alpha c}^{lagx} = \mathbf{n}_{\alpha c}^n$ are known. Thus, the interface can be placed on the cell at t^{lagx} , see Figure 22.

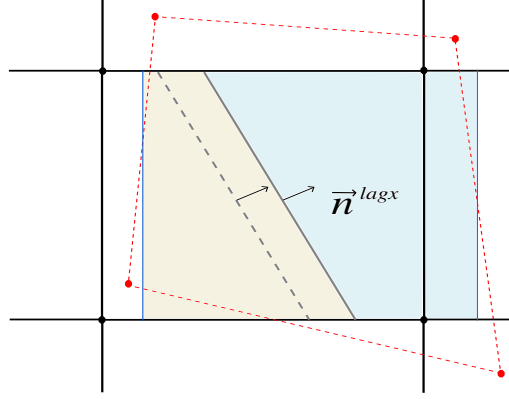


Figure 22: X-Remap. Intersection between the interface and the volume fluxes.

Now the interface is positioned on the mesh, one is able to compute its intersection with the volume fluxes at X -faces, i.e. the $dVol_f$. This simple intersection of a rectangle by a line enables to determine the $dVol_{\alpha f}$, see Figure 23.

If the intersection is empty, as it is the case in Figure 22, then volume fluxes are pure. At X -face f , in order to determine the material for the pure volume flux, one computes the scalar product:

$$(\mathbf{x}_f^{lagx} - \mathbf{x}_i^{lagx}) \cdot \mathbf{n}_{\alpha c}^{lagx}$$

where \mathbf{x}_f^{lagx} and \mathbf{x}_i^{lagx} respectively denote the middles of X -face f and the interface i . The sign of this quantity enables to conclude.

Once the $dVol_{\alpha f}$ have been determined, variable can be reconstructed at first or second order at the faces for each material. Unlike the mono-material case, the volume of each material has to be remapped, this gives $k_{\alpha c}^{projx}$. The rest is strictly equivalent as what is done in the mono-material case. Let us write down the equations:

$$Vol_{\alpha c}^{projx} = Vol_{\alpha c}^{lagx} + dVol_{\alpha l} - dVol_{\alpha r}$$

$$k_{\alpha c}^{projx} = \frac{Vol_{\alpha c}^{projx}}{Vol_c^n} = \frac{Vol_{\alpha c}^{projx}}{\Delta x \Delta y}$$

$$m_{\alpha c}^{projx} = m_{\alpha c}^{lagx} + \rho_{\alpha l}^{lagx \ o2} dVol_{\alpha l} - \rho_{\alpha r}^{lagx \ o2} dVol_{\alpha r}$$

$$m_{\alpha c}^{projx} e_{\alpha c}^{projx} = m_{\alpha c}^{lagx} e_{\alpha c}^{lagx} + \rho_{\alpha l}^{lagx \ o2} e_{\alpha l}^{lagx \ o2} dVol_{\alpha l} - \rho_{\alpha r}^{lagx \ o2} e_{\alpha r}^{lagx \ o2} dVol_{\alpha r}$$

Remark: Compared to the mono-material case, another condition is added for the limitation of second order linear reconstructions at faces. If the face is in the

neighbourhood of a contact discontinuity, i.e. if it is not surrounded only by pure cells, the reconstruction degenerates to order 1.

Y-Remap phase

Since volume fractions have been remapped, the hypothesis of the model at this time gives: $k_{\alpha c}^{lagy} = k_{\alpha c}^{projx}$ and $\mathbf{n}_{\alpha c}^{lagy} = \mathbf{n}_{\alpha c}^{projx}$, that are in general different from $k_{\alpha c}^n$ and $\mathbf{n}_{\alpha c}^n$. In Figure 23, the interface has rotated.

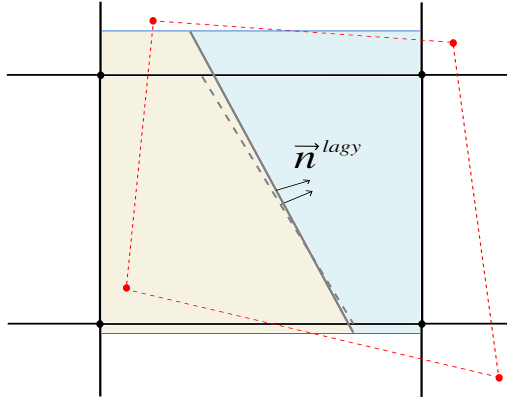


Figure 23: Y-Remap. Intersection between the interface and the volume fluxes.

Apart from that point, all the calculations are similar to those of the X-remap.

4.3 Direct Remap

Let us recall that in the mono-material case, volume fluxes at the faces are exactly the same as those in AD remap. But in the multi-material case, because of the interface positioning, fluxes are different. Indeed, what is performed in this remap can be summarized as follows. Let us define i^n as the interface at t^n .

The computation of the $dVol_{\alpha f}$ along both X and Y directions are performed at the same time. Along X , volume fluxes at X -faces are computed, the interface i^n is displaced along X (this new interface is denoted i_X^n) such that the volume fraction remains equal to $k_{\alpha c}^n$ in the X -Lagrangian cell (which is the same as in the X -remap in the AD case), and partial volume fluxes are deduced at X -faces. Along Y , volume fluxes at Y -faces are computed, the interface i^n is displaced along Y (this new interface is denoted i_Y^n) such that the volume fraction remains equal to $k_{\alpha c}^n$ in the Y -Lagrangian cell (which is the same as in the Y -remap in the AD case), and partial volume fluxes are deduced at Y -faces.

Remark: Two different interfaces i_X^n and i_Y^n are used respectively to determine partial volume fluxes at X and Y -faces. They both have the same normal vector as i^n , unlike the AD remap.

Finally, the formula for the remap of the volume gives:

$$Vol_{\alpha c}^{projx} = Vol_{\alpha c}^{lagx} + dVol_{\alpha l} - dVol_{\alpha r} + dVol_{\alpha b} - dVol_{\alpha t}$$

The other equations are logically deduced from this one.

4.4 Direct Remap with Corner Fluxes

Since the main idea behind the DirectCF remap is the accuracy of geometric representation, one single Lagrangian interface will be used when intersecting all volume fluxes. In order to place this interface thanks to $k_{\alpha c}^n$, $\mathbf{n}_{\alpha c}^n$ and Vol^{lagCF} , an approximation has to be made. Indeed, the Lagrangian representation of the volume is quite complex, see Figure 13. To avoid expensive calculations, the choice is to reconstruct the interface on a rectangular approximation of the Lagrangian cell. This approximation is very simple: it is the only rectangle whom faces pass through the middles of the four Lagrangian faces, see Figure 24.

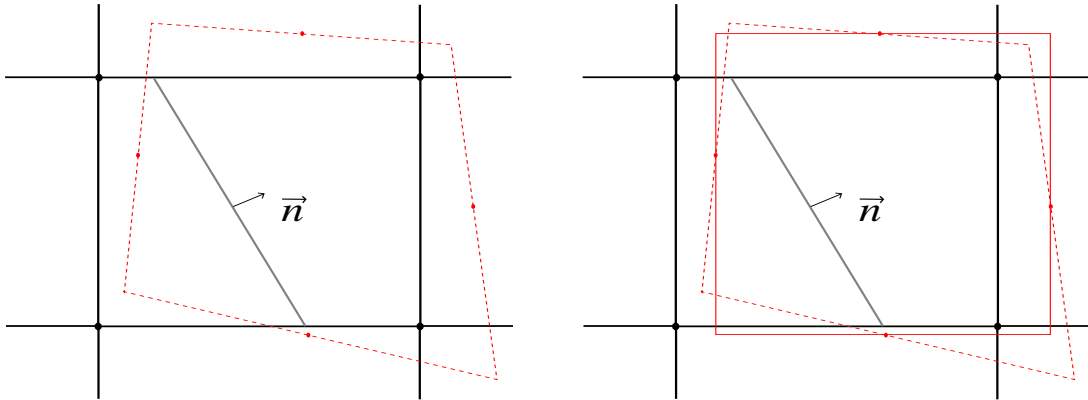


Figure 24: Rectangular approximation of Lagrangian cell.

Remark: The error committed by this approximation on the lagrangian geometry is assumed to be reasonable because this rectangular volume is exactly the Lagrangian volume taken into consideration in the AD remap, see section 3.1.

Now the approximated Lagrangian cell has been defined, and chosen rectangular, it is easy to place the interface using $k_{\alpha c}^n$, $\mathbf{n}_{\alpha c}^n$, see Figure 25.

Then, the equation of the interface enables to perform the intersection with all volume fluxes, at faces and corners. And the values of partial volume fluxes at faces and corners come out.

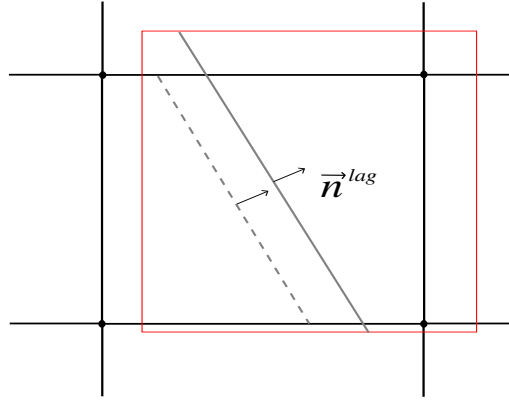


Figure 25: Placement of Lagrangian interface.

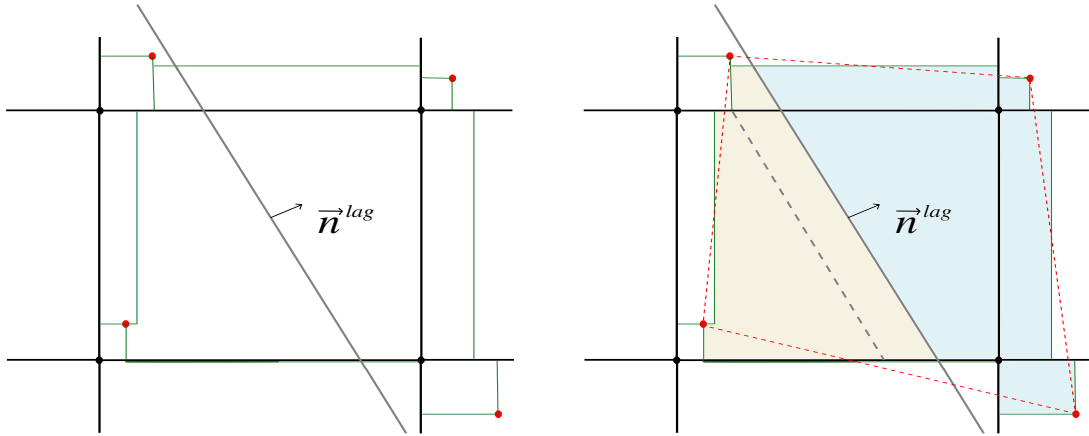


Figure 26: DirectCF remap. Intersection between the interface and volume fluxes.

Remark 1: In the same manner as for the faces, in the case of a pure corner flux at node p , the sign of $(\mathbf{x}_p^{n+1/2} - \mathbf{x}_i^{lagCF}) \cdot \mathbf{n}_{ac}^{lagCF}$ indicates the material.

Remark 2: Reconstructing the interface on a volume which is different from the one that enables to compute volume fluxes does not ensure the constancy of volume fractions during the Lagrangian phase. Such a little deviation from the incompressibility model could create artificial compressions or expansions of partial volumes. In order to fix this, the values of lagrangian partial volumes actually considered by the remap (i.e. yellow and blue volumes in Figure 26) need to be known. The too many possible situations make this computation extremely costly. Thus, this error will be neglected. In practice, even on severe benchmarks, it actually seems to be negligible.

Remark 3: In this case, there are one single normal and one single interface for all directions and volume fluxes, which reduces the calculation time. It is also a gain of accuracy since a single-interface (per iteration) model is obviously closer to the

exact case than a multiple-interface one.

Let us finally write the remap equation of the volume:

$$Vol_{\alpha c}^{proj} = Vol_{\alpha c}^{lagCF} + \sum_{faces} dVol_{\alpha f} + \sum_{corners} dVol_{\alpha p}$$

The rest follows from this formula.

5 Numerical results

In order to validate and verify the scheme including Direct remap with Corner Fluxes, several classical test-cases have been ran. They are presented in the following paragraphs. This section also aims at comparing this remap to the AD and Direct remaps. For each test-case, initial state is written, results are shown and some comments are made. The DirectCF implemented is second order at both faces and corners, with diagonal linear reconstruction at the corners.

5.1 First observations and convergence

Firstly, the accuracy of the three remaps are tested on classical test-cases of linear advection and solid rotation, in both mono-material and multi-material cases.

5.1.1 Mono-material

Advection

Linear advection and return of a square ($\rho = 10 \text{ kg.m}^{-3}$) into a field ($\rho = 0.1 \text{ kg.m}^{-3}$) with velocity $\mathbf{u} = (5 ; 5) \text{ m.s}^{-1}$.

Meshes: 50×50 , 100×100 , 200×200 , 400×400 .

Mono-material, Perfect Gas $p = (\gamma - 1)\rho e$ with $\gamma = 1.4$.

In this test-case, quantities are either constant, either transported. The goal is to measure the diffusivity of the scheme.

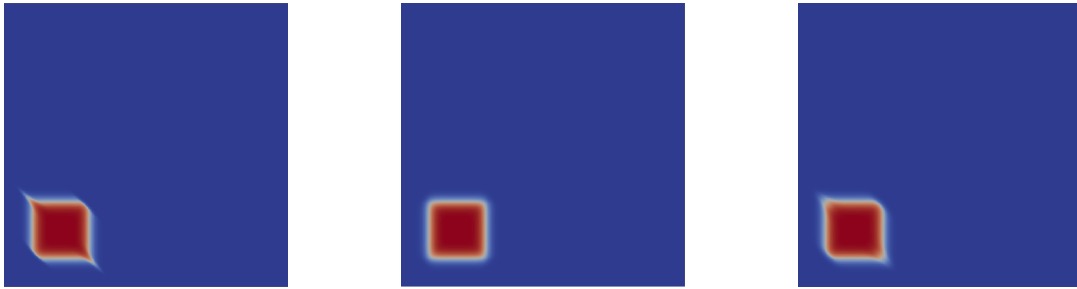


Figure 27: Monomat advection. Direct (left), AD (centre) and DirectCF (right).

Figure 27 shows the density after one advection and return on a mesh of 400×400 cells. The Direct shows the same deformation as the DirectCF, even more marked, whereas ADI draws a nice square. Regarding the L^2 error on the density, see Figure 28, despite its deformation at the corners along the $x = -y$ direction, the DirectCF error is quite comparable to AD, but the Direct is always less accurate, no matter the mesh.

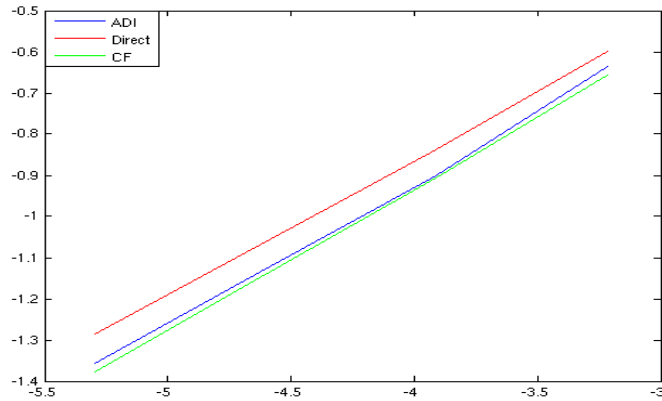


Figure 28: Monomat advection. $\log \|\rho^{final} - \rho^0\|_{L^2}$ in function of $\log(\Delta x)$

Rotation

Solid rotation (2π) of a square ($\rho = 10 \text{ kg.m}^{-3}$) into a field ($\rho = 0.1 \text{ kg.m}^{-3}$) with velocity $\mathbf{u}(r) = r\omega_0 \mathbf{e}_\theta$ and $\omega_0 = 2\pi \text{ s}^{-1}$ at each iteration.

Meshes: 50×50 , 100×100 , 200×200 , 400×400 . Mono-material, Perfect Gas $p = (\gamma - 1)\rho e$ with $\gamma = 1.4$.

Since velocities are imposed at each time step, this test-case especially evaluates geometric properties of the remap.

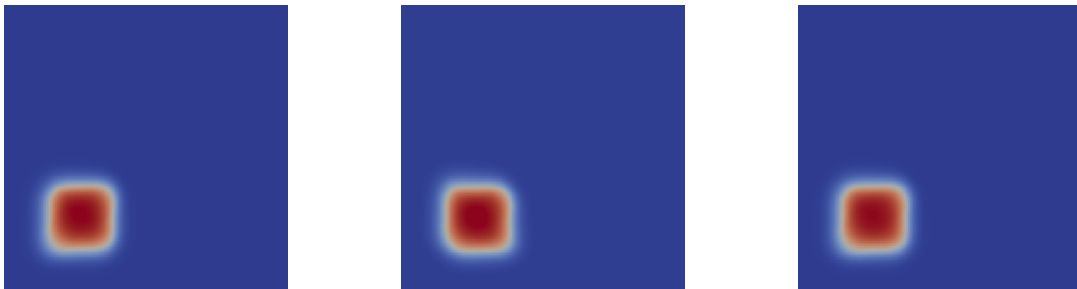


Figure 29: Monomat rotation. Direct (left), AD (centre) and DirectCF (right).

Here again, the density is represented on a 400×400 mesh after one rotation of angle 2π , see Figure 29. The Direct has broken the symmetry with respect to $x = y$, which is not the case for the two other remaps, that seem to be comparable to each other. The curves plotted on Figure 30 show that all remaps are equivalent in terms of the L^2 error.

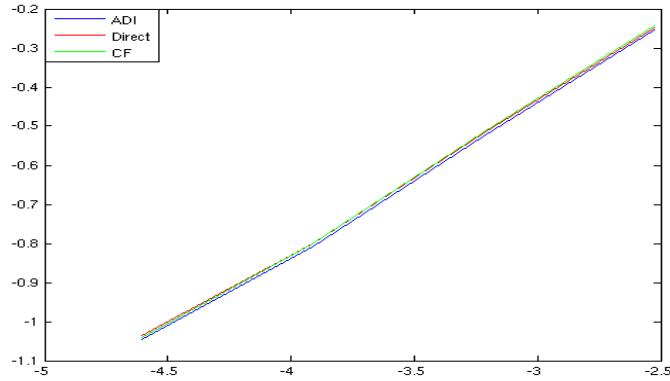


Figure 30: Monomat rotation. $\log \|\rho^{final} - \rho^0\|_{L^2}$ in function of $\log(\Delta x)$

5.1.2 Multi-material

Advection

Linear advection and return of a square of air ($\rho_a = 1.29 \text{ kg.m}^{-3}$) with velocity $\mathbf{u} = (5 ; 5) \text{ m.s}^{-1}$.

Meshes: 50×50 , 100×100 , 200×200 , 400×400 .

Multimaterial, air: Perfect Gas $p = (\gamma - 1)\rho e$ with $\gamma = 1.4$

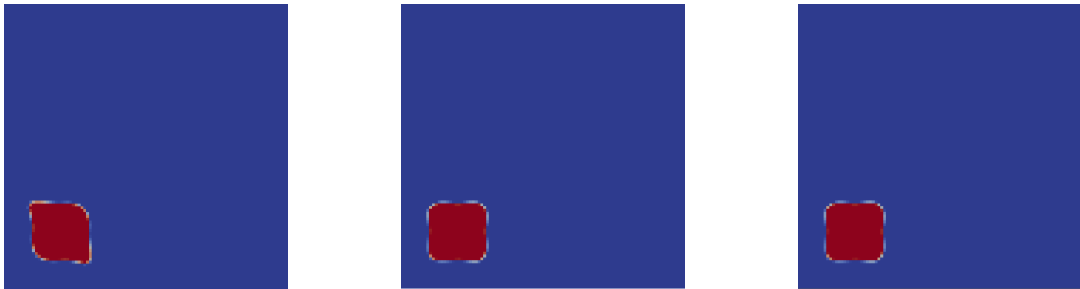


Figure 31: Multimat advection. Direct (left), AD (centre) and DirectCF (right).

In Figure 31, the volume fraction is represented after the two materials have been advected forwards then backwards along the $x = y$ direction on a 100×100 cells domain. The same difficulties as in the mono-material case are shown by the Direct remap, and they are even emphasized by sharp interface reconstruction. The corners in the direction of propagation are diffused whereas the corners in the orthogonal direction are spread. On the contrary, the AD and DirectCF remaps give nice-shaped results. When having a look at Figure 32, it is quite clear that

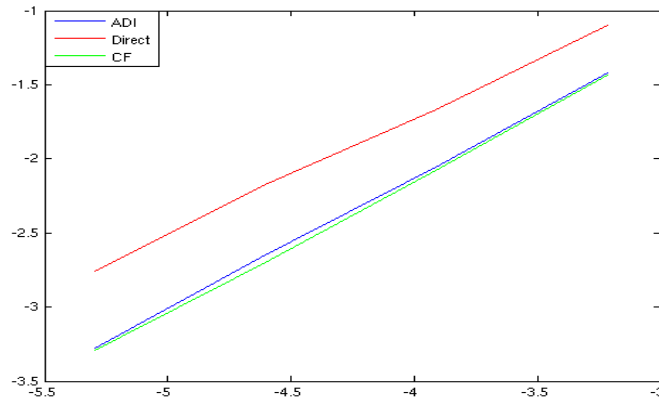


Figure 32: Multimat advection. $\log \|k_1^{final} - k_1^0\|_{L^2}$ in function of $\log(\Delta x)$

the Direct remap can be ruled out because of its lack of accuracy. The rest of this report will focus on the two other remaps.

5.2 Robustness

Robustness is a mandatory characteristic for a scheme to be considered valid, especially in the industry, where in general stability and robustness prevail over accuracy. All simulations presented below are multi-material simulations because they strongly test scheme abilities.

Water-air rotation

Solid rotation (2π) of a square of water into air with velocity $\mathbf{u}(r) = r\omega_0 \mathbf{e}_\theta$ and $\omega_0 = 2\pi 10^3 \text{ s}^{-1}$ at each iteration.

Multimaterial, Air: Perfect Gas $p = (\gamma - 1)\rho e$ with $\gamma = 1.4$

Water: Stiffened Gas $p = (\gamma - 1)\rho e - \pi$ with $(\gamma; \pi) = (7; 2.1 10^9 \text{ Pa})$

Mesh: 400×400 on the domain $[0, 4] \times [0, 4]$.

Even if a simple solid rotation is simulated, this tests robustness because of the gap between the respective densities of air and water, little mistakes can lead to noticeable effects on the variables. By the way, since water sound speed is very high, the time step is very small, which enlightens the defaults of the schemes. However, Figure 33 shows almost no difference between the two results.



Figure 33: Volume fraction at $t = 1\text{ms}$. AD (left) and DirectCF (right).

HAAS test [7]

Interaction between a shock in air and a bubble of Helium. Mesh 1000×90 on the domain $[0, 1000] \times [0, 9]$ cm.

Multimaterial, both Helium and air are Perfect Gases.

All initial data is gathered in the tabular below.

Initial state	Left air state (shock)	Right air state	Bubble
Density ρ (kg.m ³)	1.376363	1	0.18187
Velocity $\mathbf{u} \cdot \mathbf{e}_x$ (m.s ⁻¹)	124.824	0	0
Pressure p (Pa)	$1.5698 \cdot 10^5$	10^5	10^5
Gamma γ	1.4	1.4	1.66

At the initial state, the bubble is at rest and the shock is propagating with velocity over 100 m.s^{-1} .

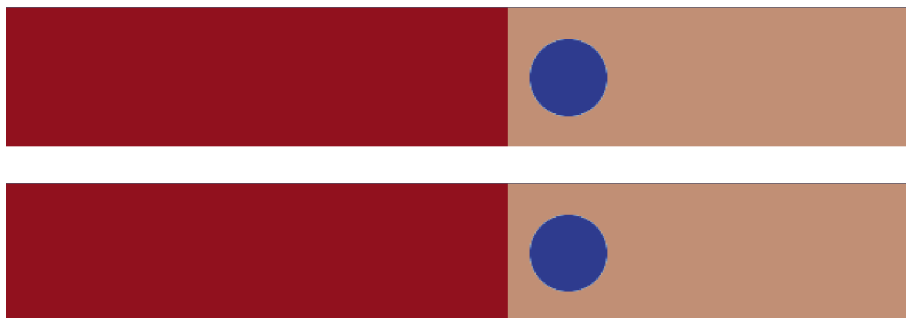


Figure 34: Density at initial state, $t = 0 \text{ s}$. AD top, DirectCF bottom.

The propagation of the shock through the bubble creates a Richtmyer-Meshkov

instability, see Figure 35.



Figure 35: Density at final state, $t = 1$ ms. AD top, DirectCF bottom.



Figure 36: Zoom, density at final state, $t = 1$ ms. AD left, DirectCF right.

Both schemes give the same mushroom-shaped instability, which is the result expected. However, one can observe some differences, in particular at the extremities of the trickle.

Water-water impact

Impact Equal Density Test: Impact of a water drop into air on a water wall. Mesh 320×160 on the domain $[0, 10] \times [0, 5]$ cm. Multimaterial, air Perfect Gas, and water Stiffened Gas. All initial data is gathered in the tabular below.

This test simulates a shock involving materials with very different densities and compressibilities. All figures have been represented using mirror symmetry so that

on each one, results given by the AD remap are on the left, and the symmetric of those given by the DirectCF remap are on the right.

Initial state	Air	Wall	Drop
Density ρ (kg.m ³)	1.29	1000	1000
Velocity $\mathbf{u} \cdot \mathbf{e}_x$ (m.s ⁻¹)	-1000	0	-1000
Pressure p (Pa)	10^5	10^5	10^5
Gamma γ	1.4	7	7
Pi π (Pa)	0	$2.1 \cdot 10^9$	$2.1 \cdot 10^9$

At the initial state, the water wall is at rest and the drop is propagating at 1 km.s^{-1} in the direction of the wall, see Figure 37, $t = 0 \text{ ms}$.

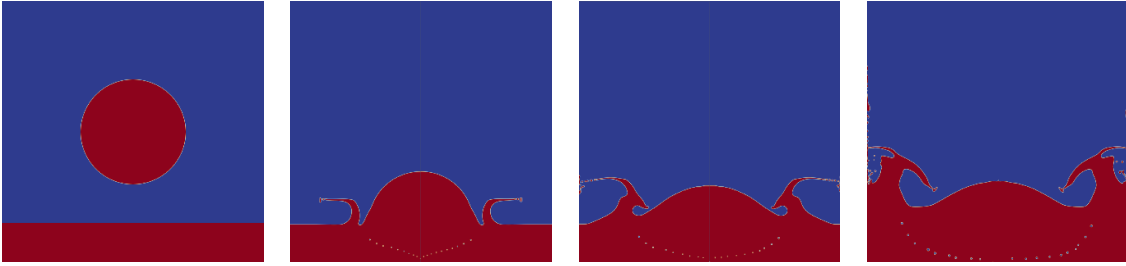


Figure 37: Volume fraction at $t = 0, 3.5, 4.5$ and 6 ms . AD left, DirectCF right.

The drop splashes onto to wall, which creates a jet, see Figure 37 at $t = 3.5 \text{ ms}$. At this step, results given by both schemes are very similar. The only hardly visible difference is the shape of the extremity of the jet. Then, at $t = 4.5 \text{ ms}$, craters have appeared on both sides of the drop, and the jets have reached lateral walls (boundary conditions). The second splashing phenomenon shows some differences between the AD and DirectCF remaps, which are more noticeable at $t = 6 \text{ ms}$, see Figure 38. The shock wave has propagated up to the boundaries and this creates jets in the upper direction. Droplets in those jets seem to slide higher along lateral walls in the AD case than in the DirectCF case.

Once again, the differences between the two remaps are quite weak compared to the severity of this benchmark. Besides, succeeding in simulating this test is a proof of good robustness. Indeed, at the time of the impact, air is enclosed between two water volumes, which generates extremely high pressure bubbles of air trapped into water, that can be seen on Figure 38. There is also high rarefaction of water in the jet, which could create negative pressures of water in some cells, and make the simulation crash.

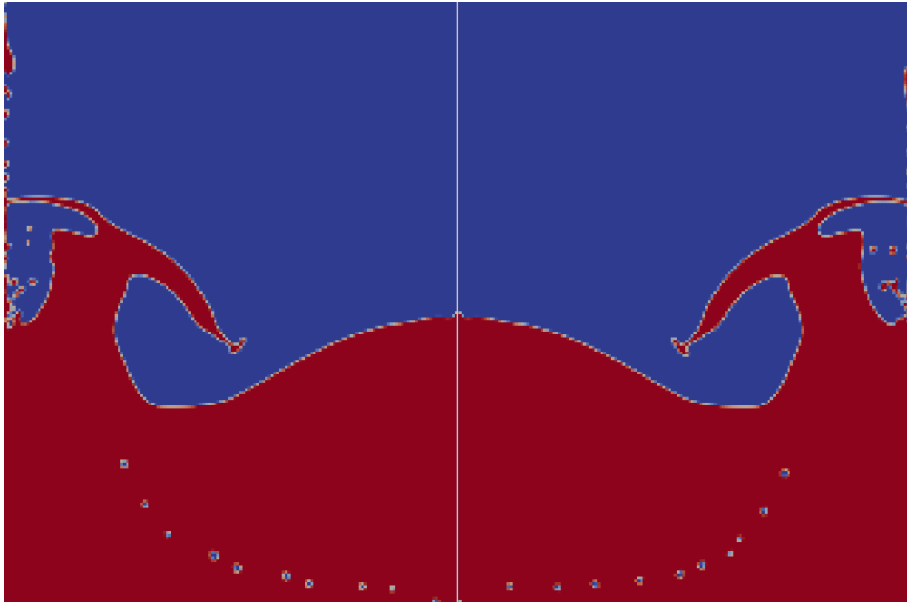


Figure 38: Zoom, volume fraction at final state, $t = 6$ ms. AD left, DirectCF right.

Conclusion

This internship aimed at proposing and testing an alternative remap to the AD remap, more adapted to exascale computers constraints. The Direct remap with Corner Fluxes has been set up theoretically. This remap enables to catch corner effects even better than the AD remap while performing the remap phase in one step. Those two features are crucial since they impact respectively the accuracy and scalability of the scheme. The remap has also been extended to multi-material flows with sharp interface reconstruction. In SHY code, the Direct and DirectCF remaps have been implemented, first in the mono-material case. Then the code has been enhanced with two different multi-fluid models: a mixing model and a model of non-miscible fluids with sharp interface reconstruction.

Numerical results obtained show that the Direct remap is not a good option because of its lack of accuracy, especially in multi-material simulations. They also show that the AD and DirectCF remaps are comparable, in terms of robustness and accuracy, even on severe benchmarks. What follows naturally from this work is the parallelization of the code. Indeed, since the goal was to make the remap scalable in order to be more efficient when running in parallel, the next step will aim at testing this.

Regarding the scheme itself, many perspectives can be considered so far. The main short-term one is probably to fix problems of second order reconstruction of variables at the corners. Direct remapping enables multidimensional reconstruction. So one can imagine a polynomial interpolation on the 9 points stencil of the DirectCF remap.

One could also take into account the rotation of the interface during the Lagrangian phase by computing the vorticity, i.e. $\text{rot } \mathbf{u}$, at the centres of the cells. Another interesting perspective is the implementation of this remap in a cell-centred Lagrangian scheme (i.e. with velocities at the centres of the cells) because first, those schemes are strictly conservative in mass, momentum and total energy, and second, the fact that all variables are centred means that they will be remapped exactly the same way. Thus, one could plan to vectorize the remap in order to run on vector processors. Besides, in the context of multi-material flows simulation, the mixed-cell model could be more advanced, by considering one velocity per material and solving Riemann problems at the interface, see [1]. But in order to do so, it is mandatory for the Lagrangian scheme to be cell-centred.

A Annexe: Étude de la diffusion de la vorticit  en hydrodynamique par diff rents types de sch mas eul riens

Introduction

Le but de cette  tude est de comparer trois diff rents types de sch mas num riques eul riens appliqu s   l'hydrodynamique. Les trois appartiennent   la famille des sch mas Lagrange+Projection. On rappelle que ces sch mas pr sentent deux phases successives pour chaque pas de temps : une phase de d formation du maillage fixe eul rien suivant le mouvement lagrangien du fluide (phase lagrangienne), puis une phase de projection des quantit s lagrangiennes sur le maillage fixe initial. Nous distinguerons donc trois instants : l'instant initial t^n , l'instant correspondant   la fin de la phase lagrangienne t^{lag} , et celui marquant la fin de la phase de projection t^{n+1} , qui est aussi l'instant final de l'it ration n .

Plus pr cis ment, on s'int resse au traitement de la vorticit  sur maillage carr . La m thode mise en  uvre est d' tudier le comportement de plusieurs types de vortex lorsqu'on les laisse  voluer sur un pas temps, pour les sch mas BBC [10], MYR (type VNR [11]) et GLACE [2] (ou EUCCLHYD [9]). Pour cela, on cherche   obtenir des expressions analytiques du champ de vitesse et des grandeurs thermodynamiques associ es   chaque type de vortex, qu'on projette ensuite sur le maillage. Apr s un pas de temps, chaque sch ma donnera un nouvel ensemble de valeurs discr tes pour la vitesse, desquelles on d duira les caract ristiques du nouveau vortex. Il s'agit finalement de comparer les comportements des sch mas en pr sence d'un vortex pur.

A.1 Cadre théorique

On se place dans le cadre de l'hydrodynamique compressible en monomatériau. En négligeant la viscosité du fluide, nous disposons des équations d'Euler, dont on rappelle l'expression ci-dessous :

$$\left\{ \begin{array}{l} \rho \frac{d}{dt} \left(\frac{1}{\rho} \right) - \operatorname{div}(\mathbf{u}) = 0 \\ \rho \frac{d}{dt} \mathbf{u} + \nabla P = 0 \\ \rho \frac{d}{dt} E + \operatorname{div}(\rho u) = 0 \end{array} \right. \quad (6)$$

On ajoute la loi d'état des gaz parfaits (EOS) :

$$P = (\gamma - 1)\rho e$$

A.1.1 Principe

On cherche une solution analytique de (6) sous la forme d'un "vortex pur", en régime stationnaire. En d'autres termes, on impose une vitesse de type $\mathbf{u}(t, r, \theta) = u_\theta(r)\mathbf{e}_\theta$ en polaires (on se place dans le cas 2D). Cette condition implique que $\operatorname{div}(\mathbf{u}) = 0$, ce qui nous place dans le cas incompressible.

On peut donc écrire qu'il existe un vecteur Ψ appelé vecteur fonction de courant tel que $\mathbf{u} = \operatorname{rot}(\Psi)$. On a alors :

$$\operatorname{rot} \operatorname{rot}(\Psi) = \nabla(\operatorname{div}(\Psi)) - \Delta\Psi = \operatorname{rot}(\mathbf{u})$$

En 2D, on a $\operatorname{rot}(\mathbf{u}) = \operatorname{rot}(\mathbf{u})\mathbf{e}_z$. Si de plus on choisit Ψ à divergence nulle, et qu'on pose $\Psi = \psi_z \mathbf{e}_z$, il vient :

$$\operatorname{rot}(\mathbf{u}) = -\Delta\psi_z \quad (7)$$

Autrement dit, étant donné un champ de vortacité $\operatorname{rot}(\mathbf{u})$, ψ_z est solution de l'équation de Poisson.

Dans le cadre de notre problème, comme \mathbf{u} ne dépend que de r , ψ_z aussi. L'équation (7) devient :

$$\frac{1}{r} \frac{\partial}{\partial r} (ru_\theta) = -\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \psi_z}{\partial r} \right)$$

$$\text{d'où } \boxed{u_\theta = -\frac{\partial \psi_z}{\partial r}}$$

Ainsi, si on fixe un type de vortex, i.e un champ de vortacité $\operatorname{rot}(\mathbf{u})$, on peut retrouver le u_θ associé par l'intermédiaire de ψ_z , obtenu en résolvant (7). Les expressions des autres grandeurs du problème s'obtiennent en injectant u_θ dans le système (6).

A.1.2 Rotationnel numérique

Dans la mesure où on s'intéresse à la vorticit , il s'agit de pouvoir la mesurer   partir de valeurs num riques. Nous devons donc   ce stade d finir un rotationnel discret. On sait d'apr s la formule de Stokes que :

$$\iint_S \text{rot} \mathbf{u} \, dS = \oint_{\partial S} \mathbf{u} \cdot d\mathbf{l}$$

On veut pouvoir d finir $\text{rot} \mathbf{u}$ en un point du maillage. Nous avons a priori 3 lieux possibles : aux centres des mailles primales, aux n uds (centres des mailles duales), ou aux milieux des faces. Usuellement, le contour choisi pour le calcul de la circulation a la m me taille et la m me forme qu'une maille (dans notre cas un carr  de c t  Δx), ce qui permet de donner une valeur au centre de ce dernier. Mais pour cela il faut  tre capable de d finir les vitesses sur les faces du contour. De fait, les vitesses n' tant pas localis es aux m mes endroits pour les trois sch mas, si on veut que notre  tude comparative ait un sens, il va falloir distinguer les cas pour que le choix du contour soit coh rent avec le sch ma  tudi . En termes plus math matiques, on va chercher   d finir $\text{rot}(\mathbf{u})$ aux centres des mailles d'un maillage conforme dans $H(\text{rot})$.

Sch mas d cal s, vitesses aux n uds (MYR) : ∂S contour d'une maille primaire

Pour ce type de sch mas, les vitesses sont aux n uds du maillage primal. On peut montrer qu'il s'agit d'un bon maillage pour d finir le rotationnel puisqu'il est conforme dans $H^1 \supset H(\text{rot})$. D'un point de vue pratique, en reconstruisant lin airement la vitesse sur une face (comme on le fait pour calculer les flux dans la projection de MYR), on est capable de donner la valeur de l'int grale de $\mathbf{u} \cdot d\mathbf{l}$ sur cette derni re. Si on indice sur p les sommets de la maille c au centre de laquelle on cherche   calculer le rotationnel, il vient :

$$\text{rot}(\mathbf{u})_{c,D} = \frac{1}{2\Delta x} \sum_{p \in \{c\}} (\mathbf{u}_p + \mathbf{u}_{p+1}) \cdot \mathbf{T}_{p,p+1}$$

o  $\mathbf{T}_{p,p+1}$ repr sente le vecteur tangent   la face $[p, p+1]$, dirig  dans le sens direct (voir Figure 39).

Sch mas centr s, vitesses aux centres des mailles (GLACE) : ∂S contour d'une maille duale

Ces sch mas ont les vitesses localis es aux centres des mailles primales, qui correspondent aux n uds du maillage dual. Ici,   l'inverse, le maillage primal n'est conforme que dans L^2 , et c'est le maillage dual, conforme dans $H^1 \supset H(\text{rot})$, qui sera adapt    l' criture d'un rotationnel discret. On proc de donc de la m me mani re que pour les sch mas d cal s, mais sur le maillage dual. Cette fois la vitesse est constante par morceaux sur une face de maille duale. Si on l'int gre, on obtient

toutefois le même résultat que pour une reconstruction linéaire, en indiquant sur c les mailles primales qui partagent le nœud commun p , auquel on évalue la valeur de $\text{rot}(\mathbf{u})$ en p .

$$\text{rot}(\mathbf{u})_{p,C} = \frac{1}{2\Delta x} \sum_{c \in \{p\}} (\mathbf{u}_c + \mathbf{u}_{c+1}) \cdot \mathbf{T}_{c,c+1}$$

où $\mathbf{T}_{c,c+1}$ représente le vecteur tangent à la face duale reliant les centres des mailles c et $c + 1$, dirigé dans le sens direct (voir Figure 39).

Schémas décalés, vitesses aux faces (BBC) : ∂S contour d'une maille duale

Dans ce dernier cas, les vitesses sont définies aux milieux des faces du maillage primal, et dirigées selon les normales à ces faces. On peut montrer dans ce cas que le maillage primal est conforme dans $H(\text{div})$, et que le maillage dual est lui conforme dans $H(\text{rot})$. En pratique, si on choisissait une maille primaire comme contour, on obtiendrait un résultat nul (vitesses orthogonales aux vecteurs tangents aux faces). Il convient donc là encore de placer les valeurs du rotationnel aux nœuds du maillage primal. On connaît alors la vitesse tangentielle sur chaque face du contour, ce qui permet d'écrire :

$$\text{rot}(\mathbf{u})_{p,BBC} = \frac{1}{\Delta x} \sum_{c \in \{p\}} \mathbf{u}_{c,c+1} \cdot \mathbf{T}_{c,c+1}$$

où $\mathbf{u}_{c,c+1}$ correspond à la vitesse de la face séparant les mailles c et $c + 1$ (voir Figure 39).

Il reste maintenant à faire évoluer la solution sur un pas de temps (entre t^n et $t^{n+1} = t^n + \Delta t$) pour chaque type de schémas. On calculera ensuite le rotationnel numérique à l'origine dans chaque cas, en prenant soin de placer l'origine sur un nœud dans les cas centré et BBC, et au centre d'une maille dans le cas décalé. De plus, comme on s'intéresse au caractère diffusif, on calculera également le rapport entre les rotationnels é t^n et t^{n+1} , pour s'affranchir des éventuelles différences dues à la discrétisation par les schémas (par exemple il se peut qu'on ait $\text{rot}(\mathbf{u})_{p,C}^n \neq \text{rot}(\mathbf{u})_{p,BBC}^n$, voir 2.4). On comparera ce rapport à 1, et on fera apparaître un coefficient de diffusion numérique.

A.2 Cas d'un vortex ponctuel

La représentation qui semble la plus naturelle pour le champ de vorticité créé par un vortex ponctuel placé à l'origine est la masse de Dirac. On note σ_0 le moment cinétique introduit par le vortex.

$$\text{rot}(\mathbf{u}) = \sigma_0 \delta_0$$

Il s'agit bien d'un moment car en dimension 2, $[\delta_0(r)] = m^{-2}$ et $[\text{rot}(\mathbf{u})] = s^{-1}$, donc σ_0 s'exprime en $m^2 \cdot s^{-1}$. Un vortex ponctuel correspond en fait à un vortex qui conserve le moment cinétique.

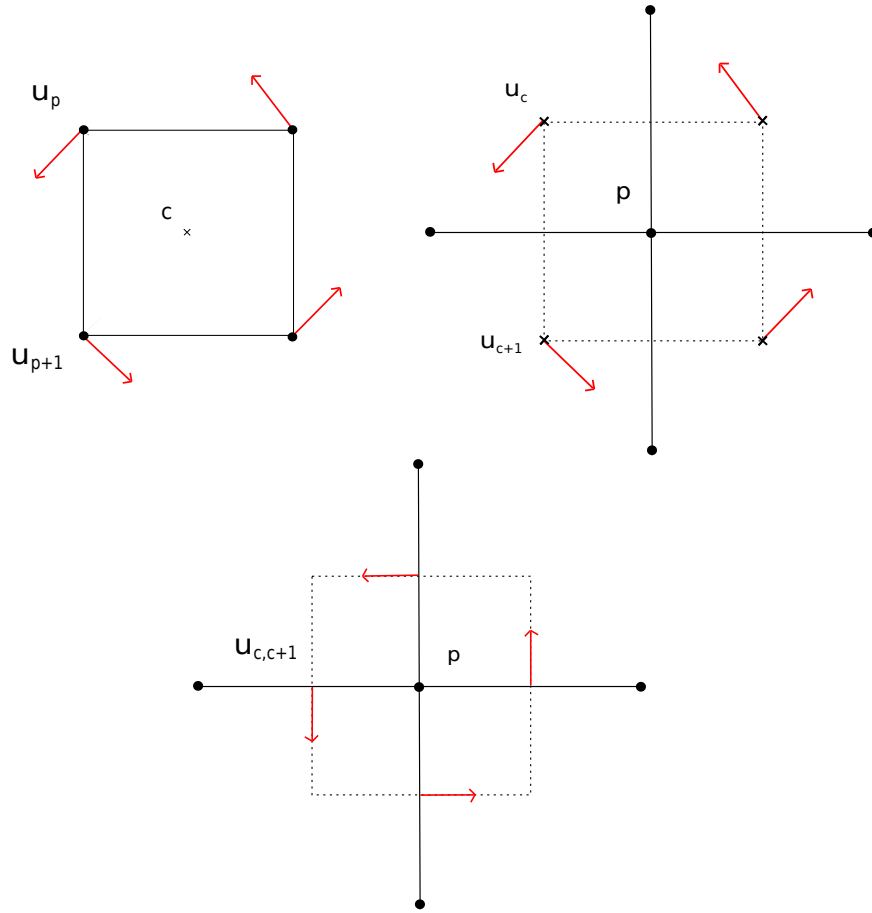


Figure 39: Schéma de la maille sur laquelle on définit $\text{rot}(\mathbf{u})$ pour les schémas décalé, centré et BBC (de gauche à droite et de haut en bas).

A.2.1 Calcul de la solution analytique

D'après le paragraphe 1.1, pour trouver l'expression de la fonction de courant, on est donc amené à résoudre l'équation :

$$-\Delta\psi_z = \sigma_0\delta_0 = \text{rot}(\mathbf{u})$$

La solution n'est autre que la fonction de Green du Laplacien, i.e en 2D

$$\psi_z(r) = -\frac{\sigma_0}{2\pi} \ln r$$

On obtient donc finalement :

$$u_\theta(r) = -\frac{\partial\psi_z}{\partial r} = \frac{\sigma_0}{2\pi r}$$

On va montrer qu'il existe une densité ρ (et une pression P associée) telle que (\mathbf{u}, ρ, P) soit solution du système d'Euler.

On a déjà vu en 1.1 que $\text{div}(\mathbf{u}) = 0$. On s'intéresse à la deuxième équation de (6), qui donne elle :

$$-\frac{u_\theta^2}{r} = -\frac{1}{\rho} \frac{\partial P}{\partial r}$$

$$\text{d'où } \rho(r) = \rho_0 \exp\left(-\frac{K_0}{r^2}\right), \text{ avec } K_0 = \frac{\sigma_0^2}{8\pi^2(\gamma-1)e_0}$$

On voit que $\rho(r) \rightarrow \rho_0$ lorsque $r \rightarrow +\infty$.

On définit α_0 tel que $u_\theta\left(\frac{\Delta x}{\sqrt{2}}\right) = \sqrt{2} \alpha_0 \Delta x$. α_0 est une constante dépendante du pas du maillage et de σ_0 , qu'on introduit pour simplifier les calculs.

$$\alpha_0 = \frac{\sigma_0}{2\pi \Delta x^2}$$

Elle peut être vue comme une vitesse angulaire.

On peut considérer que $\rho \simeq \rho_0$ si $\frac{K_0}{R^2} \ll 1$, où R représente le "rayon" d'une maille. Cette condition implique que :

$$\Delta x \gg \frac{\sigma_0}{2\pi \sqrt{(\gamma-1)e_0}} = \frac{\sigma_0}{2\pi c_0}$$

Ainsi, pour une valeur de σ_0 suffisamment petite devant la vitesse du son, et un pas de maillage grand devant ce rapport, on peut considérer que sur notre maillage, la solution analytique de (6) associée à un vortex ponctuel est assimilable au triplet $\left(\frac{\sigma_0}{2\pi r} \mathbf{e}_\theta, \rho_0, (\gamma-1)\rho_0 e_0\right)$.

A.2.2 Etude pour les schémas décalés

Le modèle de schéma utilisé est le schéma MYR d'ordre 1 en espace. Il présente la même discrétisation en espace que VNR, mais la discrétisation en temps de type saute-mouton a été remplacé par un schéma prédicteur-correcteur. Comme on l'a vu, les vitesses sont localisées aux nœuds, et le reste des grandeurs aux centres des mailles. Le centre du vortex est placé au centre d'une maille.

Détermination de \mathbf{u} sur le maillage

On commence par calculer les vitesses aux nœuds à t^n (obtenues à partir de la solution analytique), dont nous avons besoin pour pouvoir faire évoluer le maillage durant la phase lagrangienne.

$$r_1 = \frac{\Delta x}{\sqrt{2}}, \quad r_2 = \sqrt{\frac{5}{2}} \Delta x, \quad r_3 = 3 \frac{\Delta x}{\sqrt{2}}$$

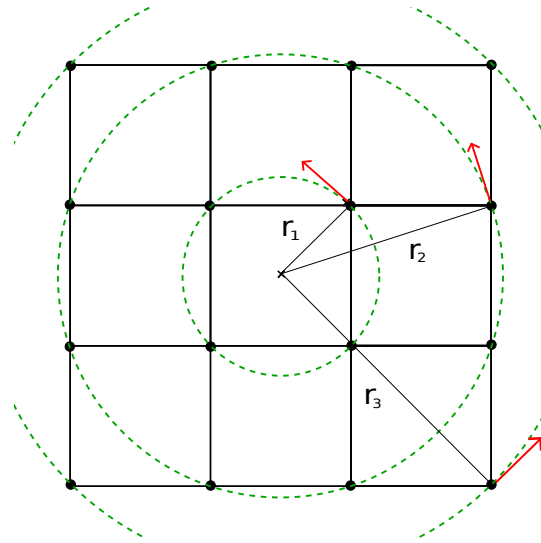


Figure 40: Schéma du champ de vitesses sur le maillage dans le cas décalé.

Et donc, pour les normes des vitesses associées à ces rayons :

$$u_1 = \sqrt{2}\alpha_0\Delta x, \quad u_2 = \sqrt{\frac{2}{5}}\alpha_0\Delta x, \quad u_3 = \frac{\sqrt{2}}{3}\alpha_0\Delta x$$

Il ne reste qu'à multiplier par le \mathbf{e}_θ correspondant pour obtenir les vecteurs vitesse.

Phase lagrangienne

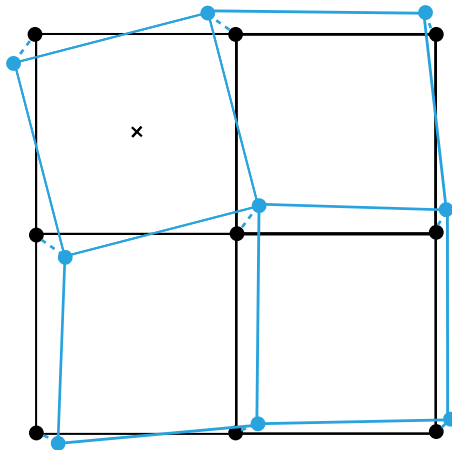


Figure 41: Schéma de la déformation lagrangienne dans le cas décalé pour un vortex ponctuel.

Puisque les vitesses nodales sont connues à t^n , on connaît directement la déformation lagrangienne du maillage, voir Figure 41. On garde les notations utilisées dans [3]. Les cellules sont indicées sur c et les nœuds sur p . On a, au nœud p :

$$\mathbf{u}_p^{lag} = \mathbf{u}_p^n + \Delta t \boldsymbol{\gamma}_p^{lag}$$

$$\text{avec } \gamma_p^{lag} = \frac{1}{m_p} \sum_{c \in \{p\}} (P_c^{lag} + Q_c^n) L_{cp}^{lag} \mathbf{N}_{cp}^{lag} = 0$$

Cela vient du fait que nous avons un champ de pression uniforme $P_c^{lag} = P_0, \forall c$, et que $\sum L_{cp}^{lag} \mathbf{N}_{cp}^{lag} = 0$, ce qui traduit la conservation de volume de chaque maille. Cette conservation de volume justifie au passage la nullité du terme de pseudo-viscosité Q_c^n .

Il vient donc, pour tout noeud p ,

$$\boxed{\mathbf{u}_p^{lag} = \mathbf{u}_p^n}$$

Phase de projection

La projection effectuée est une projection directe. On calcule les flux de masse aux faces de la maille nodale ϕ_p^α , ($\alpha = E, O, N, S$), en faisant la moyenne entre les flux aux faces des 4 mailles voisines du noeud. Ensuite, on écrit la conservation de la quantité de mouvement dans la maille nodale.

$$m_p^{proj} \mathbf{u}_p^{proj} = m_p^{lag} \mathbf{u}_p^{lag} + \sum_{\alpha} \phi_p^\alpha \mathbf{u}_p^\alpha \quad (8)$$

Dans notre cas, on a $m_p^{proj} = m_p^{lag} = \rho_0 \Delta x^2$ puisque le volume et la densité sont conservés.

Pour illustrer cette étape, prenons le noeud p correspondant au noeud en bas à droite (les autres s'en déduisent par symétrie). A l'ordre 1, les vitesses \mathbf{u}_p^α sont choisies comme suit :

$$\mathbf{u}_p^O = \alpha_0 \Delta x \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \mathbf{u}_p^S = \frac{\alpha_0 \Delta x}{5} \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \quad \mathbf{u}_p^E = \frac{\alpha_0 \Delta x}{5} \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \mathbf{u}_p^N = \alpha_0 \Delta x \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Ce qui donne donc, à l'ordre 1 :

$$\boxed{\mathbf{u}_{p,1}^{n+1} = \alpha_0 \Delta x \begin{pmatrix} 1 + \frac{68}{75} \alpha_0 \Delta t \\ 1 - \frac{68}{75} \alpha_0 \Delta t \end{pmatrix}}$$

Pour obtenir une projection d'ordre 2, il suffit de procéder à une reconstruction linéaire pour les \mathbf{u}_p^α . Ce qui revient à poser :

$$\mathbf{u}_p^O = \alpha_0 \Delta x \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{u}_p^S = \frac{\alpha_0 \Delta x}{5} \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \quad \mathbf{u}_p^E = \frac{\alpha_0 \Delta x}{5} \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \quad \mathbf{u}_p^N = \alpha_0 \Delta x \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

On obtient donc, pour l'ordre 2 :

$$\boxed{\mathbf{u}_{p,2}^{n+1} = \alpha_0 \Delta x \begin{pmatrix} 1 + \frac{34}{75} \alpha_0 \Delta t \\ 1 - \frac{34}{75} \alpha_0 \Delta t \end{pmatrix}}$$

Calcul des rotationnels discrets

D'après 1.2, comme ici $\frac{1}{2}(\mathbf{u}_p^n + \mathbf{u}_{p+1}^n) \cdot \mathbf{T}_{p,p+1} = \alpha_0 \Delta x$,

$$\text{rot}(\mathbf{u})_{c,D}^n = 4 \alpha_0 = \frac{\sigma_0}{\pi (\Delta x / \sqrt{2})^2}$$

Notons que $\text{rot}(\mathbf{u})_d \rightarrow +\infty$ lorsque $\Delta x \rightarrow 0$, ce qui est cohérent avec notre choix de représentation par une masse de Dirac.

De plus, à t^{n+1} , compte tenu des résultats obtenus dans la partie précédente, on a quel que soit l'ordre :

$$\frac{1}{2}(\mathbf{u}_p^{n+1} + \mathbf{u}_{p+1}^{n+1}) \cdot \mathbf{T}_{p,p+1} = \alpha_0 \Delta x$$

et donc,
$$\boxed{\frac{\text{rot}(\mathbf{u})_{c,D}^{n+1}}{\text{rot}(\mathbf{u})_{c,D}^n} = 1}$$

Le schéma décalé ne diffuse donc pas la vorticit  des vortex ponctuels, quel que soit l'ordre choisi pour la projection.

En revanche, on remarque que $\|\mathbf{u}_p^{n+1}\| > \|\mathbf{u}_p^n\|$, ce qui peut ˆtre une source d'instabilit . De plus, la direction des vitesses change elle aussi, on a apparition d'une composante strictement positive selon \mathbf{e}_r ,  gale   $\frac{68}{75} \alpha_0 \Delta t (\sqrt{2} \alpha_0 \Delta x)$ pour la projection d'ordre 1, et   $\frac{34}{75} \alpha_0 \Delta t (\sqrt{2} \alpha_0 \Delta x)$ pour l'ordre 2. On observe donc que le vortex a tendance   s' tirer, et le ph nom ne est logiquement att nu    l'ordre 2.

A.2.3 Etude pour les sch mas centr s

Le mod le de sch ma utilis  est dans ce cas le sch ma GLACE d'ordre 1, voir [2]. Toutes les grandeurs sont centr es, et on place le centre du vortex sur un n ud du maillage.

D termination de \mathbf{u} sur le maillage

Commen ons par d terminer les valeur des vitesses centr es sur le maillage. On a cette fois :

$$r_1 = \frac{\Delta x}{\sqrt{2}}, \quad r_2 = \sqrt{\frac{5}{2}} \Delta x, \quad r_3 = \frac{3\sqrt{2}}{2} \Delta x$$

$$\text{et donc : } u_1 = \sqrt{2} \alpha_0 \Delta x, \quad u_2 = \sqrt{\frac{2}{5}} \alpha_0 \Delta x, \quad u_3 = \frac{\sqrt{2}}{3} \alpha_0 \Delta x$$

Comme pr c demment, il suffit ensuite de multiplier par le bon \mathbf{e}_θ .

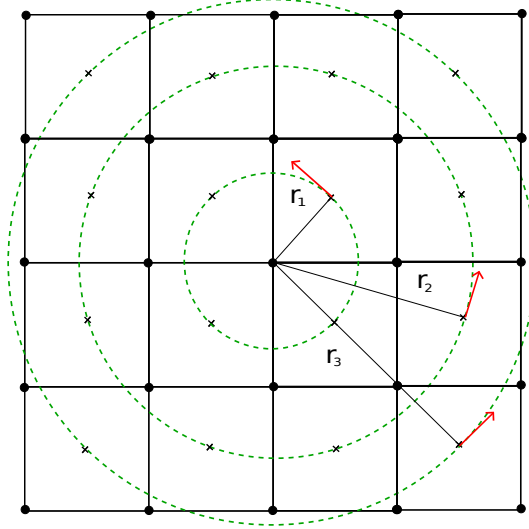


Figure 42: Schéma du champ de vitesses sur le maillage dans le cas centré.

Phase lagrangienne

Contrairement au cas décalé, on ne dispose pas des vitesses aux noeuds, il nous faut donc les calculer pour pouvoir déformer le maillage. Cette étape cruciale repose sur la construction d'un solveur aux noeuds, qui permet de déterminer non seulement les \mathbf{u}_p , mais aussi les pressions aux demi-faces voisines du noeud p . Ce solveur provient de la réécriture des équations de conservation qui, pour un champ de pression P_0 uniforme donne :

$$\sum_{c \in \{p\}} (P_{c-1/2}^{c-1} - P_{c-1/2}^c) \Delta x \mathbf{N}_{c-1}^c = 0 \quad (9)$$

$$\begin{cases} P_{c-1/2}^c &= P_0 + \rho_0 c_0 (\mathbf{u}_p - \mathbf{u}_c) \cdot \mathbf{N}_{c-1}^c \\ P_{c-1/2}^{c-1} &= P_0 - \rho_0 c_0 (\mathbf{u}_p - \mathbf{u}_{c-1}) \cdot \mathbf{N}_{c-1}^c \end{cases}$$

Une fois les vitesses aux noeuds, et donc la déformation lagrangienne, connues (voir Figure 43), on calcule les nouvelles vitesses centrées. Pour ce type de schémas, on obtient lors de la phase lagrangienne :

$$\mathbf{u}_c^{lag} = \mathbf{u}_c^n - \frac{\Delta t}{m_c} \sum_{p \in \{c\}} \frac{1}{2} (P_p^{c,p+1} + P_{p+1}^{c,p}) \Delta x \mathbf{N}_{p,p+1} \quad (10)$$

où $P_p^{c,p+1}$ représente la pression exercée sur la demi-face de $[p, p+1]$ correspondant au noeud p dans la maille c , et $\mathbf{N}_{p,p+1}$ la normale sortante à la face $[p, p+1]$.

Pour la maille en bas à droite, les calculs donnent :

$$\mathbf{u}_c^{lag} = \alpha_0 \Delta x \begin{pmatrix} 1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 \\ 1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 \end{pmatrix}$$

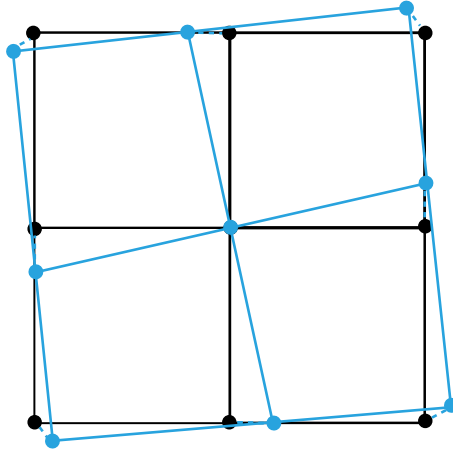


Figure 43: Schéma de la déformation lagrangienne dans le cas centré pour un vortex ponctuel.

Phase de projection

Comme pour le cas décalé, on va calculer les flux de masse algébriques, mais cette fois ce sont les flux à travers les faces de la maille réelle. Puis, comme pour (9), on écrit l'équation de la conservation de la quantité de mouvement, en utilisant ici un décentrement upwind pour le calcul des \mathbf{u}_c^α . On obtient finalement pour la maille du bas, à l'ordre 1 en Δt :

$$\mathbf{u}_c^{n+1} = \alpha_0 \Delta x \begin{pmatrix} 1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 - \frac{8}{75} \alpha_0 \Delta t \\ 1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 - \frac{76}{75} \alpha_0 \Delta t \end{pmatrix}$$

Calcul des rotationnels discrets

D'après 1.2, comme ici $\frac{1}{2}(\mathbf{u}_c^n + \mathbf{u}_{c+1}^n) \cdot \mathbf{T}_{c,c+1} = \alpha_0 \Delta x$,

$$\text{rot}(\mathbf{u})_{p,C}^n = 4 \alpha_0 = \frac{\sigma_0}{\pi (\Delta x / \sqrt{2})^2}$$

On remarque qu'on a égalité des valeurs du rotationnel numérique à t^n , ce qui confirme le bon choix des contours.

Et à t^{n+1} , à partir des résultats précédents, il vient :

$$\frac{\text{rot}(\mathbf{u})_{p,C}^{n+1}}{\text{rot}(\mathbf{u})_{p,C}^n} = 1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 - \frac{14}{25} \alpha_0 \Delta t$$

Le schéma centré a une tendance à diffuser la vorticit  des vortex ponctuels. En ce qui concerne les vitesses, contrairement au cas d cal , on a toujours $\|\mathbf{u}_p^{n+1}\| < \|\mathbf{u}_p^n\|$. On observe  galement le ph nom ne d' tirement vu dans le cas d cal , mais dans une moindre mesure.

A.2.4 Etude pour les schémas de type BBC

Le modèle de schéma utilisé est le schéma BBC, pour lequel les vitesses sont localisées aux centres des faces du maillage, et dirigées uniquement selon les normales à ces faces. Les grandeurs thermodynamiques, quant à elles, sont toujours localisées aux centres des mailles. Conformément à ce qui a été vu en 1.2, on place le centre du vortex sur un nœud du maillage.

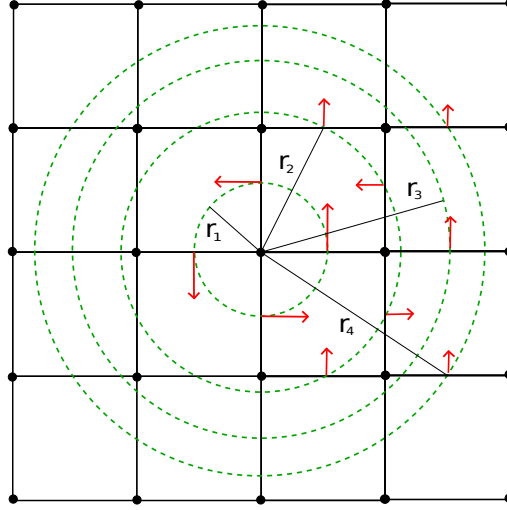


Figure 44: Schéma du champ de vitesses sur le maillage pour les schémas BBC.

Détermination de u sur le maillage

Le calcul des vitesses aux faces nécessite de connaître les rayons associés.

$$r_1 = \frac{1}{2}\Delta x, \quad r_2 = \frac{\sqrt{5}}{2}\Delta x, \quad r_3 = \frac{3}{2}\Delta x, \quad r_4 = \frac{\sqrt{13}}{2}\Delta x$$

Puis, on a pour tout $i \in [1, 4]$, $u_i = u_\theta(r_i)\mathbf{e}_{\theta_i} \cdot \mathbf{e}_x$, pour $\theta_i \in [0, \frac{\pi}{4}]$, d'où

$$u_1 = 2\alpha_0\Delta x, \quad u_2 = \frac{2}{5}\alpha_0\Delta x, \quad u_3 = \frac{2}{3}\alpha_0\Delta x, \quad u_4 = \frac{6}{13}\alpha_0\Delta x$$

Phase lagrangienne

Pour la phase lagrangienne, la discrétisation en temps est effectuée en trois étapes, de t^n à $t^{n+1/4}$, de $t^{n+1/4}$ à $t^{n+1/2}$, et de $t^{n+1/2}$ à t^{lag} . Pour une maille représentée par (i, j) , ses faces verticales sont désignées par $(i \pm 1/2, j)$, et ses faces horizontales par $(i, j \pm 1/2)$.

Première étape : On calcule les vitesses aux faces à $t^{n+1/4}$ à partir des forces de pression et des termes de pseudo-viscosité. Dans notre cas, le maillage carré, le champ

de pression uniforme et l'absence de termes de pseudo-viscosité (conservation du volume de chaque maille) donnent, pour la face de gauche :

$$ux_{i-1/2,j}^{n+1/4} = ux_{i-1/2,j}^n$$

Il en est de même pour toutes les faces, ainsi le champ de vitesse à $t^{n+1/4}$ est le même qu'à t^n .

Deuxième étape : On calcule la variation de volume

$$dV_{i,j}^{n+1/2} = \frac{\Delta x}{\Delta t/2} \left(ux_{i+1/2,j}^{n+1/4} - ux_{i-1/2,j}^{n+1/4} + uy_{i,j+1/2}^{n+1/4} - uy_{i,j-1/2}^{n+1/4} \right) = 0$$

Ce qui donne finalement pour la face de gauche (il en va là encore de même pour les autres) :

$$ux_{i-1/2,j}^{n+1/2} = ux_{i-1/2,j}^n$$

Le champ de vitesse, et les autres grandeurs, n'évoluent pas entre t^n et $t^{n+1/2}$.

Troisième étape : On calcule la nouvelle variation de volume

$$dV_{i,j}^{lag} = \frac{\Delta x}{\Delta t} \left(ux_{i+1/2,j}^{n+1/2} - ux_{i-1/2,j}^{n+1/2} + uy_{i,j+1/2}^{n+1/2} - uy_{i,j-1/2}^{n+1/2} \right) = 0$$

Ce qui assure finalement la conservation des grandeurs thermodynamiques entre t^n et t^{lag} . Pour les vitesses, on utilise un schéma centré en temps :

$$ux_{i+1/2,j}^{lag} = 2 ux_{i+1/2,j}^{n+1/2} - ux_{i+1/2,j}^n = ux_{i+1/2,j}^n$$

Le champ de vitesse à t^{lag} est donc lui aussi le même qu'à t^n .

Phase de projection

La phase de projection est effectuée par splitting d'opérateur par directions alternées. On commence par exemple par la projection en X . On calcule les flux de masse traversant chaque maille des maillages primal (pour les ux) et dual (pour les uy). On en déduit les flux de quantité de mouvement, en utilisant un décentrement upwind (à l'ordre 1) pour les vitesses. Et la conservation de la quantité de mouvement donne les nouvelles vitesses aux faces X et Y . Ces nouvelles vitesses induisent une nouvelle déformation des maillages, à partir de laquelle on effectue la projection en Y .

On cherche à calculer les vitesses aux faces situées respectivement en-dessous et à droite du nœud où est placé le centre du vortex. On se place pour cela dans la maille en bas à droite de ce sommet, que l'on note (i, j) . On s'intéresse donc aux faces $(i + 1/2, j)$ et $(i, j + 1 - 2)$.

Les calculs donnent, à l'ordre 1 en Δt :

$$\boxed{ux_{i-1/2,j}^{n+1} = uy_{i,j+1/2}^{n+1} = 2\alpha_0\Delta x \left(1 - \frac{48}{25}\alpha_0\Delta t \right)}$$

Calcul des rotationnels discrets

Comme pour les schémas centrés, on calcule la circulation sur le contour de la maille duale. Ainsi, comme on a $\mathbf{u}_{c,c+1}^n \cdot \mathbf{T}_{c,c+1} = 2\alpha_0 \Delta x$,

$$\text{rot}(\mathbf{u})_{p,BBC}^n = 8\alpha_0 = \frac{8\sigma_0}{2\pi(\Delta x)^2}$$

On remarque qu'il y a une différence d'un facteur 2 entre $\text{rot}(\mathbf{u})_{c,D}^n = \text{rot}(\mathbf{u})_{p,C}^n$ et $\text{rot}(\mathbf{u})_{p,BBC}^n$. En effet, la vitesse u_θ étant une fonction décroissante de r , les premières valeurs dont dispose le schéma BBC pour traduire la vorticit  (celles aux milieux des faces de la maille duale centrale, donc   $\Delta x/2$ du centre) sont plus  lev es que celles dont disposent les sch emas d cal  et centr  (respectivement celles aux noeuds de la maille primale centrale et de la maille duale centrale, donc   $\Delta x/\sqrt{2}$ du centre). Par ailleurs, compte tenu des valeurs des $\mathbf{u}_{c,c+1}^{n+1}$, on obtient :

$$\boxed{\frac{\text{rot}(\mathbf{u})_{p,BBC}^{n+1}}{\text{rot}(\mathbf{u})_{p,BBC}^n} = 1 - \frac{48}{25}\alpha_0 \Delta t}$$

Le sch ema BBC diffuse, comme le sch ema centr , la vorticit  des vortex ponctuels.

A.3 Cas d'un vortex "id al"

On entend par vortex "id al" un vortex qui induit un mouvement de la mati re dans son ensemble, c'est- -dire qui conserve la vitesse angulaire (rotation solide). Si la vitesse angulaire est la m me en tout point, on a donc un champ de vorticit  uniforme :

$$\text{rot}(\mathbf{u}) = \omega_0$$

A.3.1 Calcul de la solution analytique

On doit cette fois r soudre l' quation suivante pour trouver ψ_z :

$$-\Delta\psi_z = \omega_0$$

Ce qui donne directement pour la vitesse :

$$u_\theta(r) = -\frac{\partial\psi_z}{\partial r} = \frac{\omega_0}{2}r$$

On cherche alors la densit  ρ (et la pression P associ e) telle que (\mathbf{u}, ρ, P) soit solution du syst me d'Euler. On rappelle la deuxi me  quation de (6) :

$$-\frac{u_\theta^2}{r} = -\frac{1}{\rho} \frac{\partial P}{\partial r}$$

qui donne $\rho(r) = \rho_0 \exp(M_0 r^2)$, avec $M_0 = \frac{\omega_0^2}{8(\gamma - 1)e_0}$

On voit que $\rho(r) \rightarrow \rho_0$ lorsque $M_0 r^2 \rightarrow 0$.

On peut supposer que $\rho \simeq \rho_0$ si $M_0 r^2 \ll 1$, pour les r considérés ici. Pour les schémas que nous utilisons, nous ne dépassons pas $r_{max} = 2\sqrt{2} \Delta x$. La condition précédente revient alors à :

$$\Delta x \ll \frac{c_0}{\omega_0}$$

Ainsi, pour un maillage suffisamment fin, ou pour un ω_0 tel que $\omega_0 \Delta x$ petit devant la vitesse du son, le triplet $(\frac{\omega_0}{2} r \mathbf{e}_\theta, \rho_0, (\gamma - 1)\rho_0 e_0)$ représente sur notre maillage la solution analytique des équations d'Euler incompressible associée à un champ de vorticit   uniforme.

Dans la suite, pour simplifier les calculs et adopter une notation similaire au cas du vortex ponctuel, on va plut  t travailler avec $\boldsymbol{\alpha}_0 = \boldsymbol{\omega}_0/2$ de sorte que $u_\theta(r) = r\alpha_0$, qui repr  sente donc la vitesse angulaire r  elle en tout point.

A.3.2 Etude pour les sch  mas d  cal  s

D  termination de \mathbf{u} sur le maillage

Comme pour le cas du vortex ponctuel, on calcule les vitesses aux n  uds    t^n (projet  es    partir de la solution analytique). On a   videmment les m  mes rayons puisque le maillage n'a pas chang  .

$$r_1 = \frac{\Delta x}{\sqrt{2}}, \quad r_2 = \sqrt{\frac{5}{2}} \Delta x, \quad r_3 = 3 \frac{\Delta x}{\sqrt{2}}$$

$$\text{et donc : } u_1 = \frac{1}{\sqrt{2}} \alpha_0 \Delta x, \quad u_2 = \sqrt{\frac{5}{2}} \alpha_0 \Delta x, \quad u_3 = \frac{3}{\sqrt{2}} \alpha_0 \Delta x$$

L   encore il ne reste qu'   multiplier par le \mathbf{e}_θ correspondant pour obtenir les vecteurs vitesse.

Phase lagrangienne

La conservation du volume de chaque maille et le champ de pression uniforme impliquent que, pour tout n  ud p :

$$\mathbf{u}_p^{lag} = \mathbf{u}_p^n$$

On remarque   galement que la forme carr  e des mailles est conserv  e, voir Figure 45. Toute la mati  re   tant d  plac  e    la m  me vitesse par le vortex, il est normal que l'on observe une simple rotation de l'ensemble du maillage.

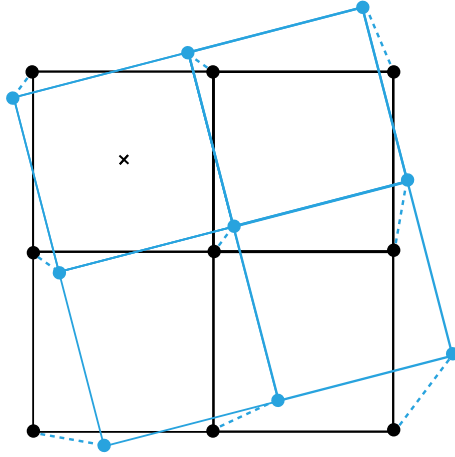


Figure 45: Schéma de la déformation lagrangienne dans le cas décalé pour un vortex idéal.

Phase de projection

Comme en 2.2.3, on détermine les flux aux faces de la maille nodale, et on applique la conservation de la quantité de mouvement (8), en prenant des valeurs pour les u_p^α correspondant à chaque ordre.

À l'ordre 1, on a les vitesses suivantes :

$$\mathbf{u}_p^O = \frac{\alpha_0 \Delta x}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \mathbf{u}_p^S = \frac{\alpha_0 \Delta x}{2} \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \quad \mathbf{u}_p^E = \frac{\alpha_0 \Delta x}{2} \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \mathbf{u}_p^N = \frac{\alpha_0 \Delta x}{2} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Ce qui donne donc, pour le nœud p :

$$\boxed{\mathbf{u}_{p,1}^{n+1} = \frac{1}{2} \alpha_0 \Delta x \begin{pmatrix} 1 + 2\alpha_0 \Delta t \\ 1 - 2\alpha_0 \Delta t \end{pmatrix}}$$

Puis, pour l'ordre 2 :

$$\mathbf{u}_p^O = \alpha_0 \Delta x \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}, \quad \mathbf{u}_p^S = \alpha_0 \Delta x \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}, \quad \mathbf{u}_p^E = \alpha_0 \Delta x \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}, \quad \mathbf{u}_p^N = \alpha_0 \Delta x \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}$$

On obtient donc à l'ordre 2 :

$$\boxed{\mathbf{u}_{p,2}^{n+1} = \frac{1}{2} \alpha_0 \Delta x \begin{pmatrix} 1 + \alpha_0 \Delta t \\ 1 - \alpha_0 \Delta t \end{pmatrix}}$$

Calcul des rotationnels discrets

On a dans ce cas-là $\frac{1}{2}(\mathbf{u}_p^n + \mathbf{u}_{p+1}^n) \cdot \mathbf{T}_{p,p+1} = \frac{1}{2} \alpha_0 \Delta x$, ce qui donne :

$$\text{rot}(\mathbf{u})_{c,D}^n = 2 \alpha_0 = \omega_0$$

Ce qui n'est pas étonnant compte tenu du choix d'un champ de vorticit e uniforme et  egal  a ω_0 .

Puis,  a t^{n+1} , on a quel que soit l'ordre :

$$\frac{1}{2}(\mathbf{u}_p^{n+1} + \mathbf{u}_{p+1}^{n+1}) \cdot \mathbf{T}_{p,p+1} = \frac{1}{2}\alpha_0\Delta x$$

et donc,
$$\boxed{\frac{\text{rot}(\mathbf{u})_{c,D}^{n+1}}{\text{rot}(\mathbf{u})_{c,D}^n} = 1}$$

Les sch emas d ecal es ne diffusent donc pas les vortex id eaux non plus. Et comme c' etait le cas pour les vortex ponctuels, on observe une amplification de la norme des vitesses nodales.

A.3.3 Etude pour les sch emas centr es

D etermination de \mathbf{u} sur le maillage

Les rayons  etant les m emes qu'en 2.3, on a, conform ement aux notations pr ec edentes :

$$u_1 = \frac{1}{\sqrt{2}} \alpha_0 \Delta x, \quad u_2 = \sqrt{\frac{5}{2}} \alpha_0 \Delta x, \quad u_3 = \frac{3\sqrt{2}}{2} \alpha_0 \Delta x,$$

Phase lagrangienne

On acc ede aux vitesses nodales par l'interm ediare de (9), et on obtient la d eformation lagrangienne suivante (voir Figure 46). On remarque qu'il n'y a aucune diff erence avec le cas d ecal e, l'ensemble du maillage est en rotation solide.

Ensuite, il reste  a calculer les vitesses centr ees  a t^{lag} avec (10). Les calculs m enent au r esultat suivant pour la vitesse au centre de la maille c en bas  a droite :

$$\boxed{\mathbf{u}_c^{lag} = \mathbf{u}_c^n = \frac{1}{2}\alpha_0\Delta x \begin{pmatrix} 1 \\ 1 \end{pmatrix}}$$

Phase de projection

Enfin, pour la projection, on proc ede exactement comme en 2.3. Apr es calcul des flux de masse aux faces de la maille c , et application de la conservation de quantit e de mouvement, il vient :

$$\boxed{\mathbf{u}_c^{n+1} = \frac{1}{2}\alpha_0\Delta x \begin{pmatrix} 1 + \alpha_0\Delta t \\ 1 - \alpha_0\Delta t \end{pmatrix}}$$

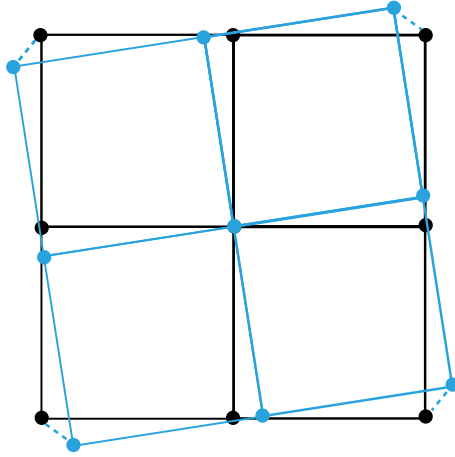


Figure 46: Schéma de la déformation lagrangienne dans le cas centré pour un vortex idéal.

Calcul des rotationnels discrets

Comme cette fois aussi $\frac{1}{2}(\mathbf{u}_c^n + \mathbf{u}_{c+1}^n) \cdot \mathbf{T}_{c,c+1} = \frac{1}{2}\alpha_0\Delta x$,

$$\text{rot}(\mathbf{u})_{p,C}^n = 2\alpha_0 = \omega_0$$

De plus, à t^{n+1} , on obtient :

$$\frac{1}{2}(\mathbf{u}_c^{n+1} + \mathbf{u}_{c+1}^{n+1}) \cdot \mathbf{T}_{c,c+1} = \frac{1}{2}\alpha_0\Delta x$$

d'où
$$\boxed{\frac{\text{rot}(\mathbf{u})_{p,C}^{n+1}}{\text{rot}(\mathbf{u})_{p,C}^n} = 1}$$

Contrairement au cas des vortex ponctuels, les schémas centrés ne vont pas diffuser la vorticit  induite par des vortex id aux. En effet, pour ce type de vortex, les sch mas d cal s (avec projection d'ordre 2) et centr s donnent exactement les m mes r sultats : m mes d formations lagrangiennes et m mes vitesses   $r = \Delta x/\sqrt{2}$ du centre du vortex,   t^{n+1} . Il est donc logique qu'on observe une conservation de la vorticit  pour les deux. On note  galement le m me  tirement du vortex et la m me amplification de la norme des vitesses.

A.3.4 Etude pour les sch mas de type BBC

D termination de \mathbf{u} sur le maillage

Le maillage  tant le m me qu'en 2.4, et le centre du vortex ayant le m me position, on garde les valeurs des rayons, avec cette fois les vitesses

$$u_1 = \frac{1}{2}\alpha_0\Delta x, \quad u_2 = \frac{1}{2}\alpha_0\Delta x, \quad u_3 = \frac{3}{2}\alpha_0\Delta x, \quad u_4 = \frac{3}{2}\alpha_0\Delta x$$

Phase lagrangienne

La phase lagrangienne est rigoureusement identique au cas du vortex ponctuel (voir 2.4). On retrouve à t^{lag} le même champ de vitesse qu'à t^n .

Phase de projection

Conformément aux notations adoptées en 2.4, on a cette fois après projection par splitting des vitesses aux faces de la maille en bas à droite, à l'ordre 1 en Δt :

$$ux_{i-1/2,j}^{n+1} = uy_{i,j+1/2}^{n+1} = \frac{3}{2}\alpha_0\Delta x$$

Comme pour le vortex ponctuel, on voit que la symétrie radiale est conservée à l'ordre 1.

Calcul des rotationnels discrets

À l'instant t^n , on a $\mathbf{u}_{c,c+1}^n \cdot \mathbf{T}_{c,c+1} = \frac{1}{2}\alpha_0\Delta x$,

$$\text{rot}(\mathbf{u})_{p,BBC}^n = 2\alpha_0 = \omega_0$$

Pour tous les types de schémas, dans le cas d'une rotation solide, on a donc bien un rotationnel discret égal à ω_0 à t^n .

De plus, compte tenu des valeurs des $\mathbf{u}_{c,c+1}^{n+1}$, on obtient à l'ordre 1 :

$$\frac{\text{rot}(\mathbf{u})_{p,BBC}^{n+1}}{\text{rot}(\mathbf{u})_{p,BBC}^n} = 1$$

Le schéma BBC ne diffuse pas la vorticit  des vortex ponctuels (  l'ordre 1). En effet, si on pousse le calcul des $\mathbf{u}_{c,c+1}^{n+1}$ jusqu'  l'ordre 2 en Δt , on voit appara tre un terme qui brise la sym trie du syst me (qui d'ailleurs existe aussi pour le cas du vortex ponctuel) d    la m thode de projection par splitting directionnel. Il en r sulte l'apparition d'un terme diffusif  gal   $\frac{1}{2}(\alpha_0\Delta t)^2$ dans le rapport des rotationnels. Ce qui n'est pas le cas pour les sch mas centr s et d cal s, qui conservent exactement la vorticit  des vortex id aux.

Conclusion

Effectuons pour terminer un bilan des résultats obtenus. On rassemble dans un tableau les différentes valeurs du rapport $\frac{\text{rot}(\mathbf{u})^{n+1}}{\text{rot}(\mathbf{u})^n}$ pour chacun des cas traités.

	Vortex ponctuel	Vortex idéal
Schéma décalé	1	1
Schéma centré	$1 - \frac{4}{3} \frac{\Delta t}{\Delta x} c_0 - \frac{14}{25} \alpha_0 \Delta t$	1
Schéma BBC	$1 - \frac{48}{25} \alpha_0 \Delta t$	1

Il en ressort que seul le schéma décalé ne diffuse ni la vorticit  des vortex ponctuels, ni celle des vortex id aux.

Une premi re explication peut venir du fait que le maillage primal n'est conforme dans $H(\text{rot})$ que pour ces sch mas-l . En effet, pour les autres sch mas, bien que le choix du point de calcul du rotationnel (i.e le choix du centre du vortex) ait  t  fait en prenant ce fait en consid ration, il se trouve que le champ de vitesse  volue sur le maillage primal, qui lui n'est pas conforme dans $H(\text{rot})$. Ce qui peut  tre source de diffusion num rique.

Une deuxi me explication est li e aux stencils de chaque type de sch mas. Dans le cas d cal , pour permettre le calcul des vitesses   t^{n+1} n cessaires   la d termination de $\text{rot}(\mathbf{u})$, il faut utiliser les valeurs des vitesses nodales de la maille centrale et des 4 mailles directement voisines (sch ma   5 points), ce qui fait au total 12 vitesses. Alors que pour les sch mas BBC et centr s, il faut utiliser les vitesses aux noeuds de la maille duale centrale et des 8 mailles duales voisines (sch ma   9 points), d'o  au total 16 vitesses. On fait les calculs avec des vitesses plus  loign es du centre du vortex, ce qui explique, dans le cas ponctuel, comme u_θ est une fonction d croissante de r , une diffusion plus importante pour ces sch mas-l .

Enfin, on observe qu'on a un crit re portant sur la CFL   partir duquel le sch ma centr  devient moins diffusif que le sch ma BBC. Pour commencer, le sch ma BBC est stable si :

$$\alpha_0 \Delta t = \frac{\Delta t}{\Delta x} \frac{\sigma_0}{2\pi \Delta x} \leq \frac{25}{48}$$

De plus, si BBC diffuse plus la vorticit  que les sch mas centr s alors cela signifie que

$$\frac{4}{3} \frac{\Delta t}{\Delta x} c_0 + \frac{14}{25} \alpha_0 \Delta t \leq \frac{48}{25} \alpha_0 \Delta t,$$

ce qui donne finalement :

$$\boxed{\text{CFL} = \frac{\Delta t}{\Delta x} c_0 \leq \frac{17}{32} = 0,53125}$$

Ce qui le cas puisque $\text{CFL}_{BBC} = 0,34$ classiquement.

References

- [1] Bruno Blais, Jean-Philippe Braeunig, Daniel Chauveheid, Jean-Michel Ghidaglia, and Raphaël Loubère. Dealing with more than two materials in the fvcf-enip method. *European Journal of Mechanics-B/Fluids*, 42:1–9, 2013.
- [2] Gilles Carré, Steéphane Del Pino, Bruno Després, and Emmanuel Labourasse. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228(14):5160–5183, 2009.
- [3] Clémentine Courtès and Mélanie Inglard. Etude de schémas numériques pour l’hydrodynamique lagrangienne : application aux schémas décalés et centrés, Rapport de stage 2012.
- [4] Quentin Debray. Etude de schémas numériques pour l’hydrodynamique compressible : un schéma lagrange projection conservatif en énergie totale, Rapport de stage 2013.
- [5] Steven Diot. *La méthode MOOD Multi-dimensional Optimal Order Detection: la première approche a posteriori aux méthodes volumes finis d’ordre très élevé*. PhD thesis, Toulouse 3, 2012.
- [6] Pete GW Stewart. *Matrix Algorithms: Volume 1, Basic Decompositions*, volume 1. Cambridge University Press, 1998.
- [7] Jean-François Haas and Bradford Sturtevant. Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *Journal of Fluid Mechanics*, 181:41–76, 1987.
- [8] Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.
- [9] Pierre-Henri Maire. Contribution to the numerical modeling of inertial confinement fusion. Habilitation à diriger des recherches 2011.
- [10] Laurent Saas. Schémas numériques euler bbc, 2014.
- [11] John VonNeumann and Robert D Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3):232–237, 1950.
- [12] Mark L Wilkins. Calculation of elastic-plastic flow. Technical report, DTIC Document, 1963.
- [13] David L Youngs. Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics*, 24:273–285, 1982.