

LAGRANGIAN METHOD ENHANCED WITH EDGE SWAPPING FOR THE FREE FALL AND CONTACT PROBLEM*

ÉTIENNE BERNARD¹, STÉPHANE DEL PINO², ERWAN DERIAZ³, BRUNO DESPRÉS⁴,
KATERINA JURKOVA⁵ AND FRÉDÉRIC LAGOUTIÈRE⁶

Abstract. We discuss various algorithms that one can use in conjunction with Lagrangian methods for the treatment of important deformation of the mesh. We develop a new contact algorithm. The free fall of a droplet of fluid in a light gas serves to illustrate the potentialities of these techniques.

Résumé. Nous étudions un problème modèle de chute libre d'une goutte de fluide dans de l'air à l'aide de méthodes numériques lagrangiennes. Nous développons et étudions un nouvel algorithme de contact qui permet (numériquement) à la goutte de toucher le sol.

INTRODUCTION

We refer to [1, 8] for a general introduction to Lagrangian methods for compressible fluid flow problems [9]. See also [11, 13]. The 2D Lagrangian equations that we consider are

$$\begin{cases} \rho D_t \tau = \nabla \cdot \mathbf{u}, & \tau = \frac{1}{\rho}, \\ \rho D_t \mathbf{u} = \nabla p + \rho \mathbf{g}, & p = (\gamma - 1) \rho \left(e - \frac{1}{2} |\mathbf{u}|^2 \right), \\ \rho D_t e = \nabla \cdot (p \mathbf{u}) + \rho (\mathbf{g}, \mathbf{u}). \end{cases} \quad (1)$$

The unknowns are the density $\rho > 0$, the velocity $\mathbf{u} \in \mathbb{R}^2$ and the total energy e . The pressure is p . The gravity is $\mathbf{g} \in \mathbb{R}^2$. The displacement of the frame is defined by

$$D_t \mathbf{x} = \mathbf{u}. \quad (2)$$

In this work we consider the problem of computing the free fall of a droplet of fluid inside a gas with a Lagrangian scheme based on the discretization of (1-2). Our interest in this problem stems from the fact that Lagrangian methods are very powerful algorithms for multimaterial fluid flow problems until some important

* *DEFORMA project, CEMRACS 2007*

¹ Université Paris-Diderot (Paris 7) and Laboratoire Jacques-Louis Lions (CNRS UMR 7598), 175, rue du Chevaleret, 75013 Paris, France (bernard@ann.jussieu.fr)

² CEA/DIF, 91680 Bruyères-le-Châtel, France (stephane.delpino@cea.fr)

³ Institute of Mathematics, Polish Academy of Sciences. ul. Sniadeckich 8, 00-956 Warszawa, Poland (E.Deriaz@impan.gov.pl)

⁴ CEA/DIF, 91680 Bruyères-le-Châtel, France (bruno.despres@cea.fr)

⁵ Technical University of Liberec (katerina.jurkova@tul.cz)

⁶ Université Paris-Diderot (Paris 7) and Laboratoire Jacques-Louis Lions (CNRS UMR 7598), 175, rue du Chevaleret, 75013 Paris, France (lagoutie@math.jussieu.fr)

mesh deformation occur. The coupling of Lagrangian methods with remeshing techniques is absolutely needed to cope with extreme mesh deformations. This issue is discussed in the literature since the 80's: see [12] for an overview of challenges and methods. In the following we use remeshing algorithms and couple them to the Lagrangian scheme. The most challenging part is the treatment of the contact between the droplet and the bottom wall.

The free fall problem intends to be representative of some of the numerical difficulties one encounters when using Lagrangian methods. For example let us consider the result of a Lagrangian computation for the free fall problem. On the left of Figure 1 is the initial condition. On the right is the result of the computation. Clearly the extreme deformation of the mesh makes questionable the accuracy of the numerical computation. Moreover such an approach is by construction unable to treat the contact.

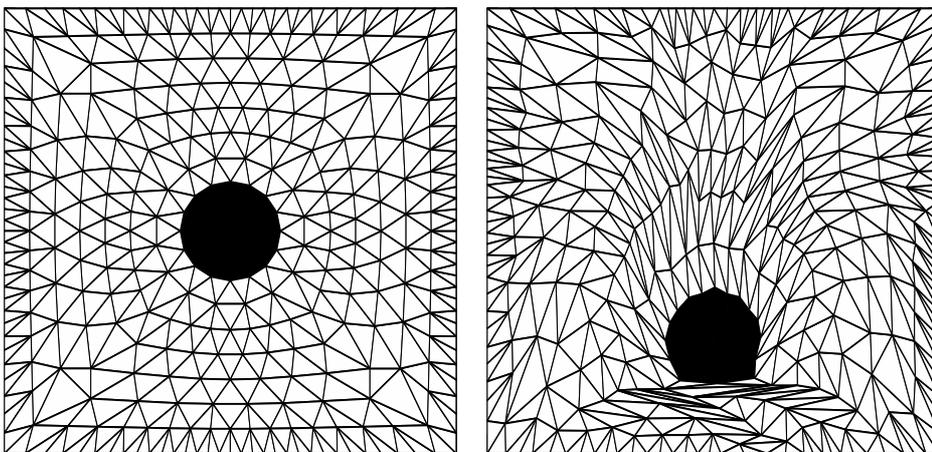


FIGURE 1. The droplet of fluid is in black. The mesh represents air with lower density. The gravity is directed toward the bottom. It is clear on the right that the mesh suffers from extreme deformation. It makes this Lagrangian computation useless after finite time.

We shall show that the situation is not desperate. Our idea is twofold. First, to use the newly developed cell centered Lagrangian scheme [4, 5] (which has been used to compute Figure 1; actually any other Lagrangian cell centered scheme, such as in [10], could be used instead) and to combine this scheme with 2D algorithms which are known to be efficient for the treatment of large mesh deformations. These algorithms consist in a swap procedure based on the empty ball criterion (or Delaunay criterion). Second (this is the originality of the present work), to design a procedure for the contact.

In what follows we give the main ideas behind each of the specific algorithms used in this study.

1. THE LAGRANGIAN SCHEME IN DIMENSION TWO

1.1. The basic 2D well-balanced Lagrangian scheme

The numerical used for the Lagrangian step of the method is based on some recent works [4, 5]. The schemes are Lagrangian which means that the mesh is moving at the fluid pace. The mass of the cells is constant. The scheme is cell-centered and the fluxes are vertices-based. Moreover the scheme is well-balanced in the sense that the gravity is incorporated in the design of the fluxes, see the seminal work [7] about well-balanced schemes.

Let us consider the mesh at time step $k\Delta t$. Each cell j is surrounded by nodes with indices r . We construct some normal \mathbf{n}_{jr}^k at the corner r of the cell j by

$$l_{jr}^k \mathbf{n}_{jr}^k = \frac{1}{2} \begin{pmatrix} y_{r+1}^k - y_{r-1}^k \\ -x_{r+1}^k + x_{r-1}^k \end{pmatrix}, \quad |\mathbf{n}_{jr}^k| = 1, \quad l_{jr}^k > 0,$$

with the property $\sum_r l_{jr}^k \mathbf{n}_{jr}^k = 0, \quad \forall j$. The basic algorithm is composed of (3-6). The equations for the nodal

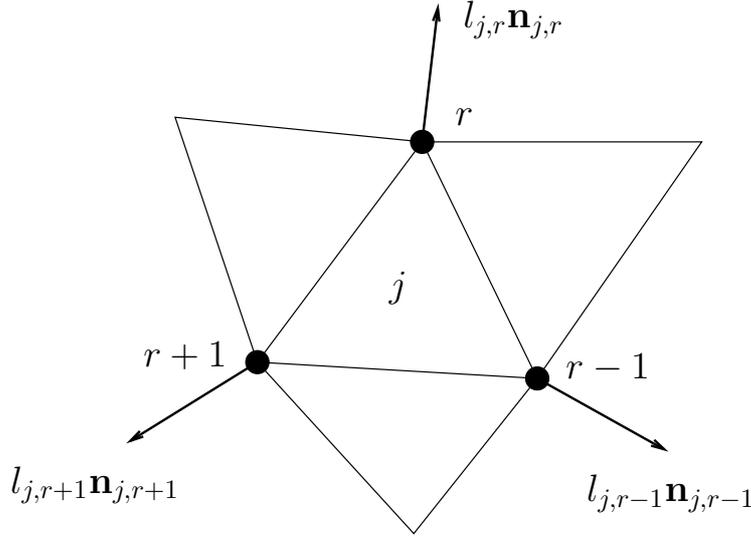


FIGURE 2. Notations: the cell j has three vertices $r-1$, r and $r+1$. The normal at corner r is $\mathbf{n}_{j,r}$.

fluxes are

$$\begin{cases} p_{jr}^* + \rho_j^k c_j^k (\mathbf{n}_{jr}^k, \mathbf{u}_r^*) = p_j^k + \rho_j^k c_j^k (\mathbf{n}_{jr}^k, \mathbf{u}_j^k) + \rho_j^k (\mathbf{g}, \mathbf{x}_r^k - \mathbf{x}_j^k), & j = 1, \dots, J, \\ \sum_j l_{jr}^k \mathbf{n}_{jr}^k p_{jr}^* = 0. \end{cases} \quad (3)$$

The equations for the displacement of the mesh are

$$\mathbf{x}_r^{k+1} = \mathbf{x}_r^k + \Delta t \mathbf{u}_r^*. \quad (4)$$

Then we compute s_j^{k+1} the new area of the cell. The density is

$$\rho_j^{k+1} = \frac{s_j^k}{s_j^{k+1}} \rho_j^k, \iff M_j = s_j^k \rho_j^k = s_j^{k+1} \rho_j^{k+1}. \quad (5)$$

The Newton's law and the total energy equation are

$$\begin{cases} M_j \mathbf{u}_j^{k+1} = M_j \mathbf{u}_j^k - \Delta t \sum_r l_{jr}^k \mathbf{n}_{jr}^k p_r^* + \Delta t M_j \mathbf{g}, \\ M_j e_j^{k+1} = M_j e_j^k - \Delta t \sum_r l_{jr}^k (\mathbf{n}_{jr}^k, \mathbf{u}_r^*) p_r^* + \Delta t M_j \left(\mathbf{g}, \frac{\mathbf{u}_j^{k+1} + \mathbf{u}_j^k}{2} \right) \end{cases} \quad (6)$$

By construction this scheme is conservative in total mass, and is conservative in total impulse and total energy in the case $\mathbf{g} = 0$. The scheme is stable under a standard CFL condition.

1.2. Swapping procedure

During the history of the mesh which is moving at a velocity \mathbf{u}_r^* close to the fluid velocity \mathbf{u}_j^k , tangling occur each time the stretching is too large. We describe a very simple method which aims at enhancing the quality of the mesh, modifying it into a Delaunay mesh. We refer to [6] and therein for a general introduction to mesh generation and mesh modifications.

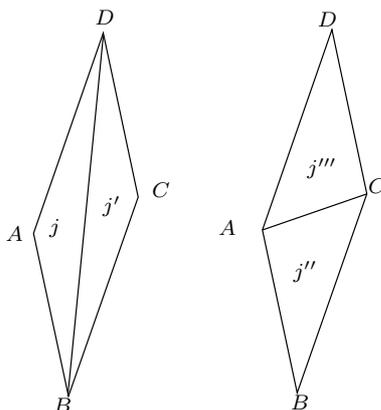


FIGURE 3. The triangle j with vertices (A, B, D) and the triangle j' with vertices (B, C, D) are flipped into the triangle j'' with vertices (A, B, C) and the triangle j''' with vertices (A, C, D) .

The idea of the method is to look at each pair of triangles *that contains the same fluid*. If the pair does not satisfy the empty ball criterion we flip it (see [6, 12]) according to the following example.

During this operation we remap the density of mass, the impulse and the total energy with a simple first order procedure. That is $s_j + s_{j'} = s_{j''} + s_{j'''}$ and

$$\rho_{j''} = \rho_{j'''} = \frac{s_j \rho_j + s_{j'} \rho_{j'}}{s_j + s_{j'}},$$

$$\rho_{j''} \mathbf{u}_{j''} = \rho_{j'''} \mathbf{u}_{j'''} = \frac{s_j \rho_j \mathbf{u}_j + s_{j'} \rho_{j'} \mathbf{u}_{j'}}{s_j + s_{j'}} \quad \text{and} \quad \rho_{j''} e_{j''} = \rho_{j'''} e_{j'''} = \frac{s_j \rho_j e_j + s_{j'} \rho_{j'} e_{j'}}{s_j + s_{j'}}.$$

By construction this method is conservative in total mass, total impulse and total energy.

Remark As specified, the flip of two adjacent cells is performed only if the two cells contain the same fluid: liquid or gas. Otherwise this would lead to introduce numerical mixing and lose the advantage of the Lagrangian scheme. During the computation of the fall of the droplet, when the droplet is near the ground, with only one layer of (stretched) air cells, this smoothing of the mesh is then excluded. This will motivate another special treatment in section 1.4: the flooding.

1.3. The initial mesh

In order to run the simulation of the free fall of a droplet, we customize an initial mesh (that is at $t = 0$) with triangles of relatively homogeneous size and shape. First we mesh the disk with the process indicated in Figure 4. This mesh has a rotational symmetry. We construct the mesh iteratively, starting from the center of the disk. For each layer, if the outer edge of the triangle is greater than $\sqrt{2}$ times the width of the layer, then we refine this layer with middle points M_i as indicated in Figure 4. In this way, all triangles have more or less the same shape and the same dimension. Then we deform the outer layer (the one filled with air) until it fits the square. Thus the triangles of the mesh do not elongate to much, and present almost equal areas.

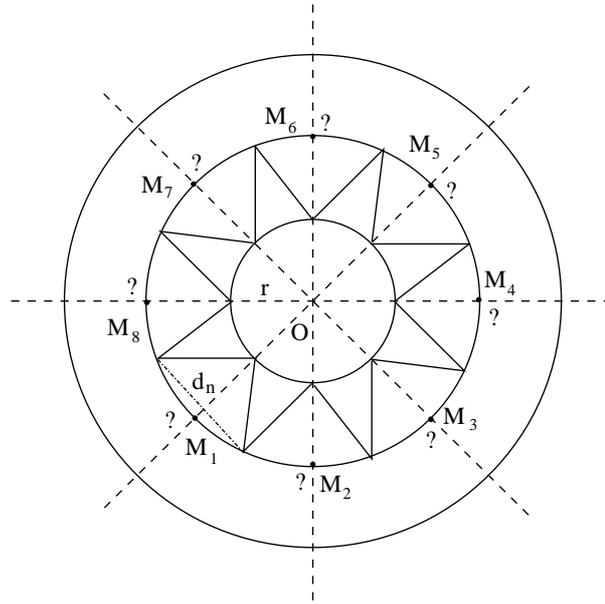


FIGURE 4. The equilibrated mesh is generated in a rotational symmetric fashion, the points M_i are added if and only if $d_n > \sqrt{2}r$.

1.4. Cell flooding

The change of topology of the mesh can't take place if we just apply the swapping algorithm. Hence, the domain allocated to air, which is a perforated surface is not able to turn itself into a surface isomorph to a disk. This makes the connection of the droplet with the boundary impossible.

In order to circumvent this drawback, we apply a non-physical operation on the mesh. This operation enforces the contact between the droplet and the boundaries under some particular stretching conditions on the mesh, so that the error we commit remain small.

This operation consists in *flooding* cells which are in contact with the droplet and which undergo a major stretch.

One can report to Figure 6 to check the notations. If the height h of a cell filled with air is inferior to a minimal height h_0 (i.e. the cell is stretched enough), and if the droplet is going toward the cell (which corresponds to $\mathbf{v}_1 \cdot \mathbf{n} > 0$ or $\mathbf{v}_2 \cdot \mathbf{n} > 0$), then we change this air cell into a fluid cell. That is the physical nature of the cell changes. This is a very important modification with the initial Lagrangian philosophy of the method.

In order to conserve the mass, the moment and the energy of each physical component (air and fluid), we take the fluid which floods the cell C from the adjacent fluid cell C_a and distribute the air which was inside the cell C to the other air cells. The corresponding equations are given by:

$$\rho(C_a) = \rho(C) = \frac{\text{mass}(C_a)}{\text{surface}(C_a) + \text{surface}(C)}$$

and

$$\mathbf{u}(C) = \mathbf{u}(C_a), \quad e(C) = e(C_a)$$

where \mathbf{u} denotes the moment and e denotes the energy per mass unit.

For all air cells C' :

$$\begin{aligned} \rho(C') &= \rho(C') + \alpha(C') * \text{mass}(C) \\ \mathbf{u}(C') &= \mathbf{u}(C') + \alpha(C') * \mathbf{u}(C) \end{aligned}$$

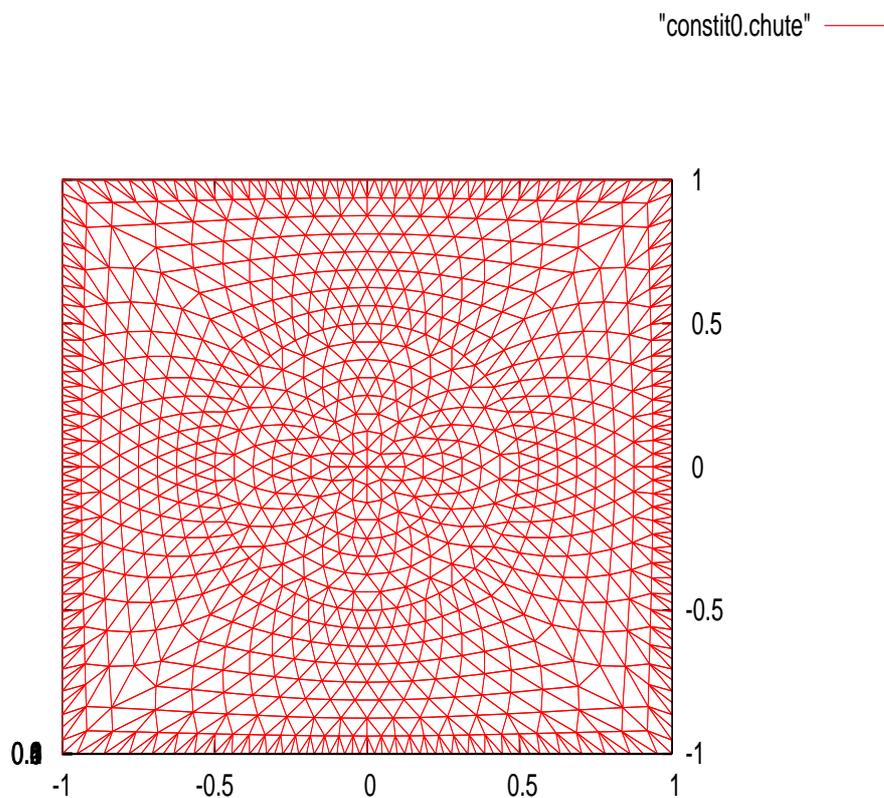


FIGURE 5. The resulting mesh issued from the process indicated in figure 4 is rather equilibrated.

$$e(C') = e(C') + \alpha(C') * e(C)$$

where $\alpha(C')$ is a weight coefficient which is proportional to the mass of the cell C' and inversely proportional to the distance between the cell C' and the flooded cell C . The weights $\alpha(C')$ satisfy:

$$\sum_{C' \text{ is an air cell}} \alpha(C') = 1$$

Without such an operation, the droplet should never touch the ground and the code should freeze.

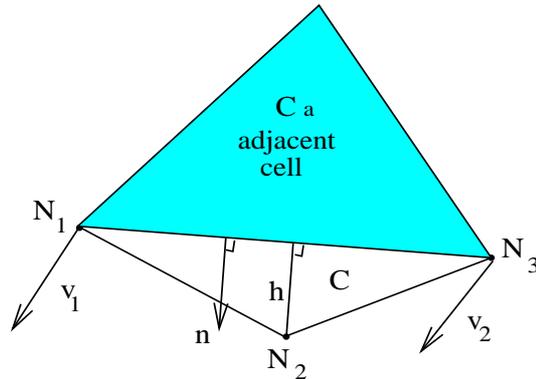


FIGURE 6. The flooding of the cell underneath occurs under particular stretching conditions.

1.5. Refinement – coarsening

1.5.1. *Small cells near the boundary*

During the simulation, it is possible that very small cells appear near the boundary of the domain. This is not in contradiction with the swapping algorithm for remeshing. This phenomenon has been observed in various problems with such algorithms. The problem is that the smallest the cell, the smallest the time step. In this case the CPU time may become quite important. This is why a specific remeshing algorithm has been developed.

We test if very small cells appear at the boundary of the domain. If so, we regroup them provided they belong to the same material (the fluid or the gas in our simulations).

1.5.2. *Grid change on the boundaries*

The algorithm proceeds as follows: we fix two thresholds d_{\min} and d_{\max} , the minimum and the maximum edge sizes on the boundaries.

If an edge is too small, the adjacent cell is removed, if it is possible, by merging it with a neighboring cell. As we do not want to change the number of points on each boundary, the removed point is moved to the middle of the largest edge of the same boundary. The merged cell takes the average values of the coalescent cells for the density, energy and pressure. The two new cells formed from the split cell take the same values as it was previously.

Similarly, if a boundary edge is too large, we split it. We take a point from the smallest edge of the same boundary, and put it in the middle of the large edge.

Doing so, we avoid small cells on the boundaries, so the Lagrangian code run with an increased ease.

2. NUMERICAL EXPERIMENTS

2.1. Test case

The aim of this study was to provide effective solutions to mesh problems appearing in a Lagrangian formulation of flow dynamics for compressible gas. The simulation of a droplet falling inside a glass (Figure 7) includes two main problems: the deformation of the mesh induced by the flow circulation - this problem gets an answer thanks to swapping algorithm - and the reconnection of domains - for which we tested the flooding operation.

The box has dimension 2×2 (in meters). We start with a velocity equal to 0. The radius of the droplet is 0.2. In this paper, we present four cases:

- the case with no swapping and no flooding operation (Figure 1),

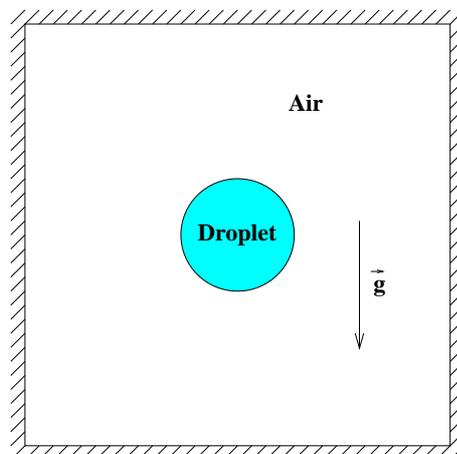


FIGURE 7. Our test case consists of a droplet falling in a closed box.

- the case with just swapping (Figure 8 and 9),
- the case where we used the swapping and the flooding operation (Figure 10),
- and the case with swapping, flooding and coarsening of the boundary points which makes the code more fluent (Figure 11).

Typical h_0 - minimal height for flooding - is chosen equal to 0.1 times the standard size of the initial edges. The minimal value of the edges on the boundaries is also chosen equal to h_0 .

2.2. Numerical results with Lagrangian method in two dimensions

In Figure 8 we plot the result computed with the Lagrangian scheme and the swapping algorithm (at each time step).

The initial data are the following. In the black circle (radius $R = 0.2$) there is at $t = 0$ a high density fluid ($\rho = 1$ and $\gamma = 4$) which may represent the fluid. In the white region there is a light fluid ($\rho_{air} = 0.01$ and $\gamma = 1.4$) which represents air. The initial pressure is, in the white region, the hydrostatic pressure $p = 1 - (y - 1)\rho g$ where g is the magnitude of the gravity force $\mathbf{g} = (0, -g)$. Inside the black region the pressure is uniform $p = 1 - (0 - 1)\rho_{air}g$, 0 is the elevation of the center. The boundary condition is a sliding wall $(\mathbf{u}, \mathbf{n}) = 0$ where \mathbf{n} is the exterior normal.

Clearly the results are much better than without the swapping. It must be stressed that the contact is by construction never possible if one uses only the swapping algorithm. This is visible in Figure 9, which is a different view of the last result in Figure 8.

However if one uses the the flooding algorithm, then the numerical contact of the droplet of fluid on the boundary is possible. This is represented in Figure 10.

A problem with the results of Figure 10 is that very small cells appear near the boundaries. So the time step goes to zero due to stability requirements. In Figure 11 we relax these constraints using a specific method to gather these small cells. In practice we observed the time step is much bigger, which makes the simulation possible in a reasonable time.

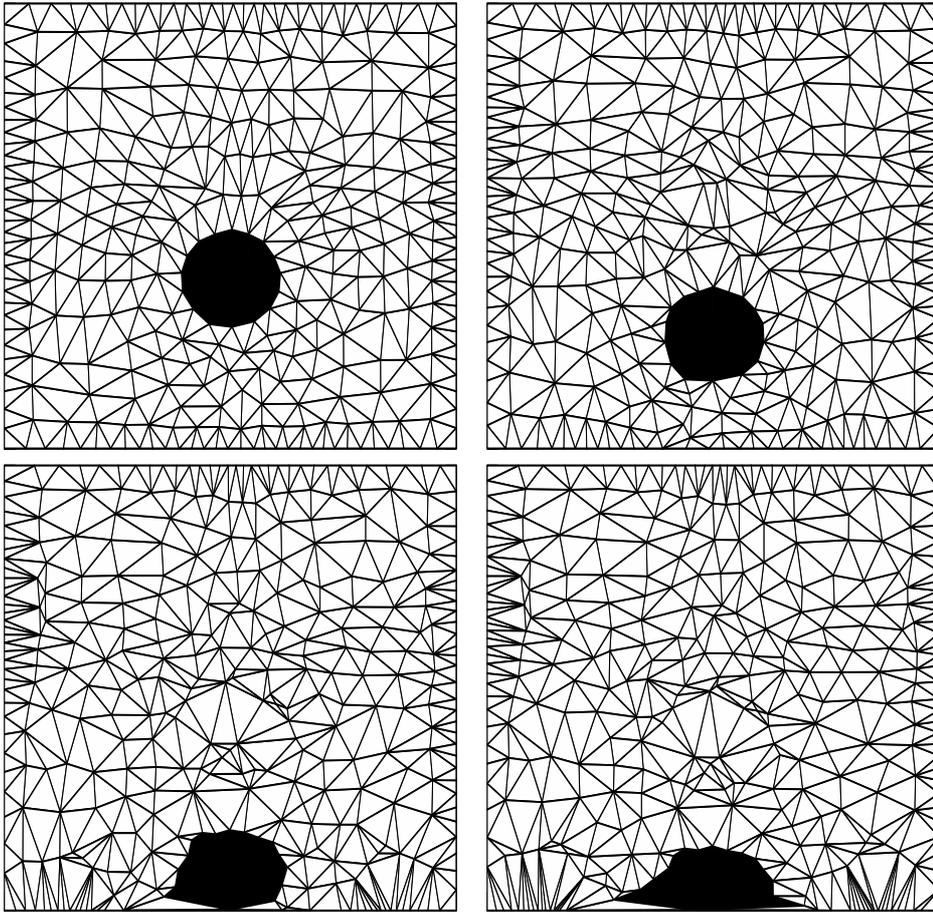


FIGURE 8. Lagrangian simulation of the free fall with just swapping. One can observe that the droplet is unable to make the contact with the boundary.

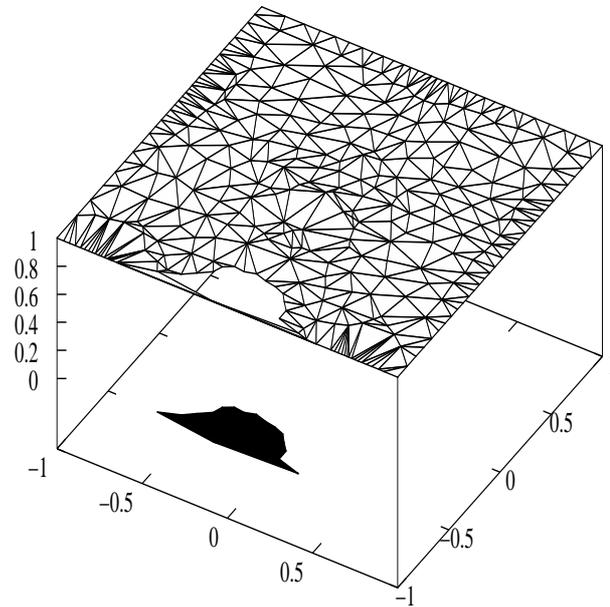


FIGURE 9. Representation in 3D: the numerical contact is never possible without the flooding algorithm

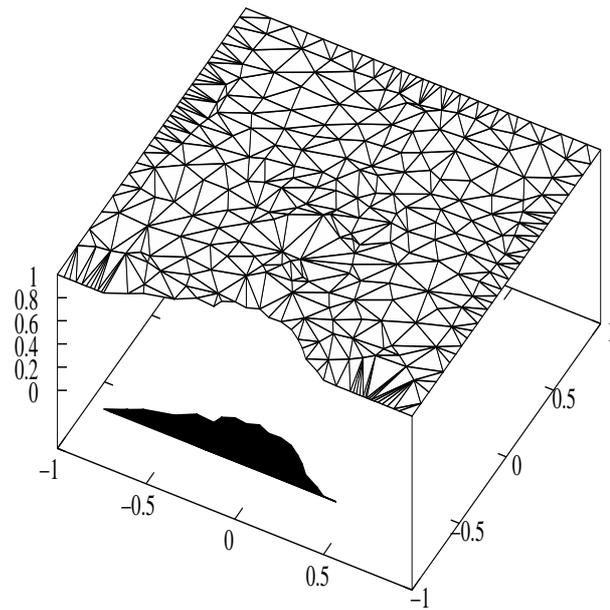


FIGURE 10. The numerical contact is feasible with the flooding method

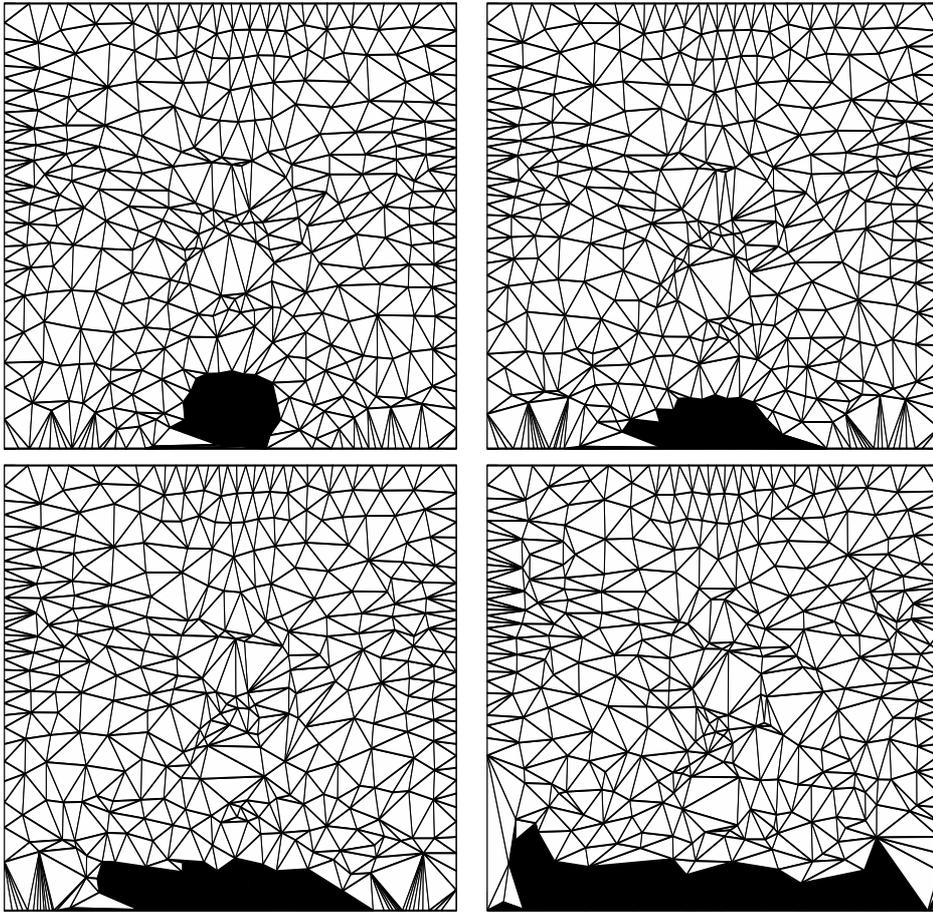


FIGURE 11. The free fall experiment in a Lagrangian scheme with swapping, flooding operation, and grid change on the boundaries.

2.3. Comparison with results obtained through an Eulerian method

Here we provide some numerical results obtained with an Eulerian algorithm. The advantage of Eulerian methods is that the mesh is fixed, so that there is no need to propose a special treatment for the topological changes of the flow. The drawback comes from the fact that the mesh is fixed: a consequence is that it does not follow the matter, thus, e.g., the mass fractions of the liquid and the gas are projected on the mesh (they take values different from 0 and 1), and unlike in the Lagrangian case, there is no clear frontier between the fluid and the air. This leads to the numerical artifact called “numerical diffusion”. In order to reduce this well-known drawback, we use an anti-dissipative algorithm developed in [3]. The principle of this algorithm is the following.

The fixed mesh is Cartesian (in the following test-cases, composed of squares) and the algorithm follows an Alternate Direction Splitting strategy. In each of the two resulting phases (horizontal and vertical), a Lagrange+remap algorithm is used: this corresponds to a Lagrange+remeshing method where the mesh would be fixed. The Lagrange part is exactly the one described above, and the core of the special anti-dissipative treatment is the remapping phase. It is based on a limited downwind flux for transport equations and is described and analyzed in [2], [3]. The Eulerian computation is done with a very fine mesh: 200×200 cells. We consider it as the reference solution.

We compare the Lagrangian and the Eulerian results in figure 12. We observe a very good qualitative agreement despite the fact that the algorithms are completely different. The position of the droplet is almost the same on both sides of the figure (we observe that the vertical difference is approximatively of one cell length). The swapping does not introduce any visible artifact before the contact on the ground. On the other hand, the last result, at time $t = 0.4$, shows small differences at the contact point. Figure 2.3 allows to compare the position of the center of mass of the droplet as a function of time with the Lagrangian scheme and with the Eulerian scheme (the third line represents the exact position of the center of mass of the droplet in vacuum) and we observe a good agreement. The Eulerian computation is very close to the reference curve: probably because the number of cells is large in this computation. The Lagrangian computation is very close, but the difference with the reference solution is more important. However the number of cells is small, and this is probably an important contribution to the error.

In this work we do not discuss the accuracy of both simulations after the contact time. We leave this issue for further research.

CONCLUSION

In this article, we were interested in adapting a new cell centered Lagrangian method for gas dynamics [5] to the problem of the free fall of a droplet inside a 2D glass (filled with air). In this experiment, major deformation problems arise. We evicted the numerical obstacles in three steps:

- the swapping algorithm flips adjacent triangles avoiding bad conditioned triangulations in the interior of the domain,
- the flooding algorithm allows us to change the topology of the mesh regarding the nature (fluid or gas) (air or fluid) of each cell,
- the remeshing of the boundaries which improves the mesh on the boundaries and allows the code to run faster under the CFL condition.

All the discussed algorithms are compatible, both from the implementation point of view and from the numerical point of view (the resulting method is stable). We think the methodology developed in this paper (together with the algorithms) could serve as a guideline for future research about cell-centered Lagrangian methods in the context of high deformation.

We have also presented some comparison with an Eulerian method. More comparisons are required to assess the accuracy of the algorithms discussed in this paper after the contact time. At this moment, this is an open problem.

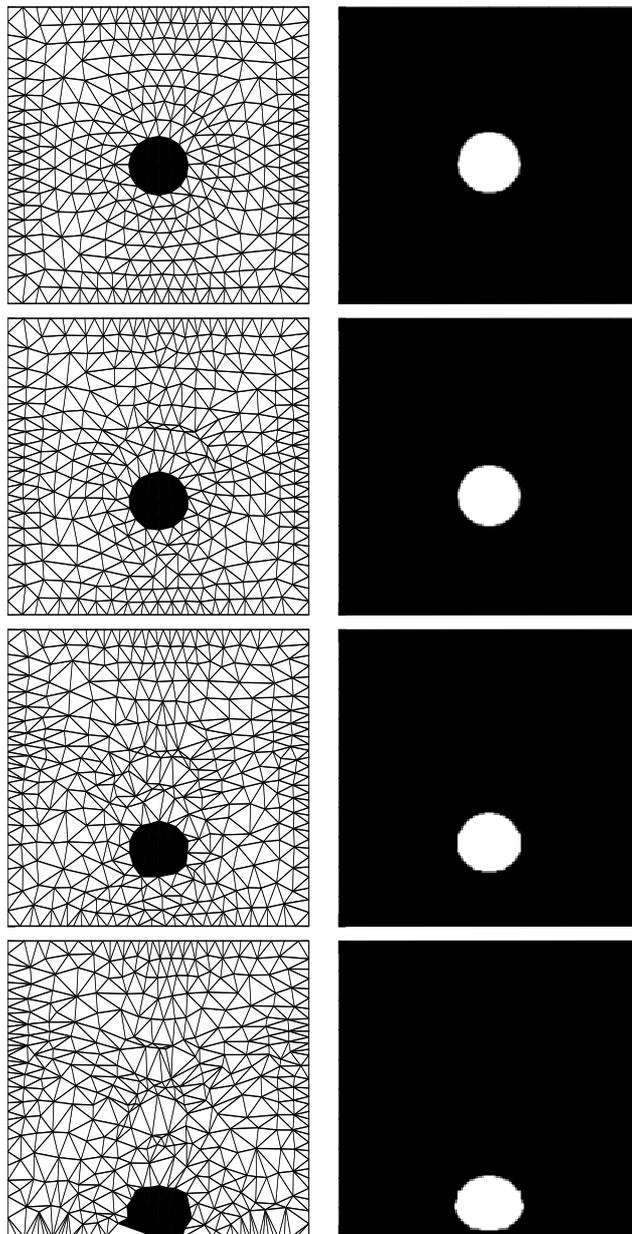


FIGURE 12. The free fall experiment. Left, Lagrangian scheme with swapping, flooding operation, and grid change on the boundaries. Right: Anti-dissipative Eulerian scheme. Time, for top to bottom: 0.1, 0.2, 0.3, 0.4

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their valuable remarks.

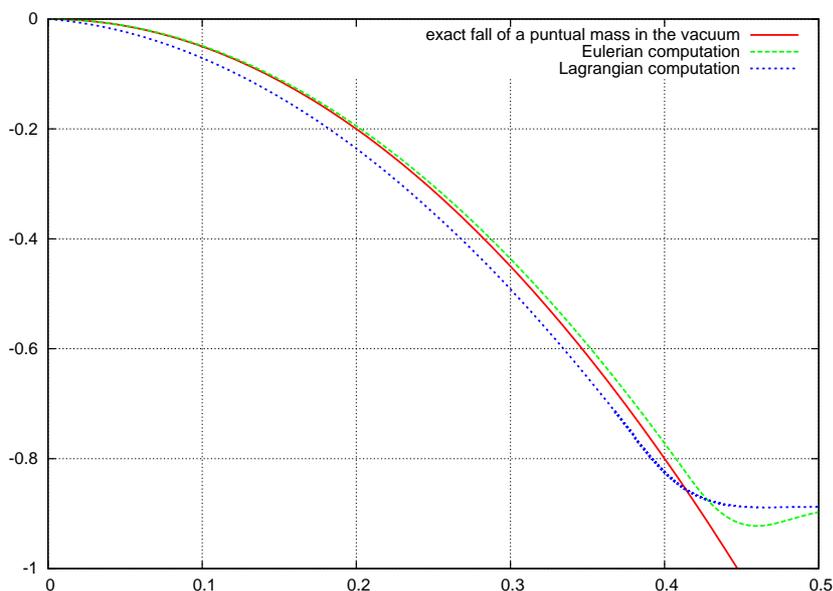


FIGURE 13. Vertical position of the center of mass of the droplet (in abscissa: time). We plot three curves: one is the reference solution $t \mapsto -\frac{g}{2}t^2$, the closest to the first one is computed in Eulerian configuration with a large number of cells, the third one is computed with the Lagrange code with swapping and flooding but with much less cells. All three curves are in close agreement

REFERENCES

- [1] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Computer Methods in Applied Mechanics and Engineering* 99 (1992), 235–394.
- [2] B. Després and F. Lagoutière, Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *J. Sci. Comput.* 16 (2001), no. 4, 479–524 (2002).
- [3] B. Després and F. Lagoutière, Numerical resolution of a two-component compressible fluid model with interfaces. *Prog. Comput. Fluid Dyn.* 7 (2007), no. 6, 295–310.
- [4] B. Després and C. Mazeran, Symmetrization of Lagrangian gas dynamics and multidimensional solvers, *CRAS Mécanique*, 331, 475-480, 2003.
- [5] B. Després and C. Mazeran, *Lagrangian Gas Dynamics In Dimension Two and Lagrangian systems*, Arch. Rat. Mech. Anal., 2005.
- [6] P. Frey and P. L. George, *Mesh generation. Application to Finite Elements*, Hermes Sciences, Paris-Oxford, 2001.
- [7] J. M. Greenberg and A. Y. LeRoux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, *Siam J. Numer. Anal.*, 33, 1, 1–16, 1996.
- [8] W. H. Hui, Unified coordinate system in computational fluid dynamics, *Comm. Comp. Phys.*, 2, 577–610, 2006.
- [9] R.J. LeVeque, *Numerical methods for conservation laws*. (ETHZ Zurich, Birkhauser, Basel 1992).
- [10] P.-H. Maire, R. Abgrall, J. Breil, J. Ovardia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM J. Sci. Comput.* 29 no 4 (2007), 1781–1824.
- [11] G. Scovazzi, E. Love and M. Shashkov, A multi/scale Q1P0 approach to Lagrangian shock hydrodynamics, Sandia report, Sand 2007-1423.
- [12] H.E. Trease, M. J. Fritts and W. P. Crowley editor, *Free Lagrange Methods*, Lecture Notes in Physics, 395, Springer Verlag, 1991.
- [13] R. Liska, M. Shashkov and B. Wendroff, Lagrangian composite schemes on triangular unstructured grids, Los Alamos National Lab. Report 02-7834, 2002.