Interface reconstruction for 2D compressible multi-materials flows

Supervisors: Laurent Dumas, Philippe Hoch, Thomas Leroy

Students: Théo Corot, Igor Chollet, Giulia Lissoni

August 22, 2018

Contents

Interface reconstruction: context

The context of the problem is :

- image segmentation
- multimaterial hydrodynamic simulations with ALE
- ...

The different families of methods in the literature are:

- Interface reconstruction method, such as **Young's method**:
  - conservation of partial volumes, robust, CPU cost
  - discontinuous interface
- Interface tracking (**Level set method**)
  - continuous interface, robust
  - no conservation of partial volumes
- Anti-diffusive methods (**VoFiRe method**)

Interface reconstruction

A new method in the family of IR was recently proposed :

DPIR method.

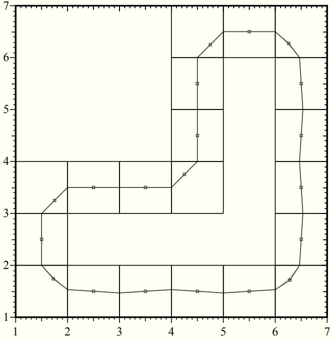
It combines all the advantages of the previous methods :

1. **continuity of the interface**
2. **preservation of volume fractions**
3. robustness on cartesian meshes
4. moderate computational cost

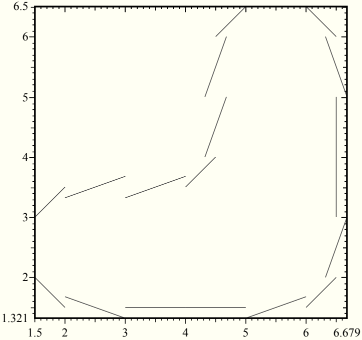
Example : the J test case

Reconstruction of a J with two materials on a cartesian mesh.

DPIR:



YOUNG'S:



DPIR method

"DPIR" method : minimization step

"DPIR" = Dynamic Programming Interface Reconstruction

[L. Dumas, J.M. Ghidaglia, P. Jaisson, R. Motte, A new volume-preserving and continuous interface reconstruction method for 2D multi-material flows, Int. J. Numer. Meth. Fluids (2017).]

It is made of two steps. The first one consists in:

- 1) the minimization of a cost functional based on volume fractions least square errors by using **dynamic programming**:

$$\min_{P_0,\dots,P_N} \left\{ \sum_{i=0}^{N-1} |\mathrm{vol}(P_i, P_{i+1}) - \mathrm{vol}_{\mathrm{target}}|^p + \lambda \|P_i - P_{i+1}\| \right\}$$

subject to $P_0 = P_N$, with $P_i = x_i I_i + (1 - x_i) E_i$, $x_i \in [0, 1]$.

Dynamic Programming : context

Let S be a state set ($\#S < \infty$ in our case) and G an application such that:

$$G : (x, y) \in S \times S \mapsto G(x, y) \in \bar{\mathbb{R}}.$$

The *horizon* $- N$ dynamic programming problem writes as :

$$\inf_{x_1,\dots,x_N}\{G(x_0, x_1) + G(x_1, x_2) + ... + G(x_{N-1}, x_N) + K_{x_N}\}$$

with $K_{x_N} \in \bar{\mathbb{R}}$.

Dynamic Programming : Resolution

The dynamic programming equation can be written as follows :

$$v_x^0 = K_x, \qquad \forall x \in S,$$
$$v_x^n = \inf_{y \in S}(G(x,y) + v_y^{n-1}), \qquad \forall 1 \leq n \leq N, \forall x \in S.$$

Using Bellman operator :

$$\mathbb{B} : \bar{\mathbb{R}}^S \to \bar{\mathbb{R}}^S, (\mathbb{B}(w))_x \mapsto \inf_{y \in S}(G(x,y) + w_y),$$

this equation becomes :

$$v^n = \mathbb{B}(v^{n-1}).$$

In term of flops : $O(N|S|)$ instead of $O(|S|^N)$ with a naive paths enumeration.

[J. F. Bonnans, S. Gaubert, Recherche Opérationnelle : aspects mathématiques et applications, 2017.]

"DPIR" method : correction step

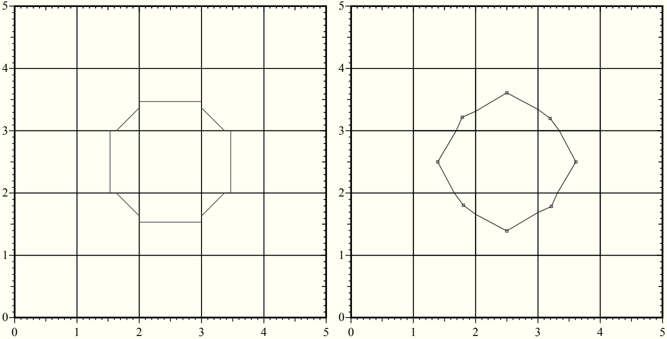
"DPIR" = Dynamic Programming Interface Reconstruction

[L. Dumas, J.M. Ghidaglia, P. Jaisson, R. Motte, A new volume-preserving and continuous interface reconstruction method for 2D multi-material flows, Int. J. Numer. Meth. Fluids (2017).]

The second step consists in:

- 2) a local **correction phase**: we add a control point on the perpendicular bisector of each segment $P_i P_{i+1}$ in order to obtain the desired volume fraction.

Comparison DPIR - Young's



On the left: the reconstruction of the cercle with Young's.
On the right: the reconstruction of the cercle with DPIR ($\lambda = 0.2$).

Goals of the project

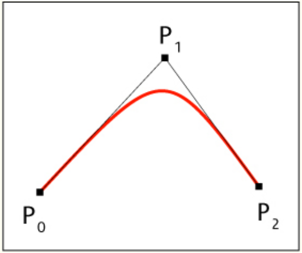
Goals of the project

1. Passing from a polygonal to a **curve reconstruction** for the interface
2. Improve the **robustness** of the method in the case of unstructured meshes
3. Extend the method to **three materials**

Quadratic rationnal Bézier Curve

It is a parametric curve, to which we associate:

- a control point P_1
- a weight $\omega \in [0, +\infty]$



We can compute its area in the following way:

$$\text{Area} = f(\omega) \cdot A(P_0, P_1, P_2)$$

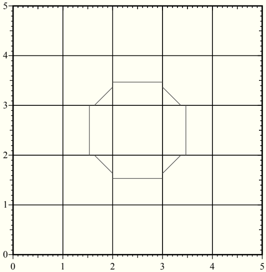
Quadratic rationnal Bézier Curve

$$\text{Area} = f(\omega) \cdot \text{A}(P_0, P_1, P_2)$$

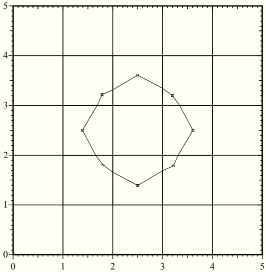
where:

$$f(\omega) = \begin{cases} 0, & \text{if} \quad \omega = 0 \\ \dfrac{2\omega}{1-\omega^2}\left(\dfrac{1}{1-\omega^2}\arctan\left(\sqrt{\dfrac{1-\omega}{1+\omega}}\right) - \dfrac{\omega}{2}\right), & \text{if} \quad \omega \in (0,1) \\ \dfrac{2}{3}, & \text{if} \quad \omega = 1 \\ \dfrac{\omega}{\omega^2-1}\left(\omega + \dfrac{1}{\sqrt{\omega^2-1}}\ln\left(\omega - \sqrt{\omega^2-1}\right)\right), & \text{if} \quad \omega > 1 \end{cases}$$

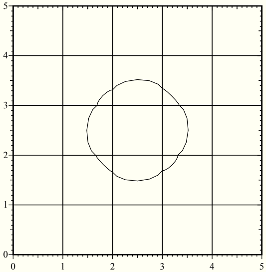
Correction comparison



Young's method



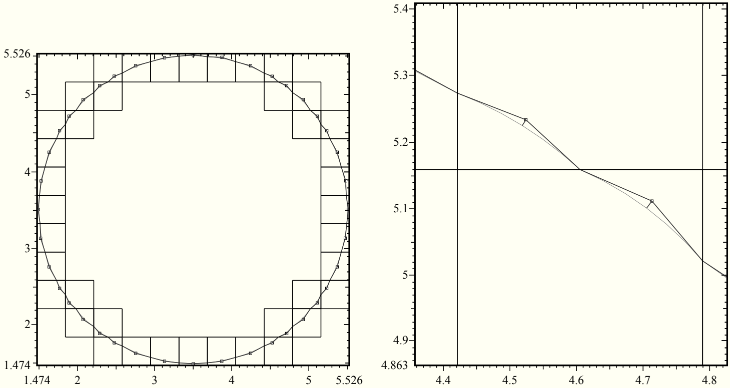
Polygonal (DPIR) $(\lambda = 0.2)$



Curve (DPIR) $(\lambda = 0.2)$

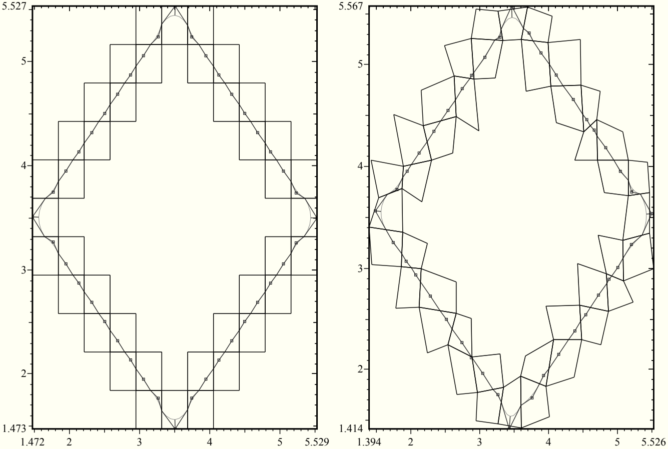
Structured meshes

Reconstruction of a circle



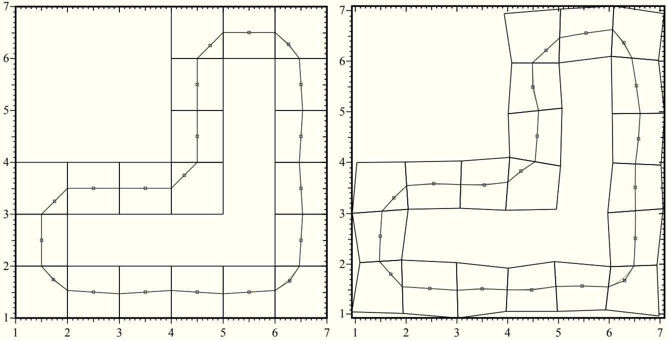
On the left: the reconstructed circle with Bezier curves and control points (computed with ω=1, $\lambda = 0.01$).
On the right: a zoom that shows the control point (in black) and the Bezier curve (in grey).

Reconstruction of a square



On the left: the square on a cartesian mesh.
On the right: the square on a perturbed cartesian mesh.
For both cases, we reconstructed with $\omega = 1$ and $\lambda = 0.01$.

Reconstruction of a J



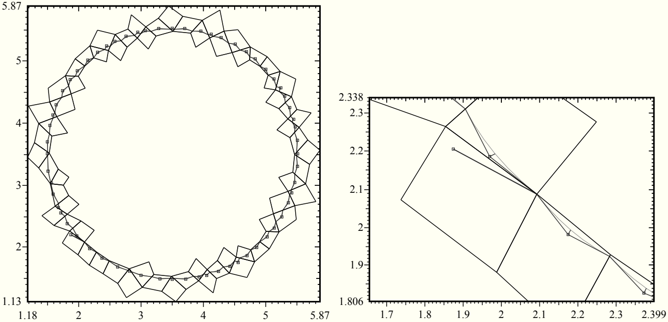
On the left: the J on a cartesian mesh.

On the right: the J on a perturbed cartesian mesh.

For both cases, we reconstructed with and $\omega = 1$ and $\lambda = 0.01$.

Unstructured meshes

Reconstruction of a circle : problem



On the left: the reconstructed circle.
On the right: a zoom on the "singular" points. We remark a problem when the curve passes really close to the nodes of the mesh.

Improvements of robustness

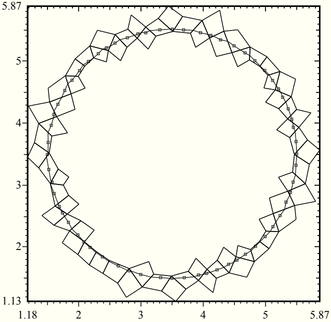
- **Discretization :** switch to Chebyshev nodes : for $k = 1, \dots n$, $x_k = \cos\left(\frac{2k-1}{2n}\pi\right)$
- **Search direction of the control points :** towards the center of the cell
- **Correction of the functional :** We added a further penalization $\widetilde{\lambda}$ to the cost functional:

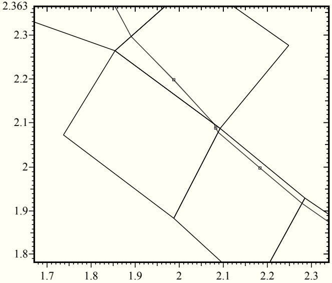
$$\widetilde{\lambda} = \frac{|\text{vol}_{\text{target}} - \text{vol}(P_i, P_{i+1})|}{\text{vol}(P_i, P_{i+1})},$$

that now becomes:

$$\min_{P_0,\dots,P_N} \left\{ \sum_{i=0}^{N-1} |\text{vol}(P_i, P_{i+1}) - \text{vol}_{\text{target}}|^p + \lambda \|P_i - P_{i+1}\| + \widetilde{\lambda} \right\}$$

Reconstruction of a circle : correction





On the left: the reconstructed circle.
On the right: a zoom on the corrected "singular" points.

Extension to three materials

DPIR: extension to three materials

We apply a new strategy, that is made of three steps:

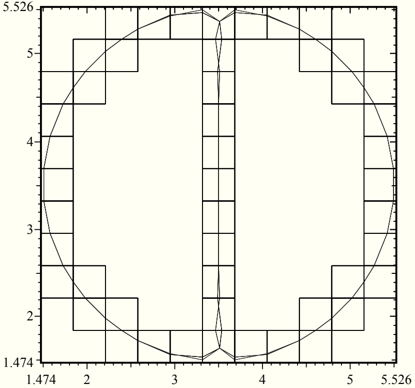
1) **Dynamic Programming:** We reconstruct separately the interfaces of the three materials by minimizing the same functional of the two material's case:

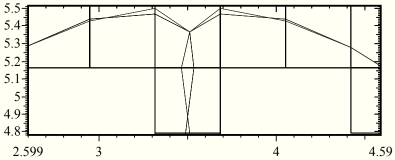
$$\min_{P_0^k,\dots,P_N^k} \left\{ \sum_{i=0}^{N-1} |\mathrm{vol}(P_i^k, P_{i+1}^k) - (\mathrm{vol}_{\mathrm{target}})^k|^p + \lambda \| P_i^k - P_{i+1}^k \| \right\}$$

with $k = 1, 2, 3$ is the index of the material.

At this stage, we have three sets of different "interface points", one per material.

Result after Dynamic Programming





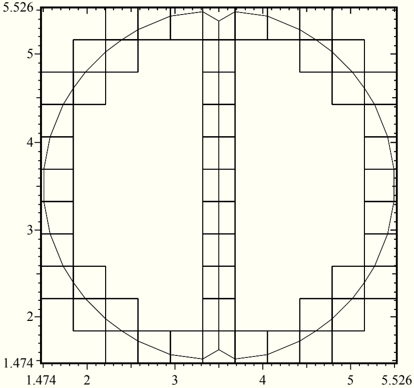
On the left: the cercle after the Dynamic programming.
On the right: a zoom on the three interfaces.

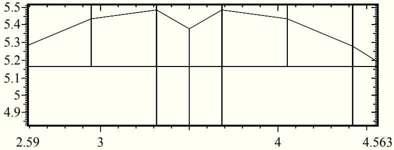
DPIR: extension to three materials

We apply a new strategy, that is made of three steps:

2) **Correction of the optimal coordinates:** We obtain one general set of optimal coordinates on the edges crossed by the interface by averaging the ones we had at the previous step.

Result after the first correction





On the left: the cercle after the correction of the optimal coordinates.

On the right: a zoom on new interface.

DPIR: extension to three materials

We apply a new strategy, that is made of three steps:

3) **Correction of volumes:** At this point, we treat separetly the mixed cells, by distinguish the ones with two or three materials.

- Two materials cells: we apply the "standard" correction of DPIR (with Bezier curves) since we have just two materials. we evaluate the signs of the corrections, that determine an order of treatment that ensures robustness.

DPIR: extension to three materials

We apply a new strategy, that is made of three steps:

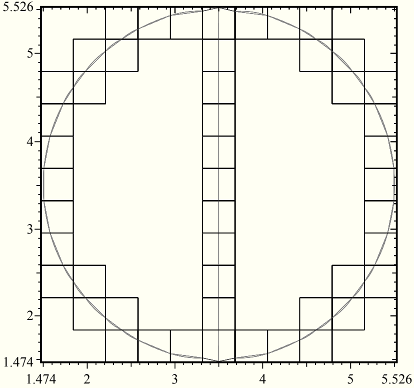
3) **Correction of volumes:** At this point, we treat separetly the mixed cells, by distinguish the ones with two or three materials.

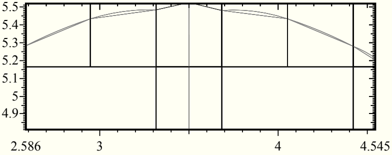
- Three materials cells: Defining

$$v := \begin{pmatrix} \mathrm{vol}(P_i^1, P_{i+1}^1) - (\mathrm{vol}_{\mathrm{target}})^1 \\ \mathrm{vol}(P_i^2, P_{i+1}^2) - (\mathrm{vol}_{\mathrm{target}})^2 \\ \mathrm{vol}(P_i^3, P_{i+1}^3) - (\mathrm{vol}_{\mathrm{target}})^3 \end{pmatrix}$$

  we evaluate the signs of the corrections, that determine an order of treatment that ensures robustness.

Result after the second correction

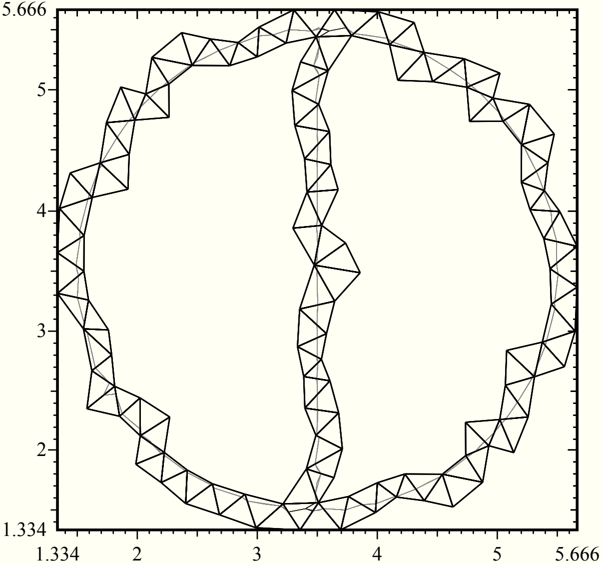




On the left: the cercle after the correction of the volumes.
On the right: a zoom on the final interface ($\lambda = 0.01$, $\omega = 1$).

Filament issue

Detection of the issue



Filament issue

DPIR can't handle the case in which the interface **crosses twice** an edge of a cell. This can happen :

- when we deal with compressible fluids,
- if the mesh is not sufficiently refined,
- near the triple points in the three materials case.

Strategies:

- locally refine the mesh and correctly repartition the volume fractions
- increase the number of dof in the cost functional (that induces a different topology at the DynPro level)
- change the discretizations on the segments
- merge the filament cells and treat them together in a single correction step

Conclusions and Perspectives

Conclusion

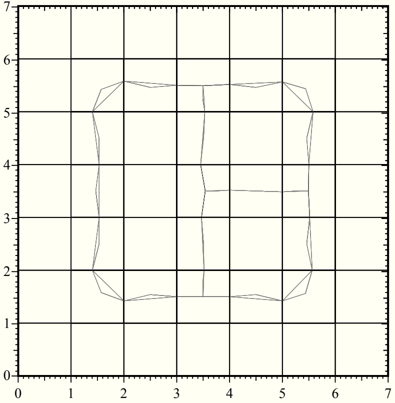
What we did:

- We modified DPIR in order to reconstruct curve interfaces,
- We improved the robustness of DPIR in the case of unstructured meshes:
  - by adding the penalization $\widetilde{\lambda}$
  - by changing the discretization of the edges at the DynPro level by using Chebyshev nodes
  - by modifiyng the search direction of the control point
- We extended DPIR to three materials for structured meshes.

Perspectives

Short term:

- treat the filament issue,
- locally adapt to each cell the choice of the Bézier weight ω,
- other tests for the three materials case.



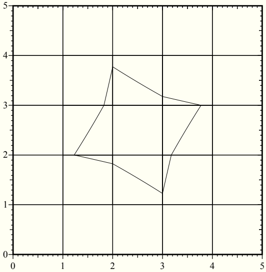
Long term:

- rewrite the DynPro taking into account the new improvements,
- extend to n materials,
- coupling to ALE.

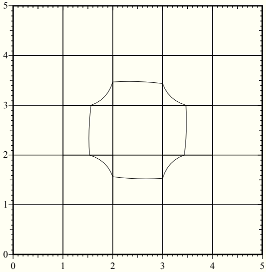
1. [D.L. Youngs, Time-dependent multi-material flow with large fluid distorsion, Numerical Methods for Fluid Dynamics, 24 (1982), 273-285]
2. [P.O. Persson, The Level-Set method, Lecture Notes, MIT 16, 920J / 2,097J / 6,339J, Numerical Methods for Partial Differential Equations, 2005]
3. [L. Dumas, J.M. Ghidaglia, P. Jaisson, R. Motte, A new volume-preserving and continuous interface reconstruction method for 2D multi-material flows, Int. J. Numer. Meth. Fluids (2017)]
4. [X. Roynard, P. Hoch, S. Borel-Sandou, Extension du schéma VoFiRe aux maillages à bords coniques, rapport stage 2013]

Thank you for your attention!

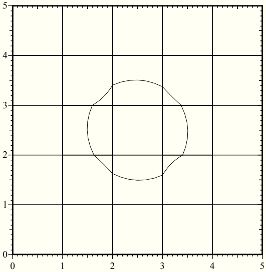
Penalization comparison



$\lambda = 0$

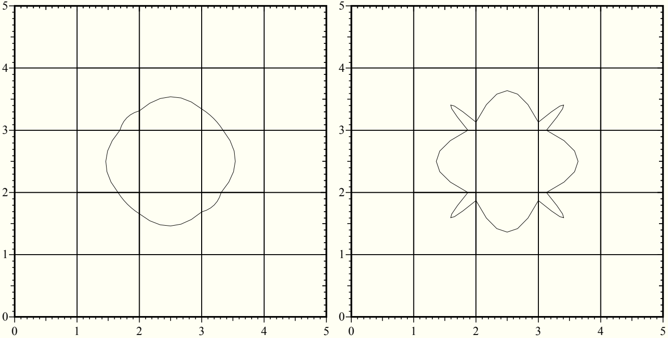


$\lambda = 0.01$



$\lambda = 0.1$

Penalization comparison



$\lambda = 0.2$ $\lambda = 0.5$