L'ALLIANCE
Centrale Nantes | Audencia Nantes École de Management | nantes ensa architecture

Advanced numerical techniques for multiphase flows in image-based simulations

Génération de maillage à partir d'imagerie 3D in-situ de l'écoulement et calcul

L. Silva
Ecole Centrale de Nantes, Institut de Calcul Intensif

P. Laure, T. Coupez, H. Digonnet

MINES ParisTech – CMM
J.-X. Zhao
E. Décencière
D. Jeulin

INP Grenoble – 3SR
T. Laurencin
L. Orgéas
P. Dumont
S. Rolland du Roscoat

ICI – Ecole Centrale de Nantes

○ Centre de recherche et technologie en Calcul Intensif

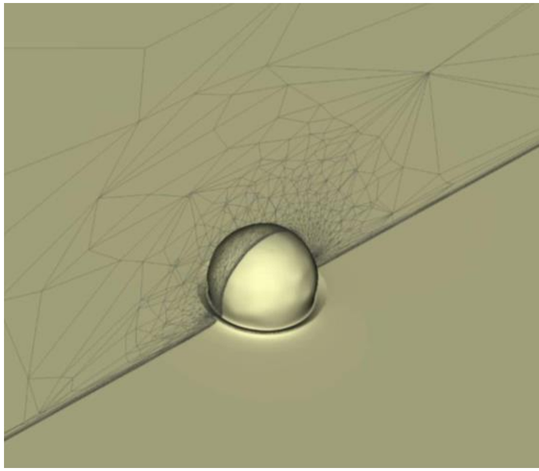
- Crée le 1er Septembre 2014, projet Connect Talent (Pays de la Loire) et Ecole Centrale de Nantes
- Equipe Sep 2015: 10 personnes
- Equipe depuis Sep 2016: ~ 30 personnes
  - 9 permanents (2 ECs titulaires, 5 ECs contractuels, 1 IR, 1 IE)
  - 1 assistante de direction
  - ~ 20 doctorants et stagiaires

- **Structuration**: **laboratoire de recherche** et **mésocentre de calcul** pour
  - la démocratisation du calcul intensif et l'usage par le plus grand nombre
  - développer un mésocentre régional (Tier2) attractif: puissance de calcul à min 1/10 du plus gros supercalculateur national -> **LIGER**
  - développer un laboratoire de recherche en modélisation par le calcul intensif
    - développer des applications logicielles en collaboration avec les autres laboratoires d'ECN, d'autres partenaires académiques, des centres techniques et des industriels de la région
  - développer des outils de formation et de transfert au calcul intensif

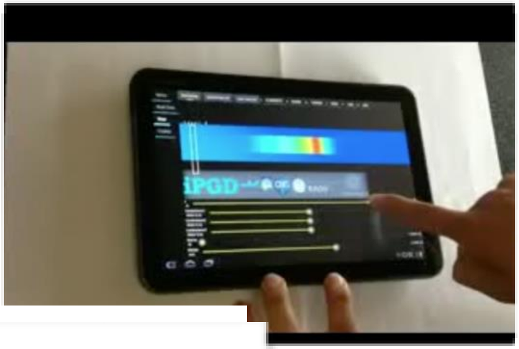
ICI – Ecole Centrale de Nantes

○ Centre de recherche

- **Thématiques de Recherche**



  – Equipe HPC: méthodes numériques pour une plateforme logicielle massivement parallèle, **ICITech**
    - Approche monolithique, interfaces implicites et immersion, éléments finis
    - Génération et adaptation de maillage anisotrope automatique
    - Calcul et programmation massivement parallèle, « cloud computing » ou CI en ligne, lien avec les grands volumes de données pour le calcul scientifique
    - Visualisation parallèle, immersive 3D et réaliste



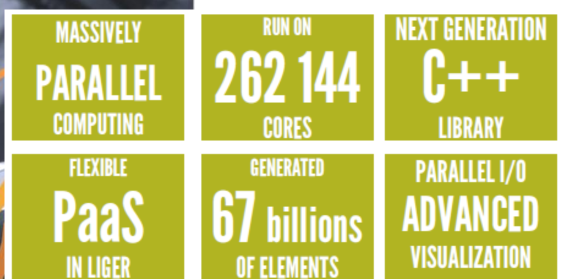
iPGD: Real-time Simulation in hand-held devices

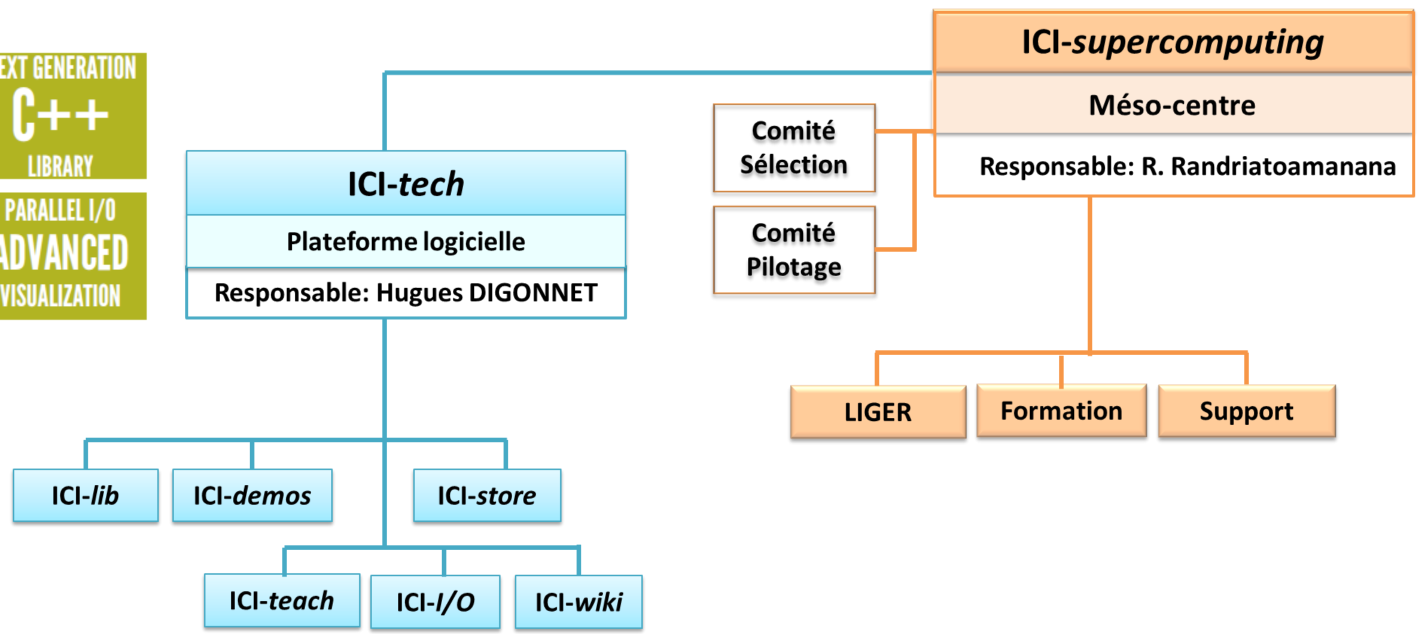
  – Equipe DATAbest: apprentissage automatique et **réduction de modèle**
    - Numerical Methods for Parametric and High-dimensional Problems: PGD
    - Data Dimensionality Reduction
    - Multi-resolution and Non-intrusive Approaches
    - Dynamic Data-Driven Application Systems (DDDAS)
    - Simulation Apps and Real-time Simulation

ICI-Tech

Structure

- ICI-Tech est un environnement pour la création d'applications logicielles et qui comprend, entre autres, les items suivants :
  - ICI-lib, une bibliothèque d'objets de calcul scientifique écrite en C++, basée sur des éléments finis et associée à une librairie de maillage, les deux hautement parallèles. Cette librairie est libre, avec une licence type CeCILL-B ;
  - ICI-store, des applications orientées utilisateurs, qui peuvent inclure les scripts pour les composer, ainsi que des fichiers sources spécifiques, qui ne seront pas mis dans la distribution ICIlib commune, pour des raisons de confidentialité/exclusivité. Ces fichiers source devront concerner des développements très spécifiques et en aucun cas toucher au coeur de la librairie.





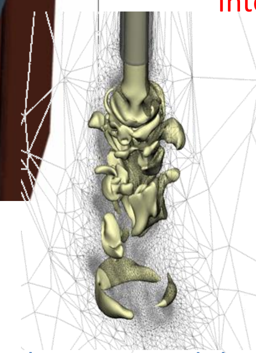
ICI-Tech

- Applications

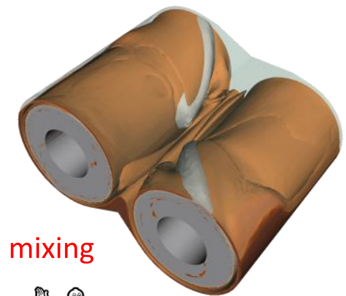
Materials and processes

Composites

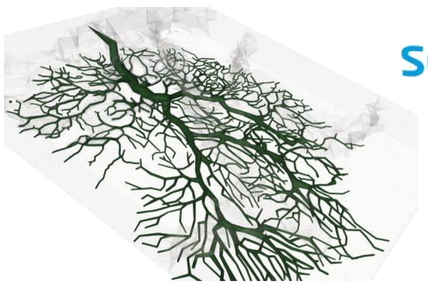
Wiping and steel treatments

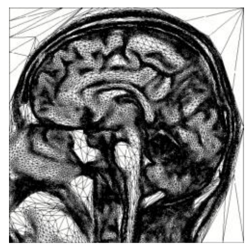


Internal mixing



Biological systems



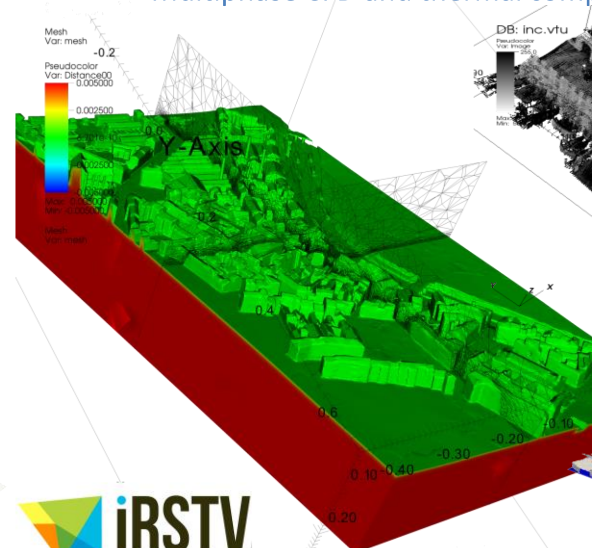


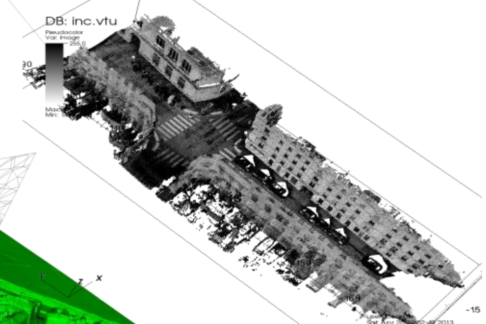
Neuron reconstruction, 3D imaging

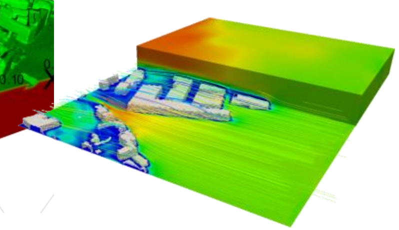
Urban environments

3D reconstruction

Multiphase CFD and thermal computations at different scales

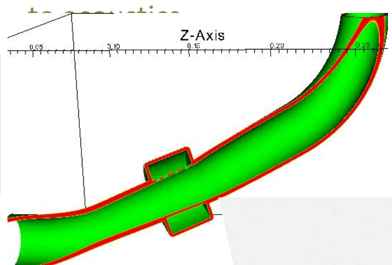






Energy

Air flows in ducts, linking



Offshore wind farms

Equipe HPC
Méthodes numériques

Motivations (I)

- Example of wiping: an extreme process

Heavy Liquid (molten Zinc) moving up along a vertical surface at 2 m/s

High surface tension coefficient (10 times the water/air value)

Geometry: 0.20 m large, 0.60 cm high

Air jet impacting the liquid surface: 200 m/s

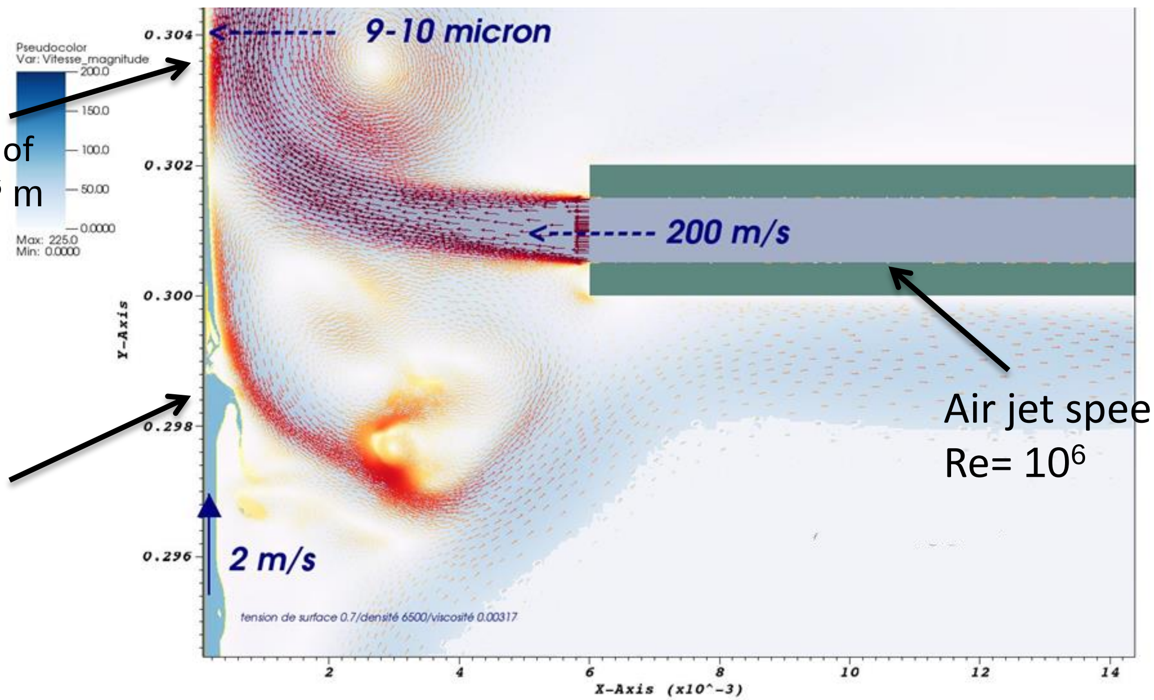
Liquid thickness above the impacting jet: less that 10 µm = 10^{-5} m

Smallest edge of the mesh = 1 µm = 10^{-6} m

uniform mesh would require : 120 billions of nodes

Present anisotropic adaptive calculation: **50 000 nodes**

Film of liquid thinning of less than 10 µm = 10^{-6} m



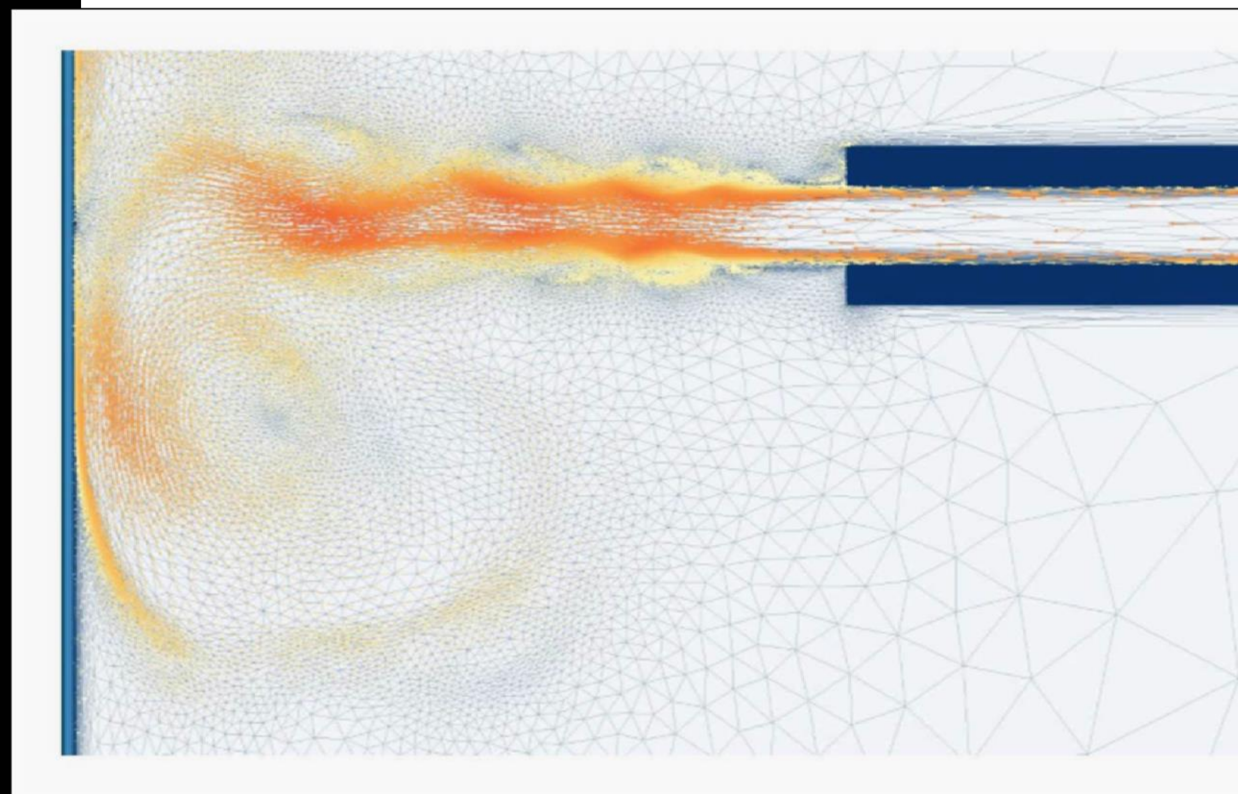
Droplet formation

Air jet speed: 200 m/s
Re= 10^6

[Coupez, 2013]

Motivations (II)

- Example of wiping: an extreme process

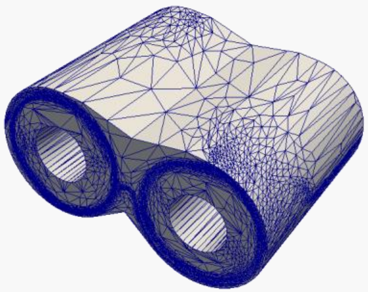


Approche numérique

○ Description courte et composantes principales

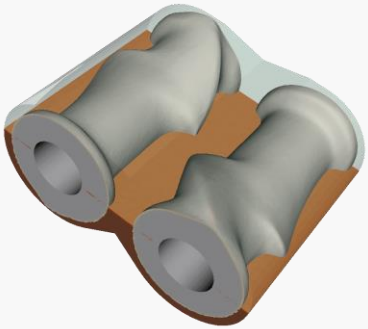
- Approche parallèle, monolithique et adaptative

Un seul maillage



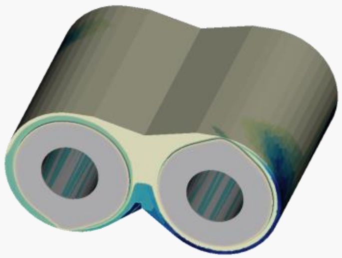
Maillage anisotrope automatique

Fonction de phase pour chaque object (2 rotors et un fluide)

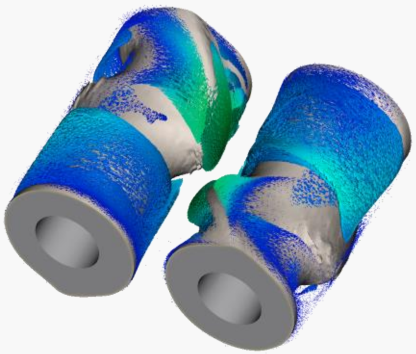


Rotors: fonction de phase calculée exactement, le mouvement est imposé
Fluide: fonction de phase obtenue par résolution d'une équation de transport

Propriétés matériau: détermination des propriétés par une loi de mélange



Vitesse et **pression**: calculé par résolution des équations de NS avec des EFs stabilisés et la librairie PETSc



- Tous les composants massivement parallélisés

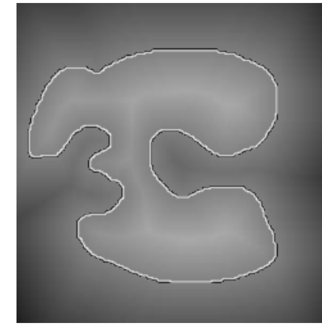
Frontières implicites (I)

- Réprésentation et ε-regularisation

- Signed distance function

$$\alpha = \bar{d}(x,\Gamma) = \begin{cases} d(x,\Gamma) \text{ if } x \in \omega \\ -d(x,\Gamma) \text{ if } x \notin \omega \end{cases}$$

α



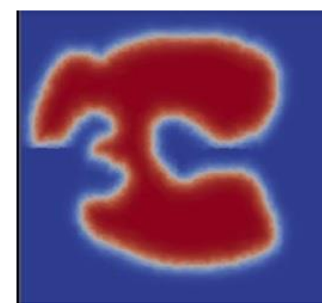
- Heaviside function $H(\alpha) = 1_\omega(x) = \begin{cases} 1 \text{ if } \alpha > 0 \\ 0 \text{ if } \alpha < 0 \end{cases},$

$H(\alpha)$

- Smoothing $\boxed{u_\varepsilon = u(\alpha,\varepsilon) = \varepsilon \tanh(\alpha/\varepsilon)}$

$$H_\varepsilon(u,\varepsilon) = \frac{1}{2}(1 + \frac{u_\varepsilon}{\varepsilon})$$

u_ε



[Ville et al, 2011], [Silva et al, 2014]

Frontières implicites (II)

ArcelorMittal

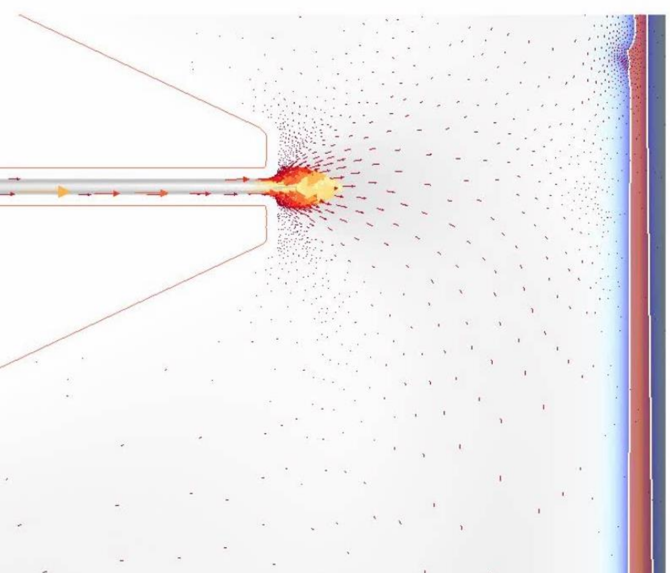
Interfaces implicites dynamiques

- Driven motion: convective reinitialization
  - Advection of the interface by solving a classical hyperbolic equation $\frac{\partial u_\epsilon}{\partial t} + v \cdot \nabla u_\epsilon = 0$
  - Loss of the metric properties of the phase function
  - Reinitialization $$\frac{\partial u_\epsilon}{\partial \tau} = s(u_\epsilon)(|\nabla u_\epsilon| - (1 - (\frac{u_\epsilon}{\epsilon})^2))$$
  - Original idea: combine convection and reinitialization

$$\boxed{\frac{\partial u_\epsilon}{\partial t} + (v + \lambda v_r) \cdot \nabla u_\epsilon = \lambda s(u_\epsilon)(1 - (\frac{u_\epsilon}{2})^2)}$$

$v_r = s(u_\epsilon)\frac{u_\epsilon}{|\nabla u_\epsilon|}$ is the reinitialization velocity

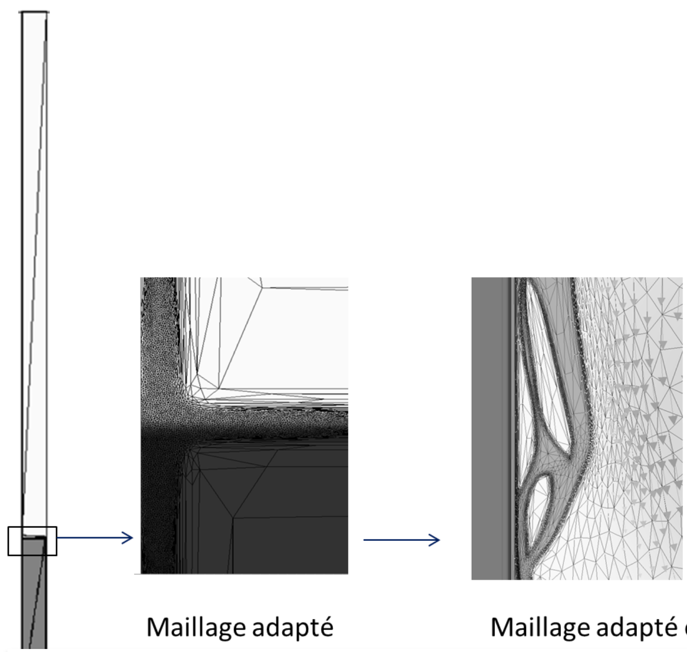
$\lambda \sim \frac{\partial \tau}{\partial t}$ will guarantee one mesh contour displacement per time step, enough if CFL



Anisotropic H-R-P adaptation

A short summary

- On each core the mesher works using a metrics field
- Computation of the metrics field is based on an a posteriori error estimator directly computed at nodes and equi distribution of the error under the constraint of a imposed number of nodes
- Is extended to multi-solution fields: phase functions and their Dirac, velocity
- Combined with R-adaptation (mesh velocity)
- New developments: P-adaptation



Maillage adapté   Maillage adapté en

[Coupez, 2011], [Coupez et al, 2013]

An adaptive anisotropic VMS formulation (I)

- Starting point: the heterogeneous Navier-Stokes equations

- Incompressible Navier-Stokes equations, with surface tension

$$\begin{cases} \rho \dfrac{dv}{dt} - \nabla \cdot (2\eta\varepsilon(v)) + \nabla p = \rho g + f_{st} \\ \nabla \cdot v = 0 \end{cases}$$

- Stabilized VMS formulation

- Multiphase incompressible Navier-Stokes equations
  - Mixture laws $\xi = \xi_\epsilon = \xi_\omega H_\epsilon + \xi_{\Omega\setminus\omega}(1 - H_\epsilon)$

- Boundary conditions by penalisation of the motion or of the deformation, in a volumic way, using Lagrange multipliers

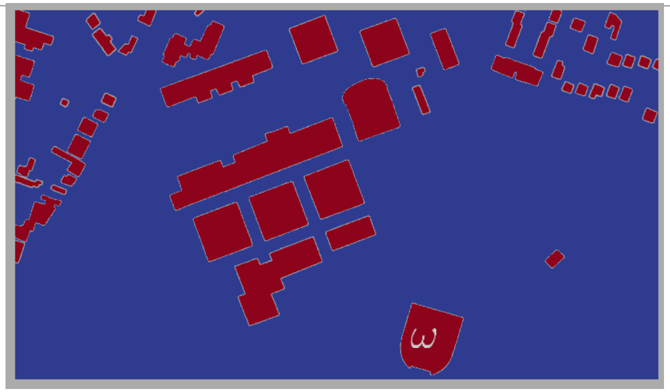
- Numerical resolution: in //
  - Newton for linearisation
  - PETSc library (Krylov + multigrid methods)

[Coupez et al, JCP 2013]

Multiphase flows (I)

Implicit boundary conditions

Ω



- Imposing boundary conditions in a volume

$$(u - u_0, w)_\Gamma = 0 \qquad (\delta_\Gamma(u - u_0), w)_\Omega = 0$$

$$(\delta_\epsilon[u - u_0], w)_\Omega = 0$$

  - The Dirac function is the derivative of the Heaviside (in our ε-smooth manner),

$$\delta_\epsilon = H'_\epsilon = \frac{1}{2\epsilon}[1 - (\frac{u_\epsilon}{\epsilon})^2]$$

- Changes in the variational formulation -> weak imposition

$$\mathcal{L}(u, \lambda) = \tfrac{1}{2}(\nabla u, \nabla u)_\Omega + \tfrac{r}{2}(\delta_\epsilon[u - u_0], u)^2_\Omega + \lambda(\delta_\epsilon[u - u_0], u)_\Omega$$

  - Resolution using an Uzawa's

- Important:
  - no need of the sign of the distance function, meaning less problems when using the point clouds
  - the Lagrange multiplier gives us the force, interesting for other applications

[Santoso et al, 2016]

Images et simulation

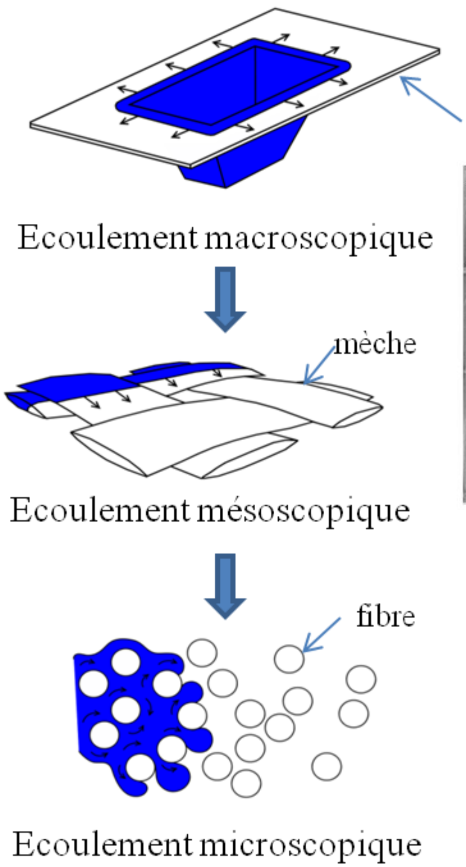
Plan

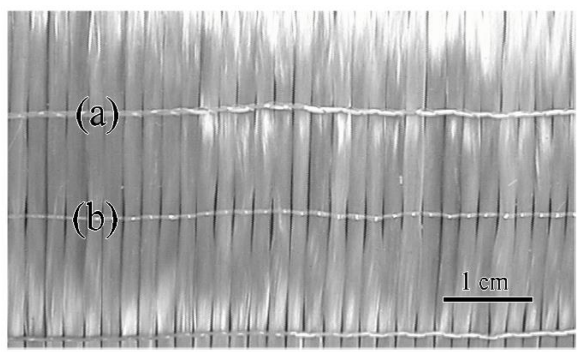
- Contexte
- Premiers travaux sur la simulation directe sur image
- Mailleur d'images
- Construction des fonctions de phase par une méthode de redistanciation
- Ecoulements multiphasiques
- Applications
  - Calcul de perméabilité 3D
- Conclusions et perspectives

Contexte

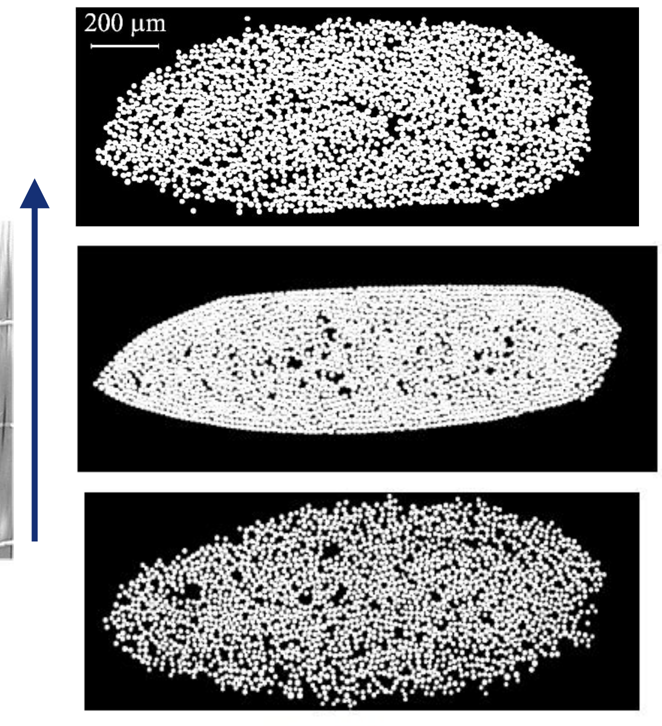
La triple échelle

- Calcul aux échelles micro/meso, en prenant en compte le caractère aléatoire du renfort





Mèches de 2000 fibres dans un UD



M-CFD: calcul multiphasique basé mécanique des fluides

- notre choix: Efs stabilisés, une approche Eulérienne et des fonctions de phase implicites
  - Interface diffuse et épaisseur d'interface ε
- **calcul directe à partir des images**

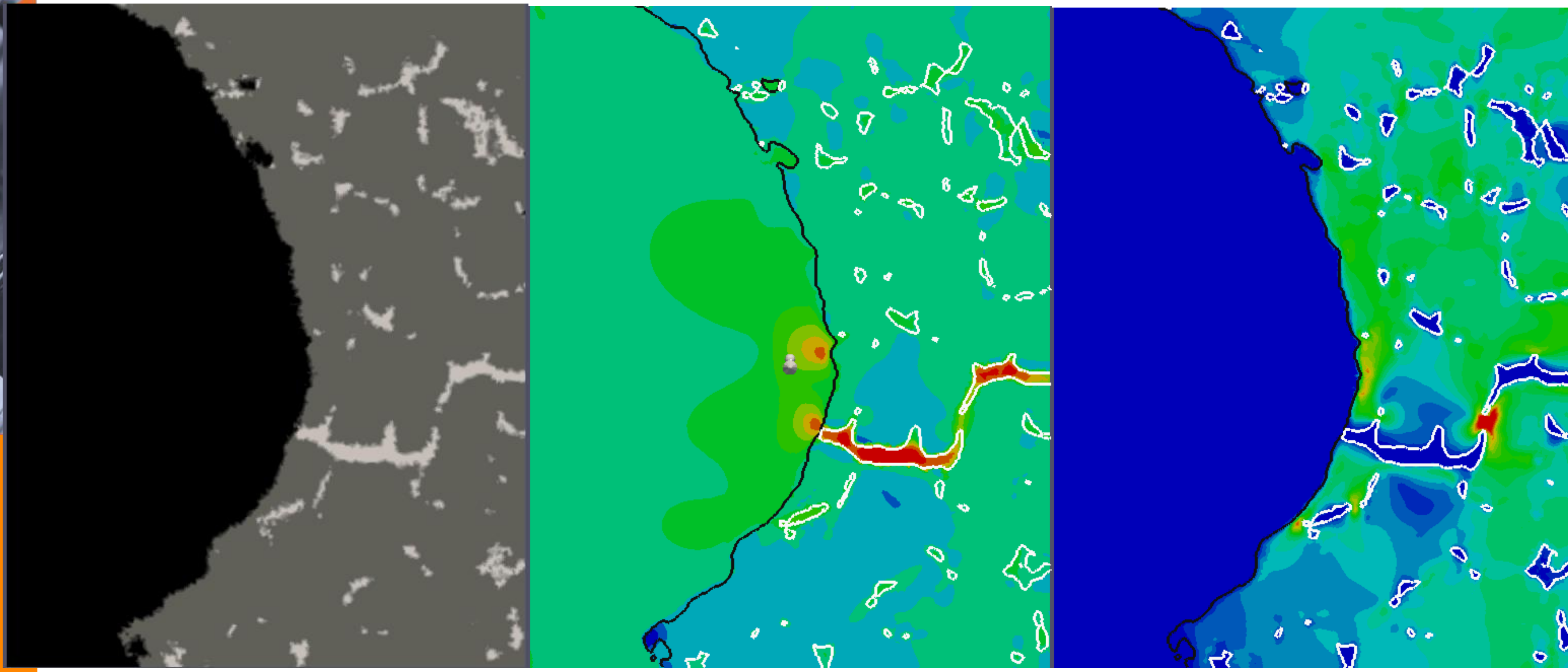
[Gantois (2012)], [Bizet (2004)]

Simulation directe sur image: l'avant

- Exemple : traction d'un aliage à l'état pâteux

ANR Simuzal (2008-2011): essais /carac SIMAP (L. Salvo)
Essai de traction d'un Al-8%Cu
Tomographie-X 3D (ESRF) in-situ
Cylindre de 2mm de diamètre et 8mm de longueur
400 images de 1024x1024x1024 voxels

Maillage: 3 Mnoeuds
Temps de calcul: 4 jours sur 50 coeurs
Comportement viscoplastique (Loi Norton-Hoff)



[Zaragoci, Silva et al (2011)]

Centrale Nantes

Simulation directe sur image: l'avant

- Exemple : traction d'un aliage à l'état pâteux

ANR Simuzal (2008-2011): essais /carac SIMAP (L. Salvo)

Essai de traction d'un Al-8%Cu

Tomographie-X 3D (ESRF) in-situ

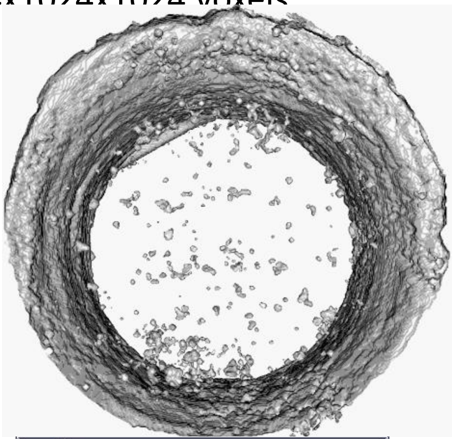
Cylindre de 2mm de diamètre et 8mm de longueur

400 images de 1024x1024x1024 voxels

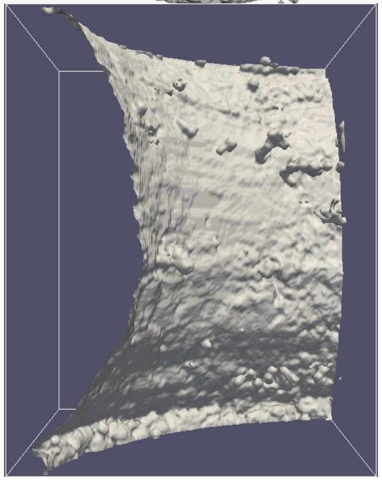
Maillage: 3 Mnoeuds

Temps de calcul: 4 jours sur 50 coeurs

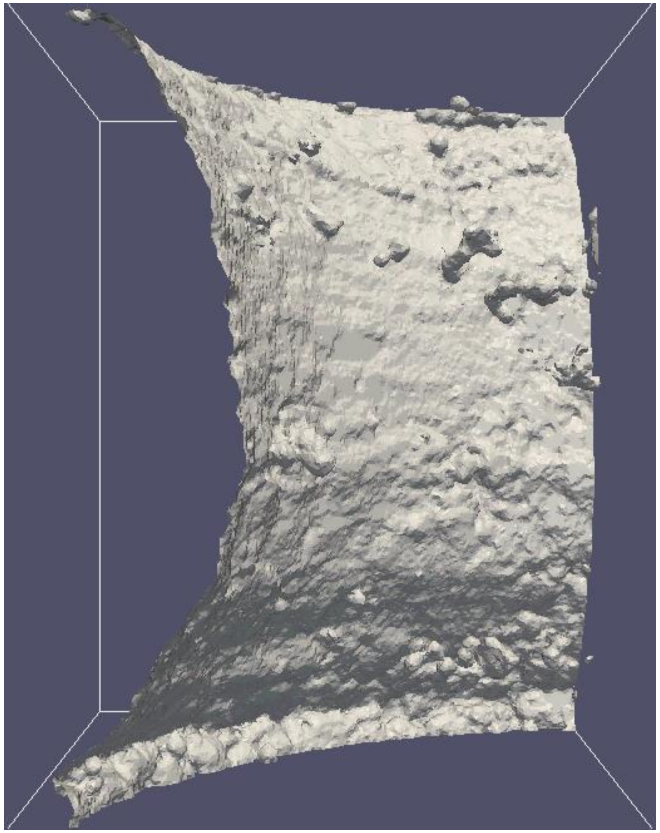
Comportement viscoplastique (Loi Norton-Hoff)



IMAGE



CALCUL

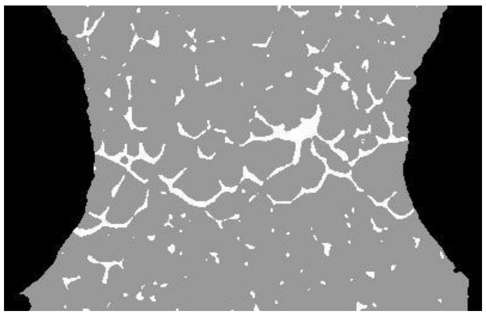


[Zaragoci, Silva et al (2011)]

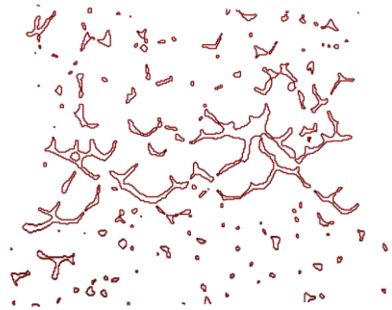
Simulation directe sur image: l’avant

Exemple : traction d’un aliage à l’état pâteux

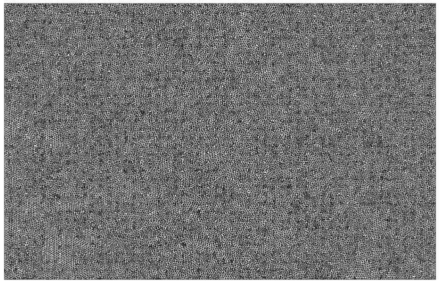
- Problème: bloquage à 3 millions de noeuds
  - sur la géneration du maillage initial et de la répresentation des différentes phases et sur le suivi des phases au cours de la traction



Segmentation (triphasique)

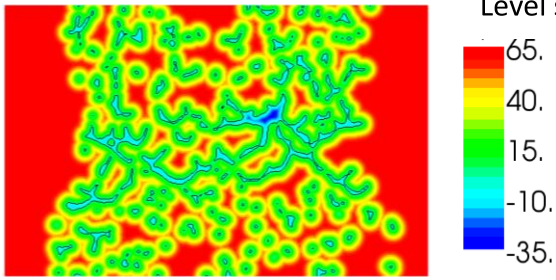


Marching Cubes



Immersion

Level set liquide (µm)



- Géneration directe du **maillage** de la microstructure sur l’image, avec sa **segmentation**

  - sur notre solveur itérative « classique » (Krylov) qui a une complexité non-linéaire ($\mathcal{O}(n^{3/2})$)
    - Implementation d’une méthode multigrille interfacée avec PETSc

- Application au suspensions de fibres dans une matrice polymère

[Zaragoci, Silva et al (2011)]

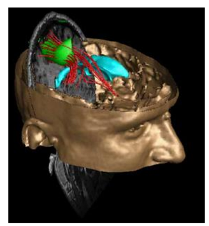
Un mailleur d’images

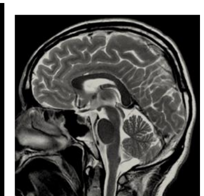
Un mailleur d'images

Acquisition et image

- Calcul de la valeur du voxel dans le maillage: interpolation directe

Image Techniques





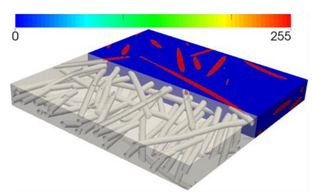
MRI : magnetic resonance imaging

Medical applications

- Image $\hat{u}$: Pixel (2D), Voxel(3D).

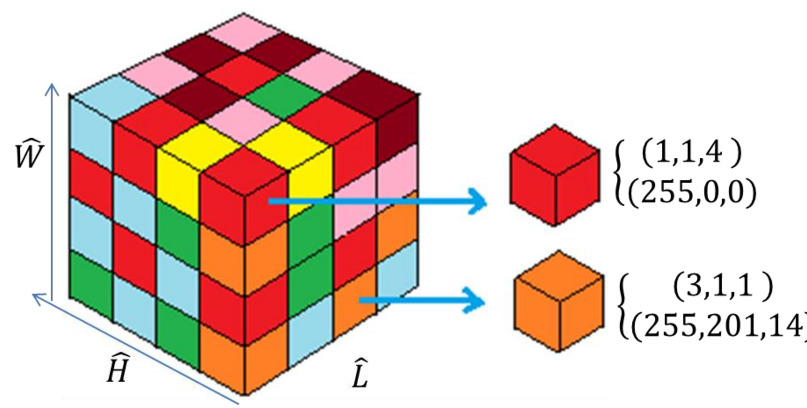
$$\hat{T} = \hat{L} \times \hat{H} = \hat{L} \times \hat{H} \times \hat{W}$$

X-ray micro-tomography



Material applications

Smartphone

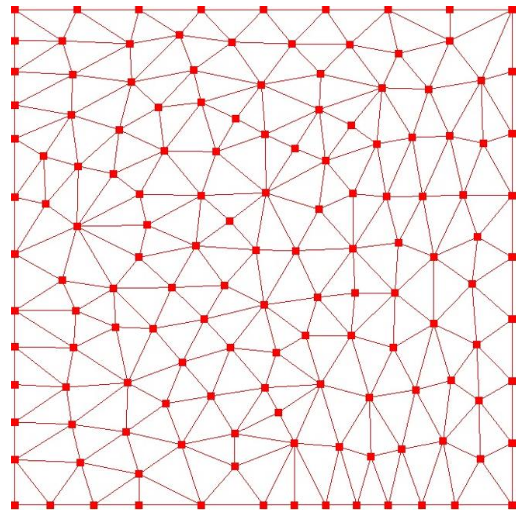


Un mailleur d'images

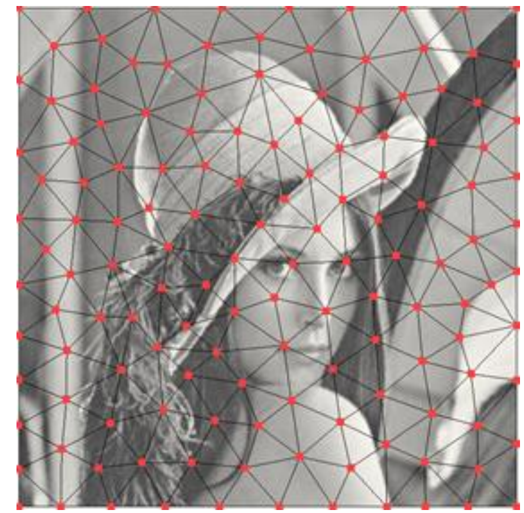
La méthode d'immersion d'image

- Calcul de la valeur du voxel dans le maillage: interpolation directe

Image $\hat{u}$
$(512 \times 512 = \hat{L} \times \hat{H} = 262114)$



Mesh 140 Nodes
$[0,1] \times [0,1] = [0,\mathrm{X}] \times [0,\mathrm{Y}]$



Mapping

$\mathbf{X}^{\boldsymbol{i}}$, noeud du maillage avec coordonnée (x^i, y^i) et valeur $u(\mathbf{X}^{\boldsymbol{i}})$

$$\mathbf{X}^i \xleftrightarrow{\;u(\mathbf{X}^i) = \hat{u}(Pixel^k)\;} \mathbf{Pixel}^k$$

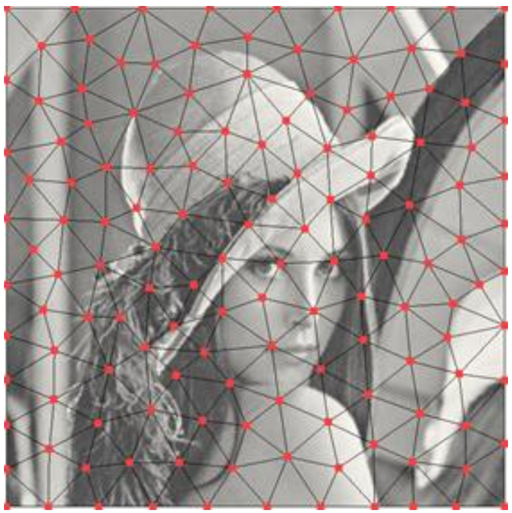
$$\widehat{l^k} = \mathrm{int}\left(\frac{x^i}{\mathrm{X}} \cdot (\hat{L} - 1) + 1\right), \widehat{h^k} = \mathrm{int}\left(\frac{y^i}{\mathrm{Y}} \cdot (\hat{H} - 1) + 1\right)$$

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

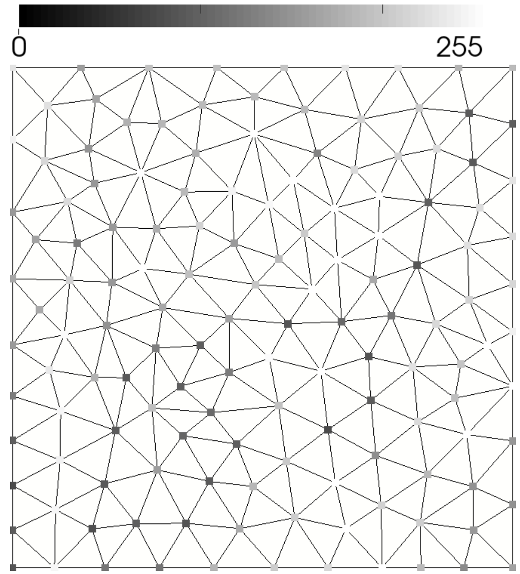
Un mailleur d'images

La méthode d'**immersion d'image**

- Calcul de la valeur du voxel dans le maillage: interpolation directe



Mapping



Valeurs aux noeuds

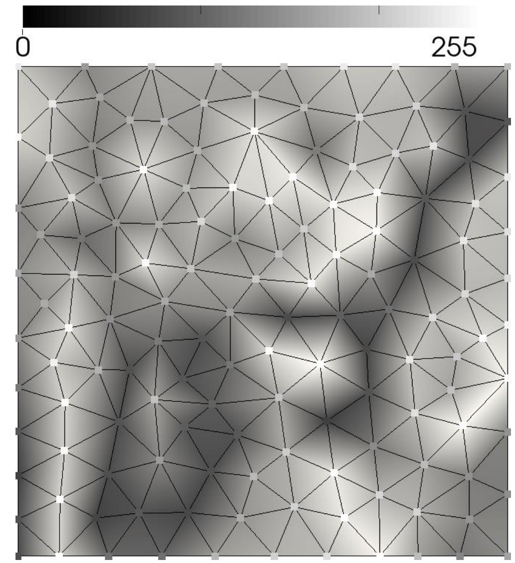


Image interpolée P1

$$\prod_h u(\mathbf{X}^i) = U^i$$

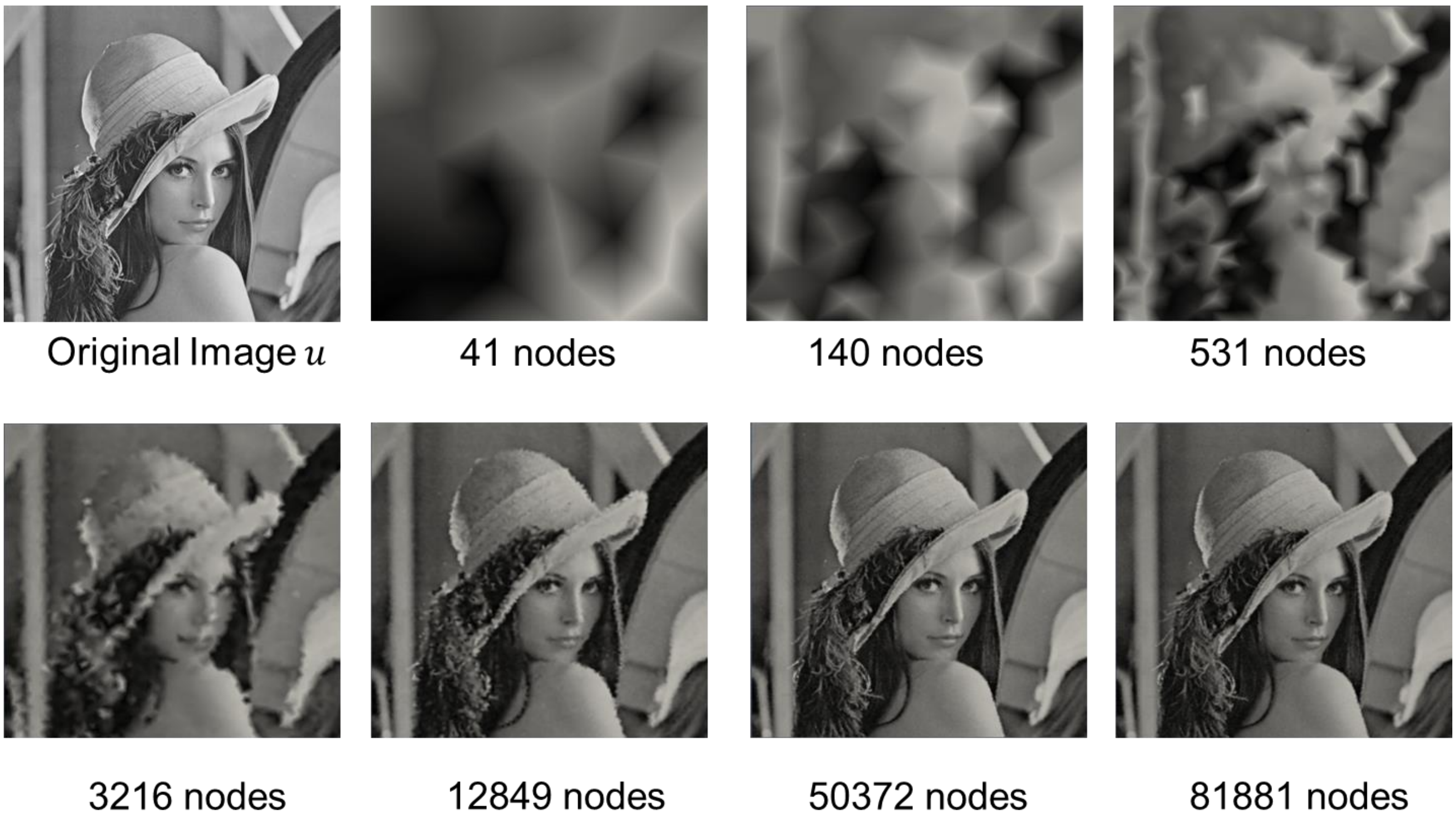
Interpolation de Lagrange

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Un mailleur d'images

La méthode d'immersion d'image

- Calcul de la valeur du voxel dans le maillage: augmenter le nombre de noeuds

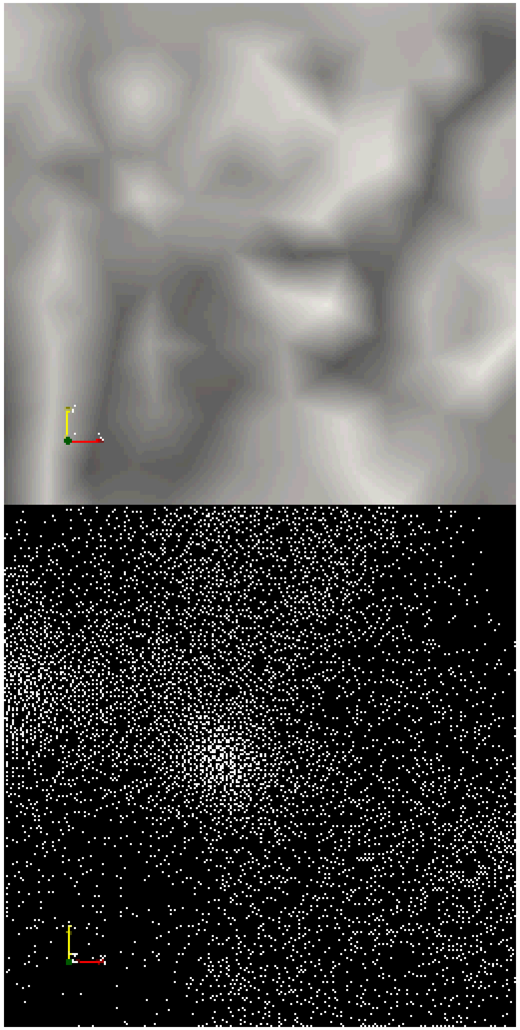


[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Un mailleur d’images

Compression via adaptation

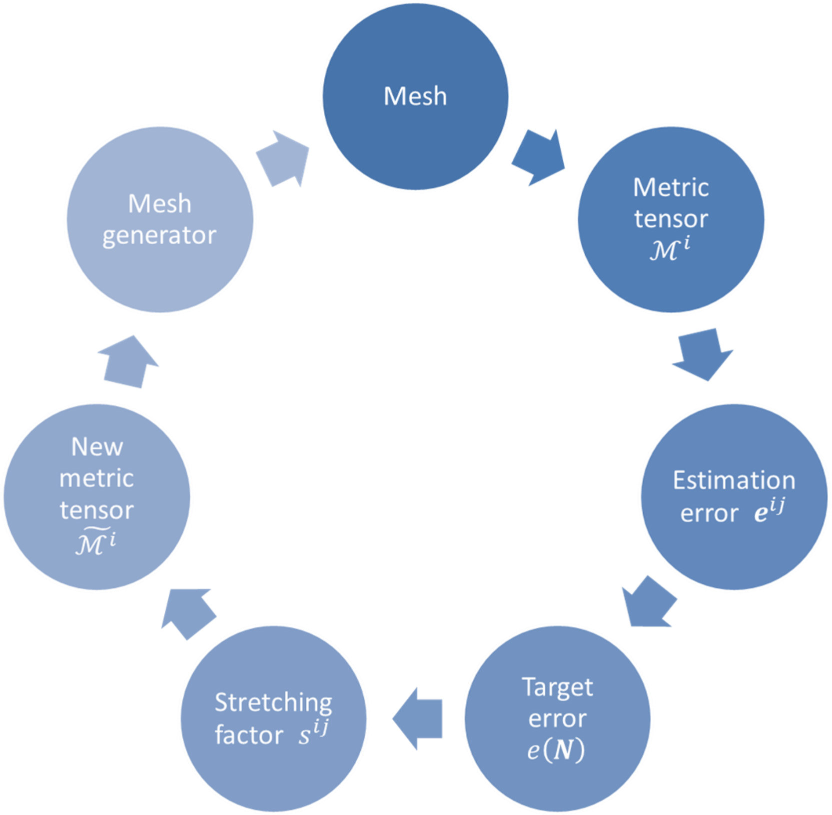
- Calcul de la valeur du voxel dans le maillage: augmenter **de façon intelligente** le nombre de nœuds: comme une superpixelisation



Un mailleur d'images

Compression via adaptation

- Idée: optimiser le placement des nœuds par adaptation anisotrope à l'aide d'un mailleur topologique par optimisation locale (parallèle)



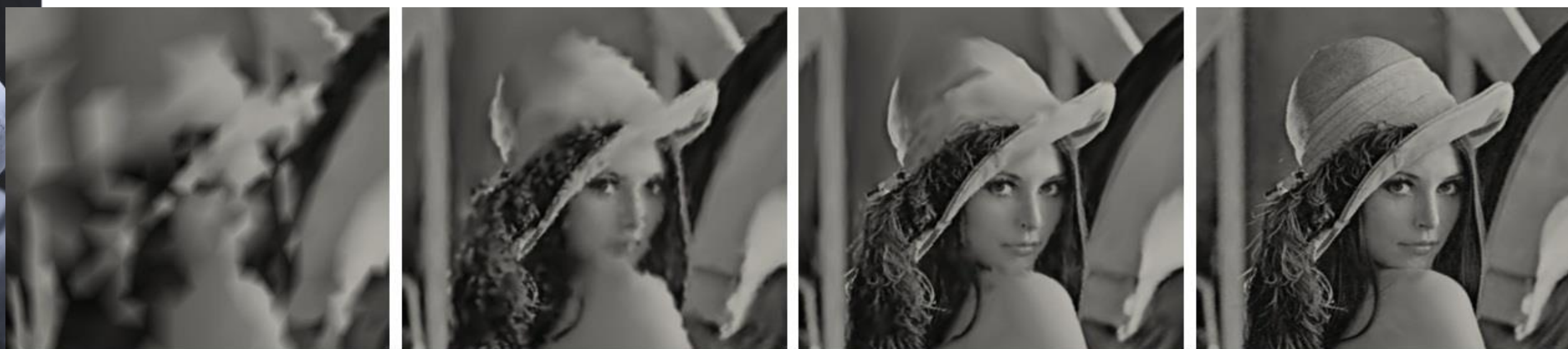
- Utilisée dans un contexte parallèle avec un repartitionnement dynamique

[Coupez (2011), Silva et al (IJCFD 2016)]

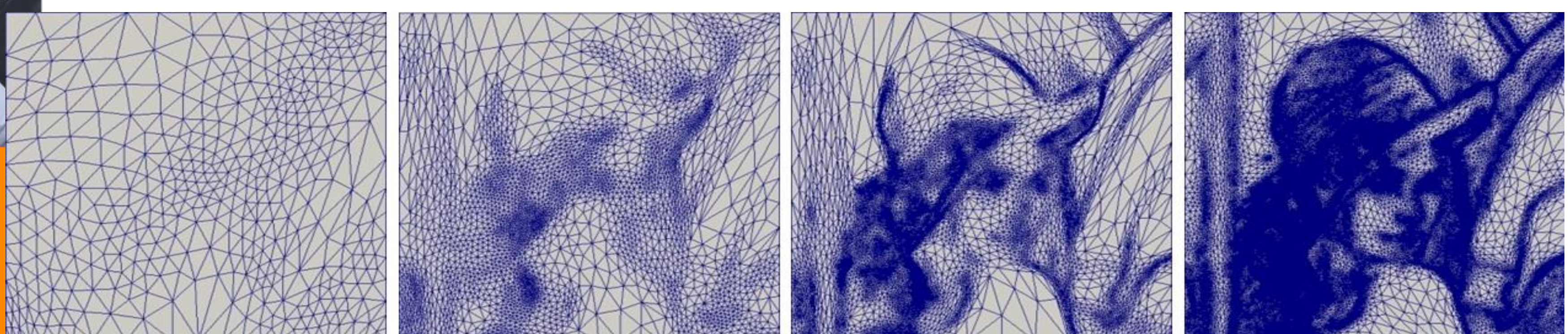
Un mailleur d'images

- Compression via adaptation

- Calcul de la valeur du voxel dans le maillage: augmenter **de façon intelligente** le nombre de nœuds: comme une superpixelisation



(a) 500 nodes (b) 3214 nodes (c) 12758 nodes (d) 82427 nodes



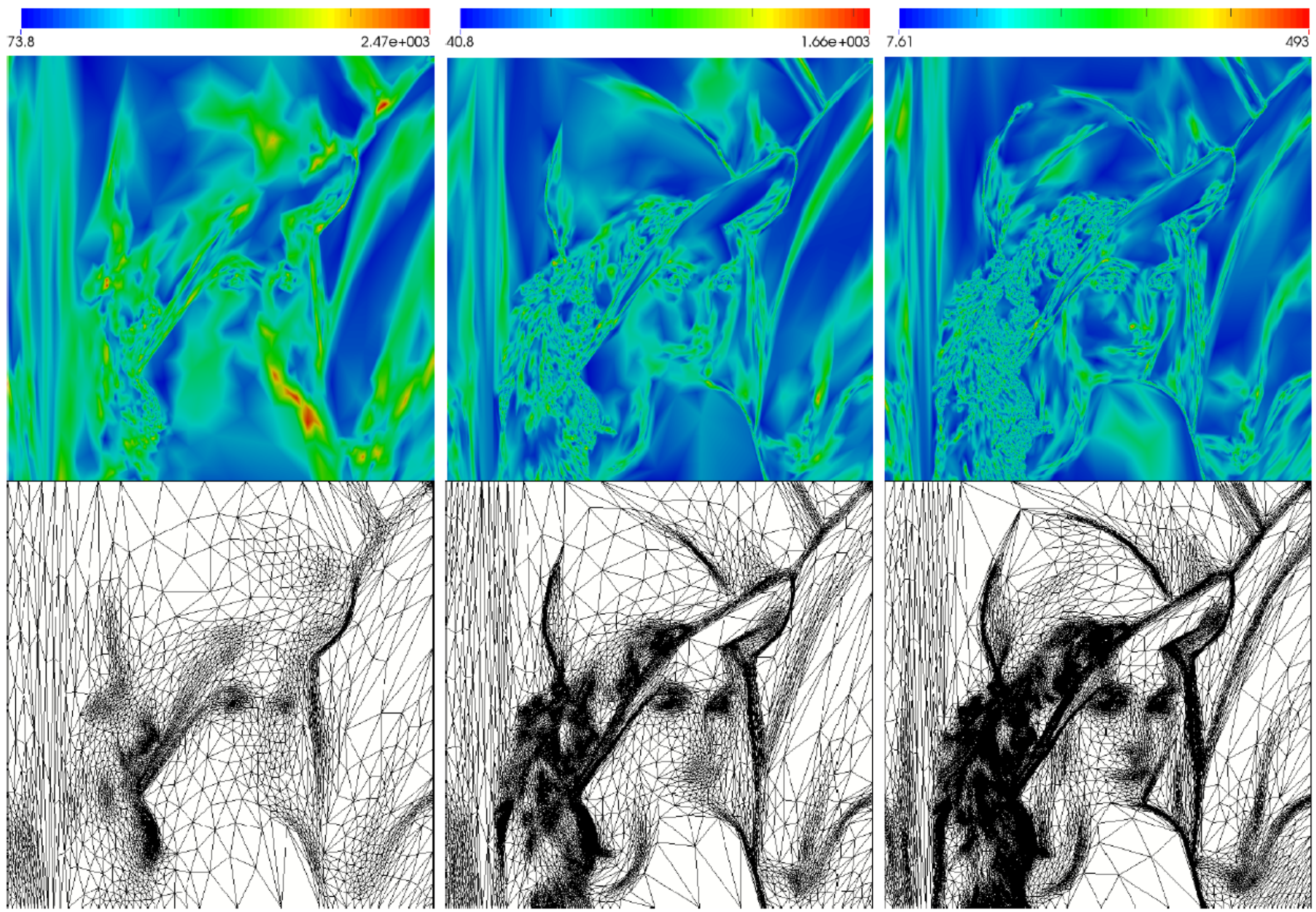
(a) 500 nodes (b) 3214 nodes (c) 12758 nodes (d) 82427 nodes

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Un mailleur d'images

○ Compression via adaptation

- Calcul de la valeur du voxel dans le maillage: augmenter **de façon intelligente** le nombre de nœuds: comme une superpixelisation



(a) $N = 3200$ (b) $N = 13000$ (c) $N = 35000$

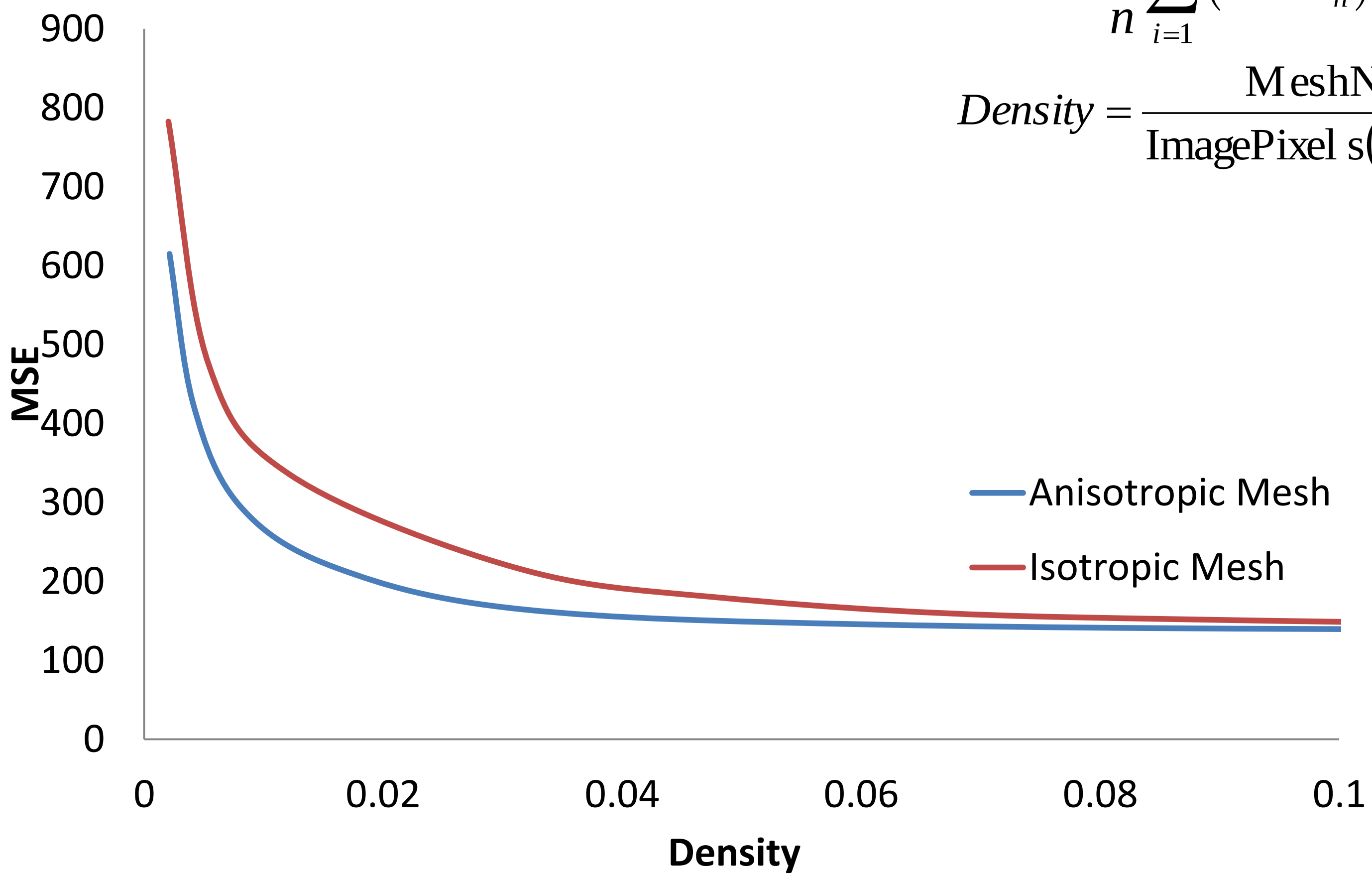
[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Un mailleur d'images

- La méthode d'immersion d'image

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{u}^{i} - u_h^{i}\right)^2$$

$$Density = \frac{\mathrm{MeshNodes}}{\mathrm{ImagePixels}(512\times 512)}$$



[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Applications images 3D

- Gros volumes spatio-temporels: performance de la méthode d'adaptation

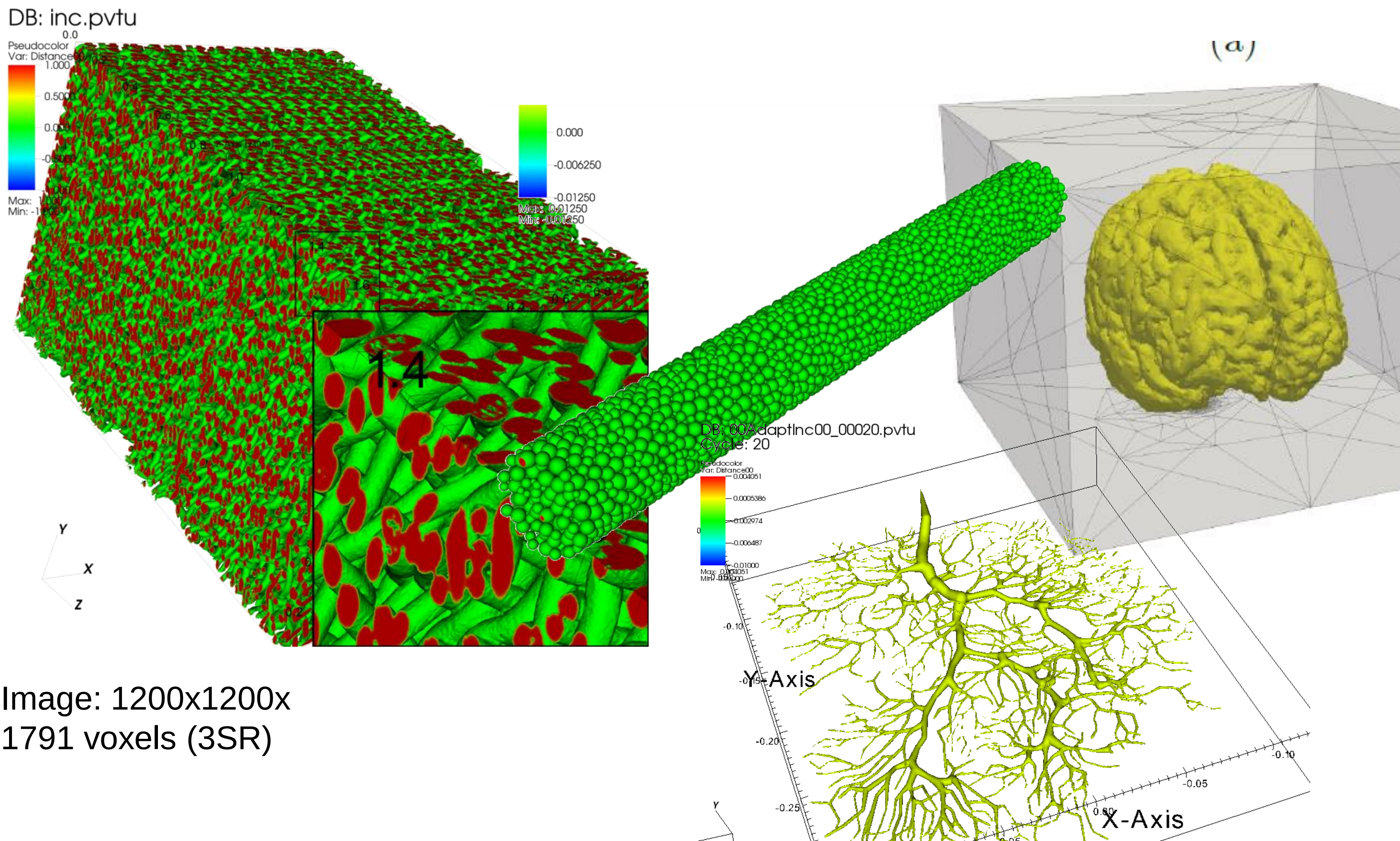
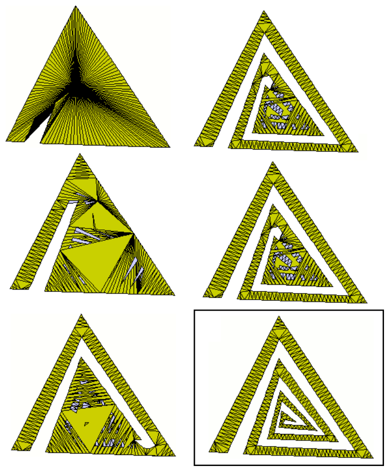


Image: 1200x1200x
1791 voxels (3SR)

Parallel strategy (I)

Goals



- Firstly: we dispose of a sequential topological mesher
  - Meshing by local mesh modification
  - Generation by local modification, volume minimization and selection is based on a quality criteria
  - Results in an iterative improvement of an initial mesh
  - Input is a metric field

- Secondly: meshing parallelization must
  - not be intrusive
  - be able to deal efficiently with isotropic and anisotropic mesh sizes
  - use unstructured an unhierarchical simplex meshes

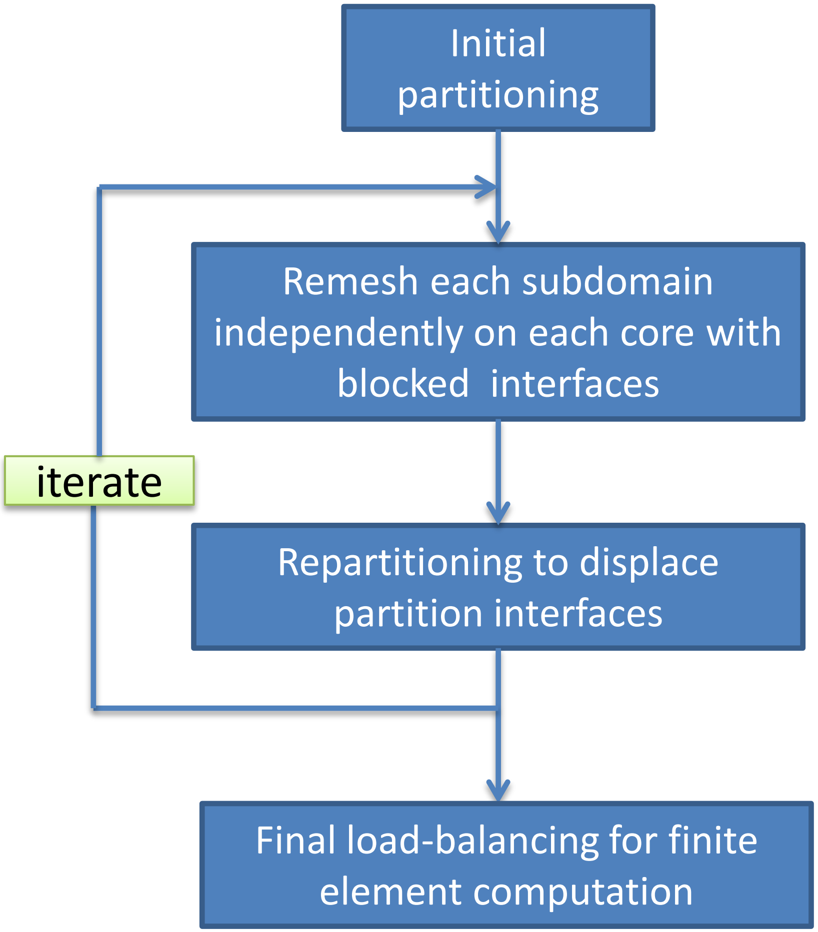
- Finally, the chosen solution is

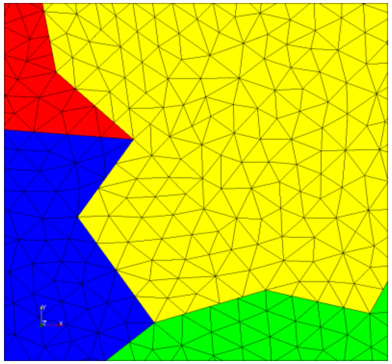
→ we don't parallelize directly the mesher, but we use it in a parallel context coupled with a parallel mesh (migration) repartitioner: **DRAMA**

[Coupez, 2000], [Digonnet, 2001]

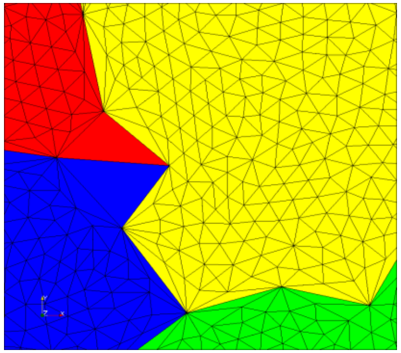
Parallel strategy (II)

- Parallelization





Without constraint : we don't have a global mesh !



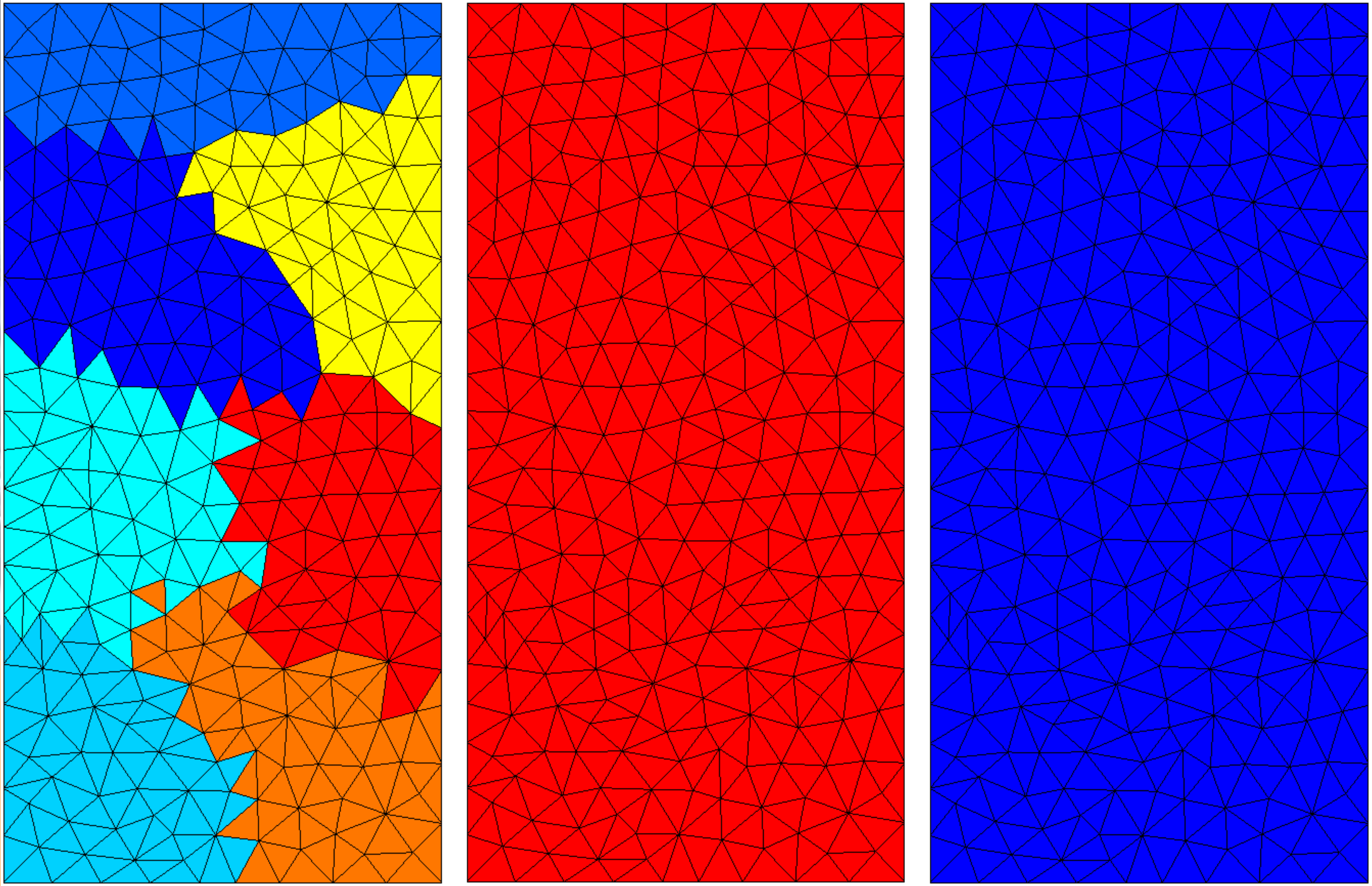
Constrained (frozen interfaces): we have a global but not perfect mesh

[[Digonnet, 2001]

Parallel strategy (III)

- Illustration: 2d case

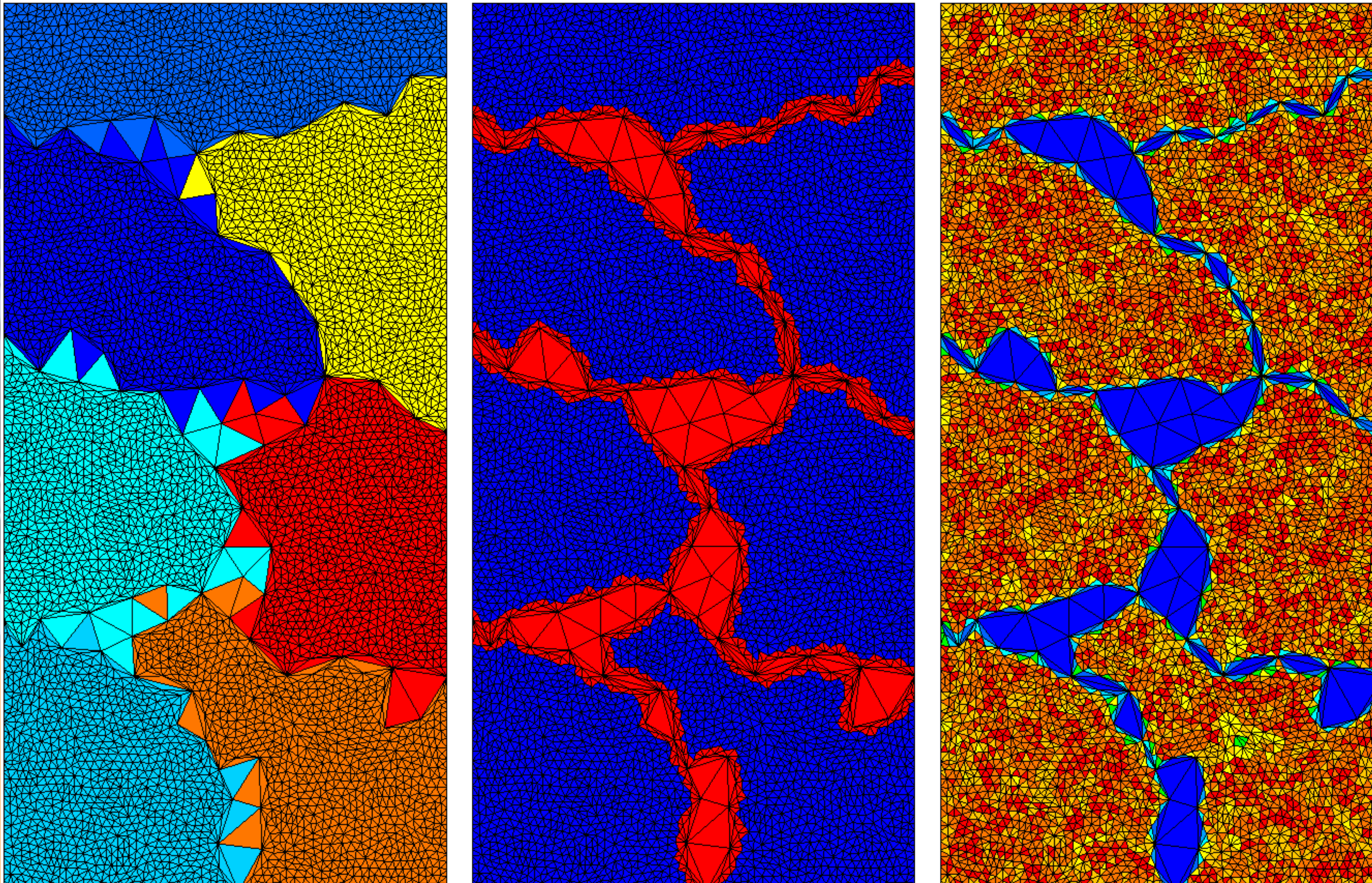
Initial partition on 7 cores



Parallel strategy (III)

- Illustration: 2d case

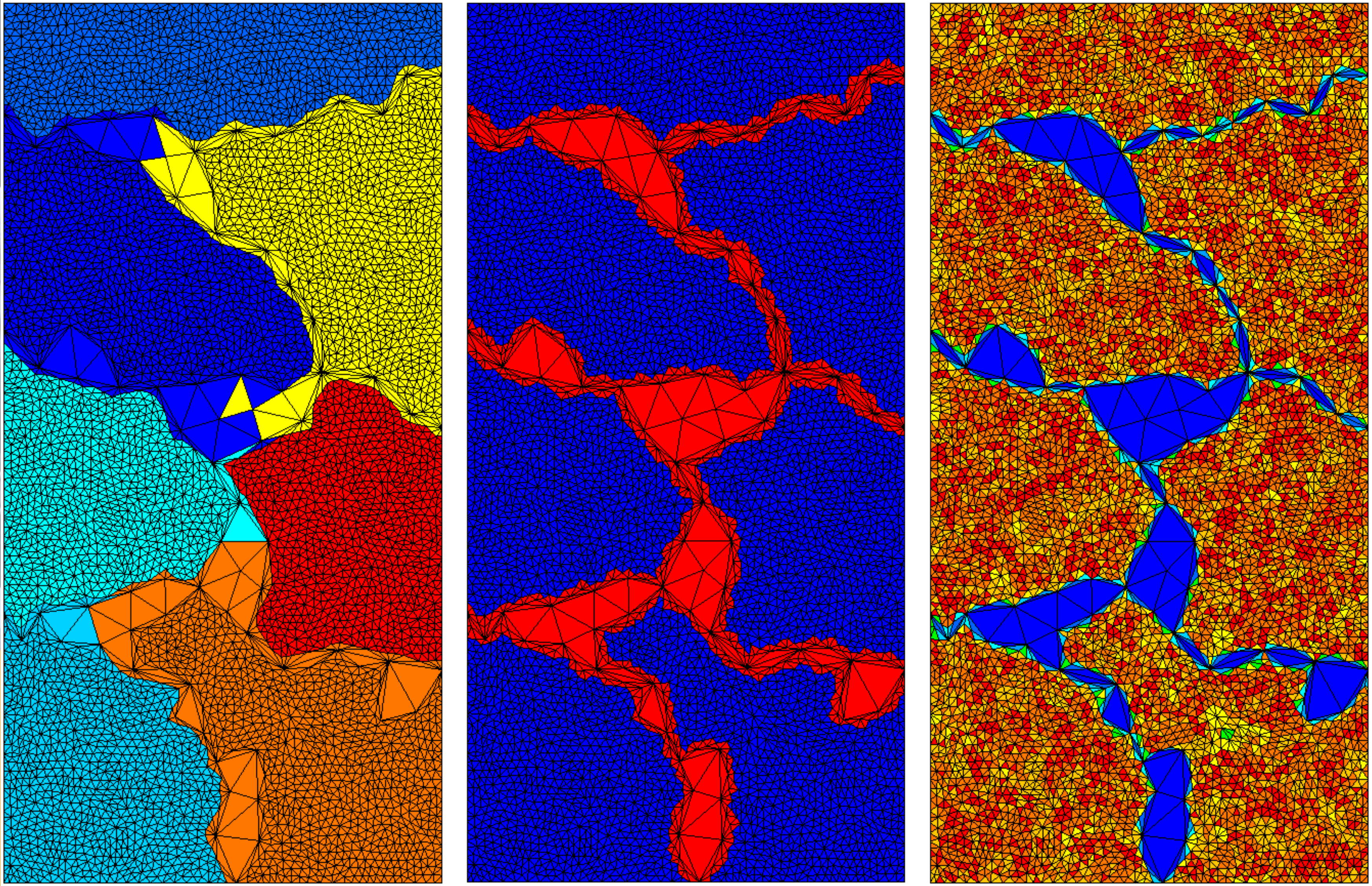
1st remeshing with blocked interfaces



Parallel strategy (III)

- Illustration: 2d case

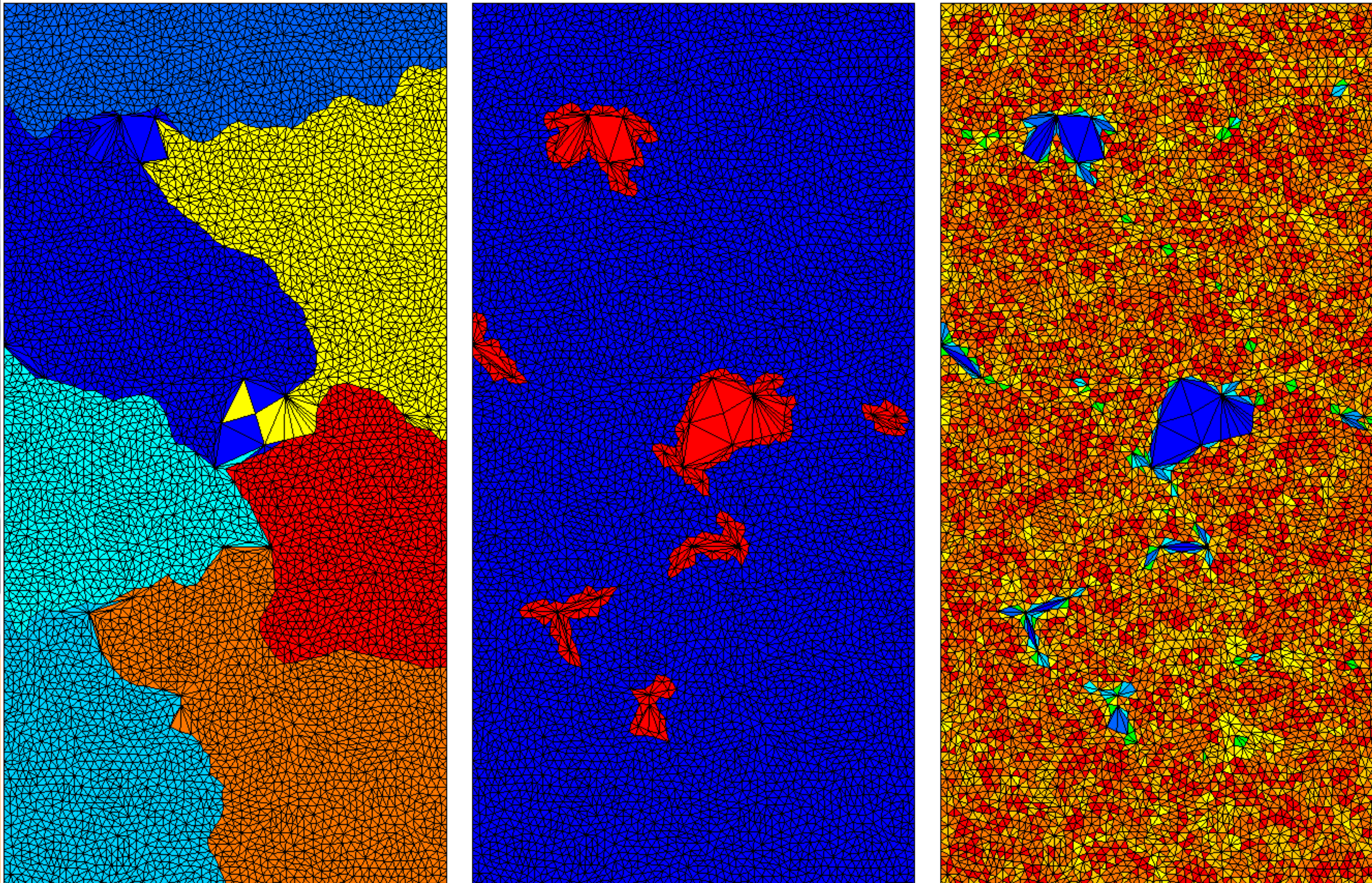
1st repartitioning to displace interfaces inside the subdomains



Parallel strategy (III)

- Illustration: 2d case

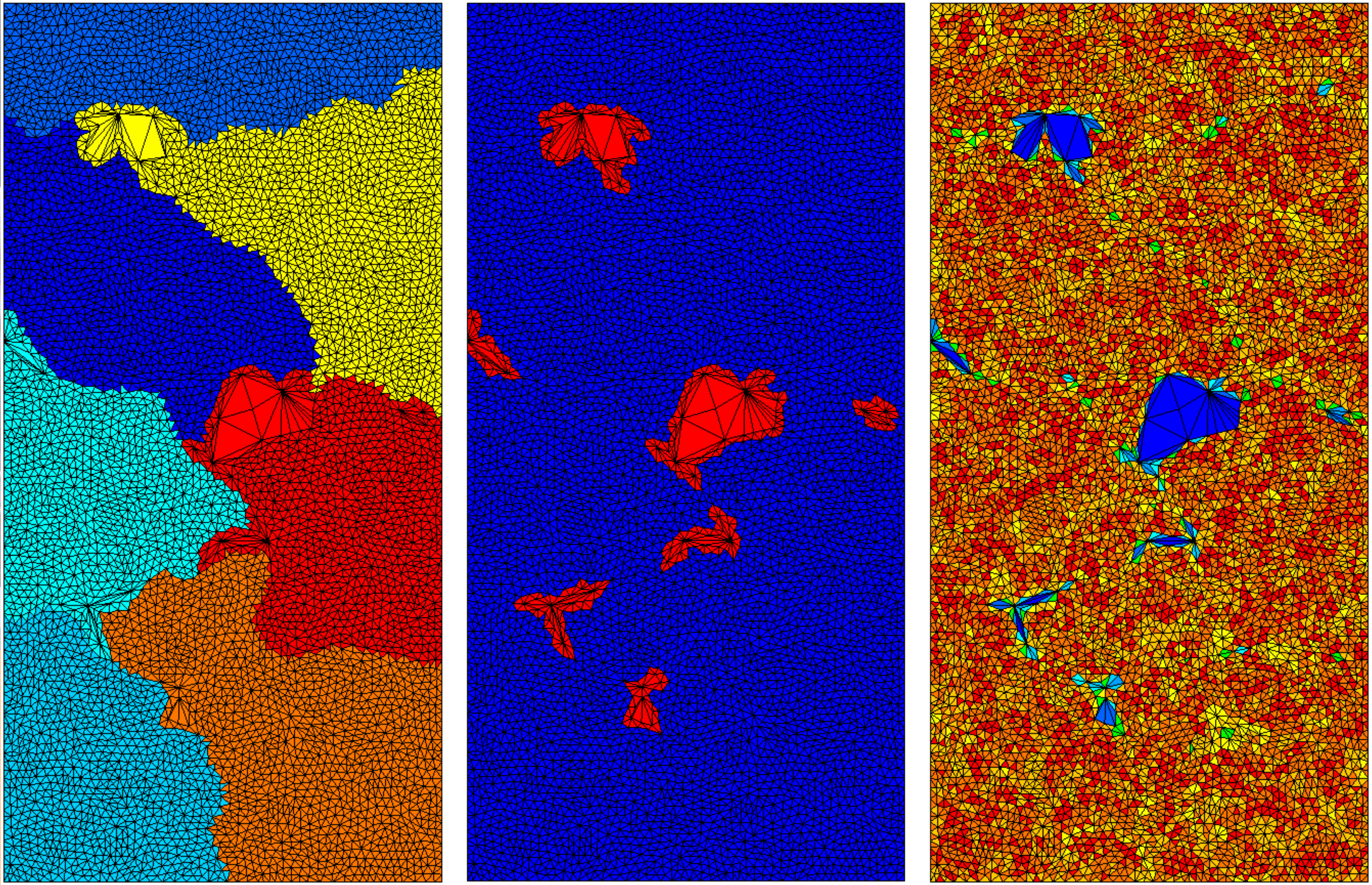
2nd remeshing



Parallel strategy (III)

- Illustration: 2d case

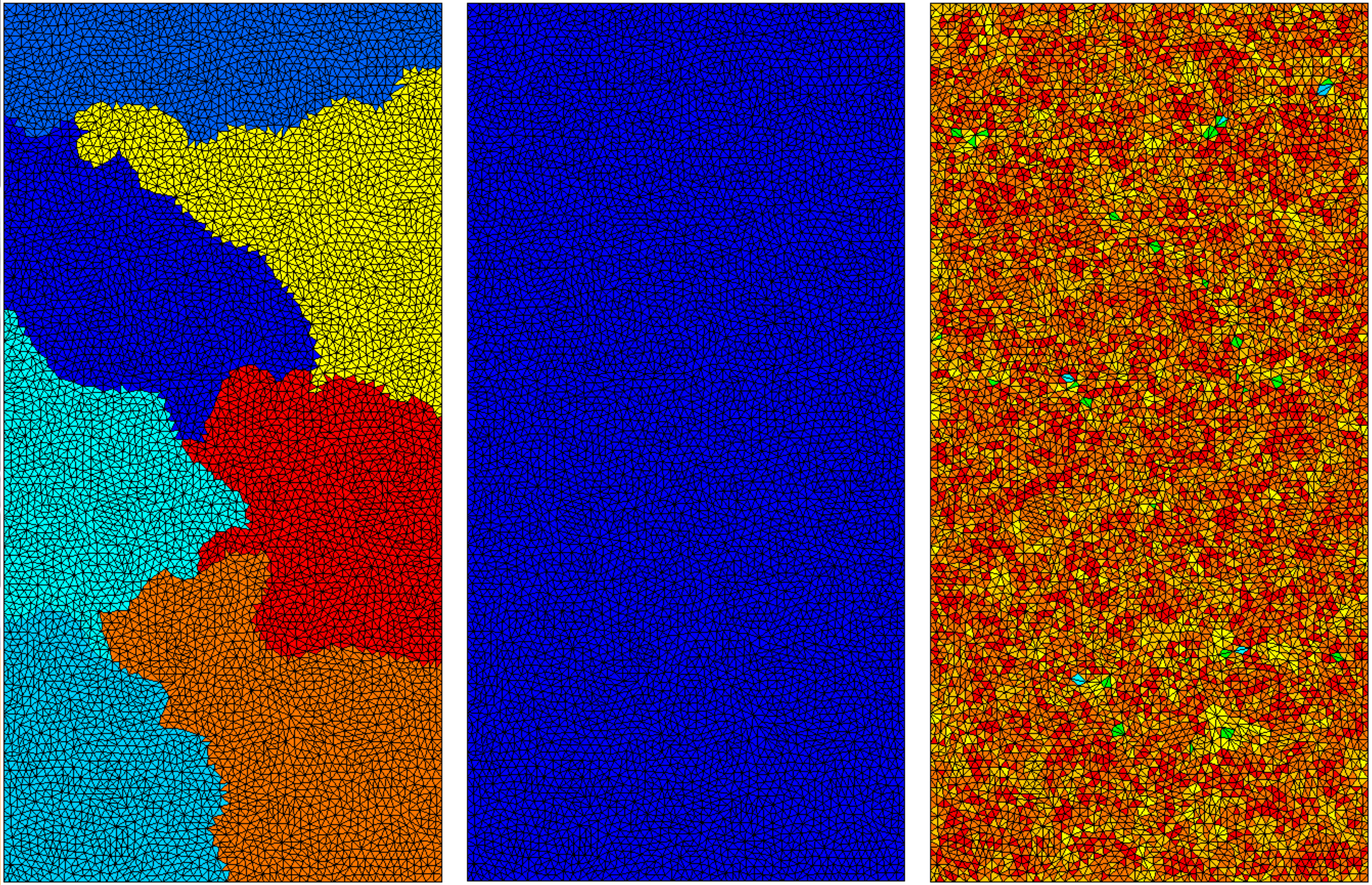
2^nd repartitioning



Parallel strategy (III)

- Illustration: 2d case

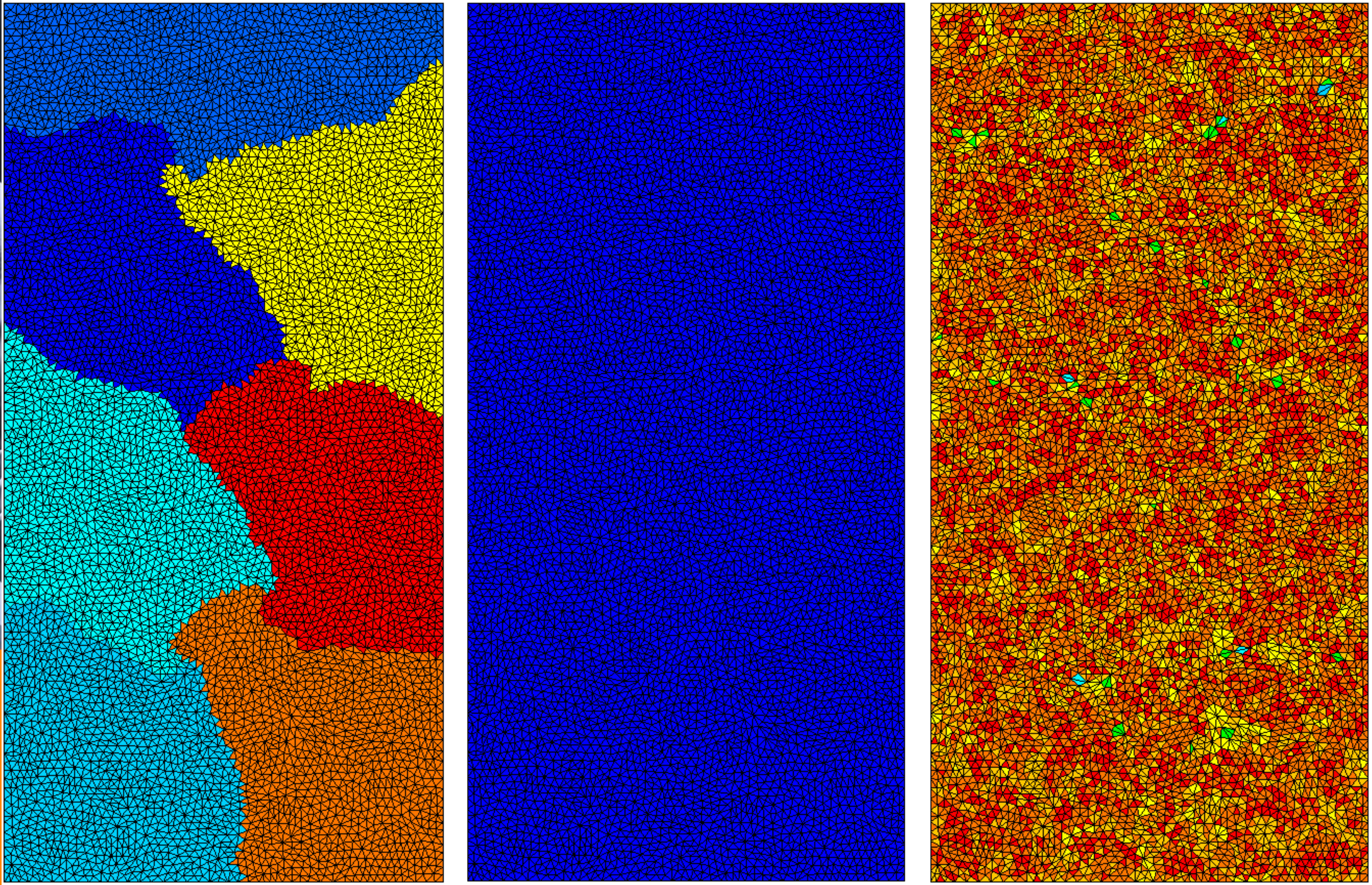
3rd remeshing



Parallel strategy (III)

- Illustration: 2d case

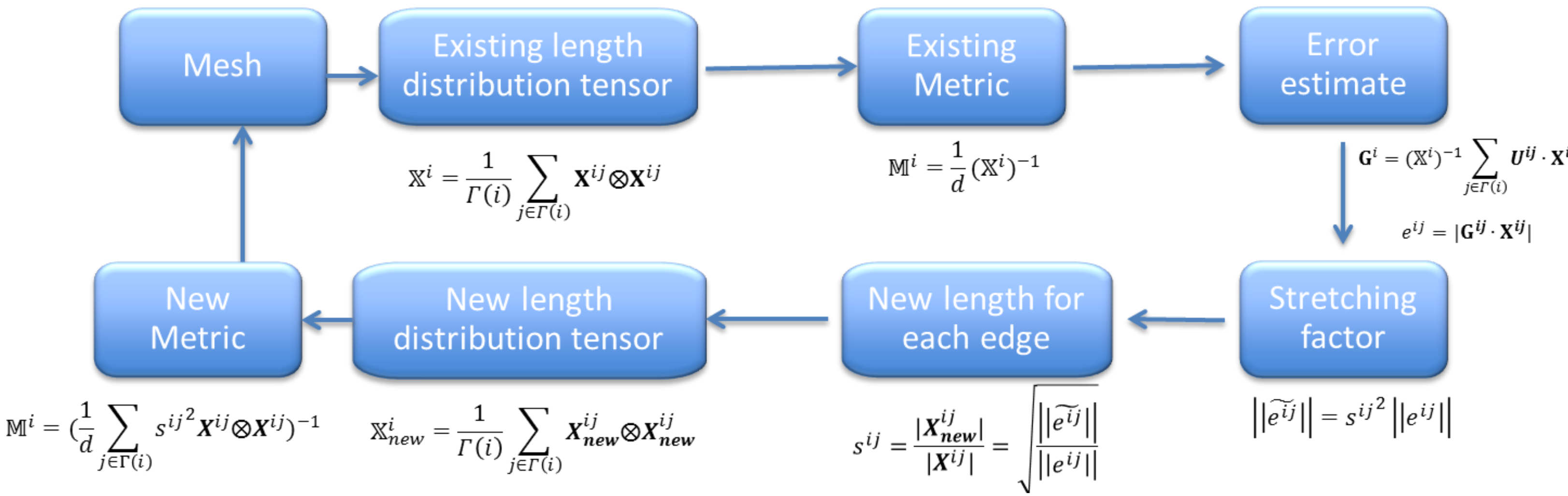
Last repartitioning for FE load balancing



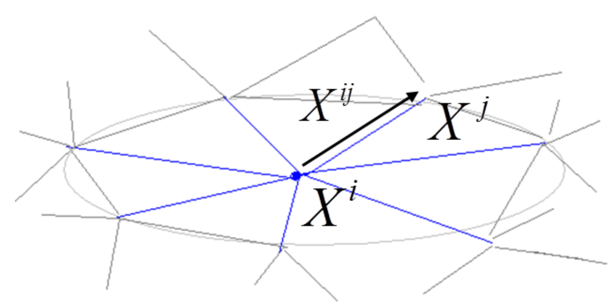
Anisotropic mesh adaptation (I)

A short summary

- On each core the mesher works using a metrics field
- Computation of the metrics field is based on an a posteriori error estimator directly computed at nodes and equi distribution of the error under the constraint of a imposed number of nodes



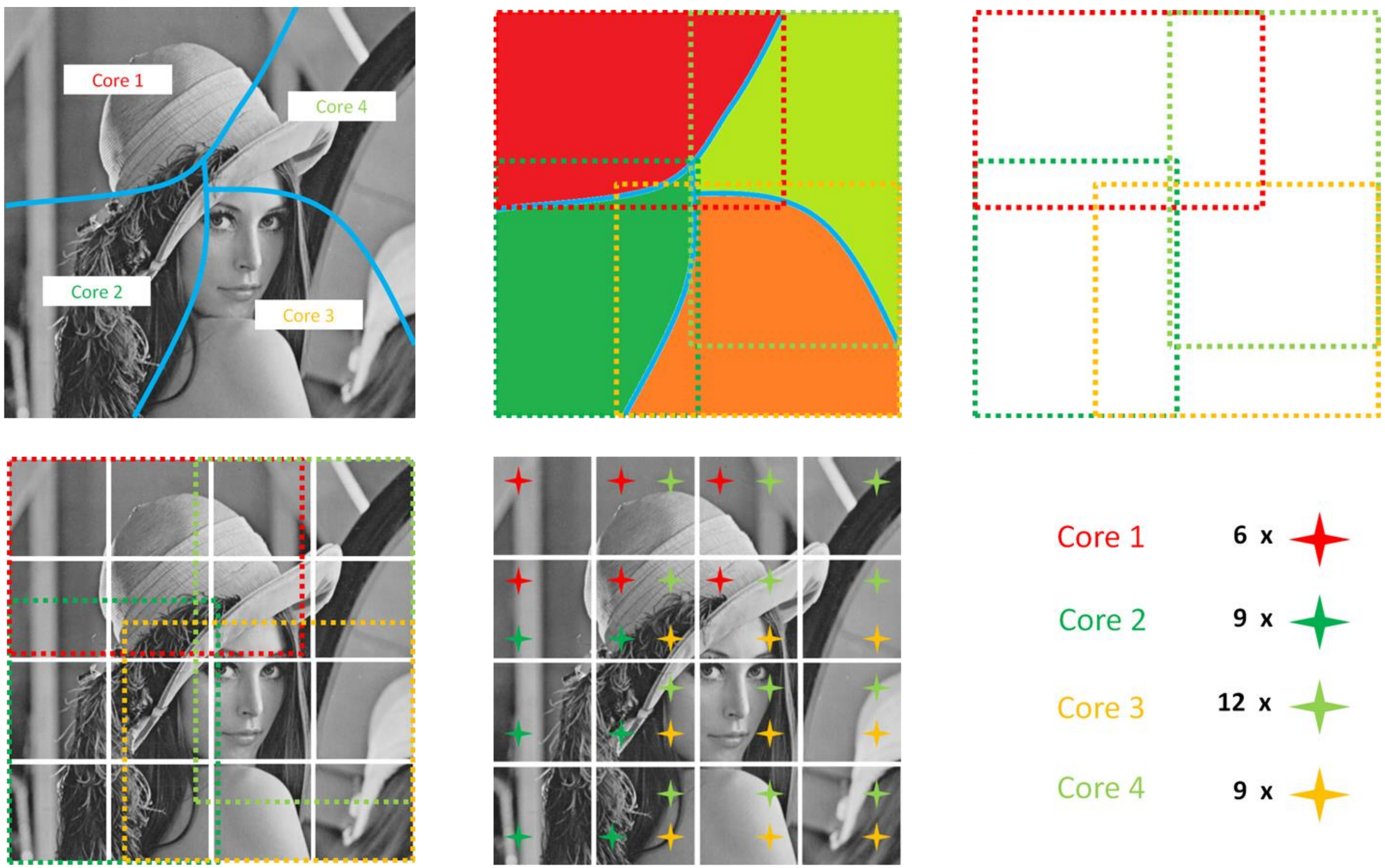
- Is extended to multi-solution fields



[Coupez, 2011], [Coupez, 2013]

Applications images 3D

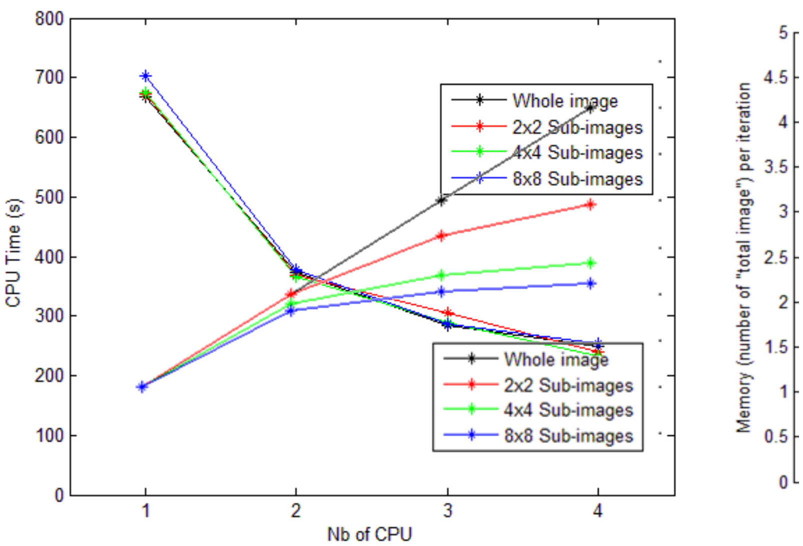
- Gros volumes spatio-temporels: performance de la méthode d'adaptation



Scalabilité

Adaptation de maillage

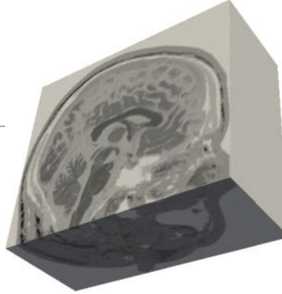
- Parallelisation interpolation avec découpe des images



Scalabilité

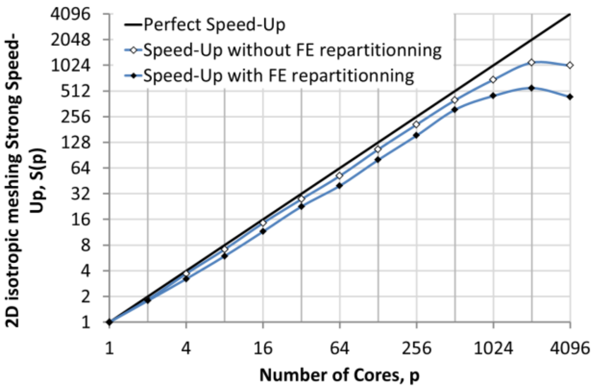
○ Adaptation de maillage

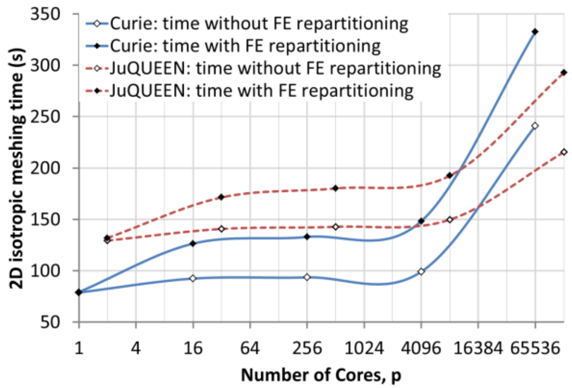
- Parallelisation interpolation avec découpe des images

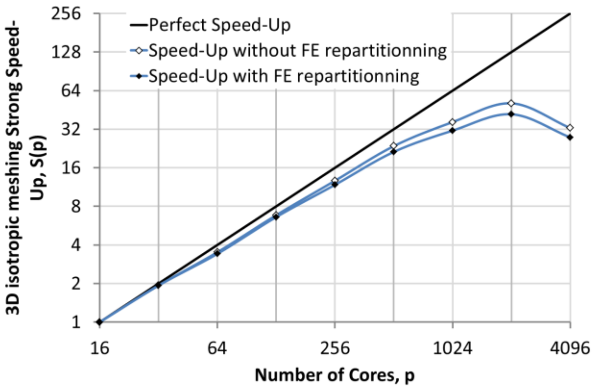


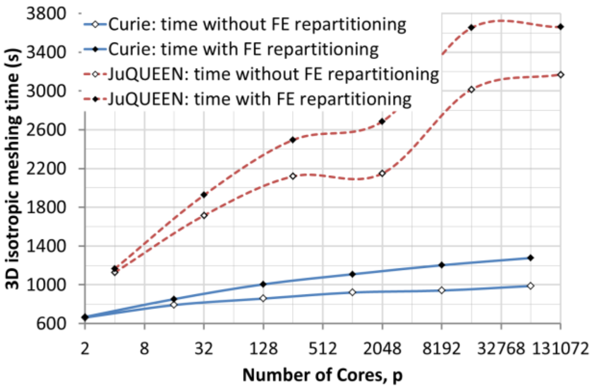
Hard Speed-Up: adaptation sur un maillage initial de 5 à final de 21 millions de noeuds (2D) et de 3.6 à 30 millions de nœuds (3D) , jusqu'à 4096 coeurs (Curie)

WeakSpeed-Up: 1 à 131 072 cœurs, avec une charge de travail constante par cœur (500 000 nœuds sur Curie et 250 000 sur JuQUEEN). Maillage final: 33,3 milliards de nœuds et 67 milliards d'éléments









Excellentes performances jusqu'à 1024 cœurs et dégradation au-delà, compte tenu de la charge par coeur par rapport au coût des communications

[Silva et al (IJCFD 2016)]

Segmentation par redistanciation

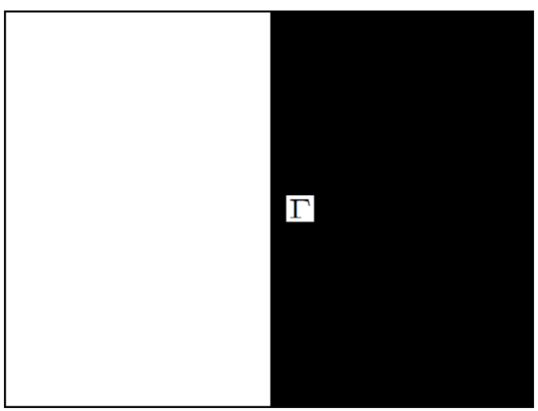
Segmentation par redistanciation

Problematique

- Sauf.. comment obtenir les fonctions de phase implicites?

Mesh adaptation → Greyscale image → Regularized function (Second order gradient)

Image (Greyscale) → Image processing → Segmented image (Binary)



Mesh adaptation → Binary image → Discontinuous function

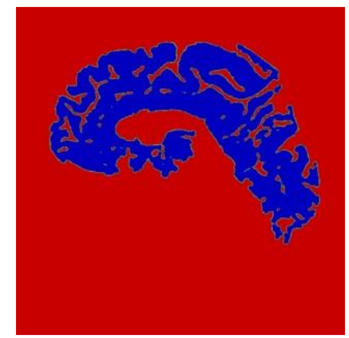
[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Segmentation par redistanciation

Problematique

- Sauf.. comment obtenir les fonctions de phase implicites?

Discontinuous function → Signed distance function u_d, Γ



- Distance signée $\alpha = \bar{d}(x,\Gamma) = \begin{cases} d(x,\Gamma) \text{ if } x \in \omega \\ -d(x,\Gamma) \text{ if } x \notin \omega \end{cases}$

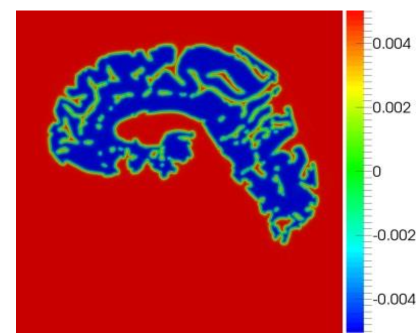
- Heaviside $H(\alpha) = 1_\omega(x) = \begin{cases} 1 \text{ if } \alpha > 0 \\ 0 \text{ if } \alpha < 0 \end{cases},$

Discontinuous function → Final redistanced function $u_\varepsilon{}^\tau(\varepsilon, S, \Delta\tau) = u_\varepsilon(\varepsilon, u_d)$

- Smoothing $\boxed{u_\varepsilon = u(\alpha,\varepsilon) = \varepsilon \tanh(\alpha/\varepsilon)}$

$$H_\varepsilon(u,\varepsilon) = \frac{1}{2}\left(1 + \frac{u_\varepsilon}{\varepsilon}\right)$$

u_ε



$\boxed{u_\epsilon(\alpha) \approx \alpha \text{ when } \alpha << \epsilon}$ et $\boxed{\lim_{\epsilon \to 0} H_\epsilon \longrightarrow H}$

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Segmentation par redistanciation

- Resolution d'une equation type Hamilton-Jacobi: réinitialisation convective

- Connaissant seulement le signe (qui peut venir d'une segmentation "grossière"), on peut reconstruire u_ε directement par resolution d'une équation type Hamilton-Jacobi

$$\begin{cases} \dfrac{\partial u}{\partial \tau} = s(u)(|\nabla u| - (1 - (\dfrac{u}{\epsilon})^2)) \\ u(\tau = 0) = 0 \text{ on } \Gamma \end{cases}$$ parce que $u'_\epsilon(\alpha) = 1 - (\frac{u_\epsilon}{\epsilon})^2$

- Vitesse de réinitialisation $v_r = S(\tilde{u}) \dfrac{\nabla u_\varepsilon}{||\nabla u_\varepsilon||_2}$

- Equation de réinitialisation convective $\dfrac{\partial u_\varepsilon}{\partial \tau} + v_r \cdot \nabla u_\varepsilon = S(\tilde{u}) g(u_\varepsilon)$

- Résolution par éléments finis stabilisés

- Etape de correction

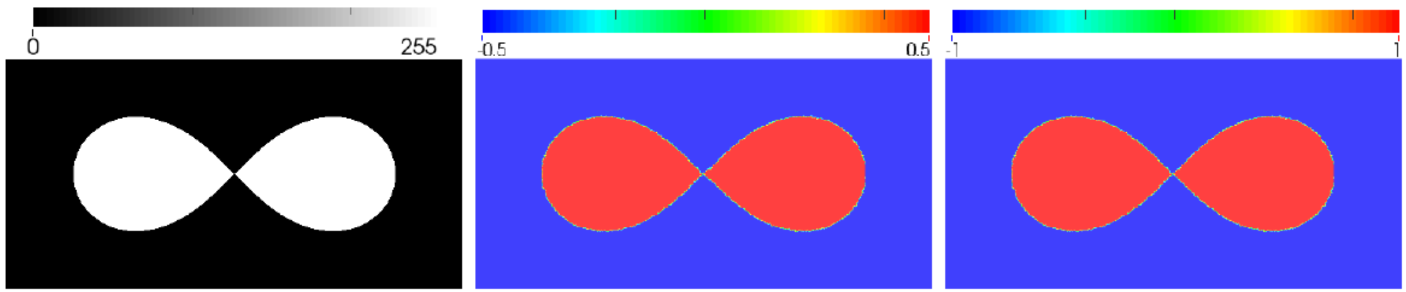
$$S(\mathbf{X}^i) = \begin{cases} 1, if\ \hat{u}(Pixel^k / Voxel^k) > threshold\ value \\ 0, if\ \hat{u}(Pixel^k / Voxel^k) = threshold\ value \\ -1, if\ \hat{u}(Pixel^k / Voxel^k) < threshold\ value \end{cases}$$

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

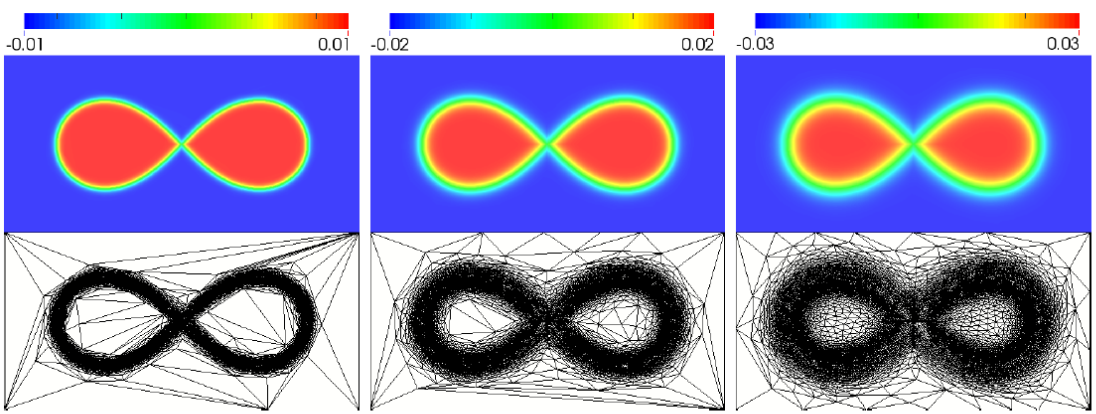
Segmentation par redistanciation

Exemple et validation

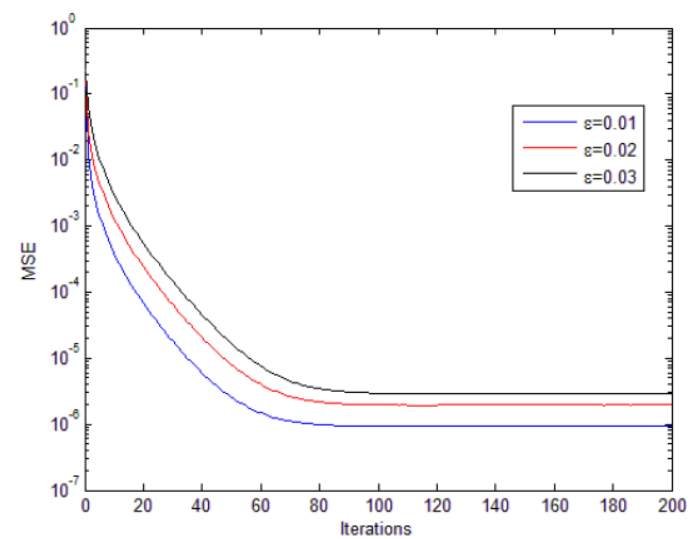
- Application avec une image construite



(a) $\widehat{u}$ (b) $u^0_\varepsilon(S, \varepsilon, \tau = 0) = u_h$ (c) S_h



(a) $\varepsilon = 0.01$ (b) $\varepsilon = 0.02$ (c) $\varepsilon = 0.03$



[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

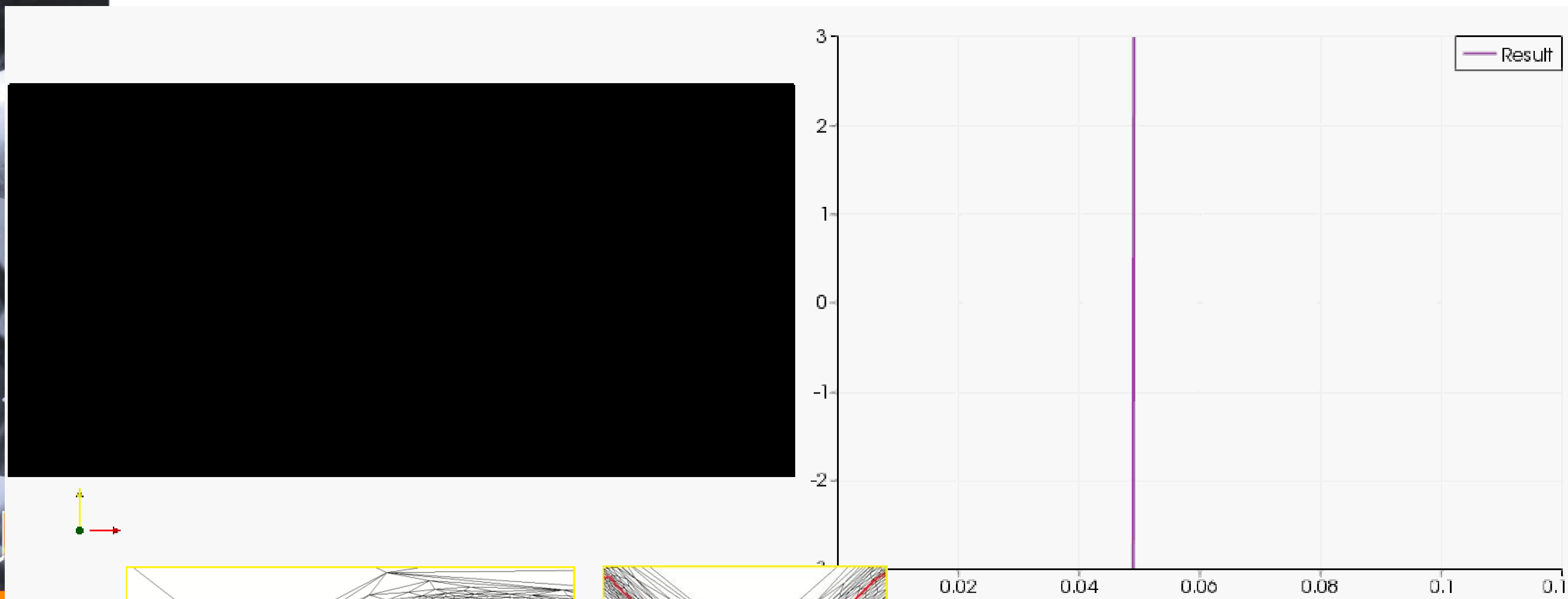
Segmentation par redistanciation

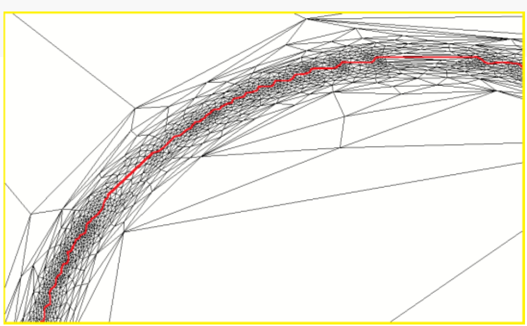
Exemple et validation

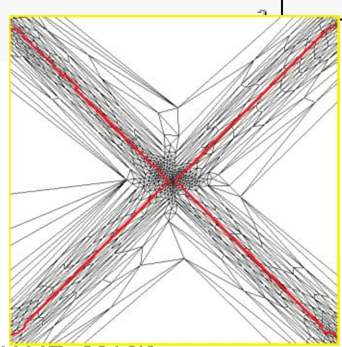
- Application avec une image construite

Mesh adaptation, $N = 10000$

$\frac{{u_\varepsilon}^\tau}{\varepsilon}$ with $\varepsilon = 0.01$







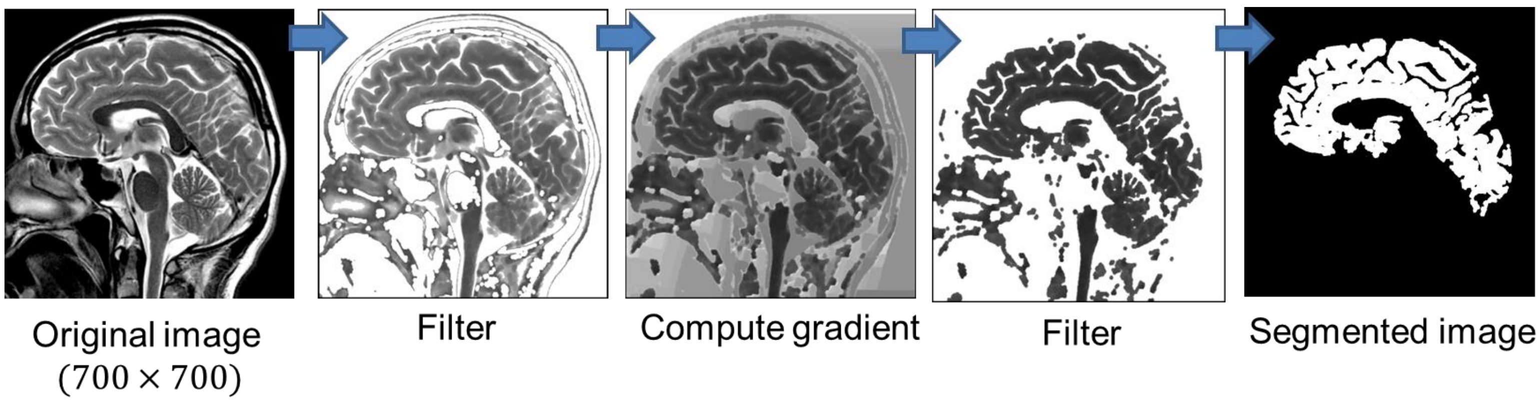
1. ε >pixel size
2. Mesh size<pixel size

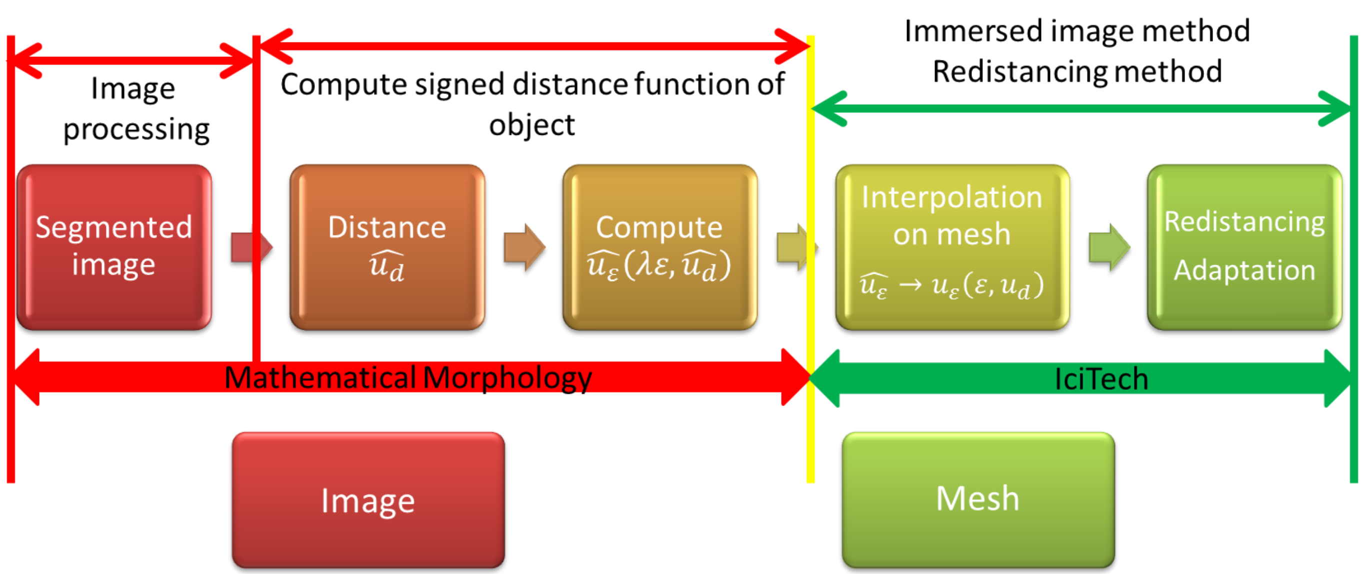
[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Segmentation par redistanciation

Problematique

- Et si on travaillait sur l'image?
  - Mathematical morphology tools (Morph-M) have been interfaced to accelerate distancing



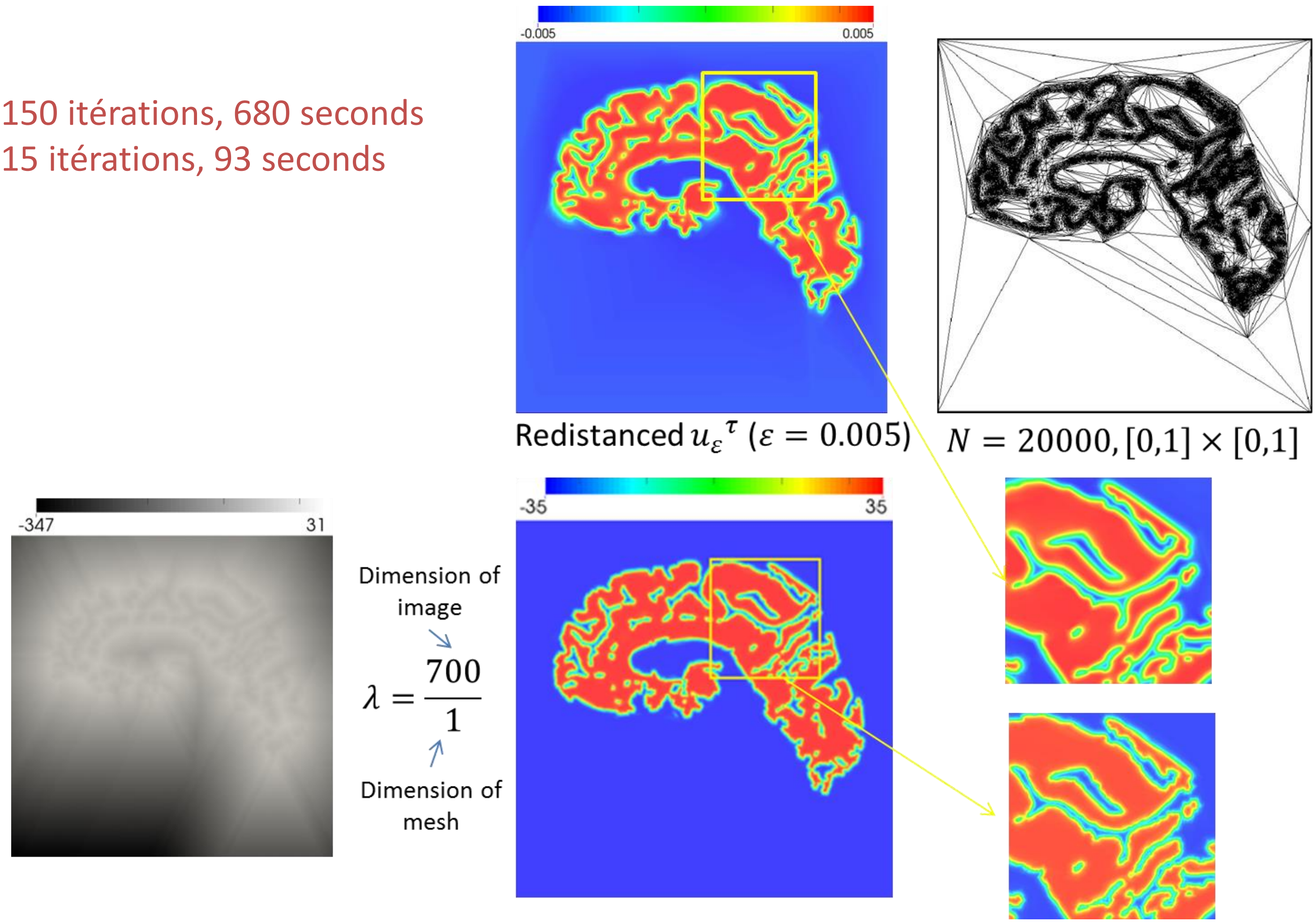


Segmentation par redistanciation

Problematique

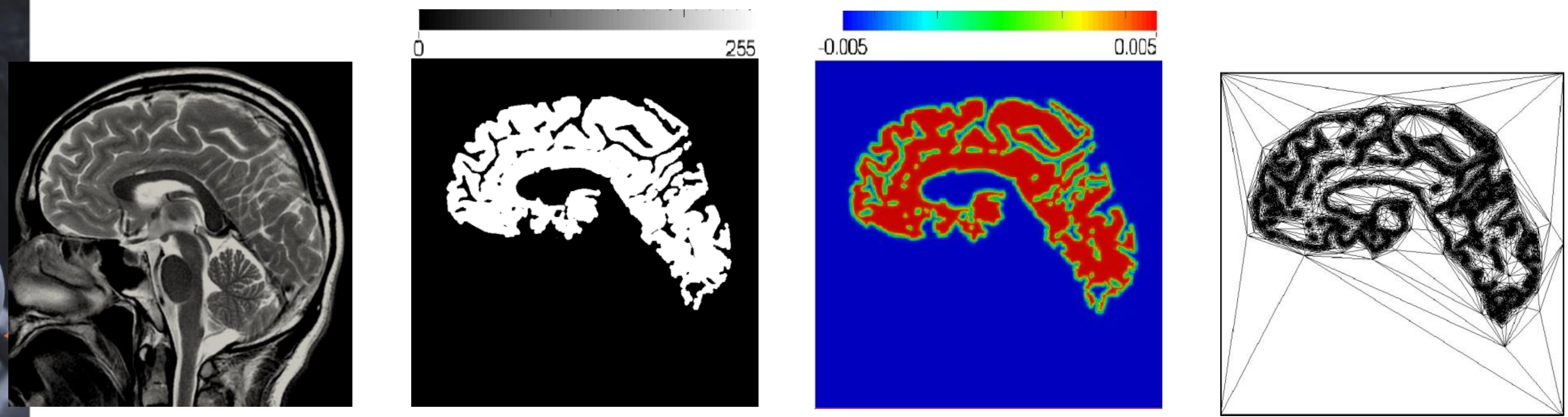
- Et si on travaillait sur l'image?
  - Mathematical morphology tools (Morph-M) have been interfaced to accelerate distancing

Avant: 150 itérations, 680 seconds
Après: 15 itérations, 93 seconds

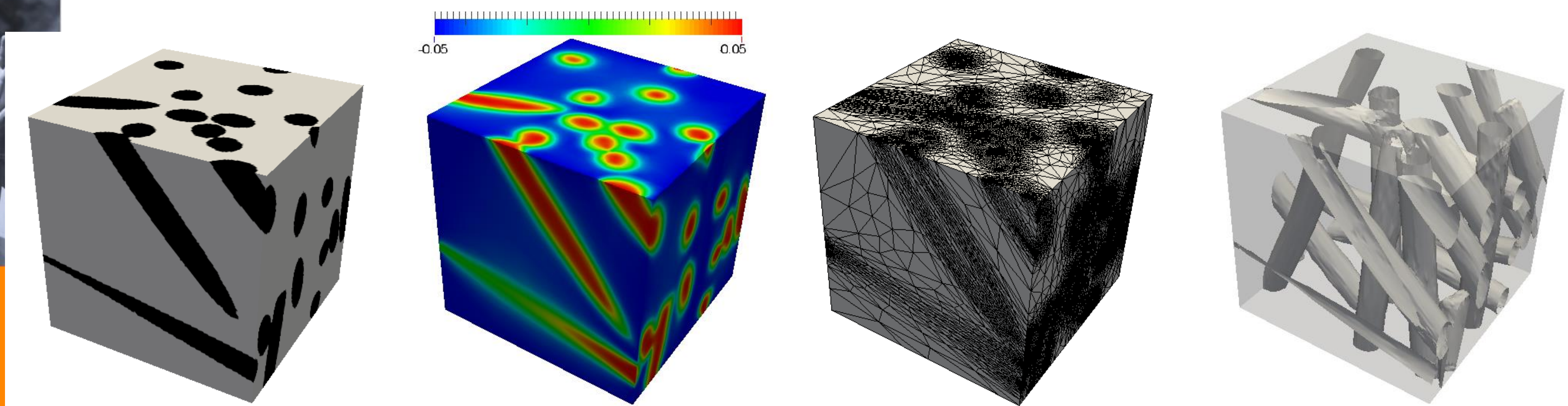


Segmentation par redistanciation

- Exemples en 2D et 3D



$\tilde{u}$ $\quad$ $\tilde{u}_{seg}$ $\quad$ $u_\varepsilon(\varepsilon = 0.005)$ $\quad$ $N = 20000$

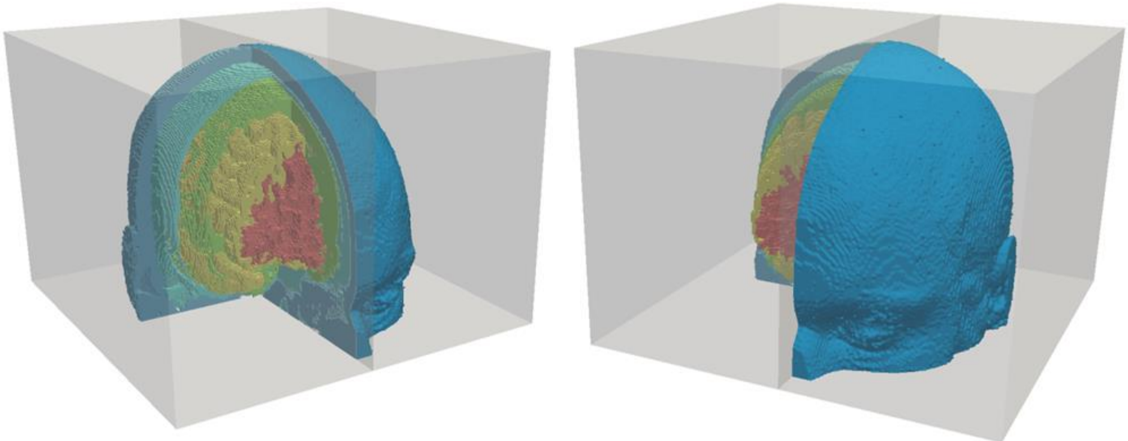


$\tilde{u}(200 \times 200 \times 200)$ $\quad$ $u_\varepsilon(\varepsilon = 0.05)$ $\quad$ $N = 200000$ $\quad$ Iso-surface $u_\varepsilon = 0$

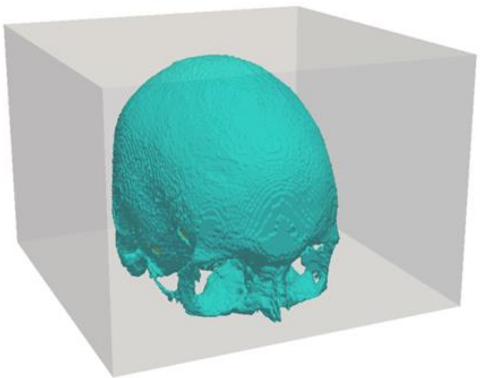
[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Segmentation par redistanciation

- N phases

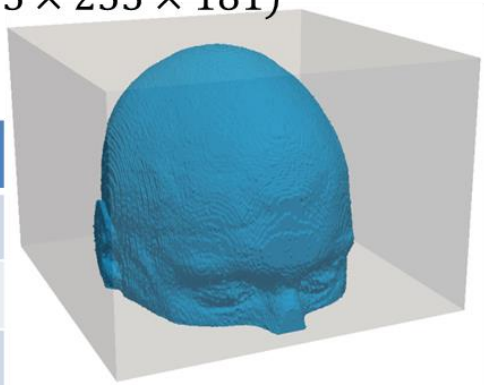


Segmented image $\widehat{u_{seg}}$ $(255 \times 255 \times 181)$

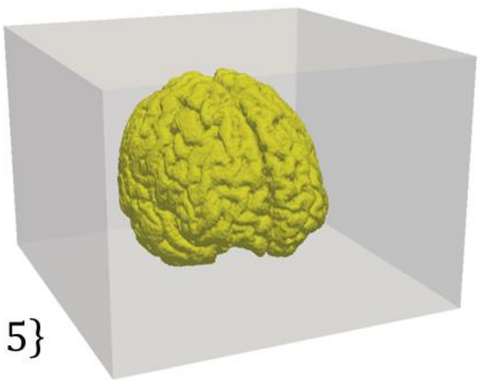


$\widehat{u_{\text{skull}}} = \{2\}$

Value of $\widehat{u_{seg}}$	
0	Outer
1	Skin,eye,etc
2	Skull
3	CSF (cerebrospinal)
4	Grey matter
5	White matter



$\widehat{u_{head}} = \{1 + 2 + 3 + 4 + 5\}$

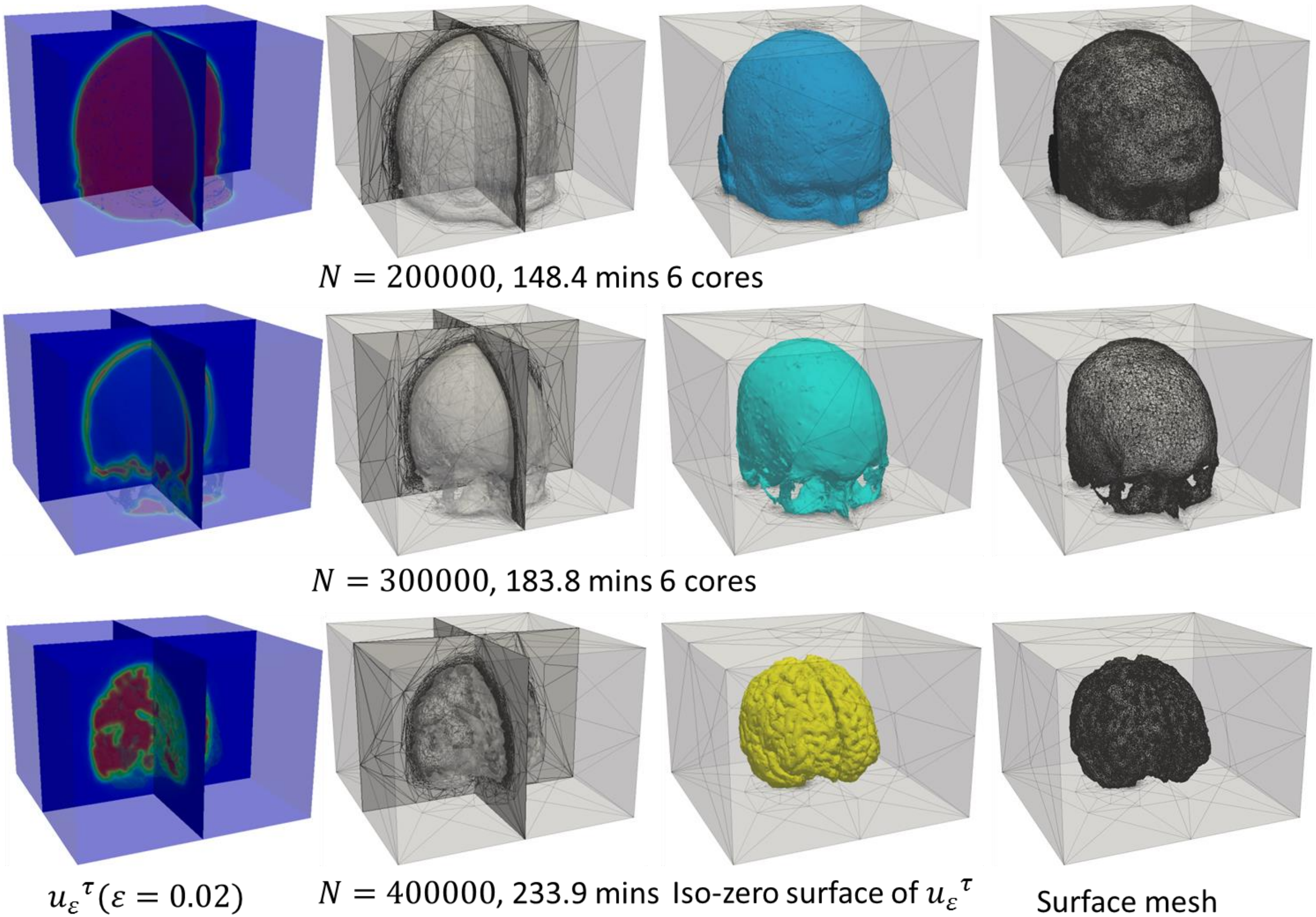


$\widehat{u_{\text{brain}}} = \{4 + 5\}$

[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Segmentation par redistanciation

- N phases



[Silva, Zhao et al (2014)] [Zhao , Silva et al (CMAME, 2016)]

Les écoulements

Méthodes immergées

Les équations de Navier-Stokes en multiphasique

- Equations de Navier-Stokes incompressible

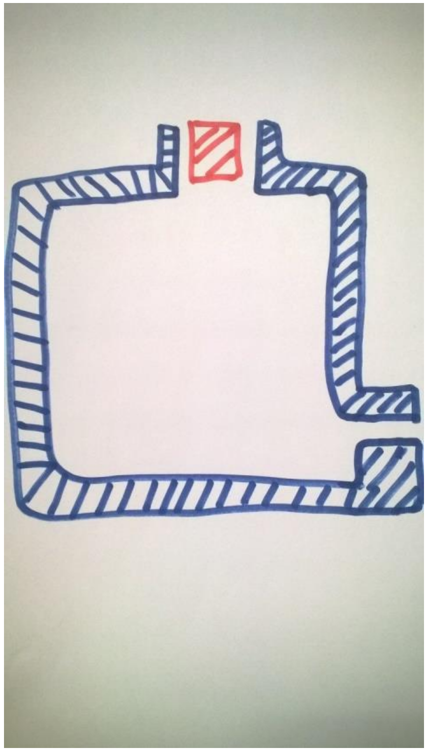
$$\begin{cases} \rho \dfrac{dv}{dt} - \nabla \cdot (2\eta\varepsilon(v) + S) + \nabla p = \rho g + F \\ \\ \nabla \cdot v = 0 \end{cases}$$

- Lois de mélange régularisées $\xi = \xi_\epsilon = \xi_\omega H_\epsilon + \xi_{\Omega\backslash\omega}(1 - H_\epsilon)$

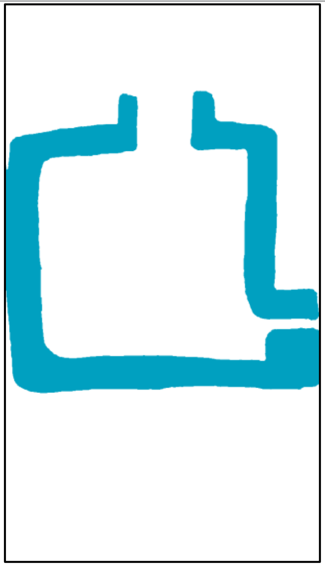
- Résolution numérique : en //
  - Méthodes éléments finis stabilisés
  - Newton pour la linéarisation
  - Librairie PETSc (Krylov + multigrille) > résolution massivement parallèle
  - Conditions limites par pénalisation + Lagrangian augmenté (calcul de la force)

[Coupez et al, JCP 2013]

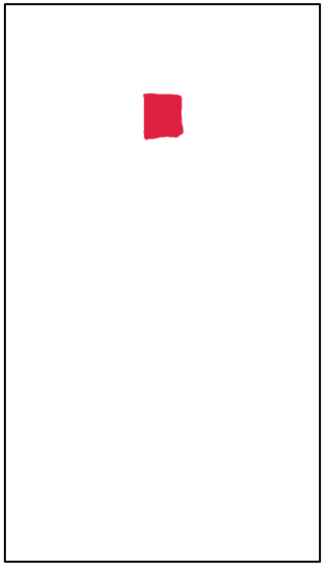
Les écoulements



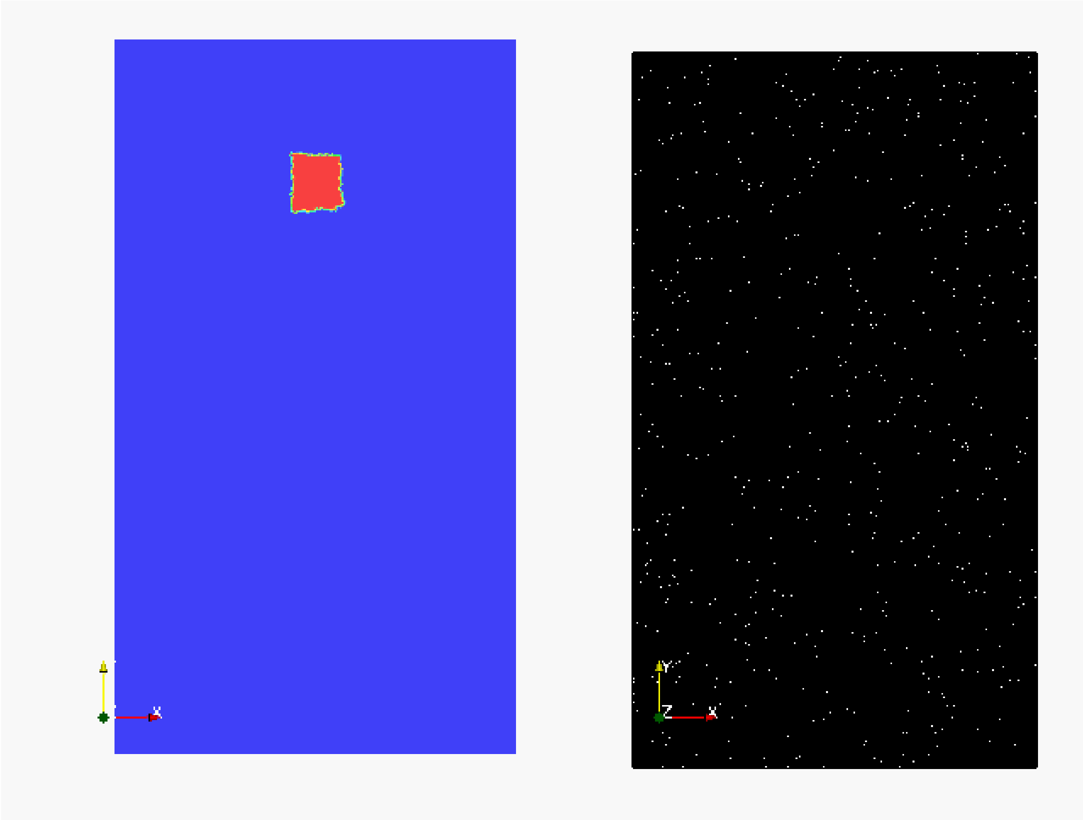
Photography of a drawing



Rigid part

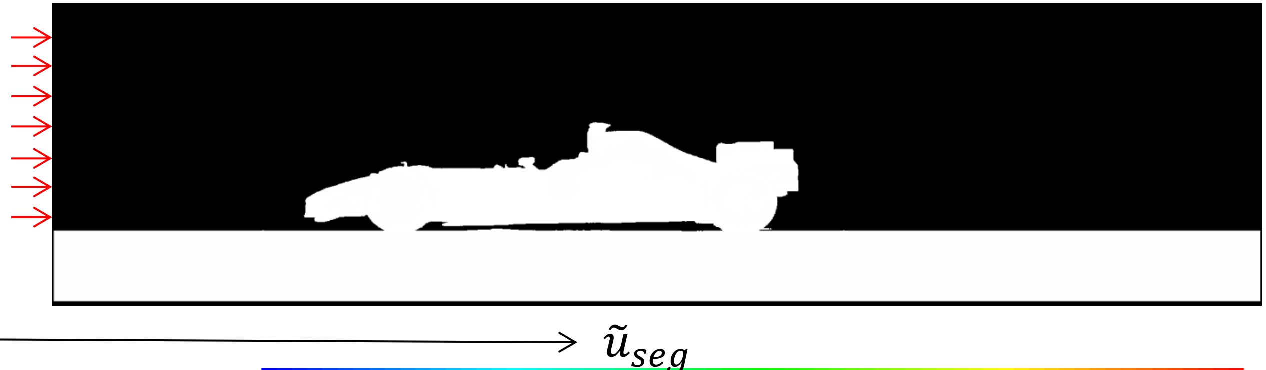


Flow inlet



$\varepsilon = 0.01, \Delta\tau = 0.00025, N = 10000$

1 core run on 185 minutes, with
$\Delta t = 0.00025$ and 1500 increments



$\tilde{u}$
$\tilde{u}_{seg}$

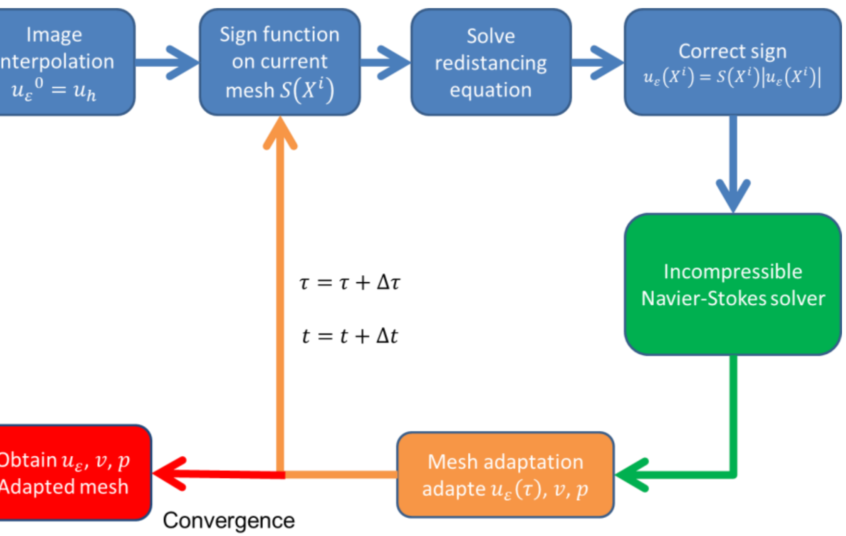
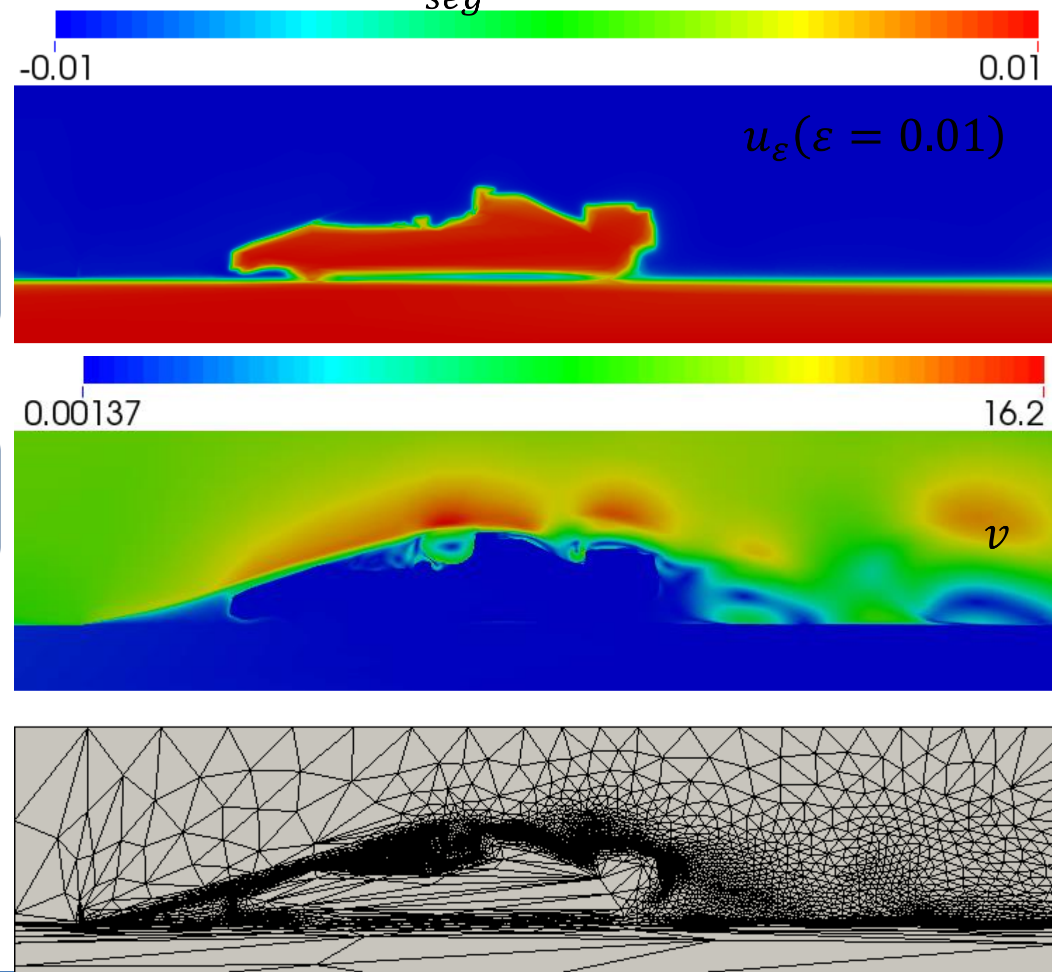


Image interpolation $u_\varepsilon{}^0 = u_h$
Sign function on current mesh $S(X^i)$
Solve redistancing equation
Correct sign $u_\varepsilon(X^i) = S(X^i)|u_\varepsilon(X^i)|$
Incompressible Navier-Stokes solver
Mesh adaptation adapte $u_\varepsilon(\tau)$, v, p
$\tau = \tau + \Delta\tau$
$t = t + \Delta t$
Obtain u_ε, v, p Adapted mesh
Convergence



-0.01
0.01
$u_\varepsilon(\varepsilon = 0.01)$
0.00137
16.2
v

Scalabilité

○ Méthode multigrille //

- Parallélisation de la résolution
  - préconditionneur de type multigrille PCMG
  - optimisation du calcul des opérateurs de restriction/interpolation

WeakSpeed-Up: Résolution des équations de Stokes sur maillage adapté, exécution de 1 à 262 144 cœurs pour une convergence relative à 1e-9, cas final à 100 milliards de degrés de libertés en 3D (21 milliards en 2D)

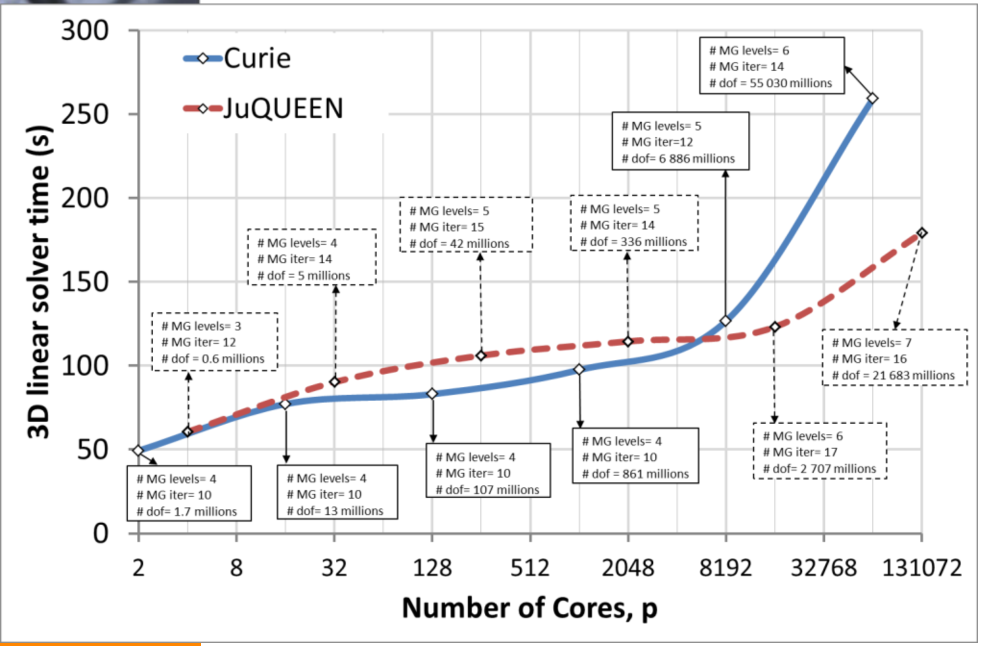
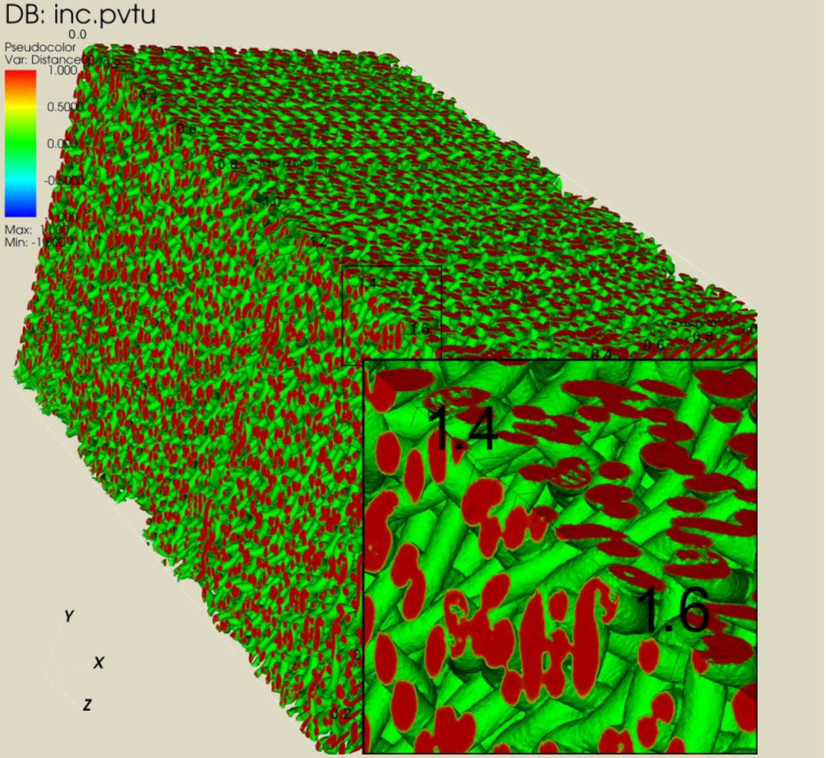


Image: 1200x1200x1791 voxels (3SR)
Maillage: 400 millions de nœuds sur 4096 coeurs
Génération/adaptation: 5h
Ecoulement (calcul multiphasique): 10 min pour 1.6 millards de degrés de liberté

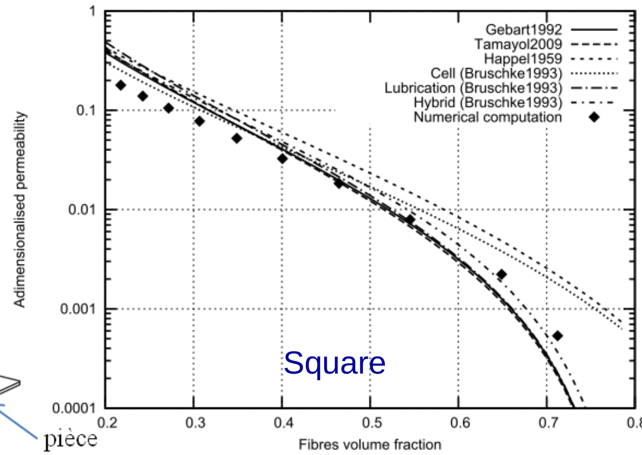


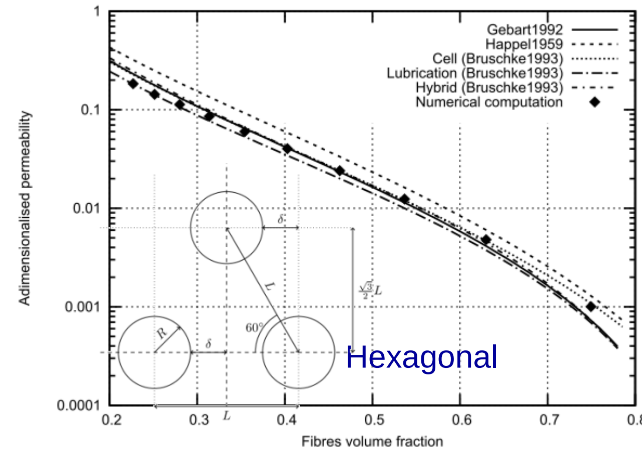
[Silva et al (IJCFD, 2016)]

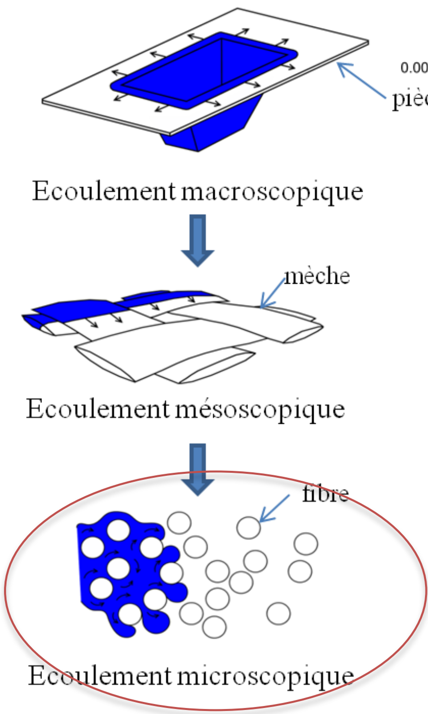
Les écoulements dans les composites

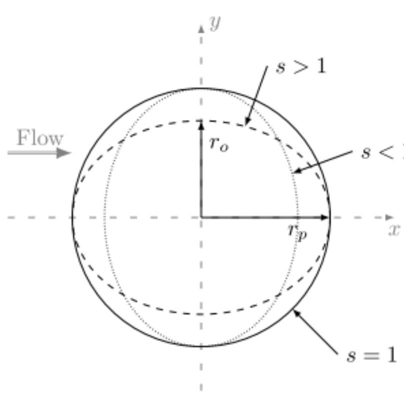
Comparaison avec des solutions de réference

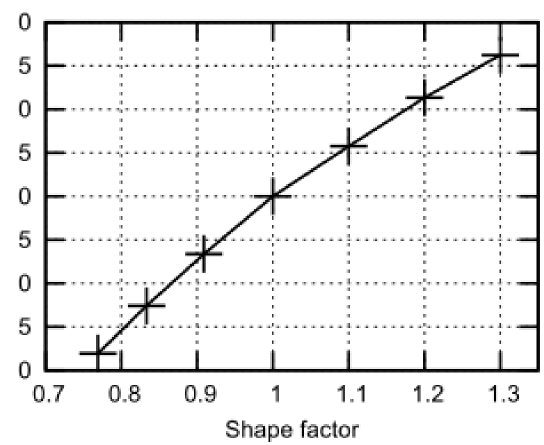
- Echelle microscopique: Perméabilité fonction du type d'arrangement, régularité, conditions aux limites, taille du VER…





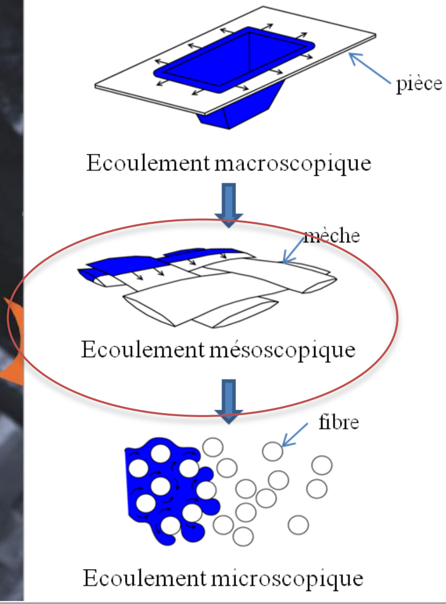


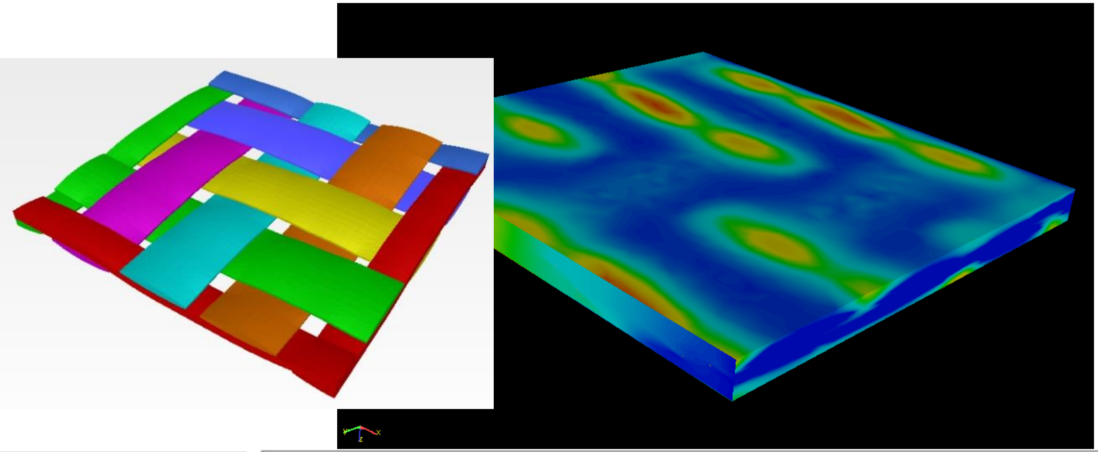


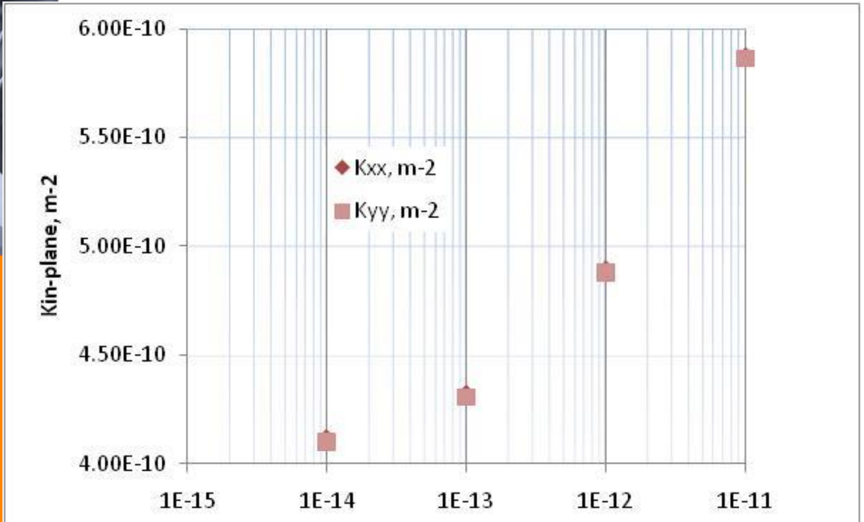


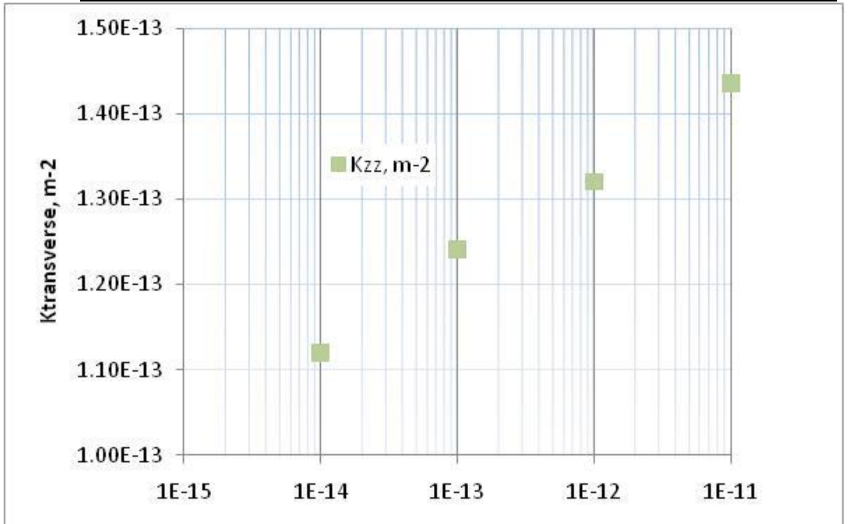
Comparaison avec des solutions de réference

- Echelle mesoscopique: Perméabilité fonction du type d’architecture, dans le plan et transverse, nesting…









Kxx ~ Kyy ~ 10^{E-10}

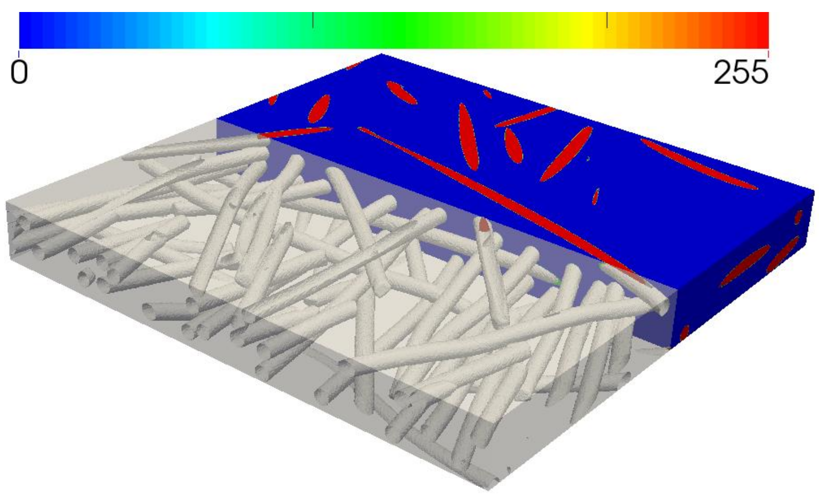
Kzz ~ 10^{E-13}

Résultats 3D: perméabilité « intrinsèque »

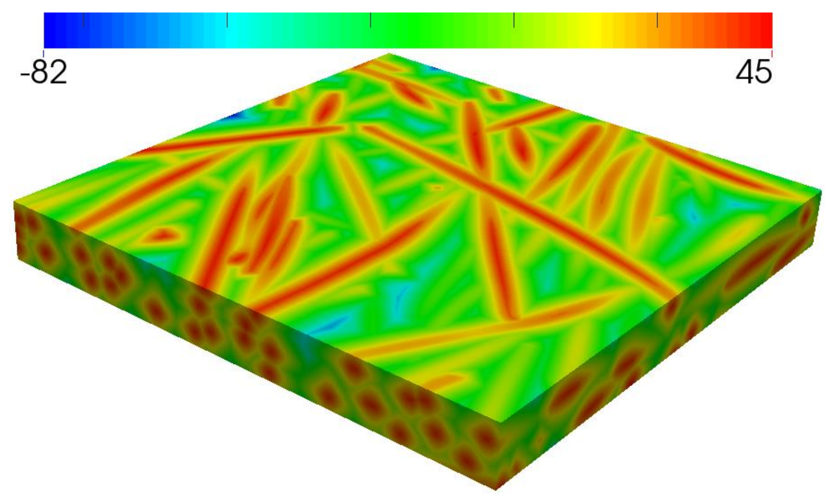
Arrangement irrégulier 3D: éléments finis

- Arrangement 3D irrégulier obtenu par imagerie 3D *[Orgéas et al, 3SR]*
- Caractéristiques renfort: R=0.1mm, L=10mm, F=0.83
- Images acquises par microtomographie-X, taille du voxel = 10x10x10 μm^3
- Image reconstruite, segmenté et binarisé, correspondante à

900x900x100 et 900x900x220



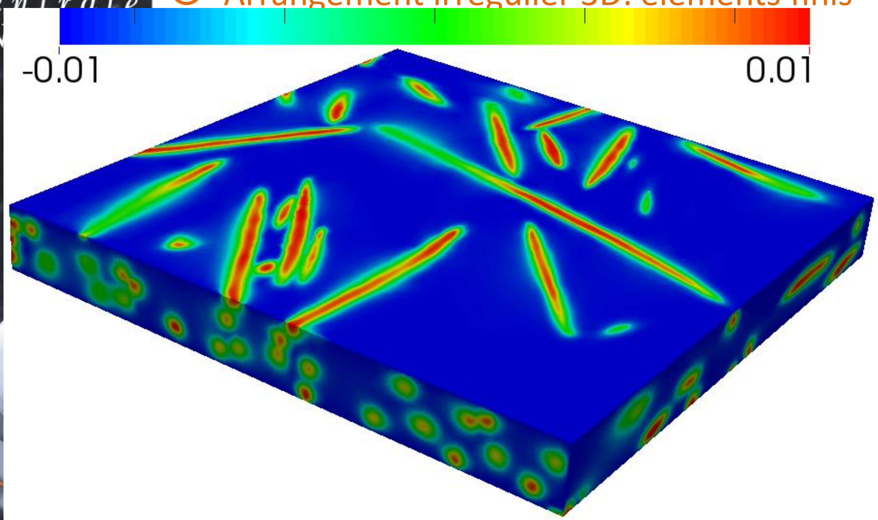
Distance signée $\widehat{u_d}$ dans l'image



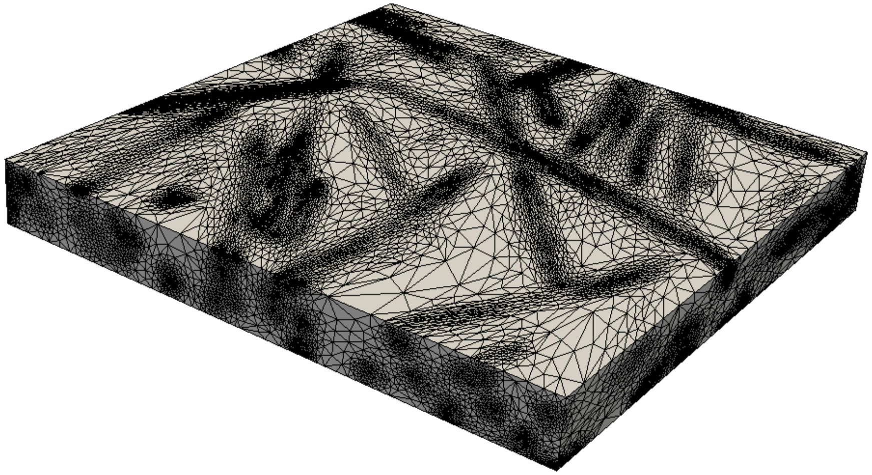
- Calcul d'écoulement sur le volume

Résultats 3D: perméabilité « intrinsèque »

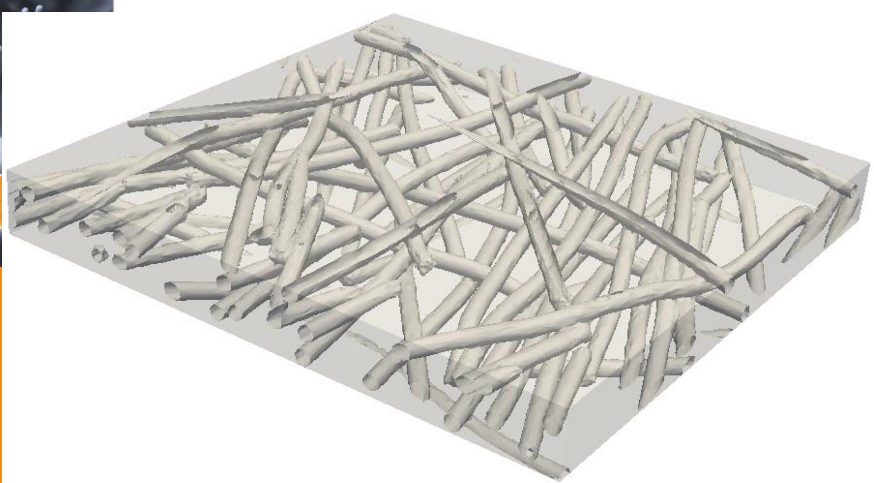
- Arrangement irrégulier 3D: éléments finis



Redistanced ${u_\varepsilon}^\tau(\varepsilon = 0.01)$



Adapted volume mesh, $N = 400000$



Iso-zero surface of ${u_\varepsilon}^\tau$

Surface mesh of the Iso-zero surface of ${u_\varepsilon}^\tau$

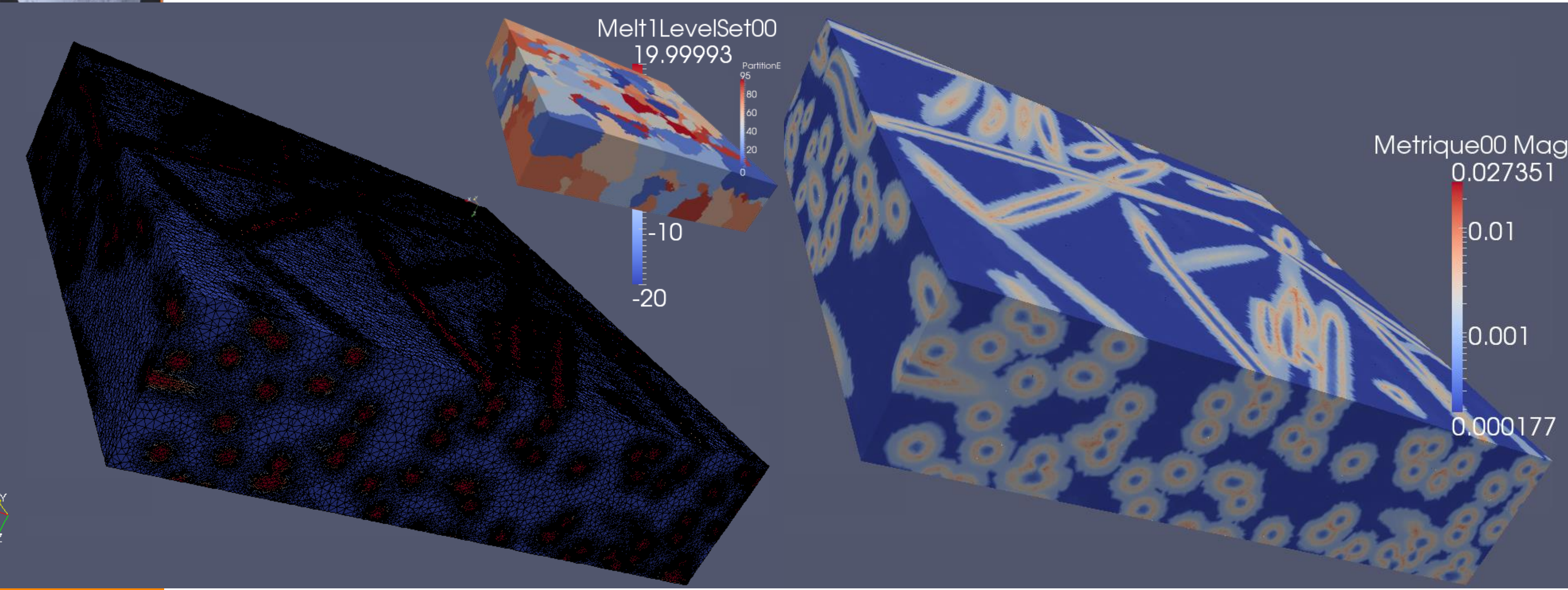
6 cores for 25 iterations, 213.8 minutes

[Silva et al (IJCFD 2016)]

Résultats 3D: perméabilité « intrinsèque »

Arrangement irrégulier 3D: éléments finis

- Génération du maillage éléments finis: interpolation + adaptation, comme précédemment (sur **96 cœurs**, en //)
- Taille du maillage sur le gros volume: ~**5 millions de nœuds** (par rapport à **178.2 millions de voxels, sur le 900x900x220 s**)



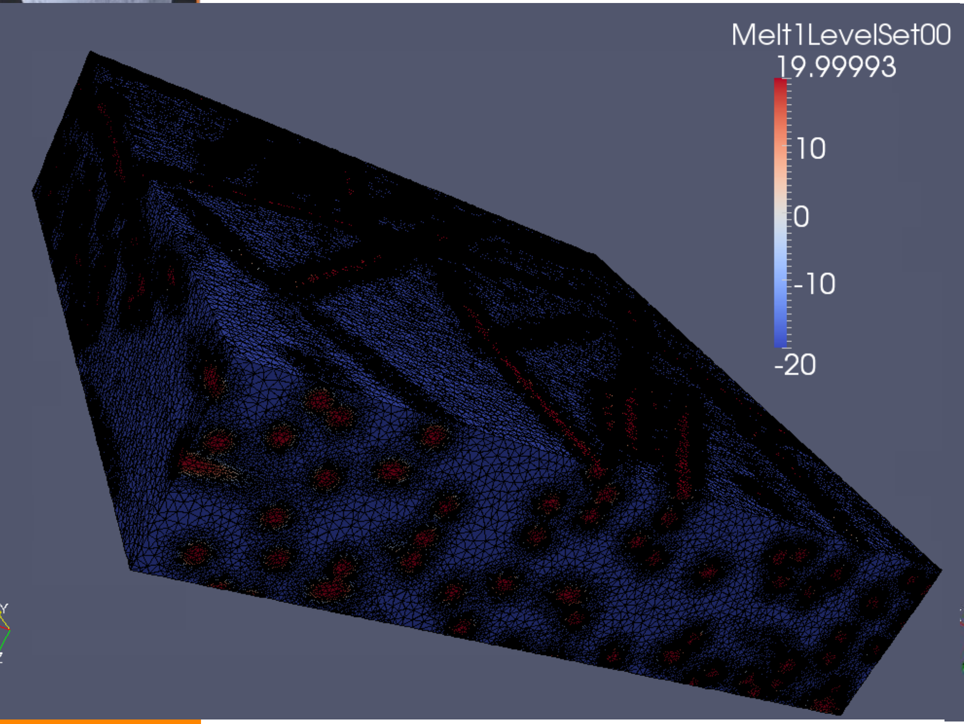
Maillage adapté ***Norme de la métrique calculée pour l'adaptation***

[Digonnet , Silva et al (soumis à IJHPCA 2015)]

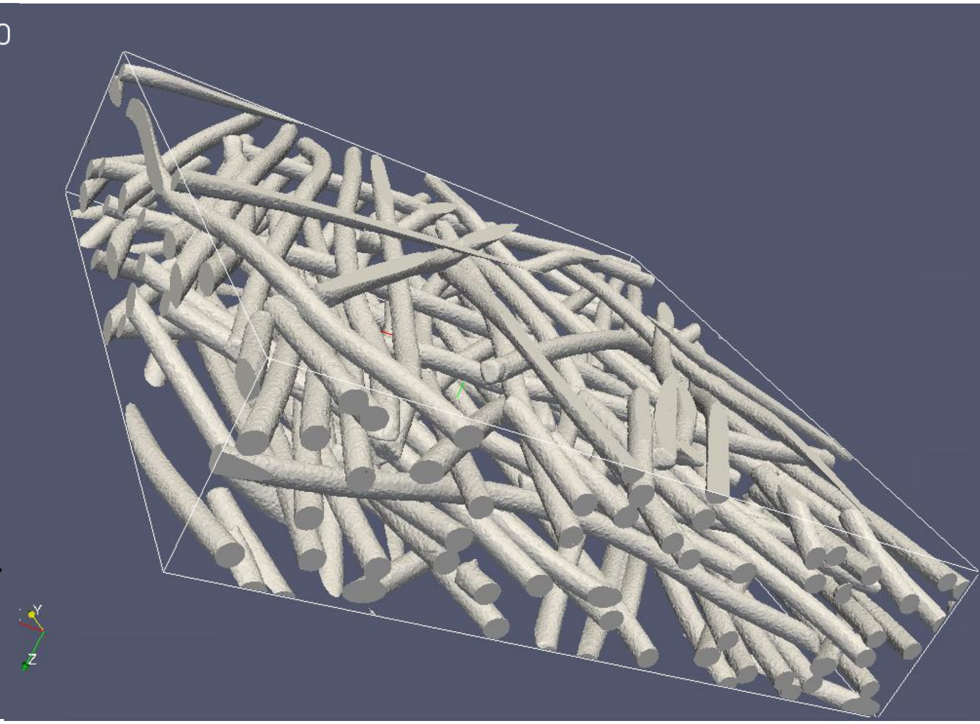
Résultats 3D: perméabilité « intrinsèque »

Arrangement irrégulier 3D: éléments finis

- Génération du maillage éléments finis: interpolation + adaptation, comme précédemment (sur **96 cœurs**)
- Taille du maillage sur le gros volume: ~**5 millions de nœuds** (par rapport à **178.2 millions de voxels,** sur le 900x900x220)



Maillage adapté



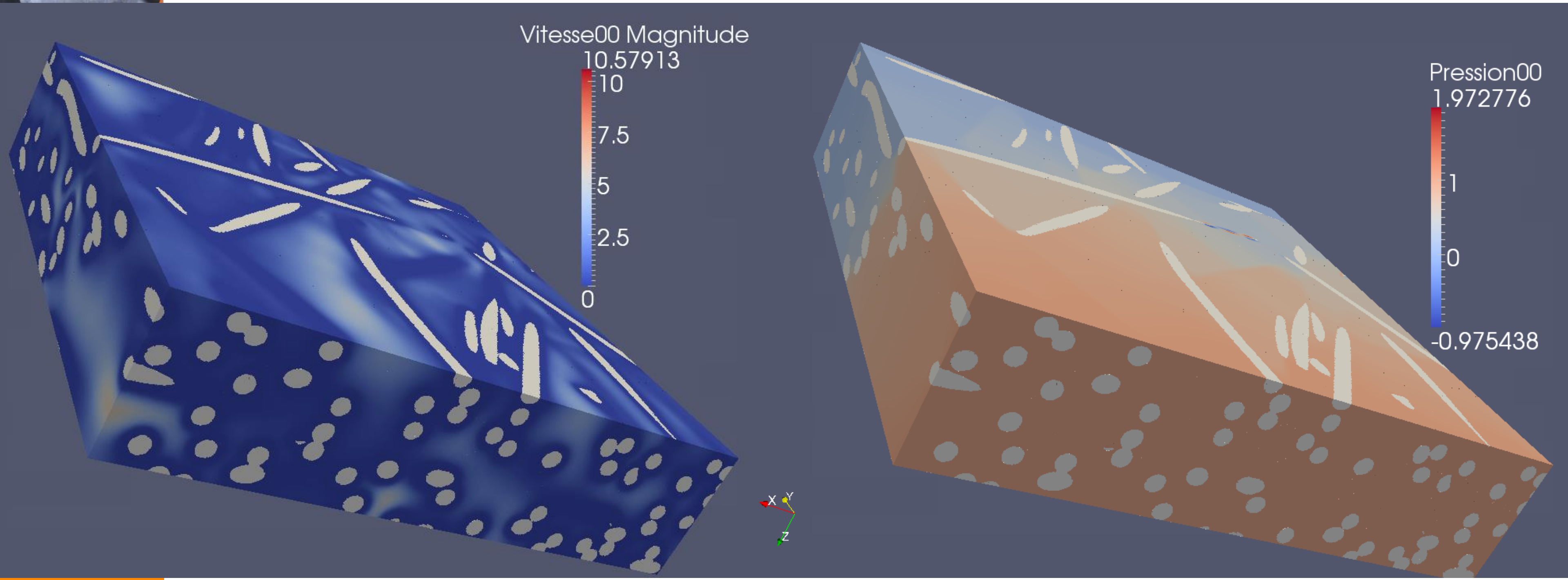
Isovaleur 0 de la distance calculée sur le maillage

[Digonnet , Silva et al (soumis à IJHPCA 2015)]

Résultats 3D: perméabilité « intrinsèque »

Arrangement irrégulier 3D: éléments finis

- Calcul de l'écoulement
- Valeurs de perméabilité ~10^{e-9} à 10^{e-10}



Norme de la vitesse ***Pression***

[Digonnet , Silva et al (soumis à IJHPCA 2015)]

Remerciements

- A Laurent Orgeas, Pierre Dumont, Sabine Rolland du Roscoat & Pierre Latil (3SR – INP Grenoble)
- A Etienne Decenciere et Dominique Jeulin (CMM-MINES ParisTech)
- Au GENCI (Grand Equipement National de Calcul Intensif) et à PRACE pour avoir permis l'accès à Curie, Occigen et JuQUEEN, dans les projets Real2HPC et BubNoz
- A l'Institut MINES Telecom pour le soutien au projet phare « De l'image au maillage »