Goal-oriented mesh adaptation for FSI problems

Éléonore Gauci[*+], **Frédéric Alauzet**[+], **Alain Dervieux**[*]

([+]) INRIA, Team Gamma3, Saclay, France
([*]) INRIA, Team Ecuador, Sophia-Antipolis, France

Réunion ANR Maidesc, 25 Octobre 2016

Motivations

Objective

Combine goal-oriented mesh adaptation and simulations with moving bodies.

Main numerical difficulty

How to handle the geometry displacement ?

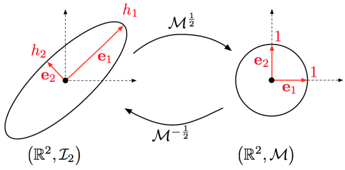
Main idea : introduce the use of metrics field, and notion of unit mesh.

[George, Hecht and Vallet., Adv Eng. Software 1991]

- Riemannian metric space: $\mathcal{M}$: $d \times d$ symmetric definite positive matrix

$$\langle u, v\rangle_{\mathcal{M}} = {}^t\,\mathbf{u}\mathcal{M}\mathbf{v} \Rightarrow \ell_{\mathcal{M}}(\mathbf{a},\mathbf{b}) = \int_{\mathbf{0}}^{\mathbf{1}} \sqrt{{}^{\mathbf{t}}\mathbf{ab}\ \mathcal{M}(\mathbf{a}+\mathbf{tab})\ \mathbf{ab}}\ \mathrm{d}\mathbf{t}$$

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}}\ \mathrm{d}|K|$$



continuous **Metric Field**$\rightarrow$ discrete **Mesh**.

Time-accurate Feature-based Mesh Adaptation

Deriving the best mesh to compute the characteristics of a given solution w in space and time

[Tam et al.,CMAME 2000], [Picasso, SIAMJSC 2003], [Formaggia et al, ANM 2004], [Frey and Alauzet CMAME 2005], [Gruau and Coupez, CMAME 2005], [Huang, JCP 2005], [Compere et al., 2007]

- **Discrete** space-time mesh adaptation problem :

Find $\mathcal{H}_{L^p}^{opt}$ having N_{st} vertices such that

$$\mathcal{H}_{L^p}^{opt} = \mathrm{Argmin}_{\mathcal{H}} ||u - \Pi_h u||_{\mathcal{H},\mathbf{L}^p(\Omega\times[0,T])}$$

- Well-posed **Continuous** space-time mesh adaptation problem :

Find $\mathcal{M}_{\mathbf{L^p}}^{\mathbf{opt}}$ of complexity N_{st} such that

$$E_{L^p}(\mathcal{M}_{L^p}^{opt}) = \min_{\mathcal{M}} \left(\int_0^T \int_\Omega \mathrm{Trace}(\mathcal{M}(\mathbf{x},t)^{-\frac{1}{2}} |H_u(\mathbf{x},t)| \mathcal{M}(\mathbf{x},t)^{-\frac{1}{2}})^p \mathrm{d}\mathbf{x}\mathrm{d}t \right)^{\frac{1}{p}}$$

$\Rightarrow$ Solved by variational calculus

Time-accurate Feature-based Mesh Adaptation

Optimal Mesh

$$\mathcal{M}_{L^p}^{opt} = N_{st}^{\frac{2}{3}} \left(\int_0^T \tau(t)^{\frac{-2p}{2p+3}} \mathcal{K}(t) \mathrm{d}t \right)^{-\frac{2}{3}} \tau(t)^{\frac{2}{2p+3}} \left(\det |H_u(\mathbf{x}, t)|\right)^{\frac{-1}{2p+3}} |H_u(\mathbf{x}, t)|$$

with $\mathcal{K}(t) = \left(\int_\Omega \left(\det |H_u(\mathbf{x}, t)|\right)^{\frac{p}{2p+3}} \mathrm{d}\mathbf{x} \right)$

Global normalization term requires the whole computation

- A global fixed-point algorithm
  - $\Rightarrow$ to compute the space-time metric complexity
  - $\Rightarrow$ to converge the **non-linear** mesh adaptation problem
  - $\Rightarrow$ to predict the solution evolution
- Split the simulation into several time sub-intervals and set an adapted mesh for each sub-interval
  - $\Rightarrow$ to limit the number of meshes

$$[0, T] = [t_0 = 0, t_1] \cup [t_1, t_2] \cup \ldots [t_{kmax}, T]$$

Unsteady Feature-based Mesh Adaptation : Algorithm

For j=1,nptfx

  For i=1,nadap

- $\mathcal{S}^j_{0,i} = \texttt{InterpolateSolution}(\mathcal{H}^j_{i-1}, \mathcal{S}^j_{i-1}, \mathcal{H}^j_i)$
- $\mathcal{S}^j_i = \texttt{SolveState}(\mathcal{S}^j_{0,i}, \mathcal{H}^j_i)$
- $|H_{\max}|^j_i = \texttt{ComputeHessianMetric}(\mathcal{H}^j_i, \{\mathcal{S}^j_i(k)\}_{k=1,nk})$

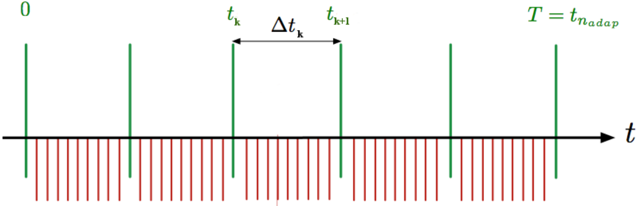
  End for $\mathcal{C}^j = \texttt{ComputeSpaceTimeComplexity}(\{|H_{\max}|^j_i\}_{i=1,nadap})$

  $\mathcal{M}^{j-1}_i = \texttt{ComputeUnsteadyLpMetrics}(\mathcal{C}^{j-1}, |H_{\max}|^{j-1}_i)$

  $\mathcal{H}^j_i = \texttt{GenerateAdaptedMeshes}(\mathcal{H}^{j-1}_i, \mathcal{M}^{j-1}_i)$

End for

$$[0, T] = [t_0 = 0, t_1] \cup [t_1, t_2] \cup \ldots [t_{kmax}, T]$$



For j=1,nptfx

For i=1,nadap

- $\mathcal{S}^j_{0,i} = \texttt{InterpolateSolution}(\mathcal{H}^j_{i-1}, \mathcal{S}^j_{i-1}, \mathcal{H}^j_i)$
- $\mathcal{S}^j_i = \texttt{SolveState}(\mathcal{S}^j_{0,i}, \mathcal{H}^j_i)$
- $|H_{\max}|^j_i = \texttt{ComputeHessianMetric}(\mathcal{H}^j_i, \{\mathcal{S}^j_i(k)\}_{k=1,nk})$

End for $\mathcal{C}^j = \texttt{ComputeSpaceTimeComplexity}(\{|H_{\max}|^j_i\}_{i=1,nadap})$

$\mathcal{M}^{j-1}_i = \texttt{ComputeUnsteadyLpMetrics}(\mathcal{C}^{j-1}, |H_{\max}|^{j-1}_i)$

$\mathcal{H}^j_i = \texttt{GenerateAdaptedMeshes}(\mathcal{H}^{j-1}_i, \mathcal{M}^{j-1}_i)$

End for

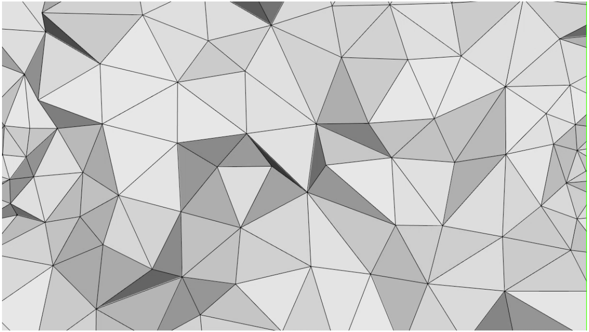
$$[0,T] = [t_0 = 0, t_1] \cup [t_1, t_2] \cup \ldots [t_{kmax}, T]$$

It is then possible to take into account the mesh motion inside the error estimate
$\Longrightarrow$ optimal space-time adapted mesh in ALE framework

Mesh optimization for moving meshes

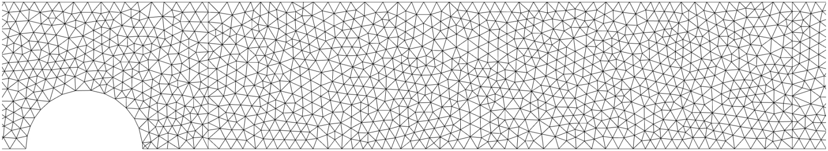
As mesh quality tends to decrease while the mesh is moving
$\Rightarrow$ regular **optimization phases** must be performed : smoothing and edge swapping

- Two planes moved at Mach 0.4 inside an inert air.
- The planes are translated and rotating
- 50 sub intervals and 3 adaptation loops
- Total space time complexity: $36,000,000$ vertices, average mesh size: $732,000$ vertices, $80,000$ timesteps

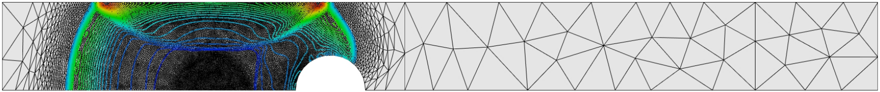


2D Pilot case

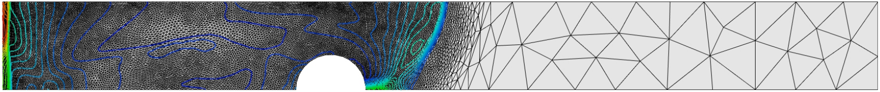
- Blast initialization : high density (10,**0**,25) and air (1,**0**,2.5)
- 2D geometry : ($5 \times 0.5\ m^2$) : 2395 Vertices
- cost function j : $j : j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$



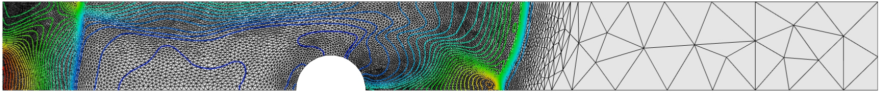
2D Pilot case



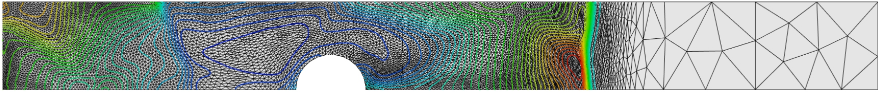
Time: 0.175



Time: 0.35



Time: 0.525



Time: 0.7

Deriving the best mesh to observe a given output scalar functional

$$j(w) = \langle g, w \rangle$$

[Venditti and Darmofal, JCP 2003], [Jones et al., AIAA 2006], [Power et al, CMA 2006], [Wintzer et al., AIAA 2008], [Leicht and Hartmann, JCP 2010]

- Let W be the solution of the state equation $\Psi(W) = 0$
- Choose a scalar functional j.
- Minimize $\delta j_h = |j - j_h| = |(g, W) - (g, W_h)|$
- Introduce the adjoint state W^*

$$\left(\frac{\partial \Psi_h}{\partial W_h} \varphi_h, W_h^* \right) = (g, \varphi_h)$$

to estimate the error $\delta j_h \; \approx \; (W^*, \Psi_h(W) - \Psi(W))$

- Minimize δj_h with an *a priori* error estimate

Unsteady Adjoint Resolution (Euler Equation)

The **continuous** state model on $\Omega \times [0, T]$ obeys to :

$$\Psi(W) = 0$$

The **discrete** state model writes :

$$W_h{}^n = W_h{}^{n-1} + \delta t^n \Phi_h(W_h^{n-1})$$

Consider a time-dependent functional :

$$j(W) = \int_0^T \int_\Gamma j_\Gamma(W(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t$$

The **continuous** adjoint state on $\Omega \times [0, T]$ obeys to :

$$\Psi^*(W, W^*) = 0 \quad \text{or} \quad -\frac{\partial W^*}{\partial t} - \left(\frac{\partial \mathcal{F}}{\partial W}\right) \nabla W^* = g \qquad (1)$$

The **discrete** adjoint state writes :

$$W_h{}^{*,n-1} = W_h{}^{*,n} + \delta t^n \frac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - \delta t^n (W_h{}^{*,n})^T \frac{\partial \Phi_h}{\partial W_h^{n-1}}(W_h^{n-1})$$

Time accurate Goal-oriented mesh adaptation

Difficulties

Computing $W^{*,n-1}$ at time t^{n-1} requires the knowledge of state W^{n-1} and adjoint state $W^{*,n}$

$\Rightarrow$ the knowledge of all states $W^n, n = 1 \cdots N$ is needed

Solve state foreward: $\Psi(W) = 0$

Solve adjoint state backward: $\Psi^*(W, W^*) = 0$

$\Rightarrow$ Large memory storage effort in 3D

(10^6 vertices & 10^3 iterations request 37.25 Gb)

Adopted solution

Solve state once to get checkpoints

State interpolation between two memory storage

For j=1,nptfx

For i=1,nadap

- $\mathcal{S}^j_{0,i} = \texttt{ConservativeSolutionTransfer}(\mathcal{H}^j_{i-1}, \mathcal{S}^j_{i-1}, \mathcal{H}^j_i)$
- $\mathcal{S}^j_i = \texttt{SolveState}(\mathcal{S}^j_{0,i}, \mathcal{H}^j_i)$

End for

For i=nadap,1

- $(\mathcal{S}^*)^j_i = \texttt{AdjointStateTransfer}(\mathcal{H}^j_{i+1}, (\mathcal{S}^*_0)^j_{i+1}, \mathcal{H}^j_i)$
- $\{\mathcal{S}^j_i(k), (\mathcal{S}^*)^j_i(k)\} = \texttt{SolveStateAndAdjointBackward}(\mathcal{S}^j_{0,i}, (\mathcal{S}^*)^j_i, \mathcal{H}^j_i)$
- $|H_{\max}|^j_i = \texttt{ComputeGoalOrientedHessianMetric}(\mathcal{H}^j_i, \{\mathcal{S}^j_i(k), (\mathcal{S}^*)^j_i(k)\})$

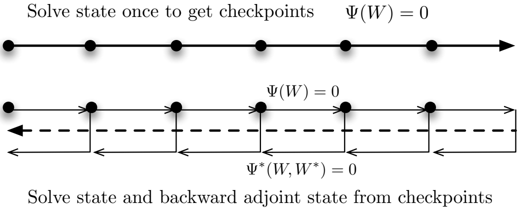
End for

$\mathcal{C}^j = \texttt{ComputeSpaceTimeComplexity}(\{|H_{\max}|^j_i\}_{i=1,nadap})$

$\mathcal{M}^j_i = \texttt{ComputeUnsteadyLpMetrics}(\mathcal{C}^{j-1}, |H_{\max}|^{j-1}_i)$

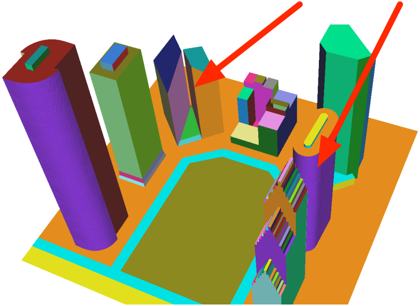
$\mathcal{H}^{j+1}_i = \texttt{GenerateAdaptedMeshes}(\mathcal{H}^j_i, \mathcal{M}^j_i)$

End for



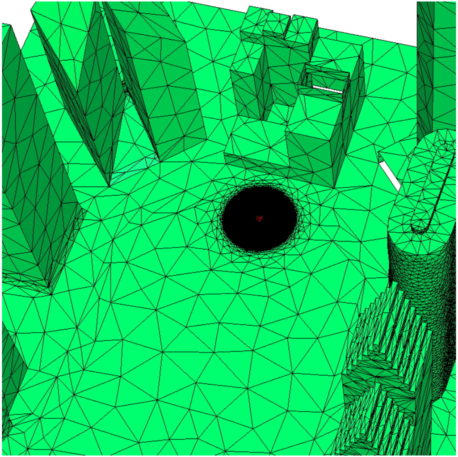
- Blast initialization : high density (10,0,0,0,25) and air (1,0,0,0,2.5)
- 3D town geometry (85 x 70 x 70 m^3) : 4187548 vertices
- cost function j : $j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$

The observation Γ are these 2 buildings



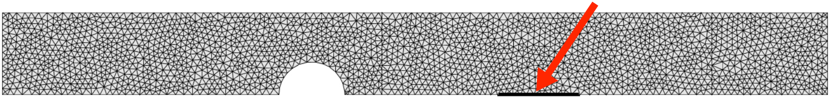
Blast in the city

- Blast initialization : high density (10,0,0,0,25) and air (1,0,0,0,2.5)
- 3D town geometry (85 x 70 x 70 m^3) : 4187548 Vertices
- cost function j : $j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t$.



2D Pilot case

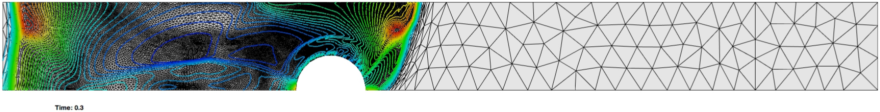
- Blast initialization : high density (10,**0**,25) and air (1,**0**,2.5)
- 2D geometry : (5 x 0.5 m^2) : 2395 Vertices
- cost function j : $j : j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$

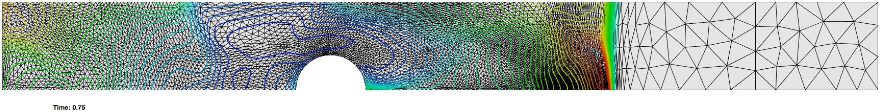


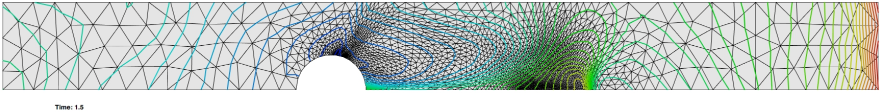
2D Pilot case

- Blast initialization : high density (10,**0**,25) and air (1,**0**,2.5)
- 2D geometry : (5 x 0.5 m^2) : 2395 Vertices
- cost function j : $j : j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$

From Top to Down : Time 0.3 - 0.75 - 1.5







Unsteady Adjoint Resolution in ALE framework

The **discrete** unsteady state model in ALE writes :

$$W_h^n = W_h^{n-1} + \delta t^n \Phi_{h,ale}(W_h^{n-1})$$

Consider a time-dependent functional :

$$j(W) = \int_0^T \int_\Gamma j_\Gamma(W(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t$$

The **continuous** unsteady adjoint state on $(\Omega, [0,T])$ in ALE obeys to :

$$\Psi^*(W, W^*) = 0 \quad \text{or} \quad -\frac{\partial W^*}{\partial t} - \left(\frac{\partial \mathcal{F}_{ale}}{\partial W}\right)\nabla W^* = g \tag{2}$$

The **discrete** unsteady adjoint state in ALE writes :

$$W_h{}^{*,n-1} = W_h^{*,n} + \delta t^n \frac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - \delta t^n (W_h^{*,n})^T \frac{\partial \Phi_{h,ale}}{\partial W_h^{n-1}}(W_h^{n-1})$$

Unsteady Adjoint Resolution in ALE : Technical Difficulties

Mesh displacement σ

- Mesh displacement σ for the adjoint resolution in backward is the same as it was in the forward state resolution
- Jacobian of fluxes : $\dfrac{\partial \Phi_{h,ale}}{\partial W_h^{n-1}}$
- Boundary conditions : Boundary fluxes take also into account displacement of geometry

Deal with connectivities

Backward Mesh : store the positions but not necessary the connectivities

$\Rightarrow$ The adjoint state W^* verify the DGCL property :

$$(|C^{n+1}| - |C^n|) - \int_n^{n+1} \int_{\partial C(t)} (\mathbf{w}^* \cdot \mathbf{n}) \mathrm{d}\mathbf{s}\mathrm{d}t = 0$$

Pilot Case

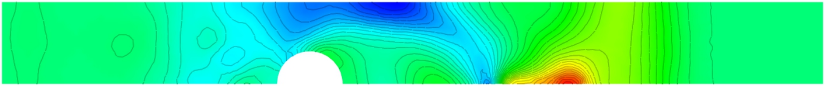
State Resolution in Forward

Fixed Mesh

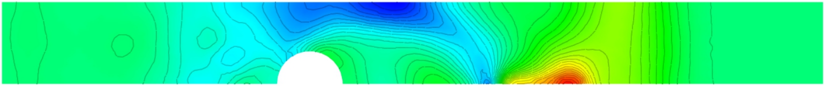
Moving Mesh

Adjoint Resolution in Backward

Fixed Mesh

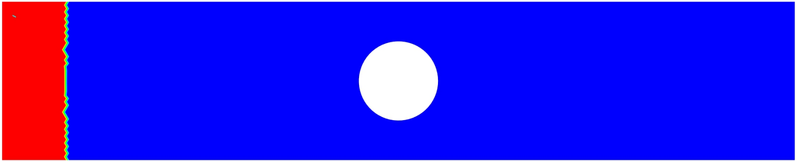


Moving Mesh

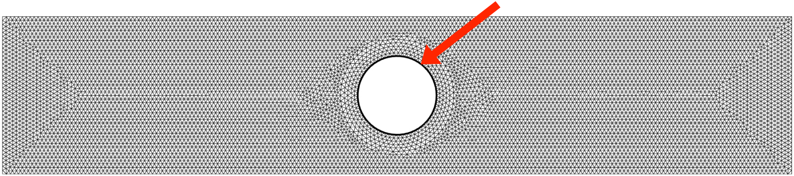


FSI case

- Lift-off of a circle by a Mach 3 Shock Wave
- 2D geometry : $(1 \times 0.2\ m^2)$: 9064 Vertices
- cost function j : $j : j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$

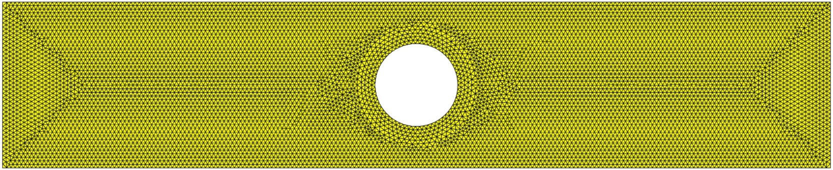


The observation is the pressure on this circle

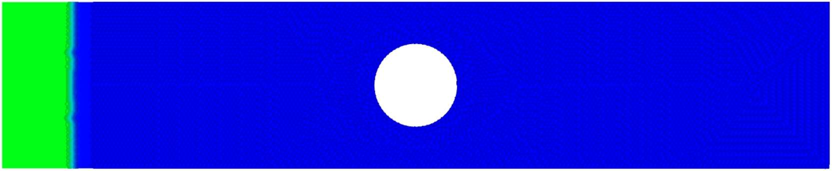


FSI Case

Mesh generated in forward

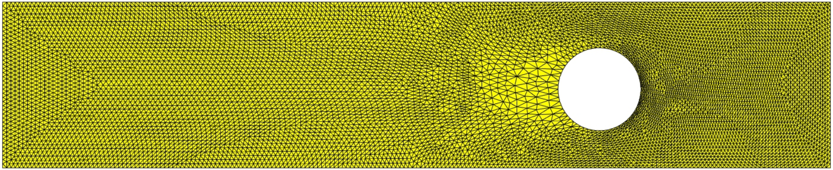


Solve State

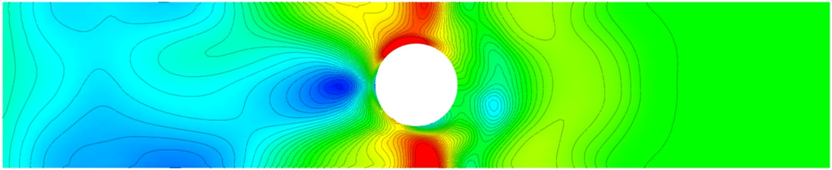


FSI Case

Mesh generated in backward



Adjoint state in backward

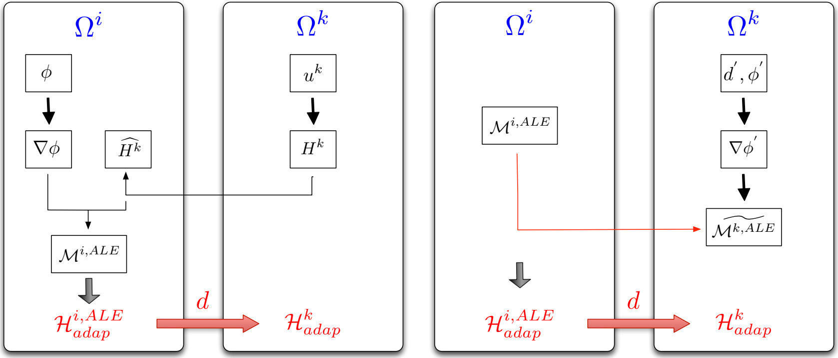


Current study : ALE metric Update

Let t^k be a time into the subset $[t^i, t^{i+1}]$,

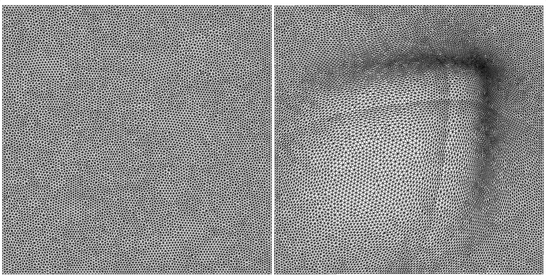
$$\begin{array}{cccc} \phi' : & \Omega^k & \rightarrow & \Omega^i \\ & \mathbf{x}^k & \mapsto & \mathbf{x}^i = \mathbf{x}^k + \mathbf{d}'(\mathbf{x}^k). \end{array}$$

$$\begin{aligned} 1 &= \left(\mathbf{e}^i(\mathbf{x}^i)\right)^T \mathcal{M}_{Lp}^{i,\mathrm{ALE}}\left(\mathbf{x}^i\right) \left(\mathbf{e}^i(\mathbf{x}^i)\right) \\ &= \left[\mathbf{e}^i\left(\boldsymbol{\phi}'(\mathbf{x}^k)\right)\right]^T \mathcal{M}_{Lp}^{i,\mathrm{ALE}}\left(\mathbf{x}^i\right) \mathbf{e}^i\left(\boldsymbol{\phi}'(\mathbf{x}^k)\right) \\ &= \left(\left[\nabla^k \boldsymbol{\phi}'(\mathbf{x}^k)\right]^T \mathbf{e}^k(\mathbf{x}^k)\right)^T \mathcal{M}_{Lp}^{i,\mathrm{ALE}}(\mathbf{x}^i) \left(\left[\nabla^k \boldsymbol{\phi}'(\mathbf{x}^k)\right]^T \mathbf{e}^k(\mathbf{x}^k)\right) \\ &= \left(\mathbf{e}^k(\mathbf{x}^k)\right)^T \left\{\nabla^k \boldsymbol{\phi}'(\mathbf{x}^k)\, \mathcal{M}_{Lp}^{i,\mathrm{ALE}}(\mathbf{x}^i) \left[\nabla^k \boldsymbol{\phi}'(\mathbf{x}^k)\right]^T\right\} \left(\mathbf{e}^k(\mathbf{x}^k)\right) . \end{aligned}$$

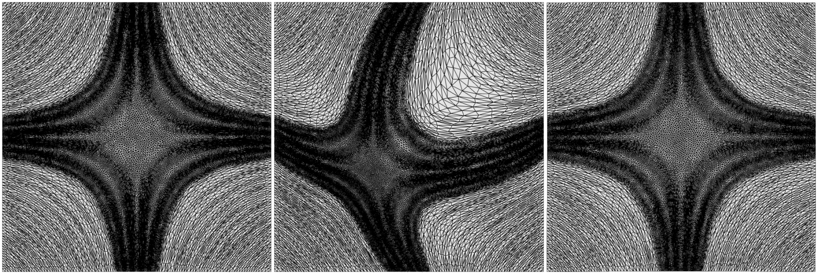


(From left to right) Metric without update, Updated metric

Current study : ALE metric Update

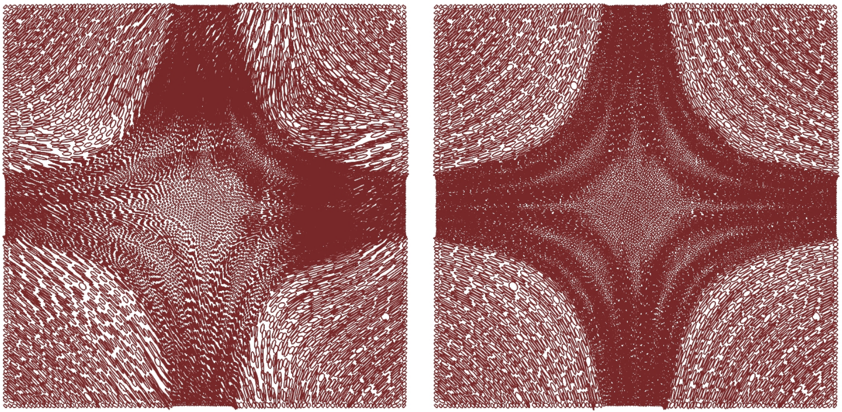


(From left to right) Mesh at t^i, Mesh at t^k



Hessian metric at t^k, Updated ALE Hessian metric at t^i, Moved and Updated ALE Hessian metric at t^k

Current study: ALE metric Update



(From left to right) Metric without update, Updated metric

Conclusions Perspectives

Conclusions

- Time accurate goal-oriented mesh adaptation
- Time accurate feature-based mesh adaptation in ALE framework
- Evaluation of Unsteady Adjoint on moving meshes
- ALE metric update

Perspectives

- Time accurate goal oriented mesh adaptation in ALE framework

THANK YOU FOR YOUR ATTENTION