

Contrôle optimal d'un réseau de neurones pour le recalage d'organe en temps réel en chirurgie augmentée

Stéphane Cotin, Guillaume Mestdagh, Alban Odot, Yannick Privat



Reproduction et développement des plantes
ENS de Lyon
46 allée d'Italie
69007 Lyon FRANCE

Courriel: guillaume.mestdagh@inria.fr

Chirurgie augmentée

Chirurgie faiblement invasive

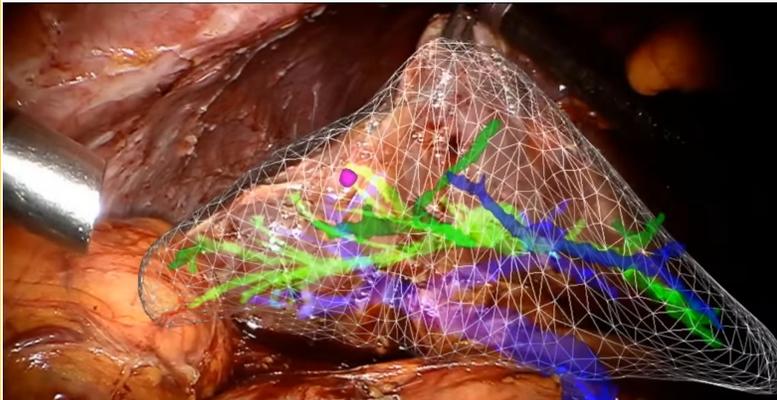


- Interaction indirecte avec l'organe
- Retour vidéo sur un écran
- Champ de vision limité, structures internes cachées
- Scène en mouvement

Une opération faiblement invasive (image : Samuel Bendet, US Air Force)

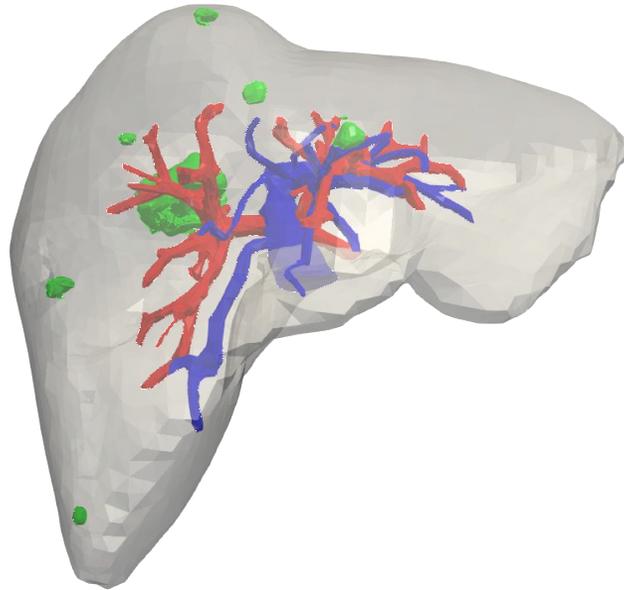
Chirurgie augmentée

Réalité augmentée



Objectif
Réalité augmentée
pendant la chirurgie

Inria, 2018



Données préopératoires
Modèle 3D du foie extrait
d'images tomographiques



Données peropératoires
Position de la surface du
foie déformée

Problème : Mettre à jour la déformation du foie pendant l'opération

Chirurgie augmentée

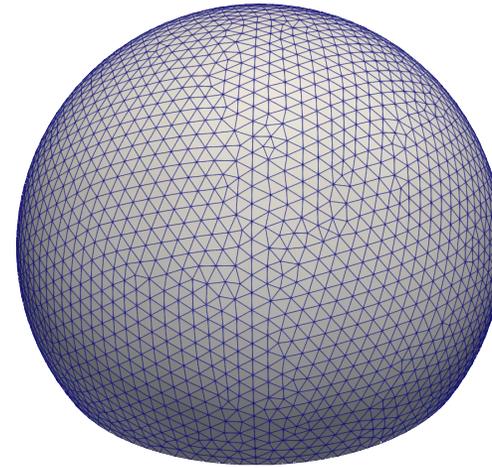
Recalage élastique

Objectif

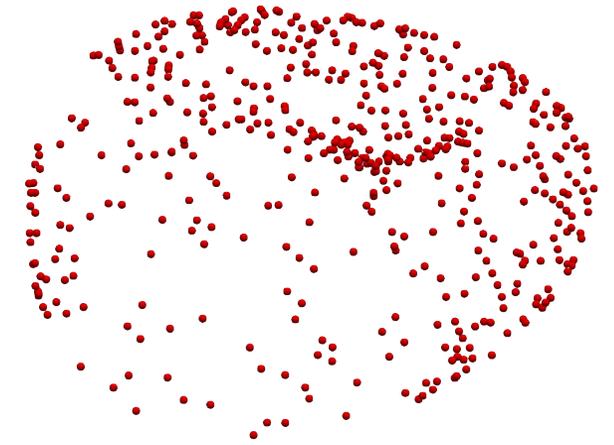
Calculer une déformation élastique à partir de données de surface

Données disponibles

- Maillage représentant le solide dans sa configuration de référence
- Modèle élastique
- Nuage de points représentant la surface déformée



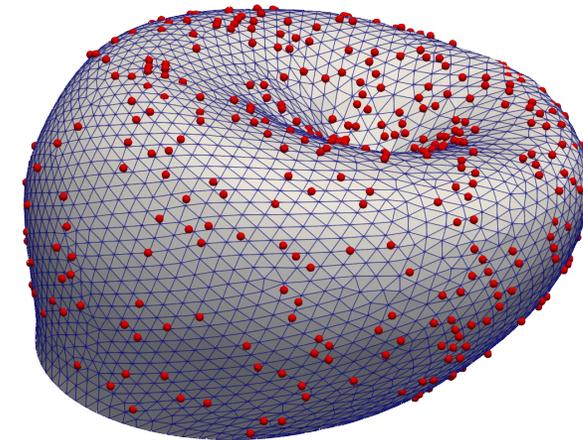
Maillage initial



Nuage de points



Calcul d'une déformation



Maillage déformé

Plan de l'exposé

1. Contexte et modélisation
2. Méthode d'adjoint classique
3. Un réseau de neurones pour recalculer plus vite

1 Contexte et modélisation

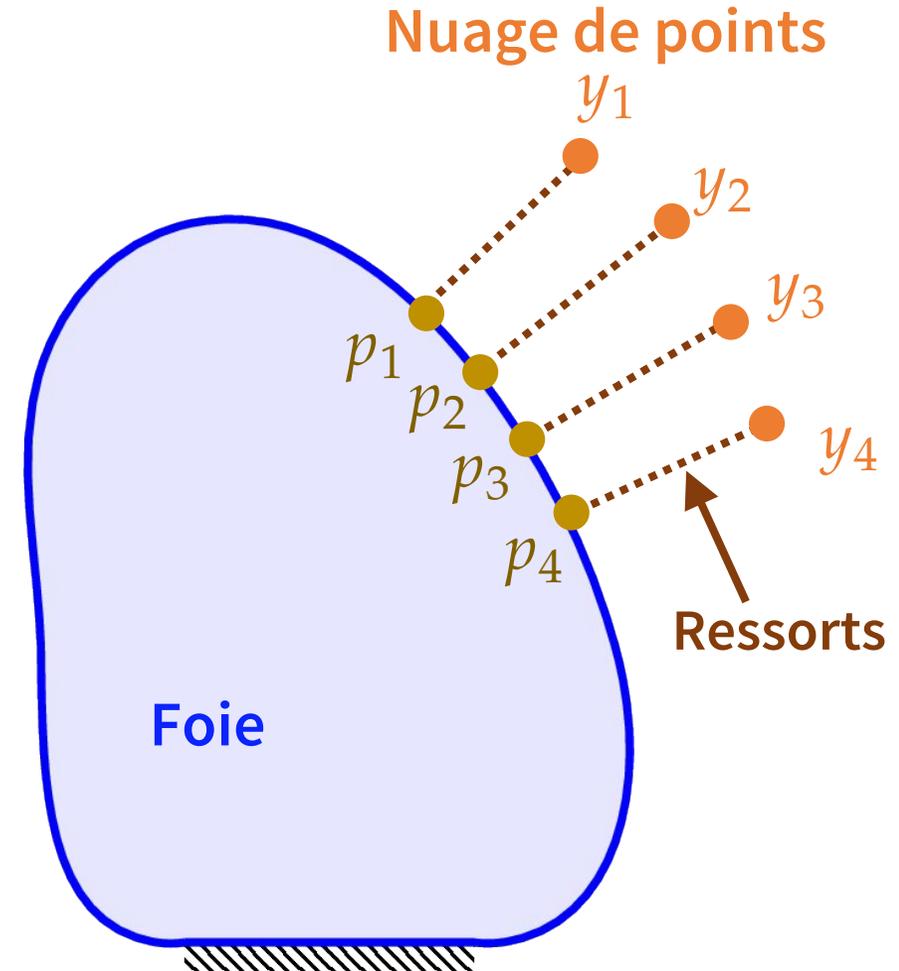
Iterative closest point et variantes

Ajout de forces fictives dans le modèle direct

- Forces attractives (ressorts) entre les p_j et les y_j
- Le modèle élastique régularise le déplacement

Limites de cette approche

- Les forces ne s'appliquent que là où il y a des données
- Pas de lien avec les causes réelles du déplacement



1 Contexte et modélisation

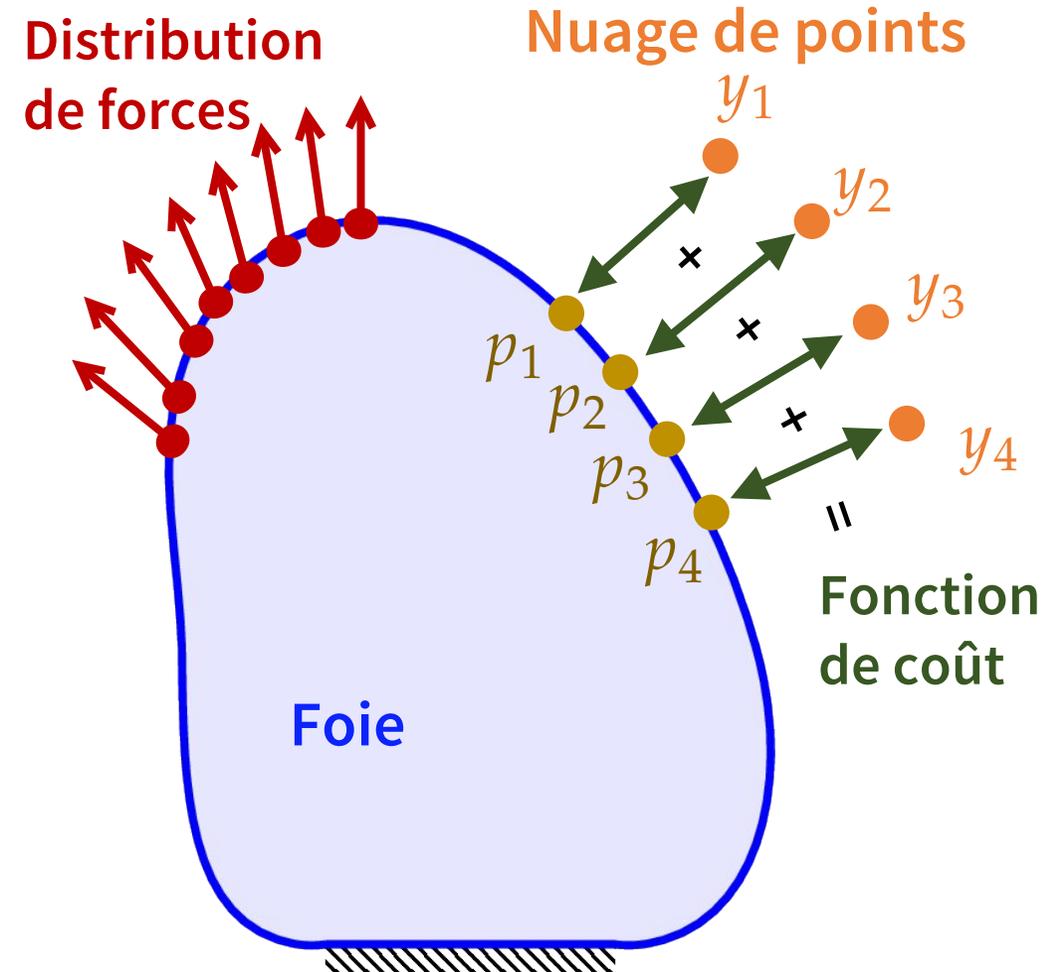
Approches par problème inverse

Prise en compte des causes du déplacement

- Hypothèses physiques sur les forces à l'origine du déplacement
- Minimisation d'une fonction de coût

Limites de ces approches

- Méthodes très spécifiques à un scénario donné
- Limité à l'élasticité linéaire

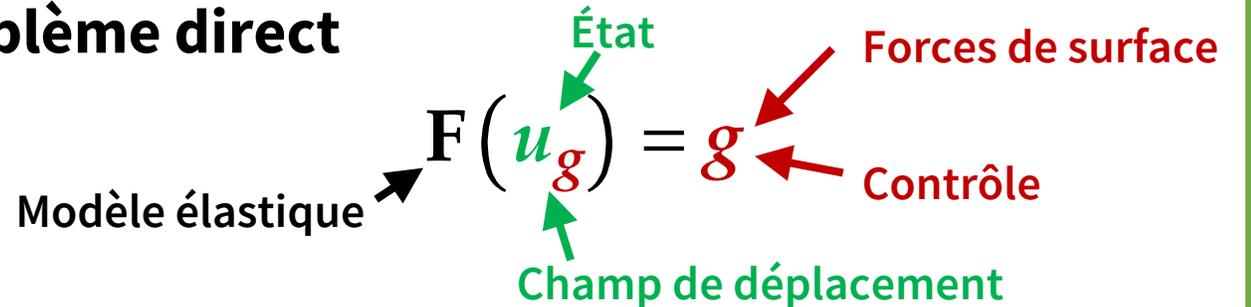


1 Contexte et modélisation

Notre approche

- Formalisme du contrôle optimal
- Problème continu indépendant du maillage
- Accès aux outils classiques de l'optimisation
- Compatible avec l'élasticité non-linéaire

Problème direct



Problème d'optimisation

$$\min_{g} J(u_g) \quad \text{sous la contrainte} \quad F(u_g) = g$$

Fonction de coût \rightarrow J

1 Contexte et modélisation

Notre approche

Formulation dépendant uniquement du contrôle

$$\min_g \phi(\mathbf{g}) \quad \text{où} \quad \phi(\mathbf{g}) = J(\mathbf{u}_{\mathbf{g}})$$

↑
Fonction objectif

- Une seule variable d'optimisation : le contrôle
- Un appel à la fonction $\phi(\mathbf{g})$ nécessite la résolution du problème élastique
- Utilisation d'une méthode d'adjoint pour la résolution numérique

2 Méthode d'adjoint classique

Implémentation de la méthode d'adjoint

Solveur d'optimisation

Engendre des itérés

Nouvel itéré \mathbf{g}

Gradient dans l'espace des \mathbf{g}

$$\phi(\mathbf{g}) = J(\mathbf{u}_{\mathbf{g}})$$

$$\nabla\phi(\mathbf{g}) = \mathbf{p}$$

Méthode d'éléments finis

Gère le problème élastique

Calcul de l'état associé

$$\mathbf{F}(\mathbf{u}_{\mathbf{g}}) = \mathbf{g}$$

Calcul de l'état adjoint

$$[\nabla\mathbf{F}(\mathbf{u}_{\mathbf{g}})]^T \mathbf{p} = \nabla J(\mathbf{u}_{\mathbf{g}})$$

Matrice tangente État adjoint

Fonction de coût

Mesure l'attache aux données

Évaluation de la fonction de coût

$$J(\mathbf{u}_{\mathbf{g}}) \quad \nabla J(\mathbf{u}_{\mathbf{g}})$$

2 Méthode d'adjoint classique

Implémentation de la méthode d'adjoint

Solveur d'optimisation

Engendre des itérés

Solveurs Scipy

- Problème sans contraintes
- Méthode de quasi-Newton (L-BFGS)

Méthode d'éléments finis

Gère le problème élastique

Fenics

- Élasticité non-linéaire
- Besoin de robustesse
- Méthode de Newton à région de confiance

Fonction de coût

Mesure l'attache aux données

Package Trimesh

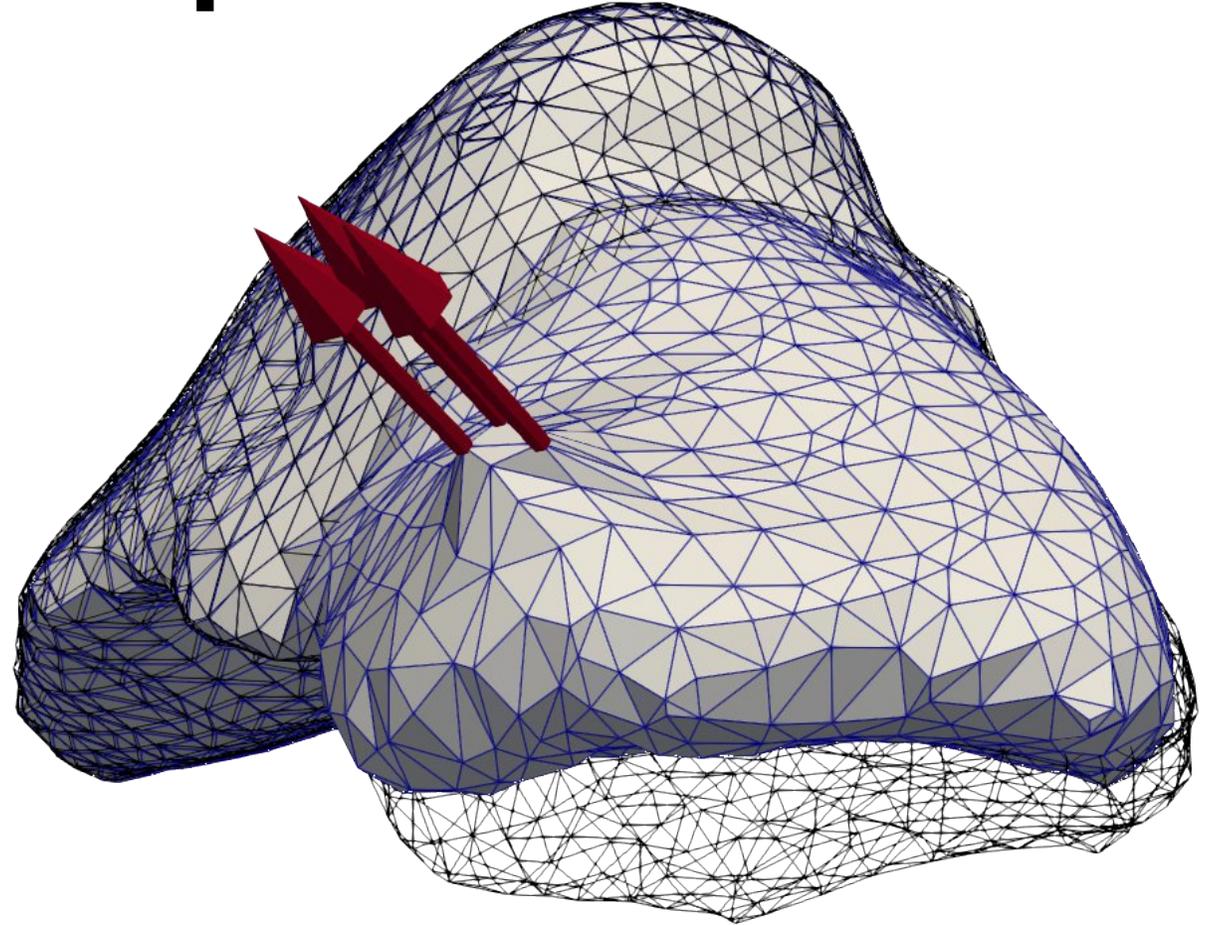
- Projections sur le maillage déformé
- Utilisation d'une structure d'arbre (R-tree)

2 Méthode d'adjoint classique

Estimation d'une force ponctuelle

Génération de données

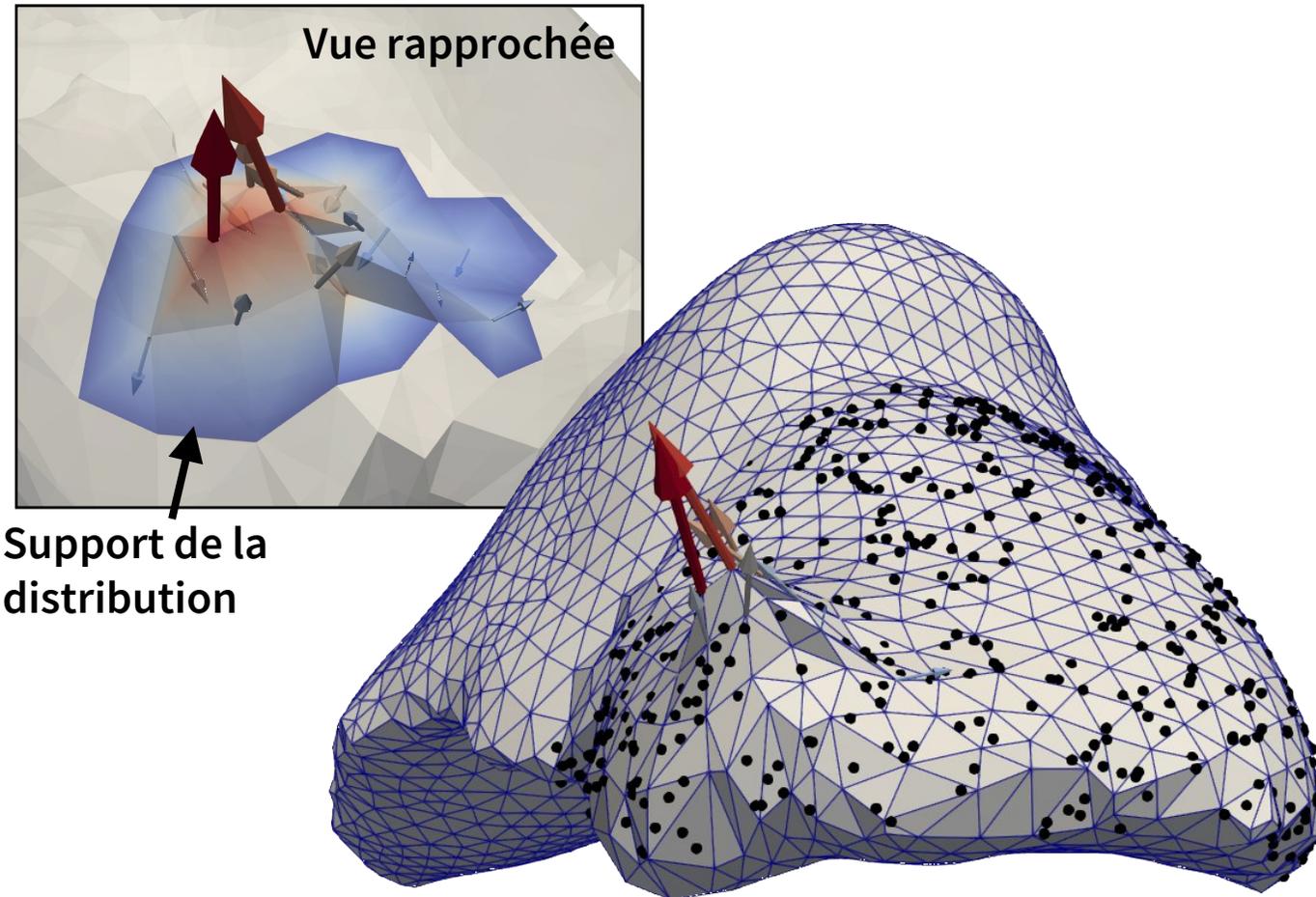
- On applique une force locale sur le maillage
- On échantillonne la surface déformée pour créer un nuage de points
- 5 séries de 50 observations, avec un petit déplacement entre observations successives



Déformation synthétique

2 Méthode d'adjoint classique

Estimation d'une force ponctuelle



Support de la distribution

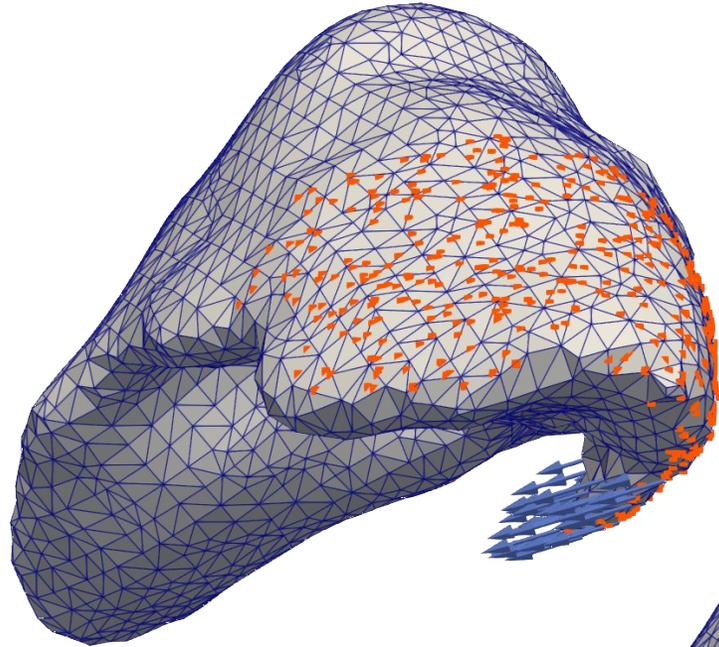
Distribution de force reconstruite

Estimation de la force

- Distribution de force restreinte à un voisinage du point d'application
- Mise à jour de l'estimation entre deux observations
- Résultante obtenue en sommant les forces à chaque nœud du maillage

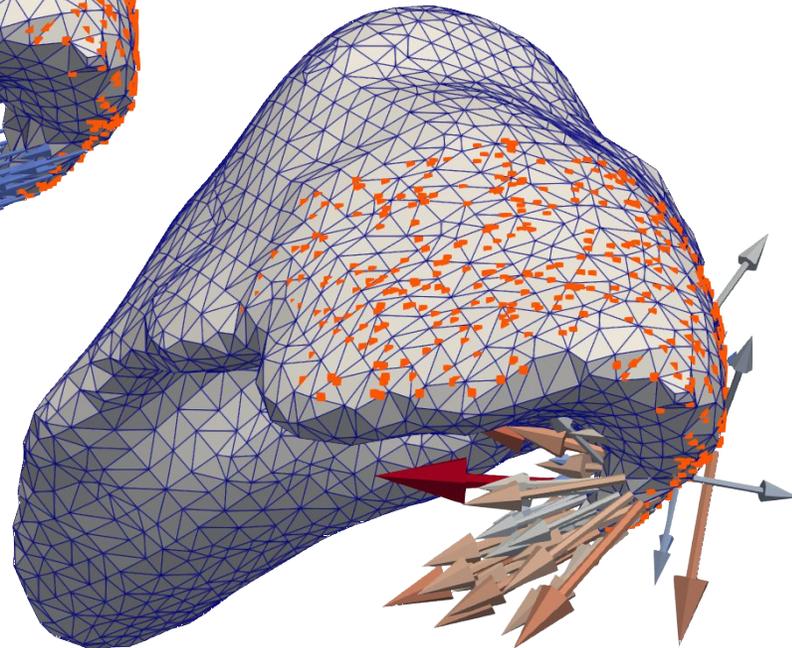
2 Méthode d'adjoint classique

Estimation d'une force ponctuelle



Déformation
synthétique

Déformation
reconstruite



Problème non-linéaire

- Néo-Hookéen
- Résolu itérativement par méthode de Newton

2 Méthode d'adjoint classique

Estimation d'une force ponctuelle

Erreur d'estimation

Valeurs moyennes pour chaque série

Cas test	Éval. objectif	Tps de mise à jour	Erreur d'estimation
Cas 1	32,8	3min 01s	7,0 %
Cas 2	5,5	1min 20s	2,5 %
Cas 3	15,7	1min 59s	1,8 %
Cas 4	24,4	2min 35s	1,1 %
Cas 5	7,6	1min 24s	11,6 %
Moyenne	17,2	2min 04s	4,8 %

Temps de calcul incompatible avec le temps réel

3 Un réseau de neurones pour recalculer plus vite

Méthode d'adjoint classique

Solveur d'optimisation

Engendre des itérés

Nouvel itéré \mathbf{g}

Gradient dans l'espace des \mathbf{g}

$$\phi(\mathbf{g}) = J(\mathbf{u}_{\mathbf{g}})$$

$$\nabla\phi(\mathbf{g}) = \mathbf{p}$$

Méthode d'éléments finis

Gère le problème élastique

Calcul de l'état associé

$$\mathbf{F}(\mathbf{u}_{\mathbf{g}}) = \mathbf{g}$$

Calcul de l'état adjoint

$$[\nabla\mathbf{F}(\mathbf{u}_{\mathbf{g}})]^T \mathbf{p} = \nabla J(\mathbf{u}_{\mathbf{g}})$$

Matrice tangente État adjoint

- Long
- Peut ne pas converger
- Fonction objectif évaluée à une certaine tolérance près

Fonction objectif

Mesure l'attache aux données

Évaluation de la fonction objectif

$$J(\mathbf{u}_{\mathbf{g}}) \quad \nabla J(\mathbf{u}_{\mathbf{g}})$$

3 Un réseau de neurones pour recalculer plus vite

Méthode d'adjoint classique

Solveur d'optimisation

Engendre des itérés

Nouvel itéré \mathbf{g}

Gradient dans l'espace des \mathbf{g}

$$\phi(\mathbf{g}) = J(\mathbf{u}_{\mathbf{g}})$$

$$\nabla\phi(\mathbf{g}) = \mathbf{p}$$

Méthode d'éléments finis

Gère le problème élastique

Calcul de l'état associé

$$\mathbf{F}(\mathbf{u}_{\mathbf{g}}) = \mathbf{g}$$

Calcul de l'état adjoint

$$[\nabla\mathbf{F}(\mathbf{u}_{\mathbf{g}})]^T \mathbf{p} = \nabla J(\mathbf{u}_{\mathbf{g}})$$

Matrice tangente État adjoint

- Long
- Peut ne pas converger
- Fonction objectif évaluée à une certaine tolérance près

Fonction objectif

Mesure l'attache aux données

Évaluation de la fonction objectif

$$J(\mathbf{u}_{\mathbf{g}}) \quad \nabla J(\mathbf{u}_{\mathbf{g}})$$

3 Un réseau de neurones pour recalculer plus vite

Méthode d'adjoint avec le réseau

Solveur d'optimisation

Engendre des itérés

Nouvel itéré \mathbf{g}

Gradient dans l'espace des \mathbf{g}

$$\phi(\mathbf{g}) = J(\mathbf{u}_{\mathbf{g}})$$

$$\nabla\phi(\mathbf{g}) = \mathbf{p}$$

Réseau de neurones

Entraîné à résoudre des problèmes élastiques

Prédiction de l'état associé

$$\mathbf{u}_{\mathbf{g}} = \mathbf{N}(\mathbf{g})$$

Rétropropagation

$$\mathbf{p} = [\nabla\mathbf{N}(\mathbf{g})]^T \nabla J(\mathbf{u}_{\mathbf{g}})$$

- Plus rapide
- Renvoie toujours une solution
- Dérivation formelle par différentiation automatique

Fonction objectif

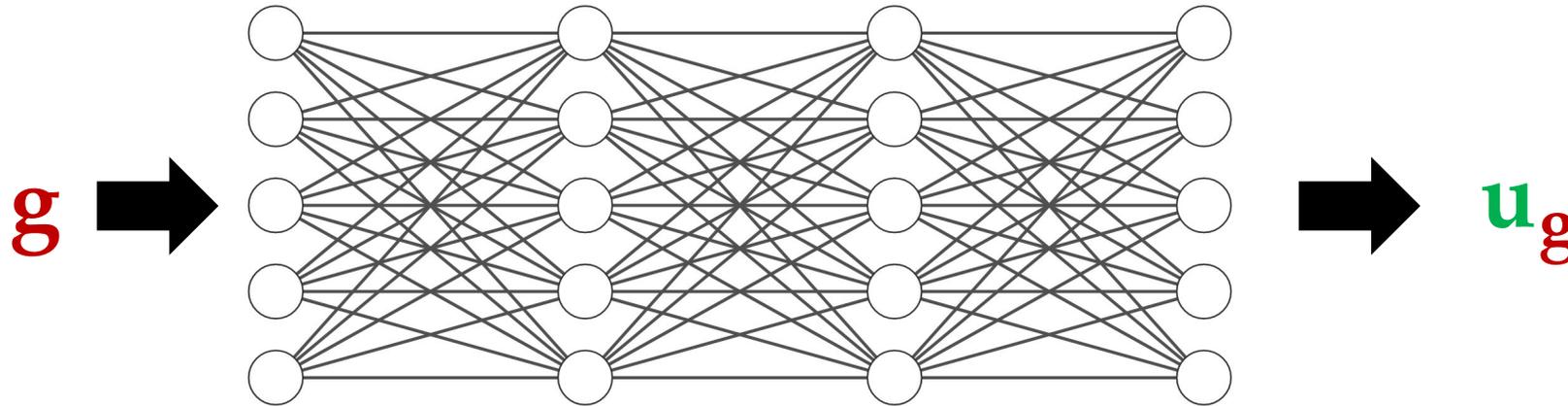
Mesure l'attache aux données

Évaluation de la fonction objectif

$$J(\mathbf{u}_{\mathbf{g}}) \quad \nabla J(\mathbf{u}_{\mathbf{g}})$$

3 Un réseau de neurones pour recalculer plus vite

Réseau de neurones



- *Fully-connected*, quatre couches, fonctions d'activation ReLU
- Uniquement des opérations matrice-vecteur directes
- Entraîné sur des simulations de déformation non-linéaire

3 Un réseau de neurones pour recalculer plus vite

Entraînement du réseau

Génération des données

- 20 000 paires $(\mathbf{g}, \mathbf{u}_{\mathbf{g}})$ générées à partir des déplacements.
- Déplacements choisis parmi les premiers modes propres du maillage
- Données complétées par des simulations où les forces sont plus localisées

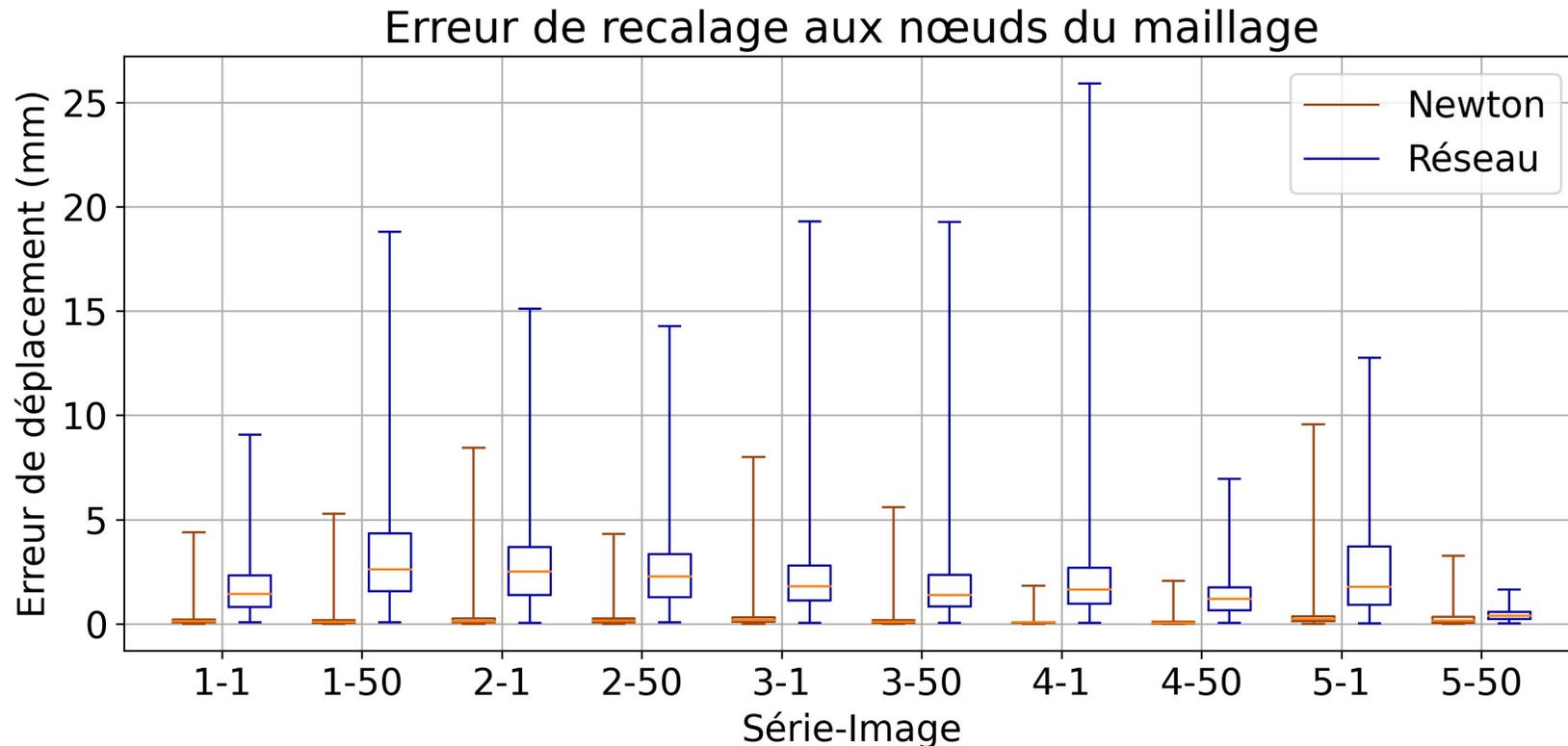
Fonction de *loss*

- Mesure d'erreur basée sur le résidu du problème élastique : $\mathbf{F}(\mathbf{u}_{\mathbf{g}}) - \mathbf{g}$

3 Un réseau de neurones pour recalibrer plus vite

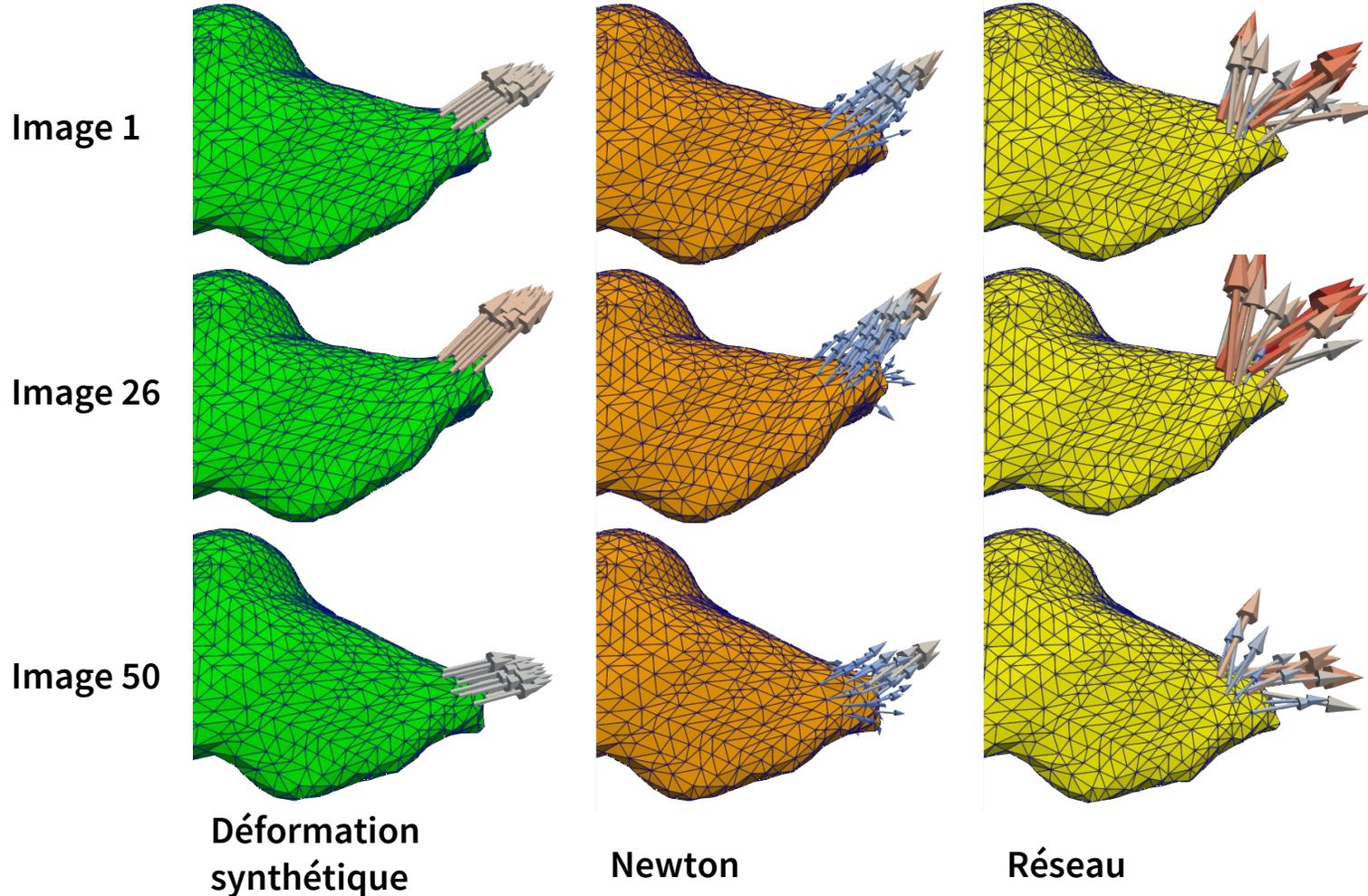
Recalage avec le réseau

Essai sur les 5 séries de déformations



3 Un réseau de neurones pour recalibrer plus vite

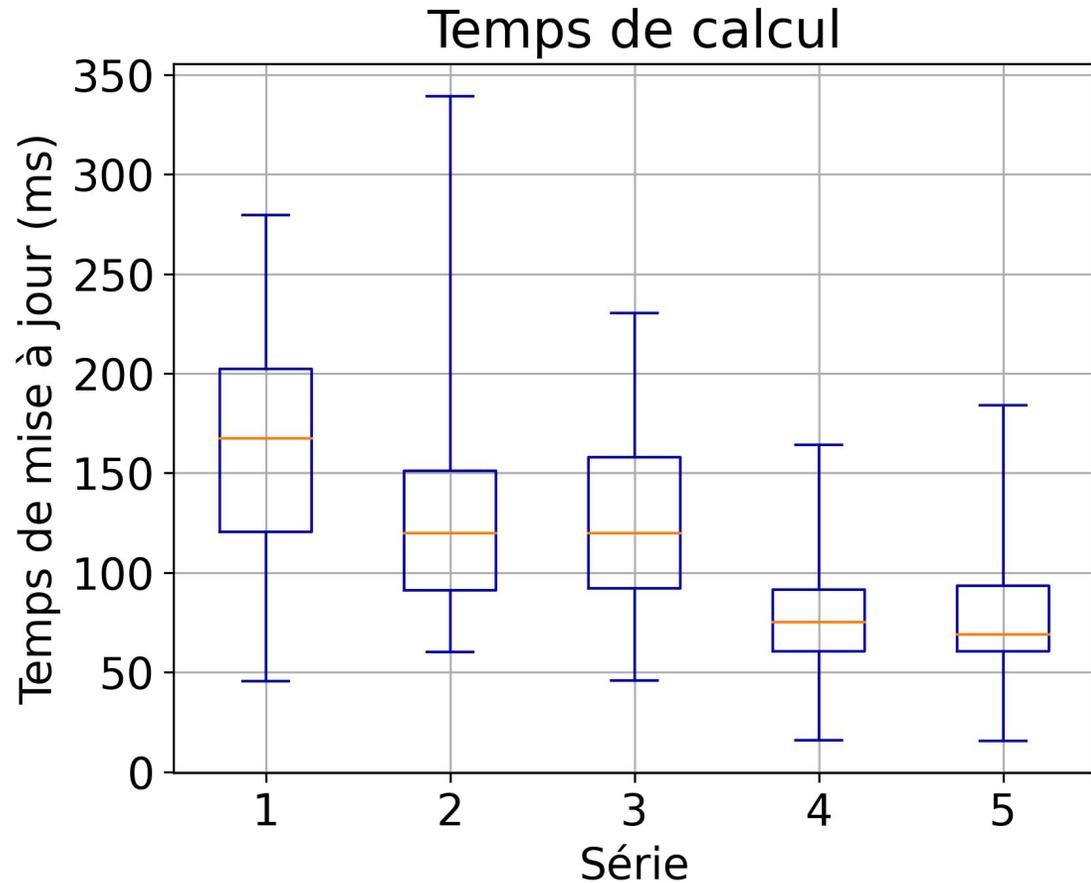
Recalage avec le réseau



- Forces reconstruites plus bruitées avec le réseau
- Le réseau : une approximation du modèle mécanique

3 Un réseau de neurones pour recalibrer plus vite

Recalage avec le réseau



Temps de mise à jour

~ 150 ms

(Méthode d'adjoint classique : 2 min)

Accélération permise par le réseau
x800

Conclusion

- Encore quelques réglages à faire pour améliorer la précision du réseau
- Contrôle optimal avec le réseau : combine la vitesse de calcul du réseau et la flexibilité d'une méthode d'adjoint classique
- Moins de difficultés liées à la résolution itérative
- Une piste intéressante pour une application en chirurgie

Merci pour votre attention !