

# Vers une paramétrisation auto-adaptative pour l'optimisation de forme aérodynamique

R.Duvigneau, B. Abou El Majd & J.-A. Désidéri

INRIA Sophia-Antipolis, Projet OPALE

CANUM 2006, Guidel, Morbihan, 29 mai - 2 juin 2006

# Introduction : optimisation de forme paramétrique

## Définition

- ▶ Choix *a priori* d'une représentation des formes (Bézier, B-Splines, NURBS, etc)
- ▶ Recherche d'une forme optimale dans un sous-espace de dimension finie

## Conséquences

- ▶ Réduction de la dimension de l'espace de recherche (nécessaire pour l'optimisation stochastique)
- ▶ Contrôle de la régularité des formes (nécessaire pour la viabilité du solveur/remailage)
- ▶ La solution dépend de la paramétrisation

# Introduction : optimisation de forme paramétrique

## Définition

- ▶ Choix *a priori* d'une représentation des formes (Bézier, B-Splines, NURBS, etc)
- ▶ Recherche d'une forme optimale dans un sous-espace de dimension finie

## Conséquences

- ▶ Réduction de la dimension de l'espace de recherche (nécessaire pour l'optimisation stochastique)
- ▶ Contrôle de la régularité des formes (nécessaire pour la viabilité du solveur/remailage)
- ▶ **La solution dépend de la paramétrisation**

# Paramétrisation auto-adaptative

## Constat

- ▶ Méthodes de paramétrisation classiques : générales, “multipurpose”
- ▶ Peu adaptées à la résolution d'un problème particulier

## Paramétrisation auto-adaptative

- ▶ Modifier une paramétrisation initiale et naïve
- ▶ Adapter la paramétrisation à un problème particulier
- ▶ S'appuyer sur une première approximation de la forme optimale

## Objectifs d'une paramétrisation auto-adaptative

- ▶ Améliorer la vitesse de convergence (conditionnement)
- ▶ Améliorer la performance finale (s'approcher de l'optimum de dimension infini)

# Paramétrisation auto-adaptative

## Constat

- ▶ Méthodes de paramétrisation classiques : générales, “multipurpose”
- ▶ Peu adaptées à la résolution d'un problème particulier

## Paramétrisation auto-adaptative

- ▶ Modifier une paramétrisation initiale et naïve
- ▶ Adapter la paramétrisation à un problème particulier
- ▶ S'appuyer sur une première approximation de la forme optimale

## Objectifs d'une paramétrisation auto-adaptative

- ▶ Améliorer la vitesse de convergence (conditionnement)
- ▶ Améliorer la performance finale (s'approcher de l'optimum de dimension infini)

# Paramétrisation auto-adaptative

## Constat

- ▶ Méthodes de paramétrisation classiques : générales, “multipurpose”
- ▶ Peu adaptées à la résolution d'un problème particulier

## Paramétrisation auto-adaptative

- ▶ Modifier une paramétrisation initiale et naïve
- ▶ Adapter la paramétrisation à un problème particulier
- ▶ S'appuyer sur une première approximation de la forme optimale

## Objectifs d'une paramétrisation auto-adaptative

- ▶ Améliorer la vitesse de convergence (conditionnement)
- ▶ Améliorer la performance finale (s'approcher de l'optimum de dimension infini)

# Principes

## Algorithme

1. Choix d'une paramétrisation initiale  $\mathcal{P}_0$  ;  $k \leftarrow 0$
2. Optimisation de la forme selon  $\mathcal{P}_k$   
obtention de la forme  $S_k$
3. Adaptation de la paramétrisation :
  - ▶ définition de variables d'adaptation
  - ▶ minimisation d'une fonctionnelle d'adaptation
  - ▶ contrainte : conservation de la forme optimale  $S_k$
4.  $k \leftarrow k + 1$  Goto (2)

## Interprétation : jeu dynamique

- ▶ **Joueur 1 : optimiser la forme** en modifiant les variables de forme (paramétrisation fixe)
- ▶ **Joueur 2 : adapter la paramétrisation** en modifiant les variables d'adaptation (forme fixe)

# Principes

## Algorithme

1. Choix d'une paramétrisation initiale  $\mathcal{P}_0$  ;  $k \leftarrow 0$
2. Optimisation de la forme selon  $\mathcal{P}_k$   
obtention de la forme  $S_k$
3. Adaptation de la paramétrisation :
  - ▶ définition de variables d'adaptation
  - ▶ minimisation d'une fonctionnelle d'adaptation
  - ▶ contrainte : conservation de la forme optimale  $S_k$
4.  $k \leftarrow k + 1$  Goto (2)

## Interprétation : jeu dynamique

- ▶ **Joueur 1 : optimiser la forme** en modifiant les variables de forme (paramétrisation fixe)
- ▶ **Joueur 2 : adapter la paramétrisation** en modifiant les variables d'adaptation (forme fixe)

# Premier cas : problème de reconstruction géométrique

## Définition du problème

On cherche l'arc  $(x(t), y(t)) \quad t \in [0, 1]$ , reliant  $(0,0)$  et  $(1,0)$ , qui minimise :

$$J_{OPT} = \frac{p^\alpha}{\mathcal{A}} \quad (1)$$

où :

$$p = \int_0^1 \sqrt{x'^2(t) + y'^2(t)} \omega(t) dt \quad \mathcal{A} = \int_0^1 y(t) x'(t) \omega(t) dt \quad (2)$$

$\alpha$  réel positif,  $\omega(t)$  une fonction positive et ajustable

## Caractéristiques

- ▶ pour certains  $\alpha$  et  $\omega(t)$  la solution est connue (cercle, profil aéro.)
- ▶ pour certains  $\alpha$  la fonctionnelle est convexe/unimodale

# Premier cas : problème de reconstruction géométrique

## Problème

- ▶ Choix  $\alpha$  et  $\omega(t)$  : reconstruction d'un profil aérodynamique

## Paramétrisation

- ▶ Représentation par une courbe de Bézier

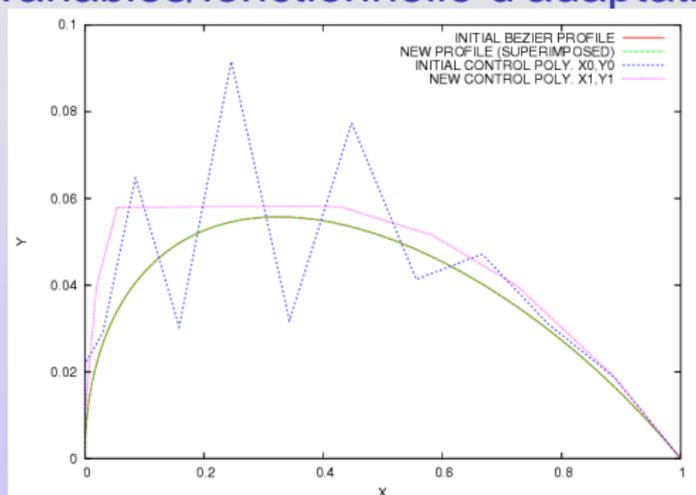
$$P(t) = \sum_{i=0}^n B_n^i(t) P_i \quad (3)$$

- ▶ ordonnées des points de contrôle = variables de forme
- ▶ abscisses des points de contrôle fixes (réguliers)

# Premier cas : choix des variables/fonctionnelle d'adaptation

## Variables d'adaptation

- ▶ abscisses des points de contrôle

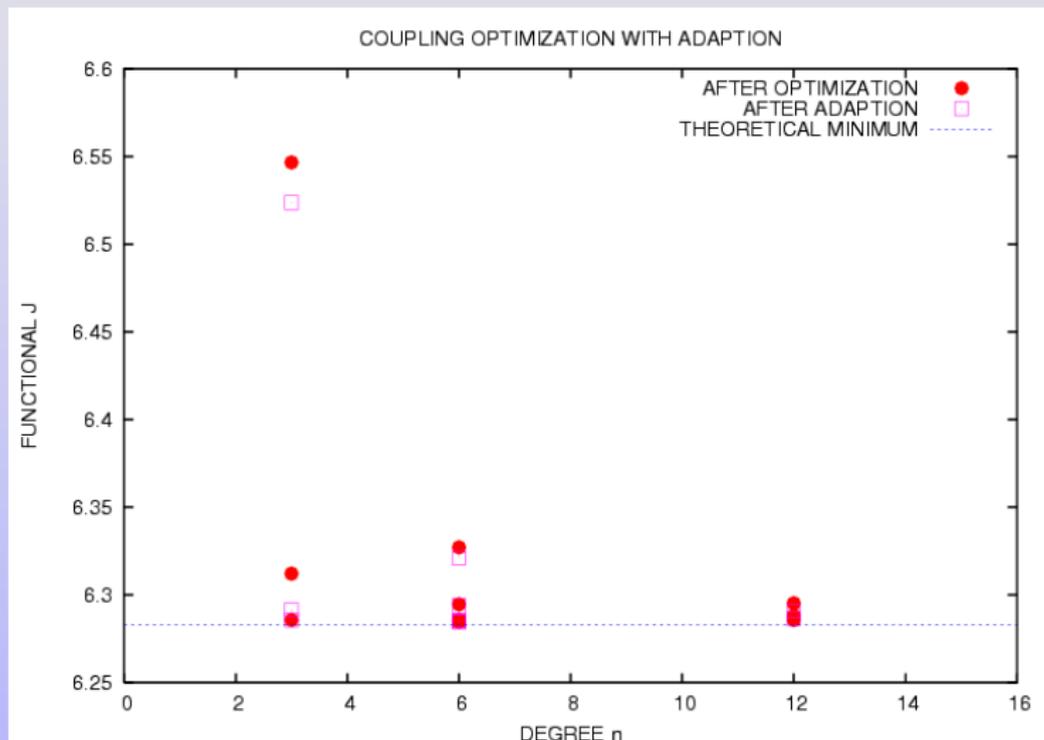


## Fonctionnelle d'adaptation

- ▶ Régulariser le polygone de contrôle
- ▶ critère = variation totale des ordonnées des points de contrôle (approximation au moindres carrés de la forme optimale)

$$TV(\{y_i\}) = \sum_{i=1}^n |y_i - y_{i-1}| \approx \int_0^1 |y'(t)| dt \quad (4)$$

## Premier cas : Comparaison des résultats



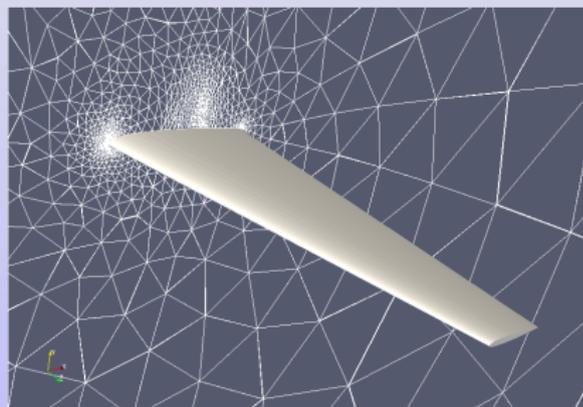
# Second cas : optimisation aérodynamique d'une aile

## Objectif

- ▶ Réduction de la trainée aérodynamique
- ▶ Contrainte de portance

## Analyse

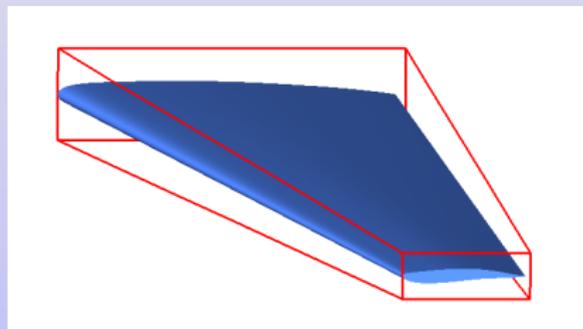
- ▶ Résolution des équations d'Euler
- ▶ Méthode Mixte éléments-volumes finis



## Second cas : optimisation aérodynamique d'une aile

### Paramétrisation par méthode FFD (Free-Form Deformation)

- Définition d'une boîte englobante  
coordonnées internes  $(\xi, \eta, \zeta) \in [0, 1] \times [0, 1] \times [0, 1]$



- déplacement  $\Delta q$  d'un point  $q$  intérieur à la boîte :

$$\Delta q = \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_i^{n_i}(s_q) B_j^{n_j}(t_q) B_k^{n_k}(u_q) \Delta P_{ijk} \quad (5)$$

où  $(s_q, t_q, u_q)$  une carte (*mapping*) de  $(\xi_q, \eta_q, \zeta_q)$

# Second cas : choix des variables/fonctionnelle d'adaptation

## Variables d'adaptation

- ▶ carte produisant  $(s_q, t_q, u_q)$  à partir de  $(\xi_q, \eta_q, \zeta_q)$

$$s = \phi(\xi) \quad t = \psi(\eta) \quad u = \theta(\zeta) \quad (6)$$

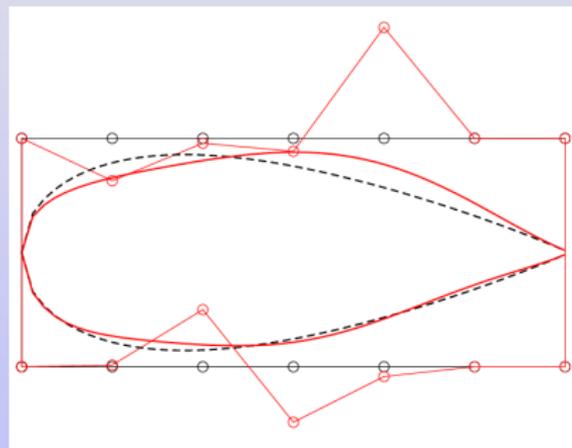
$$\phi(\xi) = \sum_{i=0}^{n'_i} B_i^{n'_i}(\xi)\phi_i \quad \psi(\eta) = \sum_{j=0}^{n'_j} B_j^{n'_j}(\eta)\psi_j \quad \theta(\zeta) = \sum_{k=0}^{n'_k} B_k^{n'_k}(\zeta)\theta_k \quad (7)$$

## Fonctionnelle d'adaptation

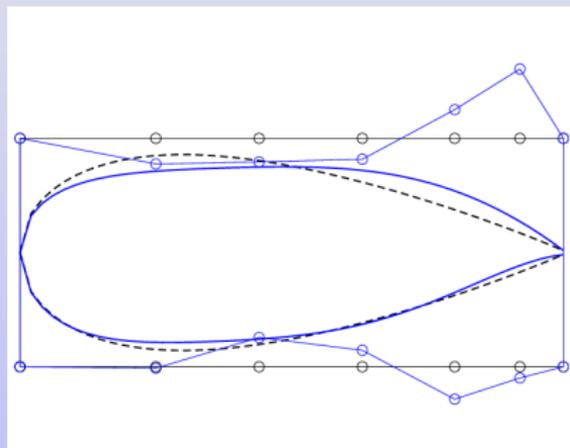
- ▶ Régulariser le champ de déformation

$$J_{AD} = \frac{1}{n_i n_j n_k} \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \left\| \overline{\nabla} \delta P_{ijk} \right\| \approx \int_0^1 \int_0^1 \int_0^1 \left\| \overline{\nabla} \delta P(\xi, \eta, \zeta) \right\| d\xi d\eta d\zeta \quad (8)$$

## Second cas : un exemple 2D

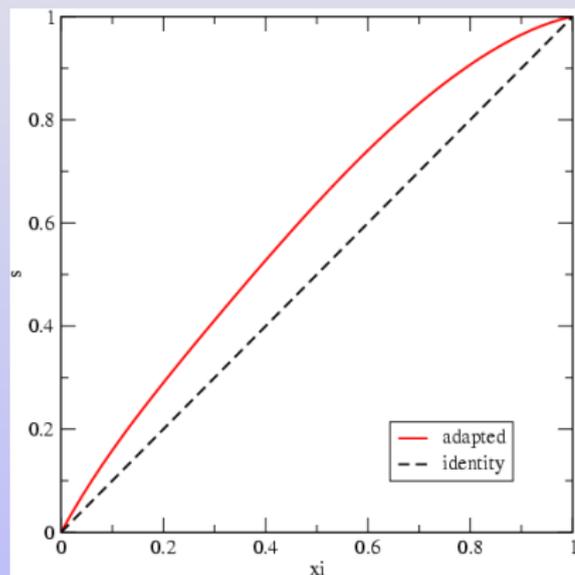


paramétrisation initiale

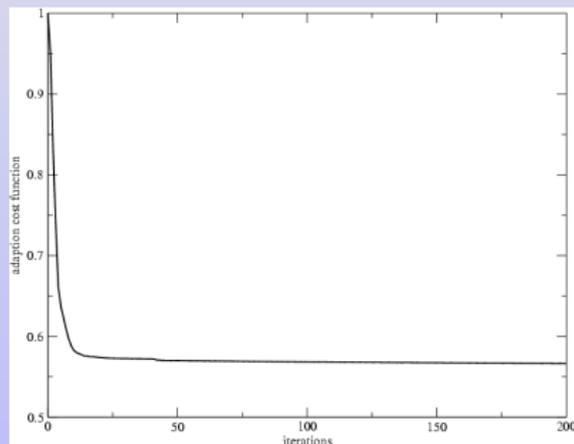


paramétrisation adaptée

## Second cas : un exemple 2D

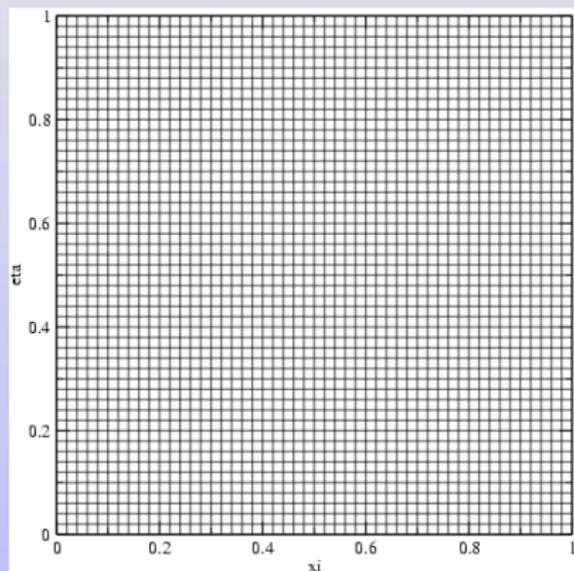


fonction  $s = \phi(\xi)$

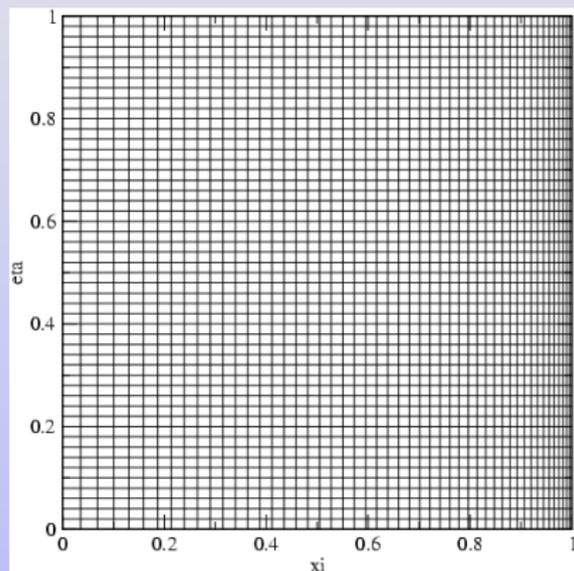


fonctionnelle d'adaptation

## Second cas : un exemple 2D



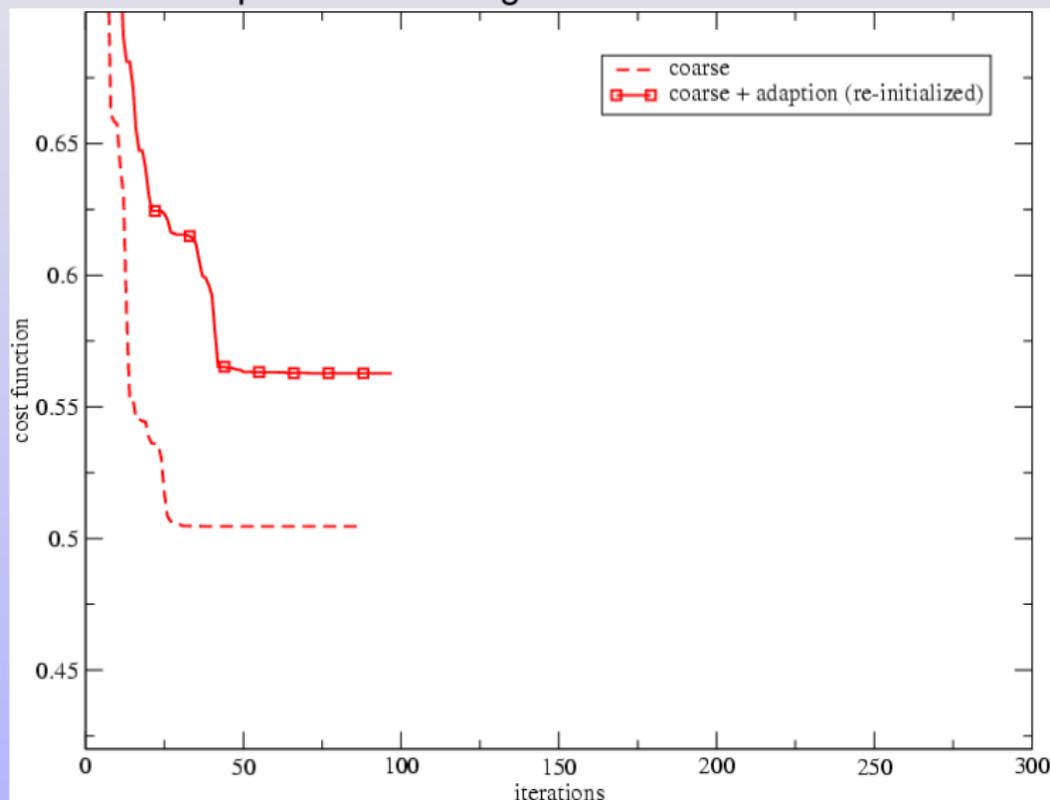
carte initiale : iso-(s,t)



carte adaptée : iso-(s,t)

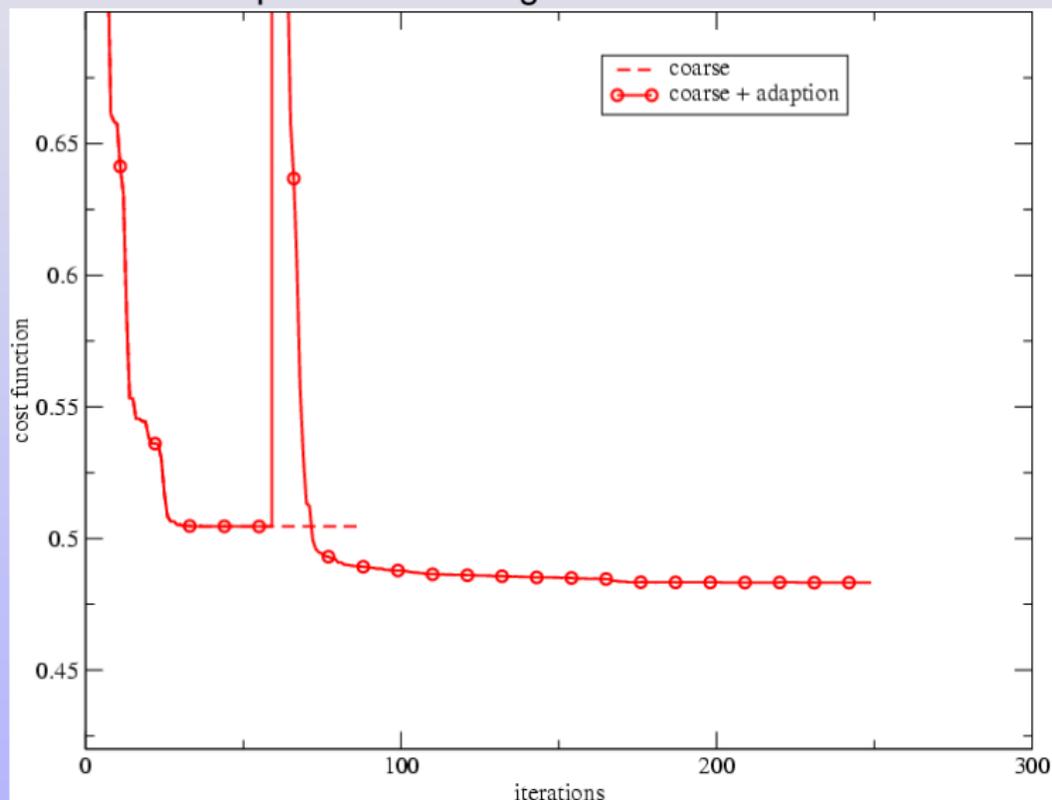
## Second cas : paramétrisation grossière (8 ddl)

Adaptation à convergence et ré-initialisation



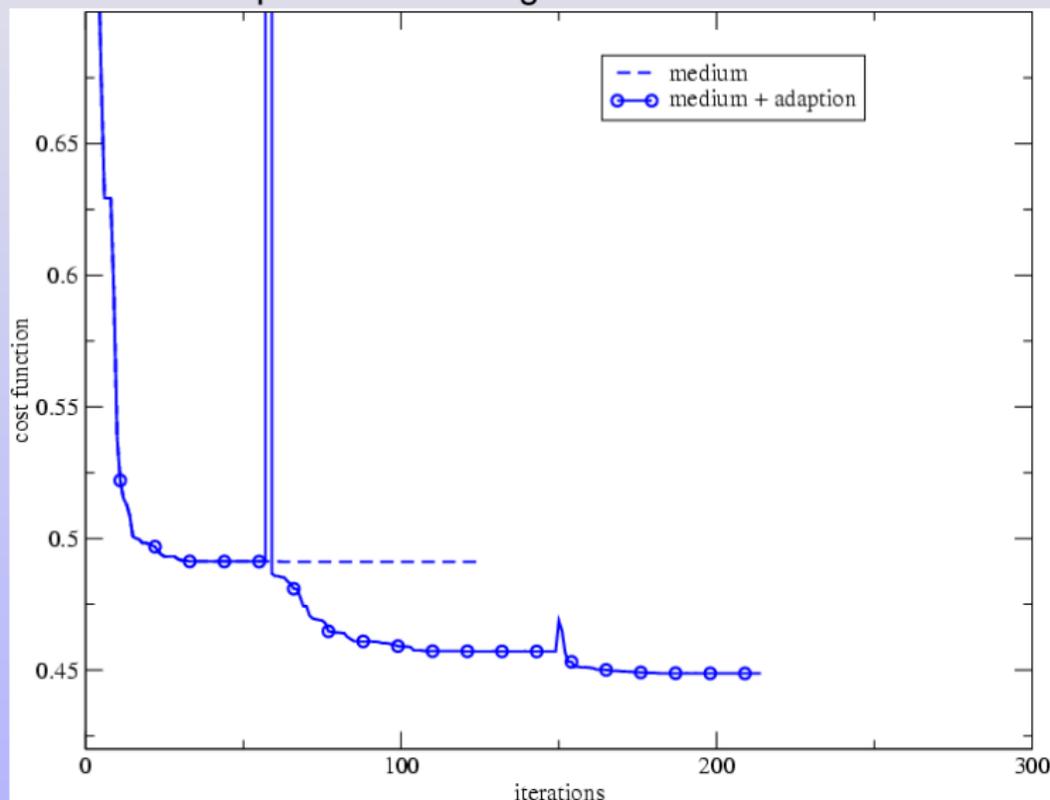
## Second cas : paramétrisation grossière (8 ddl)

Adaptation à convergence et continuation



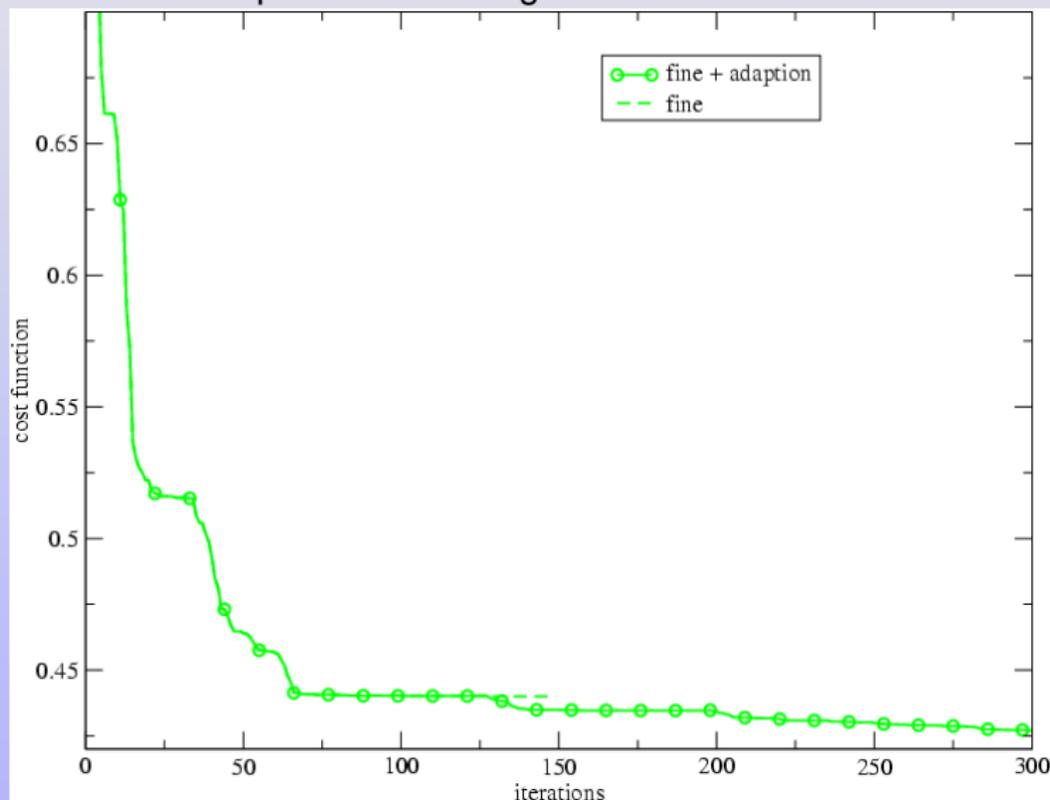
## Second cas : paramétrisation moyenne (20 ddl)

### Adaptation à convergence et continuation



## Second cas : paramétrisation fine (32 ddl)

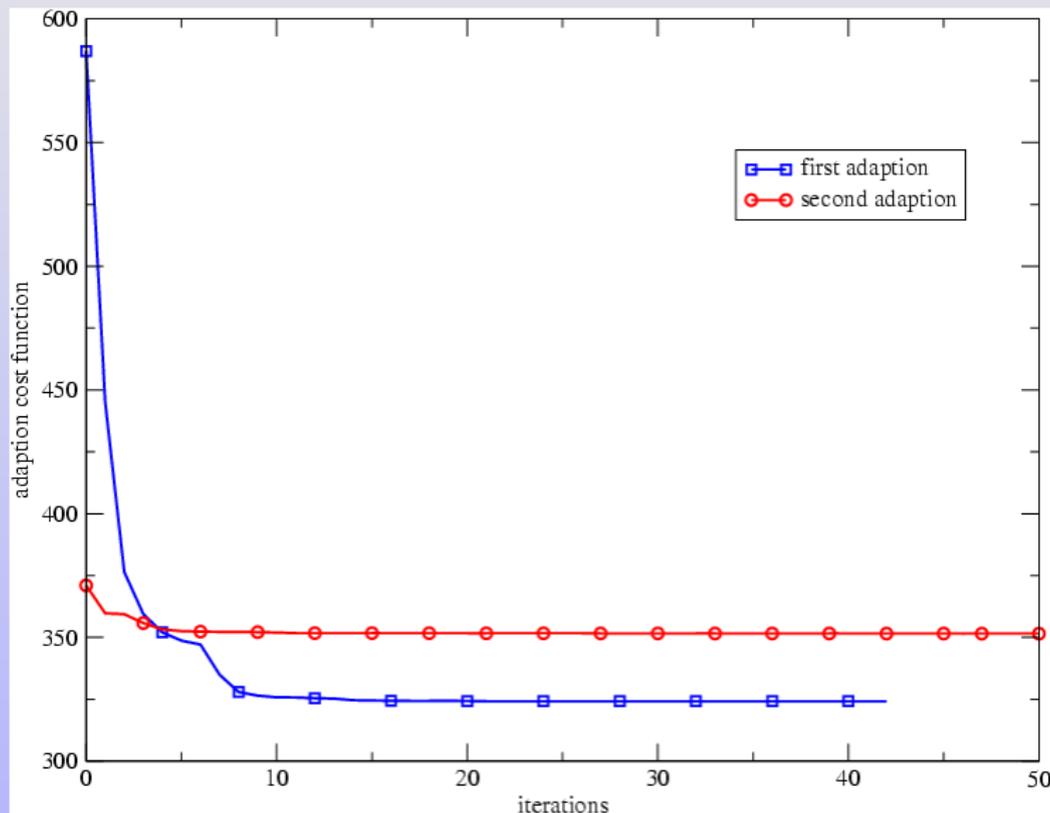
Adaptation à convergence et continuation



## Second cas : comparaison des résultats

paramétrisation	adaptation	fonction coût
grossière (8 d.d.l.)	non	0.5045
grossière (8 d.d.l.)	oui	0.4832
moyenne (20 d.d.l.)	non	0.4911
moyenne (20 d.d.l.)	oui	0.4487
fine (32 d.d.l.)	non	0.4401
fine (32 d.d.l.)	oui	0.4259

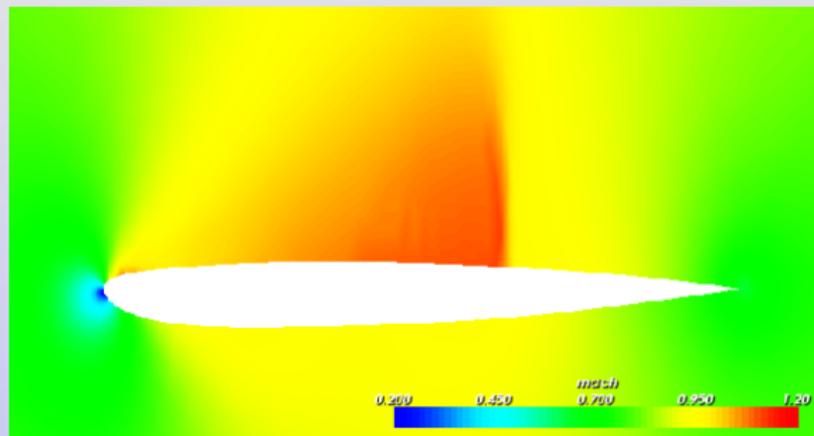
## Second cas : fonctionnelle d'adaptation (moyenne)



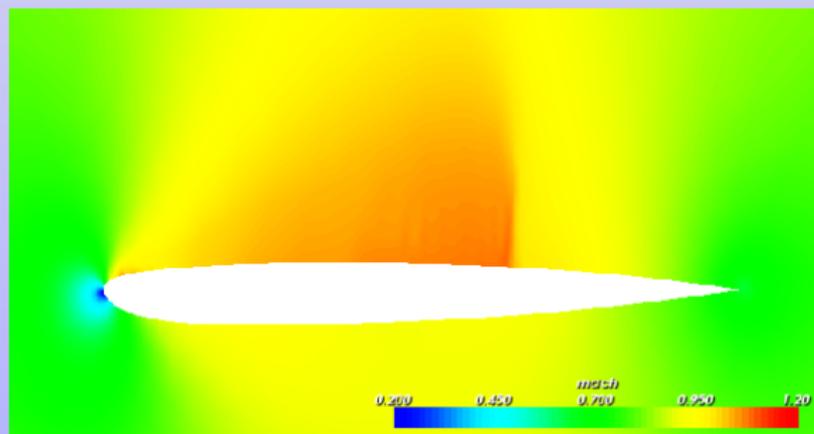
## Second cas : Comparaison des formes (moyenne)



## Second cas : Champs de Mach à la base (moyenne)

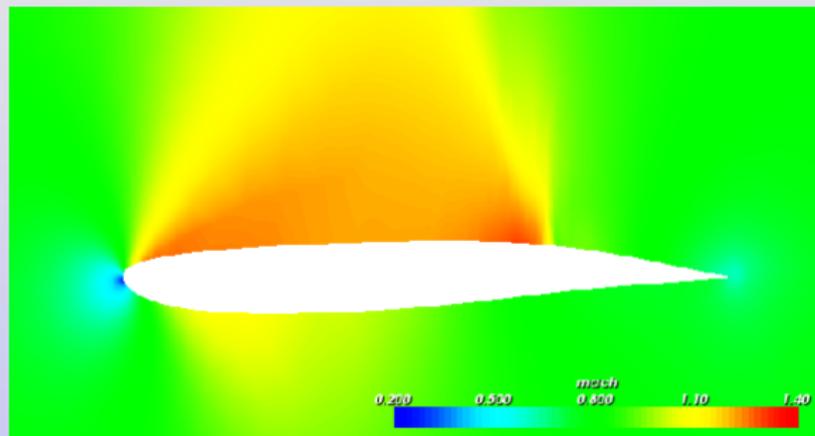


param. initiale

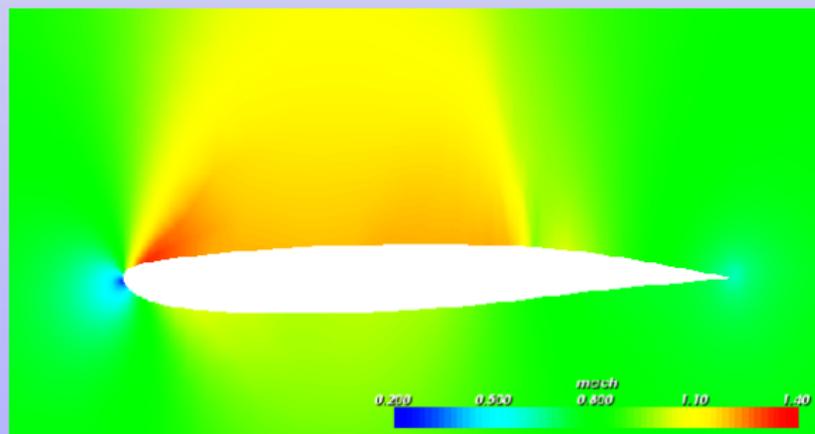


param. adaptée

## Second cas : Champs de Mach à l'extrémité (moyenne)



param. initiale



param. adaptée

# Conclusion et perspectives

## Conclusion

- ▶ Développement d'une procédure d'adaptation de la paramétrisation
- ▶ Résultats encourageants en optimisation de forme aérodynamique
- ▶ Forte sensibilité de la performance optimale aux modifications de la carte

## Conclusion

- ▶ adaptation continue
- ▶ mise en oeuvre avec une optimisation stochastique
- ▶ améliorer la définition de la carte
- ▶ modèle théorique : efficacité de la régularisation