

# Une nouvelle formulation pour le calcul d'adjoint

## ECINADS: Volet DA

### Réunion de travail

Alcin-Hascoet-Dervieux

INRIA

Groupe ECINADS 19 avril 2011

## Contexte continu

- $W$  représente l'écoulement.

- $W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}$

- l'équation de Navier Stokes :  $\Psi(W) = 0$
- $\Psi(W) = W_t + \text{div} \mathcal{F}(W)$  sur  $\Omega \times ]0, T[$  + cond. bords et initiales
- La fonctionnelle  $j = (g, W)$ ,  $g = g(x, t)$

- $W_h$  représente la solution discrète.
- l'équation de Navier Stokes discrétisée s'écrit :  $\Psi_h(W_h) = 0$

Trouver un maillage fixe “optimal”  $\mathcal{M}_{opt}$

- Minimisant l'erreur  $\delta j(\mathcal{M}) = |j(\mathbf{W}) - j(\mathbf{W}_{\mathcal{M}})|$ .

Définition du maillage  $M$  continu sur un domaine  $\Omega \subset R^3$

$$M = (\mathcal{M}(x))_{x \in \Omega}$$

Avec  $\mathcal{M}(x)$  une forme quadratique définie positive  
 $M$  est donc une métrique Riemannienne.

Trouver le maillage optimal consiste à trouver la métrique optimale.

## Lemme :

La métrieque optimale s'exprime en fonction des états direct et adjoint:

$$\mathcal{M}_{opt} = \text{Fonct}(W, W^*)$$

$W^*$  est l'état adjoint, solution de:

$$\left(\frac{\partial \Psi}{\partial W} \Big|_W\right)^* W^* = \left(\frac{\partial J}{\partial W} \Big|_W\right)^*$$

En pratique,  $W$  et  $W^*$  sont approchés par leurs discrétisés  $W_h$  et  $W_h^*$

# Métrique optimale dans des équations d'Euler

Posons  $\mathbf{H}(\mathbf{x}) = \sum_{n=1}^m \sum_{j=1}^5 ([\Delta t]_j(\mathbf{x})) + [\Delta x]_j(\mathbf{x}) + [\Delta y]_j(\mathbf{x}) + [\Delta z]_j(\mathbf{x})$ , alors:

$$[\Delta t]_j(\mathbf{x}) = \int_0^T |W_j^*(\mathbf{x}, t)| \cdot |H((W_{j,t}))(\mathbf{x}, t)| dt, \quad (1)$$

$$[\Delta x]_j(\mathbf{x}) = \int_0^T \left| \frac{\partial W_j^*}{\partial x}(\mathbf{x}, t) \right| \cdot |H(\mathcal{F}_1(W_j))(\mathbf{x}, t)| dt, \quad (2)$$

$$[\Delta y]_j(\mathbf{x}) = \int_0^T \left| \frac{\partial W_j^*}{\partial y}(\mathbf{x}, t) \right| \cdot |H(\mathcal{F}_2(W_j))(\mathbf{x}, t)| dt, \quad (3)$$

$$[\Delta z]_j(\mathbf{x}) = \int_0^T \left| \frac{\partial W_j^*}{\partial z}(\mathbf{x}, t) \right| \cdot |H(\mathcal{F}_3(W_j))(\mathbf{x}, t)| dt. \quad (4)$$

où  $W_j^*$  est la  $j^{me}$  composante du vecteur adjoint  $W^*$  et  $H(\mathcal{F}_i(W_j))$  le hessien de la  $j^{ime}$  composante du vecteur  $\mathcal{F}_i(W)$ .

## La métrieque optimale dans le cas d'Euler

(cf. Anca et al 2011 pour le cas Euler)

$$\mathcal{M}_{opt}(\mathbf{x}) = C \det(|\mathbf{H}(\mathbf{x})|)^{-\frac{1}{5}} |\mathbf{H}(\mathbf{x})|$$

où la constante  $C$  dépendant de  $N$ :

$$C = N^{\frac{2}{3}} \left( \int_{\Omega} \det(|\mathbf{H}(\mathbf{x})|)^{\frac{1}{5}} d\mathbf{x} \right)^{\frac{2}{3}}.$$



On se propose de décrire comment l'état adjoint discret va être calculé à l'aide d'un logiciel dérivé (différentié) du logiciel de simulation.

L'état adjoint  $W_h^*$  est défini par la résolution du système linéaire:

$$\left(\frac{\partial \Psi_h}{\partial W_h} \Big|_{(\kappa, W_h)}\right)^* W_h^* = \left(\frac{\partial J}{\partial W_h} \Big|_{(\kappa, W_h)}\right)^*$$

De manière classique, le logiciel de simulation calcule l'état  $W_h$ , solution de:

$$\Psi_h(W_h) = 0.$$

On utilise un artifice reposant sur un nouveau paramètre  $\kappa$ :

$$j_h(\kappa) = (g, W_h(\kappa))$$
$$\bar{\Psi}_h = \Psi_h(\kappa, W_h(\kappa)) + \kappa$$

- $\frac{\partial j}{\partial \kappa} = (g, \frac{\partial W(\kappa)}{\partial \kappa})$
- Par le théorème des fonctions implicites:

$$\frac{dW(\kappa)}{d\kappa} = - \frac{\Psi(\kappa, W(\kappa))^{-1} \Psi(\kappa, W(\kappa))}{\partial W}$$

donc

$$\frac{\partial j}{\partial \kappa} = (g, - \frac{\Psi(\kappa, W(\kappa))^{-1} \Psi(\kappa, W(\kappa))}{\partial W})$$

- En posant  $\bar{\Psi} = \Psi + \kappa$  on a

$$\frac{\partial j}{\partial \kappa} \cdot \delta = (g, \frac{\partial W}{\partial \kappa} \delta) = - (\frac{\partial \bar{\Psi}}{\partial W}^{-*} g, \frac{\partial \bar{\Psi}}{\partial \kappa} \delta)$$

$$\text{or } W^* = \frac{\partial \Psi}{\partial W}^{-*} g = \frac{\partial \bar{\Psi}}{\partial W}^{-*} g \text{ d'où}$$

$$\frac{\partial j}{\partial \kappa} \cdot \delta = -(W^*, \delta)$$

## Lemme :

avec :

$$j_h(\kappa) = (g, W_h(\kappa))$$

$$\Psi_h(\kappa, W_h(\kappa)) = \kappa$$

On a:

$$w_h^* = -\left. \frac{dj_h}{d\kappa} \right|_{\kappa=0}.$$

## La top routine

On a besoin d'une routine, la "top routine" ayant:

- $g$  comme paramètre,
- $\kappa$  comme paramètre à différentier, "independent input variable",
- $j_h = (g, W_h)$  comme sortie "dependent output variable".

Supposons que l'état  $W$  soit constitué de  $ktmax$  champs instantanés chacun de dimension  $ns \times 5$

- le paramètre  $g$  est de dimension  $ktmax \times ns \times 5$ .
- $\kappa$  est de dimension  $ktmax \times ns \times 5$ .
- l'adjoint  $W_h^*$  est de dimension  $ktmax \times ns \times 5$ .
- l'état  $W_h$  devra être stocké, il est de dimension  $ktmax \times ns \times 5$ .

## La différentiation automatique

- Elle différencie les programmes.
- Elle identifie les programmes avec des compositions de fonctions mathématiques.

Tout programme  $P$  composé d'une suite d'instructions  $I_k$ ,  $k \in [1 \dots p]$  et qui implémente une fonction  $f$  de  $R^m$  dans  $R^n$

$$f = f_p \circ f_{p-1} \circ \dots \circ f_1$$

$f_k$  représente la fonction élémentaire implémentée par l'instruction  $I_k$ .

- Le mode direct:

$$f'(x) = (f'_p \circ f_{p-1} \circ f_{p-2} \circ \cdots \circ f_1(x)) \\ \cdot (f'_{p-1} \circ f_{p-2} \circ \cdots \circ f_1(x)) \\ \cdots \\ f'_1(x)$$

- Le mode reverse:

$$f'^t(x) = f_1'^t(x) \\ f_2'^t \circ f_1(x) \\ \cdots \\ (f_p'^t \circ f_{p-1} \circ f_{p-2} \circ \cdots \circ f_1(x))$$

Si  $f$  est de dimension 1, le mode reverse nous donne le gradient de  $f$ . C'est le mode que nous choisissons.



Choix du mode inverse :

- $j_h$  va de  $R^p$  dans  $R$  de même que sa différentielle appliquée en un point.
- Le calcul de la différentielle de l'adjoint de  $j_h$  est judicieux puisque qu'elle va de  $R^p$  dans  $R$ .
- Par linéarité on a

$$Dj_h^*(a).h = h(Dj_h^*(a).1)$$

Conclusion : pour connaître la différentielle de l'adjoint de  $\Psi$

Il suffit de connaître  $D\Psi^*(a).1$

$$j(kappa) = (g, w(kappa))$$

On se propose donc d'abord construire :

$$kappa \mapsto w(kappa) \mapsto j(kappa)$$

avec un avancement explicite.

Cela comprend donc la timeloop.

Qui comprend l'évaluation du pas de temps.

On commence par l'ordre un explicite RK1 ordre un.

Cas test possible de validation: onde acoustique dans un tube “a choc” TUBE909 (BackupAisu/MESH3D/TUBE909) Comparaison possible avec Wolf.

Etape 1: Stockage complet des resultats intermediaires, traitement des problemes MPI.

Etape 1bis: passage a l'ordre deux.

Etape 2: Stockage-recalcul.

Etape 3: Envisager l'implicite.

## Méthode mixte differentiation/non-differentiation

$$\text{delta}W^{n+1} = \text{Solve}(\text{Mat}(W^n), \text{FLUX}(W^n), \text{Pre})$$

ou  $\text{Solve}(\text{Mat}, \text{Flu})$  trouve itérativement avec un preconditionnement  $\text{Pre}$  la solution  $\text{delta}W$  de  $\text{Mat} \text{delta}W = \text{Flu}$ :

$$\text{delta}W = \text{Mat}^{-1} \text{Flu}$$

Mais on cherche à dériver cela:

$$\text{Mat}(\text{delta}W + dW) - (\text{Flu} + d\text{Flu}) \implies dW = \text{Mat}^{-1} d\text{Flu}$$

Dérivation directe:

$$\text{Solve}_d(\text{Mat}, \text{Flu}, \text{Mat}_d, \text{Flu}_d, \text{Pre}) = \text{Solve}(\text{Mat}_d, \text{Flu}_d, \text{Pre})$$

Dérivation adjointe:  $\text{Pre}^*$  adjoint de  $\text{Pre} = \text{ILU}$ :

$$\text{Solve}_b(\text{Mat}, \text{Flu}, \text{Mat}_b, \text{Flu}_b, \text{Pre}) = \text{Solve}(\text{Mat}_b, \text{Flu}_b, \text{Pre}^*)$$

On a identifié les sous-programmes à différentier.

On a proposé une nouvelle formulation pour tirer un adjoint à partir d'une dérivée du code direct.

Ces approches seront appliquées au code parallèle AIRONUM.

Thank you