

Réunion ANR - 3 avril 2012

Bruno Koobus, Carine Moussaed
(en coopération avec l'INRIA, LEMMA et l'univ.de Pise)

Institut de Mathématiques et de Modélisation de Montpellier



Institut de Mathématiques
et de Modélisation
de Montpellier



Plan

- 1 Préconditionnement par déflation (et balancing) : utilisation de la bibliothèque MUMPS dans AIRONUM (avec H. Alcin)
- 2 Déflation 3-niveaux : description de l'algorithme et état des lieux dans AIRONUM
- 3 Modélisation de la turbulence : VMS-LES dynamique, hybride RANS/VMS-LES dynamique (avec A. Dervieux et M.-V. Salvetti)
- 4 Mise en oeuvre du cas test du cylindre circulaire : Reynolds 20K, 500K-3M (avec S. Wornom)

Plan

- 1 Préconditionnement par déflation (et balancing) : utilisation de la bibliothèque MUMPS dans AIRONUM (avec H. Alcin)
- 2 Déflation 3-niveaux : description de l'algorithme et état des lieux dans AIRONUM
- 3 Modélisation de la turbulence : VMS-LES dynamique, hybride RANS/VMS-LES dynamique (avec A. Dervieux et M.-V. Salvetti)
- 4 Mise en oeuvre du cas test du cylindre circulaire : Reynolds 20K, 500K-3M (avec S. Wornom)

Solveur linéaire : option de base

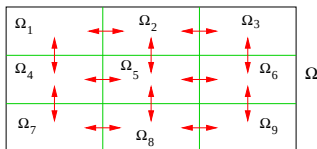
Option de base dans AIRONUM pour résoudre $\mathbf{A}\mathbf{u} = \mathbf{b}$ où A matrice de discrétisation implicite des équations de Navier-Stokes :

après mise à l'échelle (préconditionnement à gauche par D^{-1} où D =diagonale(A)), **GMRES** preconditionné à droite par **Schwarz additif restreint** avec résolution des problèmes locaux par **ILU(0)** :

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{M}^{-1}\mathbf{v} = \mathbf{D}^{-1}\mathbf{b} \text{ avec } \mathbf{M}^{-1} = \sum_{i=1}^N \mathbf{R}_i^0 (\mathbf{D}^{-1}\mathbf{A})_i^{-1} \mathbf{R}_i$$

$$\text{et } (\mathbf{D}^{-1}\mathbf{A})_i^{-1} = ((\mathbf{R}_i(\mathbf{D}^{-1}\mathbf{A})\mathbf{R}_i)_{|\Omega_i})^{-1} \approx (\mathbf{ILU}_i)^{-1}$$

avec $\mathbf{u} = \mathbf{M}^{-1}\mathbf{v}$;



Défaut, solutions considérées

- ⇒ **Mauvaise scalabilité** du solveur de base : dégradation du taux de convergence avec le nombre de sous-domaines
- ⇒ Introduire une ou plusieurs **grilles grossières**
- ⇒ Options considérées : preconditionnement par **déflation** (ou **balancing**).

Option A : Déflation

Problème : trouver $\mathbf{u}$ solution de $\mathbf{D}^{-1}\mathbf{A}\mathbf{u} = \mathbf{D}^{-1}\mathbf{b}$

Opérateurs de base :

- Φ_j = fonctions de base caractéristiques associées au sous-domaine Ω_j ($\Phi_j(x_i) = 1$ si $x_i \in \Omega_j$, 0 sinon).
- $\mathbf{Z}$ = opérateur de prolongement défini par $\mathbf{Z}_{ij} = \Phi_j(\mathbf{x}_i)$
($i=1, \dots, NS$ et $j=1, \dots, N$, $NS=nb$ noeuds, $N=nb$ sous-domaines, $NS \gg N$).
- $\mathbf{E} = \mathbf{Z}^T \mathbf{D}^{-1} \mathbf{A} \mathbf{Z}$ = matrice grille grossière
- opérateurs de projection :
 - $\mathbf{P} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A} \mathbf{Z} \mathbf{E}^{-1} \mathbf{Z}^T$
 - $\mathbf{Q} = \mathbf{I} - \mathbf{Z} \mathbf{E}^{-1} \mathbf{Z}^T \mathbf{D}^{-1} \mathbf{A}$

Remarque pour preconditionner à droite : $\mathbf{P} \mathbf{D}^{-1} \mathbf{A} = \mathbf{D}^{-1} \mathbf{A} \mathbf{Q}$

Option A : Déflation (precond. à droite)

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{u} = \mathbf{D}^{-1}\mathbf{b}$$

$$\Rightarrow \mathbf{u} = (\mathbf{I} - \mathbf{Q})\mathbf{u} + \mathbf{Q}\mathbf{u} = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T\mathbf{b} + \mathbf{Q}\tilde{\mathbf{u}}$$

$$\text{avec } \tilde{\mathbf{u}} = \mathbf{M}^{-1}\tilde{\mathbf{v}}$$

$$\text{où } \mathbf{M}^{-1} = \sum_{i=1}^N \mathbf{R}_i^0 (\mathbf{D}^{-1}\mathbf{A})_i^{-1} \mathbf{R}_i$$

$$\text{et } \tilde{\mathbf{v}} \text{ solution de } \mathbf{D}^{-1}\mathbf{A}\mathbf{Q}\mathbf{M}^{-1}\tilde{\mathbf{v}} = \mathbf{P}\mathbf{D}^{-1}\mathbf{b}$$

Option B : Balancing

Problème : trouver $\mathbf{u}$ solution de $\mathbf{D}^{-1}\mathbf{A}\mathbf{u} = \mathbf{D}^{-1}\mathbf{b}$

Opérateurs de base :

- Φ_j = fonctions de base caractéristique associée au sous-domaine Ω_j ($\Phi_j(x_i) = 1$ si $x_i \in \Omega_j$, 0 sinon).
- $\mathbf{Z}$ = opérateur de prolongement défini par $\mathbf{Z}_{ij} = \Phi_j(\mathbf{x}_i)$ ($i=1, \dots, NS$ et $j=1, \dots, N$, $NS=\text{nb noeuds}$, $N=\text{nb sous-domaines}$, $NS \gg N$).
- $\mathbf{E} = \mathbf{Z}^T \mathbf{D}^{-1} \mathbf{A} \mathbf{Z}$ = matrice grille grossière
- opérateurs de projection :
 - $\mathbf{P} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A} \mathbf{Z} \mathbf{E}^{-1} \mathbf{Z}^T$
 - $\mathbf{Q} = \mathbf{I} - \mathbf{Z} \mathbf{E}^{-1} \mathbf{Z}^T \mathbf{D}^{-1} \mathbf{A}$

Option B : Balancing (precond. à droite)

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{u} = \mathbf{D}^{-1}\mathbf{b}$$

$$\Rightarrow \mathbf{u} = \mathbf{P}_B \mathbf{v}$$

$$\text{où } \mathbf{P}_B = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T + \mathbf{Q}\mathbf{M}^{-1}\mathbf{P}$$

$$\text{et } \mathbf{v} \text{ vérifie } \mathbf{D}^{-1}\mathbf{A}\mathbf{P}_B \mathbf{v} = \mathbf{D}^{-1}\mathbf{b}$$

Utilisation de la bibliothèque MUMPS

- MUMPS : Solveur direct parallèle pour la résolution des systèmes creux, développé à l'ENS Lyon + ENSEEIHT Toulouse.
- Motivation : Réduire le coût de la résolution du problème grossier dans les méthodes de déflation et balancing (utilisation de Gauss-Jordan dans la 1ère version).
- Pourquoi MUMPS : Solveur efficace + solveur parallèle.
- Points faibles : Portabilité, compilation non triviale (utilisation de BLAS, BLACS, ScaLAPACK, SCOTCH, METIS).

Implémentation de MUMPS dans AIRONUM

MUMPS repose sur 3 étapes principales :

- **Phase d'analyse**
- **Phase de factorisation LU**
- **Phase de résolution**

(+ étape d'initialisation + étape de clôture).

Implémentation de MUMPS dans AIRONUM

- **Initialisation** de MUMPS avant la boucle en temps :

mumps_par%COMM = MPI_COMM_WORLD (def. du communicateur MPI pour MUMPS)

mumps_par%JOB = -1 (phase d'initialisation)

mumps_par%SYM = 0 (matrice non symétrique)

mumps_par%PAR = 1 (le processeur 0 participe aussi à la factorisation+résolution)

Implémentation de MUMPS dans AIRONUM

Une seule fois par itération en temps :

- **Phase d'analyse et de factorisation LU :**

On définit la matrice du système grossier :

$mumps_par\%A(k) = MumpsE(k)$ (k-ième coef. non nul de E)

$mumps_par\%IRN(k) = MumpsI(k)$ (indice i du coef. non nul)

$mumps_par\%JCN(k) = MumpsJ(k)$ (indice j du coef. non nul)

puis appel de MUMPS :

$mumps_par\%JOB = 1$

$CALL DMUMPS(mumps_par)$ (analyse, phase préparatoire)

$mumps_par\%JOB = 2$

$CALL DMUMPS(mumps_par)$ (factorisation LU)

Implémentation de MUMPS dans AIRONUM

Pour résoudre chaque système grossier apparaissant dans le preconditionnement de GMRES :

- **Phase de résolution :**

On définit le RHS du système grossier :

$$mumps_par\%RHS((ithd-1)*nvar+i) = W(i,ithd)$$

puis appel de MUMPS :

$$mumps_par\%JOB = 3$$

CALL DMUMPS(mumps_par) (résolution)

Implémentation de MUMPS dans AIRONUM

- **Clôture** de MUMPS après la boucle en temps :

```
mumps_par%JOB = -2  
CALL DMUMPS(mumps_par)
```

Déflation avec l'utilisation de MUMPS : expériences

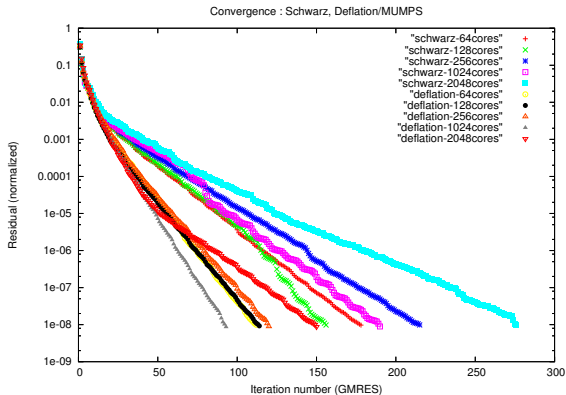
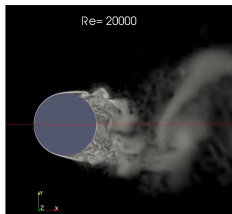
- Mise en oeuvre informatique du préconditionneur déflation + MUMPS dans le code de calcul parallèle AIRONUM d'écoulements compressibles.
- Résolution de $\mathbf{A}\mathbf{u} = \mathbf{b}$ où $\mathbf{A}$ est la matrice de discrétisation associée à un calcul turbulent (cylindre circulaire, LES Smagorinsky, maillage 1.8M de noeuds, aspect ratio < 500) pour un temps donné et pour CFL 100 et CFL 1000.
- Utilisation de 64 à 2048 coeurs sur la SGI Altix ICE JADE du CINES $\Rightarrow$ quelques difficultés techniques sur plus de 1000 coeurs avec la *Direct Access Programming Library (DAPL)*, actuellement résolues (merci à l'équipe du CINES!) :

Modif. du script PBS pour qu'AIronum tourne sur ≥ 1000 coeurs :

```
##PBS -S /bin/ksh
##PBS -N MUMPSStwolevel
##PBS -l walltime=02 :00 :00
##PBS -l select=256 :ncpus=8 :mpiprocs=8
##PBS -e test.err
##PBS -o test.log
cat $PBS_NODEFILE
cd $PBS_O_WORKDIR
## Chargement environnement module pour ksh
. /etc/profile.d/modules.sh
module unload MPT
module load intelmpi/4.0.1.007
cat $PBS_NODEFILE | uniq | mpd.hosts
mpdcleanup -f mpd.hosts -rsh=ssh
## Demarrage du daemon mpd de intelmpi
mpdboot -rsh=ssh -v -n 'cat mpd.hosts|wc -l' -f mpd.hosts
## Recuperation du nombre de coeurs utilises
nprocs=$(cat $PBS_NODEFILE|wc -l)
rm -f $PBS_O_WORKDIR/*.F
rm -f $PBS_O_WORKDIR/*.f
rm -f $PBS_O_WORKDIR/*.cpp
## Pour resoudre les pbs de fichiers non trouves lors de l'execution
export LD_PRELOAD=/work/SGI/misc/getcwd_wrapper.so
export I_MPI_MPD_RSH=ssh
export I_MPI_USE_DYNAMIC_CONNECTIONS=0
export I_MPI_FABRICS_LIST="ofa,dapl,tcp,tmi"
export I_MPI_DEBUG=5
export I_MPI_FALLBACK_DEVICE=1
mpiexec -np $nprocs $PBS_O_WORKDIR/airo.x
##Arret des daemons mpd
mpdallexit
test.log > impression.out
```

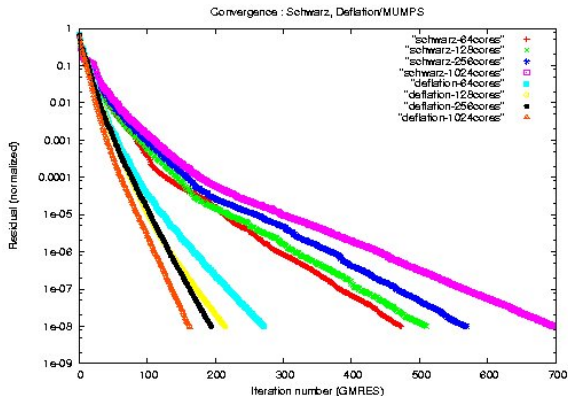
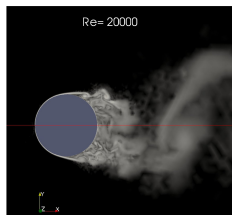
Déflation avec l'utilisation de MUMPS : expériences

LES, cyl. circulaire, $Re=20000$, 1.8M noeuds, aspect ratio < 500 , CFL=100.



Déflation avec l'utilisation de MUMPS : expériences

LES, cyl. circulaire, $Re=20000$, 1.8M noeuds, aspect ratio < 500 ,
CFL=1000.



Plan

- 1 Préconditionnement par déflation (et balancing) : utilisation de la bibliothèque MUMPS dans AIRONUM (avec H. Alcin)
- 2 Déflation 3-niveaux : description de l'algorithme et état des lieux dans AIRONUM
- 3 Modélisation de la turbulence : VMS-LES dynamique, hybride RANS/VMS-LES dynamique (avec A. Dervieux et M.-V. Salvetti)
- 4 Mise en oeuvre du cas test du cylindre circulaire : Reynolds 20K, 500K-3M (avec S. Wornom)

Déflation 3-niveaux

Motivation :

- Le solveur local n'est pas exact (ILU)
⇒ dégradation de la convergence quand beaucoup de noeuds par sous-domaine
- Remède : ajout d'une grille grossière dans chaque sous-domaine

Déflation 3-niveaux

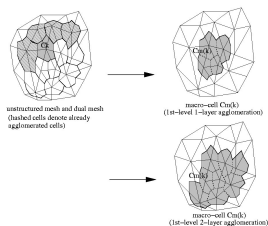
Problème : trouver $\mathbf{u}$ solution de $\mathbf{D}^{-1}\mathbf{A}\mathbf{u} = \mathbf{D}^{-1}\mathbf{b}$

Opérateurs de base associés à chaque sous-domaine Ω_i :

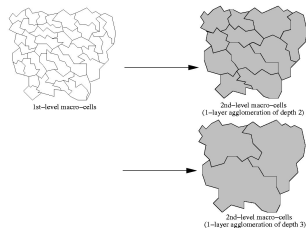
- Partitionnement des sous-domaines : $\Omega_i = \bigcup_{k=1}^{N_i} \Omega_{ik}, \bigcap_{k=1}^{N_i} \Omega_{ik} = \emptyset$
- Φ_{ik} = fonctions de base caractéristiques associées au sous-ensemble Ω_{ik} (pour $x_j \in \Omega_i$, $\Phi_{ik}(x_j) = 1$ si $x_j \in \Omega_{ik}$, 0 sinon).
- $\mathbf{Z}_i$ = opérateur de prolongement défini par $(\mathbf{Z}_i)_{jk} = \Phi_{ik}(\mathbf{x}_j)$ ($i=1, \dots, N$, $j=1, \dots, NS_i$ et $k=1, \dots, N_i$, où $N=\text{nb}$ sous-domaines, $NS_i=\text{nb}$ noeuds dans Ω_i et $N_i=\text{nb}$ sous-ensembles Ω_{ik} dans Ω_i , $N_i \approx 20 - 30$).
- $\mathbf{E}_i = \mathbf{Z}_i^T (\mathbf{D}^{-1}\mathbf{A})|_{\Omega_i} \mathbf{Z}_i$ = matrice grille grossière intermédiaire
- opérateurs de projection :
 - $\mathbf{P}_i = \mathbf{I}|_{\Omega_i} - (\mathbf{D}^{-1}\mathbf{A})|_{\Omega_i} \mathbf{Z}_i \mathbf{E}_i^{-1} \mathbf{Z}_i^T$
 - $\mathbf{Q}_i = \mathbf{I}|_{\Omega_i} - \mathbf{Z}_i \mathbf{E}_i^{-1} \mathbf{Z}_i^T (\mathbf{D}^{-1}\mathbf{A})|_{\Omega_i}$

Déflation 3-niveaux : choix de la macro-agglomération

(A)



(B)



Choix de la macro-agglomération (B) pour partitionner les sous-domaines Ω_i

Déflation 3-niveaux

Algorithme déflation 3-niveaux :

Déflation sur la grille grossière globale + Balancing sur la grille grossière intermédiaire :

$$\mathbf{u} = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T\mathbf{b} + \mathbf{Q}\tilde{\mathbf{u}}$$

où $\tilde{\mathbf{u}} = \mathbf{M}^{-1}\tilde{\mathbf{v}}$ et $\tilde{\mathbf{v}}$ solution de $\mathbf{D}^{-1}\mathbf{A}\mathbf{Q}\mathbf{M}^{-1}\tilde{\mathbf{v}} = \mathbf{P}\mathbf{D}^{-1}\mathbf{b}$

$$\text{avec } \mathbf{M}^{-1} = \sum_{i=1}^N \mathbf{R}_i^0 (\mathbf{D}^{-1}\mathbf{A})_i^{-1} \mathbf{R}_i$$

$$\text{et } (\mathbf{D}^{-1}\mathbf{A})_i^{-1} \approx \mathbf{Z}_i\mathbf{E}_i^{-1}\mathbf{Z}_i^T + \mathbf{Q}_i\mathbf{ILU}_i^{-1}\mathbf{P}_i$$

(rappel pour 2-niveaux : $(\mathbf{D}^{-1}\mathbf{A})_i^{-1} \approx \mathbf{ILU}_i^{-1}$)

Déflation 3-niveaux

Etat des lieux :

- Implémentation en cours de finalisation dans AIRONUM :

(pour un sous-domaine Ω_i , U = vecteur à $NS_i \times 5$ composantes, W = vecteur à $N_i \times 5$ composantes)

ZonagMacro.f : partitionnement de chaque sous-domaine (Ω_{ik})

AssCGsys_3lev.f : assemblage matrice grille grossière intermédiaire (E_i)

GaussJordanInv.f : inversion de la matrice grille grossière intermédiaire (E_i^{-1})

MatVecZW_3lev.f : calcul de $U = Z_i * W$ (prolongement du vecteur W)

MatVecZtU_3lev.f : calcul de $W = Z_i^T * U$ (restriction du vecteur U)

MatVecEinvW_3lev.f : calcul de $E_i^{-1} * W$

MatVecPDU-PDUT_3lev.f : calcul de $P_i * U$ et de $Q_i * U$

MatPreILU-PDefl_3lev.f : calcul de $(Z_i E_i^{-1} Z_i^T + Q_i ILU_i^{-1} P_i) * U$

- Puis étape de validation et d'expérimentation...

Plan

- 1 Préconditionnement par déflation (et balancing) : utilisation de la bibliothèque MUMPS dans AIRONUM (avec H. Alcin)
- 2 Déflation 3-niveaux : description de l'algorithme et état des lieux dans AIRONUM
- 3 **Modélisation de la turbulence : VMS-LES dynamique, hybride RANS/VMS-LES dynamique (avec A. Dervieux et M.-V. Salvetti)**
- 4 Mise en oeuvre du cas test du cylindre circulaire : Reynolds 20K, 500K-3M (avec S. Wornom)

Modélisation de la turbulence : VMS-LES dynamique

- Développement d'une version VMS-LES (et LES) dynamique efficace en non structuré pour les modèles de sous-maille de Smagorinsky et WALE, et extension à l'approche hybride RANS/VMS-LES.
- Mise en oeuvre informatique de ces modèles dynamiques dans AIRONUM.
- Expérimentations et validations pour le cas test du cylindre circulaire à Reynolds 20K (VMS-LES) et 500K-3M (Hybride RANS/VMS-LES).

L'approche VMS-LES dans AIRONUM

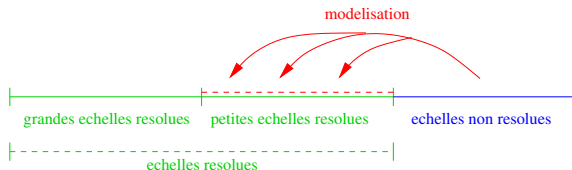
Extension du travail de Hughes et al. (2000, 2001, écoulements incompressibles + méthodes spectrales + turbulence homogène isotrope) pour pouvoir prendre en compte :

- les équations de **Navier-Stokes compressibles 3D**,
- des maillages **non structurés**,
- un modèle de discrétisation mixte **EF/VF**,
- des écoulements avec **détachements tourbillonnaires**.

(Comput. Methods Appl. Mech. Engrg., 2004 et 2006.)

Principales caractéristiques de l'approche VMS-LES

- **Construction** du modèle qui **repose sur la méthode de discrétisation spatiale** utilisée,
- Approche basée sur des **projections variationnelles** des équations de Navier-Stokes $\Rightarrow$ **équations régissant des échelles différentes de la solution** (grandes échelles résolues, petites échelles résolues, et échelles non résolues),
- **Effets des échelles non résolues modélisés** dans les équations régissant les **petites échelles résolues, mais pas** dans les équations associées aux **grandes échelles résolues** :



Version dynamique

Motivation : améliorer la prédiction du modèle VMS-LES (et LES) tout en préservant l'efficacité en non structuré.

Constante du modèle de sous-maille qui varie en temps et en espace (voir annexe pour les détails) :

- Smagorinsky :

- $\mu_{SGS}^{VMS} = \bar{\rho}(c_s \Delta^2)(\mathbf{x}, \mathbf{t}) |S'|$

- $\mu_{SGS}^{LES} = \rho(c_s \Delta^2)(\mathbf{x}, \mathbf{t}) |S|$

- WALE :

- $\mu_{SGS}^{VMS} = \bar{\rho}(c_w \Delta^2)(\mathbf{x}, \mathbf{t}) f(S', S'^d)$

- $\mu_{SGS}^{LES} = \rho(c_w \Delta^2)(\mathbf{x}, \mathbf{t}) f(S, S^d)$

+ modification de la constante dynamique du modèle WALE afin de retrouver le comportement asymptotique correcte du modèle original à la paroi (Baya Toda *et al.* 2010).

Approche hybride : nos motivations

- Calculer en maillages non structurés des écoulements à **grand nombre de Reynolds** caractérisés par des **détachements tourbillonnaires importants**.
- RANS : problèmes de précision pour ces écoulements.
- LES : plus coûteux que RANS, et requiert une résolution très fine des couches limites à grand nombre de Reynolds $\Rightarrow$ coût souvent prohibitif.
- Développer un **modèle hybride** qui combine **RANS** et **LES** pour exploiter au mieux les avantages des deux approches :
 - moins exigeant en ressource de calcul comparé à LES,
 - meilleure précision que RANS,et qui puisse **passer automatiquement et progressivement** de l'une à l'autre des approches.
- Développer un **modèle hybride** qui puisse **intégrer la VMS-LES**.

Approche hybride : l'option choisie

- **Idée centrale** de cette approche hybride de turbulence :
 - **Résoudre** les équations **RANS dans tout le domaine**,
 - **Corriger** le champ d'écoulement moyen obtenu en **ajoutant des fluctuations données par un modèle VMS-LES** dans les régions de résolution de grille adéquates pour VMS-LES.
- **Ingrédients de base** de cette approche hybride :
 - un modèle **RANS**,
 - un modèle **VMS-LES**,
 - une fonction d'**hybridation**.

Approche hybride : les équations

$$\left(\frac{\partial W_h}{\partial t}, \mathcal{X}_i \right) + (\nabla \cdot F(W_h), \mathcal{X}_i, \Phi_i) = -\theta(\tau^{RANS}(W_h), \Phi_i) - (1 - \theta)(\tau^{LES}(W'_h), \Phi'_i)$$

$$\text{où } \theta = \tanh \left[\left(\frac{\Delta}{l_{RANS}} \right)^2 \right]$$

avec $l_{RANS} = \frac{k^{3/2}}{\varepsilon}$ et $\Delta =$ taille locale de maille.



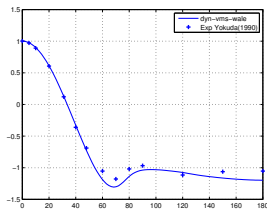
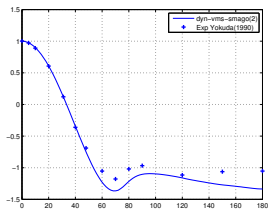
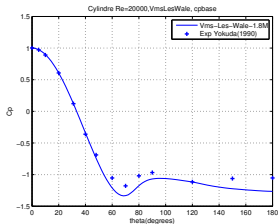
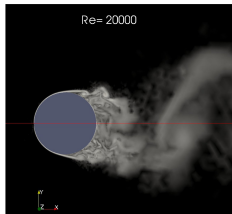
$\Delta \ll l_{RANS} : \theta \rightarrow 0$ (mode VMS-LES)

$\Delta \gg l_{RANS} : \theta \rightarrow 1$ (mode RANS)

Plan

- 1 Préconditionnement par déflation (et balancing) : utilisation de la bibliothèque MUMPS dans AIRONUM (avec H. Alcin)
- 2 Déflation 3-niveaux : description de l'algorithme et état des lieux dans AIRONUM
- 3 Modélisation de la turbulence : VMS-LES dynamique, hybride RANS/VMS-LES dynamique (avec A. Dervieux et M.-V. Salvetti)
- 4 Mise en oeuvre du cas test du cylindre circulaire : Reynolds 20K, 500K-3M (avec S. Wornom)

Cylindre Reynolds 20000 (VMS-LES)

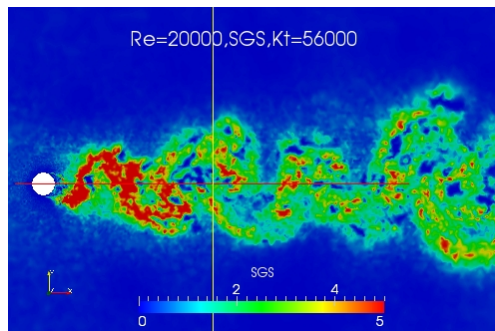


Cp moyen sur le cylindre (maillage 1.8M de noeuds) : VMS-LES WALE, Smagorinsky dyn. et WALE dyn.

Cylindre Reynolds 20000 (VMS-LES)

Simulation	$\overline{C_d}$	C_{Lrms}	l_r	$-\overline{C_{p_b}}$	θ_{sep}	St
LES Smagorinsky	1.295	0.574	0.779	1.3	86	.197
LES Smagorinsky dyn	1.24	0.44	0.9	1.24	84	0.19
LES WALE dyn	1.17	0.4	1.182	1.149	84	0.197
VMS-LES Smagorinsky dyn	1.296	0.563	0.852	1.336	86.24	0.186
VMS-LES Wale dyn	1.20	0.44	1.0733	1.197	84	0.191
Experiments (Norberg 2003,Basu 1985,Yokuda 1990)	[1.10-1.20]	[0.4-0.6]	–	[1.03-1.09]	–	0.194
LES (Salvatici 2003)	[0.94-1.28]	[0.17-0.65]	[0.7-1.4]	[0.83-1.38]	–	–
LES (Aradag 2009)	–	–	1.	[1.04-1.25]	–	–

Cylindre Reynolds 20000 (VMS-LES)



Isovaleurs instantanées du rapport μ_{SGS}/μ :
VMS-LES WALE dyn., maillage 3.8M.

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

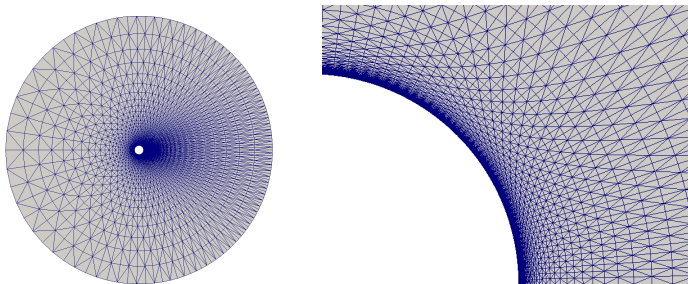


FIG.: Snapshot of the surface mesh for Reynolds number $1M$, with $274K$ cells, $y+_{match} = 100$ and $y+_{mesh} = 10$; zoom on the near-wall mesh (image on the right).

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

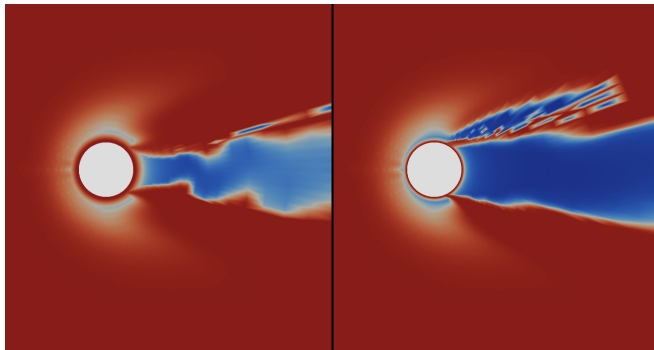


FIG.: Hybrid computation : Visualisation of the blending function for Reynolds number 1M (left image) vs 3M (image on the right).

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

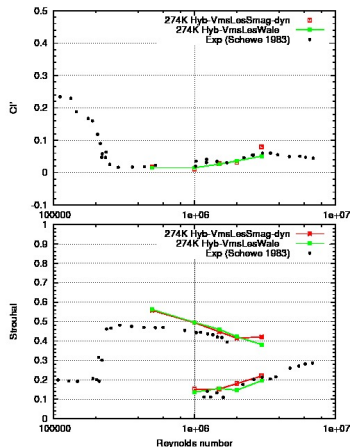


FIG.: C_l' and Strouhal vs. Reynolds curve : Hybrid-VMS Smagorinsky and WALE dyn. computations + Exp. data of Schewe.

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

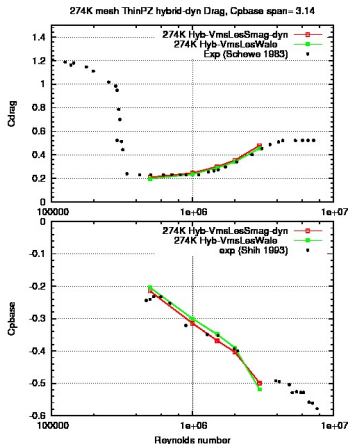


FIG.: C_d and C_{pBase} vs. Reynolds curve : Hybrid-VMS Smagorinsky and WALE dyn. computations, Exp. data of Schewe.

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

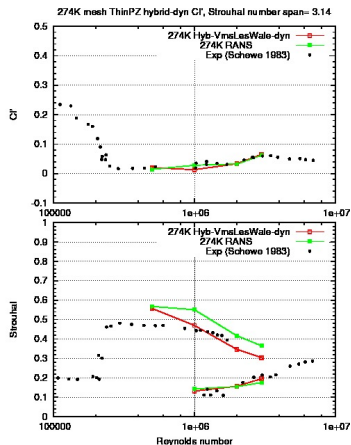


FIG.: C_f and Strouhal vs. Reynolds curve : Hybrid-VMS WALE dyn. and RANS computations, Exp. data of Schewe.

Cylindre Reynolds 500K-3M (Hybride RANS/VMS-LES)

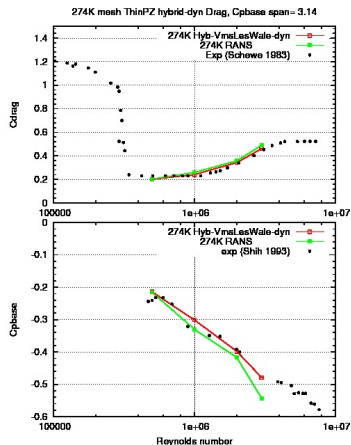


FIG.: C_d and C_{pbase} vs. Reynolds curve : Hybrid-VMS WALE dyn. and RANS computations, Exp. data of Schewe.

Annexe : WALE dynamique

Notations : $\bar{\mathbf{u}}$ = filtre de grille et $\tilde{\mathbf{u}} = \overline{\rho \mathbf{u}} / \bar{\rho}$ = filtre de Favre.

Filtre de grille :

Soit $M_{ij} = \overline{\rho \mathbf{u}_i \mathbf{u}_j} - \bar{\rho} \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_j$ le tenseur des contraintes de sous-maille.

On a :

$$M_{ij} = M_{ij} - \frac{1}{3} M_{kk} \delta_{ij} + \frac{1}{3} M_{kk} \delta_{ij} = T_{ij} + \frac{1}{3} M_{kk} \delta_{ij}$$

On modélise T_{ij} par $T_{ij} = -\mu_{SGS} \tilde{P}_{ij}$ avec

$$\tilde{P}_{ij} = -\frac{2}{3} \tilde{S}_{kk} \delta_{ij} + 2 \tilde{S}_{ij}$$

où :

$$\tilde{S}_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{\mathbf{u}}_i}{\partial x_j} + \frac{\partial \tilde{\mathbf{u}}_j}{\partial x_i} \right)$$

et où :

$$\mu_{SGS} = \bar{\rho} (c_W \Delta)^2 g(\tilde{\mathbf{u}}) \quad ; \quad g(\tilde{\mathbf{u}}) = \frac{(\tilde{S}_{ij}^d \tilde{S}_{ij}^d)^{\frac{3}{2}}}{(\tilde{S}_{ij} \tilde{S}_{ij})^{\frac{5}{2}} + (\tilde{S}_{ij}^d \tilde{S}_{ij}^d)^{\frac{5}{4}}}$$

avec :

$$\tilde{S}_{ij}^d = \frac{1}{2} \left(\tilde{g}_{ij}^2 + \tilde{g}_{ji}^2 \right) - \frac{1}{3} \tilde{g}_{kk}^2 \quad ; \quad \tilde{g}_{ij}^2 = \tilde{g}_{ik} \tilde{g}_{kj} \quad \text{et} \quad \tilde{g}_{ij} = \frac{\partial \tilde{\mathbf{u}}_i}{\partial x_j} .$$

Annexe : WALE dynamique

Filtre test (2ème niveau de filtre) :

On a le tenseur des contraintes de “subtest-scale” $M_{ij}^{test} = \widehat{\rho \mathbf{u}_i \mathbf{u}_j} - (\hat{\rho})^{-1} \left(\widehat{\rho \mathbf{u}_i} \widehat{\rho \mathbf{u}_j} \right)$

Comme précédemment avec T_{ij} , on modélise la partie déviatorique de M_{ij}^{test} par

$$M_{ij}^{test} - \frac{1}{3} M_{kk}^{test} \delta_{ij} = -c \hat{\Delta}^2 \hat{\rho} g(\hat{\mathbf{u}}) \hat{\hat{P}}_{ij} \quad (c = c_W^2)$$

D'autre part on a $\mathcal{L}_{ij} = M_{ij}^{test} - \hat{M}_{ij} = \widehat{\tilde{\rho} \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_j} - (\hat{\rho})^{-1} \left(\widehat{\tilde{\rho} \tilde{\mathbf{u}}_i} \widehat{\tilde{\rho} \tilde{\mathbf{u}}_j} \right)$

On a aussi $L_{ij} = \mathcal{L}_{ij} - \frac{1}{3} \mathcal{L}_{kk} \delta_{ij} = (c \Delta^2) B_{ij}$ avec

$$B_{ij} = \widehat{\tilde{\rho} g(\tilde{\mathbf{u}}) \tilde{\hat{P}}_{ij}} - \left(\frac{\hat{\Delta}}{\Delta} \right)^2 \hat{\rho} g(\hat{\mathbf{u}}) \hat{\hat{P}}_{ij}$$

L'inconnue est $(c \Delta^2)$ qui doit satisfaire $L_{ij} = (c \Delta^2) B_{ij}$. Ce système de 6 equations à une inconnue est “résolu” par moindres carrés :

$$(c \Delta^2) \text{ minimise } Q = (L_{ij} - (c \Delta^2) B_{ij})^2$$

$$\frac{\partial Q}{\partial (c \Delta^2)} = 0 \quad \Rightarrow \quad (c \Delta^2) = \frac{L_{ij} B_{ij}}{B_{pq} B_{pq}}.$$

N.B. Le “filtre test” choisi permet de déterminer $\frac{\hat{\Delta}}{\Delta}$.

Annexe : WALE dynamique

• Exemple de “filtre test” :

On définit pour une fonction ψ définie aux noeuds sa test-filtrée $\hat{\psi}$ au noeud k par :

$$\hat{\psi}_k = \frac{\sum_{T, k \in T} V(T)/4 (\psi_{T,1} + \psi_{T,2} + \psi_{T,3} + \psi_{T,4})}{\sum_{T, k \in T} V(T)} .$$

où T, i est le i -ième sommet du tétraèdre T .

Dans ce cas, $\hat{\Delta}/\Delta$ est évalué à $(N_i)^{\frac{1}{3}}$ où N_i est le nombre de tétraèdres ayant pour sommet i .

- “Lissage local” + “clipping” appliqués à $c\Delta^2$.
- Option d’implémentation : minimisation des calculs $\Rightarrow c\Delta^2$ et les quantités $\hat{\psi}$ stockés par noeuds.