

# Projet ANR ECINADS: Partenaire INRIA

H. Alcin <sup>1</sup>   A. Belme <sup>1</sup>   A. Carabias <sup>1</sup>   A. Dervieux <sup>2</sup>  
L. Hascoët <sup>1</sup>   V. Pascual <sup>1</sup>



<sup>1</sup>Équipe-projet Tropics <sup>2</sup>Tropics et Pumas  
Institut National de Recherche en Informatique et en Automatique,  
Sophia-Antipolis, France

*Réunion ECINADS, Sophia-Antipolis , 28 octobre 2010*

- Présentation de l'équipe INRIA
- Rôle dans ECINADS
- Premiers résultats

# 1. Présentation de l'équipe INRIA

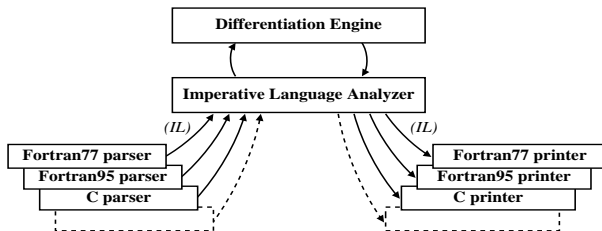
- Equipe-Projet INRIA : Tropics
  - Différentiation Automatique
  - Applications en Mécanique des Fluides
  
- Equipe-Projet INRIA : Pumas
  - Mécanique des Fluides Numérique
  - Applications à la fusion contrôlée (ITER)

# 1.1 Expertise en Différentiation Automatique (1)

## Développement du logiciel TAPENADE

TAPENADE (115 000 lines of JAVA) transforms a FORTRAN program into a FORTRAN program which computes the derivative of the given FORTRAN program. TAPENADE is organised around an intermediate step, the analysis and differentiation of an abstract imperative language.

<http://www-sop.inria.fr/tropics/>



Overall Architecture of TAPENADE

# 1.1 Expertise en Différentiation Automatique (2)

## Direct/Tangent mode :

- Differentiated code computes  $(c, \delta c) \mapsto j'(c) \cdot \delta c$ .
- Gâteaux derivative of a functional.
- Computational cost factor :  $\alpha_T \approx 4$ .

## Backward/Reverse mode :

- Differentiated code computes  $(c, \delta j) \mapsto (j'(c))^* \cdot \delta j$ .
- Gradient of a functional.
- Adjoint state.
- Computational cost factor :  $\alpha_R \approx 5$ .

## User directives :

- Directive II-LOOP allows for a more efficient reverse differentiation of parallel loops by inhibiting storages useless in “parallel” case.
- Directive NOCHECKPOINT can be used for optimal checkpointing in reverse differentiation.
- TAPENADE lets the user specify finely which procedure calls must be checkpointed or not.

## Different languages :

- TAPENADE V3 has been extended to deal with [Fortran95](#) and with [ANSI C](#).

## Higher order differentiation :

- TAPENADE is optimised for successive reverse differentiations.

# 1.1 Expertise en Différentiation Automatique (2) : Serveur TAPENADE sur WEB

Select the input language :

given by the suffix of the files   Fortran 77   Fortran 95   C

Upload source and include files, repeatedly or [copy paste your fortran source program](#).

Type the file path in, or browse :

and upload it

Name of the top routine :

Dependent output variables (separator: white space, default: all variables) :

Independent input variables (separator: white space, default: all variables) :

*(optional)* For our records, could you please give us your name and the application you have in mind (it can very well be only "test") :

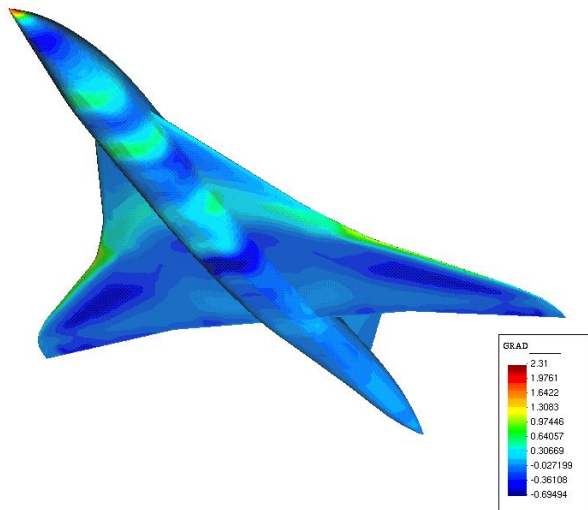
Differentiate in

# 1.1 Expertise en Différentiation Automatique (4)

## Applications de la DA

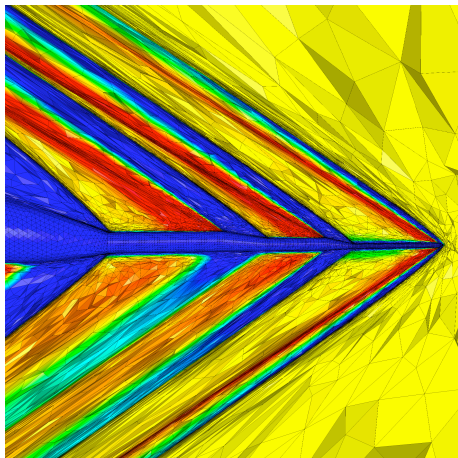
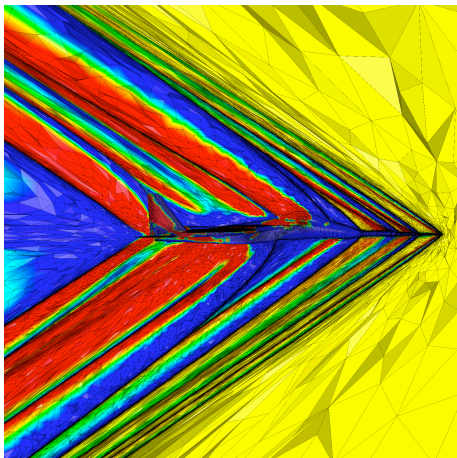
- Adjoint pour le design.
- Adjoint pour l'adaptation de maillage.
- Dérivées secondes pour la prise en compte des incertitudes.

## 1.1 Expertise en DA (5) : gradient en réduction du bang sonique





## 1.1 Expertise en DA (6) : adaptation de maillage pour le bang sonique (coopération avec l'équipe Gamma, Rocquencourt)



## 1.2 Expertise en Algorithmes de résolution Parallèles

### Parallélisme

De Suprenum à SGI-Altix en passant par la Connection Machine.

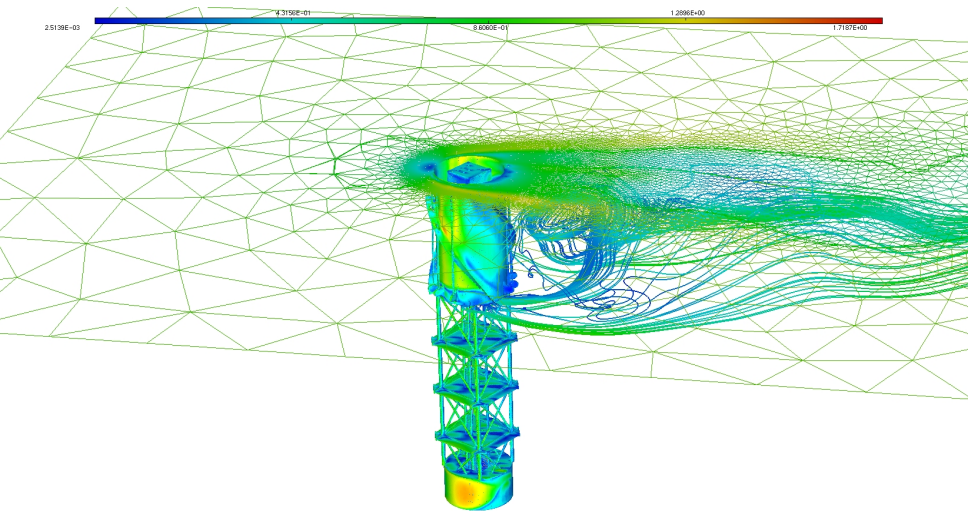
### Multigrilles

Multigrille par agglomération et leur version parallèle.

### Décomposition de domaine

Travaux de Koobus-Sarkis dans notre code AERO.

## 1.2 Expertise en Parallélisme : écoulement autour d'une géométrie offshore complexe (coopération avec Lemma)



## 2. ROLE DANS ECINADS

Task 1 : Coordination and dissemination

INRIA

Task 2 : Global preconditioning by Coarse grid

INRIA

Task 3 : Subdomain preconditioning

3.1 : Unstructured preconditioning by renumbered ILU(1)

3.2 : Preconditioning for higher order closures INRIA

## 2. ROLE DANS ECINADS(2)

Task 4 : Assembly of both preconditioners in accordance to AD strategy

T4.1 : specification of solver architecture (M1-6, [INRIA](#)) T4.2 : installation (M1-15, I3M, Lemma) of the two preconditioners in the same code version, T4.3 : identification of best preconditioning strategies (CT1, CT2,CT3, M15-24, I3M, Lemma).

Task 5 : Parallel reverse differentiation and checkpointing

by Automated Differentiation/ Adjoint iteration with parallel-mpi checkpointing

5.1 : AD strategy, [INRIA](#)

5.2 : Adjoint iteration with parallel-mpi checkpointing, [INRIA](#)

5.3 : application to correctors and mesh adaptation, [INRIA](#)

## 2. ROLE DANS ECINADS(3)

### Task 6 : Turbulence models

I3M et [INRIA](#) : nouveaux modèles hybrides, travail sur leur résolution efficace.

### Task 7 : Assessments and applications

CT1 : Turbulent flow around a cylinder at high Reynolds number : capturing of the critical regime for Reynolds 20000, 150000, 1 million, voir [Perrin 2006,2008]. [INRIA](#), adaptation de maillage.

CT2 : Pitching 3D flow around a wing in subsonic and transonic regime, experimented and computed in [Martinat],[Bourdet],[Berton],[ Seegmiller].

CT3 : Flow around a riser with coupling with elastic structure.

CT4 : Flow around a complex geometry of spar (platform) held with elastic moorings.

### 3. TRAVAIL EFFECTUE PAR L'INRIA

#### Task 2 : Global preconditioning by Coarse grid

Mise au point d'une méthode à deux niveaux de grille grossière par Hubert Alcin (co-encadrement O. Allain).

#### Task 4 : Assembly of both preconditioners in accordance to AD strategy

Combinaison grille grossière / ILU(0) (H. Alcin, co-encadrement O. Allain)

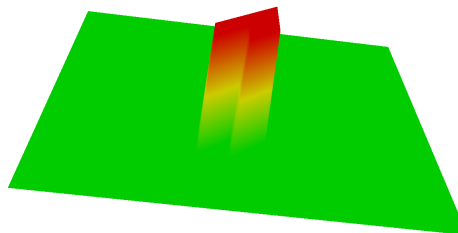
#### Task 5 : Parallel reverse differentiation and checkpointing

by Automated Differentiation/ Adjoint iteration with parallel-mpi checkpointing

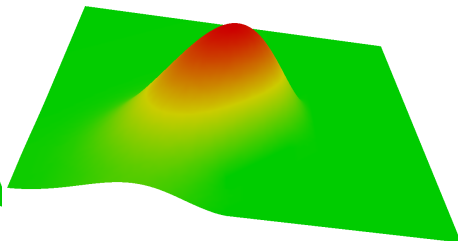
*Tâche en démarrage* (H. Alcin et L. Hascoët)

### 3. TRAVAIL EFFECTUE PAR L'INRIA, ILLUSTRATION

Définition de la base grossière



Fonction de base caractéristique

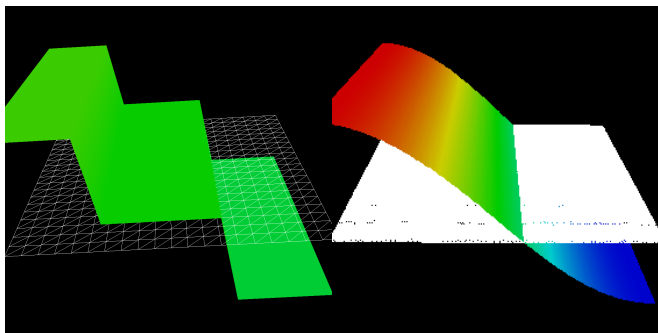


Fonction de base lissée



# Base discontinue vs lisse

Vérification sur le recalcul d'une sinusoïde (amplitude 2.)



- À gauche : solution grossière avec base caractéristique ( amplitude  $\approx 0.06$ )
- À droite : solution grossière avec base lisse ( amplitude  $\approx 1.8$ ).

Méthode : Schwarz + grille grossière "Balancing"

nb points	10000	20000	47000	94000
nb domaines	12	28	66	142
nb points par domaines	833	714	712	661
cv fct caractéristiques	480	546	750	810
cv fct lisses	400	391	444	491

- Scalabilité imparfaite pour les fonctions caractéristiques.
- Assez bonne scalabilité pour les fonctions lisses.

## 8. METHODE A TROIS NIVEAUX (en cours)

### Motivation :

Les calculs précédents ont une convergence limitée par l'efficacité de ILU sur chaque sous-domaine. Avec des sous-domaines comprenant beaucoup de points, le nombre d'itérations nécessaires pour la convergence va être plus grand.

On souhaite donc diminuer la sensibilité de la convergence au nombre de nœuds par sous-domaine.

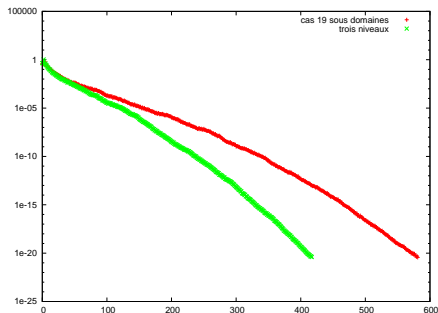
### Approche :

On ajoute une grille grossière dans chaque sous-domaine.

À terme on voudrait une fonction de base locale pour (par ex.) 100 points.

# Comparaison deux et trois niveaux

- Balancing avec 19 domaines sur 90000 nœuds (irrégulier) :
- rouge : deux niveaux dont un niveau grossier à 19 fonctions de base caractéristiques,
- vert : trois niveaux dont un niveau grossier à 19 fonctions de base caractéristiques et 341 fonctions de bases lisses réparties dans les 19 sous-domaines



## 9. CONCLUSIONS/SUITE DE L'ETUDE

- Identification d'une stratégie scalable en termes de nombre de processeurs et de nombres de points.
- Identification d'une stratégie scalable en termes de nombres de points, à confirmer.
- Étape en cours : optimisation de la condition de raccord par comparaison avec des conditions de type Robin.

## 9. CONCLUSIONS/SUITE DE L'ETUDE (2)

- L'extension au cas Navier-Stokes compressible sur machine parallèle va suivre (sous la maîtrise d'œuvre de I3M).
- L'extension au cas Navier-Stokes incompressible bifluide sur machine parallèle va suivre (sous la maîtrise d'œuvre de Lemma).

## 9. CONCLUSIONS/SUITE DE L'ETUDE (3)

- Une première différentiation du code parallèle compressible démarre en novembre (H. Alcin, L. Hascoët)
- Une deuxième étape doit optimiser les compromis stockage-recalculs.
- Une troisième étape étudiera l'optimisation des communications adjointes.