Varna, BG, September 22^{nd}-24^{th}, 2010

DISCRETE ADJOINT SOLVER FOR DESIGN IN AERONAUTICS

Pierre Trontin, Jacques Peter and Florent Renac

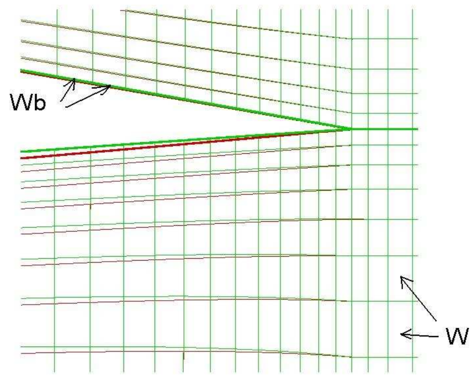
OUTLINE

Discrete adjoint solver
Layout

- **Local optimization of aeronautical shapes**
- **Minimization of an aerodynamic function $\mathcal{J}$ (drag ...) under the constraints $\mathcal{G}_k \leq 0$ ($k \in [1, n_c]$) (lift, pitching moment ...).**
- **Parametric optimization (vector of parameters $\alpha \in R^p$): $\mathcal{J}(\alpha)$, $\mathcal{G}_k(\alpha)$**
- **Gradients $d\mathcal{J}(\alpha)/d\alpha_i$ and $d\mathcal{G}_{\phi(k)}(\alpha)/d\alpha_i$ ($i \in [1, n_s]$) used in the optimization process.**

Discrete adjoint solver
Equations (1/4)

- **A finite-volume, cell centered code: elsA [a] (developped at the Onera)**
- X **is the mesh and** W **the aerodynamic conservative field**
- $R(W, X)$ **is the discrete residual**
- J **and** g_k **are defined such as** $\mathcal{J}(\alpha) = J(W(\alpha), X(\alpha), W_b(X(\alpha), W(\alpha)))$ **and** $\mathcal{G}_k(\alpha) = g_k(W(\alpha), X(\alpha), W_b(X(\alpha), W(\alpha)))$



[a] Cambier L, Veuillot J. Status of the elsA software for flow simulation and multi-disciplinary applications. 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, number AIAA Paper 2008-0664, 2008.

Discrete adjoint solver
Equations (2/4) [a]

- **Discrete equations for mechanics (set of n_a non-linear equations)**

$$R(W(\alpha), X(\alpha)) = 0$$

- **Differentiation with respect to α_i i $\in [1, n_s]$. Derivation of n_s linear system of size n_a**

$$\frac{\partial R}{\partial W}\frac{dW}{d\alpha_i} = -(\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

- **Calculation of derivatives**

$$\nabla_\alpha \mathcal{J}(\alpha) = \frac{\partial J}{\partial X}\frac{dX}{d\alpha} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha} + (\frac{\partial J}{\partial W} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W})\frac{dW}{d\alpha}$$

$$\nabla_\alpha \mathcal{G}_k(\alpha) = \frac{\partial g_k}{\partial X}\frac{dX}{d\alpha} + \frac{\partial g_k}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha} + (\frac{\partial g_k}{\partial W} + \frac{\partial g_k}{\partial W_b}\frac{\partial W_b}{\partial W})\frac{dW}{d\alpha}$$

[a] Peter J, Dwight R. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. Comp. Fluids 2010;39:373-391

Discrete adjoint solver
Equations (3/4)

- **Following equalities hold** $\forall \lambda \in \mathbb{R}^{n_a}$

$$\lambda^T \frac{\partial R}{\partial W}\frac{dW}{d\alpha_i} + \lambda^T (\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i}) = 0$$

$$\frac{d\mathcal{J}(\alpha)}{d\alpha_i} = \frac{\partial J}{\partial X}\frac{dX}{d\alpha_i} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha_i} + (\frac{\partial J}{\partial W} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W})\frac{dW}{d\alpha_i} + \lambda^T \frac{\partial R}{\partial W}\frac{dW}{d\alpha_i} + \lambda^T (\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

$$\frac{d\mathcal{J}(\alpha)}{d\alpha_i} = (\frac{\partial J}{\partial W} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W} + \lambda^T \frac{\partial R}{\partial W})\frac{dW}{d\alpha_i} \quad \frac{\partial J}{\partial X}\frac{dX}{d\alpha_i} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha_i} + \lambda^T (\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

Discrete adjoint solver
Equations (4/4)

- **Vector λ defined in order to cancel the factor of the flow sensitivity $\dfrac{dW}{d\alpha_i}$**

$$\frac{\partial J}{\partial W}+\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W}+\lambda^T\frac{\partial R}{\partial W}=0 \Longleftrightarrow \boxed{\left(\frac{\partial R}{\partial W}\right)^T\lambda=-\left(\frac{\partial J}{\partial W}+\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W}\right)^T}$$

- **Calculation of derivatives**

$$\forall i\in[1,nf]\quad \frac{d\mathcal{J}(\alpha)}{d\alpha_i}=\frac{\partial J}{\partial X}\frac{dX}{d\alpha_i}+\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha_i}+\lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

$$\nabla_\alpha\mathcal{J}(\alpha)=\frac{\partial J}{\partial X}\frac{dX}{d\alpha}+\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X}\frac{dX}{d\alpha}+\lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha})$$

- **Method with (n_c^*+1) (number of functions) and not n_s linear systems to solve**

Discrete adjoint solver dJ/dX mode[a] (1/3)

- With the classical gradient methods, the storage of $dX/d\alpha_i$ requires large memory resources $\Longrightarrow dJ/dX$ mode.
- dJ/dX mode: calculate the derivatives of the discrete equations R w.r.t. the mesh X, as well as dJ/dX and dG_k/dX

$$\nabla_\alpha \mathcal{J}(\alpha) = \left(\frac{\partial J}{\partial X} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X} + \lambda^T \frac{\partial R}{\partial X} \right) \frac{dX}{d\alpha}$$

$$\boxed{\nabla_X \mathcal{J}(X) = \frac{\partial J}{\partial X} + \frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial X} + \lambda^T \frac{\partial R}{\partial X}}$$

to be calculated and stored.

[a] Trontin P, Peter J, Nguyen-Dinh M. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates. 49th AIAA Aerospace Sciences Meeting, 4-7 January 2011. Accepted.

Discrete adjoint solver
dJ/dX mode (2/3)

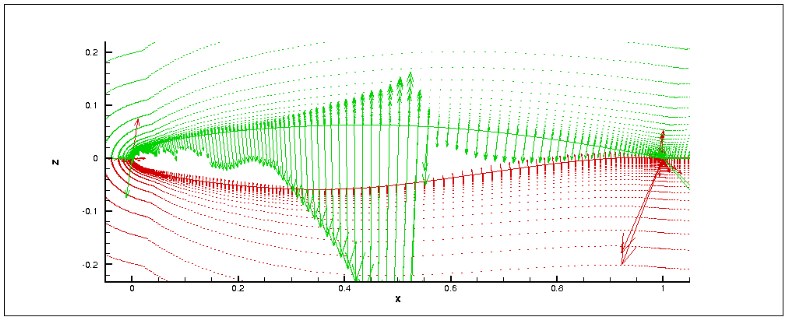
- Final product with $dX/d\alpha_i$ carried out by another computer with higher memory capacities and possibly lower computational speed.
- Surface mesh $S(\alpha)$
- Volume mesh X is an explicit function of S and $\tilde{J}(S) = J(X(S))$ (resp. $\tilde{G}_k(S) = G_k(X(S))$)

$$\frac{d\tilde{J}}{dS} = \frac{dJ}{dX}\frac{dX}{dS} \qquad \frac{d\tilde{G}_k}{dS} = \frac{dG_k}{dX}\frac{dX}{dS}$$

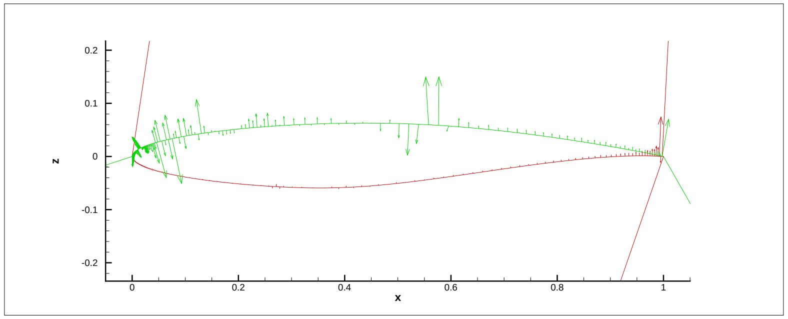
- Shape optimization independant of any set of the design parameters
- A cheaper computation of the sensitivities (multiplying last equation by $(dS/d\alpha_i)$)

Discrete adjoint solver
dJ/dX mode (3/3)

- Airfoil RAE2822
- $M_\infty = 0.734$, $Re_\infty = 6.5\,10^6$
- AoA = 2.79^o



dC_D/dX



dC_D/dS

Discrete adjoint solver
Available Numerical methods in elsA

- **Differentiated fluxes**
  - ◁ **Mean flow :**
    - ▷ **Roe flux : second order using the MUSCL formula with Van Albada limiting function**
    - ▷ **Centered flux plus scalar artificial dissipation (Jameson** *et al.*, **1981)**
    - ▷ **Viscous flux with cell-centered gradients corrected at interfaces**
  - ◁ **Turbulence quantities transport equations**
    - ▷ **Roe order one flux**
    - ▷ **Viscous flux**
    - ▷ **Source term of the differentiated models (cell centered evaluation of gradient terms)**[a]
    - ▷ **Models :** $k-\omega$ **Wilcox,** $k-\epsilon$ **Launder-Sharma, Spalart Allmaras**
- **Resolution of the adjoint equation**
  - ▷ **Newton-type or relaxation algorithm** [b]
  - ▷ **Each Newton step solved by a LU-RELAX or LU-SSOR technique**
  - ▷ **Joins: matching, line matching, plane matching**

---

[a] Renac F, Pham CT, Peter J. Sensitivity analysis for the RANS equations coupled with a linearized turbulence model. AIAA Paper Series, Paper 2007-3948, 2007

[b] Peter J, Drullion F. Large stencil viscous flux linearization for the simulation of 3D turbulent flows with backward Euler schemes. Comp. Fluids, 36:1007-1027, 2007

OUTLINE

1- DISCRETE ADJOINT SOLVER

2- RECENT ADVANCES

- ▷ **The full adjoint linearization of the Spalart Allmaras turbulence model**
- ▷ **The RPM method**
- ▷ **The chimera method**

3- APPLICATIONS

The full linearization of the Spalart Allmaras turbulent model(1/3)

$$|V|\frac{\partial \mathbf{W}}{\partial t} + \oint_{\partial V} (\mathbf{F_c} + \mathbf{F_v}) \cdot \mathbf{n} dS - \int_V \mathbf{S} dV = 0$$

- **conservative variables:**

$$\mathbf{W} = (\rho, \rho \mathbf{U}^t, \rho E, \rho k, \rho \tilde{\nu})^t$$

- **convective and viscous fluxes:**

$$\mathbf{F_c} = \begin{pmatrix} \rho \mathbf{U}^t \\ \rho \mathbf{U}\mathbf{U}^t + p\mathbf{I} \\ (\rho E + p)\mathbf{U}^t \\ \rho \tilde{\nu} \mathbf{U}^t \end{pmatrix}, \quad \mathbf{F_v} = \begin{pmatrix} 0 \\ -\boldsymbol{\tau} - \boldsymbol{\tau_r} \\ -(\boldsymbol{\tau} + \boldsymbol{\tau_r})\mathbf{U} + \mathbf{q}^t + \mathbf{q_t}^t \\ -\frac{1}{\sigma}(\mu + \rho\tilde{\nu})\nabla\tilde{\nu}^t \end{pmatrix}$$

- **constraints and heat flux:**

$$\begin{aligned} \boldsymbol{\tau} + \boldsymbol{\tau_r} &= (\mu + \mu_t)\Big(-\frac{2}{3}(\nabla \cdot \mathbf{U})\mathbf{I} + \nabla\mathbf{U} + \nabla\mathbf{U}^t \Big) \\ \mathbf{q} + \mathbf{q_t} &= -\Big(\frac{C_p \mu}{Pr} + \frac{C_p \mu_t}{Pr_t} \Big) \nabla T \end{aligned}$$

The full linearization of the Spalart Allmaras turbulent model(2/3)

- **turbulent viscosity coefficient:**

$$\mu_t = \mu \chi f_{v1} \quad \text{with} \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \quad \text{and} \quad \chi = \frac{\rho \tilde{\nu}}{\mu}$$

- **source terms:**

$$\mathbf{S} = (0, 0, 0, Prod + Cross + Dest)^t$$

  - production :

$$\begin{aligned} Prod &= C_{b1} \tilde{S} \rho \tilde{\nu} \\ \tilde{S} &= \overline{\omega} + \frac{|\rho \tilde{\nu}|}{\rho} \frac{f_{v2}}{\kappa^2 \eta^2} \\ f_{v2} &= 1 - \frac{\chi}{1 + \chi f_{v1}} \end{aligned}$$

  - cross term (diffusion) :

$$\begin{aligned} Cross &= \frac{C_{b2}}{\sigma} \rho \nabla \tilde{\nu} \cdot \nabla \tilde{\nu} \\ Cross &= \min\left(Cross, 20 \rho \tilde{\nu} \max(0, C_{b1} \tilde{S})\right) \end{aligned}$$

The full linearization of the Spalart Allmaras turbulent model(3/3)

- **source terms:**
  - destruction :

$$
\begin{aligned}
Dest &= -C_{w1} f_w \frac{|\rho\tilde{\nu}|}{\rho} \frac{\rho\tilde{\nu}}{\eta^2} \\
f_w &= g\Big(\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\Big)^{1/6} \\
g &= \tilde{r} + C_{w2}(\tilde{r}^6 - \tilde{r}) \\
\tilde{r} &= \max\Big[0, \min\Big(10, \frac{1}{\tilde{S}} \frac{|\rho\tilde{\nu}|}{\rho} \frac{1}{\kappa^2\eta^2}\Big)\Big]
\end{aligned}
$$

- **model constants:**

coefficient	value	coefficient	value
C_{b1}	$0,1355$	C_{w1}	$\frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma}$
C_{b2}	$0,622$	C_{w2}	$0,3$
σ	$\frac{2}{3}$	C_{w3}	2
κ	$0,41$	C_{v1}	$7,1$

Results on the airfoil RAE2822

- Reynolds number: $Re = \frac{\sqrt{\rho_\infty p_\infty} c}{\mu(T_\infty)} = 6.5 \times 10^6$
- Mach number: $M = \frac{U_\infty}{c_\infty} = 0, 73$
- AoA: $\alpha = 2.79$ deg.
- spatial discretization : 2 blocks of 73×229 knots
- rotation variation (for finite differences) : $\delta\alpha = 10^{-2}$ deg.
- design parameter: angle of rotation around (Oy)

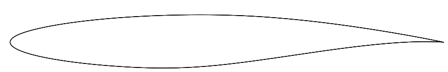


Figure 1: Airfoil RAE2822

Adjoint gradient computation on the airfoil RAE2822

function	DF	frozen turbulence assumption	full linearization
$dC_{L,p}/d\alpha$	1,703e-01	1,105e-01 (35%)	1,687e-01 $(0,9\%)$
$dC_{D,p}/d\alpha$	5,688e-03	4,741e-03 (17%)	5,703e-03 $(0,3\%)$
$dC_{D,w}/d\alpha$	2,272e-03	1,658e-03 (27%)	2,251e-03 $(0,9\%)$
$dC_{D,ff}/d\alpha$	4,092e-03	3,606e-03 (12%)	4,069e-03 $(0,5\%)$
$dC_{D,vp}/d\alpha$	1,820e-03	1,948e-03 (7%)	1,818e-03 $(0,1\%)$

OUTLINE

1- DISCRETE ADJOINT SOLVER

2- RECENT ADVANCES

- ▷ The full linearization of the Spalart Allmaras turbulent model
- ▷ The RPM method
- ▷ The chimera method

3- APPLICATIONS

Backward Euler scheme

- **The adjoint equation to be solved:**

$$\left(\frac{\partial R}{\partial W}\right)^{(ACC)T} \lambda = -\left(\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W} + \frac{\partial J}{\partial W}\right)^T$$

- **For solving the linear system:**

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

with $\mathbf{A} = \left(\frac{\partial R}{\partial W}\right)^{(ACC)T}$, $\mathbf{x} = \lambda$ and $\mathbf{b} = -\left(\frac{\partial J}{\partial W_b}\frac{\partial W_b}{\partial W} + \frac{\partial J}{\partial W}\right)^T$

- **we use the iterative algorithm:**

$$\mathbf{x}^{(l+1)} = \mathbf{F}(\mathbf{x}^{(l)}) = \mathbf{\Phi}\mathbf{x}^{(l)} + \mathbf{M}^{-1}\mathbf{b}$$

  - iteration matrix:

$$\mathbf{\Phi} = \mathbf{I} - \mathbf{M}^{-1}\mathbf{A}$$

  - implicit matrix:

$$\mathbf{M} = \frac{1}{\Delta t}\mathbf{I} + \frac{\partial \mathbf{R}}{\partial \mathbf{W}}^{(APP)}$$

Recursive Projection Method (RPM) [a]
Stability of the iterative scheme

- Theorem: convergence of the iterative algorithm

  Let $\mathbf{M},\ \mathbf{A} \in \mathbb{C}^{N \times N}$ nonsingular, then

$$\forall \mathbf{x}^{(0)}, \quad \lim_{l \to \infty} \mathbf{x}^{(l)} = \mathbf{A}^{-1}\mathbf{b} \quad \Leftrightarrow \quad \rho(\mathbf{\Phi}) < 1$$

- Divergent eigenspace definition
  - divergent eigenmodes $(\lambda_i, \mathbf{e_i})$: $\quad |\lambda_1| \geq \cdots \geq |\lambda_m| \geq 1$
  - divergent eigenspace:

$$\mathbb{P} = \text{span}\{\mathbf{e_1}, \cdots, \mathbf{e_m}\} \quad ; \quad \mathbb{Q} = \mathbb{P}^{\perp}$$

  - orthogonal projectors:

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad \mathbf{x} = \mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{x} = \mathbf{x}_p + \mathbf{x}_q$$

[a] Renac F. Aerodynamic sensitivity analysis of the RANS equations via the recursive projection method. AIAA Paper Series, Paper 2010-4364, 2010

Recursive Projection Method (RPM) Stabilization procedure

Let $\mathbf{V} \in \mathbb{R}^{N\times m}$ be an orthonormal basis for $\mathbb{P}$:

$$\mathbf{P} = \mathbf{V}\mathbf{V}^T, \quad \mathbf{Q} = \mathbf{I} - \mathbf{V}\mathbf{V}^T, \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

- RPM algorithm

$$\begin{aligned}
\mathbf{x}_q^{(l+1)} &= (\mathbf{I}-\mathbf{P})\mathbf{F}(\mathbf{x}_p^{(l)} + \mathbf{x}_q^{(l)}) \\
\mathbf{x}_p^{(l+1)} &= \mathbf{x}_p^{(l)} + (\mathbf{I}-\mathbf{P}\mathbf{\Phi}\mathbf{P})^{-1}(\mathbf{P}\mathbf{F}(\mathbf{x}_p^{(l)} + \mathbf{x}_q^{(l)}) - \mathbf{x}_p^{(l)}) \\
\mathbf{x}^{(l+1)} &= \mathbf{x}_p^{(l+1)} + \mathbf{x}_q^{(l+1)}
\end{aligned}$$

- Theorem: unconditional convergence of the RPM iteration
  Let $\mathbf{M},\ \mathbf{A} \in \mathbb{C}^{N\times N}$ nonsingular and such that $\mathbf{P}\mathbf{\Phi}\mathbf{Q} = 0$, then

$$\forall \mathbf{x}^{(0)}, \quad \lim_{l\to\infty} \mathbf{x}_p^{(l)} + \mathbf{x}_q^{(l)} = \mathbf{A}^{-1}\mathbf{b}$$

Recursive Projection Method (RPM) Krylov acceptance criterion

Goal: construction of the divergent basis $\mathbf{V}$

Method:

- let $k \in \mathbb{N}$
- define the Krylov subspace generated by $\mathbf{v}_1 = \Delta\mathbf{x}_q^{(l-k+1)}$ and $\hat{\mathbf{A}} = \mathbf{Q\Phi Q}$:

$$\mathbb{K}_k = \text{span}\{\mathbf{v}_1, \hat{\mathbf{A}}\mathbf{v}_1, \ldots, \hat{\mathbf{A}}^{k-1}\mathbf{v}_1\}$$

  where $\Delta\mathbf{x}_q^{(j)} = \hat{\mathbf{A}}^j\mathbf{x}_q^{(0)} = \mathbf{x}_q^{(j+1)} - \mathbf{x}_q^{(j)}$.

- set

$$\mathbf{K}_k = (\mathbf{v}_1, \hat{\mathbf{A}}\mathbf{v}_1, \ldots, \hat{\mathbf{A}}^{k-1}\mathbf{v}_1)$$

- and compute the QR factorization

$$\mathbf{K}_k = \mathbf{Q}_k\mathbf{R}_k$$

Recursive Projection Method (RPM) Application to Onera M6 wing (1/2) [a]

- Test-case : M6 wing. $Re = 1.46\,10^7$, $M_\infty = 0.84$, $AoA = 0^o$
- RANS equations + Spalart-Allmaras
- Structured mesh with 10 blocks and 620 042 nodes
- Design parameter: angle of rotation around (O, y)

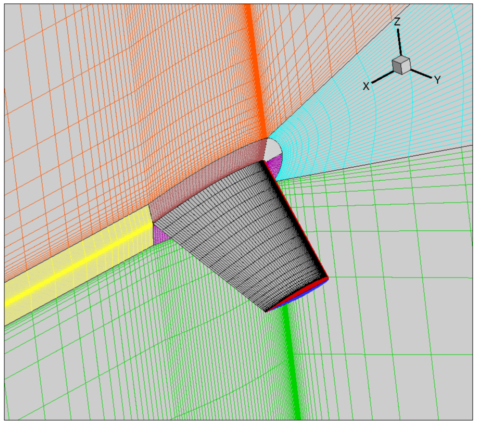
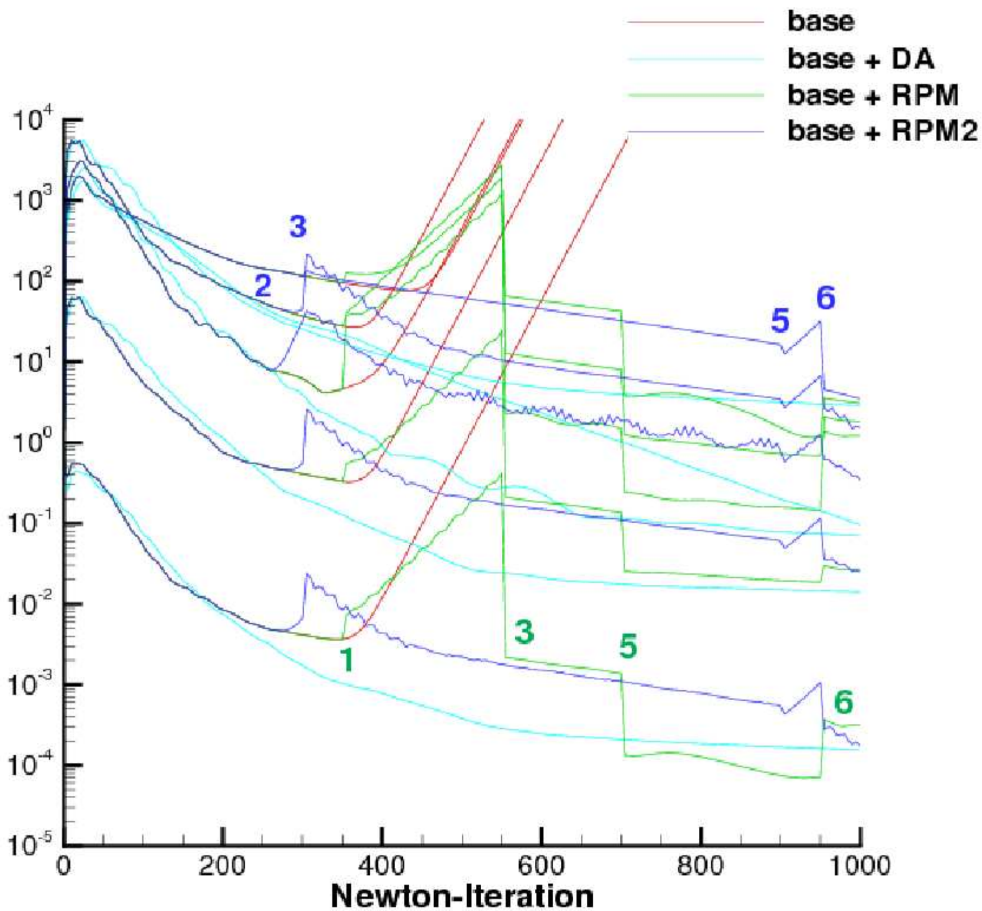


Figure 2: M6 wing

[a] Renac F. Aerodynamic sensitivity analysis of the RANS equations via the recursive projection method. AIAA Paper Series, Paper 2010-4364, 2010

Recursive Projection Method (RPM)
Application to Onera M6 wing (2/2)



OUTLINE

1- DISCRETE ADJOINT SOLVER

2- RECENT ADVANCES

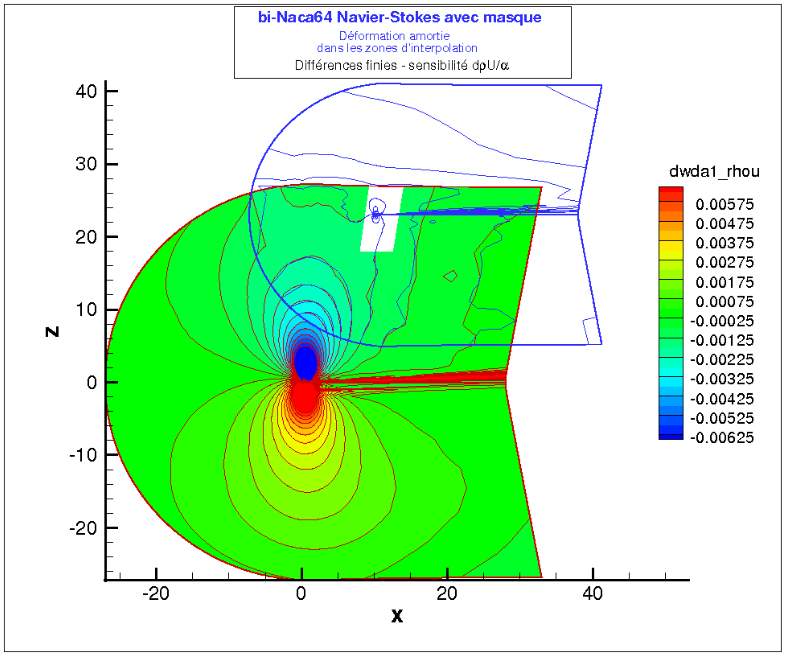
- ▷ **The full linearization of the Spalart Allmaras turbulent model**
- ▷ **The RPM method**
- ▷ **The chimera method**

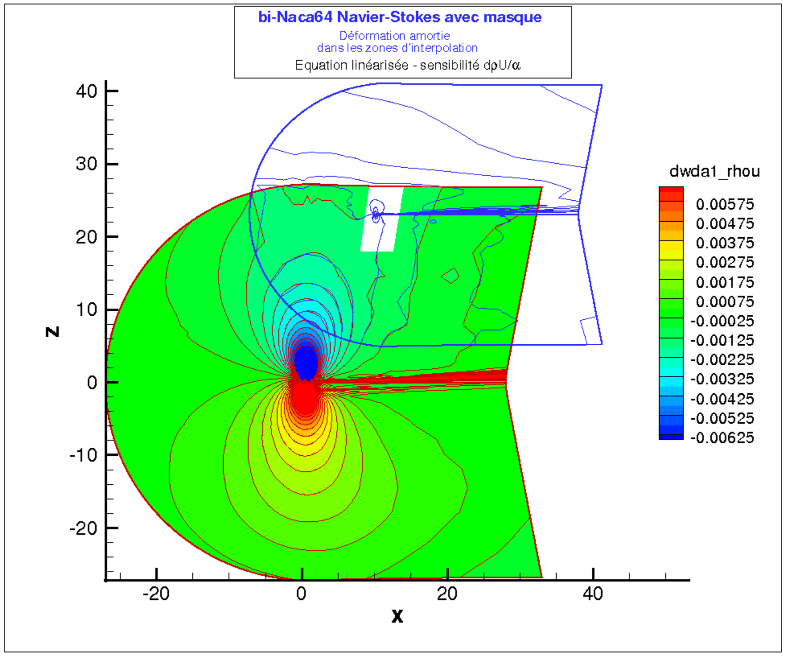
3- APPLICATIONS

The chimera method (1/2)

- Bi-NACA64A212 with mask
- 2D RANS equations. $M_\infty = 0.71$, $AoA = 0.25^o$, $Re = 2.\,10^6$
- 2 blocks: 257×63 and 257×65
- Design parameter: camber of the trailing edge of the upper profile
- Limit: blanking and interpolation zones not deformed by the design parameter

The chimera method (2/2)





	finite difference	adjoint equation	relative error
$dCLp/d\alpha$	-0.5099	-0.5220	2.37 %
$dCDp/d\alpha$	$-1.411\,10^{-2}$	$-1.494\,10^{-2}$	5.88 %

OUTLINE

1- DISCRETE ADJOINT SOLVER

2- RECENT ADVANCES

3- APPLICATIONS

Aircraft [a] [b]

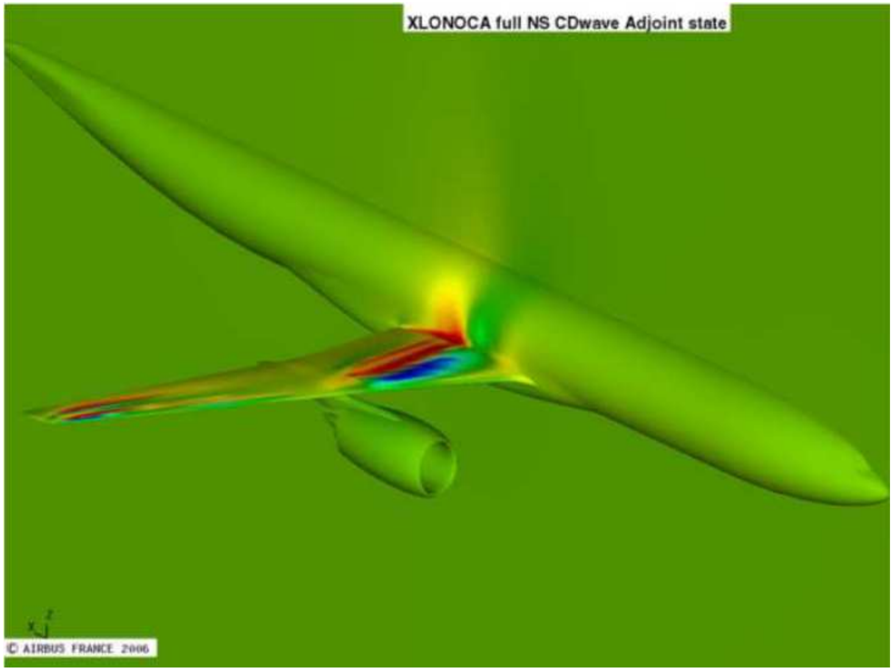


Figure 3: Adjoint vector component for CDw

[a] Salah El Din I, Carrier G, Mouton S. Discrete adjoint method in elsA (Part 2): Application to aerodynamic design optimization. Proceedings of the 7th ONERA-DLR Aerospace Symposium (ODAS), Toulouse, 2006

[b] Carrier G. Single and multi-point aerodynamic optimizations of a supersonic transport aircraft wing using optimization strategies involving adjoint method and genetic algorithm. Proceedings of ERCOFACT Workshop "Design optimization: methods and applications", Las Palmas, 2008

Turbomachinery

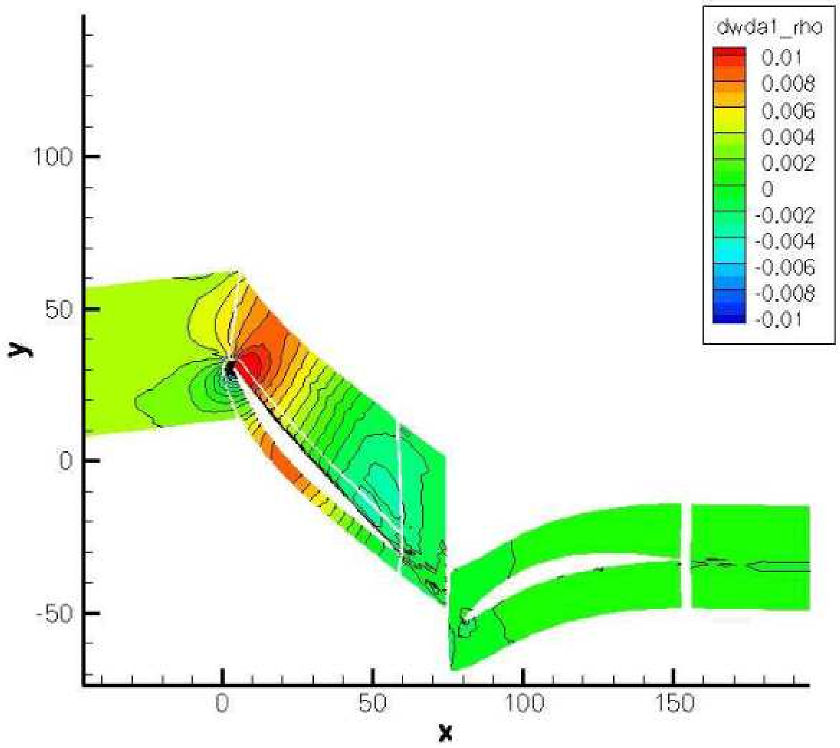


Figure 4: CME2 3D, rotor stator with clearance. $d\rho/d\alpha$

Rotor in hover (1/2) [a]

- Rotor ERATO
- 3D RANS equations. $M_{tip} = 0.616$, $Re_{tip} = 1.93\,10^6$
- 1 block: 807 457 nodes
- Design parameters: poles of Bezier curve driving chord, twist and sweep

[a] Dumont A, Le Pape A, Peter J, Huberson S. Aerodynamic shape optimization of hovering rotors using a discrete adjoint of the RANS equations. Proceedings of 5th American Helicopter Society annual forum, Grapevine, Texas, USA, 2009. To appear in AHS Journal.

Rotor in hover (2/2)

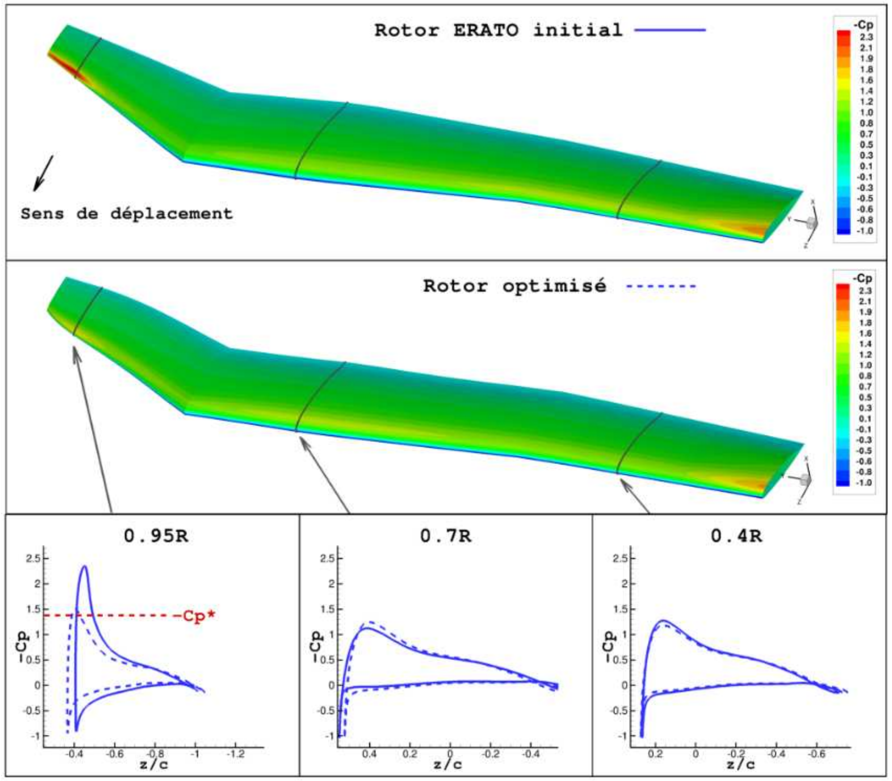


Figure 5: Rotor ERATO

THANK YOU FOR YOUR ATTENTION