# Discrete adjoint for CFD

J. Peter[1]

[1]ONERA DMFN

November 2018

# Outline

1. Discrete gradient calculation method

2. Discrete gradient methods, intuition and checks

3. Discrete adjoint for goal oriented mesh-refinement

# Outline



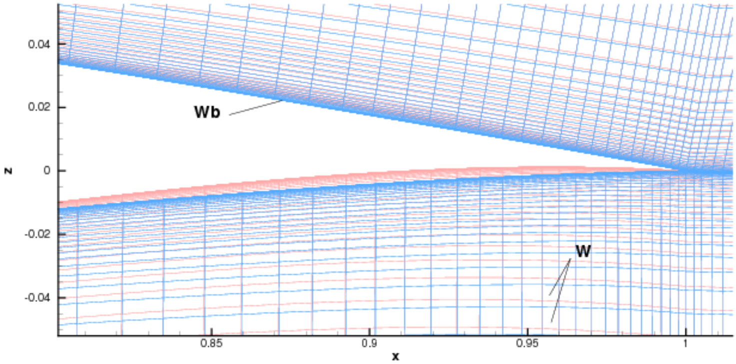
ONERA
THE FRENCH AEROSPACE LAB

# Discrete adjoint method. Parameters (1/2)

- Framework: compressible flow simulation using finite volume method. Discrete approach for sensitivity analysis

- Notations
  - Volume mesh $X$, flowfield $W$ (size $n_W$)
  - Wall surface mesh $S$
  - Residual $R$, $C^1$ regular w.r.t. $X$ and $W$ – steady state: $R(W,X) = 0$
  - Vector of design parameters $\alpha$ (size $n_\alpha$), $X(\alpha)$ $S(\alpha)$ $C^1$ regular

- Assumption of implicit function theorem
  - $\forall\ (W_i, X_i)\ /\ R(W_i, X_i) = 0 \quad (\partial R/\partial W)(W_i, X_i) \neq 0$
  - Unique steady flow corresponding to a mesh


0.04
0.02
0
-0.02
-0.04
z
0.85
0.9
0.95
1
x
Wb
W

# Discrete adjoint method. Parameters (2/2)

- Functions of interest
  - $\mathcal{J}_k(\alpha) = J_k(W(\alpha), X(\alpha))\ k \in [1, n_f]$
  - Flowfield and volume mesh linked by flow equations $R(W(\alpha), X(\alpha)) = 0$

- Sensitivities $d\mathcal{J}_k/d\alpha_i \quad k \in [1, n_f] \quad i \in [1, n_\alpha]$ to be computed

- Discrete gradient computation methods
  - Finite differences – $2n_\alpha$ flow computations (non linear problems, size $n_W$)
  - Direct differentiation method – $n_\alpha$ linear systems (size $n_W$)
  - Adjoint vector method – $n_f$ linear systems (size $n_W$)

ONERA
THE FRENCH AEROSPACE LAB

# Discrete adjoint method. Mesh (1/2)

- Framework: compressible flow simulation using finite volume method. Discrete approach for sensitivity analysis
- Notations
  - Volume mesh $X$, flowfield $W$ (size $n_W$)
  - Wall surface mesh $S$
  - Residual $R$, $C^1$ regular w.r.t. $X$ and $W$ – steady state: $R(W,X) = 0$
- Assumption of implicit function theorem
  - $\forall\ (W_i, X_i)\ /\ R(W_i, X_i) = 0 \quad (\partial R/\partial W)(W_i, X_i) \neq 0$
  - Unique steady flow corresponding to a mesh

ONERA
THE FRENCH AEROSPACE LAB

# Discrete adjoint method. Mesh (2/2)

- Functions of interest
  - $\bar{J}_k(X) = J_k(W,X)\ k \in [1, n_f]$
  - Flowfield and volume mesh linked by flow equations $R(W,X) = 0$

- Calculate $d\bar{J}_k/dX \quad k \in [1, n_f]$ to be computed

- Discrete adjoint only gradient computation
- Direct counterpart of adjoint-mesh requires calculation of $dW/dX$ which is $n_W \times n_X$ field ... not sustainable

ONERA
THE FRENCH AEROSPACE LAB

# Direct differentiation method

- Discrete equations for mechanics (set of $n_W$ non-linear equations )

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

- Differentiation with respect to $\alpha_i$ i $\in [1, n_\alpha]$. Derivation of $n_\alpha$ linear systems of size $n_W$

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

- Calculation of derivatives

$$\frac{d\mathcal{J}_k}{d\alpha_i} = \frac{\partial J_k}{\partial X}\frac{dX}{d\alpha_i} + \frac{\partial J_k}{\partial W}\frac{dW}{d\alpha_i}$$

ONERA
THE FRENCH AEROSPACE LAB

# Discrete adjoint parameter method (1/2)

- Several ways of deriving the equations of discrete adjoint method. The following also helps understanding continuous adjoint
- Following equalities hold $\forall \lambda \in \mathbb{R}^{nw}$

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

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

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

# Discrete adjoint parameter method (2/2)

- Vector $\lambda$ defined in order to cancel the factor of the flow sensitivity $\frac{dW}{d\alpha_i}$... the adjoint equation
- Then associated to $J_k$ and denoted $\lambda_k$

$$\frac{\partial J_k}{\partial W} + \lambda_k^T \frac{\partial R}{\partial W} = 0$$

- Calculation of derivatives

$$\forall i \in [1, n_\alpha] \qquad \frac{d\mathcal{J}_k(\alpha)}{d\alpha_i} = \frac{\partial J_k}{\partial X}\frac{dX}{d\alpha_i} + \lambda_k^T (\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

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

- Method with $n_f$ and not $n_\alpha$ linear systems (size $n_W$) to solve

# Iterative solution of direct and adjoint equation (1/5)

- CFD teams tend to mimic the solution of steady state flow altough flow equations are non-linear whereas direct/adjoint equations are linear
- Storing the Jacobian of the scheme and sending to direct solver has been done but is rare and is not tractable for large cases
- Iterative resolution is much more common.
  - Newton/relaxation algorithm

$$\left(\frac{\partial R}{\partial W}\right)^{(APP)\ T}\left(\lambda_k^{(l+1)}-\lambda_k^{(l)}\right)=-\left(\left(\frac{\partial R}{\partial W}\right)^T\lambda_k^{(l)}+\left(\frac{\partial J_k}{\partial W}\right)^T\right)$$

  - $p$-iteration restarted GMRES (General Minimum RESidual)

$$\left(\frac{\partial R}{\partial W}\right)^T\lambda_k^{(l+1)}=-\left(\frac{\partial J_k}{\partial W}\right)^T \quad \text{initialized by} \quad \lambda_k^{(l)}$$

# Iterative solution of direct and adjoint equation (2/5)

- Common Newton/relaxation algorithm for adjoint

$$\left(\frac{\partial R}{\partial W}\right)^{(APP)\ T}\left(\lambda_k^{(l+1)}-\lambda_k^{(l)}\right)=-\left((\frac{\partial R}{\partial W})^T\lambda_k^{(l)}+(\frac{\partial J_k}{\partial W})^T\right)$$

- Common Newton/relaxation algorithm for direct

$$\left(\frac{\partial R}{\partial W}\right)^{(APP)}\left((\frac{dW}{d\alpha_i})^{(l+1)}-(\frac{dW}{d\alpha_i})^{(l)}\right)=-\left((\frac{\partial R}{\partial W})\frac{dW}{d\alpha_i}^{(l)}+\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i}\right)$$

- Defining an approximate Jacobian $(\frac{\partial R}{\partial W})^{(APP)}$ is an old subject in compressible CFD (definition of implicit stages for backward-Euler schemes...)
  - upwind approximate linearization of convective flux
  - neglecting cross derivatives in linearization of viscous fluxes
  - ...
- Possibly adapting implicit stages and mutigrid algorithm (flow solver to adjoint solver)

ONERA
THE FRENCH AEROSPACE LAB

# Iterative solution of direct and adjoint equation (3/5)

- GMRES algorithm for approximate solution of exact linear problem

$$\left(\frac{\partial R}{\partial W}\right)^T \lambda_k^{(l+1)} = -(\frac{\partial J_k}{\partial W})^T$$

- GMRES seeks the vector of minimal $L_2$ residual on a Krylov space
- With $\lambda_k^{(l)}$ as initial guess,

$$r_0 = -(\frac{\partial J_k}{\partial W})^T - (\frac{\partial R}{\partial W})^T \lambda_k^{(l)} \qquad v_0 = r_0/||r_0||$$

- dimension $p$ Krylov subspace $\mathcal{K}$. Let us denote $A = (\frac{\partial R}{\partial W})^T$

$$\mathcal{K} = (v_0, Av_0, A^2 v_0, ..., A^{p-1} v_0)$$

# Iterative solution of direct and adjoint equation (4/5)

- GMRES algorithm for approximate solution of exact linear problem

$$\left(\frac{\partial R}{\partial W}\right)^T \lambda_k^{(l+1)} = -\left(\frac{\partial J_k}{\partial W}\right)^T$$

- dimension $p$ Krylov subspace $\mathcal{K}$
- Exact-Jacobian times vector is available in an adjoint code based on Newton-relaxation
- Compute $V_p$, orthogonal basis of $\mathcal{K}$, solve the minimization for $y_p$ coordinates on $V_p$, update with the approximate solution of linear system

$$\lambda_k^{(l+1)} = \lambda_k^{(l)} + V_p y_p$$

ONERA
THE FRENCH AEROSPACE LAB

# Iterative solution of direct and adjoint equation (5/5)

- Right-preconditionned GMRES is GMRES for

$$\left(\frac{\partial R}{\partial W}\right)^T \mathbf{M}^{-1} \gamma_k^{(l+1)} = -(\frac{\partial J_k}{\partial W})^T$$

$$\forall l \quad \mathbf{M}^{-1} \gamma_k^{(l)} = \lambda_k^{(l)}$$

- $v_0$ derived from $\lambda_k^{(l)}$ $(\gamma_k^{(l)})$
- dimension $p$ modified Krylov subspace and modified update ( $A = (\frac{\partial R}{\partial W})^T$ )

$$\mathcal{K} = (v_0, (A\mathbf{M}^{-1})v_0, (A\mathbf{M}^{-1})^2 v_0, ..., (A\mathbf{M}^{-1})^{p-1} v_0)$$

$$\lambda_k^{(l+1)} = \lambda_k^{(l)} + \mathbf{M}^{-1} V_p y_p$$

- Typical preconditionner (for elsA/Opt) are approximate-Jacobian LU-relaxation or GMRES preconditionned with approximate-Jacobian LU-relaxation

# Discrete adjoint mesh method (1/3)

- Vector $\lambda_k$ defined by

$$\frac{\partial J_k}{\partial W} + \lambda_k^T \frac{\partial R}{\partial W} = 0$$

- Calculation of derivatives

$$\forall i \in [1, n_f] \qquad \frac{d\mathcal{J}_k(\alpha)}{d\alpha_i} = \frac{\partial J_k}{\partial X}\frac{dX}{d\alpha_i} + \lambda_k^T (\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i})$$

$$\forall i \in [1, n_f] \qquad \frac{d\mathcal{J}_k(\alpha)}{d\alpha_i} = (\frac{\partial J_k}{\partial X} + \lambda_k^T \frac{\partial R}{\partial X})\frac{dX}{d\alpha_i}$$

- Obvious mathematical factorization. Huge practical importance.

# Discrete adjoint mesh method (2/3)

- Solve for adjoint vectors. CFD gradient computation code computes "only"

$$\frac{d\overline{J_k}}{dX} = \frac{\partial J_k}{\partial X} + \lambda_k^T \frac{\partial R}{\partial X}$$

  The functional outputs sensitivities $d\mathcal{J}_k(\alpha)/d\alpha_i$ are calculated later by a mesh/geometrical tool

- Pros : CFD has no knowledge of parametrization. Huge memory savings [Nielsen, Park 2005] Try several parametrization. Check $\frac{dJ_k}{dS}$ with engineers
- Cons: Matrix $\frac{\partial R}{\partial X}$ has to be explicitely computed (instead of $\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i}$ computable by finite differences) Hard work...

- Check adjoint-mesh mode $\frac{d\overline{J_k}}{dX}$ by individual nodes displacement, flow convergence, finite difference for function...

# Discrete adjoint mesh method (3/3)

- Solve for adjoint vectors computes "only" and compute

$$\frac{d\bar{J}_k}{dX} = \frac{\partial J_k}{\partial X} + \lambda_k^T \frac{\partial R}{\partial X}$$

- Cons: Matrix $\frac{\partial R}{\partial X}$ has to be explicitely computed (instead of $\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i}$ computable by finite differences) Hard work...

- How to calculate $\frac{d\tilde{J}_k}{dS}$ ?
  - Explicit link between $X$ and $S$

$$\frac{d\tilde{J}_k}{dS} = \frac{d\bar{J}_k}{dX}\frac{dX}{dS} \qquad \frac{d\mathcal{J}_k}{d\alpha_i} = \left[\frac{d\bar{J}_k}{dX}\frac{dX}{dS}\right]\frac{dS}{d\alpha_i}$$

  - Implicit link between $X$ and $S$ [Nielsen, Park 2005] adjoint equation for mesh deformation

# Outline

# Intuition about / interpretation of adjoint vector ? (1/2)

- Vector $\lambda$ defined by

$$\frac{\partial J}{\partial W} + \lambda^T \frac{\partial R}{\partial W} = 0$$

- Assume an arbitrary change $\delta R$ (very small fixed numbers) in $R$. $\delta W$ change in $W$ so that $W + \delta W$ satisfies new discrete flow equations

$$(R + \delta R)(W + \delta W, X) = 0 \qquad [\ R(W,X) + \ ]\ \ \delta R + \frac{\partial R}{\partial W}\delta W \simeq 0$$

- Change in $J$ due to change in flow $\delta W$

$$J(W + \delta W, X) \simeq J(W,X) + (\frac{\partial J}{\partial W})\delta W \qquad \delta J = -\frac{\partial J}{\partial W}(\frac{\partial R}{\partial W})^{-1}\delta R$$

- Identification of $\lambda^T$

$$\delta J = \lambda^T \delta R$$

ONERA
THE FRENCH AEROSPACE LAB

# Intuition about / interpretation of adjoint vector ? (2/2)

- Vector $\lambda$ defined by
$$\frac{\partial J}{\partial W} + \lambda^T \frac{\partial R}{\partial W} = 0$$
- Assume an arbitrary change $\delta R$ (very small fixed numbers) in $R$. $\delta W$ change in $W$ so that $W + \delta W$ satisfies new discrete flow equations. Change $\delta J$ in $J$ due to change in flow $\delta W$
$$\delta J = \lambda^T \delta R$$
- One tedious way to check one component of $\lambda$ (in one cell) is to locally perturbate the corresponding residual $R$ (one cell, one component). Calculate perturbed flow, perturbed function, function sensitivity, divide by residual perturbation...
- Interpretation of adjoint vector = “strong” in zones of strong influence on the function of interest. [[ Restriction: remind that $\lambda$ is not intrinsic. Only $\lambda$ times (derivtives of) $R$ are ]]

ONERA
THE FRENCH AEROSPACE LAB

# Some intuitions about adjoint vector ? (1/5)

- You can get understanding about adjoint from *continuous adjoint* where you see backwards propagation wrt to flow equations and a source term on the support of the function of interest. No time to give here an extensive presentation
- In 1D scalar toy problems
  - Time derivative $\frac{\partial u}{\partial t}$ gets $-\frac{\partial \lambda}{\partial t}$ Backward time integration
  - Convection term $\frac{\partial u}{\partial x}$ gets $-\frac{\partial \lambda}{\partial x}$ "Backward propagation" in adjoint state
    In a simple 1D linear convection problem, the state variable is to be given at initial time and inlet bound whereas the adjoint variable is to be given at final time and outlet bound
  - Diffusion term $\frac{\partial^2 u}{\partial x^2}$ gets $\frac{\partial^2 \lambda}{\partial x^2}$

ONERA
THE FRENCH AEROSPACE LAB

# Some intuitions about adjoint vector ? (2/5)

- Supersonic inviscid flow $M_\infty = 1.5$ $AoA = 1^o$

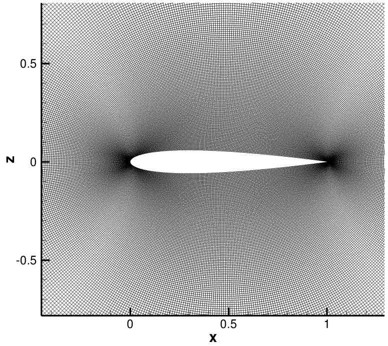



Figure: 513 × 513 mesh

# Some intuitions about adjoint vector ? (3/5)

- Supersonic inviscid flow $M_\infty = 1.5$ $AoA = 1^o$

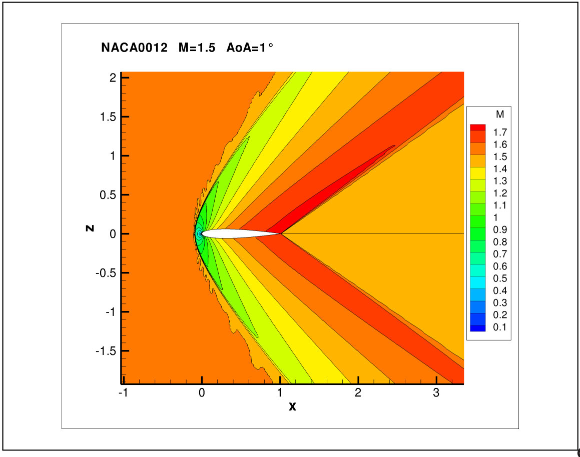



Figure: iso-lines of Mach number

ONERA
THE FRENCH AEROSPACE LAB

# Some intuitions about adjoint vector ? (4/5)

- Supersonic inviscid flow $M_\infty = 1.5$ $AoA = 1^o$

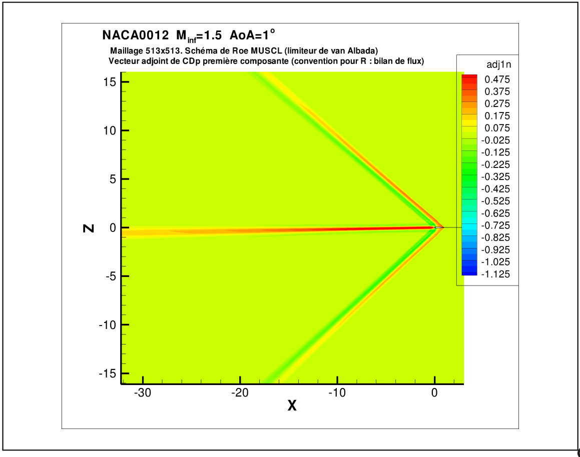



Figure: First component of adjoint vector for $CDp$

ONERA THE FRENCH AEROSPACE LAB

# Some intuitions about adjoint vector ? (5/5)

- Supersonic inviscid flow $M_\infty = 1.5$ $AoA = 1^o$

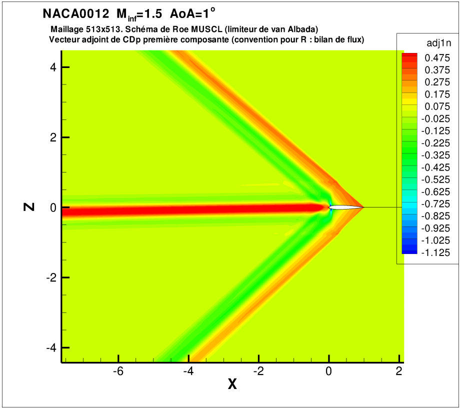



Figure: First component of adjoint vector for $CDp$ (closer)

# Checking direct differentiation method

- Gradient vectors

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

- First check the flow sensitivities (solution of direct equation) using finite differences

$$R(W(\alpha+\delta\alpha_i), X(\alpha+\delta\alpha_i)) = 0 \quad R(W(\alpha-\delta\alpha_i), X(\alpha-\delta\alpha_i)) = 0$$

$$\frac{dW}{d\alpha_i} ? \simeq \frac{W(\alpha+\delta\alpha_i) - W(\alpha-\delta\alpha_i)}{2\delta\alpha_i}$$

- Then check the outputs sensitivities

$$\frac{d\mathcal{J}_k}{d\alpha_i} ? \simeq \frac{J_k(W(\alpha+\delta\alpha_i), X(\alpha+\delta\alpha_i)) - J_k(W(\alpha-\delta\alpha_i), X(\alpha+\delta\alpha_i))}{2\delta\alpha_i}$$

ONERA
THE FRENCH AEROSPACE LAB

# Checking discrete adjoint-parameter method (1/2)

- Checking adjoint method... much more difficult than checking direct differentiation method.

- If
$$\frac{d\mathcal{J}_k}{d\alpha_i} <> \frac{J_k(W(\alpha+\delta\alpha_i),X(\alpha+\delta\alpha_i)) - J_k(W(\alpha-\delta\alpha_i),X(\alpha+\delta\alpha_i))}{2\delta\alpha_i}$$
no easy checking procedure

- In an iterative resolution method is used, of course the gradient accuracy depends on the $(\frac{\partial R}{\partial W})^T \lambda_k^{(l)}$ operation plus gathering of gradient terms

ONERA
THE FRENCH AEROSPACE LAB

# Verification of discrete adjoint-parameter method (2/2)

- Interpretation of adjoint as sensitivity of function to residual may be used to check $\lambda$.

$$\delta J = \lambda^T \delta R$$

  Rarely done (?)

- With adjoint & direct iterative solvers in the same framework, with well-checked direct solver, use duality checks. $(U, V)$ two column vectors of $\mathbb{R}^{nw}$

$$U^T(\frac{\partial R}{\partial W})V = \left(U^T(\frac{\partial R}{\partial W})\right)_{adj-code}.V = U^T.\left((\frac{\partial R}{\partial W})V\right)_{lin-code}$$

- Valid for individual flux routine. Valid for part of the interfaces (border, joins, interior)... useful with *elsA*. Uncomment specific parts of elsA/Opt to run duality tests.

# Verification of discrete adjoint-mesh method

- It is possible to compute a finite difference of reference

$$R(W(X+\delta X_l), X+\delta X_l) = 0 \quad R(W(X-\delta X_l), X-\delta X_l) = 0$$

$$\frac{d\bar{J}_k}{dX_l} \simeq \frac{J_k(W(X+\delta X_l), X+\delta X_l) - J_k(W(X-\delta X_l), X-\delta X_l)}{2\delta X_l}$$

- If adjoint vector is well checked but ajoint mesh sensitivity is wrong...

$$\frac{d\bar{J}_k}{dX} = \frac{\partial J_k}{\partial X} + \lambda_k^T \frac{\partial R}{\partial X}$$

  bug is an explicit operation and should be found after some checks (finite-difference for $\frac{\partial R}{\partial X}$, $\frac{\partial J_k}{\partial X}$ ...)

- If volume mesh sensitivity is correct but surface mesh sensitivity is not, check $\frac{dX}{dS}$ and corresponding product

ONERA
THE FRENCH AEROSPACE LAB

# Outline



ONERA
THE FRENCH AEROSPACE LAB

# References and principle

- References
  - Venditti and Darmofal formulas (stemms from Giles et al.)
  - Grid adaptation for functionnal outputs : application to two-dimensional inviscid flows. JCP 176,40-69 (2002)
  - Anisotropic grid adaptation for functionnal outputs : application to two-dimensional viscous flows. JCP 187,22-46 (2003)

ONERA
THE FRENCH AEROSPACE LAB

# References and principle

- Principle
  - direct problem: solve $Au = f$ before computing $g^T u$ approximate solve $Au_h \simeq f$ before computing $g^T u_h$
  - adjoint problem: solve $A^T v = g$ before computing $f^T v \quad (= g^T u)$ approximate solve $A^T v_h \simeq g$ before computing $f^T v_h$
  - what about error $g^T(u - u_h)$ ?

$$
\begin{aligned}
g^T(u - u_h) &= v^T A(u - u_h) \\
g^T(u - u_h) &= v_h^T A(u - u_h) + (v - v_h)^T A(u - u_h) \\
g^T(u - u_h) &= v_h^T (f - Au_h) + (v - v_h)^T A(u - u_h)
\end{aligned}
$$

# Goal oriented mesh adaption

## Objective & Notations

**Get a good approximation $J_h(W_h, X_h)$ from current level computations ???**

- Current level $H$ mesh $X_H$, flowfield $W_H$, finite-volume equations for fluid dynamics $R(W_H, X_H) = 0$
- Fine level $h$ mesh $X_h$, flow field $W_h$, finite-volume equations for fluid dynamics $R(W_h, X_h) = 0$
- Aerodynamic coefficient $J$ fine level $J(W_h, X_h)$ , current level $J(W_H, X_H)$
- Abridged notations for this section

$$R(W, X_H) = R_H(W) \qquad R(W, X_h) = R_h(W)$$

$$J(W, X_H) = J_H(W) \qquad J(W, X_h) = J_h(W)$$

# Goal oriented mesh adaption

## Venditti & Darmofal

- Current level analysis $R_H(W_H) = 0$
  Interpolated flow: $W_h^H$
  No fine level analysis allowed (but for checks)

- Possibly current level adjoint for $J$ $\quad \lambda_H^T(\frac{\partial R_H}{\partial W}) = -(\frac{\partial J}{\partial W})_H$
  Interpolated adjoint vector $(\lambda_h^H)$
  Possibly adjoint equation at extrapolated flow
  $(\lambda_h\big|_{W_h^H})^T(\frac{\partial R_h}{\partial W_h}\Big|_{W_h^H}) = -\frac{\partial J_h}{\partial W_h}\Big|_{W_h^H}$

- Aerodynamic coefficient $J$
  current level $J_H(W_H)$
  fine level $J_h(W_h)$ (unknown but in case of a check)
  fine level for extrapolated flow $J_h(W_h^H)$

# Goal oriented mesh adaption

## Venditti & Darmofal

- Exact formula

$$J_h(W_h) = J_h(W_h^H) + (\lambda_h\Big|_{W_h^H})^T R_h(W_h^H) + \mathcal{O}(||W_h - W_h^H||^2)$$

- Useful formula for actual CFD

$$J_h(W_h) = J_h(W_h^H) + \underbrace{(\lambda_h^H)^T R_h(W_h^H)}_{computable\ \ correction} + \underbrace{((\lambda_h\Big|_{W_h^H})^T - (\lambda_h^H)^T) R_h(W_h^H)}_{error\ \ in\ \ computable\ \ correction} + \mathcal{O}(||W_h - W_h^H||^2)$$

- correction = what you would like to compute
- error in correction = what is hopefully neglectible
- NB Just the same formula from any other approximate field on fine grid (badly converged, assimilated...)

# Demonstration

- Taylor expansion of aerodynamic function $J$

$$J_h(W_h) = J_h(W_h^H) + (\frac{\partial J}{\partial W}\Big|_{W_h^H})(W_h - W_h^H) + \mathcal{O}(||W_h - W_h^H||^2)$$

- Adjoint-like equation

$$(\lambda_h\big|_{W_h^H})^T(\frac{\partial R_h}{\partial W_h}\Big|_{W_h^H}) = -\frac{\partial J_h}{\partial W_h}\Big|_{W_h^H}$$

- Expression of fine grid function value

$$\begin{aligned} J_h(W_h) &= J_h(W_h^H) - (\lambda_h\big|_{W_h^H})^T(\frac{\partial R_h}{\partial W_h}\Big|_{W_h^H})(W_h - W_h^H) + \mathcal{O}(||W_h - W_h^H||^2) \\ J_h(W_h) &= J_h(W_h^H) + (\lambda_h\Big|_{W_h^H})^T R_h(W_h^H) + \mathcal{O}(||W_h - W_h^H||^2) \end{aligned}$$

# Error in computational correction

## Three expressions

- $$ECC_1 = \left( \left( \lambda_h \Big|_{W_h^H} \right)^T - (\lambda_h^H)^T \right) R_h(W_h^H)$$

- Residual of adjoint equation at interpolated flow

$$R_h^\lambda(\lambda) = \left[ \frac{\partial R_h}{\partial W_h} \Big|_{W_h^H} \right]^T \lambda - \left( \frac{\partial J_h}{\partial W_h} \Big|_{W_h^H} \right)^T = \left[ \frac{\partial R_h}{\partial W_h} \Big|_{W_h^H} \right]^T (\lambda - \lambda_h|_{W_h^H})$$

$$ECC_2 = ECC_1 = - \left( R_h^\lambda(\lambda_h^H) \right)^T \left[ \frac{\partial R_h}{\partial W_h} \Big|_{W_h^H} \right]^{-1} R_h(W_h^H)$$

- Use $R_h(W_h^H) = R_h(W_h^H) - R_h(W_h)$

$$ECC_3 = \left( R_h^\lambda(\lambda_h^H) \right)^T (W_h - W_h^H) \simeq ECC_{1-2}$$

# Goal oriented mesh refinement stategy

## Venditti & Darmofal

- Method proposed by Venditti and Darmofal
- New "function evaluation" $J_h(W_H^h) + \lambda_h^{H\ T} R_h(W_h^H)$
- Reduce error in computable solution

$$ECC_2 = -\left(R_h^\lambda(\lambda_h^H)\right)^T \left[\left.\frac{\partial R_h}{\partial W_h}\right|_{W_h^H}\right]^{-1} R_h(W_h^H)$$

- Reducing simultaneously $R_h^\lambda(\lambda_h^H)$ and $R_h(W_h^H)$
- Successfully demonstrated in the articles quoted before

# Goal oriented mesh refinement stategy

## Alternative approach

- Alternative strategy if $J_H(W_H)$ is close to $J_h(W_h^H)$
- General property
  $J_h(W_h) = J_h(W_h^H) + (\lambda_h|_{W_h^H})^T R_h(W_h^H) + \mathcal{O}(||W_h - W_h^H||^2)$

  Specifically in this case
  $J_h(W_h) \simeq J_H(W_H) + (\lambda_h|_{W_h^H})^T R_h(W_h^H) + \mathcal{O}(||W_h - W_h^H||^2)$
- The correction term is an indicator for goal-oriented mesh refinement (once again, not the standard one)

ONERA
THE FRENCH AEROSPACE LAB

# Research activities on adjoint method

- J. Peter and R.P. Dwight. *Computers and Fluids* 2010
- A. Dumont et al. *AHS Journal* 2011

- J. Peter et al *Computers and Fluids* 2012
- M. Nguyen-Dinh et al. *European Journal of Mechanics B/Fluids* 2014
- G. Todarello et al. *Journal of Computational Physics* 2016
- A. Resmini et al. *International Journal for Numerical Methods in Fluids* 2016

- Last four articles propose a method for goal-oriented mesh adaptation
- This method provides an error indicator based on for order changes in $J$ when nodes move individually in the polygon defined by their neighbors

# d$J/dX$ for mesh adaptation ? (1/2)

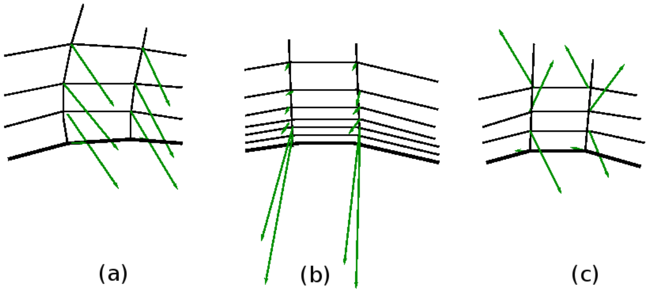
- Is $dJ/dX$ (vector field) or $||dJ/dX||$ valuable information for $J$-oriented mesh adaptation ?

  Research activity started 2012 at ONERA

- $||dJ/dX||$ times local characteristic size is a useful indicator for $J$-oriented mesh adaptation

ONERA
THE FRENCH AEROSPACE LAB

# $dJ/dX$ for mesh adaptation ? (2/2)




- Visualization of $dJ/dX$ (vector field) or $||dJ/dX||$ (scalar field)
- Analysis based on $J(X + dX) - J(X) \simeq (dJ/dX).dX$
  - Mesh (a) not well-suited for $J$ calculation
  - Mesh (b) possibly well-suited for $J$ calculation
  - Mesh (c) for $J$ calculation. Questionable

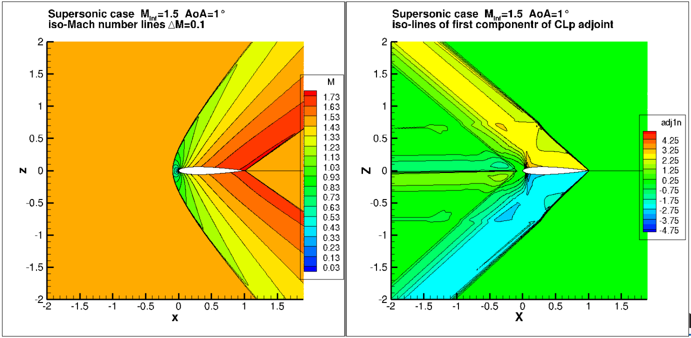
# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

- ONERA *elsA* code [Cambier, Heib, Plot 2013]
- Unstructured mesh. Roe-MUSCL scheme (van Albada limiting function)
- Adjoint capability. $\theta$-based refinement
- Remeshing MMG2D/MMG3D [Dobrzynksi 2012]
- Giovanni Todarello and Floris Vonck (Master of Science. TU Delft)

- Series of caclulation and mesh adaptation for NACA0012. Transonic $M_\infty = 0.85$ $AoA = 2^o$, $M_\infty = 0.95$ $AoA = 0^o$ – and supersonic – $M_\infty = 1.5$ $AoA = 1^o$ – flow conditions
- Final mesh compared to the one published in [Dwight 2008]
- Also consistent with mesh obtained by [Venditti Darmofal 2002] reference method

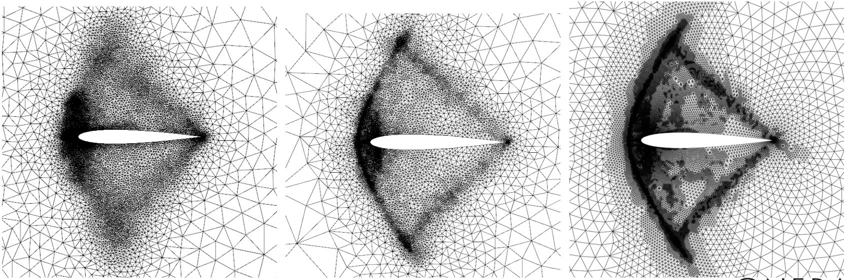
# d$J/dX$ for$J$-oriented unstructured mesh adaptation

- NACA0012 AoA=1$^o$, M=1.5 CLp calculation.
- Roe-MUSCL scheme (van Albada limiting function)
- Supersonic flow conditions. iso-Mach number lines (left), iso-$\lambda^1_{CLp}$ lines (right)



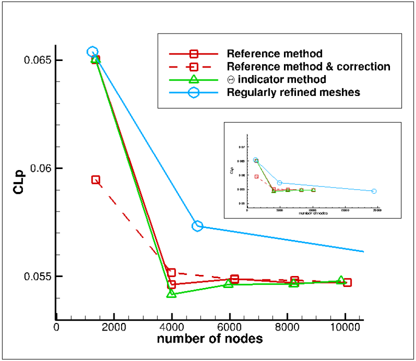
# d$J$/$dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=$1^o$, M=1.5 CLp calculation. Analysis of refined mesh zones
- Roe-MUSCL scheme (van Albada limiting function)
- Adapted meshes. Venditti and Darmofal's method, proposed $\theta$-indicator method, Dwight's method (left to right)



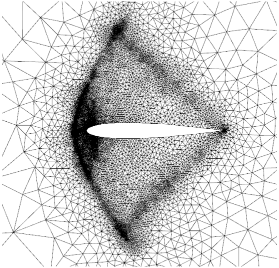
# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

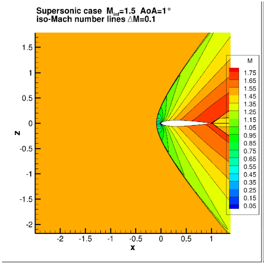
- NACA0012 AoA=1$^o$, M=1.5 CLp calculation.
- Roe-MUSCL scheme (van Albada limiting function)
- Comparison of convergence towards CLp limiting value ( CLp-lim = 0.05478 )

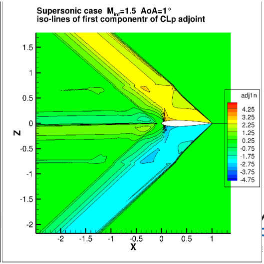


# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=1°, M=1.5 CLp calculation. (Roe-MUSCL scheme van Albada limiting function)
- Analysis of adjoint field from simple waves theory for (continuous adjoint equation of) supersonic flow
- Three simple waves starting from function support (source term in continuous adjoint equation) with theoretical angles derived from constant flow assumption
- Downwind bound of dense zones OK. Upwind bound ???



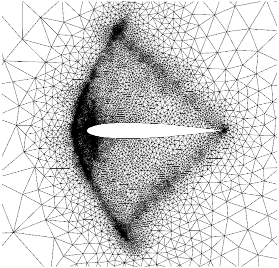


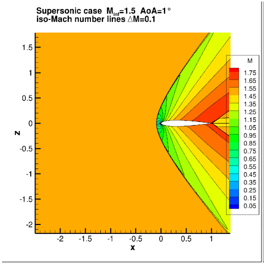



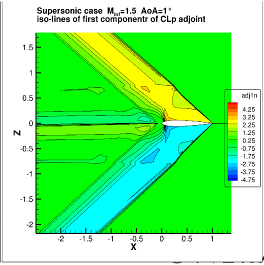
# d$J/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=1$^o$, M=1.5 CLp calculation. (Roe-MUSCL scheme van Albada limiting function)

- Low densitiy mesh zone upwind the section of shock-wave ?
- Zone of constant supersonic flow. No refinement
- In terms of $\theta$-indicator, low $\partial R/\partial X$

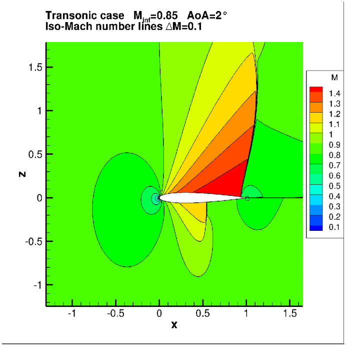


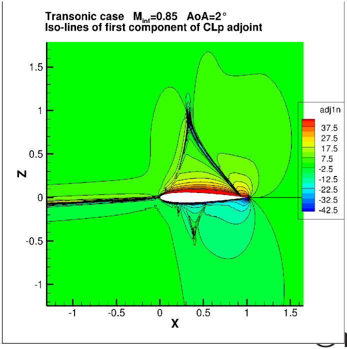


# d$J/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=2$^o$, M=0.85 CLp calculation. Analysis of adjoint field
- Roe-MUSCL scheme (van Albada limiting function)
- Transonic flow conditions. iso-Mach number lines (left), iso-$\lambda^1_{CLp}$ lines (right)



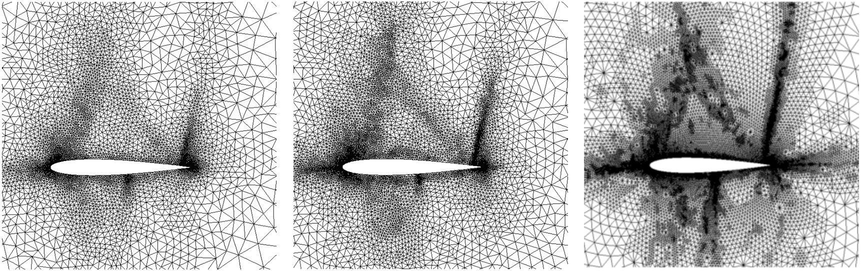




ONERA

THE FRENCH AEROSPACE LAB

# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

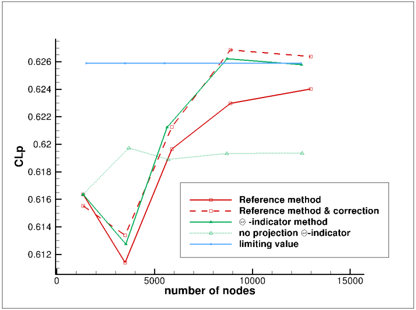
- NACA0012 AoA=2$^o$, M=.85 CLp calculation.
- Roe-MUSCL scheme (van Albada limiting function)
- Transonic flow conditions. Venditti and Darmofal's method, proposed $\theta$-indicator method, Dwight's method (left to right)



ONERA
THE FRENCH AEROSPACE LAB

# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=2°, M=.85 CLp calculation.
- Roe-MUSCL scheme (van Albada limiting function)
- Convergence towards CLp limiting value ( CLp-lim=0.6258 )




ONERA
THE FRENCH AEROSPACE LAB

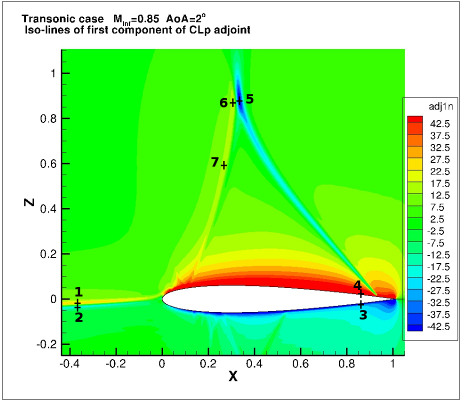
# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=$2^o$, M=0.85 CLp calculation. Analysis of adjoint field
- Dense mesh zones
  - Strong gradient of flow = shock waves
  - Strong gradient of adjoint
- Reason for hat-shaped zone of strong value / strong gradient of $\lambda_{CLp}$ ?
  - $\lambda_{CLp}$ ($\lambda^1_{CLp}$) is the sensitivity of CLp to a change in explicit residual $R$ ($R_1$) (reconverging flow-field)
  - Most often one interpretation of adjoint
  - Here actually coded in elsA

ONERA
THE FRENCH AEROSPACE LAB

# $dJ/dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=2°, M=0.85 CLp calculation. Analysis of adjoint field
- Selection of points for explicit residual perturbation



# d$J$/$dX$ for $J$-oriented unstructured mesh adaptation

- NACA0012 AoA=2$^o$, M=.85 CLp calculation. Analysis of adjoint field
- Change of flow at point 5 and 7 due to explicit residual perturbation