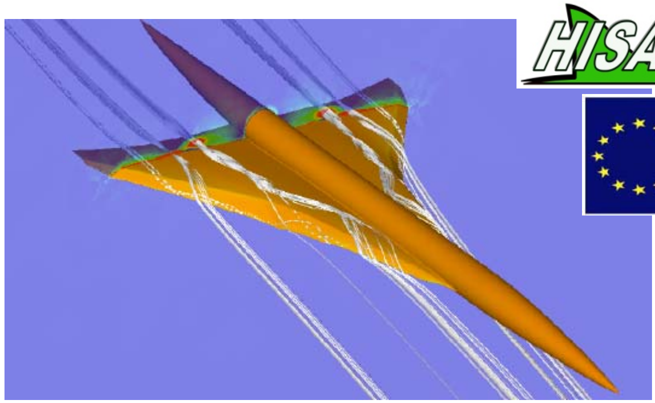
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
Parallel CFD & Optimization Unit
Laboratory of Thermal Turbomachines

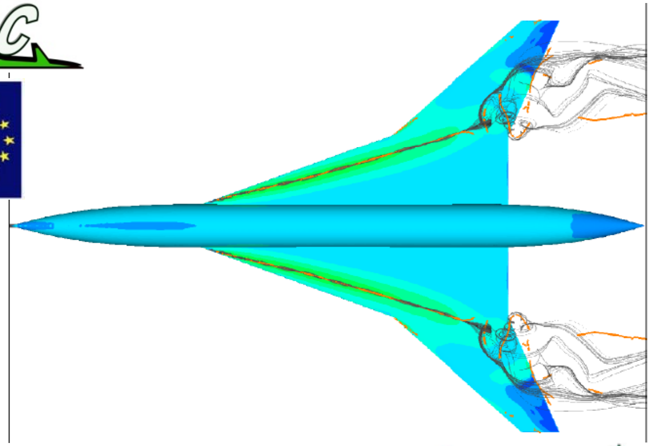
The Continuous Adjoint Method in Aerodynamic Shape Optimization, Robust Design, Flow Control & Topology Optimization

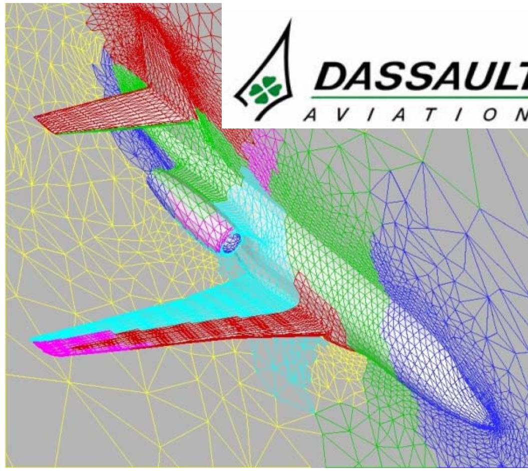
AIAC-2013
Ankara-Turkey

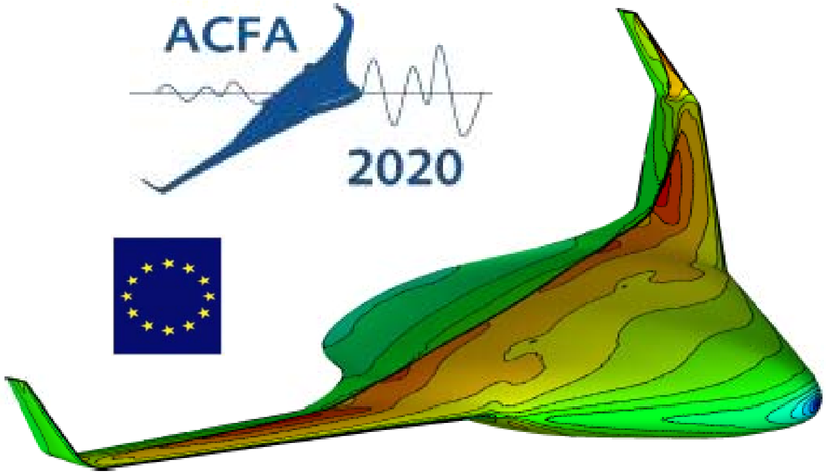
Kyriakos C. GIANNAKOGLOU, Professor NTUA
kgianna@central.ntua.gr
http://velos0.ltt.mech.ntua.gr/research/

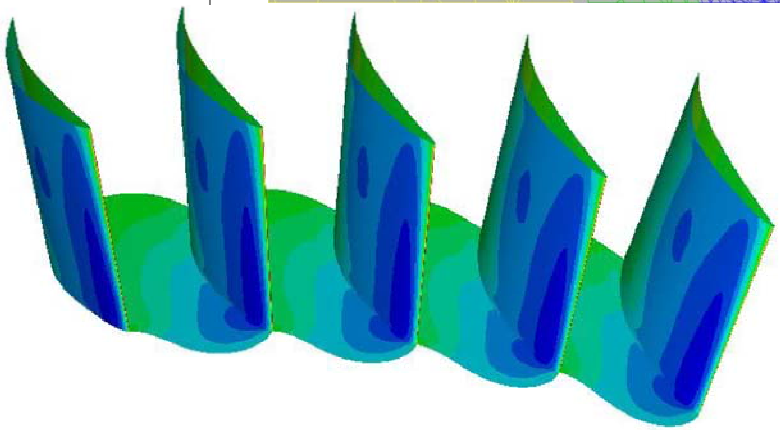
CFD Simulations based on the in-house PUMA code









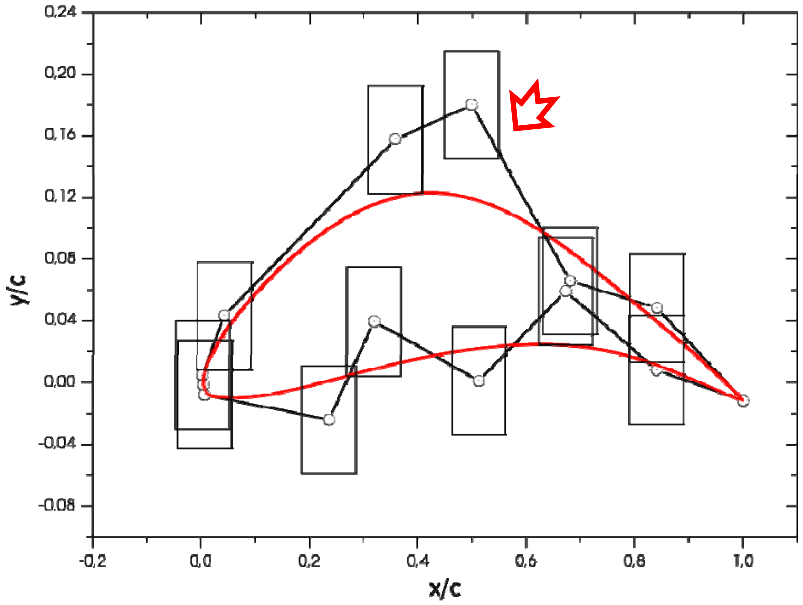


Compressible flow simulations are based on the fully-parallelized and GPU-enabled in-house PUMA code. Incompressible flows are simulated using either the inco-PUMA code or OpenFOAM. Supported by grid generation methods/software.

EASY: The Evolutionary Algorithms SYstem

A generic-purpose optimization platform which

- **accommodates any evaluation tool (CFD, CSM, CEM, etc),**
- **runs on any parallel system,**
- **solves single- and multi-objective optimization problems (SOO & MOO),**
- **solves constrained or unconstrained problems and**
- **is suitable for computationally expensive problems.**



$$x_j(t) = \sum_{i=0}^{M-1} C_i(t) X_{ij}$$

$$b_i, \ i = 1, \dots, N$$

$$\vec{b} \epsilon R^N$$

Objective Function (min.):

$$F = \frac{1}{2} \int_0^1 \left(p(x) - \underbrace{p_{tar}(x)}_{given}\right)^2 dx$$

Requested:

$$\frac{\delta F}{\delta b_m} = \int_0^1 \left(p(x) - p_{tar}(x)\right) \frac{\delta p}{\delta b_m} dx$$

Constraint:

Flow Equations = 0

Optimization Method:

Steepest Descent,

Quasi Newton,

Newton,…

Direct Differentiation (DD) Approach (instead of …)

Discrete DD:

$$\frac{dF}{db_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{db_i} \qquad \frac{dR_m}{db_i} = \frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{dU_k}{db_i} = 0$$

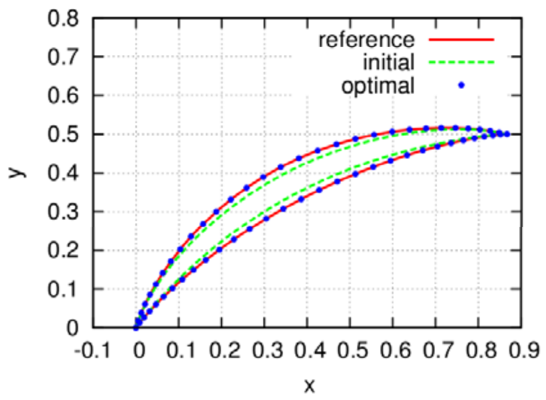
Continuous DD:

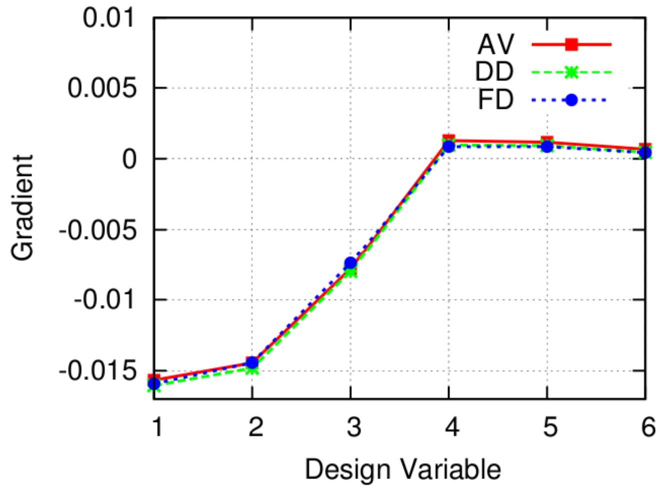
$$\frac{\delta}{\delta b_i}\left(\frac{\partial f^{inv}_{nk}}{\partial x_k}\right) = 0 \Rightarrow \frac{\partial}{\partial x_k}\left(A_{nmk}\frac{\partial U_m}{\partial b_i}\right) + \frac{\partial^2 f^{inv}_{nk}}{\partial x_k \partial x_l}\frac{\delta x_l}{\delta b_i} = 0$$

… plus the same for the state boundary conditions.

Why DD?

► **Validate the adjoint-based sensitivities (easily programmable, expensive).**





Inverse Design of a 2D Compressor Cascade

(Continuous Adjoint)

Comparison of grad(F) from:

AV=*adjoint (variable) method*

DD=*direct-differentiation*

FD=*finite-differences*

► **DD is an indispensable component of methods computing higher-order derivatives.**

Adjoint Methods: Discrete or Continuous Approach?

Discrete Adjoint:

First-discretize, then-differentiate

Continuous Adjoint:

First-differentiate, then-discretize

Hybrid Adjoint:

"half" discrete, "half" adjoint

The *Think-Discrete-Do-Continuous* Approach

Continuous adjoint where the adjoint PDEs are discretized in a way that reproduces the result of discrete adjoint.

Activities related to the development of Adjoint Methods

- **Development of both <u>continuous</u> and discrete adjoint methods.**
- **For compressible fluids (in-house, primitive variable solver, GPU-enabled).**
- **For incompressible fluids (OpenFOAM or in-house code. Pseudo-compressibility, GPU-enabled).**
- **For steady & unsteady flows (check-pointing, storage of approximates).**
- **For shape, flow-control, robust-design and topology optimization problems.**
- **Internal (turbomachinery) & external aerodynamics (cars, wings).**
- **Emphasis to continuous adjoint method for turbulence models.**
- **Calculation of high-order sensitivities, using both continuous & discrete adjoint.**

Acknowledgement

Dr. Dimitrios Papadimitriou
Dr. Evangelos Papoutsis-Kiachagias
Dr. Alexandros Zymaris
Dr. Evgenia Kontoleontos
Dr. Varvara Asouti
Ioannis Kavvadias
Konstantinos Tsiakas

The commonly used approach - The "frozen turbulence assumption"

Demonstrated for incompressible flows, exists & runs also for compressible flows

- **State Equations**

$$R^p = \frac{\partial v_j}{\partial x_j} = 0$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] = 0$$

(plus the turbulence model eqs.)

- **Development of the Adjoint Equations & Boundary Conditions**

For any objective function F:

$$F_{aug} = F + \int_\Omega u_i R_i^v d\Omega + \int_\Omega q R^p d\Omega$$

Differentiate F_{aug} w.r.t. to b_m, where b_m are the N design variables…

- **Adjoint Equations**

$$R^q = \frac{\partial u_j}{\partial x_j} = 0$$

$$R_i^u = -v_j \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - (\nu + \nu_t)\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{\partial q}{\partial x_i} = 0$$

and the adjoint boundary conditions…

Sensitivity Derivatives including only Boundary Integrals

Even if the objective function includes Field Integrals

• <u>Sensitivity Derivatives</u>

$$\frac{\delta F}{\delta b_m} = \int_{S_W} \frac{\partial F_{S_W}}{\partial x_k}\frac{\delta x_k}{\delta b_m} dS + \int_{S_W} F_{S_W}\frac{\delta (dS)}{\delta b_m} - \int_{S_W}\left[(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)n_j - q n_i\right]\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b_m} dS$$
$$+ \int_{S_W} u_i R_i^v \frac{\delta x_k}{\delta b_m} n_k dS + \int_{S_W} q R^p \frac{\delta x_k}{\delta b_m} n_k dS + \int_{S_W} (\nu+\nu_t)\frac{\partial F_{S_W}}{\partial p}\frac{\partial}{\partial x_k}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\frac{\delta x_k}{\delta b_m} n_i n_j dS$$
$$+ \int_{S_W} (\nu+\nu_t)\frac{\partial F_{S_W}}{\partial p}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\frac{\delta (n_i n_j)}{\delta b_m} dS$$

Advantages!

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'A Continuous Adjoint Method with Objective Function Derivatives Based on Boundary Integrals for Inviscid and Viscous Flows', Computers & Fluids, Vol. 36, pp. 325-341, 2007.

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Total Pressure Losses Minimization in Turbomachinery Cascades, Using a New Continuous Adjoint Formulation', Proc. IMechE, Part A: Journal of Power and Energy (Special Issue on Turbomachinery), Vol. 221, pp. 865-872, 2007.

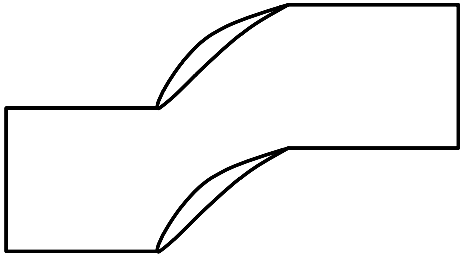
Types of F often used in Turbomachinery Applications

Apart from Lift/Drag etc used in External Aerodynamics

♦ $$F = \frac{1}{2}\int_{S_w} (p - p_{target})^2 dS$$

- Inverse design.
- Functional and design variables correspond to the same boundary !!!

♦ $$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$



- Losses Minimization.
- Functional and design variables correspond to different boundaries !!!

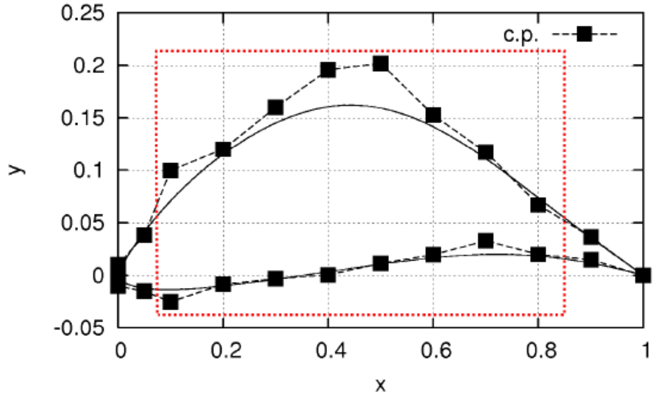
♦ $$F = \int_{S_{out}} \rho V_n s dS - \int_{S_{in}} \rho V_n s dS = \int_{\Omega} \rho u_i \frac{\partial s}{\partial x_i} d\Omega = \int_{\Omega} \frac{1}{T} \tau_{ij} \frac{\partial u_i}{\partial x_j} d\Omega$$

- Losses Minimization.
- Transformation of the inlet/outlet integral to a field integral !!!

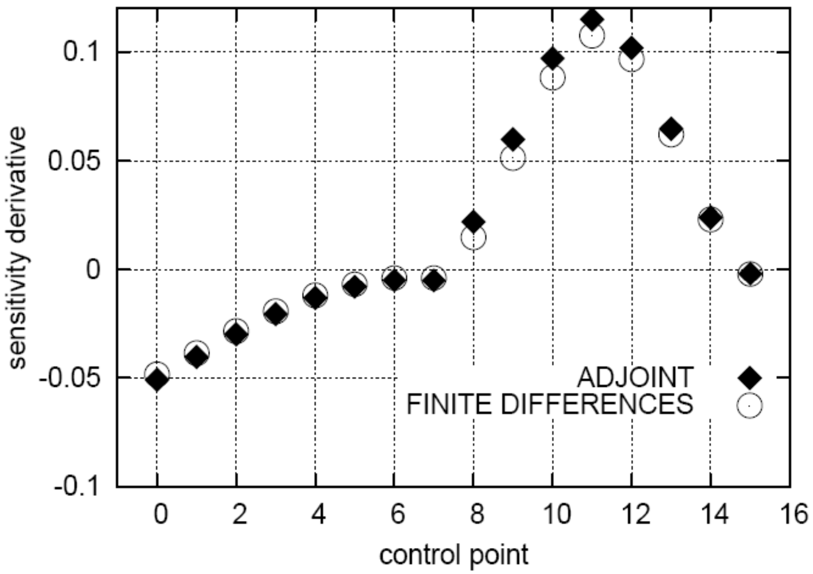
Computation of Sensitivity Derivatives on the starting airfoil

Laminar Flow, Subsonic Flow, stagger angle & solidity are fixed

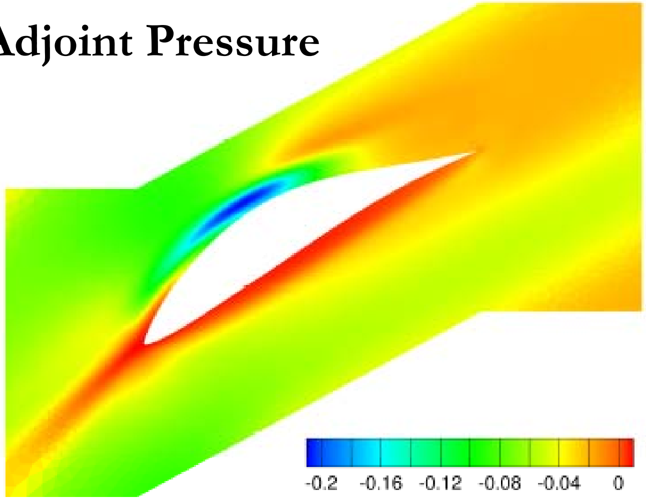
- Without running the Optimization Loop -



$$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$



Adjoint Pressure



Exact Differentiation of the Turbulence Model Eqs.

Demonstrated for incompressible flows, exists & runs also for compressible flows

Demonstrated for the Spalart-Allmaras model. Exists for k-ε & k-ω SST

$$R^p = \frac{\partial v_j}{\partial x_j} = 0$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] = 0 \qquad \nu_t = \widetilde{\nu} f_{v_1}$$

$$R^{\widetilde{\nu}} = \frac{\partial (v_j \widetilde{\nu})}{\partial x_j} - \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[(\nu + \widetilde{\nu})\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu} P(\widetilde{\nu}) + \widetilde{\nu} D(\widetilde{\nu}) = 0$$

$$F_{aug} = F + \int_\Omega u_i R_i^v d\Omega + \int_\Omega q R^p d\Omega + \boxed{\int_\Omega \widetilde{\nu_a} R^{\widetilde{\nu}} d\Omega}$$

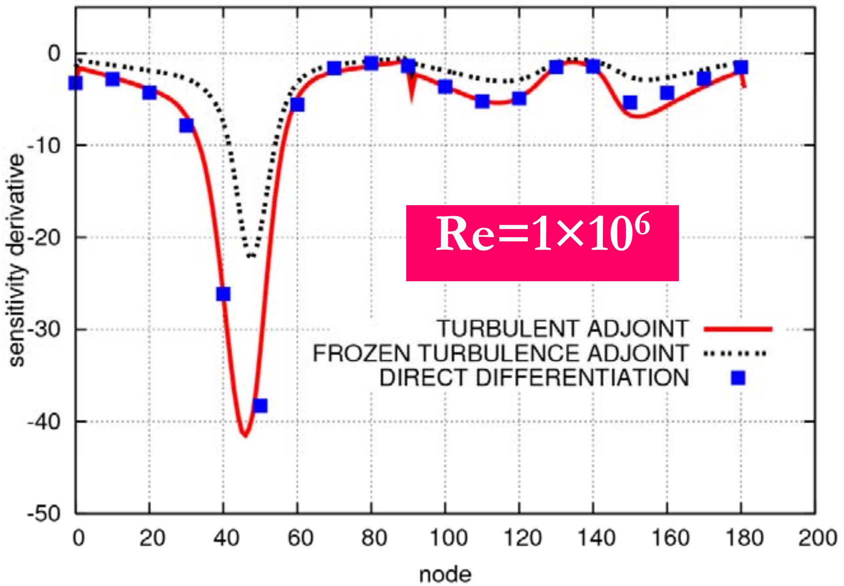
p	pressure	q	Adjoint pressure
v_i	velocities	u_i	Adjoint velocities
$\widetilde{\nu}$	turbulence variable	$\widetilde{\nu_a}$	Adjoint turbulence variable

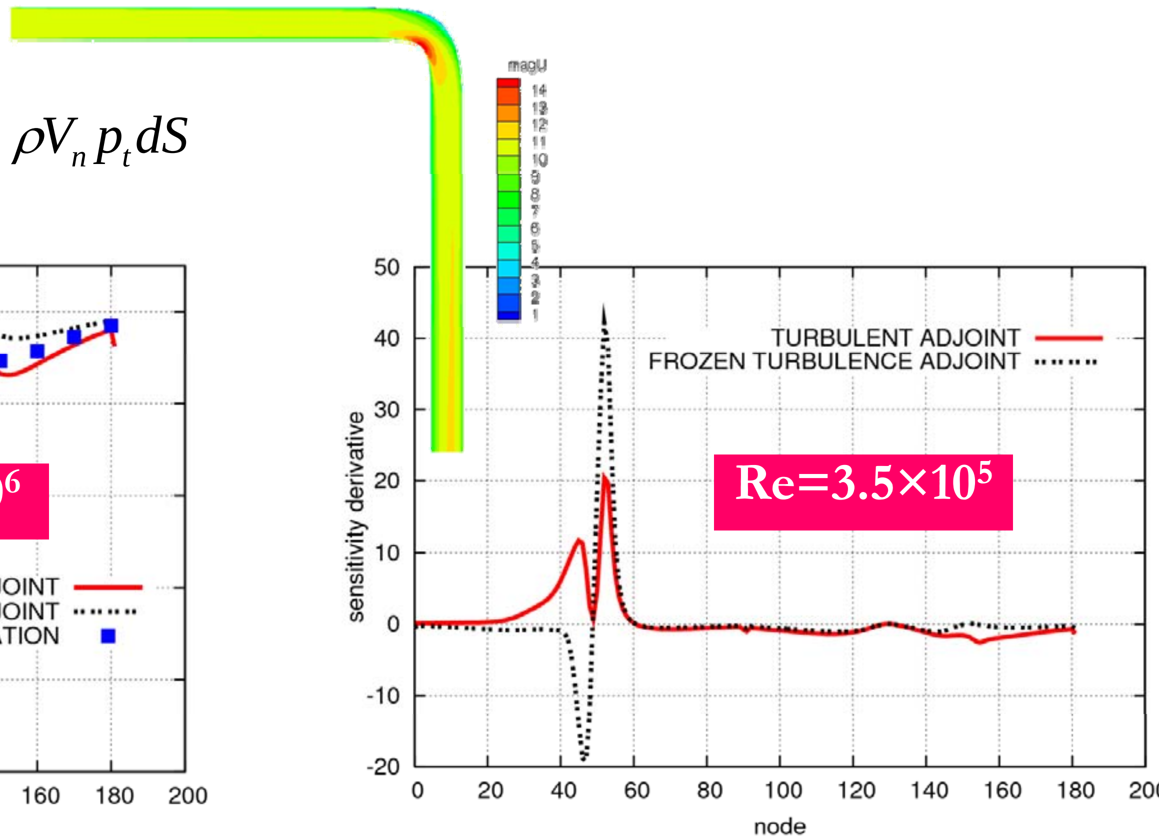
How Important is to Differentiate the Turbulence Model Eqs.?

Depending on the case & the Reynolds number, the "frozen turbulence assumption" may lead to wrongly signed sensitivity derivatives!

The computationally expensive Direct Differentiation (DD) method is used to compute reference sensitivities (to compare with).

$$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$

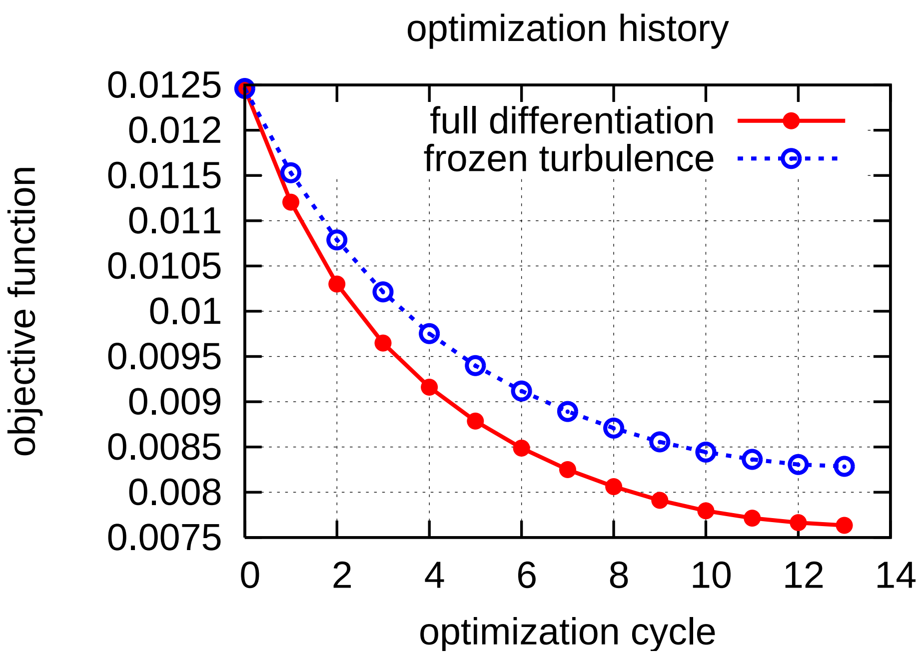




Does this affect the Optimization Turnaround time?

Demonstration using Steepest Descent, with the same step η

$$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$



Extra equations/terms & computational effort

New terms may have a completely different importance

- **An additional adjoint PDE (*the adjoint to the S-A model eq.*)**

$$\frac{\partial \widetilde{\nu_a}}{\partial x_j} v_j + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu_a}}{\partial x_j}\right] = \frac{1}{\sigma}\frac{\partial \widetilde{\nu_a}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_j} + 2\frac{c_{b2}}{\sigma}\frac{\partial}{\partial x_j}\left(\widetilde{\nu_a}\frac{\partial \widetilde{\nu}}{\partial x_j}\right) + \widetilde{\nu_a}\widetilde{\nu}\,\mathcal{C}_{\widetilde{\nu}}(\widetilde{\nu},\vec{v})$$
$$+\frac{\delta \nu_t}{\delta \widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + (-P + D)\,\widetilde{\nu_a} + \frac{\partial F_\Omega}{\partial \widetilde{\nu}}$$

(…plus boundary conditions)

- **New terms in the adjoint momentum eqs. (by far the most important!)**

$$-v_j\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \frac{\partial q}{\partial x_i} \boxed{-\widetilde{\nu}\frac{\partial \widetilde{\nu_a}}{\partial x_i} - \frac{\partial}{\partial x_l}\left(e_{jli}e_{jmq}\frac{\mathcal{C}_S}{S}\frac{\partial v_q}{\partial x_m}\widetilde{\nu}\widetilde{\nu_a}\right)} = -\frac{\partial F_\Omega}{\partial v_i}$$

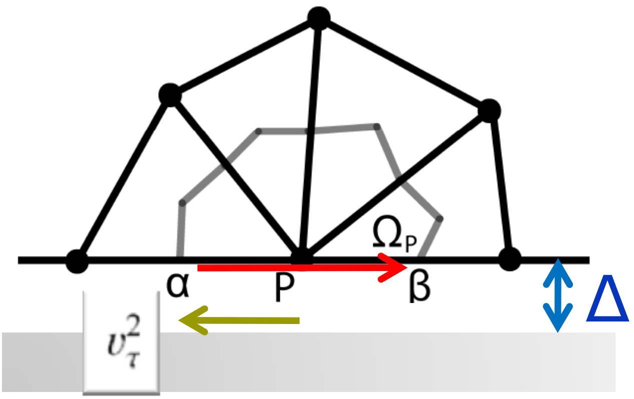
- **New terms in the adjoint boundary conditions.**
- **New terms in the sensitivity derivative expressions.**

A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: ' Continuous Adjoint Approach to the Spalart-Allmaras Turbulence Model for Incompressible Flows', Computers & Fluids, 38, pp. 1528-1538, 2009.

Differentiation of High-Re Turbulence Models

A New Adjoint Law of the Wall

Demonstrated for the k-ε model. Exists for Spalart-Allmaras & k-ω



$$\frac{\delta F_{aug}}{\delta b_m} = \frac{\delta F}{\delta b_m} + \int_\Omega u_i \frac{\delta R_i^v}{\delta b_m} d\Omega + \int_\Omega q \frac{\delta R^p}{\delta b_m} d\Omega + \int_\Omega k_a \frac{\delta R^k}{\delta b_m} d\Omega + \int_\Omega \varepsilon_a \frac{\delta R^\varepsilon}{\delta b_m} d\Omega + \int_\Omega u_i R_i^v \frac{\delta(d\Omega)}{\delta b_m} + \int_\Omega q R^p \frac{\delta(d\Omega)}{\delta b_m} + \int_\Omega k_a R^k \frac{\delta(d\Omega)}{\delta b_m} + \int_\Omega \varepsilon_a R^\varepsilon \frac{\delta(d\Omega)}{\delta b_m}$$

Friction velocity

$$v_\tau^2 = (\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) n_j t_i$$

Adjoint friction velocity

$$u_\tau^2 = (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j t_i$$

$$u_\tau^2 = \frac{1}{c_v}\left[2u_k t_k v_\tau - \left(\nu + \frac{\nu_t}{Pr_k}\right)\frac{\partial k_a}{\partial x_j} n_j \frac{\delta k}{\delta v_\tau} - \left(\nu + \frac{\nu_t}{Pr_\varepsilon}\right)\frac{\partial \varepsilon_a}{\partial x_j} n_j \frac{\delta \varepsilon}{\delta v_\tau}\right]$$

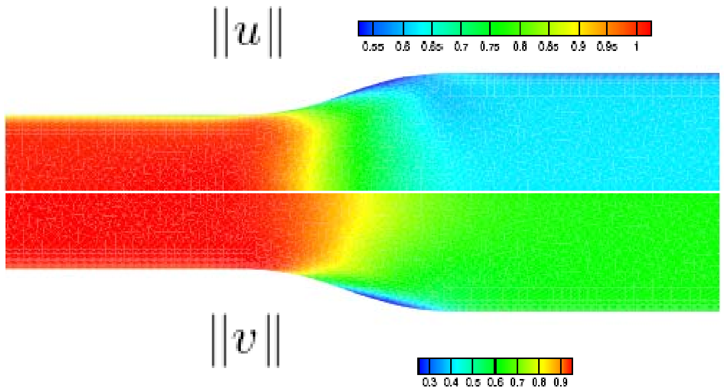
A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: 'Adjoint Wall Functions: A New Concept for Use in Aerodynamic Shape Optimization', J. Comp.Physics, 229, pp. 5228–5245 , 2010.

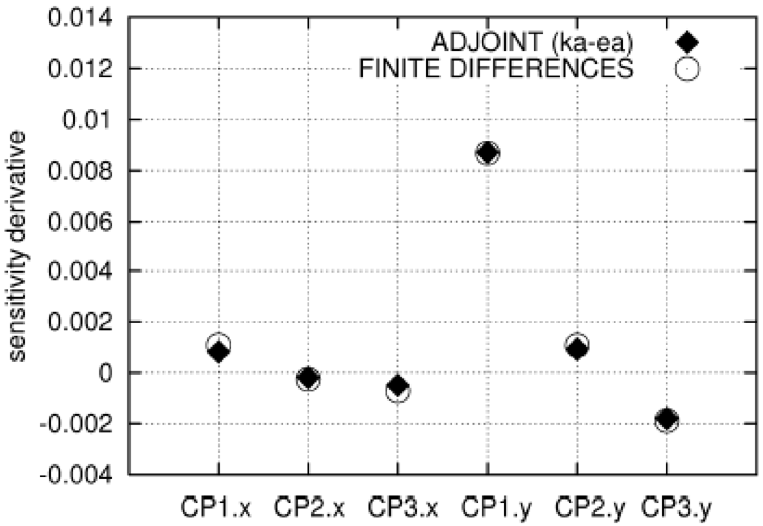
Computation of Sensitivity Derivatives on the starting geometry

Subsonic Flow in an axial diffuser, with incipient separation, Re=1x10^6

Objective function: mass-averaged total pressure losses

Without running the Optimization Loop



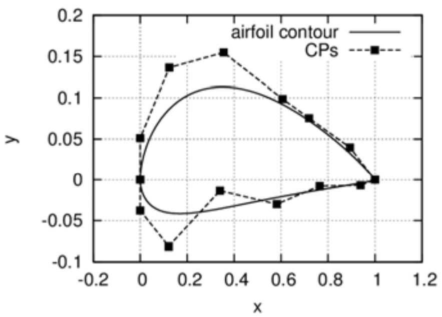


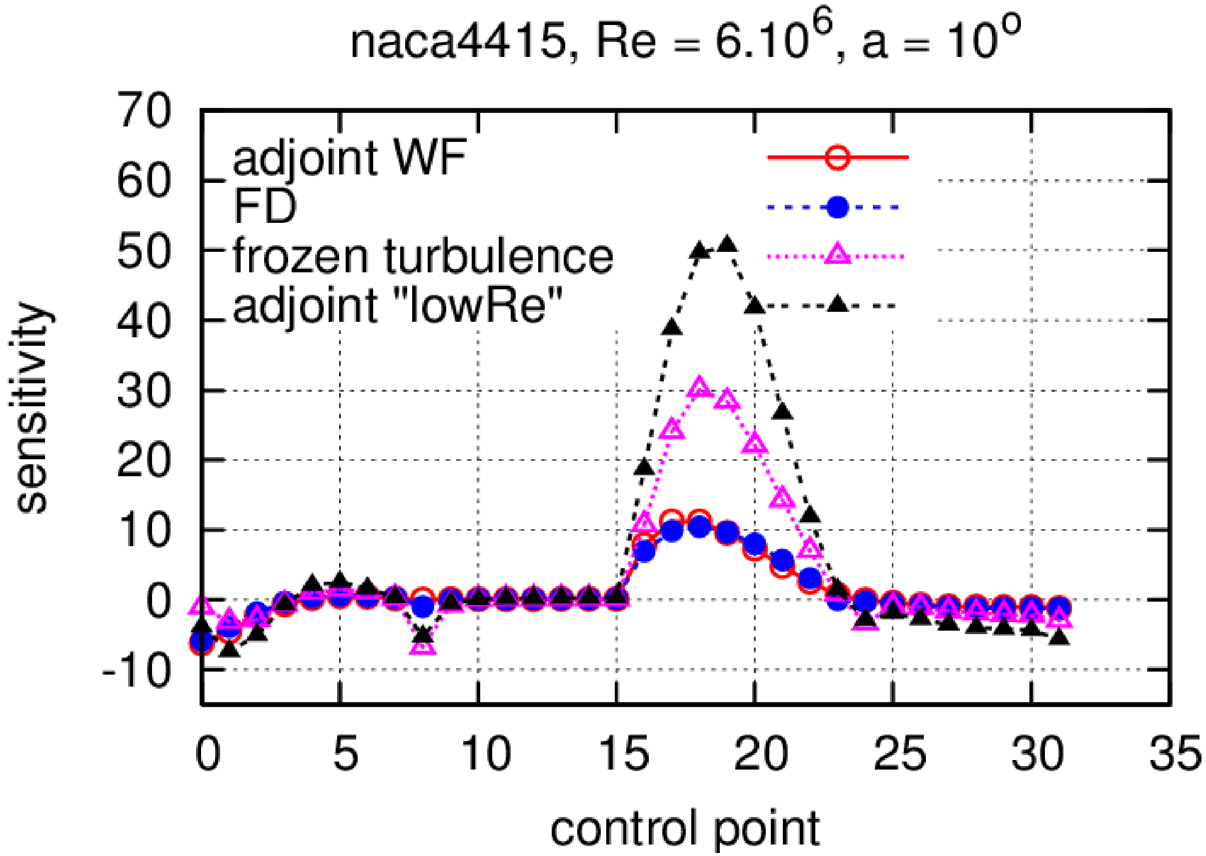
Design of an axial diffuser for min. total pressure losses (Re=1x10^6).
(*Objective: mass-averaged p_t losses*)

Why to do it? First example!

Subsonic Flow around NACA4415

Important Finding: Using the adjoint "low-Re" model yields worst results than the "Frozen Turbulence Assumption"!!!



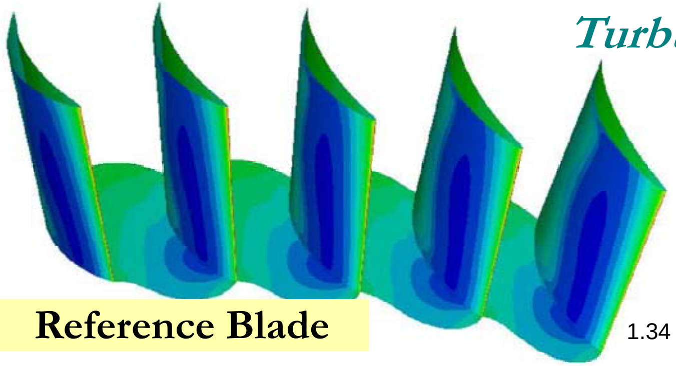


Design-Optimization of two Peripheral Compressor Cascades

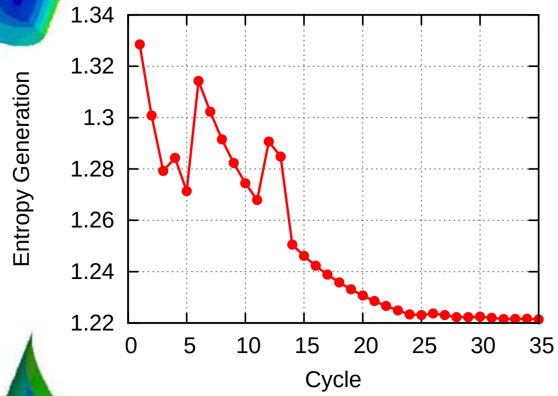
Target: Minimum Viscous Losses

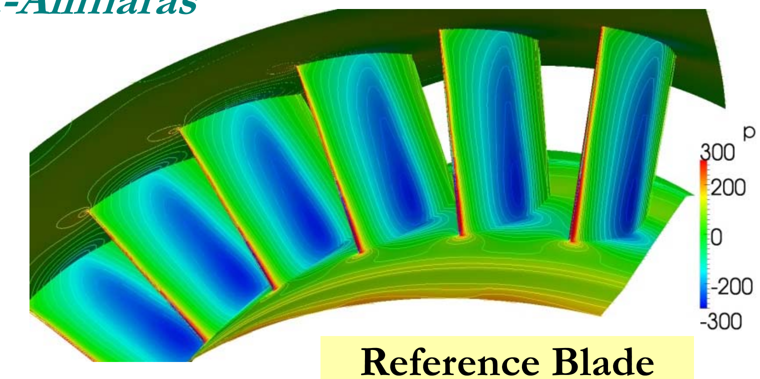
Constraints on the Flow Turning & the Blade Thickness

Turbulence Model: Spalart-Allmaras



Reference Blade

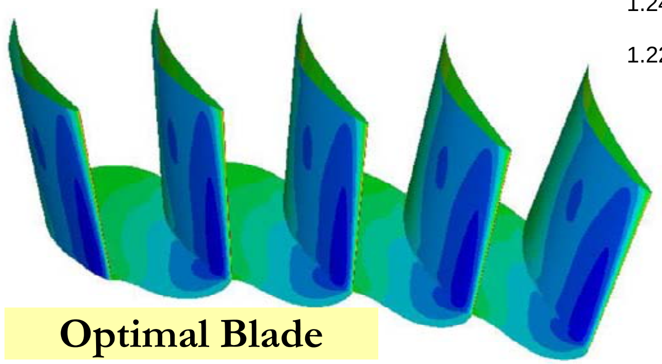




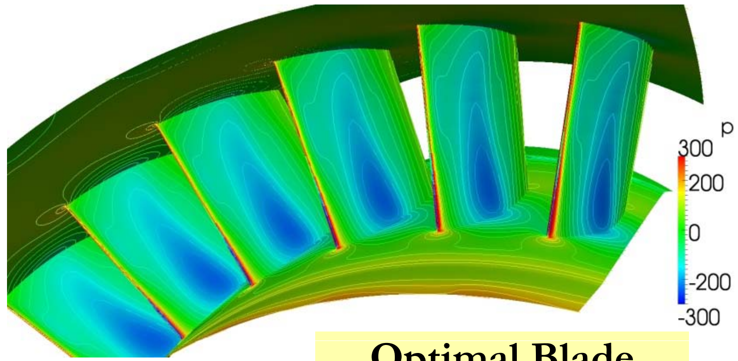
Reference Blade

Row 1

Row 2



Optimal Blade



Optimal Blade

Applied for Turbulence Models involving the Distance from the Wall

Including Wall Functions

Inspired by the AIAA J. paper, March 2012 by Bueno-Orovio, et al.

Differentiate the Hamilton-Jacobi eq., governing the distance Δ

$$\frac{\delta F_{aug}}{\delta b_n} = -\int_{S_{W_p}} \left[(\nu+\nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j - q n_i \right] \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS + \boxed{\int_{\Omega} \tilde{\nu} \tilde{\nu_a} \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega}$$

New State Eq.: $$R^{\Delta} = \frac{\partial (c_j \Delta)}{\partial x_j} - \Delta \frac{\partial^2 \Delta}{\partial x_j^2} - 1 = 0 \quad , \quad c_j = \partial \Delta / \partial x_j$$

$$F_{aug} = F + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega + \int_{\Omega} \tilde{\nu_a} R^{\tilde{\nu}} d\Omega + \boxed{\int_{\Omega} \Delta_a R^{\Delta} d\Omega}$$

New Adjoint Eq. (decoupled): $$R^{\Delta_a} = -2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) + \tilde{\nu} \tilde{\nu_a} \mathcal{C}_{\Delta} = 0$$

New Sensitivity Derivatives:

$$\frac{\delta F_{aug}}{\delta b_n} = -\int_{S_{W_p}} \left[(\nu+\nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j - q n_i \right] \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \underbrace{- \int_{S_{W_p}} 2 \Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS}$$

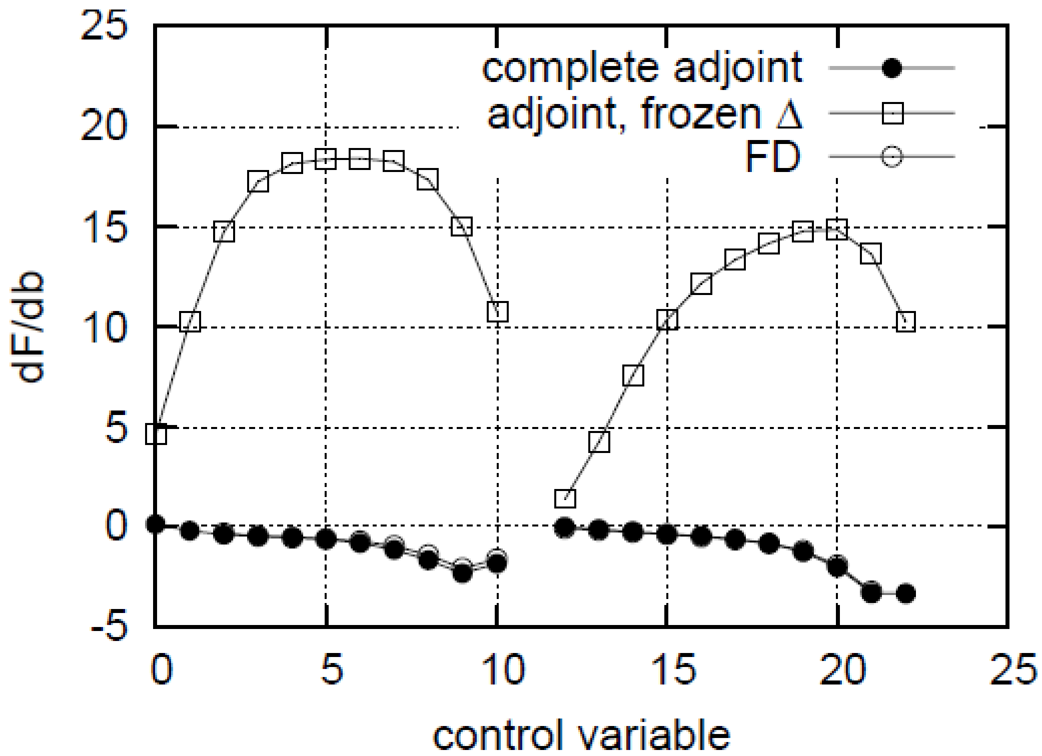
Demo: In some cases, ignoring δ(Δ) might be detrimental

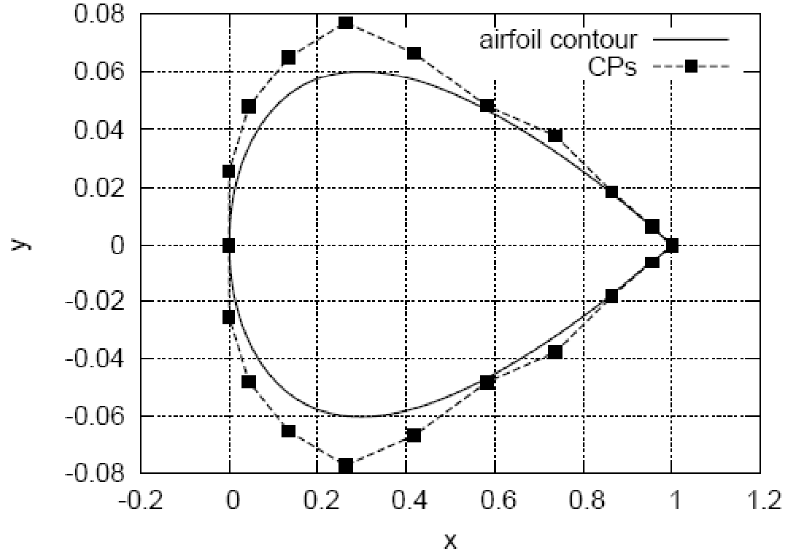
NACA12 Airfoil, Re=6x10^6, a_{inf}=3^o

NACA12 F= -Lift, Sensitivities wrt the y of Bezier control points

Spalart-allmaras, low-Re model, Re=6x10^6, a_{inf}=3^o

Important: In this case, the "frozen distance assumption" yields error in the sign!





The straightforward way to compute the Hessian

Twice application of the Direct Differentiation Method (DD-DD)

Shown in Discrete. Formulated and programmed also in Continuous Mode

Very expensive! Nothing to gain from the use of the Newton's method.

Newton Method:

$$b_i^{n+1} = b_i^n + db_i$$

$$\frac{d^2F}{db_i db_j} db_j = -\frac{dF}{db_i}$$

$$\frac{dF}{db_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{db_i}$$

k=1,…,N design variables

$$\frac{d^2F}{db_i db_j} = \frac{\partial^2 F}{\partial b_i \partial b_j} + \frac{\partial^2 F}{\partial b_i \partial U_k}\frac{dU_k}{db_j} + \frac{\partial^2 F}{\partial U_k \partial b_j}\frac{dU_k}{db_i} + \frac{\partial^2 F}{\partial U_k \partial U_m}\frac{dU_k}{db_i}\frac{dU_m}{db_j} + \frac{\partial F}{\partial U_k}\frac{d^2U_k}{db_i db_j}$$

$$\frac{dR_m}{db_i} = \frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{dU_k}{db_i} = 0$$

$$\frac{d^2R_n}{db_i db_j} = \frac{\partial^2 R_n}{\partial b_i \partial b_j} + \frac{\partial^2 R_n}{\partial b_i \partial U_k}\frac{dU_k}{db_j} + \frac{\partial^2 R_n}{\partial U_k \partial b_j}\frac{dU_k}{db_i} + \frac{\partial^2 R_n}{\partial U_k \partial U_m}\frac{dU_k}{db_i}\frac{dU_m}{db_j} + \frac{\partial R_n}{\partial U_k}\frac{d^2U_k}{db_i db_j} = 0$$

► **Cost of the DD-DD approach scales with N^2.**

Computation of the Hessian Matrix, via DD-AV

How to compute the Hessian with the lowest CPU cost

DD-AV, equivalent to "tangent mode, then reverse mode"

Shown in Discrete. Formulated and programmed also in Continuous Mode

The gain from using the Newton's method (if any) depends on N

$$\frac{dR_m}{db_i} = \frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{dU_k}{db_i} = 0 \quad \Rightarrow \quad \boxed{\frac{dU_k}{db_i}} \quad \boxed{N}$$

System solutions (EFS)

$$\frac{\partial F}{\partial U_k} + \hat{\Psi}_n \frac{\partial R_n}{\partial U_k} = 0 \quad \Rightarrow \quad \boxed{\hat{\Psi}_n} \quad \boxed{1}$$

EFS

The Adjoint equation is the same with that solved to compute the Gradient !!!

$$\frac{d^2\hat{F}}{db_i db_j} = \frac{\partial^2 F}{\partial b_i \partial b_j} + \hat{\Psi}_n \frac{\partial^2 R_n}{\partial b_i \partial b_j} + \frac{\partial^2 F}{\partial U_k \partial U_m}\frac{dU_k}{db_i}\frac{dU_m}{db_j} + \hat{\Psi}_n \frac{\partial^2 R_n}{\partial U_k \partial U_m}\frac{dU_k}{db_i}\frac{dU_m}{db_j}$$
$$+ \frac{\partial^2 F}{\partial b_i \partial U_k}\frac{dU_k}{db_j} + \hat{\Psi}_n \frac{\partial^2 R_n}{\partial b_i \partial U_k}\frac{dU_k}{db_j} + \frac{\partial^2 F}{\partial U_k \partial b_j}\frac{dU_k}{db_i} + \hat{\Psi}_n \frac{\partial^2 R_n}{\partial U_k \partial b_j}\frac{dU_k}{db_i}$$
$$+ \cancel{\left(\frac{\partial F}{\partial U_k} + \hat{\Psi}_n \frac{\partial R_n}{\partial U_k}\right)} \frac{d^2 U_k}{db_i db_j}$$

$$\frac{d^2 F^\lambda}{db_i db_j} db_j^\lambda = -\frac{dF^\lambda}{db_i}$$

► **The cost per Newton cycle is N+1+1=N+2 EFS! Scales with N, not N^2.**

Computation of the Hessian Matrix, via DD-AV

With Continuous Adjoint

See references (on both discrete & continuous approaches)

$$\frac{\delta F_{aug}}{\delta b_j} = \frac{\delta F}{\delta b_j} + \int_\Omega \Psi_n \frac{\partial}{\partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) d\Omega + \int_S \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta x_l}{\delta b_j} n_l dS + \int_\Omega \frac{\partial \Psi_n}{\partial b_j}\frac{\partial f_{nk}^{inv}}{\partial x_k} d\Omega$$

$$\begin{aligned}
\frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} &= \frac{\delta^2 F}{\delta b_i \delta b_j} + \int_\Omega \Psi_n \frac{\partial^2}{\partial b_i \partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) d\Omega \\
&+ \int_\Omega \frac{\partial^2 \Psi_n}{\partial b_i \partial b_j}\frac{\partial f_{nk}^{inv}}{\partial x_k} d\Omega + \int_\Omega \frac{\partial \Psi_n}{\partial b_i}\frac{\partial^2 f_{nk}^{inv}}{\partial x_k \partial b_j} d\Omega + \int_\Omega \frac{\partial \Psi_n}{\partial b_j}\frac{\partial^2 f_{nk}^{inv}}{\partial x_k \partial b_i} d\Omega \\
&+ \int_S \frac{\partial \Psi_n}{\partial b_i}\frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta x_l}{\delta b_j} n_l dS + \int_S \frac{\partial \Psi_n}{\partial b_j}\frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta x_l}{\delta b_i} n_l dS + \\
&+ \int_S \Psi_n \frac{\partial}{\partial b_i}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right)\frac{\delta x_l}{\delta b_j} n_l dS + \int_S \Psi_n \frac{\partial}{\partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right)\frac{\delta x_l}{\delta b_i} n_l dS \\
&+ \int_S \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta^2 x_l}{\delta b_i \delta b_j} n_l dS + \int_S \frac{\partial \Psi_n}{\partial x_m}\frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta x_m}{\delta b_i}\frac{\delta x_l}{\delta b_j} n_l dS \\
&+ \int_S \Psi_n \frac{\partial^2 f_{nk}^{inv}}{\partial x_k \partial x_l}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j} n_m dS + \int_S \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_k}\frac{\delta x_l}{\delta b_j}\frac{\delta (n_l dS)}{\delta b_i}
\end{aligned}$$

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Direct, Adjoint and Mixed Approaches for the Computation of Hessian in Airfoil Design Problems', Int. Num. Meth. in Fluids, 56, 1929-1943, 2008.

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Computation of the Hessian Matrix in Aerodynamic Inverse Design using Continuous Adjoint Formulations', Computers & Fluids, 37, 1029-1039, 2008.

K.C. GIANNAKOGLOU, D.I. PAPADIMITRIOU: 'Adjoint Methods for gradient- and Hessian-based Aerodynamic Shape Optimization', EUROGEN 2007, Jyvaskyla, Finland, June 11-13, 2007.

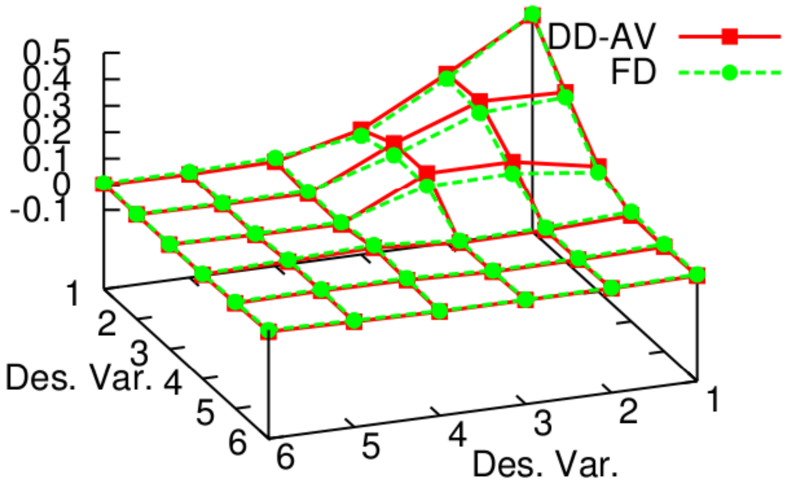
D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Aerodynamic Shape Optimization using Adjoint and Direct Approaches', Arch. Comp.Meth. Engi.(State of the Art Reviews), Vol. 15(4), pp. 447-488, 2008 .

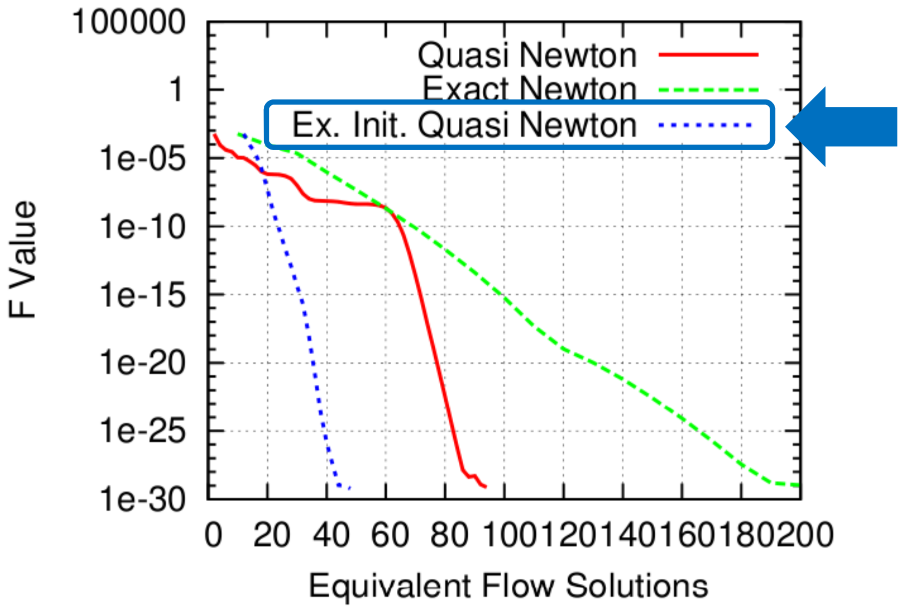
D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'The Continuous Direct-Adjoint Approach for Second Order Sensitivities in Viscous Aerodynamic Inverse Design Problems', Computers & Fluids, 38, 1539-1548, 2009.

The Exactly-Initialized-then-Quasi-Newton method

Application: Inverse design of a Compressor blading (6 design variables)

Compute the Hessian only in the first cycle, then switch to quasi-Newton method (BFGS)





$$\frac{5\times 6}{2}\times 4 = 60 \qquad \text{vs.} \qquad 6+1=7\ \text{EFS}$$

(FD) **(DD-AV)**

The only way to efficiently handle problems with N>>

Compute Hessian-vector products instead of the Hessian itself

__Inspired by:__
The Conjugate Gradient (CG) method for solving systems of linear equations

$$\boxed{Ax = q}$$

requires only matrix-vector products.

$$b_i^{n+1} = b_i^n + db_i$$

$$\frac{d^2F}{db_i db_j} db_j = -\frac{dF}{db_i}$$

$k \leftarrow 0$

$x \leftarrow \text{init}()$

$r^0 \leftarrow \boxed{Ax} - q;\ p \leftarrow -r^0$

while $r^k \neq 0$ **and** $k \leq M_{CG}$ **do**

$\quad \eta \leftarrow \frac{(r^k)^T r^k}{p^T \boxed{Ap}}$

$\quad x \leftarrow x + \eta p$

$\quad r^{k+1} \leftarrow r^k + \eta \boxed{Ap}$

$\quad \beta \leftarrow \frac{(r^{k+1})^T r^{k+1}}{(r^k)^T r^k}$

$\quad p \leftarrow -r_{k+1} + \beta p$

$\quad k \leftarrow k + 1$

end while

The AV-DD Truncated Newton Method (with CG)

$k \leftarrow 0$
$b_j \leftarrow \text{init}()$
while $k \le k_{max}$ **do**
  $U_n \leftarrow$ Flow Equations [1 EFS]
  $\Psi_n \leftarrow$ Adjoint Equations [1 EFS]
  $r_j^0 = \frac{dF}{db_j} \leftarrow$ Gradient Expression
  $db_j^0 \leftarrow \text{init}(0)$
  $p_j \leftarrow -r_j^0$
  $m \leftarrow 0$
  while $r^m \neq 0$ **and** $m \le M_{CG}$ **do**
    $\frac{dU_n}{db_j} p_j \leftarrow$ DD (Flow Equations) [1 EFS]
    $\frac{d\Psi_n}{db_j} p_j \leftarrow$ DD (Adjoint Equations) [1 EFS]
    $w_i = \frac{d^2F}{db_i db_j} p_j \leftarrow$ Hessian Expression
    $\eta \leftarrow \frac{r_i^m r_i^m}{p_j w_j}$
    $db_j^{m+1} \leftarrow db_j^m + \eta p_j$
    $r_j^{m+1} \leftarrow r_j^m + \eta w_j$
    $\beta \leftarrow \frac{r_i^{m+1} r_i^{m+1}}{r_j^m r_j^m}$
    $p_j \leftarrow -r_j^{m+1} + \beta p_j$
    $m \leftarrow m + 1$
  end while
  $b_j \leftarrow b_j + db_j$
  $k \leftarrow k + 1$
end while

$$b_i^{n+1} = b_i^n + db_i$$

$$\frac{d^2F}{db_i db_j} db_j = -\frac{dF}{db_i}$$

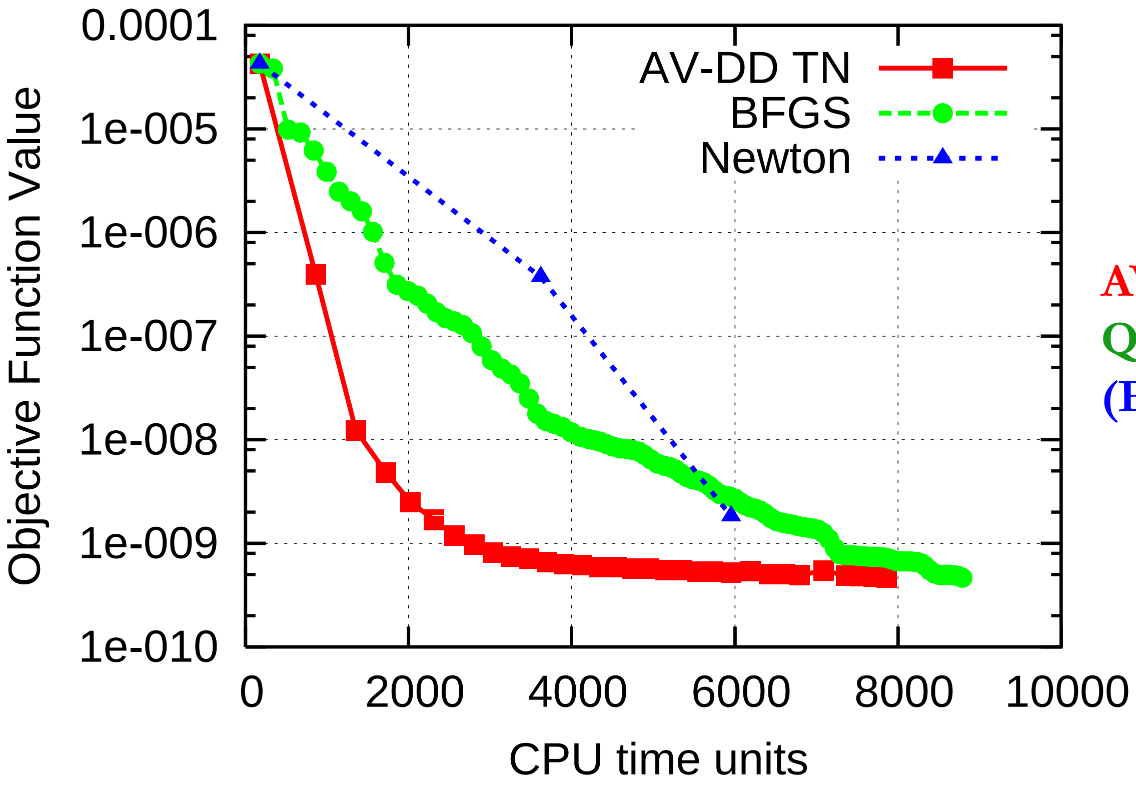
Total Cost= $2+2M_{CG} << N$

AV-DD Truncated Newton method – Why?

Application: Inverse design of an isolated airfoil, N=42 DOFs

Compute Hessian-vector products instead of the Hessian itself

Comparison of three solution methods



AV-DD Truncated Newton method

Quasi-Newton BFGS

(Exact) Newton

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Aerodynamic design using the truncated Newton algorithm and the continuous adjoint approach', Int. J.for Numerical Methods in Fluids, 68, 6, pp. 724-739, 2012.

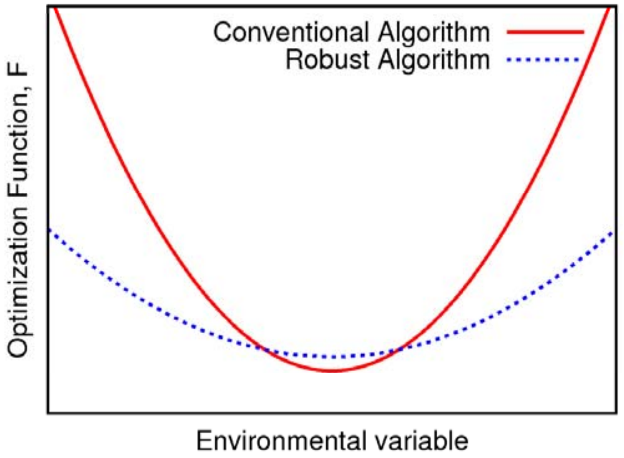
The Second-Order, Second-Moment (SOSM) Approach

For N design (b_i) & M environmental (c_i) variables

Minimize the estimated mean & standard deviation of F

Third-order mixed derivatives must be computed

Proposed method: DD_c-DD_c-Av_b (if M<N)



$$\widehat{F} = \hat{\mu_F} + k\hat{\sigma_F}$$

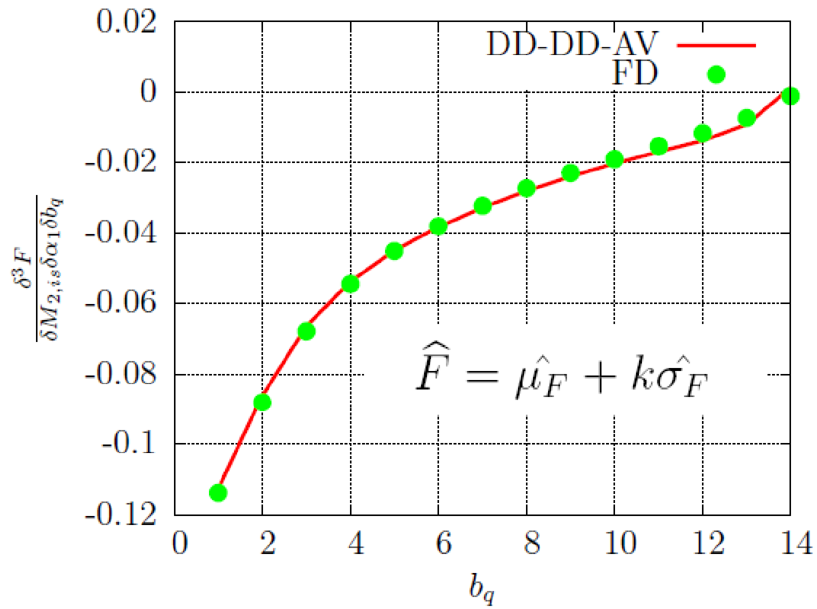
$$\hat{\mu_F} = F_{\mathrm{D}} + \frac{1}{2}\left[\frac{d^2F}{dc_i^2}\right]_{\mathrm{D}}\sigma_i^2$$

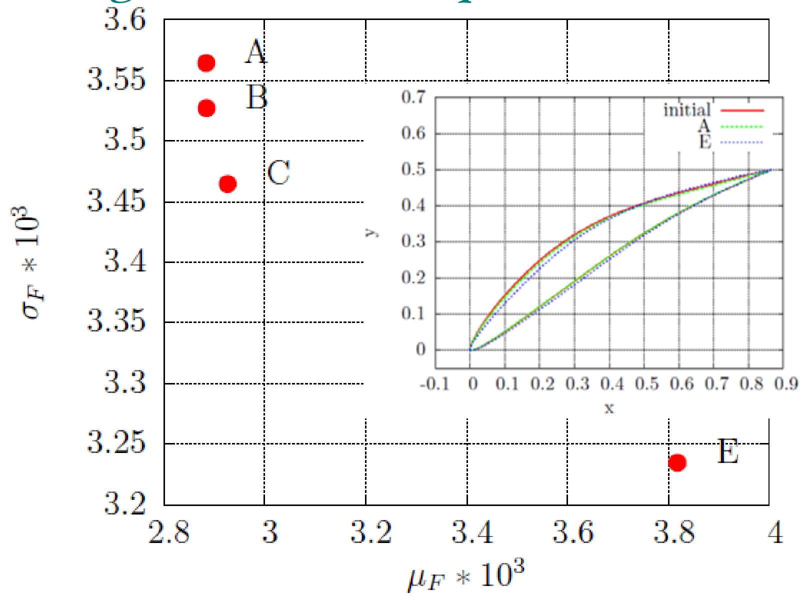
$$\hat{\sigma_F} = \sqrt{\left[\frac{dF}{dc_i}\right]_{\mathrm{D}}^2\sigma_i^2 + \frac{1}{2}\left[\frac{d^2F}{dc_i dc_j}\right]_{\mathrm{D}}^2\sigma_i^2\sigma_j^2}$$

$$\frac{d\widehat{F}}{db_l} = \frac{dF}{db_l} + \frac{1}{2}\frac{d^3F}{dc_i^2 db_l}\sigma_i^2 + k\frac{2\frac{dF}{dc_i}\frac{d^2F}{dc_i db_l}\sigma_i^2 + \frac{d^2F}{dc_i dc_j}\frac{d^3F}{dc_i dc_j db_l}\sigma_i^2\sigma_j^2}{2\sqrt{\left[\frac{dF}{dc_i}\right]^2\sigma_i^2 + \frac{1}{2}\left[\frac{d^2F}{dc_i dc_j}\right]^2\sigma_i^2\sigma_j^2}}$$

Robust Design of a Compressor Cascade

Two environmental variables (M=2): Inlet flow angle & exit isentropic Mach number



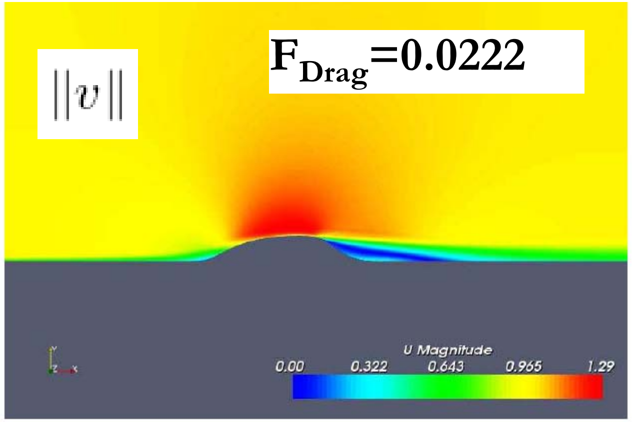


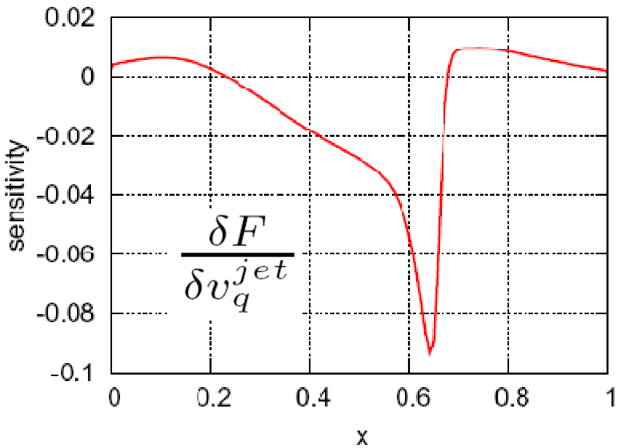
E.M. PAPOUTSIS-KIACHAGIAS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Robust Design in Aerodynamics using 3rd-Order Sensitivity Analysis based on Discrete Adjoint. Application to Quasi-1D Flows', International Journal for Numerical Methods in Fluids, Vol. 69, No. 3, pp. 691-709, 2012.
E.M. PAPOUTSIS-KIACHAGIAS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: Discrete and Continuous Adjoint Methods in Aerodynamic Robust Design problems, CFD and Optimization 2011, ECCOMAS Thematic Conference, Antalya, Turkey, May 23-25, 2011.
D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Third-Order Sensitivity Analysis for Robust Aerodynamic Design using Continuous Adjoint', International Journal for Numerical Methods in Fluids, Vol. 71, No. 5, pp. 652-670, 2013.

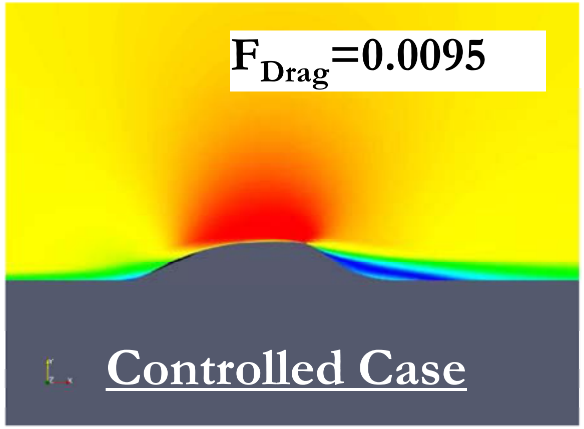
Flow Control Optimization

Optimal flow control using suction/blowing/pulsating jets

Idea: Compute the sensitivity derivatives by solving the flow & adjoint problem once, for normal_jet_velocity=0. Use the computed sensitivity maps to optimally locate the jets and their sign to decide whether suction or blowing is needed.
Stop here or iterate to optimize all jet parameters.





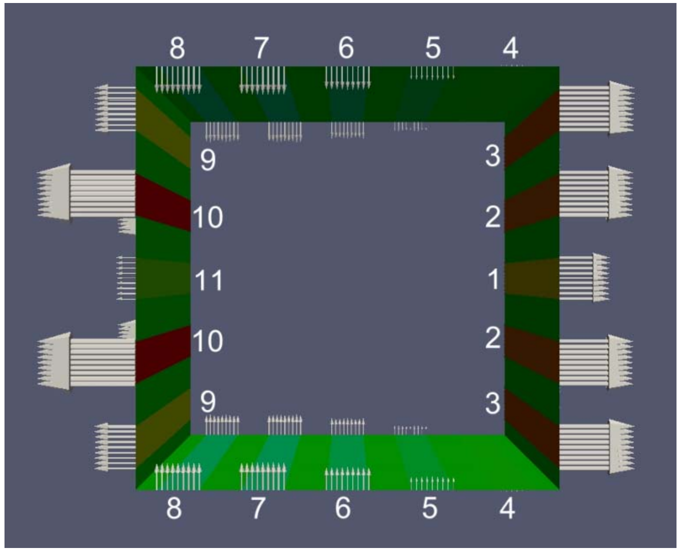


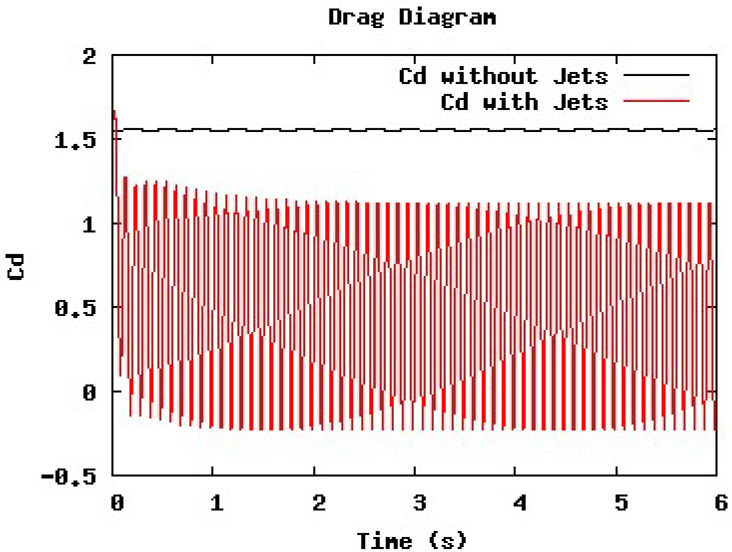
A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: 'Optimal Location of Suction or Blowing Jets Using the Continuous Adjoint Approach', ECCOMAS CFD 2010, Lisbon, June 14-17, 2010.

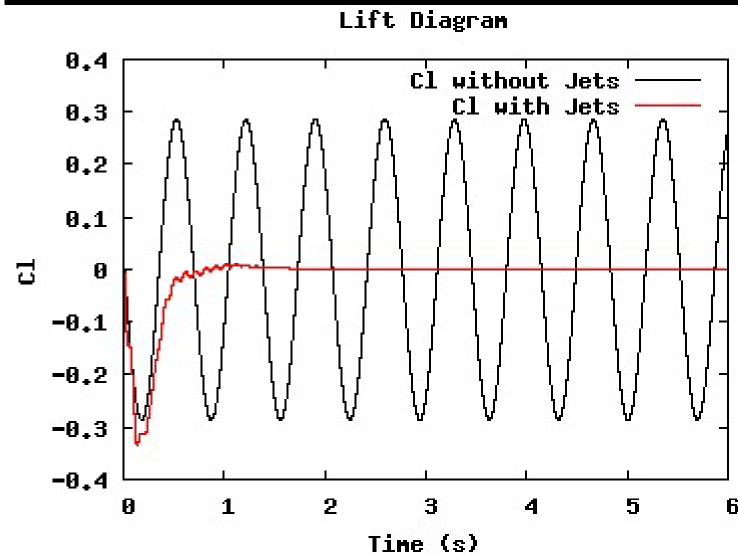
A.S. ZYMARIS, D.I. PAPADIMITRIOU, E.M. PAPOUTSIS-KIACHAGIAS, K.C. GIANNAKOGLOU, C. OTHMER: 'The Continuous Adjoint Method as a Guide for the Design of Flow Control Systems Based on Jets", Engineering Computations, to appear 2013.

Flow around a square (Re=100) – Control with Pulsating Jets

Slot	Amplitude
1	0.0484
2	0.0707
3	0.0721
4	0.0186
5	-0.0124
6	-0.0218
7	-0.0264
8	-0.0260
9	0.0400
10	0.0948
11	0.0193

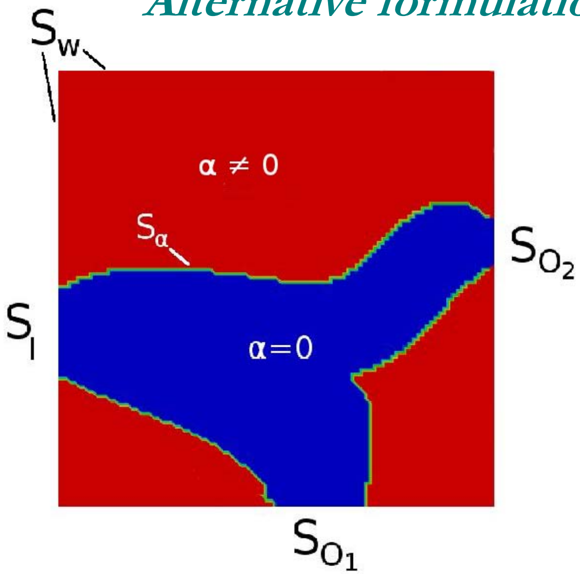






Formulations based on porosity (a)

Alternative formulations based on the level-set method excluded for this talk



Flow Model:
Incompressible fluid
Turbulent flow
With heat transfer

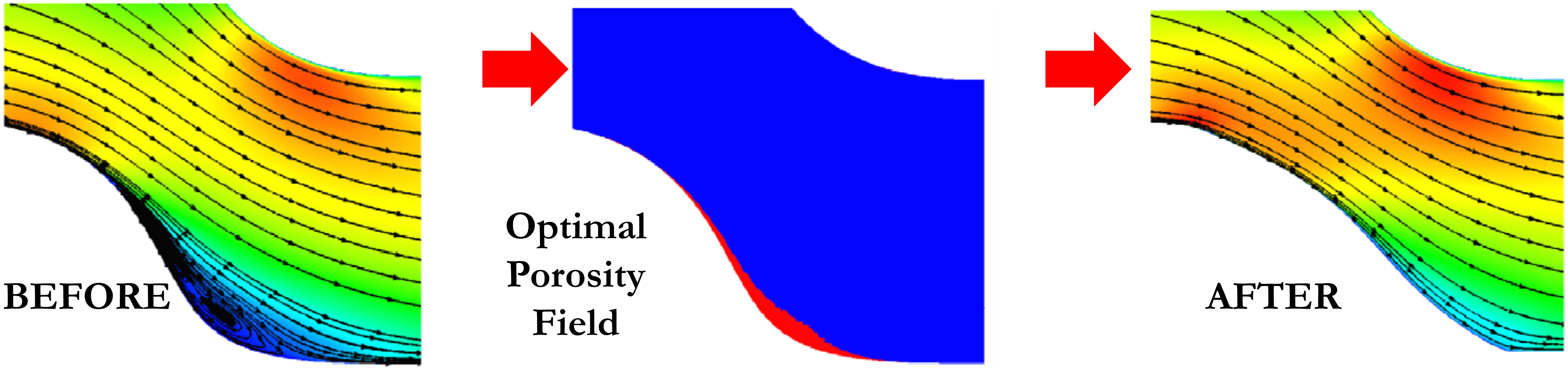
$$R_p=0, \quad R_{v_i}=0, \quad R_T=0, \quad R_{\tilde{\nu}}=0$$

$$\begin{aligned}
R_p &= \frac{\partial v_j}{\partial x_j}\\
R_{v_i} &= v_j\frac{\partial v_i}{\partial x_j}+\frac{\partial p}{\partial x_i}-\frac{\partial}{\partial x_j}\left[(\nu+\nu_t)\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\right]\underline{+\alpha v_i}\\
R_T &= v_j\frac{\partial T}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\frac{\nu}{Pr}+\frac{\nu_t}{Pr_t}\right)\frac{\partial T}{\partial x_j}\right]\underline{+\alpha\left(T-T_{wall}\right)}\\
R_{\tilde{\nu}} &= v_j\frac{\partial \tilde{\nu}}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\tilde{\nu}}{\sigma}\right)\frac{\partial \tilde{\nu}}{\partial x_j}\right]-\frac{c_{b_2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial x_j}\right)^2-\tilde{\nu}P\left(\tilde{\nu}\right)+\tilde{\nu}D\left(\tilde{\nu}\right)\underline{+\alpha\tilde{\nu}}
\end{aligned}$$

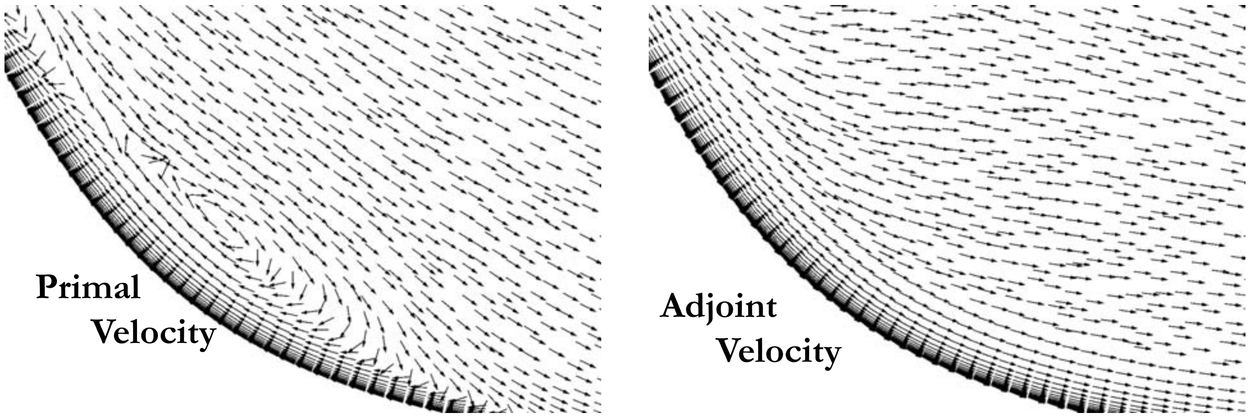
Adjoint equations

$$
\begin{aligned}
R_q &= \frac{\partial u_j}{\partial x_j} \\
R_{u_i} &= -v_j\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\frac{\partial q}{\partial x_i}-\frac{\partial}{\partial x_j}\left[(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\right] \\
&- \widetilde{\nu}\frac{\partial \widetilde{\nu_a}}{\partial x_i}-\frac{\partial}{\partial x_k}\left(e_{jki}e_{jmq}\frac{\mathcal{C}_S}{S}\frac{\partial v_q}{\partial x_m}\widetilde{\nu}\widetilde{\nu_a}\right)-T\frac{\partial T_a}{\partial x_i}+\alpha u_i \\
R_{T_a} &= -v_j\frac{\partial T_a}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\frac{\nu}{Pr}+\frac{\nu_t}{Pr_t}\right)\frac{\partial T_a}{\partial x_j}\right]+\alpha T_a \\
R_{\widetilde{\nu_a}} &= -v_j\frac{\partial \widetilde{\nu_a}}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu_a}}{\partial x_j}\right]+\frac{1}{\sigma}\frac{\partial \widetilde{\nu_a}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_j}+2\frac{c_{b2}}{\sigma}\frac{\partial}{\partial x_j}\left(\widetilde{\nu_a}\frac{\partial \widetilde{\nu}}{\partial x_j}\right) \\
&+ \widetilde{\nu_a}\widetilde{\nu}\,\mathcal{C}_{\widetilde{\nu}}(\widetilde{\nu},\vec{v})+(-P+D)\,\widetilde{\nu_a}+\frac{\delta \nu_t}{\delta \widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right) \\
&+ \frac{\delta \nu_t}{\delta \widetilde{\nu}}\frac{1}{Pr_t}\frac{\partial T_a}{\partial x_j}\frac{\partial T}{\partial x_j}+\alpha\widetilde{\nu_a}
\end{aligned}
$$

E.A. KONTOLEONTOS, E.M. PAPOUTSIS-KIACHAGIAS, A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Adjoint-based constrained topology optimization for viscous flows, including heat transfer, Engineering Optimization, 2012.

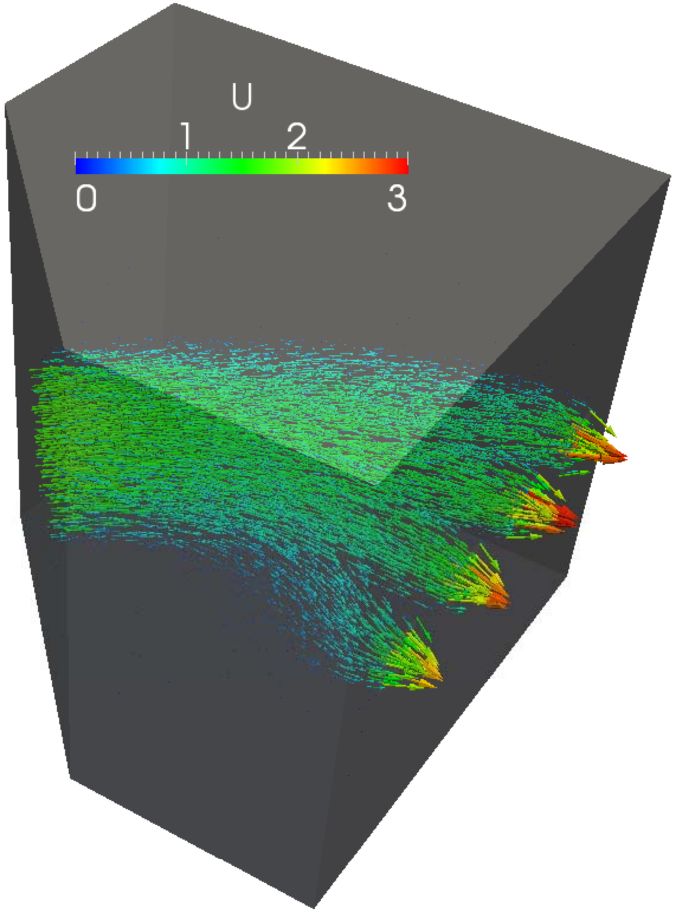


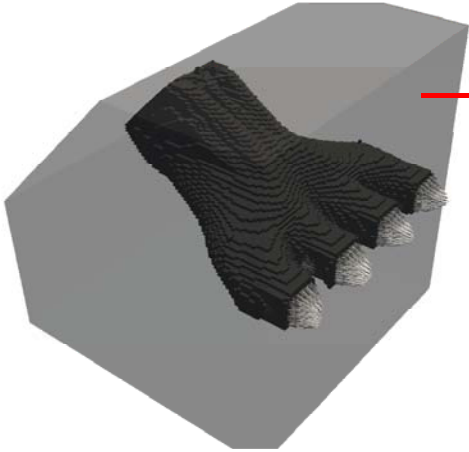
Objective: Min. pt Losses – Continuous Adjoint to [RANS & Spalart-Allmaras].
Recirculation areas disappeared - 15% total pressure losses reduction.



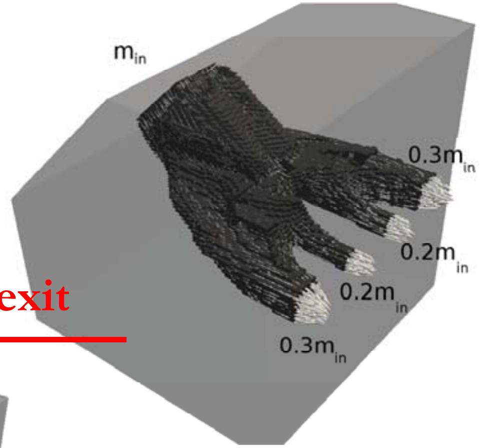
$$\frac{\delta F_{aug}}{\delta \alpha} = \boxed{v_i u_i} + \tilde{\nu}\tilde{\nu}_a + \int_{\Omega} \tilde{\nu}_a \tilde{\nu} \mathcal{C}_d(\tilde{\nu}, \vec{v}) \frac{\partial d}{\partial \alpha} d\Omega$$

Topology optimization of a manifold at laminar flow conditions.

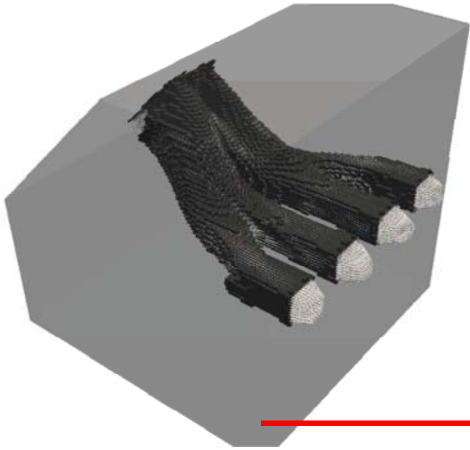




Unconstrained



With constraint on the mass flowrate per exit



With constraint on the Flow swirl at the exit

Closure

- ► Working with continuous adjoint is nice because you gain insight into adjoint PDEs & their BCs or clearly understand/control the assumptions made.
- ► Stop working with the "frozen-turbulence assumption".
- ► The adjoint law of the wall is a useful tool for industrial applications.
- ► High-order derivatives can be computed using continuous or discrete adjoint. Interesting alternatives: (one-shot) exactly-initialized quasi-Newton algorithm, truncated Newton. Useful in adjoint-based robust design.
- ► Continuous adjoint is neither better nor worse than discrete. Any problem whichcan be solved with discrete, can also be solved with continuous adjoint <u>and vice-versa</u>.

<u>On-going research:</u>

- ► Think-discrete-do-continuous...
- ► Robustness of adjoint solvers…
- ► Efficient adjoint methods for Pareto optimization…