NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Parallel CFD & Optimization Unit

Laboratory of Thermal Turbomachines

FLOWHEAD

Fluid Optimisation Workflows for Highly Effective Automotive Development Processes

Development and Application of Continuous Adjoint Methods

Kyriakos C. GIANNAKOGLOU, Professor NTUA

kgianna@central.ntua.gr

http://velos0.ltt.mech.ntua.gr/research/

D.I. Papadimitriou, E.M. Papoutsis-Kiachagias

Development of Adjoint Methods in PCOpt/NTUA

- Development of both continuous & discrete adjoint variants.
- For shape, flow-control, robust-design and topology optimization problems
- Calculation of up to 3rd-order sensitivities.
- Development of continuous adjoint methods to widely-used turbulence models.
- Development on the in-house PUMA codes, OpenFOAM and FINEOpen.
- The in-house adjoint codes are all GPU-enabled (CUDA)

Why Continuous Adjoint?

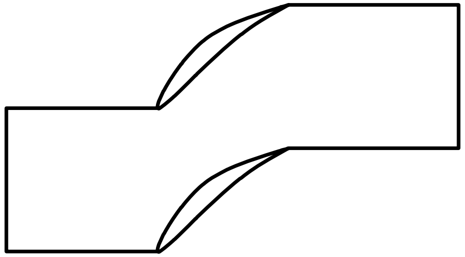
- Continuous adjoint is neither better nor worse than discrete adjoint.
- Both have advantages & (manageable) disadvantages.
- An interesting feature of continuous adjoint is that the programmer "*sees*" the adjoint PDEs and their boundary conditions (in closed form relations). If you have decided to work with discrete adjoint, it is recommended to start with continuous adjoint!
- Possibility of performing well-justified simplifications, if necessary.
- Any problem that can be solved with discrete, it can also be solved with continuous adjoint and vice-versa. In some cases, it is easier to initially work out your idea with discrete adjoint, before switching to continuous adjoint.

Getting Started: Objective Functions

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

- 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 !!!

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

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

Development of the Continuous Adjoint Method

- **State Equations**

(plus the turbulence model eq.)
The turbulence model eq. will not be differentiated ("*frozen turbulence assumption*")

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

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

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

For any objective function F:

$$\frac{\delta F_{aug}}{\delta b_m} = \frac{\delta F}{\delta b_m} + \int_\Omega u_i \frac{\partial R_i^v}{\partial b_m} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial b_m} d\Omega + \int_S \left(u_i R_i^v + q R^p\right) \frac{\delta x_k}{\delta b_m} n_k dS$$

where b_m are the design variables.
Next step: Make it independent of the variations in the state variables.

- **Adjoint Equations**

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

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

- **Adjoint Boundary Conditions**

Inlet: $u_{\langle n \rangle} = -\frac{\partial F_{S_I}}{\partial p}$ **&** $\vec{u}_{\langle t \rangle} = 0$

Outlet:

$$\begin{cases} q = u_j v_j + u_{\langle n \rangle} v_{\langle n \rangle} + (\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j n_i + \frac{\partial F_{S_O}}{\partial v_{\langle n \rangle}} \\ 0 = u_{\langle t \rangle} v_{\langle n \rangle} + (\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j t_i + \frac{\partial F_{S_O}}{\partial v_{\langle t \rangle}} \end{cases}$$

Walls: $u_{\langle n \rangle} = -\frac{\partial F_{S_W}}{\partial p}$ **&** $\vec{u}_{\langle t \rangle} = 0$

•**Sensitivity Derivatives**

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

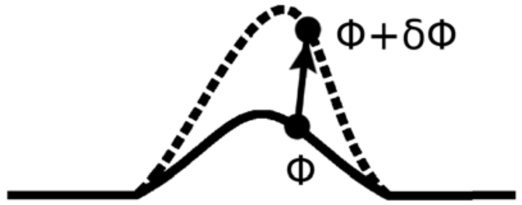
An appropriate mathematical formulation, based on the application of the Green-Gauss divergence theorem may lead to sensitivity derivatives exclusively in terms of boundary integrals (even if the objective function was a field integral !!!). Advantages!

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.

Direct Differentiation (DD) Approach

If b_i denotes the N design variables and Φ any flow quantity:



$$\frac{\delta \Phi}{\delta b_i} = \frac{\partial \Phi}{\partial b_i} + \frac{\partial \Phi}{\partial x_l}\frac{\delta x_l}{\delta b_i}$$

Total ***Partial Variation***

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

… plus the same for the state boundary conditions (Continuous DD)

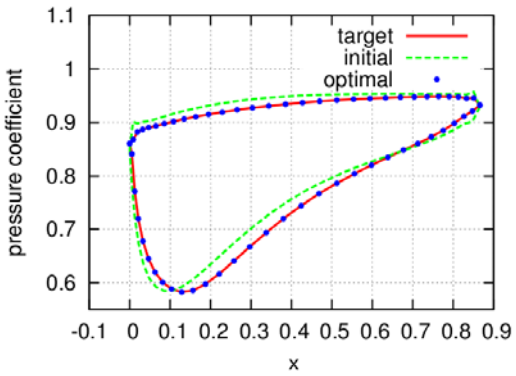
"Equivalent" to the Discrete DD:

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

►The DD method is an easily programmable (expensive, though) tool to compare the gradient computed by the Adjoint Method.

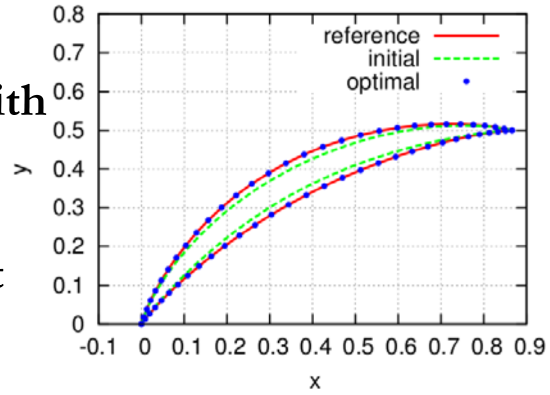
►Also, DD is an indispensable component of methods computing higher-order sensitivity derivatives.

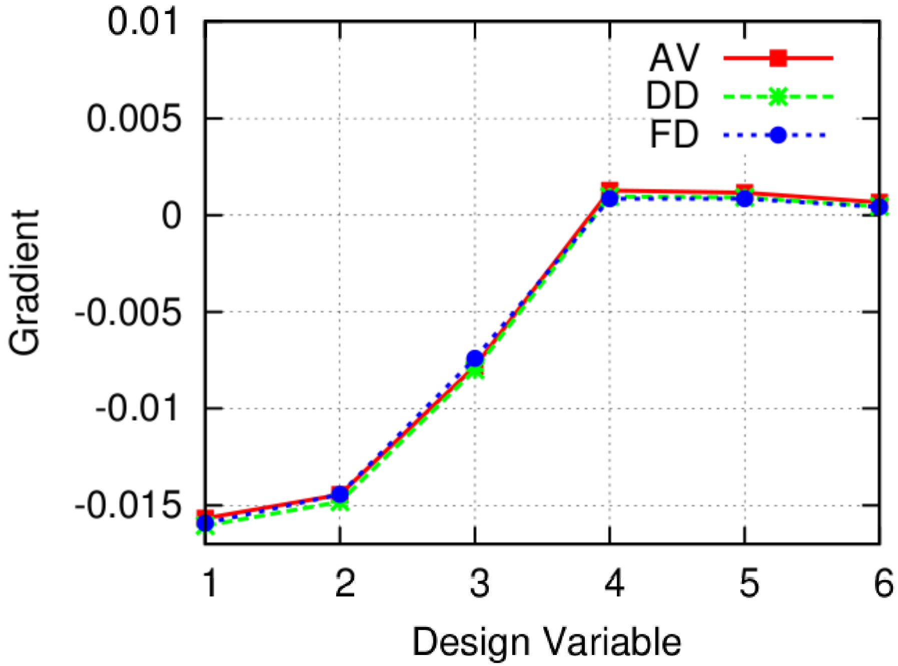
Inverse Design of a 2D Compressor Cascade

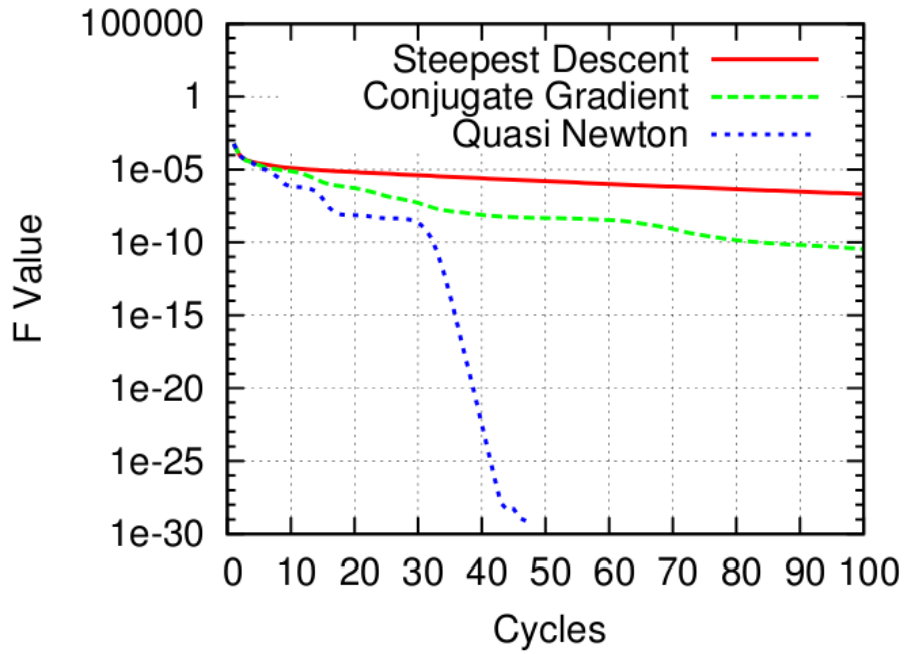


►Comparison of the gradient(F) computed by the adjoint method (**AV**) with finite-differences (**FD**) and direct-differentiation (**DD**).

►Comparison of more than one descent methods.







Adjoint to the Spalart-Allmaras (SA) Turbulence Model

__State Equations :__ ***(Turbulent Flows of an Incompressible Fluid)***

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

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] = 0\ , \qquad i = 1, 2(, 3)$$

$$\nu_t = \widetilde{\nu} f_{v_1}$$

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

The idea is to avoid making the usual assumption that "*shape variations do not affect turbulence*" ($\frac{\partial \nu_t}{\partial b_m} = 0$). To do so, we introduce the adjoint turbulent viscosity into F_{aug},

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

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

Adjoint to the Spalart-Allmaras (SA) Turbulence Model

•<u>What is new:</u>

•**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!)**

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

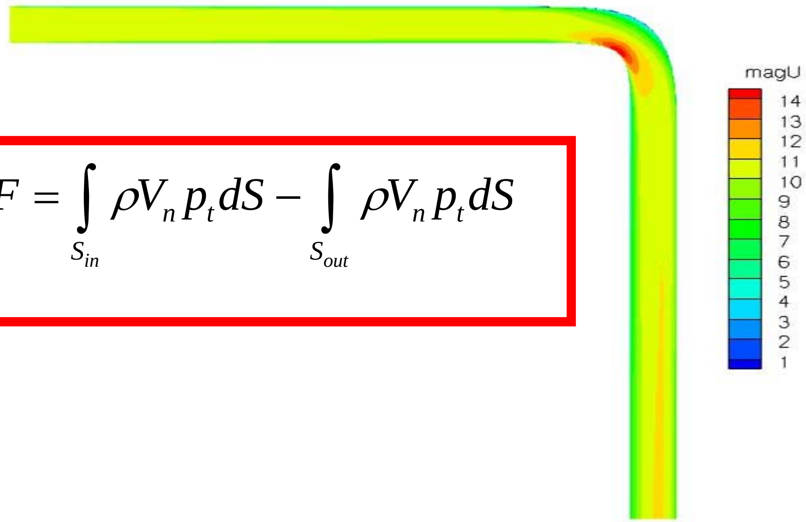
•**New terms in their boundary conditions**

•**New terms in the sensitivity derivative expressions**

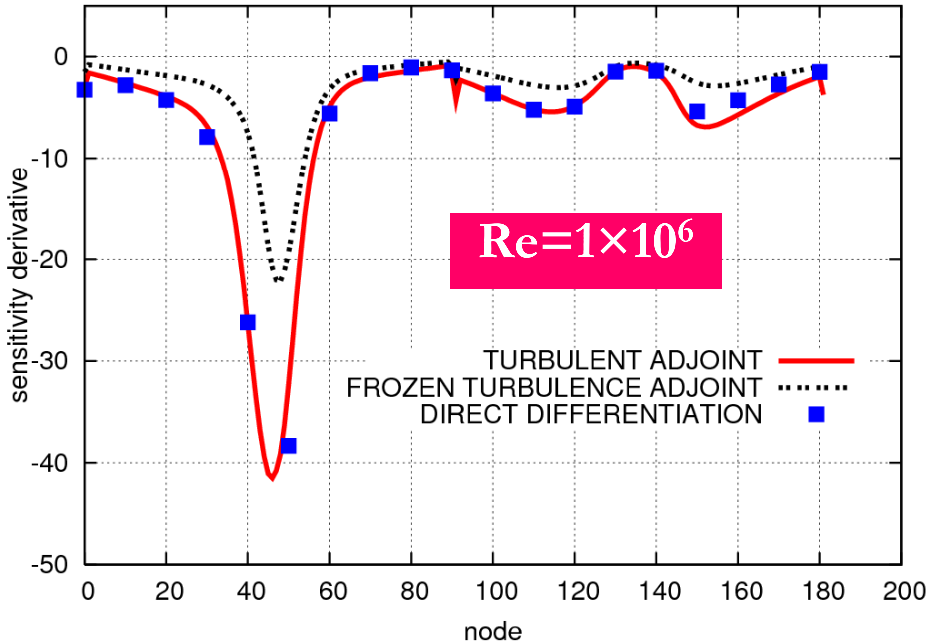
A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: ‘Continuous Adjoint Approach to the Spalart-Allmaras Turbulence Model for Incompressible Flows’, Computers & Fluids, 38, pp. 1528-1538, 2009.

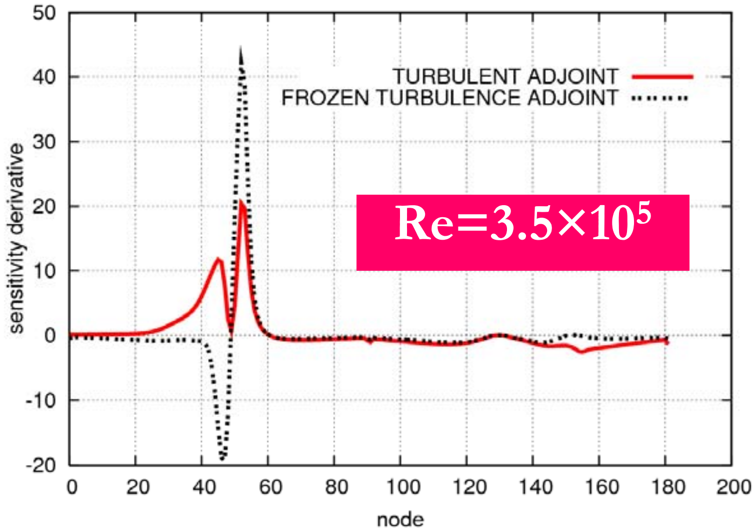
Adjoint to the Spalart-Allmaras (SA) Turbulence Model

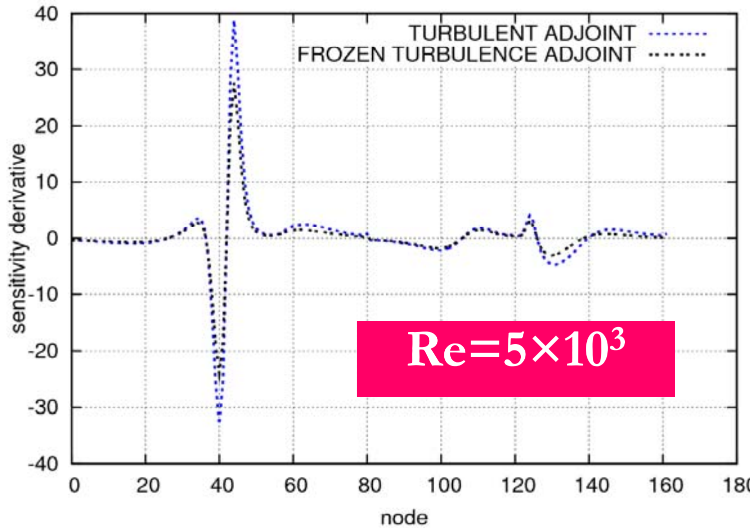
Why???



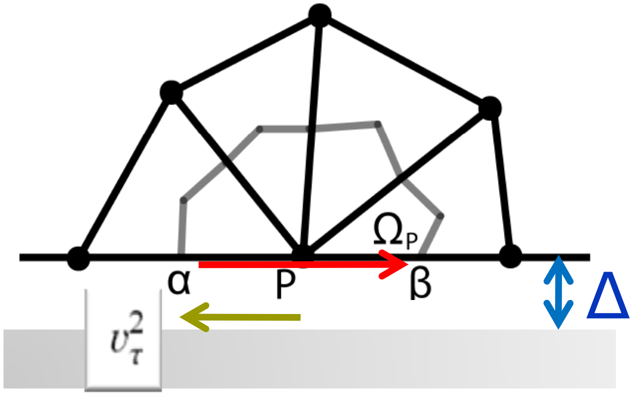
$$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$







Adjoint Wall Functions (k-ε Model)



Example: Unstructured grids, finite-volumes, k-ε turbulence model, with slip velocity at the wall

$$y^+ = \frac{\Delta \upsilon_\tau}{\nu} \qquad \upsilon^+ = \frac{\upsilon_t}{\upsilon_\tau}$$

Friction velocity :

$$\upsilon_\tau^2 = (\nu + \nu_t)\left(\frac{\partial \upsilon_i}{\partial x_j} + \frac{\partial \upsilon_j}{\partial x_i}\right) n_j t_i$$

$y^+ < y_c^+$	$y^+ \geq y_c^+$
$\upsilon^+ = y^+$	$\upsilon^+ = \frac{1}{\kappa} ln y^+ + B$
$k_P = \frac{\upsilon_\tau^2}{\sqrt{c_\mu}}\left(\frac{y^+}{y_c^+}\right)^2$	$k_P = \frac{\upsilon_\tau^2}{\sqrt{c_\mu}}$
$\varepsilon_P = k_P^{\frac{3}{2}} \frac{1+\frac{5.3\nu}{\sqrt{k_P}\Delta}}{\kappa c_\mu^{-\frac{3}{4}} \Delta}$	$\varepsilon_P = \frac{\upsilon_\tau^3}{\kappa \Delta}$

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

Adjoint Wall Functions (k-ε Model)

… after satisfying the field adjoint equations and eliminating the field integrals,

$$\frac{\delta F}{\delta b_m} = \int_{S_w}\left(D_i^u\frac{\partial v_i}{\partial b_m} + D^{k_a}\frac{\partial k}{\partial b_m} + D^{\varepsilon_a}\frac{\partial \varepsilon}{\partial b_m}\right)dS - \int_{S_w}(\nu+\nu_t)\,u_i\frac{\partial}{\partial b_m}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)n_j dS + \int_{S_w}T_k^1\frac{\delta x_k}{\delta b_m}dS$$

But all non-geometrical quantities along the "wall" depend on the friction velocity v_τ, so all we have to do is to eliminate variations in v_τ. This can be proved to be equivalent to the definition of the adjoint friction velocity

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

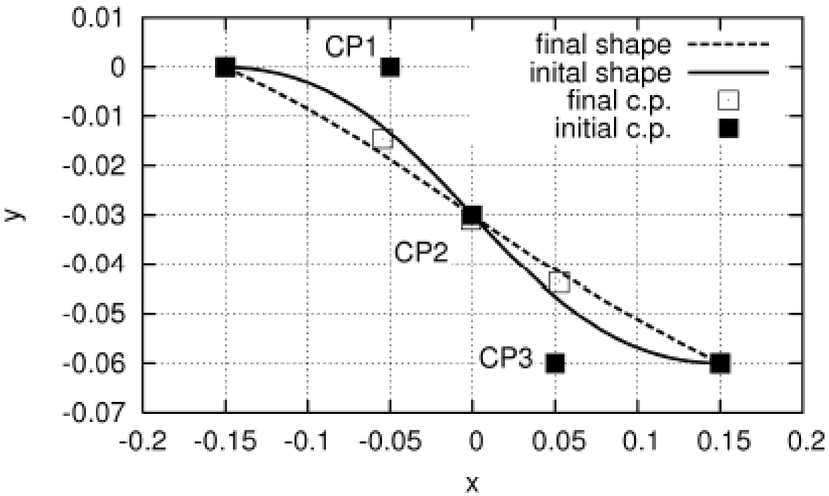
or

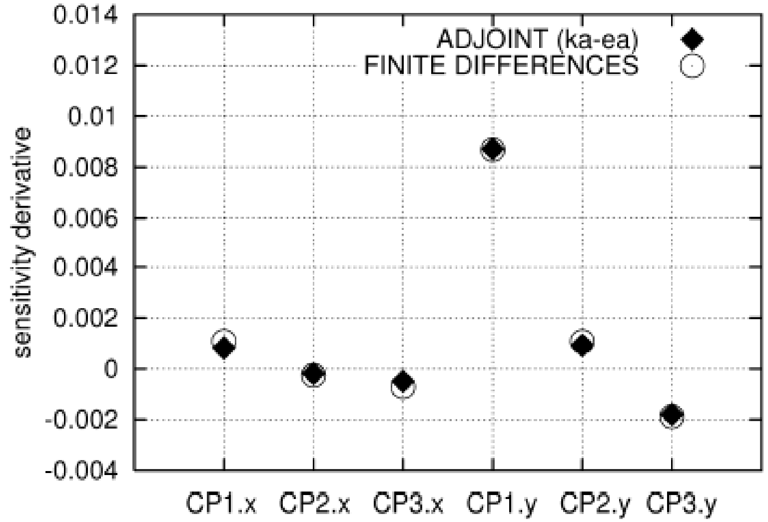
$$u_\tau^2 = \frac{1}{c_v}\left[2u_k t_k v_\tau - \left(\nu + \frac{\nu_t}{Pr_k}\right)\frac{\partial k_a}{\partial x_j}n_j\frac{\delta k}{\delta v_\tau} - \left(\nu+\frac{\nu_t}{Pr_\varepsilon}\right)\frac{\partial \varepsilon_a}{\partial x_j}n_j\frac{\delta\varepsilon}{\delta v_\tau}\right]$$

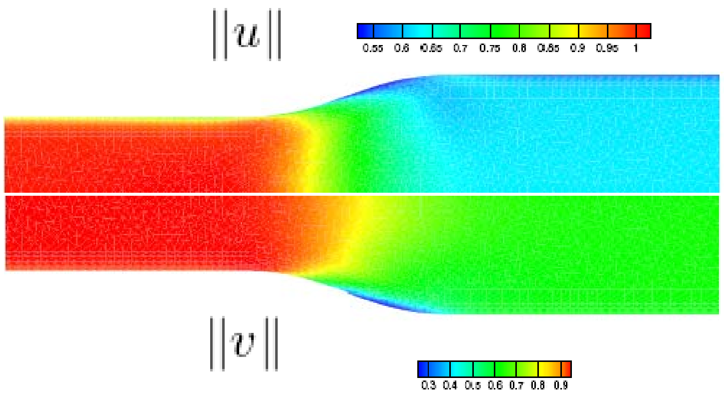
and introduces what we will refer to as the adjoint law of the wall.

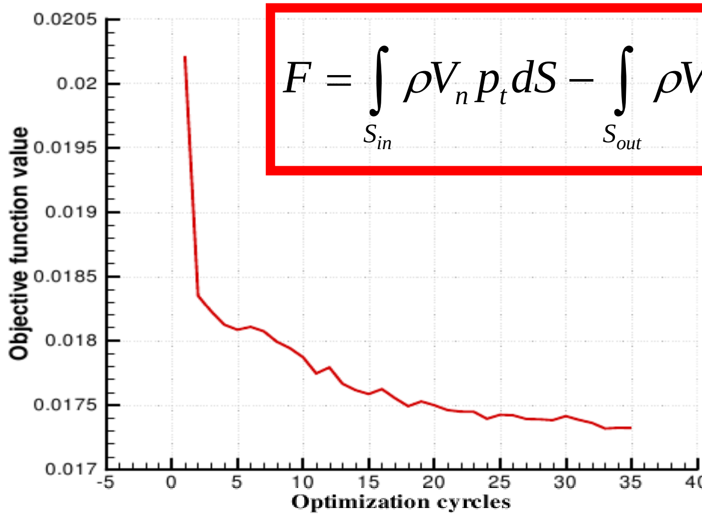
A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: 'Adjoint Wall Functions: A New Concept for Use in Aerodynamic Shape Optimization', Journal of Computational Physics, Vol. 229, pp. 5228–5245 , 2010.

Adjoint Wall Functions (k-ε Model) - Application





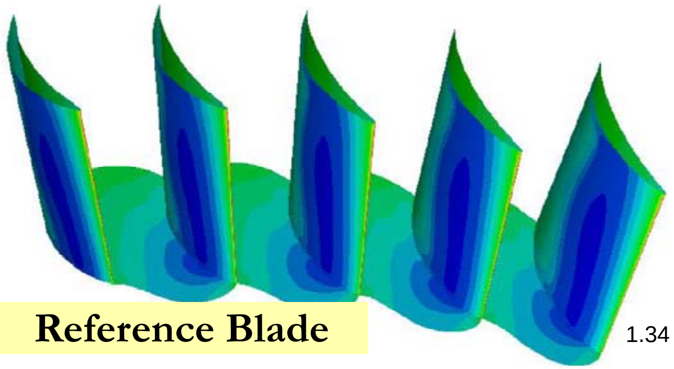




$$F = \int_{S_{in}} \rho V_n p_t dS - \int_{S_{out}} \rho V_n p_t dS$$

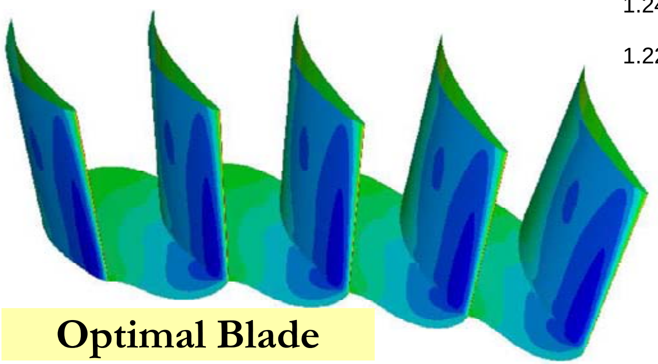
Design of an axial diffuser with minimum total pressure losses (Re=1x10^6)

Applications of the Adjoint Method in Turbomachinery

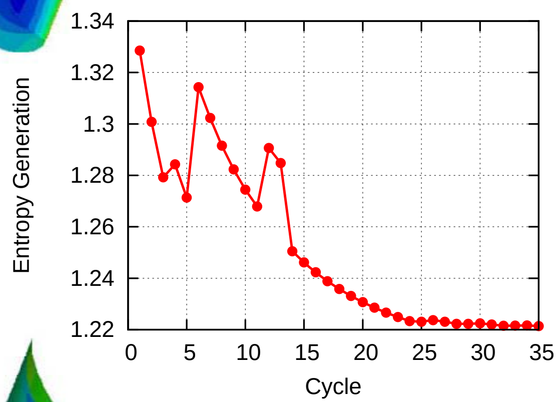


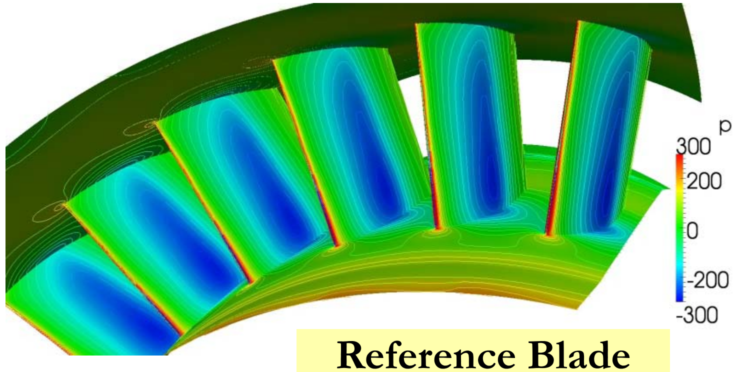
Reference Blade

Row 1



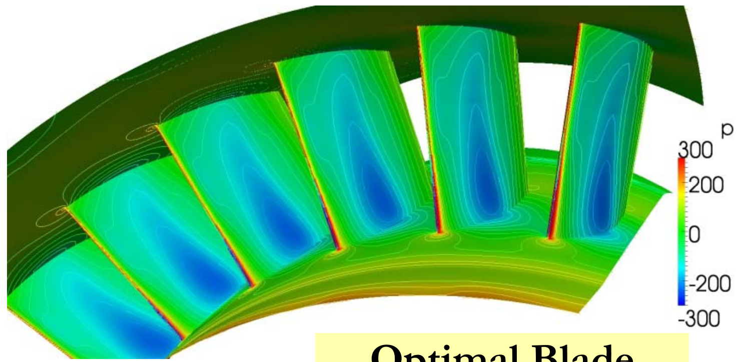
Optimal Blade





Reference Blade

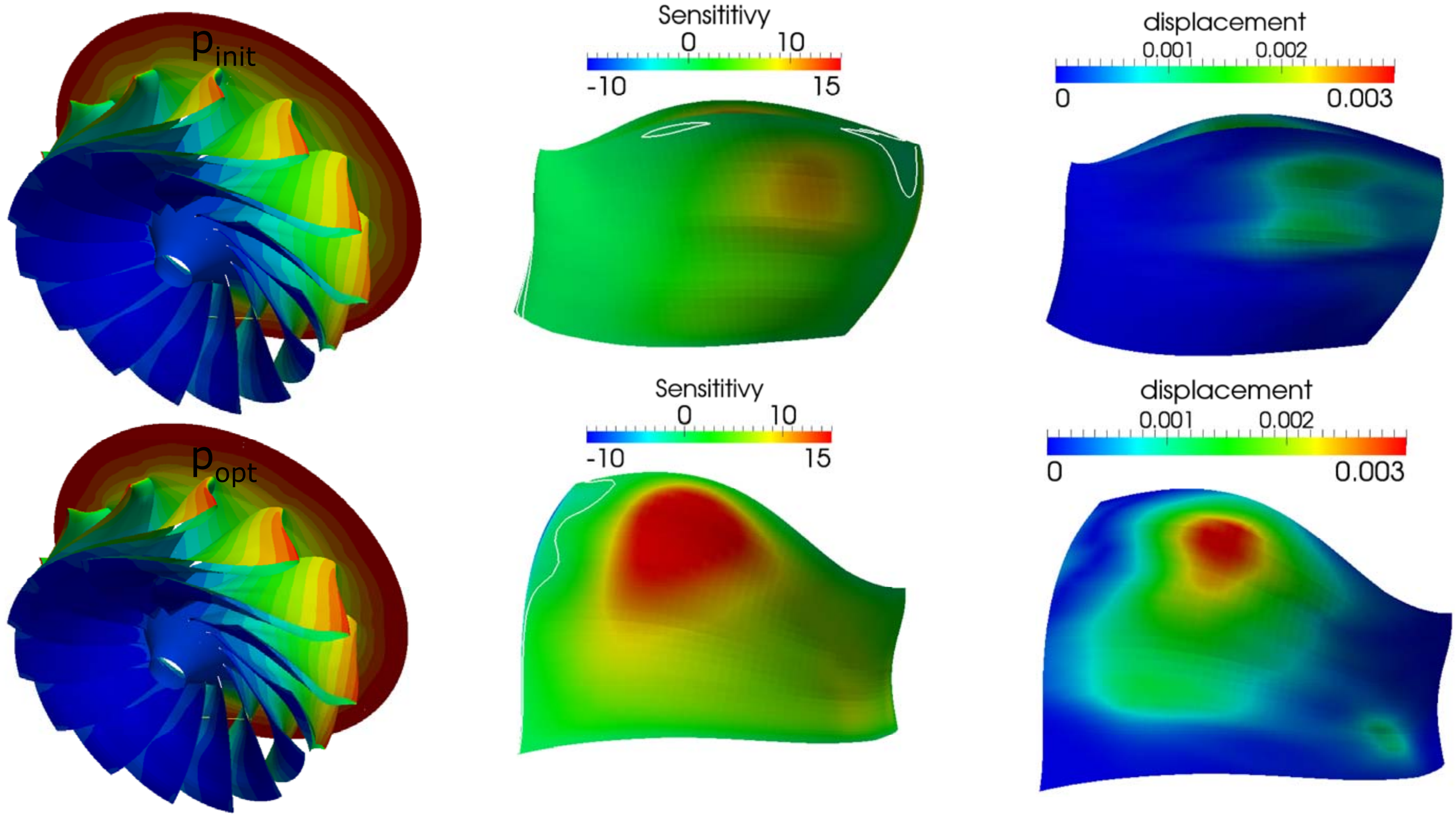
Row 2



Optimal Blade

Design-Optimization of a 3D peripheral compressor rows, for minimal viscous losses, with geometrical constraints, using the continuous adjoint method. Turbulence model : Low-Reynolds number Spalart-Allmaras.

Applications of the Adjoint Method in Turbomachinery



Optimization of a Francis turbine blade, targeting a 1.5m increase in the hydraulic height, subject to a number of flow constraints, incl. cavitation.

ANDRITZ Hydro

Hessian Matrix Computation

<u>Newton Methods :</u>

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

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

<u>**Hessian Matrix Computation using the DD-DD method:**</u>

(Think "Discrete" , program Continuous Adjoint)

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

k=1,…,N design variables

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

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

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

► **The cost for computing the Hessian via the DD-DD approach scales with N².**

Computation of the Hessian Matrix, via DD-AV

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

System solutions (EFS)

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

EFS

The Adjoint equation is <u>the same</u> with that obtained for the Gradient !!!

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

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

- ► **The cost per Newton cycle is N+1+1=N+2 EFS! Scales with N, not N^2.**
- ► **DD-AV is the <u>most efficient</u> approach (among DD-DD, AV-DD, AV-AV)!**

Computation of the Hessian Matrix

<u>Using Continuous Adjoint –</u> The DD-AV Scheme:

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

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

<u>Relevant references, in both continuous and discrete adjoint:</u>

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.

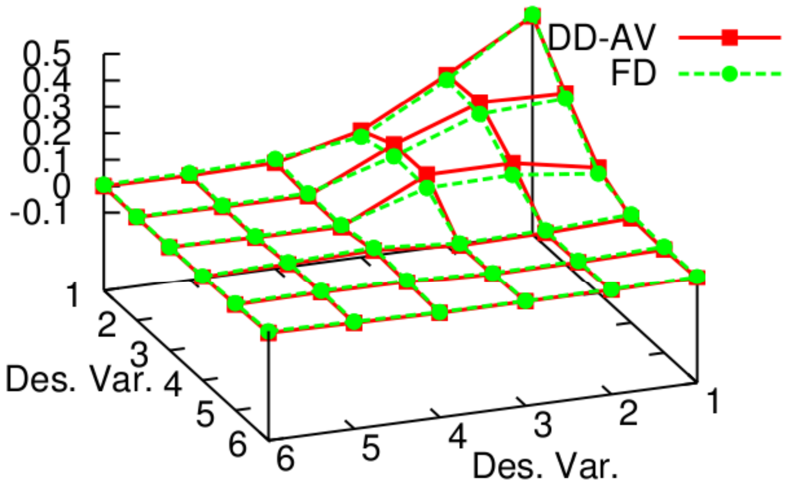
K.C. GIANNAKOGLOU, D.I. PAPADIMITRIOU: 'Adjoint Methods for gradient- and Hessian-based Aerodynamic Shape Optimization', EUROGEN 2007, Jyvaskyla, Finland, June 11-13, 2007.

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.

Computation of the Hessian Matrix - Application

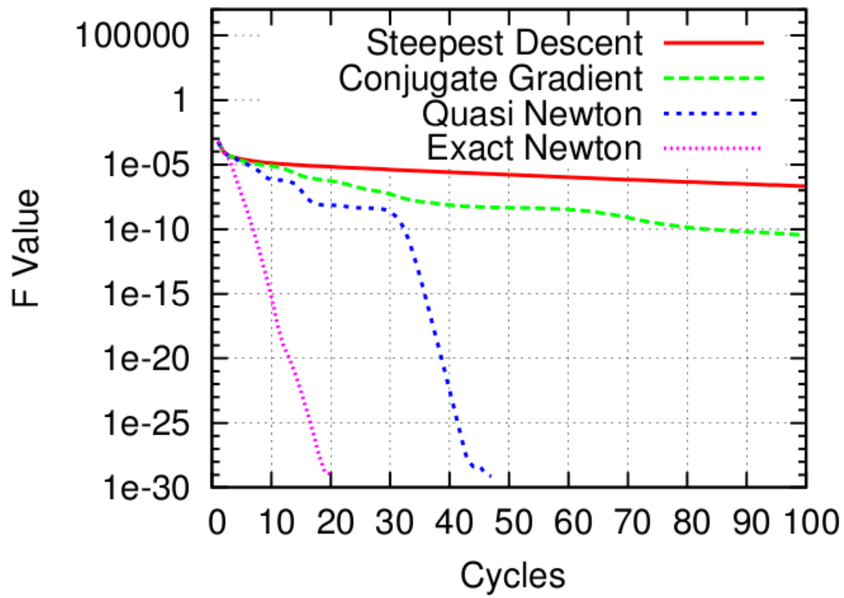
Inverse Design of a Compressor Cascade, Inviscid Flow

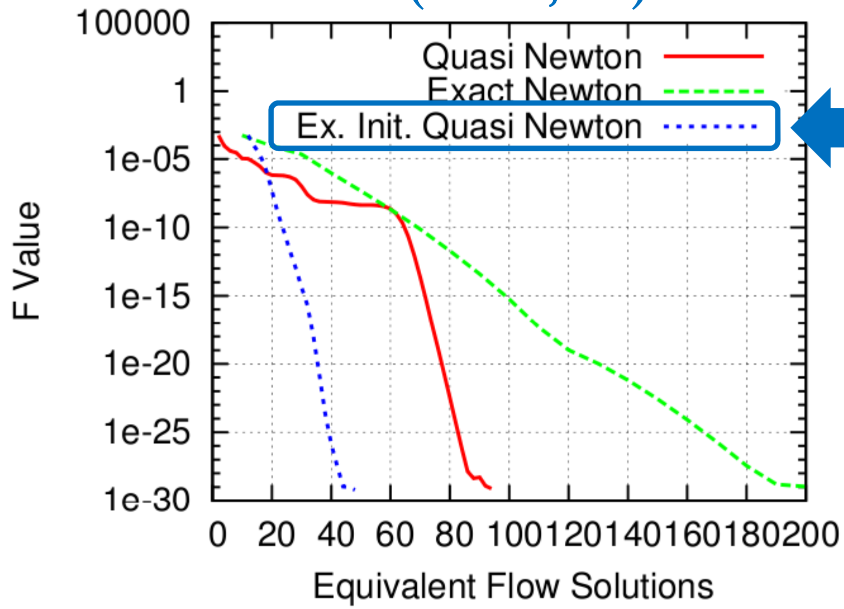


$$\frac{5\times 6}{2}\times 4 = 60 \quad \text{vs.} \quad 6+1=7 \text{ EFS}$$

(FD) (DD-AV)

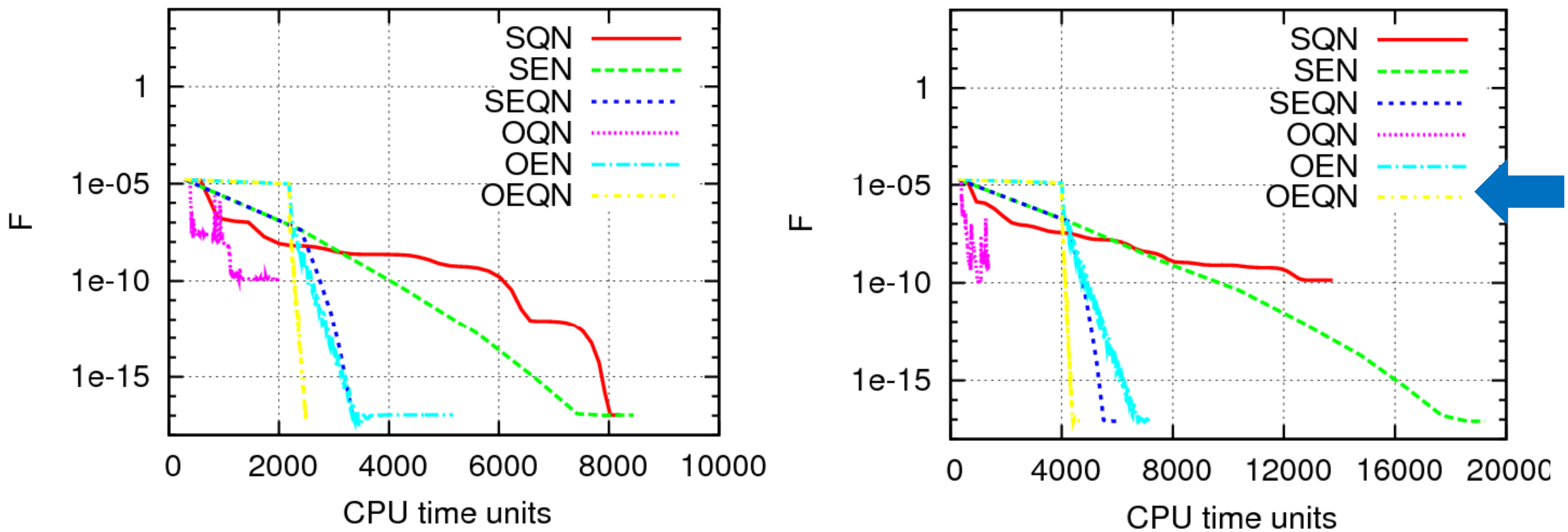
Compute the Hessian only once, in the first cycle and, then, switch to quasi-Newton methods (BFGS, etc)





One-Shot Variants – Overall CPU time comparison

SSD	▶ Segregated Steepest Descent	OSD	▶ One-Shot Steepest Descent
SQN	▶ Segregated Quasi-Newton (BFGS)	OQN	▶ One-Shot Quasi-Newton (BFGS)
SEN	▶ Segregated Exact Newton	OEN	▶ One-Shot Exact Newton
SEQN	▶ Segregated Exact(*first cycle*)-Quasi(*then*) Newton	OEQN	▶ One-Shot Exact(*first cycle*)-Quasi(*then*) Newton



Design of a Compressor Cascade, 6 (Left) & 12 (Right) Design Variables

D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'One-Shot Shape Optimization Using the Exact Hessian', ECCOMAS CFD 2010, 5th European Conference on CFD, Lisbon, Portugal, June 14-17, 2010.

Truncated Newton Methods

Why?

- ► Create more efficient Hessian-based optimization schemes.
- ► Desirable cost per cycle <<N EFS, without damaging accuracy.

Inspired by:

The Conjugate Gradient (CG) method for solving systems of linear equations

$$\boxed{Ax = q}$$

requires only matrix-vector products.

$k \leftarrow 0$

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

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

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 $k \leq k_{max}$ **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** $m \leq M_{CG}$ **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

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

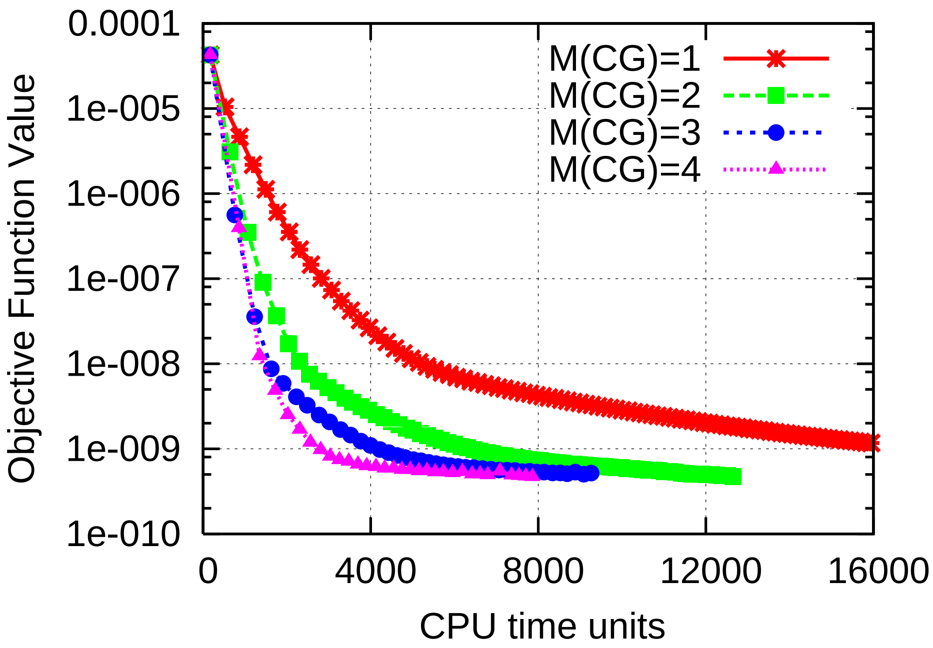
$$\frac{d^2F}{db_i db_j} db_j = -\frac{dF}{db_i}$$

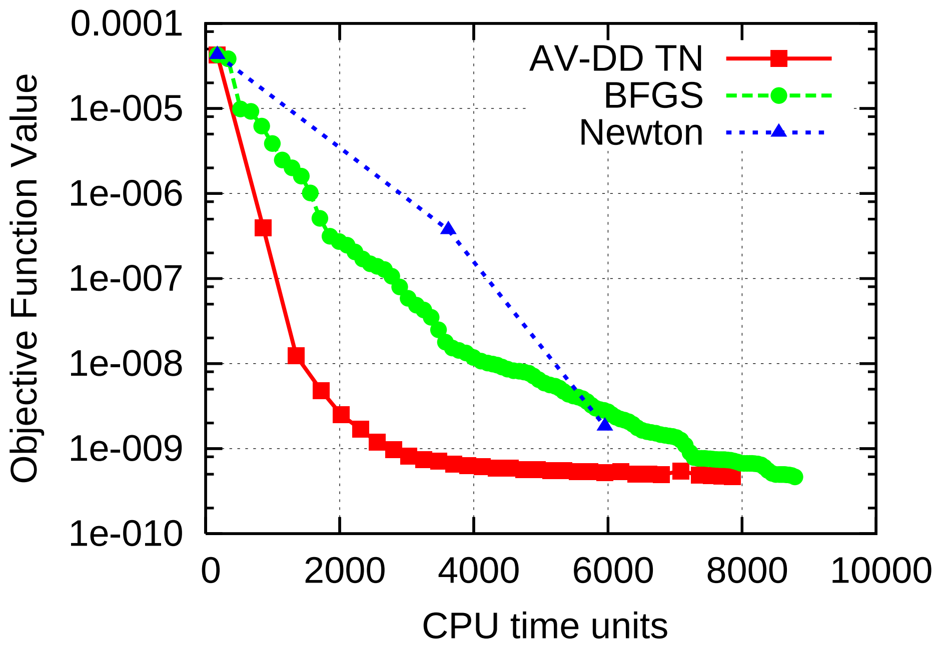
Total Cost= 2+2M$_{CG}$ << N

AV-DD Truncated Newton method – Why & How?

Parametric Study of the recommended value of the number of CG steps (M_{CG})

Comparison of
AV-DD Truncated Newton method,
quasi-Newton BFGS &
(exact) Newton

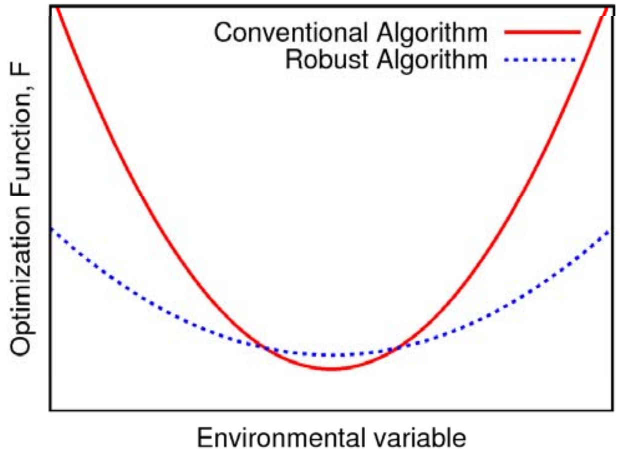




Inverse Design of an Isolated Airfoil, 42 degrees of freedom

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, 2011.

(Second-Order, Second-Moment, SOSM, approach)



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

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

where the estimated mean and the standard deviation of F are given by

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

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

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

► The recommended approach, if M<<N,

DD_c-DD_c-AV_b

exists and has been programmed in both discrete and continuous adjoint.

Robust Design

The CPU cost of the DD_c-DD_c-AV_b method:

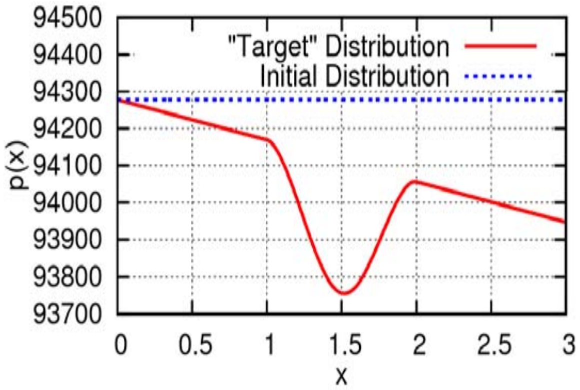
Derivative	Method	Cost	Cost when $M=1$
$\frac{dF}{db_l}$	AV	1	1
$\frac{dF}{dc_i}$	DD	M	1
$\frac{d^2F}{dc_i dc_j}$	DD-DD	$\frac{M(M+1)}{2}+M$	2
$\frac{d^2F}{dc_i db_l}$	DD-AV	$1+M$	2
$\frac{d^3F}{dc_i dc_j db_l}$	DD-DD-AV	$2+3M+M^2$	6

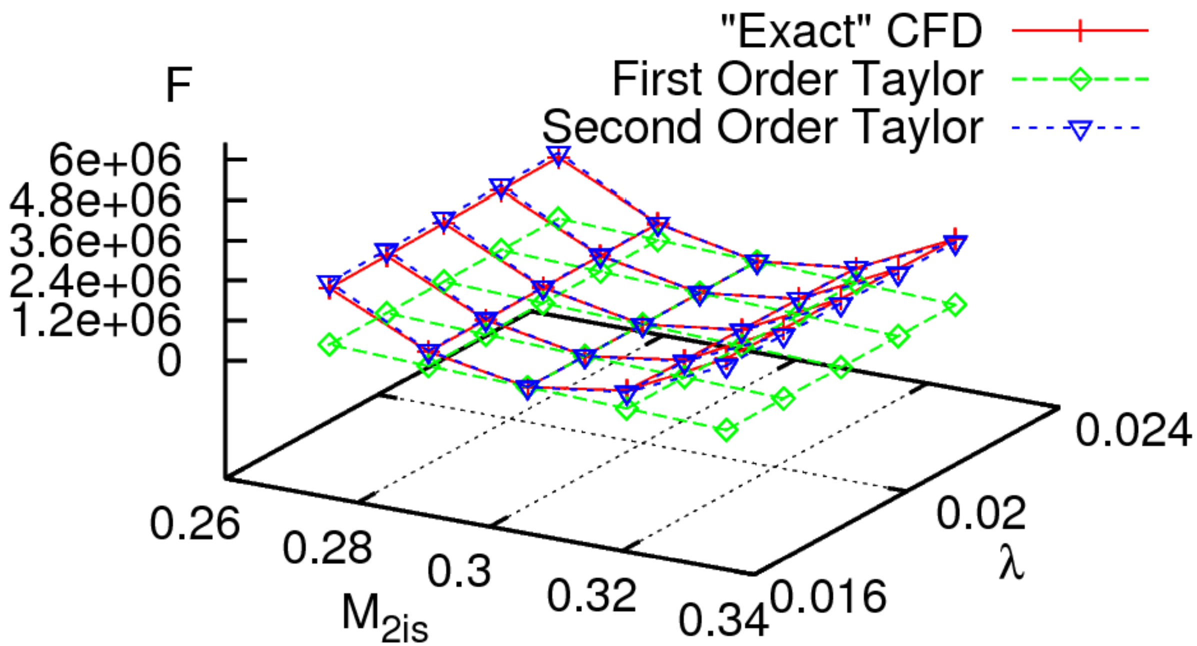
E.M. PAPOUTSIS-KIACHAGIAS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU: 'Robust Design in Aerodynamics using 3rd-Order Sensitivity Analysis based on Discrete Adjoint. Application to Quasi-1D Flows', International Journal for Numerical Methods in Fluids, to appear 2011.

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.

Robust Design – A Pseudo 1D Example

$$\frac{\partial \vec{f}}{\partial x} = \vec{q_s} + \vec{q_\nu} \qquad \vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix}, \vec{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{bmatrix}, \vec{q_s} = -\frac{1}{S}\frac{dS}{dx}\begin{bmatrix} \rho u \\ \rho u^2 \\ u(\rho E + p) \end{bmatrix}, \quad \vec{q_\nu} = -\lambda \frac{dx_i}{2D_i}\begin{bmatrix} 0 \\ \rho u^2 \\ \rho u^3 \end{bmatrix}$$



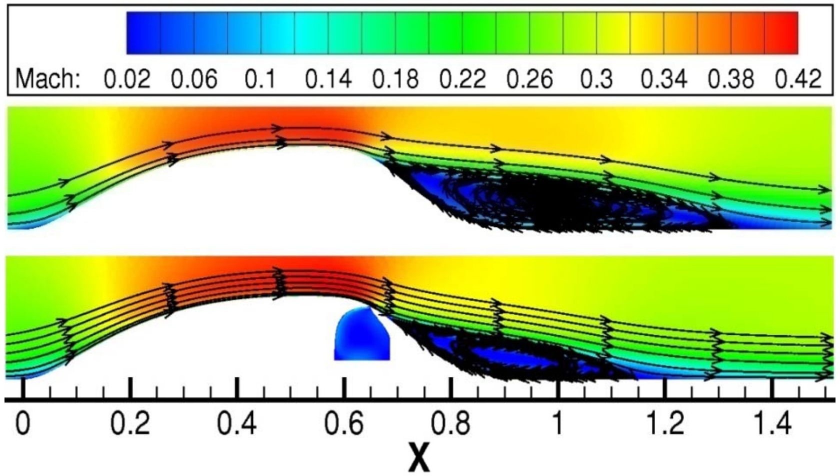


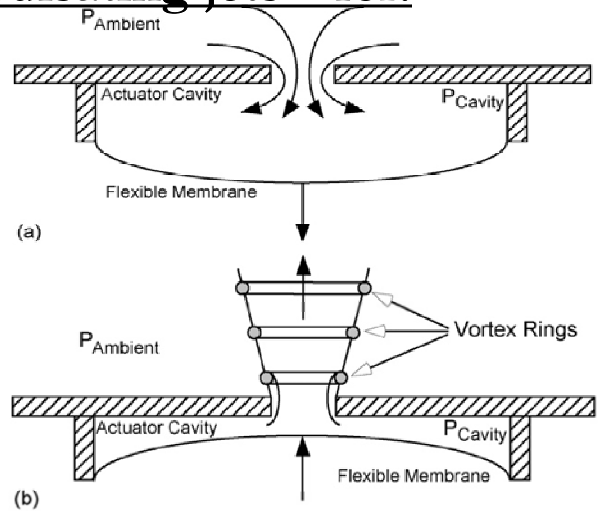
Two environmental variables:

- **Outlet Mach number M_2**
- **Darcy friction loss coefficient λ**

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, to appear 2011

<u>Flow Control – here, by means of continuous or pulsating jets – for:</u>





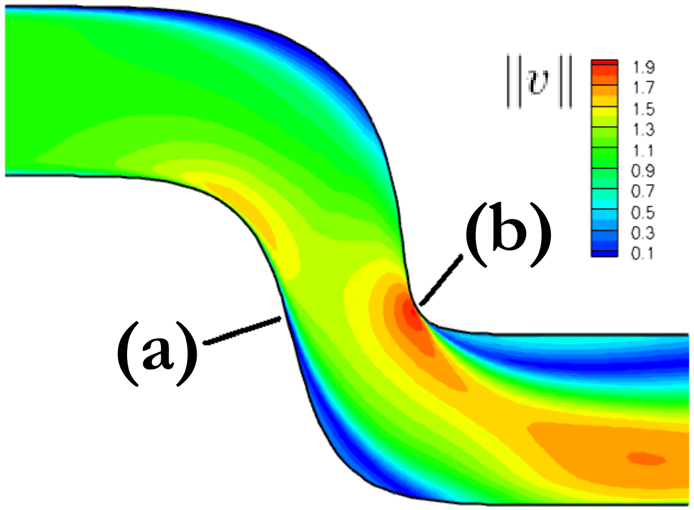
► Assume suction or blowing along the "solid" (perforated?) walls.

► Develop the (continuous) adjoint method using drag or p_t-losses as obj. function and the normal to the "wall" jet velocity as design variables (see references).

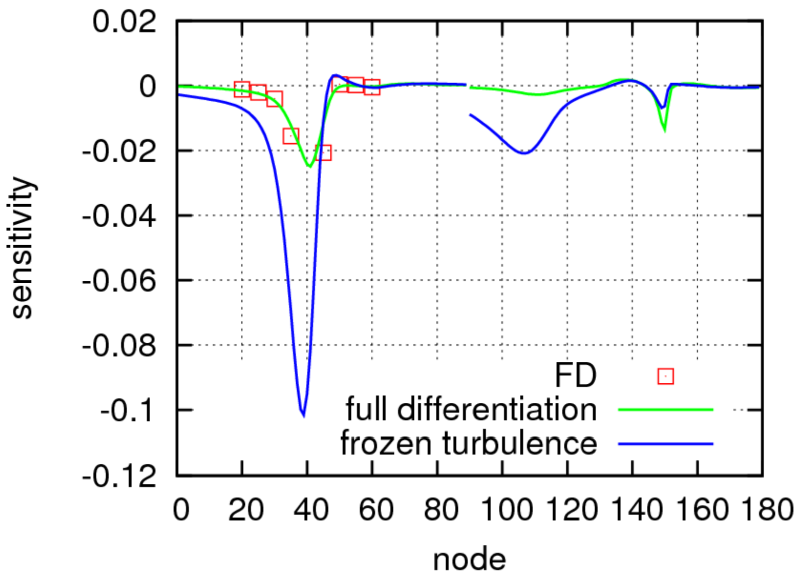
► Be careful: $$\frac{\delta \Phi}{\delta b_i} = \frac{\partial \Phi}{\partial b_i} + \cancel{\frac{\partial \Phi}{\partial x_l}\frac{\delta x_l}{\delta b_i}}$$ for b_i=normal_jet_velocity

► An optimization problem with thousands of design variables! Adjoint can make it!

► <u>Idea</u>: Compute the sensitivity derivatives by solving the flow & adjoint problem once, for normal_jet_velocity=<u>0</u>; use the so-computed sensitivity maps to optimally locate the jets. Stop here or iterate to optimize all jet parameters.



Objective function : Pt losses



$$\frac{\delta F}{\delta v_q^{jet}}$$

CROSS-CHECK: Jet application at the most promising positions:

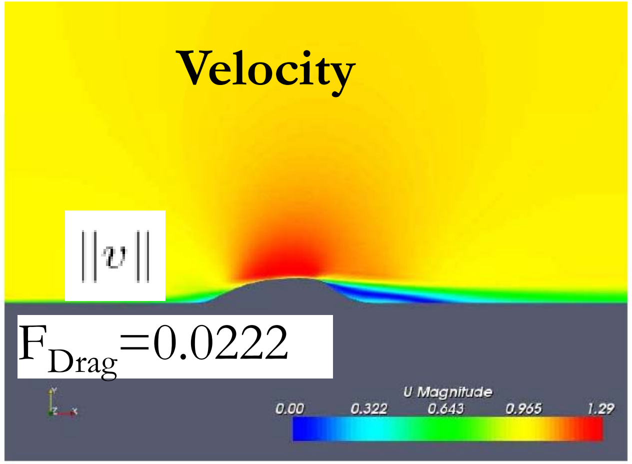
	uncontrolled	study (a)	study (b)	study (c)
F	0.01835	0.01662	0.01817	0.01649

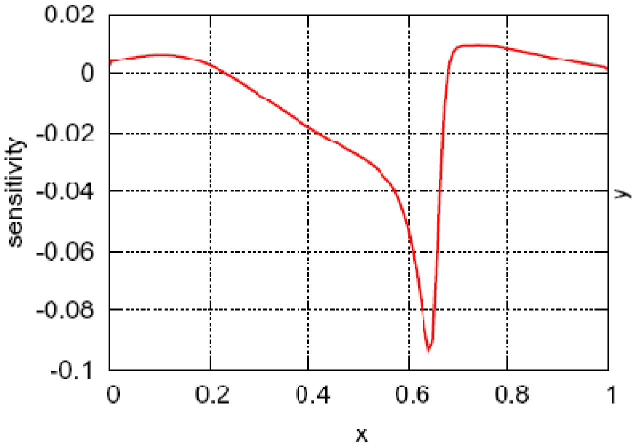
(study (c) is the combination of (a)&(b))

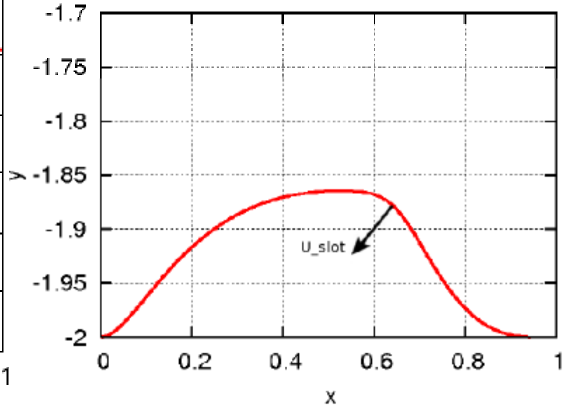
A.S. ZYMARIS, D.I. PAPADIMITRIOU, K.C. GIANNAKOGLOU, C. OTHMER: 'Optimal Location of Suction or Blowing Jets Using the Continuous Adjoint Approach', ECCOMAS CFD 2010, 5th European Conference on CFD, Lisbon, Portugal, June 14-17, 2010

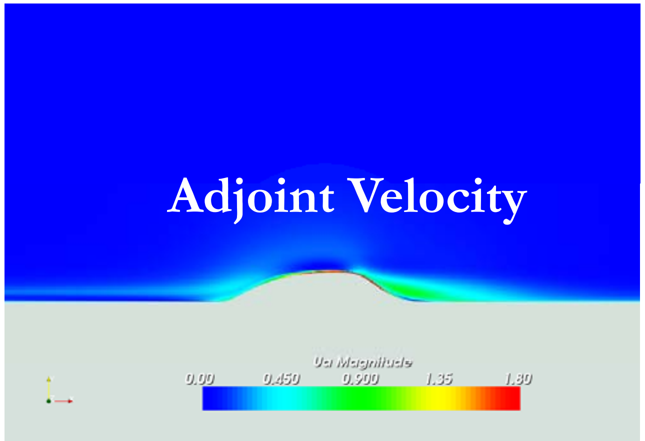
Continuous Adjoint Method for Flow Control Problems

Objective function : Drag

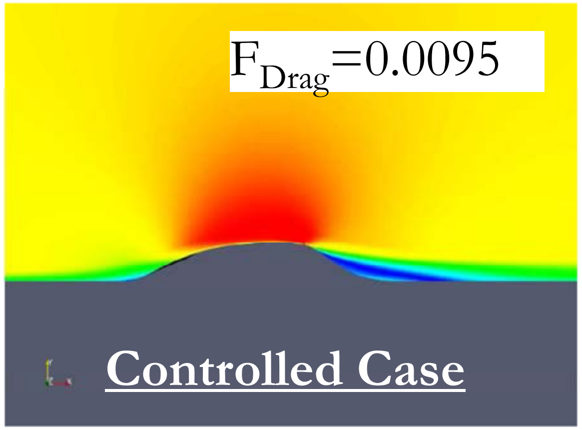






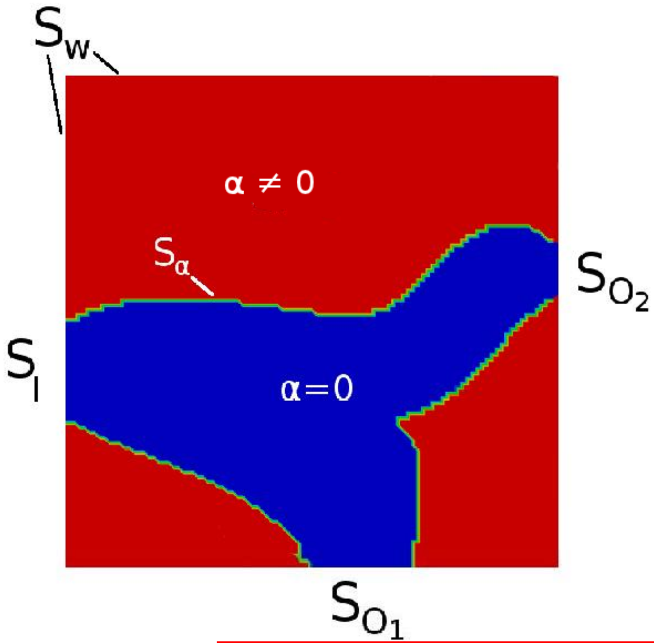


CROSS-CHECK:



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, 5th European Conference on CFD, Lisbon, Portugal, June 14-17, 2010

Topology Optimization & Continuous Adjoint Method



Flow Model:
Incompressible fluid
Turbulent flow
With heat transfer effects

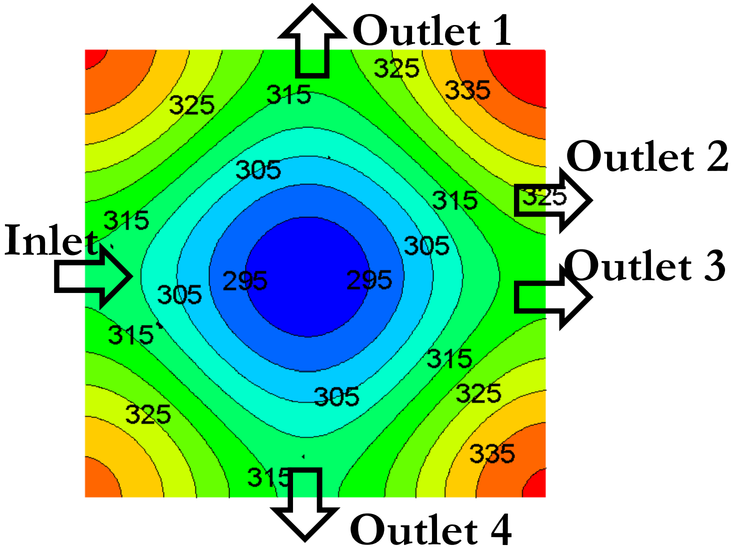
$$R_p = 0, \quad R_{v_i} = 0, \quad R_T = 0, \quad R_{\widetilde{\nu}} = 0$$

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

Adjoint Equations:

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

Topology Optimization & Continuous Adjoint Method

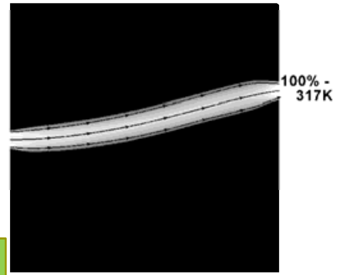


$$f_1 = -\int_{S_I}\left(p+\frac{1}{2}v_k^2\right)v_i n_i dS - \int_{S_O}\left(p+\frac{1}{2}v_k^2\right)v_i n_i dS$$

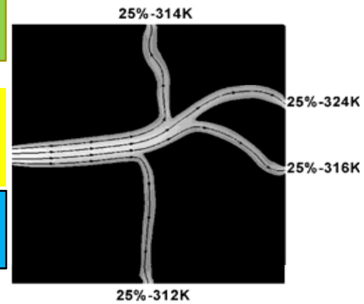
$$f_2 = \int_{S_I} T v_i n_i dS + \int_{S_O} T v_i n_i dS$$

$$F = \omega_1 f_1 - \omega_2 f_2$$

f_1=0.025, f_2=2.48

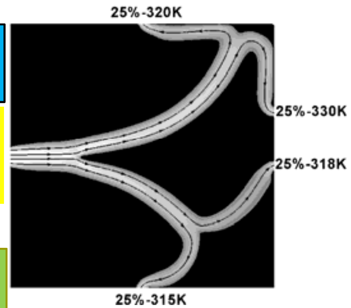


Min. Δp_t



Constraint: $m_{out\ 1,2,3,4}$ = 25% m_{inlet}

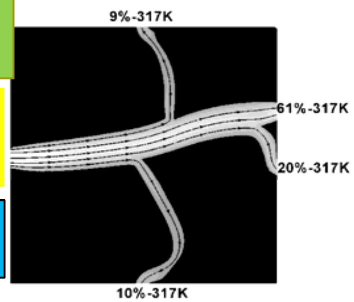
f_1=0.037, f_2=3.00



f_1=0.053, f_2=3.51

Constraint: $m_{out\ 1,2,3,4}$ = 25% m_{inlet}

Min. Δp_t
Max. ΔT

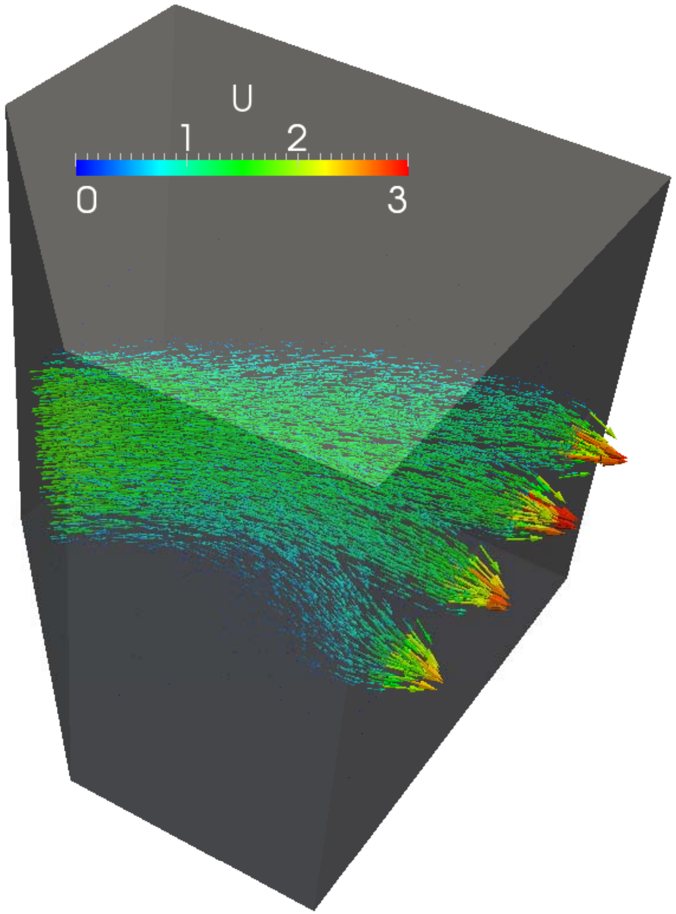


Constraint: T_{out1}= T_{out2} = T_{out3} = T_{out4}

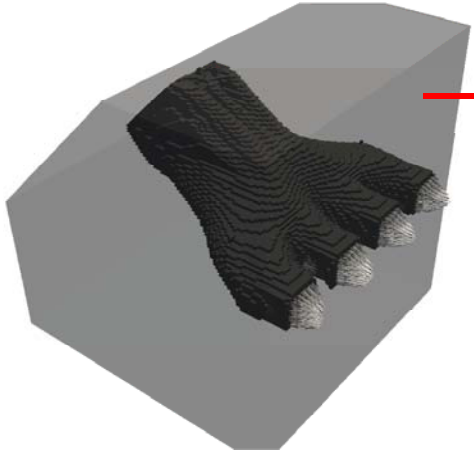
f_1=0.026, f_2=2.83

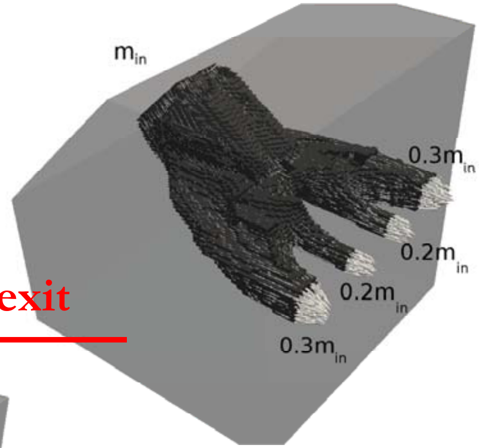
Topology Optimization & Continuous Adjoint Method

Topology optimization of a manifold at laminar flow conditions.

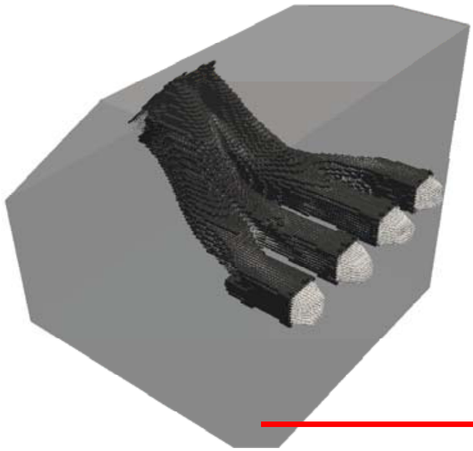


Unconstrained





With constraint on the mass flowrate per exit



With constraint on the Flow swirl at the exit

Summary /Conclusions / Acknowledgement

- Working with continuous adjoint is nice because you gain insight into adjoint PDEs & their BCs or clearly understand/control the assumptions made.
- (Exact) adjoint methods lead to expressions for the objective function gradient which comprise only boundary integrals, even if the objective function is a field integral.
- There are good reasons for developing and using the adjoint to the turbulence model equations. Stop working with the "frozen-turbulence assumption".
- The adjoint law of the wall is a useful tool for industrial applications.
- DD and AV can be used to compute the Hessian, allowing the use of exact Newton methods. For high-dimensional problems, try the (one-shot) exactly-initialized quasi-Newton algorithm, which outperforms both exact and quasi-Newton methods.
- The Truncated Newton Methods avoids Hessian matrix computations and is faster.
- Robust design methods (SOSM approach) benefit from the availability of efficient methods to compute high-order derivatives of F.

Parts of this research were funded by: