ACTIVE FLOW CONTROL OVER A CIRCULAR CYLINDER

Catalin Nae
INCAS – National Institute for Aerospace Research
Iuliu Maniu no. 220, sect. 6, 77538 Bucharest, ROMANIA
E-mail : cnae@aero.incas.ro

Keywords: *Flow control, CFD, Automatic differentiation, Optimization*

Abstract

The flow control problem is considered using microjet actuators and an optimization algorithm applied for a cost function defined as a boundary integral on the reference shape surface. The numerical approach is based on an unsteady optimization process involving a RANS flow solver, a numerical model for the actuators and a global unsteady optimization algorithm, based on the BFGS. The actuators are controlled as to minimize the cost function using gradients provided by the use of an automatic differentiation tool applied to the source code of the flow solver.

Results are presented for the numerical simulation of the actuator and the flow around the circular cylinder using one apriori given control law and for active flow control using synthetic jet actuators.

1 Introduction

Actuators like synthetic jets have the potential to change flow characteristics for some practical configurations. They can be very easily used for lift control or stall delay, and this can even be done without using active control laws [4][5][6]. For such applications, when the global effect is obvious from the beginning, from the mathematical point of view we can consider that we have a flow control problem with a given apriori law.

For more general and complex cases, the use of the jet actuators for active flow control on complex geometries may be considered as a mathematical optimization problem, reformulated as unsteady minimization of a cost function. The resulting control laws may be derived from a gradient based optimization algorithm, using mainly information on a reference surface in order to define an appropriate cost function [10][16]. As an example, we will use the case of the viscous flow over a circular cylinder [8], well known to produce a Karman vortex alley and strong oscillations both in lift and drag.

The numerical simulations in this paper are based on CFD techniques using a RANS 2D code with a modified k - ε turbulence model. A time averaged drag formulation is considered as a cost function to be minimized using control theory and an gradient based algorithm for active flow control. Gradients will be generated using automatic differentiation of the solver using several assumption made on the dependence of the cost function on the state (flow variables) [7].

2 The Unsteady Flow Control Problem

The potential of the SJ actuators for fluid control was investigated in various conditions in a previous work [17]. We will consider the problem of the circular cylinder as an unsteady optimization problem for the cost function which will be drag. The controls are numerically simulated by their external characteristics (frequency and top blowing/suction speed) at their location represented on the surface of the cylinder. The optimization algorithm will provide the optimum control laws for the actuators and the corresponding value of the minimum cost function using a gradient based method where the gradient is to be evaluated using automatic differentiation of the flow solver.

2.1 The Optimization Problem for Flow Control

We consider that the flow around the cylinder at any moment being cvasiperiodic, with a period independent of the controls . Also, because we expect that the controls influence to be computed using a time marching technique, we expect several time periods to be the representative time interval. This means that the controls are supposed to have little influence on the global motion and that the final motion is supposed to be also cvasiperiodical.

The optimization problem under these assumption will be to minimize the drag coefficient *Cd* using a set of controls on the boundary :

$$C(\varphi,\zeta)=\min_{\varphi,\zeta} J(\varphi,\zeta)=\frac{1}{n_T}\int_0^{n_T\cdot T} Cd(\varphi(t),\zeta(t))\cdot dt \qquad (1)$$

where:

$$(\varphi,\zeta)\in\{R^m x R^m x[0,T]\,|\,0\le\varphi\le\varphi_{\max},0\le\zeta\le\zeta_{\max}\} \qquad (2)$$

T is the time period of the cvasiperiodic flow solution for the case with no controls, and *Cd* is computed from a boundary integral on the cylinder surface. The controls are individually defined by 2 control values (frequency and top speed) which are bounded by imposed technological restrictions.

The time period is numerically computed using the same infinity boundary conditions and grid (after all regular precaution in CFD for grid-independent flow solutions) as for the controlled case, and ideally this time period should be related to the Strouhal number obtained experimentally for this flow at the same Mach and Reynolds numbers, i.e. St = 0.21 . This value is fixed during the optimization process, but we use several time periods for a time-averaged analysis. Since the flow solver is using a global time step integration technique(the time step used is the minimum time step as given by the local time step of every point in the domain) for the unsteady flows, the discretisation of the time dependence of the cost function is made so that the constant time step used to be lower then the global time step as resulting from flow sensitivity.

$$\Delta t\le\Delta t(P_i)=\min\left(\frac{\Delta x}{|u|+c},\frac{1}{2}\cdot\rho\cdot\mathrm{Pr}\cdot\frac{\Delta x^2}{\mu+\mu_t}\right) \qquad (3)$$

The minimization problem in a discrete formulation, using a constant global time step as given by (3), will then be (at a time moment *n*):

$$C^n(\varphi,\zeta)=\min_{\varphi,\zeta} J^n(\varphi,\zeta)=\min_{\varphi,\zeta}\sum_{i=0}^{n}\Delta t\cdot Cd(\varphi^i,\zeta^i) \qquad (4)$$

which can be formulated as :

$$C^n(\varphi,\zeta)=C^{n+1}(\varphi,\zeta)+\Delta t\cdot\min_{\varphi^n,\zeta^n} Cd(\varphi^n,\zeta^n) \qquad (5)$$

for :

$$n\in[1,N]\ ,\ N=n_T\cdot\left(1+\mathrm{int}\left(\frac{T}{\Delta t}\right)\right). \qquad (6)$$

So we have to compute at each time step the value of $\min_{\varphi^n,\zeta^n} Cd(\varphi^n,\zeta^n)$ using :

- SJ actuators as controls
- the flow solver
- an appropriate optimization algorithm

2.2 The controlling devices

For the active flow control problem, we use controls for the minimization of the unsteady cost function drag. The controls used are synthetic jet (SJ) devices that have individual characteristics for the frequency and velocity profile.

Synthetic jets (SJ) result from an oscillating diaphragm in an enclosed space, having small orifices at the top. They can be controlled electrostatically or using piezoelectric materials with frequencies in the range of 0.5 – 20 kHz. Because air is drawn into the cavity by the low-level suction pressure created by the diaphragm and then is expelled by the same diaphragm, such devices are considered to produce a zero-mass jet. The peak velocity speed and the frequency are defining parameters. For practical devices with orifice diameters like 200 μm, peak velocity may be up to 20m/s, as reported in several experiments

[3][4]. Such actuators were previously investigated for flow control on various configurations and the results are very promising [2] [5]. Several numerical studies for the actuator simulation using CFD analysis were performed in order to asses the effect of their operational characteristics [6][8].

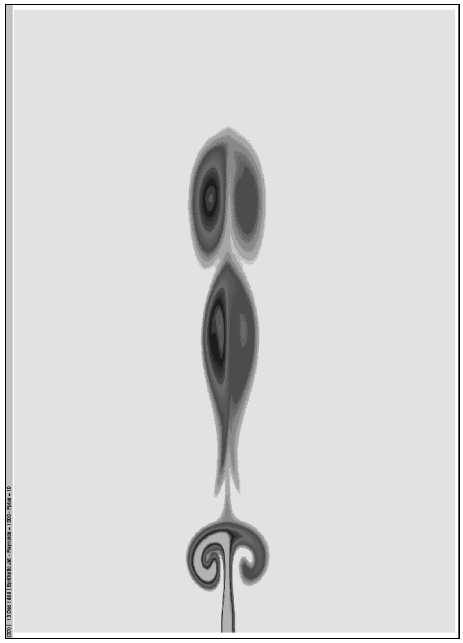

Figure 1 - SJ external flow F^{+}=10

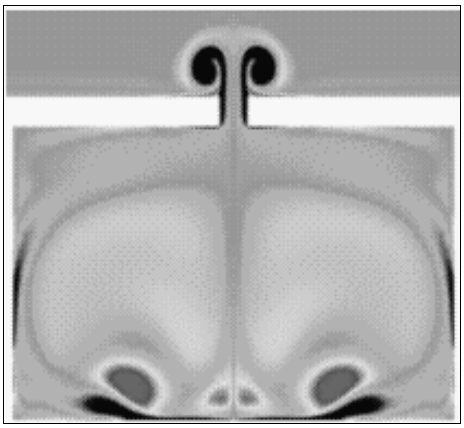

Figure 2– SJ cavity flow

From these results, for a complex analysis of their influence on a body, only the top speed and the frequency for a sinusoidal operating mode were selected in this analysis. Other characteristics of the velocity profile at the SJ exit (i.e. the influence of the external flow conditions or the geometry of the nozzle) were neglected in this phase (Figure 1, Figure 2).

Actuators are identified by the locations of the vertices on the surface and they are working in the direction given by the local normal to the surface. Such actuators are located on the surface in distinct regions. In every region, an array of actuators is operating in identical conditions. This makes an array of SJ to act like an individual actuator. If individual actuators in the same region are considered to operate at different conditions (for the top speed and frequency), then this increases the number of controls on the surface, but still uses only 2 variables for every control. Spatial definition of the velocity profile can be defined in this way for a wider actuator, using apriori given laws (7),(8) or imposing some constraints.

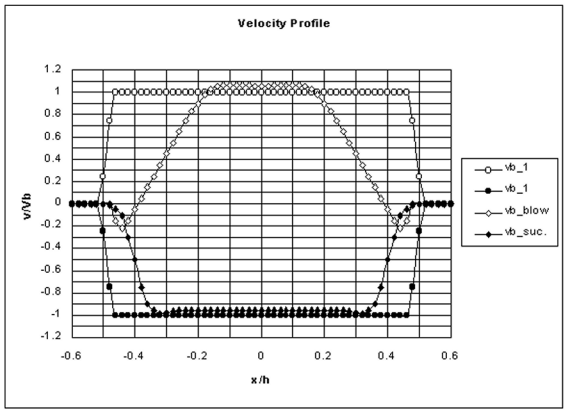

Figure 3 – SJ velocity profile

Several types of simulation were made. For the case of only external influence, the exit flow profile used was of polynomial type. The general blowing law used was :

$$v(t)=V_b(x)\sqrt{\frac{L_{ref}}{2H}}V_\infty\left[\sqrt{c_\mu}+\sqrt{\langle c_\mu\rangle}\sin\left(2\pi F^{+}\frac{V_\infty t}{L_{ref}}\right)\right] \quad (7)$$

where :

$$V_b(x)=\begin{cases} V_b^0 \\ V_b^0\cdot\sin[\pi\cdot(0.5+x)] \\ V_b^0\cdot\{\sin[\pi\cdot(0.5+x)]\}^2 \end{cases} \quad (8)$$

and global parameters given by :

$$H = \frac{1}{V_b^{0^2}} \int V_b^{\,2}(x) \cdot dx \qquad c_\mu = 2\frac{H}{L_{ref}} \left(\frac{V_b^0}{V_\infty} \right)^2 \qquad (9)$$

$$F^+ = \frac{f \cdot L_{ref}}{V_\infty} \qquad \langle c_\mu \rangle = 2\frac{H}{L_{ref}} \left(\frac{\langle v(t) \rangle}{V_\infty} \right)^2$$

In Figure 3 we present a comparison between a simple blowing/suction law as described by (7) and results obtained from a complex simulation of an actuator [17]. The results prove that a simple constant velocity profile is valid for around 50% of the nozzle in blowing and for more than 70% in suction phase. Effects like the reverse flow for the edges of the nozzle will be neglected in this phase.

2.3 The flow solver

The code used is of 2D RANS type, based on a modified k - ε turbulence model [1]. Some modifications of the initial version were made, in order to allow a better integration of SJ specific boundary conditions. The code is based on a combination of finite-volume and finite–element method, using general unstructured meshes and a choice of Roe or Osher schemes for the convective part of the system. The viscous part is solved using a typical centered finite element Galerkin technique. The global formulation used by the solver is:

$$\frac{\partial}{\partial t}(W) + \nabla \cdot F(W) = \nabla \cdot N(W) \qquad (10)$$

where :

$$W = \begin{pmatrix} \rho \\ \rho \cdot U \\ \rho \cdot V \\ \rho \cdot E \\ \rho \cdot k \\ \rho \cdot \varepsilon \end{pmatrix} \quad F_x = \begin{pmatrix} \rho \cdot U \\ \rho \cdot U^2 + p \\ \rho \cdot U \cdot V \\ (\rho \cdot E + p) \cdot U \\ \rho \cdot U \cdot k \\ \rho \cdot U \cdot \varepsilon \end{pmatrix} \quad F_y = \begin{pmatrix} \rho \cdot V \\ \rho \cdot U \cdot V \\ \rho \cdot V^2 + p \\ (\rho \cdot E + p) \cdot V \\ \rho \cdot V \cdot k \\ \rho \cdot V \cdot \varepsilon \end{pmatrix} \qquad (11)$$

and :

$$N_x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \kappa_{tot} \frac{\partial T}{\partial x} + U\tau_{xx} + V\tau_{xy} \\ (\mu + \mu_t) \cdot \frac{\partial k}{\partial x} \\ (\mu + C_\varepsilon \cdot \mu_t) \cdot \frac{\partial \varepsilon}{\partial x} \end{pmatrix} \quad N_x = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \kappa_{tot} \frac{\partial T}{\partial x} + U\tau_{xy} + V\tau_{yy} \\ (\mu + \mu_t) \cdot \frac{\partial k}{\partial y} \\ (\mu + C_\varepsilon \cdot \mu_t) \cdot \frac{\partial \varepsilon}{\partial y} \end{pmatrix} \qquad (12)$$

with :

$$\tau_{xx} = \mu_{tot} \cdot \left(2 \cdot \frac{\partial}{\partial x} U - \frac{2}{3} \cdot \nabla \cdot u \right) \qquad \nabla \cdot u = \frac{\partial}{\partial x} U + \frac{\partial}{\partial y} V \qquad (13)$$

$$\tau_{xy} = \mu_{tot} \cdot \left(\frac{\partial}{\partial y} U + \frac{\partial}{\partial x} V \right) \qquad \mu_{tot} = \mu + \mu_t$$

$$\tau_{xx} = \mu_{tot} \cdot \left(2 \cdot \frac{\partial}{\partial y} V - \frac{2}{3} \cdot \nabla \cdot u \right) \qquad \kappa_{tot} = \mu \cdot \frac{\gamma}{\mathrm{Pr}} + \mu_{tot} \cdot \frac{\gamma}{\mathrm{Pr}_t}$$

The steady state solution is obtained using an iterative time marching scheme. The algorithm is either explicit in time or implicit using a GMRES and a ILU preconditioner. For unsteady flows we will use the explicit in time formulation. It was found that a four stage Runge-Kutta scheme is the best choice for the explicit solver like:

$$\frac{\partial}{\partial t} W = RHS(W) \qquad (14)$$

and:

$$W^0 = W^n \qquad (15)$$

$$W^i = W^0 + \alpha_k \cdot \Delta t \cdot RHS\left(W^{i-1}\right) \mapsto i = 1 \ldots k$$

$$W^{n+1} = W^k$$

where α_k coefficients have been optimized for maximum accuracy and convergence speed [1].

An important feature is the time step strategy. The general formula (3), for both inviscid or viscous flows was used in order to compute the local time step at a given node . For steady state computation, a local time step strategy is commonly used. For unsteady cases, the global time step was used, as the minimum time step of all local computed time steps using the formula above. This gives up to one order of magnitude lower time step for viscous cases, so higher computational times are required [1][5].

The turbulence model used is based on the *k-ε* model. Due to the large amount of turbulent kinetic energy that is dissipated on the SJ sides edges, some important features were used in the solution approach. The grid used was designed for a $y^+ < 1$ criteria. Also, a two layer formulation was used, with a fixed distance for the low Reynolds model at $y^+ = 200$. This approach was tested and compared to the use of the wall laws technique for the low local Reynolds region. For a reasonable accurate and smooth discretisation of the sensible areas, the

two layer approach was consider to give better results than the classical wall laws model [1][5].

The boundary conditions used are based on the characteristics method for the external flow. On the cylinder, the SJ conditions were imposed for the normal velocity. Pressure and density in the SJ region was interpolated from adjacent points. Viscous conditions at the free external boundaries were fixed at a value of 10^{-5} for k and ε. Same values were used for the location of the SJ on the body.

2.4 The mesh

Due to the difficulty in obtaining reliable unstructured meshes by the Delaunay-Voronoi method (in the actuators region we request that the first grid points off the reference to be located at 10^{-5}), some local algebraic grids were used close to the surface, converted to a triangular mesh by natural triangularization. The outside region was triangulated using a Delaunay mesh generator (Figure 4). This was the initial mesh.

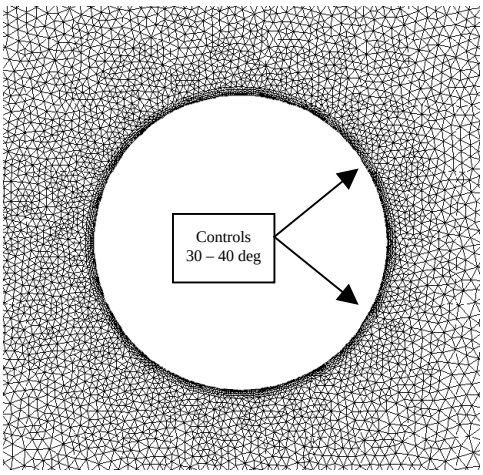

Figure 4 - Initial mesh and control location

On a such a mesh, using a first computed viscous solution for the uncontrolled case, a mesh adaptation was performed, using a mixed flow momentum and k indicator for adaptation [9]. The metric for adaptation was constructed using 4 solutions for the periodic flow. The resulting mesh (around 50.000 points, 100.000 triangles and 150.000 edges, with a smallest edge of order of 10^{-6}), was used for final viscous unsteady solution (Figure 5).

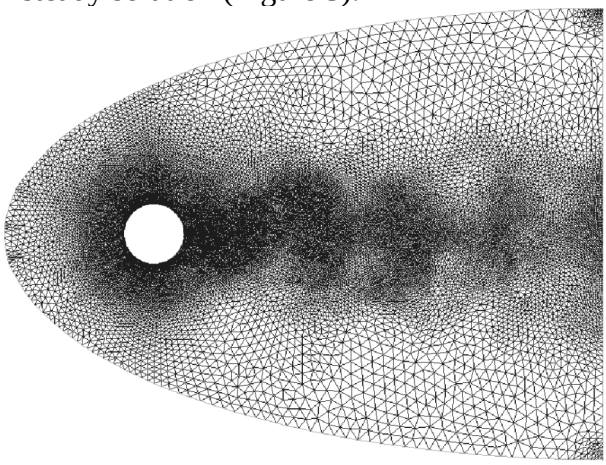

Figure 5 - Global mesh

This mesh will be used for all simulations, and initial control locations are repositioned on this new geometry. Because of the requirements of the turbulence model used, this mesh was also tested for a y+ < 1 criteria in the range of the flow solutions. A detail of the final mesh in the controls area is presented in Figure 6.

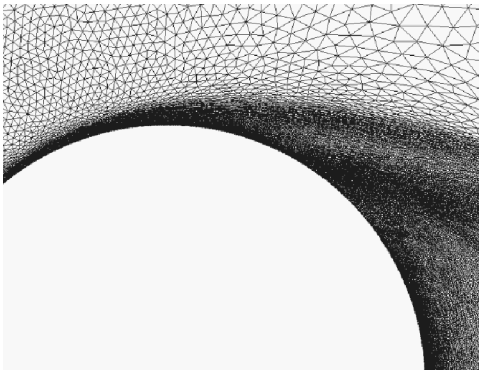

Figure 6 - Mesh detail

2.5 The Optimization Loop

The optimization algorithm used in the numerical simulations is :

1. Initial control with $(\varphi,\zeta)=(\varphi^0,\zeta^0)$
2. RANS state S_0 flow solution
3. Loop start
4. Computation of the new state S_n using the updated values for the controls;
5. Gradient computation ∇J^n ;

6. Projection step $\Pi\left(-\lambda_n \frac{dJ^n}{dF}\right)$;
7. Controls update
$\left(\varphi^n, \zeta^n\right) = \left(\varphi^{n-1}, \zeta^{n-1}\right) - \lambda_n \left[\frac{dJ^n}{dF}\right]$;
8. Time incrementation $t = t + \Delta t$
9. Loop end

We use L-BFGS-B, a limited memory algorithm for solving nonlinear optimization problems subject to simple bounds on the variables [13]. It is intended for problems in which information on the Hessian matrix is difficult to obtain[14].

3 Control Theory for Flow Control

A modern approach to solve the flow control is to formulate the problem in the framework of the mathematical theory for the control of systems governed by partial differential equations [15]. We can imagine the flow around a surface as the result of the action made by the surface that produces lift and drag by controlling the external flow. The surface design problems can now be seen as a problem in the optimal control of the flow equations using variations in the boundary conditions.

If the boundary conditions are of general type and cannot be defined by a finite number (small) of control parameters, then, for any cost function defined using information on the boundary, the Frechet derivative of the cost with respect to a function must be used. It is obvious that classical techniques using finite differences for cost derivatives with respect to control parameters are no longer possible, due to the large dimension of the problem. The control theory techniques, solving an adjoint equation with coefficients defined by the solution of the flow equations, gives the value of the gradient with a computational cost comparable to that of solving the flow equations. This means that we might expect to be able to have the gradient and the flow solution for the cost of two flow solutions, independently of the number of the control parameters.

If we define the cost function J as a function of the state flow variables W (flow parameters in the conservative form) and some physical boundary conditions, generically represented by F, then :

$$J = J\left(W(t), \nabla W, F(t)\right) \tag{16}$$

Then, for any change in F results a change for the cost function :

$$\delta J = \left[\frac{\partial J^T}{\partial W}\right]_{state} \cdot \delta W + \left[\frac{\partial J^T}{\partial F}\right]_{bound} \cdot \delta F \tag{17}$$

If we have governing flow equation which expresses the dependence of W and F inside the flow region in a generic form like (for the unsteady problem) :

$$S = S\left(W(F(t)), \frac{\partial W(F(t))}{\partial t}, \nabla W(F(t)), F(t)\right) = 0 \tag{18}$$

then we can find δW solving the following equation :

$$\delta S = \left[\frac{\delta S}{\delta W}\right]_{state} \cdot \delta W + \left[\frac{\delta S}{\delta F}\right]_{boud} \cdot \delta F = 0 \tag{19}$$

If we define a Lagrange multiplier by Λ and we multiply this equation and subtract it from the variation of δJ we finally find that if we choose Λ to satisfy the adjoint equation like :

$$\left[\frac{\partial S}{\partial W}\right]^T \cdot \Lambda = \frac{\partial J}{\partial W} \tag{20}$$

then the first term (the state dependency) is eliminated and we find that :

$$\delta J = \left[\frac{\partial J^T}{\partial F} - \Lambda^T \cdot \frac{\partial S}{\partial F}\right]_{bound} \cdot \delta F \tag{21}$$

The advantage of this equation is that it is independent of the state variables perturbation. This means that the gradient of the cost function J with respect to an arbitrary number of controls can be determined without the need of additional flow-field evaluations.

Due to the fact that (10) is a partial differential equation, the adjoint equation (20) is also a partial differential equation, so it is subjected to similar treatment for the appropriate boundary conditions which requires careful mathematical treatment [11]. In order to have numerical solutions, both the flow state

equation and the adjoint have to be discretized and solved. The control theory is applied to the set of discrete flow equations resulting from the numerical approximation using finite volume (for the Euler part) and finite element (for the viscous part) procedures. This leads directly to a set of discrete adjoint equations with a matrix which is the transpose of the Jacobian matrix of the full set of discrete nonlinear set of equations.

4 Automatic Differentiation for RANS Code

Automatic differentiation (AD) is a technique for augmenting computer programs with derivative computations. It exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations such as additions or elementary functions. By applying the chain rule of derivative calculus repeatedly to these operations, derivatives of arbitrary order can be computed .automatically, and accurate to working precision [7] .

For this work we use Odyssée, a AD tool developed at INRIA-France [7]. The automatic differentiation system Odyssée takes as input a Fortran subroutine or collection of subroutines and produces the corresponding subroutines computing the derivatives in different ways:

- Direct mode: Odyssée produces a program computing the tangent linear application (the Jacobian matrix times a vector).
- Reverse mode: Odyssée produces a program computing the cotangent linear (or adjoint) application (the transposed Jacobian matrix times a vector).

Both these modes were investigated during the present work. For large applications, the reverse mode is used as standard.

5 AD Implementation for Flow Control

The reverse mode of the AD tool is suitable for the fluid control problem with a large number(more than 6) of control variables. The resulting code is able to provide the expected results for the sensitivities with regard to the state variables. There are however some problems and conclusions resulting for several numerical tests performed with the AD tool and the optimization algorithm. They are related to the new code that is resulting from the differentiation of the state solver with respect to the state variables.

If one applies the AD tool directly, with no special precautions, the result is a huge code even in the case of a normal 2D Navier-Stokes simulation. This is because of the way the adjoint is generated where full dependencies are propagated inside the domain, even if for the particular discretisation scheme used the dependencies are limited to the neighboring cells. This makes the resulting code to be so huge, and a cure for this problem is to rewrite some of the initial code and to use AD in an incremental way under global internal loops on the vertices.

Also, from numerical results on small domains for some classical aerodynamic configurations, a well known poor dependency of the global state with respect to global flow perturbations was found [8]. This also means that any global coefficient based on a surface integral on the shape is little influenced by the flow variations induced by the controls in the whole domain. This is materialized in a week dependence of the gradient with the state [8][16]. This means that for quick aerodynamic optimizations, one may neglect the state derivative in the gradient, so only derivatives with respect to the controls will be considered. This gives a reasonable approach for a flow control problem like the one for the circular cylinder, if the controls are small. In practical applications, this assumption means that we need to implement a nested loop with a global flow solver (depending on the control variables) and an estimator for global characteristics and the cost function (based on a surface integral, including the control area). The simple approach means that we have to differentiate only the estimator, leaving the flow solver unchanged. The complex case is when we differentiate the flow solver and the estimator. Also, for a small number of controls, we can use the forward mode(i.e. 2 distinct arrays of actuators and a final 4 control parameters), or the reverse mode, for a larger number of controls. In this later

case, the effect of the above observation makes important time savings.

6 Numerical Simulations and Results

Flow control over the circular cylinder at Reynolds = 42.000 and Mach = 0.2 is analyzed using controls like synthetic jet actuators. The actuators are defined by their operating frequency and top blowing/suction speed and are located on the rear surface of the cylinder, in symmetrical regions, for an arc-length of 10 degrees, around 35 degrees location (Figure 4). They can be operated individually or as arrays. This means that the boundary conditions are expressed for a several number of control points either using the whole set of controls, or a limited subset.

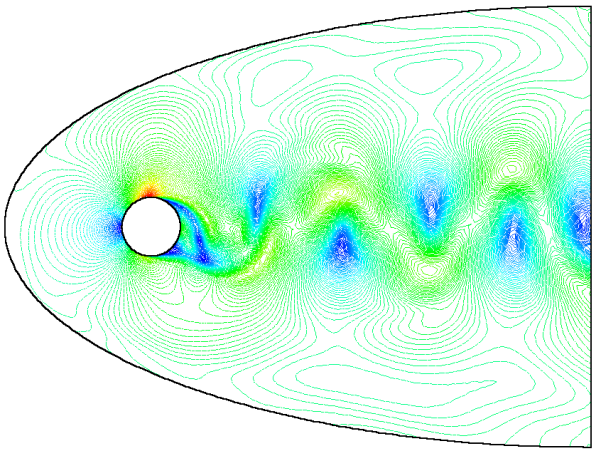

Figure 7 - Reference state (Iso-Mach lines)

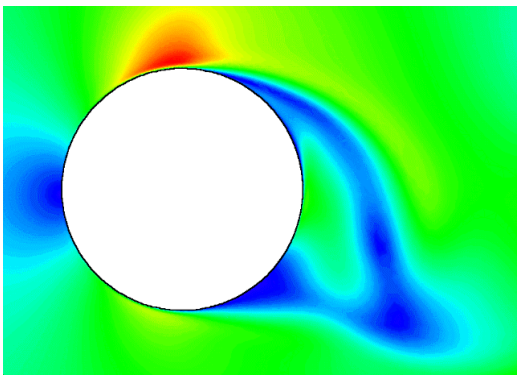

Figure 8 - Reference state detail (iso-Mach)

Operating frequencies are considered in the range of 0..2000 Hz and the maximum blowing/suction top speed is supposed to be in the range of 0..20 m/s.; nondimensionalizations were performed for the reduced frequency [2][4][6][8]. The flow induced by the actuator is supposed to have a low level of turbulence, so the same conditions for the viscous variables are considered as for free stream boundary conditions (k and ε set at 10^{-5}).

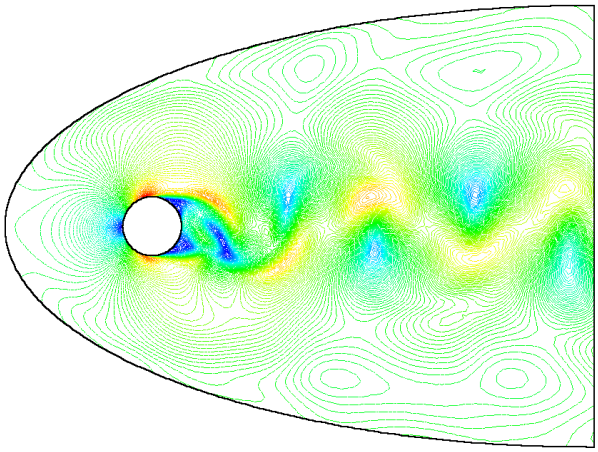

Figure 9 - Constant suction (Iso-Mach)

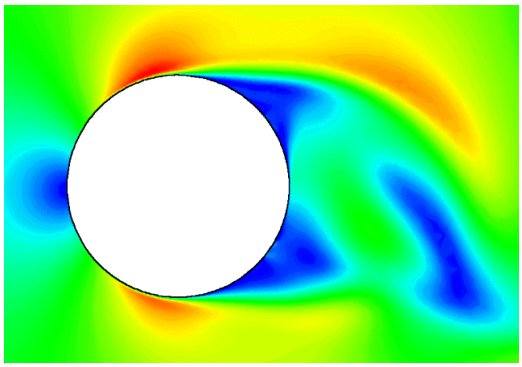

Figure 10 - Constant suction detail (Iso-Mach)

The number of points on the cylinder was 512, equally distributed, which gives a total of 28 points for the control locations, to be identified as SJ boundary conditions. So the maximum number of control parameters was 56. In the case when only arrays of SJ were considered, the minimum geometry gives 2 arrays with a total minimum controls of 4. In the case of individual actuators, only simple sinusoidal individual oscillations can be considered. For lower identification situations, when we use individual groups, complex velocity profiles may be considered as resulting from (7) or actuator simulations [8][2].

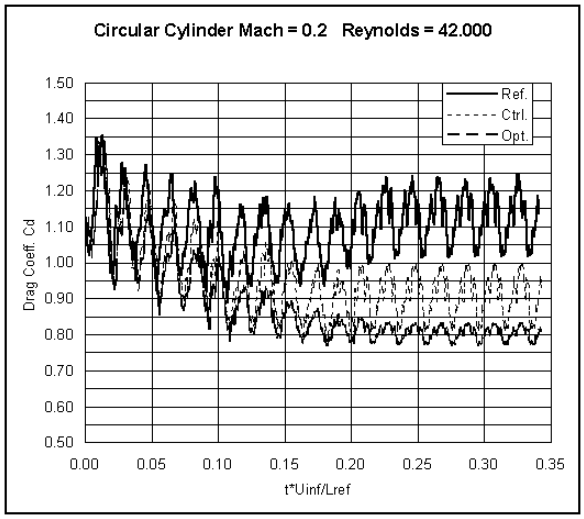

Figure 11 - Cost function history - 4 variables

All simulations were started from a reference state, the solution for the flow with no controls (Figure 7, Figure 8).

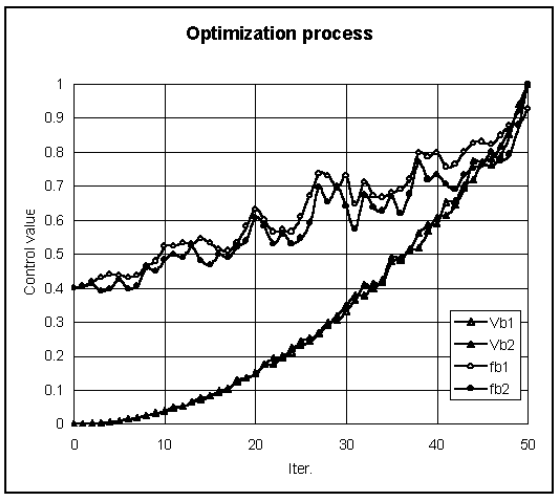

Figure 12 - Optimization process – 4 variables

From this solution, using the optimization algorithm, a minimization of drag was performed in a maximum imposed number of 50 loops. The number of cvasiperiods used for unsteady cost function averaging was 4.

An alternative solution to the problem may be considered using a different starting point, for a constant suction case (Figure 9, Figure 10). This case is using a constant suction velocity *Vb* = 0.1 for all controls located on both upper and lower area. This case will be referred as the controlled case using an apriori given law.

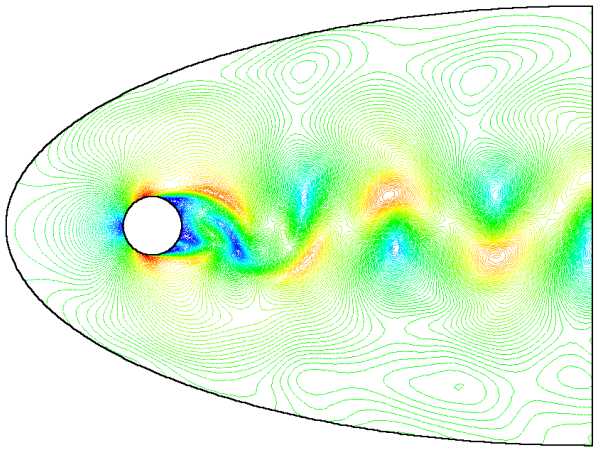

Figure 13 - Optimal state (Iso-Mach)

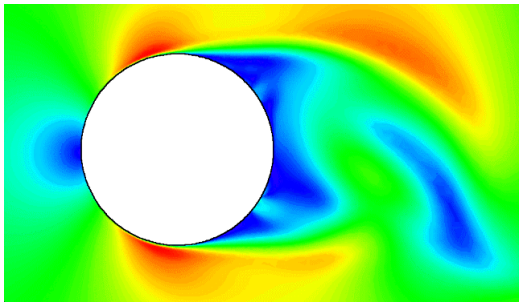

Figure 14 - Optimal state detail (Iso-Mach)

All simulations were considered up to a minimization of the residual in the RANS code on the order of 10^{-5}. For the gradient computations, a sensitivity of 10^{-3} was considered, for a global machine precision of 10^{-16}. An explicit integration of the solver was used based on global time step strategy and a 4 order RK scheme. The average time step resulting from the code was 1.45×10^{-6} .

The optimization process is presented for the cost function history Figure 11. The figure presents time history for the reference state (no controls), a controlled state using an apriori given suction law and the optimal controlled state. The optimization algorithm for the optimal case is presented in Figure 12.

The final optimal state for the 4 variable control case is presented in Figure 13, Figure 14. A comparison between 3 instantaneous states for the pressure coefficient *Cp* on the cylinder are presented in Figure 15.

7 Conclusions

A first conclusion of the simulations was that it is possible to ignore the gradient dependence to the derivative with regard to the state. This is an important finding which makes a big difference in computational time. This conclusion has to be validated in further studies.

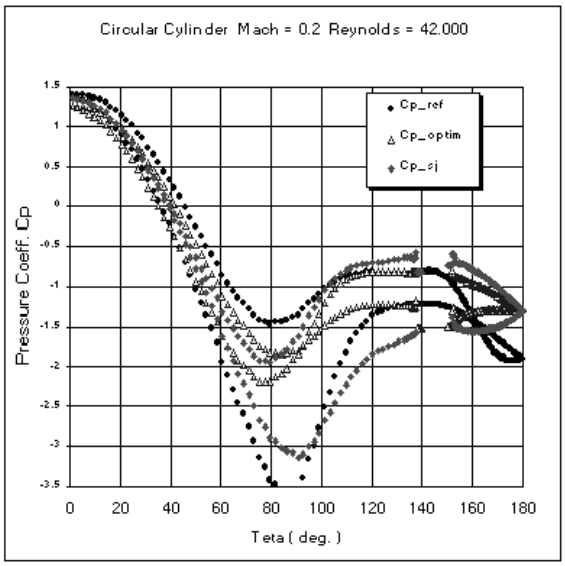

Figure 15 - Cp for instantaneous states

The algorithm is providing optimal control values for a state which is not dependent of the starting point. From the numerical results, it seams that the controls have very little different values for the resulting state, for the two starting points considered. This finding is in good agreement with the previous conclusion (independence of the state derivative for the gradient).

The reference value for the drag was Cd = 1.12 with a fluctuation of ΔCd = 0.12 . From the optimum obtained , we have a Cd = 0.81 and a fluctuation of ΔCd = 0.05 , for a symmetrical actuators control law using the top allowed blowing speed of 20 m/s and a frequency of 1272 Hz.

From these first experiments, it seams that the higher the top actuator speed is, the better the stabilization effect is obtained. Frequency is somehow to the middle allowed domain, and further analysis is indicated in order to validate these results. The control algorithm gives an optimum control law which is symmetrical, possibly because of the geometry symmetry.

Also, from the final results, it seems that flow control is possible using the SJ actuators and an gradient based optimization algorithm for the cost function drag. This finding has to be investigated in more details for complex configurations.

References

[1] Nae C. *Osher solver for unstructured grids.* INCAS report C-2070, 1998.

[2] Rizzetta D P, Visbal M R, Stanek M J. *Numerical investigation of synthetic jet flowfields.* AIAA Paper 98-2910.

[3] Amitay M, Smith B L, Glezer A. *Aerodynamic flow control using synthetic jet technology.* AIAA Paper 98-0208.

[4] Seifert A, Pack L G. *Oscillatory excitation of compressible flows over airfoils at flight reynolds numbers*. AIAA Paper 99-0925.

[5] Nae C. *Synthetic jets influence on NACA 0012 airfoil at high angles of attack.* AIAA Paper 98-4523.

[6] Kral L D, Guo D. *Characterization of jet actuators for active flow control*. AIAA Paper 99-3573.

[7] Faure C, Papegay Y. *Odyssee version 1.6*. Technical Report RT-0211, INRIA, 1997.

[8] Nae C. *Unsteady flow control using synthetic jet actuators.* AIAA Paper 2000-2403.

[9] Nae C. *Flow solver and anisotropic mesh adaptation using a change of metric based on flow variables.* AIAA Paper 2000-2250.

[10] Laporte E, Le Tallec P. *Shape optimisation in unsteady flows*. RR-3693, INRIA, 1999.

[11] Jameson A. *Aerodynamic design via control theory*. J. Sci. Comp., 3:233-260, 1988.

[12] Mohammadi B. *Practical applications to fluid flows of automatic differentiation for design problems.* VKI Lecture Series, 1997.

[13] Byrd R H, Lu P, Nocedal J, Zhu C. *A limited memory algorithm for bound constraind optimization*. Tech. Report EECS, Northwestern Univ., 1993.

[14] More J J., Toraldo G. *Algorithms for bound constraind quadratic programing problems*. Numer. Math. 55, pp. 377-400, 1989.

[15] Lions J L. *Optimal control of systems governed by partial differential equations*. Springer-Verlag, New York, 1971.

[16] Mohammadi B. *Flow control and shape optimization in aeroelastic configurations*. AIAA Paper 99-0182.

[17] Nae C. *Numerical simulation of a synthetic jet actuator.* ICA 0266, ICAS 2000.