FLOW CONTROL USING AUTOMATIC DIFFERENTIATION AND SYNTHETIC JETS

CĂTĂLIN NAE

Flow control over the circular cylinder at Reynolds 42.000 and Mach = 0.2 is analyzed using controls like synthetic jet actuators. Such actuators are defined by their operating frequency and top blowing/suction speed and are located on the rear surface of the cylinder and they can be operated individually or as arrays. The analysis is based on CFD techniques and control theory. Taking drag as a cost function, minimization of drag oscillations is achieved using an optimal control algorithm where automatic differentiation of the RANS solver is used in order to generate the gradient. Results are presented for the numerical simulation of the actuator and the flow around the cylinder using active flow control.

1. INTRODUCTION

Viscous flow over a circular cylinder is well known to produce a Karman vortex alley and strong oscillations both in lift and drag. Controlling this flow configuration is an interesting case, and many attempts have been made in this direction. Using a novel technique, based on synthetic jets actuators, this flow is numerically investigated as a promising practical solution for active flow control.

Synthetic jets result from an oscillating diaphragm in an enclosed space, having small orifices at the top. Several such devices may be considered in the form of an array acting independently or as a single unit, and being incorporated in various active devices for fluid control. 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]. Using such devices seams to represent an alternative to the traditional mechanical control systems, having also some important advantages. Because their effect is based on a zero mass transfer, they do not need dedicated air supply systems as for other blowing/suction devices, thus making them suitable for a large class of

National Institute for Aerospace Research "Elie Carafoli", B-dul Iuliu Maniu no. 220, sect. 6, code 7738, Bucharest, Romania.

Rev. Roum. Sci. Techn. – Méc. Appl., Tome 52, N°.3, P. 157–172, Bucarest, 2007

applications. Such actuators were previously investigated for flow control on various configurations and the results are very promising [2, 3, 4, 5].

The numerical simulations in this paper are based on CFD techniques using a RANS 2D code with a modified k-ε turbulence model [1]. Flow control is numerically investigated for drag, which is considered as a cost function to be minimized using control theory and a gradient based algorithm. Gradients will be generated using automatic differentiation [7] of the solver using several assumption made on the dependence of the cost function on the state (flow variables).

2. THE UNSTEADY FLOW CONTROL PROBLEM

We 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. A reference state is computed using the solver for the flow with no controls. From this state controls are activated and the optimization process begins. 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 assumptions will be to minimize the drag coefficient C_d using a set of controls on the boundary:

$$C(\varphi,\zeta)=\min_{\varphi,\zeta} J(\varphi,\zeta)=\frac{1}{n_T}\int_0^{n_T\cdot T} C_d\left(\varphi(t),\zeta(t)\right)\cdot \mathrm{d}t, \tag{1}$$

where:

$$(\varphi,\zeta)\in\left\{R^m\times R^m\times[0,T]\,|\,0\le\varphi\le\varphi_{\max},0\le\zeta\le\zeta_{\max}\right\}. \tag{2}$$

Here T is the time period of the cvasiperiodic flow solution for the case with no controls, and C_d 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 [1] is using a global time step integration technique 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

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

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):

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

or

$$C^n(\varphi,\zeta) = C^{n+1}(\varphi,\zeta) + \Delta t\cdot\min_{\varphi^n,\zeta^n} Cd\left(\varphi^n,\zeta^n\right) \text{ for } n\in[1,N], \tag{5}$$

$$N = n_T\cdot\left(1+\mathrm{int}\left(\frac{T}{\Delta t}\right)\right).$$

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

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

2.2. THE CONTROLLING DEVICES

For the cylinder problem, we use controls for the minimization of the drag. The controls used are SJ devices that have individual characteristics for the frequency and velocity profile (Fig. 2). Several numerical studies for the actuator simulation using CFD analysis were performed in order to asses the effect of their

operational characteristics [2, 6, 8]. 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. Other characteristics for the velocity profile at the SJ exit (i.e. the influence of the external flows conditions or the geometry of the nozzle) were neglected in this phase (Fig. 1).

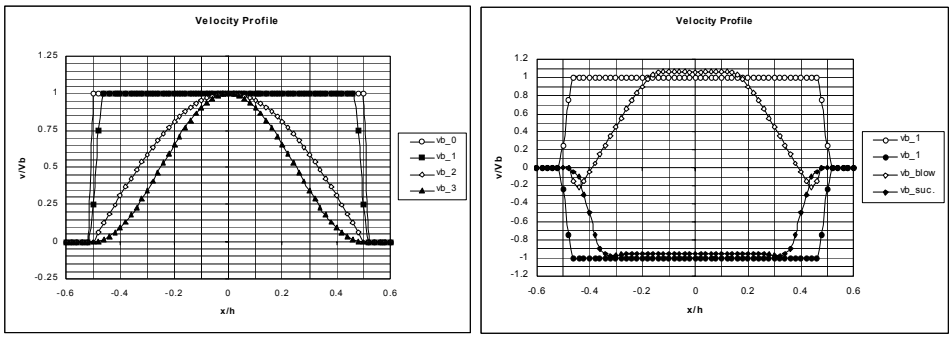

Fig. 1 – SJ exit velocity profiles.

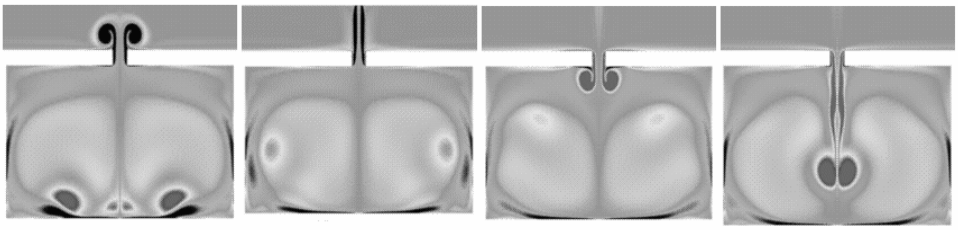

Fig. 2 – SJ cavity simulation – 1 cycle (vorticity contours).

Actuators are identified by the locations of the vertices on the surface. Because of the previous assumption, the minimal distance between two independent controls (actuators) is at least the distance between two adjacent points. 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.

Several types of simulation were made. For the case of only external influence, the exit flow profile used was of polynomial type (Fig. 1, Fig. 2). One of the blowing law used is:

$$v(t) = V_b(x) \cdot \sqrt{\frac{L_{ref}}{2H}} \cdot V_\infty \cdot \left[\sqrt{c_\mu} + \sqrt{\langle c_\mu \rangle} \cdot \sin\left(2\pi \cdot F^+ \frac{V_\infty t}{L_{ref}} \right) \right], \tag{6}$$

where:

$$V_b(x) = \begin{cases} V_b^0 \\ V_b^0 \cdot \sin[\pi \cdot (0.5 + x)] \\ V_b^0 \cdot \{\sin[\pi \cdot (0.5 + x)]\}^2 \end{cases} \tag{7}$$

and:

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

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

All computations were started from an initial viscous solution, corresponding to the no-blowing case. Then, using blowing/suction conditions (6), an oscillating converged solution was obtained, generally after 20 cycles of blowing/suction. Present results are well confirmed by experiments [3, 4, 6, 8]. This approach will be used for complex analysis of actuators interactions and complex control geometry.

2.3. THE FLOW SOLVER

The code used is of 2D RANS type, based on a modified *k*-ε turbulence model [1]. For this study, some modifications of the initial version were made. 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 connective part of the system. The viscous part is solved using a typical centered finite element Galerkin technique.

Due to the difficulty in obtaining reliable unstructured meshes by the Delaunay-Voronoi method (for blowing cases, the first grid points off the reference area were located at $1.0e^{-05}$), 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. On such a mesh, using a first computed viscous solution, a mesh adaptation was performed [9], using a flow

momentum indicator for adaptation. The resulting mesh (about 100,000 triangles), was used for final viscous unsteady solution (Fig. 3, Fig. 4).

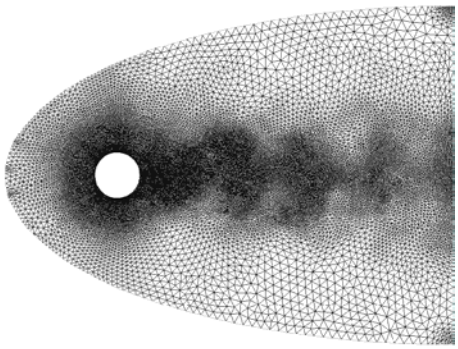

Fig. 3 – Mesh.

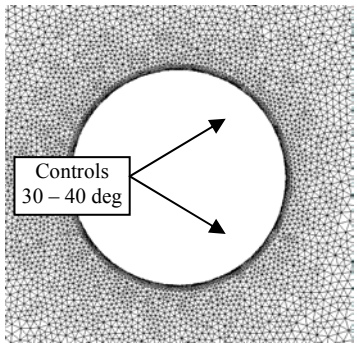

Fig. 4 – Mesh detail.

The global formulation used is:

$$\frac{\partial}{\partial t}(W)+\nabla\cdot F(W)=\nabla\cdot N(W), \tag{8}$$

where:

$$W=\begin{pmatrix}\rho\\ \rho\cdot U\\ \rho\cdot V\\ \rho\cdot E\\ \rho\cdot k\\ \rho\cdot\varepsilon\end{pmatrix}\quad F_x(W)=\begin{pmatrix}\rho\cdot U\\ \rho\cdot U^2+p\\ \rho\cdot U\cdot V\\ (\rho\cdot E+p)\cdot U\\ \rho\cdot U\cdot k\\ \rho\cdot U\cdot\varepsilon\end{pmatrix}\quad F_y(W)=\begin{pmatrix}\rho\cdot V\\ \rho\cdot U\cdot V\\ \rho\cdot V^2+p\\ (\rho\cdot E+p)\cdot V\\ \rho\cdot V\cdot k\\ \rho\cdot V\cdot\varepsilon\end{pmatrix}, \tag{9}$$

$$N_x(W)=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \kappa_{tot}\cdot\dfrac{\partial}{\partial x}T+U\cdot\tau_{xx}+V\cdot\tau_{xy}\\ (\mu+\mu_t)\cdot\dfrac{\partial}{\partial x}k\\ (\mu+C_\varepsilon\cdot\mu_t)\cdot\dfrac{\partial}{\partial x}\varepsilon\end{pmatrix}\quad N_x(W)=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \kappa_{tot}\cdot\dfrac{\partial}{\partial x}T+U\cdot\tau_{xy}+V\cdot\tau_{yy}\\ (\mu+\mu_t)\cdot\dfrac{\partial}{\partial y}k\\ (\mu+C_\varepsilon\cdot\mu_t)\cdot\dfrac{\partial}{\partial y}\varepsilon\end{pmatrix}, \tag{10}$$

and:

$$\begin{aligned}
\tau_{xx} &= \mu_{tot} \cdot \left(2 \cdot \frac{\partial}{\partial x} U - \frac{2}{3} \cdot \nabla \cdot u \right) \\
\tau_{xy} &= \mu_{tot} \cdot \left(\frac{\partial}{\partial y} U + \frac{\partial}{\partial x} V \right) \\
\tau_{xx} &= \mu_{tot} \cdot \left(2 \cdot \frac{\partial}{\partial y} V - \frac{2}{3} \cdot \nabla \cdot u \right)
\end{aligned} \quad \text{and} \quad \begin{aligned}
\nabla \cdot u &= \frac{\partial}{\partial x} U + \frac{\partial}{\partial y} V \\
\mu_{tot} &= \mu + \mu_t \\
\kappa_{tot} &= \mu \cdot \frac{\gamma}{\Pr} + \mu_{tot} \cdot \frac{\gamma}{\Pr_t} .
\end{aligned} \tag{11}$$

The steady state solution is obtained using an iterative 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 formulation. It was found that a four stage Runge-Kutta scheme is the best choice for the explicit solver:

$$\frac{\partial}{\partial t} W = RHS(W) \quad \text{and} \quad \begin{aligned}
W^0 &= W^n \\
W^i &= W^0 + \alpha_k \cdot \Delta t \cdot RHS\left(W^{i-1}\right) \mapsto i = 1 \ldots k , \\
W^{n+1} &= W^k
\end{aligned} \tag{12}$$

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

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 one order of magnitude lower time step, so higher computational times are required.

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 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 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].

2.4. THE OPTIMIZATION LOOP

The optimization algorithm used in the numerical simulations is:

1. Initial control with $(\varphi, \zeta) = \left(\varphi^0, \zeta^0\right)$
2. RANS state S_0 flow solution
3. Loop start

- Computation of the new state S_n using the updated values for the controls;
- Gradient computation ∇J^n ;
- Projection step $\Pi\left(-\lambda_n \frac{dJ^n}{dF}\right)$;
- Controls update $\left(\varphi^n, \zeta^n\right) = \left(\varphi^{n-1}, \zeta^{n-1}\right) - \lambda_n \left[\frac{dJ^n}{dF}\right]$;
- Time incrementation $t = t + \Delta t$

4. Loop end

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

At each iteration, a limited memory BFGS approximation to the Hessian is updated. This limited memory matrix is then used to define a quadratic model of the objective function. A search direction is then computed using a two-stage approach. First, the gradient projection method [14] is used to identify a set of active variables. Then the quadratic model is approximately minimized with respect to the free variables. The search direction is defined to be the vector leading from the current iterate to this approximate minimizer. Finally a line search is performed along the search direction.

The advantages of using L-BFGS-B are related to the low computational costs and storage requirements and the fact that the algorithm is independent of the properties of the objective function.

3. APPLICATION OF CONTROL THEORY

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, generally represented by F, then:

$$J = J\left(W(t), \nabla W, F(t)\right). \tag{13}$$

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

$$\delta J = \left[\frac{\partial J^T}{\partial W}\right]_{state} \cdot \delta W + \left[\frac{\partial J^T}{\partial F}\right]_{bound} \cdot \delta F. \tag{14}$$

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

$$S = S[W, \nabla W, F] = S[W(F), \nabla W(F), F] = 0 \tag{15}$$

or for the unsteady problems :

$$S = S\left[W, \frac{\partial W}{\partial t}, \nabla W, F\right] = S\left[W(F(t)), \frac{\partial W(F(t))}{\partial t}, \nabla W(F(t)), F(t)\right] = 0. \tag{16}$$

Then we can find δW solving the following equation:

$$\delta S = \left[\frac{\delta S}{\delta W}\right]_{state} \cdot \delta W + \left[\frac{\delta S}{\delta F}\right]_{boud} \cdot \delta F = 0. \tag{17}$$

If we define a Lagrange multiplier by Λ and we multiply this equation and subtract it from the variation of δJ we find:

$$\delta J = \left[\frac{\partial J^T}{\partial W}\right]_{state} \cdot \delta W + \left[\frac{\partial J^T}{\partial F}\right]_{bound} \cdot \delta F - \Lambda^T \cdot \left\{\left[\frac{\partial S}{\partial W}\right]_{state} \cdot \delta W + \left[\frac{\partial S}{\partial F}\right]_{bound} \cdot \delta F\right\} \tag{18}$$

and finally :

$$\delta J = \left[\frac{\partial J^T}{\partial W} - \Lambda^T \cdot \frac{\partial S}{\partial W}\right]_{state} \cdot \delta W + \left[\frac{\partial J^T}{\partial F} - \Lambda^T \cdot \frac{\partial S}{\partial F}\right]_{bound} \cdot \delta F. \tag{19}$$

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 (15) 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.

The discrete adjoint equations derived directly from the discrete flow equations become very complicated when the flow equations are discretized with higher order upwind biased schemes using flux limiters. If they are solved exactly, they can provide an exact gradient of an inexact cost function which results from the discretisation of the flow equations. Because they are linear, they can be solved by direct numerical inversion. The cost of an direct inversion became prohibitive as the mesh is refined, so iterative methods are more efficient. Because the similarity of the adjoint equations and the flow equations, similar iterative methods are generally used.

4. APPLICATION OF AUTOMATIC DIFFERENTIATION

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, 12].

Traditionally, two approaches to automatic differentiation have been developed: the so-called forward and reverse modes. These modes are

distinguished by how the chain rule is used to propagate derivatives through the computation.

The forward mode propagates derivatives of intermediate variables with respect to the independent variables. Given independent variables x, say, and dependent variables y, say, the linearity of differentiation allows the forward mode to compute arbitrary linear combinations $J \cdot S$ of columns of the Jacobian. For an $n \bullet p$ matrix S, the effort required is roughly $O(p)$ times the runtime and memory of the original program. In particular, when S is a vector s, we compute the directional derivative for $J \cdot s = \lim_{h \to 0} \frac{f(x + h \cdot s) - f(x)}{h}$. This type of differentiation is also used to obtain a so-called tangent linear model.

In contrast, the reverse mode of automatic differentiation propagates derivatives of the final result with respect to an intermediate quantity, adjoint quantity. To propagate adjoints, one must be able to reverse the flow of the program and must remember or recompute any intermediate value that nonlinearly impacts the final result. For a $q \cdot m$ matrix W, the reverse mode allows us to compute the row linear combination $W \cdot J$ with $O(q)$ times as many floating-point operations as required for the evaluation of f. The storage requirements are harder to predict and depend to a large extent on the nonlinearity of the program and the implementation approach chosen. The reverse mode also corresponds to a method for obtaining the discrete adjoint.

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).

5. SIMPLIFICATIONS

The reverse mode of the AD tool is suitable for the fluid control problem. 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 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 [8], a well known poor dependency of the global state with respect to global flow perturbations was found. 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. These findings have to be supported by more numerical tests, but these first simulations are giving this conclusion.

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, at 45 degrees (Fig. 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.

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; adimensionalizations 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 (*i.e.* 10^{-5}).

The number of points on the cylinder was 512, equally distributed, which gives a total of 28 point for 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 control of 4. In the case of individual actuators, only simple sinusoidal individual oscillations can be considered. For lower identification

situations, for individual groups, complex velocity profiles (Fig. 2) may be considered as resulting from (6) or actuator simulations [8, 2].

All simulations were started from a reference state, the solution for the flow with no controls (Fig. 5, Fig. 6). From this solution, using the optimization algorithm, a minimization of drag was performed in a maximum number of 50 loops. The number of cvasiperiods used was 4. An alternative solution to the problem may be considered using a different starting point, for a constant suction case (Fig. 7, Fig. 8). All simulations were considered up to a minimization of the residual in the RANS code on the order of 10^{-4}. For the gradient computations, a sensitivity of 10^{-2} was considered. An explicit integration of the solver was used based on global time step strategy and a 4 order RK scheme.

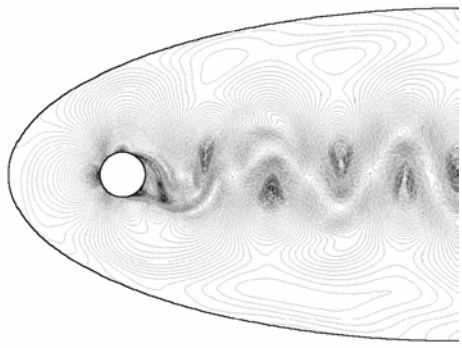

Fig. 5 – Reference state (Iso-Mach).

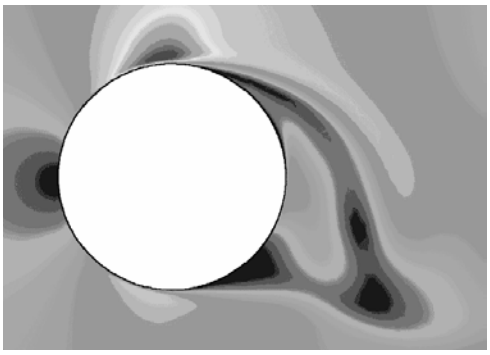

Fig. 6 – Reference state (Iso-Mach).

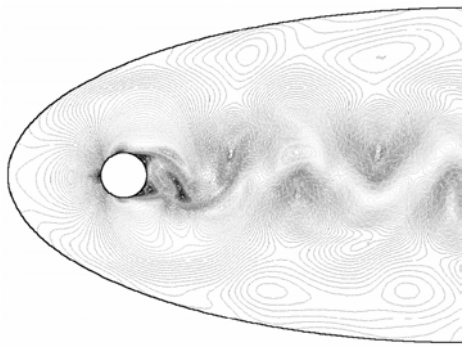

Fig. 7 – Constant suction Vb = –02.

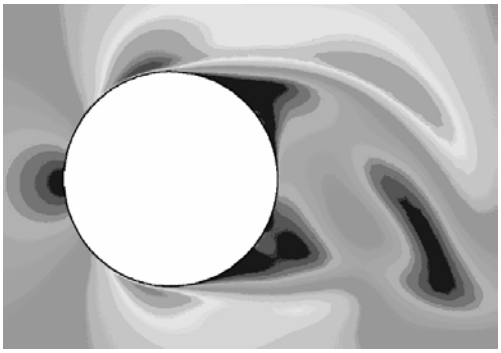

Fig. 8 – Constant top suction (Iso – Mach).

7. CONCLUDING REMARKS

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.

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 C_d = 1.12 with a fluctuation of ΔC_d = 0.12. From the optimum obtained, we have a C_d = 0.81 it a fluctuation of ΔC_d = = 0.05, for a symmetrical actuators control law using the top allowed blowing speed of 20 m/s and a frequency of 1272 Hz (Fig. 11, Fig. 12). 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 (Fig. 9, Fig. 10).

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

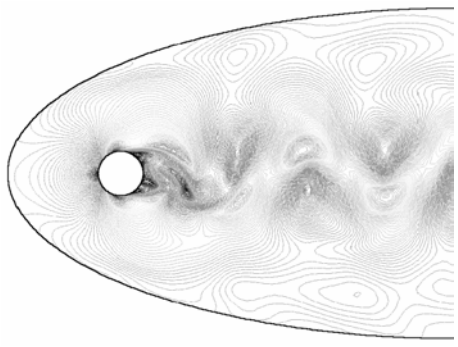

Fig. 9 – Optimal control.

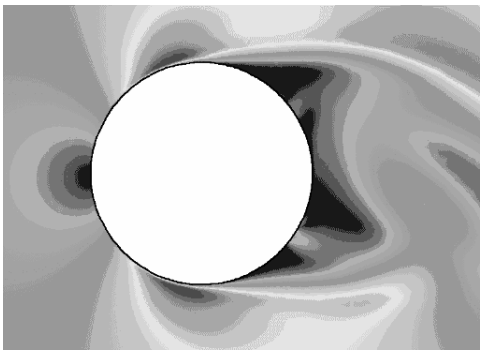

Fig. 10 – Optimal sate (Iso – Mach).

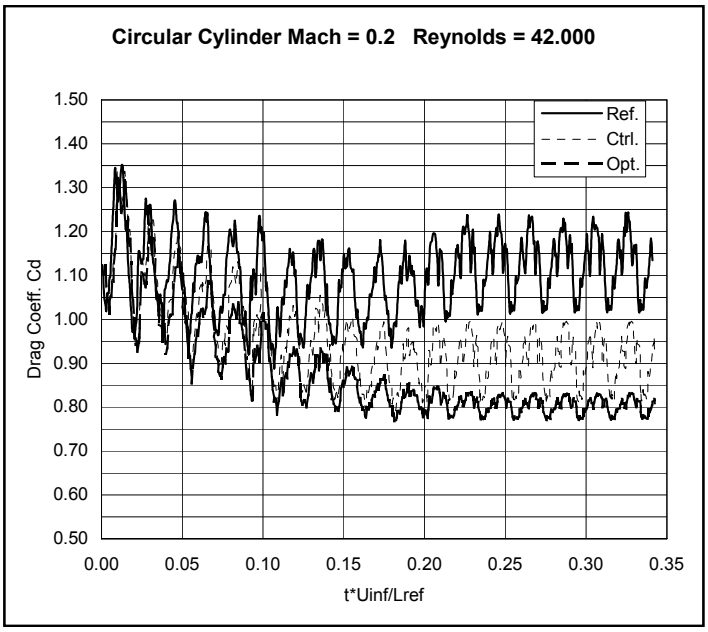

Fig. 11 – Drag oscillations.

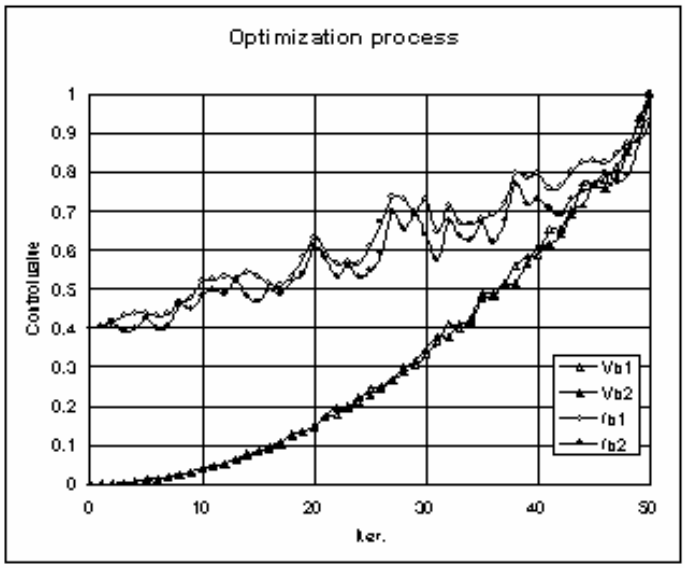

Fig. 12 – Optimization process.

Received 28 March 2000

REFERENCES

1. C. NAE, *Osher solver for unstructured grids*, INCAS report C-2070, 1998.
2. D.P. RIZZETTA, M.R. VISBAL, M.J. STANEK, *Numerical investigation of synthetic jet flowfields*, AIAA Paper 98-2910.
3. M. AMITAY, B.L. SMITH, A. GLEZER, *Aerodynamic flow control using synthetic jet technology*, AIAA Paper 98-0208.
4. A. SEIFERT, L.G. PACK, *Oscillatory excitation of compressible flows over airfoils at flight Reynolds numbers*, AIAA Paper 99-0925.
5. C. NAE, *Synthetic jets influence on NACA 0012 airfoil at high angles of attack*, AIAA Paper 98-4523.
6. L.D. KRAL, D. GUO, *Characterization of jet actuators for active flow control*, AIAA Paper 99-3573.
7. C. FAURE, Y. PAPEGAY, *Odyssee version 1.6 – the user's reference manual*, Technical Report RT-0211, INRIA Rocquencourt, 1997.
8. C. NAE, *Unsteady flow control using synthetic jet actuators*, AIAA Paper 2000-2402.
9. C. NAE, *Flow solver and anisotropic mesh adaptation using a change of metric based on flow variables*, AIAA Paper 2000-2250.
10. E. LAPORTE, P.LE TALLEC, *Shape optimisation in unsteady flows*, Raport de recherche RR-3693, INRIA Rocquencourt, 1999.
11. A. JAMESON, *Aerodynamic design via control theory*, J. Sci. Comp., **3**, pp. 233–260 (1988).
12. B. MOHAMMADI , *Practical applications to fluid flows of automatic differentiation for design problems*, VKI Lecture Series, 1997.
13. R.H. BYRD, P.LU, J. NOCEDAL, C. ZHU, *A limited memory algorithm for bound constraind optimization*, Tech. Report EECS, Northwestern Univ., 1993.
14. J.J. MORE, G. TORALDO, *Algorithms for bound constraind quadratic programing problems*, Numer. Math., **55**, pp. 377–400 (1989).
15. J.L. LIONS, *Optimal control of systems governed by partial differential equations*, Springer-Verlag, New York, 1971.
16. B. MOHAMMADI, *Flow control and shape optimization in aeroelastic configurations*, AIAA Paper 99-0182.