Aerodynamic Shape Optimization Using the Adjoint-based Truncated Newton Method

Mehdi Ghavami Nejad*
Parallel CFD & Optimization Unit, School of Mechanical Engineering, National Technical University of Athens, Greece
Email: mehdi@mail.ntua.gr

Evangelos M. Papoutsis-Kiachagias
Parallel CFD & Optimization Unit, School of Mechanical Engineering, National Technical University of Athens, Greece
Email: vaggelisp@gmail.com

Kyriakos C. Giannakoglou
Parallel CFD & Optimization Unit, School of Mechanical Engineering, National Technical University of Athens, Greece
Email: kgianna@central.ntua.gr

Keywords: Aerodynamic Shape Optimization, Continuous Adjoint, Truncated Newton Method.

Abstract

This paper presents the development and application of the truncated Newton (TN) method for shape optimization problems. The development is made for problems governed by laminar flows of incompressible fluids though its extension to turbulent flow is straightforward. The method was developed in OpenFOAM® with the aim to stress the advantages over standard gradient-based optimization algorithms. The Newton equations are solved using the conjugate gradient (CG) method which requires the computation of the product of the Hessian of the objective function with a user-defined number of vectors, escaping thus the need for computing the Hessian itself. The latter has a computational cost that scales with the number of design variables and may become unaffordable in large-scale problems with many design variables. A combination of the continuous adjoint method and direct differentiation is used to compute all Hessian-vector products. The programmed method is used to optimize the sidewall shapes of a divergent duct for minimum total pressure losses and the shape of a 2D airfoil for minimum drag.

Introduction

In aerodynamic shape optimization, the adjoint approach for the computation of the objective function gradient w.r.t the N design variables has a cost which is independent of the value of N and, for this reason, is in widespread use[1, 2, 3]. On the other hand, Newton optimization algorithms could lead to faster convergence (at least in terms of optimization cycles) than conventional optimization methods exclusively relying upon the gradient. However the cost for computing the required Hessian matrix, with the best available method[4], scales with the number of design parameters. This is the main reason for which the application of (exact) Newton methods is restricted to problems with just a few design variables. A viable alternative is the "exactly initialized quasi-Newton method" according to which the exact Hessian is computed only in the first optimization cycle and is then approximately updated, as proposed by some of the authors[5]. Though this approach may become much more efficient than the Newton method, the computation of the Hessian, even once, could be prohibitive in problems with many design variables. For this kind of problems, the TN method could be used instead. In the TN method, the Newton system of equations is solved iteratively through the CG algorithm requiring only the computation of Hessian-vector products. An approach for computing these products, based on the combination of the continuous adjoint method and direct differentiation, is presented herein, for use in shape optimization problems. This is an extension of a previous work[6] solving topology optimization problems using the TN method which proved to be very efficient. In the full paper, two cases regarding a divergent duct and an isolated airfoil will be presented. For the needs of these cases, two objective functions (volume-averaged total pressure losses and drag) are worked out. Some preliminary results are included in this extended abstract.

The TN Method

Below, the TN method is developed for a general objective function F. The Newton method accelerates the convergence of the optimization algorithm in which the design variables $b_i (i = 1, \dots, N)$ are updated by solving the Newton's system:

$$\frac{\delta^2 F}{\delta b_i \delta b_j} \Delta b_j = -\frac{\delta F}{\delta b_i} \quad (1)$$

$$b_j^{new} = b_j^{old} + \Delta b_j \quad j = 1, \dots, N$$

While the r.h.s of equation (1) can be computed via the adjoint method (since this is nothing more than the gradient of F), the Newton method requires the computation of the Hessian which, at the best case scenario, has a computational cost that scales with the number of the design parameters (N); this, in turn, makes it too expensive for large scale optimization problems. The TN method relies upon the solution of equation (1), which resembles the classic linear system of equations $\mathbf{Ax} = \mathbf{q}$, using the CG method.

Though this paper relies exclusively upon the CG method, any other iterative algorithm that requires only Hessian-vector products can be embodied in the TN algorithm. The CG algorithm for solving the system of linear equations $\mathbf{Ax} = \mathbf{q}$ is schematically given below:

$k \longleftarrow 0$
$\mathbf{x} \longleftarrow$ init()
$r_n{}^0 \longleftarrow A_{nl} x_l{}^0 - q_n{}^0; \; s_n \longleftarrow -r_n{}^0$
while $r^k \neq 0$ and $k \leq M_{CG}$ do
$w_n \longleftarrow A_{nl} s_l$
$\eta \longleftarrow \frac{(r_n{}^k)^T r_n{}^k}{s_k{}^T w_k}$
$x_l \longleftarrow x_l + \eta s_l$
$r_l{}^{k+1} \longleftarrow r_l{}^k - \eta w_l$
$\beta \longleftarrow \frac{(r_n{}^{k+1})^T r_n{}^{k+1}}{(r_k{}^k)^T r_k{}^k}$
$s_l \longleftarrow r_l{}^{k+1} + \beta s_l$
$k \longleftarrow k + 1$
end while

In the above algorithm, k is the iteration counter, **r** denotes the residual vector, η is the step and M_{CG} is the number of CG cycles which is user-defined and should be much smaller than the number of unknowns. In the above CG algorithm, each iteration costs, practically, as much as the computation of the product of **A** and **s**.

In CFD-based optimization, inspired by the previous algorithm, one optimization loop which employs the TN method comprises the following main steps:

1- Solve the flow equations.

2-Compute $\frac{\delta F}{\delta b_i}$ (for all i) by solving the adjoint equations.

3- Solve equation (1) by performing MCG iterations, where each iteration involves the computation of the product of the Hessian matrix and the **s** vector.

4-Update the design variables.

Step 1 and 2 have, more or less, the same CPU cost. To assess the efficiency of the TN method, the cost of step 3 should be kept as low as possible.

Method Formulation in Fluid Mechanics

To present the formulation of the TN method in a shape optimization problem governed by the Navier-Stokes equations, let us consider an internal flow case in which a shape that gives the minimum volume–averaged total pressure losses between the inlet (S_I) and outlet (S_O) of the domain is sought. The corresponding objective function reads

$$F_{pt} = -\int_{S_I} (p + \tfrac{1}{2} v_k{}^2)\, v_i n_i dS - \int_{S_O} (p + \tfrac{1}{2} v_k{}^2)\, v_i n_i dS$$

where n_i are the components of the outwards unit normal vector at the boundaries. It is meant that the volume flow rate is fixed, by defining a fixed velocity profile along S_I.

Whatever the shape to be designed becomes, the flow equations of an incompressible laminar flow,

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \quad (2.a)$$

$$R_i{}^v = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] = 0 \quad (2.b)$$

must be satisfied. Here, v_i are the velocity components, p is the static pressure divided by the constant density and ν is the constant bulk viscosity.

The continuous adjoint method is used to compute the gradient of F required by eq. (1). Based on a continuous adjoint development similar to those presented in[7], the field adjoint PDEs, namely

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \quad (3.a)$$

$$R_i{}^u = u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] - \frac{\partial (v_j u_i)}{\partial x_j} + \frac{\partial q}{\partial x_i} = 0 \quad (3.b)$$

(where u_i are the adjoint velocity components and q the adjoint pressure) must be discretized and solved. Eqs. (3) are associated with the adjoint boundary conditions which are omitted here in the interest of space.

Solving the adjoint PDEs, just after the solution of the flow (primal) PDEs, provides all we need to compute the sensitivity derivatives of F. If $F = F_{pt}$, these are given by

$$\frac{\delta F}{\delta b_n} = -\int_{S_W}\left[\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)n_j - qn_i\right]\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b}dS \quad (4)$$

Having computed the first-order derivatives of F via the adjoint method, the next steps are those leading to the computation of the product of the Hessian matrix and any vector **s**. To this end, eq. (4) is differentiated once more w.r.t. the design variables and, then, multiplied by **s**. We, thus, get

$$\frac{\delta^2 F}{\delta b_l \delta b_n}s_l = -\int_{S_w}\left[\nu\left(\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial \bar{u}_j}{\partial x_i}\right) - \bar{q}\delta_i{}^j\right]n_j\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b_n}dS$$
$$-\int_{S_w}\left[\nu\frac{\partial}{\partial x_m}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) - \frac{\partial q}{\partial x_m}\delta_i{}^j\right]n_j\frac{\delta x_m}{\delta b_l}s_l\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b_n}dS$$
$$-\int_{S_w}\left[\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) - q\delta_i{}^j\right]n_j\frac{\partial \bar{v}_i}{\partial x_k}\frac{\delta x_k}{\delta b_n}dS$$
$$-\int_{S_w}\left[\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) - q\delta_i{}^j\right]n_j\frac{\partial^2 v_i}{\partial x_m \partial x_k}\frac{\delta x_m}{\delta b_l}s_l\frac{\delta x_k}{\delta b_n}dS$$
$$-\int_{S_w}\left[\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) - q\delta_i{}^j\right]\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b_n}\frac{\delta(n_j dS)}{\delta b_l}s_l \quad (5)$$

where

$$\frac{\partial u_i}{\partial b_l}s_l = \bar{u}_l\,, \quad \frac{\partial v_i}{\partial b_l}s_l = \bar{v}_l\,, \quad \frac{\partial q}{\partial b_l}s_l = \bar{q}_l$$

To compute the overlined fields, the flow and adjoint equations must be differentiated w.r.t. the design variables (this process is usually referred to as Direct Differentiation or DD) and, then, multiplied by s_l. It is worth noting that the computation of $\frac{\partial u_i}{\partial b_l}$ and $\frac{\partial v_i}{\partial b_l}$ using DD would require N equivalent flow solutions (EFS; this is approximately the cost for solving the flow PDEs). Since, however, only the projection of these fields to the vector **s** are needed, two instead of 2N systems of PDEs must be solved. The DD of eqs. (2) and (3) and the subsequent contraction with **s** yields the following systems of PDEs:

$$R^{\bar{p}} = -\frac{\partial \bar{v}_j}{\partial x_j} = 0 \quad (6.a)$$
$$R_i^{\bar{v}} = \bar{v}_j\frac{\partial v_i}{\partial x_j} + v_j\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial \bar{v}_i}{\partial x_j}+\frac{\partial \bar{v}_j}{\partial x_i}\right)\right] = 0 \quad (6.b)$$

and

$$R^{\bar{q}} = -\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (7.a)$$
$$R_i^{\bar{v}} = \bar{u}_j\frac{\partial v_j}{\partial x_i} + u_j\frac{\partial \bar{v}_j}{\partial x_i} - v_j\frac{\partial \bar{u}_i}{\partial x_j} - \bar{v}_j\frac{\partial u_i}{\partial x_j} + \frac{\partial \bar{q}}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial \bar{u}_j}{\partial x_i}\right)\right] = 0 \quad (7.b)$$

As mentioned above, the cost of solving the system of eqs. (6) & (7) is 2EFS (overall) and this must be performed M_{CG} times within each optimization cycle. By adding the cost of solving the flow and adjoint PDEs (once per cycle), the cost of each cycle becomes equal to (2+2M_{CG}) EFS.

We may now summarize the TN-based algorithm:

$k \longleftarrow 0$
$b_j \longleftarrow$ *init()*
While $k \le k_{max}$ *(Outer loop/Optimization cycles) do*
$v_i, p \longleftarrow$ *Flow equations (2)*
$u_i, q \longleftarrow$ *Adjoint equations (3)*
$\Delta b_j^0 \longleftarrow$ *init(0)*
$r_j^0 = -\frac{\delta F}{\delta b_j} \longleftarrow$ *Gradient expression (4)*
$s_j \longleftarrow r_j^0$
$m \longleftarrow 0$
  while $r^m \neq 0$ *and* $m \le M_{CG}$ *(Inner loop/CG steps) do*
    $\frac{\partial v_i}{\partial b_n}s_n \longleftarrow$ *DD of flow equations(6)*
    $\frac{\partial u_i}{\partial b_n}s_n \longleftarrow$ *DD of adjoint equations(7)*
    $W_i = \frac{\delta^2 F}{\delta b_i \delta b_j}s_j \longleftarrow$ *Projected Hessian(5)*
    $\eta \longleftarrow \frac{r_i^m r_i^m}{s_k W_k}$
    $\Delta b_j^{m+1} \longleftarrow \Delta b_j^m + \eta s_j$
    $r_j^{m+1} \longleftarrow r_j^m - \eta W_j$
    $\beta \longleftarrow \frac{r_i^{m+1} r_i^{m+1}}{r_k^m r_k^m}$
    $s_j \longleftarrow r_j^{m+1} + \beta s_j$
    $m \longleftarrow$ *m+1*
  end while
$b_j \longleftarrow b_j + \Delta b_j$
$k \longleftarrow$ *k+1*
end while

Applications

The curved wall of a symmetrical divergent duct was parameterized using Bezier-Berstein control points and the shape was optimized so as to minimize the total pressure losses for the incompressible flow with Reynolds number equal to 1000. The characteristic length for computing the Reynolds number is equal to the inlet diameter.

The initial and optimized shapes of the divergent duct for minimum total pressure losses are shown in figure 1.

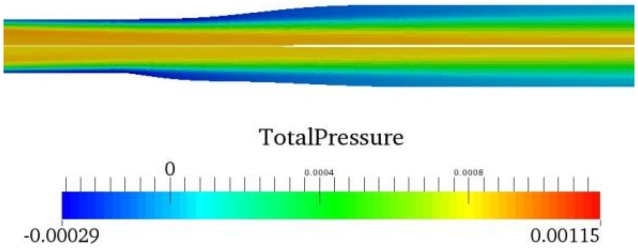

Figure 1: Total pressure in the initial (upper half) and optimized (lower half) part of the duct.

To validate the accuracy with which the adjoint method computes the gradient, a comparison with sensitivity derivatives computed using (costly) finite differences is performed in figure 2.

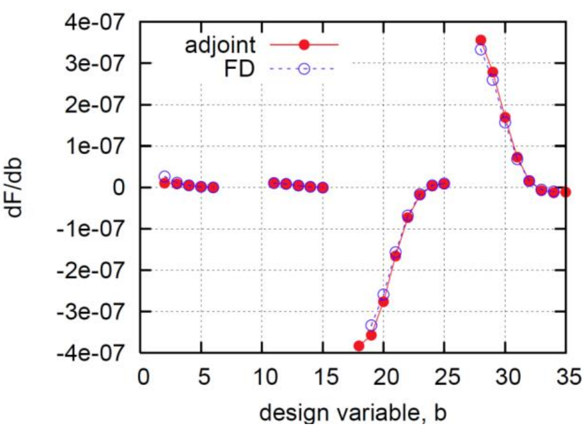

Figure 2: Comparison of the computed sensitivity derivatives with respect to the coordinates of the control points parameterizing the curved walls of the duct between adjoint and finite differences.

As shown in figure 3, the total pressure losses in the optimized shape decreases compared to that of the initial shape.

More detailed presentation of the obtained results as well as results for the second case (isolated airfoil) are deferred to the full paper. In the full paper, emphasis will be laid on the gain offered by the use of TN in shape optimization problem.

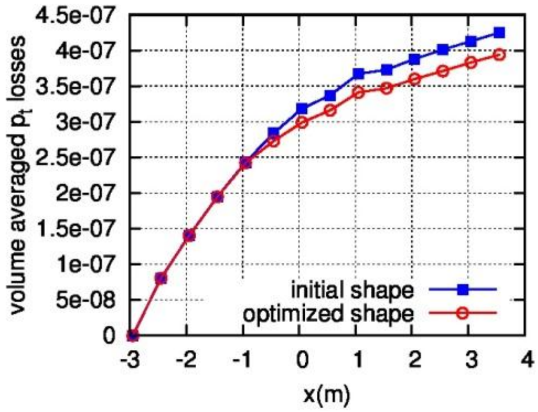

Figure 3: Comparison of the volume average total pressure losses at various transversal positions, between the initial and optimized shape.

Acknowledgment

This research was funded from the People Programme (ITN Marie Curie Actions) of the European Union's 7th Framework Programme (FP7/2007-2013) under REA grant agreement n° 317006 (AboutFLOW project). The first author is an AboutFLOW Early Stage Researcher.

References

[1] Jameson, A. Aerodynamic design via control theory. J. Sci. Comp. 3, 233-260, (1988)

[2] Anderson, WK. and Venkatakrishnan, V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA Paper 06(43), (1997).

[3] Papadimitriou, DI. and Giannakoglou, KC. A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. J. Computers & Fluids 36(2), 325-341, (2007).

[4] Papadimitriou, DI. and Giannakoglou, KC. Aerodynamic shape optimization using first and second order adjoint and direct approaches. Archives of Computational Methods in Engineering, 15(4):447-488, (2008).

[5] Zervogiannis, T., Papadimitriou, DI. and Giannakoglou, KC. Total pressure losses minimization in turbomachinery cascades using the exact Hessian. Computer Methods in Applied Mechanics and Engineering, 199:2697-2708, (2010).

[6] Papadimitriou, DI., Papoutsis-Kiachagias,

EM. and Giannakoglou, KC. Topology optimization in fluid dynamics using adjoint-based truncated Newton. ECCOMAS Conference, (2012).

[7] Papoutsis-Kiachagias, EM. and Giannakoglou KC. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization : industrial applications. Archives of Computational Methods in Engineering 1-45, (2014).