AIAA-2007-3947

18th AIAA Computational Fluid Dynamics Conference, Miami, FL

Implementation of a continuous adjoint for topology optimization of ducted flows

Carsten Othmer*
Volkswagen AG, 38436 Wolfsburg, Germany

Eugene de Villiers†
Icon CG, London, W14 9DH, U.K.

Henry G. Weller‡
OpenCFD Ltd., Reading, Berkshire, RG4 7AN, U.K.

Topology optimization of fluid dynamical systems is still in its infancy, with its first academic realizations dating back to just four years ago. In this paper, we present an approach to fluid dynamic topology optimization that is based on a continuous adjoint. We briefly introduce the theory underlying the computation of topological sensitivity maps, discuss our implementation of this methodology into the professional CFD solver OpenFOAM and present results obtained for the optimization of an airduct manifold wrt. dissipated power.

I. Fluid dynamic topology optimization

In structure mechanics, topology optimization is a well-established concept for design optimization with respect to tension or stiffness.[1] Its transfer to computational fluid dynamics, however, began just four years ago with the pioneering work of Borrvall and Petersson.[2] Since then, this topic has received significant interest in both academia and industry[3–9].

The starting point for fluid dynamic topology optimization is a volume mesh of the entire installation space. Based on a computation of the flow solution inside this domain, a suitable *local* criterion is applied to decide whether a fluid cell is favourable or counterproductive for the flow in terms of the chosen cost function. In order to iteratively remove the identified counterproductive cells from the fluid domain, they are either punished via a momentum loss term, or holes are inserted into the flow domain, with their positions being determined from an evaluation of the so-called topological asymptotic.

In the former case, the momentum loss term is usually realized via a finite cell porosity, i. e. the whole design domain is treated as a porous medium: Each cell is assigned an individual porosity α_i, which is modeled via Darcy's law. The value of α_i determines if the cell is fluid-like (low porosity values) or has a rather solid character (high values of α_i). In other words, the porosity field controls the geometry, and the α_i are the actual design variables.

Within this framework, an adjoint method can be applied to elegantly compute the sensitivities of the chosen cost function wrt. the porosity of each cell. The obtained sensitivities can then be fed into a gradient-based optimization algorithm – possibly with some penalization of intermediate porosity values in order to enforce a "digital" porosity distribution, and after several iterations, the desired optimum topology is finally extracted as an iso-surface of the obtained porosity distribution or similar post-processing operations.

In a recent study, Othmer et al.[8] were able to verify the applicability of this methodology to typical automotive objective functions, including dissipated power, equal mass flow through different outlets, flow uniformity and angular momentum of the flow in the outlet plane. In that proof-of-concept study, Automatic Differentiation techniques were applied to an academic CFD code in order to obtain a discrete adjoint solver. For industrial-sized problems, however, this code is not suitable. Therefore, we implemented the methodology via a continuous adjoint into the professional CFD environment OpenFOAM.[10] The underlying equations and their implementation will be presented in the following sections, before we demonstrate the application of this methodology to an airduct manifold.

*Computational Engineer, CAE Method Development
†Consultant Engineer, Open Source Services Devision
‡Core Developer, OpenFOAM

II. Computation of topological sensitivity maps: theory

If J stands for the cost function to be minimized, the optimization problem can be stated as follows:

$$\text{minimize } J = J(\alpha, \mathbf{v}, p) \text{ subject to } \mathcal{R}(\alpha, \mathbf{v}, p) = 0\,, \tag{1}$$

where α represent the design variables, i. e. the porosity distribution, and $\mathbf{v}$ and p stand for velocity and pressure, respectively. $\mathcal{R} = (R_1, R_2, R_3, R_4)^T$ denotes the state equations, in our case the incompressible, steady-state Navier-Stokes equations:

$$(R_1, R_2, R_3)^T = (\mathbf{v}\cdot\nabla)\,\mathbf{v} + \nabla p - \nabla\cdot\Big(2\nu\mathsf{D}(\mathbf{v})\Big) + \alpha\mathbf{v} \tag{2}$$

$$R_4 = -\nabla\cdot\mathbf{v}\,, \tag{3}$$

with the rate of strain tensor $\mathsf{D}(\mathbf{v}) = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right)$ and ν being the effective viscosity, i. e. the sum of turbulent and molecular viscosity: $\nu = \nu_t + \nu_m$. The essential component for the topology optimization methodology is the the Darcy term $\alpha\mathbf{v}$.

We thus have a constrained optimization problem, with the constraints being the state equations. Such problems are commonly tackled by introducing a Lagrange function L and reformulating the cost function as

$$L := J + \int_\Omega (\mathbf{u}, q)\,\mathcal{R}\,d\Omega\,, \tag{4}$$

where we have introduced the adjoint velocity $\mathbf{u}$ and the adjoint pressure q as Lagrange multipliers.

The first objective function of our implementation is the power dissipated by the fluid dynamic device. It can be computed as the net inward flux of energy, in our case total pressure, through the device boundaries:

$$J = -\int_\Gamma d\Gamma\,(p + v^2/2)\mathbf{v}\cdot\mathbf{n}\,. \tag{5}$$

This objective function involves only an integral over the outer surface of the flow domain and has no volume contribution from the domain itself. In such cases the adoint partial differential equations read:

$$-2\mathsf{D}(\mathbf{u})\,\mathbf{v} = -\nabla q + \nabla\cdot\Big(2\nu\mathsf{D}(\mathbf{u})\Big) - \alpha\mathbf{u} \tag{6}$$

$$\nabla\cdot\mathbf{u} = 0\,. \tag{7}$$

These are the equations that have to be implemented into the CFD solver, along with appropriate boundary conditions that depend on the chosen cost function. In the case of dissipated power optimization, these are

$$\mathbf{u}_t = 0 \tag{8}$$

$$u_n = \begin{cases} 0 & \text{at wall} \\ v_n & \text{at inlet} \end{cases} \tag{9}$$

for the adjoint velocity at wall and inlet (for details of the derivation of boundary conditions we refer to Ref. 11). At the outlet, the adjoint quantities have to satisfy

$$q = \mathbf{u}\cdot\mathbf{v} + u_n v_n + \nu(\mathbf{n}\cdot\nabla)u_n - v^2/2 - v_n^2 \tag{10}$$

$$0 = v_n(\mathbf{u}_t - \mathbf{v}_t) + \nu(\mathbf{n}\cdot\nabla)\mathbf{u}_t\,. \tag{11}$$

While Eqn. (10) defines the adjoint pressure at the outlet, Eqn. (11) is used to determine the tangential component of the adjoint velocity $\mathbf{u}_t$. For the gradient of the normal component u_n, continuity then implies

$$(\mathbf{n}\cdot\nabla)u_n = \nabla\cdot\mathbf{u} - \nabla_\parallel\cdot\mathbf{u}_t = -\nabla_\parallel\cdot\mathbf{u}_t\,. \tag{12}$$

This completes the adjoint equation system for optimization wrt. dissipated power. After solving it for $\mathbf{u}$ and q, the desired sensitivities can be computed according to Eqn. (4) as

$$\frac{\partial L}{\partial\alpha_i} = \frac{\partial J}{\partial\alpha_i} + \int_\Omega (\mathbf{u}, q)\,\frac{\partial\mathcal{R}}{\partial\alpha_i}\,d\Omega\,, \tag{13}$$

where $\partial L/\partial \alpha_i$ is the sensitivity of the cost function wrt. the porosity α_i of cell i. For the topology optimization methodology, the porosity is just an auxiliary variable to describe a continuous transition from fluid to solid. Therefore, there is no *explicit* dependence of the cost function on the porosity: $\partial J/\partial \alpha_i = 0$. Furthermore, as the porosity α_i enters the primal equation system only in cell i and only via the Darcy term, we can write

$$\frac{\partial \mathcal{R}}{\partial \alpha_i} = \begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix} \chi_i \tag{14}$$

with χ_i being the characteristic function of cell i. Hence, according to Eqn. (13), we can finally compute the desired sensitivity for each cell as the scalar product of adjoint and primal velocity times the cell volume:

$$\frac{\partial L}{\partial \alpha_i} = \mathbf{u}_i \cdot \mathbf{v}_i \, V_i \, . \tag{15}$$

III. Solver implementation

OpenFOAM (Open **F**ield **O**peration **A**nd **M**anipulation) is a CFD toolbox that can be used to simulate a broad range of physical problems. The code was chosen as the development environment for the topology optimizer due partially to its open source (GNU General Public Licence – GPL) and therefore transparent nature, but primarily because of its high level symbolic application programming interface (API). The flexibility of this interface allows for a straight forward implementation of the continuous adjoint, using previously validated components that make up the other applications in the toolbox. Fig. 1 shows the source code for the adjoint solver component of the topology optimization tool.

```
1    tmp<fvVectorMatrix> UEqn
2    (
3        fvm::div(-phi, U)
4      - fvc::grad(U) & V
5      + turbulence->divR(U)
6      + fvm::Sp(alpha, U)
7    );
8
9    solve(UEqn() == -fvc::grad(q));
10
11   volScalarField rAU = 1.0/UEqn().A();
12
13   U = rAU*UEqn().H();
14
15   phia = fvc::interpolate(U) & mesh.Sf();
16
17   fvScalarMatrix qEqn
18   (
19       fvm::laplacian(rAU, q) == fvc::div(phia)
20   ); qEqn.solve();
21
22   phia -= qEqn.flux();
23
24   U -= rAU*fvc::grad(q);
```

Figure 1. Adjoint solver implementation in OpenFOAM.

The main symbols in Fig. 1 are defined as follows:

- `phi` - primal inter-cell volume flux
- `phia` - adjoint inter-cell volume flux
- `q` - adjoint pressure
- `U` - adjoint velocity
- `V` - primal velocity

where fluxes are calculated using the velocities from the previous iteration. Throughout `fvm::` prefaces implicit finite volume operators, while `fvc::` denotes the explicit equivalents. The solver uses a segregated

approach and a SIMPLE-type algorithm to couple the adjoint velocities and pressure. Turbulence is assumed to be "frozen", so that the primal turbulent viscosity can be re-used for the adjoint diffusion term.

The adjoint momentum predictor is constructed in lines 1-7, with the first term consisting of the negative convection of the adjoint velocity by the primal (line 3). The next term (line 4), which describes the dot product of the adjoint gradient with the primal velocity, has to be represented explicitly, since the dot product of the gradient creates cross-coupling between the adjoint velocity components. The capability to solve cross-coupled equations is not currently available in the code library.

Turbulent and laminar diffusion is handled via a plug-in module, in this case denoted by the `turbulence -> divR(U)` term which is defined as:

$$-\nabla \cdot 2\left(\nu_t + \nu_m\right) \mathsf{D}(\mathbf{u}) \,, \tag{16}$$

where ν_t is the primal turbulent viscosity and $\mathsf{D}(\mathbf{u})$ was defined earlier. The current formulation additionally groups the calculation of the wall shear stress with the turbulence library, such that

$$\nu_{tw} = \tau_w / \left|\nabla \mathbf{v}_0\right| - \nu_m \,, \tag{17}$$

where τ_w, the primal wall shear stress, is calculated from some wall function. The assumption of "frozen" turbulence raises some difficult questions, particularly when applied to the correlation between the wall shear and the near-wall velocity gradient. Eqn. (17) is known to exhibit very non-linear behaviour, so a more accurate approach would be to derive the adjoint wall function equivalent and use it to calculate the adjoint of the wall shear stress. For this exploratory study the current approach is however assumed to be adequate.

The final term in the momentum equation represents the effect of porosity (`alpha`) used as a momentum sink in counterproductive cells. The adjoint momentum is then solved by setting the matrix equal to the adjoint pressure gradient (line 9). The construction of the adjoint pressure equation (lines 17-20) requires an intermediate adjoint flux found by dividing the right hand side of the `Ueqn` matrix (`.H()`) by the diagonal coefficient (`.A()`) (lines 11, 13 and 15). After the pressure has been solved the intermediate flux and velocity have to be updated so that they obey continuity (lines 22 and 24).

Once both the primal and adjoint fields have been solved, the porosity should be updated using the calculated sensitivities. For stability reasons, modifications of `alpha` have to be under-relaxed (a typical value is around 0.1). A steepest descent method can then be implemented as follows:

$$\alpha_n = \alpha_o \left(1 - \gamma_r\right) + \gamma_r \, \min\left(\max\left(\alpha_o - \lambda \left(\mathbf{u} \cdot \mathbf{v}\right), 0\right), \alpha_{max}\right) \,, \tag{18}$$

where

γ_r - relaxation factor

λ - the product of the cell volume, V_i and the sensitivity response step size δ

α_{max} - maximum allowed porosity corresponding to a "blocked" cell

α_n, α_o - porosities at new and old times respectively.

Despite being based on a continuous rather than a discrete adjoint, the current code uses the so-called "one-shot" approach for the solution procedure, i. e. the porosity update uses partially converged primal and adjoint solutions to calculate the sensitivities. Experience with simple test cases has shown that the results at convergence are comparable to using a segregated approach.

IV. Ventilation duct application

The flow splitter manifold shown in Fig. 2 represents the space envelope and inlet/outlet boundaries for a typical climatisation system component. As such it represents a relatively simple, yet topical and sufficiently interesting case for initial investigations of the topology optimization methodology. The duct consists of a single inlet with four outlets leaving the domain at a right angle to the inlet. The boxed section represents the possible design envelope. To save computational resources, the domain is halved through the use of a symmetry plane.

Since the optimization can potentially displace the porous/non-porous boundary to anywhere within the calculation domain, the space is discretised using a uniform cell size of 2x2x2 mm. The resulting mesh consists of approximately 7.5 million primarily hexahedral cells. Lower than hex elements are found on some surfaces due to projection of the mesh to the surface to produce surface conforming boundaries. The flow is assumed to be incompressible and isothermal with physical properties derived from air at 293 K. The air enters the 0.175 m diameter inlet with a fixed velocity of 21 m/s, resulting in a Reynolds number of $\approx$ 250,000. A hybrid formulation of the Spalart-Allmaras turbulence model is used and constant inlet properties are set using a turbulent length scale of 0.015 m and an intensity based on the inlet velocity of 5%. Pressure boundary conditions are zero-gradient everywhere except on the outlets, where a fixed relative pressure of 0 is enforced. The adjoint velocity at the inlet and walls is set equal to the primal velocity. At the outlet the normal components of primal and adjoint velocity are found from the pressure flux, while the tangential velocity is zero-gradient. For the adjoint pressure at the outlet, a boundary condition according to Eqn. (10) is applied.

Although inaccurate, it was found that first-order upwind discretisation had to be employed for convection terms to ensure stability. The impact of this approximation should be somewhat ameliorated by the small grid element size. The flow is assumed to be steady state and advanced using standard under-relaxation for all solution variables.

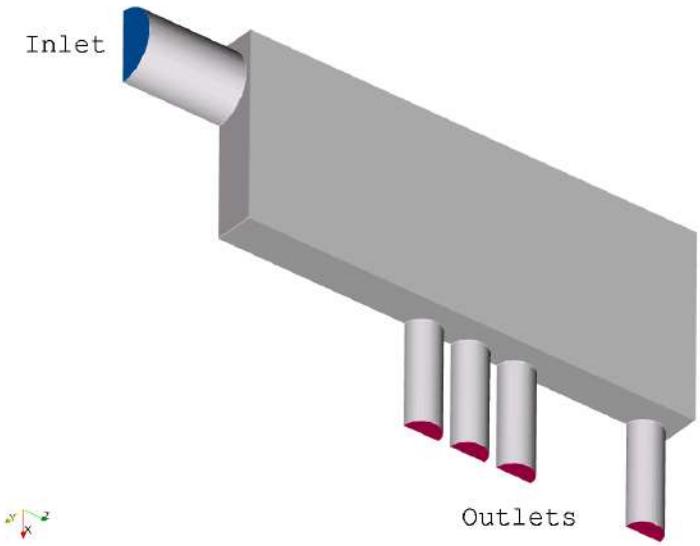
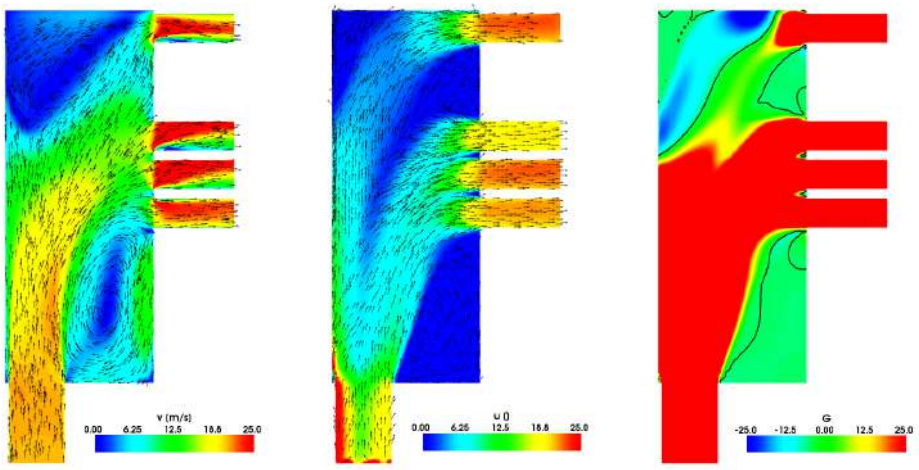


Figure 2. Flow splitter manifold geometry.

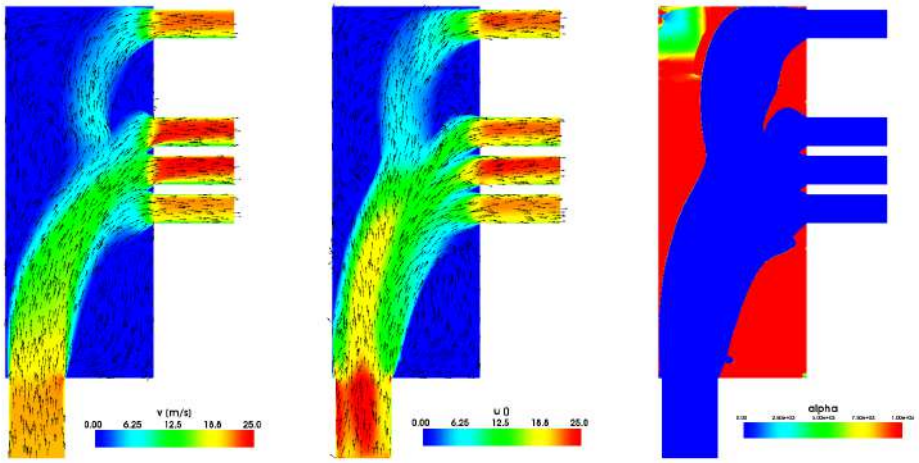
The solver is run initially without porosity updates to obtain a steady solution for both the primal and the adjoint (2000 iterations). Fig. 3 depicts results on the symmetry plane for the primal and adjoint velocities (a and b respectively) and the sensitivity derived from their dot product (c). The primal velocity (Fig. 3a) clearly shows 2 large recirculation zones at the top left and bottom right corners. In the sensitivity map (Fig. 3c) the black iso-line of zero sensitivity delineates these zones as counterproductive, i. e. once the porosity update starts they will be penalized.

After initiating porosity update, the solution is stepped forward for an additionally 20,000 iterations before a quasi-steady solution is obtained. Full convergence is not achieved for all variables as the porous region fluctuates around a mean distribution. Fig. 4 shows the solution on the symmetry plane for the adjoint and primal velocities alongside the final porosity distribution. The optimized primal velocity (Fig. 4a) is seen to be much more compact than in the baseline case (Fig. 3a), with negligible amounts of flow in recirculation regions. The reason for this is obvious from the optimized porosity distribution (Fig. 4c), with all counterproductive cells having been penalised with high porosity, while the main channels are free of blockage. In addition, the approaches to the outlet sections have been tapered, which has the effect of removing the previously observed small recirculation regions downstream of the outlet duct throats.



(a) (b) (c)

Figure 3. Initial results for the flow splitter manifold: (a) Primal velocity, (b) Adjoint velocity, (c) Sensitivities (blue: counterproductive, red: favourable, black line = 0).



(a) (b) (c)

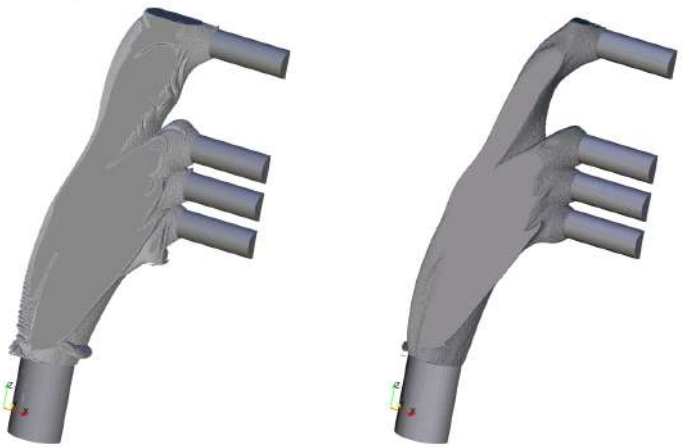
Figure 4. Optimized results for the flow splitter manifold: (a) Primal velocity, (b) Adjoint velocity, (c) Porosity (blue: no blockage, red: maximum blockage).

According to Eqn. (5), the power dissipated by a constant density fluid flowing through a duct can be computed as

$$J = -\int_{inlet} dA\,(p + v^2/2)\mathbf{v}\cdot\mathbf{n} \quad -\int_{outlets} dA\,(p + v^2/2)\mathbf{v}\cdot\mathbf{n}\,. \tag{19}$$

For the baseline case this gives a power dissipation of 60.8 W, while the optimized duct displays a dissipation of 34.8 W, an improvement of 43%. While the baseline case clearly represents a worst-case scenario, the magnitude of the improvement is encouraging considering the relative simplicity of the implementation.

While the porosity can be used directly to extract an iso-surface from the solution for design purposes, experience has shown that the current methodology does not in general produce a smooth manufacturable duct. An alternative is to use an iso-surface of velocity magnitude to exclude very low velocity regions from the educed shape, which tends to include small unwanted features produced via instabilities and other inaccuracies. A comparison of the shapes produced via these two methods is shown in Fig. 5.



(a) (b)

Figure 5. Optimized duct surface: (a) Zero porosity, (b) Velocity magnitude ($|\mathbf{v}| > 3$ m/s).

V. Summary and Outlook

We have presented the theory underlying the computation of topological sensitivity maps and its implementation into the CFD environment OpenFOAM. The application of the developed code to the minimization of power dissipation of a 3D airduct manifold demonstrated the potential of this methodology.

Even though the obtained results are very promising, the method suffers from some deficiencies: Currently, no account is taken of wall functions in the adjoint, and the turbulence field is assumed to be "frozen". Future work will therefore be dedicated to remedy these shortcomings. In addition, extension of the code to other objective functions is planned, which for a large number of objective functions only involves modification of the adjoint boundary conditions. Thanks to the flexible high level symbolic API of OpenFOAM, such adaptations within the developed adjoint solver framework are straight forward.

References

[1]M.P. Bendsøe and O. Sigmund, Topology optimization: theory, methods, and applications, Springer, Berlin, 2004.

[2]T. Borrvall and J. Petersson, Topology optimization of fluids in Stokes flow, *Int. J. Num. Meth. Fluids*, **41**, p. 77, 2003.

[3]O. Sigmund, A. Gersborg-Hansen, and R.B. Haber, Topology optimization for multi-physics problems: A future FEMLAB application?, *Proc. Nordic MATLAB Conf. 2003*, L. Gregersen (Ed.), Comsol A/S, Søborg, Denmark, p. 237, 2003.

[4]L.H. Olesen, F. Okkels, and H. Bruus, Topology optimization of Navier-Stokes flow in microfluidics, *ECCOMAS 2004*, Jyväskylä, 2004.

[5]O. Moos, F.R. Klimetzek, and R. Rossmann, Bionic optimization of air-guiding systems, *SAE 2004-01-1377*, 2004.

[6]C. Othmer and Th. Grahs, Approaches to fluid dynamic optimization in the car development process, *EUROGEN 2005*, Munich, 2005.

[7]J.K. Guest and J.H. Prévost, Topology optimization of creeping flows using a Darcy-Stokes finite element, *Int. J. Num. Meth. Engng.*, **66** (3), p. 461, 2006.

[8]C. Othmer, Th. Kaminski, and R. Giering, Computation of topological sensitivities in fluid dynamics: Cost function versatility, *ECCOMAS CFD 2006*, Delft, 2006.

[9]L.H. Olesen, F. Okkels, and H. Bruus, A High-level Programming-language Implementation of Topology Optimization Applied to Steady-state Navier-Stokes Flow, *Int. J. Num. Meth. Eng.*, **65**, p. 975, 2006.

[10]http://www.opencfd.co.uk/openfoam/www.opencfd.org

[11]C. Othmer, A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows, priv. commun., 2007.