INDUSTRIAL APPLICATION OF CONTINUOUS ADJOINT FLOW SOLVERS FOR THE OPTIMIZATION OF AUTOMOTIVE EXHAUST SYSTEMS

C. Hinterberger*, M. Olesen*

*Faurecia Emissions Control Technologies, Germany GmbH
e-mail: Christof.Hinterberger, Mark.Olesen@faurecia.com

Key words: Geometry Optimization, Adjoint Flow Solver, Exhaust Systems

Abstract. A continuous adjoint geometry optimization tool (CAGO), has been developed at Faurecia Emissions Control Technologies [1], which finds suitable shapes for catalyst inlet cones directly from the package space. CAGO produces a design proposal, which is used as a reference surface in the CAD process. Subsequently, the CAD design is validated in a fully compressible flow analysis. For further optimization sensitivities are computed by solving the adjoint flow fields with a frozen density and frozen viscosity assumption. This pseudo-compressible continuous adjoint method is described in the paper in detail in a general form including scalar transport.
1 OPTIMIZATION OF EXHAUST SYSTEMS

Meeting backpressure and flow uniformity requirements within severe packaging constraints presents a particular challenge in the layout of catalyst inlet cones. Figure 1 illustrates how the flow uniformity of a closed coupled catalyst can be influenced by the design of the inlet cone.

Especially for complex package spaces it can prove difficult to find designs that fulfill the uniformity targets. To address this problem, a continuous adjoint geometry optimization tool (CAGO) has been developed at Faurecia Emissions Control Technologies that finds suitable shapes for catalyst inlet cones directly from the package space[1]. The tool is based on the continuous adjoint formulation derived and implemented by Othmer et al.[2, 3]. The implementation uses the open source CFD toolbox OpenFOAM\(^1\) [5]. As shown in Figure 2, CAGO begins from the provided package space and produces a design proposal that can be used as a reference surface (in IGES format) during the CAD process.

Once a corresponding CAD geometry has been established — including consideration of manufacturing and durability constraints — a fully compressible flow analysis is performed, in which the catalysts are modelled as anisotropic porosities. In this analysis, the surface sensitivities for the optimization of the flow uniformity and the pressure drop (see Figure 2b) are also computed by solving the adjoint flow fields with a frozen density and frozen viscosity assumption. This pseudo-compressible continuous adjoint method is described in the methodology section of the paper. For systems that are equipped with a fuel vaporizer — which is an exhaust system component for introducing additional fuel to support DPF (diesel particulate filter) regeneration — the surface sensitivities for the uniformity of the fuel vapour distribution in front of a DOC (diesel oxidation catalyst) can

\(^1\)OpenFOAM® is a registered trademark of OpenCFD Ltd.
be also calculated. The resulting surface sensitivities can be used to assess the effect of geometry modifications and to identify areas that are critical with respect to manufacturing tolerances. In the current optimization workflow, geometry modifications at this later stage are either applied manually in the CAD model or by morphing the CFD surface mesh directly. Since the number of additional constraints (e.g., manufacturing) increases at these later design stages, further automation is difficult to define or implement and some degree of manual intervention must be accepted. Nevertheless, by enhancing the usability of the geometry manipulation tools, further improvement of the design process is still possible.

Figure 2: (a, b) Workflow of design and optimization process. (b) surface sensitivities for pressure drop and flow uniformity, showing areas which have to be expanded (green) or shrunk (red) to improve results.

Figure 3: Schematic of the optimization with CAGO: (a) package space and (b) computational mesh and boundary conditions.
Figure 4: Automatic geometry optimization with CAGO; (a, c) volumetric sensitivities for energy dissipation, (b, d) volumetric sensitivities for flow uniformity.
2 CONTINUOUS ADJOINT METHOD

Our optimization process uses two different adjoint solvers. The first is the solver used in our automatic geometry optimization tool CAGO, which is based on an incompressible continuous adjoint solver originally developed by Othmer et al.[2], and which is described in detail in[1]. CAGO operates on a fixed (non-moving) mesh for the package space and uses a level set method to describe the geometry (Figure 3). The geometry is adjusted automatically according to computed sensitivity fields. This is shown in Figure 4 where the development of the cone geometry over the solver iterations can be seen. CAGO is used to generate a design proposal for the CAD designer.

The CAD design is then validated using a fully compressible CFD simulation. For the further optimization of this model, sensitivities are calculated using a pseudo-compressible continuous adjoint method, which uses a frozen density and frozen turbulence assumption. This method is described in this section in a general form including scalar transport, so that it can be easily transferred to other applications (e.g. optimization of heat exchangers, as seen in [4]). The derivation of the equations follows the theoretical paper of Othmer[3].

2.1 Concept of the adjoint CFD method

At the beginning of an optimization problem, a cost function \( J = J(c, \xi) \), which is to be minimized, is defined. The cost function depends on the geometry, which is specified via a design vector \( c \) and it depends on the flow field \( \xi \). The total variation of the cost function with respect to a design change is thus given as follows

\[
\delta J = \delta_{\text{geometry}} J + \delta_{\text{flow}} J = \partial J/\partial c \cdot \delta c + \partial J/\partial \xi \cdot \delta \xi
\]  

(1)

When the design changes by an amount \( \delta c \), the flow field changes accordingly by \( \delta \xi \) and this affects the cost function \( J \), causing the variation \( \delta J \). Additionally, there could be a direct (geometrical) dependence of \( J \) on the geometry, causing the variation \( \delta_{\text{geometry}} J \). The variation can also be written as \( \delta_{\text{c}} J = \partial J/\partial c \cdot \delta c = \partial c J \cdot \delta c \).

For geometry optimization, it is quite convenient to define a sensitivity field in the following form:

\[
\delta J = \partial_{\text{sensitivity}} L \cdot \delta c \]  

(2)

in which the sensitivity field \( \partial_{c} L \) relates how the cost function \( J \) is affected by an arbitrary, small, and reasonable smooth, design variation \( \delta c \). Since the sensitivities \( \partial_{c} L \) have to account for how design variations influence the flow field \( \xi \), an augmented cost function

\[
L = J + \int_{\Omega} \Psi \cdot R
\]  

(3)
C. Hinterberger, M. Olesen

introduces the state equations \( R(\xi) = 0 \), which are the Navier-Stokes equations written in residual form, as an constraint into the optimization problem, with the adjoint flow field \( \Psi \) acting as Lagrange multiplier.

The augmented cost function \( L \) depends on the design \( c \) and on the flow field \( \xi \), and therefore its total variation is

\[
\delta L = \delta_{\text{geometry}} L + \delta_{\text{flow}} L = \partial_c L \cdot \delta c + \partial_\xi L \cdot \delta \xi
\]  

(4)

By requiring

\[
\delta_\xi L \equiv 0
\]

(5)

the total variation (4) becomes \( \delta L \equiv \delta_c L = \partial_c L \cdot \delta c \), which includes the sensitivity \( \partial_c L \) defined in (11). This requirement defines the adjoint flow field \( \Psi \). With \( \delta (\Psi R) = \Psi \delta R + R \delta \Psi \) and \( R = 0 \), the variation of the augmented cost function (3) is given as

\[
\delta L = \delta J + \int_\Omega \Psi \cdot \delta R
\]

(6)

and (5) expands to

\[
\delta_\xi L = \delta_\xi J + \int_\Omega \Psi \cdot \delta_\xi R \equiv 0
\]

(7)

From this, an equation system \( A(\Psi) = 0 \) can be derived for the adjoint flow field \( \Psi \), in which the boundary conditions and source terms depend on \( J \) (see section 2.5.2). The adjoint equations \( A(\Psi) = 0 \) are very similar to the flow equations \( R(\xi) = 0 \), and can be solved with a similar numerical cost. Once the adjoint equations are solved, the variation of the cost function with respect to an arbitrary design change, can be computed directly from the primal flow \( \xi \) and adjoint flow field \( \Psi \), as follows

\[
\delta_c L = \delta_c J + \int_\Omega \Psi \cdot \delta_c R
\]

(8)

without the need of an extra CFD–solution of the state equations. This can be easily applied for the computation of volumetric sensitivities, as it will be shown later, but for surface sensitivities it is difficult to compute \( \delta_c R \). However, since the total variation of the state equation is zero,

\[
\delta R = \delta_c R + \delta_\xi R = 0
\]

(9)

the computation of shape sensitivities can be defined alternatively:

\[
\delta_c L = \delta_c J - \int_\Omega \Psi \cdot \delta_\xi R
\]

(10)
Combining (1), (7) and (10), shows that (2) holds, and gives a descriptive relation between $\delta_c L$ and $\delta J$:

$$\delta L = \delta_c L = \delta_c L \cdot \delta c = \delta c J + \delta J = \delta J \quad (11)$$

Therefore, the quantity of main interest — the total variation $\delta J$ of the original cost function with respect to a design change (including alterations in flow field arising from this design change) — is equal to $\delta_c L$, and this quantity can be easily computed for any arbitrary small geometry variation $\delta c$ from the sensitivity field $\partial_c L$ via (8) for volumetric sensitivities and via (10) for shape sensitivities. The adaption of the flow field $\delta \xi$ to a design variation $\delta c$ is included via the term $\delta_c J$. The term $\delta_c J$ describes the direct (geometrical) dependence of $J$ on the design $c$. In many practical cases, the definition of $J$ itself is independent of the design vector $c$ and thus $\delta_c J = 0$.

2.2 Primal flow in exhaust systems

For the CFD optimization of automotive exhaust systems, the compressible flow at a maximum mass flow rate, which occurs at full blow down conditions, is typically calculated. The flow field $(v, \rho, p, T, k, \epsilon, \Phi_i)$ is computed by solving the steady-state, fully compressible flow equations, with the standard high-Re $k-\epsilon$ model. The catalysts are modeled as anisotropic porosities and the exhaust gas is treated as an ideal gas with $p = \rho RT$.

2.3 Cost functions

For the optimization of exhaust systems, the following types of cost functions are of interest:

- **dissipated power**

$$J_1 = -\int_{\text{inlet}} p_{\text{total}} v \cdot n - \int_{\text{outlet}} p_{\text{total}} v \cdot n$$

- **flow uniformity**

$$J_2 = \int_{\text{CAT}} \frac{1}{2} (v - \langle v \rangle)^2$$

- **species uniformity**

$$J_3 = \int_{\text{CAT}} \frac{1}{2} (\Phi - \langle \Phi \rangle)^2$$

These cost functions can either be optimized individually or be combined with weighting factors $w_i$ as $J = \sum_i w_i J_i$ for a multiobjective optimization.
2.3.1 Generic cost function

For the description of the adjoint method used, we consider a generic cost function

\[ J = \int_{\Omega} J_{\Omega} + \int_{\Gamma} J_{\Gamma} \]

(15)

with local contributions \( J_{\Omega} \left[ \frac{|\mathbf{u}|}{m^3} \right] \) and \( J_{\Gamma} \left[ \frac{|\mathbf{u}|}{m^2} \right] \) from the volume \( \Omega \) and the surface \( \Gamma \) of the flow domain. The compact notation \( \int_{\Gamma} := \int_{\Gamma} d\Gamma \) and \( \int_{\Omega} := \int_{\Omega} d\Omega \) is used in the following sections for surface and volume integrals, respectively.

2.4 Control variables

For the control variables (\( c \) in section 2.1) we use, in accordance with Othmer[3], the following fields:

- a porosity field \( \alpha \), acting as 'virtual' sand inside the fluid region
- the normal surface displacement \( \beta \), a scalar field defined at the surface of the geometry.

This provides a convenient distinction between volumetric sensitivities \( \partial_{\alpha} L \) and surface shape sensitivities \( \partial_{\beta} L \). The surface sensitivities can be used to adjust the geometry (e.g., via morphing), whereas the volumetric sensitivities \( \partial_{\alpha} L \) are useful to identify regions within the fluid volume that are detrimental to the cost function (e.g., recirculation regions for pressure loss). The sensitivities of the cost functions to changes in the control variables are computed via a continuous adjoint CFD method as described previously in section 2.1.

2.5 Pseudo-compressible continuous adjoint method

The formulation of the continuous adjoint method is simplified somewhat by making a frozen density and frozen turbulence assumption. For our applications, the density variations linked to design variations are quite small: The density depends primarily on the system temperature and on the system pressure, which is dominated by the pressure drop across the catalysts. Although Zymaris et. al. [6] showed that the frozen turbulence assumption can have quite an significant influence on the overall scaling of the computed sensitivities, using an adjoint turbulence model like Zymaris et. al. [6, 7] increases the overall complexity of the simulations significantly and potential numerical issues (e.g., stability) could make it difficult for use as a standard engineering tool.
2.5.1 State equation

We consider (a subset of) the computed primal flow field

\[ \xi = \begin{pmatrix} v \\ p \\ \Phi \end{pmatrix} = \begin{pmatrix} \text{velocity} \\ \text{pressure} \\ \text{passive scalar – e.g., species concentration} \end{pmatrix} \]

(16)
to be the solution of the pseudo-compressible steady state Navier-Stokes equations (NSE), written in residual form as

\[ \mathbf{R}(\xi) = 0 \]

(17)

\[ \begin{pmatrix} R_1, R_2, R_3 \\ R_4 \\ R_5 \end{pmatrix} = \begin{pmatrix} \text{momentum equations} \\ \text{continuity equation} \\ \text{scalar transport} \end{pmatrix} \]

\[ \begin{pmatrix} \text{convection} \\ \text{diffusion} \\ \text{pressure} \end{pmatrix} \]

\[ \begin{pmatrix} \text{scalar convection} \\ \text{scalar diffusion} \end{pmatrix} \]

with \( D(v) = \frac{1}{2}(\nabla v + v \nabla) \). The tilda operator \( \tilde{\cdot} \) has been introduced as a convenience to denote a multiplication with the density \( \rho \). When the \( \tilde{\cdot} \) operator is omitted, the incompressible equations are recovered. The continuity equation for a steady-state compressible flow \( \nabla \cdot \tilde{v} = 0 \) was used, i.e. \( \nabla (\tilde{v} \Phi) = (\tilde{v} \cdot \nabla) \Phi + \Phi \nabla \cdot \tilde{v} = (\tilde{v} \cdot \nabla) \Phi \), to write the pseudo-compressible NSE in a non-conservative form, convenient for the further manipulations. The density \( \rho \), the effective turbulent viscosity \( \nu \) and the effective turbulent diffusivity \( \Gamma \) are provided by the fully compressible primal flow solver. These fields are variable in space, but assumed to be constant with respect to a design variation. For the computation of the sensitivities, their dependence on geometry variations is considered negligible. The porous resistance term \( f = \tilde{\alpha} \mathbf{v} \) serves as a ‘virtual’ resistance to compute volumetric sensitivities \( \partial_\alpha L \), but is also present in the porosity model of the catalysts, where \( \alpha := \alpha_{ij} \) is an anisotropic porosity.
2.5.2 Derivation of the adjoint equations

The starting point for adjoint equation system is (7), also given here in index notation:

\[ \delta \xi L = \delta \xi J + \int_{\Omega} \Psi \cdot \delta \xi R = 0 \]  
\[ = \sum_i \frac{\partial J}{\partial \xi_i} \delta \xi_i + \sum_{i,j} \int_{\Omega} \Psi_j \frac{\partial R_j}{\partial \xi_i} \delta \xi_i \]

with \( \xi = (v, p, \Phi)^T \) and \( \Psi = (u, q, \varphi)^T \). The physical dimension of the adjoint field is \([\Psi_j] = [J] \cdot \frac{m^3}{s} \cdot \frac{m^3}{\xi_j} \), where \([J]\) is the physical dimension of the cost function and \([R_j]\) the dimension of the \(j\)-th component of the state equation (17). This gives, for example, \([u] = \frac{m^2}{k^3}\) as the dimension of the adjoint velocity.

The generic cost function (15) has volumetric and surface contributions and its variation \( \delta \xi J \) with respect to the flow field is:

\[ \delta \xi J = \delta \xi J_{\Omega} + \delta \xi J_{\Gamma} = \sum_i \left( \int_{\Omega} \frac{\partial J_{\Omega}}{\partial \xi_i} \delta \xi_i + \int_{\Gamma} \frac{\partial J_{\Gamma}}{\partial \xi_i} \delta \xi_i \right) \]  

The variation of (17) with respect to the flow field is

\[ \delta \xi R = \delta_v R + \delta_p R + \delta_\Phi R \]  

with

\[ \delta_v (R_1, R_2, R_3)^T = (\delta \tilde{v} \cdot \nabla) v + (\tilde{v} \cdot \nabla) \delta v - \nabla \cdot (2 \tilde{v} D (\delta v)) + \tilde{\alpha} \delta v \]
\[ \delta_v R_4 = -\nabla \cdot \delta \tilde{v} \]
\[ \delta_v R_5 = (\delta \tilde{v} \cdot \nabla) \Phi \]

\[ \delta_p (R_1, R_2, R_3)^T = \nabla \delta p \]
\[ \delta_p R_4 = 0 \]
\[ \delta_p R_5 = 0 \]  

\[ \delta_\Phi (R_1, R_2, R_3)^T = 0 \]
\[ \delta_\Phi R_4 = 0 \]
\[ \delta_\Phi R_5 = (\tilde{v} \cdot \nabla) \delta \Phi - \nabla \cdot \left( \tilde{\Gamma} \nabla \delta \Phi \right) \]
Placing (19) and (21) into (18) yields:

$$\delta_\xi L \equiv 0$$

$$= \int_\Omega \frac{\partial J_0}{\partial \xi} \delta \xi + \int_\Gamma \frac{\partial J_1}{\partial \xi} \delta \xi + \int_\Omega \Psi \cdot \delta \xi R$$

$$= \int_\Omega \frac{\partial J_0}{\partial \xi} \delta v + \int_\Gamma \frac{\partial J_1}{\partial \xi} \delta v + \int_\Omega \left( \begin{array}{c} u \\ q \\ \varphi \end{array} \right) \cdot \left( \begin{array}{c} \nabla \delta v + (v \cdot \nabla)\delta v - \nabla \cdot (2\tilde{\nu} D(\delta v)) + \tilde{\nu} \delta v \\ -\nabla \cdot \delta \tilde{v} \\ \delta \cdot \nabla \Phi \end{array} \right)$$

$$+ \left( \begin{array}{c} u \\ q \\ \varphi \end{array} \right) \cdot \left( \begin{array}{c} \nabla \delta p \\ 0 \\ 0 \end{array} \right)$$

$$+ \int_\Omega \frac{\partial J_0}{\partial \xi} \delta \Phi + \int_\Gamma \frac{\partial J_1}{\partial \xi} \delta \Phi + \int_\Omega \left( \begin{array}{c} u \\ q \\ \varphi \end{array} \right) \cdot \left( \begin{array}{c} 0 \\ 0 \\ (\tilde{v} \cdot \nabla) \delta \Phi - \nabla \cdot (\tilde{\Gamma} \nabla \delta \Phi) \end{array} \right)$$

(22)

The variations of the flow field $\delta \xi = (\delta v, \delta p, \delta \Phi)^T$ in (22) within the fluid domain $\Omega$ are unknown a priori, but can be separated by partial integration. After some manipulation (see 2.5.3) we obtain from (22) the basis for the adjoint equation system:

$$\delta_\xi L := 0$$

$$\frac{\partial L}{\partial \xi} = \int_\Omega \left( \begin{array}{c} \delta v \\ \delta p \\ \delta \Phi \end{array} \right) \cdot \left( \begin{array}{c} -\nabla \cdot \tilde{v} - (\tilde{v} \cdot \nabla) u - \nabla \cdot (2\tilde{\nu} D(u)) + \rho (\alpha u + \nabla q - \Phi \nabla \varphi) + \frac{\partial J_0}{\partial \nu} \\ -\nabla \cdot u + \frac{\partial J_1}{\partial \nu} \\ -(\tilde{v} \cdot \nabla) \Phi + \frac{\partial J_1}{\partial \Phi} \end{array} \right)$$

adjoint flow equations $A(\Psi)$

$$+ \int_\Gamma \left( \begin{array}{c} \delta v \\ \delta p \\ \delta \Phi \end{array} \right) \cdot \left( \begin{array}{c} n(u \cdot \tilde{v}) + u(\tilde{v} \cdot n) + 2\tilde{\nu} n \cdot D(u) + \rho (\Phi n - q n) + \frac{\partial J_0}{\partial n} \\ u \cdot n + \frac{\partial J_1}{\partial n} \\ \varphi (\tilde{v} \cdot n) + \tilde{\Gamma} n \cdot \nabla \varphi + \frac{\partial J_1}{\partial \Phi} \end{array} \right)$$

adj. BC1

$$+ \int_\Gamma \left( \begin{array}{c} u \\ q \\ \varphi \end{array} \right) \cdot \left( \begin{array}{c} -2\tilde{\nu} n \cdot D(\delta v) \\ 0 \\ -\tilde{\Gamma} n \cdot \nabla (\delta \Phi) \end{array} \right)$$

adj. BC2

Equation (23) can be fulfilled for arbitrary variations of the flow field $\delta \xi = (\delta v, \delta p, \delta \Phi)^T$ by solving the adjoint flow equations $A(\Psi) = 0$ along with the adjoint boundary conditions $(BC1 + BC2 = 0)$. 

11
2.5.3 Notes on the partial integration of (22)

For completeness, we provide some details for the partial integration of (22). The following mathematical rules:

- Gauss’s theorem, to convert volume integral to surface integral
  \[ \int_{\Omega} \nabla a = \int_{\Gamma} a \, n \quad \int_{\Omega} \nabla \cdot a = \int_{\Gamma} a \cdot n \]

- chain rule
  \[ \nabla (ab) = a \nabla b + b \nabla a \quad \Rightarrow \quad a \nabla b = \nabla (ab) - b \nabla a \]

- partial integration
  \[ \int_{\Omega} a \left( \nabla \cdot \mathbf{b} \right) = \int_{\Omega} \nabla (ab) - \int_{\Omega} b \cdot \nabla a \quad \Rightarrow \quad \int_{\Omega} \nabla (ab) = \int_{\Omega} a \left( \nabla \cdot \mathbf{b} \right) - \int_{\Omega} b \cdot \nabla a \]

lead to specific rules used for the partial integration of (22):

\[ a) \quad \int_{\Omega} a \left( \nabla \cdot \mathbf{b} \right) = \int_{\Omega} \nabla (ab) - \int_{\Omega} b \cdot \nabla a \quad \Rightarrow \quad \int_{\Omega} a \left( \nabla \cdot \mathbf{b} \right) = \int_{\Omega} b \cdot \mathbf{a} - \int_{\Omega} b \cdot \nabla a \]
\[ b) \quad \int_{\Omega} a \cdot (\nabla b) = \int_{\Omega} \nabla (ab) - \int_{\Omega} b \cdot \nabla a \quad \Rightarrow \quad \int_{\Omega} a \cdot (\nabla b) = \int_{\Omega} b \cdot \mathbf{a} - \int_{\Omega} b \cdot \nabla a \]
\[ c) \quad \int_{\Omega} a \cdot (\nabla b) = \int_{\Omega} \nabla (ab) - \int_{\Omega} b \cdot \nabla a \quad \Rightarrow \quad \int_{\Omega} a \cdot (\nabla b) = \int_{\Omega} b \cdot \mathbf{a} - \int_{\Omega} b \cdot \nabla a \]

Using these rules, together with some more complex relations derived by Othmer[8], which are not shown here (indicated with *), we obtain for the different terms of (22):

\[ \int_{\Omega} \mathbf{u} \cdot ((\delta \nabla \cdot \nabla) \mathbf{v}) \]
\[ \int_{\Omega} \mathbf{u} \cdot ((\nabla \cdot \nabla) \delta \nabla) \]
\[ \int_{\Omega} \mathbf{u} \cdot (-\nabla \cdot (2\delta D(\delta \nabla))) \]
\[ \int_{\Omega} \mathbf{q} \cdot (-\nabla \cdot \delta \nabla) \]
\[ \int_{\Omega} \varphi ((\delta \nabla \cdot \nabla) \Phi) \]
\[ \int_{\Omega} \mathbf{u} \cdot (\nabla \delta p) \]
\[ \int_{\Omega} \varphi ((\nabla \cdot \nabla) \delta \Phi) \]
\[ \int_{\Omega} \varphi (-\nabla \cdot (\Gamma \nabla \delta \Phi)) \]

Using these rules, together with some more complex relations derived by Othmer[8], which are not shown here (indicated with *), we obtain for the different terms of (22):
2.6 Continuous adjoint equation system

The adjoint equation system $A(\Psi) = 0$, which originates from (23) is summarized below:

- adjoint momentum eqn.

\[
- \rho \mathbf{v} \cdot 2D (\mathbf{u}) = -\rho \nabla q + \nabla \cdot (2\rho \nu D (\mathbf{u})) - \alpha \rho \mathbf{u} + \rho \Phi \nabla \varphi - \frac{\partial J_\Omega}{\partial \mathbf{v}}
\]

- upstream convection

- pressure

- diffusion

- friction

- scalar

- cost function

- adjoint continuity eqn.

\[
\nabla \cdot \mathbf{u} = \frac{\partial J_\Omega}{\partial p}
\]

- cost function

- adjoint scalar transport

\[
- \rho \mathbf{v} \cdot \nabla \varphi = \nabla \cdot (\rho \Gamma \nabla \varphi) - \frac{\partial J_\Omega}{\partial \Phi}
\]

- upstream convection

- diffusion

- cost function

The adjoint equation system is very similar to the primal NSE system. The main difference being that the adjoint convection is upstream to the primal flow, and there are volumetric sources $\partial_\xi J_\Omega$, if there is a volumetric contribution $J_\Omega$ to the cost function.

Our OpenFOAM-based adjoint solver uses the same numerical method for the adjoint flow as for the primal flow, but some adaptations in the convection scheme are applied to improve stability and convergence. For applications involving species transport, using non-diffusive convection schemes proved to be important. Automatically generated polyhedral meshes were used, in conjunction with OpenFOAM’s limitedLinear scheme (2nd order central with TVD limiter).

2.6.1 Adjoint boundary conditions

The boundary conditions of $A(\Psi) = 0$ depend on the surface contribution $J_\Gamma$ of the cost function. The derivation of the adjoint boundary conditions is shown in [3].

- For inlets and walls with $\delta \mathbf{v} = \mathbf{0}$, $\delta p \neq 0$ and $\partial_n \delta \Phi = 0$ (for adiabatic walls) the adjoint boundary conditions\(^2\) are fulfilled with $u_n = -\partial_p J_\Gamma$ and $\mathbf{u}_t = \mathbf{0}$. Neumann boundaries $\partial_n q = 0$, $\partial_n \varphi = 0$ can be applied for the adjoint pressure and adjoint scalar.

\(^2\)The condition $\mathbf{u}_t = \mathbf{0}$ follows from BC2, where $\nabla \cdot \delta \mathbf{v} = 0$ leads to $2\tilde{v} \mathbf{n} \cdot D (\delta \mathbf{v}) = \tilde{v} \partial_n \delta \mathbf{v}_t \neq \mathbf{0}$. $u_n = \mathbf{u} \cdot \mathbf{n}$ is the surface normal adjoint velocity component and $\mathbf{u}_t = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}) \mathbf{n}$ the surface tangential adjoint velocity vector.
• At an outlet (pressure boundary and Neumann condition \( \partial_n \xi = 0 \)) is \( \delta v \neq 0 \), \( \delta p = 0 \) and \( \partial_n \delta \xi = 0 \) and BC2 is fulfilled. From BC1 follows \( \varphi = -\partial_n J_T/v_n \) for the adjoint scalar, and \( q = u \cdot \tilde{v} + u_n \tilde{v}_n + \tilde{v} \partial_n u_n + \varphi \tilde{\Phi} + \partial_n J_T \) for the adjoint pressure, and \( 0 = \tilde{v}_n u_t + \tilde{v} \partial_n u_t + \partial_n J_T \) for the tangential adjoint velocity.

In a multi-objective optimization, an adjoint system can be solved for each individual cost function, or a single adjoint system can be solved if the cost functions are first combined into a single global cost function \( J = \sum_i w_i J_i \) with individual weighting factors \( w_i \). If scalar transport is unimportant for the cost function, setting \( \varphi = 0 \) leads directly to a simpler adjoint equation system.

### 2.7 Computation of sensitivities

#### 2.7.1 Volumetric sensitivity

A volumetric sensitivity follows easily from (8) by using the porosity \( \alpha \), which can be viewed as a type of “virtual sand” for the design variable \( (c \equiv \alpha) \). The usual cost functions do not depend directly on \( \alpha \), and thus \( \delta \alpha J = 0 \):

\[
\delta \alpha L = \partial \alpha L \cdot \delta \alpha = \delta \alpha J + \int_\Omega \Psi \cdot \delta \alpha R = 0 + \int_\Omega \begin{pmatrix} u \\ q \\ \varphi \end{pmatrix} \cdot \begin{pmatrix} \tilde{v} \delta \alpha \\ 0 \\ 0 \end{pmatrix} = \int_\Omega u \cdot \tilde{v} \delta \alpha \\
\Rightarrow \partial \alpha L = u \cdot \tilde{v}
\]

\[ (24) \]

Generic volumetric sensitivity

In addition to the ‘virtual’ friction term \( \tilde{\alpha} v \) in the momentum equation, a generic sink \( \gamma_i \xi_i \) could also be considered, which leads to a generic volumetric sensitivity \( \partial \gamma_i L_i = \Psi_i \xi_i \) that indicates how the cost function is influenced by local momentum, mass or scalar sources within the flow domain. This valuable information can be used to modify the design manually. For example, a momentum sink could be introduced by blocking a part of the flow domain. Or if an region is sensitive to a scalar source, flow with a higher scalar concentration could be guided into that area via baffles or by relocating a scalar injection point (e.g., fuel vaporizer application).

#### 2.7.2 Surface sensitivities

The starting point for the computation of surface sensitivities \( \partial \beta L \) is (10) with \( c \equiv \beta \):

\[
\delta J = \delta \beta L \cdot \delta \beta = \delta \beta J - \int_\Omega \Psi \cdot \delta \xi R
\]

\[ (25) \]
From the relation (7) and the adjoint equation system (23) with \( A(\Psi) = 0 \) and \( BC_2 = 0 \) follows:

\[
\delta_\beta L = \delta_\beta J - \int_\Omega \Psi \cdot \delta_\xi R = \delta_\beta J + \delta_\xi J - \delta_\xi L \tag{26}
\]

\[
= \delta_\beta J + \left( \int_\Omega \delta_\xi \cdot \partial_\xi J_\Omega + \int_\Gamma \delta_\xi \cdot \partial_\xi J_\Gamma \right) - \delta_\xi L \tag{23}
\]

\[
(23) \delta_\beta \cdot \partial_\beta J + \int_\Omega \delta_\xi \cdot \partial_\xi J_\Omega - \int_\Gamma \delta_\xi \cdot \left( \frac{u \cdot \tilde{v}}{\varphi(\tilde{v} \cdot n) + \Gamma n \cdot \nabla \varphi} \right)\]

\[
(= - \partial_\xi J_\Gamma)
\]

The variations \( \delta_\xi \) of the flow field with respect to a geometry variation are unknown within the flow domain. At walls they are typically \( \delta_\xi = 0 \). Equation (26) simply reflects the definition of \( \delta J \), since \( \delta_\xi L = 0 \). To compute the local surface sensitivities \( \partial_\beta L \), the flow field has to be held constant, since (26) was derived from (8), in which (with \( c \equiv \beta \)) only the partial derivative \( \partial_\beta \) appears. Thus \( \delta_\xi = \delta_\beta \xi = 0 \) within the whole flow domain \( \Omega \), except at the boundary \( \Gamma \), where a local wall normal displacement of \( \delta_\beta \xi \) causes a local variation \( \delta_\beta \xi \). According to [3] and [9], the flow field can be approximated by a Taylor series \( \xi(\beta) = \xi(0) + \partial_n \xi \beta + O(\beta^2) \), where \( \beta \) is the wall distance (in direction of the outward normal vector \( n \)) and \( \partial_n \) is the wall normal gradient operator, and \( \delta_\xi \) can thus be written as

\[
\delta_\xi \equiv \delta_\beta \xi = \partial_\beta \xi \cdot \delta_\beta \approx \partial_n \xi \cdot \delta_\beta \tag{27}
\]

which can be inserted into (26) to yield locally\(^3\):

\[
\partial_\beta L = \partial_\beta J - \partial_\beta \xi \cdot \left( \frac{u \cdot \tilde{v}}{\varphi(\tilde{v} \cdot n) + \Gamma n \cdot \nabla \varphi} \right) \tag{28}
\]

By considering the no-slip condition at the wall\(^4\), combined with the adjoint boundary conditions, the local surface sensitivity to a surface normal displacement can be derived [3]:

\[
\partial_\beta L = -\tilde{v} \partial_n u_t \partial_n v_t - \tilde{\Gamma} \partial_\beta \Phi \partial_n \varphi \tag{29}
\]

where \( u_t = u - (u \cdot n) n \) and \( v_t \) are the surface tangential velocity components. Equation (29) is only valid within regions that are not part of the definition of \( J \), otherwise the term \( \partial_\beta J \) has to be taken into account. For adiabatic walls \( \partial_\beta \Phi = 0 \) and therefore

\[
\partial_\beta L = -\tilde{v} \partial_n u_t \partial_n v_t \tag{30}
\]

\(^3\)the result can also be derived directly by partial integration of (8), with \( c \equiv \beta \), by applying the wall boundary condition at the displaced wall to gain \( \delta_\beta R = \delta_\beta R \cdot \delta_\beta \approx \delta_\xi R \partial_\beta \xi \cdot \delta_\beta = -\partial_n \xi \partial_\xi R \cdot \delta_\beta \).

\(^4\)At the wall: \( \xi_1 = v = 0, \nabla \cdot v = 0, \partial_n v_n = 0, \partial_n p = 0 \) and \( n \cdot 2D(u) = (n \cdot \nabla) u_t = \partial_n u_t \).
3 CONCLUDING REMARKS

An industrial CFD optimization workflow for exhaust systems has been presented in the paper that utilizes two adjoint CFD solvers. The continuous adjoint geometry optimization tool (CAGO) is an innovative form optimization tool — useful in the early stages of the design process to get an design proposal for a given package space – that provides a design proposal, which can be used as a reference during CAD design. Further optimization of CAD geometries use a fully compressible CFD solution, in combination with a pseudo-compressible continuous adjoint method, which has also been presented here in detail. Although the frozen turbulence and frozen density approximations reduce the absolute precision of the sensitivities, this adjoint method provides valuable insight to guide the manual geometry adjustment, which also needs to consider production processes and other constraints such as thermal behaviour, durability and product costs. The focus of the next developments lies on increasing the level of automation in this CAD-based adjustment process. As demonstrated by [10], the computed sensitivities can also be used to adjust CAD shape parameters directly.

REFERENCES


