LOCALIZED AERODYNAMIC SHAPE OPTIMIZATION FOR AUTOMOTIVE APPLICATIONS

Maurice Nayman[1] & Ruben E. Perez[2]

[1]Master's Student, Department of Mechanical and Aerospace Engineering, Royal Military College of Canada, Kingston, ON, K7K 7B4, Canada
[2]Professor, Department of Mechanical and Aerospace Engineering, Royal Military College of Canada, Kingston, ON, K7K 7B4, Canada

Abstract

This paper investigates the use of RANS-based unconstrained gradient-based optimization to reduce drag in automotive applications. Three optimization algorithms are tested around the roof spoiler and front fascia regions of the *DrivAer Estateback* to evaluate the efficacy of the different algorithms in regions with differing flow physics. Two variants of the *Estateback* were used, defined previously by the authors. These were the *base* Estateback and an *improved* variant, which had been previously modified by the authors to have less aerodynamic drag using qualitative gradient-based methods. Drag optimization of the *Estateback* was most successful around the roof spoiler, achieving up to 7%. Optimization around the front fascia showed no measurable improvement, due to the complex wake features around the front tires not being fully resolved by the RANS solver. DDES post-optimality CFD was used to confirm the results predicted by the optimization algorithms, which showed the RANS optimizers were most accurate when applied to the *improved Estateback* variant.

Keywords: Automotive Aerodynamics, Aerodynamic Shape Optimization, Continuous Adjoints

Nomenclature

α_K	Scaling value of the search direction to compute the new design vector
ε_a	Absolute objective function convergence tolerance
ε_g	Gradient convergence tolerance
ε_r	Relative objective function convergence tolerance
$\mathbf{p_K}$	Search direction for a given outer iteration
$\mathbf{x_K}$	Design variable vector at major iteration K
$\mathbf{x}^*$	Design vector at which the objective function is minimized
∇J	Gradient of function f
D	Aerodynamic drag [N]
$H(\mathbf{x})$	Hessian matrix
J	Adjoint objective function

l Car wheelbase [mm]

U_∞ Freestream velocity [m/s]

V_K Approximate inverse hessian matrix

x Streamwise location [mm]

y Spanwise location [mm]

z Vertical location [mm]

K Number of optimization loops

k Equivalent total number of CFD solutions

Dimensionless Groups

C_D Drag Coefficient, $\frac{D}{0.5\rho_\infty U_\infty^2 A}$

Acronyms

CFD Computational Fluid Dynamics

MRF Moving Reference Frame

RANS Reynolds Averaged Navier Stokes

SA Spalart Allmaras

SIMPLE Semi-Implicit Method for Pressure Linked Equations

1. Introduction

With the looming climate crisis, many sectors of engineering have worked towards improving efficiency, reducing emissions, and creating more effective design to limit global carbon dioxide emissions. This has included the use of electrification, new sustainable fuels, and a greater focus on aerodynamic drag in the aerospace and automotive industries. With passenger vehicles making up 41% of emissions in the transportation sector [[1]], electrification has become more common, with the attempt to divest from fossil fuels.

With electric vehicles benefiting from greater efficiencies in their drivetrains, aerodynamic drag has become even more of a concern. When Porsche developed their Taycan electric model, they found that aerodynamic drag comprised 29% of the car's total losses, compared to roughly 8% for their internal combustion vehicles [2]. As such, minimizing aerodynamic drag can be four times as important for electric vehicles compared to internal combustion vehicles. This gives rise for a need in effective optimization methods, which can be used in key stages of a car's design cycle to improve its drag.

The automotive industry differs from aerospace, in that the exterior design of a car is largely form driven, rather than function driven. Manufacturers will conceptualize a design language to be expressed with the vehicle, with aerodynamic targets not necessarily driving that language. Therefore, it is not uncommon for aerodynamic sacrifices to be made, in order to achieve a desired aesthetic. Therefore, traditional multi-disciplinary optimization that incorporates the exterior styling can be an

inefficient use of engineering efforts, as it is likely that it would not meet the styling requirements of the vehicle. Therefore, optimization methods may not be feasible until the end of the design cycle, for final improvements. This use of gradient-based optimizers should therefore be efficient, and be able to be applied to certain areas of the car with either low styling concern, or noticeable design freedom.

Aerodynamic shape optimization has been investigated in the automotive industry, comprising of gradient-based and gradient-free optimizers to improve certain regions of a car[3, 4, 5, 6, 7, 8]. The majority of recent work has employed gradient-based optimization. This has been driven by the continuous adjoint method for the calculation of the sensitivities, which is an efficient sensitivity analysis method in the presence of many design variables. The key contributor to this work in the open-source community is the National Technical University of Athens (NTUA) Parallel Computing and Optimization Unit (PCOpt), who developed the continuous solver for OpenFOAM [[9]], adjointOptimisationFoam [10]. Numerous investigations have been performed, such as the optimization of a Volkswagen iD3's rear spoiler, which showed drag improvements around 5%, performed by Papoutsis et al [7]. Further studies have also investigated the effects of multiple operating conditions on the minimization of multi-objective optimization problems for a car subject to a cross-wind [6].

What these works do not quantify is the effect of different unconstrained gradient-based optimization methods on the quality of the results. Since industrial automotive flows are particularly complex due to interaction wakes, rotation wheels, and large regions of separation; there tends to be poorer numerical convergence of the primal solutions [11]. In turn, this introduces inaccuracies and difficulties in converging the adjoint equations. Therefore, certain optimization methods may perform more effectively for automotive purposes, specifically those that yield smaller displacements at each optimization iteration. Furthermore, much of the existing literature has focused on optimizing the shape of the rear decklid of automotive geometries [12, 7, 3]. These do not consider regions with more complex flow physics that will be more sensitive to the primal solution's convergence, such as around the front wheels. This area is of particular importance for drag reduction and as such, it is valuable to quantify the effectiveness of different optimization methods in these areas of a car.

This paper acts as an extension of previous work by the authors using gradient-based methods to improve the aerodynamic design of passenger cars [13]. This is meant to serve as the final step in a typical automotive design routine, where improvements would be made to the vehicle through expert-informed decisions, at which point optimization methods can be used to extract the last portion of performance from the vehicle. This paper therefore serves as the final component in a more unified automotive design process, making effective use of gradient-based methods to improve engineering efficiency. In doing so, this paper seeks to determine how important expert-driven modifications are in the drag reduction process of a car. It is unlikely that computer-driven optimization alone is sufficient to achieve the full potential of a car, especially in the presence of styling constraints that cannot be quantified within an optimization routine. All of this is attempted to be met within CFD software typically used by the automotive industry.

Furthermore, this paper provides an examination of the role the optimization algorithm, and initial design vector play for a highly-detailed automotive geometry. Previous work has investigated these effects for automotive bodies, but these have featured simplified geometries, without complex wake structures arising from the tires [12]. Part of the present work will be the use of higher fidelity post-optimality CFD simulations to confirm the improvement prediction accuracy of the optimizers, which has not been previously presented for automotive applications. This will provide quantification of previously undocumented optimizer performance across a range of initial design vectors pertinent to the automotive industry.

The use of unconstrained gradient-based optimization is applied to the *DrivAer Estateback* model, to evaluate its efficacy in industrial applications valuable to the automotive industry. This optimization will be performed on both the baseline *Estateback* model and the improved *Estateback* model previously described by the authors [13]. The continuous adjoint formulation, optimization methods, and optimization problem are presented in Section 2. The application of gradient-based optimization to the two *Estateback* variants using three different step update methods is presented in Section 3. Additionally, this section will also present post-optimality computational fluid dynamics (CFD) simulations to verify the improvement predicted by the optimization algorithms. The conclusions reached and discussion of future work is presented in Section 4.

2. Methodology

This section presents the definition and formulation of the optimizers and their use within OpenFOAM, as well as the test-case and numerical setup that will be used to apply the optimization routines. Finally, the specific optimization problem will be defined.

2.1 Gradient-Based Optimizers

For all gradient-based optimizers, the derivative of the objective function is used to define a search direction to either maximize or minimize the objective function, J. For CFD solutions, the most efficient method with which the Jacobian and Hessian can be calculated is the adjoint method. Within the OpenFOAM [[9]] framework used, the continuous adjoint method has been implemented within its continuous adjoint solver, adjointOptimisationFoam, developed by the National Technical University of Athens [10]. The governing equations for the continuous adjoint method have been previously described by the authors [[13]], and as such, will not be detailed here.

Unconstrained gradient-based optimizers in *adjointOptimisationFoam* are applied in the following steps, with an initial major iteration, K = 0, and an initial design vector: $\mathbf{x}_K$:

1. Convergence is tested, by evaluating the objective function value at the current design vector, $\mathbf{x}_K$, and comparing it to the previous major iteration's objective function value (if available). If convergence criteria are met, the solution is stopped.
2. A search direction, $\mathbf{p}_K$, is computed using gradient information from the objective function. A number of search direction methods can be chosen to compute this vector.
3. Compute the step length to find a positive scalar, α_K, such that the *Armijo* conditions are met as implemented in *adjointOptimisationFoam* [14].
4. The design vector is updated such that, $\mathbf{x}_{K+1} = \mathbf{x}_K + \alpha_K \mathbf{p}_K$.

   K = K + 1, and steps 1 - 4 are repeated if K is less than the maximum number of desired iterations.

As mentioned, *adjointOptimisationFoam* employs the *Armijo* condition for the line search, which is an inexact line search method. It is formulated such that the line search converges when Condition 1 is satisfied:

$$J\left(\mathbf{x_K} + \alpha_K \mathbf{p_K}\right) \leq \mu_1 \alpha_K \nabla J\left(\mathbf{x_K}\right)^T \mathbf{p_K} \tag{1}$$

In Condition 1, μ_1 is a small scaling factor, which is 10^{-4} by default in *adjointOptimisationFoam* [14]. This ensured that some minimum improvement is required for the line search to have converged. It should be noted that the line search in *adjointOptimisationFoam* can be limited to a certain number of iterations, which could result in line search termination before an improvement is found. As such, this implementation within *adjointOptimisationFoam* was not guaranteed to result in an improvement in the objective function at each major optimization iteration. For all optimization simulations in this work, the line search was allowed to run for up to five function evaluations before terminating, to balance computational expense with finding an improved solution.

Since the line search method implemented does not make use of any curvature conditions, each subsequent iteration of the line search is computed through backtracking by the following method:

1. The initial step length $\bar{\alpha}$ is set to one, and some reduction ratio, $0 < \rho < 1$ is chosen
2. If the *Armijo* condition outlined in Equation 1 is met, $\alpha_K = \bar{\alpha}$, and the line search is terminated
3. If the condition is not met, set $\bar{\alpha} = \rho\bar{\alpha}$ and repeat step two until convergence is met, or the maximum number of line search iterations is reached

In this paper, three search direction methods are explored: Conjugate Gradient (CG), the Broyden-Fletcher-Goldfarb-Shanno (BFGS), and the Damped-BFGS (DBFGS) algorithm.

2.1.1 Conjugate Gradient

The Conjugate Gradient method is a modification to a steepest descent method, whereby the gradient search direction is *pushed* towards the solution more rapidly by using previous optimization iterations' gradient information [15]. It can be formulated by assuming that the objective function space can be represented by a quadratic function:

$$J(x) = \frac{1}{2}x^T Qx - x^T b, Q = Q^T \tag{2}$$

Therefore, the gradient of the objective function becomes:

$$\nabla J = Qx - b \tag{3}$$

Using this function and its gradient, when applying the typical steepest descent search direction, which is the negative gradient, the search direction for a Conjugate Gradient problem can be formulated as:

$$\mathbf{p}_{K+1} = -\nabla J(\mathbf{x}_{K+1}) + \beta_K \mathbf{p}_K \tag{4}$$

The computation of β_K requires it to be Q-conjugate at every optimization iteration, K. To negate the need to compute Q, so the search direction is solely dependent on the gradient of the objective function, the Fletcher-Reeves approximation is applied. This approximation is given by [16]:

$$\beta_K = -\frac{\nabla J(\mathbf{x}_{K+1})^T \nabla J(\mathbf{x}_{K+1})}{\nabla J(\mathbf{x}_K)^T \nabla J(\mathbf{x}_K)} \tag{5}$$

This allows for curvature information to be approximated, without the need for computing the Q matrix.

2.1.2 BFGS

The BFGS method is a quasi-Newton method, whereby the Hessian matrix is approximated, to try and represent the curvature of the design space [16]. Since the inverse Hessian matrix is what will actually be needed, instead of solving for $H_K(\mathbf{x})$, the inverse Hessian, $V_K(\mathbf{x})$ is directly solved for at each major iteration, K. Assuming that a Taylor-series expansion of the gradient of a function can be written as:

$$\nabla J(\mathbf{x}_K + \mathbf{s}_K) = \nabla J(\mathbf{x}_K) + H_K \mathbf{s}_K + \ldots \tag{6}$$

In Equation 6, $\mathbf{s}_K = \alpha_K \mathbf{p}_K$, and $\mathbf{y}_K = \nabla J(\mathbf{x}_K + \mathbf{s}_K) - \nabla J(\mathbf{x}_K)$. The Hessian can then be given by:

$$H_K \mathbf{s}_K = \mathbf{y}_K \tag{7}$$

Which then places a requirement for the inverse Hessian to meet the following criterion:

$$V_{K+1}\mathbf{y}_K = \mathbf{s}_K \tag{8}$$

Using this information, the inverse Hessian matrix is solved for by attempting to minimize the norm of the difference between the inverse Hessian at iteration $K+1$ and the inverse Hessian at iteration K. The inverse Hessian is then solved using the following equation:

$$V_{K+1} = \left[I - \frac{\mathbf{s}_K \mathbf{y}_K{}^T}{\mathbf{s}_K{}^T \mathbf{y}_K}\right] V_K \left[I - \frac{\mathbf{y}_K \mathbf{s}_K{}^T}{\mathbf{s}_K{}^T \mathbf{y}_K}\right] + \frac{\mathbf{s}_K \mathbf{s}_K{}^T}{\mathbf{y}_K{}^T \mathbf{s}_K} \tag{9}$$

Using this approximation for the inverse Hessian matrix, the search direction for a BFGS problem can then be defined by:

$$\mathbf{p}_K = -V_K \nabla J(\mathbf{x}_K) \tag{10}$$

By using the approximated Hessian matrix, the BFGS algorithm leverages curvature information about the design space, leading to larger steps across the objective function topology. This makes the accuracy of BFGS dependent on how well-behaved the objective function's topology is, as well as how accurately the sensitivity derivatives can be computed, since it directly impacts the inverse Hessian calculation.

2.1.3 DBFGS

The Damped BFGS algorithm was developed to act as a more stable version of the BFGS algorithm [17]. It is similar to the presented BFGS algorithm, except it is derived from a BFGS algorithm that computes the Hessian directly and takes its inverse, rather than solving for the inverse Hessian problem that BFGS does. This makes the DBFGS algorithm more akin to the Davidon-Fletcher-Powell (DFP) algorithm, which also directly computes the Hessian, then inverts it to compute the new search direction. The damping of the Hessian is performed to maintain the positive-definiteness of the matrix, and occurs by examining the net curvature at the current design vector. For some user-defined scalar γ, if the following inequality is met, the solution will proceed in its damped form:

$$\mathbf{s}_K \mathbf{y}_K{}^T < \gamma \mathbf{s}_K{}^T H_K \mathbf{s}_K \tag{11}$$

Whether the solution is in its damped form or not, the solution of the Hessian matrix takes the same form as the DFP algorithm, being:

$$H_{K+1} = H_K - \frac{H_K \mathbf{s}_K \mathbf{s}_K{}^T H_K}{\mathbf{s}_K{}^T H_K \mathbf{s}_K} + \frac{\mathbf{r}_K \mathbf{r}_K{}^T}{\mathbf{s}_K{}^T \mathbf{r}_K} \tag{12}$$

The $\mathbf{r}_K$ in Equation 12 is a damped version version of the change in the gradient, $\mathbf{y}_K$. This damped term is computed by the following equation:

$$\mathbf{r}_K = \theta \mathbf{y}_K + (1-\theta)\mathbf{s}_K H_K \tag{13}$$

When the inequality shown in Equation 11 is not met, θ is equal to unity, returning a typical DFP solution. When the inequality is met, θ is given by the following:

$$\theta = \frac{(1-\gamma)\mathbf{s}_K{}^T H_K \mathbf{s}_K}{\mathbf{s}_K{}^T H_K \mathbf{s}_K - \mathbf{y}_K{}^T \mathbf{s}_K} \tag{14}$$

Through this process, the DBFGS algorithm helps maintain positive-definiteness of the Hessian matrix, but it still requires the Hessian to be inverted. This can become numerically ill-conditioned, resulting in inaccurate or poorly conditioned search directions. When applied to *adjointOptimisationFoam*, this could lead to extremely large deformations of the geometry, resulting in mesh quality violations. For the DBFGS method, the search direction is calculated as:

$$\mathbf{p}_K = -H_K^{-1} \nabla J(\mathbf{x}_K) \tag{15}$$

2.2 Freeform-Deformation Boxes

For shape optimization problems, freeform deformation boxes (FFD) are an efficient means of defining the design variables for an optimization problem. For a three-dimensional problem, each control point of an FFD has three design variables, for each of its three axes. Therefore, for an FFD with n points, it will comprise $3n$ design variables in three-dimensional space. An example of how an FFD can be used to displace a triangulated surface is shown in Figure 1. A similar approach is implemented in *OpenFOAM*.

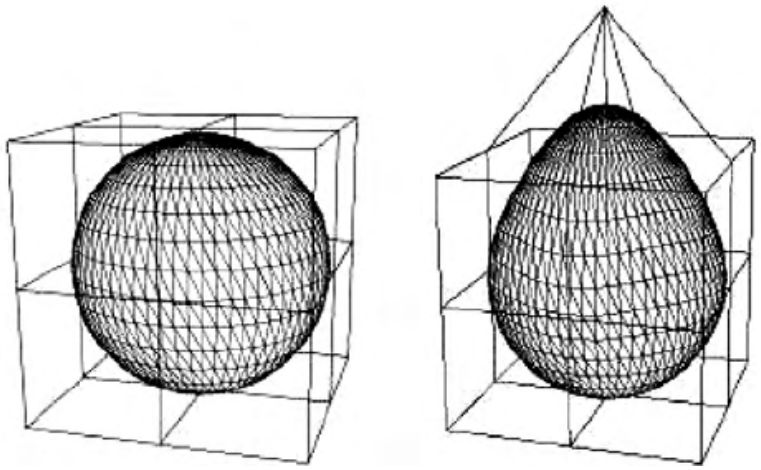


Figure 1 – Example surface displacement that can be achieved with an FFD, from [[18]]

Within *OpenFOAM*, the FFD method is derived from NURBS B-spline radial basis functions [14]. Rather than using a global coordinate system-aligned parameterization for an FFD, a local cartesian coordinate system is created, aligned to be orthogonal to the rectangular box, such that a new coordinate system with axes u, v, w is defined. The FFD is therefore comprised of Cartesian points in x, y, z that can then be parameterized in u, v, w space, with radial basis functions U, V, W. For *OpenFOAM's* implementation of the FFD boxes, the box must be an orthogonal rectangular prism.

The degree of each radial basis function U, V, W can then be described as pu, pv, and pw. For an FFD with a number of points in the u, v, w directions of: $I+1$, $J+1$, and $K+1$, the mth component of a mesh point, x, can be calculated by Equation 16.

$$x_m(u,v,w) = \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K} U_{i,pu}(u)V_{i,pv}(v)W_{i,pw}(w)b_m^{ijk} \tag{16}$$

Using the initial mesh, a mapping is found for each mesh point $\boldsymbol{x}(x, y, z)$ in u, v, w space. Using the known knot vectors $\mathbf{r}$ = (x_r, y_r, z_r), each internal and boundary point of the finite volume mesh that lives within the FFD can be parameterized in u, v, w space by iteratively solving the following system of linear equations:

$$\mathbf{R}(u,v,w) = \begin{bmatrix} x(u,v,w) - x_r = 0 \\ y(u,v,w) - y_r = 0 \\ z(u,v,w) - z_r = 0 \end{bmatrix} \tag{17}$$

This mapping is independent for each mesh point and as such, the B-spline implementation within *OpenFOAM* solves for these parameterization using parallel computations for decomposed problems [14]. At the end of each optimization iteration, displacements are computed for each point within the FFD(s). Once the updated points b_m^{ijk} are known, Equation 16 is used to update all of the mesh points within the FFD. When using *adjointOptimisationFoam* in double-precision, these parameterizations can be calculated to machine precision, allowing for accurate deformations of complex geometries.

When FFD points are used as the design variables of an optimization problem, a search direction will be computed at each point, in each of its three directions. Once a step length is then computed from the line search, each control point is displaced. This displacement is then mapped back to the mesh points within the FFD, which are displaced according to the NURBS radial basis functions.

2.3 Test Case - *DrivAer Estateback* Variants

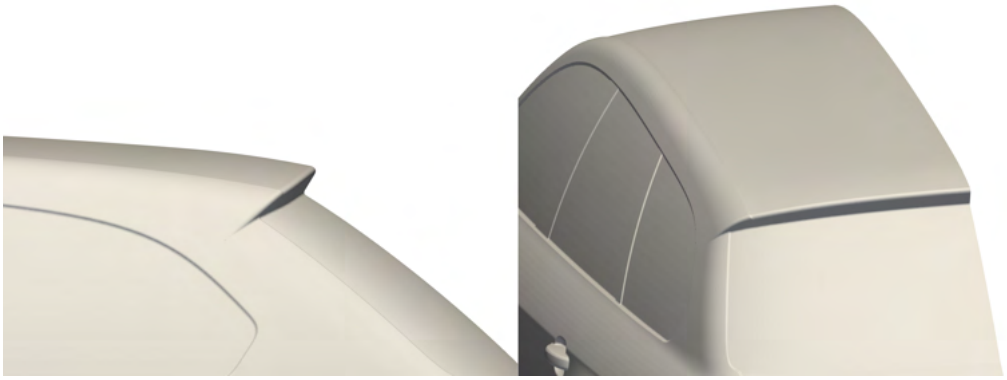
The *DrivAer* geometry is a modular, generic automotive body that was introduced in 2011 [19]. This body was meant to bridge the gap between existing generic bodies such as the Ahmed body [20] or SAE bodies [21]. While these bodies had some more complex features, such as A-pillar flows and rear deck flows, they lacked wheels, wheelwells, and the option for complex underbodies and engine bays. The *DrivAer* model featured all of these components, allowing for more complex flow structures to be modelled, without the need for more representative CAD models from a manufacturer. As such, the authors have used the *Estateback* variant of the *DrivAer* platform to develop continuous adjoint applications for automotive aerodynamic design [13].

For this research, the two variants of the *Estateback* studied are the *base Estateback*, and the improved *Estateback* previously presented by the authors [13]. The *improved* variant was modified using gradient information in the form of the Momentum Contribution Field (MCF) method, previously defined by the authors [13]. This method is a qualitative application of gradient information to provide the designer with information regarding how the local flow momentum increased or decreased the drag of the car. Using this qualitative method, the front bumper, roof spoiler, underbody, and rear fenders were modified to reduce the aerodynamic drag with respect to the *base* variant.

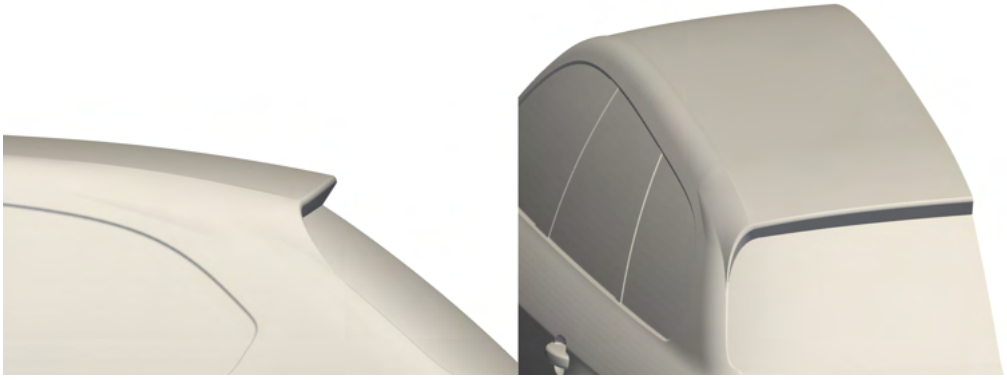
These two variants were chosen to highlight how gradient-based optimization can be applied to a design routine. The *base Estateback* variant represents a car that is closer to the beginning of its development, whereas the improved variant represents a car that is closer to the end of its development. This will be used to quantify if the optimizer yields greater improvements to the car in its base form, as well as what can be achieved when combining expert-driven design with computer-driven optimization.

Two different regions of the car are chosen on which gradient-based optimization will be performed, the front bumper cheeks and the rear spoiler. This decision was made given the difference in flow fields and how accurately the CFD solver can resolve these features. The front bumper cheek flow directly interacts with the front tires, which is a large drag contributor to the car. Predicting these wakes and the upstream flow interactions can be difficult for CFD solvers, so it is of interest to see how the optimizer can perform in around this geometry. For the roof spoiler, while certain CFD methods may under-predict the separation on the rear fascia, the upstream flow is largely attached, meaning the CFD solver should be less error-prone. Furthermore, having a better structure to the wake can yield large drag savings, which is of industrial benefit to investigate. Additionally, these areas can be of lower design studio importance, tending to be less sensitive to styling input, meaning they are better candidates on which optimization would be performed in an industrial setting.

The front bumper cheek and rear spoiler geometries for the *baseline* and *improved Estateback* variants are shown below in Figures 2 and 3. This highlights the key differences in the two geometries in the regions of interest for the optimization. The front bumper cheek in the *improved* variant has an outward flick to outwash air flow around the front wheels, reducing the pressure buildup on the front of the tires, along with a more bluff splitter and front underbody, which was previously described [13]. The rear spoiler on the improved variant has its trailing edge pulled down and rearwards, helping to accelerate the flow on the roof, which has been previously shown to be effective at reducing drag [13]. The geometries are shown for a half-car model, as the vehicle is symmetric, which will be used to reduce the computational cost of simulations.



(a) *Base* Variant



(b) *Improved* Variant

Figure 2 – Roof spoiler geometry

(a) *Base* Variant

(b) *Improved* Variant

Figure 3 – Front bumper cheek geometry

2.4 Numerical Setup

The vehicle is placed in a 153.6 m long, 25.6 m wide, and 25.6 m high computational domain, to create the half-car model about the Y = 0 plane. This domain results in a blockage ratio of 0.16%. A half-car simulation reduces the cell count of the mesh by roughly half, resulting in the desired decrease in computational cost. This is deemed acceptable, as the relative efficacy of the various step update methods was desired, rather than their absolute performance on a full car geometry. It is expected that the efficacy of the different algorithms would scale to a full-car simulation. Meshing methods have been previously described by the authors, with the same approach used as with previous research [13].

When performing an automated optimization loop in adjointOptimisationFoam, a RANS primal solution is performed to compute the primal velocity, pressure, and turbulence fields. For bluff bodies with significant areas of areas of separation, RANS models tend to underpredict the amount of separation along the body, but tend to give reliable changes in results between different designs. Both the primal and adjoint RANS equations are solved using the SIMPLE [22] algorithm, with the RANS equations closed using the Spalart-Allmaras (SA) turbulence model. Averaging of the primal fields and objective function occurs over the final 1000 iterations of each objective function evaluation.

Since adjointOptimisationFoam stores the previous optimization iteration's fields to initialize the current optimization iteration (other than the initial function evaluation), fewer SIMPLE iterations are needed to find a suitable solution for the flow field. As such, the initial function evaluation solves for 2500 iterations, with each subsequent function evaluation solving for 1500 iterations. This allows for a computational speed-up with each objective function evaluation, reducing the total cost of the simulation. The adjoint equations are solved for 1000 iterations every time the continuous adjoint fields are computed.

RANS models can be insufficient for the prediction of automotive aerodynamics. As such, Delayed Detached Eddy Simulations (DDES) are used on select geometries from the various optimization runs to perform post-optimality CFD simulations. The deformed mesh from the optimization runs are used for these simulations, reducing the grid-dependence on the DDES simulations. This is meant to serve as a confirmation of the changes predicted by the optimization solver, and to see if there are any scenarios where the optimizer predicts an increase in drag after the inexact line search, but the DDES simulation predicts a decrease in drag.

The DDES equations are solved using OpenFOAM's pimpleFoam solver, which is a combination of the PISO [23] and SIMPLE algorithms, for time-stepping solutions. The combined PISO-SIMPLE algorithm is abbreviated as the PIMPLE algorithm[24]. The benefit of the PIMPLE algorithm over the standard PISO algorithm is it does not require the maximum Courant number to not exceed one, as it uses SIMPLE-based inner pressure correction iterations to allow for higher Courant numbers at each time step. Second-order spatial and temporal schemes are employed with the PIMPLE algorithm to ensure simulation accuracy. The simulation is allowed to run for five seconds, with averaging of the flow fields and forces occuring over the last two and a half seconds, with time steps of one millisecond. The $k-\omega$ SST DDES turbulence model is used to close the unsteady equations.

2.5 Optimization Problem

For a given FFD around a portion of the geometry, the optimization problem will be formulated as:

$$\begin{aligned} \text{minimize} \quad & C_D \\ \text{w.r.t.} \quad & \mathbf{x} = (x_{11}, x_{12}, x_{13}, ..., x_{n3})^T \end{aligned} \tag{18}$$

In Equation 18, x_{i1} represents the x-component of the ith active design variable, x_{i2} represents the y-component of the ith active design variable, and x_{i3} represents the z-component of the ith active

design variable. When defining a deformation box within OpenFOAM, the user has the ability to deactivate certain design variables within the box. It is typically required to constrain all of the design variables at the boundary of an FFD, which prevents the deformed mesh from moving around points not parameterized by the FFD. Since a symmetry plane is applied to the *DrivAer* geometry, any FFD boxes that pass through the symmetry plane are only constrained in the parameterized V direction for the points not on the rectangular bounds of the box. This allows for motion along the symmetry plane, while constraining plane-normal displacements, representative of a symmetry deformation constraint.

For this problem, two different FFD boxes were evaluated, one around the front bumper cheeks, and one around the rear spoiler. The positions of which are highlighted in Figures 4 - 5. The FFD's are coloured by their activation of the X and Z degrees of freedom (red for active, blue for inactive). This highlights that on the symmetry plane of the vehicle, the spoiler's FFD has a proper symmetry condition, as previously described. The front bumper cheek FFD was made to be relatively wider than the surface geometry to allow for sufficient deformation of the surface, without artificially constraining the results. These boxes were evaluated for each of the two *Estateback* variants, for each of the three step update methods described in Section 22.1

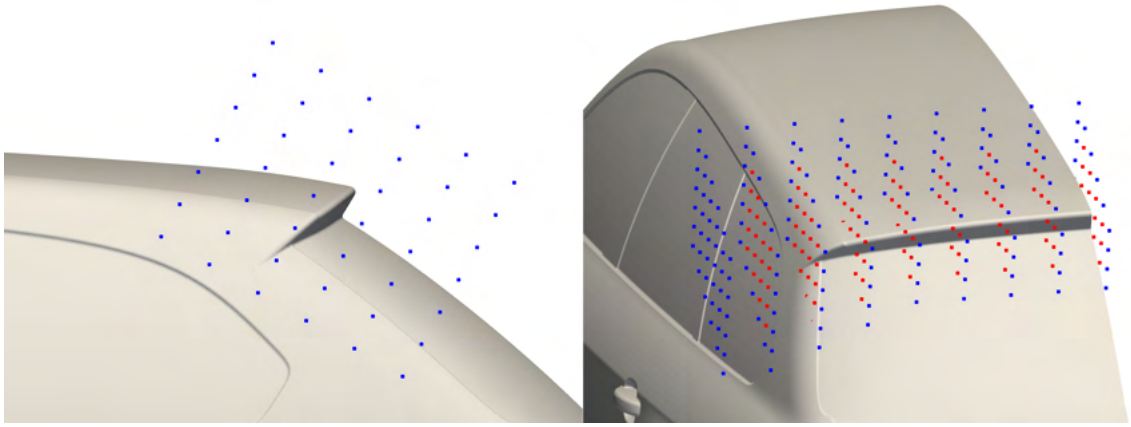


Figure 4 – Freeform deformation box placed around the rear spoiler. The car is shown in its symmetric state to highlight the symmetry plane

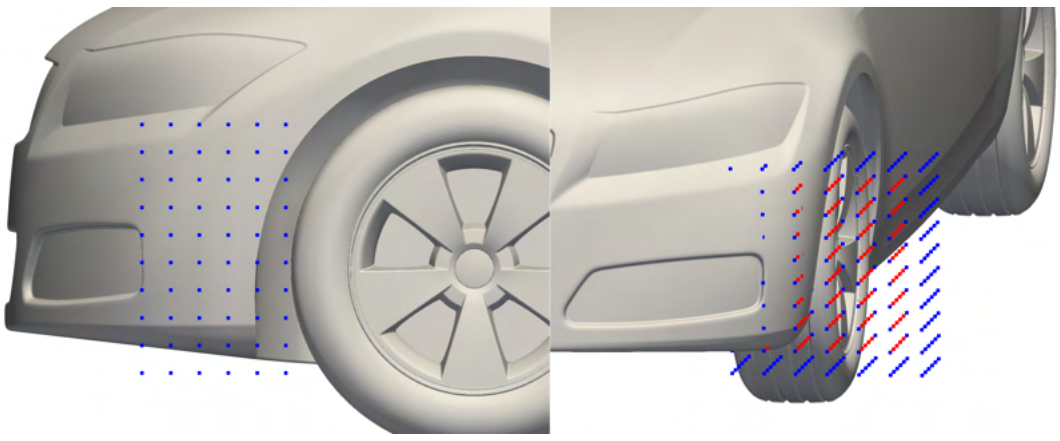


Figure 5 – Freeform deformation box placed around the left front bumper cheek

The number of points in the U, V, W directions as well as the degree of the NURBS radial basis function for each FFD is shown below in Table 1. All directions were defined with a third degree NURBS radial basis function. This ensures that there can be curvature continuity within the deformed surface, while preventing artificial damping of the deformations due to a high-degree radial basis function.

Table 1 – Number of points and degree of the NURBS radial basis function in the U, V, W directions for each FFD

FFD	U Points	V Points	W Points	Degree U	Degree V	Degree W
Rear Spoiler	6	9	7	3	3	3
Front Bumper Cheek	6	7	10	3	3	3

Convergence of the optimization problem is monitored for both the L2-norm of the gradient, as well as the change in objective function value for successive iterations. These criteria are defined as follows:

$$||\nabla J(\mathbf{x_K})|| \leq \varepsilon_g \tag{19}$$

$$|J(\mathbf{x_{K+1}}) - J(\mathbf{x_K})| \leq \varepsilon_a + \varepsilon_r |J(\mathbf{x_K})| \tag{20}$$

The convergence criterion in Equation 20 must be met for two successive before it is considered as having been met. The solver reaching either of these two convergence metrics is sufficient for the optimization loop to be concluded. For all models, ε_g was set to 0.001, ε_a was set to 0.0005, and ε_r was set to 0.0005. Since a half model is employed, this means the relative change in objective function should be C_DA = 0.001 + 0.1% of the previous iteration's value. This limit was placed as drag improvements of less than 0.001 C_D are typically considered to be within the error of the CFD simulation, for automotive purposes.

In some cases, poor numerical conditioning of the gradient at some control points of the FFD results in extremely large deformations of the mesh, violating the mesh quality criteria. This can stem from a number of reasons. Some of the control points can live within the geometry, therefore not being a part of the active flow region. Mapping the external flow solution to these points can then result in large gradients, due to large flow gradients that are present around the mesh points that influence the control point(s). Furthermore, scaling of the design variables only occurs at the first optimization iteration. For complex industrial problems, not scaling the design variables more frequently can yield large displacements, for which the local search direction is an inaccurate approximation over such a large step. Additionally, the poor conditioning of the Jacobian can be caused by the aforementioned numerical convergence issues of both the primal and adjoint solvers. In previous work by the authors, inaccuracies in the gradients were not necessarily identified, since the adjoint information was used qualitatively, and with engineering judgement in their interpretation. The optimization algorithms used due not have this same judgement and are susceptible to these convergence issues.

These mesh quality violations cause the following optimization iteration to fail, as the simulation cannot compute a proper solution. In these cases, an "improved" solution can be isolated, prior to the mesh quality violation occurring. While this is not strictly an optimized geometry, it can still be valuable from an industrial standpoint. The main purpose of applying shape optimization to automotive applications is to search for further improvements on the design. If an improvement is realized, this can still be useful for a car's development, despite it not being a rigorously optimized geometry.

3. Results and Discussion

This section examines the results of the optimization studies and post-optimality CFD for each of the two variants and their FFDs, for the three different step update algorithms tested. The optimization results for each step update will be first compared for the *Estateback* variant they were tested on, then compared for the two *Estateback* variants. This will highlight what the relative improvement for each geometry was, as well as the total improvement expected for the updated *Estateback* variant in comparison to the *baseline Estateback*.

3.1 Roof Spoiler Optimization

The results of the roof spoiler optimization will be presented by first investigation the evolution of the objective function (drag) as a function of optimization iteration, K, for the *base* and *improved* variants of the *Estateback*. These results of each variant will then be compared to each other, and the resultant geometry deformations will be compared.

3.1.1 Base Variant Results

The evolution of aerodynamic drag normalized to the initial function evaluation as a function of optimization iteration is presented in Figure 6 for the three optimization algorithms. It is noteworthy that the DBFGS did not make it past the first optimization iteration. This means that the first iteration for which the inverse Hessian matrix was calculated, resulting in a poorly-conditioned search direction, yielding non-physical mesh deformations, violating the mesh quality. This poor conditioning was due to the inversion of the Hessian matrix, which was discussed in Section 22.12.1.3 Since the DBFGS algorithm yielded mesh quality violations after the first optimization iteration, all three optimization algorithms would have had the same Jacobian vector calculated, because they all perform a steepest descent evaluation for the first optimization iteration. Since neither the CG, nor the BFGS algorithm yielded mesh quality issues, and *OpenFOAM* did not report the damping term being activated for the DBFGS solver, the problem in the search direction had to stem from the inversion of the Hessian matrix performed by the DBFGS algorithm. This indicates that the problem was poorly-conditioned in nature, and was non-convex/multimodal.

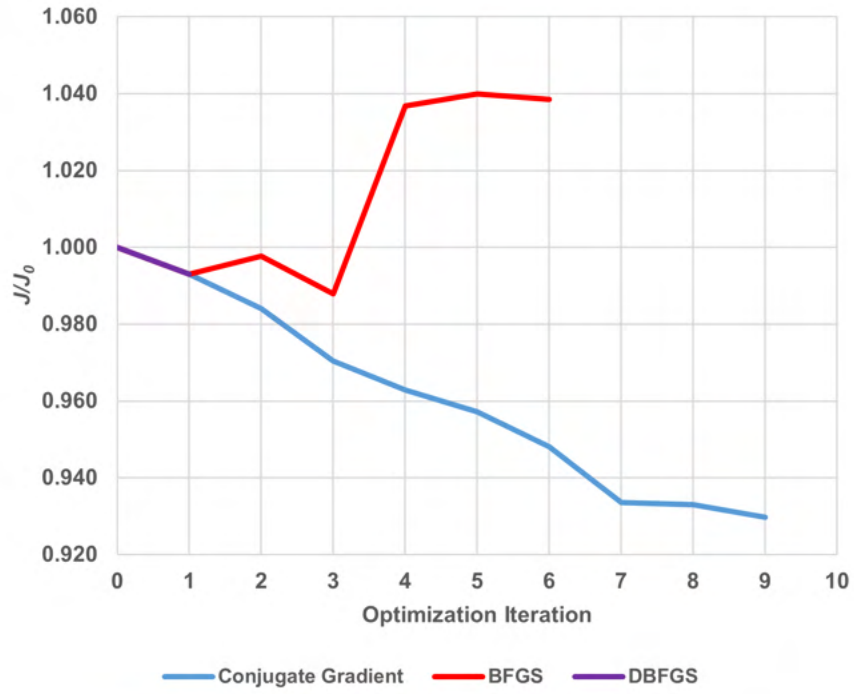


Figure 6 – Optimization results for the three optimization algorithms. The objective function is normalized to the drag of the *base* variant at the initial optimization iteration

The BFGS trace is of interest, as it results in an increase in the objective function at the second optimization iteration, as well as at the fourth and fifth optimization iterations. This increase is feasible, due to the maximum number of iterations that are specified for the line search. It is also interesting to note that the solution converged, despite it being to a value greater than the initial objective function value. This suggests that there was some degree of multi-modality in the objective function's topology, with this being a local minimum. This multi-modality will be further discussed in the results of the *improved* variant's optimization results.

The Conjugate Gradient trace shows that there was constant improvement in the objective function, and a converged solution could be found. A maximum improvement in the objective function of 7.0% was predicted from the optimization algorithm, relative to the initial design vector. Since the Conjugate Gradient method largely relies on gradient information, with slight "steering" of the search direction from the β term, it is less sensitive to numerical error in the sensitivity derivatives, and will be a more robust method. The Conjugate Gradient method takes longer to yield an improvement since it will suggest smaller displacements at each optimization iteration, but this is favourable in the presence of a more complex objective topology, as is clearly exhibited.

3.1.2 Improved Variant Results

The evolution of aerodynamic drag normalized to the initial function evaluation, as a function of optimization iteration is presented in Figure 7, for the *improved Estateback* variant. As with the *base* variant, the DBFGS optimizer failed after the first optimization iteration, which would be due to a poorly conditioned search direction. It is evident that for industrial cases, the inversion step of the DBFGS algorithm yields numerical instabilities in the sensitivities.

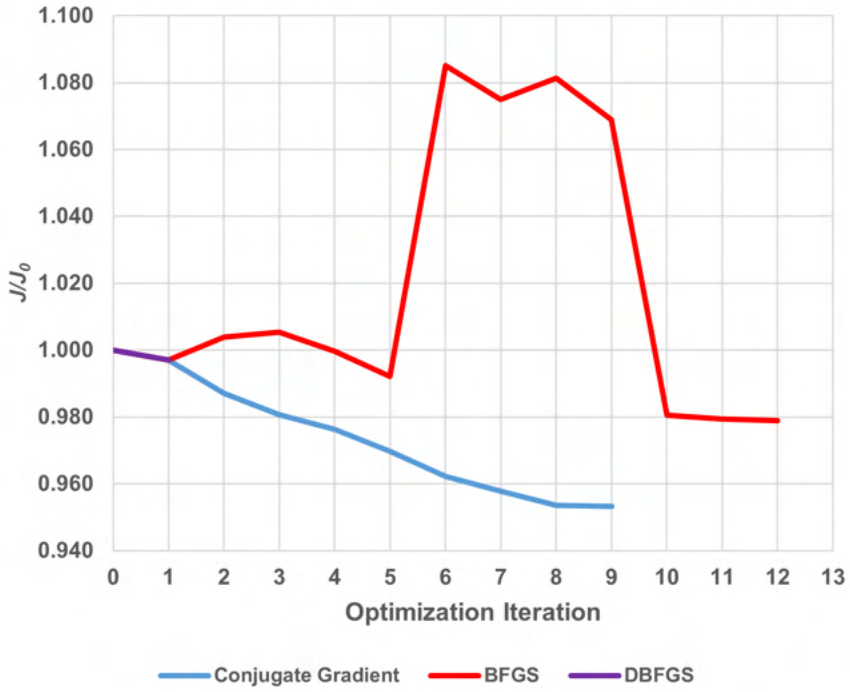


Figure 7 – Optimization results for each of the three optimization algorithms

The BFGS trace differed from the *base* variant, since its converged solution reduced the car's drag. However, in the process the objective function was increased and subsequent iterations oscillated around this value, before converging to a solution with its drag reduced. This once again highlights that for industrial automotive applications, the BFGS algorithm is less robust, due to insufficient convergence of the primal equations, as well to the possibility of large displacements at each iteration, degrading the accuracy of the local Jacobian and inverse Hessian approximations. The fact that it converged for the *improved* variant and not the *base* variant is not indicative of the user-defined modifications resulting in a more favourably conditions objective function topology, but to rather to the change in the initial design vector that was sufficient to place the car in a more favourable region of the objective function's topology, such that it could be guided towards a local minimum.

The Conjugate Gradient trace shows that a converged, improved objective function value could be reached. It is also valuable to note that this converged objective function value predicts greater improvement than the BFGS' converged solution. As with the *base* variant, the Conjugate Gradient

method was not only more robust, but it was also more effective at reducing the drag of the car. It is also valuable to note that the Conjugate Gradient optimization of the *improved* variant only resulted in a 4.7% improvement in drag, compared to 7.0% for the *base* variant. This is due to the user-driven modifications, that would have moved the initial design vector closer to the converged minimum. As such, a smaller total step was needed to reach that minimum, leading to a more accurate Jacobian prediction due to staying within the region of space in which the adjoint approximation is valid (convex).

The results of the spoiler optimization for the *improved* variant hint at more global multi-modality in the design space. Both the Conjugate Gradient and BFGS solutions yielded a converged solution, however the predicted drag reduction was different for both points. This suggests that the design space was not truly convex, which would extend to the *base Estateback* variant. These local minima were verified by performing an additional optimization run on the converged geometries. The repeat of the Conjugate Gradient model converged immediately due to insufficient change in the objective function. This showed that the Conjugate Gradient method found a local minimum, with less than 0.5% change in the drag when the optimizer was restarted using the new initial design vector. In the case of the BFGS repeat, it converged to a new solution after five major iterations, predicted an additional 2.5% reduction in drag. Since the BFGS mapping was capable of finding a new, stronger minimum than previously found, this further highlights the multi-modality of the design space, and suggests that the Conjugate Gradient algorithm is better suited to finding stronger minima in automotive applications.

This multi-modality is partly an artifact of the lack of wake resolution within the RANS solver, but is also due to the wake interactions between the roof flow and the underbody/rear fender flow. Minimizing with respect to ${C_D}^2$ could help improve the convexity of the design space and improve solution quality. This objective function was not readily available and its derivation was out of the scope of this research.

3.1.3 Comparing Optimization Results

This section details how the optimization of both variants compared to each other, normalized by the initial objective function value for the *base* variant, denoted by $J_{0,Base}$. The objective function evolution traces for each algorithm and variant are presented in Figure 8. As expected, the value of the objective function at optimization iteration one is the same for all algorithms for a given variant, since all three algorithms employed a steepest descent initial step, and were therefore limited to the same maximum initial displacement.

Comparing the Conjugate Gradient traces for both *Estateback* variants highlights that there was a noticeably larger net improvement in the *base* variant's drag, but the optimized *improved* variant is still predicted to have less drag than the optimized *base* variant. At the initial optimization iteration, the *improved* variant was predicted to have 4.8% less drag than the *base* variant, but the converged design vector was only predicted to have an improvement of 2.2%. This therefore shows the importance of using expert-driven design to reduce the net drag of a car. While the optimizer was capable of improving the *base* variant more than the *improved* variant, the optimized *improved* variant was still better than the optimized *base* variant. Since the MCF method that was used to drive the modifications of the *improved* variant, it is more favourable to use this method during the early phases of a design cycle to respect styling constraints, with optimization being reserved for later stages, when further improvements would be desired.

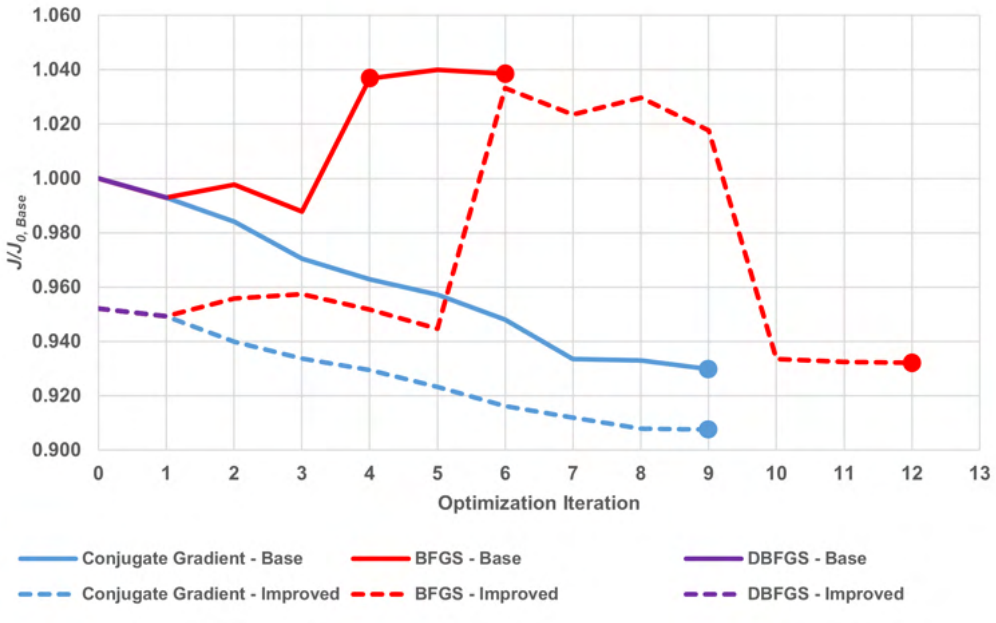


Figure 8 – Optimization results for each of the three optimization algorithms, for both *Estateback* variants. Both variants' drag is normalized by the drag of the *base* variant at the initial optimization iteration ($J_{0,Base}$). Design vectors for post-optimality CFD are marked

Table 2 highlights the performance of all of the optimizers for the two variants. The minimized objective function value relative to the initial objective function value for a variant (J^*/J_0), as well as the minimized objective function value relative to the *base* variant's initial drag ($J^*/J_{0,Base}$) are presented to highlight the relative and net improvements found from each algorithm. J_K/J_0 is the normalized value of the aerodynamic drag for the final optimization iteration, K. If J^*/J_0 and J_K/J_0 differ, this is not an indication of a converged solution not being found, but that the converged solution was not the minimal solution found during the optimization process. For the equivalent number of flow solutions, k, this value is in relation to the number of iterations the initial primal solution took. All subsequent primal evaluations cost 0.6 equivalent flow solutions, and all adjoint solutions cost 0.4 equivalent flow solutions.

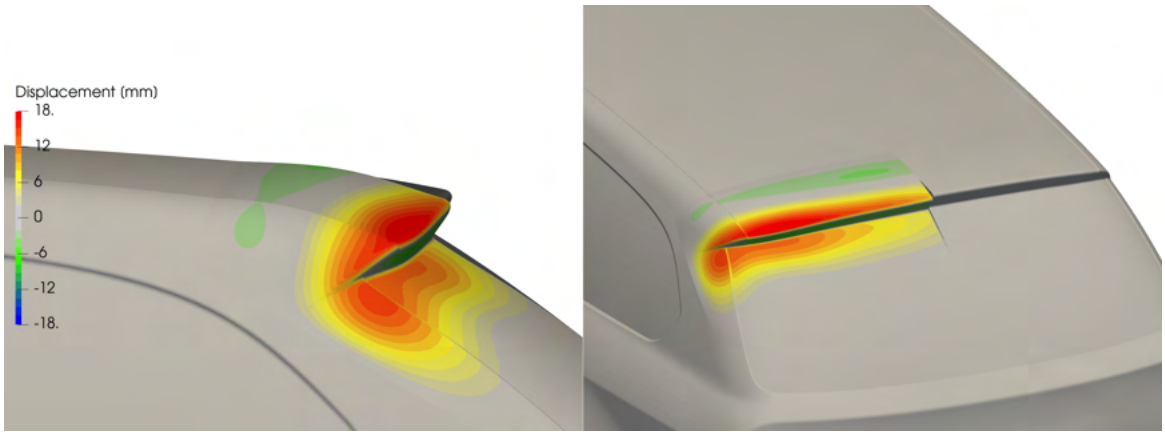
Table 2 – Summary of optimization progress for each algorithm and variant studied, for the spoiler FFD

Algorithm	Variant	K	k	J^*/J_0	$J^*/J_{0,Base}$	J_K/J_0	Converged
CG	*Base*	9	13.4	0.930	0.930	0.930	Yes
	Improved	9	10.4	0.953	0.908	0.953	Yes
BFGS	*Base*	6	16.2	0.988	0.988	1.038	Yes
	Improved	12	27.2	0.979	0.932	0.979	Yes
DBFGS	*Base*	1	2.4	0.993	0.993	0.993	No
	Improved	1	2.4	0.997	0.949	0.997	No

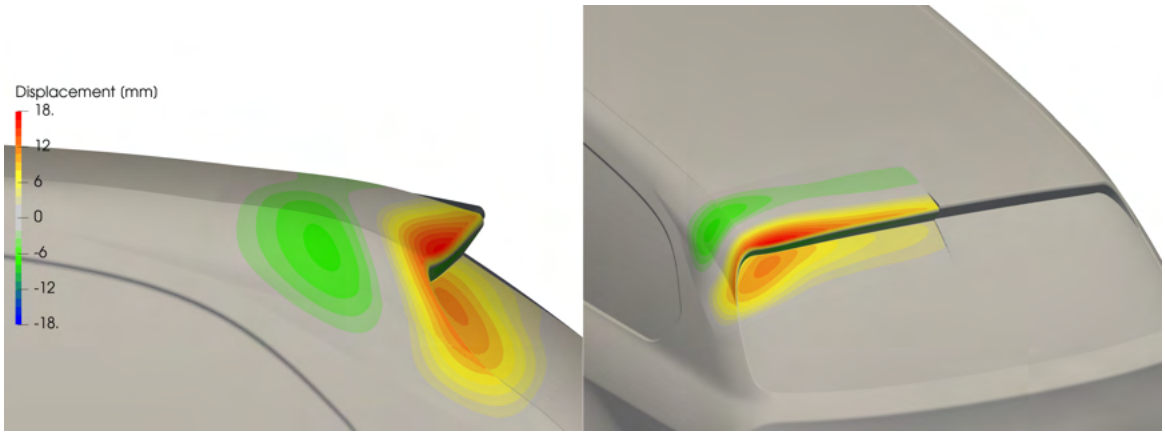
From Table 2, the most improved optimization iteration results for the *base* and *improved* variants were predicted to be from the Conjugate Gradient optimization. Furthermore, the Conjugate Gradient evaluations were able to perform reduce drag in fewer equivalent function evaluations than the BFGS algorithm did. This was because the BFGS optimization loops required many line search iterations before either the iteration limit, or an improved design was found. The Conjugate Gradient models tended to require only one function evaluation per line search execution, which further supports the Conjugate Gradient's search direction being more robust, but also more efficient than the other tested methods.

To understand the modifications to the *base* and *improved* variants, the deformed geometry for the *base* variant at optimization Iteration 9 and the deformed geometry of the *improved* variant at optimization Iteration 9 are presented in Figure 9, for the Conjugate Gradient models. When examining the displacement, a positive displacement means the surface was deformed inwards, along the surface normal, and a negative displacement is outwards, against the surface normal. The BFGS and DBFGS models will not be presented as they did not produce a comparable reduction in drag.

For both the *base* and *improved* variant, the outboard end of the spoiler was flared downwards more than the inboard end, creating more downward curvature on the outboard end of the roofline, leading to more outboard flow acceleration. Additionally, having the trailing edge lower on the outboard end of the roof can help contract the overall size of the wake. This would require validation with higher fidelity methods, as RANS tends to underpredict the amount of separation on the rear fascia. Thus, it would perform relatively accurately up until separation occurred on the roofline.



(a) *Base* Variant, Conjugate Gradient Iteration 9



(b) *Improved* Variant, Conjugate Gradient Iteration 9

Figure 9 – Deformed surfaces, subject to the spoiler FFD

On the *base* variant, upstream of the spoiler, a bump was added to the roof, allowing for more flow acceleration upstream of the spoiler. This bump was not present on the *improved* variant, due to previous modifications that were guided by the MCF [13]. Since the MCF yielded information that increasing flow acceleration at the trailing edge of the roof would help decrease drag, the roof was already given additional downwards curvature. As such, a bump was not needed to be generated to achieve this effect. It is also interesting to note that this bump was present on the *base* variant, since this was information that was directly distilled from the MCF guidance. However, the bump being developed was a result of the optimizer exploiting the geometry of the spoiler FFD, rather than being a physical solution. It can be seen that the bump present in the *base* variant was not

tangent to the rest of the roofline. This suggested that the roofline should have been raised higher to accommodate for the additional curvature generated downstream of the bump. Having the bump present would create a local pressure buildup and potential thickening of the boundary layer, that would not be desired. Nonetheless, this highlights that the MCF and optimization should be used in tandem, whereby the MCF information guides early work in a car's design cycle to target key areas where drag can be minimized, with optimization extracting the final performance from the design.

3.2 Roof Spoiler Post-Optimality CFD

The designs that were chosen to perform post-optimality CFD on were Conjugate Gradient optimization iteration 9 and BFGS iterations 4 and 6 for the *base* variant, as well as Conjugate Gradient iteration 9 and BFGS iteration 12 for the *improved* variant, as marked in Figure 8. The BFGS iterations were chosen to determine the accuracy of the RANS solver employed in the optimization process when large deformations were generated between optimization iterations, which is one of the hallmarks of the BFGS method. Additionally, BFGS iterations 3 and 6 for the base variant were chosen to see if iteration 3 actually resulted in an improvement in drag, and if iteration 6 resulted in an increase in drag. The remaining options are chosen to verify if the predicted improvement is able to be replicated within a DDES model. The DBFGS iterations were not chosen for either variant, as only one optimization iteration was completed in both cases.

Figure 10 presents the percent difference between the post-optimality and optimizer's prediction of J/J_0 for the spoiler optimization. For the *base* variant, the change direction predicted by the optimizer was different than the direction calculated by the DDES model. In all three cases, if the optimizer predicted a change that increased drag, post-optimality CFD suggested it decreased drag, and vice versa. This trend was not present in the *improved* variant's optimization, with both analyzed designs predicting a reduction in drag with the post-optimality CFD also predicting a drag reduction. It is also valuable to see that in both cases of the *improved* variant's post-optimality analysis, they predicted roughly 1.4% more drag reduction in the post-optimality CFD, compared to the optimization value. It was not expected that the post-optimality CFD would provide the actual drag savings from an optimization design vector, but rather a more accurate sense of the improvement predicted. However, drag reduction predicted by post-optimality CFD was trusted, given the consistency across different designs shown when validating the DDES models. Experimental validation of the designs would be required to understand the true drag savings achieved.

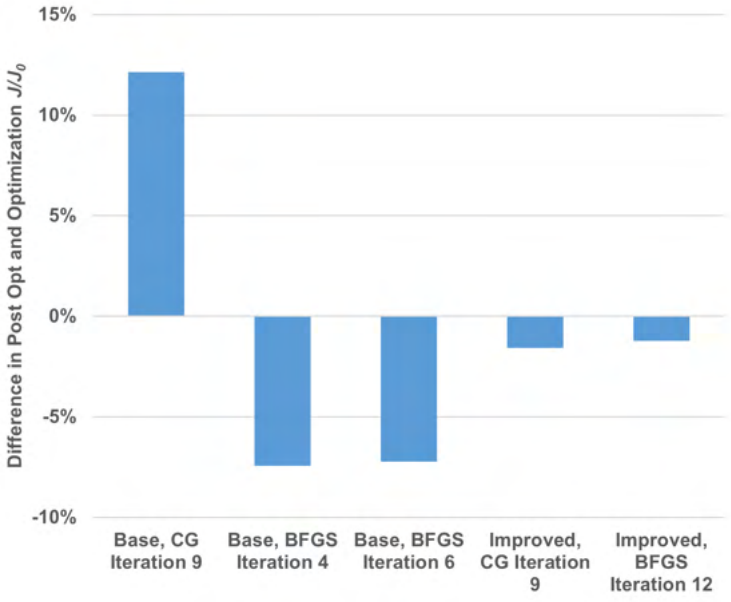


Figure 10 – Percent difference in J/J_0 from post-optimality CFD to optimizer predictions, for the spoiler design vectors chosen

In summary, the post-optimality CFD had better agreement with the optimizer for the *improved* variant simulations. This was due to the modifications that defined the *improved* variant, which sought to reduce the size of the rear wake and make it more uniform. As a result, the optimizer's underlying RANS solver was better able to deform the geometry in a manner that would yield a drag reduction. This highlights the importance of expert-driven design modifications prior to the application of optimizers for automotive purposes.

3.3 Front Bumper Cheek Optimization

The results of the front bumper cheek optimization will be presented in the same manner as the spoiler optimization.

3.3.1 Base Variant Results

The evolution of aerodynamic drag normalized to the initial function evaluation as a function of optimization iteration is presented in Figure 11 for the front bumper cheek of the *base* variant. In contrast to the *base* variant's spoiler optimization, none of the algorithms showed a clear path of improvement by deforming the front bumper cheek. There was a considerable amount of *zig-zagging* by the various algorithms, with a clear improvement being elusive. This was due to the more complex flow physics which occurred around the front bumper cheek, compared to the spoiler region. The flow that comes off the front bumper cheeks was largely dominated by the interactions it had with the front tires and their wake. Since the optimizer makes use of a SA RANS model, the ability to predict these wakes was susceptible to significant modelling errors due to the lack of resolved turbulent kinetic energy. As such, the sensitivities in this area may not only be incorrect, but the resultant primal evaluations may not have given a reasonable evaluation of the flow field.

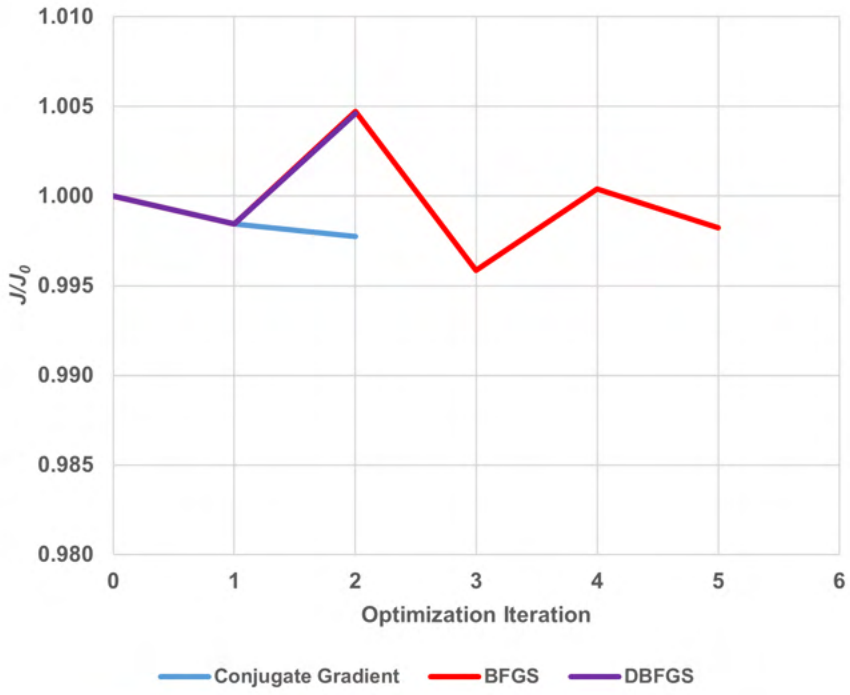


Figure 11 – Optimization results for each of the three optimization algorithms. The objective function is normalized to the drag of the *base* variant at the initial optimization iteration

In the case of the BFGS optimization, large deformations were generated for the front bumper cheek, which would not be acceptable from a styling perspective. However, the BFGS algorithm was capable of yielding more improvement than the Conjugate Gradient approach, as it converged early due to small surface displacements. This is in contrast to the spoiler optimization where the BFGS algorithm was not as effective as Conjugate Gradient.

3.3.2 Improved Variant Results

The evolution of aerodynamic drag normalized to the initial function evaluation as a function of optimization iteration is presented in Figure 12 for the *improved Estateback* variant.

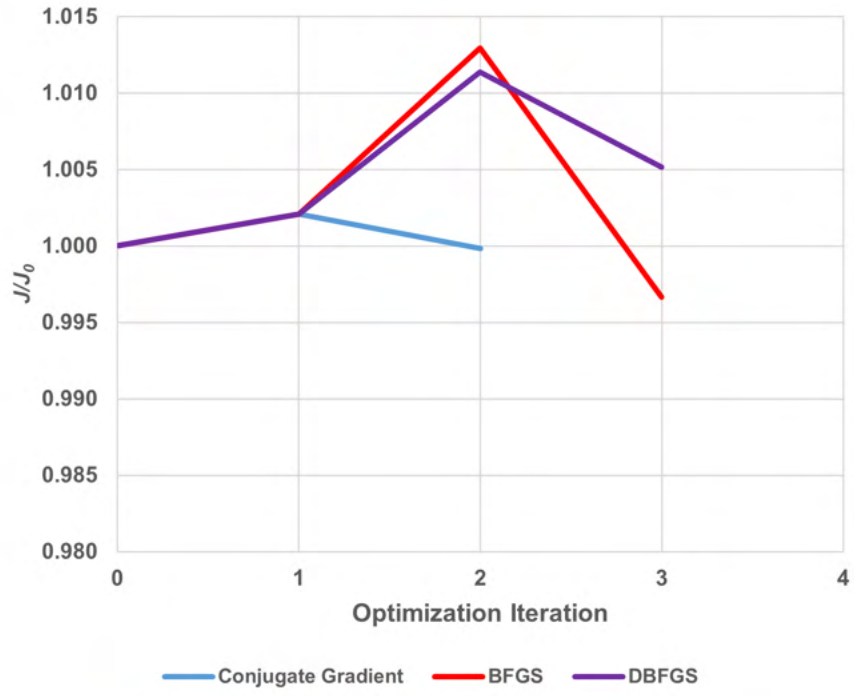


Figure 12 – Optimization results for each of the three optimization algorithms

It is noteworthy that the BFGS and DBFGS traces differed so significantly. This can be be attributed to the inversion of Hessian matrix, occurring in the DBFGS algorithm causing variations in the search direction. These will not occur for the BFGS algorithm, as the *OpenFOAM* implementation directly solves for the inverse Hessian matrix. As with the *base* variant, the BFGS algorithm predicted greater improvement in drag than the Conjugate Gradient algorithm, largely driven by larger surface displacements. As with the *base* variant, little improvement was found by modifying the front bumper cheek region, due to the complex flow physics that are difficult to model using a RANS solver. This further highlights that optimization was best suited to areas in which the RANS solver could yield reasonable predictions: in areas that are not dominated by unsteady flow phenomena.

3.3.3 Comparing Optimization Results

This section details how the optimization of both variants compared to each other, normalized by the initial objective function value for the *base* variant, denoted by $J_{0,Base}$. The objective function evolution traces for each algorithm and variant are presented in Figure 13. As with the optimization of the spoiler, the *improved* variant displayed less relative drag reduction when deforming the front bumper cheeks. This should be expected, since the purpose of the *improved* variant was to push the *DrivAer Estateback* closer to a minimal drag design, meaning it should result in less reduction in drag from the optimization routine than the *base* variant. However, significantly less improvement was predicted by deforming the front bumper cheeks, since the maximum relative improvement was less than 0.5% compared to up to 7% relative improvement by deforming the spoiler. This is once again due to the more complex flow physics in this region, which can be difficult for a RANS-based optimizer to yield a reduction in drag.

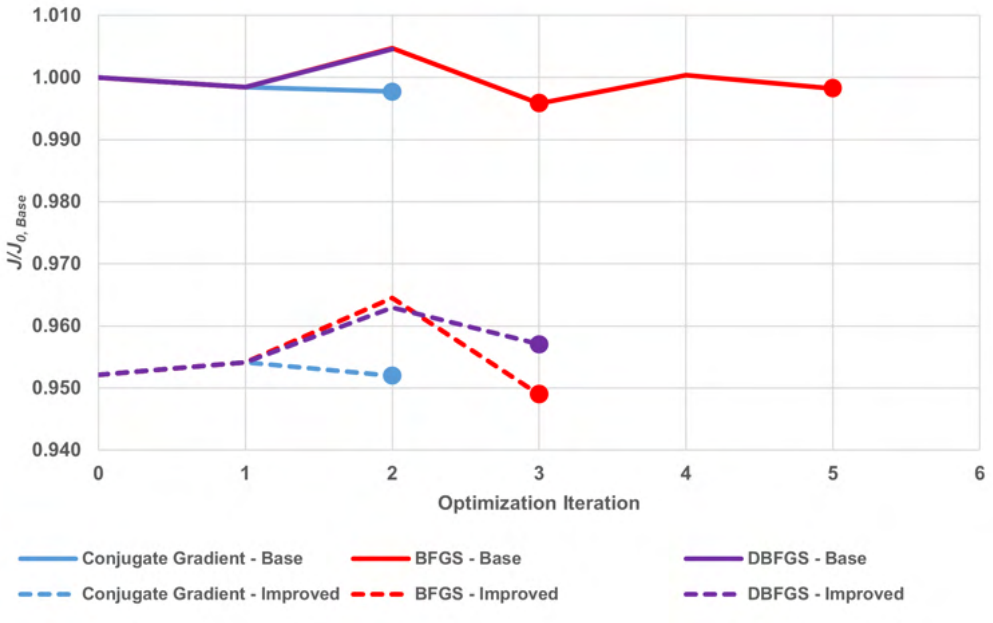


Figure 13 – Optimization results for each of the three optimization algorithms, for both *Estateback* variants. Both variants' drag is normalized by the drag of the *base* variant at the initial optimization iteration ($J_{0,Base}$). Design vectors for post-optimality CFD are marked

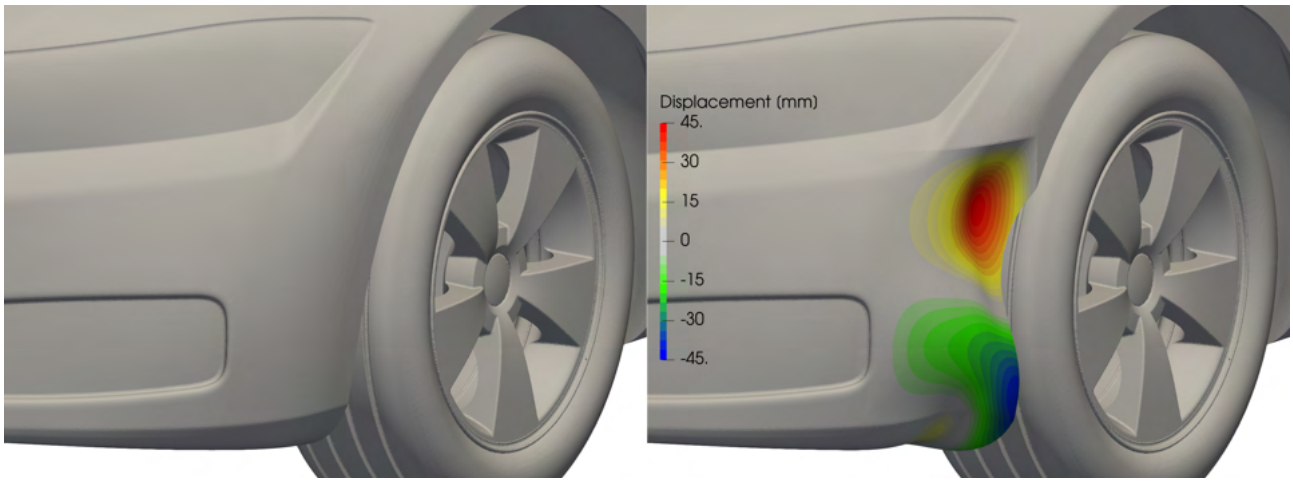
Table 3 highlights the performance of all the optimizers for the two variants for the front bumper cheek optimization, similar to the spoiler optimization. Unlike the spoiler optimization, the BFGS algorithm resulted in the most improvement in drag for both the *base* and *improved* variants. However, it did not find a converged solution, meaning these designs can only be considered improvements, rather than optima. Part of the reason the Conjugate Gradient method did not yield a noticeable improvement, is because the first few iterations result in little total displacement of the surface with the maximum displacement being around three millimeters. Since this region was dominated by complex flow physics, these small displacements were difficult to resolve with a RANS solver. Even though drag reduction was predicted from deformations of similar magnitude in previous research [[13]], these designs were evaluated using DDES models which were better suited to resolving the front tire wakes. As such, the Conjugate Gradient evaluations converged early due to the convergence criteria used. Therefore, it could be more valuable to perform a BFGS optimization sweep first, on which the best iteration could then have a Conjugate Gradient optimization sweep performed.

Table 3 – Summary of optimization progress for each algorithm and variant studied, for the front bumper cheek FFD box

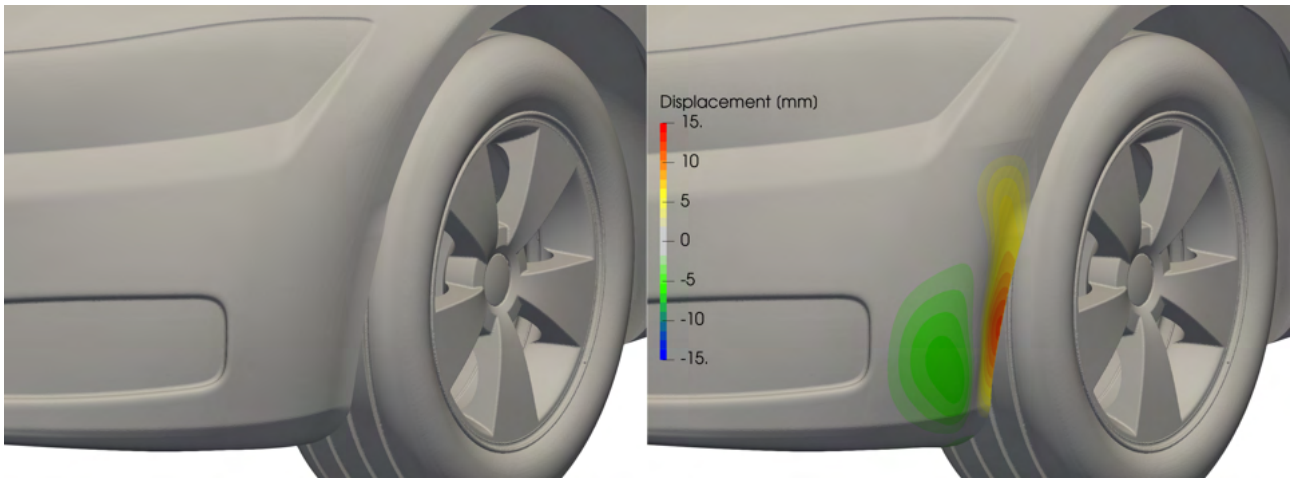
Algorithm	Variant	K	k	J^*/J_0	$J^*/J_{0,Base}$	J_K/J_0	Converged
CG	*Base*	2	3.4	0.998	0.998	0.998	Yes
	Improved	2	5.8	1.000	0.952	1.000	Yes
BFGS	*Base*	5	11.8	0.996	0.996	0.998	No
	Improved	3	9.2	0.997	0.949	0.997	No
DBFGS	*Base*	2	5.8	0.998	0.998	1.005	No
	Improved	3	9.2	1.000	1.000	1.005	No

One aspect of the results that was consistent between the spoiler and front bumper cheek results was that the BFGS algorithm takes consistently more function evaluations when performing an optimization routine. As with the spoiler optimization, the BFGS algorithm resulted in a search direction where an improvement in the objective function required more than one line search iteration. This can be attributed to the additional curvature information that the BFGS algorithm provided the solver, compared to the Conjugate Gradient case. This allows for a larger traversal of the objective function topology, but it may then require additional line search iterations to meet the *Armijo* conditions.

From Table 3, the most improved optimization iteration results for the *base* and *improved* variants were predicted to be from the BFGS algorithm. The deformed geometry for the *base* variant at optimization Iteration 3 and the deformed geometry of the *improved* variant at optimization Iteration 3 are presented in Figure 14.



(a) *Base* Variant



(b) *Improved* Variant

Figure 14 – Deformed surfaces, subject to the front bumper cheek FFD

The *base* variant showed roughly three times as much deformation than the *improved* variant did in their most improved forms. The deformation realized for the *base* variant would not be practical however, from neither a manufacturing nor an aesthetics perspective. However, it is interesting to note that on the lower portion of the front bumper cheek, the optimizer sought to grow a tire air deflector, to push air outwards around the lower portion of the tire. The optimizer also tried to push the bumper cheek outwards, which would generate some local thrust due to the additional curvature, but would also help align the flow to be more parallel to the face of the front tires. This was present on the *improved* variant as well, to a lesser extent.

The inwards deformation on the upper half of the front bumper cheek for the *base* variant is an effect of the RANS solver being unable to resolve the flow physics due to the front bumper cheek's interactions with the tires. While some inward curvature would help generate suction and local thrust, the bumper cheek was pushed in too much, causing some of the front tires to be exposed. There was some inwards deformation of the *improved* variant, however it was far more subtle, seeming to reduce the height of the flick. When this flick was initially generated, only the flick length along the front bumper cheek was changed, with all tested variants being 5 mm tall [13]. Based on these deformations, it is evident that a RANS-driven gradient-based optimizer will perform more poorly in areas where the flow is not well behaved and features large areas of separation.

3.4 Front Bumper Cheek Post-Optimality CFD

The front bumper cheek designs on which post-optimality CFD was performed were: were Conjugate Gradient optimization Iteration 2 and BFGS Iterations 3 and 5 for the *base* variant, as well as Conjugate Gradient Iteration 2, BFGS Iteration 3, and DBFGS Iteration 3 for the *improved* variant. The DBFGS result for the *improved* variant was analyzed since the search direction for BFGS and DBFGS should have been similar, but there was a noticeable difference in their objective function values.

Figure 15 highlights how the chosen optimization iterations performed in the post-optimality DDES model for the front bumper cheek deformations. For both variants, there was no real trend in the post-optimality results, largely due to the complex flow physics around the front bumper cheek, which were less prevalent around the roof spoiler. Since the drag in this region was largely dominated by the front wheel drag, these are features that the RANS-based optimizer could not resolve with sufficient fidelity to yield effective change. However, as with the spoiler, the *improved* variant saw better drag correlation between the post-optimality CFD and the optimzation results.

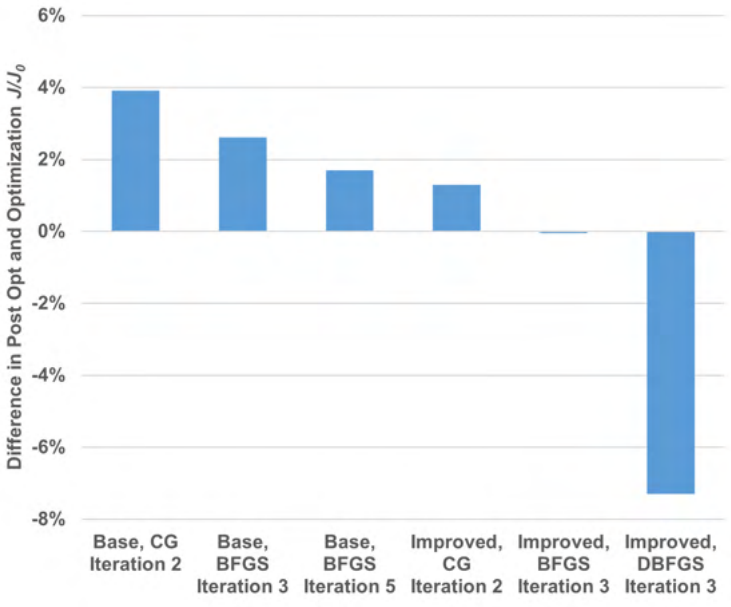


Figure 15 – Percent difference in J/J_0 from post-optimality CFD to optimizer predictions, for the front bumper cheek design vectors chosen

In summary, since the front bumper cheek region was dominated by complex flow structures, the optimizer did not reduce drag measurably, and did not agree with the post-optimality CFD. This highlights the importance of the solver fidelity in the optimization routine. In areas where the flow physics are complex, it would be more effective to use a gradient-free optimizer, which could use DDES models for its drag evaluation, instead of a gradient-based RANS optimizer. This negates inaccuracies in a RANS model around adverse pressure gradients, and remove the need for post-optimality CFD.

4. Conclusions and Future Work

This paper presented an analysis on the efficacy of different RANS-based unconstrained gradient-based optimizers for reducing drag of a canonical automotive geometry. This involved performing optimization using freeform deformation boxes around the spoiler and front bumper cheeks for two variants of the *DrivAer Estateback*, which determined how the different algorithms performed in the presence of different flow features.

For the spoiler region optimization, a relative drag reduction of up to 7% was predicted for the *base* variant, and up to 4.7% for the *improved* variant. The baseline variant predicted greater drag reduction, since the modifications previously devised by the authors of the *improved* variant placed it closer to a local minimum. However, the *improved* variant predicted a greater net improvement in drag when combining the Momentum Contribution Field and optimization methods than just MCF or optimization modifications made to the *base* variant. For both variants of the *Estateback*, the Conjugate Gradient algorithm was found to be the fastest and most robust for reducing drag.

Optimization of the front bumper cheek highlighted the inaccuracies of the RANS solver and subsequent sensitivities in areas dominated by separation. Negligible relative improvement was predicted for either *Estateback* variant, due to the complex flow physics around and downstream of the front tires. For both regions, it was noted that the BFGS algorithm tended to yield more displacement earlier in the optimization routine, which tended to cause oscillatory changes in drag because of the inexact line search method used. Additionally, the BFGS and DBFGS algorithms tended to yield poorly conditioned sensitivities, due to the curvature information contained in their Hessian approximations, which caused the optimization routine to fail due to mesh quality violations. RANS-based gradient optimizers are insufficient for realistic automotive geometries in regions dominated by adverse pressure gradients and separation as such.

Post-optimality CFD of key designs for the roof spoiler and front bumper cheek optimization routines highlighted areas in which the RANS-based optimizer suffered in accuracy. For the roof spoiler optimization, the *base* variant's predicted change did not show a consistent trend with the post-optimality CFD, influenced by adverse pressure gradients not effectively resolved by the RANS solver. The *improved* variant's post-optimality CFD showed the same direction of change with a consistent 1.4% additional drag reduction predicted, compared to the RANS optimization. Post-optimality CFD for the front bumper cheek optimization further emphasized the inability of effective drag minimization from a RANS-based solver. Greater disagreement was found in both the drag change direction and magnitude between the post-optimality and optimization CFD.

A number of improvements to the tested optimizer were noted: the first would be the ability to place restrictions on the total displacement of the freeform deformation box in each direction. Limiting the total displacement could still yield an optimzation iteration where the objective function increases due to the inexact line search, but it could damp control point displacement guiding the solution back towards a local minimum. Furthermore, being able to rescale the design variables at each iteration would be valuable. This could work with total displacement limitations to damp certain control points that lead to excessive deformations, improving optimizer robustness. Additionally, gradient-free methods such as response surface methods, may be more effective for optimizing regions dominated by complex flow physics, such as the front bumper cheeks. The underlying solver would benefit from more accurate function evaluations, such as DDES models, and the solution would not be susceptible to inaccurate gradient information from an adjoint computation.

5. Contact Author(s) Email Address(es)

mailto: maurice.nayman@rmc.ca
mailto: ruben.perez@rmc.ca

6. Copyright Statement

References

[1] IEA, "Energy Technology Perspectives 2020," Tech. rep., IEA, 2020.

[2] Hinterbuchinger, M., "Aerodynamics electrified, Porsche Taycan Turbo," *12th FKFS Conference Progress in Vehicle Aerodynamics and Thermal Management, Stuttgart Germany*, September 2019.

[3] Papoutsis-Kiachagias, E. M. and Giannakoglou, K. C., "Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications," *Archives of Computational Methods in Engineering*, Vol. 23, No. 2, December 2014, pp. 255–299.

[4] Papoutsis-Kiachagias, E., Magoulas, N., Mueller, J., Othmer, C., and Giannakoglou, K., "Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions," *Computers & Fluids*, Vol. 122, November 2015, pp. 223–232.

[5] Karpouzas, G., Papoutsis-Kiachagias, E., Schumacher, T., De Villiers, E., Giannakoglou, K., and Othmer, C., "Adjoint optimization for vehicle external aerodynamics," *International Journal of Automotive Engineering*, Vol. 7, 01 2016, pp. 1–7.

[6] Papoutsis-Kiachagias, E. M., Asouti, V. G., Giannakoglou, K. C., Gkagkas, K., Shimokawa, S., and Itakura, E., "Multi-point aerodynamic shape optimization of cars based on continuous adjoint," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 2, October 2018, pp. 675–694.

[7] Papoutsis, E., Giannakoglou, K., W., R., N., M., and Keromnes, V., "From scanned CAD to an optimized car: aerodynamic shape optimization of an electric vehicle based on adjointOptimisationFoam," *ESI OpenFOAM Conference 2021*, 2021.

[8] Nastov, A. and Caples, D., "Aerodynamic Development of the 2019 Chevrolet Corvette C7 ZR1," *SAE Technical Paper Series*, SAE International, April 2019.

[9] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, Vol. 12, No. 6, 1998, pp. 620.

[10] Papoutsis-Kiachagias, E., Gkaragkounis, K., Margetis, A.-S., Skamagkis, T., Asouti, V., and Giannakoglou, K., "adjointOptimisationFoam: An OpenFOAM-Based Framework for Adjoint-Assisted Optimisation," *14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, Athens, Greece*, No. 7960-18382, Institute of Structural Analysis and Antiseismic Research National Technical University of Athens, 2021.

[11] Othmer, C., "Adjoint methods for car aerodynamics," *Journal of Mathematics in Industry*, Vol. 4, No. 1, 2014, pp. 6.

[12] He, P., Mader, C. A., Martins, J. R., and Maki, K. J., "An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM," *Computers & Fluids*, Vol. 168, may 2018, pp. 285–303.

[13] Nayman, M. N. and Perez, R. E., "A Sensitivity-Driven Approach to Automotive Aerodynamic Design," *AIAA AVIATION 2023 Forum, San Diego*, No. 2023-3387, American Institute of Aeronautics and Astronautics, June 2023.

[14] Parallel CFD and Optimization Unit, National Technical University of Athens, *adjointOptimisationFoam, an OpenFOAM-based optmisation tool*, Parallel CFD and Optimization Unit, June 2020.

[15] Fletcher, R., "Function minimization by conjugate gradients," *The Computer Journal*, Vol. 7, No. 2, February 1964, pp. 149–154.

[16] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer, 1999.

[17] Powell, M. J. D., "Algorithms for nonlinear constraints that use lagrangian functions," *Mathematical Programming*, Vol. 14, No. 1, December 1978, pp. 224–248.

[18] Perry, E. C., Benzley, S. E., Landon, M., and Johnson, R., "Shape optimization of fluid flow systems," *Proceedings of ASME FEDSM'00. 2000 ASME Fluids Engineering Summer Conference*, 2000.

[19] Heft, A., Indinger, T., and Adams, N., "Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations," *SAE 2012 World Congress & Exhibition*, SAE International, 04 2012.

[20] Ahmed, S., Ramm, G., and Faltin, G., "Some Salient Features Of The Time-Averaged Ground Vehicle Wake," *SAE International Congress and Exposition 1984*, SAE International, February 1984.

[21] Wood, D., Passmore, M. A., and Perry, A.-K., "Experimental Data for the Validation of Numerical Methods

- SAE Reference Notchback Model," *SAE International Journal of Passenger Cars - Mechanical Systems*, Vol. 7, No. 1, April 2014, pp. 145–154.

[22] Patankar, S. and Spalding, D., "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, Vol. 15, No. 10, October 1972, pp. 1787–1806.

[23] Issa, R., Gosman, A., and Watkins, A., "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme," *Journal of Computational Physics*, Vol. 62, No. 1, January 1986, pp. 66–82.

[24] OpenCFD Ltd., "OpenFOAM: User Guide v2206 pimpleFoam," 2022, Online; Accessed April 20, 2023.