A computational fluid dynamics (CFD) code has been combined with a genetic algorithm (GA) to perform a two-dimensional drag minimization on the base region of tractor trailers. Few studies have been conducted with CFD driven by a GA due to extensive run times and the general non autonomous characteristics of meshing a suitable CFD geometry. This study solves the two-dimensional Reynolds-averaged Navier-Stokes (RANS) equations to obtain a drag calculation used by the GA. Custom FORTRAN subroutines were written to handle the structured meshing of the tractor trailer with a variable geometry base slat for the entire design space. The mesh generator was used as a subroutine in the GA with the only input being the geometric variables. The objective function for the GA consists primarily of the mesh generator code and the CFD solver. The base optimization involved placing flaps of varying size, position, and curvature in the base region to determine a minimum drag configuration. The front of the tractor trailer remained unchanged, but was included in the CFD analysis. The base flaps were defined by a cubic function requiring 4 variables plus one additional variable to set the length of the flaps. For this study the flap length was limited to a maximum of four feet aft of the trailer base. The resulting minimum drag configuration was shown to reduce drag by over 50 percent when compared to the CFD run with no base flaps.

Nomenclature

\[ C_D = \text{Drag Coefficient} \]
\[ i, j = \text{Grid Indexing Parameter} \]
\[ y^+ = \text{Normalized Turbulence Length} \]
\[ H_1 = \text{Base Slat Parameter: Distance from truck centerline to intersection of slat and truck base} \]
\[ H_2 = \text{Base Slat Parameter: Distance from truck centerline to free end of slat} \]
\[ \beta = \text{Base Slat Parameter: Total angle of slat} \]
\[ L = \text{Base Slat Parameter: Distance slat extends from base of truck} \]
\[ \theta_1 = \text{Base Slat Parameter: Initial angle of slat} \]
\[ \theta_2 = \text{Base Slat Parameter: Final angle of slat} \]

I. Introduction

The use of genetic algorithms (GAs) as a design tool has been demonstrated in many areas of the aerospace industry.\(^1\)\(^2\) Aerospace applications have included the design of spacecraft controls\(^1\)\(^2\), turbines\(^3\)\(^8\)\(^15\), helicopter

---

\(^1\)Graduate Student, Department of Aerospace Engineering, AIAA Member.
\(^2\)Professor, Department of Aerospace Engineering, Associate Fellow AIAA.
\(^1\)Associate Professor, Department of Aerospace and Ocean Engineering, Associate Fellow AIAA.
controls, flight trajectories, wings and airfoils, inlets, rockets, missiles, and propellers. This study uses the IMPROVE© code which is a binary encoded tournament based GA used in References 13, 17, 18, and 21-23. The GA features many capabilities such as a pareto option, nicheing, creep, and elitism.

Genetic Algorithms are loosely based on the theory of evolution by having members compete with each other over generations. Members having the best survival characteristics, or in this case geometric parameters, are more prone to survive. For these two optimizations, fitness is determined by the computation of drag carried out by a RANS CFD simulation. The members in each case will be different geometries based on the selection of variable parameters within a given design space. The GA passes these members to the objective function. The objective function consists of a mesh generator for the given geometry and computational fluid dynamics (CFD) solver to obtain the drag value.

Performance derivatives with respect to each parameter are very costly when using Navier-Stokes CFD as an objective function. Geometric parameters also produce highly coupled results in complex flowfields. These factors justify using a GA rather than a gradient based optimization scheme for the drag minimization study presented in this thesis.

The steady-state Reynolds-averaged Navier-Stokes equations with an SST $k-\omega$ turbulence model and wall functions were solved for each member evaluated by the GA. The CFD solver, Fluent, uses a cell centered finite volume method for integration of the governing equations. The segregated solver option was used to decouple continuity and momentum equations. The energy equation was deactivated because the simulations were conducted at low speed and constant temperature. A second-order upwind discretization was used for the momentum equation while a first-order upwind discretization was used for turbulent kinetic energy and specific dissipation rate.

The driving force behind this study is the potential fuel cost savings of an aerodynamically optimized tractor trailer. In 2006 tractor trailers logged 143.6 billion miles in the United States alone at an average of 5.9 miles per gallon corresponding to 24.3 billion gallons of diesel fuel or the equivalent of 600 million barrels or oil. The direct economic impact is about 50 billion dollars for a crude oil price of $90 per barrel.

Figure 1 shows the horsepower contribution of aerodynamic drag and rolling friction resistance. For typical highway speeds, aerodynamic drag accounts for more than half of the engine load. A drag reduction of 20% would result in approximately 10% fuel savings. This amount of fuel savings would result in a savings of 4 billion dollars annually for trucks in the United States. Even if drag reduction efforts can only reduce drag by 5%, a significant financial motivation still exists for the aerodynamic optimization of tractor trailers.

This study concentrated on the base region of the tractor trailer. The GA was used to determine the optimal size, position, and curvature of the slats for base drag reduction. References 24, 28, and 29 describe previous efforts of drag reduction by using base slats. A front end geometry previously shown to produce no flow separation was employed. In order to reduce CFD run times, only half of the model was meshed and the symmetry boundary condition was used. A total of 5 variables were used for the modeling of the base region slats. Only symmetrical flap configurations were considered. Figure 2 shows a schematic of the base slat configuration.
II. Base Slat Optimization

This section details how the optimization was set up prior to the GA run. Base slat parametrics are discussed in section A. The grid details, are presented in section B. Section C covers a grid refinement study. Convergence criteria for the CFD runs are discussed in section D while the GA structure and computational expense is discussed in section E.

A. Base Slat Modeling

Figure 3 illustrates the base region of the two dimensional tractor trailer model used in this study. The dotted line represents a line of symmetry used to reduce computational expense. A cubic polynomial is used to describe the slat geometry here, hence six variables are shown. Five of the six variables shown can completely describe the curvature, location, and length of the slat. $\beta$ and $H_2$ are redundant. $\theta_1$ and $\theta_2$ are initial and final angles of the slat with respect to the horizontal. $L$ is the length the slat extends past the rear of the trailer. $H_1$ is the distance from the centerline to the point of attachment of the slat. $H_2$ is the vertical distance from the centerline to the end of the slat. Due to federal regulations, $L$ is limited to four feet.

Initially, $H_2$ was planned to be non-dimensionalized by $H_1$ and allowed to vary between zero and one. After grid convergence issues arose, the variable was changed to $\beta$, the angle between the attached end and free end of the slat. $H_2$ and $\beta$ are related by:

$$
\tan \beta = \frac{H_1 - H_2}{L}
$$

(1)
In this manner, $\beta$ is limited to a reasonable angle to avoid CFD convergence problems associated with large angles.

$H_1$, $\beta$, $\theta_1$, $\theta_2$, and $L$ provide just enough information to fully define the cubic curve. $H_1$ and the length of the truck provide a location for the first point. $H_2$ and $L$ provide a location for a second point of the curve. Finally, the initial and final angles, $\theta_1$ and $\theta_2$, are needed to define the curvature. The following process was used to convert the variables into coefficients used for the cubic. The equation for a cubic curve can be written as

$$y = Ax^3 + Bx^2 + Cx + D \quad (2)$$

The coefficients were solved as functions of known values for starting and ending coordinates, $x_1, x_2, y_1, y_2$, and angles $\theta_1$ and $\theta_2$. The coefficients can be expressed as follows:

$$A = \frac{-2y_1 + 2y_2}{x_1^3 - 3x_1^2x_2 + 3x_2^2x_1 - x_2^3} + \frac{\tan(-\theta_1) + \tan(-\theta_2)}{x_1^2 + x_2^2 - 2x_1x_2} \quad (6)$$

$$B = \frac{3y_1(x_1 + x_2) - 3y_2(x_1 + x_2)}{x_1^3 - 3x_1^2x_2 + 3x_2^2x_1 - x_2^3} + \frac{-\tan(-\theta_1)(x_1 + 2x_2) - \tan(-\theta_2)(2x_1 + x_2)}{x_1^2 + x_2^2 - 2x_1x_2} \quad (7)$$

$$C = \frac{-6y_1x_1x_2 + 6y_2x_1x_2}{x_1^3 - 3x_1^2x_2 + 3x_2^2x_1 - x_2^3} + \frac{\tan(-\theta_1)(2x_1 + x_2)x_2 + \tan(-\theta_2)(2x_1 + x_2)x_1}{x_1^2 + x_2^2 - 2x_1x_2} \quad (8)$$

$$D = \frac{y_1(3x_1 - x_2)x_2^2 + y_2(x_1 - 3x_2)x_1^2}{x_1^3 - 3x_1^2x_2 + 3x_2^2x_1 - x_2^3} + \frac{-\tan(-\theta_1)x_1x_2^2 - \tan(-\theta_2)x_1^2x_2}{x_1^2 + x_2^2 - 2x_1x_2} \quad (9)$$

The coefficients are substituted into the simple cubic expression to solve for any $y$ on the slat as a function of $x$.

**B. Gridding**

Meshing the truck was not possible with a single zone “c” mesh. The lack of a tapered trailing edge created a necessity for two zones in the truck grid. Figure 4 shows how the truck mesh is broken into two zones. Indexing for zone 1 starts at the lower right corner of the zone and extends left for the increasing $i$ direction and up for increasing $j$ direction. For this problem, a typical mesh with a base flap consists of about 90,000 to 100,000 nodes total with approximately 60,000 to 65,000 in zone 1 and 30,000 to 35,000 in zone 2. A grid refinement study (Section 3.3) was performed to ensure that this number of grid points was sufficient. Zone 1 generally consists of around 430 nodes in the $i$ direction and 140 nodes in the $j$ direction which extends approximately two truck lengths outward from the truck surface.
Figures 5 and 6 show zone 2 in greater detail. The border between zones is a horizontal line if no slats exist. With the addition of slats, the entire region aft of the tractor trailer is shifted vertically causing a curved border between zones. Indexing starts in the lower left corner with increasing $i$ in moving right and increasing $j$ moving up. Zone 2 with a flap present typically consists of 300 nodes in the $i$ direction and 100 nodes in the $j$ direction. The far right side of the grid ends approximately two truck lengths aft of the rear of the truck. Note the high density of points close to the surface which extends above and to the right of the truck past where the walls exist. This was an unavoidable scenario due to the use of a structured grid.
Figure 7 shows a close up of the front end of the truck. Spacing in the *i* direction was held constant at just under one inch from the top of the truck’s curved corner in the increasing *i* direction all the way to the line of symmetry. The primary purpose of modeling the front end of the truck was to have an accurately developed boundary layer flow approaching the base, therefore, one inch spacing on the front end of a full scale truck model was deemed reasonable. Any drag error obtained by *i* spacing in the front end of the truck will be present on every model tested by the GA and will not affect which geometry is chosen as the optimal solution.

Spacing in the first nodes away from the walls was kept nearly constant at around 0.0022 inches in order to obtain $y^+$ values of 30 to 60 on the surface which was required for the wall functions. This number was calculated using the same process as the mirror optimization problem from reference 38. The chord length used for Reynold’s number calculation was the length of the truck while the desired $y^+$ value was between 30 and 60 rather than the $y^+$ of 1 used for the mirror. The $y^+$ difference with wall functions is necessary because a $k$-$\omega$ SST turbulence model is used in place of the two equation $k$-$\varepsilon$ turbulence model with integration to the wall. Figure 8 shows the detail in the grid at the upper right corner of the truck base region magnified greatly. The spacing was carefully set up on the borders of zones 1 and 2 to prevent drastic changes in aspect ratio in adjacent cells.

While the grid generator produces suitable grids for most areas within the intended design space, excessively skewed cells lead to numerical instabilities for certain slat geometries. Large angles with respect to the horizontal anywhere on the slat create significantly skewed cells. Figure 9 shows the area of concern for a high angled slat. Ideally, cell boundary lines should extend perpendicular to the surface to better capture the boundary layer aerodynamics. Slat angles were thus limited by the GA to avoid excessively skewed cells.

### C. Grid Refinement Study

A grid refinement study was conducted on the truck model without slats to ensure that the chosen grid spacing will give accurate results. Due to limitations on grid point spacing near the surface of walls, it was not possible to create the coarse mesh from the fine mesh by means of removing every other grid point in the *i* and *j* directions. The $y^+$ value near the surface of the wall needed to be the same for both the coarse and fine grids. The differences were mainly in the spacing of grid points in the base region and the spacing of points running parallel to the surface wall. Table 1 presents the number of grid points in both directions of both zones for the coarse and fine grids.

![Figure 7. Front End of Truck](image)

![Figure 8. Upper Corner of Truck Base Region](image)

![Figure 9. Potential Area of Highly Skewed Cells](image)
The spacing variables were tweaked in the grid generator routine to obtain a fine grid that had roughly twice the number of points in each direction as the coarse grid without changing the $y^+$ needed to accurately capture the boundary layer. The ultimate goal of this study was to show that no significant difference exists between the converged drag values of both the coarse and fine grids.

Figure 10 shows the coarse and fine grids compared next to each other. The coarse grid is on the left with the fine grid on the right. The front end and base region close-ups show the coarse grid has roughly twice the space between nodes compared to the fine grid in the direction parallel to the surface. Refinement normal to the surface can be seen farther away from the surface.

<table>
<thead>
<tr>
<th>Zone 1, $i$ direction</th>
<th>Coarse Grid</th>
<th>Fine Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>432</td>
<td>836</td>
</tr>
<tr>
<td>Zone 1, $j$ direction</td>
<td>141</td>
<td>269</td>
</tr>
<tr>
<td>Zone 2, $i$ direction</td>
<td>289</td>
<td>542</td>
</tr>
<tr>
<td>Zone 2, $j$ direction</td>
<td>103</td>
<td>183</td>
</tr>
<tr>
<td>Total Nodes</td>
<td>90679</td>
<td>324070</td>
</tr>
</tbody>
</table>

The spacing variables were tweaked in the grid generator routine to obtain a fine grid that had roughly twice the number of points in each direction as the coarse grid without changing the $y^+$ needed to accurately capture the boundary layer. The ultimate goal of this study was to show that no significant difference exists between the converged drag values of both the coarse and fine grids.

Figure 10 shows the coarse and fine grids compared next to each other. The coarse grid is on the left with the fine grid on the right. The front end and base region close-ups show the coarse grid has roughly twice the space between nodes compared to the fine grid in the direction parallel to the surface. Refinement normal to the surface can be seen farther away from the surface.

Figure 10. Coarse and Fine Grid Comparison
The drag coefficient for both cases converged to approximately 0.296. The drag convergence for the truck without slats is illustrated in Fig. 11. The fine grid required over 100,000 iterations to converge while the coarse grid required only about 25,000 iterations.

Table 2 gives a breakdown of viscous and pressure drag on the truck. Both the coarse and fine grids returned similar values for pressure and viscous drag. Viscous drag accounted for 13.5% of the total drag on the tractor trailer model. Due to the similar results of the coarse and fine grid, the coarse grid was determined to be sufficient for the calculation of drag. Slatted grids for the GA run use the node spacing properties of the coarse grid.

<table>
<thead>
<tr>
<th></th>
<th>Pressure Drag (N/m)</th>
<th>Viscous Drag (N/m)</th>
<th>Total Drag (N/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse Grid</td>
<td>223.733</td>
<td>34.588</td>
<td>258.321</td>
</tr>
<tr>
<td>Fine Grid</td>
<td>223.470</td>
<td>34.969</td>
<td>258.439</td>
</tr>
</tbody>
</table>

D. Convergence Criteria

Many CFD runs were conducted to determine optimal convergence criteria. Since Fluent does not have an option to set convergence based on a drag value, a residual convergence for continuity, x-momentum, y-momentum, k, and ω based on where drag was suitably converged was employed. Two test cases with the same slat and different base spacing parameters were run to test sensitivity of the grid spacing. The drag histories for both cases are shown in Fig. 12. Both cases converged to the same drag value and required approximately the same number of iterations, thus showing the coarser of the two grids was sufficient.

Residuals for the coarse base spacing grid are shown in Fig. 13. Continuity converges long before drag, but the other residuals converge just after drag. CFD residual convergence values for X-momentum equation, Y-momentum equation, k, and ω were set to values from Fig. 13 at 60,000 iterations.
iterations. Residual convergence for all equations was set to $1.00E-7$.

Initial GA run attempts showed that the selected residual convergence criteria were not universally effective throughout the entire design space. Frequently cases would hit the residual convergence criteria long before drag converged. This resulted in the GA choosing its best performers to be cases that had not converged and had drag values that were increasing with iteration.

The fix for this problem involved creating a simple subroutine to check the drag convergence from the outputted drag file after the case converged. If the drag was found to vary by more than 0.5% over the last 3000 iterations, the case was “disqualified” by setting the drag to a high number rather than taking the final drag value. The cases could have been run longer rather than thrown out, but there was no guarantee that a case would ever get to the chosen convergence criteria.

Setting the drag to a high value essentially kills off traits that cause difficulty in drag convergence as the GA progresses through generations. The obvious downfall to this approach is the possible elimination of a good solution just because it is ill conditioned for drag convergence. Testing all cases that did not converge would require additional computational expense and may or may not provide useful results. Further investigation in this area is recommended.

### E. Code Structure and Computational Expense

The computations were performed on a 60-processor Linux cluster located in Auburn’s Samuel Ginn College of Engineering. The cluster has 30 nodes, where each node is composed of two AMD Opteron 242 (64-bit) chips, for a total of 60 available processors. Communication between the nodes is achieved with high-speed Infiniband interconnects, and the cluster has a total of 0.5 terabytes of disk space.

Initialization errors often occurred when attempting to solve for the flowfield using one case with more than four processors on the cluster. Since these errors could be prevented by running the same case with fewer processors, the problem was believed to be related to the domain decomposition of the given grid among the number of specified processors. Because of this error, the maximum number of processors allowed for each case was capped at four. Run times for a single geometry typically ranged from 10 to 12 hours while running in parallel on four processors.

Initial GA run time calculations were found to be far too long when the CFD computations were run in serial (i.e., one at a time). 15 generations with 20 members in each generation would take approximately 3600 hours or 150 days. It was necessary to modify the GA to have the ability to run multiple cases at a time. Since the objective function consisted of calling a script to execute Fluent, it was possible for the code to run more than one case at a time without using parallel computing such as MPI.

The GA was configured with five available slots for CFD runs. At the beginning of the generation, the objective function was called five times, once for each of the five slots. Whenever a case finished, it was replaced with another case until all 20 members had been run through the objective function. One generation typically took two days to complete (12 hours * 20 members / 5 at a time = 48 hours). A 15 generation GA run would take about one month to complete on 20 processors. If larger computing resources were available, the run times could be reduced to approximately one week by running all 20 members of the population at once.
Figure 14 shows the flowchart for the revised GA. The only changes made to the GA were implemented in the subroutine where the objective function was called. Due to shared cluster usage issues, the code was configured to change the number of processors being used at any time during a GA run. This was accomplished by changing the scripts which initiated Fluent. The scripts were called from the FORTRAN code, but unlike the code itself, the scripts could be edited while the code was running to change the number of processors used to solve each case. The changes would take effect the next time the GA loaded a case.

The design space for the slat optimization is given in Table 3. After initial CFD testing, it was found that large angles in the slat caused the CFD solution to diverge due to gridding issues. Conservative limitations were placed on the ranges of $\theta_1$, $\theta_2$, and $\beta$. Finding an optimal slat within the limited design space was considered more important than expanding the design space and potentially allowing the GA to converge on an erroneous CFD run.
III. Results

The tractor trailer model without slats produced a drag coefficient of 0.296. For comparison, reference 24 gives drag coefficients ranging from 0.277 to 0.474 depending on the turbulence model for a simplified, streamlined three-dimensional tractor trailer. The upper value of 0.474 comes from a model with known problems with drag predictions on similar flows. Reference 24 also gives 0.6 as the wind averaged drag coefficient of a typical Class 8 tractor trailer.

Figures 15 and 16 show the flow field around the front of the truck. All pressures shown in contour plots are in Pascals and are relative to freestream static pressure. Unless otherwise specified, all lengths are in meters with the front end of the truck having a longitudinal coordinate of 0 meters. No flow separation occurs at the front as expected. The maximum velocity of 55 m/s occurs at the corner of the “nose”. Stagnation occurs as expected at the flat portion of the nose along the centerline.

After ten generations, the GA managed to reduce drag by over 50% from the two-dimensional tractor trailer with no slats. The optimized member has a drag coefficient of 0.1216. By assuming the difference between the 0.6 drag coefficient from reference 24 and the 0.296 drag coefficient from the two-dimensional model without slats remains unaffected by the addition of slats, a rough estimation of drag reduction on a three-dimensional truck with protuberances can be obtained. The drag coefficient can be estimated to drop approximately 29% from 0.6 without slats to 0.426 with slats. For comparison, reference 29 gives wind tunnel test data for two unoptimized slat configurations that reduce the wind averaged drag coefficient by 16.4% and 18.8% from a baseline (no slats) configuration with a 0.59 drag coefficient.

The drag reduction for this “clean” vehicle can only loosely be related to the drag reduction for a complete operable rig for several reasons.

<table>
<thead>
<tr>
<th>H_1</th>
<th>maximum</th>
<th>minimum</th>
<th>increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.995</td>
<td>0.3</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td>\beta</td>
<td>15.0°</td>
<td>0.0°</td>
<td>0.2°</td>
</tr>
<tr>
<td>\theta_1</td>
<td>15.0°</td>
<td>0.0°</td>
<td>0.1°</td>
</tr>
<tr>
<td>\theta_2</td>
<td>15.0°</td>
<td>0.0°</td>
<td>0.1°</td>
</tr>
</tbody>
</table>
The “clean” vehicle is a simplified two-dimensional model of an overhead projection of a tractor trailer. A symmetry boundary condition was used down the centerline of the truck. No upstream protuberances exist, making the flow unrealistically streamlined as it approaches the rear of the truck. However, it is still important to note that even though this is just an estimate, the principle of substantial drag reduction using optimized slats has been clearly demonstrated.

Figure 17 shows the performance of each generation. A large drag reduction occurred after the 2nd generation, while each successive generation only slightly improved on the design. In looking at the geometry, this appears to be closely correlated to the value of $H_1$.

A total of five cases did not initialize or converge properly. They are not shown in the drag distribution plot due to the assigned penalty drag value. The five cases were from generations 1, 2, 5, and 10. All five cases converged properly when tested as individual runs outside of the GA. This meant the cases did not properly initialize while the GA was running, and could have failed due to a lack of available shared Fluent licenses on the cluster.

Figure 18 shows $H_1$ for each member of every generation. The GA quickly learns the benefit of placing the slat close to the edge of the truck. By the 4th generation, the GA rarely creates a member with the slat far from the edge of the truck. The slat can get close to the edge; however, due to limits of the grid generating subroutine, the slat can not consist of the top edge (maximum $j$) grid points in zone 2. The GA input file was set to allow $H_1$ range between 0.3 and 0.995 percent of the truck half width, with the latter value corresponding to 0.25 inches from the edge.

The $H_2$ distribution by generation is shown in Fig. 19. Initially a wide range of $H_2$ values exist due to the scattering of $H_1$ values. As $H_1$ approaches the edge of the truck, $H_2$ follows suit due to its dependence on $H_1$ by the angle between end points on the slat. Figure 20 shows how the angle between end points on the slat, $\beta$, varies with generation. Every generation has a $\beta$ of 13°, and low values of $\beta$ are eliminated by the 4th generation.

Figure 21 shows that $\theta_1$ moves toward small angles as the GA progresses. With the slat being placed on the edge of the truck, a small angle ensures that large flow separation will not occur at the beginning of the slat. One interesting detail is the string of larger $\theta_1$ values that survived up through the 8th generation. Nearly all of these cases had shorter length slats. In general, slats with shorter lengths tended to perform better with a larger $\theta_1$ values than slats with longer lengths. Flowfield illustrations with streamlines further validate this observance.

Figure 22 shows the distribution by generation of the slat final angle, $\theta_2$. $\theta_2$ approaches its 15° limit as the GA
progresses. After the 4th generation, all small $\theta_2$ values disappear. The convergence towards the maximum $\theta_2$ allowed by the design space indicates that there likely exists an optimal final slat angle greater than 15º. Due to CFD convergence issues, this design area could not be examined without a significant amount of additional work. However, in three-dimensional flows, it is unlikely that very large final turning angles could be maintained without flow separation, particularly in cross flow.

The slat length variable, $L$, does not have a strong tendency to converge to either a short or long slat as shown by Fig. 23. The last two generations indicate a shift towards a longer slat, but more generations are needed to determine whether this trend would continue. Because pressure drag is much more dominant than the viscous drag, intuition suggests the optimal member would have a large slat length. This would allow the slat to slow the flow down as much as possible without flow separation, thereby raising the base pressure. The GA converges to a range between 25 and 40 inches after the 4th generation.
The GA optimized slat is shown in Fig. 40 with its variables in Table 4. The slat is connected to the truck 0.816 inches from the edge and extends out with an initial angle of 1.5°. The total slat length is 36.5 inches.

The importance of the Coanda effect is illustrated by the geometry of the best performing slat. The Coanda effect is the tendency of a fluid to remain attached to a curved surface.\(^{26}\) Other than the small separated flow region created by the backward-facing step at the beginning of the slat (see Fig. 13), the airflow directly above the slat remains attached to the slat due to the pressure differential between the airflow above the slat and the vacuum that would be created on the slat if the flow were to become detached. Ultimately, the GA attempts to find a slat that keeps the flow attached to the surface while the slat curves toward the centerline and increases the static pressure of the flow. Previous studies have shown that two-dimensional RANS CFD solutions with a \(k-\omega\) turbulence model perform fairly well when predicting certain aspects (primarily the separation point) of Coanda effects.\(^{37}\)

Figures 25 and 26 contain drag and residual history for the optimized solution. The drag coefficient is fully converged by 50000 iterations. Residuals oscillated up until 15000 iterations before settling into a smooth trend. The turbulence residuals, \(\omega\) and \(k\), remain nearly constant from 15000 to 40000 iterations before decreasing throughout the next 20000 iterations. The drag coefficient appears to have a strong dependence on the continuity residual as both change very little after 15000 iterations.

<table>
<thead>
<tr>
<th>Table 4. GA Optimized Slat Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H_1) (from edge)</td>
</tr>
<tr>
<td>(\beta)</td>
</tr>
<tr>
<td>(L)</td>
</tr>
<tr>
<td>(\theta_1)</td>
</tr>
<tr>
<td>(\theta_2)</td>
</tr>
<tr>
<td>(C_D)</td>
</tr>
</tbody>
</table>

Figures 23 and 24. Slat Length Distribution by Generation

Figure 24. GA Optimized Slat

Figures 25 and 26. Drag and residual history for the optimized solution.
The wake regions of the no slat case and optimal slat case are shown in Fig. 27 and 28. Static Pressure differences on the base of the truck are drastically different for the two cases. With no slats, the static pressure drops dramatically as the flow separates at the aft end of the truck. The gauge pressure remains nearly constant along the base of the truck and throughout most of the wake at approximately -200 pascals. The slats force the pressure drop to occur before the base region of the truck while increasing the gauge pressure along the base of the truck to around -80 pascals. As expected, the optimal slat keeps the flow attached along its entire length. The main vortex length is reduced by 1.5 meters with the addition of the optimal slats.
Figures 29 and 30 zoom in on the counter rotating vortices in the corners of the slat. The grid resolution is fine enough to capture four distinct vortices in the wake region on the truck. The first is the large clockwise rotating vortex spanning over four meters in the wake. A second half meter vortex is captured just underneath the slat rotating counterclockwise. The third and fourth vortices both rotate clockwise and are wedged in the corner of where the slat meets the truck. They are approximately 1-2 inches in size.

While the winning case mentioned above has the least drag of all the GA cases, its length makes it more impractical than some of the shorter members examined by the GA came across. Generation 5 had a slat design with a drag coefficient of 0.138 (compared to 0.122 for the Generation 10 case) and a length of 17 inches. Figure 31 depicts the short slat while its variables are shown in Table 5. The short slat is positioned at the same place, but starts with a greater initial angle than the long slat.

<table>
<thead>
<tr>
<th>Table 5. Effective Short Slat Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_i$ (from edge)</td>
</tr>
<tr>
<td>$\beta$</td>
</tr>
<tr>
<td>$L$</td>
</tr>
<tr>
<td>$\theta_1$</td>
</tr>
<tr>
<td>$\theta_2$</td>
</tr>
<tr>
<td>$C_D$</td>
</tr>
</tbody>
</table>

Figure 29. Close-up of Optimized Slat Base (1.5m scale)

Figure 30. Close-up of Optimized Slat Base (0.3m scale)

Figure 31. Effective Short Slat
Figure 32 shows the flowfield around the base region of the short slat. Static pressure on the base of the truck remains around -100 Pascals gauge while the drag coefficient is 0.138. The wake zone extends five meters downstream of the truck.

After obtaining a GA solution, two test cases were run to see if the solution could be improved upon. Generally, if given enough generations, the GA will find the optimal solution within the design space. Due to the grid concerns, the design space was limited. Figure 33 shows two geometries selected based on a combination of intuition and general convergence direction of the GA. Both geometries exceed the maximum angle limits set within the GA run. Slat variables are given in Table 6.

Experimental slat “A” performed better than the optimal solution from the GA with a drag coefficient of 0.107. Slat “B” did not perform very well and had a drag coefficient of 0.175. Static gauge pressure on the base of the truck was around -40 pascals with slat “A” and -100 pascals with slat “B”. Unlike the optimized GA slat, flow separation at the free end of the slat occurs on both experimental slats. It is also important to note that greater flow separation is more likely to occur in a three-dimensional analysis with cross flow and a rougher upstream geometry. Pressure contours and streamlines for the experimental slats are shown in Fig. 34 and 35.

<table>
<thead>
<tr>
<th>Table 6. Experimental Slat Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>H₁ (from edge)</strong></td>
</tr>
<tr>
<td><strong>B</strong></td>
</tr>
<tr>
<td><strong>L</strong></td>
</tr>
<tr>
<td><strong>θ₁</strong></td>
</tr>
<tr>
<td><strong>θ₂</strong></td>
</tr>
</tbody>
</table>
IV. Conclusion

A CFD driven GA optimization has been performed on a simplified tractor trailer design. The task involved writing a customized grid generator for the model truck and slat capable of producing a suitable grid for any geometry within the design space. The grid generator coupled with a CFD solver was used as an objective function for a GA. Other components of the objective function included a drag convergence check and several scripts and batch files needed for communication between Fluent and the GA. Several scripts and batch files were required to activate the CFD solver within the objective function. The GA was modified for the base slat optimization to operate in parallel on a cluster of Linux processors to make computational run times manageable. A CFD driven GA optimization has been shown to be an effective means of drag reduction for a simple geometry. For more complicated geometries, computational expense and grid generation capabilities quickly become the limiting factor in the analysis.

The base slat optimization yielded a useful result. Drag was reduced by 59% from the clean two-dimensional model with the addition of a 36.5 inch slat and 53% with the addition of a 17 inch slat. By examining the traits of good performers, an experimental geometry was created outside of the tested GA design space that was shown to reduce drag by 64% from the clean two-dimensional model. For an encouraging but risky estimation for drag reduction on a real tractor trailer, the difference between the CFD runs without slats and with slats can be subtracted from known full tractor trailer drag coefficients. Using this extrapolation, the optimized base slat can be estimated to reduce drag by 29% from an operable tractor trailer rig.

For three-dimensional optimizations, a different grid generation approach would be much more practical. The majority of the time spent with the base slat optimization was coding and recoding the grid generator so that all cases within the design space would converge. A two-dimensional geometry with more variables or a three-dimensional geometry would be nearly impossible with the Cartesian gridding approach used in this study. Unstructured automatic grid generators would provide much more flexibility with the complexity of the geometry as well as the limitations of the design space.

Given successful results and financial motivation for even the smallest reduction in drag, future work in drag reduction via base slats is strongly encouraged. Recommendations for future work include:

- computations at multiple yaw angles to allow the computations of a wind-averaged drag coefficient
- computations at two different highway speed (e.g., 70 mph and 55 mph)
- extension of the work to more realistic 3D trailer geometries
- validation of turbulence model drag predictions when curved slats are included
References


