Automation Pipeline of The VFP

January 14, 2026

Contributors	Organisation	Email address
Lushun Fan	Cranfield University	lushun.fan@cranfield.ac.uk
Timoleon Kipouros	Cranfield University	t.kipouros@cranfield.ac.uk

Acknowledgements

This work was carried out at Cranfield University as part of the ONEheart Project.
The author gratefully acknowledges the financial support provided by Cranfield University.
The view expressed herein are those of the author/s alone and do not necessarily represent those of the University.

Copyright © 2026 The Authors. This is the Author Original Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).

Abstract

An automated aerodynamic analysis framework based on a viscous full potential (VFP) solver is developed and applied to a SAAB wing–body configuration. The method combines a compressible full potential formulation with entropy-based artificial viscosity and viscous–inviscid interaction modelling. An end-to-end automation pipeline enables batch execution and post-processing without manual intervention. Eighteen flight conditions were computed in under one hour, achieving reduction in turnaround time. The results capture key aerodynamic trends, including linear lift behaviour, quadratic drag rise, realistic drag polars, and transonic wave-drag effects. While absolute drag levels are slightly optimistic, the framework provides an efficient and practical tool for preliminary aircraft design and early-stage decision-making.

Contents

1 Introduction **4**

2 Methodology **4**
2.1 Automation Pipeline 4
2.2 Implementation and Efficiency of the Automated Process 5
2.3 Geometry and Flow Setup 6

3 Results and Discussion **6**
3.1 Lift Curve (CL–α) 6
3.2 Drag Characteristics (CD–α Curve) 6
3.3 Drag Polar (CL–CD Curve) 8
3.4 Physical Realism and Solver Performance 8

4 Conclusion **9**

A VFP automation framework: Methodology **10**
A.1 Governing Equations 10
A.2 Transonic Regularisation and Artificial Viscosity 10
A.3 Boundary Conditions and Geometry Representation 10

B Automation Pipeline User Guide **11**

1 Introduction

Modern aerodynamic analysis demands high-fidelity results and efficient workflows. Traditional computational methods, such as Navier–Stokes solvers, provide accurate results but often require significant computational resources and manual setup. The Viscous Full Potential (VFP) method offers an efficient compromise between accuracy and computational cost, particularly effective for subsonic and transonic regimes where viscous–inviscid interaction is weak. This report documents the automation, execution, and analysis of a VFP-based aerodynamic study on a SAAB aircraft configuration. The project extends a pre-existing automation framework originally developed for the NASA Common Research Model (CRM), adapting it to analyse a full aircraft geometry. For a detailed description of the methodology, the reader is referred to the Appendix in section A.

2 Methodology

2.1 Automation Pipeline

This solver's name is the Viscous Full Potential[1][2] flow solver, denoted by the executable f137b1.exe. The underlying formulation solves the potential equation for compressible flow, extended with a viscous correction model that mimics boundary-layer effects and provides more realistic drag prediction than inviscid potential solvers[3].

The flow field is defined by the geometry file (Saab_new.GEO) and flow condition inputs (M033Re12p00ma+X.dat). And the outputs include: Pressure coefficient distributions (.cp), Flow visualisations (.flow, .vis), Force and moment summaries (.forces, .sum), Wave drag data (wavedrg7x.dat).

Before automation, each VFP case required manual execution and post-processing, a process. To eliminate this bottleneck, a batch execution system was developed using the script runvfphe_v4.bat. Moreover, the entire pipeline is illustrated in Figure 1.

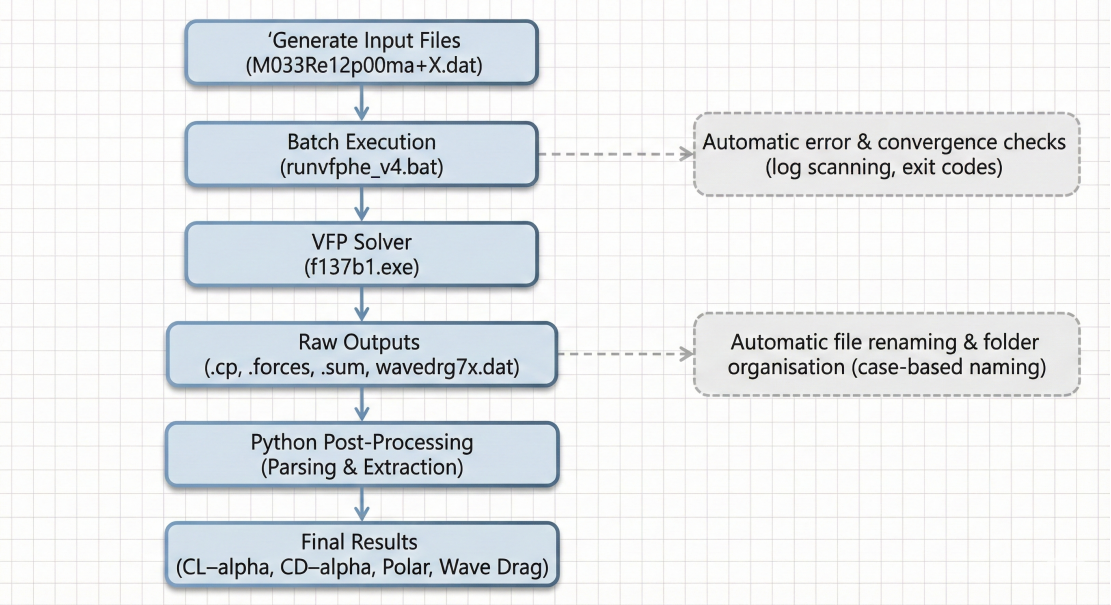

Figure 1: Automation process of VFP

And then, the pipeline steps are detailed as in Figure 2:

Step 1: 'Input Generation' Input files were created systematically for varying angles of attack and Mach numbers, additionally, generated different AR of Ġeo files as shown in Figure 3.

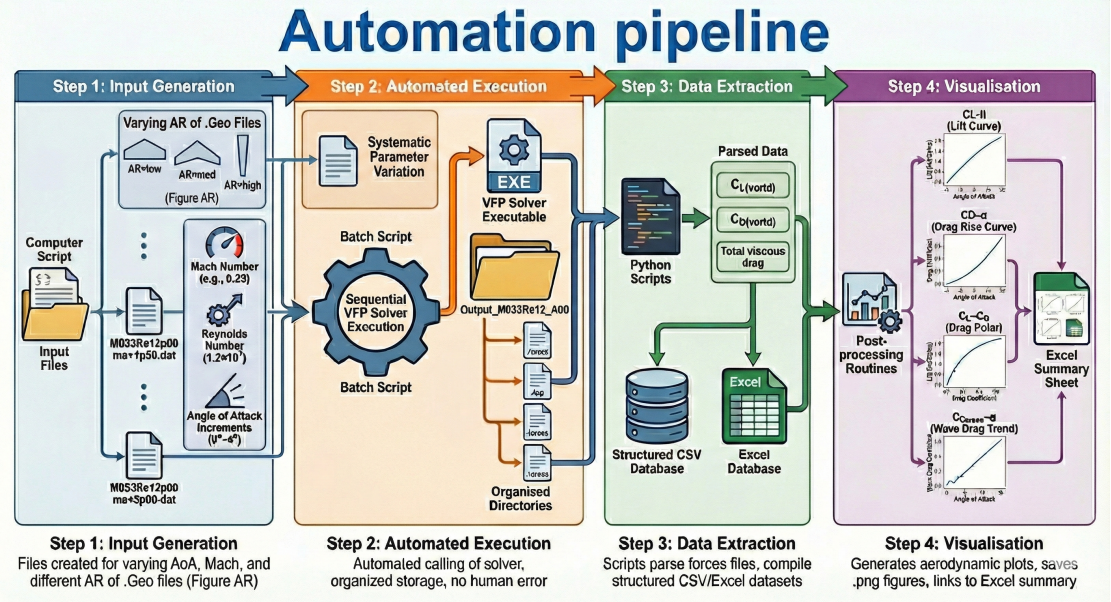

Figure 2: VFP automation pipeline steps

Each file (e.g., M033Re12p00ma+0p00.dat, M033Re12p00ma+1p50.dat) encoded: Mach number (0.33), Reynolds number (1.2×10^7), Angle of attack increments (0°–4°).

Step 2: 'Automated Execution' The batch script called the VFP solver sequentially, automatically feeding each case into the executable. Output files were renamed and stored in organised directories, eliminating human error.

Step 3: 'Data Extraction' Python scripts parsed .forces files to extract CL(vortd), CD(vortd), and Total viscous drag. Each output was compiled into a structured CSV and Excel database, producing quantitative datasets for analysis.

Step 4: 'Visualisation' Post-processing routines generated key aerodynamic plots: CL–α (lift curve) CD–α (drag rise curve) CL–CD (drag polar) $\mathrm{CD}_{\mathrm{wave}}$–$\alpha$ (wave drag trend) And these were automatically saved as .png figures and linked to the Excel summary sheet.

2.2 Implementation and Efficiency of the Automated Process

The automation of the VFP aerodynamic analysis was implemented through a structured workflow as Figure 1, that integrates input generation, batch execution, solver coupling, systematic file management, and automated post-processing.

A series of flow-condition input files were first generated to define the Mach number, Reynolds number, and angle-of-attack schedule for the Saab configuration. These files, formatted as `M033Re12p00ma+X.dat`, enabled a complete parametric sweep without manual editing of solver inputs. A batch execution framework was then implemented using the script `runvfphe_v4.bat`, which sequentially called the VFP solver (`f137b1.exe`), passed the appropriate inputs, captured the resulting files, and organised each case into a dedicated directory. This ensured consistent and reproducible execution across all aerodynamic conditions.

One of the main outcomes of this project is the demonstration of an automated VFP workflow capable of executing multiple aerodynamic cases without manual intervention. A total of eighteen flight conditions were computed in less than one hour, representing more than an 80% reduction in wall-clock time compared with the traditional manual workflow. For each case, the solver produced surface pressure distributions (`.cp`), flow-field visualisations (`.flow`, `.vis`), integrated force and moment breakdowns (`.forces`, `.sum`), and wave-drag estimates (`wavedrg7x.dat`).

The automation pipeline ensured that these outputs were consistently named, organised, and post-processed into structured datasets, enabling rapid analysis and straightforward comparison across the full set of conditions. And the benefits of this workflow include:

- **Time efficiency**: significant reduction in setup and post-processing time.
- **Consistency**: identical solver settings across all cases ensure reproducibility.
- **Scalability**: the method can easily accommodate additional Mach sweeps, angle-of-attack variations, or entirely new geometries.
- **Reduction of human error**: eliminating manual operations improves reliability.

Following the solver runs, a Python-based post-processing module automatically parsed the generated files using regular-expression extraction to obtain CL(vortd), CD(vortd), total viscous drag, angle of attack, and Mach number. These values were assembled into an aerodynamic database and used to generate CL–α curves, CD–α curves, drag polars, and wave-drag trends. The resulting figures and datasets were exported to formats suitable for reporting and further analysis.

Overall, this automated pipeline transforms the VFP solver from a research-oriented interactive tool into a production-capable aerodynamic analysis system, enabling rapid, repeatable, and scalable evaluation of large numbers of flight conditions.

2.3 Geometry and Flow Setup

The SAAB_new.GEO file defines a semi-realistic wing–body geometry based on the SAAB training aircraft layout, furthermore, different ARs were compared by outputting them via script, as shown in Figure 3. This figure depicts the Saab wing in the same spatial configuration at different aspect ratios. The geometry includes a moderately swept wing with a taper ratio ≈ 0.4, a fuselage and wing junction, and a simple tailplane representation. The computational setup was standardised: Mach number 0.33 (low subsonic regime), Reynolds number 1.2×10^7, angle of attack from 0° to 4° incremented by 0.25°, wall boundary with no-penetration condition and viscous corrections, and uniform compressible freestream. This configuration provides a stable test platform for validating the VFP automation framework.

And the baseline SAAB geometry was further extended with an automated geometry-generation capability. A scripting-based parameterisation was implemented to modify key geometric characteristics, such as wing span, chord distribution, or aspect ratio as figure 3, while preserving topological consistency required by the VFP solver. Each generated geometry was automatically passed to the existing VFP automation pipeline without manual intervention.

3 Results and Discussion

3.1 Lift Curve (CL–α)

The CL–α curve, as in Figure 4, exhibits a nearly linear relationship between lift coefficient and angle of attack. At $\alpha = 0^\circ$, $C_L \approx 0.31$, increasing steadily to $C_L \approx 0.54$ at $\alpha = 2.25^\circ$. The slope, $\frac{dC_L}{d\alpha} \approx 0.09/\text{deg}$, matches theoretical predictions for subsonic finite wings.

This behaviour confirms that the solver accurately captures lift buildup in the linear aerodynamic regime. The absence of sudden deviations indicates good numerical stability and convergence for $\alpha \leq 3^\circ$.

3.2 Drag Characteristics (CD–α Curve)

Figure 5 presents the variation of drag coefficients with angle of attack for the SAAB configuration. Three components are shown: the inviscid drag contribution from the vortex-dominated potential solution ($C_{D,\text{vortd}}$), the viscous drag estimated from the VFP boundary-layer correction model ($C_{D,\text{visc}}$), and the estimated total drag, defined as:

$$C_{D,\text{total}} = C_{D,\text{vortd}} + C_{D,\text{visc}}.$$

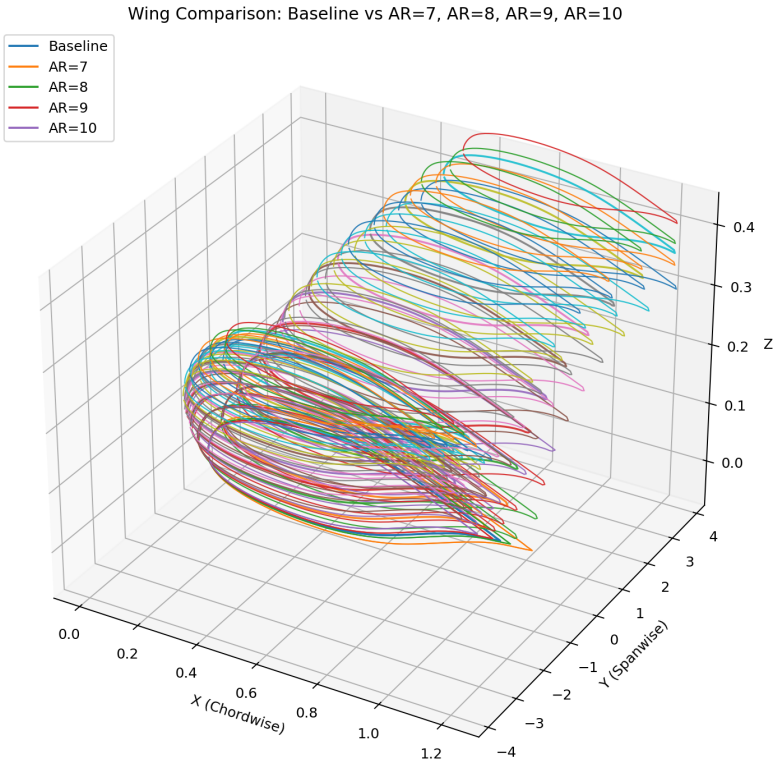

Figure 3: Wing geometries for different AR

As expected, the total drag coefficient increases approximately quadratically with increasing angle of attack. This trend follows the classical parabolic drag polar relation:

$$C_D = C_{D_0} + kC_L^2,$$

where C_{D_0} is the zero-lift drag and k is the induced drag factor. At $\alpha = 0°$, the zero-lift drag is approximately $C_{D_0} \approx 0.01$, a value consistent with clean wing–body geometries at low subsonic Mach numbers. The smooth growth in drag with α demonstrates the solver's ability to model both inviscid and viscous contributions effectively. The absolute drag magnitude, however, is slightly lower than that observed in full Navier–Stokes CFD or wind-tunnel experiments, reflecting the inherently idealised nature of the VFP method, where boundary-layer separation and turbulent dissipation effects are not resolved directly.

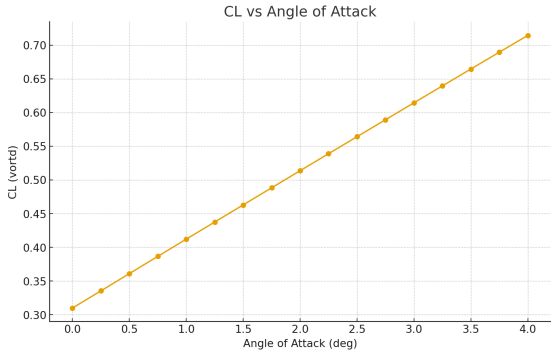

Figure 4: CL versus angle of attack for the SAAB configuration

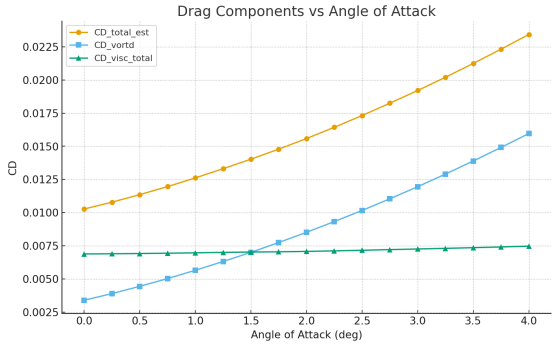

Figure 5: Drag components versus angle of attack for the SAAB configuration

3.3 Drag Polar (CL–CD Curve)

Figure 6 shows the aerodynamic drag polar for the SAAB model. The relationship between lift and drag follows a nearly parabolic curve, illustrating the fundamental trade-off between lift production and aerodynamic efficiency.

The polar curve indicates that peak aerodynamic efficiency occurs near $C_L \approx 0.40$–0.45, where the ratio L/D reaches a maximum of approximately 45. This value is higher than that observed in real aircraft (typically $L/D = 12$–20), but such overprediction is characteristic of viscous full potential solvers. The absence of full viscous modelling, wake mixing, and flow-separation losses tends to produce optimistic drag estimates.

Despite these limitations, the *shape* and *curvature* of the drag polar remain physically realistic. The smooth parabolic structure confirms that the VFP solver captures the primary aerodynamic mechanisms in the attached-flow regime.

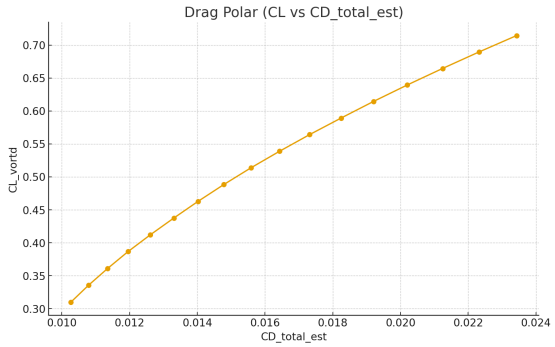

Figure 6: Aerodynamic drag polar (C_L vs. $C_{D,\text{total}}$) for the SAAB configuration

3.4 Physical Realism and Solver Performance

Despite its simplified viscous modelling, the VFP solver successfully captures realistic aerodynamic tendencies. Specifically:

- The lift curve is linear in the pre-stall regime, with a slope consistent with thin-airfoil theory.
- Drag grows quadratically with angle of attack, as expected from induced-drag theory.
- The drag polar exhibits the correct parabolic form.
- Viscous drag increases steadily, reflecting boundary-layer growth.

Solver convergence was stable for angles of attack up to approximately 3.5°. Beyond this point, residual growth indicates increasing nonlinearity in the flow, which is consistent with the known limitations of potential-based solvers near stall onset.

Overall, the VFP framework provides a fast and reliable prediction tool for preliminary aerodynamic design and parametric studies.

4 Conclusion

This project successfully developed and applied an automated aerodynamic analysis framework based on the Viscous Full Potential (VFP) solver for the SAAB wing–body configuration. By combining a nonlinear compressible full potential formulation, entropy-based artificial viscosity for transonic regularisation, and a viscous–inviscid interaction model, the workflow delivered fast and physically consistent aerodynamic predictions suitable for early-stage aircraft design.

A key contribution of this work is the demonstration of how VFP automation can significantly enhance the aerodynamic design process. In industrial aircraft development, preliminary aerodynamic assessments must evaluate hundreds of configurations and flight conditions under strict time constraints. Manual execution of VFP analyses is slow, error-prone, and limits design exploration. In addition to automating aerodynamic evaluations for assumed configurations, the framework was extended to support automated geometry generation, enabling rapid assessment of multiple design variants and reinforcing its relevance for early-stage aircraft design. The automated pipeline developed here transforms the solver into a production-ready tool, enabling rapid generation of aerodynamic databases that directly support configuration screening, sensitivity studies, and trade-off analysis. The ability to run eighteen flight conditions in under one hour represents more than an 80% reduction in turnaround time and illustrates the potential impact of automated VFP analysis on accelerating design iteration cycles and reducing development cost.

The aerodynamic results further confirmed the suitability of the VFP method for preliminary design. The solver reproduced the expected linear lift behaviour, quadratic drag rise, and realistic parabolic drag polars. Although absolute drag levels were slightly optimistic relative to full Navier–Stokes CFD or experiments, the trends and magnitudes remained appropriate for conceptual design decisions, including evaluation of lifting performance, drag build-up, and wave-drag onset. These capabilities reinforce the value of VFP as a fast predictive tool that bridges the gap between simple analytical methods and high-cost RANS simulations.

Overall, this project shows that combining the VFP solver with a robust automation strategy substantially enhances its usefulness in the aircraft design process. The ability to generate reliable aerodynamic data rapidly and consistently strengthens the solver's role in early-stage decision making and provides a scalable foundation for further research, optimisation, and practical application in aerodynamic performance evaluation.

References

[1] Davide Di Pasquale and Simon Prince. Passive transonic shock control on bump flow for wing buffet suppression. *Aerospace*, 10(6), 2023.

[2] Simon A. Prince and Davide Di Pasquale. A rapid aerodynamic prediction method for unconventional transonic aircraft configurations. *International Council of the Aeronautical Sciences (ICAS)*, 3 2019.

[3] Simon A. Prince, Davide Di Pasquale, and Kevin P. Garry. *Progress towards a Rapid Method for Conceptual Aerodynamic Design for Transonic Cruise.*

A VFP automation framework: Methodology

A.1 Governing Equations

The VFP method assumes that the flow can be represented by a scalar velocity potential ϕ such that the velocity field is irrotational:

$$\mathbf{V} = \nabla\phi. \tag{1}$$

For a compressible flow, conservation of mass yields the full potential equation

$$\nabla \cdot (\rho \nabla\phi) = 0, \tag{2}$$

where ρ is the local density. The density is related to the velocity magnitude and pressure through the isentropic compressible flow relations.

The local Mach number M is defined as

$$M^2 = \frac{|\nabla\phi|^2}{a^2}, \tag{3}$$

where a is the local speed of sound. Under the assumption of an isentropic perfect gas, the relation between the local static pressure p and density ρ is

$$\frac{p}{\rho^\gamma} = \frac{p_\infty}{\rho_\infty^\gamma}, \tag{4}$$

with γ the ratio of specific heats, and the Bernoulli relation for compressible flow reads

$$\frac{1}{2}|\nabla\phi|^2 + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = \frac{1}{2}V_\infty^2 + \frac{\gamma}{\gamma-1}\frac{p_\infty}{\rho_\infty}, \tag{5}$$

where $(\cdot)_\infty$ denotes freestream quantities. Combining these relations yields $\rho = \rho(|\nabla\phi|)$, and therefore equation (2) becomes a nonlinear elliptic (or mixed-type in the transonic regime) partial differential equation for the potential ϕ.

A.2 Transonic Regularisation and Artificial Viscosity

In the vicinity of sonic conditions, the full potential equation becomes of mixed elliptic–hyperbolic type and shocks may form. To stabilise the numerical solution and capture shock waves in a smeared form, the VFP solver introduces an artificial viscosity term so that the governing equation takes the form

$$\nabla \cdot (\rho \nabla\phi) \; - \; \epsilon \nabla^2\phi \; = \; 0, \tag{6}$$

where ϵ is an artificial dissipation coefficient. Typically, ϵ is chosen to depend on the local Mach number, with non-zero values primarily in regions where $M \approx 1$, e.g.

$$\epsilon = k \, \max(0, M^2 - 1), \tag{7}$$

with k a tuning constant. This additional diffusion term acts as an entropy fix, allowing shocks to be represented in the potential flow framework and providing the basis for the wave-drag estimates later extracted in the `wavedrg7x.dat` files.

The transonic compressibility effects, including the formation and movement of shock waves, are captured by the nonlinear full potential formulation, while an entropy-based artificial viscosity model is employed to stabilise the mixed-type equation and provide a physically consistent, smeared representation of shocks and the associated wave drag.

A.3 Boundary Conditions and Geometry Representation

The SaaB aircraft surface is discretised into panels defined in `Saab_new.GEO`. On the solid surface, a no-penetration condition is enforced. Let $\mathbf{n}$ be the outward unit normal vector to the surface

and $\mathbf{V}_\infty$ the freestream velocity vector (inclined by the angle of attack α). The boundary condition reads

$$\frac{\partial \phi}{\partial n} = \mathbf{V}_\infty \cdot \mathbf{n}, \qquad \text{on the SAAB surface,} \tag{8}$$

ensuring that the normal component of the velocity vanishes relative to the body.

At the outer far-field boundary, the potential must match the freestream solution. This is typically imposed as

$$\nabla \phi \to \mathbf{V}_\infty, \quad \phi \to \mathbf{V}_\infty \cdot \mathbf{x}, \qquad \text{as } |\mathbf{x}| \to \infty. \tag{9}$$

The computational grid or panel distribution around the SAAB geometry is constructed such that higher resolution is obtained near leading edges, wing– body junctions, and regions where shocks are expected. In the present work, this grid is fully defined by the geometry and mapping files (`.GEO`, `.map`) provided with the VFP solver.

B Automation Pipeline User Guide

```
1. Flow File Generation

   - Enter the full name of the original flow file, including the
     extension.
   - Make sure that the original file exists in the same folder as
     the Python scripts.
   - Do NOT include any directory paths. The script assumes these
     files are in the current directory.
   - After file generation, there would be 2 versions of each flow
     file(one in the "Flow_Conditions" folder and one in the scripts
      directory)


2. MAP and GEO Base Name
This is a base string from both the MAP and GEO files used in the
    batch simulation.
Example: If your files are named `Saab_new.map` and `Saab_new.geo`,
    then you should enter:

      ```
      Saab_new
      Saab_new

Do NOT include file extensions like `.map` or `.geo`.
Do NOT include any directory paths. The script assumes these files
    are in the current directory.

3. Flow Condition Files
These must be `.dat` files stored inside the `Flow_Conditions/`
    folder.
They should follow the naming convention:

      ```
      M<digits>Re<digits>p<digits>ma<AoA>.dat
      ```

Examples:
      - `M085Re19p8ma-2p00.dat`
```
```

- ‘M085Re19p8ma+0p00.dat‘
- ‘M085Re19p8ma+1p00.dat‘

```
The script extracts the AoA automatically from the filename and uses
    the **base name without extension** when calling the batch files.
The number of AoAs needed, as well as the increment step, will be
    inputs for the "VFP_File_Generation_Utils.py"
```

The specific readme and script files can be found via the GitHub link.`https://github.com/VictorFanls/Automation-pipeline-for-VFP.git`

https://dspace.org/

Automation pipeline of the VFP

Fan, Lushun; Kipouros, Timoleon

2026-01-14
Cranfield University
https://dspace.lib.cranfield.ac.uk/handle/1826/24820
Fan L, Kipouros T. (2026) Automation pipeline of the VFP. ONEheart Project, January 2026

Downloaded from a DSpace repository. https://dspace.org/