DOI: 10.13009/EUCASS2023-315
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

# Temporal Evolution of Transition Onset along Trajectories of Generic Flight Vehicles

*Frederik Jacobs*⋆† *and Johan Steelant*⋆
⋆*European Space Agency (ESTEC)*
*Keplerlaan 1, 2201AZ Noordwijk, The Netherlands*
Frederik.Jacobs@esa.int · Johan.Steelant@esa.int
†Corresponding author

## Abstract
To assess the location of transition on generic flight vehicles, a tool has been developed [5] which applies existing correlations to complex CFD solutions. This paper presents two improvements to this tool. First of all, a new interpolation algorithm which allows to evaluate the transition evolution of complex vehicles along their flight paths. Secondly, we present an improved methodology for the first estimation of the boundary layer thickness. A first validation case, based on experimental data found in literature, shows good agreement to the current methodology. Finally, a geometric analysis on a launcher fairing is presented as a first application and shows the strength of the tool.

## 1. Introduction

During the design of flight vehicles, quite some uncertainty exists about the onset of the laminar to turbulent boundary layer transition. In [9], it was highlighted that pending on the exact location of the transition onset, the vehicle weight could differ with a factor of two. Additionally, for future reusable spacecraft, the location of transition onset is equally important to estimate accurately the heat load at each position to size the thermal protection system accordingly. A lot of correlations exist to estimate the transition onset, but they usually require information like the displacement thickness, momentum thickness, wall or edge values,... which are not calculated by standard computational fluid dynamics (CFD) tools. Therefore, a tool [5], called Boundary Layer Identification and Transition Zone Detection (BLITZ) Code, was developed to post-process the data of a CFD simulation in such way that the existing correlations can be applied.

Starting from an existing laminar CFD solution, the code processes the data in different steps. At first the solution is mapped into an internal data structure. A first initial guess is made of the boundary layer thickness along the domain, based upon nearest neighbour projection. For each wallpoint an array of physical points is then created normal to the wall up to the detected boundary layer thickness (including an additional margin). Typically, about 100 points are used in the normal direction for each wallpoint. The flow properties at these points are then derived from the available CFD flowfield, for which two methods were implemented. The first approach consists in projecting the flow properties of the nearest CFD mesh node onto each of the array points. In this case the profile will not be smooth (typically for nearest neighbour projection) and therefore a reconstruction is applied to the variables to get a piecewise linear profile [5]. A second method is based on Delaunay triangulation which aims to group the three-dimensional CFD nodes such that non-overlapping tetrahedra are formed. This approach, performed using the Qhull library [2], allows more accurate property calculation but comes at a substantial increase of computational effort. Once the flow properties on the discretized boundary layer profile are known, the boundary layer thickness is calculated based on multiple selection criteria like maximal value, derivatives and so on as discussed in [5]. To assess the transition onset, the streamline length is calculated based on streamline tracing along the mesh for all wallpoints. Once all these data are reconstructed, correlations can be applied for natural and bypass transition as well as for estimations of the local critical roughness, backwards or forwards facing step etc.

In this paper two improvements on the existing tool are presented. A first improvement focuses on the determination of the properties of the discretized boundary layer points. Using nearest neighbour projection is fast but not very accurate. On the other hand, the linear interpolation using the Delaunay triangulation is more accurate but very expensive and can show accuracy problems under certain conditions. A new method, called the octant method, is proposed which can be smartly parallelized to increase the speed of the interpolation at a high accuracy. Additionally, an improved method for the initial boundary layer edge detection is implemented. To discretize the boundary layer, an initial guess is made. Therefore, an analysis is performed in the wall normal direction, ranging from the wall to

DOI: 10.13009/EUCASS2023-315



the edge of the domain, using a fixed number of points. When the boundary layer is very thin with respect to the domain, the initial guess can be far from the actual boundary layer thickness. To improve this, a method using multiple searches is developed. Doing this does not only improve the accuracy but also eliminates problems for blunt bodies in supersonic flows, in general improving the capability of the tool.

Applying simple correlations to complex flight vehicles might result in non-accurate results. To assess this problem, it is important to validate the developed tool. Yurchenko et al. [11] present transition data of a Russian rocket launcher. Based on an extensive literature search, the geometry of the vehicle could be determined and numerical simulations are compared to the presented data. During the initial design phase, designers might select geometry based on arbitrary choices. However, this might result in early transition. To determine the evolution of the transition location throughout a flight, a wraparound is developed which allows to process different transition analyses along the flight trajectory. In this way it is possible to design vehicles and do e.g., parametric studies to optimize the design. An example is presented for a launcher fairing, which shows that delaying the transition can be achieved by ensuring continuity in second derivative of the geometry.

## 2. Development

### 2.1 Octant interpolation

Using Delaunay triangulation becomes extremely expensive for three-dimensional applications. Furthermore, it provides accuracy problems for very finely meshed boundary layers. A more robust and faster algorithm, called the octant method, was worked out to avoid the aforementioned problems.

The methodology uses, for each point separately, the closest CFD node in each octant surrounding the selected point. A graphical representation of the octants is given in Figure 1[1]. The point, on which the properties are needed, is located in the origin while each octant is shown in a different color. To find the closest node in each octant, the distance to all CFD nodes needs to be calculated. However, as all the points are located along the same normal of one wallpoint, they are by definition close to each other. Therefore, only the CFD nodes close to this normal profile are of interest. In Figure 3 the green dots are the CFD mesh nodes while the red dots are the discretized points of the local boundary layer discretization along the normal profile for which interpolated values are needed. By only using the nodes located within the blue circle for the weight calculation, the calculation time is drastically reduced. In this case the distance to all CFD nodes with respect to the midpoint of the array only needs to be calculated once for each array. The cost for this expensive operation is divided over all points of the array. As the distance to the CFD nodes is calculated more quickly afterwards, a large speed up and therefore a faster interpolation are obtained.

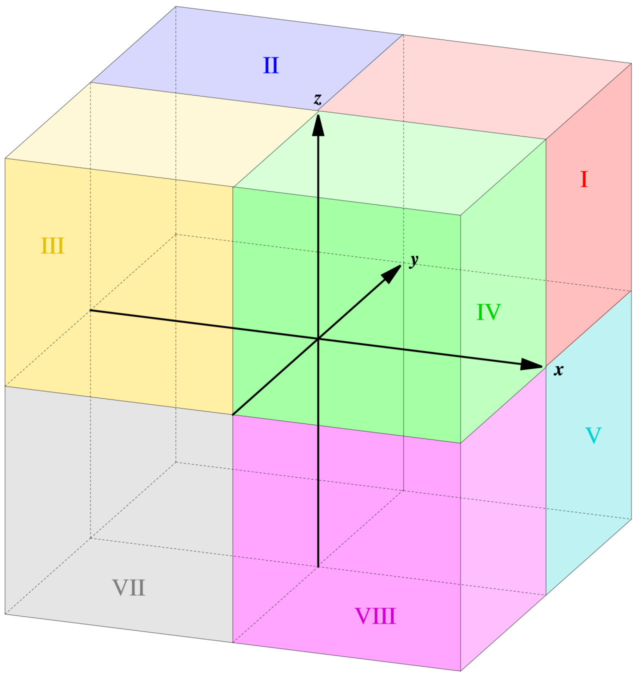



Figure 1: Graphical illustration of the different octants

[1]Obtained from https://commons.wikimedia.org/wiki/File:Octant_numbers.svg

DOI: 10.13009/EUCASS2023-315



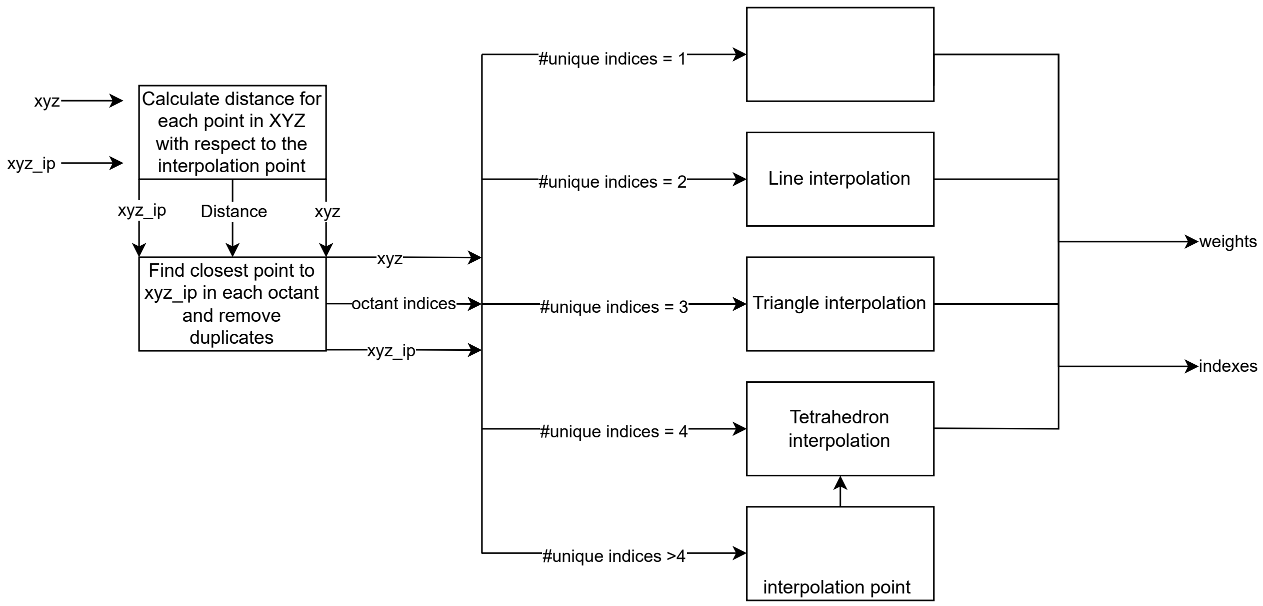



Figure 2: Graphical representation of the weight calculation function

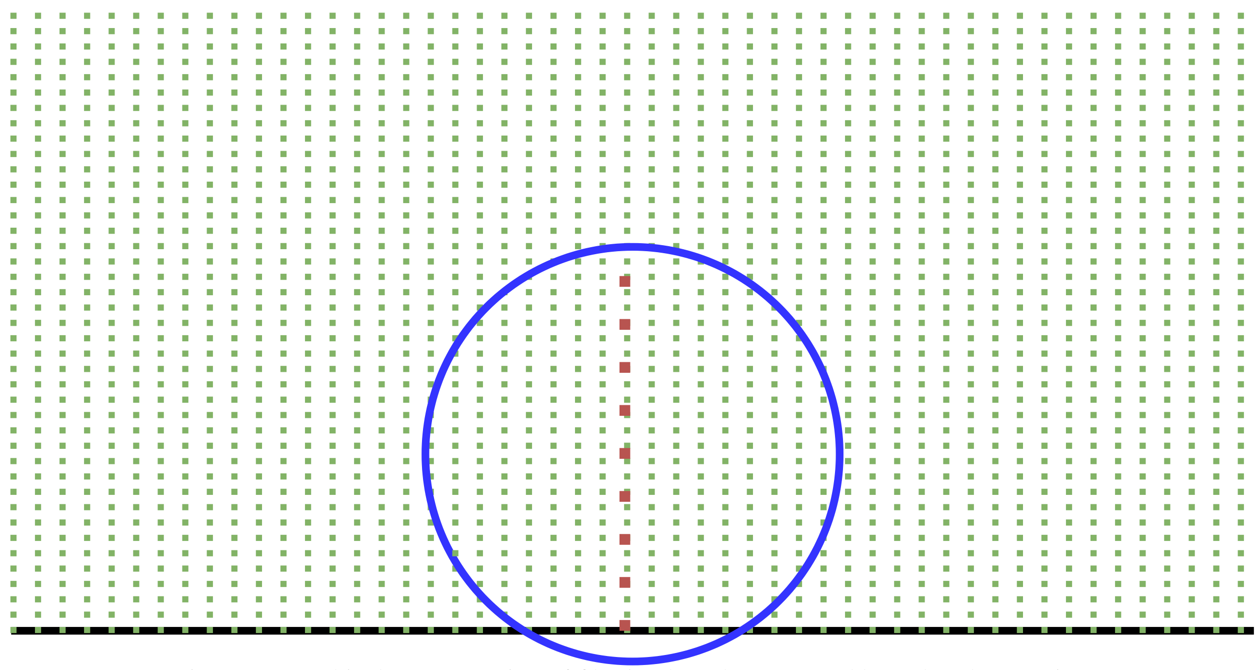


Figure 3: Graphical representation of filtering based on grouped boundary layer points

Finally, the interpolation is performed using the eight closest CFD nodes (one in each octant) only. As the octant contains its border planes (pre-defined in this way), nodes might appear multiple times. A distinction is then made based on the number of remaining unique nodes. For example, if only three distinct nodes remain, the point is then located within the triangle formed by these nodes. A simple two-dimensional interpolation in this triangle is then sufficient. The methodology for different number of unique nodes is graphically represented in Figure 2. For all potential cases, different checks are implemented to validate the assumptions. Furthermore, if no successful elements are found within the circle shown in Figure 3, the radius is increased in different steps to ensure a successful interpolation.

In Table 1 an overview of different test cases can be seen, including information about timing of the complete analysis. The total time is shown since for the nearest interpolation method additional time is spent throughout the code in reconstructing a piecewise linear profile from the original interpolation for the different boundary layer property profiles. It can be seen that for smaller test cases, i.e., the two flat plates, the interpolation time for nearest neighbour projection and the octant interpolation method are very similar. However, it is also clear that the interpolation time for linear interpolation using Qhull is significantly higher. When going to larger test cases it can be seen that the difference

DOI: 10.13009/EUCASS2023-315



in calculation time between nearest neighbour projection and octant interpolation is increasing, nevertheless also the accuracy is increasing drastically.

Table 1: Interpolation time comparison for different methods and test cases

| Case | Cores | Interpolation points | CFD nodes | Nearest | Octant | Linear |
|---|---|---|---|---|---|---|
| **Flat plate 01** | 12 | 176 800 | 195 838 | 3min34s | 3min42s | 10min51s |
| **Flat plate 02** | 12 | 199 160 | 426 620 | 2min46s | 2min41s | 12min32s |
| **Cylinder** | 12 | 4 084 340 | 3 468 778 | 39min25s | 52min49s | 184min40s |
| **Fairing** | 64 | 4 605 900 | 5 401 418 | 18min53s | 40min32s | Failed |
| **HXI** | 64 | 13 212 810 | 5 045 700 | 54min17s | 76min18s | Failed |

The test case called Cylinder is a cylinder of unit radius in a Mach 2 cross flow, of which the Mach contours can be seen in Figure 4. Table 1 shows that the time for octant interpolation is slightly above the nearest neighbour interpolation but still well below the time for linear interpolation. Note that the linear interpolation using Qhull results in different erroneous results due to accuracy problems. This is also confirmed in Figure 5a, where it is seen that the profile is far off from the other two. At a first glance the results for the nearest neighbour and octant interpolation seem to be very similar, which is due to the reconstruction. However, when zooming in (which is also shown in Figure 5a) it can be seen that the profile of the nearest neighbour method actually shows inaccurate oscillations. Figure 5b shows the derivative of the profile for both cases. It is clear that the newly implemented method increases the accuracy of the derivative of the profile drastically, which allows for more accurate boundary layer edge estimation.

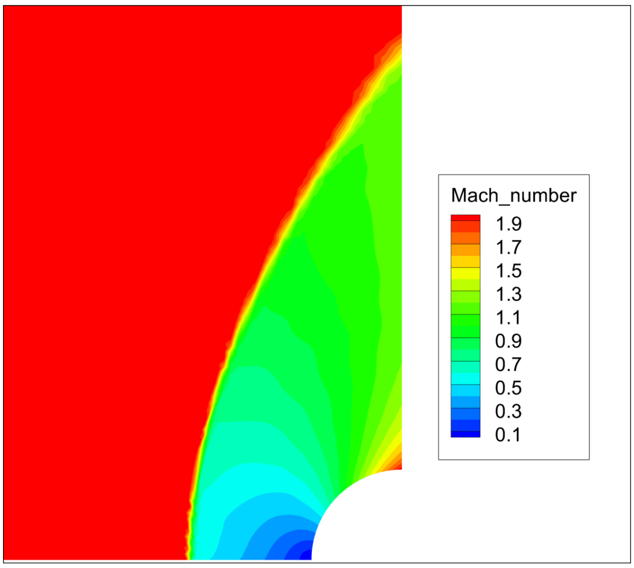



Figure 4: Mach contours of a cylinder in Mach 2 flow

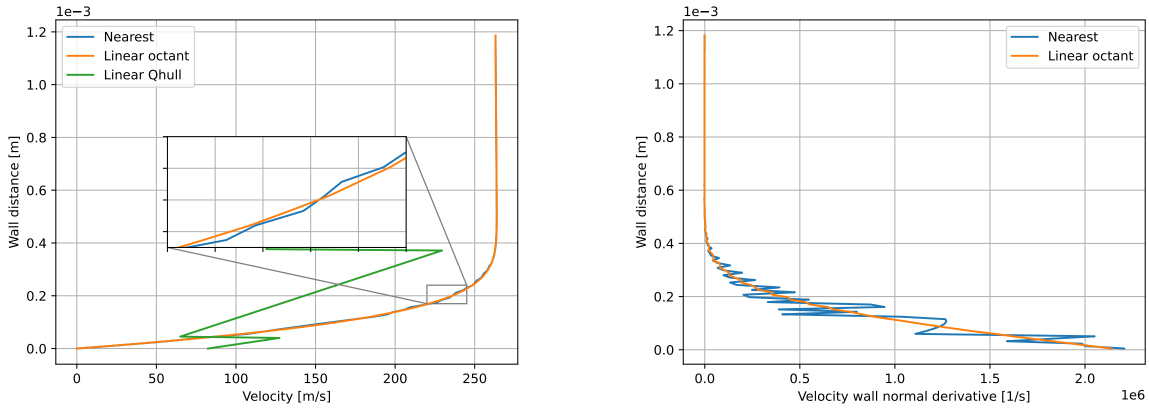
For cases with large differences in element size, e.g., the fairing test case, the Qhull based linear interpolation typically showed accuracy problems which were not able to be fixed. For the newly presented octant-based method, the linear interpolation does not suffer from this problem. Additionally, it increases the accuracy compared to the previous nearest neighbour methodology and reduces the amount of erroneous results. The presented method allows fast and accurate analysis of a solution, which opens the door towards analyses of complex flight vehicles along their complete trajectory.

### 2.2 Initial boundary layer guess

To calculate the boundary layer profiles, a discretization is created in the wall normal direction using a first estimate of the boundary layer thickness. This first estimate was originally made by discretizing until the border of the domain, using a fixed number of points. On this discretization the boundary layer thickness is estimated, which could then result in problems. A first problem occurs when the boundary layer is fully embedded within the first point away from the wall. For large domains, this can result in an initial guess being too far from the surface. An alternative problem

DOI: 10.13009/EUCASS2023-315






(a) Velocity (b) Velocity derivative in wall normal direction

Figure 5: Velocity profile and its derivative for nearest interpolation and linear interpolation using the octant method for the cylindrical test case

that was detected is the guess being at the final point of the interpolation, i.e., on the edge of the domain. Both effects result in a low accuracy on the final boundary layer edge detection and should therefore be avoided.

Since this first guess is essential for the proper final detection of the boundary layer thickness, an improved method is proposed. The initial detection is happening on the original, non-reconstructed, nearest neighbour projection properties. This interpolation is relatively fast and is therefore performed multiple times. A graphical representation of the newly implemented method is shown in Figure 6. Initially, a wall normal line is discretized up to the wall of the domain for each point. On this discretization, the boundary layer is detected using nearest neighbour interpolation. Afterwards, the boundary layer is refined by adding a layer equal to the 90th percentile of the detected boundary layer thicknesses (including a safety factor) as maximal interpolation distance over the complete vehicle, using the same number of points. This is done four times and shown as the red loop in Figure 6. Afterwards, the boundary layer profile is refined locally by discretizing up to the previously detected boundary layer thickness at that location. At this point the outliers, i.e., above 90th percentile and below 10th percentile, are corrected with a safety factor. This is represented by the green loop in Figure 6 and is repeated three times. Finally, the final boundary layer thickness is defined after which the boundary layer detection algorithm is ran using linear interpolation to increase the accuracy, making use of derivative based detection algorithms. In Appendix A the process is written down as implemented.

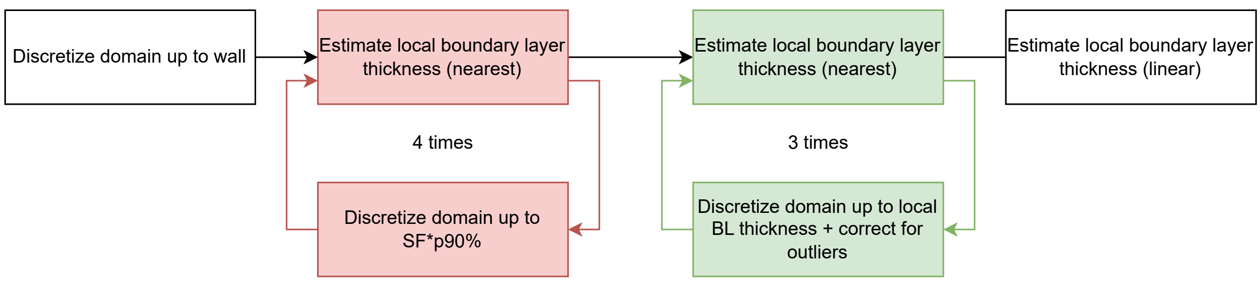



Figure 6: Improved initial boundary layer guess algorithm

## 3. Validation: comparison to flight data

### 3.1 Geometry definition

To validate the tool for launch vehicles, a comparison is made with experimental data presented by Yurchenko et al. [11]. Based on graphics presented in their report, as well as further investigation based on online news sources it is assumed that they report on the Angara 1.2pp launch on 9th of July 2014, a month before the first submission of the presented data. This assumption is based on graphical comparison of images presented in the paper and online graphics

DOI: 10.13009/EUCASS2023-315



of the Angara1.2pp vehicle as shown in Figure 7[2]. A good agreement between the different pictures is seen. The flight was carrying different sensors to provide technical data along the flight, which explains the provided data. As detailed information on the fairing geometry was missing, the technical drawings of the fairing were taken from the manuals of Rockot [10] and Angara [7].

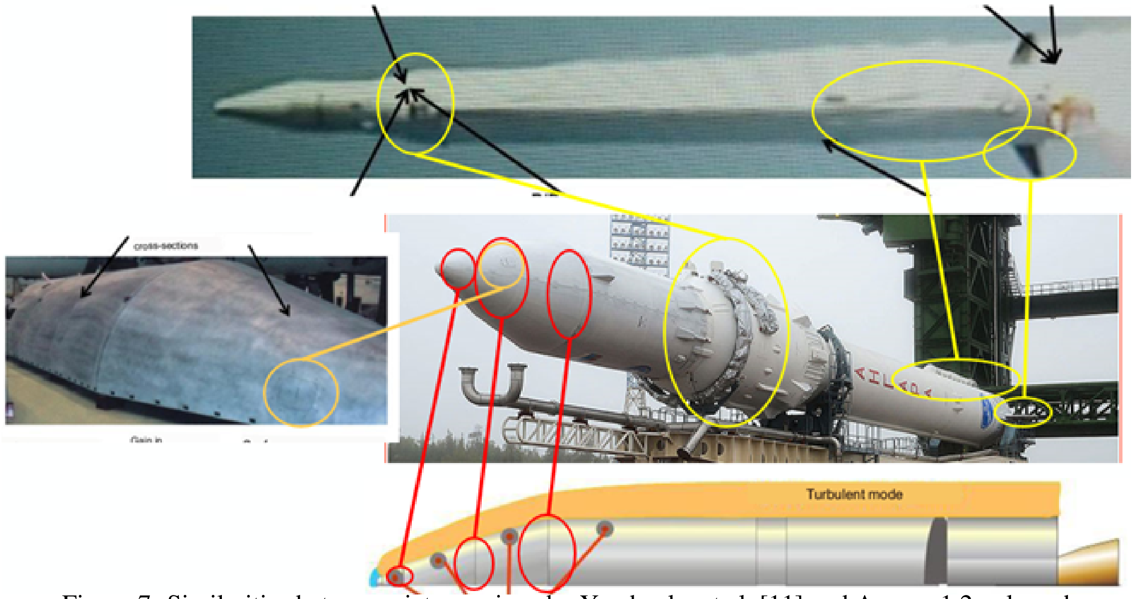



Figure 7: Similarities between pictures given by Yurchenko et al. [11] and Angara 1.2pp launch

### 3.2 Simulation comparison

In [11] it can be seen that the transition zone starts to move backwards from the nose tip at T+105 s at 36 km altitude and at Mach 5.6. The complete vehicle is fully laminar from T+120 s at 46 km altitude and Mach 7.7. To reproduce this transition shift, CFD simulations were run ranging from T+85 s to T+140 s, linearly extrapolating the altitude and Mach number based on the two provided points. First of all, an analysis on natural transition is performed based on the transition implementation by Karsch et al. [6]. Figure 8 shows the location of natural transition, based on the criterion by Bowcutt et al. [4] which uses Reynolds number based on streamline length and edge Mach number. In this case, the transition is already fully completed by T+105 s and thus does not match the experimental data. Therefore, a further assessment is made based on the allowed surface roughness.

Based on the correlation by Reda [8], the critical distributed roughness can be calculated at different points along the geometry. The selected locations are the middle of the front arc and the middle of the different conical sections as well as a position one meter behind the end of the last conical section. Figure 9 shows the critical roughness at these different locations. In the publication of Yurchenko et al. a critical roughness at the spherical cap of 0.1 mm is mentioned. At T+105 s the critical roughness at this spherical cap is reaching 0.1 mm, indeed confirming that further along the flight path natural transition takes over. Combining natural and roughness-based transition assuming 0.1 mm distributed roughness is shown in Figure 10 and shows indeed the transition location moving backwards almost instantly at T+105 s. However, the spherical nose cap is probably the most smoothly finished part of the fairing. Further downstream, higher roughness is probably present. Assuming the same 0.1 mm roughness up to the end of the first conical section, and 0.5 mm behind this point, the transition moves backwards more gradually and the vehicle reaches a fully laminar state at T+120 s as can be seen in Figure 11, matching the experimental data.

It can be concluded that the turbulent-to-laminar prediction can be predicted accurately using a combination of natural and roughness induced transition. Nevertheless, in this case it mainly shows that the methodology works for roughness induced transition and that the added vibrations due to the engines are not triggering earlier transition. To further validate the methodology, it would be required to evaluate multiple types of flight vehicles both for bypass and natural transition.

[2]The central figure is obtained from https://www.russianspaceweb.com/angara1pp.html

DOI: 10.13009/EUCASS2023-315



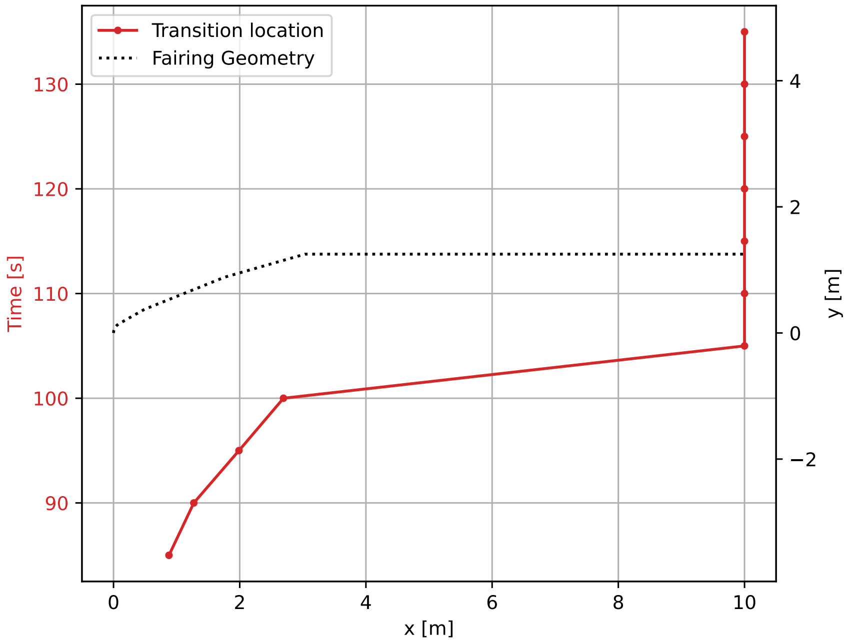



Figure 8: Natural transition location

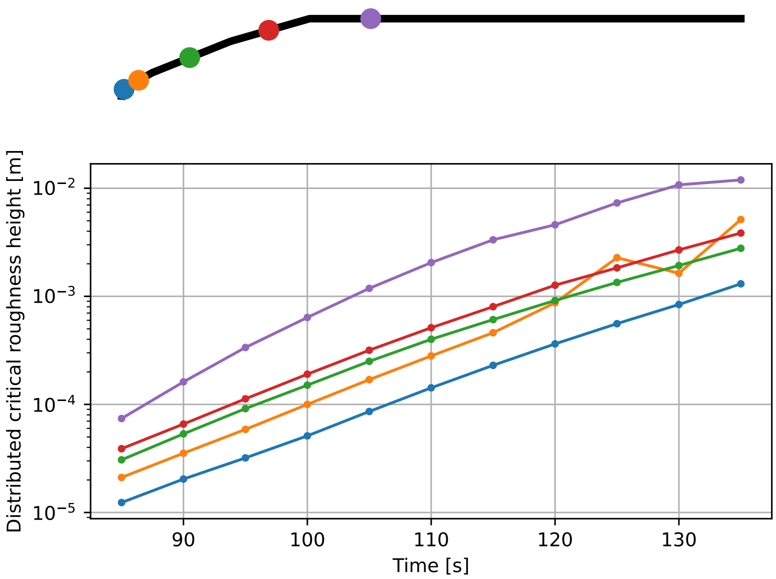



Figure 9: Critical distributed roughness

## 4. Application: blunted geometry

A study is performed to assess alternative fairing geometries with respect to transition onset.

### 4.1 Baseline geometry

A first assessment is made on the fairing of Ariane 5, used as a baseline case. The external geometry is ogive shaped [3] and has an outer diameter of 5.4 m and a length of 17 m [1]. The altitude and velocity of an Ariane 5 launch as a function of time are used to determine the environmental parameters. Simulations ranging from T+30 s up to T+150 s

[3]https://www.arianespace.com/mission-update/launcher-build-up-is-completed-for-arianespaces-dual-payload-ariane-5-flight-on-august-6/

DOI: 10.13009/EUCASS2023-315



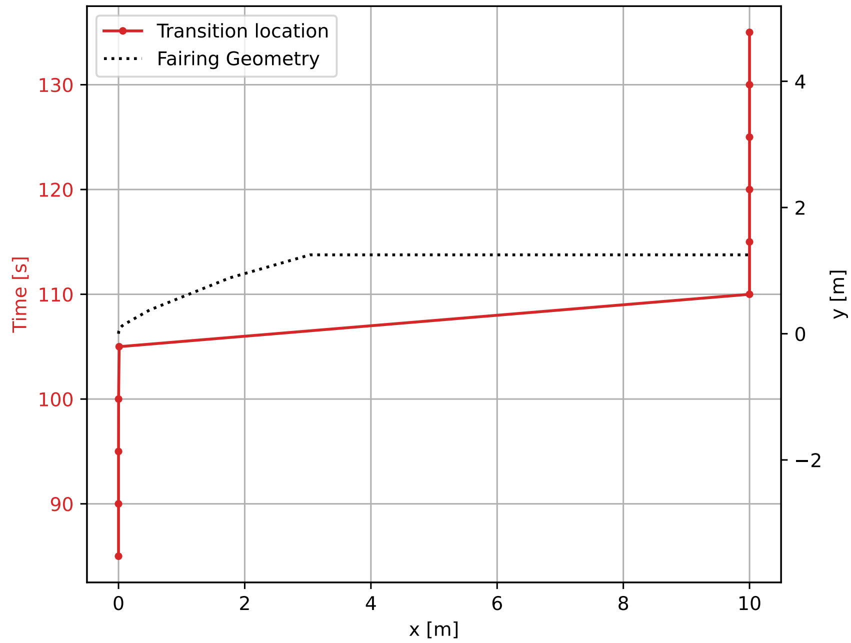



Figure 10: Combined natural and roughness induced transition location with uniform roughness of 0.1 mm

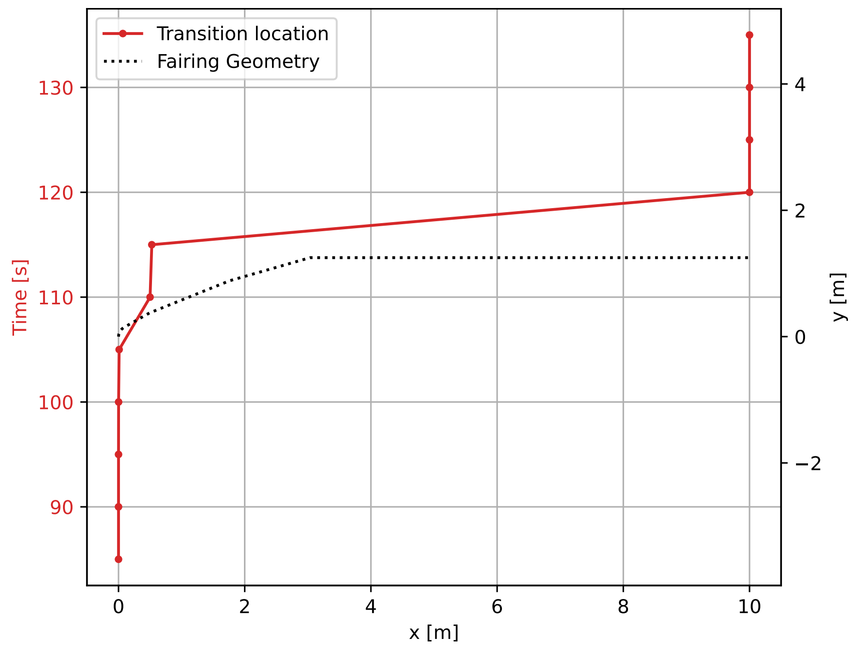



Figure 11: Combined natural and roughness induced transition location with 0.1 mm roughness for the first two sections and 0.5 mm for the other sections

are performed with a 2 s time step, corresponding to altitudes ranging from 2.1 km to 77.4 km and Mach numbers from 0.6 to 7. Figure 12 shows the altitude and Mach number as function of time as found in the Ariane 5 manual [1].

The location of natural transition (again based on the criterion by Bowcutt et al. [4]) is shown in Figure 13. Up to T+80 s the transition location is located at the connection of the two ogives. Afterwards, the transition location moves backwards up to T+98 s, at which moment the transition location stays for about 8s around the location of the connection between the second ogive and the cylindrical part. It is interesting to note that these locations are the locations behind a discontinuity in the second derivative of the profile.

Besides natural transition, it is also important to look at roughness induced transition. Therefore, one can look at different parameters like e.g., the critical distributed roughness which is shown at various locations in Figure 14, based

DOI: 10.13009/EUCASS2023-315



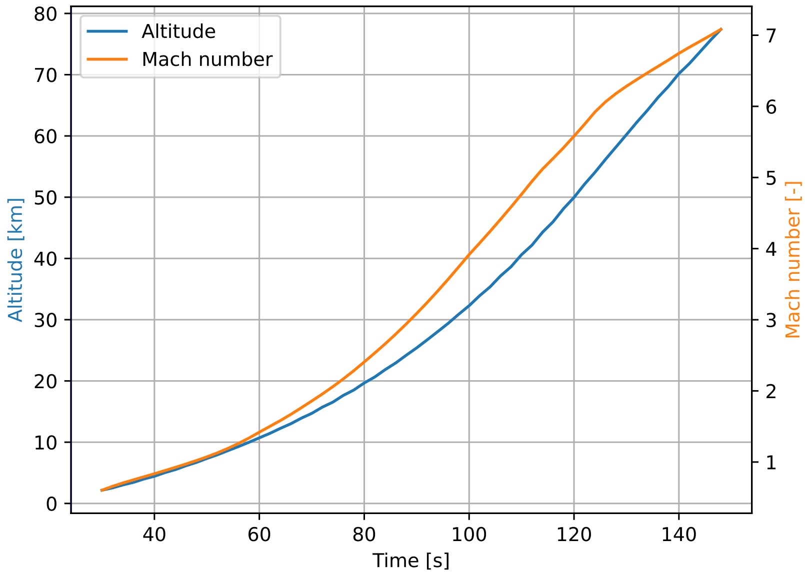



Figure 12: Mach number and altitude as a function of time for an Ariane 5 launch [1]

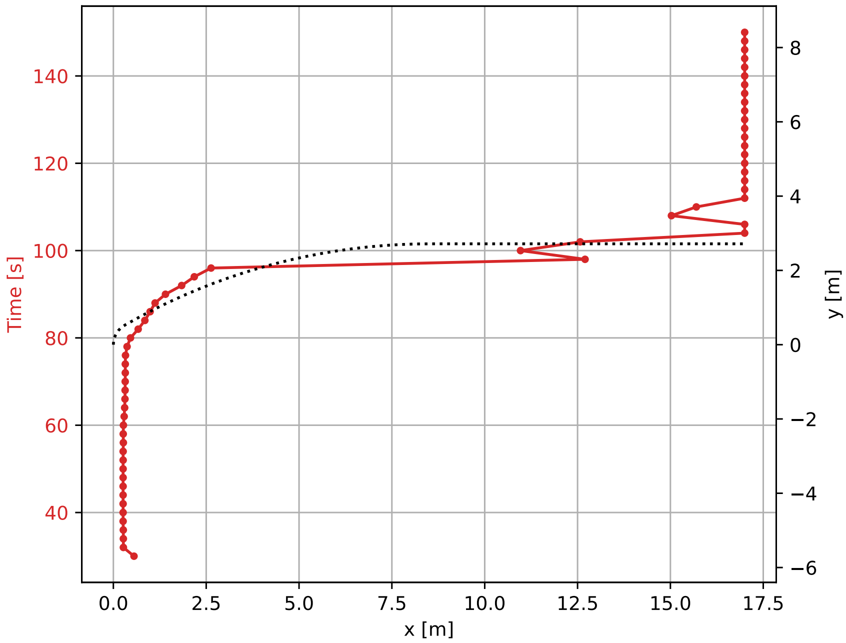



Figure 13: Natural transition location on Ariane 5 fairing

on the correlation by Reda [8]. Imagine a distributed roughness of 0.1 mm along the complete fairing. This would mean that transition would be triggered at the nose up to T+100 s. From that point on, the transition is defined by natural transition since all downstream locations have a higher critical roughness.

Other parameters like the critical discrete roughness, cavity length and depth or forward and backward facing steps can be considered as well. Based on these plots, it can be decided where the roughness should be reduced. In the given example, a roughness of 0.02 mm would result in a shift of the transition location backwards at about the same time as the natural transition would occur. This therefore means that it does not make sense to reduce the roughness at the nose below 0.02 mm since for lower roughness transition would still happen due to natural transition.

DOI: 10.13009/EUCASS2023-315



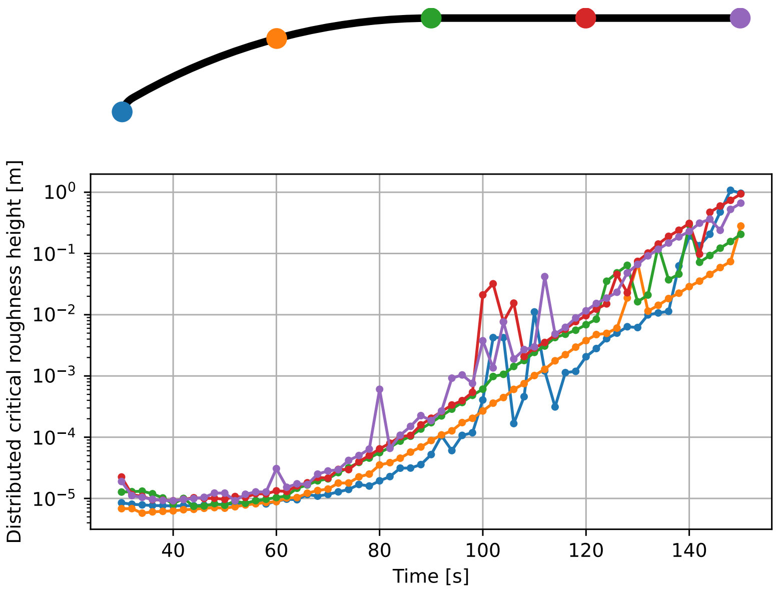



Figure 14: Critical distributed roughness on Ariane 5 fairing

### 4.2 Adapted geometries

As mentioned before, transition occurs slightly behind the locations of discontinuities in second derivative. We propose different alternative geometries which eliminate the discontinuity in second derivative along the geometry with a limited change in contour.

A first approach uses two cubics to match the first ogive to the cylindrical part. In this case the cubics are matched such that they are connected and have continuous first and second derivative both at the connection to the spherical cap and to the cylindrical part, as well in their connection location. Therefore, the connection location is considered as an additional unknown such that nine unknowns can be matched with nine equations. The length from the tip to the cylindrical part and the final angle of the spherical cap are then used to match the geometry as closely as possible to the baseline geometry.

A second case uses an ellipse which is defined below and uses a parameter $a$ to match the geometry coordinates ($x$ and $y$) as close as possible. In this case $r$ is the radius of the cylindrical section downstream of the curved fairing.

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{r}\right)^2 = 1 \tag{1}$$

A final case is based on a superellipse which is similar to a normal ellipse but using higher powers $m$ and $n$ as defined below.

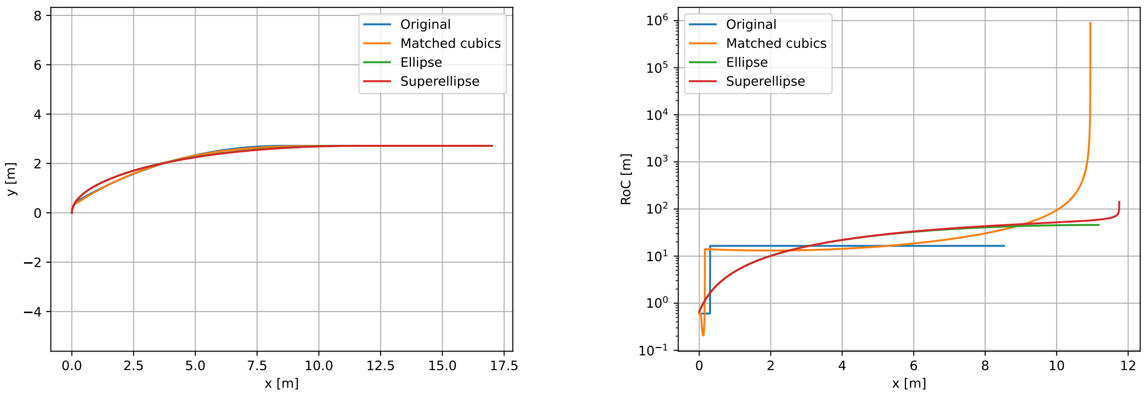
$$\left(\frac{x}{a}\right)^n + \left(\frac{y}{r}\right)^m = 1 \tag{2}$$

In this case if $n$ is not chosen equal to two, the second derivative becomes zero at the horizontal location. This results in a profile that is completely continuous up to the second derivative. Again, the different parameters are selected such that the difference with the original profile is as low as possible and $n$ is forced to be above 2.1, to ensure the second derivative going to zero. The original and the three adapted contours can be seen in Figure 15a, the local radius of curvature of the profile can be seen in Figure 15b. It can be seen that the radius of curvature increases steadily to infinity for the matched cubics and superelliptical case. For the others a jump is present at that location since an infinite radius of curvature (in the flow direction) is present at the cylindrical part.

Figure 16 shows the location of natural transition for the different curves. It can be seen that turbulent-to-laminar transition of the flight vehicle happens much earlier for the continuous shapes. Nevertheless, for all cases the transition location sticks around the connection to the cylindrical parts as here the radius of curvature is going to infinity and therefore transition might be triggered due to the fast increase in radius of curvature. This could be eliminated by using a superellipse (or similar shape) along the complete vehicle, eliminating this fast increase. However, a transition to a cylindrical section will always be required for practical reasons.

DOI: 10.13009/EUCASS2023-315






(a) Geometry (b) Radius of curvature

Figure 15: Geometry and radius of curvature for different alternatives for the Ariane 5 fairing

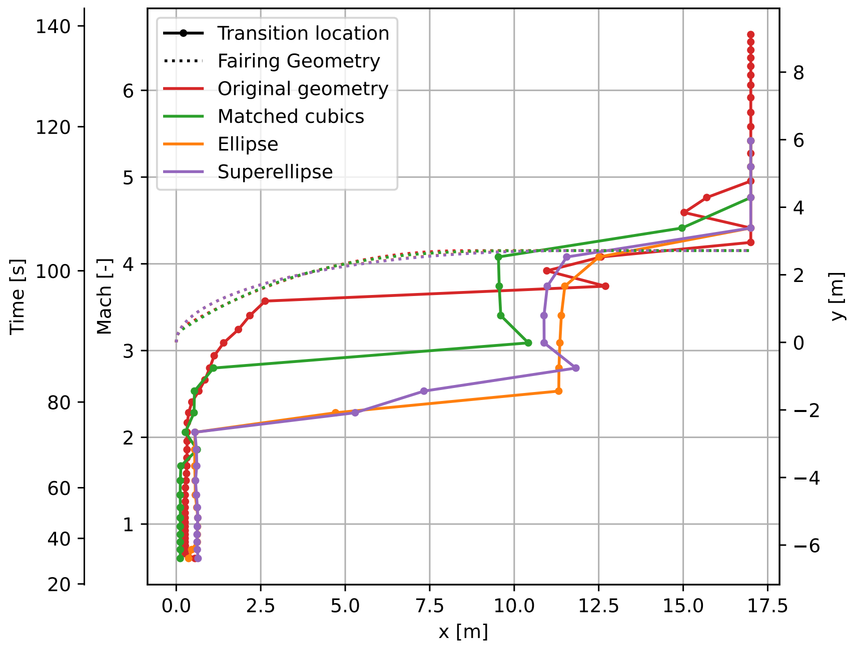



Figure 16: Natural transition location for different geometries for the Ariane 5 fairing

Figure 17 shows the critical discrete roughness along the geometry for the different geometries based on the correlation by Berry et al. [3]. The critical roughness is only very slightly influenced by the selected geometry. Therefore, it should be assessed whether changing the geometry is beneficial since transition might be induced by roughness rather than natural transition.

Note that for the Ariane 5 fairing (and blunt bodies in general) the pressure drag is much higher compared to the viscous drag and therefore delaying the transition will only have a minor influence on the total drag. Nevertheless, the heat load into the fairing as well as the acoustic load caused by the turbulent boundary layer are affected by the transition. This will be further investigated in future developments. With this first study we aim to show the strength of the tool which could be used for more complex flight vehicles to do a similar analysis in the future.

## 5. Conclusion

This paper presented several improvements on the existing transition analysis tool developed by Hoffmann et al. [5]. Most importantly, a new interpolation algorithm was implemented which is based on the closest point in each of the octants surrounding the point where one wants to interpolate. Since the points of the boundary layer array are close to each other, smart parallelization can be applied for which only a limited subset of the CFD nodes are considered as

DOI: 10.13009/EUCASS2023-315



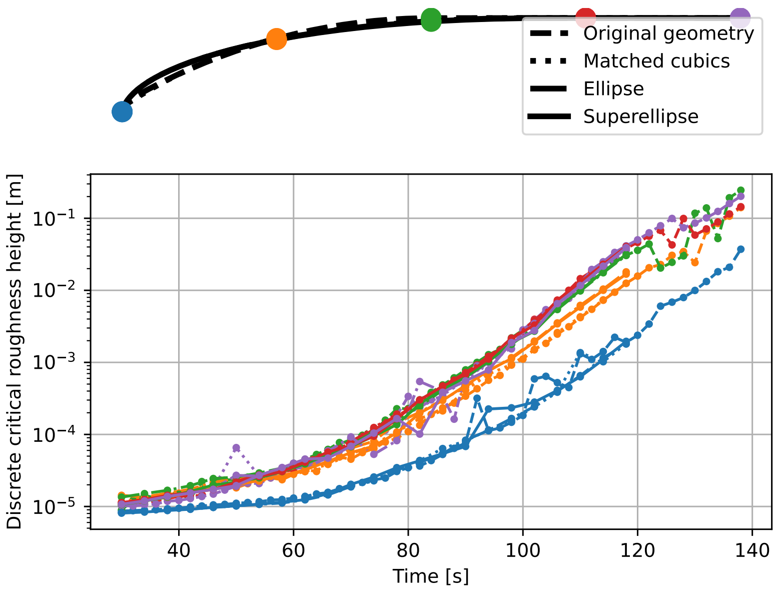



Figure 17: Critical discrete roughness for different alternative geometries for the Ariane 5 fairing

possible closest points. This method has the accuracy of linear interpolation at only a slight increase of calculation time with respect to nearest neighbour projection. A second improvement consists of a new initial boundary layer detection algorithm, which allows for more refined discretization of the domain and therefore a better detection of the finally selected boundary layer thickness and better calculation of the properties like displacement thickness, momentum thickness and so on.

To validate the code, a transition assessment is compared to experimental data published by Yurchenko et al. [11]. A good agreement is seen with the experimental data when both natural and roughness induced transition are considered. However, to further validate the tool a more extensive investigation is required for both natural and roughness induced transition for different kind of geometries. Nevertheless, a good first indication of the validity of the tool has been presented.

As a first application, a geometric assessment is performed on the Ariane 5 fairing. It has been shown that by removing the discontinuity in second derivative it is possible to have earlier turbulent-to-laminar transition. For this application, the influence on drag is limited due to the large influence of the pressure-based drag on blunt bodies, but a more severe impact might be present on the integrated heat load or acoustic load of the vehicle. Knowing the natural transition location could be very useful in an early stage of a flight vehicle design to immediately optimize the geometry. On top of this, knowing e.g., the critical backwards facing step can allow to design joints in such a way that the critical limit is not crossed and therefore no roughness induced transition is taking place.

To further validate the tool, it is recommended to validate over a wide range of geometries and flow conditions. Additionally, effort could be put in blending laminar and turbulent simulations to assess the forces on a flight vehicle and further investigate the heat flux and acoustic load into the fairing. Therefore, a blending methodology between turbulent and laminar CFD simulations will be developed. Finally, assessments of different vehicles along complete trajectories are planned to be performed.

## References

[1] Ariane 5 user's manual issue 5 revision 2, 2016. URL `https://www.arianespace.com/wp-content/uploads/2011/07/Ariane5_Users-Manual_October2016.pdf`.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22:469–483, 12 1996. ISSN 0098-3500. doi: 10.1145/235815.235821. URL `https://dl.acm.org/doi/10.1145/235815.235821`.

[3] S. Berry, S. Bouslog, G. Brauckmann, J. Caram, S. Berry, S. Bouslog, G. Brauckmann, and J. Caram. Boundary

DOI: 10.13009/EUCASS2023-315



layer transition due to isolated roughness - shuttle results from the larc 20-inch mach 6 tunnel. American Institute of Aeronautics and Astronautics, 1 1997. doi: 10.2514/6.1997-273.

[4] K. G. Bowcutt, J. D. Anderson, and D. Capriotti. Viscous optimized hypersonic waveriders. American Institute of Aeronautics and Astronautics, 3 1987. doi: 10.2514/6.1987-272. URL https://arc.aiaa.org/doi/10.2514/6.1987-272.

[5] P. Hoffmann, J. V. D. Eynde, and J. Steelant. Boundary layer and transition onset assessment on generic geometries. *CEAS Space Journal*, 2023.

[6] M. Karsch, J. V. den Eynde, and J. Steelant. Linearly combined transition model based on empirical spot growth correlations. *CEAS Space Journal*, 5 2023. ISSN 1868-2502. doi: 10.1007/s12567-023-00499-8. URL https://link.springer.com/10.1007/s12567-023-00499-8.

[7] A. Medvedev, A. K. Nedaivoda, M. F. Jensen, E. F. Laursen, and M. Albrecht. Angara launch system mission planner's guide, 12 2002. URL https://www.mach5lowdown.com/wp-content/uploads/PUG/Angara-Mission-Planners-Guide-Rev-0-2002-12.pdf.

[8] D. C. Reda. Correlation of nosetip boundary-layer transition data measured in ballistics-range experiments. *AIAA Journal*, 19:329–339, 3 1981. ISSN 0001-1452. doi: 10.2514/3.50952.

[9] J. F. Shea. Report of the defense science board task force on the national aerospace plane (nasp), 9 1988.

[10] Y. Viertel, M. Kinnersley, T. Miski, and P. Bamberg. Rockot user's guide, 8 2011. URL http://www.eurockot.com/wp-content/uploads/2012/10/UsersGuideIss5Rev0web.pdf.

[11] I. Yurchenko, I. Karakotin, and A. Kudinov. The turbulent-laminar transition on the rocket surface during the injection. *Science and Education of the Bauman MSTU*, 14, 9 2014. ISSN 19940408. doi: 10.7463/0914.0726268. URL http://technomag.bmstu.ru/doc/726268.html.

## A. Improved initial boundary layer guess algorithm

```
SF=2#adding a safety factor
xyz_ip=CreateInterpolationDiscretization(maxwalldistance,xyz,wptIndex)
BLThicknessGuess=MakeBLGuessUsingNearestIP(xyz_ip,xyz,uvw,reconstruction=False)

for j in range(4):
        discretizationlength=[percentile(BLThicknessGuess*SF,90),]*len(wptIndex)
        xyz_ip=CreateInterpolationDiscretization(discretizationlength,xyz,wptIndex)
        BLThicknessGuess=MakeBLGuessUsingNearestIP(xyz_ip,xyz,uvw,reconstruction=False)

for j in range(3):
        discretizationlength=BLThicknessGuess
        for i in range(len(wptIndex)):
                if BLThicknessGuess[i]>percentile(BLThicknessGuess,90)*SF:
                        BLThicknessGuess[i]=percentile(BLThicknessGuess,90)*SF #if extremly laying
                            outwards the point is assumed to be wrong and the guess is limited to
                            the 99th percentile, including a safety factor
                elif BLThicknessGuess[i]<percentile(BLThicknessGuess,10)/SF:
                        BLThicknessGuess[i]=percentile(BLThicknessGuess,10)/SF #if extremly laying
                            outwards the point is assumed to be wrong and the guess is limited to
                            the 99th percentile, including a safety factor

        xyz_ip=CreateInterpolationDiscretization(discretizationlength,xyz,wptIndex)
        BLThicknessGuess=MakeBLGuessUsingNearestIP(xyz_ip,xyz,uvw,reconstruction=True)
```

Listing 1: Algorithm to improve the initial boundary layer guess.