

Comparison of Triangular, Cell-Centered, and Node-Centered Prismatic Grid methods

Aaron Katz^{*}, Shaun Harris[†], Dalon Work[‡], Oisin Tong[§], and Jon Thorne[¶]

Department of Mechanical and Aerospace Engineering, Utah State University, Logan, UT 84322

This paper compares three separate algorithms for computing simple geometries and simple cases. The cases are then analyzed and validated to compare the error. It is concluded which method yields higher order results and therefore yielding a more accurate result. More validation must be undertaken to back up this conclusion.

I. Introduction

Three separate algorithms for computing a flat plate and a bump case are shown and compared. A triangular grid case with flux correction, a cell-centered grid case, and a node-centered grid case with flux correction are explained and a flat plate case and bump case are analyzed for each grid. The purpose of this study is to analyze the error associated with each of the three algorithms mentioned above for a few simple cases. This abstract will present preliminary visualization of a flat plate case and a bump case. Additionally, the node-centered prismatic grid was also run at transonic speeds with a limiter inclusion on a bump case. The limiter computation and results are shown. The resulting shock is shown and analyzed.

II. Numerical Methods

A. Computational Approaches

This section will outline the basic description of each of the three algorithms mentioned above. We will not go into detail on the computation as previous papers have already done so. We will outline the limiter inclusion computation into the node-centered grid algorithm.

B. Grids

1. Triangular Grid

The triangular grid formulation uses computation discussed elsewhere [1]. A section of a two dimensional flat plate is shown in figure 1a.

2. Cell-Centered Grid

The cell-centered grid uses a prismatic approach to define the grid. Figure 1b shows a section of a flat plate outlined with identification numbers at the cell center points. All numerics are computed using this grid identified at the center of each cell as referenced previously [2].

3. Node-Centered Grid

The node-centered grid uses a prismatic approach to define the grid. Figure 1c shows a section of a flat plate outlined with identification numbers at the node center points. All numerics are computed using this grid identified at the center of each node as referenced previously [2].

^{*} Assistant Professor, AIAA Member

[†] Undergraduate Student, AIAA Student Member

[‡] PhD Candidate, AIAA Student Member

[§] PHD Candidate, AIAA Student Member

[¶] Masters Student, AIAA Student Member

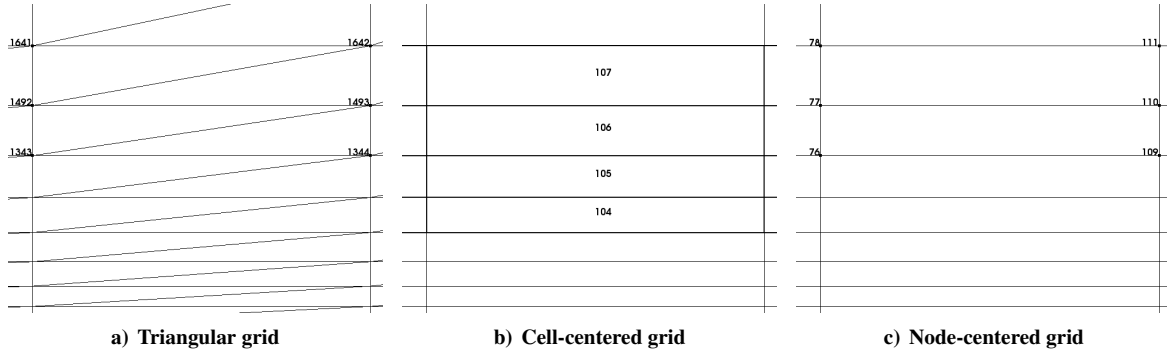


Figure 1. Visual overview of the three separate grids used in this study. Each figure displays the identification number associated with the respective node or cell used in the calculations.

C. Limiter Inclusion

Capturing a shock wave using a higher order method can be difficult. The node-centered prismatic code mentioned previously [2] employs a local extremum diminishing (LED) scheme. A one dimension example of this method is shown in figure C. With this scheme we can employ a limiter. One method used to retain high order accuracy is through limited linear reconstruction, in which the left and right states for the artificial diffusion are reconstructed from surrounding nodal values to the control volume interface. For example, at the $j + \frac{1}{2}$ interface yields

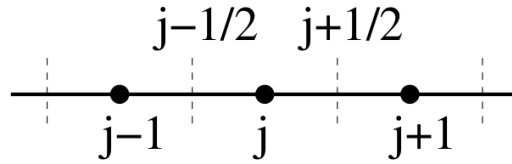


Figure 2. One-dimensional finite-volume stencil

$$Q_L = Q_j + \frac{s}{2}(x_{j+1} - x_j) \frac{dQ}{dx} \Big|_j, Q_R = Q_{j+1} - \frac{s}{2}(x_{j+1} - x_j) \frac{dQ}{dx} \Big|_{j+1} \quad (1)$$

where s is a limiter and the gradients of Q are estimates using surrounding nodal values. Using the reconstructed values actually renders the scheme non-LED. However, we really only need the LED property at local extrema. The role of the limiter, s , is to detect these local extrema, at which points we set $s = 0$. Otherwise, $s \approx 1$. This may be accomplished at $j + \frac{1}{2}$ by setting

$$s = 1 - \left| \frac{a - b}{|a| + |b|} \right|^q, a = Q_{j+2} - Q_{j+1}, b = Q_j - Q_{j+1} \quad (2)$$

where q is a positive integer taken to be three in these node-centered code cases. If a and b are of opposite signs, which would be the case at a local extremum, then $s = 0$. In smooth regions, the limiter is nearly unity.

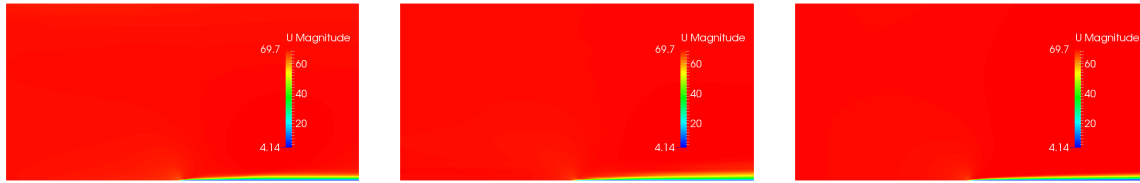
In theory, this algorithm can be employed to result in a lower order approximation at the local extrema to capture a shock wave using a lower order approximation. The rest of the domain will be approximated using a higher order scheme.

III. Preliminary Results

We will compare the results from the flat plate and bump cases from each of the grid codes.

A. Flat Plate

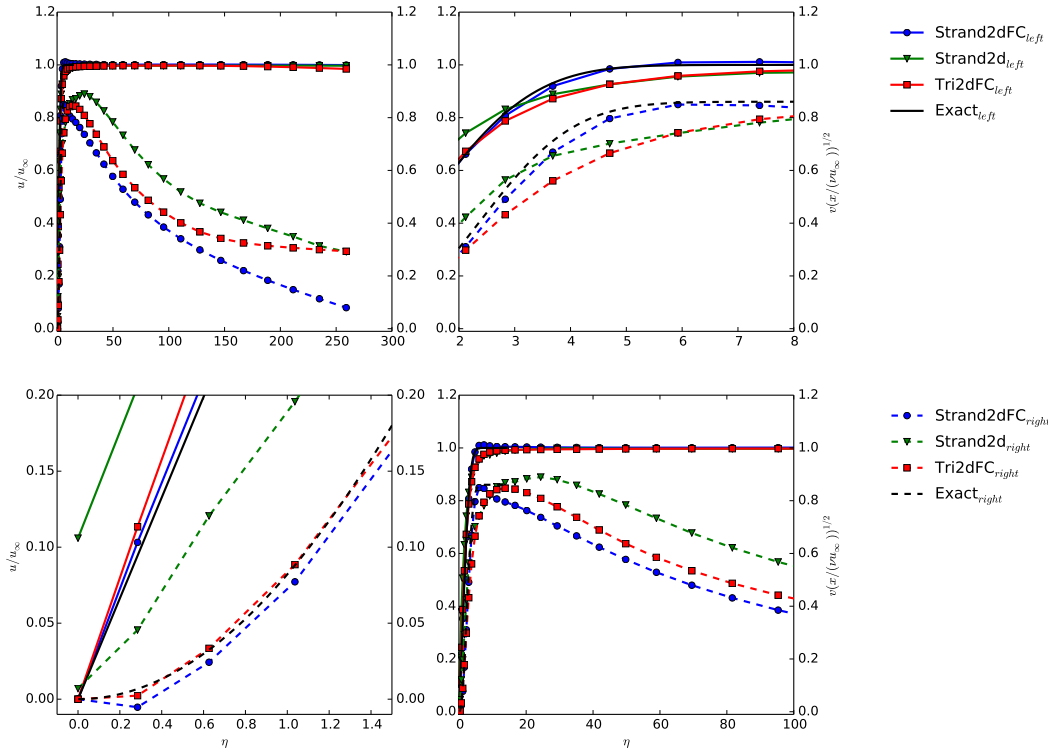
Each grid ran a case of a flat plate at Mach 0.2 and $Re = 10,000$. The visual comparison of each of the three codes run is shown in figure 3.



a) Triangular grid

b) Cell-centered grid

c) Node-centered grid



d) Blasius Solution comparison of all three grids. The left axis shows the u velocity direction and the right axis shows the v velocity direction.

Figure 3. Each figure displays the flat plate case executed by each of the three algorithms associated with the respective node or cell used in the calculations. The comparison of these solutions vs the Blasius solution is also shown in four plots.

Each grid algorithm was run using a similar plate comparison of the flat plate stuff against Blasius solutions. The plot is shown in four separate views below in figure 3d.

We can observe by visual inspection that the Strand2dFC line fits closer to the exact Blasius solution. The Strand2dFC stands for a two-dimensional prismatic code with a flux correction. This is also known as the previously stated node-centered prismatic grid code. We also note that there is a significant error associated with the Strand2d code also known as the cell-centered prismatic grid code. This error will need to be examined before the final work is submitted.

Additionally, a study will be conducted comparing these results formally against a validation study. This will give further insight into the performance of these grid codes.

B. Bump

1. Subsonic

The final paper will give the results of and comparison against a validation study of a Mach 0.2 $Re = 3$ million case. The results will be compared and the error will give an indication for conclusive statements of the performance of the codes.

2. Transonic Inviscid with Limiter Inclusion

The node-centered grid algorithm was taken and run at a faster speed of Mach 0.8. The bump here is an inviscid boundary condition. The resulting shock on the bump is shown in figure 4.

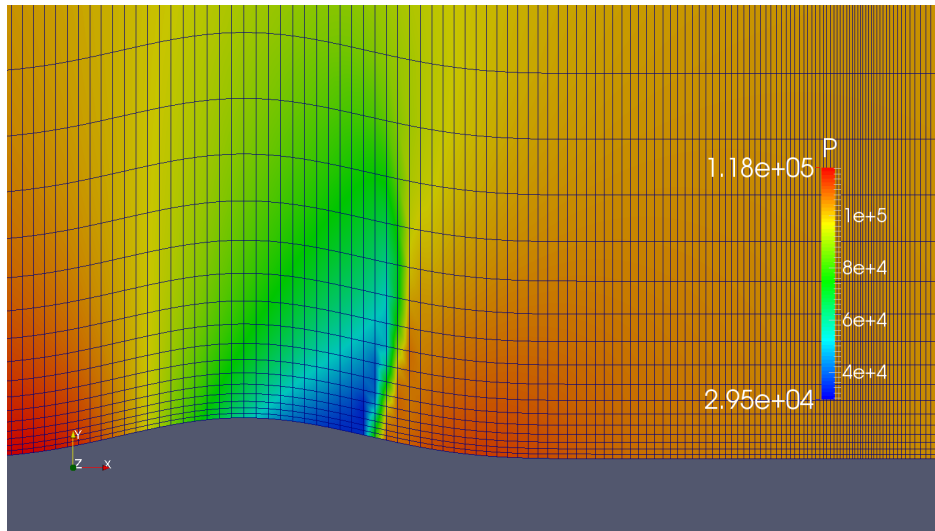


Figure 4. Inviscid transonic bump case at Mach 0.8

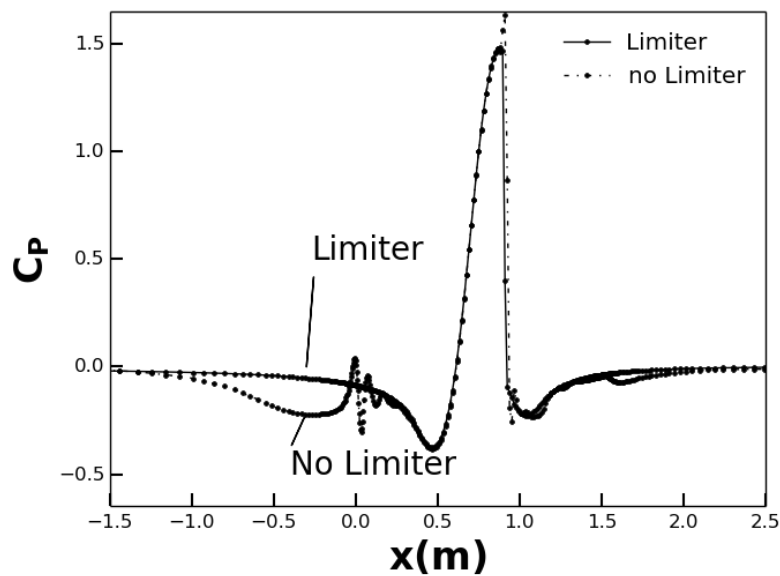


Figure 5. C_p value along the bump at 0.8 Mach speed. This compares the results with and without the limiter inclusion.

IV. Final Paper

The final paper research will conclude the validation error analysis of the three codes. It will conclude analyzing the flat plate case by comparing it against a validation study.

The final paper will include results from the subsonic case which will have an error analysis and validation study. This, along with the flat plate cases, will yield further insight into the higher order solution given by node-centered prismatic grid algorithms.

Additionally, the impact of the limiter will be analyzed. Other limiter calculations will be compared and contrasted. The impact of having high aspect ratios and geometry will also be analyzed with respect to this limiter.

Finally, the final paper will show the value of the node-centered scheme in calculating high-order solutions for simple geometry. This will include other studies and geometries such as a hemisphere cylinder.

V. Conclusions

The preliminary results from the flat plate case has shown the higher-order accuracy of the node-centered prismatic grid algorithm code. Further validation and analysis will theoretically back up this conclusion.

The transonic bump case with the limiter inclusion also shows the advantage of capturing a shockwave while maintaining higher-order accuracy. Further validation and analysis will be applied to this limiter inclusion. The limiter computation may also be subject to change depending on analysis.

With simple geometry and cases, we have preliminary results showing the higher-order accuracy of the node-centered prismatic grid algorithm. Further investigation is still required.

References

- [1] Work, D., *NUMERICAL EXAMINATION OF FLUX CORRECTION FOR SOLVING THE NAVIER-STOKES EQUATIONS ON UNSTRUCTURED MESHES*, Master's thesis, Utah State University, 2014.
- [2] Katz, A. and Work, D., "High-order flux correction/finite difference schemes for strand grids," *Journal of Computational Physics*, 2015.