Turbomachinery in SU2: From Blade Geometry generation to Adjoint Design Optimization

Aman uz zaman Baig,
Dr. Mark Turner

baigan@mail.uc.edu

Gas Turbine Simulation Lab
Aerospace Engineering Dept.
University of Cincinnati

August 9th, 2019

Outline

1. Introduction
2. Pre-processing
3. Results
4. Recommendations
5. Conclusion

Intent of the presentation:

- Explain basic procedure to run Turbomachinery cases in SU2
- Discussion of results of two cases: 3 Blade Row case, IGVs only (distorted inlet NASA profile)
- I will keep things very brief

T-Blade3: 3D blade geometry builder

Quick overview:

- A parametric 3D blade geometry builder
- The geometric and aerodynamic parameters are used to create 2D airfoils which are then stacked on the desired stacking axis
- Available on github.com/GTSL-UC/T-Blade3
- Can output a .geomTurbo file which can be used to create grids in Numeca Autogrid

T-Blade3: 3D blade geometry builder

Quick overview:

- A parametric 3D blade geometry builder
- The geometric and aerodynamic parameters are used to create 2D airfoils which are then stacked on the desired stacking axis

- Available on github.com/GTSL-UC/T-Blade3
- Can output a .geomTurbo file which can be used to create grids in Numeca Autogrid

Grid generation using Autogrid

Quick overview:

- .geomTurbo is an input for Numeca Autogrid that can create a multi-block structured grid
- The solution is then run in Fine/Turbo for a reference solution
- Export a plot3D file for Pointwise

Grid generation using Autogrid

Quick overview:

- .geomTurbo is an input for Numeca Autogrid that can create a multi-block structured grid
- The solution is then run in Fine/Turbo for a reference solution
- Export a plot3D file for Pointwise

Pointwise: Exporting in .su2 format

- The key is to merge the blocks by connecting the overlapping domains (or faces)
- Multi-block grid converts to a single-block grid
- A .su2 format can then be exported
- The same procedure has to be followed for each blade row

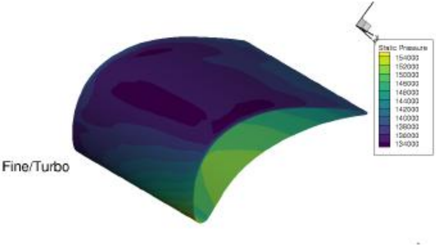
Generating the periodic mesh

- SU2_PER or SU2_MSH for creating periodic meshes
- SU2_PER much faster
- (Issue) The periodicity is only matched with a certain donor, receiver pair, if it is reversed the periodic points cannot be found.
- The same procedure has to be followed for each blade row and combines zones later on

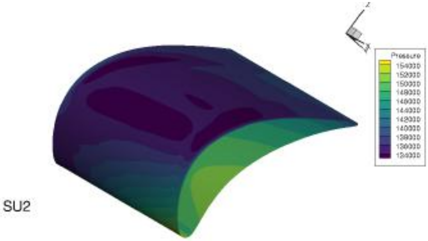
Case overview:

- Standard Aachen Turbine Case
- Constant Inlet/Outlet conditions
- No tip gap, issues in Pointwise
- Venkatakrishnan (slope limiter values used: 0.05, 0.2, 0.9), too many fluctuations, diverged in all cases.
- Van Albada Edge, all fluctuations nearly smoothed out

Pressure distribution over the Rotor

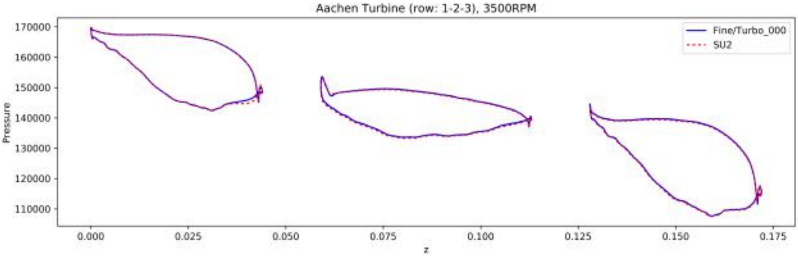


Fine/Turbo



SU2

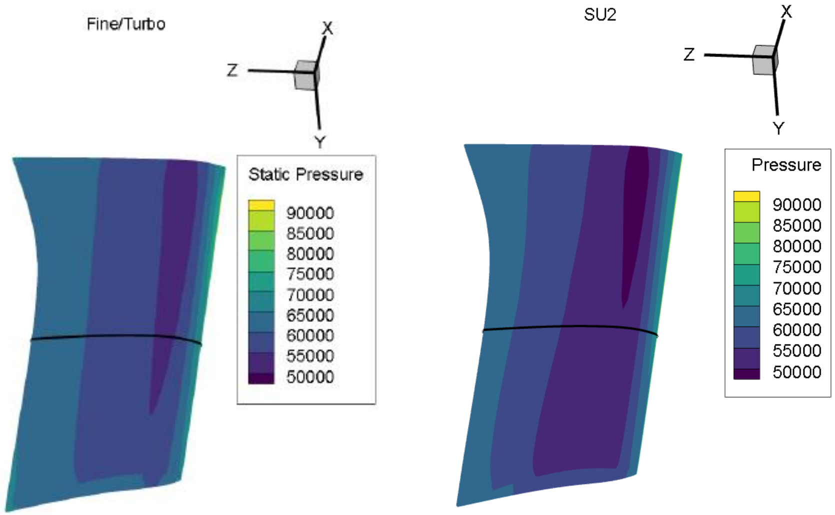
Static Pressure comparison



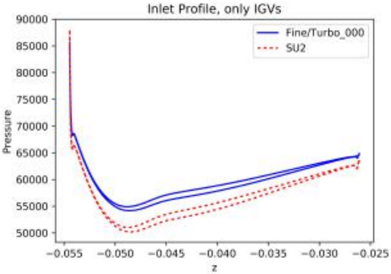
NASA BLI Case:

- Boundary Layer Ingestion Propulsor design with NASA
- Inlet boundary layer distortion profile
- Need an inlet.dat file with coordinates of all grid points, including the halo cells
- Required a customized python program to use periodic_halo files for the 2D and 3D meshes for the inlet face from Pointwise
- Worked with matching point accuracy of 1e-10.

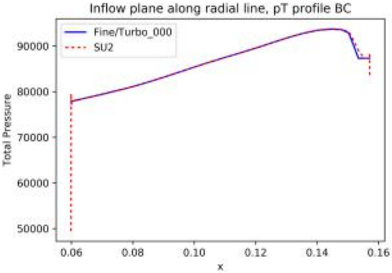
Pressure distribution over IGVs



Distorted profile results



Pressure distribution over the IGV at a spanwise section



BC verification in SU2

Issues and Recommendations:

- No multi-grid for Periodic Boundaries, makes at least 10 times slower than Fine/Turbo
- No interpolation for inlet files yet
- inlet_example.dat does not include periodic points
- Inlet file for multi-zone grids is not found in the working directory
- Separate config files for every zone to initialize differently

Future Work:

- Working on grid differentiation in T-Blade3
- Adjoint Optimization
- Harmonic Balance
- Integrate with structural solver for FSI

Thanks for your attention.

Questions?