# A Reimagining of the DPLR Code

Presented by: Josh Finkbeiner

THERMAL & FLUIDS
ANALYSIS WORKSHOP
Ames Research Center 2025

Dr. Joshua R. Finkbeiner

NASA Glenn Research Center

**Thermal & Fluids Analysis Workshop 2025**
**NASA Ames Research Center**
San Jose, CA
August 4-7, 2025

# Introduction

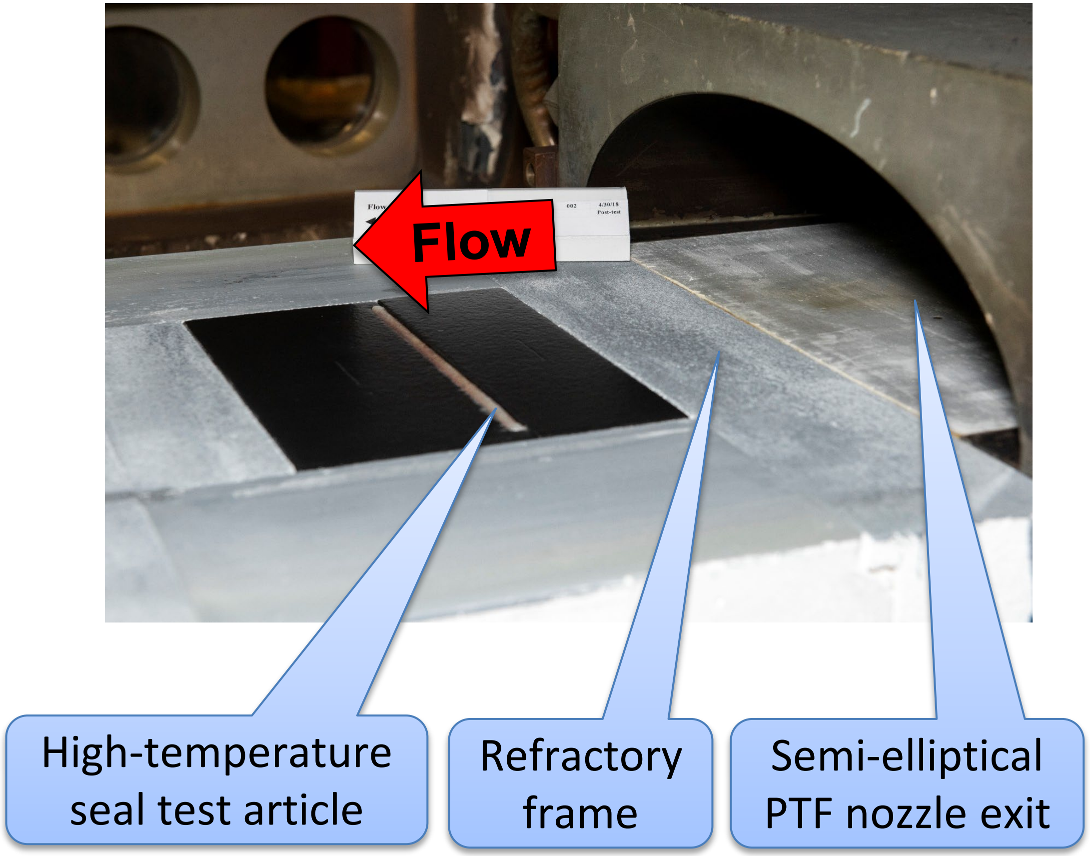
## New implementation of the DPLR code

1. Motivation: Laminar-to-turbulent transition modeling in wall-bounded arc jets
2. Description of DPLR code
3. Challenges encountered while modeling arc jet flows
4. Reimplementation of the code
5. Case study: Corrections to viscous fluxes along domain edges/corners
6. Comparisons of sample case results

# Motivation

- Validate conductive/convective seal thermal models
  - Determine enthalpy profile through Panel Test Facility (PTF) boundary layer
  - Predict surface pressure distribution
- Laminar-to-turbulent transition affects boundary layer profiles
- Gokcen and Alluni (2013) determined that transition occurred within PTF nozzle
- Finkbeiner (2021) identified likely location of transition in PTF nozzle
  - Transition occurs in vicinity of upstream edge of the boundary conditioning plate
  - Became basis for Ph.D. dissertation

**High-temperature seal test article installed in PTF**

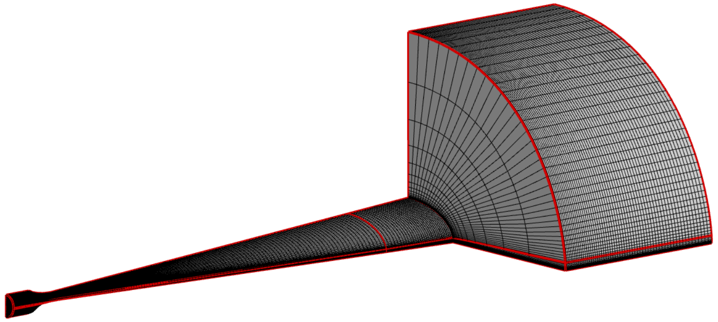


# Description of DPLR

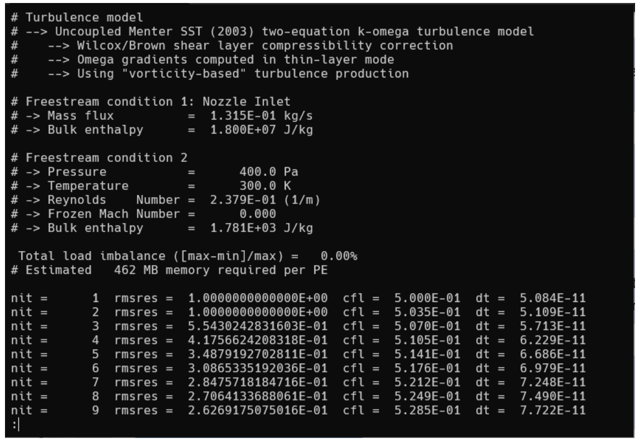
- What is DPLR? See Tang (2016)
  - Ames-developed hypersonic continuum structured CFD code
  - Massively parallel Data Parallel Line Relaxation (DPLR) solver
  - Gas dynamics models for high-speed Earth and planetary entry
  - Laminar or RANS turbulence models
  - NASA Software of the Year (2007)
- Significant heritage in modeling arc jet flows

# DPLR CFD Workflow

**Mesh Generation**
- Model geometry (CAD)
- Discretize domain volume into a mesh
- Check mesh quality



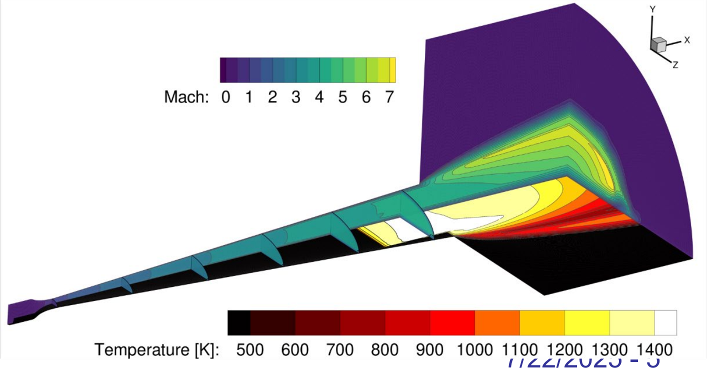
**CFD Preprocessing**
- Decompose mesh for parallel processing
- Detect multi-block interfaces
- Populate initial guess



```
# Turbulence model
# --> Uncoupled Menter SST (2003) two-equation k-omega turbulence model
#    --> Wilcox/Brown shear layer compressibility correction
#    --> Omega gradients computed in thin-layer mode
#    --> Using "vorticity-based" turbulence production

# Freestream condition 1: Nozzle Inlet
# -> Mass flux          =  1.315E-01 kg/s
# -> Bulk enthalpy      =  1.800E+07 J/kg

# Freestream condition 2
# -> Pressure           =      400.0 Pa
# -> Temperature        =      300.0 K
# -> Reynolds    Number =  2.379E-01 (1/m)
# -> Frozen Mach Number =      0.000
# -> Bulk enthalpy      =  1.781E+03 J/kg

 Total load imbalance ([max-min]/max) =   0.00%
# Estimated   462 MB memory required per PE

nit =      1  rmsres =  1.0000000000000E+00  cfl =  5.000E-01  dt =  5.084E-11
nit =      2  rmsres =  1.0000000000000E+00  cfl =  5.035E-01  dt =  5.109E-11
nit =      3  rmsres =  5.5430242831603E-01  cfl =  5.070E-01  dt =  5.713E-11
nit =      4  rmsres =  4.1756624208318E-01  cfl =  5.105E-01  dt =  6.229E-11
nit =      5  rmsres =  3.4879192702811E-01  cfl =  5.141E-01  dt =  6.686E-11
nit =      6  rmsres =  3.0865335192036E-01  cfl =  5.176E-01  dt =  6.979E-11
nit =      7  rmsres =  2.8475718184716E-01  cfl =  5.212E-01  dt =  7.248E-11
nit =      8  rmsres =  2.7064133688061E-01  cfl =  5.249E-01  dt =  7.490E-11
nit =      9  rmsres =  2.6269175075016E-01  cfl =  5.285E-01  dt =  7.722E-11
:
```

**CFD Solution**
- Upload, queue, wait for resources
- Iterate the solution until steady-state
- Download results and check for completion

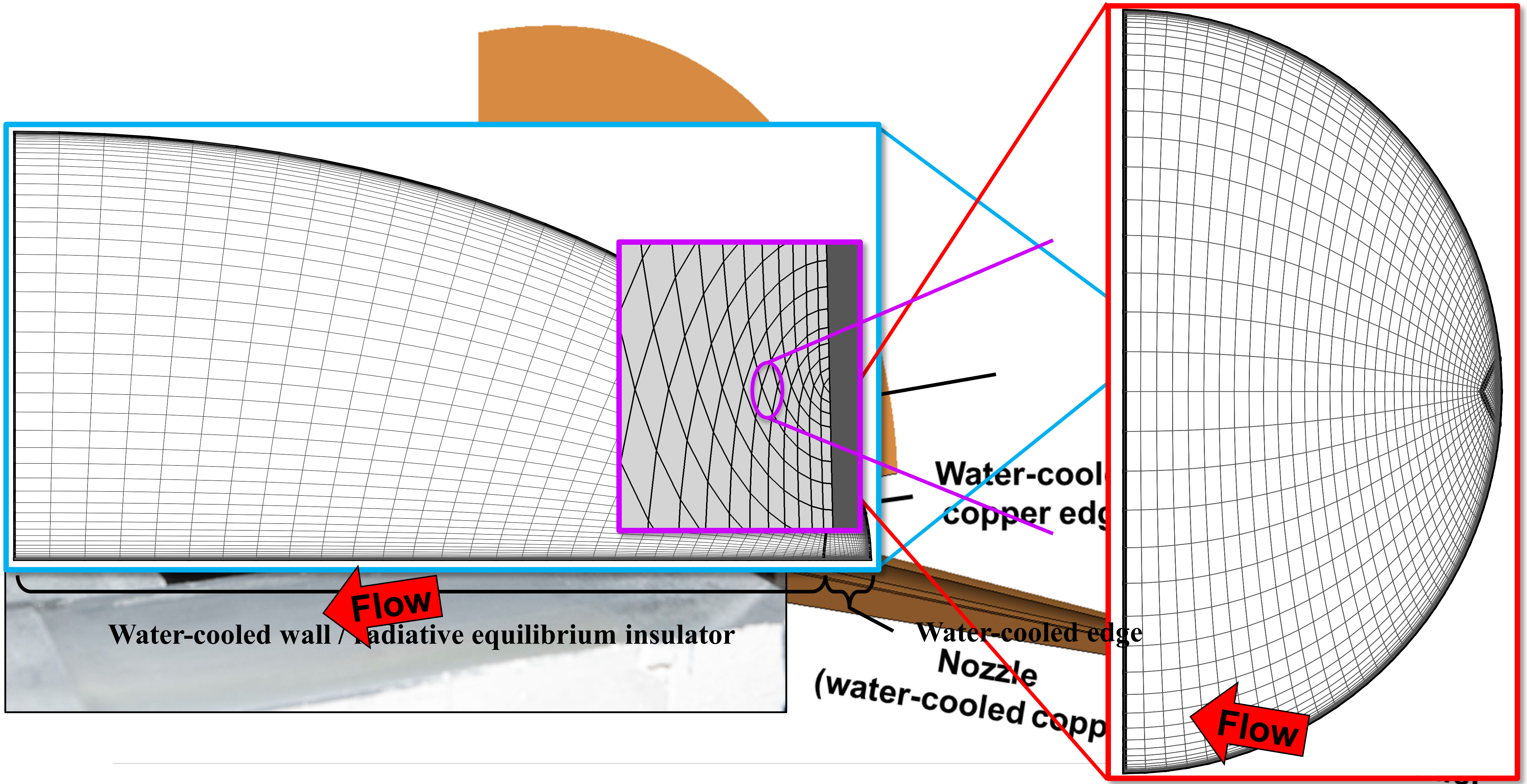
**CFD Postprocessing**
- Check solution “sanity”
- Compute primitive variables
- Generate plots



# Panel Test Facility DPLR Model

- 6-species air chemistry model ($N_2$, $O_2$, NO, N, O, Ar)
- 2-temperature nonequilibrium thermodynamics model
  - Translational/rotational temperature from statistical mechanics
  - Vibrational/electronic temperature from NASA GRC Gordon/McBride tables [McBride 2002]
- Turbulence
  - Laminar
  - Reynolds-Averaged Navier-Stokes (RANS) transition models
    - SST-$\gamma$-$Re_\theta$ (Menter SST) [Langtry 2009]
    - Imposed location (Menter SST, Baldwin-Lomax)
  - $y^+ < 1$ required throughout the nozzle

# Panel Test Facility CFD Domain

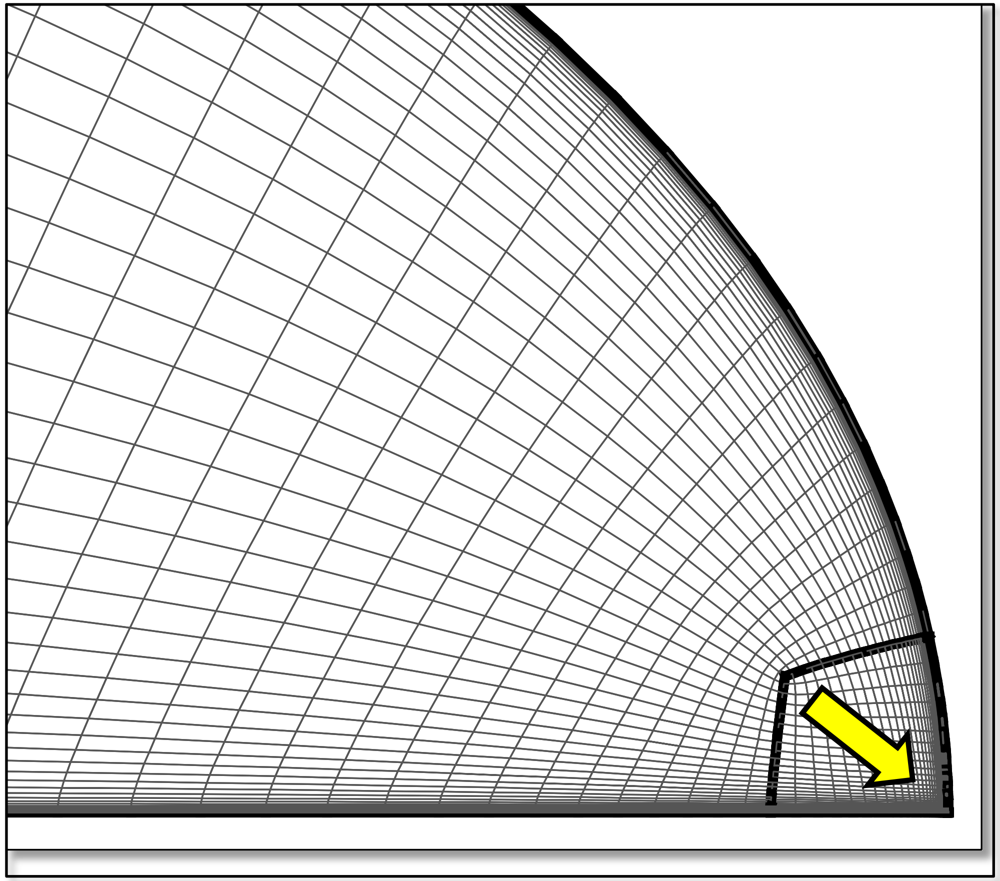





Note: Slide is animated

# Instabilities Developed During Mesh Refinement

- Wall spacing iteratively refined to $y^+ < 1$
- Mesh design and refinement performed by automated scripting
  - Wall spacing
  - Mesh stretch ratios
  - Number of points in each block dimension
  - Elliptic smoothing
- Mesh refinement moved OH block corner toward nozzle walls
  - High viscous gradients along block edges
  - Undefined values developed in tangent viscous flux computations
  - **Crash!**



# DPLR Reimplementation History

- Started modifying the DPLR code in 2014
  - Identified and corrected numerical instabilities
  - Added transition modeling capability (e.g., SST-$\gamma$-$Re_\theta$)
  - Improved robustness of build system
- DPLR can be difficult to read
  - Officially Fortran 90 but more closely aligned with Fortran 77
  - Data dependencies are difficult to trace
    - Fortran 90 modules used for global variables but not code
    - No **intent** statements
- 2014 federal furlough required DPLR work stoppage…
- …but, an enrolled Ph.D. student could modify the code without violating the Federal Anti-Deficiency Act…

# Reimplementation of DPLR Not Intended as Criticism

"It should be emphasized that the original DPLR software is an excellent analysis tool that has been a cornerstone of modern aerothermal analysis at NASA. The rewrite of this software should not be viewed as a criticism of the original authors of DPLR; rather, it should be seen as a distillation of their original ideas into a more approachable and understandable form. The improved readability and organization of the rewritten code potentially allows the concepts included in the original DPLR code to become even more influential in the CFD community." [Finkbeiner (2025)]

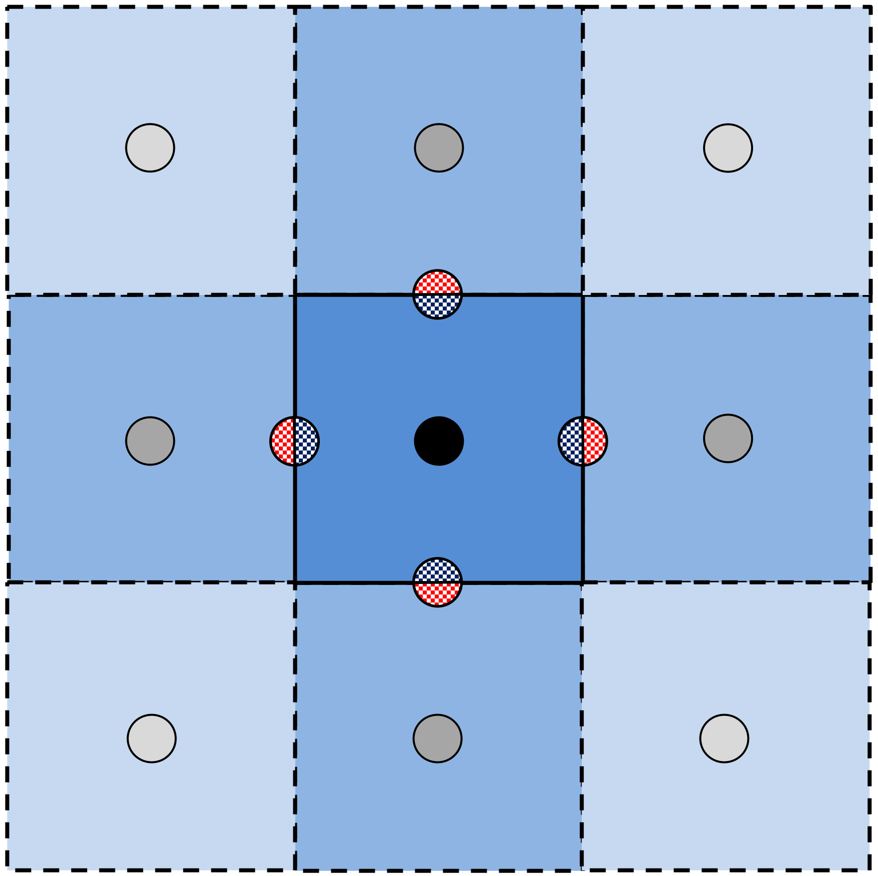
# Summary of DPLR Code Reimplementation

1. Utilized Fortran 2008 conventions (modules, derived types, etc.)
2. Removed duplicate code – “Don’t Repeat Yourself” [Hunt (2000)]
   - 2D and 3D solvers combined into a single universal solver
   - Code organized into modules, Modules organized into libraries, etc.
3. Incorporated standard numerical libraries (e.g., LAPACK)
4. Corrected tangential viscous gradients at edge/corner ghost cells
5. implemented kd-tree to accelerate:
   - Computation of distance to nearest wall
   - Preprocessor interface detection
6. Implemented lagged communications across DPLR line boundaries
7. Reorganized turbulence model subsystem into modular library
8. Accelerated Jacobian computations by storing modal internal energies
9. Corrected MUSCL stencil implementation near viscous walls
10. Added mass flux-based subsonic nozzle inlet boundary condition
11. Added subsonic nozzle initial condition to accelerate solutions

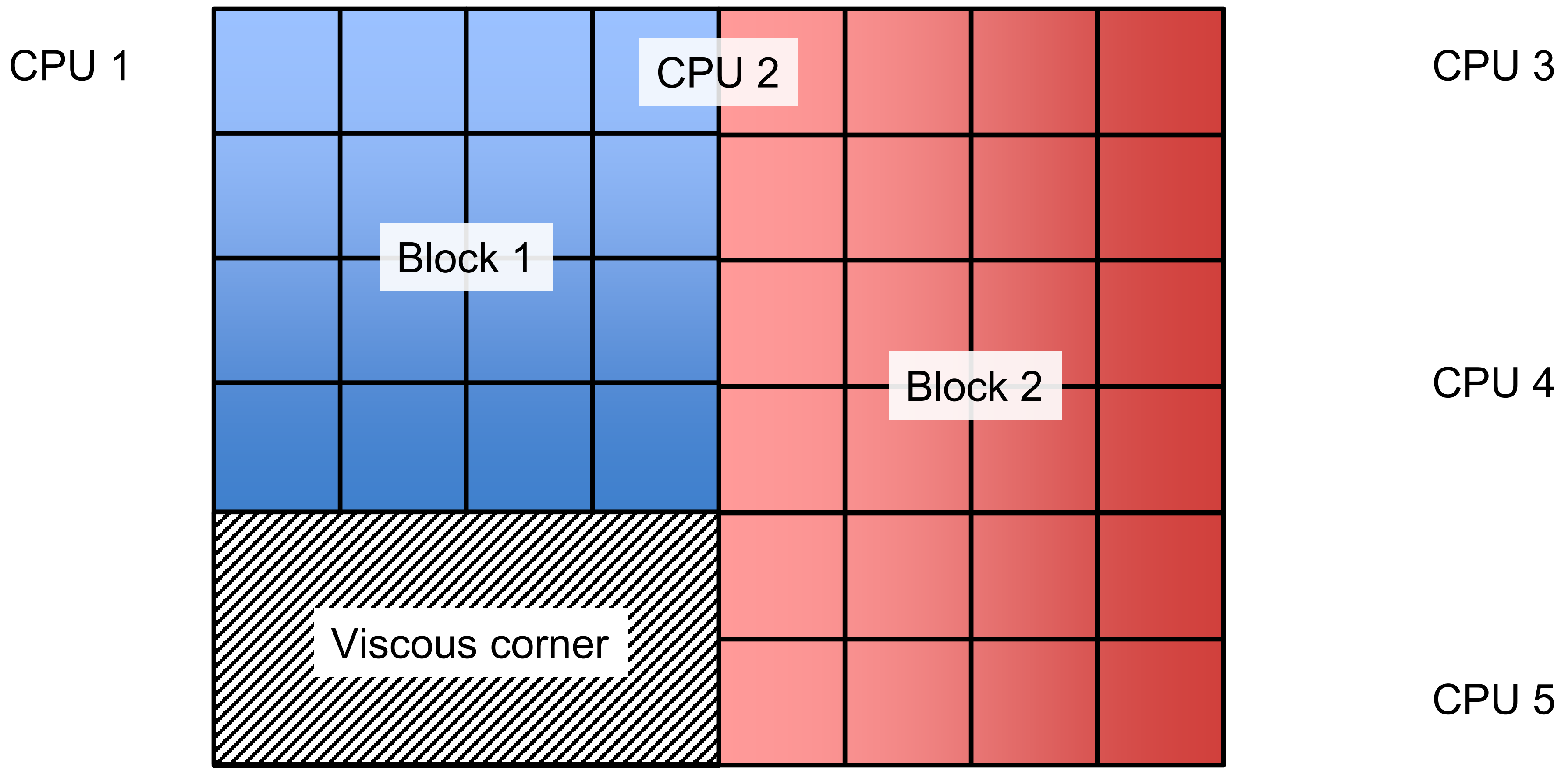
# Viscous Stencil in Corners

# Finite Volume Method

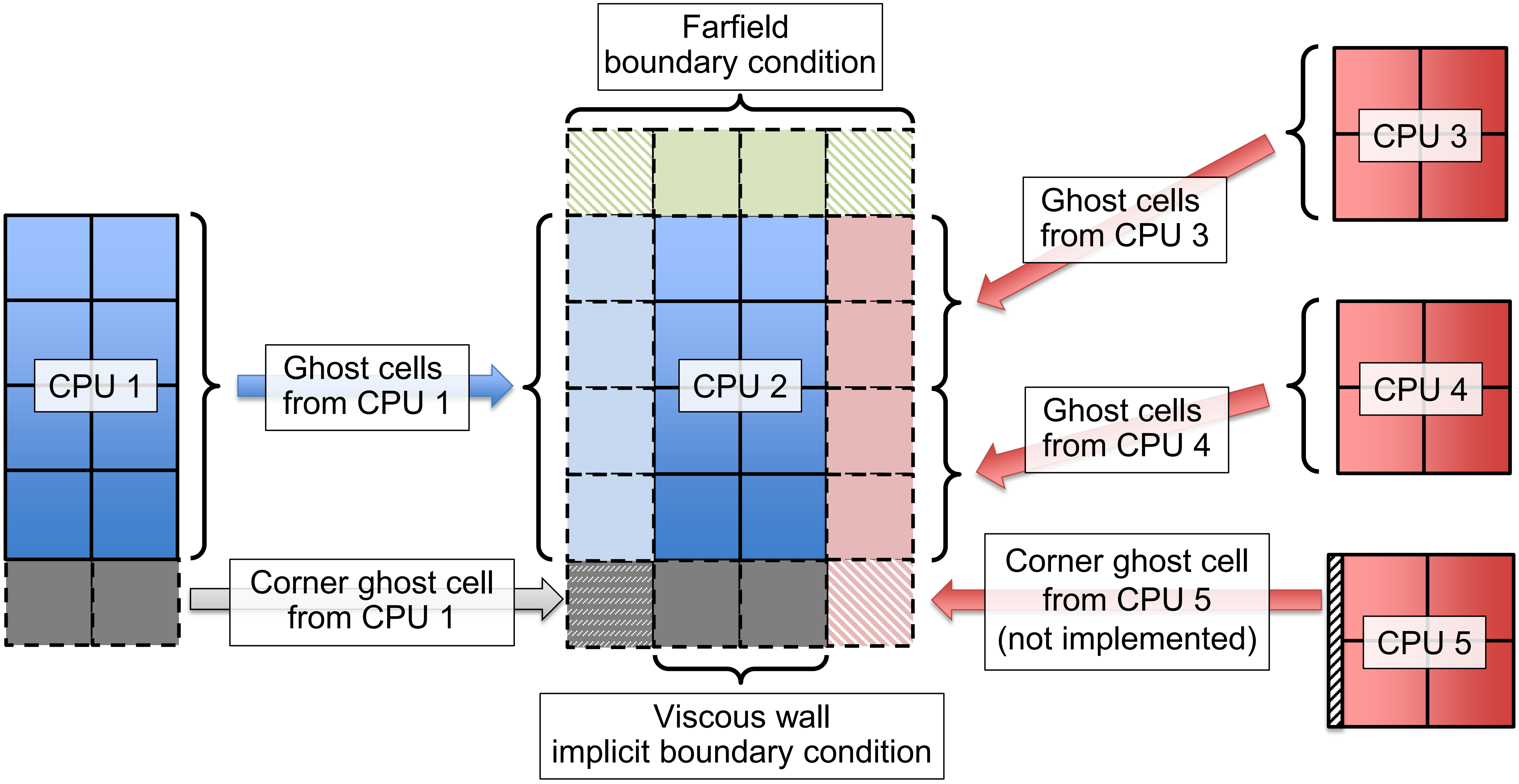
- Solution assumed piecewise constant
  - Flow properties stored at cell centers
  - Solution updates derived from interpolated values at face centers
- Coupled system of algebraic equations
  - Inviscid flux vector computed from left and right face-centered **properties** (vector)
  - Viscous fluxes computed from face-centered property **gradients** (tensor)
- Time integration scheme
  - Explicit: face-centered interpolations derived from current values
  - Implicit: face-centered interpolations derived from future values



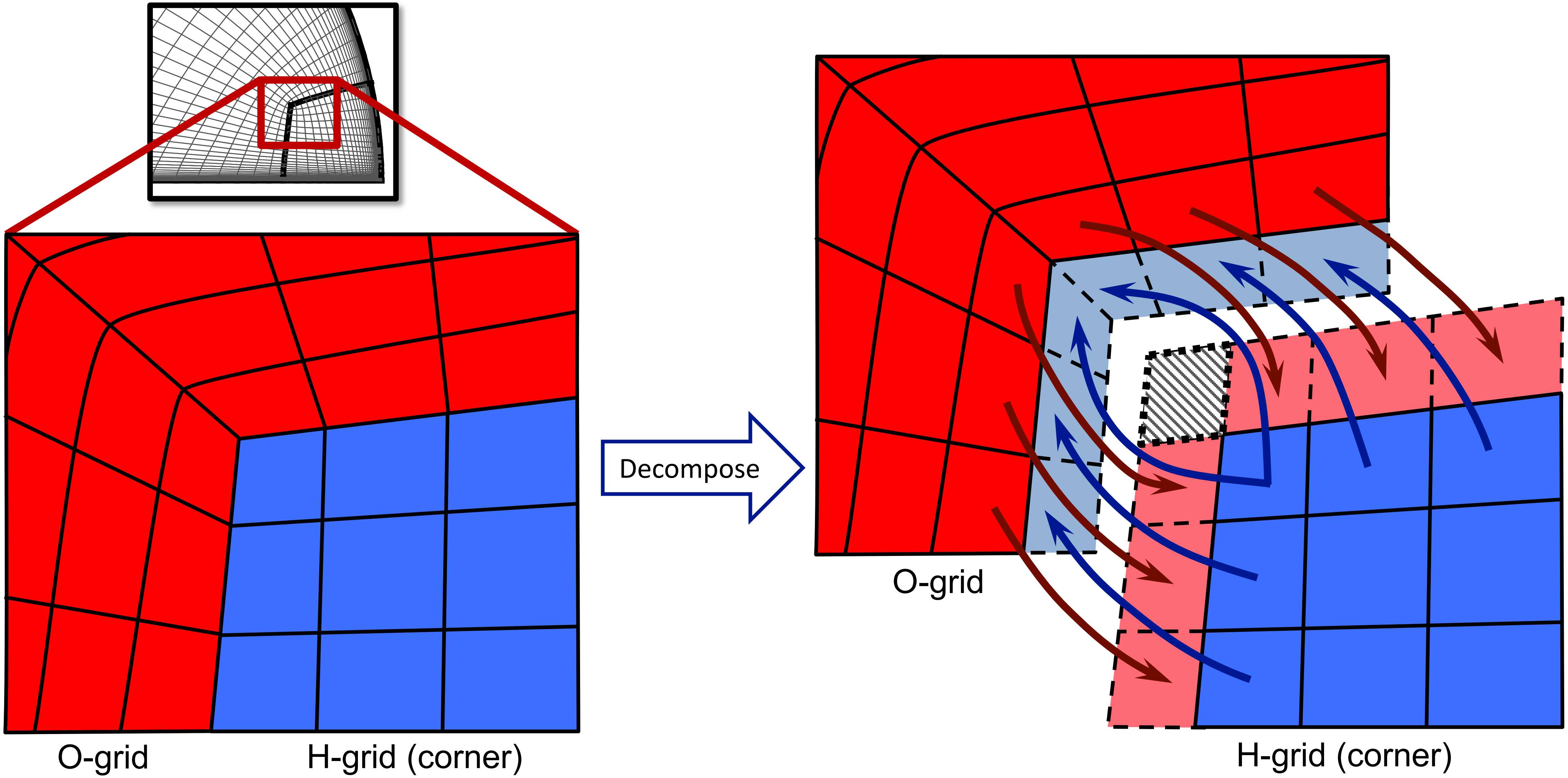
# Multiblock Domain Decomposition



# Ghost Cells and Boundary Conditions



# O-H Mesh MPI Communication



# Viscous Flux Computation in Domain Corners

## Normal gradients

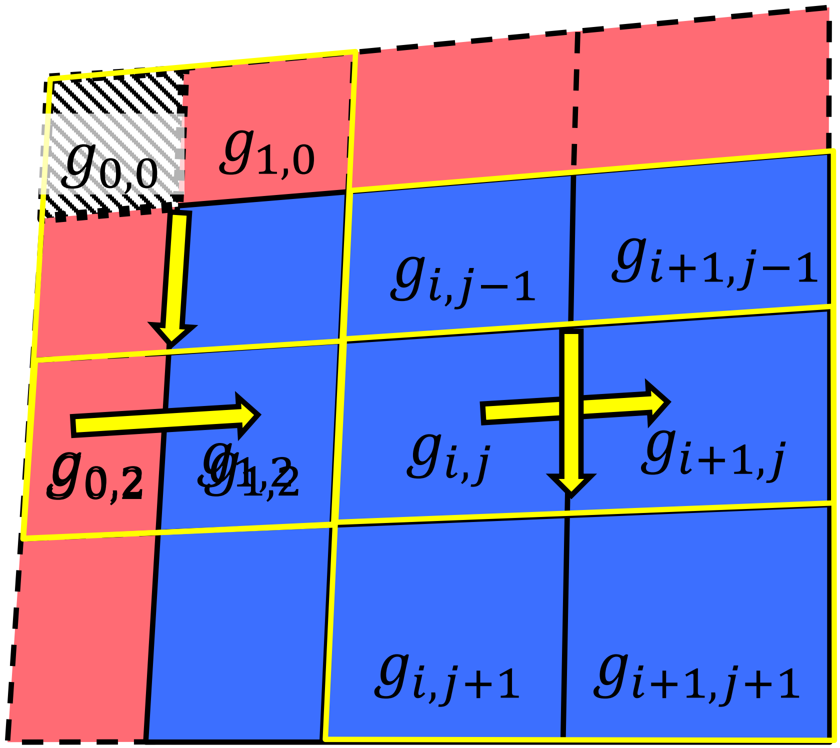
$$\frac{\partial g}{\partial x} = \frac{g_{i+1,j} - g_{i,j}}{\Delta x}$$

- Well-posed throughout domain interior

## Boundary normal gradients

$$\frac{\partial g}{\partial x} = \frac{g_{1,j} - g_{0,j}}{\Delta x}$$

- Well-posed along all boundaries
- Straightforward extension of domain stencil




## Tangent gradients

$$\frac{\partial g}{\partial y} = \frac{(g_{i,j+1} + g_{i+1,j+1}) - (g_{i,j-1} + g_{i+1,j-1})}{4\Delta y}$$

- Well-posed throughout domain interior

## Edge tangent gradients

$$\frac{\partial g}{\partial y} = \frac{(g_{0,2} + g_{1,2}) - (g_{0,0} + g_{1,0})}{4\Delta y}$$

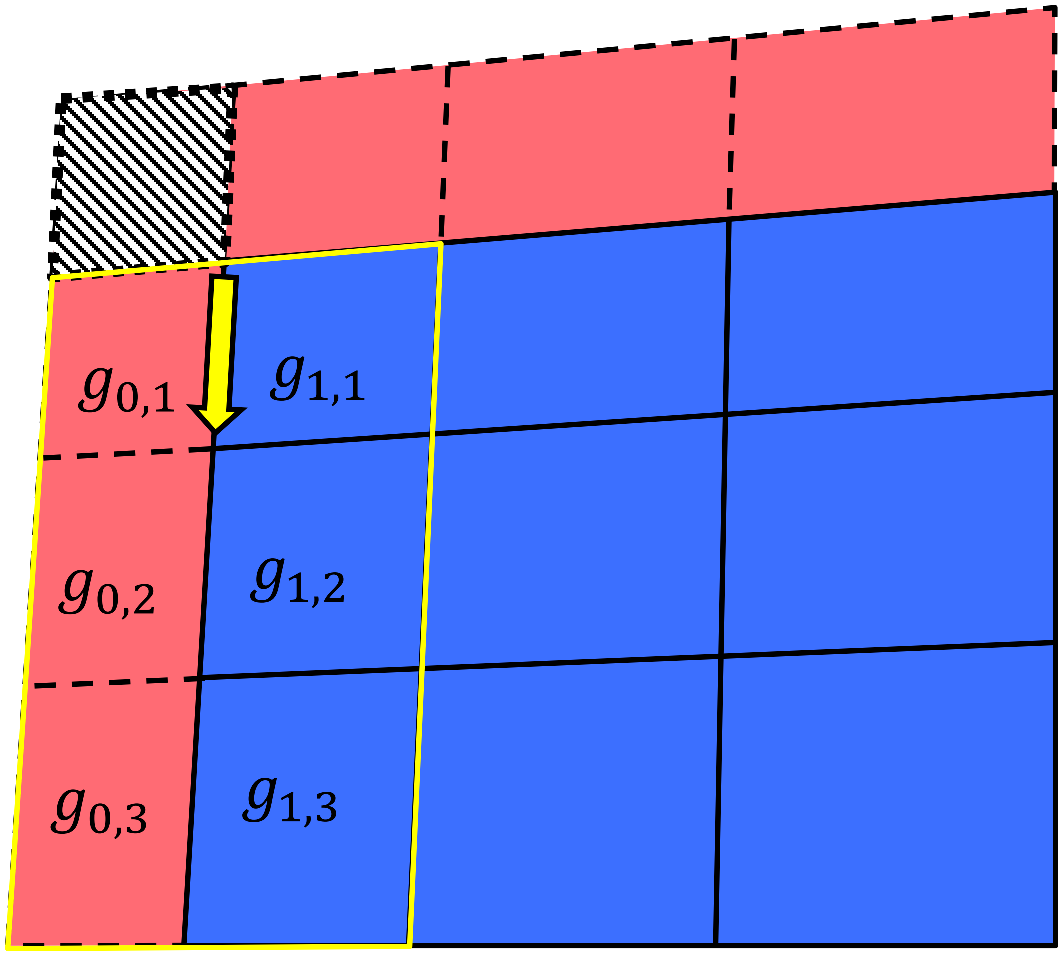
- Cells (0,2) and (1,0) specified by boundary conditions
- Cell (0,0) is undefined
- Caused instability as wall spacing was reduced

# One-Sided Viscous Flux Stencils

- Applied one-sided finite differences

$$\frac{\partial g}{\partial y} = \frac{-3(g_{0,1} + g_{1,1}) + 4(g_{0,2} + g_{1,2}) - (g_{0,3} + g_{1,3})}{4\Delta y}$$

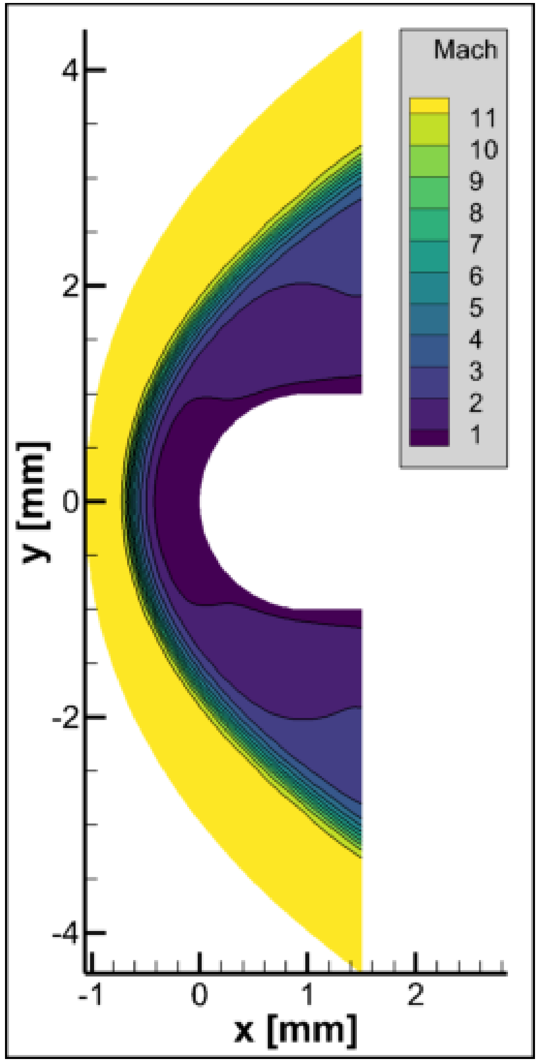
- Benefits
  - Avoids referencing edge ghost cells and maintains numerical stability
  - Retains second-order accuracy
  - Halves number of MPI communications
  - Removes hierarchical corner boundary condition logic



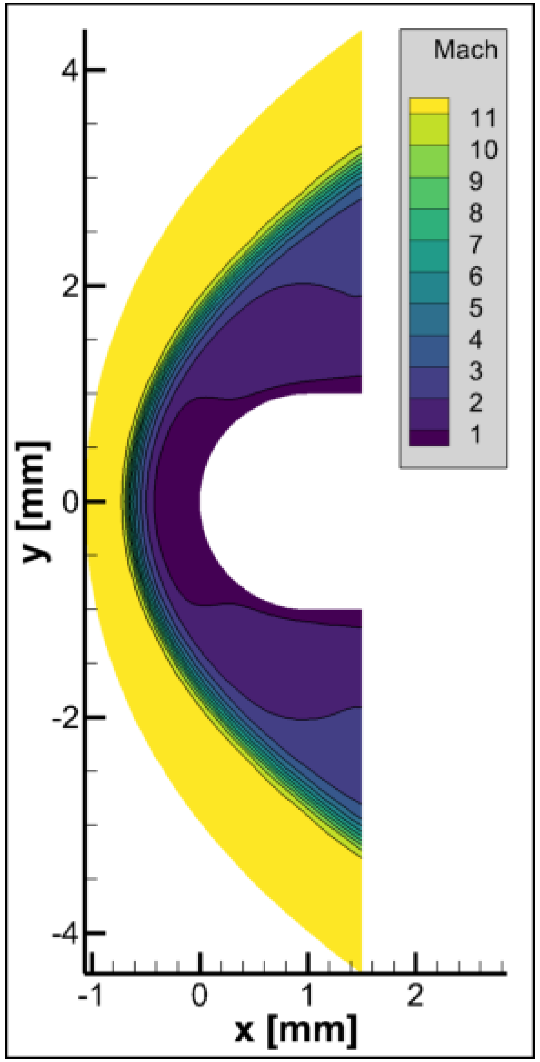
# Comparisons

- DPLR 3.05 included example configuration cases
  - Desktop-scale
  - Mixture of dimensionalities, model specifications, etc.
- 2D 2mm-diameter hemisphere in Mach 11 air
  - 6-species air
  - 2-temperature model (stat. mech)
- Axisymmetric IHF nozzle
  - 8-species air
  - 2-temperature model (stat. mech)

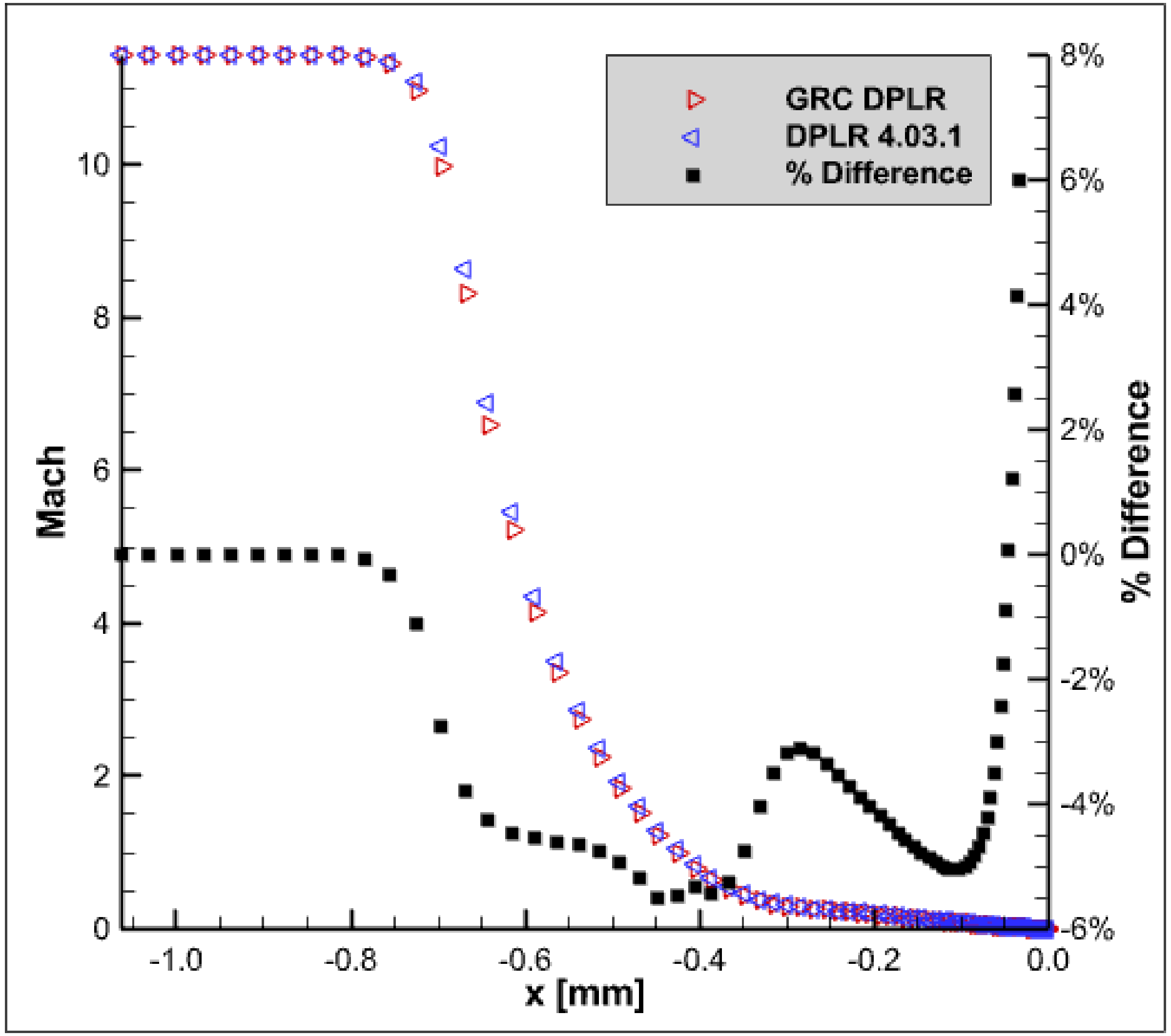
# 2D Cylinder Comparison (DPLR 3.05 example)




DPLR 4.03.1

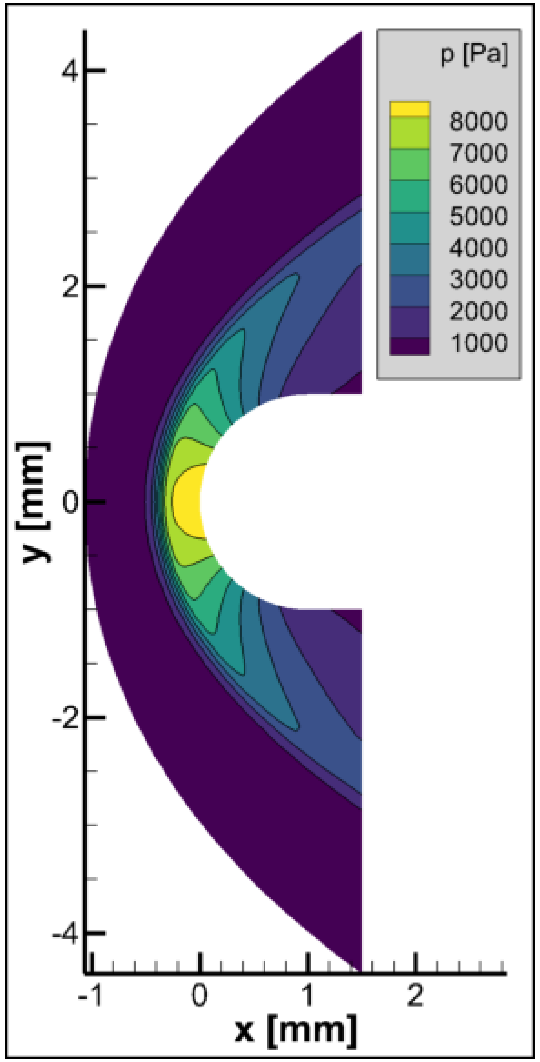



GRC DPLR

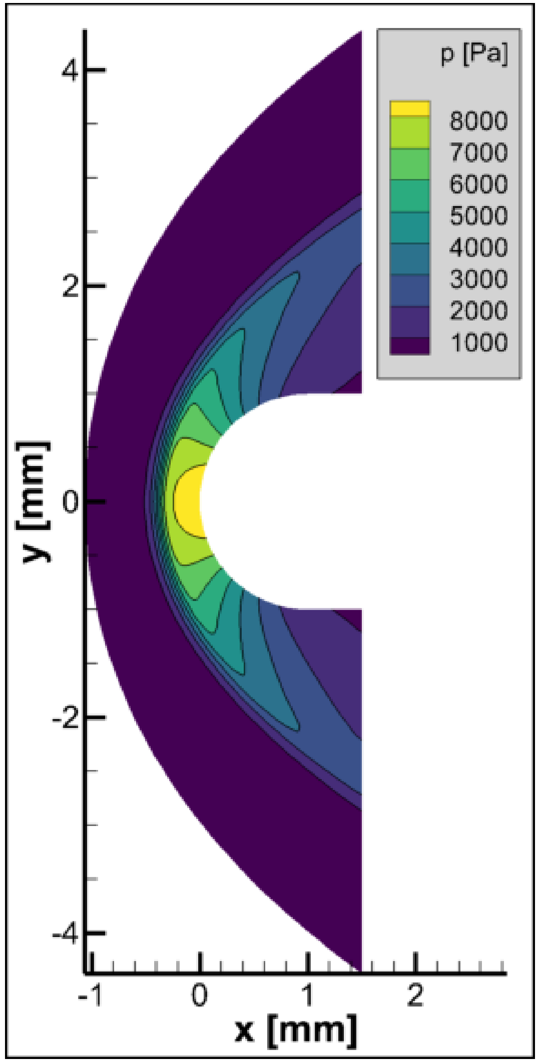



Mach number distributions along $y = 0$

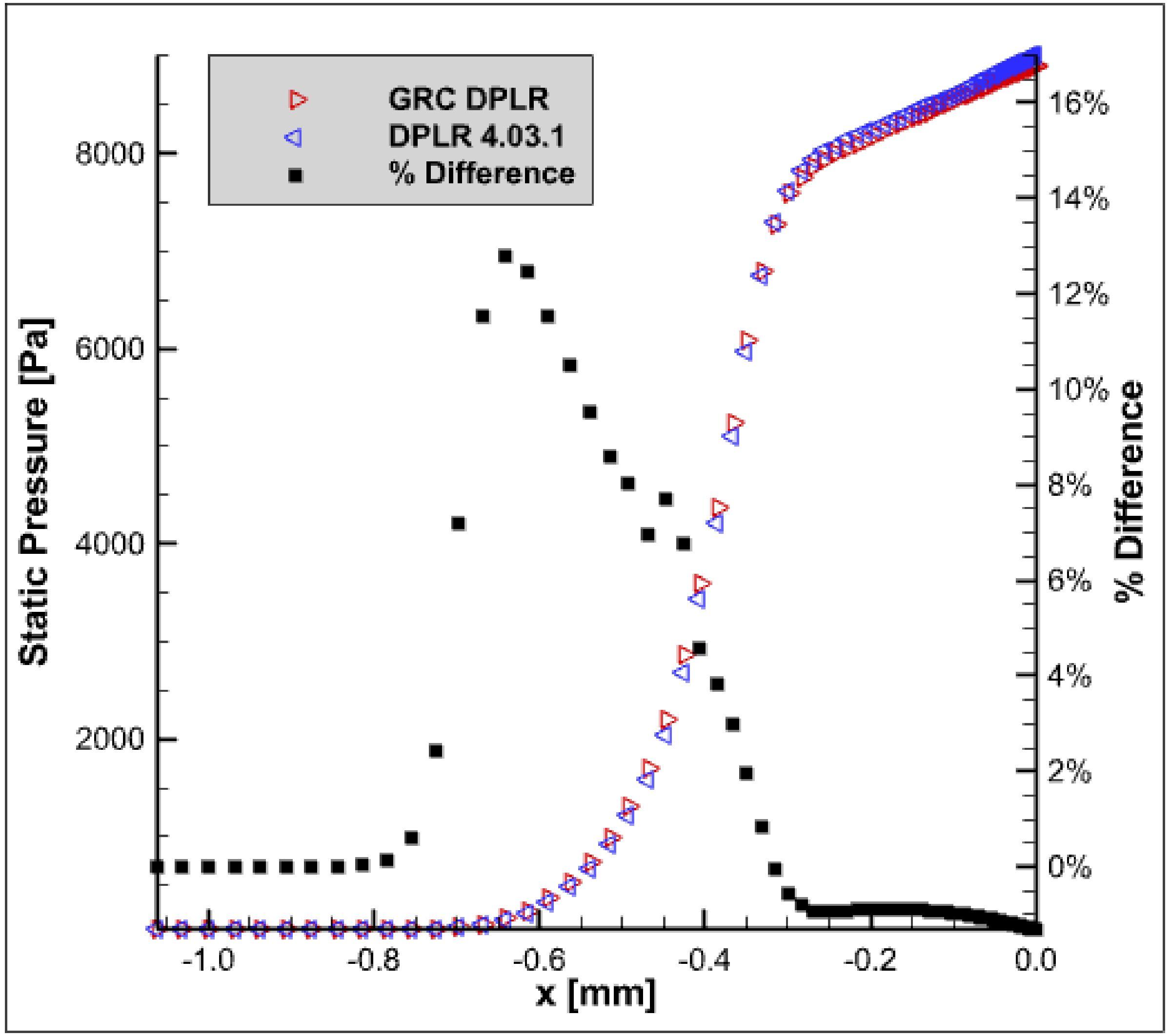
# 2D Cylinder Comparison (DPLR 3.05 example)




DPLR 4.03.1

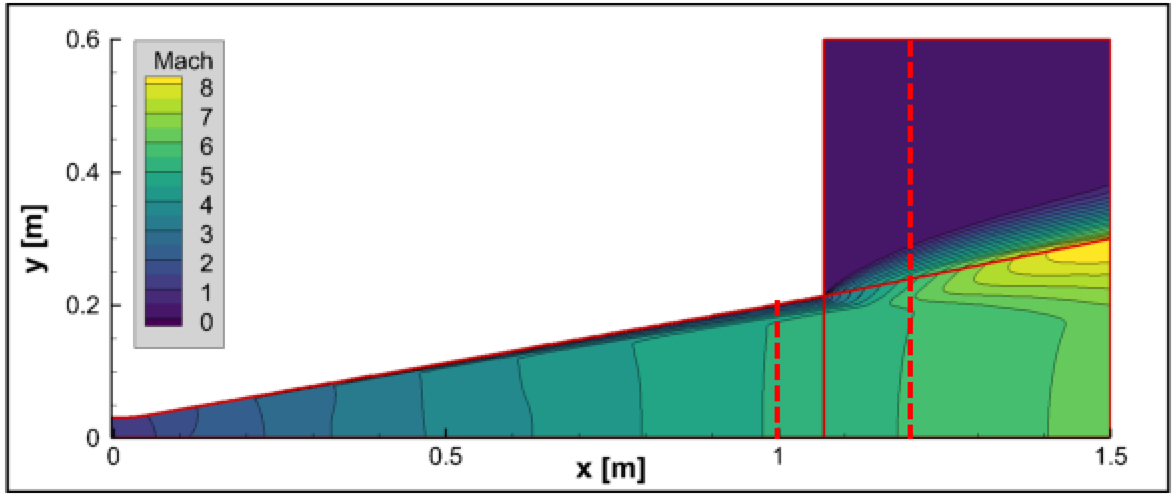



GRC DPLR

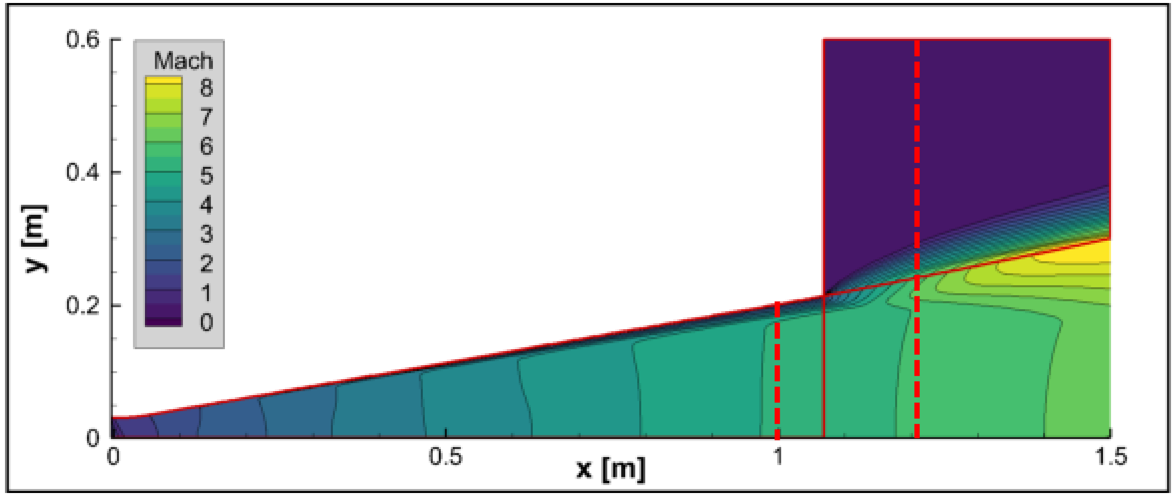



Static pressure distributions along $y = 0$

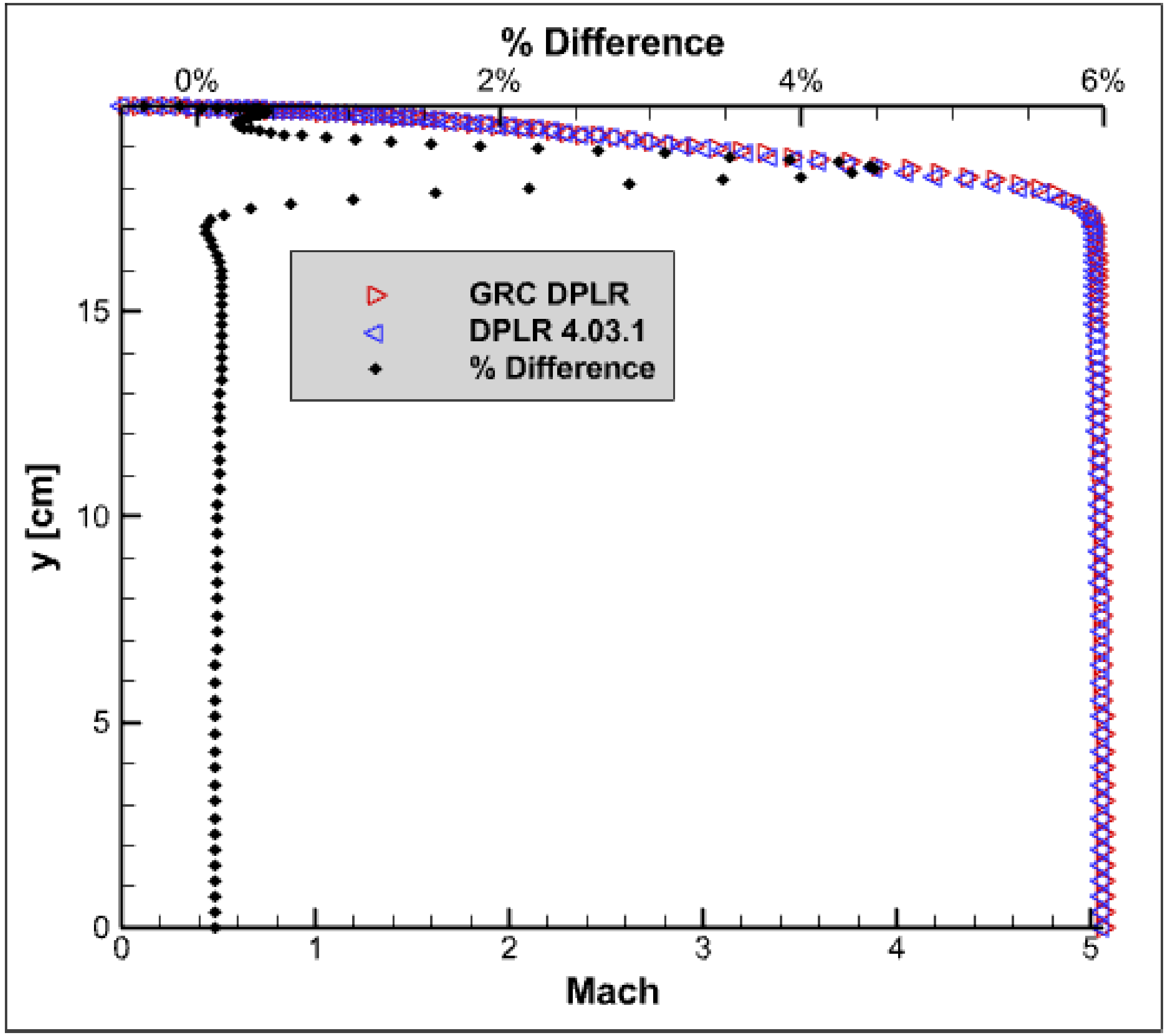
# IHF Nozzle Comparison (DPLR 3.05 example)




DPLR 4.03.1

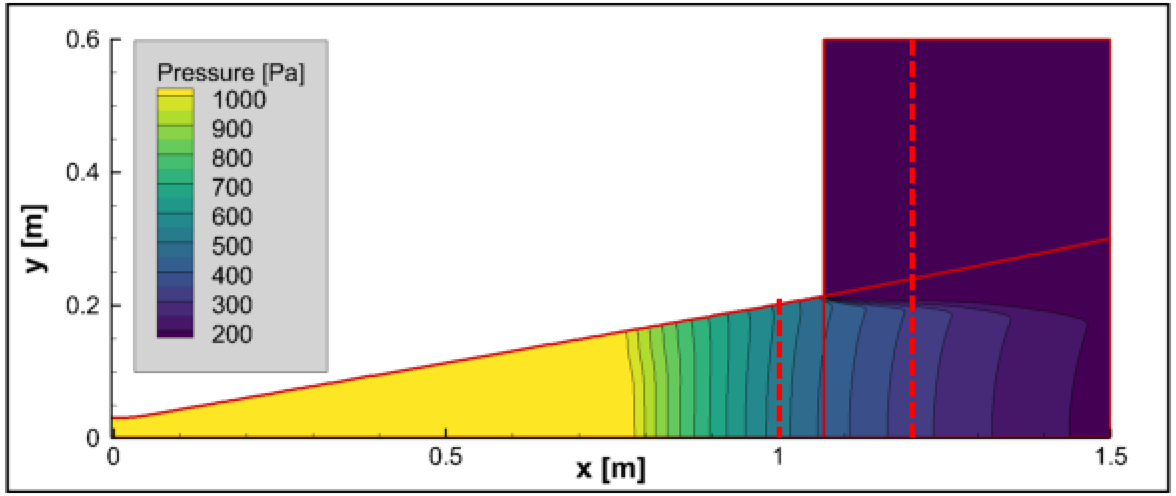



GRC DPLR

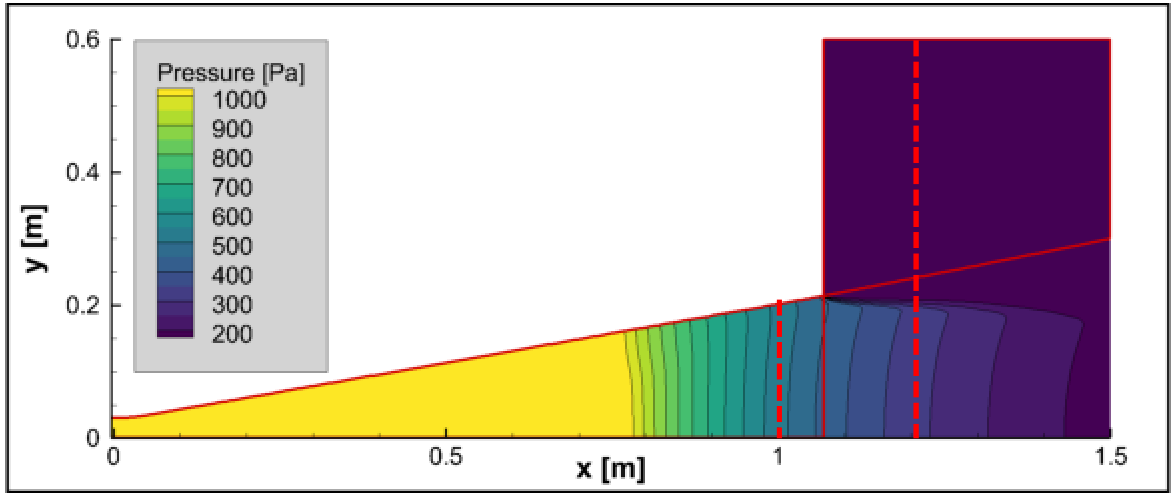



$x = 1.2\,\mathrm{m}$

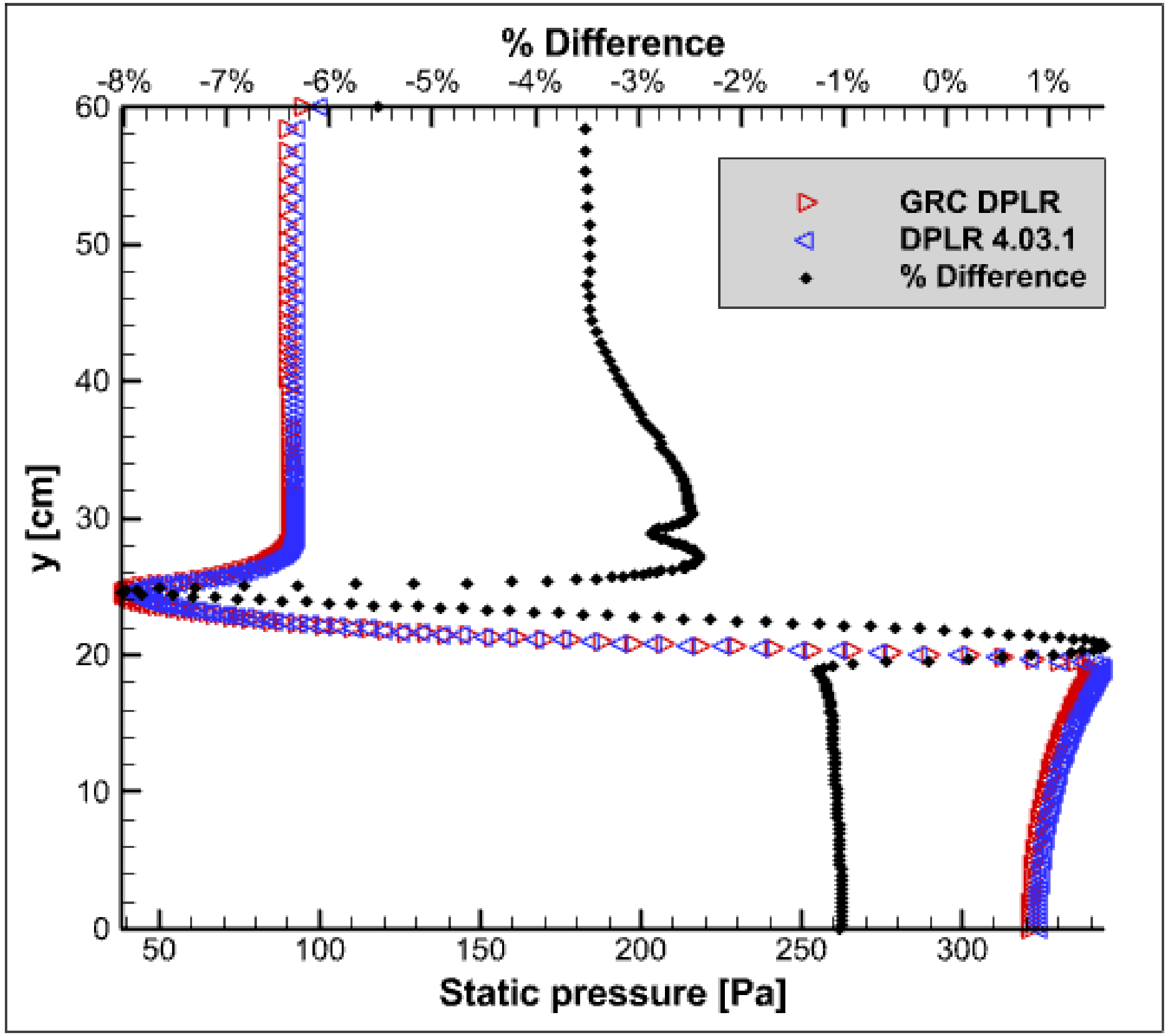
# IHF Nozzle Comparison (DPLR 3.05 example)




**DPLR 4.03.1**

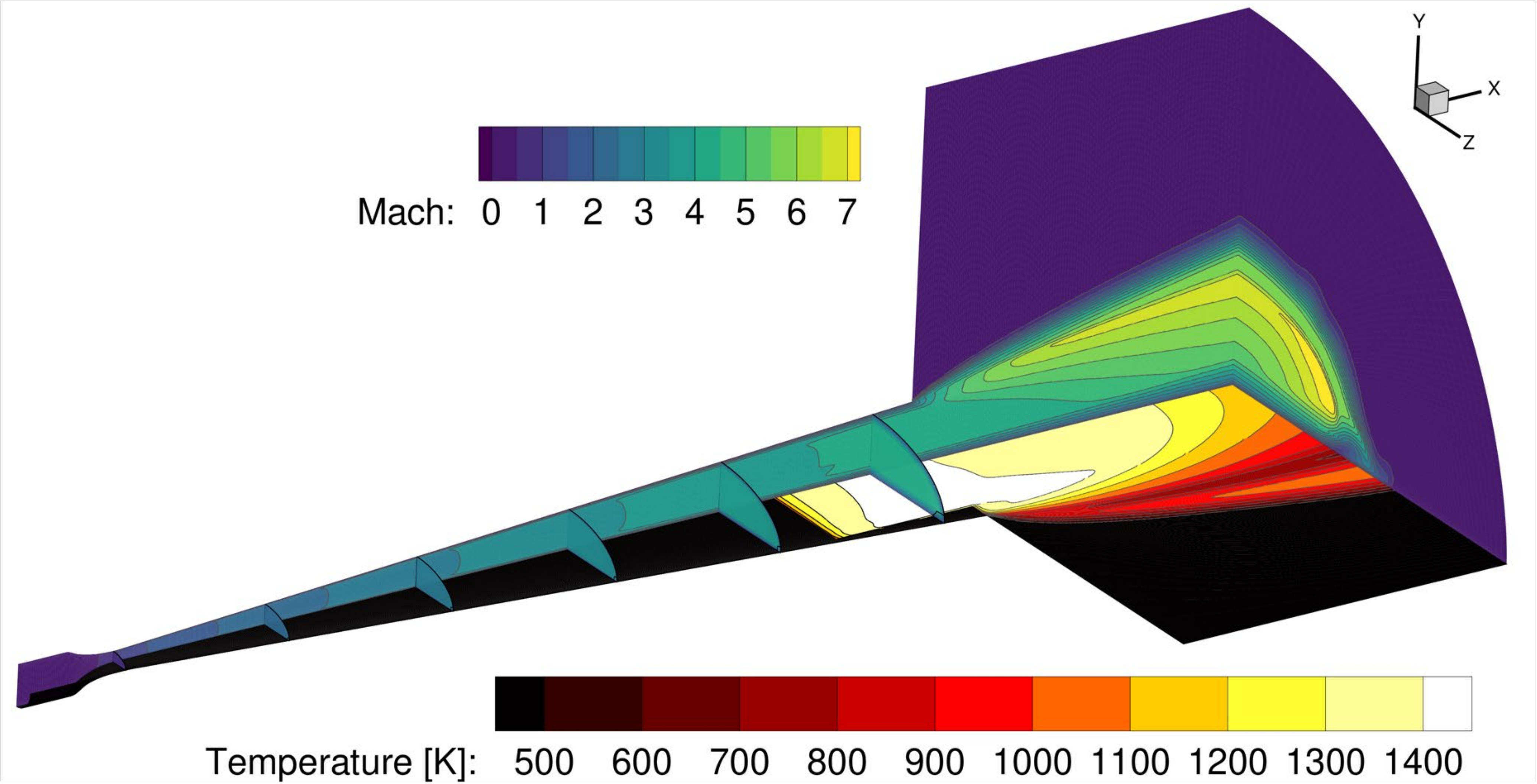



**GRC DPLR**




$x = 1.2$ m

# Success




**Example laminar-to-turbulent DPLR solution of the NASA Ames Panel Test Facility Nozzle**

# Summary

## Reimplemented the DPLR code

1. Utilized modern software conventions
2. Enabled laminar-to-turbulent transition modeling in wall-bounded arc jets with $y^+ < 1$
3. Corrected viscous fluxes along domain edges/corners
4. Implemented numerous corrections and improvements
5. Demonstrated good agreement in sample case results with original Ames DPLR

# References


- Gökçen, Tahir, and Antonella I. Alunni. "On Laminar-to-Turbulent Transition of Arc-Jet Flow in the NASA Panel Test Facility." Journal of Thermophysics and Heat Transfer 27, no. 3 (July 2013): 549–62. https://doi.org/10.2514/1.T3984.
- Finkbeiner, Joshua R. "Modeling Laminar-to-Turbulent Transition in the Panel Test Facility Arcjet." Journal of Thermophysics and Heat Transfer 35, no. 2 (April 2021): 386–401. https://doi.org/10.2514/1.T6098.
- Tang, Chun Y. "Introduction to DPLR." Thermal & Fluids Analysis Workshop. NASA Ames Research Center, 2016. https://ntrs.nasa.gov/citations/20160010056.
- McBride, Bonnie J., Michael J. Zehe, and Sanford Gordon. "NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species." NASA Technical Paper. Cleveland, OH: NASA Glenn Research Center, September 1, 2002. https://ntrs.nasa.gov/citations/20020085330.
- Langtry, Robin B., and Florian R. Menter. "Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes." AIAA Journal 47, no. 12 (December 2009): 2894–2906. https://doi.org/10.2514/1.42362.
- Finkbeiner, Joshua R. "Modeling Laminar-to-Turbulent Transition in the Panel Test Facility Arcjet." Ph.D. Dissertation, Purdue University, 2025.
- Hunt, Andrew, and David Thomas. The Pragmatic Programmer: From Journeyman to Master. USA: Addison-Wesley Longman Publishing Co., Inc., 2000.