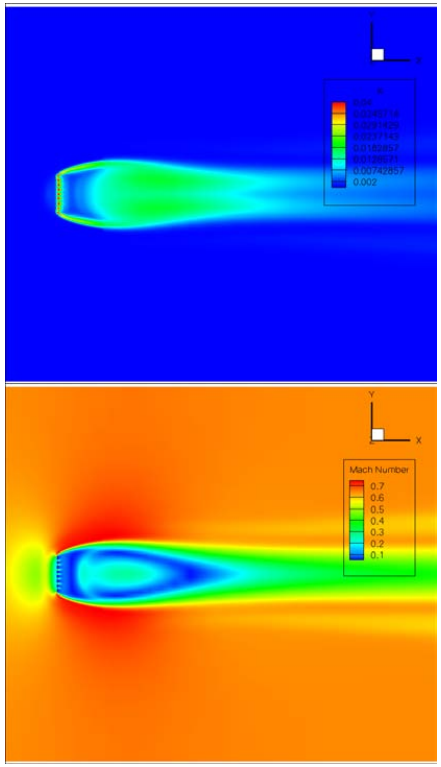
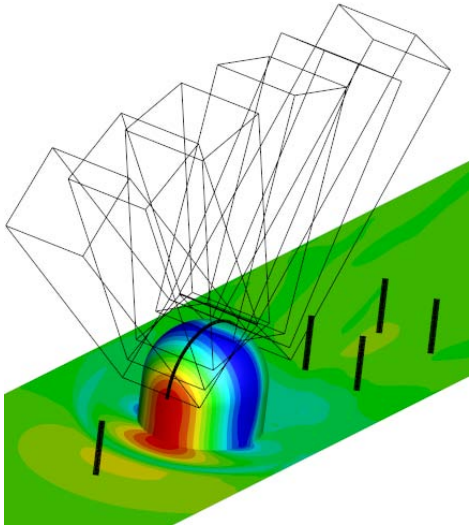
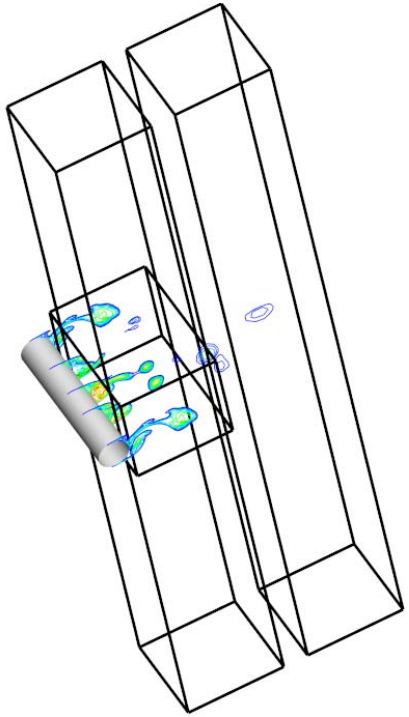
KRATOS
DEFENSE & SECURITY SOLUTIONS

Integration of Aero-Optics and Distributed Porosity Models within OVERFLOW



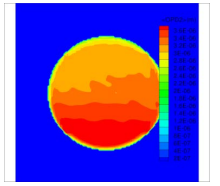




William J. Coirier, Ph.D.
Director, Engineering and Aerosciences Branch
Advanced Technology Division

Presented at the

10th Symposium on Overset Composite Grids and Solution Technology
September 23, 2010



KRATOS
DEFENSE & SECURITY SOLUTIONS

Outline

- "Porous Media" Model
  - Identification of need and approach
  - Implementation in OVERFLOW
  - Comparisons/Demonstrations
- Aero-Optics Model Integration
  - Paraxial Beam Equations and Solver
  - Development and Integration into OVERFLOW: Approach #1
    - Validation and Demonstrations
  - Development and Integration into OVERFLOW (and others): Approach #2
    - Description of approach
    - Public API

KRATOS
DEFENSE & SECURITY SOLUTIONS

Porous Fence Model: Identification of Need

- Customer needs help mitigating effects of open cavities upon instruments located inside aircraft.
- An aerodynamic fence is presently being used on one cavity, while the other cavity might be improved with the addition of a fence there as well.
- Modeling either fence with an overset grid system is possible, but prohibitive:
    - "Fence #1" requires approximately 70 Million grid points for the fence alone for a total model size of about 105 Million grid points
- A simple body force model was applied earlier (Tramel), but needed some improvements:
    - Parallelization, body force same in all directions, flow realignment "fix"

Approach

- Develop a "Distributed Porosity Model" for incorporation into OVERFLOW
- Requirements:
    - Parallelizable
    - "Easy" to specify volumetric and areal porosities and regions

KRATOS
DEFENSE & SECURITY SOLUTIONS

Porous Fence Model

- Distributed Body Forces:
  - Simulate the effect of obstructions not resolved by mesh
  - Motivated by forest and urban canopy modeling used in numerical weather models and in urban CFD transport and dispersion models

$$S_x = \frac{A_x}{\beta}\frac{\rho}{2}u|U| = \Gamma_x \rho u|U| \qquad S_y = \frac{A_y}{\beta}\frac{\rho}{2}v|U| = \Gamma_y \rho v|U| \qquad S_z = \frac{A_z}{\beta}\frac{\rho}{2}w|U| = \Gamma_z \rho w|U| \qquad S_5 = uS_x + vS_y + wS_z$$

- Source terms added to the momentum and energy equations
  - There are terms that can be added to t.k.e. and dissipation rate equations, but these are neglected due to past experience
- Linearization of body force terms is included in two LHS operators:

$$J_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\Gamma_x u U & \Gamma_x \dfrac{\left(U^2 + u^2\right)}{U} & \Gamma_x \dfrac{uv}{U} & \Gamma_x \dfrac{uw}{U} & 0 \\ -\Gamma_y v U & \Gamma_y \dfrac{uv}{U} & \Gamma_y \dfrac{\left(U^2 + v^2\right)}{U} & \Gamma_y \dfrac{wv}{U} & 0 \\ -\Gamma_z w U & \Gamma_z \dfrac{uw}{U} & \Gamma_z \dfrac{vw}{U} & \Gamma_z \dfrac{\left(U^2 + w^2\right)}{U} & 0 \\ \dfrac{\partial S_5}{\partial q_1} & \dfrac{\partial S_5}{\partial q_2} & \dfrac{\partial S_5}{\partial q_3} & \dfrac{\partial S_5}{\partial q_4} & 0 \end{pmatrix}$$

Implementation Details

- Definition of porous regions and porosity values:
  - (j,k,l) range of regions and "type" of porosity definitions:
    - Ax, Ay, Az constant
    - Ax, Ay, Az define magnitude of areal porosity and are applied normal to the local coordinate surfaces using metrics (3 surfaces)
    - Ax, Ay, Az defined from a q-file
  - In practice, we use the q-file approach:
    - Given the fence geometry (hole pattern, hole diameters) and grid, each grid point is assigned the corresponding areal and volumetric porosity

$$n_i = \frac{\xi_i}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}$$

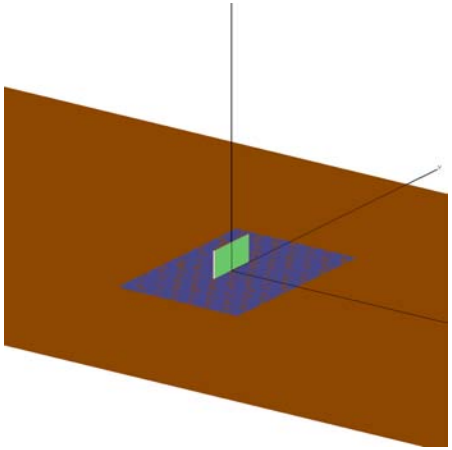
- Parallelization
  - After domain decomposition in OVERFLOW, on each processor each porous media "block" is intersected with the blocks/zones being solved on this processor resulting in a list of intersected porous media sub-blocks used for body force processing on this processor
- OVERFLOW entry points:
  - OVERRU, OVERST, STEP, RHSH, OVERGL, DISTGL, OVERDO
- Limitations:
  - Not set up for grid coarsening (FMG=.true.) or multi-grid or moving grids or …
  - Linearization only for 32 bit SSOR and BW (ILHS=0 or 5, in the j-factor split)

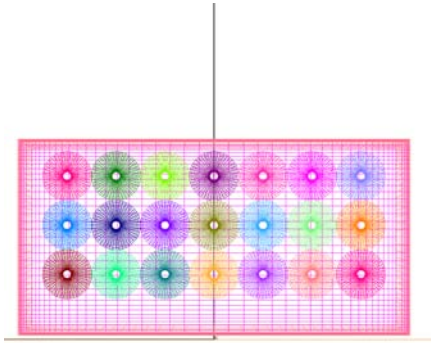
KRATOS
DEFENSE & SECURITY SOLUTIONS

Porous Fence Model "Validation"

- We have been using the porous fence model to perform design trade studies for a particular aircraft in a particular program funded by a particular agency.
  - DES calculations of a complete aircraft with open instrument bays, simplistic models of instruments inside
  - Estimate improvements upon unsteady forces and moments imposed on instrument models with different fence designs
    - Fence height, inclination angle, location upstream, porosity
  - Flight tests have backed up our design trends and increments
  - Our recommendations are flying now

What we can show you:
- Spoiler on a plate
- Spoiler with 21 holes on a plate
- Compare overset results (resolving plate and holes) with porous media model results
- RANS (SST), HLLC, 2nd Order…

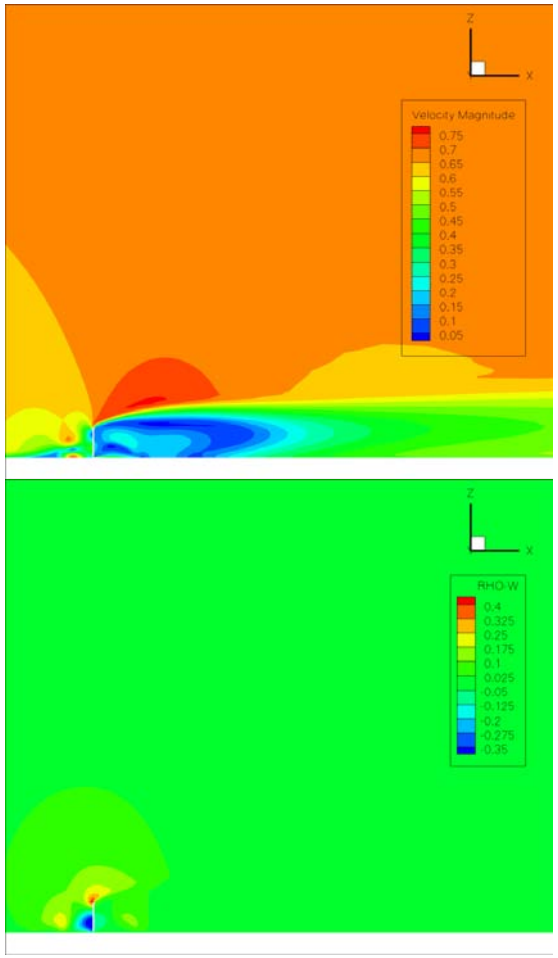




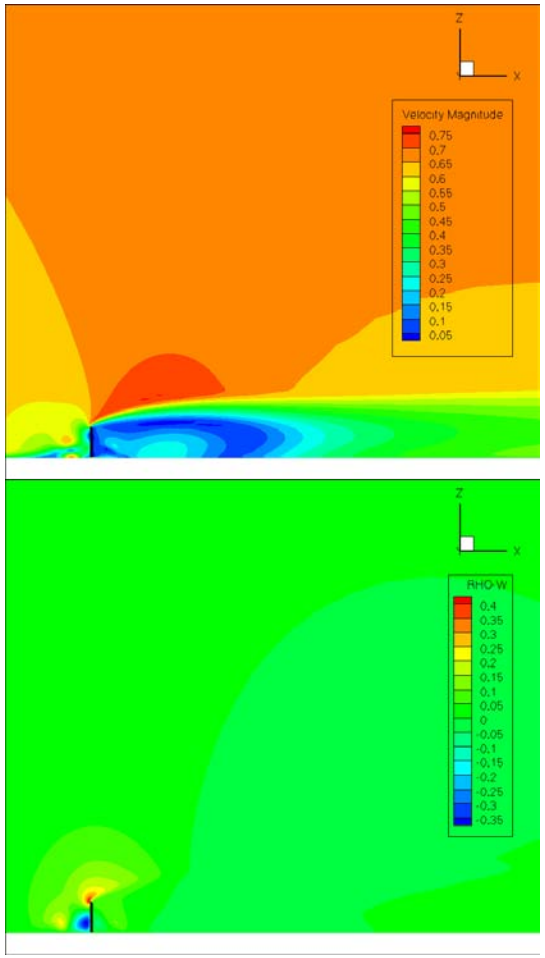
KRATOS
DEFENSE & SECURITY SOLUTIONS

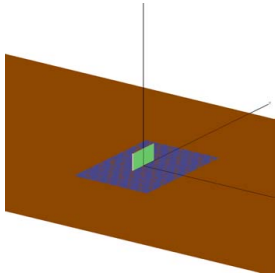
Porous Fence Model "Validation"

Gridded Spoiler



Porous Media Spoiler





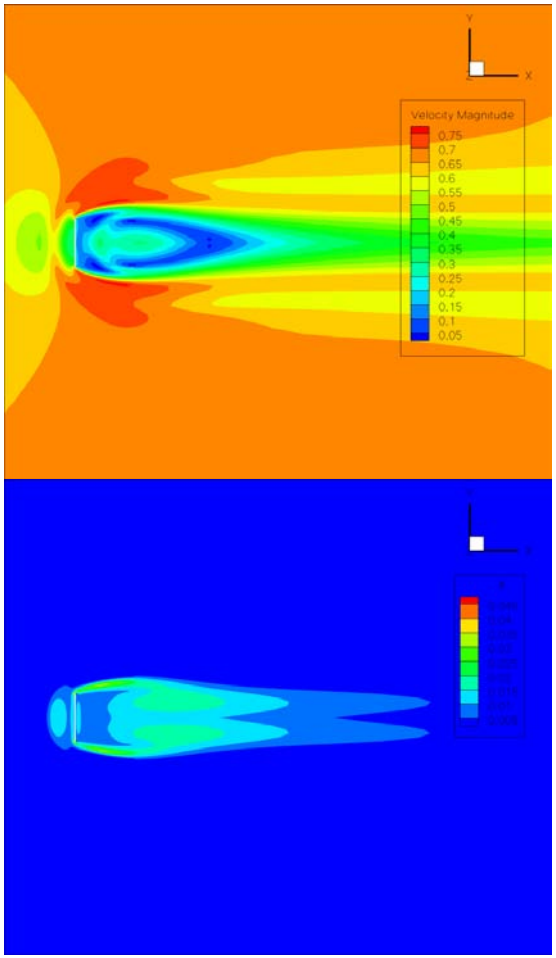
Velocity Magnitude

Velocity Normal to Plate

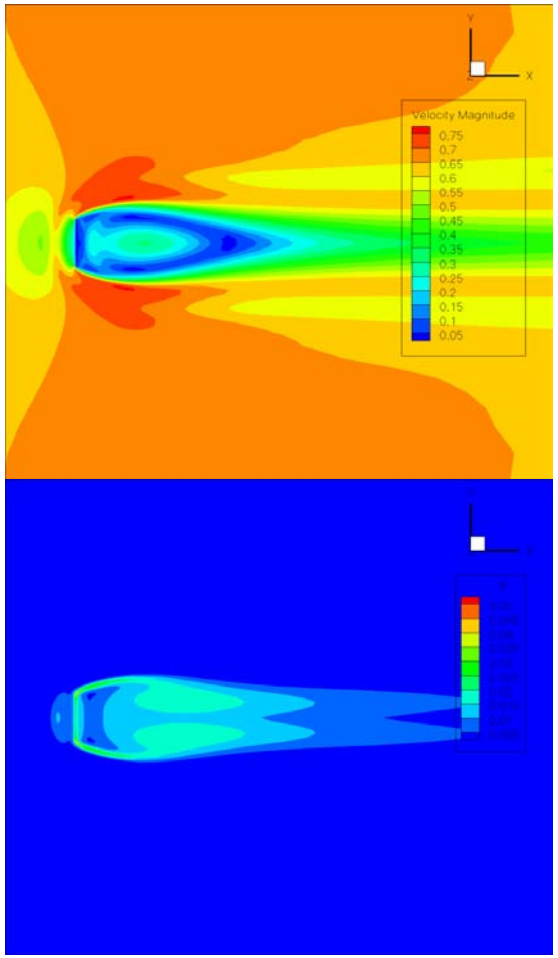
KRATOS
DEFENSE & SECURITY SOLUTIONS

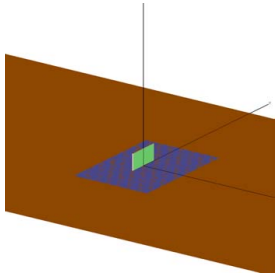
Porous Fence Model "Validation"

Gridded Spoiler



Porous Media Spoiler





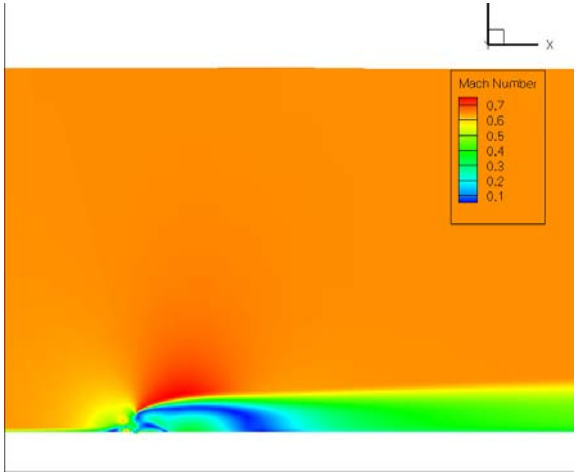
Velocity Magnitude

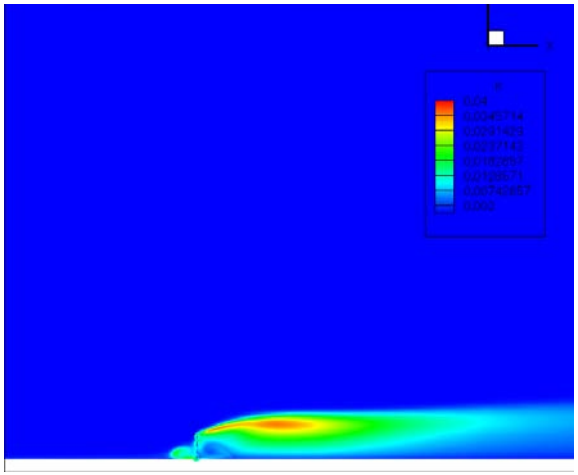
Turbulence Kinetic Energy

KRATOS
DEFENSE & SECURITY SOLUTIONS

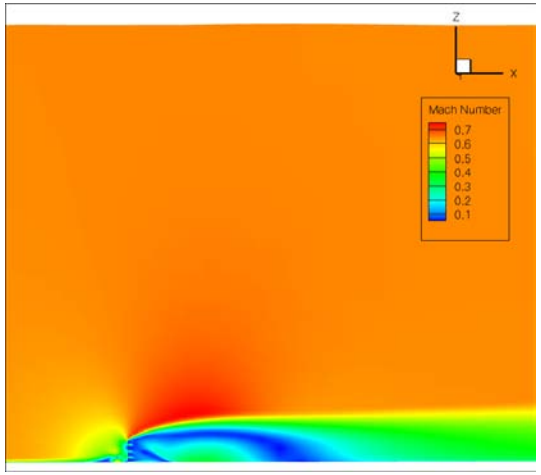
Porous Fence Model Improvements

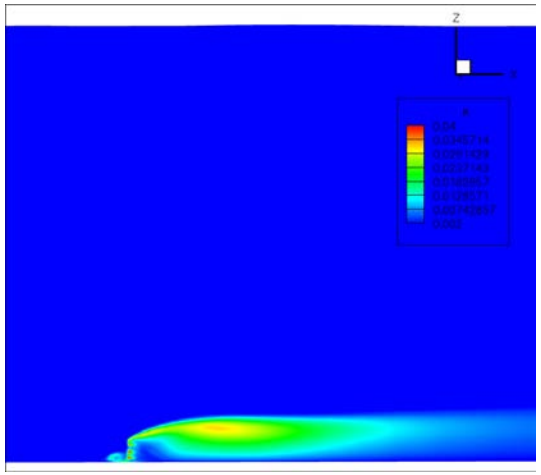
Gridded 21 hole Spoiler

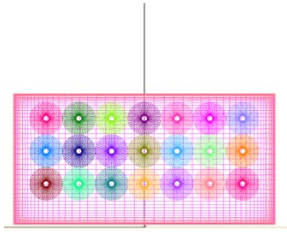




Porous Media “21 hole” Spoiler







Mach Number

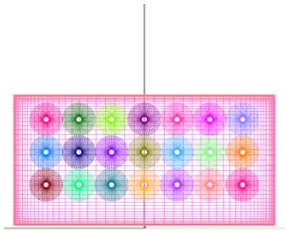
Turbulence Kinetic Energy

KRATOS
DEFENSE & SECURITY SOLUTIONS

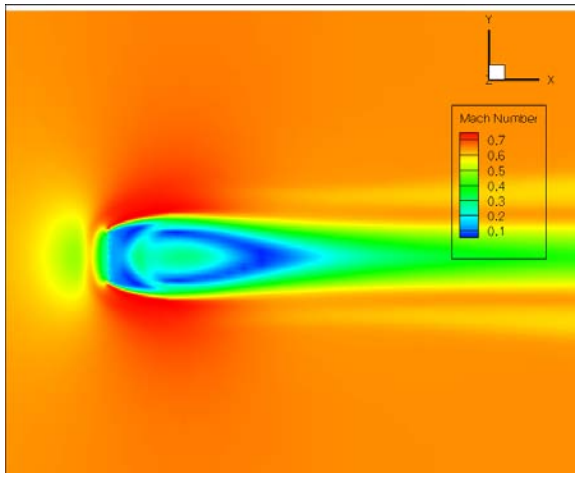
Porous Fence Model Improvements

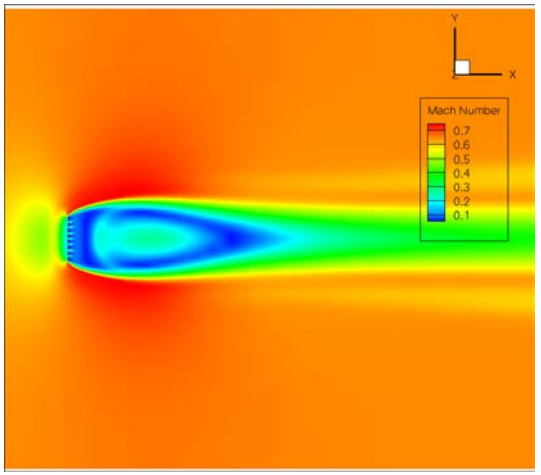
Gridded 21 hole Spoiler

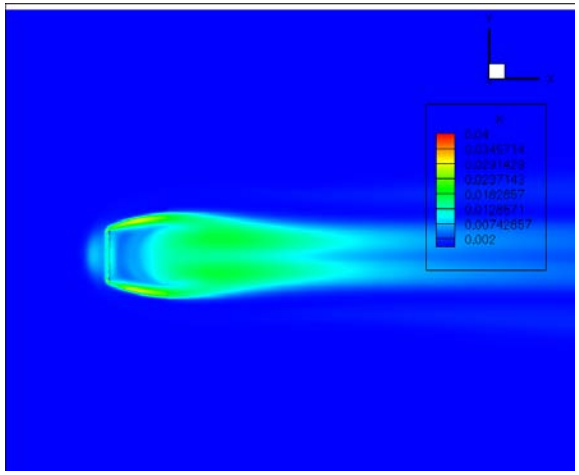
Porous Media “21 hole” Spoiler

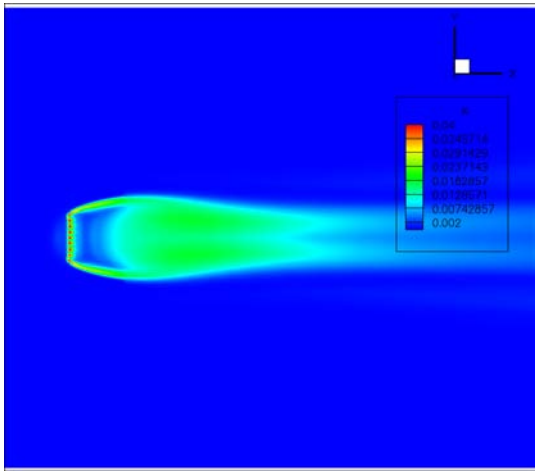


Mach number









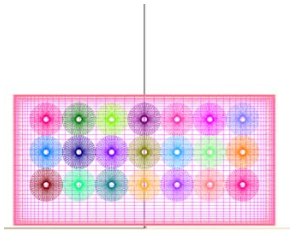
Turbulence Kinetic Energy

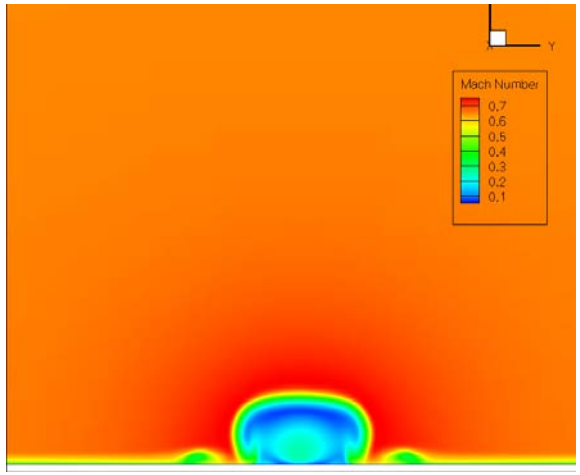
KRATOS
DEFENSE & SECURITY SOLUTIONS

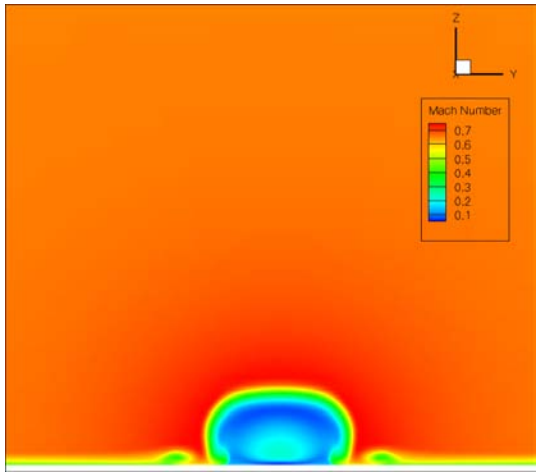
Porous Fence Model Improvements

Gridded 21 hole Spoiler

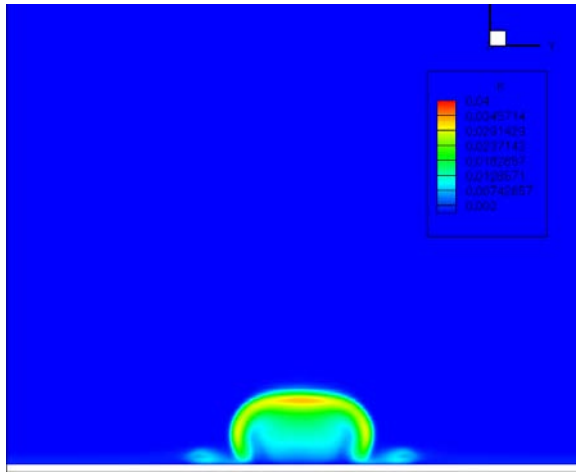
Porous Media “21 hole” Spoiler

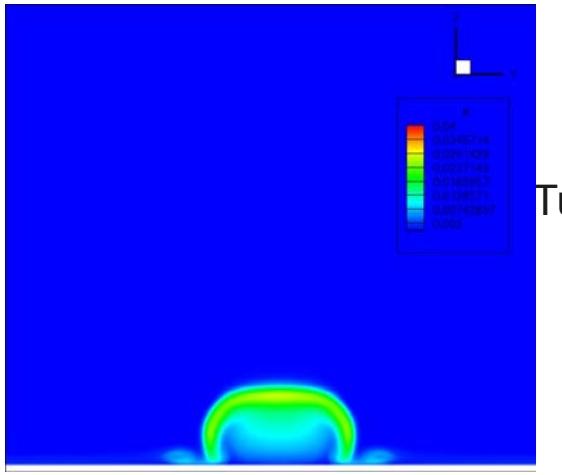






Mach Number





Turbulence Kinetic Energy

KRATOS
DEFENSE & SECURITY SOLUTIONS

Coupled CFD/Aero-Optics Modeling

"Then"

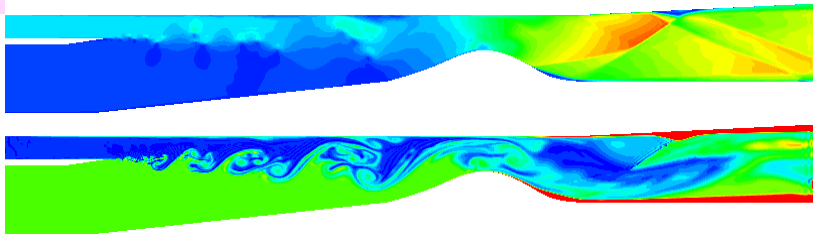
OVERFLOW 2.1x

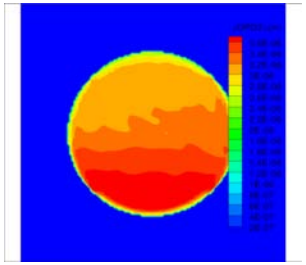
aeroOpticsModule

Beam Profiles

Post Processors

Aero-Optic Metrics





"Now"

Mesh Preprocessor

Paraxial Solver Class Library

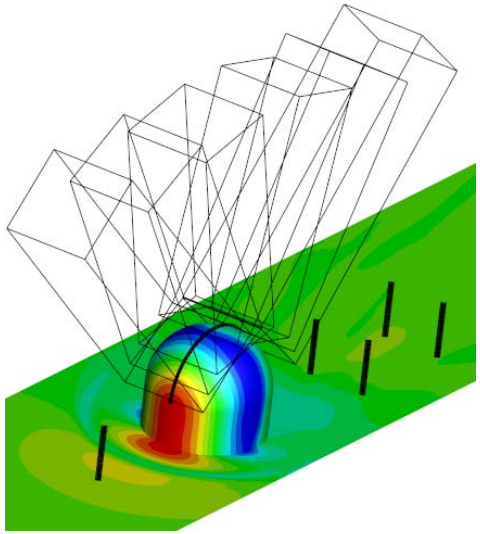
Overset or Unstructured CFD Solver

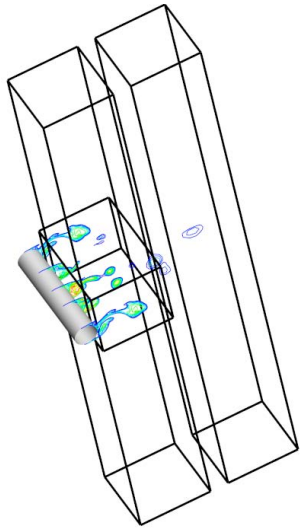
Paraxial Solver Class Library

Propagated Beam Data Files

Paraxial Solver Class Library

Aero Optical Quality Metrics





KRATOS
DEFENSE & SECURITY SOLUTIONS

Acknowledgements

• Dr. Scott Sherer AFRL/RBAT, Computational Aerosciences Branch: Technical Monitor/ Sponsor

• Aerosciences and Engineering Analysis Group (Kratos/DFI):

• Bob Tramel (ex-DFI, now Kord Technologies)

    • Original PI, formulation of Paraxial Beam Solver

KRATOS
DEFENSE & SECURITY SOLUTIONS

General Description of Problem

- Characterize the near-field, aerodynamically-induced distortion of optical beams as they are propagated through the unsteady, compressible, turbulent flow fields in the vicinity of air vehicles.
    - "Aero-Optics" as opposed to "Atmospheric Propagation"
- Optical/Beam distortion caused by changes in index of refraction and beam spreading/attenuation:
    - Spatial variations in density (index of refraction) induce phase differences in planar waves
        - Beam phase and amplitude distribution related to beam spreading perpendicular to propagation direction
    - Temporal variations of density induce temporal variations in beam patterns
- Need to model:

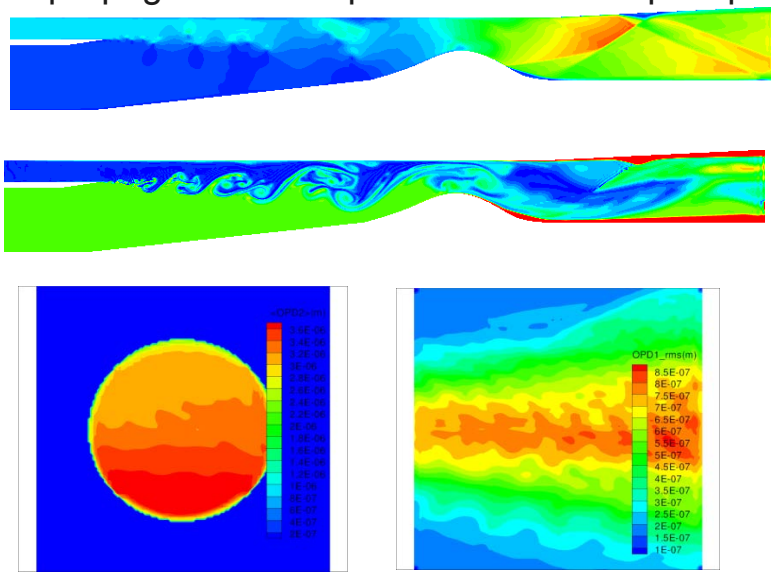
Compressible flow field to capture the large-scale density variations.
Turbulent fluctuations to capture the small scale density variations.

KRATOS
DEFENSE & SECURITY SOLUTIONS

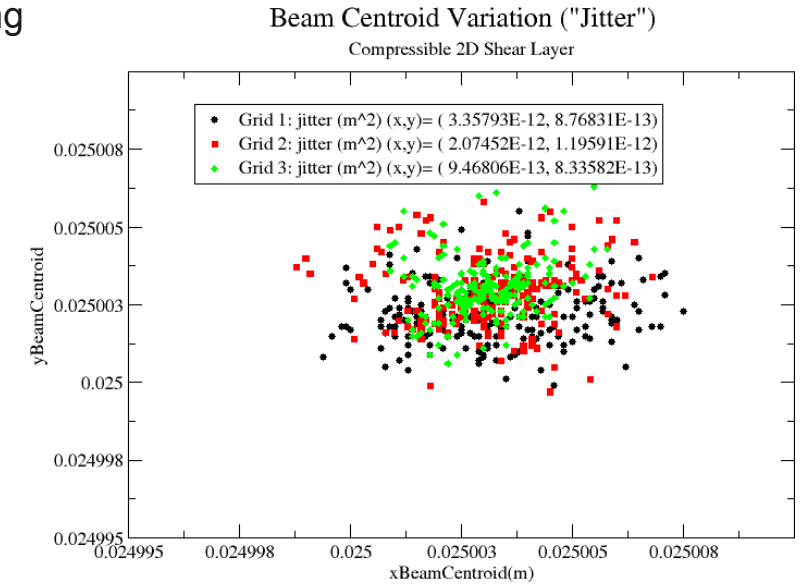
Original Approach

OVERFLOW 2.1x
aeroOpticsModule
→ Beam Profiles → Post Processors → Aero-Optic Metrics

- Integrate a Paraxial Beam Equation solver into OVERFLOW 2.1x
- Propagate Beams at specified time steps through “optical grids”
- Deposit propagated beam profiles to disk for post processing



- Post Processing utilities:
  - Produce visualization files (beam amplitude, OPD…)
  - Compute AO statistics/Metrics



FROM STRENGTH TO SUCCESS™

KRATOS
DEFENSE & SECURITY SOLUTIONS

Paraxial Beam Equation Solver Formulation

- Reduce Maxwell's equations to the Paraxial Beam equation by:
  - Non-conducting air, constant magnetic permeability, flow characteristic time >> emag propagation time, planar wave…
  - Paraxial Approximation

$$\nabla_{\perp}^{2} A + \frac{\partial^2 A}{\partial z^2} + 2jk\frac{\partial A}{\partial z} + \left(\frac{n^2\omega 2}{c_0^2} - k^2\right)A = 0$$

$$\frac{\partial^2 A}{\partial z^2} << k\frac{\partial A}{\partial z}$$

- Non-Dimensionalization by wavelength results in:

$$\frac{\partial^2 A}{\partial \xi^2} + \frac{\partial^2 A}{\partial \eta^2} + 4\pi j\frac{\partial A}{\partial \zeta} + 4\pi^2\left[\left(\frac{n}{n_r}\right)^2 - 1\right]A = 0$$

- Parabolic, Operator Splitting Approach:
  - Phase Shift:

$$j\frac{\partial A}{\partial \zeta} + \pi\left[\left(\frac{n}{n_r}\right)^2 - 1\right]A = 0$$

  - Wave Advance:

$$\frac{\partial^2 A}{\partial \xi^2} + \frac{\partial^2 A}{\partial \eta^2} + 4\pi j\frac{\partial A}{\partial \zeta} = 0$$

Phase Shift

$$A^*_{kl} = A_{kl}(\zeta)e^{j\pi\left[\left(\frac{n}{n_r}\right)^2 - 1\right]\Delta\zeta}$$

Wave Advance

$$\hat{A}_{kl}(\zeta + \Delta\zeta) = \hat{A}^*_{kl}e^{-j\left[\left(\frac{2\pi k}{N_x\Delta\xi}\right)^2 + \left(\frac{2\pi l}{N_y\Delta\eta}\right)^2\right]\frac{\Delta\zeta}{4\pi}}$$

KRATOS
DEFENSE & SECURITY SOLUTIONS

Integration with OVERFLOW 2.1x

- aeroOptics Solver is a Fortran90 Module with procedures called by the master process (IAM.eq.MNGR)
- "Optics" (CFD) grids donate density to "Spectral" (aeroOptics) grids
  - Spectral grid dimensions not tied to optics (CFD) grid dimensions
  - Identify which CFD grids are Optics grids, and associate aeroOptics data with them (such as spectral grid size, imposed beam type, …)
- Minimal changes to OVERFLOW:
  - OVERST: Initialization
  - OVERGL: read in "aeroOptics" namelist, distribute limited amount of namelist data to children processes
  - INEND: Parallel Gather of the density data onto the optics grid, Call the aeroOptics solver (from MNGR)
  - OVERDO: Finalization, freeing of memory
- Use the FFTW3 library for the Discrete Fourier Transforms

Aero-Optical Quality Metrics

- OVERFLOW-aeroOptics writes out (appends) beam data to a binary data file for each optics grid for every paraxial beam solve.
- Post process these files for visualization and Aero-Optics Quality Metrics:
  - Beam Centroids for each time slice
  - Average Beam Centroid
  - Jitter
  - Boresight Error
  - Contained Energy Diameters

$$I(f) = \iint_x \int_y AA^* f(x,y)\,dx\,dy$$

$$x_n = \frac{I(x)}{I(1)} \qquad y_n = \frac{I(y)}{I(1)}$$

$$\bar{x}_n = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \bar{y}_n = \frac{1}{N}\sum_{n=1}^{N} y_n$$

$$\sigma_x = \frac{1}{N}\sum_{n=1}^{N}\left(x_n - \bar{x}_n\right)^2 \qquad \sigma_y = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \bar{y}_n\right)^2$$

$$\varepsilon_x = \left|\bar{x}_n - x_0\right| \qquad \varepsilon_y = \left|\bar{y}_n - y_0\right|$$

Optical Path Differences:
- By Path Integral
- By Phase Differences between propagated and reference beam

$$OPD_1 = K_{gd}\lambda\int_0^{L_z} \rho\, d\zeta \qquad OPD_2 = \frac{\lambda}{2\pi}\left(\phi_L - \phi_r\right)$$

KRATOS
DEFENSE & SECURITY SOLUTIONS

Validation: Gaussian Beam Propagation

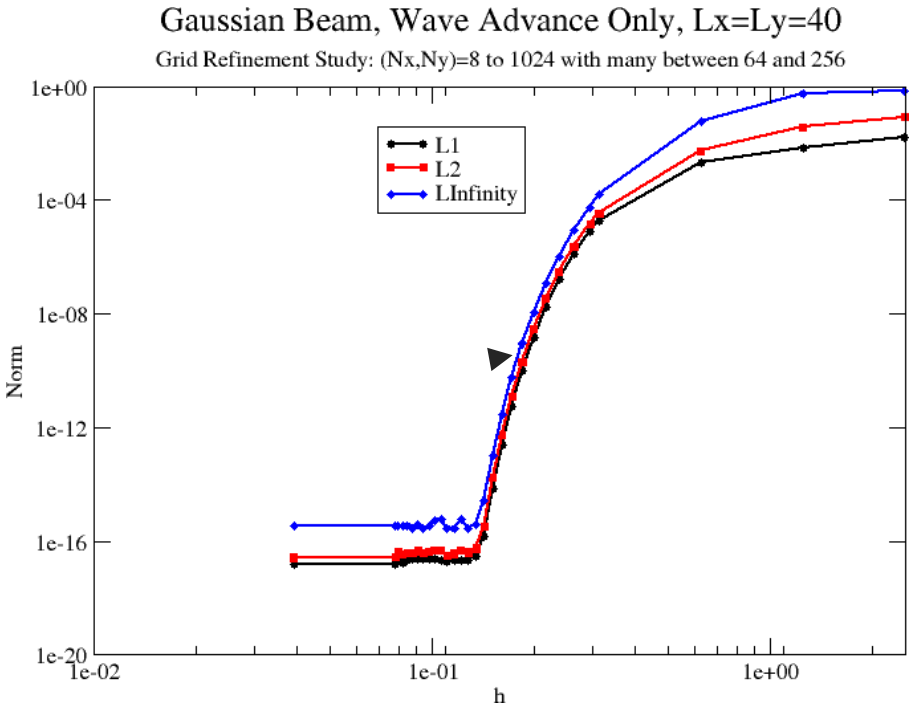
- Propagate a Gaussian Beam Profile through vacuum (no wave advance)
- Analytical Solution

$$A = \frac{A_0}{\left(1 + j\tilde{\alpha}\zeta\right)} e^{-\frac{\pi\tilde{\alpha}\tilde{r}^2}{\left(1+j\tilde{\alpha}\zeta\right)}} \qquad \tilde{\alpha} = \frac{1}{\pi\tilde{r}_0^2} + \frac{j}{\tilde{f}}$$

$$\tilde{r}_0 \equiv r_0/\lambda \qquad \tilde{f} = f/\lambda$$

- Evaluate Error by mesh refinement
- Exhibits “Spectral Convergence”

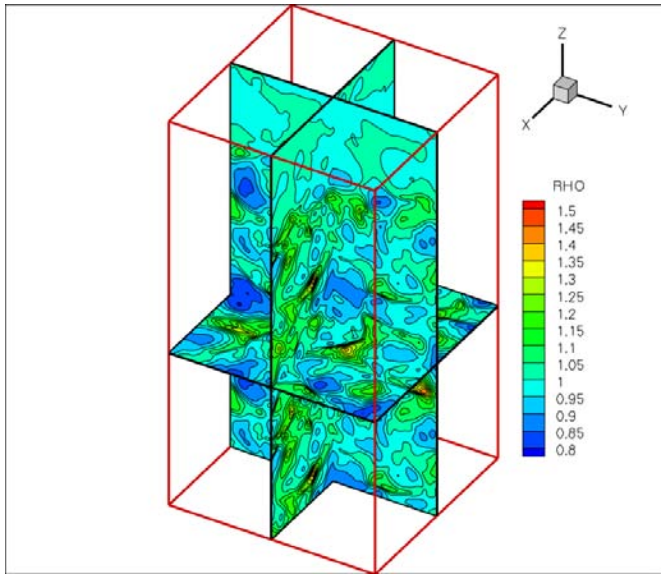
”Spectral” convergence:
Error decays faster than $\varepsilon \approx 1/N_x^k$
for any value of k



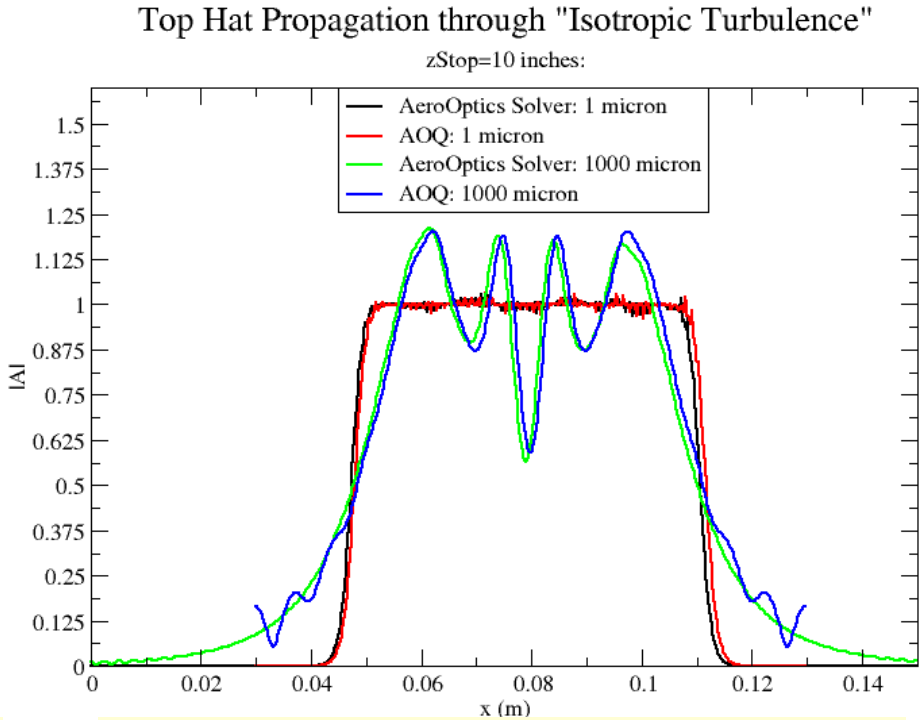
KRATOS
DEFENSE & SECURITY SOLUTIONS

Propagation of a Top Hat Profile

- Compare aeroOptics solver to AOQ for propagation of Top Hat through isotropic turbulence field to Lz=10 inches
- Some differences between the aeroOptics grid and AOQ:
  - AOQ: 10x10 cm
  - aeroOptics: $(x, y) \in [-\pi, \pi]$ inches



Isotropic turbulence field computed via FDL3DI (supplied by AFRL/RBAC)



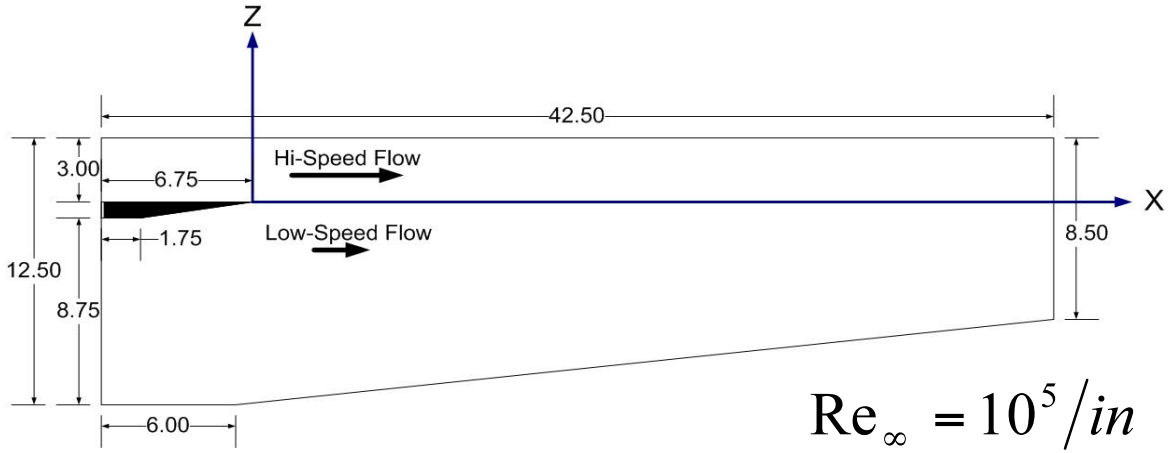
Compares well to AOQ: Note effect of smaller domain size upon AOQ solution

Two-dimensional Compressible Shear Layer

- Corresponds to a run in the Notre Dame Compressible Shear Layer Wind Tunnel (CSLWT)

$M_{high} = 0.65$

$M_{low} = 0.12$



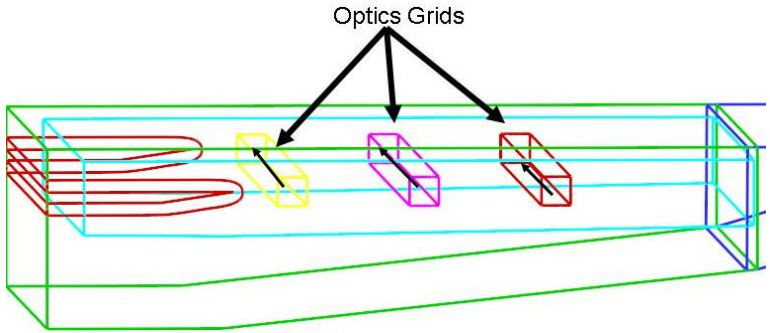
$\mathrm{Re}_\infty = 10^5 / in$

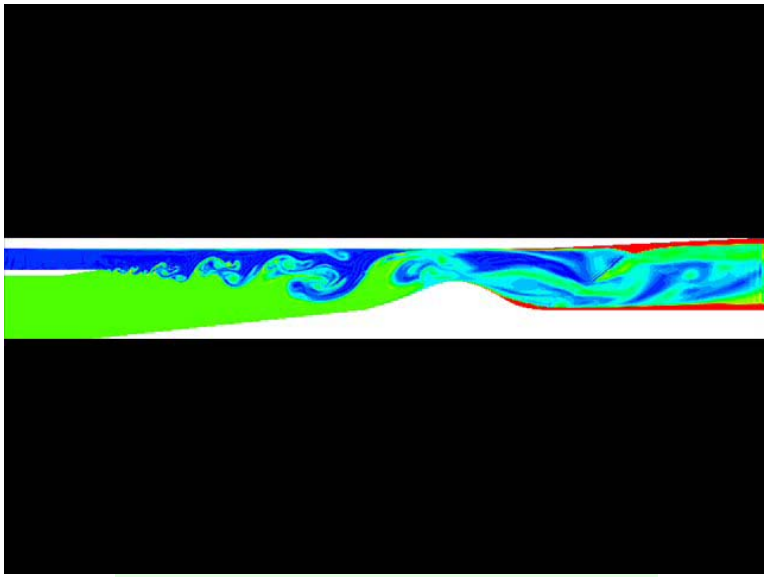
- OVERFLOW 2.1x run:
  - Dual-time stepping, 2nd order in time, dtPhys=0.05, 5 sub-iterations per time step.
  - WENOM 5th-order spatial scheme
  - DDES using Menter's SST model with compressibility correction turned on
  - Two-dimensional problem (3 planes)
  - "FMG" run for 2 levels of mesh sequencing after initializing flow to a very crude initial condition
  - 2000 time steps to settle flow
  - Parallel on 56 processors on "bender" (Linux, DFI)
  - 3328 sec (flow) 110 sec (paraxial) (3.3%)

KRATOS
DEFENSE & SECURITY SOLUTIONS

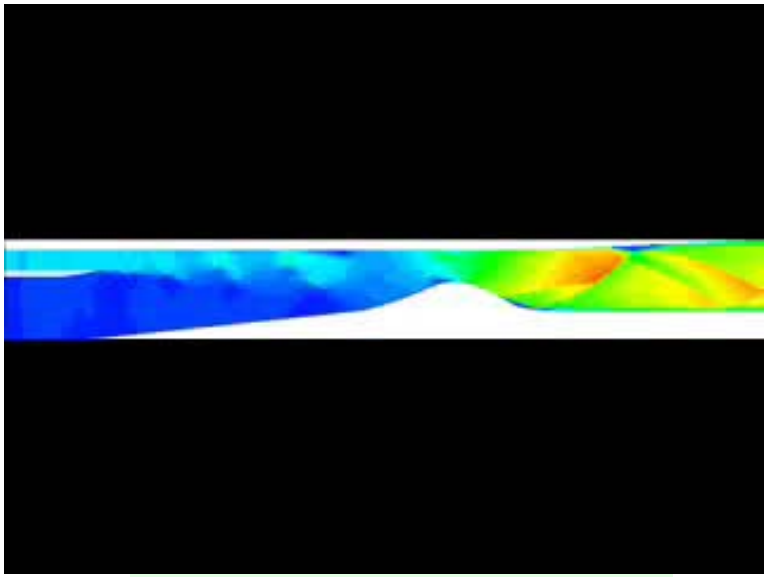
Two-dimensional Compressible Shear Layer

- "Optics" grids generated in the shear layer region
- LEVEL2 must be false in these grids, otherwise large portions of the optics grids can be blanked out.





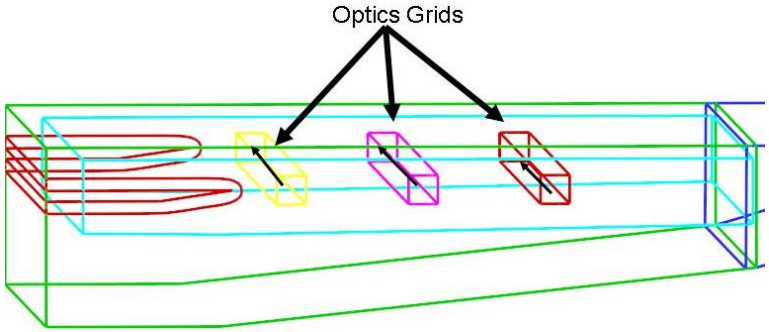
S1 entropy measure



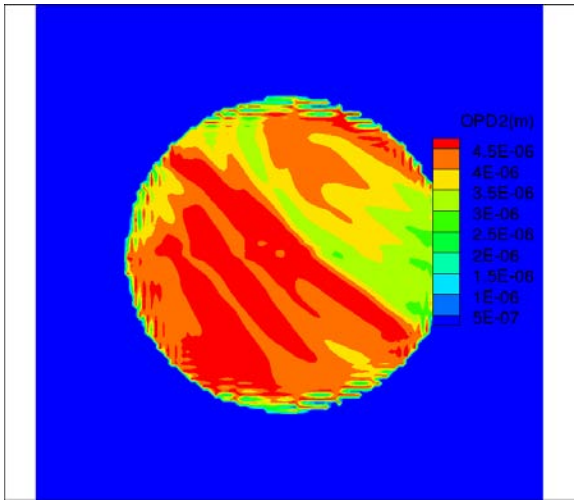
Mach Number

Two-dimensional Compressible Shear Layer

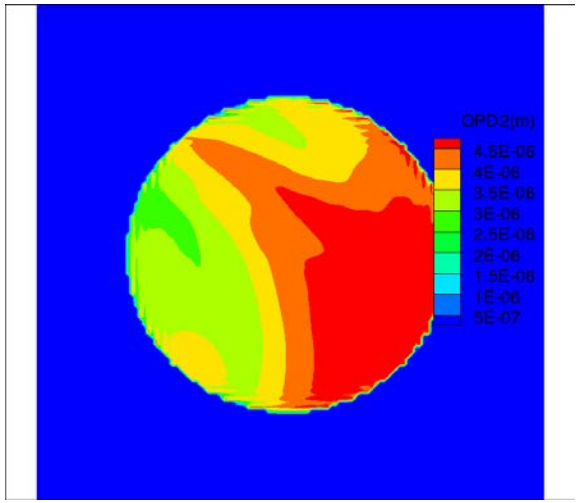
- OPD2 calculated via phase difference



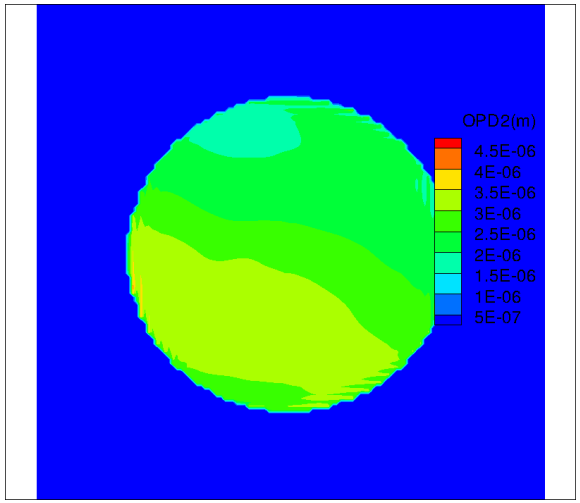
$$\lambda = 10^{-5}\,(meters)$$



Optics Grid 1



Optics Grid 2

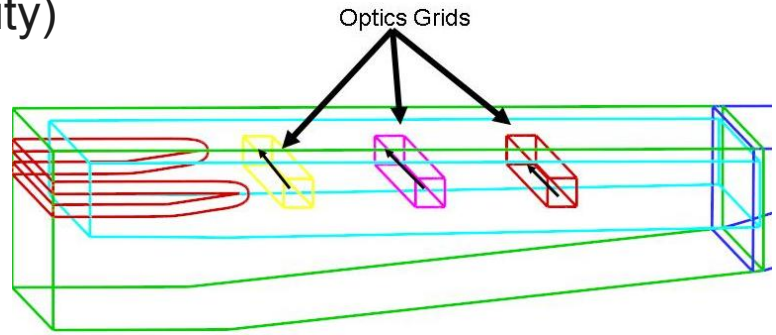


Optics Grid 3

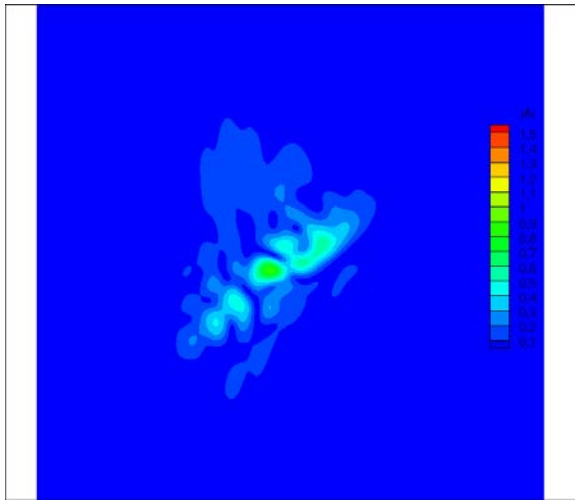
KRATOS
DEFENSE & SECURITY SOLUTIONS

Two-dimensional Compressible Shear Layer

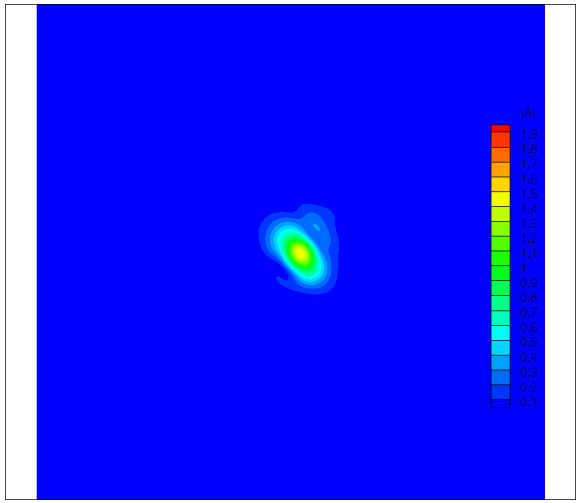
• Magnitude of beam propagated through tunnel then 10 meters more through air (constant density)



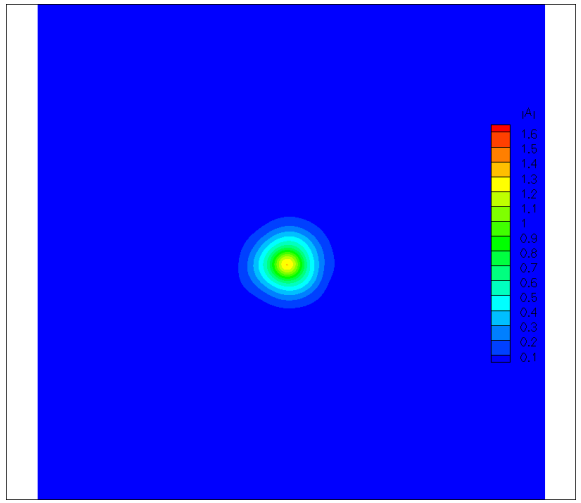
$$\lambda = 10^{-5}\,(meters)$$



Optics Grid 1



Optics Grid 2



Optics Grid 3

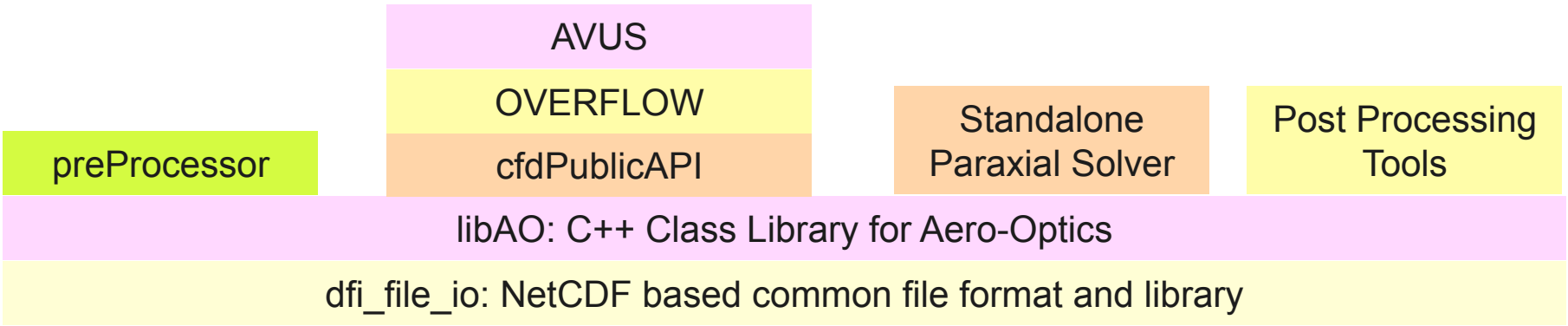
KRATOS
DEFENSE & SECURITY SOLUTIONS

Lessons Learned and Recommendations

- Optics Grids tied too closely to CFD Grids: Must build optics grid into mesh system and they must win any blanking battles
- Solver too tightly integrated: Can't integrate with other codes easily, can't run standalone…

Recommendations

- Develop aero-optics solver library for integration into multiple CFD codes
- Modular approach to allow re-use of software (class libraries)
- Use 3rd party libraries and software wherever possible to reduce build time



KRATOS
DEFENSE & SECURITY SOLUTIONS

dfi_file_io: NetCDF-based File Format

- NetCDF: Network Common Data Format library and tools
  - C, C++, F77/F90 API to create and access data files for array-oriented scientific data
  - Open source, free, supported by Unidata (UCAR/NSF)
- dfi_file_io:
  - Library/namespace built upon NetCDF to read/write named std::vectors of basic data types (ints, floats, doubles, strings, complex)
  - Easy to write/read data to/from a commonly shared file

```
dfi_file_io::read(commonFileName,"X",X);
```

```
dfi_read_int(trim(preProcFileName)//CHAR(0),trim(varName)//CHAR
(0),donorCellList(p)%nDonorCells)
```

  - Built in error handling, resizes and fills in vectors on read
  - Not designed to be a full-fledged CFD common file format like CFD-DTF or CGNS
  - ncgen⇔ncdump: Used to process easy to read ASCII files for common parameter input

KRATOS
DEFENSE & SECURITY SOLUTIONS

libAO:: C++ Class Library for Aero-Optics

- Collection of classes, functions and executables related to aero-optics, paraxial beam solution, CFD pre-processing, CFD model integration and post-processing
- Example classes/functionality
  - paraxialSolverClass: encapsulates the paraxial solver
  - paraxialSolverExecutorClass: Used to marshall collections of paraxial solves
  - beamGridClass: defines beam grid extents
  - paraxInterpolatorClass: derived from beamGridClass, used to perform fast interpolation needed by paraxialSolver, is member data of paraxialSolverClass
  - cartSearchSpace<T>: templated spatial search class to find nearest neighbors
  - oversetPreProcessorClass: overset mesh pre-processor class
  - unstructuredFVMPreProcessorClass: unstructured FVM mesh pre-processor class
- Use third-party, open-source and standard libraries:
  - boost: Some useful classes
  - fftw3: Fastest Fourier Transform in the West (3)

KRATOS
DEFENSE & SECURITY SOLUTIONS

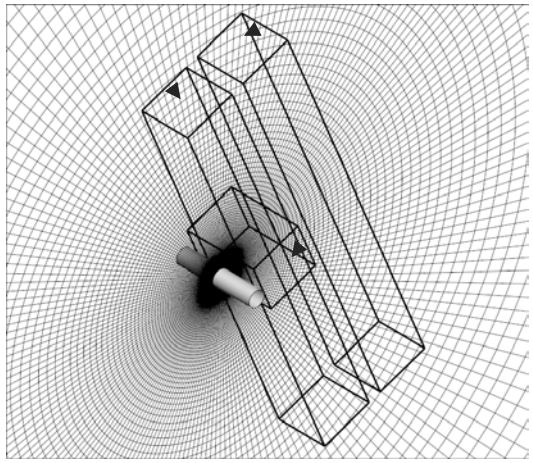
Mesh Pre-Processing Phase

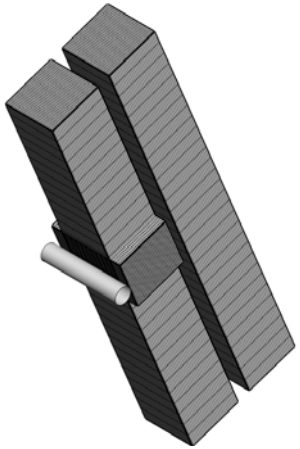
• Given a CFD mesh system (overset, unstructured) and a collection of beamGrids, pre-compute the CFD model donor nodes(cells) interpolants for a given interpolator grid size.

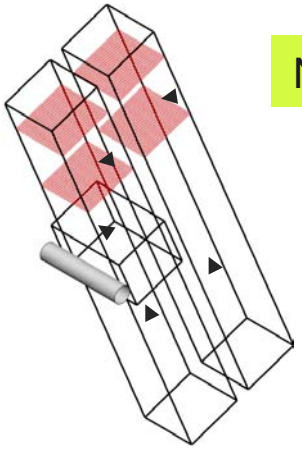
beamGrids: define extent and direction of beam propagation

interpolatorGrids: pre-compute CFD donor interpolants (nodes, weights) onto interpolator grids

Spectral Grids: Marching planes in propagation direction







N_x x N_y

• Pre-processing computes and stores CFD donor interpolants (nodes, cells, weights) to interpolate CFD solution onto interpolator grid

• Interpolator grid provides fast interpolation onto spectral grid

KRATOS
DEFENSE & SECURITY SOLUTIONS

Integration within OVERFLOW (and others)

- Provide a minimal set of API called by the flow solver

```
initializeParaxialBeamSolvers(commonFileName);
for each timestep
          . . .
          for each beam
                    setRhoDonor(beamNumber,RhoDonor,timeStamp);
                    runParaxialBeamSolver(beamNumber,timeStamp);
finalizeParaxialBeamSolvers
```

- initializeParaxialBeamSolvers:
  - Reads common file produced by pre-processor (donors, weights, …)
  - Initializes executor that runs paraxial solvers
- setRhoDonor:
  - CFD solver collects only the needed donor density data and supplies it to the executor
- runParaxialBeamSolver:
  - Solves paraxial equations, writes donor data, beam profiles,… to be used later
- finalizeParaxialBeamSolvers:
  - Frees up memory, closes files, finalizes things

KRATOS
DEFENSE & SECURITY SOLUTIONS

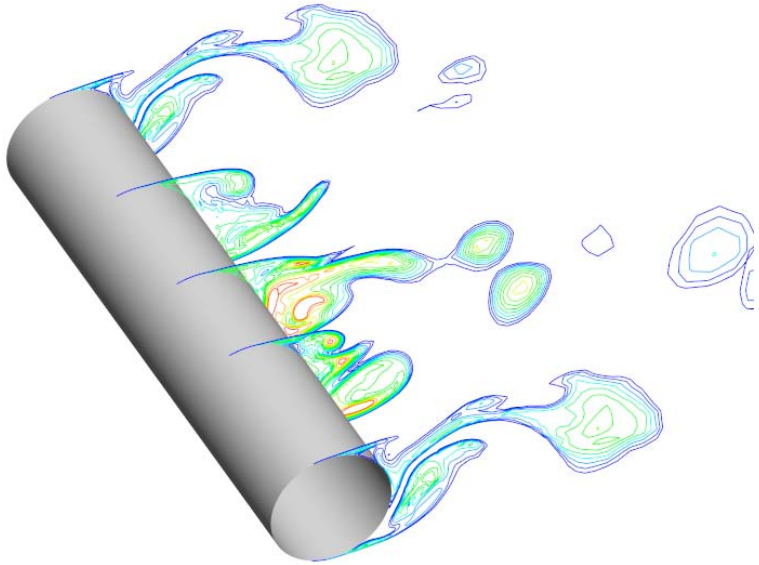
Integration within OVERFLOW (and others)

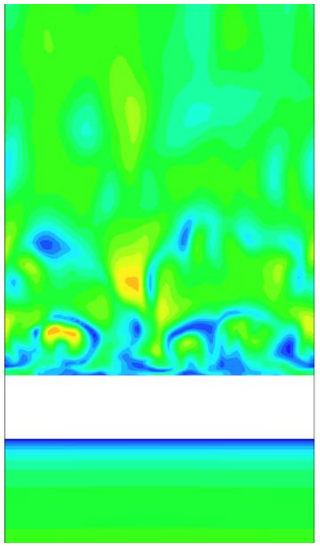
- Minimal hassle integrating within OVERFLOW:
  - OVERST:
    - initializeParaxialBeamSolvers
  - OVERGL:
    - Read an aeroOptics namelist, read common file, scatter data
  - INEND:
    - Gather donor density, setRhoDonor, runParaxialBeamSolver
  - OVERDO:
    - finalizeParaxialBeamSolvers
- Easily integrated into AVUS
- Master process does all the paraxLib invocations, density gather, i/o etc.

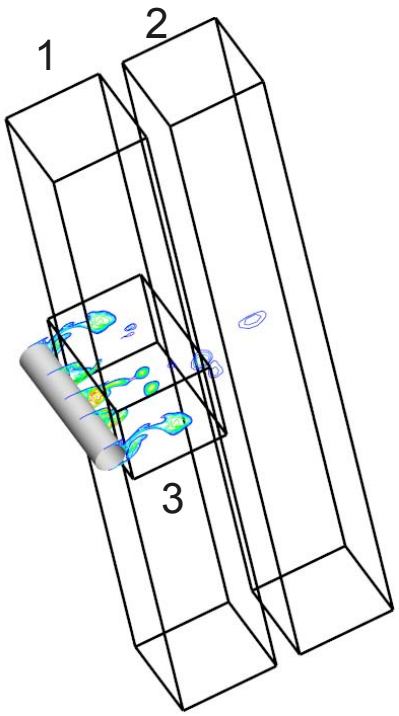
KRATOS
DEFENSE & SECURITY SOLUTIONS

Example Problem: Shedding Cylinder

- Circular cylinder, diameter=1 inch
- M=0.2, Strouhal number used to get an estimate on time step:
  - 200 steps per shedding cycle
- Wall functions, SA-DES, third-order upwind, HLLC, 16000 time steps, dual time-stepping scheme



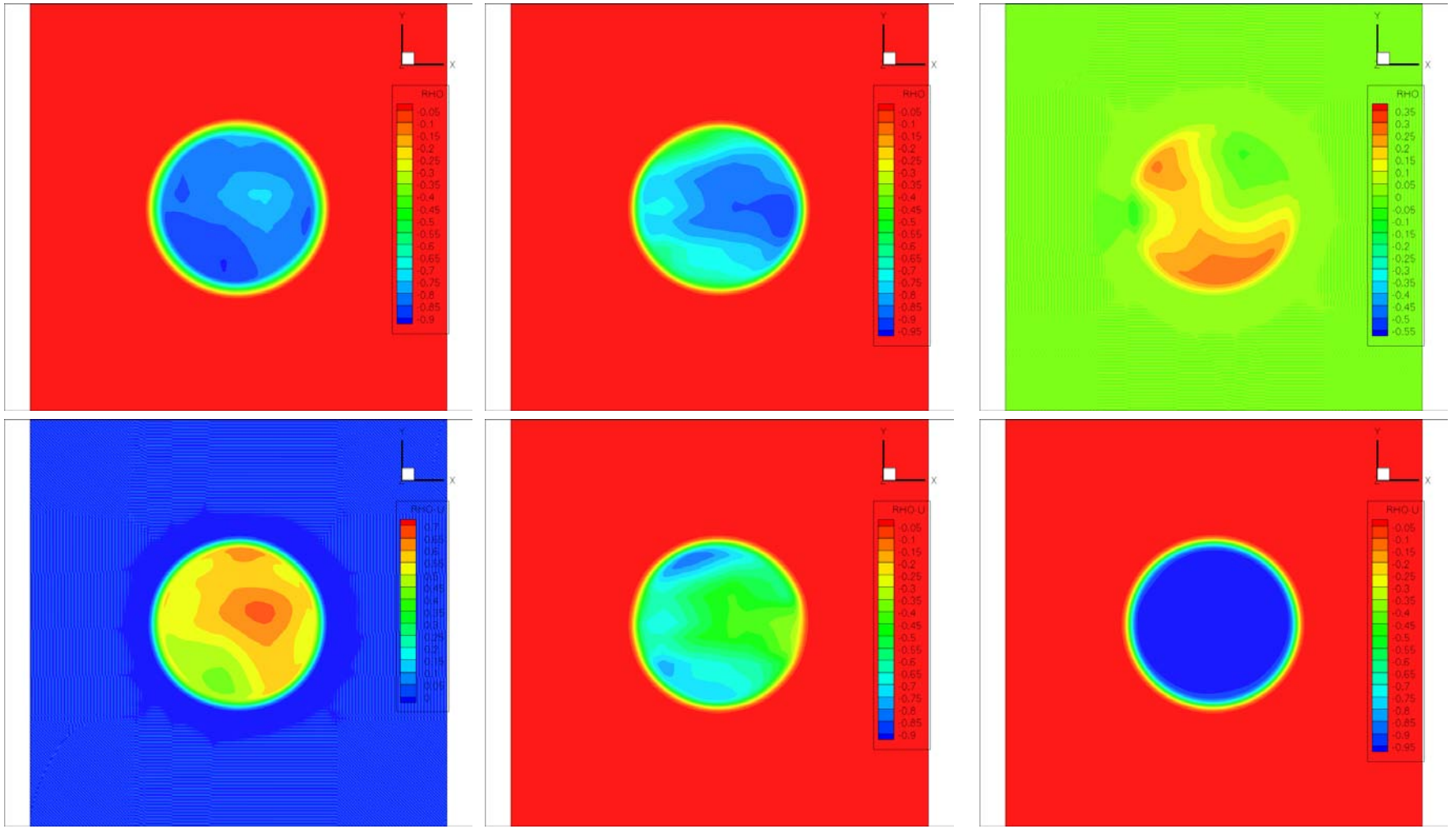




KRATOS
DEFENSE & SECURITY SOLUTIONS

Example Problem: Shedding Cylinder

- Real (top) and Imaginary Components of beam propagated through 200 timesteps of flow



KRATOS
DEFENSE & SECURITY SOLUTIONS

Summary

- Integrated library into OVERFLOW 2.1z:
    - Will integrate into 2.2 soon…
- Integrated into AVUS
    - Demonstrated for the same shedding cylinder case
- Potential integration into FDL3DI
- Need to finish post-processing
- Validation cases:
    - Hemispherical turret
    - Flat windowed turret with pins
    - Other cases as needed
- Documentation, installation scripts, …