PHASTA
(Parallel Hierarchic Adaptive Stabilized Transient Analysis)

Kenneth E. Jansen

Outline

- Introduction
- Compressible Flow Equations
- Finite Element Discretization
- PHASTA Algorithms and Past Parallel performance
- PHASTA Applications
- Demo

https://github.com/PHASTA

  - Test cases
  - Clone/build/run
  - Viz, input, outputs, etc.

PHASTA: Core of 2 Exascale Early Science Projects

Aurora Early Science Projects

Simulation

Simulation Co-PIs CO:
Jed Brown CS
John Evans
John Farnsworth
Philippe Spalart

Extreme-Scale Unstructured Adaptive CFD
PI Kenneth Jansen, University of Colorado at Boulder

Extreme-Scale Cosmological Hydrodynamics
PI Katrin Heitmann, Argonne National Laboratory

NWChemEx: Tackling Chemical, Materials & Biochemical Challenges in the Exascale Era
PI Theresa Windus, Iowa State University and Ames Laboratory

High-Fidelity Simulation of Fusion Reactor Boundary Plasmas
PI C.S. Chang, Princeton Plasma Physics Laboratory

Extending Moore's Law Computing with Quantum Monte Carlo
PI Anouar Benali, Argonne National Laboratory

Data

Data Co-PIs CO:
Stephen Becker APPM
Jed Brown CS
Alireza Doostan
John Evans
John Farnsworth
Philippe Spalart

Extreme-Scale In-Situ Visualization and Analysis of Fluid-Structure-Interaction Simulations
PI Amanda Randles, Duke University and Oak Ridge National Laboratory

Simulating and Learning in the ATLAS Detector at the Exascale
PI Walter Hopkins, Argonne National Laboratory

Data Analytics and Machine Learning for Exascale Computational Fluid Dynamics
PI Ken Jansen, University of Colorado Boulder

Dark Sky Mining
PI Salman Habib, Argonne National Laboratory

Exascale Computational Catalysis
PI David Bross, Argonne National Laboratory

Learning

Accelerated Deep Learning Discovery in Fusion Energy Science
PI William Tang, Princeton Plasma Physics Laboratory

Virtual Drug Response Prediction
PI Rick Stevens, Argonne National Laboratory

Many-Body Perturbation Theory Meets Machine Learning to Discover Singlet Fission Materials
PI Noa Marom, Carnegie Mellon University

Enabling Connectomics at Exascale to Facilitate Discoveries in Neuroscience
PI Nicola Ferrier, Argonne National Laboratory

Machine Learning for Lattice Quantum Chromodynamics
PI William Detmold, Massachusetts Institute of Technology

https://aurora.alcf.anl.gov

Introduction to CFD

- Unstructured grid methods offer:
  - Complex geometry discretization
  - Adaptivity to locally match length scales to those dictated by physics
- Downside of unstructured grid CFD
  - Lack of structure → complicated code → more computational effort per point
  - Post processing solution more difficult
  - Less mature
- Popular methods
  - Finite Volume (least complex, upwind diss., high order?)
  - Discontinuous Galerkin (natural for Euler Equations, viscous flows less natural)
  - Stabilized Finite Element (robust, minimally diss., high order)

Governing Equations

- Compressible Navier-Stokes equations
  - Conservation of Mass

$$\rho_{,t} + [\rho u_i]_{,i} = 0$$

  - Conservation of Momentum (vector for i=1, 2, 3 is x_1, x_2, x_3 direction)

$$[\rho u_i]_{,t} + \lfloor \rho u_i u_j \rfloor_{,j} = \lfloor - p\delta_{ij} + \tau_{ij} \rfloor_{,j} + \rho b_i$$

  - Conservation of Energy

$$[\rho e_{tot}]_{,t} + [\rho u_i e_{tot}]_{,i} = \lfloor - p u_i + \tau_{ij} u_j - q_i^{heat} \rfloor_{,i} + \rho(b_i u_i + \gamma)$$

- Indicial notation $[\ ]_{,t}$ is a time derivative, $[\ ]_{,i}$ is a spatial derivative in ith direction, contraction (implied sum) on repeated indices.

Constituitive Laws

- To close the N-S equations requires
  - Gas law (e.g., ideal gas)
  - Viscous stress model
    - Laminar Newtonian or Non-Newtonian
    - Turbulence Model
      - DNS, LES or RANS (may require additional transport equations)
  - Heat flux model
    - Same set typically extended from viscous law through Prandtl analogy

Flow Equations as a System

- The conservation equations in residual form

$$\underline{U}_{,t} + \underline{F}_{i,i} - \underline{S} = \underline{0}$$

$$where \quad F_i = F_i^{adv} + F_i^{diff}$$

$$U = \left\{\begin{matrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{matrix}\right\} = \rho \left\{\begin{matrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{tot} \end{matrix}\right\}$$

$$F_i^{adv} = \rho u_i \left\{\begin{matrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{tot} \end{matrix}\right\} + p \left\{\begin{matrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{matrix}\right\}$$

$$F_i^{diff} = \left\{\begin{matrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} u_j \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ -q_i^{heat} \end{matrix}\right\}$$

$$S = \rho \left\{\begin{matrix} 0 \\ b_1 \\ b_2 \\ b_3 \\ b_i u_i + \gamma \end{matrix}\right\}$$

Notation

- Define a quasi-linear operator on the variable vector as

$$L \equiv A_0 \frac{\partial}{\partial t} + A_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}\left(K_{ij} \frac{\partial}{\partial x_j} \right)$$

and $L = L_t + L_{adv} + L_{diff}$

- Here $A_i = F_{i,Y}^{adv}$ is the *i'th* Euler Jacobean matrix,

  K_{ij} is the diffusivity matrix, defined such that

$$K_{ij} Y_{,j} = F_i^{diff}$$

  and $A_0 = U_{,Y}$ is the change of variables metric.

- Using this notation we can write our equations in a simplified form as

$$LY = S$$

Finite Element Weak Form

- Differential form is multiplied by a smooth weighting function $W \in W_h^k$
- Product is integrated over an open space-time domain
- The solution is sought in the solution space

$$S_h^k = \left\{ v \middle| v(;,t) \in H^1(\Omega)^m,\ t \in [0,T],\ v|_{x\in\bar{\Omega}_e} \in P_k(\Omega)_e^m,\ v(.,t) = \hat{g}\ on\ \Gamma_g \right\}$$

- And the weight function space is given by

$$W_h^k = \left\{ w \middle| w(;,t) \in H^1(\Omega)^m,\ t \in [0,T],\ w|_{x\in\bar{\Omega}_e} \in P_k(\Omega)_e^m,\ w(.,t) = 0\ on\ \Gamma_g \right\}$$

- Galerkin weak form : find $Y \in S_h^k$ such that

$$\int_\Omega (W \cdot U_{,t} - W_{,i} \cdot F_i^{adv} + W_{,i} \cdot F_i^{diff})\, d\Omega - \int_\Gamma W \cdot (-F_i^{adv} + F_i^{diff})\, n_i d\Gamma = 0$$

$$0 = Galerkin\ +\ \sum_{e=1}^{nel} \int_{\Omega_e} \overline{\hat{L}W \cdot \tau(LU - \wp)\, d\Omega_e} \longrightarrow$$

Stabilization term

Discontinuity Capturing Operator

- Hughes and Mallet suggested

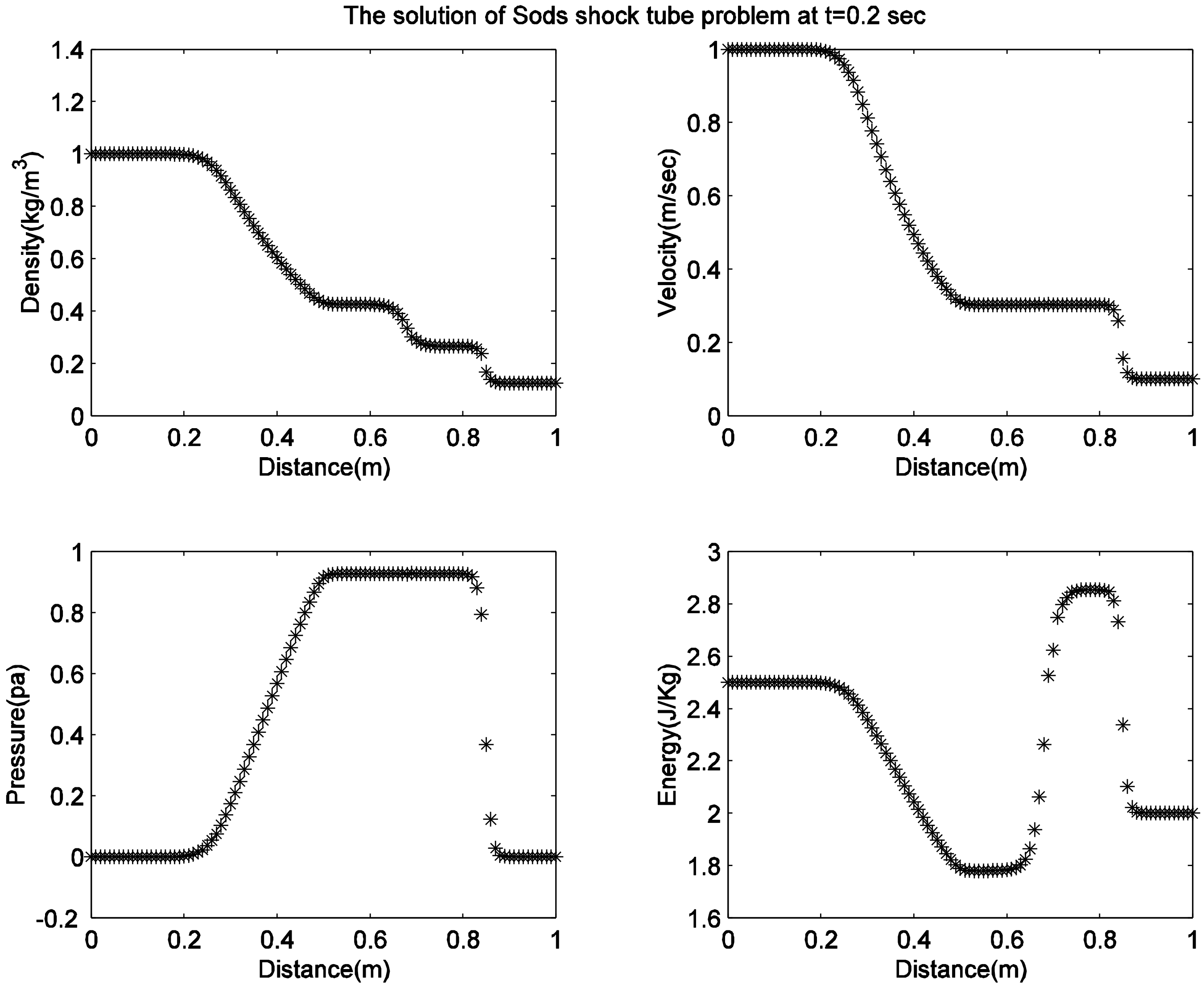
$$\sum_{e=1}^{nel}\int_{\Omega_n^e} \upsilon^h \hat{\nabla}_\xi W^h \cdot \left[\tilde{A}_o\right] \hat{\nabla}_\xi V^h \, d\Omega$$

$$\nu^h_{HM} = \max\left(0,\ \left[\frac{(LY-S)\cdot\tilde{A}_0^{-1}(LY-S)}{g^{ij}Y_{,i}\cdot A_0^{DC}Y_{,j}}\right]^{\frac{1}{2}} - \left[\frac{(LY-S)\cdot\tilde{\tau}(LY-S)}{g^{ij}Y_{,i}\cdot A_0^{DC}Y_{,j}}\right]\right)$$

DC Operator

- It should act in the direction of the gradient
- for consistency, it should be proportional to the residual
- and for accuracy, it should vanish quickly in smooth regions of the solution

Sod’s Shock Tube Problem



Overview of PHASTA CFD Solver

- Massively parallel MPI Navier-Stokes flow solver
- Models compressible or incompressible, turbulent, unsteady flows
- Finite element discretization in space => Complex geometries
- Accuracy h^{k+1}; k=polynomial order, e.g., k=1 linear, 2nd order accurate
- Minimally dissipative stabilization key to scale-resolving turbulence
- Fully implicit in time => Δt governed by the physics
- Mesh adaptivity => - Grid matches physical scale
  - Anisotropic (boundary and shear layers)
- Variety of scale resolving (DNS, LES), turbulence models (RANS), and hybrids (DES, DDES, IDDES).
- Parallel scaling to 768K cores and 3.1M MPI instances
- Applied to a number of flows that demonstrate the progress but additional need for improved scale-resolving simulations

Computational Work Kernels

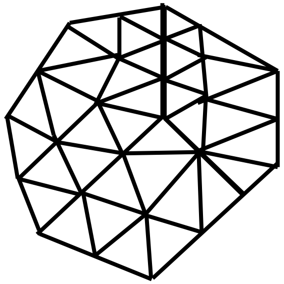
- Implicit non-linear FEM solver with two phases of computation:
  - **Equation formation** (*Eqn. form.*) – depends on *elements*

PDE/strong form – $\mathscr{L}Y = \mathscr{S}$

Stab FEM – $\int_\Omega (\cdot)^h d\Omega + \int_\Gamma (\cdot)^h d\Gamma$

Quadrature – $\sum_{qp}^{vol} c_{qp}(.)^h + \sum_{qp}^{bdy} d_{qp}(.)^h$

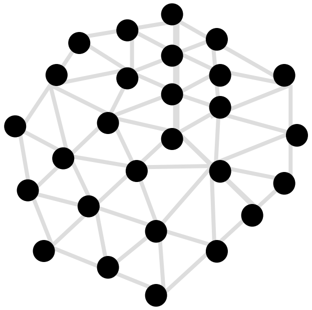
Assembly – $\mathbf{A}\mathbf{x} = \mathbf{b}$



  - **Equation solution** (*Eqn. sol.*) – depends on *degrees-of-freedom* (*dofs*):

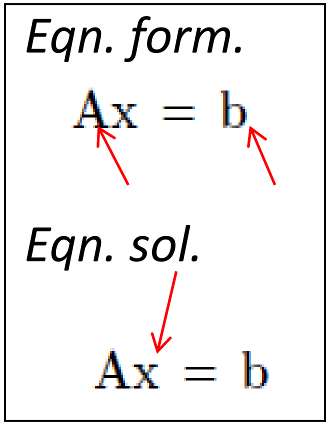
Krylov Iterative solver (GMRES)

```
$\mathbf{A}\mathbf{x} = \mathbf{b}$
$\mathbf{p} = \mathbf{b}$
while
    $\boxed{\mathbf{q} = \mathbf{A}\mathbf{p}}$
    Orthonormalize $\mathbf{q}$
```



Current approach – Parallelization

- Parallel strategy:
  - Domain decomposition approach based on the elements
  - Both compute stages operate off the same mesh partition
  - Partition defines inter-part relations (part-to-part comm.)

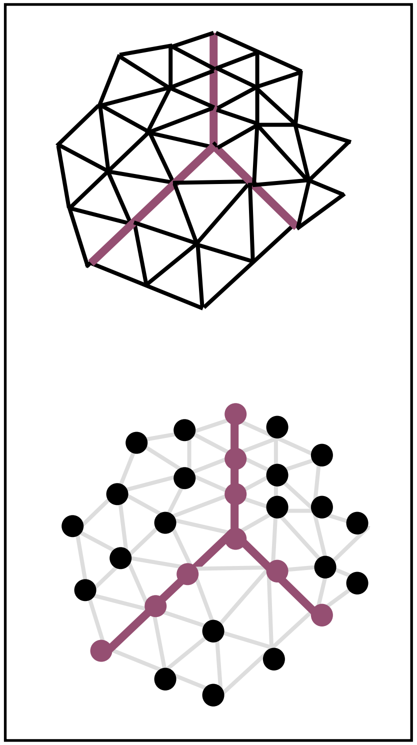


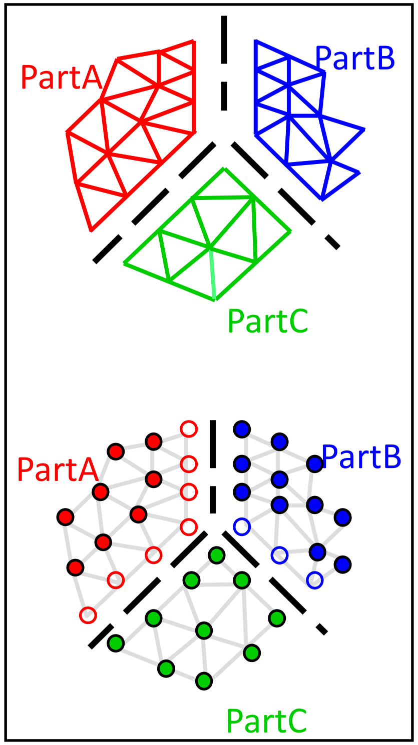
Eqn. form.

$$Ax = b$$

Eqn. sol.

$$Ax = b$$





Locally, *incomplete* values (in ***b***, ***A***, ***q***, etc.) for shared *dofs.*

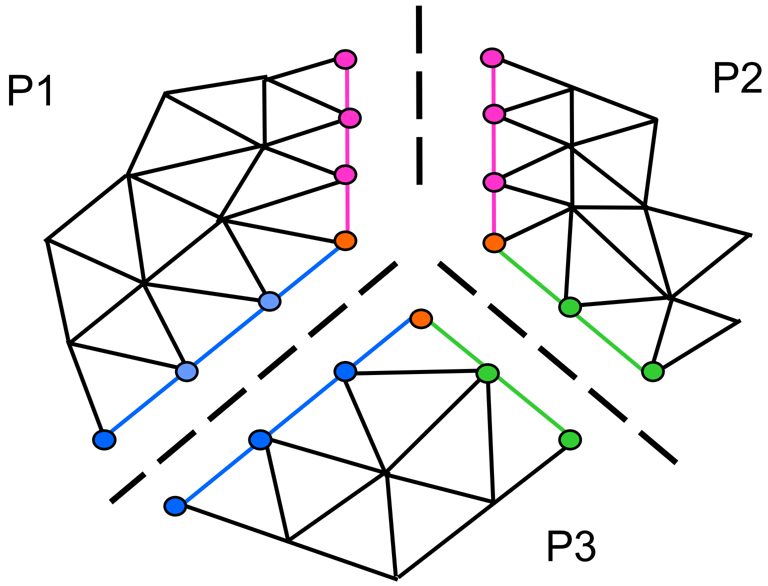
Apply communications to *complete* values/entries (in ***b***, ***q*** only NOT full A)

$\textcircled{b}$ during *Eqn. form.*

$\textcircled{q}$ during *Eqn. sol.*

PHASTA Flow Solver Parallel Paradigm

- **Equation formation:**
  - O(40) peer-to-peer non-blocking comms
  - Overlapping comms with comp
  - Scales well on many machines
- **Implicit, iterative equation solution:**
  - Each Krylov vector is:
    - q=Ap (matrix-vector product)
    - Same peer-to-peer comm of q PLUS
    - Orthonormalize against prior vectors
    - Dot product and norm computations =>



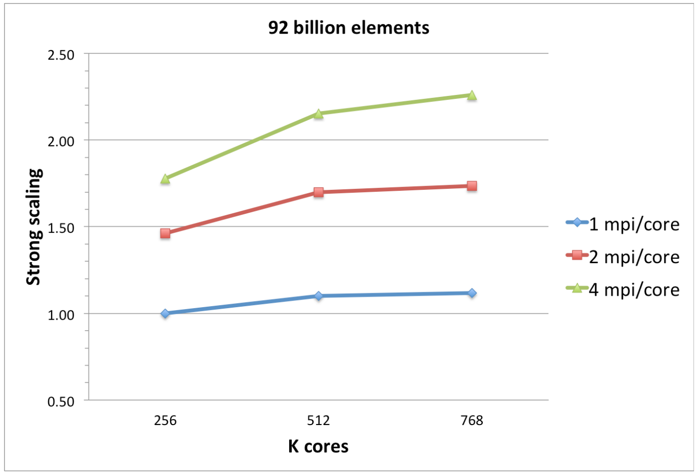
MPI_Allreduce (1 to 200 doubles)

PHASTA Scaling on MIRA

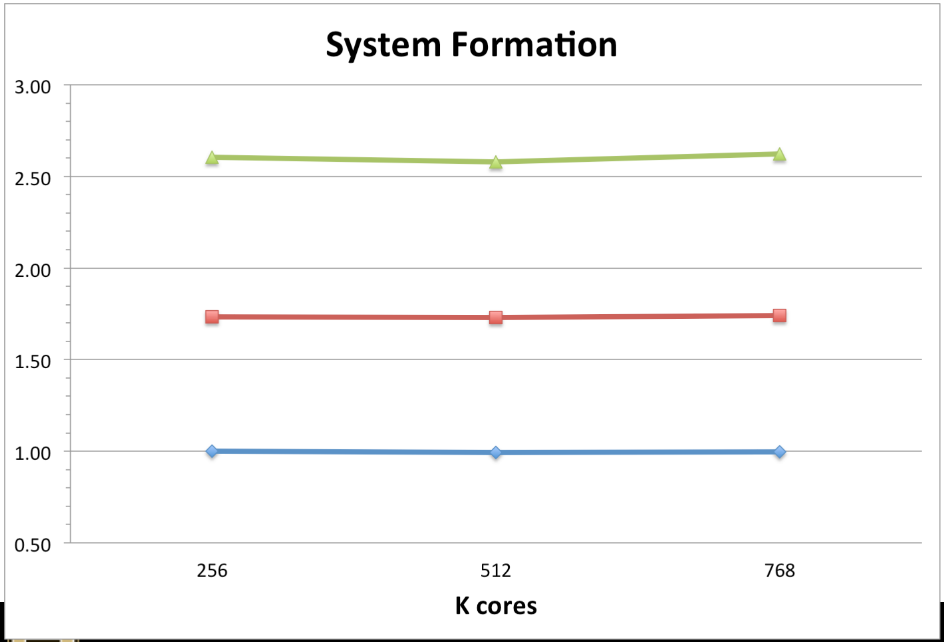
- **92B** element mesh (with resp. 8B and 64B *dofs*)
  - Strong scaling: from 256K to **768K** cores (Mira)
  - 1, 2 and **4** MPI processes per core
  - **3,145,728** MPI processes on full Mira system

- Scaling factor defined as ($t_{base} \times ncore_{base}$)/($t \times ncore$):
  - For 1 MPI process per core: scaling factor 1 means perfect scaling
  - For 2 or 4 MPI processes per core: scaling factor means acceleration wrt. 1 MPI process per core
  - Baseline: 256K cores for 92B elements

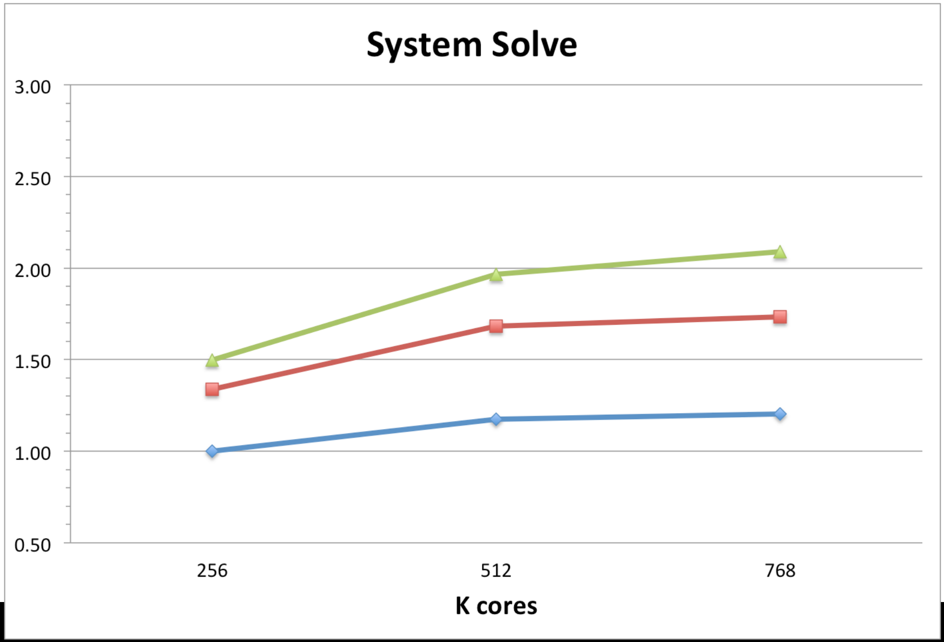
PHASTA Scaling on MIRA



92 billion elements

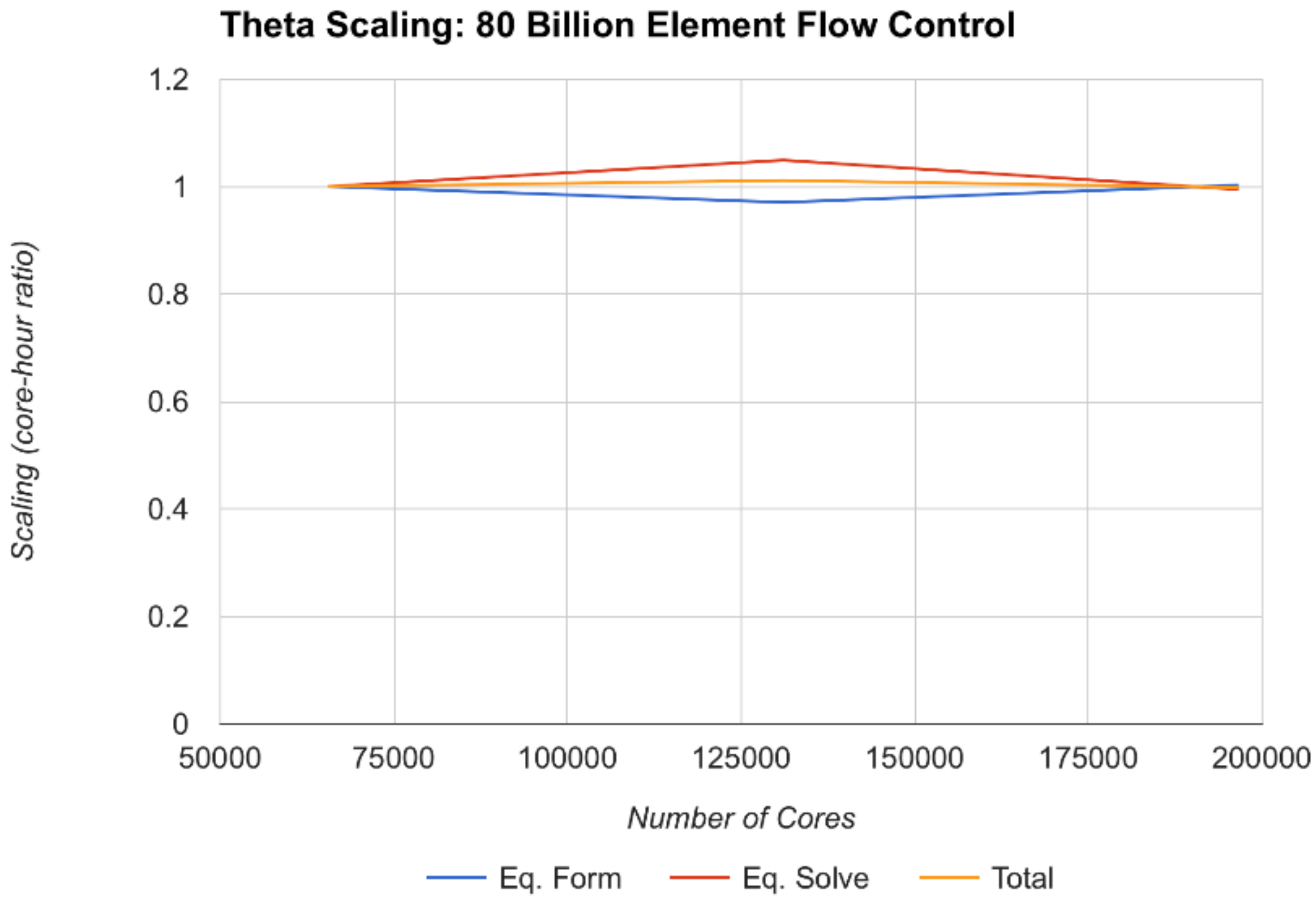


System Formation

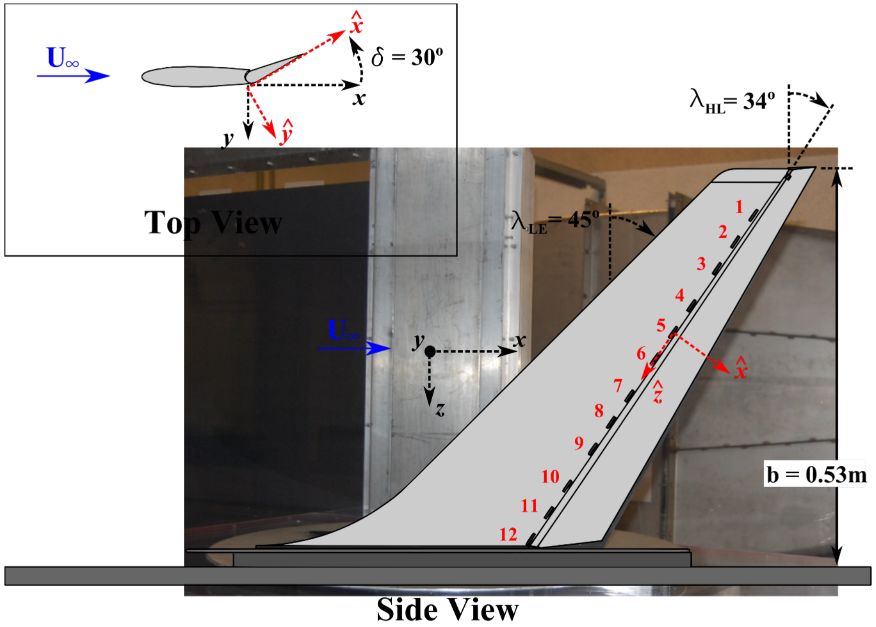


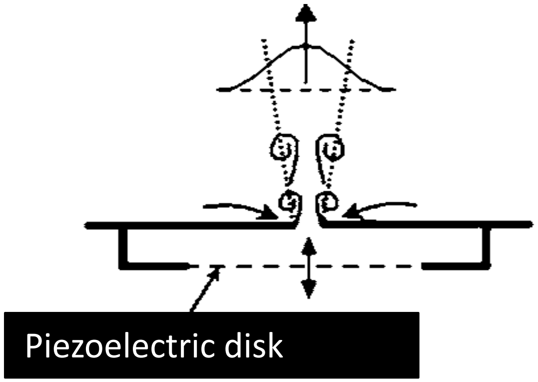
System Solve

PHASTA Scaling on Theta



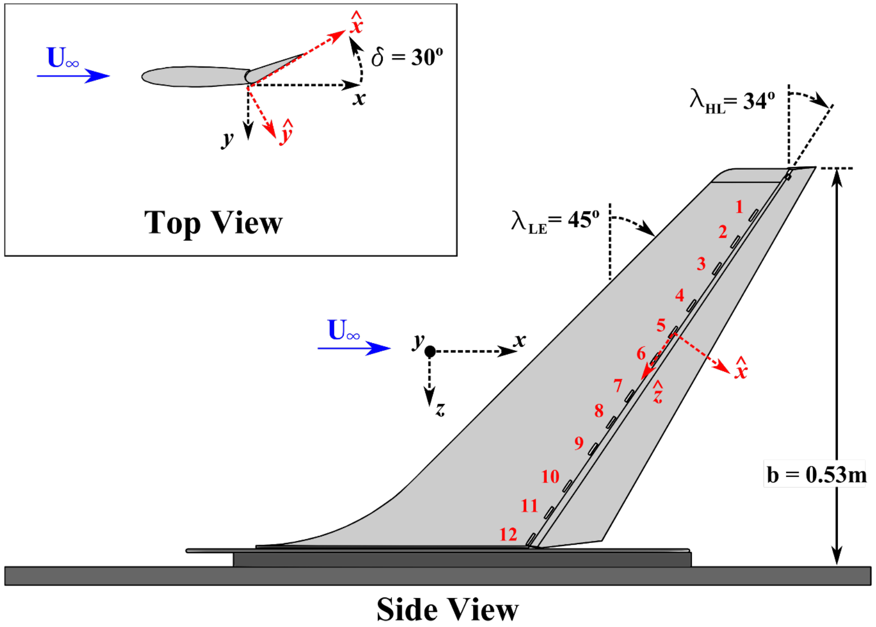
Tunnel-Scale Model

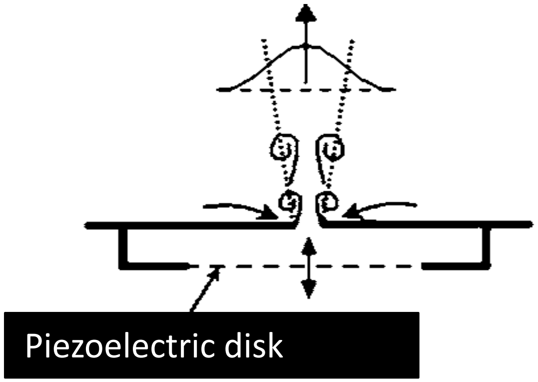




Sketch of a synthetic jet

Tunnel-Scale Model

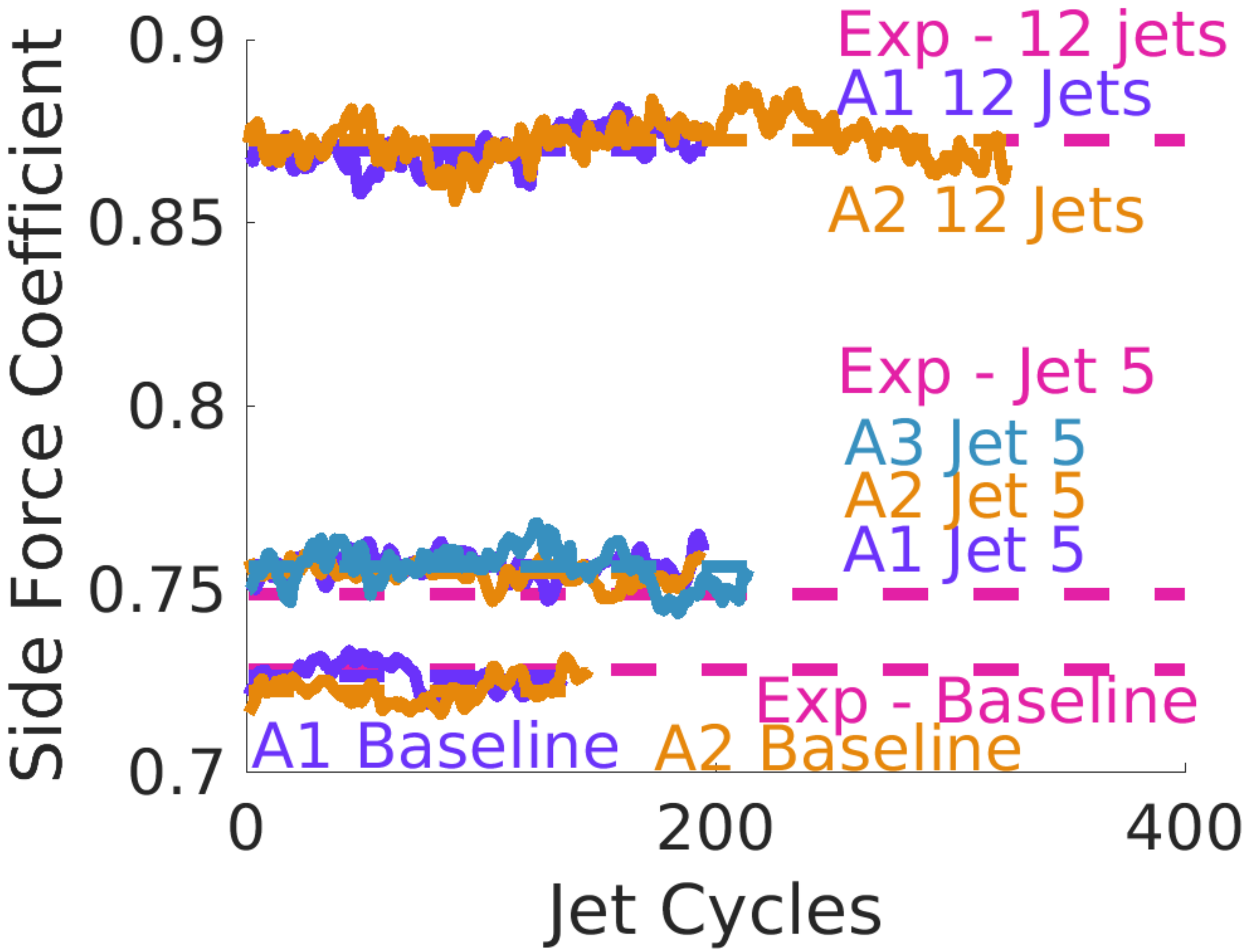




Sketch of a synthetic jet

Jet 5 Blowing: Q isosurface colored by speed

Force Coefficient: DDES Compared to Experiment



Grids for Scale Resolving Methods

DNS:

- Mesh size increases as $\mathcal{O}\left({Re_\theta}^{10/4}\right)$

LES$_{\text{DNS}}$:

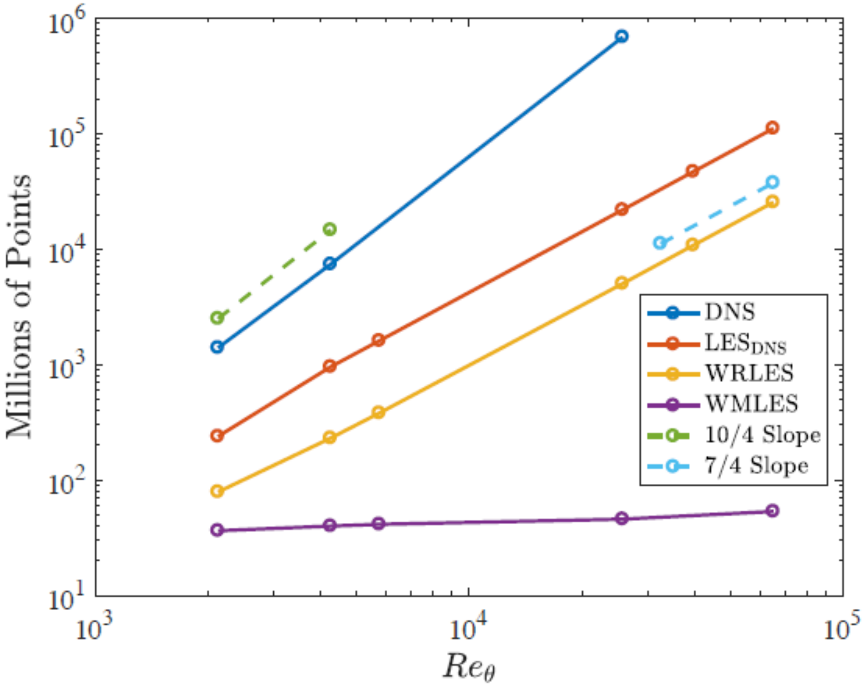
- Mesh size increases as $\mathcal{O}\left({Re_\theta}^{7/4}\right)$
- Wall normal spacing is allowed to grow geometrically in the outer layer only until $\delta/20 - \delta/64$
- Same Δx and Δz grid sizes as DNS below y^+=200

WRLES:

- Mesh size increases as $\mathcal{O}\left({Re_\theta}^{7/4}\right)$
- Wall normal spacing is allowed to grow geometrically from the wall until $\delta/20 - \delta/64$
- Typical LES Δx and Δz grid sizes

WMLES:

- Wall stress or hybrid/RANS (larger LES Δx and Δz grid sizes)



Mach 3 Direct Numerical Simulation on 4K cores