An efficient preconditioner for a monolithic high-order fluid-structure interaction solver

Matteo Franciolini
NPP Fellow, NASA Ames Research Center

Anirban Garai
Science and Technology Corporation, NASA Ames Research Center

Scott Murman
NASA Ames Research Center

Background

- Developing an implicit space-time high-order multi-physics solver (*eddy)*

- Requires solution of nonlinear system to machine precision every steady/ unsteady step
  - Matrix-Free, Newton-Krylov algorithm

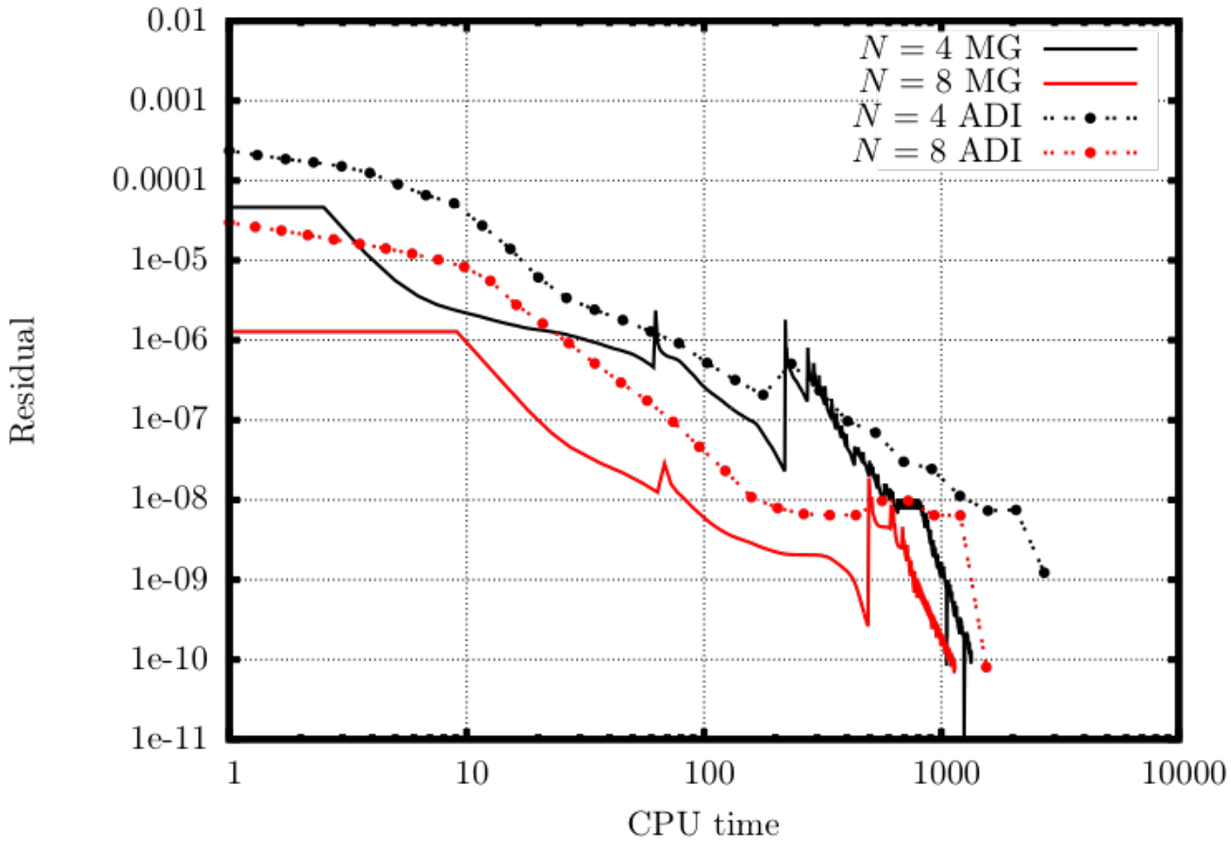
- Standard preconditioning methods developed for 2nd-order solvers, based upon a full Jacobian approximation, scale too poorly with increasing polynomial order - both memory and computational cost - to be utilized

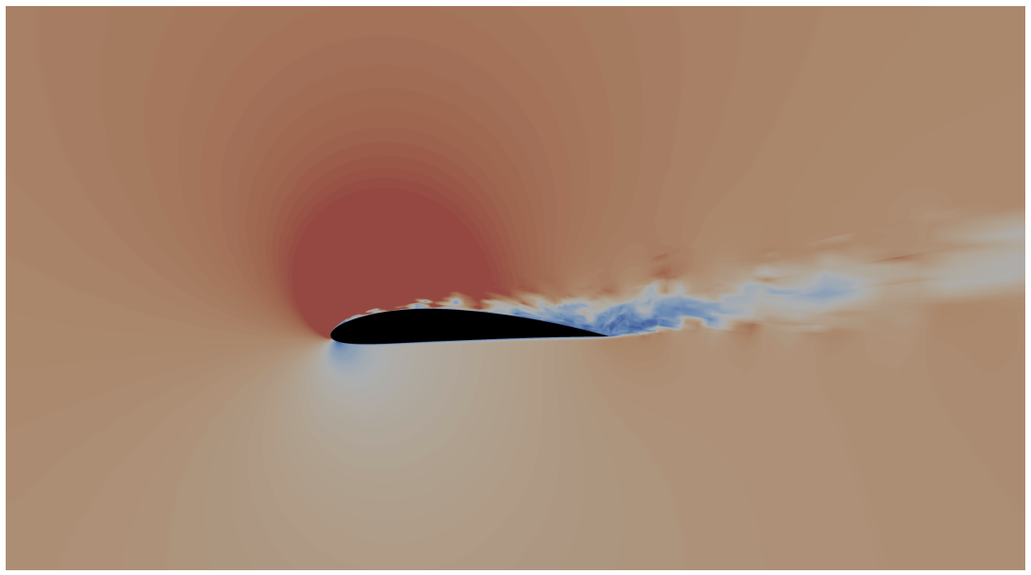
- Developed a strategy based upon a hierarchical approach to preconditioning for high-order discretizations
  - Leverage standard methods at low order (N=2)
  - Develop efficient preconditioners at high order (N > 2)
  - Implement in a spectral multigrid linear preconditioner

Navier-Stokes

- Example for the Navier-Stokes equations (SciTech 2020)

- Block - ILU0 at N=2
- Tensor-product ADI preconditioner at N > 2 (Diosady & Murman, JCP)
- Solution of linear adjoint system for NACA 4412 airfoil at Re=50k
- Reduces computational time by factor of 2, even though very few viscous-dominated elements





Outline

- Extend hierarchical strategy to systems for mesh deformation and structural solver
- Unlike N-S equations, these use an entity-based continuous-Galerkin formulation
- Still tensor-product based, and efficient at arbitrary order
- These building blocks can then be combined with fluid and other solvers in a general monolithic framework by considering couplings between different physical systems
  - Provides an efficient, stable, arbitrary-order multi-physics solver

Mesh deformation/structural solver preconditioning

- Degrees of Freedom arranged in an entity-based fashion
  - Linear system reflects entity-by-entity substructure
  - Gather-scatter DoF to assemble residuals

$$\begin{bmatrix} \mathbf{A}_{v,v} & \mathbf{A}_{v,e} & \mathbf{A}_{v,f} & \mathbf{A}_{v,\kappa} \\ \mathbf{A}_{e,v} & \mathbf{A}_{e,e} & \mathbf{A}_{e,f} & \mathbf{A}_{e,\kappa} \\ \mathbf{A}_{f,v} & \mathbf{A}_{f,e} & \mathbf{A}_{f,f} & \mathbf{A}_{f,\kappa} \\ \mathbf{A}_{\kappa,v} & \mathbf{A}_{\kappa,e} & \mathbf{A}_{\kappa,f} & \mathbf{A}_{\kappa,\kappa} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_v \\ \mathbf{U}_e \\ \mathbf{U}_f \\ \mathbf{U}_\kappa \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_v \\ \mathbf{R}_e \\ \mathbf{R}_f \\ \mathbf{R}_\kappa \end{Bmatrix}$$

- We can't assemble the full Jacobian explicitly
  - BDDC, FETI, require explicit assembly of $\mathbf{A}_{i,j}$
  - Not clear how to do that efficiently

- Direct factorization of the elemental entities allows to compute only a portion of the matrix

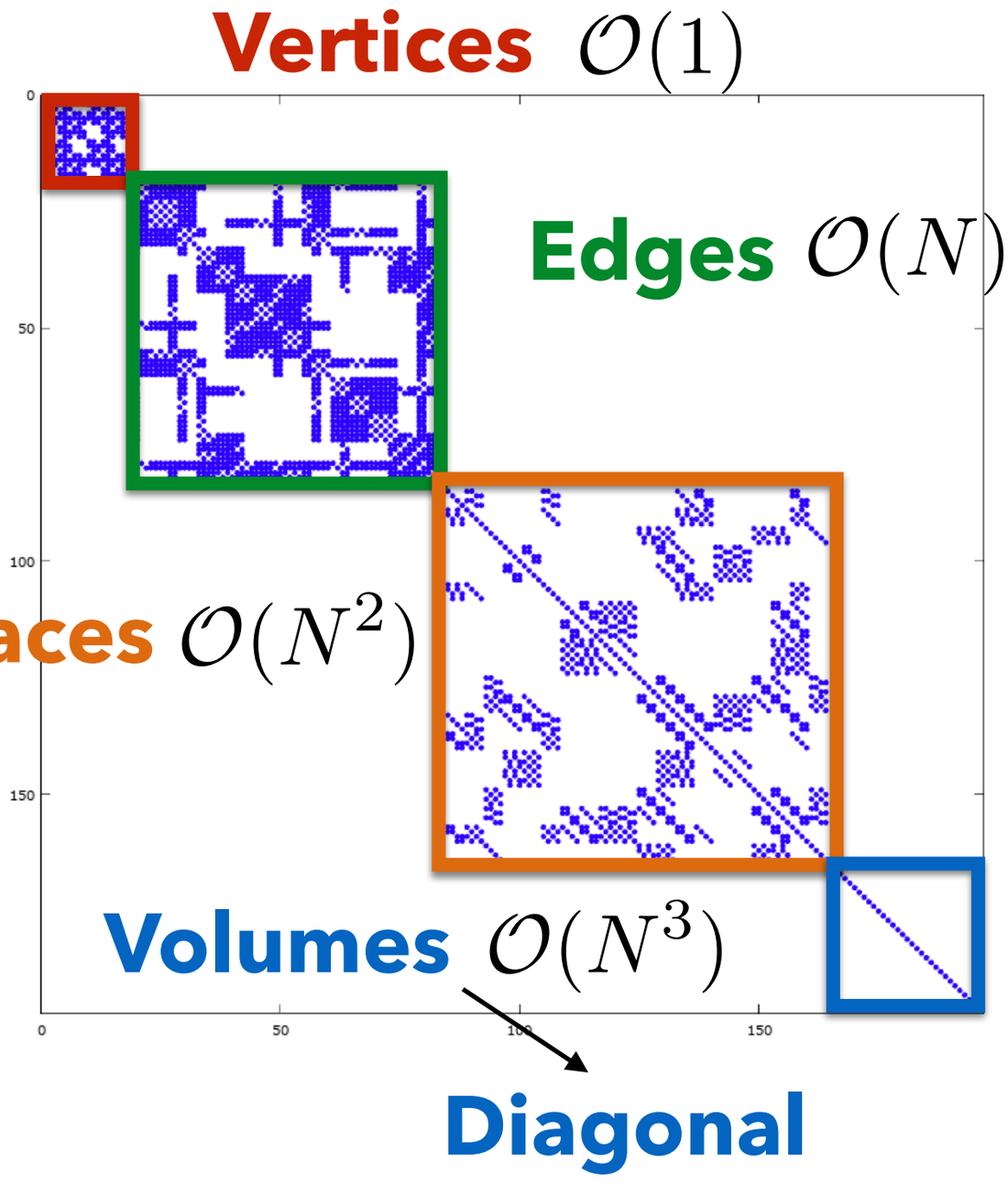
Mesh deformation/structural solver preconditioning

- Preconditioning strategy based on factorization of entity-wise Jacobian blocks
  - Diagonal approximation of volume terms for memory-lean implementation

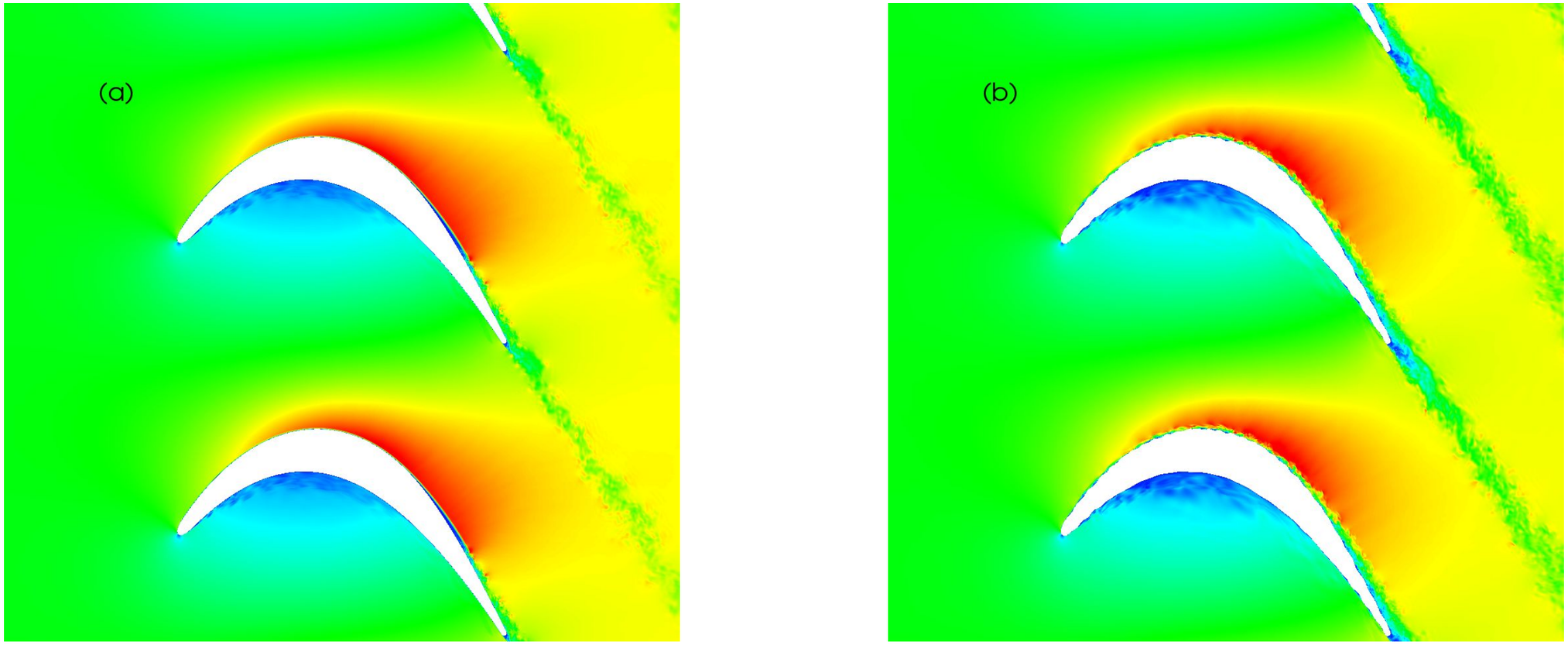
- Lower dimensional entities do the majority of the work for very low computational cost

- 4 small factorizations
  - one is just a simple diagonal inversion

- Further reduce the computational cost just by considering the block-diagonal of Vert, Edges, Faces.

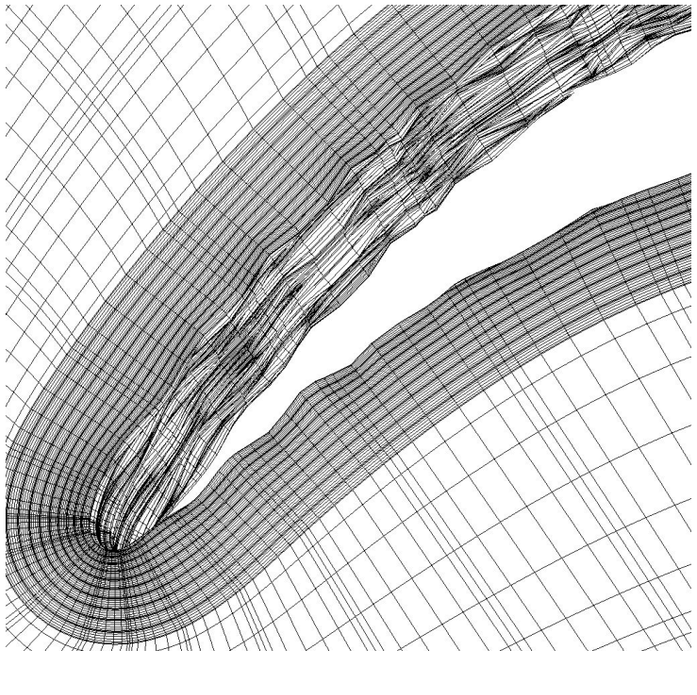


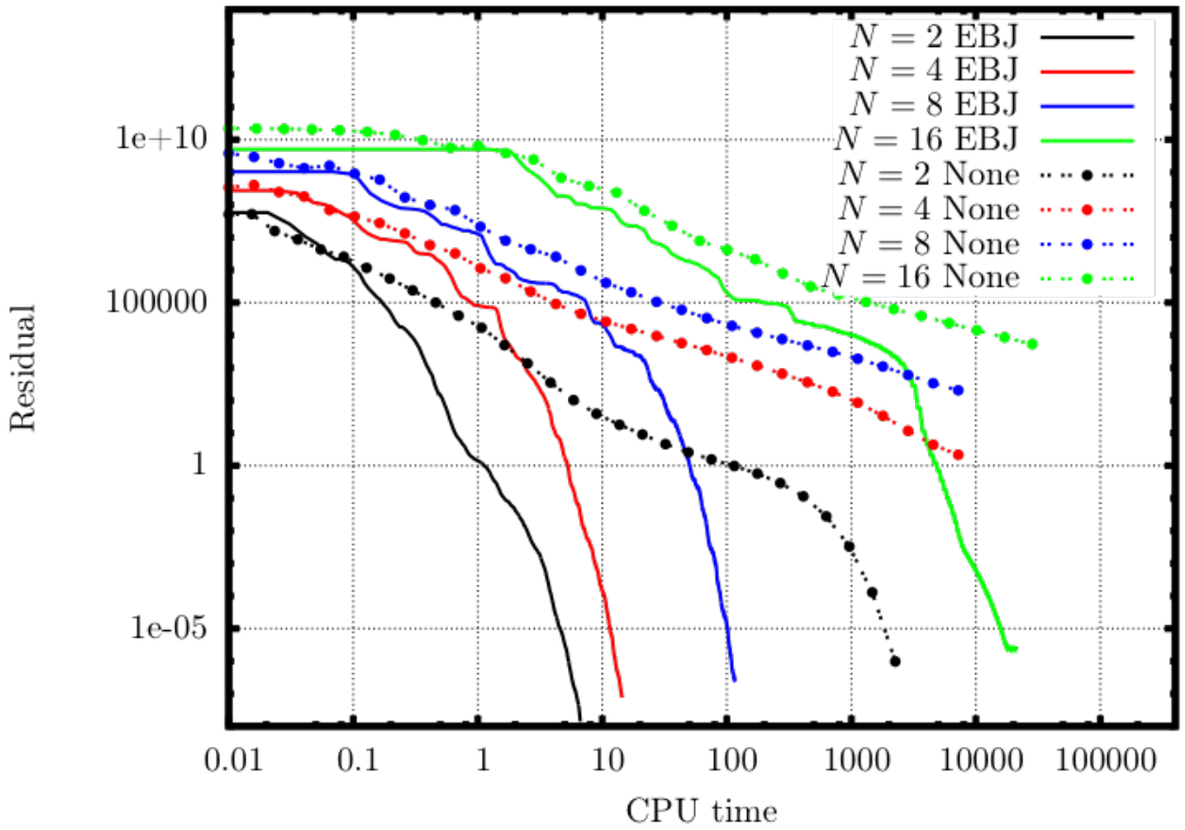
Mesh deformation (structural solver)



- Flow around LPT Airfoil highly dependent on surface roughness
- Presence of modeled surface roughness results in different transition behavior and size of separation bubble
- Better agreement with experiments
  - Garai et al. (2018 IGTI)
- Obtaining deformed mesh is computationally **expensive at high-order**

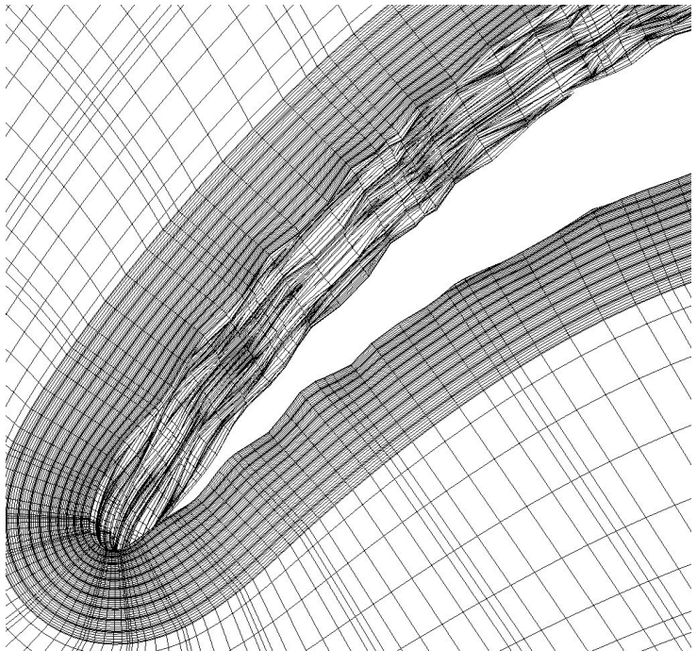
Mesh deformation/structural solver preconditioning

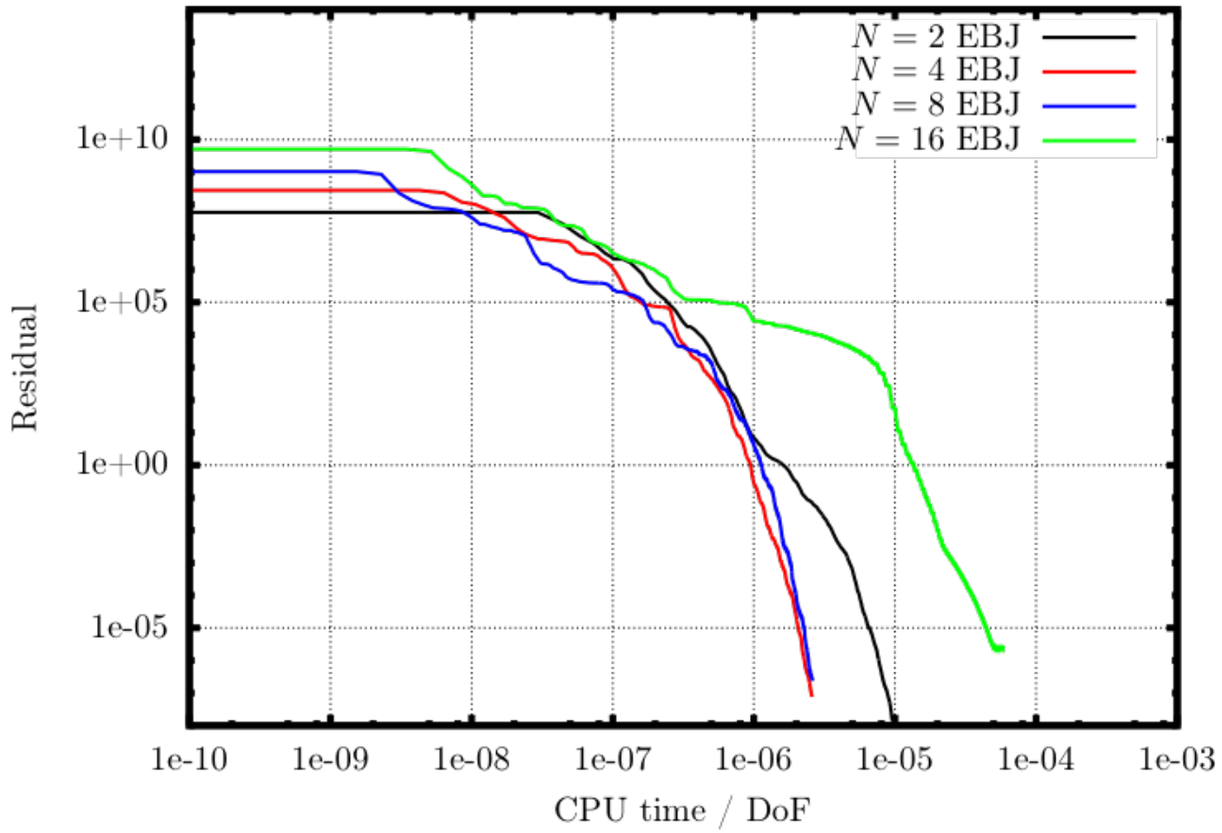




- Turbine blade test case with roughness
  - deformation applied to mesh nodal points with increasing N
  - Up to **200X** improvements w.r.t. un-preconditioned case
  - Parallel implementation shows linear scaling with number of cores
  - Constant cost per Dof

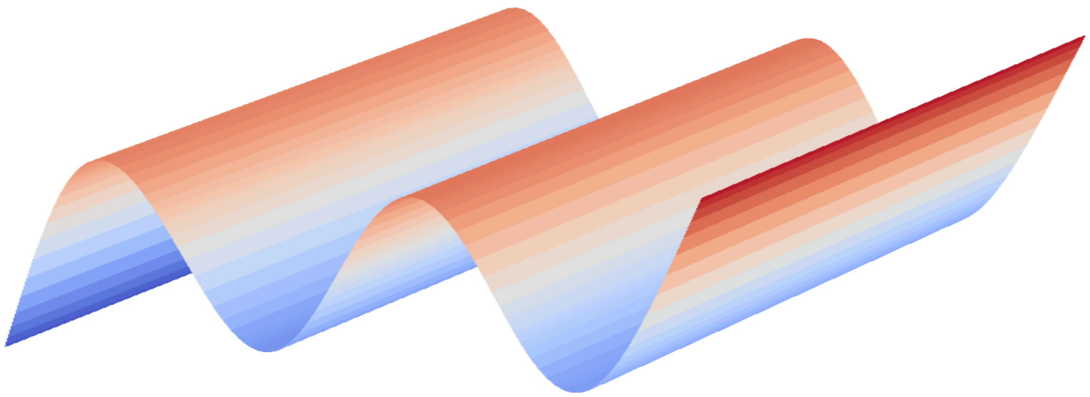
Mesh deformation/structural solver preconditioning

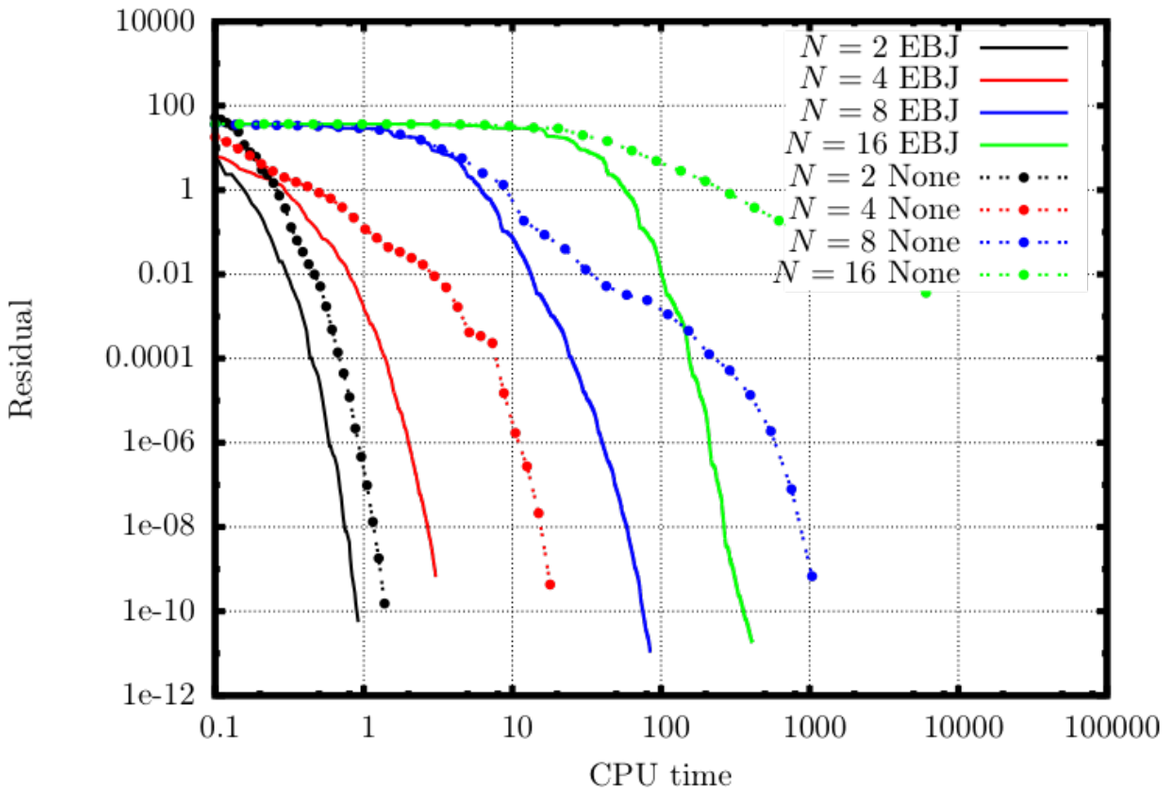




- Turbine blade test case with roughness
  - deformation applied to mesh nodal points with increasing N
  - Up to **200X** improvements w.r.t. un-preconditioned case
  - Parallel implementation shows linear scaling with number of cores
  - Constant cost per Dof

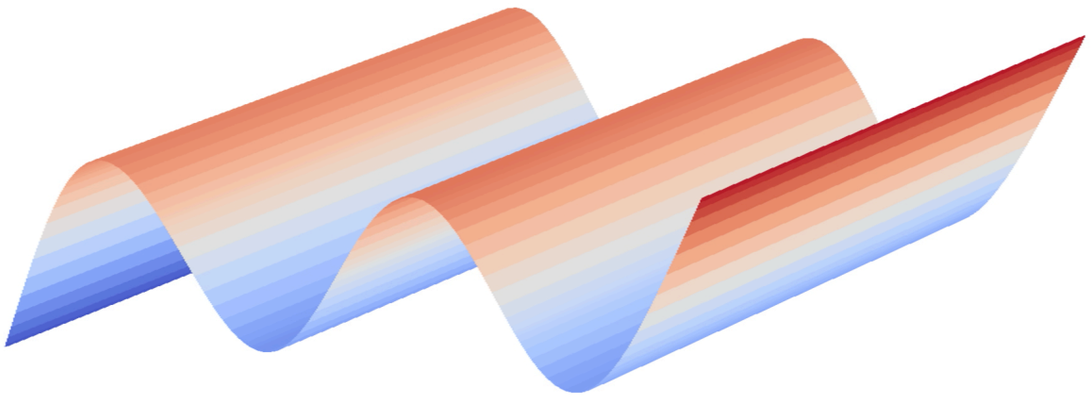
Mesh deformation/structural solver preconditioning

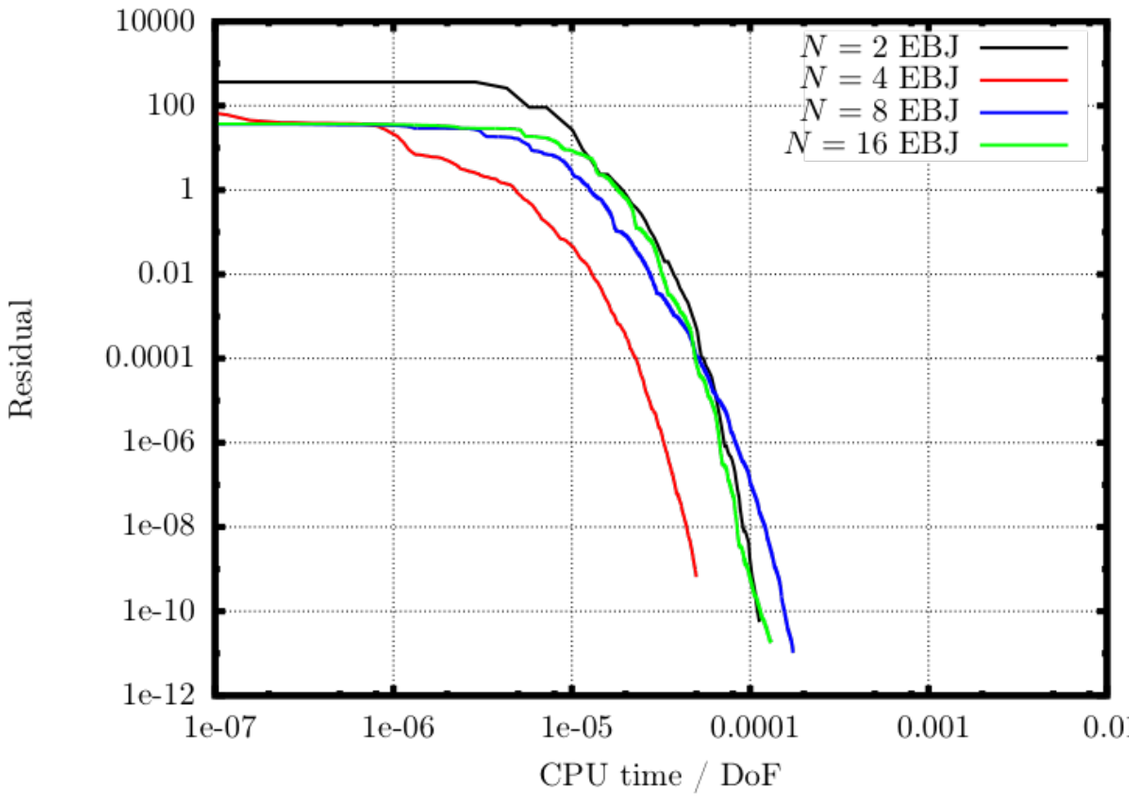




- Free-Free plate vibration in phase with natural frequency
- 8x8 mesh elements, spatial order: 2,4,8,16; temporal order: 4
- Entity Block-Jacobi enables implicit space-time solutions within seconds for very high-order cases
- Constant cost per DoF

Mesh deformation/structural solver preconditioning

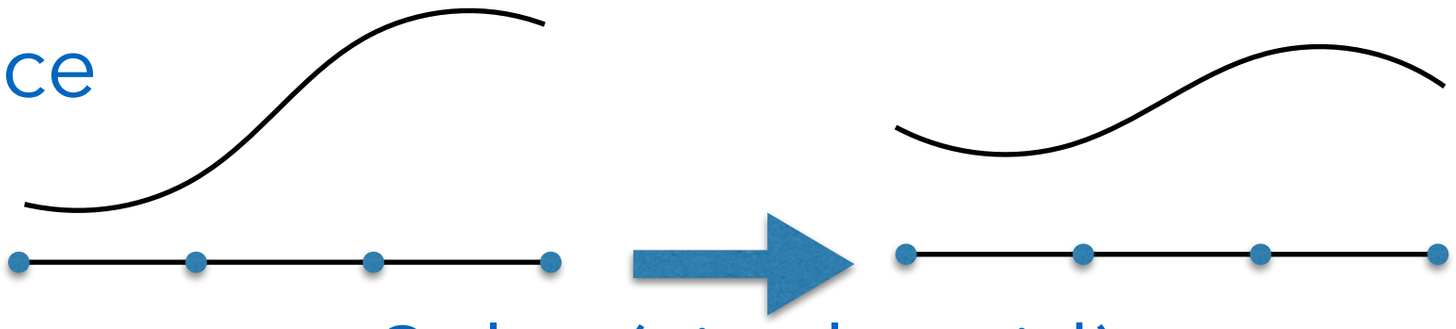




- Free-Free plate vibration in phase with natural frequency
- 8x8 mesh elements, spatial order: 2,4,8,16; temporal order: 4
- Entity Block-Jacobi enables implicit space-time solutions within seconds for very high-order cases
- Constant cost per DoF

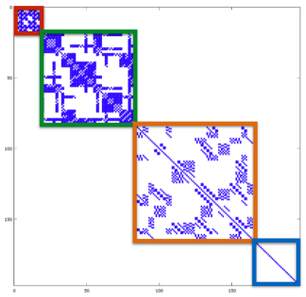
Spectral multigrid method (structural solver)

Solution space
(high order)



Solve (single grid)

EBJ



Spectral multigrid method (structural solver)

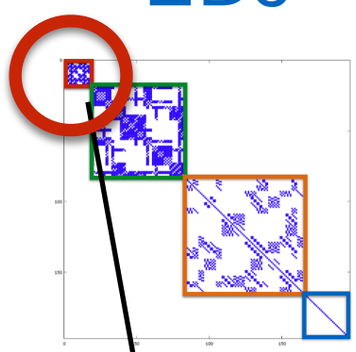
Solution space
(high order)

Solve (single grid)

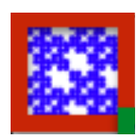
Restrict

Prolong

EBJ



ILU

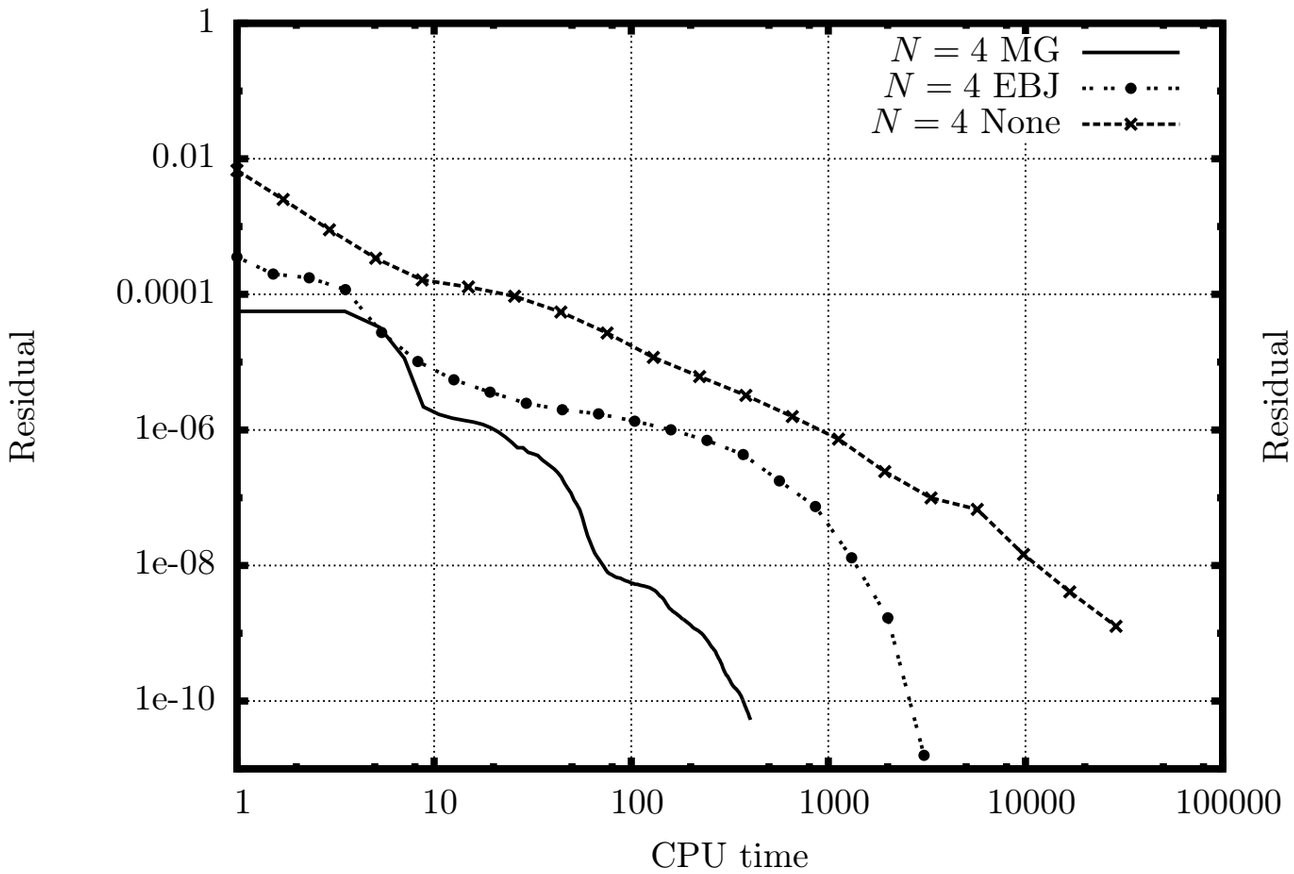


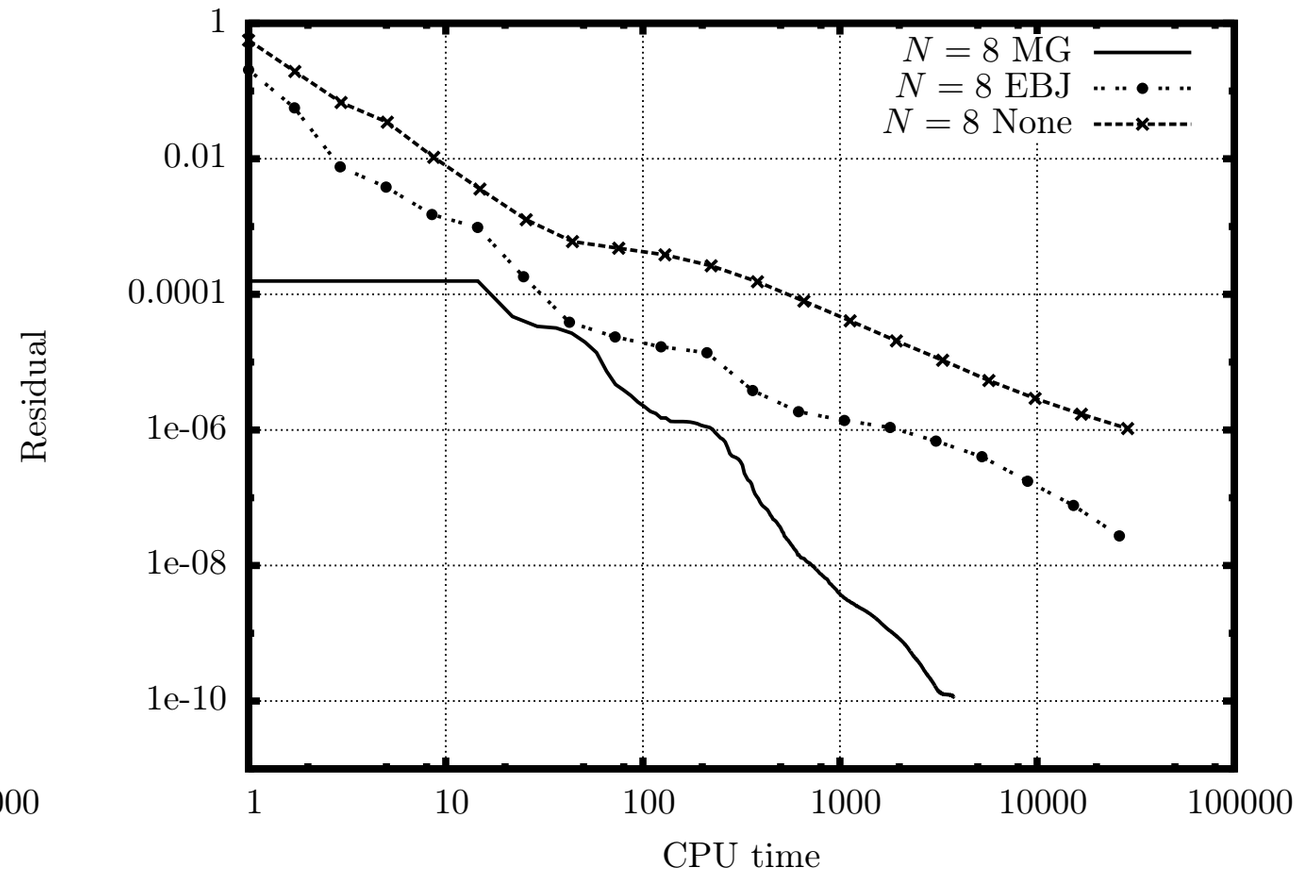
Coarse space
(low order)

Solve (multigrid)

C0 restriction

Multigrid performance (structural solver)





- NACA 0012 - heaving mesh deformation (HOW)
- 4096 stretched mesh elements with fixed farfield
  - Single-grid EBJ preconditioner struggles
- 10X further improvement in time-to-solution with multigrid

Summary

- Developed a suite of preconditioning strategies for different physics/ discretizations
  - Required to enable FSI simulations

- Implementation of preconditioning operator for structural solver
  - enabling use of mesh deformation as high order grid generation tool
  - enabling use of a linear shell structural solver for parachute fabric

- Improved solver robustness using multigrid preconditioning with block factorization of coarse space Jacobian matrices
  - cost reduction for ill-conditioned cases (mesh deformations, diffusion, adjoints)
  - enabling technology for solving stiff source terms (RANS, hypersonics)