Scalable Algorithms for High-Order Solutions on Common Research Model (CR1)

Behzad R. Ahrabi, Michael J. Brazell, and
Dimitri J. Mavriplis

University of Wyoming

5th International Workshop on High-Order CFD Methods
AIAA SciTech
January 6, 2018

Introduction

- Finite-element methods (FEM)
  - Compact stencils
  - Clear path to exact linearization
    - Reaching machine convergence is common and expected
  - High number of computation operations per memory fetch
    - More suitable for emerging hardware architectures
  - Seeking to increase accuracy, robustness, and efficiency over more established discretizations
- Standardized test cases (TMR, DPW, HLPW, HiOCFD, …)
  - Code verification
  - Performance assessment (memory consumption and computational cost)
    - Mesh convergence study
    - Linear and nonlinear convergence study (less studied)
- Focus of this talk: Solution strategies and convergence histories
  - More details on mesh convergence results in Marshall Galbraith's talk

Description of Code

HOMA Solver (High-Order Multilevel Adaptive Solver)

- SUPG, RANS, neg-SA, strong and weak implementation of BCs.
- Fully implicit with exact linearization through automatic diff.
- Non-Linear Strategies:
  - Pseudo Transient Continuation (PTC)
  - P-multigrid (PMG) solver based on Full Approximation Scheme (FAS)
- Principal Linear Solver: Flexible GMRes (FGMRes)
  - Built-ins:
    - Local ILU(k)
    - Implicit Line Relaxation (with Double-CFL Strategy)
    - Additive Schwarz (Restrictive)
  - External Packages: PETSc (Used only for comparing with home-developed solvers.)

Description of Code

- References:

  - Ahrabi, B. R. and Mavriplis D. J., "Scalable Solution Strategies for Stabilized Finite-Element Flow Solvers on Unstructured Meshes", 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0517, Dallas, TX, January 2017.

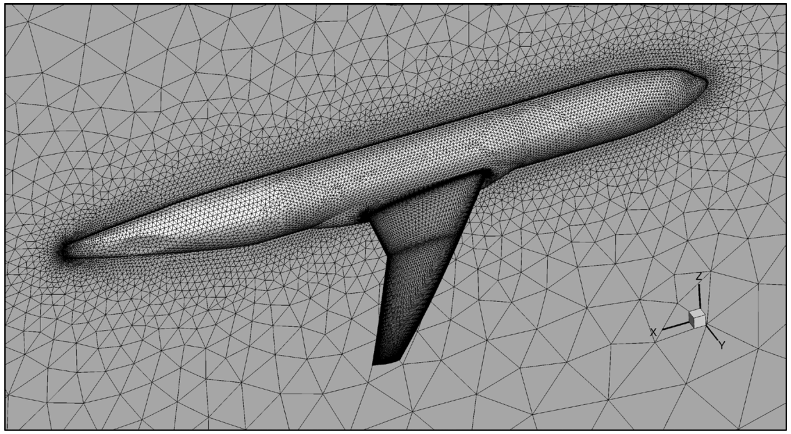
  - Ahrabi, B. R. and Mavriplis D. J., "Scalable Solution Strategies for Stabilized Finite-Element Flow Solvers on Unstructured Meshes, Part II", 23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, AIAA Paper 2017-4275, Denver, CO, June 2017.

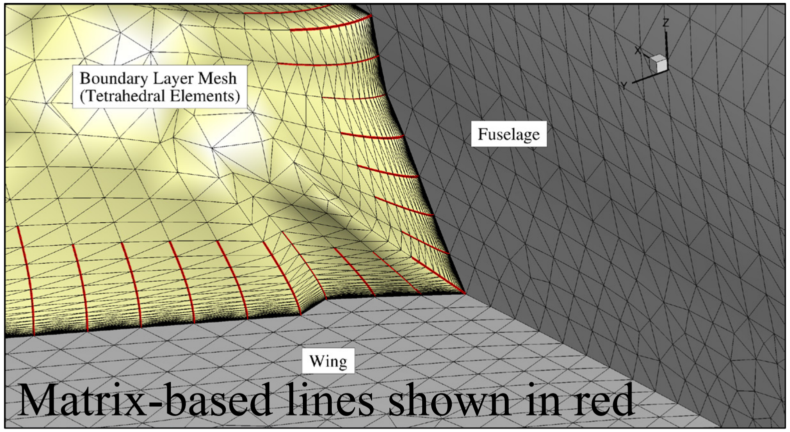
  - Ahrabi, B. R., Brazell, M. J., and Mavriplis D. J., "An Investigation of Continuous and Discontinuous Finite-Element Discretizations on Benchmark 3D Turbulent Flows (Invited)", 2018 AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2018.

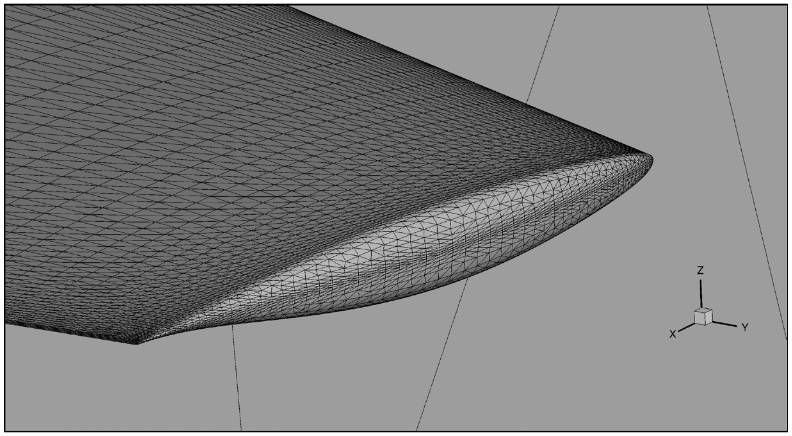
Transonic and Subsonic Turbulent Flow over DPW-6 Configuration

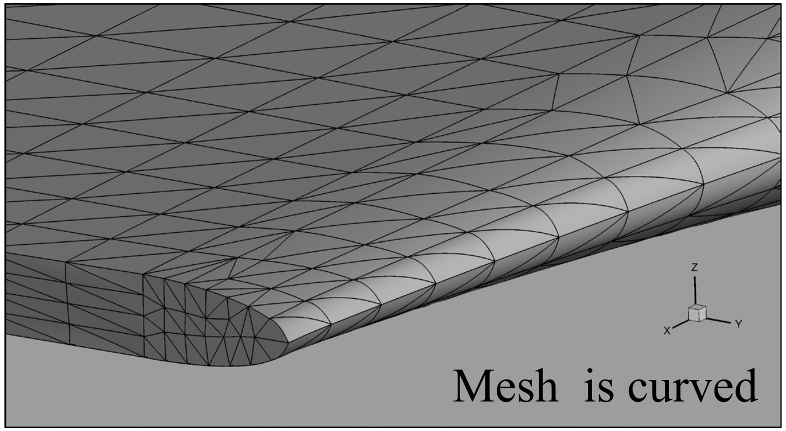
HiOCFD5 Coarse Mesh

433,893 DoFs for P1, and 3,401,021 DoFs for P2



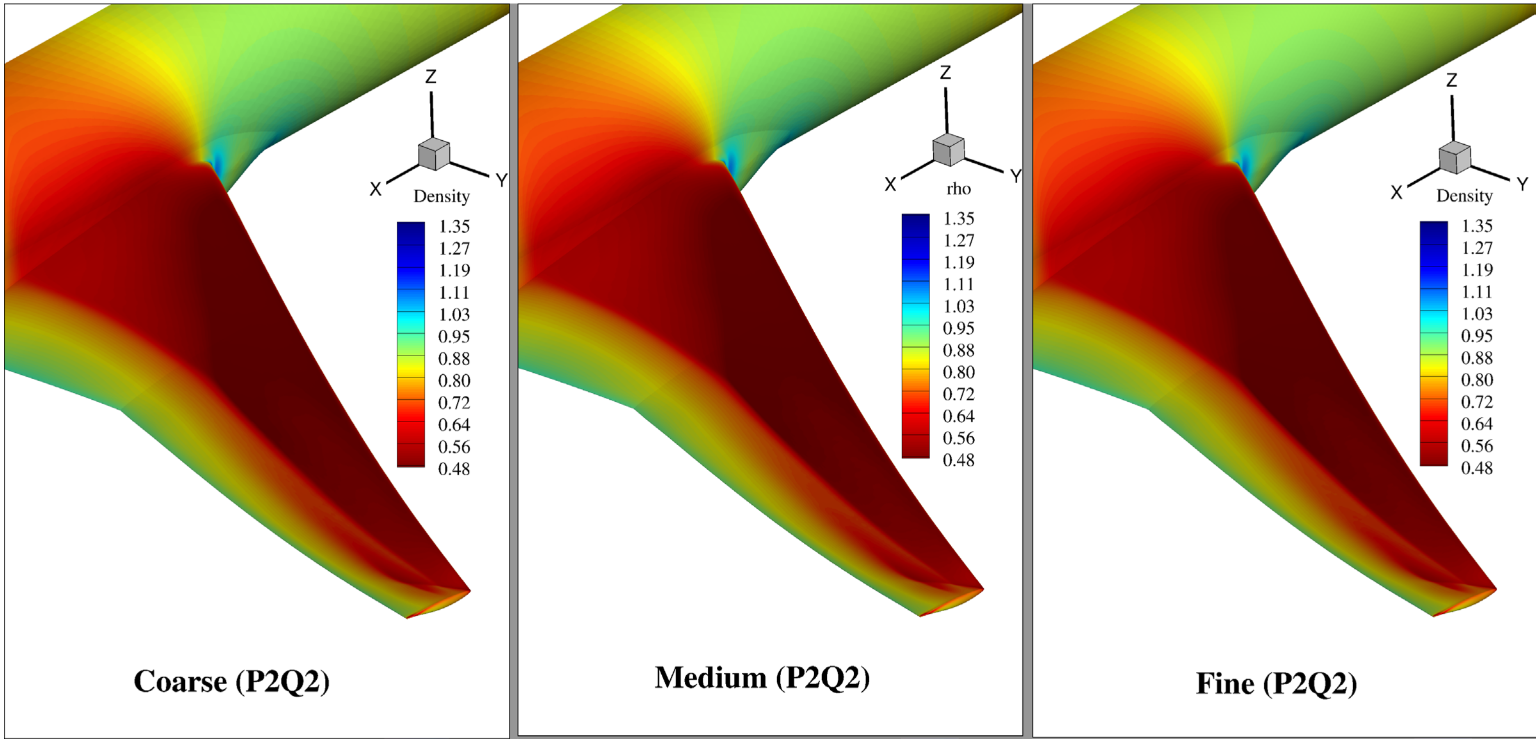






Transonic Turbulent Flow over DPW-6 Configuration
Flow Visualization

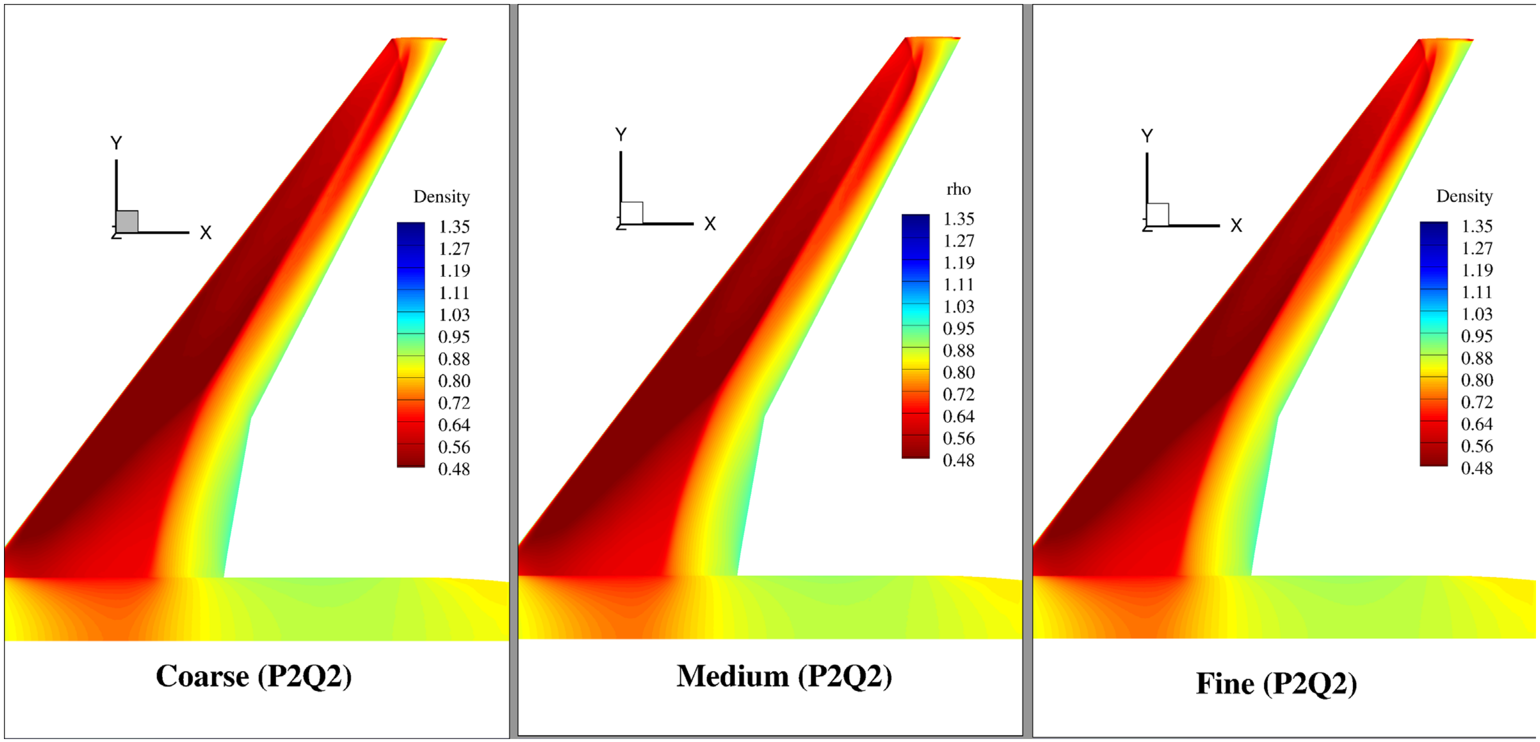
- Free stream conditions: M = 0.85, Re = 5e+6, Alpha = 2.75^{o}



- Good shock resolution even on the coarse mesh

Transonic Turbulent Flow over DPW-6 Configuration
Flow Visualization

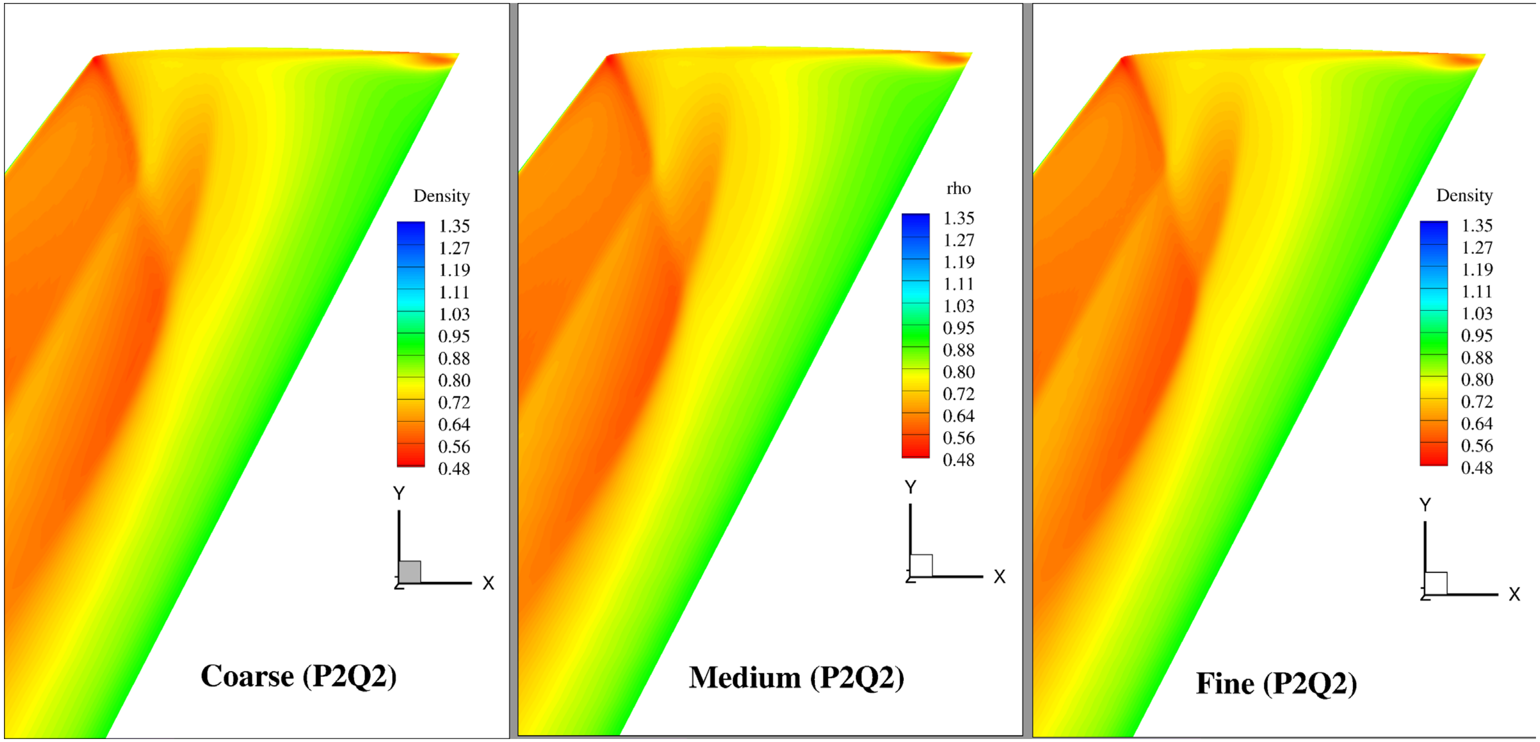
- Free stream conditions: M = 0.85, Re = 5e+6, Alpha = 2.75^o



- Good shock resolution even on the coarse mesh

Transonic Turbulent Flow over DPW-6 Configuration
Flow Visualization

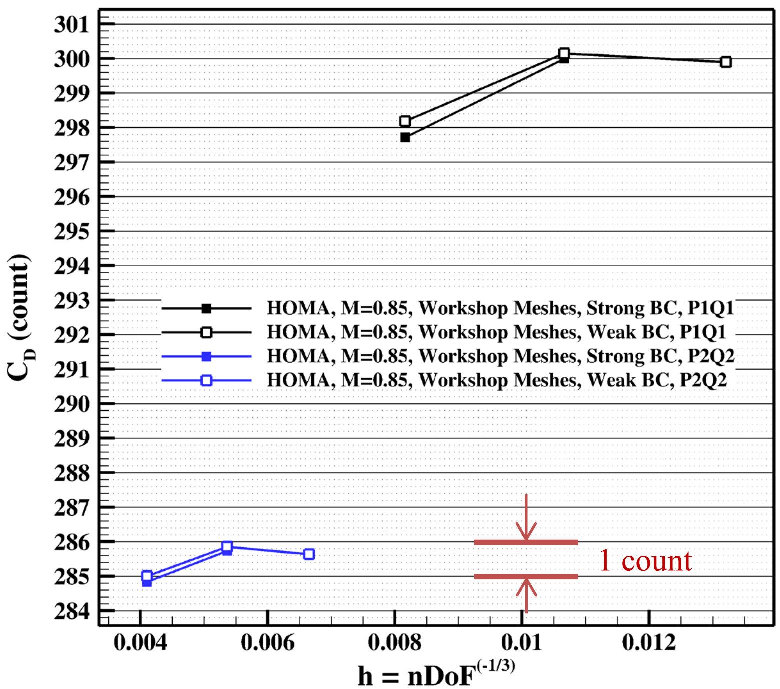
- Free stream conditions: M = 0.85, Re = 5e+6, Alpha = 2.75^{o}

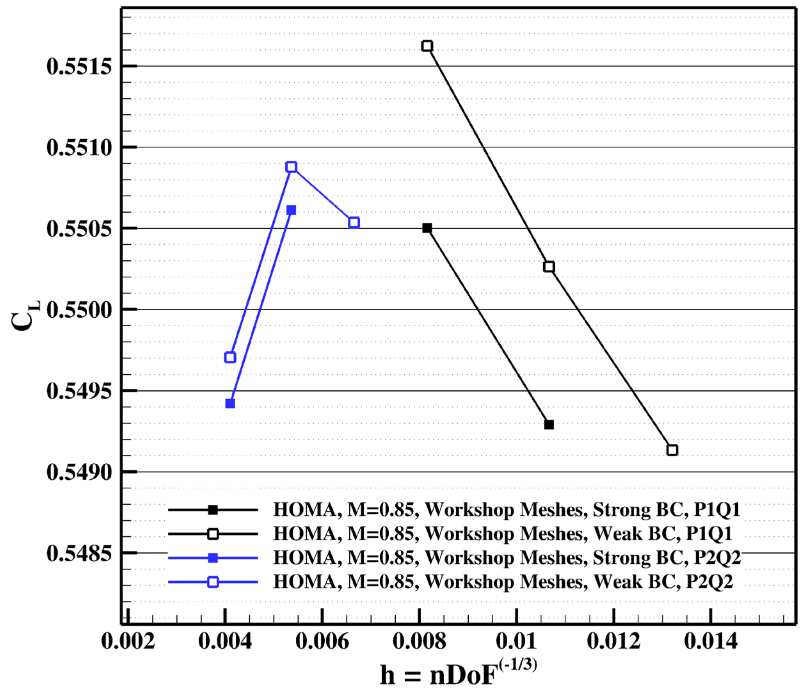


- Good shock resolution even on the coarse mesh

Transonic Turbulent Flow over DPW-6 Configuration
Aerodynamic Forces

- Free stream conditions: M = 0.85, Re = 5e+6, Alpha = 2.75°

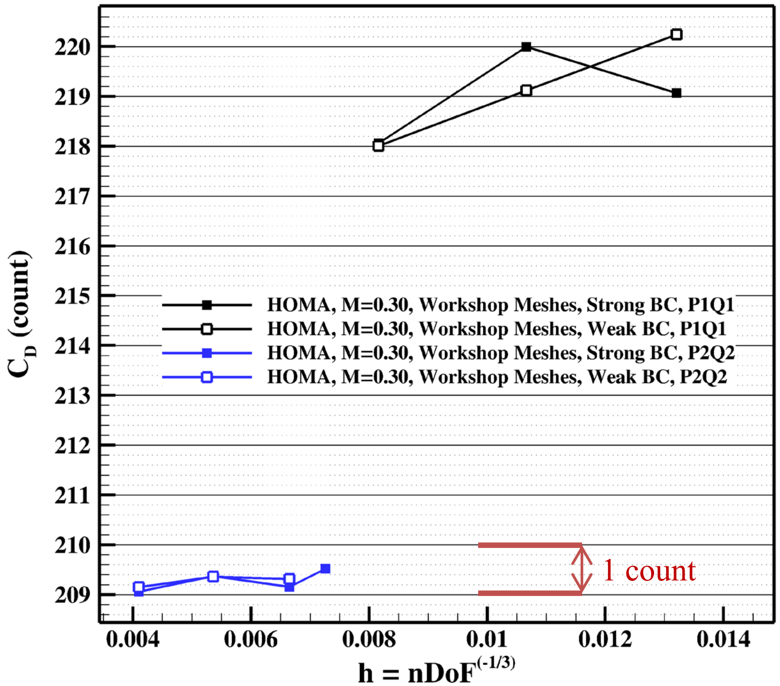


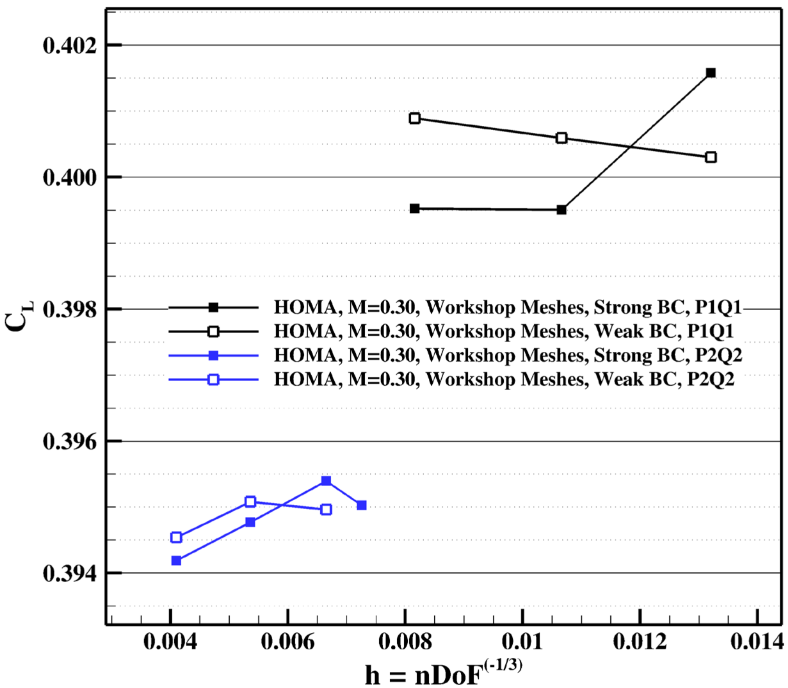


- Meshes are coarse for these simulations
  - Large difference between P1 and P2 solutions

Subsonic Turbulent Flow over DPW-6 Configuration
Aerodynamic Forces

- Free stream conditions: M = 0.3, Re = 5e+6, Alpha = 2.75°





- Meshes are coarse for these simulations
  - Large difference between P1 and P2 solutions

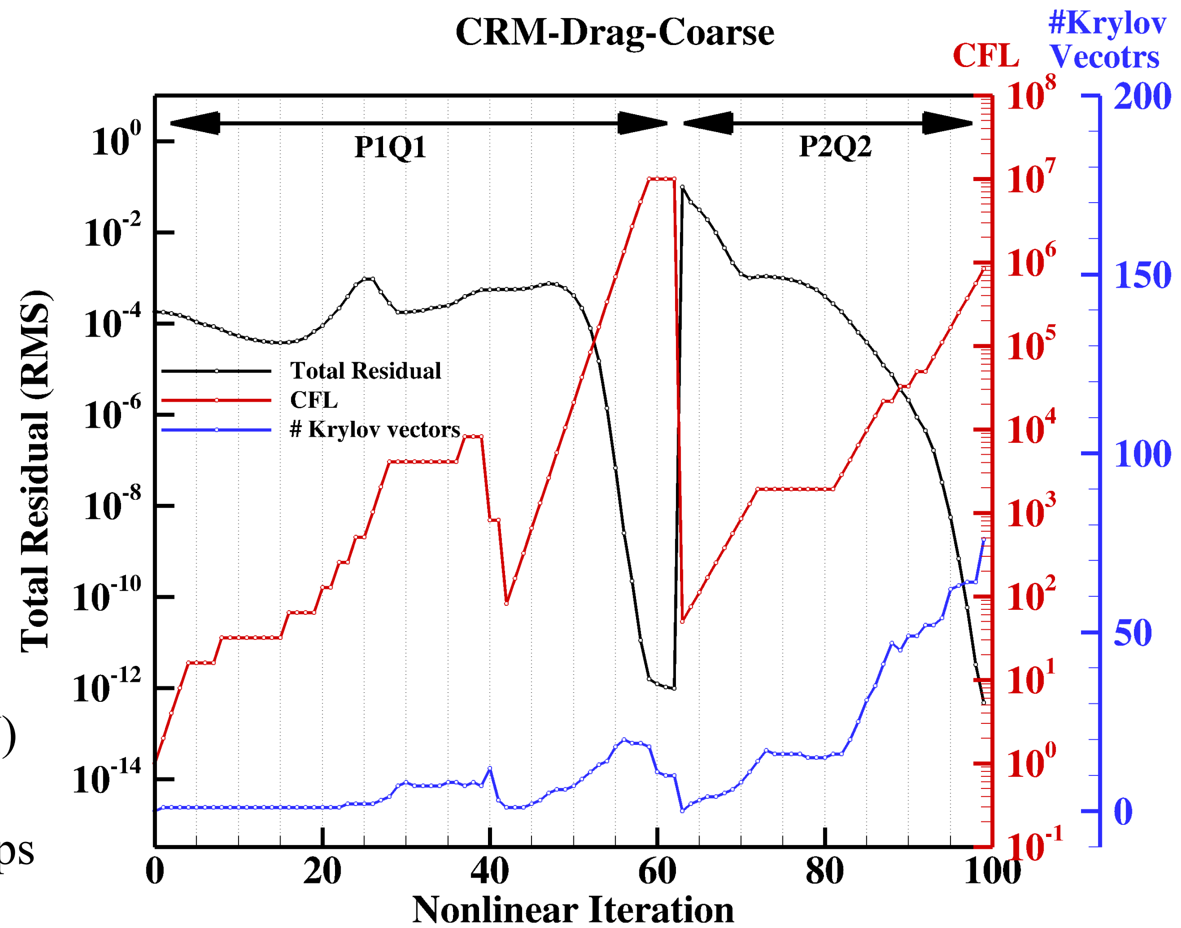
Transonic Turbulent Flow over DPW-6 Configuration
Convergence Histories

Free stream conditions:
- M = 0.85
- Re = 5e+6
- Alpha = 2.75°

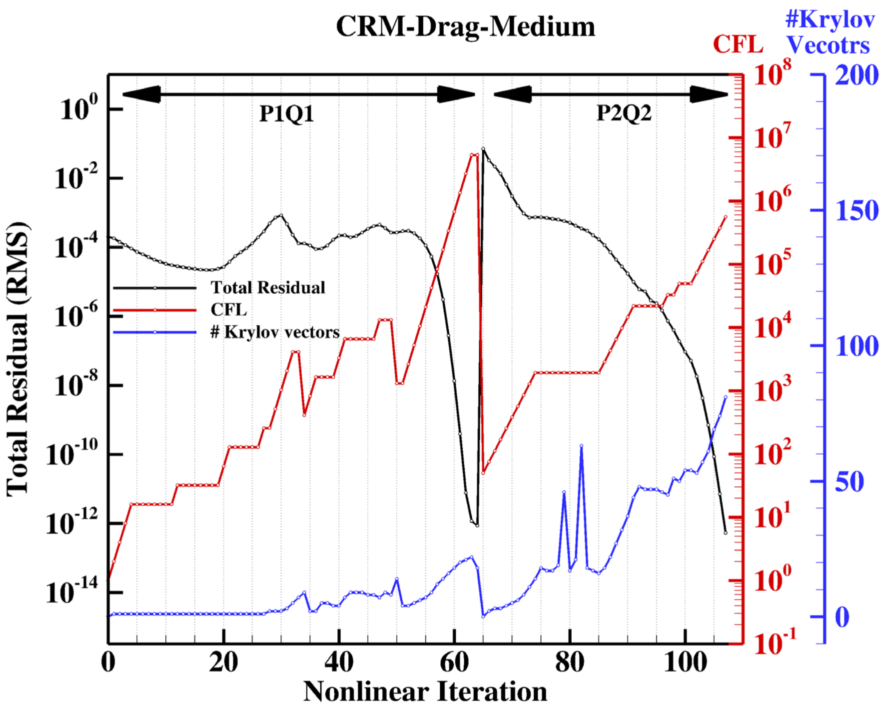
Nonlinear solver:
- PTC

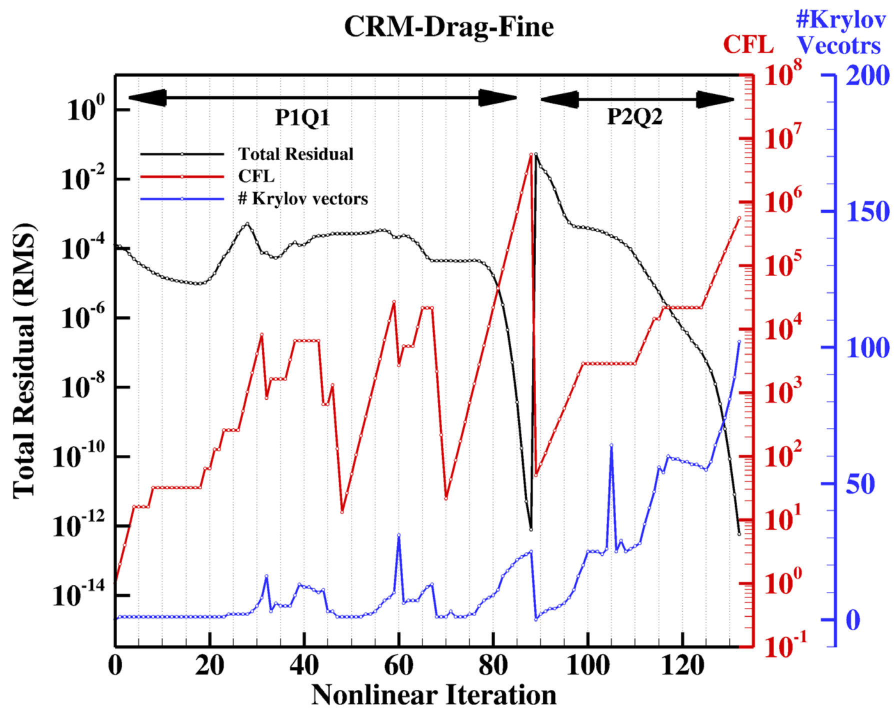
Linear solver:
- FGMRes
- Relative tol. = 10^{-4}

Linear preconditioner:
- Dual-CFL line solver (PILJ) See Refs. [1,2]
- Maximum number of sweeps per line = 200



Transonic Turbulent Flow over DPW-6 Configuration
Convergence Histories





- Slight dependency on the mesh resolution,
  But, remember that finer grids are not much finer!

- But why line preconditioner? Why not ILU(k)?
- We go with lines because…
  1. Strong scalability
     - Linear and nonlinear convergence are independent of number of partitions
  2. Significantly less memory (hundreds of times)
     - More suitable for emerging HPC architectures
  3. More tunability
     - Increased effectiveness using more iterations in preconditioning
  4. More computational efficiency

Slide From
AIAA Paper 2017-4275

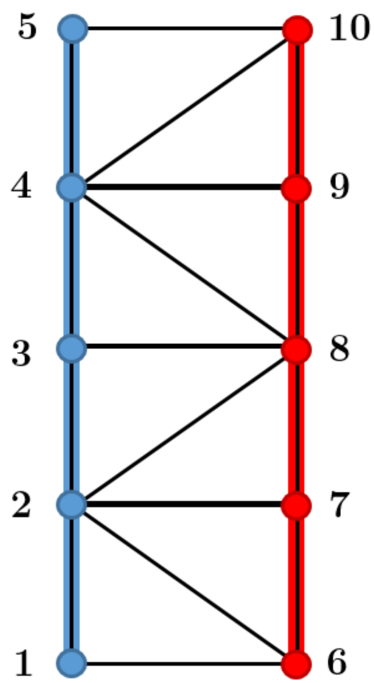
Implicit Line Preconditioner

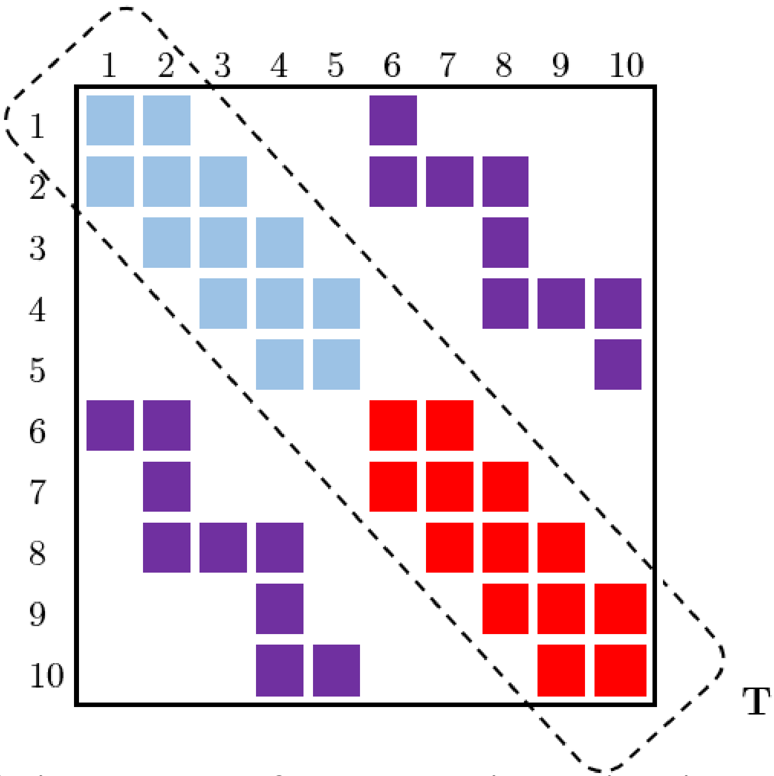
- Identify and solve implicitly along strong connections.
- Attempt to reproduce success of line solver observed in FV.
- Works well on 1st order Jacobian matrix BUT not on 2nd order. (Diagonal dominancy)
- To solve

$$[\mathbf{A}]\mathbf{x} = \mathbf{b}$$

$$[\mathbf{A}] = [\mathbf{T}] + [\mathbf{O}]$$

$$\begin{aligned}
&\text{for } k = 1, n_i \\
&\quad \mathbf{r}^{k-1} = \mathbf{b} - [\mathbf{A}]\mathbf{x}^{k-1} \\
&\quad \textit{Solve } [\mathbf{T}]d\mathbf{x}^k = \mathbf{r}^{k-1} \\
&\quad \mathbf{x}^k = \mathbf{x}^{k-1} + \omega \cdot d\mathbf{x}^k \\
&\text{end for}
\end{aligned}$$





Connectivity pattern for a **P1** Discretization

Slide From
AIAA Paper 2017-4275

Implicit Line Preconditioner

- Identify and solve implicitly along strong connections.
- Attempt to reproduce success of line solver observed in FV.
- Works well on 1st order Jacobian matrix BUT not on 2nd order. (Diagonal dominancy)
- To solve

$$[\mathbf{A}]\mathbf{x} = \mathbf{b}$$

$$[\mathbf{A}] = [\mathbf{T}] + [\mathbf{O}]$$

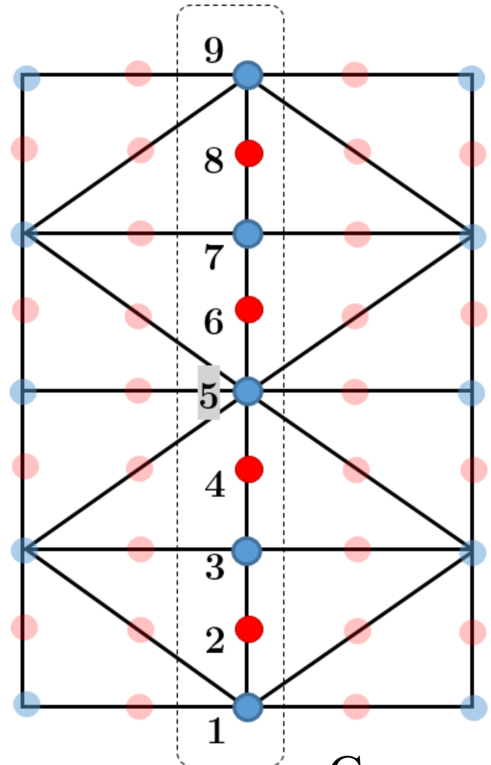
for $k = 1, n_i$

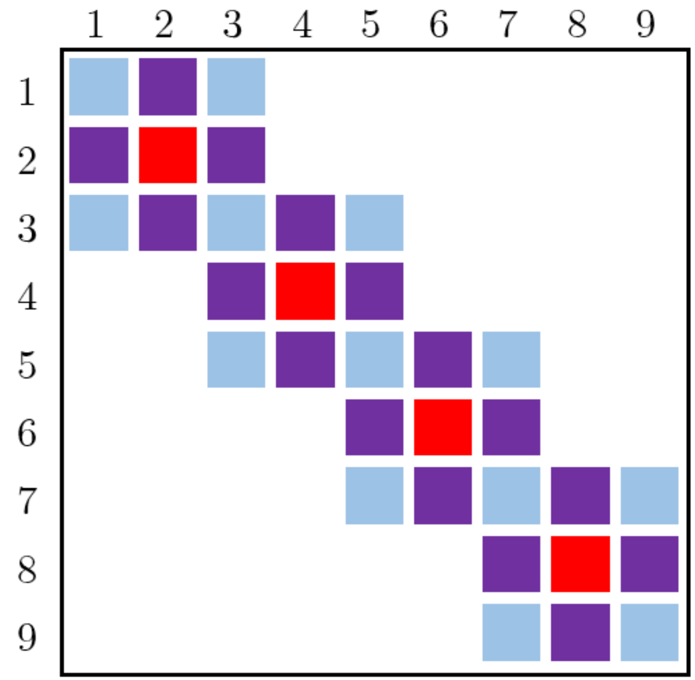
$\quad \mathbf{r}^{k-1} = \mathbf{b} - [\mathbf{A}]\mathbf{x}^{k-1}$

$\quad \textit{Solve}\ [\mathbf{T}]d\mathbf{x}^k = \mathbf{r}^{k-1}$

$\quad \mathbf{x}^k = \mathbf{x}^{k-1} + \omega \cdot d\mathbf{x}^k$

end for





Connectivity pattern for a **P2** discretization

- Gauss elimination does not produce any fill-ins
- ILU(0) gives exact factorization

Slide From
AIAA Paper 2017-0517

Dual CFL Strategy for Implicit Line Solver

- Newton-Krylov system: $\left[\mathbf{A}_{\mathrm{CFL_{NK}}}\right]\mathbf{x} = \mathbf{b}$ Solved using FGMRes
- Preconditioner system: $\left[\mathbf{A}_{\mathrm{CFL_{NK}}}\right]\mathbf{x} = \mathbf{b}$ Solved using tridiagonal solver
- Reformulate the preconditioner system as a defect correction method:

$$\mathbf{r} = \mathbf{b} - \left[\mathbf{A}_{\mathrm{CFL_{NK}}}\right]\mathbf{x} \qquad n_o \text{ outer iterations}$$

Dual CFL Strategy

$$\left[\mathbf{A}_{\mathrm{CFL_{NK}}}\right]\Delta\mathbf{x} = \mathbf{r}$$

$$\left[\mathbf{A}_{\mathrm{CFL_{Line}}}\right]\Delta\mathbf{x} = \mathbf{r} \qquad n_i \text{ inner iterations}$$

$$\mathrm{CFL_{Line}} = \min(\mathrm{CFL_{NK}}, \mathrm{CFL_{Cap}})$$

- Solved using block tridiagonal/pentadiagonal solver:

$$[\mathbf{T}]\Delta(\Delta\mathbf{x}) = \mathbf{r} - \left[\mathbf{A}_{\mathrm{CFL_{Line}}}\right]\Delta\mathbf{x}$$

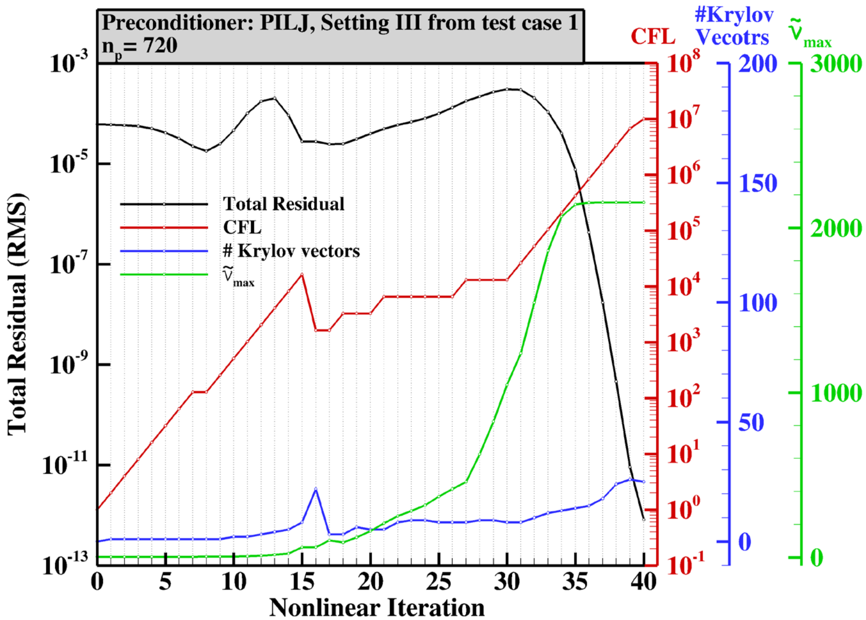
- **Explained as a Preconditioned Implicit Line Jacobi (PILJ) method**

Comparison of line and ILU(k) preconditioners on the Subsonic Turbulent Flow over DPW-6 Configuration

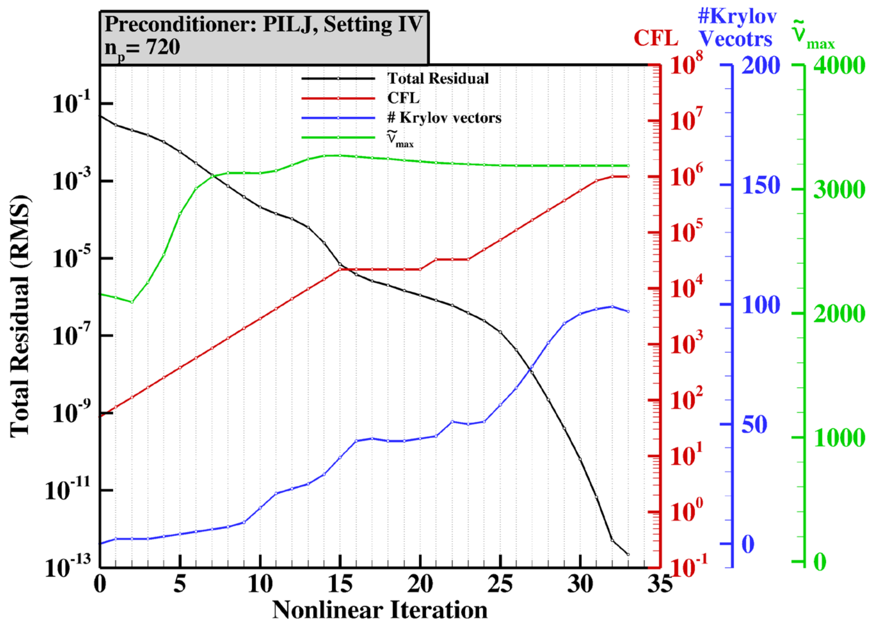
- Memory efficiency
- Computational efficiency

Subsonic Turbulent Flow over DPW-6 Configuration

- Free stream conditions: M = 0.3, Re = 5e+6, Alpha = 4.0°
- HiOCFD5 coarse mesh
- Preconditioner: Line (PILJ)



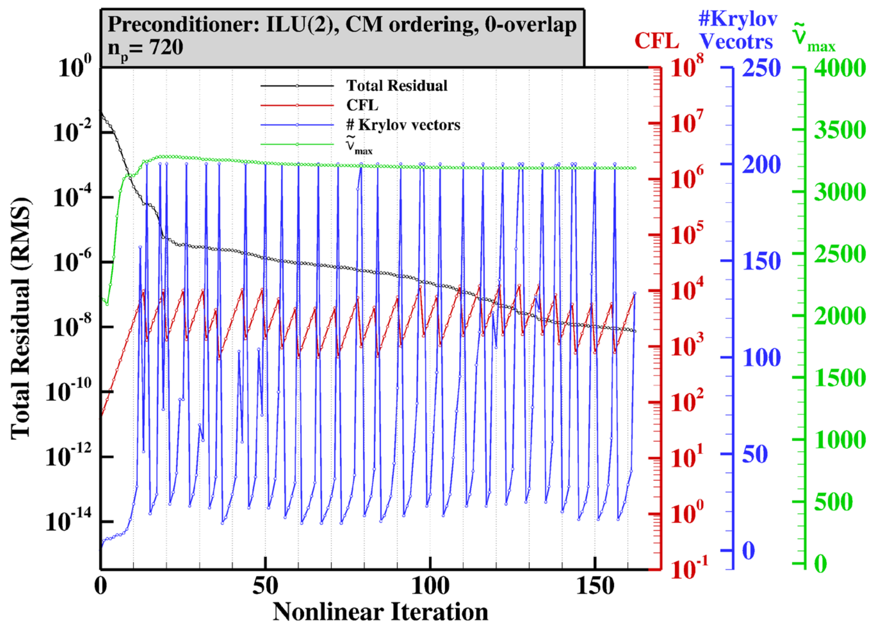
P1 solution



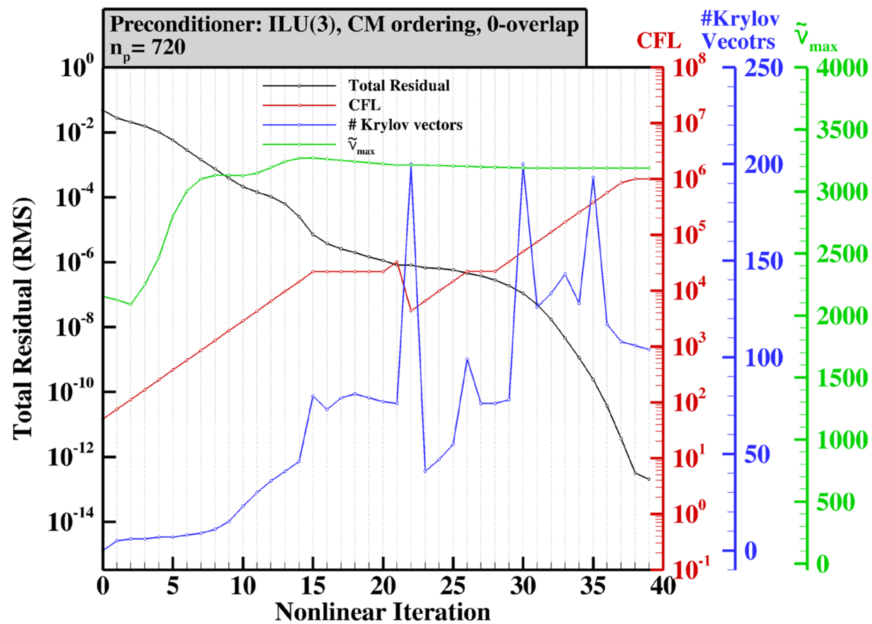
P2 solution
(Initialized by converged P1 solution)

Subsonic Turbulent Flow over DPW-6 Configuration

- Free stream conditions: M = 0.3, Re = 5e+6, Alpha = 4.0º
- HiOCFD5 coarse mesh
- Preconditioner: ILU(k)



P2 solution, with ILU(2), 0-overlap

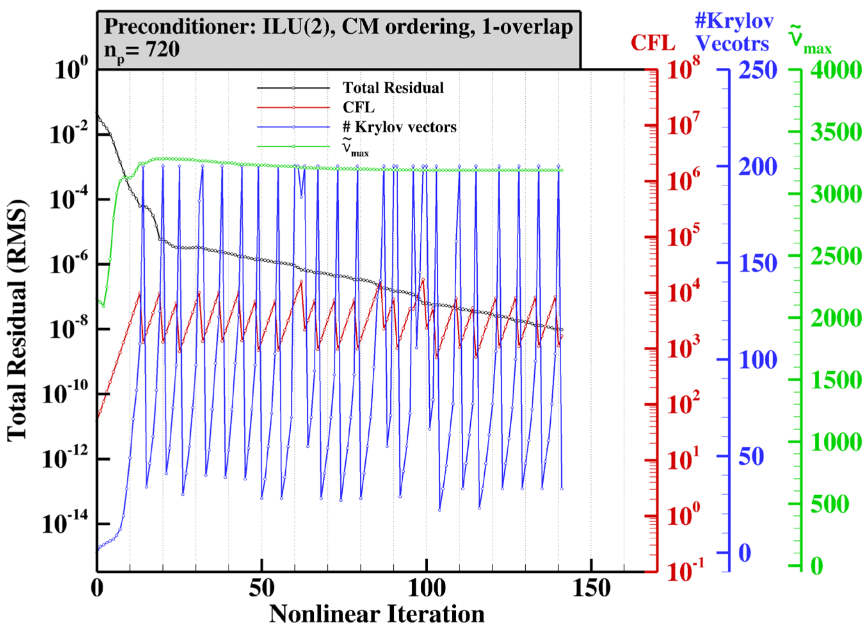


P2 solution, with ILU(3), 0-overlap
(Initialized by converged P1 solution)

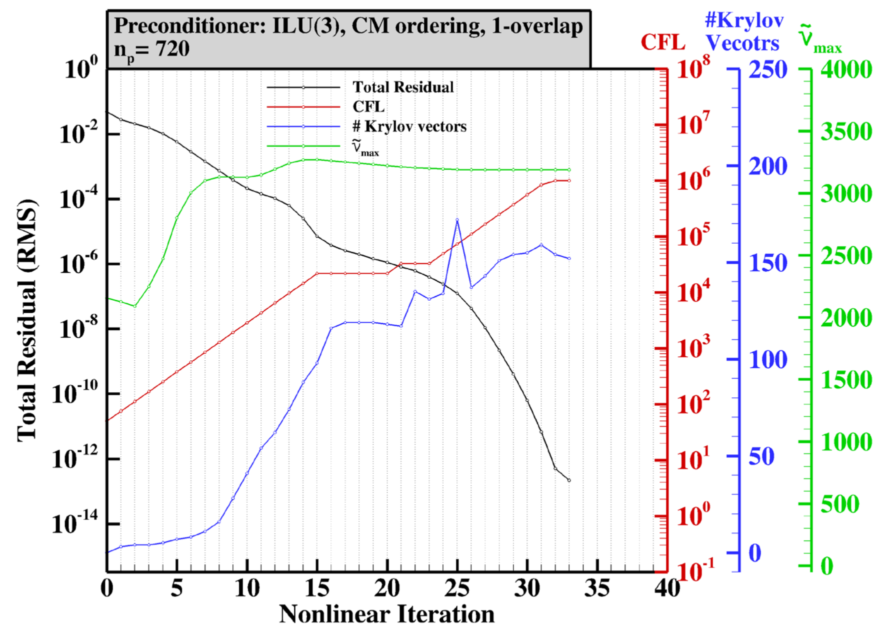
❑ ILU(2) is sufficient to solve the P1 problem)

Subsonic Turbulent Flow over DPW-6 Configuration

- Free stream conditions: M = 0.3, Re = 5e+6, Alpha = 4.0º
- HiOCFD5 coarse mesh
- Preconditioner: ILU(k) + Restrictive Additive Schwartz (RAS)



P2 solution, with ILU(2), 1-overlap



P2 solution, with ILU(3), 1-overlap
(Initialized by converged P1 solution)

❑ Overlapping does not make any significant improvements

Subsonic Turbulent Flow over DPW-6 Configuration
Comparison of Memory Consumption

Fill = ratio of the number of non-zero entries in the factorized matrix over the number of non-zero entries in the original matrix

$$\frac{\text{Fill (ILU(3))}}{\text{Fill (PILJ)}} = 340$$

And this is just for k=3

PILJ (Line Preconditioner) Fill = 0.066				
	Setting IV			
n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
180	32	1269	99	4.075E4
360	32	1269	99	2.067E4
720	32	1269	99	1.059E4

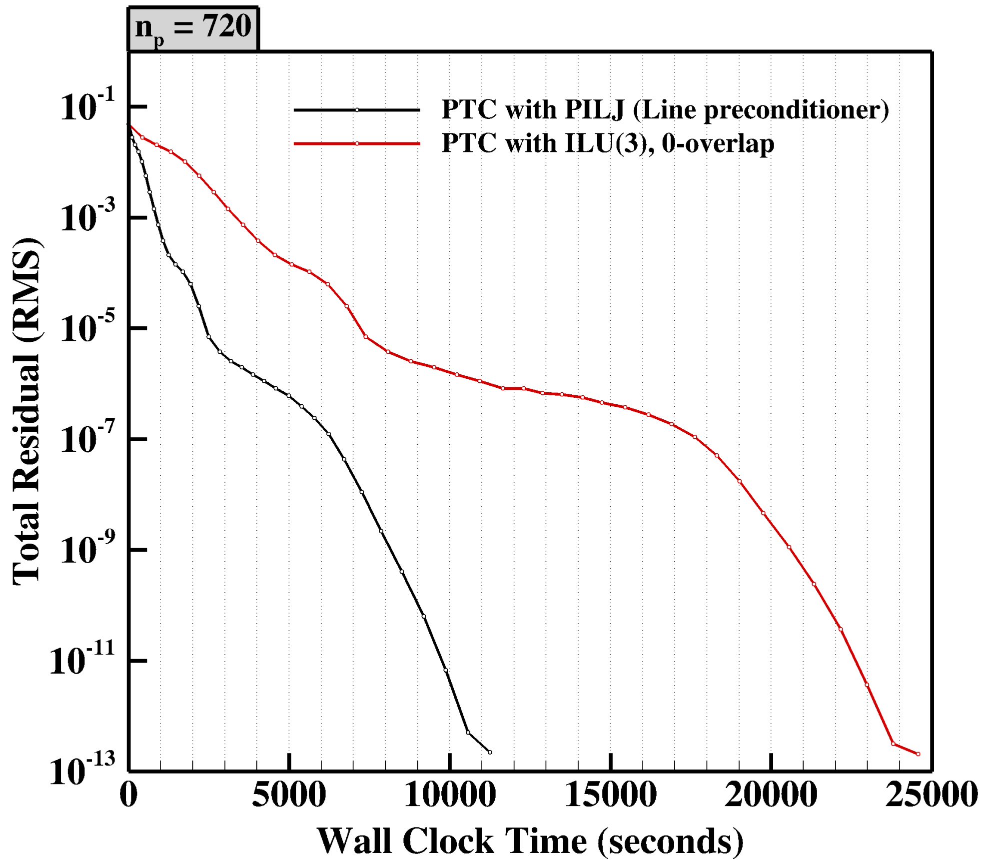
Setting IV:

$\text{CFL}_{cap} = 250, n_o = 10, n_i = 20, \omega_L = 0.1$

CM ordering									
		ILU (2) (Fill=12.36)				ILU (3) (Fill=22.46)			
Overlap	n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
0	180	NM	NM	NM	NM	NM	NM	NM	NM
	360	SC	SC	200	SC	NM	NM	NM	NM
	720	SC	SC	200	SC	38	2721	200	2.38E4
1	180	NM	NM	NM	NM	NM	NM	NM	NM
	360	SC	SC	200	SC	NM	NM	NM	NM
	720	SC	SC	200	SC	32	2988	200	2.44E4

Subsonic Turbulent Flow over DPW-6 Configuration
Comparison of Run Time

- Free stream conditions: M = 0.3, Re = 5e+6, Alpha = 4.0°
- HiOCFD5 coarse mesh



Conclusions

- Feasibility of a high-order SUPG solver was demonstrated for the transonic flow over common research model.
- The effectiveness of the implicit line preconditioner was demonstrated for high-order continuous finite-element methods.
- Nonlinear convergence showed slight dependency on the mesh resolution.
- Memory and computational efficiency of the line preconditioner was compared with ILU(k)

Acknowledgments

"This research was sponsored by NASA's Transformational Tools and Technologies (TTT) Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate."

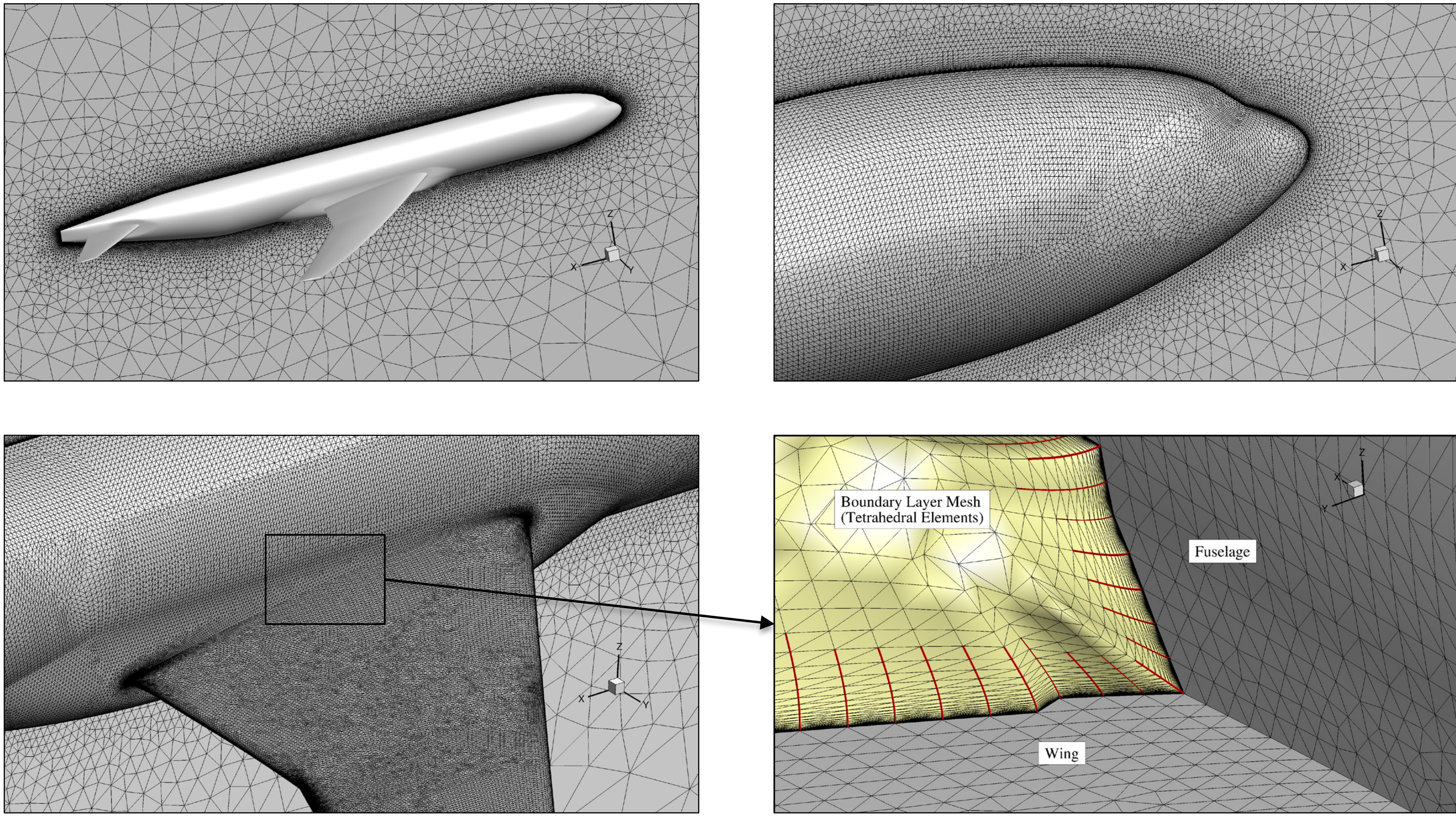
Thank you!
Any questions?

Backup Slide:
- More details on the line preconditioner
- Demonstration of the strong scalability
- Effect of different node orderings on ILU(k)

Lines on P1 and DPW-4 Configuration (Transonic)

- A transonic turbulent test case
- Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0^{o}
- Comparison of PILJ and ILU(k) on a P1 problem

Transonic Turbulent Flow over DPW-4 Configuration
Custom Unstructured Mesh
6,861,035 DOFs for P1



Matrix-based lines shown in red

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°

PILJ (Line Preconditioner)												
	Setting I				Setting II				Setting III			
n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
90	Not tested				125	2801	120	1.439E5	**100**	**710**	**49**	**9.709E4**
180	Not tested				125	2801	120	7.380E4	**100**	**710**	**49**	**4.979E4**
360	Not tested				125	2801	120	3.837E4	**100**	**710**	**49**	**2.589E4**
720	SC	SC	200	SC	125	2801	120	2.013E4	**100**	**710**	**49**	**1.358E4**

Nomenclature:

n_p = Number of processors (and partitions)

n_{NL} = Number of non-linear steps to reach the full convergence (up to machine precision)

k_{max} = Maximum number of Krylov vectors used at the linear solutions during the nonlinear steps

t_{tot} = Total elapsed time in seconds

NC = No convergence (the nonlinear solver was very slow and the simulation was not continued)

SC = Slow convergence (due to frequent failure of the linear solver the nonlinear proceeds with low CFL)

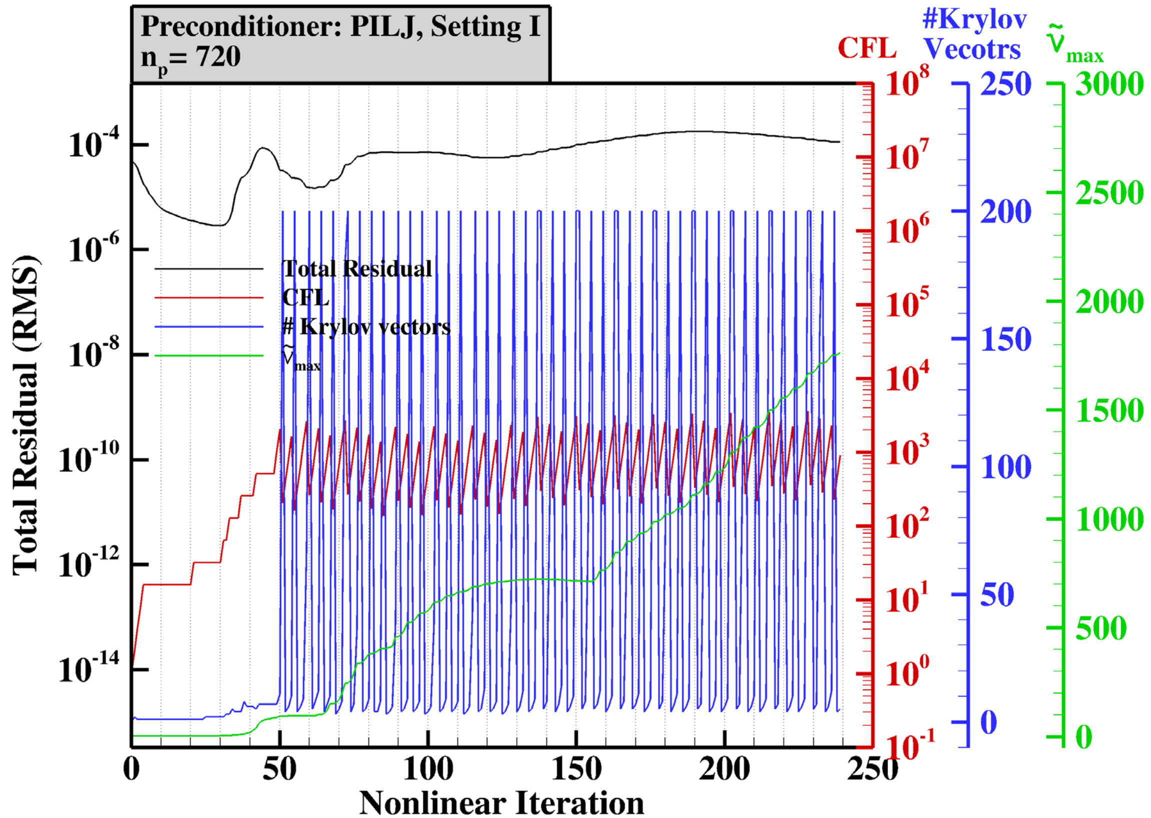
Setting I: $\mathrm{CFL}_{cap} = 10^8, n_o = 5, n_i = 10, \omega_L = 0.3$ (see Algorithms 1 and 2)

Setting II: $\mathrm{CFL}_{cap} = 10^3, n_o = 5, n_i = 10, \omega_L = 0.3$ (see Algorithms 1 and 2)

Setting III: $\mathrm{CFL}_{cap} = 10^3, n_o = 10, n_i = 20, \omega_L = 0.3$ (see Algorithms 1 and 2)

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°



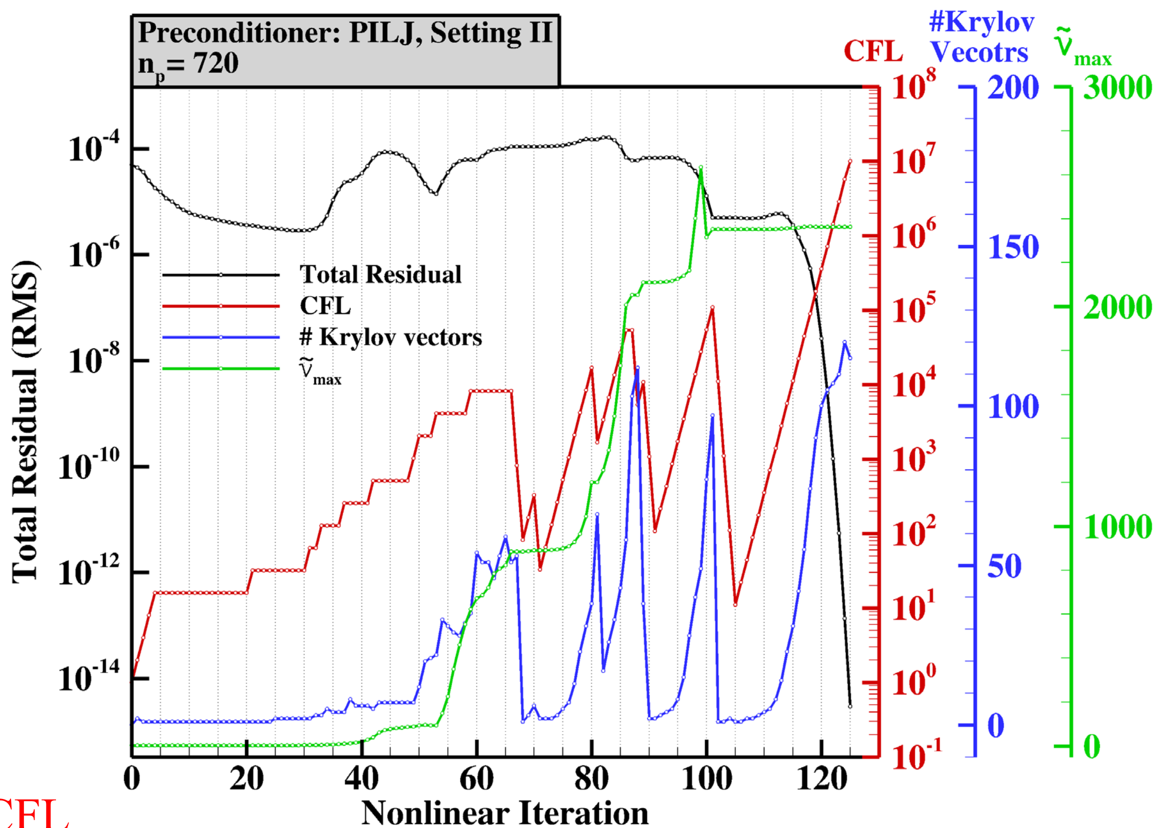
Setting I: $\boxed{\mathrm{CFL}_{cap} = 10^8}, n_o = 5, n_i = 10, \omega_L = 0.3$

Deactivates the dual-CFL strategy

→ Slow Progress

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°



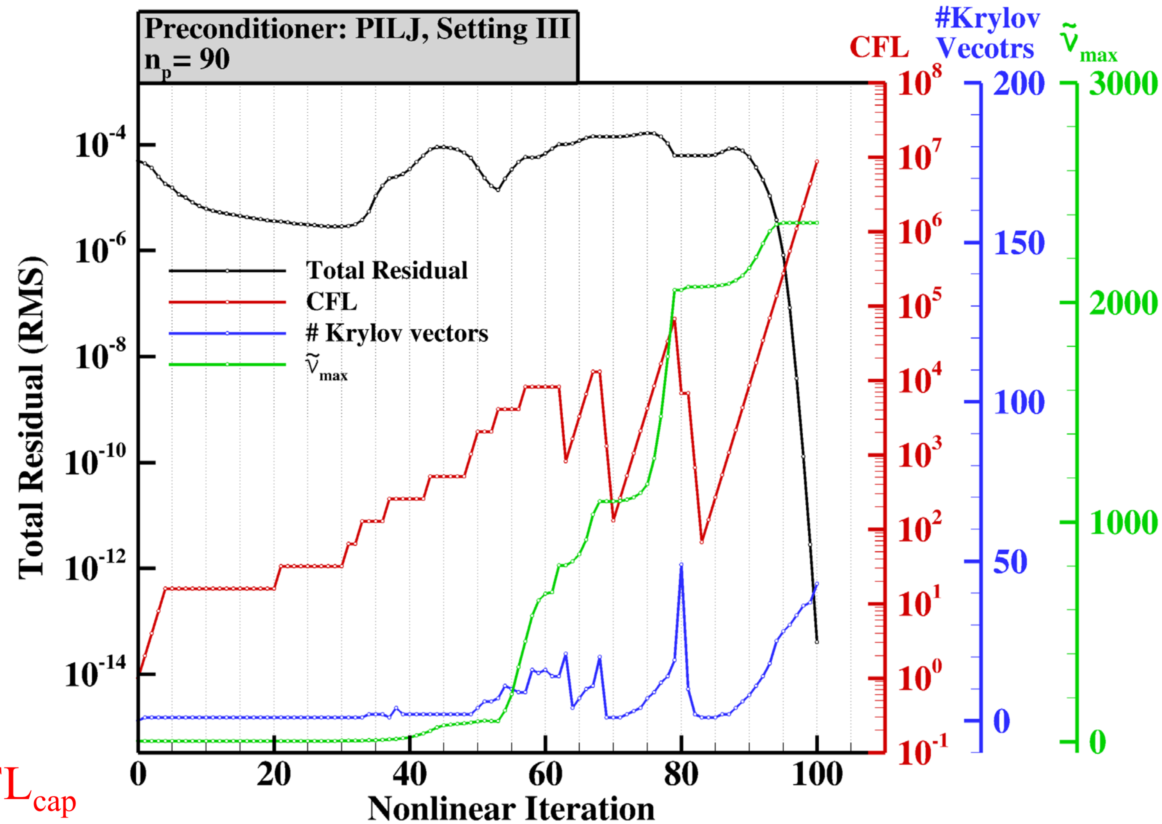
By decreasing CFL_{cap}

Setting II: $\mathrm{CFL}_{cap} = 10^3, n_o = 5, n_i = 10, \omega_L = 0.3$

→ Fast Progress

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°



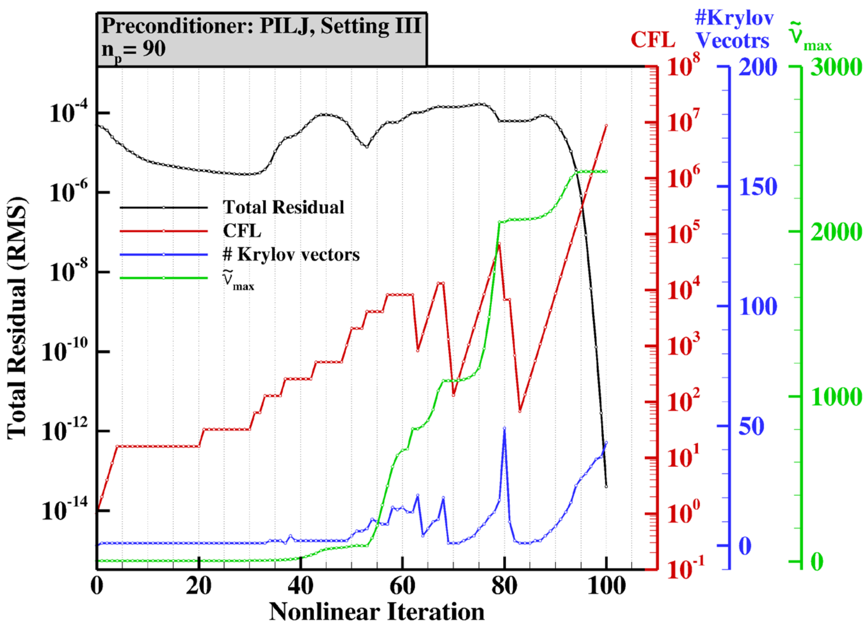
By decreasing CFL_{cap} and using more sweeps:

$$\text{Setting III: } \text{CFL}_{cap} = 10^3, n_o = 10, n_i = 20, \omega_L = 0.3$$

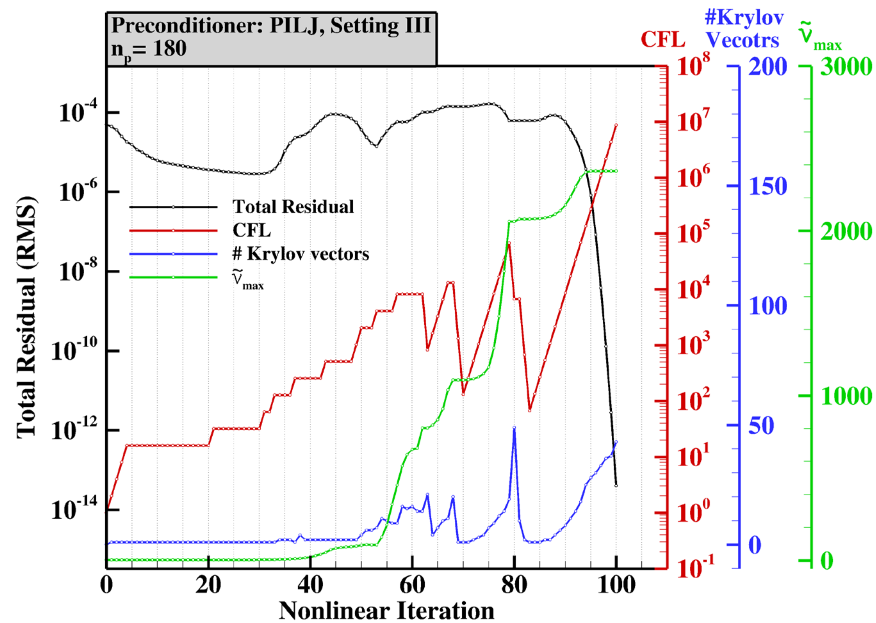
→ Faster Progress

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°



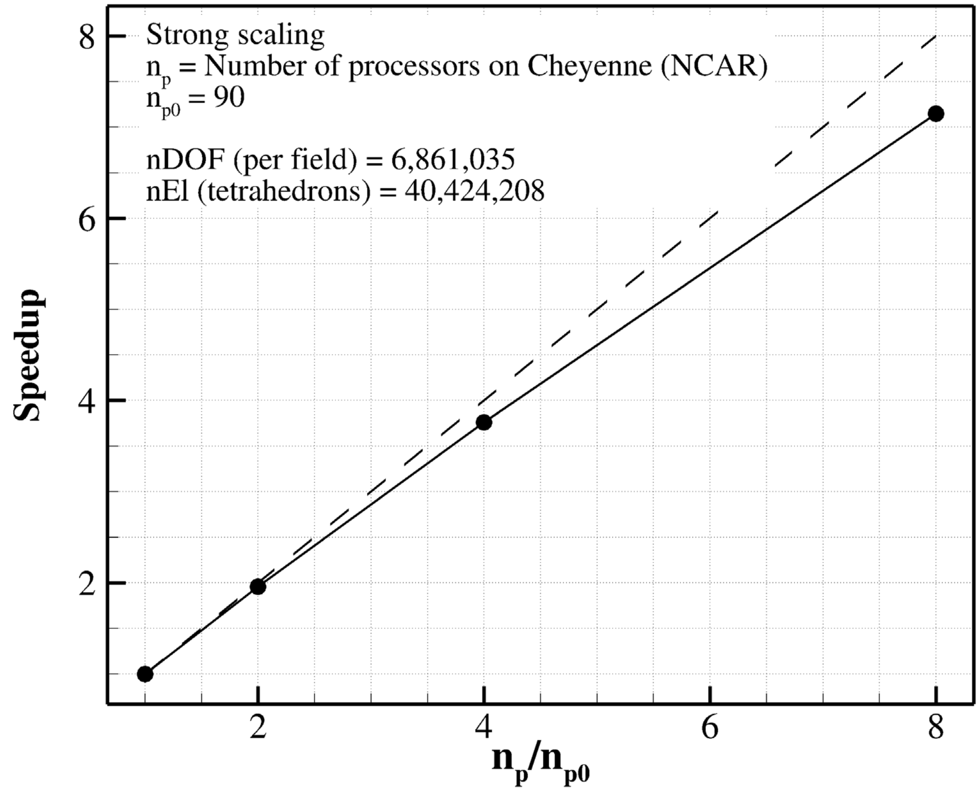
$n_p = 90$



$n_p = 180$

Convergence behavior is independent of number of partitions
→ Strong scalability

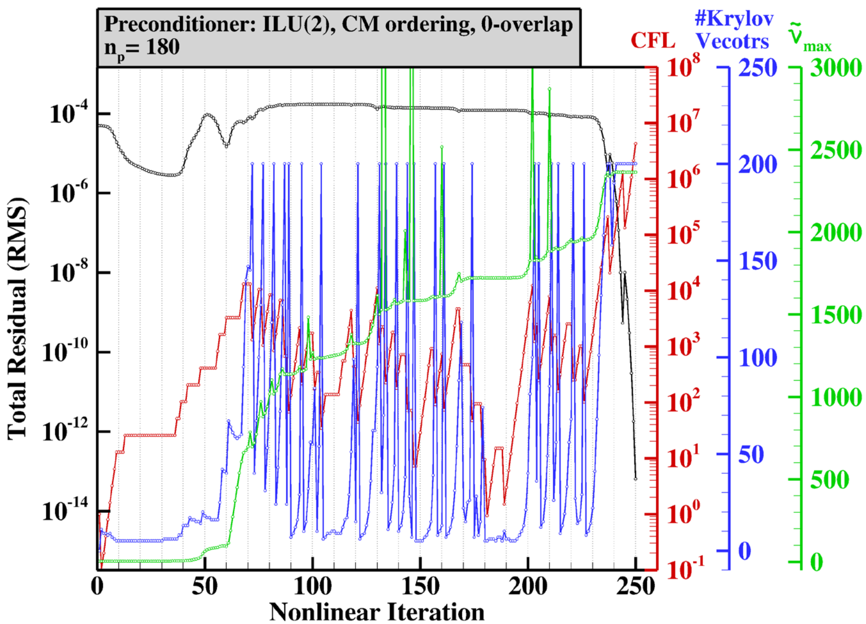
Transonic Turbulent Flow over DPW-4 Configuration
Strong Scaling using PILJ



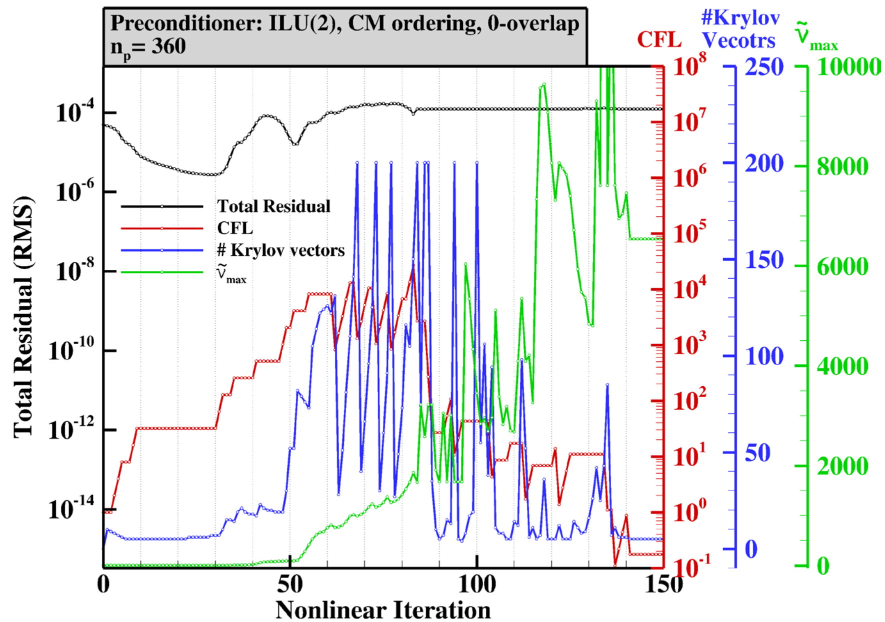
Can we get the same behavior with ILU(k)? Not always…

Transonic Turbulent Flow over DPW-4 Configuration

Flow conditions: M = 0.85, Re = 5 million, alpha = 2.0°



$n_p = 180$



$n_p = 360$

Do overlapping or reordering methods solve this issue? Not always…

Transonic Turbulent Flow over DPW-4 Configuration

CM ordering													
		ILU (2) (Fill=4.57)				ILU (3) (Fill=8.08)				ILU (4) (Fill=12.86)			
Overlap	n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
0	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	250	18078	200	9.13E4	NM	NM	NM	NM	NM	NM	NM	NM
	360	NC	NC	200	NC	115	6161	200	2.32E4	NM	NM	NM	NM
	720	NC	NC	200	NC	136	8132	200	1.44E4	118	7399	200	1.76E4
1	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	NC	NC	200	NC	NM	NM	NM	NM	NM	NM	NM	NM
	360	NC	NC	200	NC	105	4499	200	2.43E4	NM	NM	NM	NM
	720	151	7046	200	1.57E4	128	6554	200	1.64E4	NC	NC	200	NC

Nomenclature:

n_p = Number of processors (and partitions)

n_{NL} = Number of non-linear steps to reach the full convergence (up to machine precision)

k_{max}= Maximum number of Krylov vectors used at the linear solutions during the nonlinear steps

t_{tot} = Total elapsed time in seconds

NM = Not enough memory to run the case

NC = No convergence (the nonlinear solver stalled before the getting close to the final solution)

Fill = The ratio of the number of non-zero entries in the factorized matrixed over the number of non-zero entries in the original matrix (averaged over all partition numbers)

Overlap = overlap level for the Restrictive Additive Schwarz (RAS) method

Transonic Turbulent Flow over DPW-4 Configuration

RCM ordering													
		ILU (2) (Fill=4.47)				ILU (3) (Fill=8.04)				ILU (4) (Fill=12.68)			
Overlap	n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
0	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	NC	NC	200	NC	NM	NM	NM	NM	NM	NM	NM	NM
	360	171	15811	200	3.07E4	NC	NC	200	NC	NM	NM	NM	NM
	720	NC	NC	200	NC	NC	NC	200	NC	128	8160	200	1.48E4
1	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	138	7936	200	5.54E4	NM	NM	NM	NM	NM	NM	NM	NM
	360	NC	NC	200	NC	NC	NC	200	NC	NM	NM	NM	NM
	720	NC	NC	200	NC	NC	NC	200	NC	NC	NC	200	NC

Line ordering													
		ILU (2) (Fill=4.83)				ILU (3) (Fill=8.54)				ILU (4) (Fill=13.53)			
Overlap	n_p	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}	n_{NL}	$n_{L,tot}$	k_{max}	t_{tot}
0	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	NC	NC	200	NC	NM	NM	NM	NM	NM	NM	NM	NM
	360	NC	NC	200	NC	NC	NC	200	NC	NM	NM	NM	NM
	720	NC	NC	200	NC	NC	NC	200	NC	127	9948	200	2.21E4
1	90	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	180	NC	NC	200	NC	NM	NM	NM	NM	NM	NM	NM	NM
	360	NC	NC	200	NC	165	8148	200	4.45E4	NM	NM	NM	NM
	720	NC	NC	200	NC	193	9509	200	2.78E5	171	8583	200	4.05E4