This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2021-4535C

Sandia National Laboratories

EXASCALE COMPUTING PROJECT

Polynomial Preconditioning in Trilinos

Jennifer Loe and Erik Boman

ECP Annual Meeting 2021

This research was supported by the Exascale Computing Project (17-SC-20-SC),a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Polynomial Preconditioning in Belos

Polynomial Preconditioning:

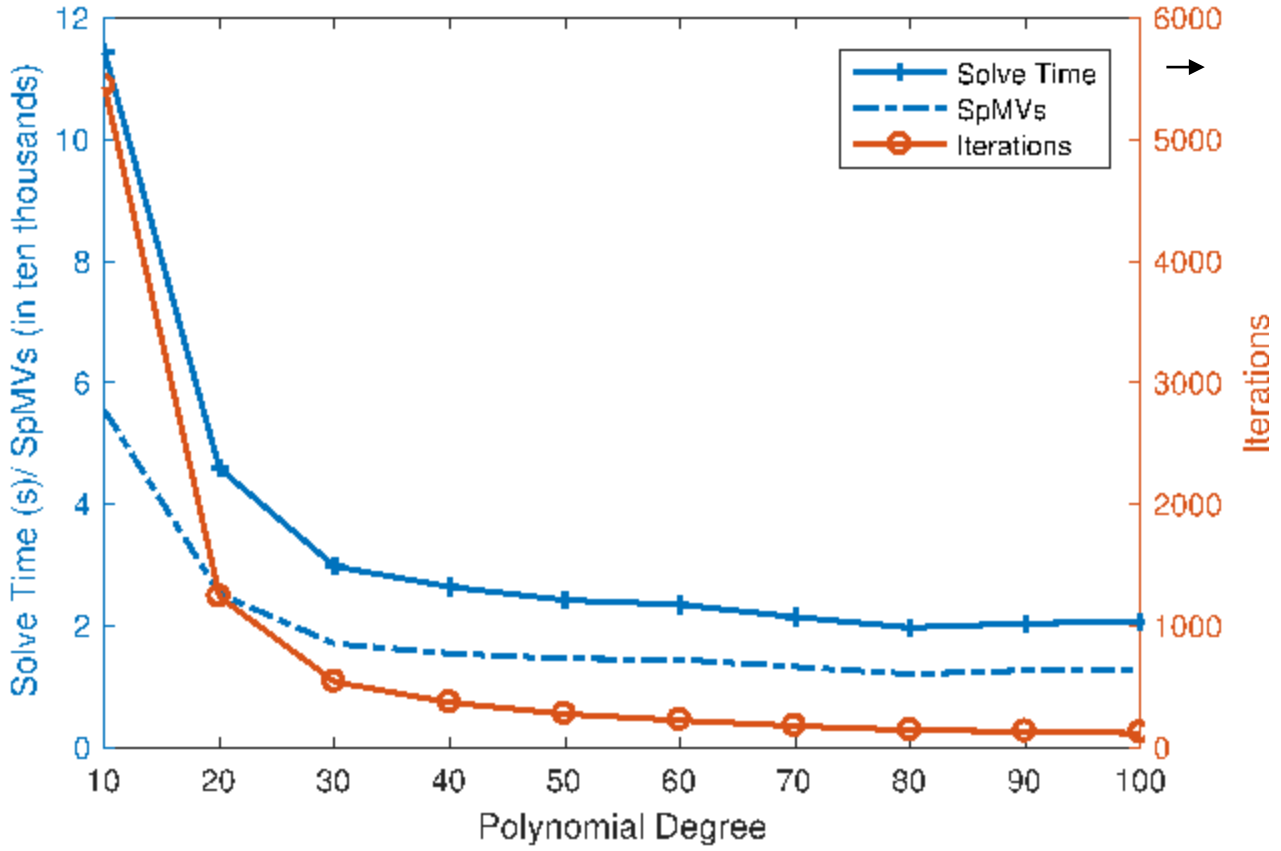
- $Ax = b \rightarrow p(A)Ax = p(A)b$
- $p(x)$ is chosen to be a GMRES polynomial for A.
- Available now in the Trilinos linear solvers package Belos!

New stable implementation of polynomial preconditioner; expected to provide good performance for parallel computing.

- New "roots" implementation that is stable at high degrees.
- Stand-alone or combine with other standard preconditioners, such as ILU.
- More SpMVs in exchange for fewer dot products requiring global synchronization.

**For polynomial preconditioning details, see:
Jennifer Loe, Erik Boman, and Heidi Thornquist. *Polynomial Preconditioned GMRES in Trilinos: Practical Considerations for High-Performance Computing*

Polynomial Preconditioning Speedup:



Matrix is cfd2, n=123440. Running GMRES(50) with random right-hand side. 32 MPI processes over 1 node.

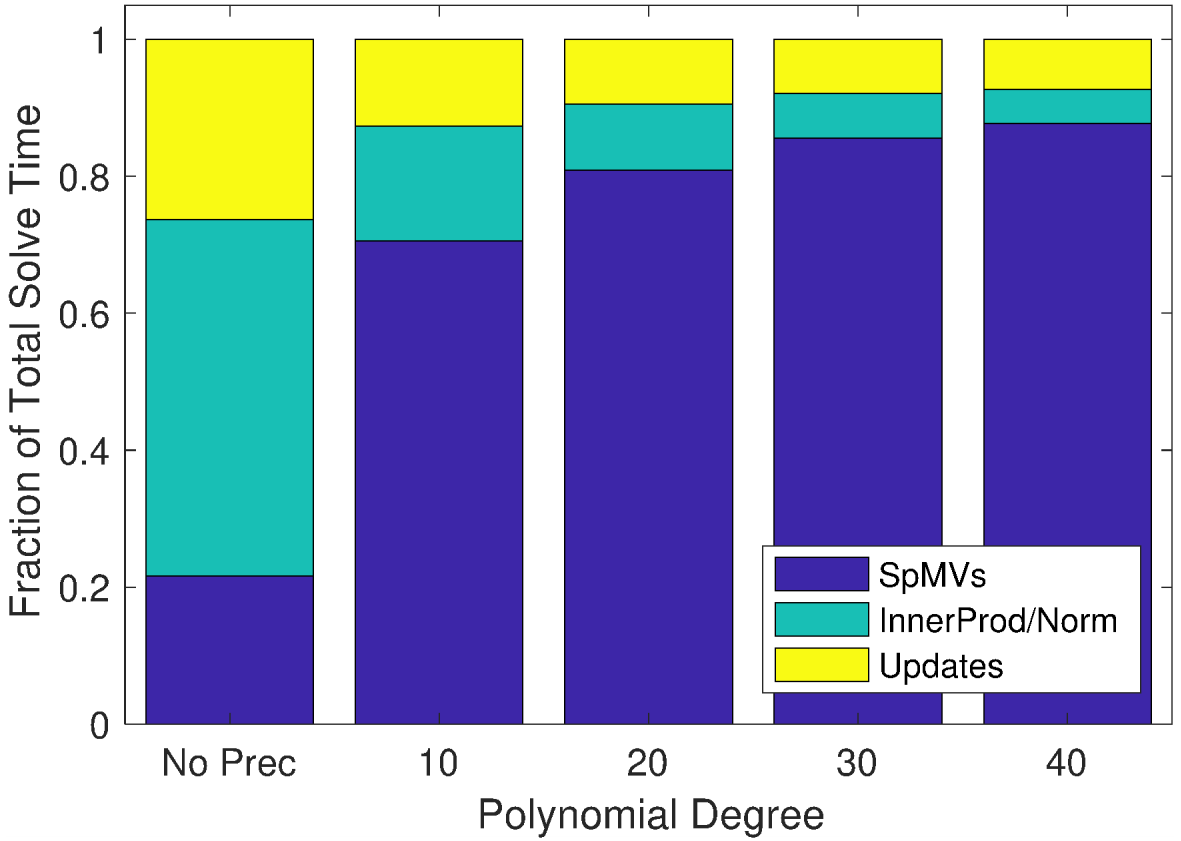
Can reduce SpMVs, iterations, and solve time- less time in orthogonalization kernels.

For this example, degree 80 polynomial gives 57x speedup over no preconditioning, with over 14x reduction in SpMVs.

Shifting global synchronization to local synchronization:

Matrix **cfd2**, *A* is SPD, *n* = 123440.

Operations requiring expensive global synchronization reduce from **52%** of solve time to less than **10%** of solve time.



Speedups from polynomial preconditioning:

Matrix	No Prec		Degree 20			Degree 40			Degree 60		
	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters
cfd2	102	85116	15740	3.82	779	11850	2.52	286	12600	2.533	198
Transport	1042	40268	6048	26.16	285	3948	15.19	93	4032	14.51	63
FEM_3D_thermal2	0.73	480	630	0.151	29	748	0.16	16	804	0.193	11
thermal2	xxx	xxx	26200	78.98	1297	14280	37.01	353	6466	17.1	104
xenon2	5.73	3304	2080	0.626	102	2200	0.521	54	2340	0.606	38
ML_Geer	xxx	xxx	260500	3214	12897	61580	731.5	1487	29570	346.7	472
G3_circuit	xxx	xxx	xxx	xxx	xxx	20080	35.88	473	xxx	xxx	xxx
G3_circuit DAMPED	N/A	N/A	47380	108.7	2233	32100	57.74	775	28730	45.82	451

Polynomial preconditioning can be used to **accelerate standard preconditioners:**

Matrix	ILU Only		ILU + Deg 5			ILU + Deg 10			ILU + Deg 20		
	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters
Transport	55.9	1879	1595	22.59	315	920	10.81	91	960	10.22	47
FEM_3D_thermal2	0.06	20	30	0.099	5	40	0.058	3	45	0.064	2
thermal2	301	12902	7310	64.73	1447	4710	34.58	466	2180	16.52	107
G3_circuit	23.9	982	1120	9.474	221	1160	7.502	114	1080	5.484	53

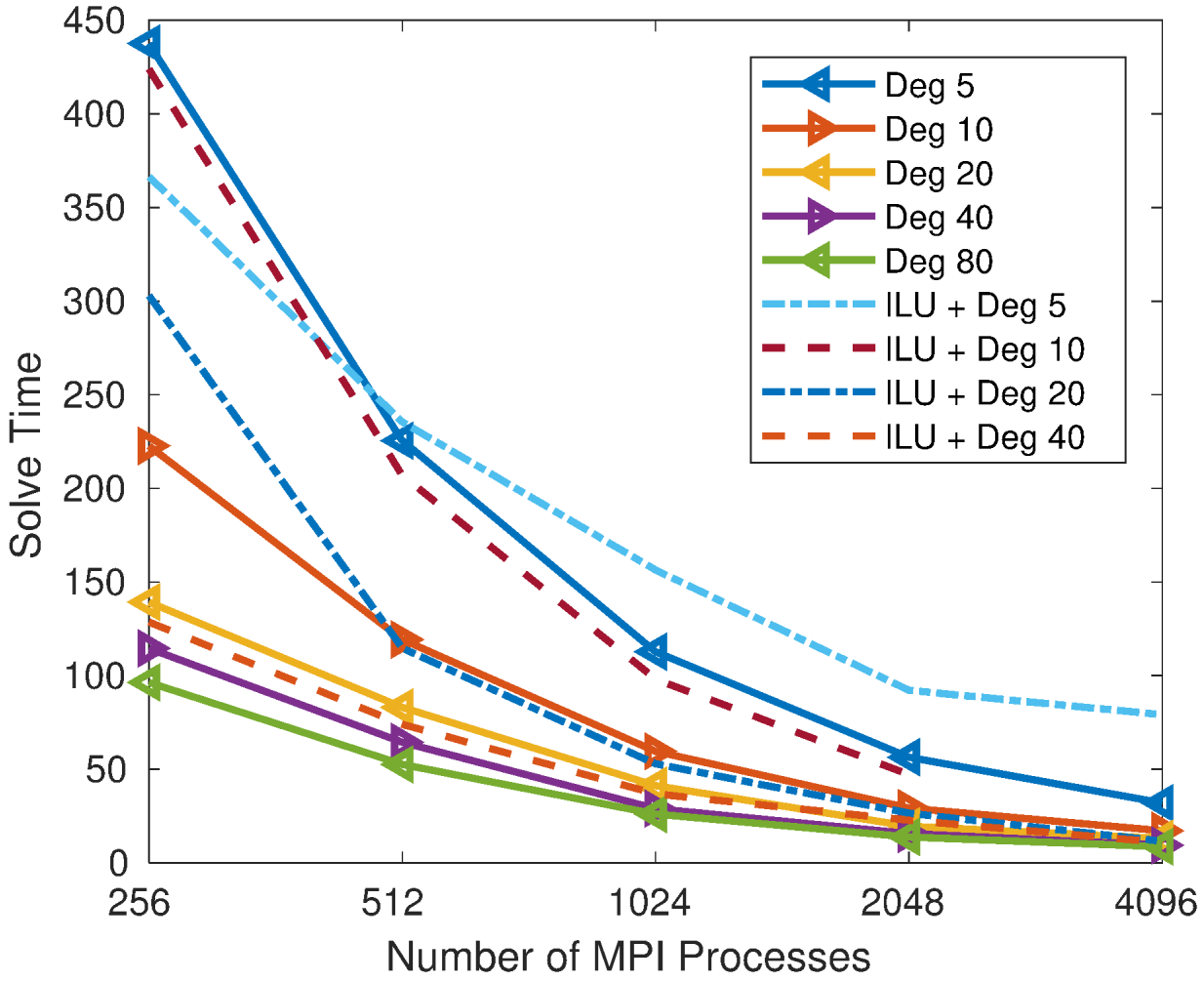
Speedups from polynomial preconditioning:

Matrix	No Prec		Degree 20			Degree 40			Degree 60		
	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters
cfd2	102	85116	15740	3.82	779	11850	2.52	286	12600	2.533	198
Transport	1042	40268	6048	26.16	285	3948	15.19	93	4032	14.51	63
FEM_3D_thermal2	0.73	480	630	0.151	29	748	0.16	16	804	0.193	11
thermal2	xxx	xxx	26200	78.98	1297	14280	37.01	353	6466	17.1	104
xenon2	5.73	3304	2080	0.626	102	2200	0.521	54	2340	0.606	38
ML_Geer	xxx	xxx	260500	3214	12897	61580	731.5	1487	29570	346.7	472
G3_circuit	xxx	xxx	xxx	xxx	xxx	20080	35.88	473	xxx	xxx	xxx
G3_circuit DAMPED	N/A	N/A	47380	108.7	2233	32100	57.74	775	28730	45.82	451

Polynomial preconditioning can be used to **accelerate standard preconditioners:**

Matrix	ILU Only		ILU + Deg 5			ILU + Deg 10			ILU + Deg 20		
	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters	SPMVs	Time	Iters
Transport	55.9	1879	1595	22.59	315	920	10.81	91	960	10.22	47
FEM_3D_thermal2	0.06	20	30	0.099	5	40	0.058	3	45	0.064	2
thermal2	301	12902	7310	64.73	1447	4710	34.58	466	2180	16.52	107
G3_circuit	23.9	982	1120	9.474	221	1160	7.502	114	1080	5.484	53

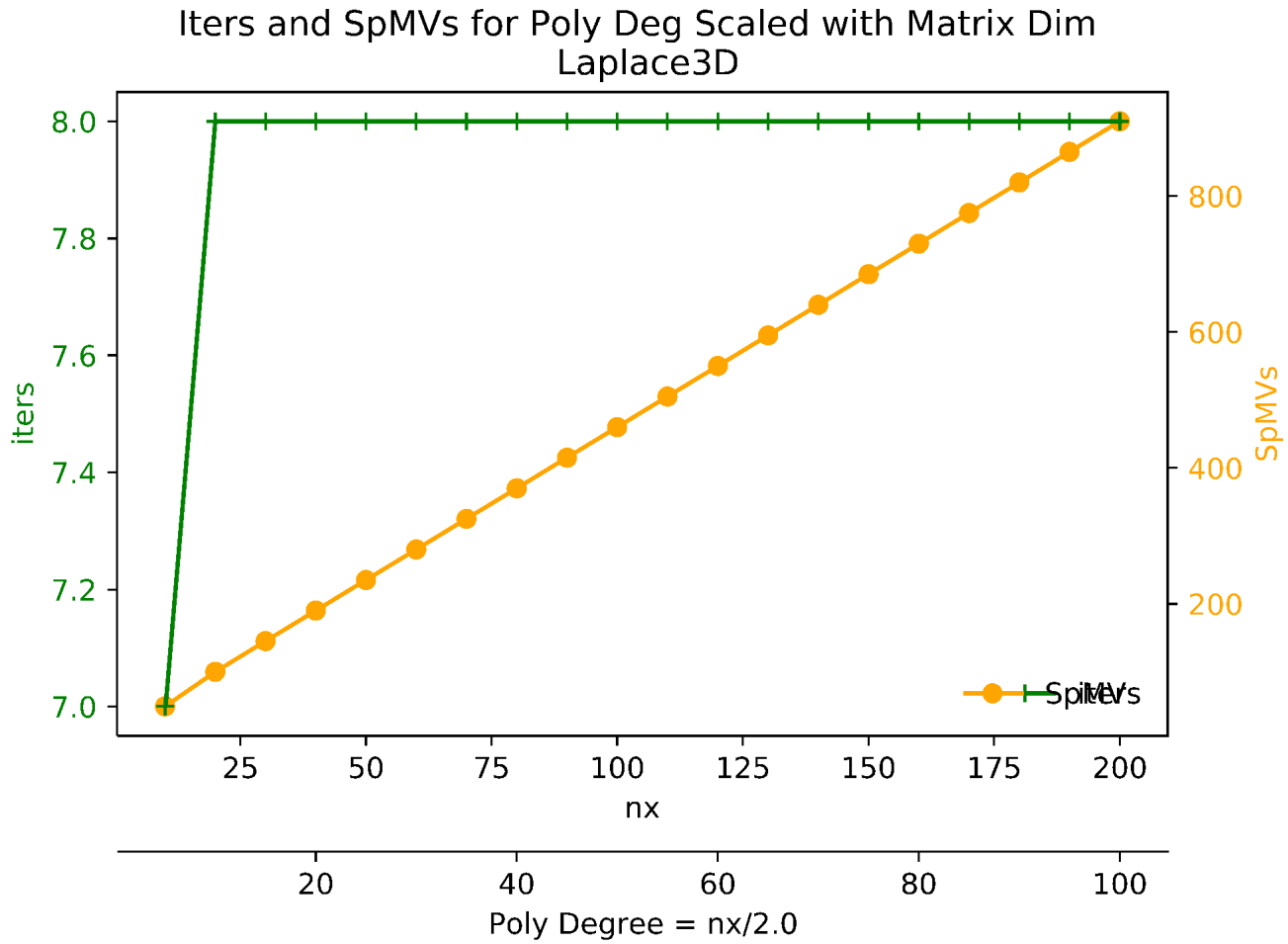
Strong Scaling with MPI:



3D Laplacian, n = 166 million,
1.1 billion nonzero elements

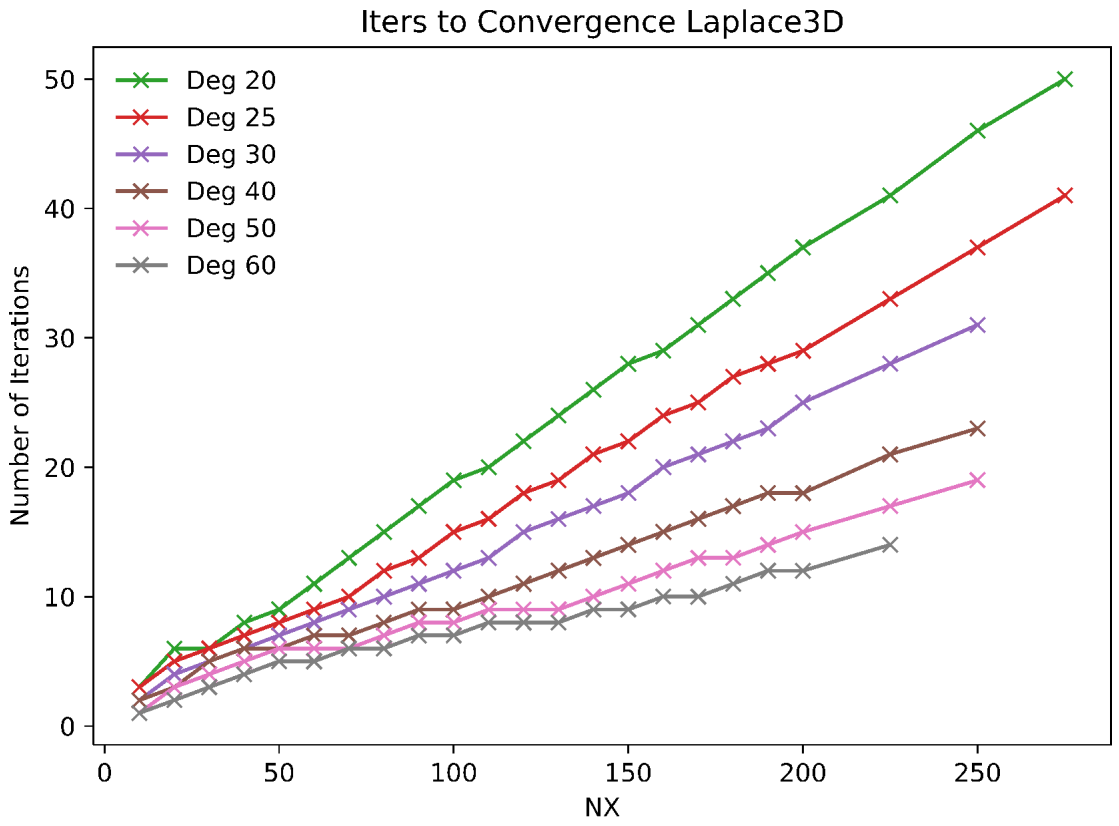
“Weak” Scaling:

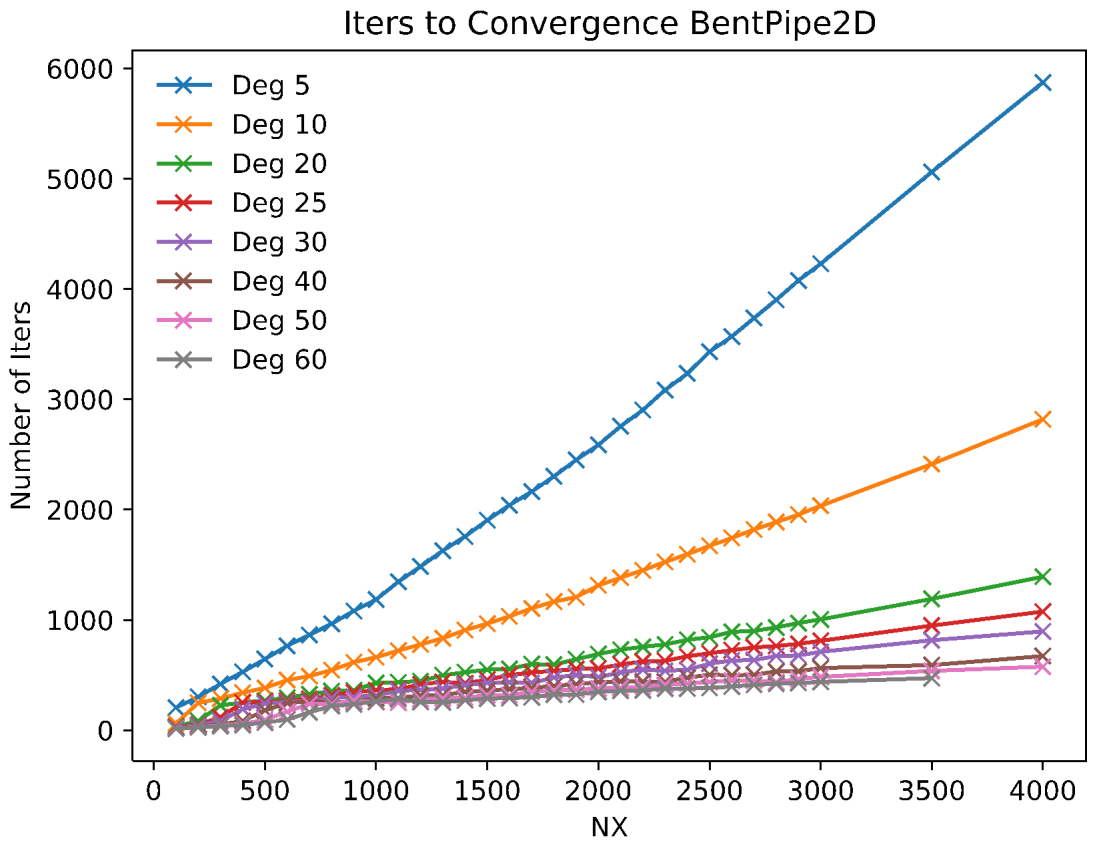
To maintain consistent number of iterations as mesh is refined, double polynomial degree for every double of nx.



NX	Poly Deg	Iters
10	5	7
20	10	8
30	15	8
40	20	8
50	25	8
60	30	8
70	35	8
80	40	8
90	45	8
100	50	8
110	55	8
120	60	8
130	65	8
140	70	8
150	75	8
160	80	8
170	85	8
180	90	8
190	95	8
200	100	8

More scaling measures with fixed polynomial degree:

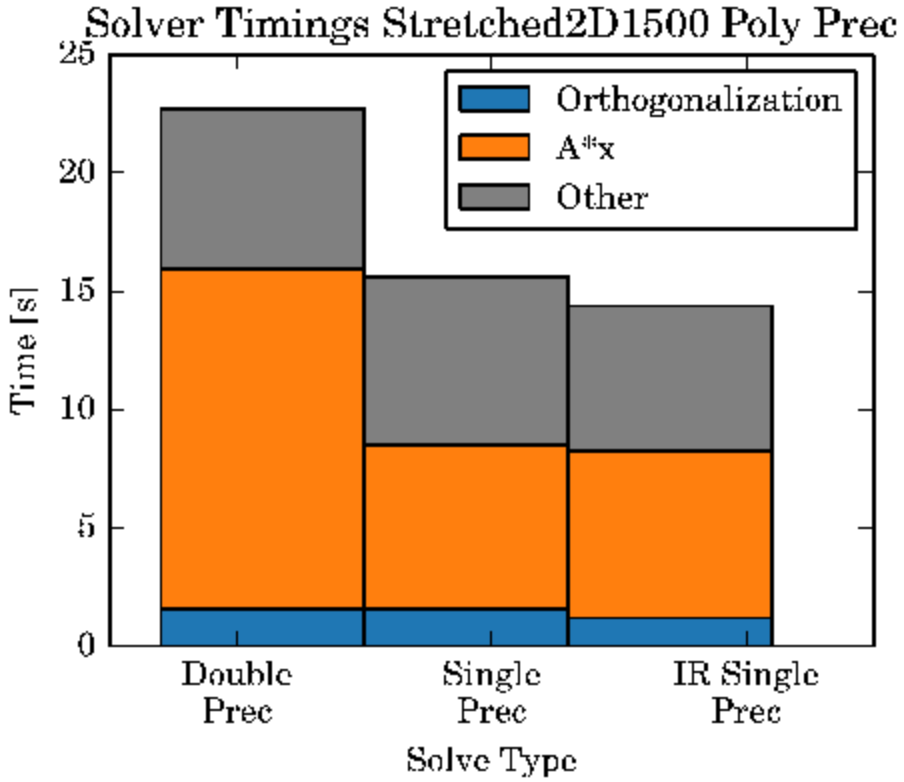




With a “high enough” polynomial degree, the iteration count grows close to linearly with respect to the number of grid points on the PDE domain.

Connections to ECP Multiprecision

Polynomial Preconditioning in Varied Precisions



Stretched2D: Laplacian on a stretched grid.

Double precision GMRES(50) with double precision preconditioner. GMRES(50)-IR with single precision preconditioner.

GMRES-IR gives consistent solve time improvement ~33% over double precision with preconditioning.

Polynomial preconditioning shifts main expense to SpMV rather than dense orthogonalization kernels.

Future for polynomial preconditioning in Trilinos:

- **Short Term (by end of 2021):**
  - Performance testing on NVIDIA GPUs
  - Testing on AMD GPUs
- **Longer Term:**
  - Optimize for GPU performance
  - Test with large-scale applications

Future Improvements to Belos Linear Solvers Package in Trilinos:

Making Belos Efficient for GPUs:

- Belos currently uses Teuchos library for serial dense linear algebra; all the data must be stored on host.
- Requires extra data movement to and from GPU.
- Will translate Teuchos linear algebra -> Kokkos linear algebra for GPU performance and portability.
- Performance profiling for large-scale applications.

We expect that these changes will improve performance of all Trilinos iterative linear solvers on GPU systems!