Highly Parallel Smoothers for PSCToolkit on GPUs

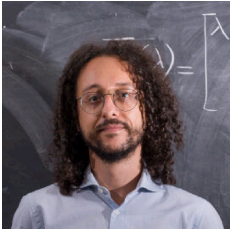
Pasqua D'Ambra
Institute for Applied Computing, National Research Council (IAC-CNR)
and CINI Lab on HPC-KTT

pasqua.dambra@cnr.it

2024 SIAM Conference on Applied Linear Algebra
Special Session on "Sparse Linear Systems at Large Scale"

May 16, 2024

Co-authors



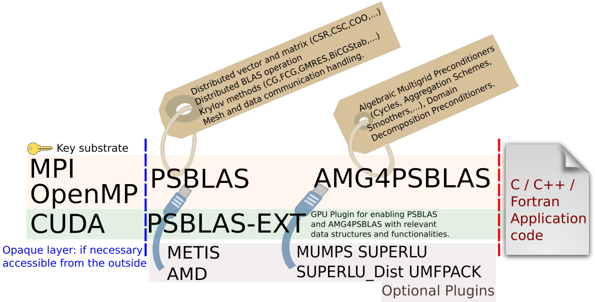
Fabio Durastante
University of Pisa

Salvatore Filippone
University of Rome "Tor-Vergata"

Stefano Massei
University of Pisa

Stephen Thomas
AMD

Parallel Sparse Computation Toolkit (psctoolkit.github.io)



recognized as "Excellent Science Innovation" by the EU Innovation Radar

Parallel Sparse Computation Toolkit (psctoolkit.github.io)

parallel sparse BLAS-1/2/3, Krylov solvers, algebraic interface with support for mesh handling and partitioning, effective handling of large index spaces for dealing with billions of dofs and of halo data exchange

Parallel Sparse Computation Toolkit (psctoolkit.github.io)

parallel sparse BLAS-1/2/3, Krylov solvers, algebraic interface with support for mesh handling and partitioning, effective handling of large index spaces for dealing with billions of dofs and of halo data exchange

additional matrix storage formats, interfaces to two external libraries for sparse BLAS-1/2 on GPUs and on multi-core CPUs

Parallel Sparse Computation Toolkit (psctoolkit.github.io)

parallel sparse BLAS-1/2/3, Krylov solvers, algebraic interface with support for mesh handling and partitioning, effective handling of large index spaces for dealing with billions of dofs and of halo data exchange

additional matrix storage formats, interfaces to two external libraries for sparse BLAS-1/2 on GPUs and on multi-core CPUs

parallel algebraic multigrid (AMG) preconditioners, specifically designed for extreme scalability

MultiGrid methods

V-cycle$\left(l, nlev, A^l, b^l, x^l\right)$

if $(l \neq nlev)$ then

$$x^l = x^l + (M^l)^{-1}(b^l - A^l x^l)$$

$$b^{l+1} = (P^l)^T\,(b^l - A^l x^l)$$

$$x^{l+1} = \text{V-cycle}\left(l+1, A^{l+1}, b^{l+1}, 0\right)$$

$$x^l = x^l + P^l x^{l+1}$$

$$x^l = x^l + (M^l)^{-T}(b^l - A^l x^l)$$

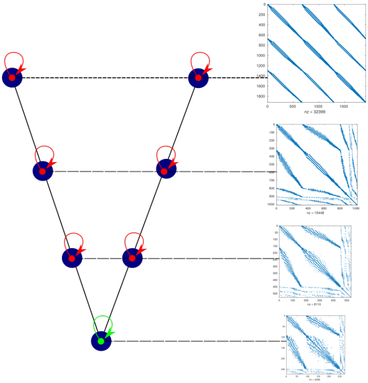
else

$$x^l = \left(A^l\right)^{-1} b^l$$

endif

return x^l

end



Smoother

$$M^l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}$$

"damping high frequencies"

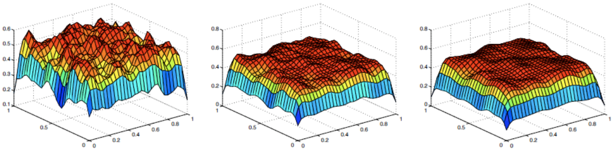
Prolongator

$$P^l : \mathbb{R}^{n_{l+1}} \rightarrow \mathbb{R}^{n_l}$$

"transferring low frequencies"

Algebraic MultiGrid (Brandt, McCormick and Ruge, 1984)

Algebraic MultiGrid methods do not explicitly use the (eventual) problem geometry but rely only on matrix entries to generate coarse-grids by using characterizations of *algebraic smoothness*



Key issue

errors not reduced by the (chosen) smoother (algebraic smoothness)

$$(Aw)_i = r_i \approx 0 \Longrightarrow w_{i+1} \approx w_i$$

have to be well represented on the coarse grid and well interpolated back $\mathbf{w} = (w_i) \in \mathcal{R}ange(P^l)$

MultiGrid Convergence

Theorem (McCormick 1985, Vassilevski 2008)

If M^l is a contraction at each level l, i.e., $\|I - (M^l)^{-1}A^l\|_{A^l} < 1$, the V-cycle preconditioner B defined as the multiplicative composition of the iteration matrix:

$$I - (B^l)^{-1}A^l = (I - (M^l)^{-T}A^l)(I - P^l((P^l)^T A^l P^l)^{-1}(P^l)^T A^l)(I - (M^l)^{-1}A^l)$$

has the following error bound:

$$\|E\|_A^2 = \|I - B^{-1}A\|_A^2 \leq 1 - \frac{1}{C} \quad \textit{with}$$

$$C = max_l C^l$$

where $C^l = sup_{v \in \mathcal{R}ange(P^l)^{\perp_A} \setminus 0} \frac{\|v\|^2_{\tilde{M}^l}}{\|v\|^2_A} \geq 1$ *is the approximation constant and* $\tilde{M}^l = M^l(M^l + (M^l)^T - A^l)^{-1}(M^l)^T$ *is the symmetrized smoother.*

MultiGrid Convergence

Theorem (McCormick 1985, Vassilevski 2008)

If M^l is a contraction at each level l, i.e., $\|I - (M^l)^{-1}A^l\|_{A^l} < 1$, the V-cycle preconditioner B defined as the multiplicative composition of the iteration matrix:

$$I - (B^l)^{-1}A^l = (I - (M^l)^{-T}A^l)(I - P^l((P^l)^T A^l P^l)^{-1}(P^l)^T A^l)(I - (M^l)^{-1}A^l)$$

has the following error bound:

$$\|E\|_A^2 = \|I - B^{-1}A\|_A^2 \leq 1 - \frac{1}{C} \quad \textit{with}$$

$$C = max_l C^l$$

where $C^l = sup_{v \in \mathcal{R}ange(P^l)^{\perp_A} \setminus 0} \frac{\|v\|_{\tilde{M}^l}^2}{\|v\|_A^2} \geq 1$ *is the approximation constant and* $\tilde{M}^l = M^l(M^l + (M^l)^T - A^l)^{-1}(M^l)^T$ *is the symmetrized smoother.*

Optimal Convergence (independent of problem size and number of levels)

MultiGrid Convergence

Theorem (McCormick 1985, Vassilevski 2008)

If M^l is a contraction at each level l, i.e., $\|I - (M^l)^{-1}A^l\|_{A^l} < 1$, the V-cycle preconditioner B defined as the multiplicative composition of the iteration matrix:

$$I - (B^l)^{-1}A^l = (I - (M^l)^{-T}A^l)(I - P^l((P^l)^T A^l P^l)^{-1}(P^l)^T A^l)(I - (M^l)^{-1}A^l)$$

has the following error bound:

$$\|E\|_A^2 = \|I - B^{-1}A\|_A^2 \leq 1 - \frac{1}{C} \quad \textit{with}$$

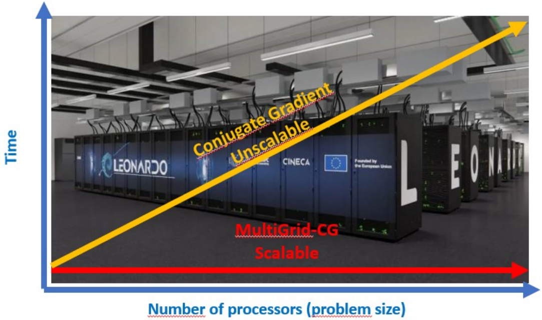
$$C = max_l C^l$$

where $C^l = sup_{v \in \mathcal{R}ange(P^l)^{\perp_A} \setminus 0} \frac{\|v\|_{\tilde{M}^l}^2}{\|v\|_A^2} \geq 1$ is the approximation constant and $\tilde{M}^l = M^l(M^l + (M^l)^T - A^l)^{-1}(M^l)^T$ is the symmetrized smoother.

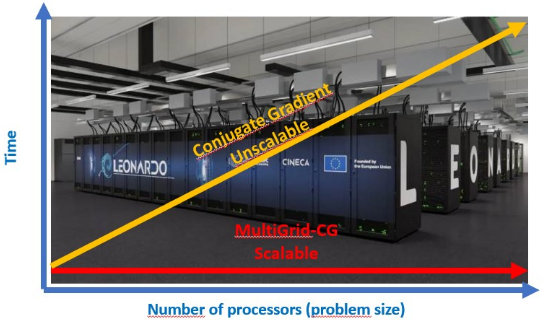
The smaller the approximation constant at each level the smaller the error!

Scalable (AMG) preconditioners

(courtesy of Rob Falgout)



(courtesy of Rob Falgout)

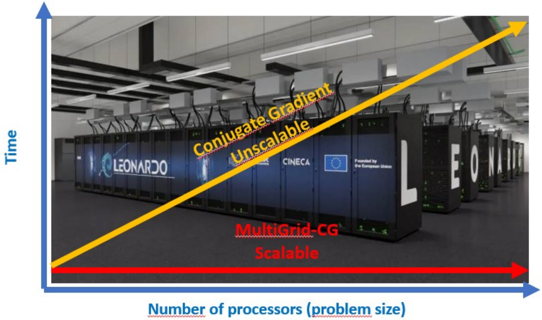


AMG can be optimal ($\mathcal{O}(n)$ flops) and hence have good scalability potential

Optimal complexity is not sufficient in parallel!

Scalable (AMG) preconditioners

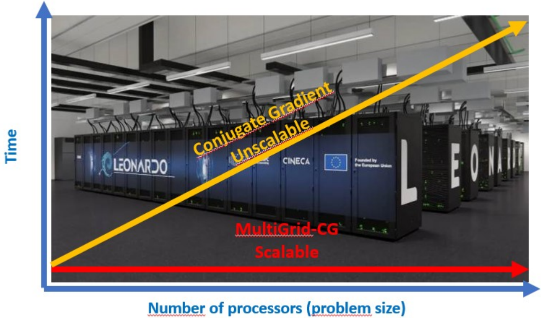
(courtesy of Rob Falgout)



- $\|E\|_A^2 < 1$ being independent of n (algorithmic scalability)
  true only for Laplacian and surroundings!

Scalable (AMG) preconditioners

(courtesy of Rob Falgout)



- $\|E\|_A^2 < 1$ being independent of n (algorithmic scalability) true only for Laplacian and surroundings!
- B should be composed of local actions essentially based on a "hierarchy" of sparse matrix-vector products (implementation scalability)

Parallel Smoothers

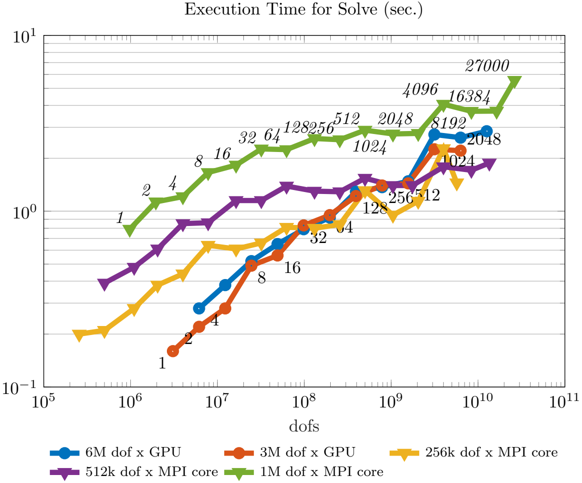
Let M be the spd (convergent) ℓ_1-Jacobi smoother:

$$G = (I - M^{-1}A), \qquad \begin{array}{l} M = \text{diag}(M_{ii})_{i=1,\ldots,n} \\ M_{ii} = a_{ii} + \sum_{j \neq i} |a_{ij}| \end{array}$$

Pros: simple and cheap to setup, only based on sparse matrix-vector product and local vector updates well suited for high-throughput SIMD processors

Cons: larger approximation constant than parallel (hybrid) Gauss-Seidel iterations (in our AMG setting the constant is larger of a factor about 4 for homogeneous 3D Poisson problem)

Some results on Piz Daint: MPI-HGS vs MPI/GPU-l1Jac



the hybrid approach permits up to ≈ 50% savings in solve time and energy consumption for 10 billion dofs

Parallel Smoothers

Polynomial accelerators (Adams et al. 2003, Kraus et al. 2012)

$$G = p_k((M^l)^{-1}A^l), \text{ for } p_k(x) \in \Pi_k[x]$$
$$\text{s.t. } p_k(0) = 1 \text{ and } |p_k(x)| < 1 \text{ for } 0 < x \leq 1$$

Key issue: choose polynomials to optimize V-cycle approximation constant

V-cycle Convergence & Polynomial Smoothers

Let $G = (I - (M^l)^{-1}A^l)$ be the error propagation matrix of an spd smoother M^l such that $\rho((M^l)^{-1}A^l) \leq 1$, let be $G = p_k((M^l)^{-1}A^l)$, for $p_k(x) \in \Pi_k[x]$ s.t. $p_k(0) = 1$ and $|p_k(x)| < 1$ for $0 < x \leq 1$.

Theorem (Lottes, 2023)

The V-cycle error propagation matrix has following bound:

$$\|E\|_A^2 \leq max_l \frac{C^l}{C^l + (\gamma_k^l)^{-1}},$$

where C^l is the approximation property constant at the level l and

$$\gamma_k^l = \sup_{0<\lambda\leq 1} \frac{\lambda p_k(\lambda)^2}{1 - p_k(\lambda)^2}$$

V-cycle Convergence & Polynomial Smoothers

Let $G = (I - (M^l)^{-1}A^l)$ be the error propagation matrix of an spd smoother M^l such that $\rho((M^l)^{-1}A^l) \leq 1$, let be $G = p_k((M^l)^{-1}A^l)$, for $p_k(x) \in \Pi_k[x]$ s.t. $p_k(0) = 1$ and $|p_k(x)| < 1$ for $0 < x \leq 1$.

Theorem (Lottes, 2023)

The V-cycle error propagation matrix has following bound:

$$\|E\|_A^2 \leq max_l \frac{C^l}{C^l + (\gamma_k^l)^{-1}},$$

where C^l is the approximation property constant at the level l and

$$\gamma_k^l = \sup_{0<\lambda\leq 1} \frac{\lambda p_k(\lambda)^2}{1 - p_k(\lambda)^2}$$

γ_k^l depends only on the polynomials

V-cycle Convergence & Polynomial Smoothers

Let $G = (I - (M^l)^{-1}A^l)$ be the error propagation matrix of an spd smoother M^l such that $\rho((M^l)^{-1}A^l) \leq 1$, let be $G = p_k((M^l)^{-1}A^l)$, for $p_k(x) \in \Pi_k[x]$ s.t. $p_k(0) = 1$ and $|p_k(x)| < 1$ for $0 < x \leq 1$.

Theorem (Lottes, 2023)

The V-cycle error propagation matrix has following bound:

$$\|E\|_A^2 \leq max_l \frac{C^l}{C^l + (\gamma_k^l)^{-1}},$$

where C^l is the approximation property constant at the level l and

$$\gamma_k^l = \sup_{0<\lambda\leq 1} \frac{\lambda p_k(\lambda)^2}{1 - p_k(\lambda)^2}$$

the smaller γ^l at each level the smaller the error!

γ bounds & 4th-kind Chebyshev polynomials

Minimax problem

$$\gamma_k := \min_{p_k(x)\in\Pi_k} \max_{x\in(0,1]} \left| \frac{x p_k(x)^2}{1 - p_k(x)^2} \right|$$

$$\text{s.t. } p_k(0) = 1 \text{ and } |p_k(x)| < 1 \text{ for } 0 < x \leq 1$$

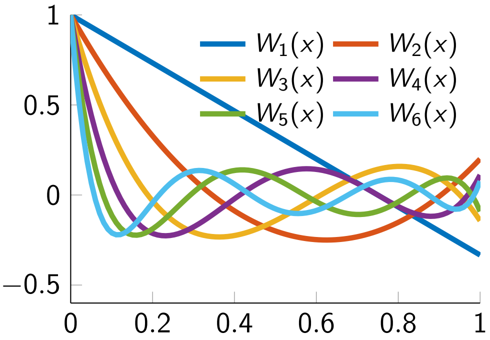
γ bounds & 4th-kind Chebyshev polynomials

Quasi-optimal 4th-kind Chebyshev polynomials (Lottes 2023)

$$W_k(x) = \frac{\sin(k + 1/2)\theta}{\sin(\theta/2)}, \; k \geq 0, \; x = \cos(\theta),$$

- $W_k(x) = argmin_{p_k(x)\in\Pi_k} \max_{x\in(0,1]} \left|xp_k(x)^2\right|$ and $\gamma_k = \frac{1}{4/3k(k+1)}$
- no information about spectra of matrices are needed
- can be applied as a simple 3-terms recurrence requiring sparse matrix-vector products and vector updates

γ bounds & 4th-kind Chebyshev polynomials



$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{z}^{(k)}$$

$$\begin{cases} \mathbf{z}^{(0)} = \mathbf{0} \\ \mathbf{z}^{(k)} = \dfrac{2k-3}{2k+1}\mathbf{z}^{(k-1)} + \dfrac{8k-4}{2k+1} M^{-1}(\mathbf{b} - A\mathbf{x}^{(k-1)}) \end{cases}$$

γ bounds & 4th-kind Chebyshev polynomials

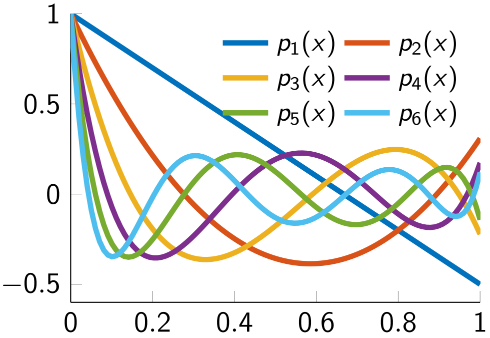
Approximate optimal 4th-kind Chebyshev polynomials (Lottes 2023)

$$p_k(x) = \sum_{j=0}^{k} \frac{\beta_{j,k} - \beta_{j+1,k}}{2j+1} W_j(1-2x),$$

$$\beta_{0,k} = 1, \quad \beta_{k+1,k} = 0 \ \ \forall k \geq 0.$$

- $p_k(x)$ improves the quasi-optimal bound: $\gamma_k \approx \frac{1}{4/\pi^2(2k+1)^2 - 2/3}$ for sufficiently large k
- coefficients $\beta_{j,k}$ can be computed by Newton's method applied to a system of non-linear eq.

γ bounds & 4th-kind Chebyshev polynomials



$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \beta_k \mathbf{z}^{(k)}$$

$$\begin{cases} \mathbf{z}^{(0)} = \mathbf{0} \\ \mathbf{z}^{(k)} = \dfrac{2k-3}{2k+1}\mathbf{z}^{(k-1)} + \dfrac{8k-4}{2k+1} M^{-1}(\mathbf{b} - A\mathbf{x}^{(k-1)}) \end{cases}$$

γ bounds & 1st-kind Chebyshev polynomials

Rewriting the minimax problem

$$\gamma_k = \min_{p_k(x)\in\Pi_k} \max_{x\in(0,1]} x \left| 1 - \frac{1}{1 - p_k(x)^2} \right|,$$

s.t. $p_k(0) = 1$ and $|p_k(x)| < 1$ for $0 < x \leq 1$

γ bounds & 1st-kind Chebyshev polynomials

Rewriting the minimax problem

$$\gamma_k = \min_{p_k(x)\in\Pi_k} \max_{x\in(0,1]} x \left| 1 - \frac{1}{1-p_k(x)^2} \right|,$$

$$\text{s.t. } p_k(0) = 1 \text{ and } |p_k(x)| < 1 \text{ for } 0 < x \leq 1$$

Quasi-Optimal 1st-kind Chebyshev polynomials

$$\tau_k(x) = \frac{1}{2}\left[(x + \sqrt{x^2-1})^k + (x - \sqrt{x^2-1})^k \right]$$

- $\tau_k(x)$ provides the optimal solution in the interval $[a_k, 1]$, for any $a_k \in (0, 1)$
- optimal values of a_k and corresponding γ_k can be numerically obtained by solving a scalar non-linear equation
- can be applied as a simple 3-terms recurrence requiring sparse matrix-vector products and vector updates

γ bounds & 1st-kind Chebyshev polynomials

Theorem (PD, Durastante, Massei, Filippone, Thomas, 2024)

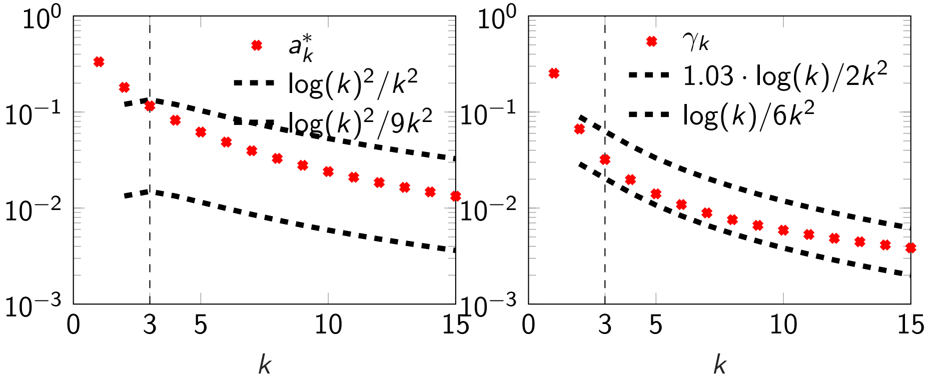
Let $a_k^* \in (0,1)$ *be such that*

$$\max_{x\in(0,1]} x \left| 1 - \frac{1}{1 - \tau_k^{[a_k^*,1]}(x)^2} \right| = \gamma_k.$$

If $k \geq 3$*, then*

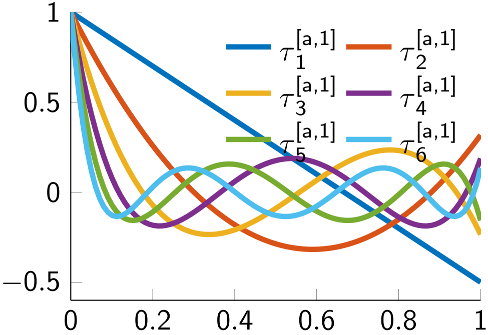
$$\frac{\log(k)^2}{9k^2} \leq a_k^* \leq \frac{\log(k)^2}{k^2}, \qquad \textit{and} \qquad \frac{\log(k)}{6k^2} \leq \gamma_k \leq 1.03\frac{\log(k)}{2k^2}.$$

γ bounds & 1st-kind Chebyshev polynomials



Bounds and computed quantities for the optimal parameters a_k^* for the 1st-kind Chebyshev polynomials and the smoothing constant γ_k, $k = 1, \dots, 15$

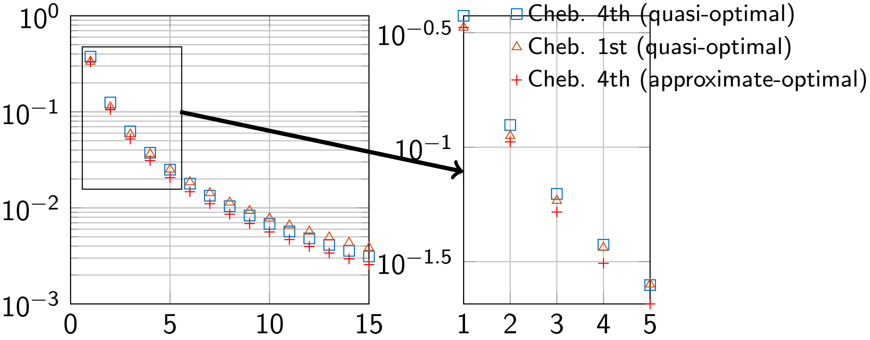
γ bounds & 1st-kind Chebyshev polynomials



$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{z}^{(k-1)}$$

$$\begin{cases} \mathbf{z}^{(0)} = \frac{2}{1+a^*} M^{-1}(\mathbf{b} - A\mathbf{x}^{(0)}) \ \rho_0 = \frac{1-a^*}{1+a^*} \\ \rho_k = \left(\frac{2(1+a^*)}{1-a^*} - \rho_{k-1} \right)^{-1} \\ \mathbf{z}^{(k)} = \rho_k \rho_{k-1} \mathbf{z}^{(k-1)} + \dfrac{4\rho_k}{(1-a^*)} M^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}) \end{cases}$$

γ bounds



if $k \leq 5$, the quasi-optimal 1st-kind Cheb.

are better than (or comparable with)

the quasi-optimal 4th-kind Cheb.

Test case: Poisson equation

$$-\Delta u = 1 \text{ on unit cube, with DBC}$$

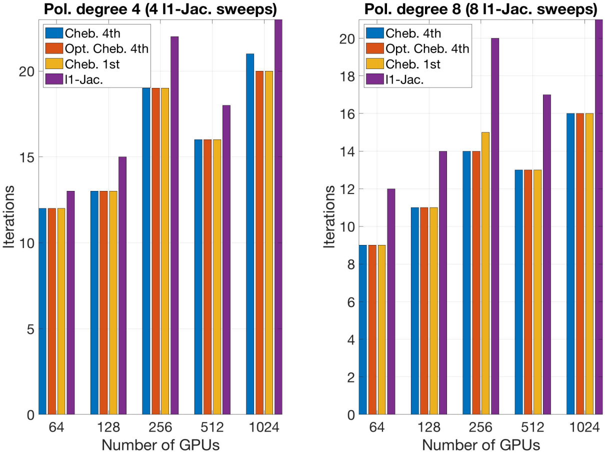
Solver/preconditioner settings

- AMG as preconditioner of CG, stopped when $\|\mathbf{r}^k\|_2/\|\mathbf{b}\|_2 \leq 10^{-7}$, or $itmax = 500$

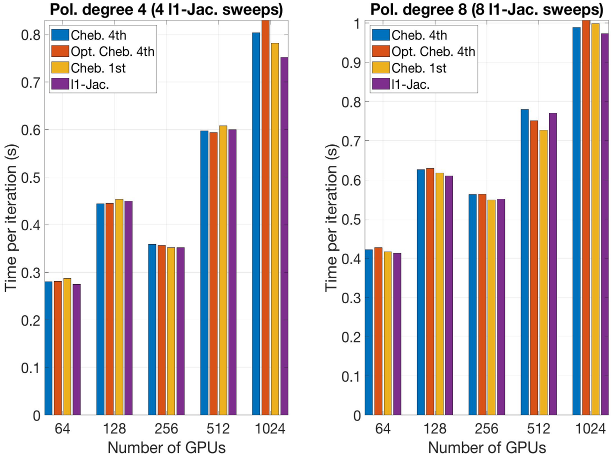
  VSMATCH V-cycle for matching-based coarsening with aggregates of max size 8, smoothed prolongators
- coarsest matrix size $n_c \leq 200np$, with np number of tasks (GPUs)
- ℓ_1-Jacobi iterations, quasi-opt. 4th-kind Cheb., approximate opt. 4th-kind Chebyshev and quasi opt. 1st-kind Cheb. accelerations; 30 iterations of ℓ_1-Jacobi at the coarsest level.

Platform: Leonardo booster, ranked 6th in the last Top500 list (BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband)

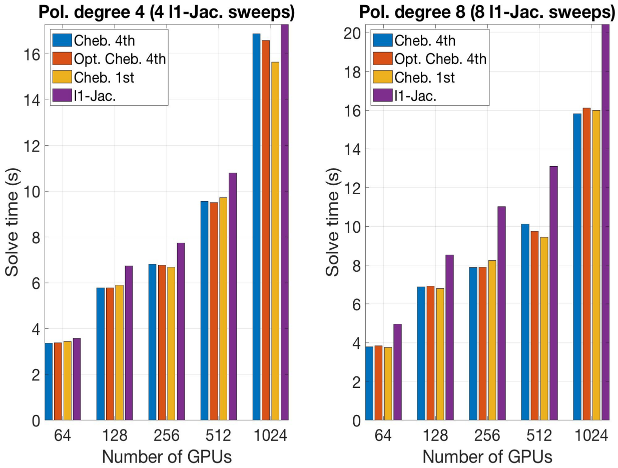
Results:Iterations



Results:Time per Iteration



Results: Solve Time



Concluding remarks and work in progress

- PSCToolkit is a software project addressing extreme scalability for scientific computing on heterogeneous architectures
- new GPU supports for polynomial smoothers have been included in PSCToolkit and demonstrate benefits in solving benchmark systems up to 6 billion dofs on up to 1024 GPUs of the Leonardo supercomputer
- applications to systems arising from CFD for sustainable energy are work in progress (Fabio's talk, @MS13, last Monday morning)

Thanks for Your Attention

This work was performed with support of the European Union under HORIZON-EUROHPC-JU-2023-COE-01 agreement N. 101144014 — EoCoE-III and of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR - Next Generation EU (NGEU)