Improving Parallel Scalability of Spectral Element Method Pressure Poisson Preconditioners (With 4th-kind Chebyshev Smoothing)

Malachi Phillips[1] Stefan Kerkemeier[2] Paul Fischer[1,2,3]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign

[2]Mathematics and Computer Science, Argonne National Laboratory, Lemont, IL 60439

[3]Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign

Big Picture

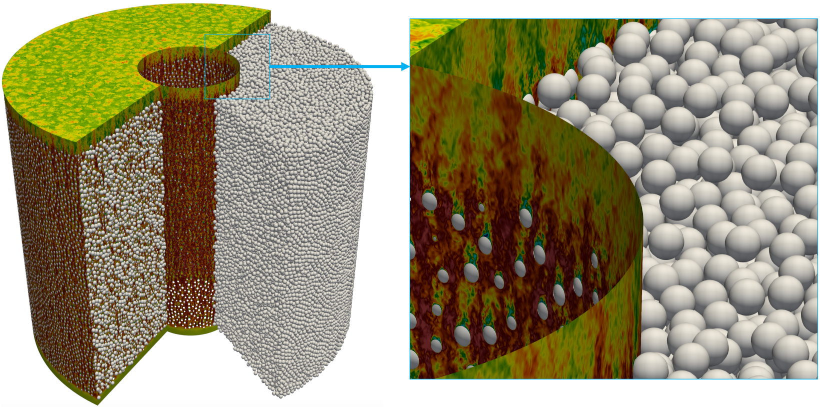
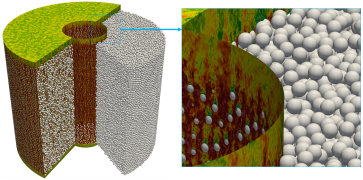


Figure: 352K pebble geometry from[1], $n = 51B$, $P = 27648$ V100s on Summit.

[1] Min, Lan, Fischer, Merzari, Kerkemeier, Phillips, Rathnayake, Novak, Gaston, Chalmers, et al., "Optimization of full-core reactor simulations on summit", 2022.

Motivation



nekRS Timing Breakdown: n=51B, 2000 Steps				
	pre-tuning		post-tuning	
Operation	time (s)	%	time (s)	%
computation	1.19+03	100	5.47+02	100
advection	5.82+01	5	4.49+01	8
viscous update	5.38+01	5	5.98+01	11
pressure solve	1.08+03	90	4.39+02	80
precond.	9.29+02	78	3.67+02	67
coarse grid	5.40+02	45	6.04+01	11
projection	6.78+00	1	1.21+01	2
dotp	4.92+01	4	1.92+01	4

Table: Runtime statistics for the 352K pebble geometry of fig. 1 on $P = 27648$ V100s on Summit.

Poisson

Solve series of Poisson problems using SE discretization:

$$-\nabla^2 \tilde{u} = \tilde{f} \text{ for } \tilde{u}, \tilde{f} \in \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}. \tag{1}$$

Weak formulation: *find* $u^m(\mathbf{x}) \in X_0^N \subset \mathcal{H}_0^1$ *such that*

$$\int_\Omega \nabla v \cdot \nabla u^m \, dV = \int_\Omega v \, f^m \, dV \quad \forall \, v \in X_0^N, \tag{2}$$

$$X_0^N = \text{span}\{\phi_j(\mathbf{x})\} \tag{3}$$

Discrete problem – solve $A\underline{u}^m = \underline{b}^m$:

$$a_{ij} = \int_\Omega \nabla \phi_i \cdot \nabla \phi_j \, dV. \tag{4}$$

How to solve? Multigrid.

Polynomial Smoothers

Polynomial smoother $G_j = (I - \omega S_j A_j)^k$ is k steps of simple smoothing iteration:

$$\left(\underline{x}_{i+1}\right)_j = \left(\underline{x}_i\right)_j + \omega S_j(\underline{b}_j - A_j \left(\underline{x}_i\right)_j). \tag{5}$$

Can we do better?

$$\min_{p_k \in \mathbb{P}_k, p_k(0)=1} \max_{\lambda \in [\lambda_{min}, \lambda_{max}]} |p(t)|. \tag{6}$$

1st-kind Chebyshev Smoother[2,3]

Minimax solution:

$$\hat{T}_k(\lambda) = \frac{1}{\sigma_k} T_k\left(\frac{\theta - \lambda}{\delta}\right) \text{ with } \sigma_k := T_k\left(\frac{\theta}{\delta}\right). \quad (7)$$

T_k are Chebyshev polynomials of the 1st-kind:

$$\begin{aligned} T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x) \\ T_0(x) &= 1 \\ T_1(x) &= x. \end{aligned} \quad (8)$$

θ is the midpoint of the interval $[\lambda_{min}, \lambda_{max}]$:

$$\theta = \frac{\lambda_{min} + \lambda_{max}}{2}.$$

δ is the mid-width of the interval:

$$\delta = \frac{\lambda_{max} - \lambda_{min}}{2}.$$

[2] Adams, Brezina, Hu, and Tuminaro, "Parallel multigrid smoothing: polynomial versus Gauss–Seidel", 2003.

[3] Kronbichler and Ljungkvist, "Multigrid for matrix-free high-order finite element computations on graphics processors", 2019.

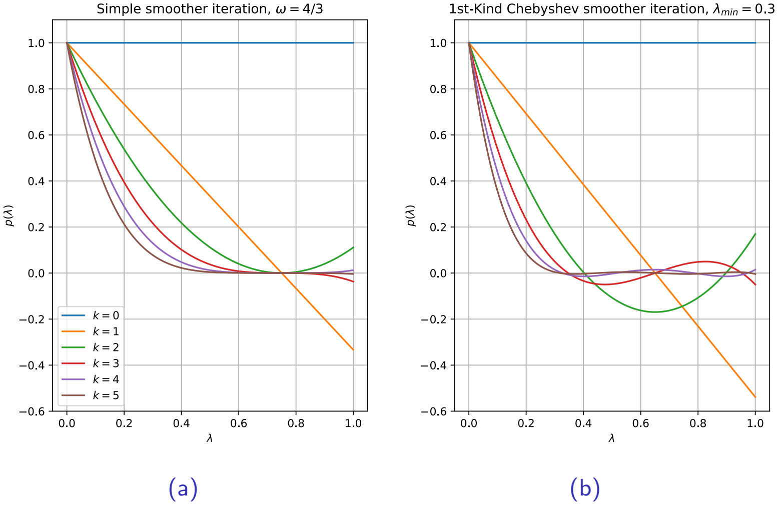


Figure: Smoother polynomials for the simple smoother (a) and the 1st-kind Chebyshev smoother (b).

Algorithm Chebyshev smoother, 1st-kind

$$\theta = \frac{1}{2}(\lambda_{max} + \lambda_{min}),\ \delta = \frac{1}{2}(\lambda_{max} - \lambda_{min}),\ \sigma = \frac{\theta}{\delta},\ \rho_0 = \frac{1}{\sigma}$$

$\underline{x}_0 = \underline{x}, \underline{r}_0 = S(\underline{b} - A\underline{x}_0),\ \underline{d}_0 = \frac{1}{\theta}\underline{r}_0$

for $i = 1, \ldots, k-1$ **do**

$\quad \underline{x}_i = \underline{x}_{i-1} + \underline{d}_{i-1}$

$\quad \underline{r}_i = \underline{r}_{i-1} - SA\underline{d}_{i-1},\ \rho_i = \frac{1}{2\sigma - \rho_{i-1}}$

$\quad \underline{d}_i = \rho_i \rho_{i-1} \underline{d}_{i-1} + \frac{2\rho_i}{\delta}\underline{r}_i$

end for

$\underline{x}_k = \underline{x}_{k-1} + \underline{d}_{k-1}$

return $\underline{x}_k$

4th-kind Chebyshev Smoother[5]

wlog $\rho(SA) = 1$.
Two-level Hackbusch bound[4]:

$$\begin{aligned} \|E_{\searrow}\|_A &= \left\|(I - PA_c^{-1}P^TA)G_k\right\|_A \\ &\le C^{1/2} \sup_{0<\lambda\le 1} \lambda^{1/2}|p_k(\lambda)|. \end{aligned} \tag{9}$$

What p_k minimizes this error bound?

[4] Hackbusch, "Multi-grid convergence theory", 1982.
[5] Lottes, "Optimal polynomial smoothers for multigrid V-cycles", 2022.

Weighted minimax solution:

$$p_k(\lambda) = \frac{1}{2k+1} W_k(1 - 2\lambda), \tag{10}$$

W_k are 4th-kind Chebyshev polynomial[6]:

$$\begin{aligned} W_n(x) &= 2xW_{n-1}(x) - W_{n-2}(x) \\ W_0(x) &= 1 \\ W_1(x) &= 2x + 1. \end{aligned} \tag{11}$$

Can we do *even* better? What about the multi-level case?

[6] Mason, "Chebyshev polynomials of the second, third and fourth kinds in approximation, indefinite integration, and integral transforms", 1993.

Lemma[7]:

Let the smoother iteration (on each level j) be given by

$$G_j = p_{k_j}(S_j A_j)$$

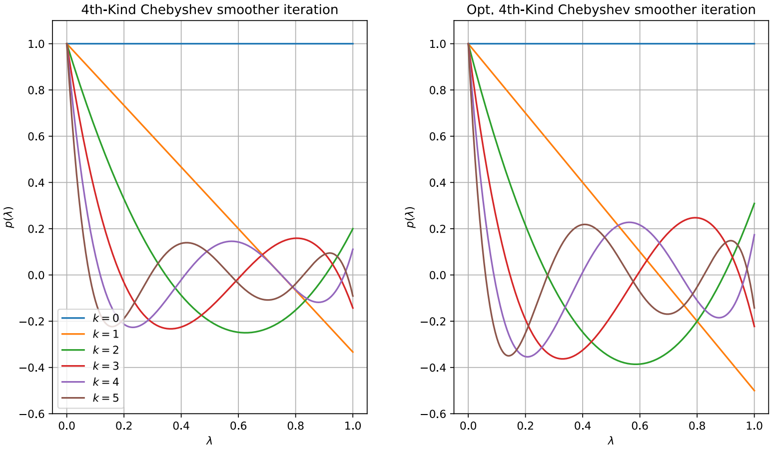
where S_j is SPD, $\rho(S_j A_j) = 1$, and $p_{k_j}(x)$ is a k_j-order polynomial satisfying $p_{k_j}(0) = 1$ and $|p_{k_j}(x)| < 1$ for $0 < x \leq 1$, possibly different on each level. Then the V-cycle contraction factor

$$\|E_{\nearrow}\|_A^2 \leq \max_{j \in 0,\ldots,\ell-1} \frac{C_j}{C_j + \gamma_j^{-1}} \tag{12}$$

where C_j is the approximation property constant for level j, and

$$\gamma_j = \sup_{0<\lambda\leq 1} \frac{\lambda\, p_{k_j}(\lambda)^2}{1 - p_{k_j}(\lambda)^2}. \tag{13}$$

[7] Lottes, "Optimal polynomial smoothers for multigrid V-cycles", 2022.



(a) (b)

Figure: 4th-kind Chebyshev smoother (a) and the 4th-kind Chebyshev smoother optimized with respect to previous error bound (b).

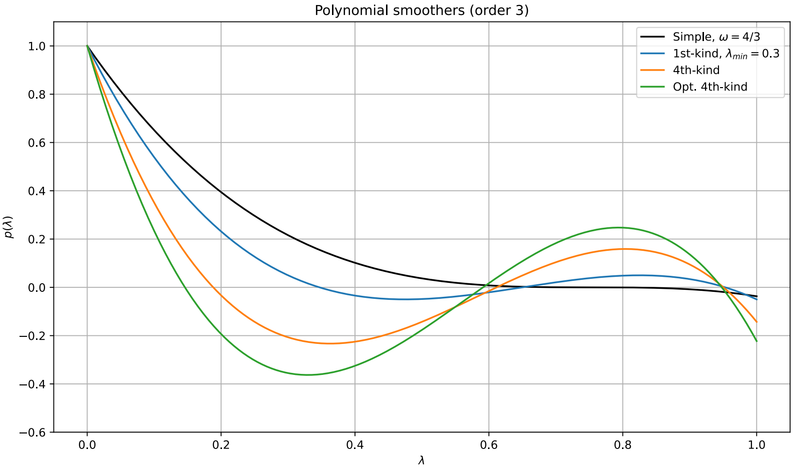


Figure: All smoother polynomials at order 3.

$$\|E_{\nearrow}\|_A^2 \leq \frac{C}{C + \gamma^{-1}}$$

Polynomial Smoother	$\gamma^{-1}, k \to \infty$
Simple multi-sweep, damping	$2\omega k$
1st-kind Chebyshev, *fixed* λ_{min}	$2\sqrt{\frac{1}{\lambda_{min}}}k$
1st-kind Chebyshev, λ_{min}^* optimizes γ^{-1}	$2.38k^{1.78}$
4th-kind Chebyshev	$\frac{4}{3}k(k+1)$
4th-kind optimal Chebyshev	$\frac{4}{\pi^2}(2k+1)^2 - \frac{2}{3}$

Algorithm Chebyshev smoother, (Opt.) 4th-kind

$\underline{x}_0 = \underline{x}$, $\underline{r}_0 = \underline{b} - A\underline{x}_0$

$\underline{d}_0 = \frac{4}{3}\frac{1}{\lambda_{max}} S\underline{r}_0$

for $i = 1, \ldots, k-1$ **do**

  $\underline{x}_i = \underline{x}_{i-1} + \beta_i \underline{d}_{i-1}, \underline{r}_i = \underline{r}_{i-1} - A\underline{d}_{i-1}$

  $\underline{d}_i = \frac{2i-1}{2i+3}\underline{d}_{i-1} + \frac{8i+4}{2i+3}\frac{1}{\lambda_{max}} S\underline{r}_i$

end for

$\underline{x}_k = \underline{x}_{k-1} + \beta_k \underline{d}_{k-1}$

return $\underline{x}_k$

Used for *p*-multigrid (pMG) and algebraic multigrid (AMG) [8].

[8] AMG implementations available for:

- hypre/boomerAMG: `https://github.com/MalachiTimothyPhillips/hypre/tree/fourth-kind-chebyshev-polynomials`
- Trilinos/MueLu: `https://github.com/MalachiTimothyPhillips/Trilinos/tree/optimal-chebyshev-polynomials`

To Post-smooth, or Not to Post-smooth?

- Given $2k$ smoother passes, what order m pre-smoothing, n post-smoothing should be used, $m + n = 2k$?
- Answer using error bound from previous Lemma:

$$\underset{m,n,m+n=2k}{\arg\max}\ C\left(\gamma^{-1}(m) + \gamma^{-1}(n)\right) + \gamma^{-1}(m)\cdot\gamma^{-1}(n) \quad (14)$$

- Check solutions for $k < 50$ in SymPy[9].
- With few exceptions, either $m = n = k$ (symmetric smoothing) or $m = 2k$, $n = 0$ (no post-smoothing) is optimal[10].
- *When* to use which?

[9] Meurer, Smith, Paprocki, Čertík, Kirpichev, Rocklin, Kumar, Ivanov, Moore, Singh, et al., "SymPy: symbolic computing in Python", 2017.

[10] Phillips and Fischer, "Optimal Chebyshev Smoothers and One-sided V-cycles", 2022.

To Post-smooth, or Not to Post-smooth?

Polynomial Smoother	When to *omit* post smoothing?
Simple multi-sweep, damping	$C > \frac{(4k - \log(4k))^2}{\log(2k)}$
1st-kind Chebyshev, *fixed* $\lambda_{min} = 0.1$	$k > 3, \ C \gtrapprox 1.55 e^{1.45k}$
1st-kind Chebyshev, λ^*_{min} optimizes γ^{-1}	$C \gtrapprox 2.38 k^{1.78}$
4th-kind Chebyshev	$C > \frac{2(k+1)^2}{3}$
4th-kind optimal Chebyshev	$C > \frac{2\left(6(2k+1)^2 - \pi^2\right)^2}{3\pi^2\left(-12(2k+1)^2 + 6(4k+1)^2 + \pi^2\right)}$

C is the multigrid approximation property constant.
Roughly $\kappa(SA)$ restricted to the A-orthogonal complement of the coarse-grid space.

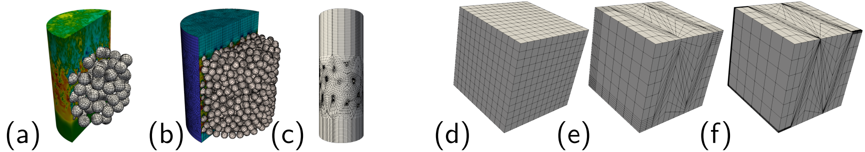
nekRS Pressure Poisson Results

Solver parameter study in nekRS:

- Consider 3 smoothers for Chebyshev-acceleration:
  - Jacobi
  - Additive Schwarz Method (ASM)
  - Restrictive Additive Schwarz (RAS)
- Consider 4 types of polynomial acceleration schemes:
  - 1st-kind Chebyshev
  - 1st-kind Chebyshev, λ_{min} optimized via random RHS
  - 4th-kind Chebyshev[11]
  - Optimized 4th-kind Chebyshev
- Vary k from 1 to 6
- Consider 2 different V-cycle approaches:
  - (k, k) symmetric V-cycle
  - $(2k, 0)$ V-cycle (no post-smoothing)

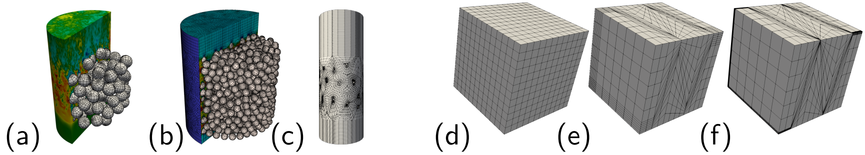
[11] Lottes, "Optimal polynomial smoothers for multigrid V-cycles", 2022.

nekRS Pressure Poisson Results



Case Name	E	N	n
146 pebble (fig. 8a)	62K	7	21M
1568 pebble (fig. 8b)	524K	7	180M
67 pebble (fig. 8c)	122K	7	42M
Kershaw ($\varepsilon = 1.0$) (fig. 8d)	47K	7	16M
Kershaw ($\varepsilon = 0.3$) (fig. 8e)	47K	7	16M
Kershaw ($\varepsilon = 0.05$) (fig. 8f)	47K	7	16M

Table: Discretization and fastest solver for the NS examples.



Case	P	Fastest Solver	T_S	Iter.	$\frac{T_D}{T_S}$	$\frac{(T_{crs})_D}{(T_{crs})_S}$
(a) pb146	6	4^{th}_{opt}-Cheb, RAS(4,4)	0.15	5.3	1.17	1.21
(b) pb67	18	4^{th}_{opt}-Cheb, RAS(12,0)	0.37	12.5	1.81	2.41
(c) pb1568	72	4^{th}-Cheb, ASM(12,0)	0.14	3	1.27	2.13
(d) K. 1	6	1^{st}-Cheb, λ^{opt}_{min}, RAS(2,2)	0.09	8	1.75	1.13
(e) K. 0.3	6	1^{st}-Cheb, λ^{opt}_{min}, RAS(5,5)	0.67	28	1.35	1.79
(f) K. 0.05	6	4^{th}_{opt}-Cheb, RAS(12,0)	2.40	88	1.75	2.31

Table: T_S: solution time of fastest solver. T_D solution time of nekRS default, 1^{st}-Cheb, ASM(3,3).

Questions?

- More details in pre-print: "Optimal Chebyshev Smoothers and One-sided V-cycles" `https://arxiv.org/abs/2210.03179`
- nekRS: `https://github.com/Nek5000/nekRS`
- 4th-kind Chebyshev implementations in popular AMG solvers:
  - hypre/boomerAMG: `https://github.com/MalachiTimothyPhillips/hypre/tree/fourth-kind-chebyshev-polynomials`
  - Trilinos/MueLu: `https://github.com/MalachiTimothyPhillips/Trilinos/tree/optimal-chebyshev-polynomials`

Supporting Materials

Operator Cost

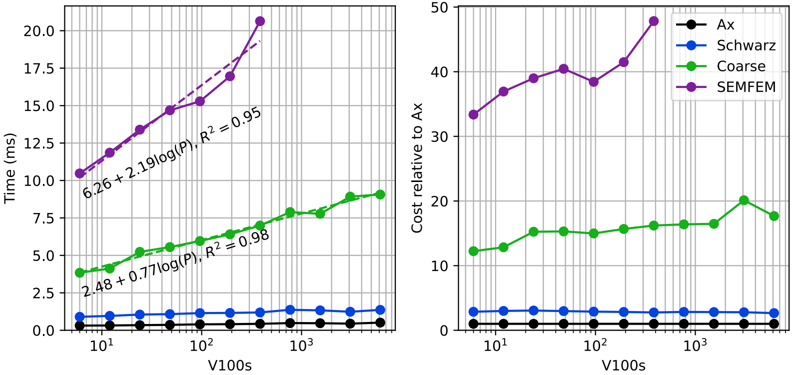


Figure: Weak scaling operator cost study for the Poisson solver for the Kershaw benchmark problem, $n/P = 2.67M$, $\varepsilon = 0.05$ [12].

[12] $\log(P)$ scaling of coarse grid solve, SEMFEM operator are expected, see Fischer, "Scaling limits for PDE-based simulation", 2015; Tufo and Fischer, "Fast parallel direct solvers for coarse grid problems", 2001.

Opt. 4th-kind Chebyshev polynomial for $\underline{e}_k = p_k(SA)\underline{e}_0$:

$$p_k(\lambda) = \sum_{i=0}^{k} \frac{\beta_i - \beta_{i+1}}{2i+1} W_i(1-2\lambda), \tag{15}$$

with $\beta_0 = 1$ and $\beta_{k+1} = 0$.

Multigrid approximation property constant

$$C_j := \left\| A_j^{-1} - P_{j+1}^j A_{j+1}^{-1} \left(P_{j+1}^j \right)^T \right\|_{A_j,S_j}^2$$
$$:= \sup_{\|\underline{f}\|_{S_j} \leq 1} \left\| \left(A_j^{-1} - P_{j+1}^j A_{j+1}^{-1} \left(P_{j+1}^j \right)^T \right) \underline{f} \right\|_{A_j}^2 . \qquad (16)$$

C_j is *roughly* $\kappa(S_j A_j)$ restricted to the A_j-orthogonal complement of the coarse $(j+1)$-space.

p-multigrid

- Matrix-free a must:
  - dofs: $n \sim Ep^3$
  - $\mathrm{nnz}(A) \sim O(Ep^6)$
  - $A\underline{x}$ cost $O(Ep^4) = O(np)$
- Drop Galerkin requirement for coarser levels
  - Each MG level has different polynomial order
    - e.g., $p = 7$, $p = 3$, $p = 1$

Schwarz-based Smoothers

SE-based additive Schwarz method (ASM) smoothers[13]:

$$S_{ASM}\underline{r} = \sum_{e=1}^{E} W_e R_e^T \bar{A}_e^{-1} R_e \underline{r} \qquad (17)$$

Or, **restrictive additive Schwarz** (RAS)[14]:

$$S_{RAS}\underline{r} = \sum_{e=1}^{E} \tilde{R}_e^T \bar{A}_e^{-1} R_e \underline{r}. \qquad (18)$$

Subdomains area extensions of element with $\bar{p}^3 = (p+3)^3$ dofs.

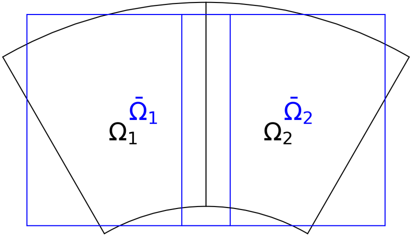
$$\bar{A}_e \neq R_e^T A_e R_e \text{ ruins } O(pn) \text{ complexity} \qquad (19)$$

---

[13] Lottes and Fischer, "Hybrid multigrid/Schwarz algorithms for the spectral element method", 2005; Loisel, Nabben, and Szyld, "On hybrid multigrid-Schwarz algorithms", 2008.

[14] Cai and Sarkis, "A restricted additive Schwarz preconditioner for general sparse linear systems", 1999.

How to apply $\bar{A}_e^{-1}$?

(a) 

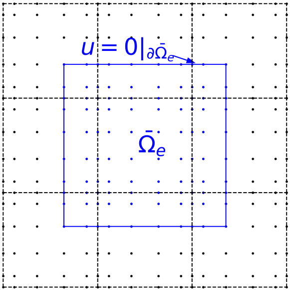
(b) 

Figure: Figure 6a Approximation of deformed elements Ω_1 and Ω_2 as box-shaped, overlapping subdomains $\bar{\Omega}_1$ and $\bar{\Omega}_2$. Figure 6b overlapping subdomain $\tilde{\Omega}_e$, constructed by overlapping two nodes in each spatial dimension and applying a homogeneous Dirichlet boundary condition on $\partial\bar{\Omega}_e$.

Fast Diagonalization Method

$$\bar{A}_e = B_z \otimes B_y \otimes A_x + B_z \otimes A_y \otimes B_x + A_z \otimes B_y \otimes B_x,$$

Generalized eigenvalue problem in x, y, z:

$$A_* \underline{s}_i = \lambda_i B_* \underline{s}_i$$

Fast, direct inverse:

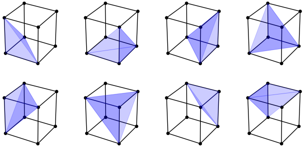
$$\bar{A}_e^{-1} = (S_z \otimes S_y \otimes S_x) D^{-1} (S_z^T \otimes S_y^T \otimes S_x^T),$$

$$D := I \otimes I \otimes \Lambda_x + I \otimes \Lambda_y \otimes I + \Lambda_z \otimes I \otimes I.$$

- Storage: $3E\bar{p}^2 + E\bar{p}^3$
- Complexity: $O(E\bar{p}^4)$
- **Use Schwarz-based smoothers in Chebyshev acceleration**

Preconditioning via Low-order Operator

- Precondition high-order system using low-order system
- Spectral equivalence $\kappa(A_F^{-1}A) \sim \pi^2/4$ in certain cases[15]
- Choice of finite element space matters[16]
  - Strong diagonal preconditioner, $M^{-1} = A_F^{-1} B_d B^{-1}$.
- Bello-Maldonado and Fischer[17] proposed one-per-vertex scheme
  - Use this with weak preconditioner, $M^{-1} = A_F^{-1}$.



[15] Orszag, "Spectral methods for problems in complex geometrics", 1979.

[16] Canuto, Gervasio, and Quarteroni, "Finite-element preconditioning of G-NI spectral methods", 2010.

[17] Bello-Maldonado and Fischer, "Scalable low-order finite element preconditioners for high-order spectral element Poisson solvers", 2019.

- How to apply A_F^{-1}? AMG!
  - PMIS coarsening
  - 0.25 strength threshold
  - Extended + i interpolation ($p_{max} = 4$)
  - L_1-Jacobi relaxation
  - One V-cycle for preconditioning
  - Smoothing on the coarsest level
- Use either AmgX[18] or boomerAMG[19] on GPU.
- Other approaches exist: Pazner, "Efficient low-order refined preconditioners for high-order matrix-free continuous and discontinuous Galerkin methods", 2020

[18] Naumov, Arsaev, Castonguay, Cohen, Demouth, Eaton, Layton, Markovskiy, Reguly, Sakharnykh, et al., "AmgX: A library for GPU accelerated algebraic multigrid and preconditioned iterative methods", 2015.

[19] Falgout, Li, Sjögreen, Wang, and Yang, "Porting hypre to heterogeneous computer architectures: Strategies and experiences", 2021.

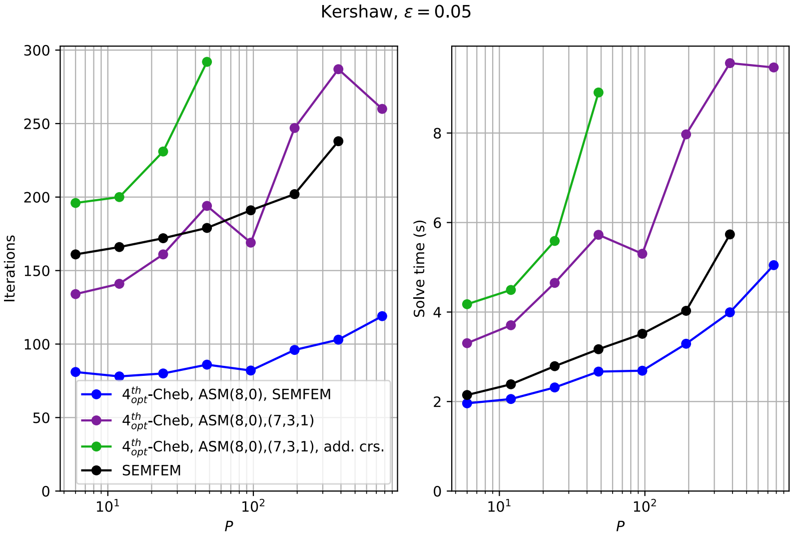


Figure: Weak scaling results for Kershaw, $\varepsilon = 0.05$.

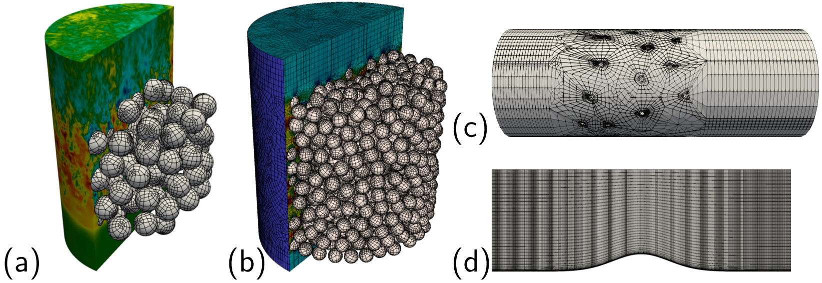


Figure: Navier-Stokes cases: pebble-beds with (a) 146, (b) 1568, and (c) 67 spheres; (d) Boeing speed bump.

Case Name	E	N	n	Fastest Solver
146 pebble (fig. 8a)	62K	7	21M	1^{st} Cheb-RAS(3,3),(7,5,3,1)
1568 pebble (fig. 8b)	524K	7	180M	SEMFEM
67 pebble (fig. 8c)	122K	7	42M	SEMFEM (**4X Speedup**)
Speed bump (fig. 8d)	885K	9	645M	1^{st} Cheb-RAS(3,3),(9,5,1)

Table: Discretization and fastest solver for the NS examples.

1st-kind Chebyshev

Correlation for λ^*_{min} with 1% relative error and 0.1% absolute error for $k \in [1, 50]$ is given by

$$\lambda^*_{min} \approx \frac{1.69}{k^{1.68} + 2.11k + 1.98}. \qquad (20)$$