Solver Development in nekRS: Optimal Chebyshev Smoothers and One-sided V-cycles

Malachi Phillips[1] 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

Poisson

- Poisson solve encompasses the majority of the solution time
- Spectral element (SE): E elements with polynomial degree p, $n \approx Ep^3$ unknowns and $\mathcal{O}(Ep^6)$ nonzeros
  - Matrix-free is a must: exploit tensor-product-sum factorization, $\mathcal{O}(Ep^4)$ cost to apply matrix-vector product[1]
  - Fast solvers require preconditioning: multigrid!

[1] Deville, Fischer, and Mund, *High-order methods for incompressible fluid flow*.

Multigrid

Algorithm 1 Multigrid V-cycle

$\underline{x} = \underline{x}_0 + \text{presmooth}(A, \underline{x}_0, \underline{b})$
$\underline{r} = \underline{b} - A\underline{x}$
$\underline{r}_C = P^T \underline{r}$
$\underline{e}_C = A_C^{-1} \underline{r}_C$ // solve coarse system/re-apply multigrid
$\underline{e} = P\underline{e}_C$
$\underline{x} = \underline{x} + \underline{e}$
$\underline{x} = \underline{x} + \text{postsmooth}(A, \underline{x}, \underline{b})$

1st Kind Chebyshev Smoother[2,3]

Algorithm 2 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 $i = 1, \dots, k-1$ **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 $\underline{x}_k$

[2] Adams et al., "Parallel multigrid smoothing: polynomial versus Gauss–Seidel".

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

4th Kind Chebyshev Smoother[4]

Algorithm 3 Chebyshev smoother, (Opt.) 4th kind

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

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

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

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

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

end for

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

return $\underline{x}_k$

- 4th kind: $\beta_i := 1$
- Opt. 4th kind: β_i from optimization
- No *ad-hoc* λ_{min} parameter, same complexity
- Could still optimize λ_{min} in 1st kind, multiple RHS

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

V-cycle Error Bounds

$$C := ||A^{-1} - PA_c^{-1}P^T||_{A,S}^2 := \sup_{||f||_S \le 1} ||(A^{-1} - PA_c^{-1}P^T)f||_A^2. \quad (1)$$

V-cycle contraction factor:

$$||E||_A^2 \le \frac{C}{C + \gamma^{-1}} \quad (2)$$

$$= V(C, k) \quad (3)$$

$$\gamma = \sup_{0<\lambda\le 1} \frac{\lambda\, p(\lambda)^2}{1 - p(\lambda)^2}. \quad (4)$$

e.g., 4th kind:

$$\gamma_4^{-1} = \frac{4}{3}k(k+1) \quad (5)$$

One-sided V-cycle

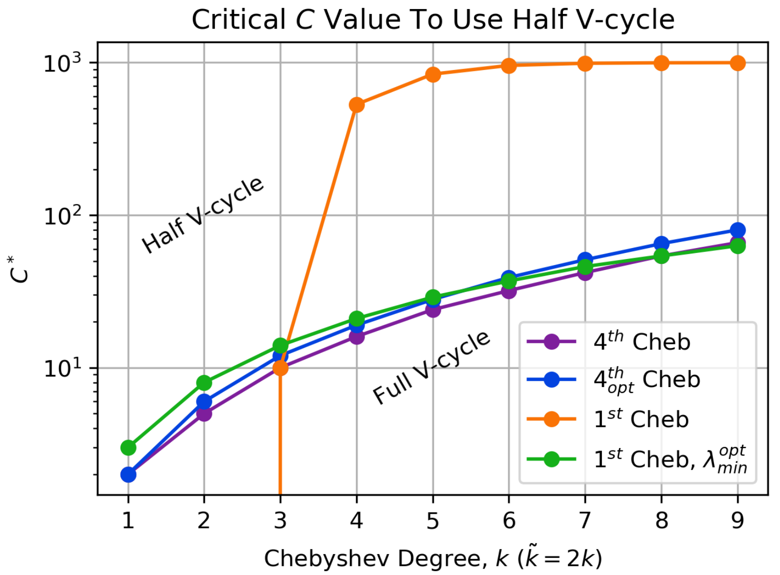
- Full, symmetric V-cycle contraction factor: $||E||_A^2 \leq V(C, k)$
- One-sided V-cycle contraction factor: $||E||_A \leq \sqrt{V(C, \tilde{k})}$
- Order k full V-cycle has same complexity as order $\tilde{k} = 2k$ one-sided V-cycle
- Better use one-sided (higher order) V-cycle at same cost?

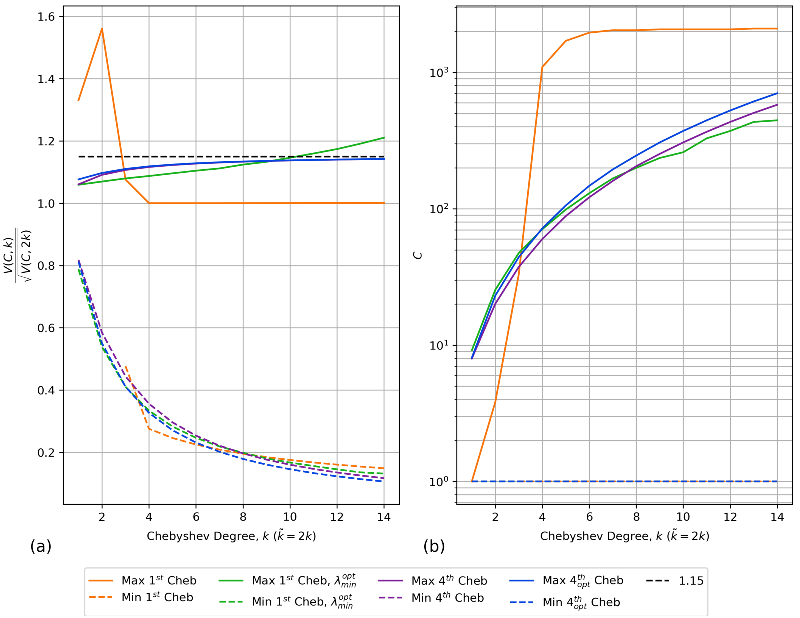
Question we'd like to answer:

$$||E||_A^2(C, k) \overset{?}{\geq} ||E||_A(C, 2k) \tag{6}$$

Easier question:

$$V(C, k) \overset{?}{\geq} \sqrt{V(C, 2k)} \tag{7}$$

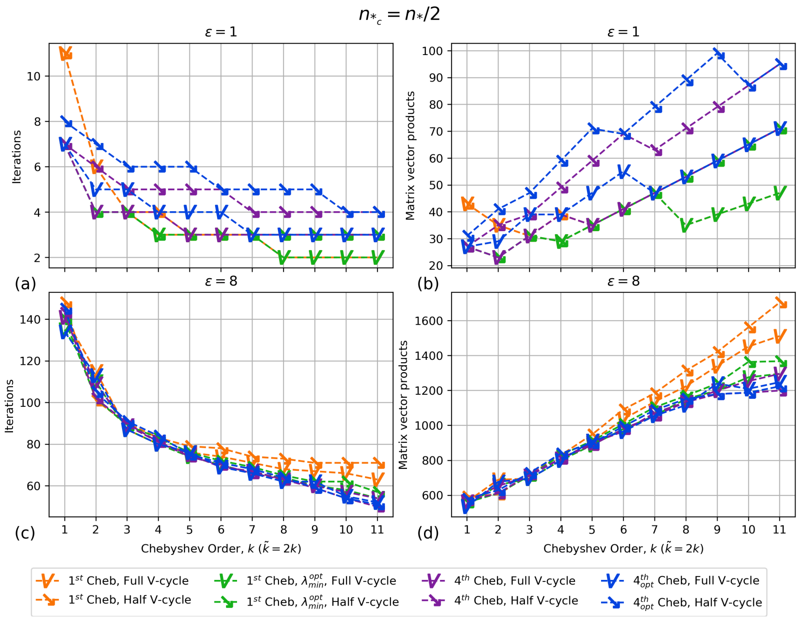

Critical C Value To Use Half V-cycle
10^3
10^2
10^1
C^*
Half V-cycle
Full V-cycle
1
2
3
4
5
6
7
8
9
Chebyshev Degree, k ($\tilde{k} = 2k$)
4^{th} Cheb
4^{th}_{opt} Cheb
1^{st} Cheb
1^{st} Cheb, λ^{opt}_{min}


1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
$\frac{V(C,k)}{\sqrt{V(C,2k)}}$
2
4
6
8
10
12
14
Chebyshev Degree, k ($\tilde{k} = 2k$)
(a)
10^3
10^2
10^1
10^0
C
Chebyshev Degree, k ($\tilde{k} = 2k$)
(b)
Max 1^{st} Cheb
Min 1^{st} Cheb
Max 1^{st} Cheb, λ_{min}^{opt}
Min 1^{st} Cheb, λ_{min}^{opt}
Max 4^{th} Cheb
Min 4^{th} Cheb
Max 4^{th}_{opt} Cheb
Min 4^{th}_{opt} Cheb
1.15

Finite Difference

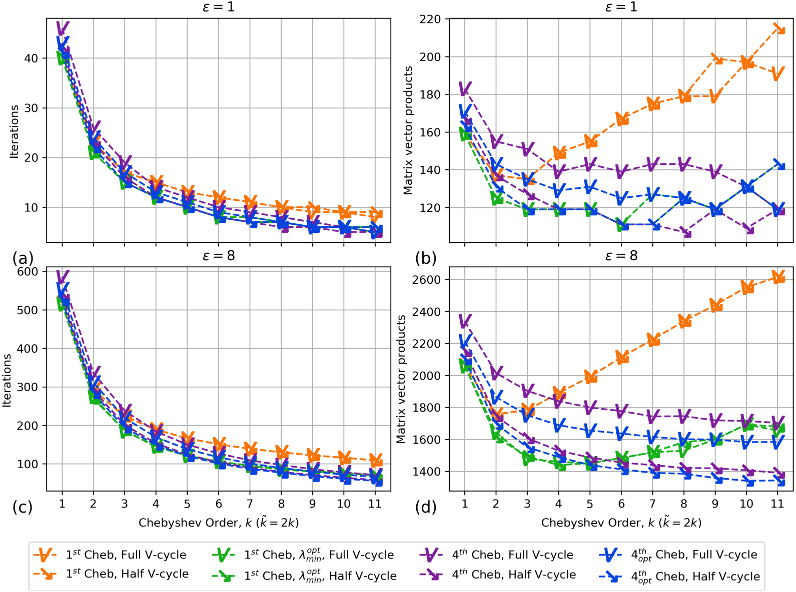
$$-\nabla^2 u = f \text{ for } u, f \in \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}. \tag{8}$$

- $\Omega := [0,1]^2$, $u|_{\partial\Omega} = 0$.
- $n = 128$, $n_x = n/\varepsilon$, $n_y = n\varepsilon$, $\varepsilon = 1, 8$.
- $(n_{x_c} + 1) \times (n_{y_c} + 1)$, $n_{*_c} = n_*/R$, $R = 2, 16$.
- $u(x,y) = \sin(3\pi x)\sin(4\pi y) + g$, g random satisfying $g|_{\partial\Omega} = 0$.
- Iterate until 10^{-6} relative residual tolerance, or 1,000 iterations.
- Use two-level geometric MG with Chebyshev-accelerated Jacobi smoothing as preconditioner for KSP.



$n_{*_c} = n_*/2$
$\varepsilon = 1$
Iterations
Matrix vector products
(a)
(b)
$\varepsilon = 8$
(c)
(d)
Chebyshev Order, k ($\tilde{k} = 2k$)
1^{st} Cheb, Full V-cycle
1^{st} Cheb, Half V-cycle
1^{st} Cheb, λ_{min}^{opt}, Full V-cycle
1^{st} Cheb, λ_{min}^{opt}, Half V-cycle
4^{th} Cheb, Full V-cycle
4^{th} Cheb, Half V-cycle
4_{opt}^{th} Cheb, Full V-cycle
4_{opt}^{th} Cheb, Half V-cycle

$n_{*_c} = n_*/16$

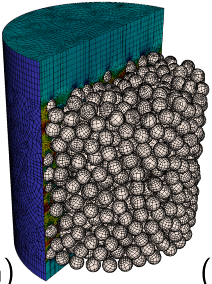


ε = 1
ε = 1
ε = 8
ε = 8
Iterations
Matrix vector products
(a)
(b)
(c)
(d)
Chebyshev Order, k (k̃ = 2k)
1st Cheb, Full V-cycle
1st Cheb, Half V-cycle
1st Cheb, λ opt min, Full V-cycle
1st Cheb, λ opt min, Half V-cycle
4th Cheb, Full V-cycle
4th Cheb, Half V-cycle
4th opt Cheb, Full V-cycle
4th opt Cheb, Half V-cycle

Finite Difference

ε	Factor	Lowest Matvec Cost Solver	Matvecs	Iterations
1	2	1^{st} Cheb, λ_{min}^{opt}, Jacobi(2), V	23	4
2	2	1^{st} Cheb, λ_{min}^{opt}, Jacobi(3), V	327	41
4	2	4^{th} Cheb, Jacobi(4), $\searrow$	317	53
8	2	4^{th}_{opt} Cheb, Jacobi(1), V	535	134
1	16	4^{th} Cheb, Jacobi(16), $\searrow$	107	6
2	16	4^{th}_{opt} Cheb, Jacobi(6), $\searrow$	383	48
4	16	1^{st} Cheb, λ_{min}^{opt}, Jacobi(4), V	699	70
8	16	4^{th}_{opt} Cheb, Jacobi(20), $\searrow$	1341	61

Navier-Stokes



(a)

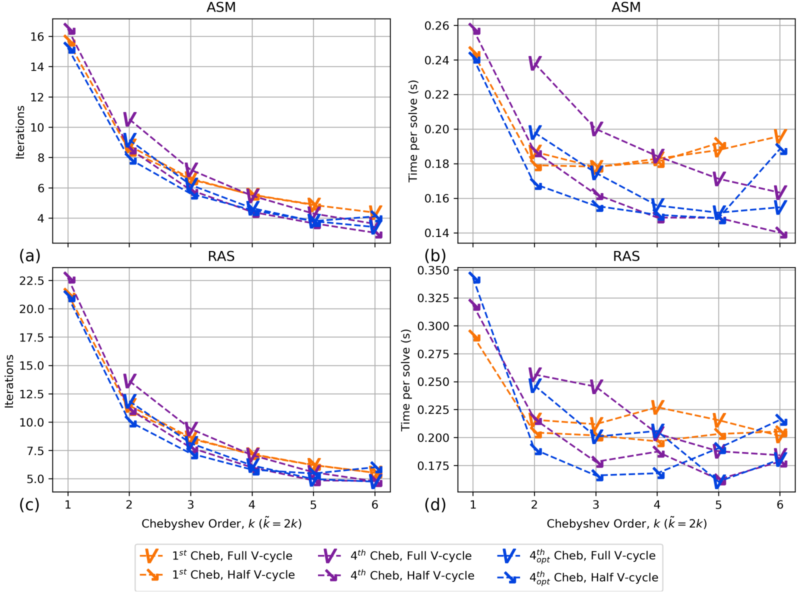
(b)

	pb1568 (a)	67 pebble (b)
E	524K	122 K
p	7	7
n	180M	42M
P	72	18
n/P	2.5M	2.3M

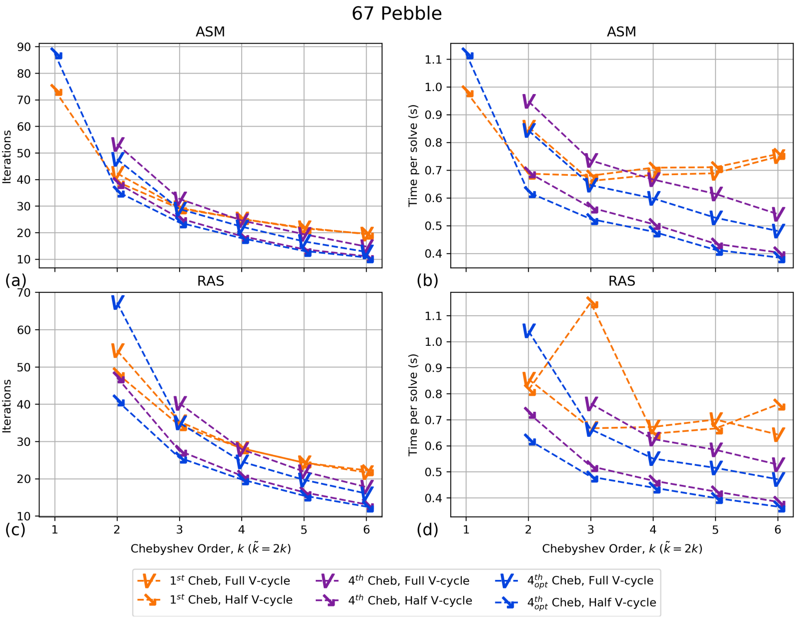
- 1568 pebble (a)[5], and 67 pebble (b)[6]
- Solve pressure Poisson using PGMRES(15) and solution projection[7].
- 10^{-4} residual tolerance, 2,000 timesteps

[5] Lan et al., "All-hex meshing strategies for densely packed spheres".
[6] Reger et al., "Large Eddy Simulation of a 67-Pebble Bed Experiment".
[7] Fischer, "Projection techniques for iterative solution of Ax= b with successive right-hand sides".

1568 Pebble



ASM
RAS
Iterations
Time per solve (s)
(a)
(b)
(c)
(d)
Chebyshev Order, k ($\tilde{k} = 2k$)
1^{st} Cheb, Full V-cycle
1^{st} Cheb, Half V-cycle
4^{th} Cheb, Full V-cycle
4^{th} Cheb, Half V-cycle
4^{th}_{opt} Cheb, Full V-cycle
4^{th}_{opt} Cheb, Half V-cycle


67 Pebble
ASM
RAS
Iterations
Time per solve (s)
(a)
(b)
(c)
(d)
Chebyshev Order, k ($\tilde{k} = 2k$)
1^{st} Cheb, Full V-cycle
1^{st} Cheb, Half V-cycle
4^{th} Cheb, Full V-cycle
4^{th} Cheb, Half V-cycle
4^{th}_{opt} Cheb, Full V-cycle
4^{th}_{opt} Cheb, Half V-cycle

Navier-Stokes Summary

	Case	Fastest Solver	T_S	Iterations	$\frac{T_D}{T_S}$
V	Kershaw($\varepsilon = 1$)	1^{st} Cheb, λ_{min}^{opt}, RAS(2)	0.09	8	1.75
	Kershaw($\varepsilon = 0.3$)	1^{st} Cheb, λ_{min}^{opt}, RAS(5)	0.67	28	1.35
	Kershaw($\varepsilon = 0.05$)	1^{st} Cheb, λ_{min}^{opt}, RAS(6)	2.60	95	1.62
	pb146	4_{opt}^{th} Cheb, RAS(4)	0.15	5.3	1.17
	pb67	4_{opt}^{th} Cheb, RAS(6)	0.47	16.0	1.40
	pb1568	4_{opt}^{th} Cheb, ASM(5)	0.15	3.8	1.17
V, $\searrow$	Kershaw ($\varepsilon = 1$)	1^{st} Cheb, λ_{min}^{opt}, RAS(2), V	0.09	8	1.75
	Kershaw ($\varepsilon = 0.3$)	1^{st} Cheb, λ_{min}^{opt}, RAS(5), V	0.67	28	1.35
	Kershaw ($\varepsilon = 0.05$)	4_{opt}^{th} Cheb, RAS(12), $\searrow$	2.40	88	1.75
	pb146	4_{opt}^{th} Cheb, RAS(4), V	0.15	5.3	1.17
	pb67	4_{opt}^{th} Cheb, RAS(12), $\searrow$	0.37	12.5	1.81
	pb1568	4^{th} Cheb, ASM(12), $\searrow$	0.14	3	1.27

Figure: T_S: solution time of fastest solver. T_D solution time of nekRS default, 1^{st} Cheb, ASM(3) V. V with k Chebyshev order has same complexity $\searrow$ with order $2k$ Chebyshev per iteration. Top half of table looks at fastest solver using full V-cycle. Bottom half of table looks at fastest solver.

Conclusion

- Speedup Navier-Stokes pressure Poisson solve around 15-30% relative to default nekRS solver.
- 4th and opt. 4th kind Chebyshev smoothers generally show improvement over 1st kind Chebyshev smoothing[8].
- Adapt Lottes's error bounds to determine *where* to use the full V-cycle versus half V-cycle.
- Continue to *reduce* pressure on coarse grid solve by reducing the iteration count, which should increasingly pay-off *at scale*.

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

Ongoing Work

- Tuning pMG solver params: https://tinyurl.com/nekrs-tune-one-sided
- 4th and Opt. 4th Kind Chebyshev Smoother implementation in AMG solvers: https://tinyurl.com/hypre-opt-cheb, https://tinyurl.com/trilinos-opt-cheb
- Porting those improvements from Hypre into nekRS: https://tinyurl.com/nekrs-amg-improv
- nekRS https://github.com/Nek5000/nekRS