Learning to Restart: Reinforcement Learning for GMRES(m)

Ryan Divan, Rishabh Mohapatra, Tyler Pellek {rd2700, rm5883, tp5523}@princeton.edu

Princeton University, Department of Mathematics

Questions

The optimal parameter m for GMRES(m) [1] is highly system-dependent, as arbitrarily slow convergence can be produced adversarially for any fixed m. What information do we need about the linear system to adaptively learn m? Can we solve this problem with reinforcement learning (RL)?

Problem

GMRES(m) is an iterative algorithm which solves the linear system $Ax = b$ for an $n \times n$ matrix A by searching the Krylov subspaces $\mathcal{K}_k(A)$ at iteration k. However, m iterations of GMRES(m) cost $\mathcal{O}(m^2)$, and convergence is not guaranteed when $m \neq n$.

Our Approach: GMRES-RL

We propose **GMRES-RL**, a reinforcement learning algorithm which learns m online. By using $\mathcal{O}(1)$ state size and a cheap Deep Q-Network, GMRES-RL can achieve faster convergence with lower overhead.

Methods: MDP Framing, Reward Shaping, and SLRL

Markov Decision Process (MDP)

- We model GMRES(m) as an MDP with states

$$s_k = (\|r_k\|_2, M_k, E_k) \tag{1}$$

and actions (restart parameters)

$$m_k \in \{1, \cdots, m_{\max}\}, \tag{2}$$

where r_k denotes the residual, and M_k and E_k are the history of last $d = 5$ parameters and residuals.

- Our work improves on formulations in [2,3], which have $\mathcal{O}(n)$ state size, fail to capture informative multistep dynamics, or use overhead-prone SAC models.

Reward Shaping

- We build on the reward defined in [2], which does not penalize larger m_k. GMRES iterations scale quadratically in the iterate number, leading to increased computation time.
- We use τ-clamped potential-based reward shaping as in [4], with

$$\Phi_\tau(r) = -\log(\max(\|r\|_2, \tau)/\tau). \tag{3}$$

This gives us the shaped reward

$$\begin{aligned} R_k = &-\lambda m_k - \Phi_\tau(r_{k-1}) + \gamma \Phi_\tau(r_k) \\ &+ B\mathbf{1}\{\|r_k\|_2 < \tau \leq \|r_{k-1}\|_2\}, \end{aligned} \tag{4}$$

where λ, γ, and B are tuned hyperparameters.

- Without trained policy on each, we use EPIC distance [6] (Pearson corr.) for reward similarity

Single Life Reinforcement Learning

- With a fixed policy, adversaries can produce right-hand side vectors which converge arbitrarily slow for GMRES(m).
- We address this with single-life reinforcement learning (SLRL) [2]: the agent learns online during the same solve, observes the current state, and selects the next Krylov subspace dimension.
- An SLRL algorithm must pay for its training cost with faster convergence to be competitive.

Experiment: GMRES-RL vs. AK-SLRL on Wall-Clock Time

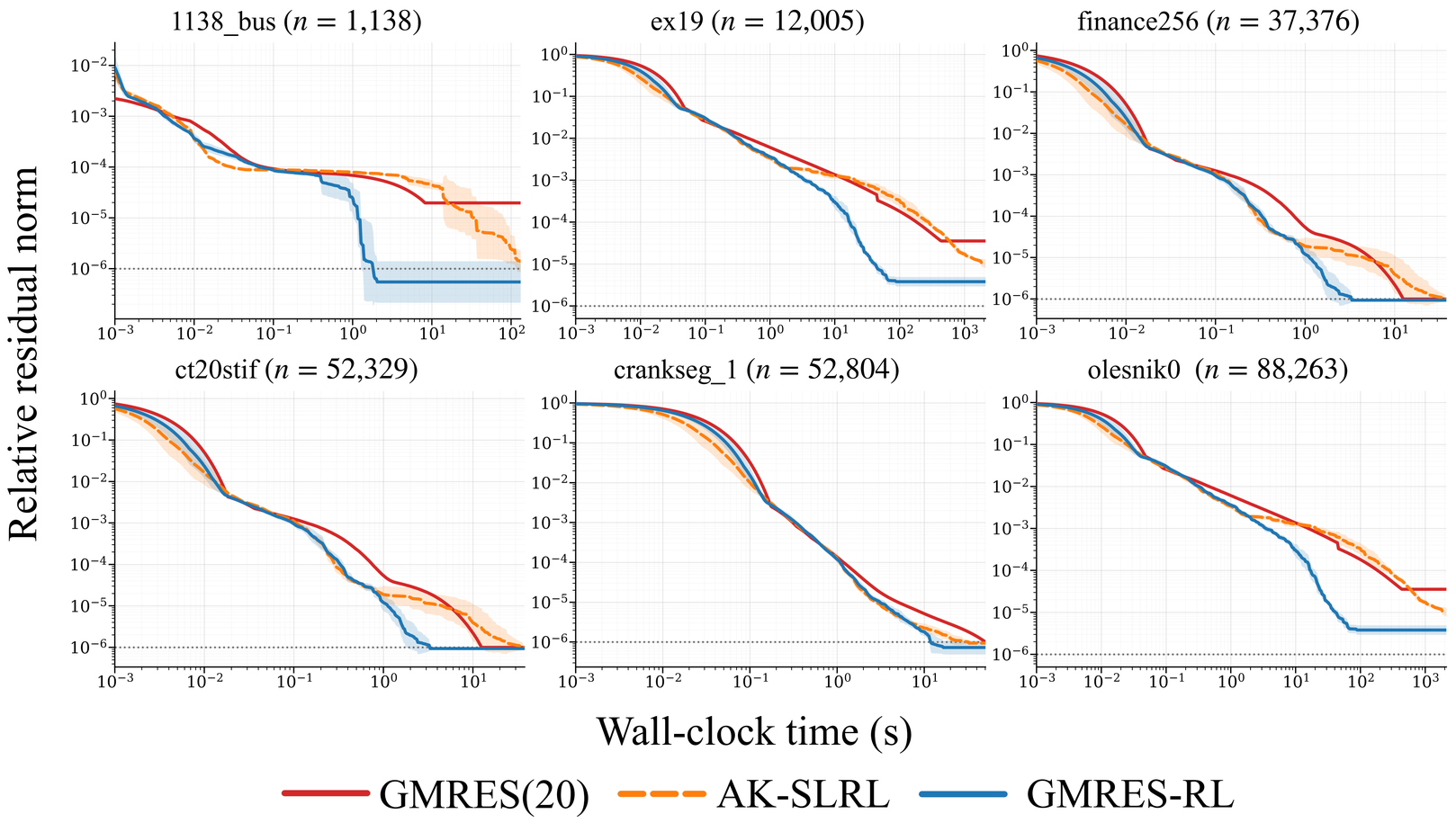


Figure 1: Convergence of GMRES(20), AK-SLRL [2], and GMRES-RL on the six matrices in [2] with common budget: relative residual norm vs. wall clock time. Solid line is mean across seeds; shaded band is ±1 std. Horizontal dashed line marks relative tolerance τ.

	Arnoldi iterations					Wall-clock (s)		Median	Conv.
Method	Mean	Median	p90	IQM	P(>GMRES(20))	Mean	p90	seed-std	rate
GMRES(20)	39,001	10,480	100,000†	1.00	—	233.4	617.9	—	70.3%
randGMRES	15,809	**2,250**	100,000†	**3.11**	0.827	73.4	**138.2**	303	89.8%
GMRES-RL (final)	**15,488**	2,350	**65,205**	2.98		**0.863**	**69.1**	145.6	**232**	**92.3%**

Table 1: Aggregated statistics on the 155-matrix benchmark, as in [5]. *Median seed-std* is across 139 matrices on which both stochastic methods converged on every seed. *Conv. rate* is conditioned on convergence before the cap of 100,000. † indicates the cap was hit.

We test GMRES-RL on 155 matrices with right-hand side $b = A\mathbf{1}$. We compare GMRES-RL to AK-SLRL on the benchmark in [2], as well as GMRES(20) and randGMRES (uniform random $m \in \{5, 10, 15, 20\}$).

Main points

- On the six-matrix benchmark from [2], GMRES-RL improves on both AK-SLRL and GMRES(20) in Arnoldi count and wall-clock time on every matrix, with the gap over AK-SLRL widening as n grows.
- GMRES-RL achieves similar mean and median performance to randGMRES, which picks m uniformly at random from $\{5, 10, 15, 20\}$, but dominates on the tail where randGMRES can fail to converge.

Scaling with Condition Number

- We consider a class of matrices which discretize 1D convection-diffusion operators.
- The Arnoldi work of GMRES-RL increases slowest.
- As $\kappa_2(A)$ increases, the advantage of GMRES-RL over randGMRES widens.

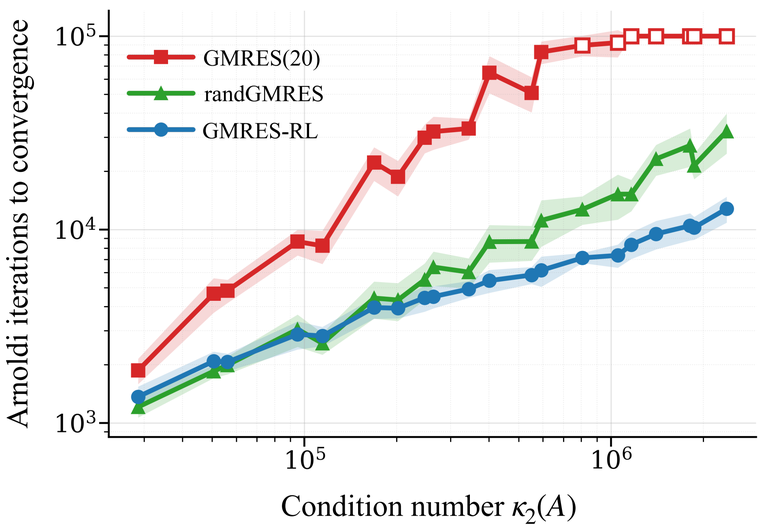


Figure 2: Arnoldi iterations to convergence on 20 1D convection-diffusion matrices vs $\kappa_2(A)$ for GMRES(20), randGMRES, and GMRES-RL. Shaded band is ±1 std, solid line is mean. Hollow marker indicates not all runs reached tolerance.

Conclusion

We pose GMRES(m) as an RL problem, and learn m online with our proposed GMRES-RL algorithm. We show that the state space can be simplified and simultaneously improve performance. GMRES-RL matches or beats AK-SLRL on every metric on the benchmark posed [2]. On our 155 matrix SuiteSparse benchmark, GMRES-RL matches the performance of randGMRES on average while still converging when randGMRES exceeded the computational budget.

References & Links

[1] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7(3):856– 869, 1986.
[2] H. Keramati and F. Hamdullahpur. AK-SLRL: Adaptive Krylov subspace exploration using single-life reinforcement learning. arXiv:2502.00227, 2025.
[3] L. Peairs and T.-Y. Chen. Using reinforcement learning to vary the m in GMRES(m). Procedia Comput. Sci., 4:2257–2266, 2011.
[4] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations. In Proc. ICML, pp. 278–287, 1999.
[5] R. Agarwal et al. Deep RL at the edge of the statistical precipice. CoRR abs/2108.13264, 2021.
[6] Gleave, A., Dennis, M., Legg, S., Russell, S., and Leike, J. (2021). Quantifying differences in reward functions. In International Conference on Learning Representations.

- Code: `https://github.com/ryan-divan/GMRES_RL`