Symmetric Stair Preconditioning of Linear Systems for Parallel Trajectory Optimization

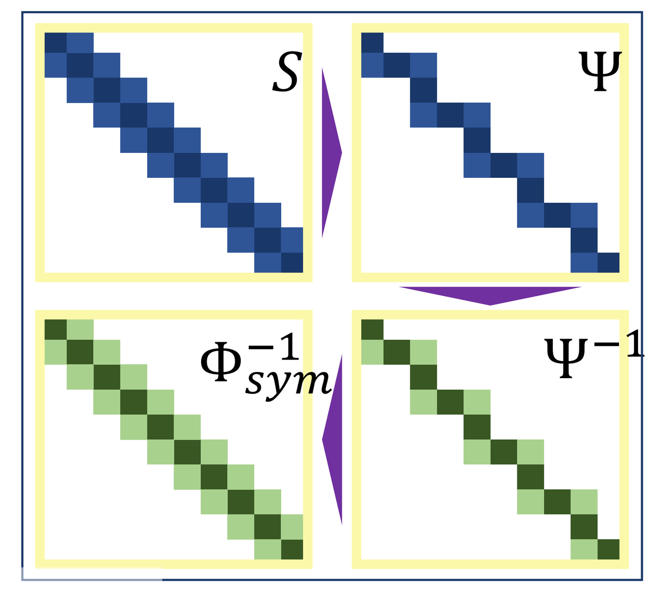
Xueyi Bu[1], Brian Plancher[2]

1: School of Engineering and Applied Science, Columbia University 2: Barnard College, Columbia University

ICRA2024 May 13-17, Yokohama, Japan

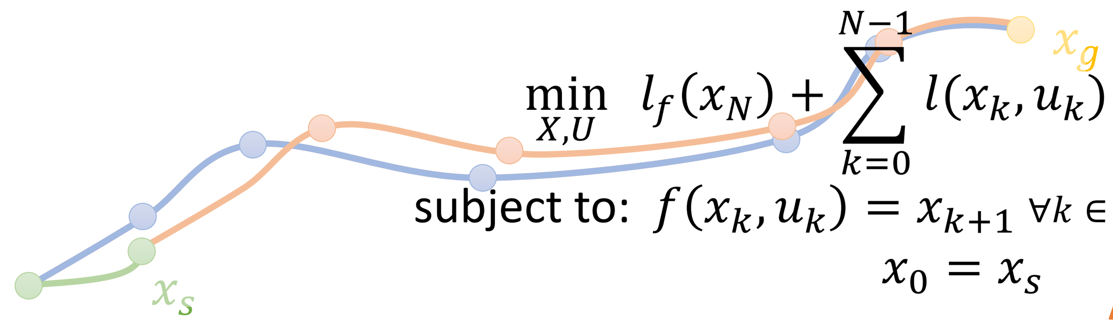
The Big Picture:

In this work, we present a **new parallel-friendly symmetric stair preconditioner** optimized for solving the linear systems underlying trajectory optimization problems. We prove that our preconditioner has **advantageous theoretical properties** when used with **iterative linear system solvers** such as a more clustered and bound Eigenspectrum. Numerical experiments with typical trajectory optimization problems reveal that our preconditioner provides up to a **34% reduction in condition number** and up to a **25% reduction in the number of linear system solver iterations**.



Direct Trajectory Optimization:

Trajectory optimization computes a robot's optimal path through an environment as a series of states, x, and controls, u, by minimizing a cost function, l, subject to discrete time dynamics, f.

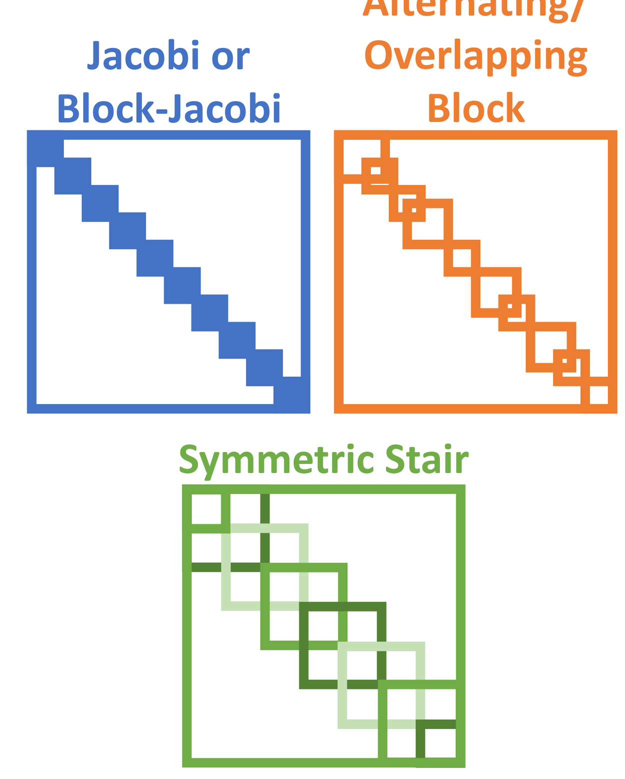


$$\min_{X,U} \; l_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

$$\text{subject to: } f(x_k, u_k) = x_{k+1} \;\forall k \in [0, N) \qquad x_0 = x_s$$

The Need for Sparse Parallel Preconditioners:

One key computation for direct trajectory optimization problems is the repeated solving of the resulting Karush-Kuhn-Tucker linear system, which can be solved using the **symmetric positive definite and block tridiagonal *Schur Complement*, S**. Iterative linear system solves can accelerate such problems on parallel processors. However, these methods **require a preconditioner, $\Phi^{-1} \approx S^{-1}$**, as their convergence properties are related to the clustering and magnitude of the eigenvalues of $\Phi^{-1}S$. Furthermore, as the resulting algorithms leverage matrix-vector products with both Φ^{-1} and S, **efficient parallel solvers require Φ^{-1} to be both sparse and parallel friendly**. Previous works usually leverage Jacobi, Block-Jacobi, as well as Alternating/Overlapping Block preconditioners.



The Symmetric Stair Preconditioner:

The stair-splitting polynomial preconditioner for block-tridiagonal systems is parallel friendly to both compute and invert, with an **analytical inverse**, however it is not symmetric, preventing its use with standard iterative methods.

$$S = \begin{bmatrix} D_1 & O_1 & 0 \\ O_1^T & D_2 & O_2 \\ 0 & O_2^T & D_3 \end{bmatrix} = \Psi_l = \begin{bmatrix} D_1 & 0 & 0 \\ O_1^T & D_2 & O_2 \\ 0 & 0 & D_3 \end{bmatrix} - P_l = -\begin{bmatrix} 0 & O_1 & 0 \\ 0 & 0 & 0 \\ 0 & O_2^T & 0 \end{bmatrix} \;\blacktriangleright\; \Psi_l^{-1} = \begin{bmatrix} D_1^{-1} & 0 & 0 \\ -D_2^{-1}O_1^T D_1^{-1} & D_2^{-1} & -D_2^{-1}O_2 D_3^{-1} \\ 0 & 0 & D_3^{-1} \end{bmatrix}$$

$$\Psi_r = \begin{bmatrix} D_1 & O_1 & 0 \\ 0 & D_2 & 0 \\ 0 & O_2^T & D_3 \end{bmatrix} - P_r = -\begin{bmatrix} 0 & 0 & 0 \\ O_1^T & 0 & O_2 \\ 0 & 0 & 0 \end{bmatrix} \;\blacktriangleright\; \Psi_r^{-1} = \begin{bmatrix} D_1^{-1} & -D_1^{-1}O_1 D_2^{-1} & 0 \\ 0 & D_2^{-1} & 0 \\ 0 & -D_3^{-1}O_2^T D_2^{-1} & D_3^{-1} \end{bmatrix}.$$

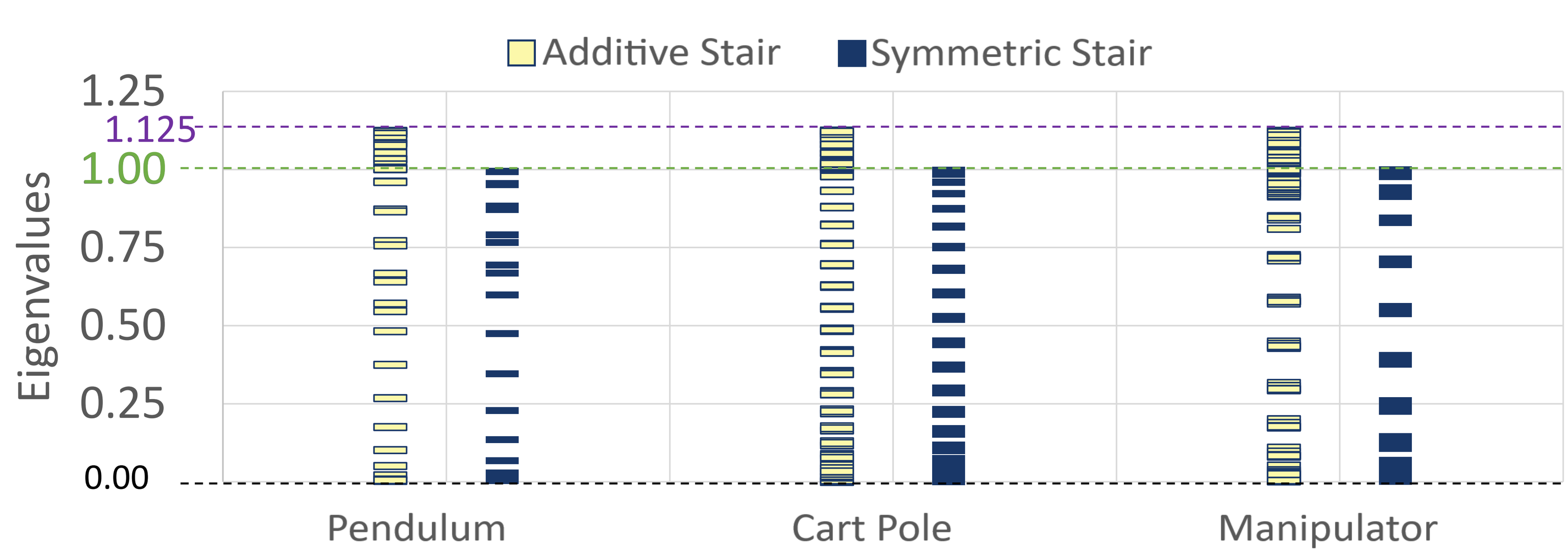
We prove that our symmetric stair preconditioner, Φ_{sym}^{-1}, provides an more clustered Eigenspectrum and bounds the spectral radius at one, unlike the existing symmetric stair-based preconditioner, the additive stair preconditioner, Φ_{add}^{-1}.

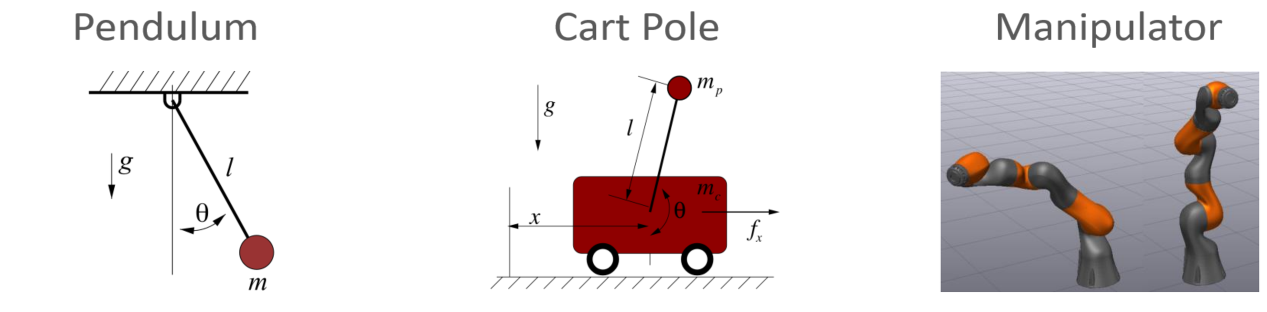
$$\Phi_{add}^{-1} = \frac{1}{2}(\Psi_l^{-1} + \Psi_r^{-1}) \;\blacktriangleright\; \lambda(\Phi_{add}^{-1}S) \in \left(0, \frac{9}{8}\right] \qquad \Phi_{sym}^{-1} = \Psi_l^{-1} + \Psi_r^{-1} - D^{-1} \;\blacktriangleright\; \lambda(\Phi_{sym}^{-1}S) \in (0, 1]$$

Experimental Results:

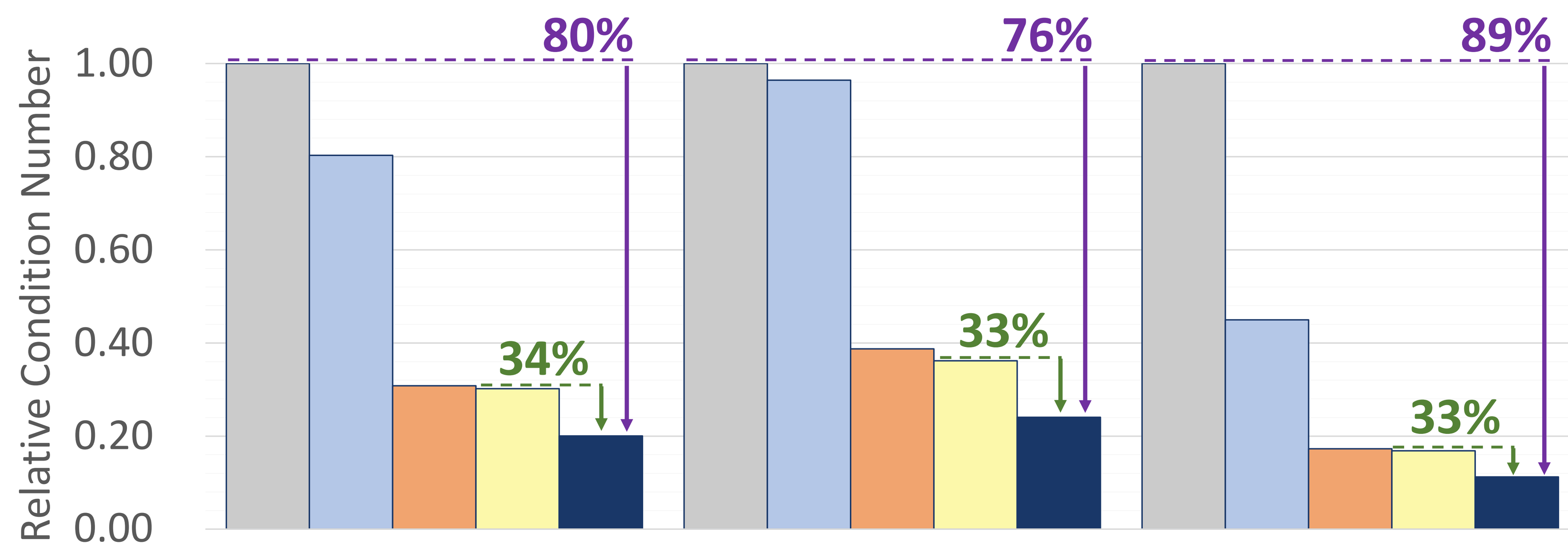
We provide numerical experiments comparing our preconditioner to previous parallel preconditioners for three canonical trajectory optimization problems: 1) a pendulum swing-up, 2) a cart-pole swing-up, and 3) a workspace motion for a 7-dof manipulator.

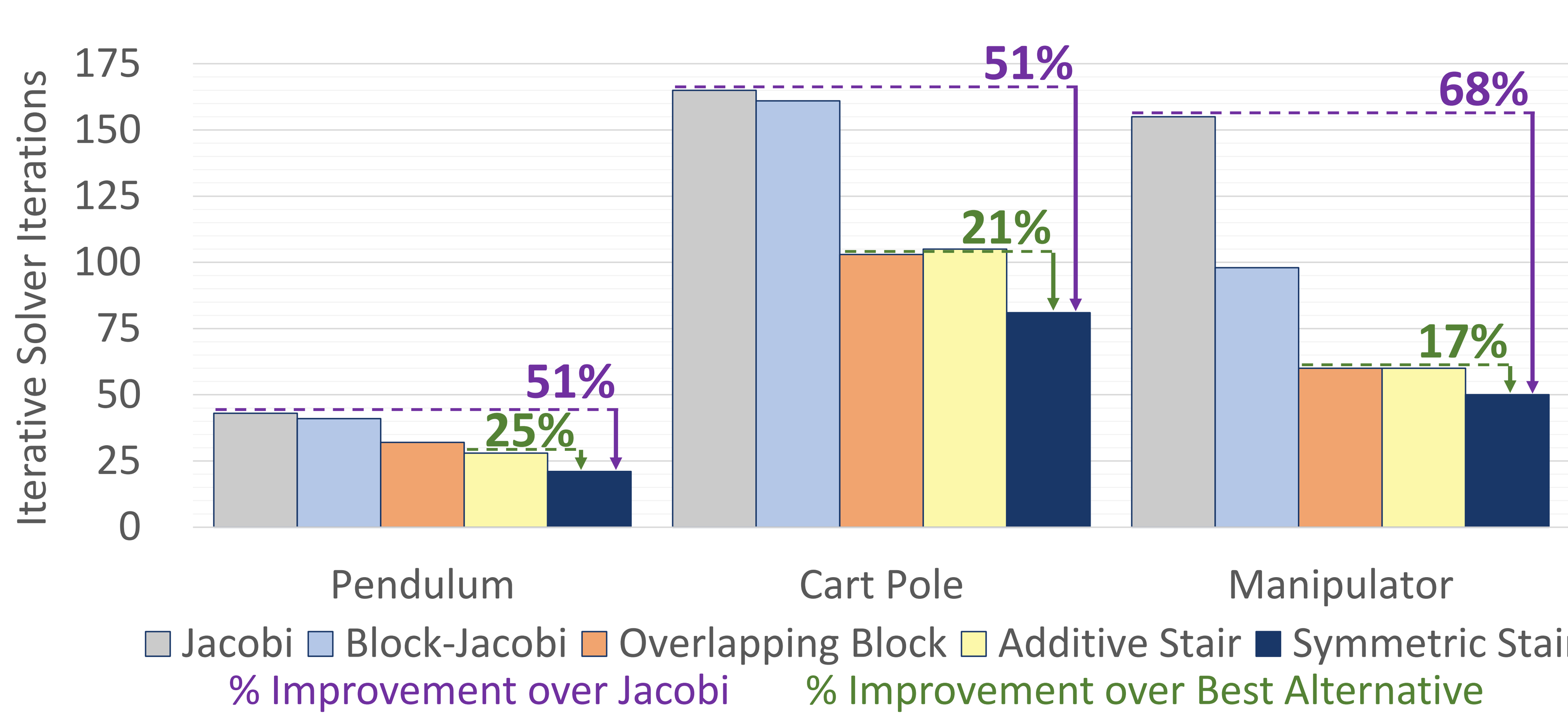
As shown in our proofs, our preconditioner provides a **clustered and bound Eigenspectrum.**





This both **reduces the condition number and, more importantly, the resulting iterations to convergence,** when used in conjunction with iterative linear system solvers.





The Barnard Accessible and Accelerated Robotics Lab

This material is based upon work supported by the National Science Foundation (under Award 2246022). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding organizations.