

Parallelization of all-at-once preconditioned solvers for time-dependent PDEs

Matthias Bolten, Ryo Yoda

Abstract

Modern high performance computers provide tremendous compute power by utilizing large amounts of cores. As a consequence, traditional spatial parallelization schemes lead to a saturation of the speedup more often. This motivated the use of parallelization in the time direction, as well. Various approaches exist and often two- and multilevel approaches like parareal or space-time multigrid are chosen that introduce a coarse level—or multiple coarse levels—to propagate information faster in time, usually in a serial manner. An alternative that is very natural from a numerical linear algebra viewpoint is to consider all-at-once systems. Consider a linear time dependent PDE:

$$\begin{aligned}\frac{\partial u(x, t)}{\partial t} &= \mathcal{L}u(x, t) + f(x, t), & (x, t) &\in \Omega \times (0, T], \\ u &= g, & (x, t) &\in \partial\Omega \times (0, T], \\ u(x, 0) &= u_0(x), & x &\in \Omega.\end{aligned}$$

Discretization using finite elements and denoting the mass matrix by M and the stiffness matrix by K in each time step with step width $\tau = \frac{T}{n}$ yields

$$M \left(\frac{\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}}{\tau} \right) = K\mathbf{u}^{(k+1)} + \mathbf{f}^{(k+1)},$$

writing the resulting n linear systems into one finally gives

$$\begin{bmatrix} M - \tau K & & & & \\ -M & M - \tau K & & & \\ & & \ddots & \ddots & \\ & & & -M & M - \tau K \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \vdots \\ \mathbf{u}^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{(1)} + M\mathbf{u}_0 \\ \mathbf{f}^{(2)} \\ \vdots \\ \mathbf{f}^{(n)} \end{bmatrix}.$$

This large system can be solved iteratively using, e.g., GMRES. Further, when M and K are symmetric or can be made symmetric simultaneously and when they are not changing like in the example provided, the matrix can be symmetrized by applying a reordering technique allowing to use methods like MINRES. If M and K do not change over time preconditioners for block Toeplitz or block Hankel matrices can be used in both cases. The GMRES case has been studied, e.g., in [3], the MINRES case was considered, e.g., in [2, 4, 5, 6]. The preconditioners studied are either block circulant or block ϵ -circulant and thus multiplication and inversion can be carried out in almost optimal, i.e., $\mathcal{O}(n \log n)$, complexity by using the FFT. Additionally, the blocks have to be solved efficiently which is usually carried out using the method used for the sequential time-stepping. While the study of the preconditioners is extensive, the actual parallel implementation is studied very little. One option to implement these kinds of methods is the usage of a parallel FFT as studied in [1]. Yet, an efficient parallelization of the FFT is relatively difficult given that it requires a lot of communication in comparison to very few arithmetic operations. This is one reason why multi-dimensional FFTs usually transpose the data such that the individual 1D-FFTs can be carried out sequentially. For the preconditioners considered in this case, in general only 1D-FFTs are needed.

Given the results obtained using efficient multi-dimensional FFTs on parallel computers as an alternative to a direct parallelization of the FFT we propose to transpose the data such that sequential 1D-FFTs can be used and to transpose it back before solving the individual blocks. The method resulting from a parallel implementation of the method proposed in [3] in this way provides an excellent scaling behavior, yielding an additional speedup after saturation of pure time-stepping with parallel solves of the spatial problem alone.

The applications of the preconditioners require the solution of usually complex block system that have the dimension of the spatial problem. Our implementation currently uses smoothed aggregation multigrid from Trilinos to solve these systems. We have used this approach on machines based on CPUs as well as on clusters with GPUs. In all cases we can achieve good scaling results, providing efficient parallelization in time by using preconditioning.

We will provide an overview over the different preconditioners that can be implemented in the proposed way, present the parallelization approach in detail, discuss the solution of the blocks and demonstrate the achieved performance.

References

- [1] G. CAKLOVIC, R. SPECK, AND M. FRANK, *A parallel-in-time collocation method using diagonalization: theory and implementation for linear problems*, 2023, arXiv:2103.12571 [math.NA].
- [2] S. HON, *Optimal block circulant preconditioners for block Toeplitz systems with application to evolutionary PDEs*, J. Comput. Appl. Math., 407 (2022), p. 15. Id/No 113965.
- [3] X.-L. LIN AND M. NG, *An all-At-once preconditioner for evolutionary partial differential equations*, SIAM J. Sci. Comput., 43 (2021), pp. a2766–a2784.
- [4] E. McDONALD, S. HON, J. PESTANA, AND A. WATHEN, *Preconditioning for nonsymmetry and time-dependence*, in Domain decomposition methods in science and engineering XXIII. Proceedings of the 23rd international conference, Jeju Island, South Korea, July 6–10, 2015, Cham: Springer, 2017, pp. 81–91.
- [5] E. McDONALD, J. PESTANA, AND A. WATHEN, *Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations*, SIAM J. Sci. Comput., 40 (2018), pp. a1012–a1033.
- [6] J. PESTANA AND A. J. WATHEN, *A preconditioned MINRES method for nonsymmetric Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 273–288.