An investigation of parallel performance of block epsilon-cirulant preconditioner for time-dependent PDEs

Ryo Yoda
ryoda@uni-wuppertal.de
University of Wuppertal
Wuppertal, Germany

Matthias Bolten
bolten@uni-wuppertal.de
University of Wuppertal
Wuppertal, Germany

1 INTRODUCTION

This work investigates the parallel performance of a memory-distributed implementation of the block ϵ-circulant (BEC) preconditioning [1] with MPI. This method is a promising parallel-in-time approach in a massively parallel environment for all-at-once linear systems arising from time-dependent PDEs. The BEC preconditioner introduces a weighted parameter ϵ into the block circulant preconditioner and achieves independent convergence for spatial mesh sizes with sufficiently small ϵ. However, its parallel performance has not been fully investigated. This work presents its parallel result for convection-diffusion problems.

2 BEC PRECONDITIONING

The BEC preconditioner is block circulant in time and introduces a weighted parameter, $\epsilon \in \mathbb{R}$, in the upper-right block. In the case of backward Euler method, P_{BEC} becomes

$$P_{\mathrm{BEC}} = \begin{bmatrix} A_0 & & & \epsilon A_1 \\ A_1 & A_0 & & \\ & \ddots & \ddots & \\ & & A_1 & A_0 \end{bmatrix} \in \mathbb{R}^{n_t n_x \times n_t n_x}. \quad (1)$$

Using the block circulant property in time, we can describe BEC preconditioning $\mathbf{z} = P_{\mathrm{BEC}}^{-1}\mathbf{y}$ as three-step procedure in Algorithm 1. The first and third steps are the FFT parts,

Algorithm 1 Three-step procedure of BEC preconditioning

1: Compute $\tilde{\mathbf{y}} = [(\mathcal{F}_{n_t} D_\epsilon) \otimes I_{n_x}]\,\mathbf{y}$ ▷ $\mathcal{F}_{n_t}$: Fourier matrix
2: Solve $B_k \tilde{z}_k = \tilde{y}_k$ for $\tilde{z}_k$ $\quad (k = 0, 1, \ldots, n_t - 1)$
3: Compute $\mathbf{z} = \left[(D_\epsilon^{-1}\mathcal{F}_{n_t}^*) \otimes I_{n_x}\right]\tilde{\mathbf{z}}$ ▷ D_ϵ: Diag. matrix

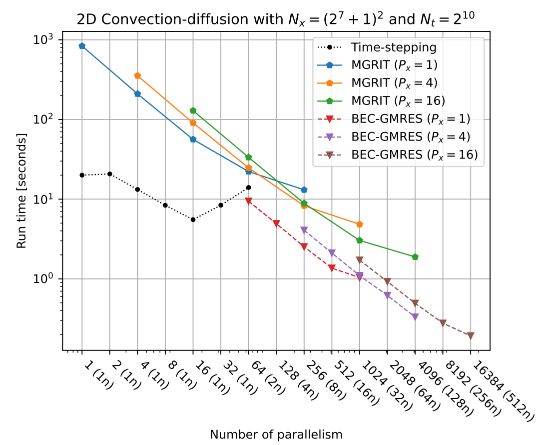
which repeat the FFT for one-dimensional time-step-sized vectors. We perform the data redistribution in the spatial domain and fully independent sequential FFTs. The second step requires a linear solver for complex-valued systems $B_k \tilde{z}_k = \tilde{y}_k$. We use the equivalent real-valued formulation, which splits complex numbers into real and imaginary parts, i.e., $B_k = B_k^{(R)} + iB_k^{(I)}$, $\tilde{z}_k = \tilde{z}_k^{(R)} + i\tilde{z}_k^{(I)}$, and $\tilde{y}_k = \tilde{y}_k^{(R)} + i\tilde{y}_k^{(I)}$.

$$\begin{bmatrix} +B_k^{(R)} & -B_k^{(R)} \\ +B_k^{(I)} & +B_k^{(I)} \end{bmatrix} \begin{bmatrix} \tilde{z}_k^{(R)} \\ \tilde{z}_k^{(I)} \end{bmatrix} = \begin{bmatrix} \tilde{y}_k^{(R)} \\ \tilde{y}_k^{(I)} \end{bmatrix} \quad (2)$$

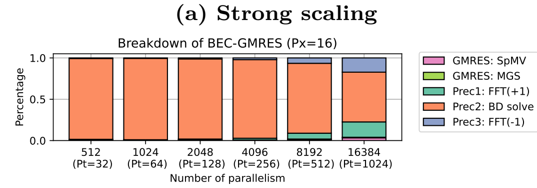
3 NUMERICAL EXPERIMENTS

We present parallel results of BEC preconditioned GMRES (BEC-GMRES) for convection-diffusion problems. The measurement environment is the Wisteria/BDEC-01 Odyssey system equipped with 2.2GHz Fujitsu A64FX. We implement with `FFTW v3.3.9` and `Trilinos v14.5` on the Fujitsu compiler and MPI with `v4.9.0`. Compile options are used `-std=c++17 -Nclang -Ofast -mcpu=a64fx -march=armv8-a -fPIC`.

Figure 1a shows that BEC-GMRES has good scaling performance with respect to temporal parallelism. Much of the time is spent in solving real-valued equivalent systems, as shown in Figure 1b.



(a) Strong scaling



(b) Breakdown of BEC-GMRES

Figure 1: Parallel tests with $n_x = (2^7+1)^2$ and $n_t = 2^{10}$

4 CONCLUSION

BEC-GMRES has achieved good scaling with respect to temporal parallelism. Future work will reduce the time in solving real-valued equivalent systems using complex-valued AMG solvers and inexact iteration in preconditioning step.

REFERENCES

[1] Xue-lei Lin and Michael Ng. 2021. An All-at-Once Preconditioner for Evolutionary Partial Differential Equations. *SIAM Journal on Scientific Computing* 43, 4 (2021), A2766–A2784. https://doi.org/10.1137/20M1316354