Acceleration of Optimized Coarse-grid Operators by Spatial Redistribution for Multigrid Reduction in Time

Ryo Yoda[1], Matthias Bolten[2], Kengo Nakajima[1], and Akihiro Fujii[3]

[1] The University of Tokyo, Japan ryo_yoda@mist.i.u-tokyo.ac.jp
[2] Wuppertal University, Germany
[3] Kogakuin University, Japan

Abstract. The multigrid reduction in time (MGRIT) method is one of the parallel-in-time approaches for time-dependent PDEs and typically uses rediscretized coarse-grid operators. As their convergence struggle with hyperbolic problems, an optimization method for coarse-grid operators has been proposed to deal with these problems. This method improves convergence using coarse-grid operators with a slightly increased number of nonzero elements. However, it is more desirable for coarse-grid operators to be cheaper than fine-grid operators, and there is room for improvement in terms of parallel implementation. This work combines the spatial redistribution technique for MGRIT, which accelerates coarse-grid solvers using agglomerated idle processors, with the above optimization method. This combination attempts to achieve better scaling performance while maintaining good convergence. Numerical experiments demonstrate a 23% runtime reduction at most among the various assignments tried with specific amount of parallelism.

Keywords: parallel-in-time approaches · multigrid methods · coarse-grid optimization · spatial redistribution

1 Introduction

This paper considers parallel numerical solvers for time-dependent partial differential equations (PDEs). In modern computing systems with increasing number of cores, spatial parallelism obtained by domain decomposition methods is exhausted because of over-decomposition. Therefore, attempts to extract temporal parallelism, called parallel-in-time approaches [4, 6], have attracted much attention recently. Some examples of the most powerful parallel-in-time solvers are space-time multigrid, Parareal, and multigrid reduction in time (MGRIT) [3]. This paper focuses on MGRIT. While the application of MGRIT has been successful for various parabolic problems, the convergence of MGRIT struggles with hyperbolic problems [1, 5]. This failure can occur even with implicit time discretization or spatial coarsening to stabilize the coarse-grid problem.

De Sterck et al. firstly identified that the convergence deterioration was due to rediscretized coarse-grid operators and proposed an optimization method for

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_29

coarse-grid operators for linear advection problems [1]. This optimization method constructs coarse-grid operators with a realistic number of nonzero elements that minimizes the spectral difference from the ideal operator, dramatically improving the convergence of MGRIT. However, it should be noted that the cost of operators increases from fine- to coarse- level, it is not negligible as mentioned below in terms of parallel performance; because when running at high temporal parallelism, the percentage of coarse-level operations increases, and the overhead is not negligible. Thus, there is room for improvement of the parallel implementation here.

The aim of this paper is to accelerate MGRIT with optimized coarse-grid operators using the spatial redistribution technique [7]. This technique accelerates the coarse-level solvers by assigning temporal agglomerated idle processors to redistributed spatial domains and reduces the overhead. Hence, the novel combination of these existing methods is expected to offset the increased cost of operators optimized for improved convergence.

2 Multigrid Reduction in Time

First, we consider sequential time-stepping for linear time-dependent PDEs. Let N_x be the number of spatial grid points and N_t be the number of time steps. We assume that the spatial and temporal discretized governing equations have the relation: $\mathbf{u}^{i+1} = \Phi\mathbf{u}^i + \mathbf{g}^{i+1}$, where $\mathbf{u}$ and $\mathbf{g}$ denote the unknown and force vector, respectively, and the temporal index i ranges from 0 to $N_t - 1$. The time-stepping proceeds sequentially N_t times according to the above relation.

Multigrid reduction in time (MGRIT) [3] is an all-at-once approach that extracts time parallelism by solving all time steps at once. It yields the linear space-time system on the temporal fine-grid based on the time-stepping method:

$$A\mathbf{u} = \begin{bmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{bmatrix} \begin{bmatrix} \mathbf{u}^0 \\ \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{g}^0 \\ \mathbf{g}^1 \\ \vdots \\ \mathbf{g}^{N_t-1} \end{bmatrix} = \mathbf{g}, \tag{1}$$

where Φ is called a fine-grid operator. MGRIT uses relaxations in parallel by delimiting the time dependency based on C-points and F-points. C-points correspond to each m-th time-step, m is called a coarsening factor, and the others are labeled F-points. F- or C-relaxation perform the time-stepping method at F-points or C-points only, respectively. FCF-relaxation also performs F-, C-, and F-relaxations in that order.

MGRIT also constructs a coarse-grid system similar to Eq. 1 with N_t/m time steps and a coarse-grid operator Ψ. The restriction and prolongation operators, which transfer between fine and coarse grids, are defined as injections on C-points. In general, we obtain Ψ with a rediscretization approach that enlarges the time-step width Δt by a m factor. On the other hand, Ψ can also be obtained by an optimization method [1], which minimizes the spectral difference from optimal

Table 1: Convergence rates of MGRIT with various numbers of nonzero elements for one-dimensional advection problem with $N_x = 2^8$, $N_t = 2^9$ and $\mathrm{nnz}(\tilde{\phi}) = 10$.

	$m = 2$	$m = 4$	$m = 8$	$m = 16$	$m = 32$
$\mathrm{nnz}(\tilde{\phi}^m)$	19	36	60	89	105
$\nu = 10$	0.092	0.119	0.190	0.469	1.233
$\nu = 11$		0.088	0.166	0.362	0.610
$\nu = 12$			0.157	0.227	0.388
$\nu = 13$			0.085	0.226	0.362
$\nu = 14$				0.183	0.307
$\nu = 15$				0.105	0.311
$\nu = 16$				0.073	0.232

operators. Here we briefly it introduce according to the notation of [1]. Let λ be the eigenvalues of fine-grid operators Φ and let μ be the eigenvalues of coarse-grid operators. Assuming periodic boundary conditions for spatial domains, using the first columns of each operator and the DFT matrix $\mathcal{F} \in \mathbb{C}^{N_x \times N_x}$, we can compute the eigenvalues $\lambda^m = \mathcal{F}\tilde{\phi}^m$ and $\mu = \mathcal{F}\psi$ for Φ^m and Ψ, respectively. Based on this computation, the optimization problem for coarse-grid operators is formulated by

$$\psi := \underset{\hat{\psi} \in \mathbb{R}^\nu}{\operatorname{argmin}} \left\| W_\lambda^{1/2} \mathcal{F} \left(\tilde{\phi}^m - \mathcal{R}^T \hat{\psi} \right) \right\|_2^2, \tag{2}$$

where $W_\lambda = \mathrm{diag}(w(|\lambda_k|))$ is a weighting matrix, and $k = 0, \dots, N_x - 1$ denotes the spatial index. We adopt $w(z) = 1/(1 - z + \epsilon)^2$ as weight function, where $\epsilon = 10^{-6}$. The operator $\mathcal{R} \in \mathbb{R}^{\nu \times N_x}$ constrains sparsity to obtain practical coarse-grid operators with $\nu \ll N_x$. By solving the normal equation of Eq. 2, we obtain the solution ψ and construct the optimized coarse-grid operator Ψ.

Finally, we briefly review the convergence improvement of the optimization approach for a one-dimensional linear advection problem based on the experiment in Fig. 6 in [1]. Table 1 shows the convergence rates of MGRIT, derived by the two-level reduction analysis [2], for the above problem with each m and ν. We confirm that this approach provides good convergence even for hyperbolic problems by slightly increasing ν for m. While this increase is acceptable and practical, there is room for improvement in terms of parallel performance, which will be addressed in the next section.

3 Spatial redistribution technique

The spatial redistribution technique [7] for MGRIT assigns temporally agglomerated idle processes to spatially redistributed domains, accelerating the coarse-grid operators. Therefore, this technique decreases temporal parallelism P_t and increases spatial parallelism P_x to reduce the coarse-level spatial solver on the coarse level. See [7] for more details.

In this paper, we use this technique to reduce the cost of optimized coarse grid operators. Moreover, it does not affect convergence; we improve scaling while maintaining good convergence. Also, it offers flexibility in how processes are agglomerated and reassigned, and there are no restrictions on application problems. Therefore, in later numerical experiments, after constructing the coarse-grid operators via the optimization approach, this solver is parallelized in the spatial direction by row-wise one-dimensional block partitioning according to spatial parallelism.

4 Numerical experiments

This section investigates the effectiveness of MGRIT with optimized coarse-grid operators and spatial redistribution. Numerical experiments are conducted for the one- or two-dimensional advection problems on structured grids with periodic boundary conditions: $\mathbf{u}_t - \alpha \mathbf{u}_x = 0$, where $(\mathbf{x}, t) \in [0,1]^d \times [0,T]$ and $\alpha = 1$ in our experiments. The initial conditions are $\sin{(\pi x)}^4$ or $\sin{(\pi x)}^4 \cdot \sin{(\pi y)}^4$, respectively. We discretize them with third-order upwind discretization for space and explicit third-order Runge-Kutta for time. The CFL number on the finest level is set to 0.85 times the CFL limit, where the rediscretization approach does not work well. The respective coefficients are based on the setup in [1].

To simplify the comparison of solvers, we use the following abbreviations: "M1" corresponds to the MGRIT with optimized coarse-grid operators. "M2" denotes a solver that combines the above with spatial redistribution. The convergence tolerance for both solvers isthat the relative residual 2-norm becomes less than 10^{-10}. We evaluate each solver on the Wisteria/BDEC-01 Odyssey supercomputer system equipped with 2.2GHz Fujitsu A64FX, which has 12×4 cores. We implement flat MPI and MPI/OpenMP modes for each solver using Fujitsu compiler and MPI v4.7.0. In the latter mode, the number of threads T_x is fixed to 12, considering A64FX architecture. The runtime is the minimum value of three measurements.

The first experiment is a one-dimensional problem with $N_x = 2^{15}$, $N_t = 2^{14}$, $T = 5.528$, and flat MPI mode. Both MGRITs use $L = 6$ levels with the coarsening factor $m = 4$. Fig. 1 shows the strong scaling results. Both figures include the same results of the sequential time-stepping solver, and we can see it reduces the runtime up to $P_x = 1,024$. However, it stagnates at $P_x = 4,096$ and indicates the exhaustion of spatial parallelism. Next, we move on to M1 on the left and M2 on the right. Both MGRITs converged after only four iterations for this problem. These optimization approaches showed good convergence while the rediscretization approach diverged (not shown in the figure). The scaling tests fix the spatial parallelism P_x from 1 to 256 and increase the temporal parallelism P_t. We can see from the left figure that while M1 scales well at small temporal parallelism, the improvement degrades as the temporal parallelism approaches 4,096, the maximum. The reason is that as temporal parallelism increases, the coarse levels occupy a larger proportion of the total runtime, and these levels use optimized coarse-grid operators that are slightly more expensive than the finest

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_29

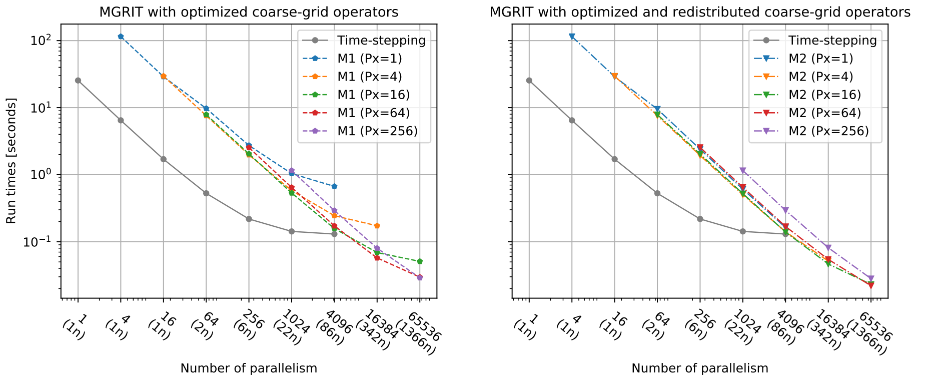

Fig. 1: Strong scaling experiment for one-dimensional advection problem with $N_x = 2^{15}$, $N_t = 2^{14}$, $L = 6$, and $M = 6$. Each color depends on the spatial parallelism specified at the finest level. "M1": MGRIT with optimized coarse-grid operators and "M2" with these and spatial redistribution.

grid. In contrast, M2 scales well without stagnation, even at high parallelism. This good scaling is achieved by accelerating the optimized coarse-grid operators. In order to see the improvement of M2 in more detail, we use a runtime breakdown of the fastest case with $P_x = 64$ and $P_t = 1,024$ in Fig. 2. This figure compares the runtime of M1 and M2 at each level, decomposing it into four representative operations: sparse matrix-vector (SpMV), vector operations, communication in space, and communication in time. In the SpMV part, we can see that the runtime increases as the level increases, except for $L = 0$, which contains the convergence check part. This increase directly corresponds to an increase in the number of nonzero elements in optimized coarse-grid operators. Since there is no temporal agglomeration and spatial redistribution on $L = 0$ and 1 for P_t and N_t in this problem, there is no significant difference in the runtime of the two solvers on these levels. After $L = 2$, M2 reduces the SpMV and vector operation parts due to the spatial redistribution. This solver also increases communication to some extent on coarse levels due to the increased spatial parallelism, but it is negligibly small.

Finally, we confirm the different parallelism assignments of each fastest case at maximum nodes 1,366 nodes in Fig. 1: M1 with $P_x = 256$ and $P_t = 256$ and M2 with $P_x = 64$ and $P_t = 1,024$. Comparing the two, M2 achieves a 23% runtime reduction over M1. The M2 parallelism assignment decreases the spatial parallelism and increases the temporal parallelism, compared to the fastest case of M1. A similar trend can be observed for smaller nodes. We believe this is because we can simultaneously benefit from the high temporal parallelism of MGRIT and the coarse-level overhead reduction due to spatial redistribution.

The second experiment is the two-dimensional problem with $N_x^2 = (2^9)^2$, $N_t = 2^{10}$, $T = 2.764$, and hybrid mode. The spatial direction is parallelized

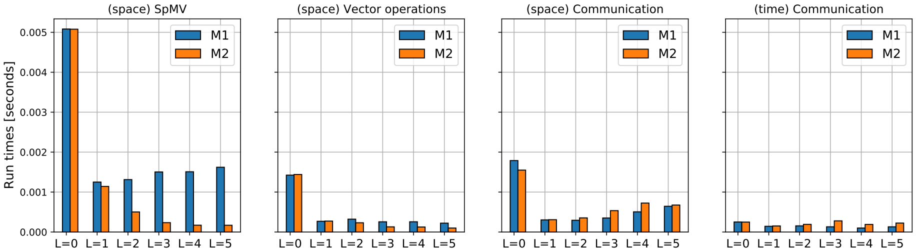

Fig. 2: Runtime breakdown with $P_x = 64$ and $P_t = 1,024$ at 1366 nodes in Fig. 1.

using a hybrid MPI/OpenMP approach with a fixed number of threads $T_x = 12$, and the temporal direction is parallelized using MPI only. We set $L = 4$ and $m = 4$ for both MGRITs. In the two-dimensional case, we obtained similar results as in the previous section. Fig. 3 shows the stagnation of time-stepping and M1 at high parallelism and the well-scaling of M2. Focusing on the results at 1,024 nodes, we compare both MGRITs' performance. M2 with $P_x = 16$, $P_t = 256$, $T_x = 12$ achieves a 17% improvement compared to M1 with $P_x = 64$, $P_t = 64$, $T_x = 12$.

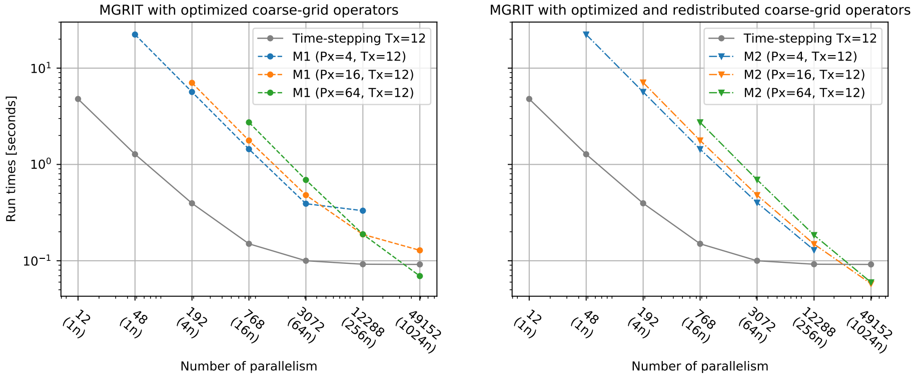

Fig. 3: Strong scaling experiment for two-dimensional advection problem with $N_x^2 = (2^9)^2$, $N_t = 2^{10}$, $L = 4$, $M = 4$, and $T_x = 12$. "M1": MGRIT with optimized coarse-grid operators and "M2" with these and spatial redistribution.

5 Conclusion

This paper accelerated MGRIT with optimized coarse-grid operators by the spatial redistribution technique. These operators provide good convergence for one- or two-dimensional advection problems, even for explicit discretizations.

On the other hand, the overhead of increasing the number of nonzero elements hinders scaling when coarse-level solvers occupy a large proportion at high temporal parallelism. The spatial redistribution reduces this overhead to achieve a good convergence and a scalable solver, using agglomerated idle processes on coarse levels. Our numerical experiments show a 23% improvement for the one-dimensional problem and a 17% improvement for the two-dimensional problem compared with the fastest parallelism assignment.

In numerical experiments on a two-dimensional problem, we evaluate a hybrid MPI/OpenMP implementation with a fixed number of threads. However, given that the cost of coarse-grid operators varies with the optimization approach and spatial parallelism changes with the spatial redistribution, the optimal configuration of the number of processes and threads will be different. Future work will investigate them as they vary on the coarse level.

Our approach can tolerate some increase in the cost of optimized coarse-level operators. Ideally, if coarse-grid operators are accelerated by m times, a coarsening factor of MGRIT, then an increase in the cost of m times is acceptable for an optimization method. The present optimization process is performed sequentially and constructs coarse-grid operators. However, for example, an optimization method is an option that does not dramatically improve convergence but has sufficient parallelism at the construction stage. We believe that this strategy may lead to new coarse-grid operator optimization methods.

Acknowledgements This work is supported by “Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures” in Japan (Project ID: jh220049).

References

1. De Sterck, H., Falgout, R.D., Friedhoff, S., Krzysik, O.A., MacLachlan, S.P.: Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. Numerical Linear Algebra with Applications **28**(4) (2021)
2. Dobrev, V., Kolev, T., Petersson, N., Schroder, J.: Two-level convergence theory for multigrid reduction in time (mgrit). SIAM Journal on Scientific Computing **39**(5), S501–S527 (2017)
3. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM Journal on Scientific Computing **36**(6), C635–C661 (2014)
4. Gander, M.J.: 50 years of time parallel time integration. In: Multiple shooting and time domain decomposition methods, pp. 69–113. Springer (2015)
5. Howse, A., Sterck, H., Falgout, R., MacLachlan, S., Schroder, J.: Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. SIAM Journal on Scientific Computing **41**(1), A538–A565 (2019)
6. Ong, B., Schroder, J.: Applications of time parallelization. Computing and Visualization in Science **23**(11) (2020)
7. Yoda, R., Bolten, M., Nakajima, K., Fujii, A.: Assignment of idle processors to spatial redistributed domains on coarse levels in multigrid reduction in time. In: HPCAsia2022. p. 41–51 (2022)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_29