AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-0340

Parallel in Time for a Fully Space-Time Adaptive Mesh Refinement Algorithm

Joshua Christopher* and Xinfeng Gao† and Stephen Guzik‡
Computational Fluid Dynamics and Propulsion Laboratory
Colorado State University, Fort Collins, CO 80523, USA

Robert D. Falgout§
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

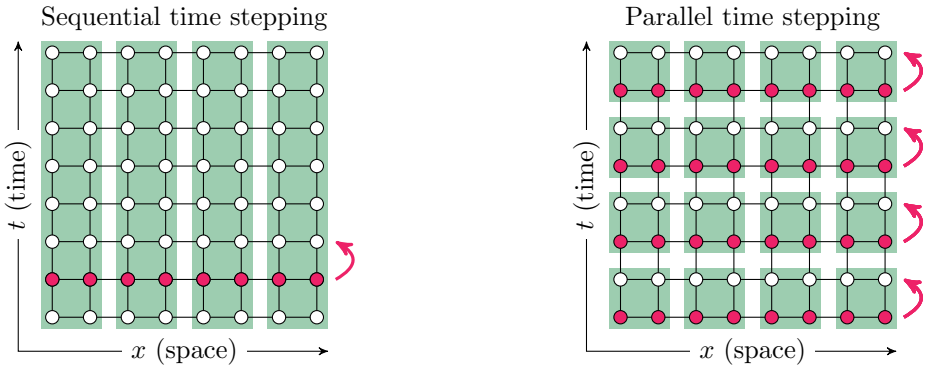
Jacob B. Schroder¶
Department of Mathematics and Statistics
University of New Mexico, Albuquerque, NM 87131, USA

In order to solve time-periodic problems in computation fluid dynamics, typically the characteristic time is iterated over multiple times to drive out transients. This can lead to a large number of steps to achieve a well-converged solution. Parallel-in-time methods excel in cases like these problems characterized by time periodicity and provide a speedup over time-sequential codes. In particular, time-parallel methods using multigrid only need to solve a single characteristic time due to the iterative nature of the method. In this study, we apply space-time multigrid with adaptivity to solve time-periodic problems. To verify and validate the implementation, Stokes second problem is solved in time-parallel and compared to the time-sequential solutions. A strong scaling test demonstrates that this time-parallel method is able to achieve speedups of $13\times$ over the time-sequential algorithm without loss of accuracy.

I. Introduction

Currently, computational fluid dynamics (CFD) algorithms are predominately parallelized in space. Figure 1a illustrates that the spatial domain is decomposed for parallelization but the time stepping is sequential. Spatial parallelism provides good parallel scaling only when the number of spatial mesh points per core is large. However, the parallel efficiency reaches a plateau when spatial parallelism saturates. Additional cores can be applied to the parallelization of time stepping to further speed up calculations. Figure 1b shows the parallelization in both space and time. Parallel-in-time (PinT) methods open a new direction for adapting CFD methods to computer architectures where the number of parallel processing units becomes extremely large. Many CFD applications in engineering that take large numbers of time steps can potentially benefit significantly from PinT methods, such as turbomachinery, wind turbines, or reciprocating engines. To harvest parallelism in both space and time, we will develop new space-time parallel CFD methods.

*Colorado State University Ph.D. Student, Member AIAA, Corr. Author, Email: joshua.christopher@colostate.edu
†Colorado State University Associate Professor, Email: xinfeng.gao@colostate.edu, Member AIAA
‡Colorado State University Assistant Professor, Email: stephen.guzik@colostate.edu, Member AIAA
§Lawrence Livermore National Lab, Email: rfalgout@llnl.gov
¶University of New Mexico Assistant Professor, Email: jbschroder@unm.edu



(a) Time-sequential **(b) Time-parallel**

Fig. 1 Parallelization strategy for a time-sequential algorithm and a time-parallel algorithm.

The present study is primarily focused on applying the new adaptive parallel-in-time framework [1, 2] to solve a time periodic problem and demonstrate the improvement of solution wall-times over sequential time stepping. Parallel-in-time methods allow significant speedups for time-periodic problems since the multigrid reduction-in-time (MGRIT) method only needs to solve for a single characteristic time interval. Classical time-sequential methods typically need to solve 5-10 periods to drive out transients. Based on the PinT multigrid capabilities of *XBraid* [3], the framework is further enhanced with the space-time adaptivity using structured adaptive mesh refinement (SAMR). The CFD application is performed by *Chord* [4–8], a fourth-order accurate finite-volume method for solving the compressible Navier-Stokes equations with Reynolds-averaged Navier-Stokes (RANS), large eddy simulation (LES), or direct numerical simulation (DNS) approaches for turbulence at high Reynolds-number. If reacting flows are of interest, Chord is capable of combustion modeling with detailed chemical kinetics. Chord is based on the *Chombo* parallel AMR library [9] and uses a mapped-grid approach to perform SAMR on complex engineering geometries. Advantages of XBraid are that it is non-intrusive and designed to use existing implementations of time-integration methods. Many CFD codes take decades to develop, validate, and mature, thus the use of XBraid avoids any major overhaul of the validated application codes.

The remainder of this study is organized as follows. Section II provides brief background information on the PinT methods used by focusing on the mathematical formulation of the system. The adaptations needed to support solving time-periodic problems with PinT methods are discussed in Section III. In Section IV, we present the results of applying the adaptive PinT method to a time-periodic Stokes flow problem. Lastly, Section V makes concluding remarks and provides several directions for future work.

II. Overview of the PinT Method

The software elements supporting the adaptive PinT infrastructure include XBraid, Chord, and Chombo. Each is a well established open source code or library. Ample literature is available for XBraid [10–13], Chombo [14–17], and Chord [5, 7, 8, 18]. Furthermore, the complete adaptive PinT infrastructure is described in our recent work [2]. For conciseness, the detailed knowledge of these elements along with the adaptive PinT infrastructure are not presented here and interested readers are referred to the literature mentioned above. Instead, we briefly review the underlying mathematical formulation of the PinT method to provide the readers with the essential background and motivation.

The central concept of XBraid [3] is the MGRIT method, which is used to enable time parallelization. MGRIT defines a temporal mesh of time points $t^i = i\delta t$ with $i = 0, \ldots, N_t$, time step size of $\delta t = T/N_t$, final time T, total number of time points N_t, and the solution state $\mathbf{U}_i \approx \mathbf{U}(t^i)$. In addition to the uniform temporal mesh described, non-uniform temporal meshes are possible with XBraid. Given a one-step time integration method Φ_i, the time discretization method can be represented as,

$$\mathbf{U}_0 = g_0, \quad \mathbf{U}_i = \Phi_i(\mathbf{U}_{i-1}) + \boldsymbol{g}_i, \quad i = 1, 2, \ldots, N_t \,. \tag{1}$$

where $\boldsymbol{g}_i$ are solution-independent terms. This can be represented (in the linear setting only, for simplicity) with the

system,

$$A\mathbf{U} \equiv \begin{pmatrix} \mathrm{I} & & & \\ -\Phi_1 & \mathrm{I} & & \\ & \ddots & \ddots & \\ & & -\Phi_N & \mathrm{I} \end{pmatrix} \begin{pmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_N \end{pmatrix} \equiv \boldsymbol{g} \,. \tag{2}$$

A forward block solve of this system corresponds to sequential time integration. While a forward solve is optimal and $O(N)$, it is serial. Applying a multigrid method allows a concurrent solve that is also $O(N)$.

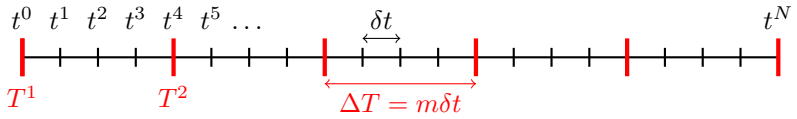
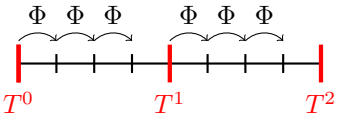
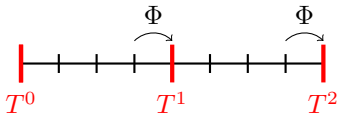


Fig. 2 A two-level time grid composed of F-points in black and C-points in red. The composition of C-points and F-points form the fine time grid while, the C-points form the coarse time grid.

With MGRIT, the temporal domain specified above is coarsened by a factor m into a hierarchy of temporal grids. Each grid is partitioned into F-points, which exist only on the fine time grid, and C-points, which exist on both the fine time grid and the next coarser time grid. A two level hierarchy is shown in Figure 2, where the fine grid is the composite of F-points (in black) and C-points (in red), and the coarse grid is just the C-points. Restriction and prolongation are simply injection at the C-points. F-relaxation applies the time propagator to the F-points in parallel while C-relaxation propagates the solution to the C-points in parallel. F-relaxation is shown in Figure 3a which performs the applications of Φ starting at T^0 concurrently with the applications of Φ starting at T^1. Shown in Figure 3b is C-relaxation which completes the application of Φ to the C-points. Like with F-relaxation, the application of Φ to points T^1 and T^2 are completed simultaneously. Successive applications of F-relaxation and C-relaxation, called FCF-relaxation, make up the relaxation strategy for MGRIT. The time propagator used in this study is the standard explicit fourth-order Runge-Kutta (RK4) method.



(a) F-relaxation is the parallelized application of Φ, the time integration operator, to each of the F-points.

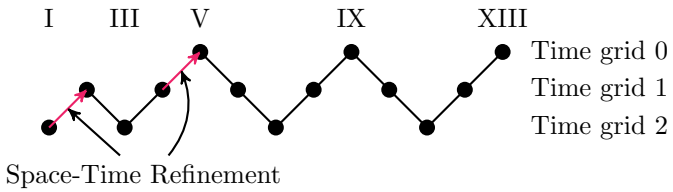


(b) C-relaxation is the parallelized application of Φ, the time integration operator, to each of the C-points.

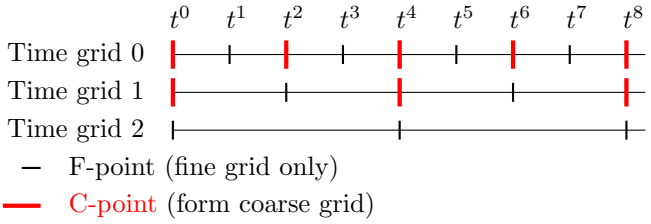
Fig. 3 Relaxation methodology of F- and C-relaxations.

The space-time adaptivity in the PinT algorithm is implemented with SAMR. The process of creating the adaptive space-time meshes is illustrated by Figure 4. Essentially, an initial uniform coarse space-time mesh is generated from input parameters. A PinT solve on this coarse mesh tags space-time intervals for refinement and a new refined space-time mesh is generated. The refined temporal mesh is passed to the MGRIT algorithm so that a new temporal grid in the MGRIT hierarchy may be created. All coarser grid information is carried over to the new fine space-time mesh in order to satisfy AMR requirements.

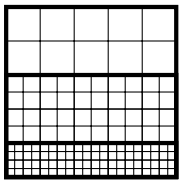
In addition, we note that the solution-adaptive nature of the SAMR method used here allows for dynamic refining and coarsening processes in space and time according to the physics-based criteria. Although the coarsening process is not demonstrated for the Stokes problem herein, we will demonstrate the full capability for CFD problems with multiphysics (such as a convection-diffusion-reaction case) in a follow-up study.



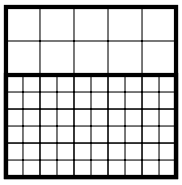
(a) Finer multigrid time grids are created by refinement in an FMG cycle



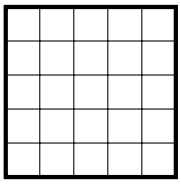
(b) After each space-time refinement, the coarser time grids are created by coarsening from the new finest time grid with a fixed coarsening factor.



(c) Spatial mesh at time grid 0



(d) Spatial mesh at time grid 1



(e) Spatial mesh at time grid 2

Fig. 4 (a) More refined time grids are created in an FMG cycle. (b)-(d) Each time point has a corresponding hierarchy of AMR meshes. An example spatial mesh is shown for time point t^4 at each level in the time grid.

III. The Multigrid Configuration for Time-Periodicity

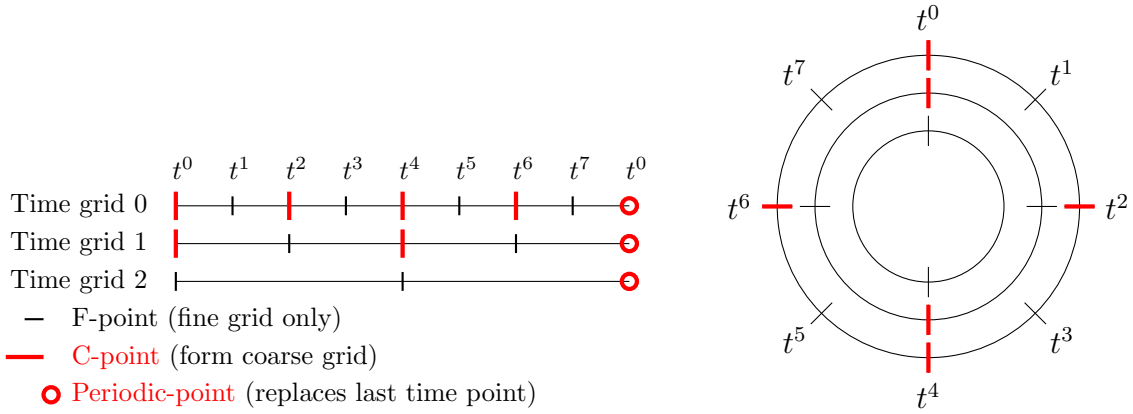
To implement time-periodicity, the matrix solved by MGRIT (Equation (2)) is modified so that the first and last points are treated as the same point,

$$\mathbf{A}\mathbf{U} \equiv \begin{pmatrix} \mathrm{I} & & & -\mathrm{I} \\ -\Phi & \mathrm{I} & & \\ & \ddots & \ddots & \\ & & -\Phi & \mathrm{I} \end{pmatrix} \begin{pmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{N_t} \end{pmatrix} \equiv \boldsymbol{g} \,. \tag{3}$$

This modification requires that the periodic time point be aligned for all coarser temporal levels, as shown in Figure 5a on a linear mesh or more suggestively in Figure 5b on a circular mesh. In order to meet this condition, the number of time points on the finest temporal grid must be a multiple of the coarsening ratios. The total coarsening factor is computed by

$$n = \prod_{\ell}^{L} m_\ell \,, \tag{4}$$

for the coarsening factor m on level ℓ out of L total levels. Therefore, the total number fine time points, N_t, must be some multiple of n. If the total number of fine time points does not meet this criteria after user-specified refinement, extra time points are added to the fine time grid until this condition is met. If additional time points are added, they are distributed evenly across the temporal domain so that no global communication requirements are imposed.



(a) The temporal grids showing the periodic time point **(b) A time-periodic temporal grid**

Fig. 5 Modification to the temporal grid so that the final time point is a periodic time point.

As with non-periodic cases, a time domain is specified on $[t^0, t^{N_t}]$ with N_t number of time steps. Special consideration is given to the final time interval from $i = N_t - 1$ to $i = N_t$. Since the last time point $i = N_t$ is treated as the same time point as the first time point (Equation (3) or as shown in Figure 5b for $N_t = 8$), it is assigned the same time value as the initial time point. To provide a consistent interface to applications, the time step from $i = N_t - 1$ to the periodic time point is treated as follows. The user application is presented with a time step from $i = N_t - 1$ to $i = N_t$ and the usual Δt step size. After the user application has stepped to $i = N_t$, the solution of $i = N_t$ is copied to $i = 0$ and the time value is reset to that of t^0. This initial point is then treated as any other C-point in the multigrid hierarchy. That is, it undergoes prolongation and restriction and receives coarse grid corrections.

In a non-periodic time-parallel case, the initial condition of the first time point is treated the same as the initial condition of time-sequential cases. Additional initial conditions at the remaining C-points can be specified [10] or be assigned a randomly generated solution. In our coupled MGRIT-AMR algorithm, the FMG process adaptively creates levels from a single coarse space-time grid (see Christopher et al. [2] for more discussion on this process). When MGRIT has a single temporal level, only a single C-point at the initial time point is created. This means that our coupled MGRIT-AMR algorithm requires only a single initial condition, located at the initial time point. With time-periodic cases, no initial conditions are required and the initial condition may simply be a random solution. However, specifying an initial condition similar to the final solution can help reduce the number of MGRIT iterations required for convergence.

IV. Results

Stokes second problem is used to demonstrate the capability of the adaptive space-time algorithm for solving problems with periodicity in time. A diagram of the computational domain is shown in Figure 6. An infinitely long flat, no-slip wall at $y = 0$ is periodically oscillating with a harmonic motion in the x-direction. A slip wall is located at $y = h$. The harmonic oscillations are proscribed by the function

$$u(0,t) = U_0 \cos(\omega t) \,, \tag{5}$$

where $U_0 = \mathrm{Re}\, \nu/h$ with $\mathrm{Re} = 20000$ being the Reynolds number, $\nu = 1.46 \times 10^{-5}\,\mathrm{m\,s^{-2}}$ the kinematic viscosity, $h = 0.1\,\mathrm{m}$ the distance between the two parallel flat walls, and $\omega = 2\pi/7$ being the oscillation frequency. The exact solution of the velocity distribution is known as $u(0,t) = U_0 \exp^{-\eta y} \cos(\omega t - \eta y)$ with $\eta = \sqrt{\omega/(2\nu)}$. With the exact solution, the algorithm accuracy can be easily assessed.

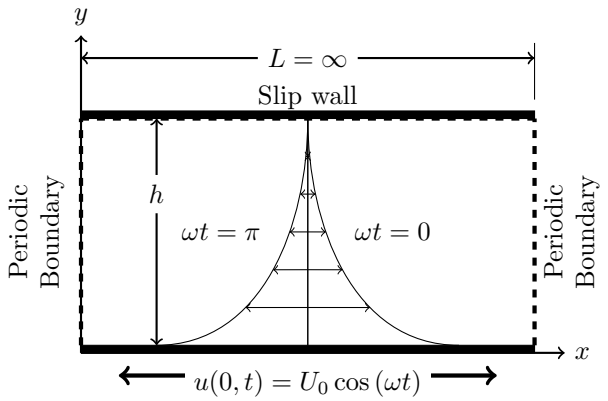


Fig. 6 A diagram of Stokes second problem.

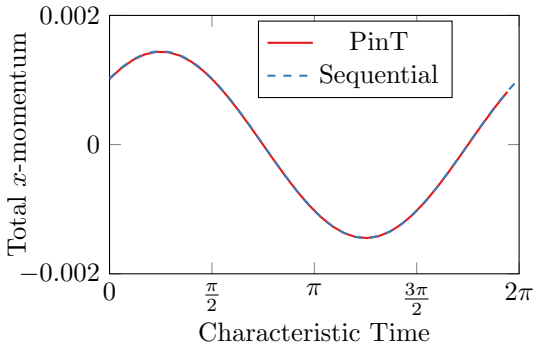
First, XBraid-Chord is expected to maintain the fourth-order accuracy of the underlying finite-volume method, which is verified by the convergence rates of error norms as shown in Table (1) and Table (2). The tables list the solution error norms (L_1, L_2, and L_∞) for six consecutive grids, with the spatial resolution ($\min(\Delta y, \Delta y)$) varying from $\Delta y = 6.25 \times 10^{-3}$ m to $\Delta y = 1.953\,125 \times 10^{-4}$ m. In addition, the error reduction rates are calculated for every two consecutive grids and clearly, both the time-sequential (Table (1)) and time-parallel (Table (2)) algorithms converge to the fourth-order accuracy as the mesh is sufficiently refined.

Table 1 Errors and convergence rate of Stokes second problem for Chord

Spatial Mesh Size	L_1 Error	L_1 Rate	L_2 Error	L_2 Rate	L_∞ Error	L_∞ Rate
16^2	2.00×10^{-7}	-	3.85×10^{-6}	-	1.14×10^{-4}	-
32^2	4.10×10^{-8}	2.29	9.61×10^{-7}	2.00	3.57×10^{-5}	1.67
64^2	5.27×10^{-9}	2.95	1.50×10^{-7}	2.68	7.84×10^{-6}	2.17
128^2	4.48×10^{-10}	3.56	1.33×10^{-8}	3.49	7.71×10^{-7}	3.35
256^2	3.24×10^{-11}	3.79	9.75×10^{-10}	3.77	5.84×10^{-8}	3.72
512^2	2.17×10^{-12}	3.90	6.62×10^{-11}	3.88	3.98×10^{-9}	3.87

Table 2 Errors and convergence rate of Stokes second problem for the space-time parallel algorithm

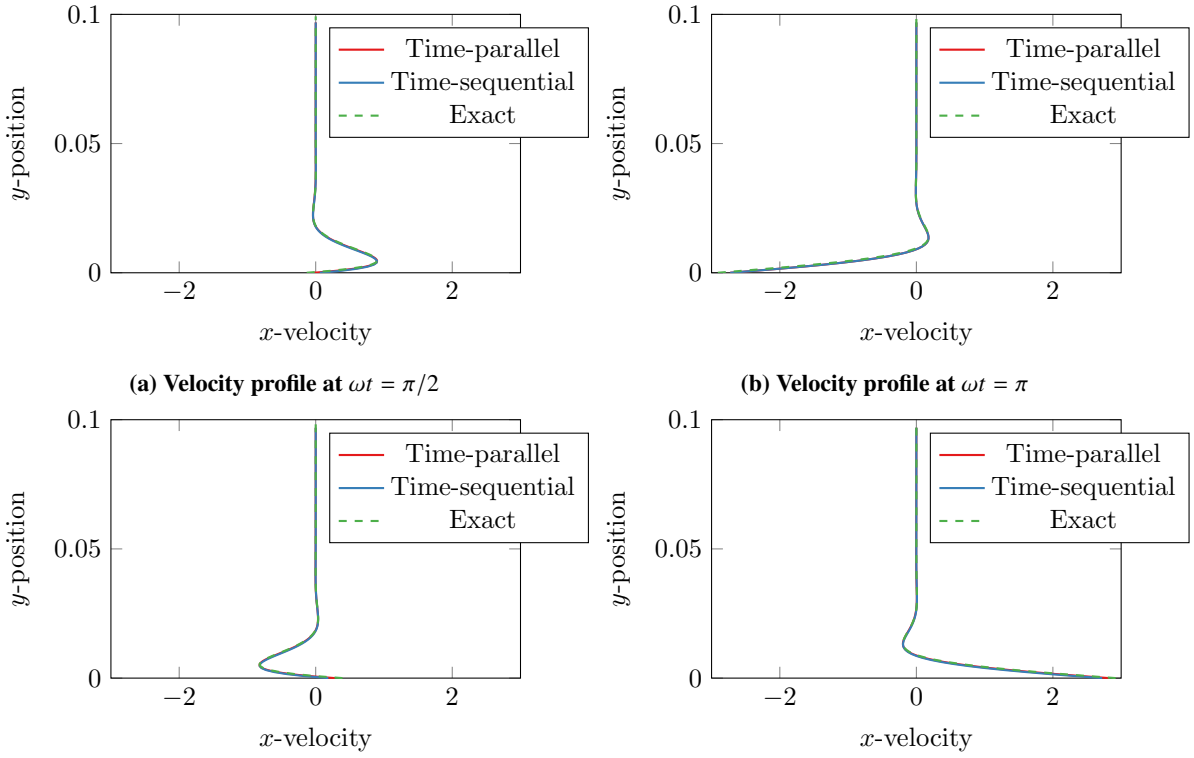
Spatial Mesh Size	L_1 Error	L_1 Rate	L_2 Error	L_2 Rate	L_∞ Error	L_∞ Rate
16^2	2.77×10^{-5}	-	5.32×10^{-4}	-	1.70×10^{-2}	-
32^2	3.56×10^{-6}	2.96	7.47×10^{-5}	2.83	2.59×10^{-3}	2.71
64^2	4.33×10^{-7}	3.04	9.98×10^{-6}	2.90	3.76×10^{-4}	2.78
128^2	3.72×10^{-8}	3.54	9.24×10^{-7}	3.43	3.75×10^{-5}	3.33
256^2	1.29×10^{-9}	4.85	6.20×10^{-8}	3.90	5.19×10^{-7}	3.79
512^2	6.62×10^{-11}	4.28	3.66×10^{-9}	4.08	3.65×10^{-8}	3.83



[H]

Fig. 7 Comparing total x-momentum with time-sequential and time-parallel.

Second, the coupled MGRIT-AMR algorithm is expected to maintain the conservation property. The SAMR methods performs flux-correction to maintain single-valued flux at spatial coarse-fine interfaces to ensure the conservation. The introduction of SAMR to the PinT algorithm should still preserve this property. Figure 7 shows the comparison of the conserved quantity, x-momentum, between the time-sequential and time-parallel solutions. In response to the oscillating wall, the total x-momentum in the domain varies according to the wall's movement. As seen in the figure, the parallel-in-time algorithm correctly tracks x-momentum over time as does the time-sequential algorithm.



(a) Velocity profile at $\omega t = \pi/2$

(b) Velocity profile at $\omega t = \pi$

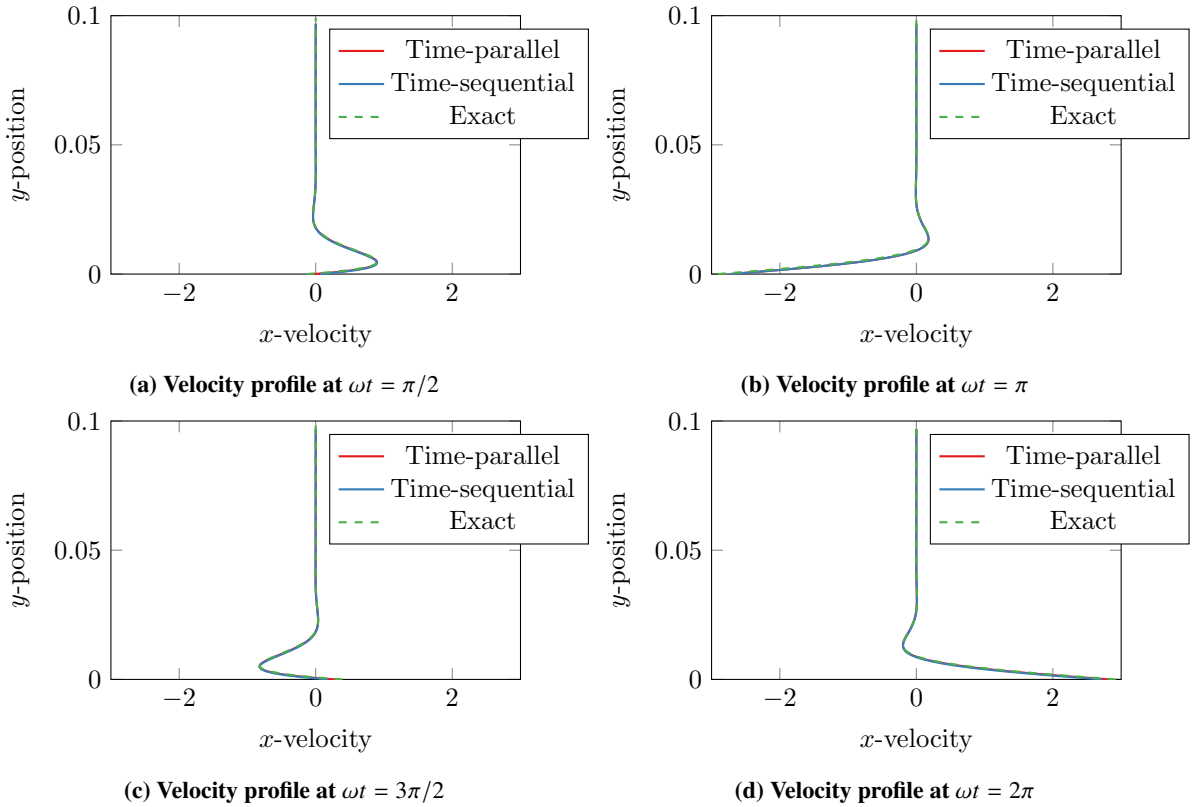
(c) Velocity profile at $\omega t = 3\pi/2$

(d) Velocity profile at $\omega t = 2\pi$

Fig. 8 Velocity profiles at the characteristic times after one iteration (time-parallel) or time interval (time-sequential).

The numerical solutions at four characteristic times are compared for the time-parallel and time-sequential algorithms against the analytic solution. We are interested in examining the solution details before and after convergence. First, in Figure 8, is the time-parallel solution after one iteration of the MGRIT code compared to one timer interval of the time-sequential solution. These velocity profiles are taken through the center line ($x = 0$) of the domain. At each

characteristic time, the velocity profiles already closely approximate the exact solution, though further iterations need to be performed to reduce the error. Figure 8a and Figure 8c show profiles taken at the moment when the wall speed is zero and the momentum of the fluid continues to push the fluid in the positive x-direction and negative x-direction, respectively. The error is at its maximum in this high gradient region. Conversely, in Figure 8b and Figure 8d the wall is at its maximum positive x-direction and negative x-direction speed, respectively. Again the error is mostly concentrated in the high gradient region near the wall.



(a) Velocity profile at $\omega t = \pi/2$

(b) Velocity profile at $\omega t = \pi$

(c) Velocity profile at $\omega t = 3\pi/2$

(d) Velocity profile at $\omega t = 2\pi$

Fig. 9 Velocity profiles at the characteristic times after convergence.

Figure 9 shows the velocity profiles after convergence is reached in both the time-parallel and time-sequential cases. This required four MGRIT iterations for the time-parallel case and ten time intervals for the time-sequential case. In all characteristic times, the time-parallel and time-sequential solutions track equally well to the exact solution. The error is still largest in the high gradient regions near the wall but is significantly lower than in the first iteration or time interval.

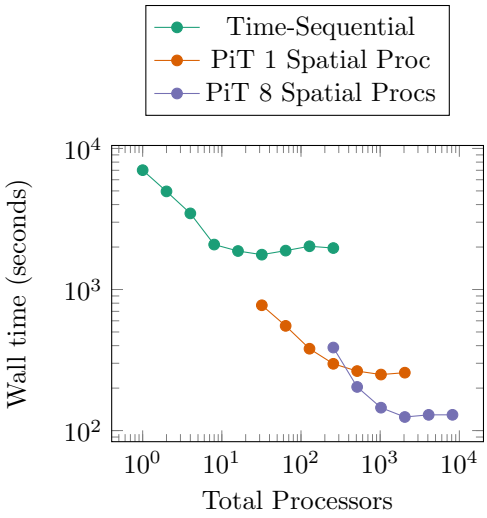


Fig. 10 Strong scaling study comparing time-sequential with time-parallel.

A strong scaling test is performed to compare the speedups obtainable by time-parallelization. A fixed problem size is used with a 16^2 base spatial mesh and four levels of refinement, leading to a five level AMR and multigrid hierarchy. Using the maximum stable time-step size on all levels leads to a total of 8960 time points on the finest temporal grid. For the time-sequential algorithm, the spatial parallelization is increased from 1 to 256 processors. Two different time-parallel cases were run; the first solely parallelizes the time domain while the second uses 8 processors for spatial parallelization. The results are shown in Figure 10 for all three cases. Spatial parallelization saturates fairly quickly with this case, the maximum speedup due to spatial parallelization occurred at 4 processors in space. When only parallelizing the temporal domain, saturation occurs around 1024 processors and is 7.0× faster than time-sequential. When combining space and time parallelization, saturation was reached at 4096 processors with a speedup of 13.7× over the time-sequential case.

V. Concluding Remarks

In this study we extend MGRIT to efficient solves of time-periodic CFD problems. We treat the first and last time point as the same time point to create periodic time grids. Additional consideration is made for the periodic time point to provide a consistent interface to user codes. The adaptive parallel in space-time method is validated by the Stokes second problem. The space-time parallel method maintains the fourth-order accuracy and the conservation property of the underlying finite-volume method. The time-parallel solution reproduces the time-sequential one and both compare well to the analytical solution. A strong scaling study shows that the time-periodic case is able to achieve significant speedups over the time sequential case. Speedups of up to 13.7× were achieved with the tested configurations.

Future work includes additional performance tuning and load balancing that needs to be performed for time-periodic cases. Also of interest is studying more complex fluid dynamics, for example introducing a source term and problems with convective physics. Eventually the goal of our work is to apply these methods to LES studies of turbomachinery.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-CONF-800089).

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of

authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

References

[1] Christopher, J., Gao, X., Guzik, S. M., Falgout, R. D., and Schroder, J. B., "Space-Time Adaptivity with Multigrid Reduction in Time for the Compressible Navier-Stokes Equations," *Proc. 19th Copper Mountain Conference On Multigrid Methods*, 2019, pp. 1–10. LLNL-CONF-765114.

[2] Christopher, J., Gao, X., Guzik, S. M., Falgout, R. D., and Schroder, J. B., "Fully Parallelized Space-Time Adaptive Meshes for the Compressible Navier-Stokes Equations Using Multigrid Reduction in Time," *Computing and Visualization in Science*, 2019. Submitted. LLNL-JRNL-798697.

[3] "XBraid: Parallel Time Integration with Multigrid," , 2019. `https://computation.llnl.gov/projects/parallel-time-integration-multigrid`.

[4] Gao, X., Guzik, S. M. J., and Colella, P., "Fourth Order Boundary Treatment for Viscous Fluxes on Cartesian Grid Finite-Volume Methods," AIAA 2014-1277, 52nd AIAA Aerospace Sciences Meeting, 2014.

[5] Guzik, S. M., Gao, X., Owen, L. D., McCorquodale, P., and Colella, P., "A Freestream-Preserving Fourth-Order Finite-Volume Method in Mapped Coordinates with Adaptive-Mesh Refinement," *Comput. Fluids*, Vol. 123, 2015, pp. 202–217.

[6] Gao, X., and Guzik, S. M. J., "A Fourth-Order Scheme for the Compressible Navier-Stokes Equations," AIAA 2015-0298, 53rd AIAA Aerospace Sciences Meeting, 2015.

[7] Gao, X., Owen, L. D., and Guzik, S. M. J., "A Parallel Adaptive Numerical Method with Generalized Curvilinear Coordinate Transformation for Compressible Navier-Stokes Equations," *Int. J. Numer. Meth. Fluids*, Vol. 82, 2016, pp. 664–688.

[8] Guzik, S. M., Gao, X., and Olschanowsky, C., "A high-performance finite-volume algorithm for solving partial differential equations governing compressible viscous flows on structured grids," *Comput. Math Appl.*, Vol. 72, 2016, pp. 2098–2118.

[9] Adams, M., Colella, P., Graves, D. T., Johnson, J. N., Johansen, H. S., Keen, N. D., Ligocki, T. J., Martin, D. F., McCorquodale, P. W., Modiano, D., Schwartz, P. O., Sternberg, T. D., and Van Straalen, B., "Chombo Software Package for AMR Applications - Design Document," Tech. Rep. LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[10] Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B., "Parallel Time Integration with Multigrid," *SIAM J. Sci. Comput.*, 2014.

[11] Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B., "Parallel Time Integration with Multigrid," *SIAM J. Sci. Comput.*, Vol. 36, No. 6, 2014, pp. C635–C661.

[12] Falgout, R. D., Friedhoff, S., Vandewalle, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B., "Multigrid Methods with Space-Time Concurrency," *Computing and Visualization in Science*, Vol. 18, No. 4-5, 2017, pp. 123–143.

[13] Falgout, R. D., Katz, A., Kolev, T. V., Schroder, J. B., Wissink, A., and Yang, U. M., "Parallel Time Integration with Multigrid Reduction for a Compressible Fluid Dynamics Application," Tech. Rep. LLNL-JRNL-663416, Lawrence Livermore National Laboratory, 2015.

[14] Colella, P., Dorr, M. R., Hittinger, J. A. F., and Martin, D. F., "High-order finite-volume methods in mapped coordinates," *J. Comput. Phys.*, Vol. 230, 2011, pp. 2952–2976.

[15] Zhang, Q., Johansen, H., and Colella, P., "A Fourth-Order Accurate Finite-Volume Method with Structured Adaptive Mesh Refinement for Solving the Advection-Diffusion Equation," *SIAM J. Sci. Comput.*, Vol. 34, No. 2, 2012, pp. B179–B201. doi:10.1137/110820105, `http://epubs.siam.org/doi/10.1137/110820105`.

[16] Martin, D., Colella, P., Anghel, M., and Alexander, F., "Adaptive Mesh Refinement for Multiscale Nonequilibrium Physics," *Comput. Sci. Eng.*, Vol. 7, No. 3, 2005, pp. 24–31. doi:10.1109/MCSE.2005.45.

[17] Martin, D. F., Colella, P., and Graves, D., "A Cell-Centered Adaptive Projection Method for the Incompressible Navier–Stokes Equations in Three Dimensions," *Journal of Computational Physics*, Vol. 227, No. 3, 2008, pp. 1863–1886. doi:10.1016/j.jcp.2007.09.032.

[18] Owen, L. D., Guzik, S. M., and Gao, X., "A high-order adaptive algorithm for multispecies gaseous flows on mapped domains," *Comput. Fluids*, Vol. 170, 2018, pp. 249 – 260. doi:https://doi.org/10.1016/j.compfluid.2018.05.010, URL `http://www.sciencedirect.com/science/article/pii/S0045793018302536`.