SPACE-TIME ADAPTIVITY WITH MULTIGRID REDUCTION IN TIME FOR THE COMPRESSIBLE NAVIER-STOKES EQUATIONS [*]

J. CHRISTOPHER [†]

In collaboration with: X. Gao, S. M. Guzik, R. D. Falgout, J. B. Schroder

Abstract. A parallelized fully-adaptive space-time mesh refinement algorithm using multigrid reduction-in-time (MGRIT) is applied to the unsteady compressible Navier-Stokes equations to solve fluids problems. Previously, fully-adaptive space-time methods have primarily used sequential time marching to integrate the time domain. Despite parallelization in the spatial domain, wall clock times have remained high due to the computational cost per time-step and the large number of desired time-steps. Furthermore, architectural trends in high-performance computing have shifted from ever-increasing clock speeds to greater concurrency. This motivates a need to parallelize the time domain. The spatial parallelization consists of a partitioning of the domain into a nested hierarchy of Cartesian grids that is adaptively refined at regions and flow features of interest. The MGRIT algorithm is demonstrated using an explicit time integration scheme applied to Couette flow, where error is compared to the analytic solution and a convergence study is performed. The eventual goal, after further development and optimization, is to conduct a performance comparison against a sequential-in-time version of the algorithm.

Key words. Multigrid Reduction in Time, Multigrid, Parallel-in-Time, Adaptive Space-Time Mesh Refinement, Finite Volume, Compressible Navier-Stokes, XBraid, Chombo

1. Introduction.

Annual increases in processor speeds have stalled from historic trends, so shifts in high-performance computing are leading towards increasing core counts and greater concurrency in computations. This shift comes at a time of research into higher-order finite-volume methods, which has a larger computational cost per time step compared to lower-order finite volume methods. Furthermore, in order to accurately resolve fast time-scale physics, small time steps are required, which leads to a large number of time steps that must be taken to reach a solution time of interest for many fluids applications. These challenges point to sequential time integration being a bottleneck that must be overcome for the next generation of fluids solvers.

Structured adaptive mesh refinement (SAMR) has been developed through several projects such as AMReX[1], Chombo[4], and SAMRAI[2]. These projects create a space-time adaptive mesh through subcycling the finer time steps, but solve the time domain sequentially. Parallelization of the spatial domain can easily reach saturation, either due to the spatial domain being small enough that further parallelization does not actualize further speedup, or due to having a large enough spatial domain that the parallelization overhead becomes the limiting factor[7].

This study explores coupling the space-time adaptivity of a SAMR algorithm to the time-parallel solve of the MGRIT algorithm[8]. Previous work has applied MGRIT algorithms to the unsteady compressible Navier-Stokes equations[9] with fixed grids. Spatial adaptivity with MGRIT has been explored for linear advection and inviscid Burgers equations[17]. Temporal non-uniform refinement was developed for the solution of the heat equation[16] to create adaptive meshes. Several challenges with coupling the a SAMR algorithm to an MGRIT algorithm are addressed in this study. One of the main features of SAMR-type algorithms is subcycling of the nested

[*] 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-765114).

[†] Colorado State University, (joshuac@rams.colostate.edu, http://cfdlab.colostate.edu/).

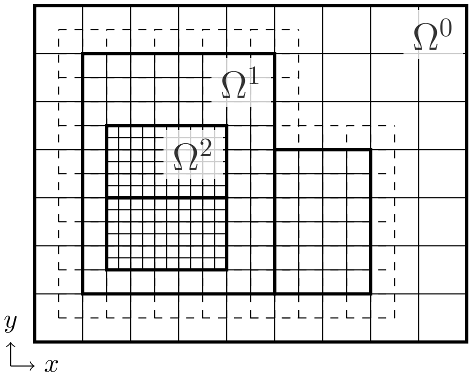


Fig. 2.1: Spatial grids are nested or overlaid upon coarser grids.

grid hierarchy. Subcycling is a method where coarse grids are advanced before fine levels, then the coarse solution is used as a predictor for information at the coarse-fine level interface. Under MGRIT's parallel-time solve, the coarse grids are not necessarily advanced ahead of fine grids which disrupts the coarse predictor in subcycling and makes ensuring conservation at coarse-fine interfaces difficult. The time propagation, restriction, and prolongation operators must be developed so that the SAMR and MGRIT algorithms can be compatible with each other.

The remainder of this section discusses the mathematical framework used in this study. Section 2 is a description of the spatial discretization and space-time mesh construction, followed by a presentation of the MGRIT algorithm. The fully-adaptive space-time algorithm developed for this study is described in Section 3, along with some of the challenges in coupling a SAMR code to an MGRIT algorithm. Section 4 presents supporting numerical results and Section 5 presents concluding remarks.

2. Background Information.

The computational domain is partitioned into a spatial and temporal discretization. A spatially parallelized fourth-order finite-volume method is used for the spatial discretization, and the standard fourth-order Runge-Kutta method is used for the time discretization which is parallelized with MGRIT.

2.1. Spatial Domain.

The software used for the spatial discretization is implemented in the numerical framework Chord[13, 12, 11], a fourth-order finite-volume library and solver for the Navier-Stokes equations. Chord makes use of Chombo[5], a library providing highly-parallelized solution adaptivity. Chord solves the mapped, transient, compressible Navier-Stokes equations with species transport and chemical reactions, but for the present study mapping, multiple species, and chemical reactions are disabled to simplify the problem space. Solution adaptivity is implemented as a nested hierarchy of overlapping grids such that finer grids are only dependent on the next coarser level for providing information at the interface of the finer level. An example of the nested hierarchy of grids is shown for three levels in Figure 2.1. Surrounding level Ω^1 is a ring of "invalid" ghost cells, which will be discussed shortly.

Consider the Navier-Stokes equations grouped into time derivatives, hyperbolic

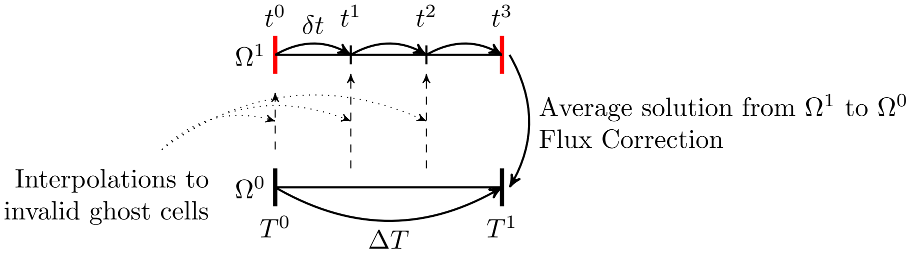


Fig. 2.2: Subcycling is the process by which many smaller time steps are taken for stability-limited fine grids.

flux, and elliptic flux,

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\nabla} \cdot \left(\vec{\mathbf{F}} - \vec{\mathbf{G}} \right) = 0 \, ,$$

with hyperbolic flux $\vec{\mathbf{F}}$ and elliptic flux $\vec{\mathbf{G}}$. The semi-discrete form of the Navier-Stokes equations is,

$$\frac{d}{dt}\langle \mathbf{U} \rangle_{\boldsymbol{i}} = -\frac{1}{\Delta x^d} \sum_{d=1}^{D} \left(\left(\langle \vec{\mathbf{F}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \vec{\mathbf{F}} \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right) - \left(\langle \vec{\mathbf{G}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \vec{\mathbf{G}} \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right) \right) \, , \tag{2.1}$$

where face-averaged quantities are represented with $\langle \cdot \rangle$. Face-averaged quantities and gradients for flux evaluation are computed using a fourth-order center-differencing scheme. The Piece-wise Parabolic Method limiter[15] is used to limit face-averaged quantities near strong solution discontinuities or shock waves. An upwind scheme is then used to resolve the Riemann problem that results from the limiting. Currently the fourth-order Runge-Kutta method is used to integrate the semi-discrete form sequentially in time.

Since an explicit time marching method is used to advance the time domain sequentially, a smaller time step must be taken for the more refined grids to maintain stability. Chord limits the time step size automatically, using a combination of the CFL condition for the hyperbolic flux and the von Neumann number for the elliptic flux[11]. In order to ensure convergence of the multigrid algorithm, only problems that are elliptic in nature are studied in the present paper, so the time step size is effectively limited only by the von Neumann number.

To minimize the number of steps taken on the coarse spatial mesh (to make advancing the coarse spatial mesh cheaper), subcycling is employed. Shown in Figure 2.2 is the process of subcycling for two levels with a temporal refinement ratio of $n_{\text{ref}} = 3$. The solution on the coarse grid may take time steps of size ΔT, and the fine grid limited to fine time step $\delta t = \Delta T / n_{\text{ref}}$. Subcycling is simply advancing the coarse spatial solution first by ΔT, then advancing the fine spatial solution n_{ref} times by δt. As shown in Figure 2.2, there are some additional operations to support the coupling of the two levels, namely interpolations to “invalid” ghost cells, fine solution averaging, and flux correction. Invalid ghost cells are the ring of cells that surround fine levels Ω^i for $i > 0$ (and shown with dashed lines around Ω^1 in Figure 2.1) that allow centered differencing operators to be used throughout the domain. These invalid ghost cells need solution values interpolated from the coarse grid at some intermediate time

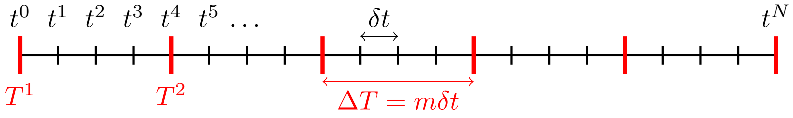


Fig. 2.3: Time grids for two levels in the MGRIT method, C-points shown in red and F-points shown in black. The C-points are the points of the coarse grid, and the composite of the C-points and F-points are the points of the fine grid.

$T^0 \leq \varphi < T^1$. Fine grid solutions are assumed to be more accurate than coarse grid solutions, so at the end of the subcycle the fine grid solution replaces the coarse grid solution through averaging. Since the solution on the coarse grid is being replaced in some coarse grid cells, flux correction is performed on $\Omega^{\ell-1}$ at the interface between Ω^{ℓ} and $\Omega^{\ell-1}$ to ensure the flux is single-valued across the interface. These operations need to be considered for creating a fully-adaptive space-time algorithm with MGRIT, so they are further discussed in Section 3.

2.2. Temporal Domain. Multigrid-reduction-in-time is a non-intrusive method used to solve and parallelize the time domain. For this study the MGRIT implementation used is the open-source library XBraid[3]. A temporal mesh is defined with $t^i = i\delta t$, for $i = 0, \ldots, N_t$, $\delta t = T/N_t$, and the solution state $\mathbf{U}_i \approx \mathbf{U}(t^i)$. A general one-step time discretization method can be represented by,

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

where Φ_i is the time integration method, and $\boldsymbol{g}_i$ represents solution-independent terms. For a linear time propagator this can be written as the application of a sequential forward solve of a system $A\mathbf{U} = \boldsymbol{g}$ as detailed by [8]. MGRIT solves this system in parallel using an $O(N_t)$ multigrid reduction iterative method.

The MGRIT method works by coarsening the time domain with a factor $m > 1$ to create a coarse time grid $T^{i_c} = i_c\Delta T$ with the coarse time step $\Delta T = m\delta t$ and coarse time points $i_c = 0, 1, \ldots, N_T = N_t/m$. A coarse time propagator $\Phi_{i_c,\Delta T}$ takes steps that are ideally cheaper to apply than the fine time propagator. The solution on the coarse time grids provides a coarse grid error correction used to accelerate the solution on the fine time grid. The relaxation provides a complementary error correction on the fine grid. The time grid is partitioned into C-points, which exist on both the fine and coarse time grid, and F-points which exist only on the fine time grid. These time points are shown in Figure 2.3 with C-points marked in red.

Relaxation is accomplished with FCF-relaxation, which is the successive applications of F-relaxation and C-relaxation. F-relaxation updates the F-point values $\mathbf{U}_i$ on the interval (T^i, T^{i+1}) by propagating the C-point value at T^i in sequence. Each of the intervals are independent of each other and are computed in parallel. C-relaxation completes the interval by propagating the value from F-point $\mathbf{U}_{mi_c-1}$ to C-point $\mathbf{U}_{mi_c}$. Each of these C-points may also be updated in parallel. Prolongation and restriction operators are discussed in Sections 3.2 and 3.3, respectively.

3. Space-Time Adaptivity. The algorithms described in Section 2 now need to be coupled. Recall from Section 2.1 that the solution on fine spatial grids is advanced at smaller time steps than on coarse spatial grids using subcycling. Likewise, the finer time domains in the MGRIT algorithm are advanced with a smaller time step than the coarse time domains. It seems natural to map the multigrid time levels to the SAMR spatial levels, one-to-one.

Mapping is done such that the coarsest spatial level, Ω^0, maps to the coarsest time grid. The first refined spatial level, Ω^1, maps to the first finer time grid, and so on for the number of levels used. When MGRIT solves the finest level, ideally only the finest spatial level is solved. Likewise for all of the coarser levels. For the remainder of this paper, Ω^ℓ denotes the coupled MGRIT-SAMR levels.

3.1. Space-Time Mesh Generation. To construct the space-time mesh, the initial coarsest space-time grid is specified as user input. An initial sweep of the coarse level "tags" time intervals for refinement and a finer space-time mesh is created. During the relaxation on the finer mesh, time intervals are tagged for refinement again and the process continues recursively until no more tagging occurs or the finest desired level is reached. After the finest space-time mesh is reached, temporal grids coarser than the finest level are generated by coarsening the temporal grid by a constant factor. Coarser spatial grids are not redefined like the temporal grid is. Instead, the spatial grid is stored for each time interval and the solution is transformed to the spatial grid as described in the following sections.

Tagging is problem specific but is generally based on physical quantities such as sharp solution gradients[10] or on truncation error estimates[14]. The process of inspecting the solution and tagging cells for refinement occurs at every time point of the coarser level. In a time interval containing cells tagged for refinement, finer spatial grids will be created for the interval using the Berger-Rigoutsos[6] algorithm. Temporal refinement adds the number of time points to the interval required to maintain stability of the time integration method on the finer spatial grid.

3.2. Prolongation-Refinement. Solution transfer from the coarse temporal grid to the fine temporal grid is simply injection at the C-points from the coarse temporal grid to the fine temporal grid as described by Falgout et al.[8]. If the spatial grid has been refined at that C-point, then the solution is interpolated from the coarse spatial grid to the fine spatial grid using a fourth-order least-squares method described by McCorquodale and Colella[15]. If a finer spatial grid does not exist at that location (see Section 3.6), then the solution is copied to an identical spatial grid at the finer time point.

3.3. Restriction-Coarsening. Restriction in the time domain is, like prolongation, simply an injection at the C-points, now from the fine grid to the coarse grid. Spatial coarsening occurs by averaging the solution from the fine level to the coarse level such that the the solution value in a coarse cell is the average of the values of the overlapping fine cells[5]. Complementary to prolongation, if the coarser time point does not contain a coarser spatial mesh, the finest available solution is copied to an identical spatial mesh at the coarser time point.

3.4. Solution Propagation with Subcycling. As mentioned in Section 2.1, additional operations need to be performed to propagate the solution on the finer spatial grids. The first is an interpolation in time to the "invalid" ghost cells that surround the finer level. Some steps of the finer level are taken at times that are intermediate to the C-points, such as the advance from t^1 to t^2 in Figure 2.2. An interpolation of the coarse solution at either time T^0 or T^1 would lead to solution instabilities as well as lose the fourth-order accuracy of the algorithm. Therefore an interpolation in time using a fourth-order Taylor polynomial[15] constructed using the solution states at T^0 and T^1 and the stage values of the RK4 method.

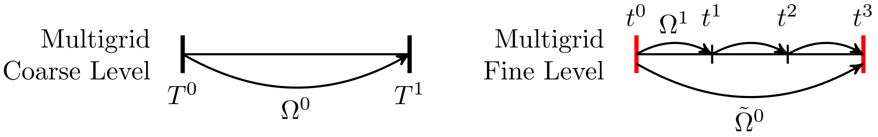


Fig. 3.1: Coarse solution data is stored locally at fine time points to support advancing the fine time steps. To advance the fine multigrid level solution, the locally stored coarse spatial level solution $\tilde{\Omega}^0$ must be advanced first.

Sequential SAMR algorithms require solving the coarse levels before the fine levels so that solution information at T^{mi} and $T^{m(i+1)}$ can be stored and made available to the time interpolator. This presents a problem for the MGRIT algorithm because when going down the multigrid cycle the coarse levels are evaluated after fine levels. Although solution information from previous multigrid iterations is available (for greater storage costs), it is not clear that previous iteration solution data would provide a good enough estimate of course solution data for the high order scheme. Similarly, solution information at a time point $T^{m(i+1)}$ is not available at T^{mi} or any time point on the fine time grid, so not enough information is available to perform this interpolation when going up the multigrid cycle either.

A related problem also occurs with the flux correction at coarse-fine spatial mesh interfaces. Flux correction is the process by which a single-valued flux, and thus conservation, is guaranteed at coarse-fine interfaces[5]. Although finite-volume methods are inherently conservative, this property of the method only works on a single grid. The nested hierarchy of the spatial meshes produces two different fluxes at the coarse-fine interface:

- flux calculated on the coarse grid that underlays the entire fine grid,
- flux calculated on the fine grid using ghost cell values interpolated from the coarse grid and values from the fine grid.

The coarse grid is conservative on the coarse grid, and the fine grid is conservative on the fine grid, but the composite grid is not necessarily conservative because coarse cell states are replaced by fine cell states in the regions they overlay. To correct for this difference, flux values from both grids are accumulated in a register during the time step (at each stage of the RK4 method, and for all subcycles of the finer grid), then reconciled at the end of the time step. As with the interpolation to ghost cells, this flux correction operation requires solution information from different time points which is not currently supported in the MGRIT implementation used.

To address these issues, at each C-point the finer level copies and stores the coarser grid state locally to the fine time point. These coarser "support" grid states are then advanced as needed by the finer grids to provide the support needed to advance the finer level. A representation of this process is shown in Figure 3.1, where the coarse state Ω^0 is copied at coarse time point T^0 to the "support" state $\tilde{\Omega}^0$ at the C-point t^0. Then, when FCF-relaxation occurs on the interval $[t^0, t^3]$, the locally stored coarse solution $\tilde{\Omega}^0$ is advanced first. This is a duplication of work, as the coarse solution is advanced once at the coarse time point, then again internally at the fine time point. These "support" steps are performed with subcycling so it is hoped that the additional computational cost is not too great. When doing prolongation-refinement and restriction-coarsening, coarser "support" states that are local to the fine time points are copied from or to the coarse time points, respectively.

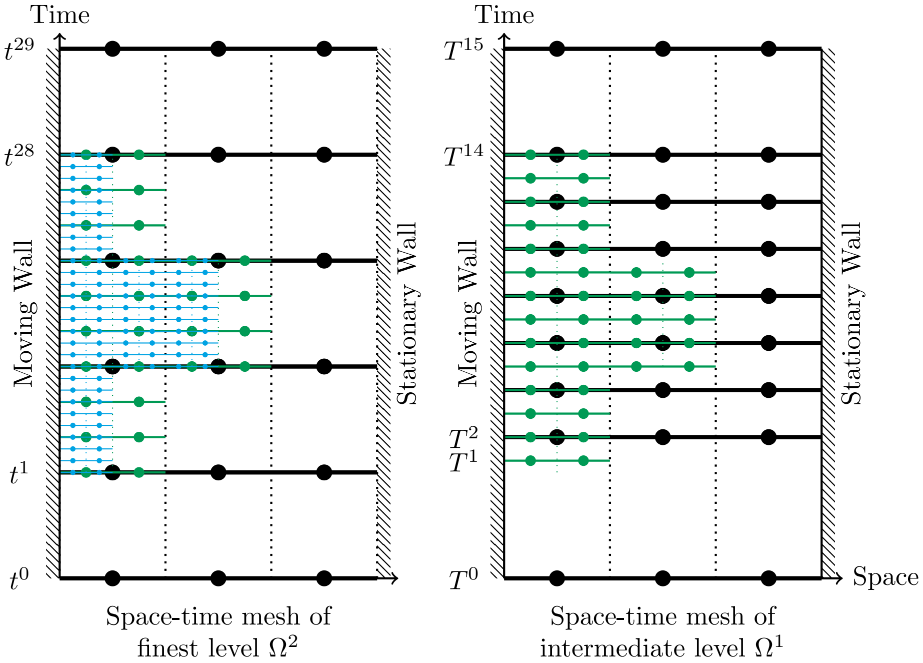


Fig. 3.2: A space-time mesh for three levels, finest level Ω^2 in blue, and an intermediate level Ω^1 in green. The C-points of the intermediate level correspond with the black time points.

An example of the space-time mesh showing the support states is depicted in Figure 3.2. In this example, the space-time mesh was refined twice with a factor of three in the temporal domain and a factor of two in the spatial domain. The finest space-time mesh, shown on the left as Ω^2, contains fine grids that are uniformly distributed within the time interval of its "support" states ($\tilde{\Omega}^1$ and $\tilde{\Omega}^0$ shown in green and black). This allows a direct application of the subcycling method as described above. Upon coarsening the finest space-time mesh to the intermediate level Ω^1, the distribution of the fine grid is no longer uniform within the time interval of its "support" state ($\tilde{\Omega}^0$ in black). In Figure 3.2, this only occurred in the first coarse time interval yet it is possible for this to occur many times if the finest level isn't temporally grouped together. A single step cannot be used on level Ω^1 to integrate from time point T^0 to the time point T^1 because the step size is larger than the stability constraint for the explicit time marching method. To remedy this, multiple "internal" steps not represented on the space-time mesh and not used as part of the MGRIT method are taken to cover these larger gaps. The coarsest level, Ω^0, is not shown but consists of the same points as $\tilde{\Omega}^0$ in Ω^1 and will likely also need to take internal steps on intervals with a larger time step size than the stability constraint.

3.5. Regridding. Adaptivity in the spatial domain provides grids that may shift in the region they discretize at different time intervals. This means when advancing the solution during FCF-Relaxation, there will be a need to advance the solution through a time point where the grids change. This occurs in Figure 3.2 on the finest level Ω^2 at t^1. Algorithms exist in SAMR-type frameworks to map the solution $\mathbf{U}$

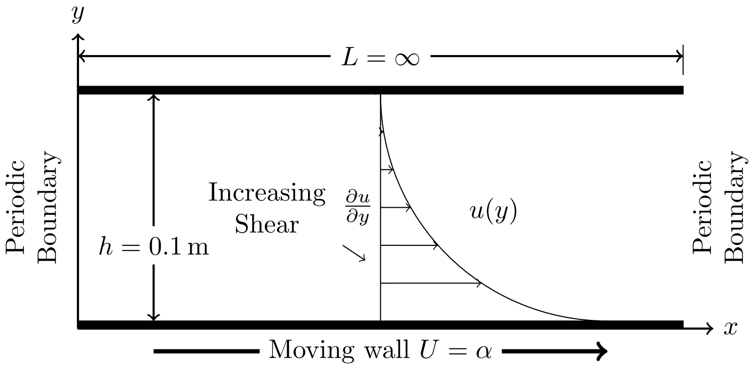


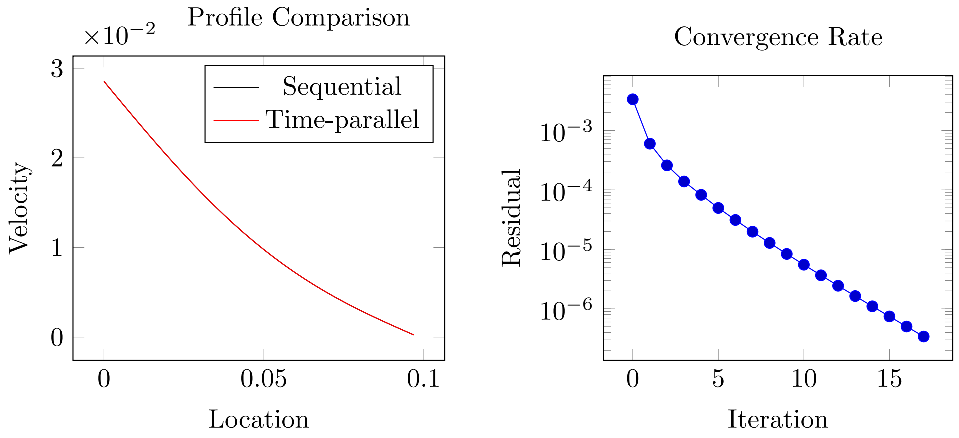
Fig. 4.1: Diagram of the transient Couette flow case showing boundary conditions and velocity profile at some point before steady state.

from one spatial grid to another[5], and they are reused here. In short, for regions of Ω^ℓ where the grid will no longer exist in the new time interval, the solution is averaged down. In regions where new grids are created for the new interval, the solution is interpolated up from $\Omega^{\ell-1}$ using the fourth-order least squares method mentioned in Section 3.2. If a region of Ω^ℓ will contain grids in both the old and new time intervals, the solution is copied identically (as the structured grid generator will create exactly the same cell locations in the same regions).

3.6. Unrefined Spatial Regions on Fine Temporal Levels. Some intervals on the fine composite time grid will not have fine spatial grids associated with it. Consider the space-time mesh of the finest level Ω^2 in Figure 3.2. On the fine interval $[t^0, t^1)$ there are no fine spatial grids, but to satisfy the multigrid scheme the solution needs to be advanced on that interval. In order to accomplish this, the coarse support grid solution $\tilde{\Omega}^0$ stored at t^0 and t^1 is advanced in this region instead.

4. Numerical Results. In order to verify and validate this space-time adaptive algorithm, the Couette flow test case is used. Couette flow is a pseudo one-dimensional flow governed by the Navier-Stokes equations. The flow is between infinite no-slip walls, with one wall moving and the other stationary. Figure 4.1 shows the boundary conditions and a representation of the velocity profile $u(y,t)$ at some time t between initial and steady state conditions. Flow is initialized at quiescent conditions and a velocity profile develops from the moving wall until steady state conditions are reached. At steady state, the shear stress in the fluid is constant throughout the domain, a linear velocity profile is reached with zero velocity at the stationary wall, and a velocity of the speed of the moving wall is reached at the other end of the domain.

The moving wall speed is $\alpha = \mathrm{Re}\,\nu/h$ where Re is the Reynolds number, and the dynamic viscosity is $\nu = 1.789 \times 10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-1}}$. The space-time mesh constructed for the Couette flow problem is similar to that shown in Figure 3.2. A base spatial grid of 32×32 cells is used with an initial 2000 time points. A two-level MGRIT solver was used for this case. The space-time mesh was constructed using an adaptive refinement with a spatial refinement ratio of two and a temporal refinement ratio of four. The MGRIT coarsening factor used was two. The coarsening factor is different from the refinement factor to help stress the algorithm. Tagging of the spatial domain is done by vorticity magnitude and the threshold is set slightly above the shear at



(a) Velocity profile comparison at the final time point. Profiles are overlapping.

(b) Plot of residual through the MGRIT iterations.

Fig. 4.2: Accuracy and convergence results from Couette flow.

steady state. After refinement the finest temporal grid contains 6470 time points. At the initial time point with quiescent conditions, no spatial refinement is done as there is no flow and no vorticity. As the shear grows during start up, the refined spatial grids grow from the moving wall towards the stationary wall. The refined spatial grids then recede and eventually vanish as the steady state condition is approached and the shear drops below the tagging threshold value.

To test the accuracy of the algorithm, the solution is compared at the final time point to both an analytic solution and the solution obtained by sequential time stepping. Shown in Figure 4.2a is a trace through the final time point plotted with the sequential solution; the solution profiles are overlapping. The L2-norm of the error of the parallel-time solution at the final time point is 4.80×10^{-8}. For comparison, the L2-norm error of the solution found with sequential time marching and the same space-time mesh is 4.24×10^{-8}.

The halting tolerance chosen for the MGRIT algorithm was 5×10^{-7} which was obtained in 17 iterations. The residual at each iteration is plotted in Figure 4.2b, showing a fairly consistent convergence rate. The residual decreases by a factor of around 0.5 for iterations 2–4, with the rate decreasing to around 0.7 by the last iteration.

5. Conclusions. In this study, the SAMR algorithm was coupled to the time-parallel MGRIT algorithm. New methods were developed to handle unique features of the SAMR algorithm such as the generation of the adaptive space-time mesh. Challenges with the subcycling process and its requirements for surrounding space-time solution information were addressed with the trade off additional storage and computational work. The developed parallel-in-time algorithm computes solutions with similar error as the sequential algorithm.

Future work includes developing optimizations to the space-time SAMR algorithm. For example, methods to reduce the duplicate work to support flux correction and invalid ghost cell time-interpolation need to be developed. Another area that needs further investigation is performing load balancing to ensure even distribution of work per processor. After optimizing, strong and weak scaling tests will be performed to determine the speedup by parallelizing the time domain. Lastly, the convergence rate of the MGRIT algorithm is passable but exploration on methods to improve the convergence rate should be undertaken.

REFERENCES

[1] *Amrex: Software framework for block structured amr.* https://amrex-codes.github.io/amrex/.

[2] *Samrai: Structured adaptive mesh refinement application infrastructure.* https://computation.llnl.gov/projects/samrai.

[3] *Xbraid: Parallel time integration with multigrid.*

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

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

[6] M. J. Berger and I. Rigoutsos, *An algorithm for point clustering and grid generation*, IEEE Transactions Systems, Man, and Cybernetics, (1991).

[7] A. Dubey, A. S. Almgren, J. B. Bell, M. Berzins, S. R. Brandt, G. Bryan, P. Colella, D. T. Graves, M. Lijewski, F. Löffler, B. O'Shea, E. Schnetter, B. van Straalen, and K. Weide, *A survey of high level frameworks in block-structured adaptive mesh refinement packages*, CoRR, abs/1610.08833 (2016).

[8] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., (2014).

[9] R. D. Falgout, A. Katz, T. V. Kolev, J. Schroder, A. M. Wissink, and U. Yang, *Parallel time integration with multigrid reduction for a compressible fluid dynamics application*, tech. report, Lawrence Livermore National Lab, 2014.

[10] X. Gao, *A parallel solution-adaptive method for turbulent non-premixed combusting flows*, PhD thesis, University of Toronto, 2008.

[11] X. Gao, L. D. Owen, and S. M. J. Guzik, *A parallel adaptive numerical method with generalized curvilinear coordinate transformation for compressible Navier-Stokes equations*, Int. J. Numer. Meth. Fluids, 82 (2016), pp. 664–688.

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

[13] S. M. Guzik, X. Gao, L. D. Owen, P. McCorquodale, and P. Colella, *A freestream-preserving fourth-order finite-volume method in mapped coordinates with adaptive-mesh refinement*, Comput. Fluids, 123 (2015), pp. 202–217.

[14] D. F. Martin, *An Adaptive Cell-Centered Projection Method for the Incompressible Euler Equations*, PhD thesis, University of California at Berkeley, 1998.

[15] P. McCorquodale and P. Colella, *A high-order finite-volume method for conservation laws on locally refined grids*, Comm. App. Math. Comput. Sci., 6 (2011), pp. 1–25.

[16] B. O'Neill, *Multigrid Reduction in Time for Nonlinear Parabolic Problems*, PhD thesis, University of Colorado, Boulder, 2017.

[17] H. D. Sterck, R. D. Falgout, A. J. M. Howse, S. P. MacLachlan, and J. B. Schroder, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations*, SIAM J. Sci. Comput, (2017). LLNL-JRNL-737050.