Parallel Time Integration – An Approaching Paradigm Shift for Scientific Computing *

Robert D. Falgout† Jacob B. Schroder‡

September 5, 2023

Abstract

This note argues that parallel-in-time methods will be necessary for doing high-fidelity time-dependent simulations in the future. A "proof" is given to support the argument and to provide a framework for debate. The effect of a parallel-in-time paradigm on scientific computing practice is also discussed.

1 Skepticism about parallel-in-time

Skepticism about the viability and necessity of parallel-in-time methods continues. Here are some of the questions and comments the authors have heard on this topic:

- *What about causality?* Parallel-in-time methods converge to the same space-time solution as time stepping, so causality is not broken.
- *What about shocks and discontinuities?* Coarse space-time problems used in parallel-in-time methods do not (by definition) need to capture fine-scale features. This has been demonstrated for both shocks [9] and discontinuities [5, 7].
- *PinT requires too much memory.* Eventually there will be ample resources. This is reminiscent of the transition from 2D-space to 3D-space in the 1990s. Also, it is possible to keep per-processor memory requirements nearly the same as time stepping [3].
- *What about hyperbolic and chaotic problems?* This is indeed a research area, but proof-of-principle has been demonstrated and progress continues [8, 10, 11, 12].
- *PinT algorithms have terrible parallel efficiencies.* Parallel efficiency alone can be misleading - slow codes are often highly efficient. Furthermore, if efficiency measures the time to solution on a given amount of resources, then sequential time stepping is often much less efficient than parallel-in-time.

*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-TR-851068).

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA, rfalgout@llnl.gov

‡Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, USA, jbschroder@unm.edu

Although parallel-in-time methods date back to the 1960s, it is only recently that research on this topic has taken off [4, 6]. As the research matures, these methods will become increasingly more practical.

2 The need for parallel-in-time

The following is a "proof" that parallel-in-time methods are inevitable for most time-dependent applications. The result relies on accepting four basic assumptions, which are meant to help frame the debate on this question and are discussed after the theorem. Note that n_t defines the *size of the temporal dimension* and is analogous to the number of time steps in a prototypical time-stepping application. There are also three concepts of time to distinguish: the temporal dimension in the simulation (lower-case t), simulation wall-clock time (capital T), and time itself (denoted τ), i.e., our past, present, and future.

Theorem 2.1. *Assume the following are true:*

1. *Scientists will continue to want higher fidelity simulations, requiring the size of the temporal dimension to increase indefinitely:*

$$n_t(\tau) \xrightarrow[\tau\to\infty]{} \infty$$

2. *Simulation time for time stepping is linear in n_t, where $T_0(\tau)$ is some atomic time:*

$$T_{seq}(\tau) \gtrsim n_t T_0(\tau)$$

3. *Simulation time for parallel-in-time is polylogarithmic in n_t:*

$$T_{pint}(\tau) \lesssim \log^q(n_t) T_0(\tau)$$

4. *Scientists have a threshold beyond which they will switch to a faster method:*

$$T_{seq} > \theta T_{pint} \quad \Rightarrow \quad \textit{switch to parallel-in-time}$$

Then, at some time in the future, scientists will adopt parallel-in-time.

Proof. Let $n = n_t$ to simplify notation. Since $\log(n) \sim \ln(n)$, use natural log without loss of generality. With

$$F(n) = \frac{n}{\ln^q(n)}, \qquad \frac{dF}{dn} = \frac{\ln(n) - q}{\ln^{q+1}(n)},$$

it is easy to verify that $F > 1$ for all $n > 1$, and F increases without bound for $n \in (e^q, \infty)$. Hence, for any $\theta_1 \geq 1$, there is a corresponding n_1 such that $F(n) > \theta_1$ for all $n > n_1$, and therefore the following holds for all $n > n_1$ with constants c_2, c_3 from assumptions 2, 3:

$$\frac{T_{seq}}{T_{pint}} \geq c_2 c_3 F(n) > \theta_1 c_2 c_3.$$

Choosing θ_1 such that $\theta_1 c_2 c_3 \geq \theta$, then $T_{seq} > \theta T_{pint}$ and the result follows from assumptions 1 and 4. □

Assumption 1: Scientists will continue to want higher fidelity simulations, requiring the size of the temporal dimension to increase indefinitely

The size of the temporal dimension affects simulation accuracy and is a function of the computer resources used. It is hard to imagine scientists not wanting to fully utilize available resources to improve simulation accuracy. The bigger question is about computer resource growth. Given that growth has been exponential and is expected to stay this way for the foreseeable future, it's reasonable to expect that this will continue for quite some time.

There are other useful ways to utilize computer resources to enhance simulation fidelity, for example, uncertainty quantification. More efficient use of memory through adaptive mesh refinement or reduced-order basis methods can also improve the accuracy of some simulations with respect to available resources. However, these techniques only slow or delay growth of n_t. They do not offset it.

Assumption 2: Simulation time for time stepping is linear in n_t

If n_t is the number of time steps and T_0 is proportional to the cost of doing a single time step, then this assumption clearly holds. Note that the assumption allows n_t to be something more exotic as long as it is proportional to the sequential temporal component of the code. The value of T_0 is partially determined by machine hardware characteristics, which change over time (e.g., clock speeds, memory/network latency/bandwidth). However, its value is also affected by the application and the application scientist. For example, a time step may depend linearly on the memory accessed (e.g., an explicit time step) or it may depend nonlinearly. Likewise, scientists may have different strategies for using the available machine resources. Writing T_0 as a function of τ allows for this variability over time.

Note that if $T_0 \sim 1/n_t$, then simulation time does not increase due to time stepping. This is what happened until about 2005 as a result of clock speed increases. It might then seem that developing approaches to decrease T_0 could circumvent the need for parallel-in-time methods, but the result in Theorem 2.1 does not depend on the value of T_0. Even though a smaller T_0 implies faster time stepping, it similarly implies faster parallel-in-time methods. The real driver is the increase in computer resources.

Assumption 3: Simulation time for parallel-in-time is polylogarithmic in n_t

Multigrid is one effective aproach for parallelizing in the time dimension. Multigrid methods have been shown to converge to discretization accuracy in time proportional to $\log^2(N)$ for some problems, where N is the problem size. For example, solve time for multigrid on parallel machines is determined by the number of grids visited, which defines the serial component of the method. The following two multigrid approaches converge to discretization accuracy with a serial component proportional to $\log^2(N)$ [2]:

- Full multigrid (FMG) - one cycle with $\log^2(N)$ grids visited;
- V-cycle multigrid - $\log(N)$ cycles with $\log(N)$ grids visited.

This is nearly optimal with respect to the $O(\log(N))$ lower bound derived in [13].

Multigrid-based parallel-in-time methods are also able to achieve the above polylogarithmic solution times for some classes of problems (e.g., diffusion). However, parallel-in-time

methods require more memory resources than time stepping. Hence, for a given memory footprint, parallel-in-time methods have a smaller value of n_t than time stepping, while the result in Theorem 2.1 relies on a common value.

To put the two methods on the same footing, consider a machine with available memory of size M and a space-time problem of size $n_s \times n_t$, where n_s and n_t are the "spatial" and temporal sizes respectively. Assume that $n_s \sim n_t^k$ with $k > 0$, and assume that the entire memory is used to solve the problem. Note that if $k = 0$, only a finite amount of memory and parallelism can be used in the spatial dimension, leaving only parallel-in-time as a means of exploiting additional resources. The diffusion problem mentioned above fits into this framework with $k = d/2$, where d is the number of spatial dimensions (assuming $\Delta t \sim \Delta x^2$). From this, utilizing all of the memory resources implies

$$\begin{array}{rll} T_{seq}: & M \sim n_s \sim n_t^k & \Rightarrow n_{t,seq} \sim M^{1/k} \\ T_{pint}: & M \sim n_s n_t \sim n_t^{k+1} & \Rightarrow n_{t,pint} \sim M^{1/(k+1)}. \end{array}$$

If the same space-time problem is solved in Theorem 2.1, then the memory used by sequential time stepping satisfies

$$n_t = n_{t,pint} \;\Rightarrow\; M_{seq} \sim M^{k/(k+1)} < M.$$

Although larger problems may be solved with sequential time stepping, the solve times are potentially much longer, and scientists may prefer to solve smaller problems faster with a parallel-in-time method. This tradeoff is captured in the threshold θ of Assumption 4. The above discussion is also valid when the parallel-in-time method is applied to successive sub-windows in time if the window size is proportional to n_t.

The main research issue here is developing parallel-in-time methods that have the right convergence behavior. Note that if these methods are not developed and T_0 remains fixed (as with current architectures), time stepping is a bottleneck and resources go unused.

Assumption 4: Scientists have a threshold beyond which they will switch to a faster method

If Assumption 1 holds and T_0 is fixed, then not having a threshold runs counter to the desire for higher fidelity simulations (it requires living with a growing bottleneck indefinitely, even when there is an alternative). If T_0 decreases indefinitely, then higher fidelity simulations could still be done, but scientist who switch to a faster method would be more competitive than those who do not. Hence, all scientists likely have a threshold, but the size of that threshold will vary depending on the relative importance of simulation time versus simulation accuracy and other considerations.

Achieving a threshold depends on the constants in the models and the availability of resources. Recall, for example, the scenario discussed above where $n_{t,seq} \sim M^{1/k}$. Applications with many spatial dimensions may not have enough resources today to reach a threshold and switch to parallel-in-time. Conversely, applications with fewer spatial dimensions and many time steps may benefit from parallel-in-time now, assuming appropriate methods exist.

3 A paradigm shift

The above "proof" argues that it's not a matter of *if* parallel-in-time methods will be needed, but *when*. Continued research and development in this area is essential, because scientific simulation with parallel-in-time methods is dramatically different from using traditional time stepping. Here are a few points of note on this approaching paradigm shift:

- Solver convergence in the time dimension becomes a new issue to consider. For time stepping methods, there is no distinction between the temporal discretization and the solution in the temporal dimension, since the latter is an exact solver.
- The choice of using an explicit versus implicit method may change, since the time to solve either with parallel-in-time is roughly the same (on the same space-time mesh). To see this, consider the following space-time stencils for an example 1D-space-time diffusion problem:

$$\text{explicit} = \begin{bmatrix} & * & \\ * & * & * \end{bmatrix} \qquad \text{implicit} = \begin{bmatrix} * & * & * \\ & * & \end{bmatrix}.$$

  These are both 4-point stencils, and from a multigrid solver point of view, there is no meaningful difference. They require similar cycle complexity to solve.
- Computational steering changes. Simulations are traditionally steered as the system evolves forward in time. With parallel-in-time, steering intervention would most likely occur at coarse temporal scales and apply to the entire space-time domain.
- Full space-time adaptivity will become commonplace. There are not many codes that currently use full space-time adaptivity. One exception is the structured adaptive mesh refinement community, which uses a technique called subcycling [1]. Most applications take a single time step for all spatial points, driven by the size of the smallest spatial resolution.
- Fully unstructured space-time grids becomes viable, that is, method of lines will no longer be needed.

References

[1] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of computational Physics, 82 (1989), pp. 64–84. 5

[2] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000. Second edition. 3

[3] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661. 1

[4] M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, eds., Springer, 2015, pp. 69–114. 2

[5] S. Günther, R. D. Falgout, P. Top, C. S. Woodward, and J. B. Schroder, *Parallel-in-time solution of power systems with unscheduled events*, in Power and Energy Society General Meeting (PESGM), 2020, IEEE, 2020. 1

[6] B. W. Ong and J. B. Schroder, *Applications of time parallelization*, Comput. Visual Sci., 23 (2020), pp. 1–15. 2

[7] J. B. Schroder, M. Lecouvez, R. D. Falgout, C. S. Woodward, and P. Top, *Parallel-in-time solution of power systems with scheduled events*, in Power and Energy Society General Meeting (PESGM), 2018, IEEE, 2018. 1

[8] H. D. Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan, *Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection*, Numer. Linear Algebra Appl., 28 (2021), p. e2367. 1

[9] H. D. Sterck, R. D. Falgout, A. J. 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., 41 (2019), pp. A538––A565. 1

[10] H. D. Sterck, R. D. Falgout, and O. A. Krzysik, *Fast multigrid reduction-in-time for advection via modified semi-lagrangian coarse-grid operators*, SIAM J. Sci. Comput., 45 (2023), pp. A1890–A1916. 1

[11] H. D. Sterck, R. D. Falgout, O. A. Krzysik, and J. B. Schroder, *Efficient multigrid reduction-in-time for method-of-lines discretizations of linear advection*, Journal of Scientific Computing, 96 (2023), pp. 1–31. 1

[12] D. A. Vargas, R. D. Falgout, S. Günther, and J. B. Schroder, *Multigrid reduction in time for chaotic dynamical systems*, SIAM J. Sci. Comput., 45 (2023), pp. A2019–A2042. 1

[13] P. H. Worley, *Limits on parallelism in the numerical solution of linear partial differential equations*, SIAM J. Stat. Comput., 12 (1991), pp. 1–35. 3