LLNL-SR-805739

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Parallel-in-Time Multigrid Methods for Hyperbolic Problems, with a Focus on the Shallow Water Equations

J. Schroder

February 26, 2020

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Final Report

“Parallel-in-Time Multigrid Methods for Hyperbolic Problems, with a Focus on the Shallow Water Equations”

Submitted by:
Jacob Schroder

Prepared for:
Lawrence Livermore National Laboratory
Attn: Sherrill Kazadzis
Supply Chain Management, L-650
Lawrence Livermore National Laboratory

Under contract number:
B633040

FINAL REPORT

Contract B633040

"Parallel-in-Time Multigrid Methods for Hyperbolic Problems, with a Focus on the Shallow Water Equations"

Progress has been made in several of the proposed areas of study, and is summarized below.

Project Abstract

The coming massive parallelism of exascale computing presents a pressing challenge for the many DOE simulations of time-dependent partial differential equations, which typically use traditional sequential time stepping methods. Since this traditional approach is inherently serial, it presents a sequential bottleneck when moving to exascale computing, because future performance gains will come through greater concurrency, not faster clock speeds. Thus, the goal of this work is to research parallelism in time, i.e., methods that compute multiple time values simultaneously, not sequentially. The focus will be on hyperbolic problems of programmatic interest to DOE, with the goal of enabling scalable simulations of time-dependent hyperbolic problems on future architectures.

The difficulty lies in the fact that hyperbolic problems are well-known to be difficult for parallel-in-time methods, with the most common parallel-in-time method, Parareal, diverging in many cases. Here, the chosen methodology for scalably solving these hyperbolic space-time equations is multigrid-reduction-in-time (MGRIT), because multigrid (when it works) is a powerful, optimal, and scalable solver for discretized PDEs. To further research in this area, this project will explore a model hyperbolic problem (the shallow water equations) in the context of recent advances (e.g., by Wingate and Haut [1]) in constructing improved coarse time-grid time-propagators by using the slow asymptotic structure of the equations. In particular, we will investigate if such coarse time-propagators offer significant advantages in an MGRIT setting, and explore any broader insights gained into parallel-in-time for hyperbolic problems.

Project Steps

- Schroder and Chaudhry interviewed three students, and selected MSc student Nicholas Abel for the RA funded by the subcontract
- The summer visit dates to LLNL were selected in consultation with Dr. Falgout
  Schroder: July 8th - August 2nd (4 weeks)
  Abel: July 8th – July 12th (1 week)
- Equipment was purchased and installed: a group workstation and a student desktop computer.
- Nicholas Abel began his summer RA on June 1st, and has been studying the method of Haut and Wingate (HW), the rotating shallow water equations, and implementing the parallel-in-time integration scheme of HW.
- Nicholas Abel began his Fall Semester RA in August, 2019. He has successfully integrated our production parallel-in-time code XBraid with a target shallow water code in Python developed by Wingate's research group (called Cyclops), and is now running research experiments.

---

[1] *An Asymptotic Parallel-in-Time Method for Highly Oscillatory PDEs,* SIAM J. Sci. Comput.

- The experiments coupling Cyclops and XBraid have consisted of considering F- versus FCF-relaxation and various multilevel settings, using combinations of the standard fine-grid time-stepping operator and the asymptotic time-averaged time-stepping operator. There is a definite benefit to using FCF-relaxation, which is a new result, but it is yet unknown if multilevel can provide a benefit. Only preliminary data supports this conclusion. See below for more details.
- Nicholas Abel presented his results at the Annual AMG Summit (workshop) in Santa Fe, NM to Dr. Falgout and other experts in the field.

Research Synopsis

As outlined in the statement-of-work (SOW), the main effort of this project will come over the summer and the Fall, when Nicholas Abel will be employed as an RA, and Schroder and Chaudhry will have summer research funding. There is no funded RA during the Spring semester. So far, we have accomplished the following.

- We have implemented a 1D shallow water model in the chosen test code (XBraid)
    - With no spatial coarsening, which limits the fine-grid time-step size, and the use of a naïve coarse time-grid time-stepper, XBraid converges in the range of 0.33-0.66. For practice, the student ran some strong and weak scaling tests.
    - With spatial coarsening, XBraid does not converge well, and generally diverges. This is not surprising, as the problem here is purely hyperbolic.
- Thus after the above warm-up step, we extended our spatial discretization to include extra terms for the 1D rotating shallow water equations (RSW), as presented by HW.
    - This included implementing a nonlinear rotational term, and a hyperviscosity term
    - The hyperviscosity term, in particular, should help XBraid.
    - The 1D RSW were implemented in Python and XBraid, using both RK4 and a library ODE solver in Python. This was done to verify the XBraid implementation.
- After consultation with Dr. Falgout during our summer visit, it was decided to develop an XBraid-Python coupling. This allows us to couple XBraid with a library Python RSW code Cyclops, written by Wingate's research group at Exeter. While we were confident in our above developed finite-difference code for the RSW, use of this code will allow us to use the specific time and space discretizations of the RSW favored by application area experts such as Wingate's group.
    - Development of this XBraid-Python interface took longer than expected, in particular the conversion of Python objects into usable C-style data was difficult, as was determining the structure of the Python callback functions required by XBraid.
    - However, the interface is now complete and has been verified, both for a simple example (similar to the existing `ex-01.c` in XBraid) and for the Cyclops code.
        - This verification followed the traditional verification steps for XBraid and a user code – comparing single and multilevel, comparing single and multiprocessor, and verifying the fixed-point iteration test.
        - Agreeing with Dr. Falgout, we carried out one last validation experiment, where we tracked the L-infinity norm of stand-alone Cyclops and Cyclops+XBraid. Some implementation of the L-infinity norm in Cyclop+XBraid was required. The convergence histories here were (nearly) identical, thus further verifying the code.
    - We have begun multilevel experiments using XBraid and Cyclops.

- Carried out experiments comparing F-relaxation versus FCF-relaxation. For larger (more practical) final time values, FCF relaxation shows some benefit, converging in 50% (or slightly fewer) iterations. In particular, the wall clock times for FCF are 50% faster
    - The faster wall clock times are due to the high cost of the coarse time-grid time-step. This "Step" function on the coarse time-grid costs 10 to 100 time as much as "Step" on the fine-grid, thus reducing MGRIT iterations with FCF greatly reduces this cost. This extra cost of "Step" could be parallelized further (as it relates to the computation of integration points), but at the cost of reduced parallel efficiency. Our conclusion is that MGRIT with FCF will still be faster.
    We consider this new result to be a major outcome of the project.
- Regarding the research goal of using multilevel solvers, the results have been mixed.
    - For the case of scale separation (when the problem parameter controlling the oscillation strength, epsilon, is 0.01), we have not found a case where a 3 or more level solver out-performs a 2-level solver
    - **However for the case of no scale separation (epsilon = 1.0), there are some preliminary numbers showing possible benefit to using a 3-level solver.** For this result, a 3-level MGRIT solver does converge more quickly in at least some cases (e.g., 7 iterations versus 10 iterations). Future work will entail exploring this further, and especially the aspects of tailoring the time-averaging window to each level in the MGRIT solver.
- We committed a version of the Python+XBraid interface in a branch in the publicly available XBraid Github repository. Given how popular Python is, this will benefit the broader community. Future work will be to push this to the main branch of XBraid, along with user's manual documentation.
- We discovered that if the dissipation in the problem gets much larger than that used in classic rotating shallow water examples (dissipation over 10^{-2}), XBraid converges very fast with no modifications.