Journal of Computational Physics 371 (2018) 483–505

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Communication-aware adaptive Parareal with application to a nonlinear hyperbolic system of partial differential equations

Allan S. Nielsen *, Gilles Brunner, Jan S. Hesthaven

Chair of Computational Mathematics and Simulation Science, Section de Mathématiques, École Polytechnique Fédérale de Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 10 May 2017
Received in revised form 17 April 2018
Accepted 29 April 2018
Available online 24 May 2018

Keywords:
Parallel-in-time
Parareal
Hyperbolic
PDE
Shallow water equations
Tsunami simulation

ABSTRACT

In the strong scaling limit, the performance of conventional spatial domain decomposition techniques for the parallel solution of PDEs saturates. When sub-domains become small, halo-communication and other overhead come to dominate. A potential path beyond this scaling limit is to introduce domain-decomposition in time, with one such popular approach being the Parareal algorithm which has received a lot of attention due to its generality and potential scalability. Low efficiency, particularly on convection dominated problems, has however limited the adoption of the method. In this paper we demonstrate trough large-scale numerical experiments that it is possible not only to obtain time-parallel speedup on the non-linear shallow water wave equation, but also that we may obtain parallel acceleration beyond what is possible using conventional spatial domain-decomposition techniques alone. Two factors were essential in achieving this. First, for Parareal to converge on the hyperbolic problem we used an approximate Riemann solver as the preconditioner. This preconditioner introduces only dissipative errors with respect to the 3rd order accurate WENO-RK discretization used to solve the PDE system. The preconditoner is comparatively expensive and convergence is slow unless time-subdomains are short. We therefore introduce a new scheduler that we denote Communication Aware Adaptive Parareal (CAAP). CAAP increases obtainable speed-up by minimizing the time-subdomain length without making communication of time-subdomains too costly whilst also adaptively overlapping consecutive cycles of Parareal so to mitigate the impact of a relatively expensive coarse operator.

1. Introduction

The ongoing and rapid evolution of computers used to model physical phenomenons in the computational sciences poses new challenges for algorithms. The growing number of cores, the increasingly convoluted cache hierarchies, and the use of accelerators all seek to boost the computational capacity of individual nodes. At the same time, the number of compute nodes in distributed memory machines has increased dramatically. The machine that currently crowns the top500 list of supercomputer Sunway TaihuLight at NSCC Wuxi, China, has more than 40.000 compute nodes, comprising a total of more than 10 millions cores [10]. This ongoing development towards increasing hardware parallelism exposes algorithmic shortcomings and requires a rethinking of the fundamental algorithms to maintain scalability and enable efficient use of the computing

* Corresponding author.
E-mail addresses: allan.nielsen@epfl.ch (A.S. Nielsen), gilles.brunner@alumni.epfl.ch (G. Brunner), jan.hesthaven@epfl.ch (J.S. Hesthaven).

https://doi.org/10.1016/j.jcp.2018.04.056
0021-9991/

platform [9]. In this paper we show how the Parareal method may be modified in such a way as to allow for parallel-in-time acceleration of a tsunami simulation beyond what is possible using conventional spatial domain-decomposition techniques alone. The underlying PDE governing the dynamics of the model is the shallow water wave equation. The equation is a purely hyperbolic system of coupled non-linear PDEs, the solutions of which typically contain both shocks and smooth regions interacting in a non-trivial manner. We thus conjecture that the positive result, presented in section 6, is an indication that it is possible for other similar systems to benefit from parallel-in-time acceleration.

1.1. Time-domain parallelism and Parareal

Solving time dependent PDEs is often done in a methods-of-line approach where the spatial components are discretized in some appropriate manner and a numerical integration technique is applied to advance in time. The approach extends to distributed memory machines by applying some form of domain decomposition, letting independent nodes communicate boundary information of their local sub-domains. The limitation to the approach lies in the strong scaling limit, i.e. increasing the number of nodes for a fixed problem size to achieve a reduction in time to compute.

One might naively suspect that one may "run out of parallelism" – i.e. as the combined number of cores become sufficiently high, there are simply not enough parallel degree's of freedom for all cores to work all the time. But consider this, solving a problem with upwards of a billion degree's of freedom in space may today be done on a potent workstation. Conversely, even the largest clusters available for researchers in the world has no more than a few million cores, everything included. This scaling limit is therefore somewhat theoretical, and not yet of much relevance for practitioners. So why does obtainable speed-up saturate in the strong-scaling limit? Consider what happens as spatial sub-domains decrease in size, given a three dimensional domain in space divided into a number of quadratic sub-domains with n elements spanning each dimension. The compute work in each sub-domain is proportional to n^3. The boundary information that needs to be exchanged with neighboring sub-domains is proportional to $\sim n^2$. As $n \to 1$, compute nodes will increasingly be spending time communicating boundary information rather than computing.

This particular limit is very much of practical concern. Moving a double between two individual compute nodes in a cluster is many orders of magnitude more expensive than a compute operation in terms of both wall-time and energy consumption. On large machines comprising thousands of nodes, this is a substantial bottleneck for scaling application efficiently and new algorithmic developments are required. A potential new path in obtaining scaling beyond what is possible with conventional methods, is to extract parallelism in the time integration procedure. Once a system of partial differential equation has been reduced to a large system of ordinary differential equations to be integrated over time, the problem is usually viewed as a sequential process. However, many attempts to extract parallelism do exists. For a complete overview of research in the direction we refer to a recent paper [12].

The focus of this paper will be on the Parareal method that has received a lot of attention over the past decade. The Parareal method, first proposed in [17], borrows from ideas in spatial domain decomposition to construct an iterative approach for solving the temporal problem in a parallel global-in-time approach. To present the method, consider a problem on the form

$$\begin{cases} \frac{d\mathbf{u}}{dt} + \mathcal{A}(t, \mathbf{u}) = 0 \\ \mathbf{u}(T_0) = \mathbf{u}_0 \quad t \in [T_0, T] \end{cases} \tag{1}$$

where $\mathcal{A} : \mathbb{R} \times V \to V'$ is a general operator depending on $\mathbf{u} : \Omega \times \mathbb{R}^+ \to V$ with V being a Hilbert space and V' its dual. Now, assume there exists a unique solution $\mathbf{u}(t)$ to (1) and decompose the time domain into n_t individual time slices

$$T_0 < T_1 < \cdots < T_{n_t - 1} < T_{n_t} = T. \tag{2}$$

Let $T_n = n\Delta T$. We now define a numerically accurate solution operator $\mathcal{F}_{\Delta T}$ that, for any $t > T_0$, advances the solution as

$$\mathcal{F}_{\Delta T}(T_n, \mathbf{u}(T_n)) = \mathbf{U}_{T_n + \Delta T} \approx \mathbf{u}(T_n + \Delta T) \tag{3}$$

To solve (1) on $[T_0, T_0 + n_t \Delta T]$, define the matrix of operators $M_{\mathcal{F}}$

$$M_{\mathcal{F}} = \begin{bmatrix} 1 & & & \\ -\mathcal{F}_{\Delta T}^{T_0} & \ddots & & \\ & \ddots & \ddots & \\ & & -\mathcal{F}_{\Delta T}^{T_{n_t-1}} & 1 \end{bmatrix} \tag{4}$$

with $\bar{\mathbf{U}} = \left[\mathbf{U}_0, \ldots, \mathbf{U}_{n_t}\right]$ and $\bar{\mathbf{U}}_0 = [u(T_0), 0, \ldots, 0]$. The sequential solution procedure is then equivalent to solving

$$M_{\mathcal{F}} \bar{\mathbf{U}} = \bar{\mathbf{U}}_0 \tag{5}$$

by forward substitution for $\bar{\mathbf{U}}$ so to recover $\mathbf{U}_0 \cdots \mathbf{U}_{n_t}$ as approximations to $\mathbf{u}(T_0) \cdots \mathbf{u}(T_{n_t})$. If we instead seek to solve the system using a point-iterative approach, i.e., we seek the solution $\bar{\mathbf{U}}^{k+1} = \bar{\mathbf{U}}^k + \left(\bar{\mathbf{U}}_0 - M_{\mathcal{F}} \bar{\mathbf{U}}^k\right)$, we observe that at the