ARTICLE IN PRESS

Journal of Computational and Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Nonlinear parallel-in-time Schur complement solvers for ordinary differential equations

Santiago Badia *, Marc Olm

Universitat Politècnica de Catalunya, Jordi Girona1-3, Edifici C1, E-08034 Barcelona, Spain
Centre Internacional de Mètodes Numèrics en Enginyeria, Parc Mediterrani de la Tecnologia, Esteve Terrades 5, E-08860 Castelldefels, Spain

ARTICLE INFO

Article history:
Received 24 March 2017
Received in revised form 14 September 2017

Keywords:
Time parallelism
Ordinary differential equations
Domain decomposition
Nonlinear solver
Scalability

ABSTRACT

In this work, we propose a parallel-in-time solver for linear and nonlinear ordinary differential equations. The approach is based on an efficient multilevel solver of the Schur complement related to a multilevel time partition. For linear problems, the scheme leads to a fast direct method. Next, two different strategies for solving nonlinear ODEs are proposed. First, we consider a Newton method over the global nonlinear ODE, using the multilevel Schur complement solver at every nonlinear iteration. Second, we state the global nonlinear problem in terms of the nonlinear Schur complement (at an arbitrary level), and perform nonlinear iterations over it. Numerical experiments show that the proposed schemes are weakly scalable, i.e., we can efficiently exploit increasing computational resources to solve for more time steps the same problem.

1. Introduction

At the beginning of the next decade supercomputers are expected to reach a peak performance of one exaflop/s, which implies a 100 times improvement with respect to current supercomputers. This improvement will not be based on faster processors, but on a much larger number of processors (in a broad sense). This situation will certainly have an impact in large scale computational science and engineering (CSE). Parallel algorithms will be required to exhibit much higher levels of concurrency, keeping good scalability properties.

When dealing with transient problems, since information always moves forward in time, one can exploit sequentiality. However, the tremendous amounts of parallelism to be exploited in the near future certainly motivates to change this paradigm. One of the motivations to exploit higher levels of parallelism will be to reduce the time-to-solution. In the simulation of ordinary differential equations (ODEs), the way to go is to exploit concurrency in time. The idea is to develop parallel-in-time solvers that provide the solution at all time values in one shot, instead of the traditional sequential approach that exploits the arrow of time. If scalable parallel-in-time solvers are available, the use of higher levels of parallelism will certainly reduce the time-to-solution.

Parallel-in-time solvers are receiving rapidly increasing attention. Different iterative methods have been considered so far, e.g., the *parareal* method [1] or spectral deferred-correction time integrators [2]. With regard to direct methods, time-parallel methods can be found in [3]. In general these methods can exploit low levels of concurrency [4] or are tailored for particular types of equations [5]. We refer to [3] for an excellent up-to-date review of time parallelism. It has also motivated the development of space–time parallel solvers for transient partial differential equations (PDEs) [6,7].

* Corresponding author at: Centre Internacional de Mètodes Numèrics en Enginyeria, Parc Mediterrani de la Tecnologia, Esteve Terrades 5, E-08860 Castelldefels, Spain.

E-mail addresses: sbadia@cimne.upc.edu (S. Badia), molm@cimne.upc.edu (M. Olm).

http://dx.doi.org/10.1016/j.cam.2017.09.033
0377-0427/© 2017 Elsevier B.V. All rights reserved.

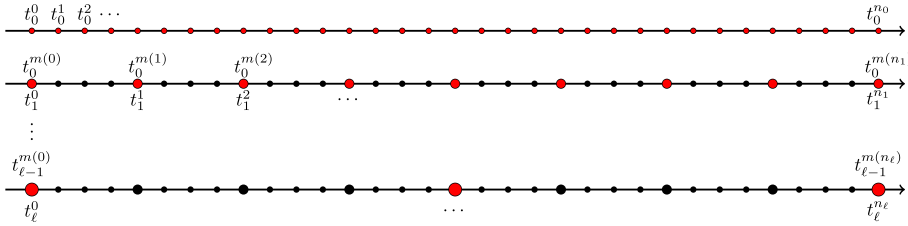

Fig. 1. Multilevel time partition of $[0, T]$. The subindex in t_α^β denotes the partition level, whereas the superindex denotes the time step in such partition. In order to relate time values of two constitutive levels, we use the notation $t_\alpha^\beta = t_{\alpha-1}^{m(\beta)}$, i.e., the time value t_α^β corresponds to the time step $m(\beta)$ at the previous level.

In this work, we propose a parallel-in-time solver for ODEs that relies on the well-known Schur complement method in linear algebra. For linear (systems of) ODEs, the approach can be understood as a Schur complement solver in time. When the coarse problem is too large compared to the local problems, due to the structure of the coarse problem, we can consider recursively the Schur complement strategy, leading to multilevel implementations, in order to push forward scalability limits. The method can be applied to θ-methods, discontinuous Galerkin (DG) methods, Runge–Kutta methods, and BDF methods. We also note that the proposed method can also be understood as a parareal scheme in which the coarse solver is automatically computed in such a way that the scheme is a direct method (convergence in one iteration is assured). (The interpretation of the parareal method as an approximation of the Schur complement problem has already been pointed out in [6].) As a result, the proposed method solves the drawback of the parareal scheme, i.e., its poor parallel efficiency, inversely proportional to the number of iterations being required by the iterative algorithm. One of the messages of this work is to show that the approximation of the Schur complement in parareal methods does not really pay the price when (just by roughly multiplying by two the number of operations) one can have a highly scalable direct parallel-in-time solver.

In order to extend these ideas to nonlinear PDEs, we consider two different strategies. First, we consider a global linearization of the problem in time using, e.g., Newton's method. We note that the idea of a global linearization of nonlinear ODEs to exploit time-parallelism is not new. It was already considered by Bellen and Zennaro in 1989 [8]. (In any case, the solvers proposed in [8] are different from the ones presented herein. They are restricted to the Steffensen's linearization method, which leads to a diagonal problem per nonlinear iterations, where parallelization can obviously be used.) After the linearization of the problem, we consider the Schur complement solver commented above at every nonlinear iteration. A second strategy consists in applying the nonlinear Schur complement strategy first, and next to consider the linearization of such operator, leading to nested nonlinear iterations.

The parallel-in-time ideas in this work can naturally be blended with domain decomposition ideas to design highly scalable space–time parallel solvers. We have combined these ideas with a multilevel balancing DD by constraints (BDDC) preconditioner (see [9,10]) in [7], by judiciously choosing the quantities to be continuous among processors in a space–time partition, i.e. [11]. The resulting space–time parallel solver has been proved to be scalable on thousands of processors for different transient (non)linear PDEs.

The outline of the article is as follows. In Section 2, we state the problem. We introduce a time-parallel direct method for linear ODEs based on the computation of a multilevel time Schur complement in Section 3. In Section 4, we extend the method to nonlinear ODEs, by combining first a Newton linearization step with a Schur complement linear solver and next considering nonlinear Schur complement problems. We present a detailed set of numerical experiments in Section 5, showing the excellent scalability properties of the proposed methods. Finally, we draw some conclusions in Section 6.

2. Statement of the problem

In this section, we develop a parallel direct solver for the numerical approximation of ordinary differential equations (ODEs). We consider a system of ODEs of size m_{unk}:

$$\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} + \boldsymbol{\kappa}(t, \boldsymbol{u}(t)) = \boldsymbol{0}, \qquad \boldsymbol{u}(t^0) = \boldsymbol{u}_0, \tag{1}$$

for $t \in (t^0 = 0, T]$. Let us assume that $\boldsymbol{\kappa}(\cdot, \cdot)$ is continuous with respect to the first argument and Lipschitz continuous with respect to the second argument, and that existence and uniqueness holds.

For the time interval $[0, T]$, we define a hierarchical multilevel partition as follows (see Fig. 1 for a detailed illustration). We define a (level-0) time partition $\{0 = t_0^0, t_0^1, \dots, t_0^{n_0} = T\}$ into n_0 time elements. Next, we consider a (level-1) coarser time partition $\{0 = t_1^0, \dots, t_1^{n_1} = T\}$ into n_1 time subdomains (or level-1 elements), defined by aggregation of elements at the previous level, i.e., for every $i \in \{0, \dots, n_1\}$ there exists an $m(i) \in \{0, \dots, n_0\}$ such that $t_1^i = t_0^{m(i)}$. We proceed recursively, creating coarser partitions for higher levels. We define the time element i at level-k as the time interval (t_k^i, t_k^{i+1}).

We will present the method in a general way that is independent of the time integration scheme being used. We define the nonlinear operators $\boldsymbol{\mathcal{A}}_0^{i+1} : \boldsymbol{u}^i \mapsto \boldsymbol{u}^{i+1}$ for $i \in \{0, \dots, n_0 - 1\}$, such that, given the initial value $\boldsymbol{u}^i$, solves (1) in (t_0^i, t_0^{i+1}), and

Please cite this article in press as: S. Badia, M. Olm, Nonlinear parallel-in-time Schur complement solvers for ordinary differential equations, Journal of Computational and Applied Mathematics (2017), http://dx.doi.org/10.1016/j.cam.2017.09.033.