Title: Parallel-in-Time collocation methods: ideas, methods, software

Abstract:

The efficient use of modern supercomputers has become one of the key challenges in computational science. For the numerical solution of time-dependent processes, time-parallel methods have opened new ways to overcome both strong and weak scaling limits. If higher order accuracy in time is needed or at least feasible, parallelization techniques based on collocation methods can provide temporal parallelism within a single time-step as well as across multiple time-steps. In this talk, we give an overview of three different approaches to introduce parallel-in-time integration for collocation methods: (1) parallel preconditioners for spectral deferred corrections, (2) the parallel full approximation scheme in space and time, and (3) diagonalization-based preconditioning for multi-step collocation problems. Those approaches can even be combined to obtain multi-time-parallel integrators.

Yet, an efficient parallel implementation is a demanding exercise and may exceed the scope of more math-based time-parallel projects. Instead, parallel-in-time research often resorts to "counting iterations" of the algorithm as a means of measuring its performance independently of its actual implementation. However, this hides many non-negligible sources of computational cost, such as communication. If the true parallel efficiency of candidate algorithms is under consideration, this needs to be obtained in a more coherent way.

In addition to the ideas of the aforementioned parallel-in-time collocation methods we also present here one prototyping library which aims to cover both aspects of time-parallel research: method development and fair efficiency testing. pySDC is a Python code that, while not providing production-level performance, allows users to detect pitfalls in parallel algorithms before committing to optimized implementations or rigorous mathematical analysis. Parallel efficiency can be estimated by comparing to various time-serial algorithms implemented in the same framework. The modular structure allows users to easily apply a new algorithm to a wide range of problems and configurations without awareness of all the intricacies of the code. pySDC is publicly hosted on GitHub and well-tested and -documented in order to provide new users with a smooth and rapid start.

This is joint work with Thomas Baumann (JSC), Ruth Schoebel (JSC), Gayatri Caklovic (former KIT), Daniel Ruprecht (TUHH), Michael Minion (former LBL)