A GENERAL SOFTWARE FRAMEWORK FOR PARALLEL ALGEBRAIC MULTIGRID METHODS

Pavel Jiránek
CERFACS, 42 Avenue Gaspard Coriolis, France
e-mail: `jiranek@cerfacs.fr`

Joint work with S. Gratton, X. Vasseur, P. Hénon

Abstract

Parallel algebraic multigrid (AMG) preconditioners constitute an essential component of fast and scalable codes for large scale computations and a number of software implementations exists nowadays in the form of freely distribuable (or proprietary) packages. They allow the users to benefit from a wide selection of the AMG methods and their various modifications and improvements discovered during the last three decades.

Most of the existing general-purpose AMG codes, however, suffer several disadvantages. In particular, the "external" developers may experience when trying to extend an existing package with new AMG approaches and algorithmic modifications or trying to reuse the existing code to be applicable on more "exotic" problems than the ones the package was originally intended for. This is mainly due to a non-modular and sometimes rather ad-hoc design. In addition, most of the packages attempt to be self-contained, they do not use much of the available and more or less standard libraries.

We propose a new AMG framework which attempts to overcome the aforementioned flaws. We heavily use an object oriented approach to design the basic environment, which allows us to create a flexible and easily customizable hierarchy of classes implementing a finely granularized setup and solve process of AMG. The multigrid levels are split in two layers: the operator layer for the solution part, which allows to apply the multigrid cycles on a hierarchy composed from arbitrarily composed operators, and the variable layer used in the AMG setup to access the particular composition of the actual operators. This approach is useful, e.g., for structured problems like indefinite saddle point ones, where a particular block structure can be employed when creating the multigrid hierarchy. The AMG setup process is realized by a set of parametrized "black-box" components with inputs and outputs connected to certain level variables, which are used, e.g., to create coupling graphs, coarsen ing, to compute prolongation operators etc., and are ran in an automatically determined order given by the data dependencies on the particular pair of subsequent multigrid levels. In addition, a different set of components can be used for different levels allowing to use various coarsening approaches per level within one hierarchy. The framework also allows to fully customize the solve process, that is, to use the user-defined smoothers, solvers, and even cycles. For solving sequences of algebraic systems, we offer ways to reuse any part of the already computed hierarchy by specifying which

level variables need to be rebuild in order to update the preconditioner. The whole framework, based on Trilinos, is designed to minimize the coding requirements for both basic and advanced users by using a factory patterns based on automatically constructed dictionaries and setting up all needed strategies and parameters within one XML file.