Commun. Comput. Phys.
doi: 10.4208/cicp.301109.080410s

Vol. **9**, No. 3, pp. 607-626
March 2011

Extrapolation-Based Acceleration of Iterative Solvers: Application to Simulation of 3D Flows

Leopold Grinberg* and George Em Karniadakis

Division of Applied Mathematics, Brown University, Providence 02912, USA.

Received 30 November 2009; Accepted (in revised version) 8 April 2010

Available online 17 September 2010

To the memory of David Gottlieb

Abstract. We investigate the effectiveness of two extrapolation-based methods aiming to approximate the initial state required by an iterative solver in simulations of unsteady flow problems. The methods lead to about a ten-fold reduction in the iteration count while requiring only negligible computational overhead. They are particularly suitable for parallel computing since they are based almost exclusively on data stored locally on each processor. Performance has been evaluated in simulations of turbulent flow in a stenosed carotid artery and also in laminar flow in a very large domain containing the human intracranial arterial tree.

AMS subject classifications: 52B10, 65D18, 68U05, 68U07

Key words: High-order methods, POD, CFD, low-energy preconditioner.

1 Introduction

Discretization of first-order (in time) PDEs

$$\frac{\partial \mathbf{u}}{\partial t} = f_a(t,\mathbf{u},\mathbf{x}) \Rightarrow \frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} = f_d(t^{n+1},\mathbf{u}^{n+1},\mathbf{u}^n,\mathbf{x})$$

often requires the iterative solution of a linear system of equations of the form $\mathbf{A}\mathbf{u}^{n+1}=\mathbf{b}$; here f_a is an analytic function and f_d is its discrete counterpart.

In general, the performance of the iterative solver is determined by: (1) the condition number of the (preconditioned) operator; (2) the number of floating points operations required at each iteration (e.g., sparsity of the operator); (3) degree of parallelization, i.e., minimization of the sequential part of an algorithm with respect to the work performed

*Corresponding author. *Email addresses:* lgrinb@dam.brown.edu (L. Grinberg), gk@dam.brown.edu (G. E. Karniadakis)

in parallel; and (4) the proximity of the initial state (guess) to the exact solution. The first two factors are related to the properties of the operator and also to the discretization method employed. The third factor is greatly affected by the choice of the numerical scheme for iterative solution and its implementation. While in the last few decades significant effort has been put in developing effective preconditioning techniques that minimize the condition number of the matrix $\mathbf{A}$ and maximize its sparsity, very little work has been published on attempts to obtain a good initial state to initialize the iterative solver. It is common practice to use the solution from a previous time step $\mathbf{u}^n$ as an initial state, however, an *approximate* solution for $\mathbf{u}^{n+1}$ can be obtained using various techniques. Depending on the method and the smoothness of $\mathbf{u}(t)$, the distance between approximate solution (hereafter denoted by $||\mathbf{u}_{ap}-\mathbf{u}^{n+1}||$) can be significantly smaller than $||\mathbf{u}^n-\mathbf{u}^{n+1}||$, hence leading to a substantial reduction in the number of iterations. This, in turn, lowers the CPU-time and enhances the overall efficiency of a solver, hence it is crucial that obtaining $\mathbf{u}_{ap}$ will require *minimal computational effort*.

Markovinović and Jansen [1] employed Proper Orthogonal Decomposition (POD) to accelerate convergence of iterative solvers, and tested it in simulations of two-phase flow through heterogeneous porous media. Specifically, a two-step projection method was proposed: in the first substep, solutions computed at previous time steps were projected onto a subspace spanned by a low number of global POD modes. Subsequently, an approximate solution $\mathbf{u}_{ap}$ was obtained by *solving* the governing equations in the reduced space and projecting the result back to the original, high-dimensional space. This procedure was completed by solving the original system of equations using the $\mathbf{u}_{ap}$ (instead of $\mathbf{u}^n$) as an improved initial state, leading to 67% reduction in the computing time. The key idea of this method is based on the observation that computing the approximate solution (initial state for the iterative solver) using the reduced basis model is less computationally demanding than performing a number of iterations required to advance the solution to the same state. In another study, Tromeur-Dervout and Vassilevski [2] also implemented POD to predict the solution field $\mathbf{u}^{n+1}$ using a fully-implicit scheme. A low-dimensional solution based on POD was used to provide a better initial state to an inexact Newton back-tracking method, and a two-fold reduction in the computing time was reported.

A different use of POD was attempted in [3], where a Galerkin-free methodology was implemented for simulating unsteady flow past a circular cylinder. Extrapolation of POD modes was pursued over *large* time steps and the predictions were used as initial conditions to a Navier-Stokes solver running over much shorter time steps. A related study but targeting the right-hand-side (RHS) of the linear system was pursued in [4] where it was proposed to reduce the iteration count in solution of a linear system of equations $\mathbf{A}\mathbf{u}^{n+1}=\mathbf{b}^{n+1}$ by using information contained in successive RHS vectors $\mathbf{b}^{n+1-k}$, $k=1,\cdots,L$ and corresponding solutions $\mathbf{u}^{n+1-k}$, $k=1,\cdots,L$. This method is based on solving a modified system $\mathbf{A}\tilde{\mathbf{u}}=\tilde{\mathbf{b}}$, where $\tilde{\mathbf{b}}$ is computed by the Gram-Schmidt orthogonalization of the successive RHS vectors. The solution $\mathbf{u}^{n+1}$ is obtained by adding a superposition of the solutions $\mathbf{u}^{n+1-k}$, $k=1,\cdots,L$ to $\tilde{\mathbf{u}}$. In [4] several variations of this method have been proposed; while the computational complexity of the methods varied,