Journal of Computational Physics 238 (2013) 32–47

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

Optimization of the deflated Conjugate Gradient algorithm for the solving of elliptic equations on massively parallel machines

Mathias Malandain *, Nicolas Maheu, Vincent Moureau

CORIA (CNRS UMR6614), Site Universitaire du Madrillet, BP 12, 76801 Saint-Étienne-du-Rouvray Cedex, France
INSA of Rouen, Avenue de l'Université, 76801 Saint-Étienne-du-Rouvray, France
University of Rouen, France

ARTICLE INFO

Article history:
Received 27 April 2012
Received in revised form 12 October 2012
Accepted 26 November 2012
Available online 13 December 2012

Keywords:
Deflation
Preconditioned Conjugate Gradient
Massively parallel simulation
Partial Differential Equation
Large Eddy simulation

ABSTRACT

The discretization of Partial Differential Equations often leads to the need of solving large symmetric linear systems. In the case of the Navier–Stokes equations for incompressible flows, solving the elliptic pressure Poisson equation can represent the most important part of the computational time required for the massively parallel simulation of the flow. The need for efficiency that this issue induces is completed with a need for stability, in particular when dealing with unstructured meshes. Here, a stable and efficient variant of the Deflated Preconditioned Conjugate Gradient (DPCG) solver is first presented. This two-level method uses an arbitrary coarse grid to reduce the computational cost of the solving. However, in the massively parallel implementation of this technique for very large linear systems, the coarse grids generated can count up to millions of cells, which makes direct solvings on the coarse level impossible. The solving on the coarse grid, performed with a Preconditioned Conjugate Gradient (PCG) solver for this reason, may involve a large number of communications, which reduces dramatically the performances on massively parallel machines. To this effect, two methods developed in order to reduce the number of iterations on the coarse level are introduced, that is the creation of improved initial guesses and the adaptation of the convergence criterion. The design of these methods make them easy to implement in any already existing DPCG solver. The structural requirements for an efficient massively parallel unstructured solver and the implementation of this solver are described. The novel DPCG method is assessed for applications involving turbulence, heat transfers and two-phase flows, with grids up to 17.8 billion elements. Numerical results show a two- to 12-fold reduction of the number of iterations on the coarse level, which implies a reduction of the computational time of the Poisson solver up to 71% and a global reduction of the proportion of communication times up to 53%. As a result, the weak scaling of the LES solver is shown to be clearly improved for massively parallel uses.

1. Introduction

1.1. State-of-the-art and motivation

Hestenes and Stiefel published in 1952 an article considered a seminal reference on the CG method [1], that has become a standard method for solving symmetric linear systems, especially those arising from physical problems. The idea of preconditioning this method dates back to the late Fifties, that is very few years after its creation, with for instance the work of

* Corresponding author at: CORIA (CNRS UMR6614), Site Universitaire du Madrillet, BP 12, 76801 Saint-Étienne-du-Rouvray Cedex, France. Tel.: +33 2 32 95 97 87.

E-mail addresses: mathias.malandain@coria.fr (M. Malandain), nicolas.maheu@coria.fr (N. Maheu), vincent.moureau@coria.fr (V. Moureau).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.11.046

Engell et al. [2], but the deflation technique came up three decades later, in 1987, thanks to Nicolaides [3]. This method has since been developed and frequently applied, among others, to linear systems arising from the discretization of Navier–Stokes equations: see for instance [4] about the creation of a Deflated CG method; Frank and Vuik [5] and Vermolen et al. [6] about the domain-decomposition deflation method; Aubry et al. [7] and Nabben and Vuik [8] for applications to physical problems. Several comparisons have been made between solvers [9,10], from which it appears that the DPCG is among the most stable and efficient numerical methods for symmetric systems; moreover, its easy implementation for massively parallel solving on unstructured meshes is a considerable asset. Nevertheless, several stability issues may occur when the method is used for the simulation of fluid flows in complex geometries. These issues may be of particular importance when dealing with Poisson equations in two-phase flows, that feature variable coefficient matrices with strong variations in its diagonal.

In an article written in 2009 by Tang et al. [11], the A-DEF2 variant of the Deflated CG is created that is proved fast and robust, even when compared to Domain Decomposition and multigrid methods, which is why this method is studied here. Some improvements were still to be used: as a matter of fact, reducing the high number of iterations required on the coarse level can become very important, as these iterations will add irreducible communication times to the overall time spent in the solver.

Two novel methods are introduced here, one of them consisting in the computation of improved initial guesses, the other one being an adaptation of the convergence criterion on the coarse grid at every iteration of the solver on the fine grid. For assessment purposes, they have been implemented in the A-DEF2 solver used for the pressure Poisson equation in an unstructured LES solver for incompressible flows. The novel solver created thereby has been tested for different settings; it has been chosen to focus here on real turbulent flow configurations.

This paper is organized as follows. After a quick introduction on the Deflated PCG method and its variant A-DEF2, the creation of initial guesses for the successive systems on the coarse grid based on the former computed solutions is discussed in Section 2, and Section 3 addresses the possibility of adjusting the convergence criterion on the coarse grid. Then, Section 4 presents the massively parallel implementation of these techniques, thanks to which the numerical results shown in Section 5 have been obtained on grids up to billions of cells. Finally, Section 6 concludes this article by summing up the developed methods and the resulting gains on computational times.

1.2. Deflation of CG: the A-DEF2 algorithm

In this paper, the creation of the CG and PCG algorithms will not be detailed, as the reader may refer to [12,13] for details.

Numerous variants of preconditioned, deflated and balancing algorithms have been introduced and compared by Tang et al. in [11], built thanks to strict combinations of the preconditioners corresponding to each method. The A-DEF2 variant of the DPCG, a novel combination of the deflation and preconditioning techniques, has been chosen here due to its stability and performance, shown in the aforementioned article. It is applied here by combining a standard geometrical deflation with non-overlapping projection vectors, as introduced by Vermolen et al. in [6], and a preconditioning by the inverse of the diagonal. These choices are practical, as they reduce the computational time required for initializing the solver. The A-DEF2 algorithm being implemented is described by Algorithm 1, with W the deflation matrix and $M = diag(A)$. In what follows, the right-hand side of the solver on the coarse grid $W^T(AM^{-1} - I)r_i$ is called b_i^C.

Algorithm 1. The A-DEF2 algorithm, to solve $Ax = b$ using the preconditioner M^{-1} and the deflation matrix W

Require: A, b, M^{-1}, W
  $\hat{A} \leftarrow W^T A W$
  Solve $\hat{A}d_{-1} = W^T b$
  $x_0 = W d_{-1}$
  $r_0 = b - Ax_0$
  Solve $\hat{A}d_0 = W^T(AM^{-1} - I)r_0$
  $w_0 = M^{-1}r_0 - Wd_0$
  $p_0 = w_0$
  for $k = 0, 1, \ldots$ until required convergence **do**
    $\alpha_{k+1} = \frac{r_k^T w_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_{k+1} p_k$
    $r_{k+1} = r_k - \alpha_{k+1} A p_k$
    Solve $\hat{A}d_{k+1} = W^T(AM^{-1} - I)r_{k+1}$
    $w_{k+1} = M^{-1}r_{k+1} - Wd_{k+1}$
    $\beta_{k+1} = \frac{r_{k+1}^T w_{k+1}}{r_k^T w_k}$
    $p_{k+1} = w_{k+1} + \beta_{k+1} p_k$
  end for