8th. World Congress on Computational Mechanics (WCCM8)
5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 July 5, 2008
Venice, Italy

Projected Krylov Methods for the Solution of Unsymmetric Augmented Systems

Dominique Orban[1]

[1] GERAD and École Polytechnique de Montréal
CP 6079, Succ. Centre-Ville, Montréal, Québec H3C-3A7, Canada
E-mail: Dominique.Orban@polymtl.ca, www.mgi.polymtl.ca/dominique.orban

1 Introduction

We consider the iterative solution of systems of the form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is square but not necessarily symmetric, $B \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$ and $d \in \mathbb{R}^m$. In a fluid flow context, systems such as (1) arise as Stokes subproblems in Navier-Stokes iterations when considering the flow of two or more immiscible fluids through a cavity and must be solved to compute a correction (u, p) in the velocity and pressure fields. The large size of such problems preclude a direct factorization of the coefficient matrix of (1). On the other hand, iterative methods applied to (1) usually perform very poorly. We are particularly interested in the case where A may not be assembled explicitly but rather, matrix-vector products with A may be obtained by calling a function. We assume that B is available explicitly.

Whenever A is symmetric and positive definite, (1) represents the first-order optimality conditions of the equality-constrained quadratic program

$$\begin{array}{ll} \underset{u\in\mathbb{R}^n}{\text{minimize}} & -b^Tu + \frac{1}{2}u^TAu \\ \text{subject to} & Bu = d. \end{array} \tag{2}$$

In this type of application, the density of the coefficient matrix is mostly due to A. Assuming that B has full row rank, it is often feasible to compute a factorization of the *projection matrix*

$$\begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}, \tag{3}$$

where G is a sparse symmetric approximation to A that is positive definite over the nullspace of B. A particularly efficient iterative method for (2) is then the projected preconditioned conjugate gradient algorithm often used in nonlinear optimization contexts. This method typically requires the factorization of a single projection matrix and one matrix-vector product with A per iteration.

We examine similar Krylov-type iterations for the case where A is unsymmetric and present a methodology by which to derive projected Krylov methods for systems of the form (1). We concentrate on the Bi-CGSTAB and TFQMR families of methods. The methods we consider are akin to so-called projection methods, which are sometimes regarded as being too expensive and only effective on systems in which A is diagonally dominant. We hope that this paper will correct that reputation by showing that efficient projections combined with the appropriate Krylov iteration make for a very competitive numerical method.

Let $Z \in \mathbb{R}^{q\times n}$ be a matrix whose rows form a basis for the nullspace of B. Any solution u^* to (1) may be written $u^* = Zu_Z^* + B^Tu_B^*$, so that (1) yields $BB^Tu_B^* = d$, which uniquely determines u_B^* and leaves u_Z^* as a solution to $Z^TAZu_Z^* = Z^T(b - AB^Tu_B^*)$. Applying any Krylov method to the latter system with a preconditioner of the form $M = Z^TGZ$, where G is such that M is positive definite, is equivalent to applying the same Krylov method with A as coefficient matrix, with preconditioning steps replaced by projections computed via (3), and without recourse to computing Z. Special care must be observed in an implementation of this scheme as severe numerical cancellation likely occurs, especially in the projection steps. We will present a remedy to this difficulty.

2 Implementation and Numerical Results

We only briefly illustrate the projected Bi-CGSTAB algorithm in this abstract. Factorization of the projection matrix is performed by the multi-frontal symmetric indefinite MA57 from the Harwell Subroutine Library. The stopping test implemented by default is triggered when the projected Bi-CG residual vector s_k is small or when the residual vector r_k satisfies a Galerkin-type condition, or if the total number of matrix-vector products exceeds $2n_A$, where n_A is the order of the $(1,1)$ block matrix A. We compare the projected Bi-CGSTAB approach with a direct LU factorization of (1). The comparison is based on the total solution time and memory requirements. The LU factorization is realized by means of the UMFPACK package. In the following test, we choose $G = I$. The test was performed with the Intel compiler 10.0 on a 2.4 GHz Intel Core 2 Duo Apple laptop with 4 GB of memory. The sparsity pattern of the coefficient matrix, of order 7761, is show in the leftmost plot of Fig.1. The $(1,1)$ block, the $(1,2)$ block and the whole augmented matrix have density 0.334%, 0.457%, and 0.351% respectively. The projection matrix being symmetric, we only store its lower triangle, which has a density of 0.047%. Its factors have a density of 0.092%—a minor example of fill-in.

Factorization of (3) was realized in 0.14 seconds and the algorithm performed $2,523$ iterations and a total of $5,046$ matrix-vector products before reaching convergence in 23.83 seconds. The total running time on this example is thus 23.97 seconds. The rightmost plot of Fig.1 shows the residual history. The norm of the residual vector is 6.69e−09, the relative residual is 1.13e−08 and the relative error with the solution found with the LU factorization is 5.84e−06. In the LU factorization, the L factor was found to have $568,780$ entries and U was found to have $543,144$, densities of 0.944% and 0.902% respectively, a substantially more important fill-in. In this small-scale example, the LU factorization was performed in a mere 0.33 seconds.

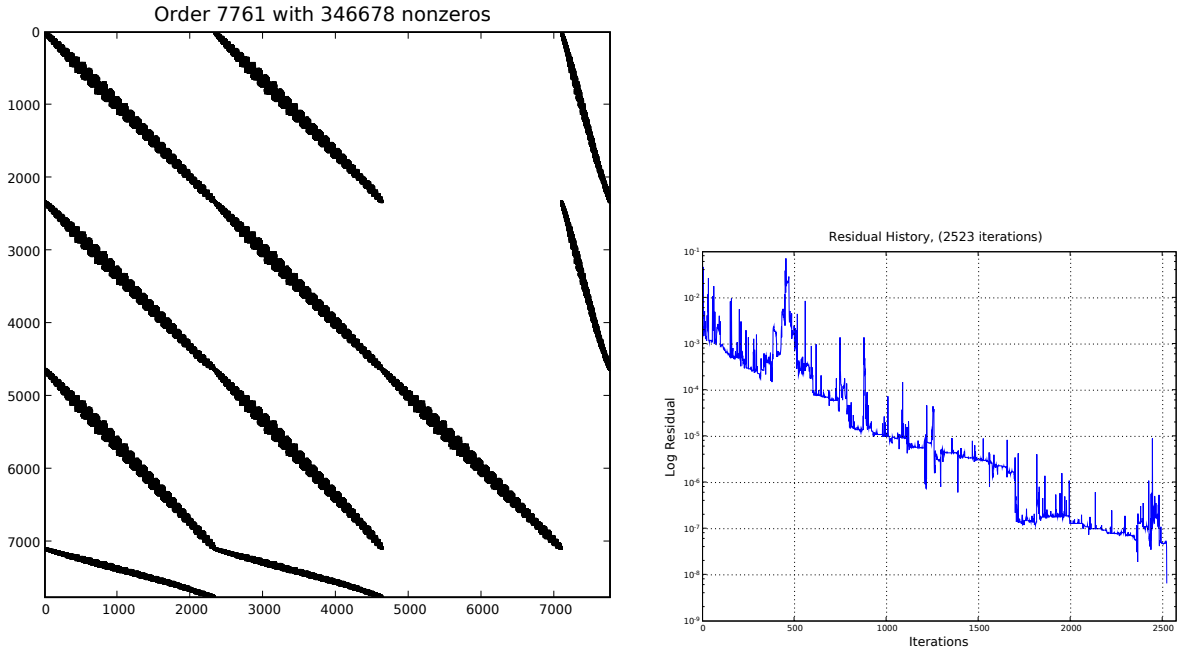

Figure 1: Left: Sparsity pattern of the linear system occurring at iteration 8 of a Navier-Stokes process. Right: Residual history of the PBCGTAB method.