
Dominique Orban
**Projected Krylov Methods for the Solution of
Unsymmetric Augmented Systems**

Ecole Polytechnique de Montreal
Department of Mathematics Industrial Engineering
CP 6079
Succ Centre-Ville
Montreal H3C 3A7
Quebec
Canada
dominique.orban@gerad.ca

We consider the iterative solution of systems of the form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} up = bd, \quad (1)$$

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 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

$$\begin{aligned} & \underset{u \in \mathbb{R}^n}{\text{minimize}} && -b^T u + \frac{1}{2} u^T A u \\ & \text{subject to} && B u = d. \end{aligned} \quad (2)$$

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*

$$\begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}, \quad (3)$$

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 $u^* = Zu_z^* + B^T u_B^*$, so that (1) yields $BB^T u_B^* = d$, which uniquely determines u_B^* and leaves u_z^* as a solution to $Z^T A Z u_z^* = Z^T (b - AB^T u_B^*)$. Applying any Krylov method to the latter system with a preconditioner of the form $M = Z^T G Z$, 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.

In our implementation, 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.

We present results on systems arising from the discretization of Navier-Stokes equations for the flow of one or more immiscible incompressible fluids. Some test cases involve a potentially moving obstacle taken into account by the fictitious domains method.