



Preconditioned iteration for saddle-point systems with bound constraints arising in contact problems

A. El maliki^{a,b}, M. Fortin^a, J. Deteix^a, A. Fortin^{a,*}

^a GIREF, Département de mathématiques et de statistique, Pavillon Vachon, 1045 avenue de la médecine, Université Laval, Québec, Canada G1V 0A6

^b Ecole Nationale de Commerce et de Gestion de Casablanca, Université Hassan II Mohammedia-Casablanca, Beau site, B.P 2725 Ain Sebaâ, Casablanca, Morocco

ARTICLE INFO

Article history:

Received 1 May 2012

Received in revised form 31 August 2012

Accepted 10 October 2012

Available online 5 November 2012

Keywords:

Mixed iteration

Generalized Conjugate Residual

Block preconditioner

Projection procedure

Hierarchical preconditioner

Schur complement approximation

ABSTRACT

We introduce a new algorithm, which we shall call Mixed Iteration, or by the acronym Mix-It, for the solution of saddle point problems arising from minimization problems under equality and/or inequality constraints. Such constrained problems are found in many applications and we shall be specially interested in contact problems in solid mechanics. The algorithm is however general and can be applied to other situations. We present a real saddle point algorithm in the sense that the primal variables and Lagrange multipliers unknowns are solved simultaneously. The effectiveness of the proposed method for large-scale simulations is improved by a block preconditioner and an adequate treatment of inequality constraints. The block preconditioner involves the solution of two subsystems associated respectively with the primal and the dual variables. We present some numerical results to illustrate the performances of the proposed method.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Although the algorithms that we shall describe are quite general, our motivation will be contact problems in solid mechanics. A realistic simulation for a three-dimensional contact problem in linear or nonlinear elasticity, as described in Section 4, implies the solution of very large saddle point systems for which efficient iterative methods capable of dealing with equality and inequality constraints are badly needed. Several approaches have been developed such as incremental methods [1,2] and mathematical programming techniques as in Nguyen and de Saxcé [3]. Other techniques take different forms such as penalty methods [3], Uzawa-type methods (see [4]) and projected gradient methods (see [5–8] and the references therein). These methods generally iterate on the dual problem (the Lagrange multipliers) and each iteration requires the solution of the primal system. For large contact problems, this is very time consuming since at each iteration, the system associated to the stiffness matrix has to be fully solved. Such contact problems can also be solved efficiently using algorithms based on an active set strategy combined with multigrid methods as was done by Wohlmuth and coworkers [9,10]. Many other efficient contact algorithms have been developed in the past few years. We refer the interested reader to [11–14]. We propose an iterative approach to saddle point systems where the primal

and dual variables are solved simultaneously. Our method can be seen as an improved version of the discrete Arrow–Hurwicz’s method introduced in [4]. We refer to the comprehensive review of Benzi, Golub and Liesen [15], for a discussion of saddle-point algorithms.

The key ingredient is to employ this variant as a preconditioner for the Conjugate Gradient Method or the Generalized Conjugate Residual (GCR) algorithm as described in Eisenstat et al. [16]. An appropriate treatment of inequality constraints is then added through a projection step. We thus obtain a what we call the Mixed Iteration with projection method (P-Mix-It-GCR). The choice of the GCR method is motivated by many factors and we shall come back to it.

Mixed formulations (primal–dual variables) lead to symmetric and indefinite systems. In the case of nonlinear elasticity, described in Section 4, the symmetry can be lost when deformation-dependent pressure loads are present or when the variation of the normal vector is included in the stiffness matrix.

The GCR method is a non symmetric variant of the conjugate gradient (CG) method. It is flexible in the sense that it allows variable preconditioners. It was shown in [17] to be equivalent to FGMRES, the flexible version of the GMRES method. When treating inequality constraints, we found that GCR which can be applied to non symmetric and indefinite systems natural to implement, which was not the case for GMRES.

To reduce storage requirements of the GCR method, the truncated version GCR (m) is employed, where m is the number of direction vectors being kept in the GCR method. We therefore

* Corresponding author.

E-mail addresses: elmaliki@giref.ulaval.ca (A. El maliki), mfortin@mat.ulaval.ca (M. Fortin), deteix@giref.ulaval.ca (J. Deteix), afortin@giref.ulaval.ca (A. Fortin).

use only the most recently computed direction vectors. The convergence properties of this method can be found in Ref. [16].

A specific preconditioning technique is introduced to accelerate convergence. It involves the approximate solution of two subsystems associated respectively with the primal and the dual variables. In our applications, the (HP) iterative method recently proposed in El Maliki et al. [18] is used for the primal variables. This method is very efficient when solving linear systems associated with stiffness matrices arising from the discretization of mechanical problems by quadratic elements.

Finding an efficient preconditioner for the block associated with the Lagrange multipliers (the Schur's complement) is a more difficult task. Some promising avenues are investigated in this paper but the optimal choice is still an open question.

An outline of the paper is as follows. The Sections 2 and 3 are devoted to the presentation of our iterative methods and preconditioning techniques. In Section 4, we shall then present the contact formulation. In Section 5, we present some techniques to approximate the stiffness and Schur's complement matrices. Numerical results are presented in the last section.

2. The mixed iteration algorithm with equality constraints

Let us start from a quadratic programming problem.

$$\inf_{(Cv=g)} \frac{1}{2} (Kv, v) - (f, v) \quad (1)$$

where $(\cdot, \cdot)$ refers to a scalar product, K is a symmetric and positive definite matrix, C is the matrix of constraints and v will be called the primal variable. Introducing a Lagrange multiplier, this can be written as a saddle point problem,

$$\inf_v \sup_{\lambda} \mathcal{L}(v, \lambda) = \frac{1}{2} (Kv, v) - (\lambda, Cv - g) - (f, v). \quad (2)$$

The optimality conditions of (2) can be written in the following block matrix form:

$$\mathcal{F} \begin{bmatrix} u \\ \lambda \end{bmatrix} := \begin{bmatrix} K & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \quad (3)$$

The Lagrange multiplier λ will be called the dual variable.

It is possible to solve directly the full system (3) but this approach has important drawbacks. The corresponding matrices are almost always too large to be solved by direct methods and moreover the system is indefinite. Its resolution by a direct method could require pivoting and is difficult by standard iterative methods.

A very common technique to solve the above problem is the dual iteration method or Uzawa's method. Given λ_0 , we solve the primal problem in u and update λ by

$$\begin{cases} Ku_k = f - C^T \lambda_k \\ \lambda_{k+1} = \lambda_k + \rho(Cu_k - g). \end{cases} \quad (4)$$

The parameter ρ can be chosen by standard methods, such as the steepest descent method. Furthermore, this algorithm can be improved by introducing a conjugate gradient variant as in Fortin and Glowinski [19]. This method is efficient but the cost of solving for u_k in (4) can be high. It was soon realized that an incomplete resolution was often sufficient. This led to what was called in [19] the Arrow–Hurwicz's method.

$$\begin{cases} u_{k+1} = u_k + \rho_1 \tilde{K}^{-1} (f - Ku_k - C^T \lambda_k) \\ \lambda_{k+1} = \lambda_k + \rho_2 (Cu_{k+1} - g) \end{cases} \quad (5)$$

$\tilde{K}^{-1}$ is some approximation of K^{-1} . This means that linear systems involving K are not solved with high accuracy. This is the case when

only a few iterations of an iterative method is used. If $\tilde{K}^{-1} = K^{-1}$ and $\rho_1 = 1$, this falls back on the dual iteration method. If the matrix K is symmetric, to obtain a symmetric preconditioner, we add a third step which modifies the displacement unknowns.

$$\begin{cases} u_{k+\frac{1}{2}} = u_k + \rho_1 \tilde{K}^{-1} (f - Ku_k - C^T \lambda_k) \\ \lambda_{k+1} = \lambda_k + \rho_2 (Cu_{k+\frac{1}{2}} - g) \\ u_{k+1} = u_{k+\frac{1}{2}} - \tilde{K}^{-1} C^T (\lambda_{k+1} - \lambda_k) \end{cases} \quad (6)$$

We shall refer to such an extension, with a small abuse of language, as being a symmetric preconditioner even if K is not.

The main drawback of this method was the lack of a good procedure to choose the parameters ρ_1 and ρ_2 . A way to circumvent this is to consider the Arrow–Hurwicz method or its full version as a preconditioner for another iterative method such as GMRES or GCR.

To do so, let us start with the approximate factorization $\mathcal{LDU}$ of the block matrix $\mathcal{F}$ from system (3).

$$\mathcal{F} \simeq \begin{bmatrix} I & 0 \\ C\tilde{K}^{-1} & I \end{bmatrix} \begin{bmatrix} \tilde{K} & 0 \\ 0 & -\tilde{S} \end{bmatrix} \begin{bmatrix} I & \tilde{K}^{-1}C^T \\ 0 & I \end{bmatrix} := \mathcal{LDU} \quad (7)$$

where $\tilde{S}$ is an approximation of the Schur complement $S = CK^{-1}C^T$ and $\tilde{K}$ is an approximation of the stiffness matrix K . If $\tilde{K} = K$ and $\tilde{S} = S$, we indeed have an exact factorization. Note that:

$$(\mathcal{LDU})^{-1} = \begin{bmatrix} I & -\tilde{K}^{-1}C^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{K}^{-1} & 0 \\ 0 & -\tilde{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -C\tilde{K}^{-1} & I \end{bmatrix} \quad (8)$$

which can be used as an approximate solver. When K is symmetric, we have $\mathcal{U} = \mathcal{L}^T$ and the full block decomposition $\mathcal{F} \simeq \mathcal{LDL}^T$ is symmetric but indefinite. In this case, the preconditioner can be combined with the conjugate gradient method or the generalized conjugate residual method.

Let (r_{u_k}, r_{λ_k}) be the residual of the system (3). Applying this approximate solver is equivalent to solve $(\delta u, \delta \lambda)^T = (\mathcal{LDU})^{-1} (r_{u_k}, r_{\lambda_k})^T$ which leads to three subsystems, two with the matrix $\tilde{K}$ and one with the matrix $\tilde{S}$:

$$\begin{cases} \delta^* = \tilde{K}^{-1} r_{u_k} \\ \delta \lambda = \tilde{S}^{-1} (C \delta^* - r_{\lambda_k}) \\ \delta u = \tilde{K}^{-1} (r_{u_k} - C^T \delta \lambda) = \delta^* - \tilde{K}^{-1} C^T \delta \lambda. \end{cases} \quad (9)$$

which is analogous to (6). Dropping $\mathcal{L}$ or $\mathcal{U}$, we obtain two unsymmetric preconditioners:

$$\mathcal{DU} = \begin{bmatrix} \tilde{K} & C^T \\ 0 & -\tilde{S} \end{bmatrix} \quad \text{with} \quad (\mathcal{DU})^{-1} = \begin{bmatrix} \tilde{K}^{-1} & \tilde{K}^{-1}C^T\tilde{S}^{-1} \\ 0 & -\tilde{S}^{-1} \end{bmatrix}. \quad (10)$$

$$\mathcal{LD} = \begin{bmatrix} \tilde{K} & 0 \\ C & -\tilde{S} \end{bmatrix} \quad \text{with} \quad (\mathcal{LD})^{-1} = \begin{bmatrix} \tilde{K}^{-1} & 0 \\ \tilde{S}^{-1}C\tilde{K}^{-1} & -\tilde{S}^{-1} \end{bmatrix}. \quad (11)$$

Applying (11) yields

$$\begin{cases} \delta u = \tilde{K}^{-1} r_{u_k} \\ \delta \lambda = \tilde{S}^{-1} (C \delta u - r_{\lambda_k}). \end{cases} \quad (12)$$

which is analogous to (5). For such a preconditioner, it has been shown by Murphy et al. [20] that a preconditioned Krylov subspace method converges in two iterations. The infrastructure for the implementation of this block triangular preconditioner is already incorporated in PETSC package [21] and was studied by Klawonn

Download English Version:

<https://daneshyari.com/en/article/6918191>

Download Persian Version:

<https://daneshyari.com/article/6918191>

[Daneshyari.com](https://daneshyari.com)