# RESOLUTION METHODS DEDICATED TO STRESS FORMULATED CONTACT PROBLEMS

**Kuss François, Lebon Frédéric**

Laboratoire de Mécanique et d'Acoustique
Chemin Joseph Aiguier – B.P. 121
F-13000 Marseille Cedex

*lebon@lma.cnrs-mrs.fr* (Lebon Frédéric)

**Abstract**

This paper deals with numerical methods for unilateral contact problems with dry friction (Signorini-Coulomb problems) written in terms of stresses (dual formulation). The formulation of the problem is first presented, we use the equilibrium finite element method to discretize it. With this method, the stress field is obtained by derivation of an Airy stress function which nodal values are taken to be the degrees of freedom. Therefore, forces can not be prescribed directly as displacements are prescribed in the classical finite element method (primal formulation). Forces are thus introduced in the system using Lagrange multipliers. For this reason, the matrix system is not positive definite and dedicated solvers have to be used. Four solvers are thus proposed, three of them are based on the condensation of the problem on the contact zone, using various partitions of the global matrix system and various linear solvers to perform condensation. The obtained condensed system is then solved and contact and friction conditions are applied using a classical Gauss Seidel relaxation algorithm. One solver is based on the use of the augmented Lagrangian version of the problem and is solved using the Uzawa algorithm. Two solvers are shown to be very efficient and allow the problem to be solved quickly.

**Keywords: Contact, friction, equilibrium finite elements, augmented Lagrangian, condensation.**

**Presenting Author's Biography**

Frédéric Lebon Professor PhD in Applied Mathematics 1989, Aix-Marseille 1 University, Habilitation in Mechanics, 1999, Montpellier 2 University. 2004-present In charge for the Contact Mechanics Modelling group, Mechanics and Acoustics Laboratory (UPR-CNRS 7051). 2003-present Professor, Provence Aix-Marseille 1 University, Mechanics and Acoustics Laboratory (UPR-CNRS 7051). 2001-2003 Professor, Claude Bernard Lyon 1 University, Civil Engineering Department, Materials and Structures Mechanics Laboratory (EA 1897), In charge for the Numerical Modelling group. 2000-2001 Visiting Professor Tor Vergata Rome 2 University, Civil Engineering Department 1989-2001 Assistant Professor, Montpellier 2 University, Mechanics and Civil Engineering Laboratory (URA CNRS 1214 then UMR 5508).

*ISBN 978-3-901608-32-2*  

# 1 Introduction

In this paper, we propose original solvers for the dual contact problem with dry friction.

In the second section, the problem is briefly presented, we use equilibrium finite elements to discretize it, due to prescribed forces and to contact and friction conditions, the problem is non positive definite and non linear.

In the third section, four solvers are proposed to solve this system. Three are based on condensation strategies, one is based on the Uzawa algorithm.

In the fourth section, we compare these solvers on two examples of various sizes. Classical preconditioners are evaluated on iterative solvers applied to the proposed examples. Then all solvers are compared in terms of computation times and of memory requirements.

# 2 The contact problem

## 2.1 Mechanical problem

We consider a deformable body, $\Omega$ in receding contact with a rigid foundation. The boundary $\Gamma$ is split into three parts, $\Gamma_D$, $\Gamma_F$ and $\Gamma_C$ so that $\Gamma = \bar{\Gamma}_D \cup \bar{\Gamma}_F \cup \bar{\Gamma}_C$.

We take $\sigma = \sigma_{ij}(u) = A_{ijkl}\epsilon_{kl}(u)$ to denote the stress tensor and $\alpha$ to denote the flexibility tensor. The body is in contact on $\Gamma_C$. The contact is governed by the Signorini unilateral conditions and the quasi Coulomb's friction law. On the contact boundary, the stress vector is decomposed into normal and tangential parts. The body is subjected to volume forces $F = (F_i)$ in $\Omega$ and to surface forces $f = (f_i)$ on $\Gamma_F$. We assume that $F \in [L^2(\Omega)]^d$ and $f \in [L^2(\Gamma_F)]^d$. The displacement is prescribed on $\Gamma_D$. Here, we focus on the stress formulation of the problem, proposed in [1]:

**Problem** $(\mathbf{P_s})$. Find a stress field $\sigma : \Omega \rightarrow H$ such that

$$\left\{ \begin{array}{l} \sigma \in \Sigma(\sigma) \\ b(\sigma, \tau - \sigma) \geq l(\tau - \sigma) \quad \forall \tau \in \Sigma(\sigma) \end{array} \right. \tag{1}$$

with $b(\sigma,\tau) = \int_\Omega A^{-1}\sigma \cdot \tau dx$
and $l(\tau) = \int_{\Gamma_D} u_0 \tau \cdot n ds$.
We define the statically admissible sets:

$$\begin{array}{l} H = \{\sigma = (\sigma_{ij}),\ \sigma_{ij} = \sigma_{ji} \in L^2(\Omega)\} \\ H_{F,f} = \{\sigma \in H;\ \sigma_{ij,j} + F_i = 0 \text{ in } \Omega\,, \\ \sigma_{ij} n_j = f_i \text{ on } \Gamma_F\} \\ \Sigma(\tau) = \{\sigma \in H_{F,f};\ \sigma_n \leq 0\,, \\ |\sigma_t| \leq -\mu\tau_n \text{ on } \Gamma_C\}. \end{array} \tag{2}$$

# 3 Discretization of stress formulations

## 3.1 Discrete formulation

In this part, we present the discretization of the dual problem, we reduce the problem to that of the complementary energy in order to simplify its expression. The problem is then written formally as a problem without any contact or friction; it is discretized using equilibrium finite elements introduced by Fraeijs de Veubeke in [2], and recently used by Kempeneers in [3]. This type of element has been used in some numerical cases: by Zavelani-Rossi [4] in a study on plane structures in the presence of plasticity and cracks, by Wieckowski and al. [5] for the elastoplastic analysis of plane structures, and by Bisegna and al. [6] [7] for solving the Signorini and Coulomb problem in the case of a plane elastic structure.

In this work, we used the Hsieh Clough and Tocher triangular element. In order to verify the local equilibrium condition, an Airy stress function is interpolated on the element, the stress field is the curl of the Airy function and is thus in equilibrium. Prescribed forces and contact forces are introduced, using Lagrange multipliers, in the complementary energy functional:

$$\begin{array}{l} \pi^* = \frac{1}{2}[\phi]^T[S][\phi] - [\phi]^T[q] + [\lambda]([C_F][\phi] - [F]) \\ +[\lambda'](\overline{[C_C][\phi]} - [F_C]) \end{array} \tag{3}$$

Since contact and friction conditions have still to be applied, the obtained matricial system is non linear, constrained and non positive definite. Dedicated solver need to be used, and are thus presented in the following section.

$$\left[ \begin{array}{cccc} S & {C_F}^T & {C_C}^T & 0 \\ C_F & 0 & 0 & 0 \\ C_C & 0 & 0 & -I \\ 0 & 0 & -I & 0 \end{array} \right] \cdot \left[ \begin{array}{c} \phi \\ \lambda \\ \lambda' \\ F_C \end{array} \right] = \left[ \begin{array}{c} q \\ F \\ 0 \\ 0 \end{array} \right] \tag{4}$$

## 3.2 Condensation on the contact boundary

The goal of the condensation is to obtain the reduced (condensed) system:

$$[D][F_C] = [G] \tag{5}$$

to be solved with a relaxation algorithm. This condensed system can be obtained using various methods: "blind condensation", condensation using the SYMMLQ solver and a three-step solver.

### 3.2.1 Blind condensation

The first way to obtain this system is to apply the basic idea of condensation, starting from the system (4) and taking the first three unknowns as the internal unknowns $\phi_i$ the symbolic system to be reduced is:

$$\left[ \begin{array}{cc} M_{11} & M_{12} \\ M_{21} & 0 \end{array} \right] \cdot \left[ \begin{array}{c} \phi_i \\ F_C \end{array} \right] = \left[ \begin{array}{c} U_i \\ 0 \end{array} \right] \tag{6}$$

By solving:

$$\phi_i = M_{11}^{-1}U_i - M_{11}^{-1}M_{12}F_C \tag{7}$$

the condensed system is obtained:

$$M_{21}M_{11}^{-1}M_{12}F_C = M_{21}M_{11}^{-1}U_i \Rightarrow [D][F_C] = [G] \tag{8}$$

 

Inverting $M_{11}$ could be very expensive, the resolution of Eq. (7) can be performed using a linear solver, $M_{11}$ is factorised one time and $\phi_i$ is obtained by $(1+n_C)$ triangular matrices multiplications. Since the matrix $M_{11}$ is not positive definite and can not be stored using efficient storage scheme, this first method is very expensive and has not been used here.

#### 3.2.2 SYMMLQ condensation

Since the slowness of the previous condensation strategy comes from the inefficiency of direct solvers applied to non positive definite matrices, we used the SYMMLQ iterative method to solve the previous $(1 + n_C)$ systems. The SYMMLQ method was introduced in [8]; it is a conjugate gradient like method for solving symmetric indefinite linear systems. It solves the projected system and keeps the residual vectors orthogonal to all previous ones. The efficiency of this algorithm highly depends on the choice of a preconditioner, an efficient diagonal block preconditioner, presented in [9], is thus used here.

#### 3.2.3 Three-step condensation

We propose here an efficient condensation method in order to take advantage of the structure of each matrix. Note that in the following, the brackets will be omitted on matrices to simplify the notations.The condensation has three major steps:

Step 1: Evaluation of $\phi$ such that:

$$\phi = S^{-1}q - S^{-1}{C_F}^T\lambda - S^{-1}{C_C}^T\lambda' \qquad (9)$$

which can be achieved by two ways:

- Use of a direct solver: $S$ is factorised one time using $LDL^T$ factorization with skyline storage and the products $S^{-1}q$, $S^{-1}{C_F}^T$ and $S^{-1}{C_C}^T$ are obtained by $(1 + n_f + n_C)$ system resolutions. The costliest part of this step is thus done only one time.
- Use of a conjugate gradient algorithm (CG): an efficient preconditioner is computed one time and allows to obtain quickly the products presented above by $(1 + n_f + n_C)$ system resolutions.

Step 2: $\phi$ is then introduced in lines 2 and 3 of Eq. (4), and $\lambda$ is obtained by $(1 + n_C)$ resolutions a full symmetric $n_f$ by $n_f$ matrix. Here again, the matrix is factorised only one time.

Step 3: $\lambda$ is the introduced in line 2 of Eq. (4) and $\lambda'$ is obtained by the same process ($(1 + n_C)$ resolutions a full symmetric $n_C$ by $n_C$ matrix). Finally, $\lambda'$ is introduced in the last line of Eq. (4) and the condensed system is obtained. Please note that the last two steps involve full symmetric matrices, but since $n_f$ and $n_C$ are small compared to $n_\phi$, their factorization is not expensive.

### 3.3 Condensed system solver

The condensed system, obtained using one of the condensation methods, is solved using a Gauss Seidel algorithm which allows to apply contact and friction conditions. This algorithm is presented in [10], but since its computation time is usually less than 1% of the total computation time, it is not presented here.

Initial guess of $\phi$, $\lambda$, $\lambda'$, and $F_C$

**Until** convergence, iteration $k$:

1- Determination of $\tilde{U}^k(\lambda^k, \lambda'^k, F_C^k)$

2- Determination of $\phi^k$ by solving $\tilde{S}\phi^k = \tilde{U}^k$

3- Update of $F_C$:
$$F_C^{k+1} = P_{\Sigma(F_C)}(C_C\phi^k - \frac{\lambda' k}{2})$$

4- Update of $\lambda$ and $\lambda'$:
$$\lambda^{k+1} = \lambda^k + \rho(C_F\phi^k - F)$$
$$\lambda'^{k+1} = \lambda'^k + \rho(C_C\phi^k - F_C)$$

Convergence if $i_C^k$ small

**End of loop**

Where $P_{\Sigma(F_C)}$ is the projection operator in the Signorini Coulomb friction cone.

Fig. 1 The Uzawa algorithm

### 3.4 Augmented Lagrangian formulation

Another way to solve Eq. (3) is to modify it to obtain an augmented Lagrangian formulation. The functional becomes:

$$\begin{array}{l} \pi_a* = \frac{1}{2}\phi^T S\phi - \phi^T q \\ +\lambda(C_F\phi - F) + \lambda'(C_C\phi - F_C) \\ +\frac{r}{2}(C_F\phi - F)^2 + \frac{r}{2}(C_C\phi - F_C)^2 \\ with\ \ F_C \in \Sigma(F_C) \end{array} \qquad (10)$$

where $r$ is the penalty parameter. Cancelling the first variation of this functional in comparison with the variables $\phi$, $\lambda$, $\lambda'$ and $F_C$ gives the augmented system, solved using the Uzawa algorithm, presented on Fig. 1, where $\rho$ is the Uzawa's step. Please note that the augmented Lagrangian formulation is used to improve the convergence rate of the Uzawa algorithm. Since many iterations are usually needed to obtain convergence, the first step of this algorithm is performed using a direct skyline solver. The factorization of $\tilde{S}$, which is the costliest operation, is thus performed only one time and the factorised matrix is used at each step of the algorithm. The convergence indicator has been defined such as:

$$i_C^k = sup_i||r^k(i)|| \qquad (11)$$

$$\text{with } r^k = C_C\phi^k - F_C^{k+1} + C_F\phi^k - F \qquad (12)$$

the limit value of $i_C$ has been chosen to be $10^{-3}$ Newtons.

## 4 Numerical results

The algorithms presented here have been implemented in the computer code LMGC90 (http://www.lmgc.univ-

 

Tab. 1 Sizes of treated examples

| Name | $n\phi$ | $n_f$ | $n_C$ |
|---|---|---|---|
| Mesh0 | 761 | 96 | 38 |
| Mesh1 | 1973 | 144 | 80 |
| Mesh2 | 3105 | 176 | 120 |
| Mesh3 | 6287 | 232 | 160 |
| Mesh4 | 8360 | 392 | 200 |
| Mesh5 | 19128 | 588 | 480 |
| Mesh6 | 26538 | 708 | 576 |
| Mesh7 | 50483 | 920 | 640 |

montp2.fr/˜dubois/LMGC90/) and tested on the example of a steel tooth, presented in Fig. 2, with prescribed displacements on its left edge, and which is in contact on a rigid foundation where $\mu = 0.2$.

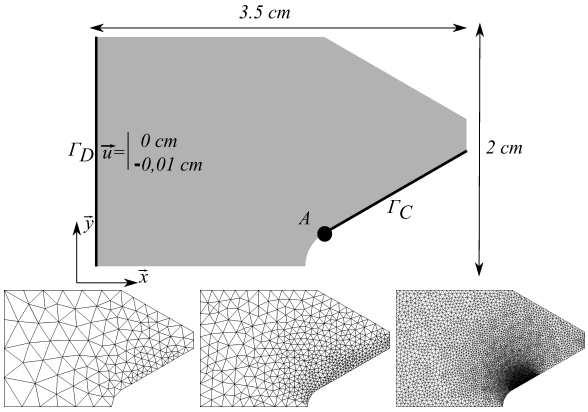


Fig. 2 The example of a tooth and its meshes (named 1, 2 and reference meshes)

#### 4.0.1 Descriptions of benchmarking examples and algorithms parameters

In this section, we compare the proposed resolution methods on the examples of Fig. 2 (Mesh0 to Mesh3) and on the same problem with four teeth (Mesh4 to Mesh7) with various sizes presented on Tab. 1. All meshes have been previously renumbered using the Cuthill McKee algorithm [11] and matrices have been equilibrated using diagonal equilibration to improve convergence rates and accuracy.

The following methods have been compared :

- The SYMMLQ condensation strategy.
- The three-step condensation using two different methods in the first level of condensation: A skyline $LDL^T$ solver, or a conjugate gradient solver. Those two methods will be named 3STEPD and 3STEPCG in the following.
- The resolution of the augmented Lagrangian formulation using the Uzawa algorithm.

When using the CG and SYMMLQ methods, the following approximations $S^*$ of $S$ have been tested as preconditioners:

- Drop tolerance incomplete Cholesky factorization (DTICx), with tolerance parameter $x \in [3; 6]$.
- Filling level incomplete Cholesky factorization (FLICx), with filling level factor $x \in [1; 5]$.
- The SSOR preconditioner (SS$\omega$), with relaxation parameter $\omega \in ]0; 2[$.
- The exact Cholesky factorization of $S$ (SKY).

Those preconditioning strategies are reviewed in [12]. Note that in the SYMMLQ case, these preconditioners are used to construct the block preconditioner. When using incomplete Cholesky factorisations, one can choose to compute line $i$ using the previous lines of the incomplete factorisation or of the full factorisation. The first strategy has been applied to the computation of FLIC and the second to the computation of DTIC. To use the Uzawa's algorithm, one has to choose the parameters $\rho$ and $r$ defined in the section 3.4. Since the parameter $r$ defines the importance of the product matrices $C_F^T C_F$ and $C_C^T C_C$ against $S$, the matrices $C_F$ and $C_C$ have been multiplied by the factors $p_F$ and $p_C$ such as:

$$p_F = \frac{\sqrt{mean(S)}}{mean(C_F)} \qquad p_C = \frac{\sqrt{mean(S)}}{mean(C_C)} \tag{13}$$

It allows to give the same importance to a chosen $r$ on all examples. In another hand, we had no rule to chose $\rho$ and $r$, thus we tried various combinations of both parameters in $[1; 20; 40; 60; 80; 100; 200; 300]$ for each example and took the best sets for the following comparisons.

Computations have been performed on an Intel P4 1.7Ghz computer with 1Gb Ram memory in order to evaluate performances on a middle-range computer.

#### 4.0.2 Influence of the preconditioner

Before comparing all methods, we analyse the influence of preconditioners on the SYMMLQ and 3STEPCG methods. The same behaviour has been observed on each example and with both methods, thus we focus on results obtained with the SYMMLQ method and on the Mesh3 example. The Computation times and memory requirements are represented on Fig. 3

First looking at the DTIC preconditioner, one can see that increasing the drop tolerance parameter decreases computation times. With the FLIC preconditioner, an average parameter exists, here this parameter is 3. By comparing both incomplete Cholesky factorisations, one can see that, even with an equivalent size (for example when comparing DTIC3 and FLIC4), the DTIC preconditioner is the more efficient, principally because it is constructed using the full Cholesky factorisation.

 

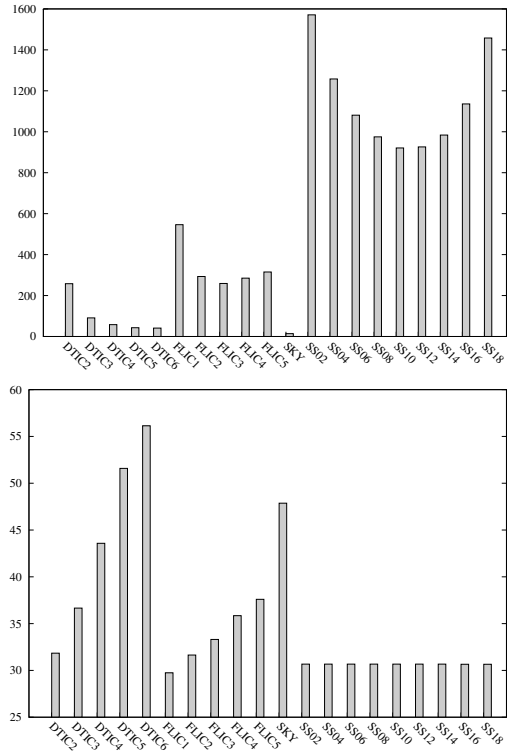


Fig. 3 Comparison of various preconditioners, up: computation times in seconds, down : Memory requirements in Mb.

With the SSOR preconditioner, the optimum parameter was around 1 on all treated examples, but this preconditioner was not very efficient on the kind of treated problems.

In the following, only the best preconditioners are used: the DTIC3 which gives a good compromise beetween computation times and memory requirements, the DTIC6 which gives very good computation times, and the full Cholesky factorisation in order to compare iterative and direct solvers.

### 4.0.3 Comparison of algorithms

The previous section permitted us to choose efficient conditioners in order to limit the number of compared methods. In this part, we use only the DTIC and SKY preconditioners with iterative methods. The following methods have been compared on examples of Tab. 1:

- The SYMMLQ condensation with the DTIC3, the DTIC6 and the SKY preconditioners, which will be respectively named SDTIC3, SDTIC6 and SSKY.
- The 3STEP condensation with the direct $LDL^T$ and the conjugate gradient solver, using the DTIC3 and the DTIC6 preconditioners, which will be respectively named 3STEPD, 3STEPCG3 and 3STEPCG6.
- The resolution of the augmented Lagrangian formulation using the Uzawa algorithm, named ALM.

The computation times and memory requirements are respectively summarised on Fig. 4 and Fig. 5.

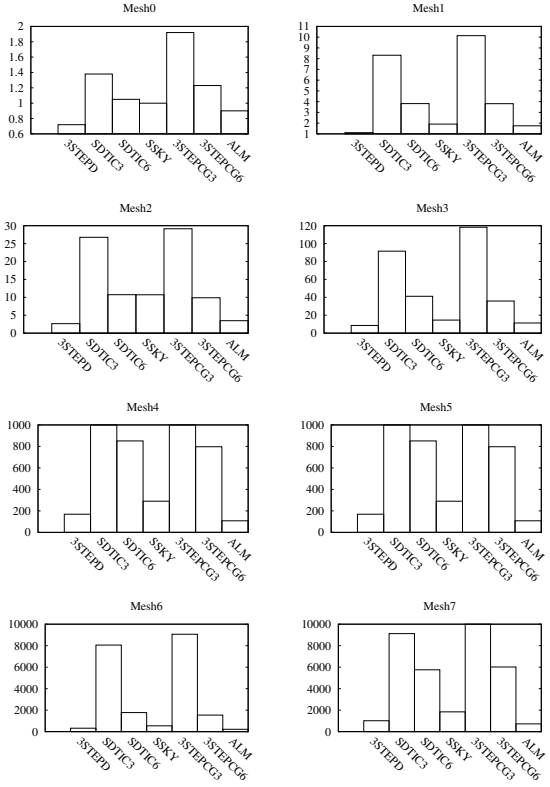


Fig. 4 Comparison of computation times (in seconds)

First, by comparing SDTIC3 with 3STEPCG3 and SDTIC6 with 3STEPCG6, one can see that the SYMMLQ method and the 3STEPCG method are equivalent when using equivalent preconditioners, but these four strategies are less efficient than others. In a general rule, iterative methods can only be as efficient as the 3STEPD and ALM methods when using very efficient preconditioners as in the case of SSKY strategy. But these methods need the storage of both the global matrix and of the preconditioner, they thus need a greater amount of memory than the 3STEPD and the ALM methods.

The 3STEPD and the ALM methods thus seams to be the more efficient methods. On all examples, both methods give equivalent computation times, the ALM method is slightly faster.

The main advantage of the ALM method is that it requires a very low memory amount whereas the 3STEPD methods needs the storage of intermediate matrices at each step of condensation. This drawback of the 3STEPD method could be overcame using an appropriated file storage.

The main advantage of the 3STEPD method is its robustness since it is not parameter dependent as the ALM method. Computation times are thus predictable since

 

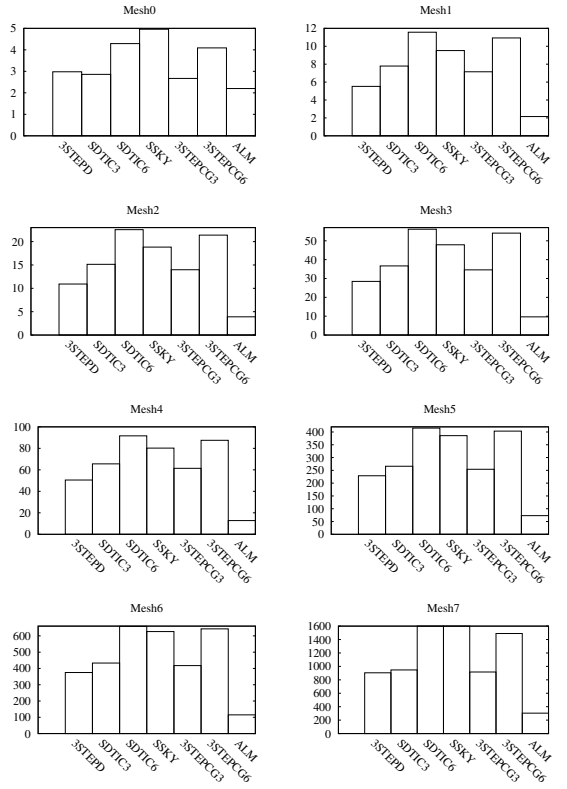


Fig. 5 Comparison of used memory (in Mb)

they principally depend on the profile of the flexibility matrix. In another hand, this method can be parallelized very easily.

## 5 Conclusion

In this paper, a dual method dedicated to unilateral contact problems with friction has been presented. Due to the type of finite elements used and to contact and friction conditions, the global system is non positive definite and non linear; it has thus to be solved using appropriated solvers.

Four solvers have been proposed, three based on the principle of condensation, one on the Uzawa algorithm. All solvers have been compared on examples of various sizes, two have been shown to be particularly efficient: the three-step direct condensation strategy and the Uzawa algorithm, the first has the advantage to be usable as a black box, the second needs a very low memory amount but needs the determination of the Uzawa step parameter.

## 6 References

[1] J.J. Telega. Quasi-static signorini's contact problem with friction and duality. *International Series on Numerical Mathematics*, 101:199–214, 1991.

[2] B. Fraeijs de Veubeke. Displacement and equilibrium models in the finite element method. *International Journal for Numerical Methods in Engineering*, 52:287–342, 2001.

[3] Martin Kempeneers, Pierre Beckers, J.P. Moitinho de Almeida, and Orlando J.B. Almeida Pereira. Modèles équilibre pour l'analyse duale. *Revue européenne des éléments finis*, 12:737–760, 2003.

[4] Adolfo Zavelani-Rossi. An equilibrium approach to plane problems. *Computers and structures*, 79:1877–1895, 2001.

[5] Zdzislaw Wieckowski, Sung-Kie Youn, and Byung-Sik Moon. Stress based finite element analysis of plane plasticity problems. *International Journal for Numerical Methods in Engineering*, 44:1505–1525, 1999.

[6] P. Bisegna, F. Lebon, and F. Maceri. Relaxation procedures for solving signorini-coulomb contact problems. *Advances in Engineering Software*, 35:595–600, 2004.

[7] P. Bisegna, F. Lebon, and F. Maceri. D-pana : a convergent block-relaxation solution method for the discretized dual formulation of the signorini-coulomb contact problem. *Comptes-Rendus Mathématiques Académie des Sciences Paris*, 333:1053–1058, 2001.

[8] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SINUM*, 12:617–629, 1975.

[9] C. Corral, I. Giménez, M. Tur, and J. J. Ródenas. Iterative preconditioned methods for the solution of contact problems by the finite element method. In B. H. V. Topping, G. Montero, and R. Montenegro, editors, *Proceedings of the Fifth International Conference on Engineering Computational Technology*, Stirlingshire, United Kingdom, 2006. Civil-Comp Press. paper 87.

[10] F Kuss and F Lebon. Dual methods for unilateral contact problems. In B. H. V. Topping, G. Montero, and R. Montenegro, editors, *Proceedings of the Fifth International Conference on Engineering Computational Technology*, Stirlingshire, United Kingdom, 2006. Civil-Comp Press. paper 106.

[11] P Lascaux and R Théodor. *Analyse numérique matricielle appliquée à l'art de l'ingénieur, tomes 1 et 2'*. Dunod, 2004.

[12] M. T Jones and P.E Plassmann. An improved incomplete cholesky factorization. *ACM transactions on mathematical software*, 21:5, 1995.

 