On the Solution of a Finite Element Approximation of a Linear Obstacle Plate Problem

Luis M. Fernandes † Isabel N. Figueiredo ‡ Joaquim J. Júdice‡

Abstract. In this paper the solution of a finite element approximation of a linear obstacle plate problem is investigated. A simple version of an interior point method and a block pivoting algorithm have been proposed for the solution of this problem. Special purpose implementations of these procedures are included and have been used in the solution of a set of test problems. The results of these experiences indicate that these procedures are quite efficient to deal with these instances and compare much favorable with the path-following PATH and the active-set MINOS codes of the commercial GAMS collection.

1 Introduction

In this paper we investigate the solution of a contact problem, that describes the equilibrium of a thin elastic clamped plate, that may come into contact with a rigid obstacle, by the action of forces. The formulation and the existence of a solution to this problem has been discussed elsewhere [8, 10]. In this paper we only address the geometrically linear plate, which corresponds to the hypothesis of small strain and originates a linear model, as described in [8].

The linear obstacle plate problem, whose unknown is the deflection of the middle plane of the plate, has a unique solution, since its variational formulation corresponds to an elliptic variational inequality [8]. In spite of the knowledge of the existence of solution, its exact analytical expression is in general impossible to determine, due to the complexity of the model. Therefore there is a need to use approximate and numerical methods in order to obtain an approximate solution. In this paper, we use the finite element method to define the discrete optimization model that determines the approximate solution.

Since the differential operator governing the linear obstacle plate problem is of fourth order, we choose the Bogner-Fox-Schmit rectangle [3], which is a finite element of class C^1, to obtain the discrete problem. Then, the resulting discrete variational inequality can be reformulated as a minimization problem with inequality constraints. In the literature, the most commonly used numerical methods chosen to solve this minimization problem, are penalty or Lagrangian multipliers approach methods [8, 10, 12], with the inherent drawbacks of ill conditioning, penalty sensitivity and lack of robustness. In this work, we investigate the efficiency of other different solution methods, that are not based in this type of approach.

An interior-point algorithm [16] and a block principal pivoting method [9] have been conveniently implemented and investigated on the solution of a set of test problems. To gain a better idea of their efficiency in these cases, we have also solved these problems by using the path-following PATH [4] and the active-set MINOS methods that are available in the commercial optimization collection GAMS (General Algebraic Modeling System) [2] . These experiences have shown that the interior-point and block-pivoting algorithms are quite efficient for dealing with the finite-dimensional obstacle problem under study and compare much favorably with those general-purpose codes of the GAMS collection.

The structure of the paper is as follows. In section 2 the differential and variational formulations and the finite element discretization are presented. The two complementarity algorithms are described in

†Departamento de Matemática, Escola Superior de Tecnologia de Tomar, 2300-313 Tomar, Portugal.
‡Departamento de Matemática, Universidade de Coimbra, Apartado 3008, 3001-454 Coimbra, Portugal.

section 3. The computational experience is reported in section 4. Finally, some conclusions are presented in the last section of the paper.

2 Finite Element Approximation

2.1 Differential Formulation

The differential equations, inequalities and boundary conditions governing the nonlinear obstacle problem, corresponding to a geometrically nonlinear and thin elastic clamped plate, can be stated in the following form

$$\text{Find} \quad (u_1, u_2, z) : \Omega \subset R^2 \to R^3, \quad \text{such that :}$$

$$\frac{Et^3}{12(1-\nu^2)} \Delta^2 z - t\Big[\sigma_{\alpha\beta}(\overline{u}, z) z_{,\beta}\Big]_{,\alpha} \geq f, \qquad \text{in} \quad \Omega, \tag{1}$$

$$z \geq \psi, \qquad \text{in} \quad \Omega, \tag{2}$$

$$\left(\frac{Et^3}{12(1-\nu^2)} \Delta^2 z - t\Big[\sigma_{\alpha\beta}(\overline{u}, z) z_{,\beta}\Big]_{,\alpha} - f\right)(z - \psi) = 0, \qquad \text{in} \quad \Omega, \tag{3}$$

$$\sigma_{\alpha\beta,\beta}(\overline{u}, z) = 0, \qquad \text{in} \quad \Omega, \tag{4}$$

$$u_1 = u_2 = z = \frac{\partial z}{\partial n} = 0, \qquad \text{in} \quad \partial\Omega. \tag{5}$$

In (1-5) the set $\Omega = \{x = (x_1, x_2) \in R^2\}$ is an open, bounded, connected subset of R^2, with Lipschtiz boundary $\partial\Omega$, and defines the middle plane of the plate. The constant t represents the thickness of the plate. The scalar functions $\psi : \Omega \to R$ and $f : \Omega \to R$ denote, respectively, the obstacle and the vertical force acting on the plate. The unknown $(u_1, u_2)(x_1, x_2)$ denotes the horizontal displacement and $z(x_1, x_2)$ the vertical displacement at the point $(x_1, x_2) \in \Omega$. We assume that the plate is made of a homogeneous and isotropic material, so the constants E and ν are respectively Young's modulus and Poisson's ratio. The greek indexes α, β... belong to the set $\{1, 2\}$; we also use the Einstein summation convention, that is $a_\alpha b_\alpha$ means $\sum_{\alpha=1}^{2} a_\alpha b_\alpha$; Δ^2 is the biharmonic operator, Δ is the Laplace operator and $\frac{\partial z}{\partial n}$ is the normal derivative of z. The notation $._{,\alpha}$ means partial derivative with respect to the component α of x and finally $\sigma = (\sigma_{\alpha\beta})$ denotes the membrane stress tensor whose definition is

$$\sigma_{\alpha\beta}(\overline{u}, z) = \frac{E}{1-\nu^2}\left[(1-\nu)\big(e_{\alpha\beta}(\overline{u}) + \frac{1}{2} z_{,\alpha} z_{,\beta}\big) + \nu\big(e_{\gamma\gamma}(\overline{u}) + \frac{1}{2} z_{,\gamma} z_{,\gamma}\big)\right] \tag{6}$$

where $e_{\alpha\beta}(\overline{u})$ are the components of the linear strain tensor defined by

$$e_{\alpha\beta}(\overline{u}) = \frac{1}{2}\Big(u_{\alpha,\beta} + u_{\beta,\alpha}\Big). \tag{7}$$

The nonlinearity of (1-5) is present in the definition of the stress tensor (6) and in the nonlinear terms of (1) and (3).

The linear obstacle plate problem, corresponding to a geometrically plate, can be obtained directly from (1-5), by neglecting the nonlinear terms, and takes the following simple form:

$$\text{Find} \quad z : \Omega \subset R^2 \to R, \quad \text{such that :}$$

$$\frac{Et^3}{12(1-\nu^2)} \Delta^2 z \geq f, \qquad \text{in} \quad \Omega, \tag{8}$$

$$z \geq \psi, \qquad \text{in} \quad \Omega, \tag{9}$$

$$\left(\frac{Et^3}{12(1-\nu^2)}\Delta^2 z - f\right)(z-\psi) = 0, \qquad \text{in} \quad \Omega, \tag{10}$$

$$z = \frac{\partial z}{\partial n} = 0, \qquad \text{in} \quad \partial\Omega, \tag{11}$$

where the vertical displacement z is the unknown. This linear problem is the subject of our investigation and represents the equilibrium position for a geometrically linear and thin elastic clamped plate that is constrained to the action of a vertical force and touches the obstacle.

2.2 Variational Formulation

In order to describe the finite element approximation, we first define the variational formulation corresponding to problem (8-11). To that end we introduce the Sobolev space

$$H_0^2(\Omega) = \left\{ z \in H^2(\Omega): \quad z_{|\partial\Omega} = 0 = \frac{\partial z}{\partial n}_{|\partial\Omega} \right\} \tag{12}$$

and the constraint set defined by the obstacle

$$K = \{z \in H_0^2(\Omega): \quad z \geq \psi \quad \text{in} \quad \Omega\}. \tag{13}$$

Then the variational formulation of problem (8-11) takes the following form:

$$\begin{aligned} &\text{Find} \quad z \in K: \\ &A(z, w-z) \geq F(w-z), \quad \forall \omega \in K. \end{aligned} \tag{14}$$

The forms appearing in (14) are defined by

$$\left[\begin{array}{l} A: H_0^2(\Omega) \times H_0^2(\Omega) \to R, \\ A(z,w) = \hat{D}\int_\Omega \{\nu\Delta z \Delta\omega + (1-\nu)\partial_{\alpha\beta} w \partial_{\alpha\beta} z\} d\Omega, \end{array}\right. \tag{15}$$

$$\left[\begin{array}{l} F: H_0^2(\Omega) \to R, \\ F(w) = \int_\Omega f w d\Omega \end{array}\right. \tag{16}$$

$$\tag{17}$$

where

$$\hat{D} = \frac{Et^3}{12(1-\nu^2)}. \tag{18}$$

We recommend [8, 10] for the justification of the variational formulation (14), its relation with the differential formulation and the proof of existence and uniqueness of solution of this model.

2.3 Discrete Problem

We assume that the domain Ω is a rectangular domain and is partitioned into a mesh of $m = n_1 n_2$ rectangles $\Omega^e = [x_1^e, y_1^e] \times [x_2^e, y_2^e]$, n_1 is the number of sub-intervals in the x_1 direction and n_2 the number of sub-intervals in the x_2 direction. The amplitudes of $[x_1^e, y_1^e]$ and $[x_2^e, y_2^e]$ are $h_1^e = y_1^e - x_1^e$ and $h_2^e = y_2^e - x_2^e$, respectively. Moreover we suppose that the mesh $\{\Omega^e\}_{e=1,\ldots,m}$ is affine equivalent to the reference element $\hat{\Omega} = [-1,+1] \times [-1,+1]$. The affine transformations are defined by the mapping

$$\begin{aligned} F^e: \quad &\Omega^e \longrightarrow \hat{\Omega} = [-1,+1] \times [-1,+1] \\ &(x_1, x_2) \longrightarrow (\xi, \eta) = \left(\tfrac{2}{h_1^e}(x_1 - x_c^e), \tfrac{2}{h_2^e}(x_2 - y_c^e)\right), \end{aligned} \tag{19}$$

where x_c^e, y_c^e are the middle points of $[x_1^e, y_1^e]$ and $[x_2^e, y_2^e]$, respectively, and (ξ, η) is a generic element of $\hat{\Omega}$.

The Bogner-Fox-Schmit finite element [3] is used for the approximation of the vertical displacement z. The 16 degrees of freedom characterizing this element are the values of z, $z_{,1}$, $z_{,2}$ and $z_{,12}$ at each vertex of Ω^e. The analytical expressions of the local shape functions, defined in the reference element $\hat{\Omega}$, are stated below:

$$\begin{array}{ll}
N_1^1(\xi,\eta)=\psi_1^0(\xi)\psi_1^0(\eta) & N_2^1(\xi,\eta)=\psi_2^0(\xi)\psi_1^0(\eta)\\
N_1^2(\xi,\eta)=\psi_1^1(\xi)\psi_1^0(\eta) & N_2^2(\xi,\eta)=\psi_2^1(\xi)\psi_1^0(\eta)\\
N_1^3(\xi,\eta)=\psi_1^0(\xi)\psi_1^1(\eta) & N_2^3(\xi,\eta)=\psi_2^0(\xi)\psi_1^1(\eta)\\
N_1^4(\xi,\eta)=\psi_1^1(\xi)\psi_1^1(\eta) & N_2^4(\xi,\eta)=\psi_2^1(\xi)\psi_1^1(\eta)\\
\\
N_3^1(\xi,\eta)=\psi_2^0(\xi)\psi_2^0(\eta) & N_4^1(\xi,\eta)=\psi_1^0(\xi)\psi_2^0(\eta)\\
N_3^2(\xi,\eta)=\psi_2^1(\xi)\psi_2^0(\eta) & N_4^2(\xi,\eta)=\psi_1^1(\xi)\psi_2^0(\eta)\\
N_3^3(\xi,\eta)=\psi_2^0(\xi)\psi_2^1(\eta) & N_4^3(\xi,\eta)=\psi_1^0(\xi)\psi_2^1(\eta)\\
N_3^4(\xi,\eta)=\psi_2^1(\xi)\psi_2^1(\eta) & N_4^4(\xi,\eta)=\psi_1^1(\xi)\psi_2^1(\eta)
\end{array} \tag{20}$$

for any $(\eta,\xi)\in[-1,+1]^2$, where ψ_1^0, ψ_1^1, ψ_2^0 and ψ_2^1 are the cubic Hermite polynomials defined on $[-1,+1]$ by

$$\begin{array}{ll}
\psi_1^0(\xi)=\frac{1}{4}(\xi-1)^2(\xi+2) & \psi_2^0(\xi)=\frac{1}{4}(\xi+1)^2(2-\xi)\\
\psi_1^1(\xi)=\frac{1}{4}(\xi-1)^2(\xi+1) & \psi_2^1(\xi)=\frac{1}{4}(\xi+1)^2(\xi-1).
\end{array} \tag{21}$$

At each finite element Ω^e the vertical displacement z is approximated by z_h, which satisfies

$$z_h(x_1,x_2)=\sum_{i=1}^{4}\big(z_i^1N_i^1+z_i^2\frac{h_1^e}{2}N_i^2+z_i^3\frac{h_2^e}{2}N_i^3+z_i^4\frac{h_1^eh_2^e}{4}N_i^4\big)\circ F^e(x_1,x_2), \tag{22}$$

where the unknowns z_i^j, for $j=1,...,4$ are the approximation values of z, $z_{,1}$, $z_{,2}$ and $z_{,12}$, respectively, at node i of Ω^e.

In order to describe the discrete problems, corresponding to (14), we must introduce some notations. Let n be the number of global nodes of the mesh, and assume the coefficients z_i^j, for $i=1,...,n$ and $j=1,...,4$, have been ordered in a linear numbering, so that $\{z_i^j\}_{(i,j)}$ may be identified with the vector $z\in R^{4n}$. Moreover, we introduce the following subsets of indices

$$\begin{array}{ll}
J_1,\ J_2\ J_3,\ J_4, & J=J_1\cup J_2\cup J_3\cup J_4\subset\{1,2,...,4n\},\\
L_1,\ L_2,\ L_3, & L=L_1\cup L_2\cup L_3\subset\{1,2,...,4n\},
\end{array} \tag{23}$$

where J_k, L_k represent the sets of indices related to the type k of global degrees of freedom and attached to the interior or boundary nodes of the mesh. The subscript $k=1$ refers to the displacement, $k=2$ to the first derivative of the displacement with respect to x_1, $k=3$ to the first derivative of the displacement with respect to x_2 and finally $k=4$ refers to the second mixed derivative of the displacement. The subset L refers to the degrees of freedom related to the boundary conditions (11) of the problem. If R and S are two sets of indices, $z\in R^{4n}$ and W is a matrix, we denote by z_R and W_{RS} the subvector of z and the submatrix of W respectively, whose components have the indices in R and S. We remark that the submatrix W_{RS} has dimension $|R|\times|S|$, where $|R|$, $|S|$ are the cardinals of the sets R, S. We also define the vector ψ of dimension n, which approximates the obstacle $\psi(.,.)$ by

$$\psi=(\psi_1,\psi_2,...,\psi_n)=\Big(\psi(x_1^1,x_2^1),\psi(x_1^2,x_2^2),...,\psi(x_1^n,x_2^n)\Big), \tag{24}$$

where (x_1^i,x_2^i) are the coordinates of the global node i of the mesh, with $i\in\{1,...,n\}$.

The approximate problem is obtained directly from (14) by replacing z by the approximation z_h defined in (22). Based on the choice of the finite elements described before and using the notations introduced in (23-24), the discrete problem associated to (14) takes the following form

$$\left\{\begin{array}{l}
\text{Find}\quad z\in R^{4n}\quad\text{such that :}\\
z_{J_1}\geq\psi,\quad z_{L_1}=z_{L_2}=z_{L_3}=0,\\
\left\{\begin{array}{l}(w-z)^T\Big[Cz-F\Big]\geq 0,\\ \forall w\in R^{4n},\ w_{J_1}\geq\psi,\quad w_{L_1}=w_{L_2}=w_{L_3}=0,\end{array}\right.
\end{array}\right. \tag{25}$$

where the matrix C is a constant sparse matrix of order $4n$, independent of z and F is a vector of dimension $4n$, associated to the forces. These matrix and vector are obtained by assembling the corresponding element matrix and element vector. In order to give the exact expressions of C and F we first define the vector of local shape functions N_i, associated with the local node i ($i = 1, 2, 3, 4$) of the Bogner-Fox-Schmit finite element

$$N_i = \begin{bmatrix} N_i^1 & N_i^2 & N_i^3 & N_i^4 \end{bmatrix} \tag{26}$$

and the associated vectors

$$N_i^e = \begin{bmatrix} N_i^1 & \frac{h_1^e}{2} N_i^2 & \frac{h_2^e}{2} N_i^3 & \frac{h_1^e h_2^e}{4} N_i^4 \end{bmatrix}, \qquad N^e = \begin{bmatrix} N_1^e & N_2^e & N_3^e & N_4^e \end{bmatrix} \tag{27}$$

where N^e is a vector of order 16. Then, the definitions of C and F at the element level are discussed below.

- The element matrix of C, denoted by C^e, is a 16×16 symmetric matrix, and it is the usual stiffness matrix for linear plate bending :

$$C^e = \frac{h_1^e h_2^e}{4} \left[\int_{\hat{\Omega}} \frac{t^2}{12} S_i^{e\,T} D S_j^e d\hat{\Omega} \right]_{i,j=1,2,3,4} \tag{28}$$

  with

$$D = \frac{tE}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}_{3\times 3} \qquad \text{and} \qquad S_i^e = \begin{bmatrix} \frac{4}{h_1^e h_1^e} N_{i,11}^e \\ \frac{4}{h_2^e h_2^e} N_{i,22}^e \\ 2\frac{4}{h_1^e h_2^e} N_{i,12}^e \end{bmatrix}_{3\times 4} . \tag{29}$$

- The element vector of F is a vector of dimension 16 , denoted F^e and such that

$$F^e = \frac{h_1^e h_2^e}{4} \int_{\hat{\Omega}} \hat{f} N^{e\,T} d\hat{\Omega} \tag{30}$$

  where the scalar function $\hat{f}$ is the force f defined on $\hat{\Omega}$.

We refer to [8], for the proof of convergence of the finite element solutions of (25) to the solution of (14).

3 Algorithms for the Discrete Problem

Consider again the finite-dimensional problem (25). It follows from the bilinear form $A(.,.)$ of (15), that C is a symmetric matrix. By considering the following partitions of the matrix C and vectors F and z

$$C = \begin{bmatrix} C_{JJ} & C_{JL} \\ C_{LJ} & C_{LL} \end{bmatrix}, \quad F = \begin{bmatrix} F_J \\ F_L \end{bmatrix}, \quad z = \begin{bmatrix} z_J \\ z_L \end{bmatrix} \tag{31}$$

where $J = J_1 \cup J_2 \cup J_3 \cup J_4$ and $L = L_1 \cup L_2 \cup L_3$, then $z_L = 0$, because of the boundary conditions, and the problem (25) corresponds to a stationary point of the following quadratic program

$$\begin{array}{l} \min \left\{ \frac{1}{2} z_J^T C_{JJ} z_J - F_J^T z_J \right\} \\ \text{subject to} \quad \{ z \in R^{|J|} : \quad z_{J_1} \geq \psi \}. \end{array} \tag{32}$$

Furthermore as $\sqrt{A(.,.)}$, defined in (15), is a norm in $H_0^2(\omega)$, equivalent to the usual norm of $H_0^2(\omega)$, the submatrix C_{JJ} of C is positive definite (PD). This implies that a stationary point $\overline{z}_J$ is unique and is exactly the unique global minimum of the quadratic program (32). Moreover, $\overline{z} = (\overline{z}_J, 0)$ is the unique solution of the discrete problem (25).

It follows from this introduction that the solution of the discrete problem reduces to a strictly convex quadratic problem with $|J| = 4(n_1 - 1)(n_2 - 1) + 2(n_1 + n_2)$ variables and $|J_1| = (n_1 - 1)(n_2 - 1)$ lower bounds, where $n_1 n_2 = m$ is the number of finite elements of the mesh, defined in section 2. It is well-known that this value of m should be large in order the optimal solution of the quadratic problem (32) to be a good approximation for the continuous variational problem (14) under consideration.

There are a number of algorithms for solving strictly convex quadratic programs with simple lower bounds [1, 11]. Among them, the so-called active-set method [11] is a robust technique that searches the unique global minimum by performing changes in a working active-set associated with each iterate. This algorithm possesses finite termination under a nondegeneracy primal assumption and can be implemented for large-scale quadratic programs by exploiting efficient techniques for updating LDL^T decompositions in the sparse case. The code MINOS of the GAMS collection [2] is considered to be an efficient implementation and has been used in the experiences reported in section 4 of the paper. The finiteness of the algorithm is only assured if the working set changes in exactly one element in each iteration. This characteristic of the active-set method is a drawback in practice, as the algorithm may take too many iterations for problems where the initial and the optimal active sets are quite different. This was the reason of our recommendation for another algorithms that are based on different philosophies. As discussed in [6], block pivoting and interior-point algorithms have been shown to process convex quadratic programs with a structure similar to that of the discrete problem (32). In this paper we investigate the efficiency of these algorithms in this case.

Both the interior-point and block pivoting algorithms search the global minimum of the strictly convex quadratic program in a primal-dual way, by solving the following mixed linear complementarity problem (MLCP)

$$\text{Find } (z, \omega) \in R^{|J|} \times R^{|J|}$$

$$w_J = C_{JJ} z_J - F_J \tag{33}$$

$$w_{J_2} = w_{J_3} = w_{J_4} = 0 \tag{34}$$

$$(z_{J_1} - \psi)^T w_{J_1} = 0 \tag{35}$$

$$z_{J_1} \geq \psi, \quad w_{J_1} \geq 0, \tag{36}$$

that corresponds to the Kuhn-Tucker necessary and sufficient optimality conditions [11] of the quadratic program (32). In this section we describe in some detail these two algorithms and their implementations for large scale quadratic programs. In this extent, we introduce the set J_f as the set of indices of the unrestricted variables

$$J_f = J_2 \cup J_3 \cup J_4 = J \setminus J_1. \tag{37}$$

3.1 Interior-Point Algorithm

In order to describe this method, we rewrite the MLCP (33-36) in the following equivalent form

$$\text{Find } (z, \omega) \in R^{|J|} \times R^{|J|}$$

$$C_{JJ} z_J - w_J - F_J = 0 \tag{38}$$

$$(Z_{|J_1|} - \Psi) W_{|J_1|} e = 0 \tag{39}$$

$$w_{J_f} = 0 \tag{40}$$

$$z_{J_1} \geq \psi, \quad w_{J_1} \geq 0, \tag{41}$$

where $e \in R^{|J_1|}$ is a vector of ones, $Z_{|J_1|}$, Ψ and $W_{|J_1|}$ are diagonal matrices whose diagonal elements are z_i, ψ_i and w_i, $i \in J_1$. The interior-point algorithm is an iterative procedure that seeks a solution of the system (38-41) by maintaining (40-41) in each iteration. Furthermore, each iterate (z^k, w^k) satisfies the inequalities (41) strictly. This is one of the special features of this type of method. Another one is the introduction in each iteration of a central path defined by

$$(z_i - \psi_i) w_i = \mu_k, \quad \text{for all} \quad i \in J_1 \tag{42}$$

such that the new iterate should be forced to follow. The so-called central parameter μ_k tends to zero as the algorithm proceeds and is usually defined by

$$\mu_k = \delta \frac{\displaystyle\sum_{i \in J_1} (z_i^k - \psi_i) w_i^k}{|J_1|} \tag{43}$$

with $0 < \delta < 1$ a fixed constant. To define the new iterate, a search direction is first found as the Newton's direction of the following system of nonlinear equations consisting of the linear equation and the central path nonlinear equations

$$\begin{aligned}
&\text{Find } (z, \omega) \in R^{|J|} \times R^{|J|} \\
&C_{JJ} z_J - w_J - F_J = 0 \qquad (44)\\
&(Z_{|J_1|} - \Psi) W_{|J_1|} e = \mu_k e \qquad (45)\\
&w_{J_f} = 0. \qquad (46)
\end{aligned}$$

Since $w_{J_f}^k = 0$, then this search direction (u, v) satisfies

$$\begin{bmatrix} C_{J_1 J_1} & C_{J_1 J_f} & -I_{|J_1|} & 0 \\ C_{J_f J_1} & C_{J_f J_f} & 0 & 0 \\ W_{|J_1|}^k & 0 & Z_{|J_1|}^k - \Psi & 0 \\ 0 & 0 & 0 & I_{|J_f|} \end{bmatrix} \begin{bmatrix} u_{J_1} \\ u_{J_f} \\ v_{J_1} \\ v_{J_f} \end{bmatrix} = \begin{bmatrix} w_{J_1}^k + F_{J_1} - C_{J_1} z^k \\ F_{J_f} - C_{J_f} z^k \\ \mu_k e - (Z_{|J_1|}^k - \Psi) W_{|J_1|}^k e \\ 0 \end{bmatrix} \tag{47}$$

where $I_{|J_1|}$ and $I_{|J_f|}$ are the identity matrices of order $|J_1|$ and $|J_f|$, $W_{|J_1|}^k$ and $Z_{|J_1|}^k$ are diagonal matrices with diagonal elements w_i^k and z_i^k, with $i \in J_1$, respectively, and C_{J_1} and C_{J_f} contain the rows of C corresponding to the sets of indices J_1 and J_f, respectively.

Simple linear algebra manipulations lead to the following expressions for calculating the search direction (u, v):

$$\begin{aligned}
&\begin{bmatrix} C_{J_1 J_1} + (Z_{|J_1|}^k - \Psi)^{-1} W_{|J_1|}^k & C_{J_1 J_f} \\ C_{J_f J_1} & C_{J_f J_f} \end{bmatrix} \begin{bmatrix} u_{J_1} \\ u_{J_f} \end{bmatrix} = \begin{bmatrix} w_{J_1}^k + F_{J_1} - C_{J_1} z^k + \mu_k (Z_{|J_1|}^k - \Psi)^{-1} e \\ F_{J_f} - C_{J_f} z^k \end{bmatrix} \\
&v_i = \frac{\mu_k}{z_i^k - \psi_i} - w_i^k \left(1 + \frac{u_i}{z_i^k - \psi_i}\right), \qquad i \in J_1 \\
&v_j = 0, \qquad j \in J_f.
\end{aligned} \tag{48}$$

After finding the search direction (u, v) a stepsize is computed, to guarantee that a move along this search direction shows some progress to reach the solution of the MLCP. Two measures are important in this process of computing this stepsize, namely the complementarity gap

$$(z_{J_1} - \psi)^T w_{J_1} = \sum_{i \in J_1} (z_i - \psi_i) w_i \tag{49}$$

and the norm of the linear infeasibility

$$\|C_{JJ} z_J - F_J - w_J\|_2, \tag{50}$$

where $\|.\|_2$ is the well-known euclidean norm in $R^{|J|}$.

In theory, the stepsize should be chosen to ensure proximity to the central path and that these two quantities (49-50) reduce in a proportional way [16]. In practice the stepsize α_k satisfies

$$\alpha_k = \min\{1, \beta \alpha_{\max}\} \tag{51}$$

where $0 < \beta < 1$ is a fixed constant and

$$\alpha_{\max} = \min\Big\{ \min\{\frac{z_i^k - \psi_i}{-u_i} : \quad u_i < 0\}, \quad \min\{\frac{w_i^k}{-v_i} : \quad v_i < 0\}, \qquad i \in J_1 \Big\}. \tag{52}$$

To understand why this choice of the stepsize is recommended in practice we consider the following merit function associated with the conditions (38-39)

$$g(z,w) = \|C_{JJ} z_J - w_J - F_J\|_2^2 + \|(Z_{|J_1|} - \Psi) W_{|J_1|} e\|_2^2. \tag{53}$$

Then it is possible to show [15] that if $\alpha_{\max} > 0$ then there exists a $0 < \beta < 1$ such that

$$g(z^k + \alpha_k u, w^k + \alpha_k v) < g(z^k, w^k). \tag{54}$$

Furthermore this result usually holds for the largest possible values of β.

After the computation of the stepsize α_k, the new iterate is given by

$$z^{k+1} = z^k + \alpha_k u, \qquad w^{k+1} = w^k + \alpha_k v \tag{55}$$

and satisfies the conditions (40) and (41) strictly. The algorithm terminates with an approximate solution z^k and w^k satisfying

$$\|C_{JJ} z_J^k - F_J - w_J^k\|_2 \leq \varepsilon_1 \tag{56}$$

and

$$\sum_{i \in J_1} (z_i^k - \psi_i) w_i^k \leq \varepsilon_2. \tag{57}$$

for some tolerances $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$.

It is now possible to state the steps of the algorithm.

Interior-Point Algorithm

1. Let $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ be two positive tolerances, $k = 0$ and z^k, w^k be vectors satisfying $w_{J_f}^k = 0$, $w_{J_1}^k > 0$ and $z_{J_1}^k > \psi$.
2. For $k = 0, 1, ...$
   - Compute μ_k by (43).
   - Find (u, v) by (48).
   - Compute α_k by (51) and (52) with $\beta = 0.99995$.
   - Update $z^{k+1} = z^k + \alpha_k u$, $w^{k+1} = w^k + \alpha_k v$.
   - Terminate if (56) and (57) holds.

It follows from the description of the algorithm that the main effort of each iteration is concerned with the solution of the linear system (48). Since C_{JJ} is a symmetric definite matrix the same holds for all the matrices $\overline{C}_{JJ}^k$ of the system (48). Furthermore C_{JJ} and $\overline{C}_{JJ}^k$ have the same sparsity pattern. These two properties of the working matrix $\overline{C}_{JJ}^k$ are quite important for the design of an efficient implementation of the algorithm capable of processing large-scale quadratic programs of the form (32). In order to briefly describe this implementation, we recall that the solution of a large-scale linear system with a symmetric positive definite matrix C_{JJ} or $\overline{C}_{JJ}^k$ consists of the following phases [5, 7] :

- Analyse, where the so-called minimum degree algorithm or one of its variants is applied to find a suitable ordering for the rows and columns of C_{JJ} (and $\overline{C}_{JJ}^k$) that leads into small fill-in during the factorization.

- Factorization, which computes the LDL^T decomposition of a principal permutation of the matrix C_{JJ} (or $\overline{C}^k_{JJ}$) according to the ordering achieved in the Analyse phase.

- Solution, which computes the solution of the linear system by processing the easy systems with the matrices L, D and L^T.

It is important to add that these three phases are performed separately and Analyse is the most costly procedure, as it usually takes 70% of the effort required for finding the solution of the system. Since $\overline{C}^k_{JJ}$ has the same sparsity partten of C_{JJ}, then this Analyse phase is only processed once during the whole application of the interior-point method. This substantial saving in the computation makes the effort of each iteration of the interior-point quite small and enables the solution of quite large convex quadratic programs. The implementation of the interior-point method exploits the ideas presented above and uses the subroutines MA27 from Harwell collection [5] for these Analyse, Factorization and Solution phases.

As is discussed in the last section of this paper, machine memory limitations may unable the solution of quite large, strictly convex quadratic programs by using this type of implementation. In this case a preconditioned conjugate-gradient algorithm [13] should be used to find the search direction in each iteration. An implementation of an interior-point method for a linear network problem based in this latter methodology has been described and fully tested in [14]. An important feature of this implementation is that the tolerances used in the stopping criterium of the conjugate-gradient method can be chosen to be monotone decreasing with the iteration count of the interior-point algorithm. This implies substantial savings in the process of finding the search direction for each iteration of the procedure. We believe that an implementation of the same type with different preconditionings can process efficiently all these large strictly convex quadratic programs.

As is discussed by many authors [11, 16] the so-called predictor-corrector interior-point algorithm is a valid alternative technique for processing the strictly convex quadratic program (32). Each iteration of this algorithm essentially differs from one of the previous interior-point method on the existence of a predictor step that finds a new point $(\overline{z}^k, \overline{w}^k)$ that is used to compute the central parameter μ_k. This implies that each iteration requires the solution of two linear systems with the same matrix $\overline{C}^k_{JJ}$. It is interesting to note that this only duplicates the computation in the Solution phase, which is almost meaningless when the implementation is based on direct solvers. However, the situation is different when an iterative solver is used to find the search direction. Furthermore, as is shown in the last section of this paper, the number of iterations of the simple interior-point algorithm is constantly small even for very large values of $|J|$. Finally, the descent property for the merit function of (53) does not hold any longer for the predictor-corrector algorithm. These considerations have lead to our decision of using the simple interior-point algorithm to process the strictly convex quadratic programming discrete problem.

3.2 A Block Principal Pivoting Algorithm

Principal pivoting algorithms are direct methods that find in a finite number of iterations the unique global minimum of the strictly convex quadratic program (32) by processing its equivalent MLCP (33-36). These algorithms use in each iteration a complementary basis solution of the MLCP. If R and S are subsets of J such that $R \cup S = J$, $R \cap S = \emptyset$ and C_{RR} is nonsingular, such a solution satisfies $\overline{z}_i = \psi_i$, for all $i \in S$ and $\overline{w}_i = 0$, for all $i \in R$, and, this implies that the remaining components are uniquely given by

$$\begin{array}{l} C_{RR}\overline{z}_R = F_R - C_{RS}\psi_S \\ \overline{w}_S = -F_S + C_{SR}\overline{z}_R + C_{SS}\psi_S. \end{array} \tag{58}$$

It is important to add that if C_{JJ} is a symmetric positive definite matrix there is a complementary basic solution for each possible partition $\{R, S\}$ of J. Since $w_{J_f} = 0$ in any solution of the MLCP (33-36), then we force J_f to be always included in the set R of any complementary basic solution that is used by the algorithm. If such a solution $\overline{z}, \overline{w}$ satisfies

$$\overline{z}_{R\cap J_1} \geq \psi_{R\cap J_1} \quad \text{and} \quad \overline{w}_S \geq 0 \tag{59}$$

then it is said to be feasible and is a solution of the MLCP (33-36). Otherwise, the so-called set of infeasibilities is considered :

$$H = \{i \in R \cap J_1 : \quad \overline{z}_i < \psi_i\} \cup \{i \in S : \quad \overline{w}_i < 0\}. \tag{60}$$

The number of elements of this set H is called the infeasibility count of the complementary basic solution. We note that $0 \leq |H| \leq |J_1|$ and $|H| = 0$ if and only if $(\overline{z}_R, \psi_S)$ is the unique solution of the MLCP.

Each iteration of a principal pivoting algorithm simply consists of replacing the sets R and S associated with a complementary basic infeasible solution ($H \neq \emptyset$) to another sets $\overline{R}$ and $\overline{S}$ corresponding to another solution of the same type. This is done by using the following formulas

$$\begin{aligned} \overline{R} &= R \setminus (R \cap H_1) \cup (S \cap H_1) \\ \overline{S} &= J \setminus \overline{R} \end{aligned} \tag{61}$$

where $H_1 \subseteq H$. The algorithms differ on the choice of the set H_1. As is discussed in [6], the use of

$$H_1 = \{ \min\{i \in H\} \} \tag{62}$$

in each iteration guarantees finite termination to the algorithm. However, as for the active-set methods, these modifications of one element usually lead to too many iterations for large-scale quadratic programs, where the initial and final partitions $\{R, S\}$ are quite different. On the other hand, the all-change modification $H_1 = H$ usually leads to small number of iterations in practice [6]. However, there is no theoretical guarantee that an algorithm solely based on these latter changes possesses finite termination. As is discussed in [6, 9], it is possible to design a principal pivoting method algorithm that combines these two features presented before. The resulting method performs all-changes modifications (61) with $H_1 = H$ in general, and one-element changes (62) are only included for assuring finite termination. The switch from one form of iterations to the other one, is done by controlling the infeasibility count, that is, the number of elements $|H|$ of the set H given by (60).

The steps of the algorithm are presented below.

Block Principal Pivoting Algorithm

1. Let $R = J_f$, $S = J \setminus R$, $p > 0$, $ninf = |J| + 1$ and $nit = 1$.
2. Compute $\overline{z}_R$ and $\overline{w}_S$ by (58) and the infeasibility set H by (60). Let $|H|$ be the number of elements of H. Then
   - If $|H| = 0$, terminate with $\overline{z} = (\overline{z}_R, \psi_S)$ the unique solution of the MLCP.
   - If $ninf > |H|$, set $ninf = |H|$ and $nit = 1$. Go to 3.
   - If $ninf \leq |H|$ and $nit \leq p$, go to 3. (if $nit = 1$ set $\overline{R} = R$ and $\overline{H} = H$).
   - If $ninf \leq |H|$ and $nit = p + 1$, go to 4, with $R = \overline{R}$ and $H = \overline{H}$.
3. Set $R = R \setminus (R \cap H) \cup (S \cap H)$, $S = J \setminus R$, $nit = nit + 1$ and go to 2.
4. Let $t = \min\{i \in H\}$. Set

$$R = \begin{cases} R \setminus \{t\}, & \text{if } t \in R \\ R \cup \{t\}, & \text{if } t \in S \end{cases} \tag{63}$$

   and $S = J \setminus R$ and go to 2.

It follows from the description of the steps of the algorithm that the integer constant p plays an important role on the efficiency of the algorithm. This value represents the maximum number of block iterations ($H_1 = H$) that are allowed to be performed without an improvement of the infeasibility count. It is obvious that this value should be small. However, too small values for p may lead to the performance of one-element modifications too often with an increase on the number of iterations.

Extensive computational experience reported in [6] has shown that $p = 10$ is usually a good choice in practice.

The algorithm can also be implemented according to a scheme similar to the interior-point method. As before, an Analyse phase is performed only once for the matrix C_{JJ}. In each iteration, the matrix C_{RR} is constructed from C_{JJ}. Once more the rows and columns with indices $i \in R$ are considered in the ordering established in the Analyse phase. This construction may be time consuming if the set R is quite large. After this step is finished, the Factorization and Solution phases are performed in order to get the vector $\overline{z}_R$. As before the subroutines MA27 are used to perform all the tasks.

The vector $\overline{w}_S$ is computed as follows

$$\overline{w}_i = -F_i + C_{i.} \begin{bmatrix} \overline{z}_R \\ \psi_S \end{bmatrix}, \quad i \in S \tag{64}$$

where $C_{i.}$ represents the row i of the matrix C_{JJ}.

As is discussed in the last section of this paper, this type of implementation is quite efficient for processing reasonable large strictly convex quadratic programs. It is also possible to derive an implementation of this algorithm based on the preconditioned conjugate-gradient algorithm for processing the linear systems $C_{RR}\overline{z}_R = F_R$ that are required in each iteration of the principal pivoting algorithm. However, the computation of the infeasibilities according to (60) prevents the use of variable monotone decreasing tolerances and this makes an iterative based implementation less attractive for the block pivoting method.

4 Computational Experience

In this section we report some numerical experiments with the algorithms on the solutions of discrete problems associated with a thin, elastic and clamped plate, whose middle plane is a square with side length $l = 100mm$ and thickness $t = 2mm$. For this plate geometry, several numerical tests have been performed with different obstacles, materials and forces. The following obstacles have been considered:

$$\begin{aligned} &\psi_1(x,y) = -2, \\ &\psi_2(x,y) = -0.1 - (\tfrac{x}{50} - 1)^2(\tfrac{y}{100} - 0.5)^2, \\ &\psi_3(x,y) = -0.5(\tfrac{x}{50} - 1)^2 - 0.02(\tfrac{x}{50} - 1) - \tfrac{0.5}{1+30(\frac{x}{50}-1)^2} \end{aligned} \tag{65}$$

for x, y in $[0, 100]$. The obstacle ψ_1 represents a plane and the obstacles ψ_2 and ψ_3 are surfaces, whose three dimensional-plots are displayed in figures 1 and 2.

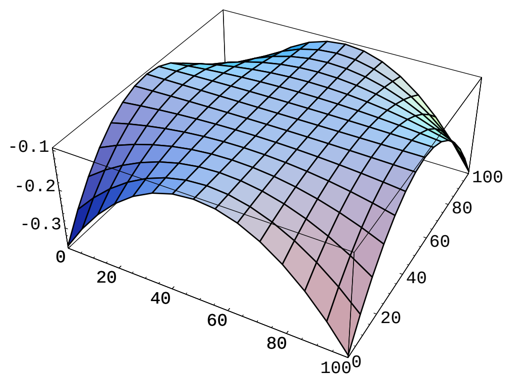

Figure 1: Obstacle ψ_2

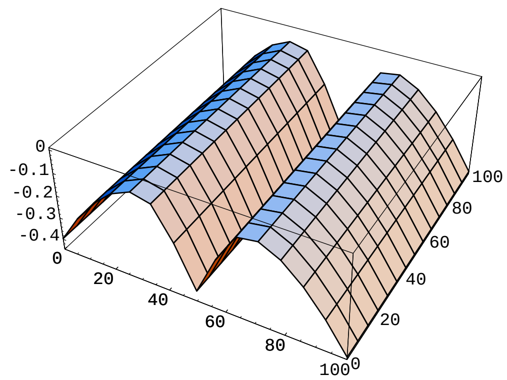

Figure 2: Obstacles ψ_3

We assume that the forces acting on the plate are constants with intensities

$$f_1(x,y) = -1, \quad f_2(x,y) = -5, \quad f_3(x,y) = -10, \tag{66}$$

(in the unit kg/mm^2). Moreover the material of the plate may be steel, bronze or lead, with Young's modulus E and Poisson's ratio ν given below

Material	E (kg/mm^2)	ν
Steel	21×10^3	0.28
Bronze	11×10^3	0.31
Lead	1.8×10^3	0.44

As already mentioned in section 2, the discretization of the infinite dimensional problems is based in the finite element method. We have chosen the Bogner-Fox-Schmit rectangle with 16 degrees of freedom in this approximation. The square $[0, 100]^2$ is discretized successively by 4×4, 10×10, 20×20 and 30×30 finite elements in our first experiments. Refined meshes for this approximation with 50×50, 70×70 and 90×90 finite elements are also considered in the second experiences.

All the experiences have been performed on a gateway G520 (256Mb RAM, Pentium II processor 350Mhz).

It follows from the description of the steps of the interior-point method that there is the need of choosing the initial point (z^0, w^0) and the tolerances ε_1, ε_2 that are required in the stopping criterium (56-57). After some experiments and based on previous computational work with similar problems [6], we have made the following choices

$$\begin{array}{ll} z_i^0 = 0, & i \in J_1 \cup J_f \\ w_i^0 = \max\{10, -F_i\}, & i \in J_1 \\ w_i^0 = 0, & i \in J_f \end{array} \tag{67}$$

and $\varepsilon_1 = 10^{-10}$, $\varepsilon_2 = 10^{-8}$. These values usually guarantee at least seven decimal digits of accuracy in the unique solution of the MLCP found by the interior-point algorithm.

Tolerances are also needed in the block pivoting method for choosing the set of infeasibilities H. In our implementation of this algorithm the following definition of H has been used

$$H = \{i \in R: \quad z_i < \psi_i - \varepsilon\} \cup \{i \in S: \quad w_i < -\varepsilon\} \tag{68}$$

where $\varepsilon = 10^{-6}$. A scaled free version of this definition should be probably more recommended, particularly for bad scaling MLCPs that arise when the mesh size is quite small. However, in our experiences the definiton (68) has proven to work quite well, as the accuracy of the solution of the MLCP found by this method is usually the same of that computed by the interior-point algorithm.

The tables 1, 2, 3, 4, 5 and 6 report the results of solving the test problems by the interior-point method (IP) and the block pivoting (BP) algorithms. In these tables, *ele* represents the number of finite elements that have been considered in the mesh, $|J|$ the corresponding number of variables of the problem (32) and $|J_1|$ the number of lower bounds. Furthermore *cn* denotes the number of variables z_i, $i \in J_1$, that attain the lower bound ψ_i at the optimal solution of the quadratic program, that is, the number of nodes at which there is contact between the plate and the obstacle. Finally, the performance of each one of the algorithms has been reported in terms of the number of iterations (it) and CPU time in seconds (CPU).

ele	$\|J\|$	Algorithms	force $f_1 = -1$			force $f_2 = -5$			force $f_3 = -10$		
			cn	it	CPU	cn	it	CPU	cn	it	CPU
16	52	IP	5	23	$1.75E-2$	9	23	$1.75E-2$	9	24	$1.95E-2$
		PATH		7	0.03		11	0.03		11	0.03
		BP		2	$0.01E-2$		1	$0.01E-2$		1	$0.01E-2$
		MINOS		95	0.30		78	0.22		72	0.31
100	364	IP	9	13	0.17	45	14	0.18	49	13	0.17
		PATH		11	0.83		128	7.91		52	0.98
		BP		8	0.17		3	0.05		2	0.03
		MINOS		550	2.98		551	2.96		541	3.01
400	1524	IP	33	12	1.73	121	12	1.74	165	12	1.75
		PATH		52	21.21		157	42.01		201	43.35
		BP		11	2.82		8	1.74		6	1.20
		MINOS		1992	35.36		1968	35.46		1955	35.37

Table 1: Plate of Steel and Obstacle ψ_1

In order to gain a better indication of the efficiency of these methodologies, we have also solved some problems by the active-set MINOS (version 5) and the path-following PATH (version 4.4a) codes that are available in the GAMS collection. It is important to add that for the PATH algorithm *it* represents the sum of the so-called inner and crash iterations. The results are displayed in the following tables 1, 2 and 3.

ele	$\|J\|$	Algorithms	force $f_1 = -1$			force $f_2 = -5$			force $f_3 = -10$		
			cn	it	CPU	cn	it	CPU	cn	it	CPU
16	52	IP	9	20	$1.66E-2$	9	21	$1.75E-2$	9	22	$1.75E-2$
		PATH		11	0.03		11	0.03		11	0.03
		BP		1	$0.01E-2$		1	$0.01E-2$		1	$0.01E-2$
		MINOS		74	0.23		73	0.28		75	0.26
100	364	IP	42	11	0.15	75	12	0.16	81	12	0.16
		PATH		44	0.98		77	1.23		77	1.22
		BP		4	0.07		2	0.03		1	0.01
		MINOS		522	2.90		487	2.90		412	2.94
400	1524	IP	142	11	1.52	230	11	1.55	283	12	1.62
		PATH		192	48.27		279	55.28		333	62.14
		BP		7	1.51		5	0.93		3	0.50
		MINOS		1952	35.32		1943	35.41		1946	35.37

Table 2: Plate of Bronze and Obstacle ψ_3

It follows from the results displayed in tables 1, 2 and 3 that both the interior-point and block pivoting algorithms are quite efficient for the solution of all the test problems. Moreover, these two algorithms have proven to perform much better in terms of iterations and CPU time than the other two GAMS general purpose algorithms, and the gap increases much with an increase on the number of elements. It is interesting to note that the number of iterations of the interior-point method does not indicate an increase with the number of elements, and consequently the dimension $|J|$ of the problem. On the other hand, there is a slight increase on the number of iterations for the block pivoting method. These conclusions become more evident with the results displayed in the table 4.

ele	$\|J\|$	Algorithms	force $f_1=-1$			force $f_2=-5$			force $f_3=-10$		
			cn	it	CPU	cn	it	CPU	cn	it	CPU
16	52	IP	9	19	$1.56E-2$	9	20	$1.66E-2$	9	21	$1.66E-2$
		PATH		11	0.02		11	0.02		11	0.02
		BP		1	$0.01E-2$		1	$0.01E-2$		1	$0.01E-2$
		MINOS		65	0.31		58	0.28		59	0.25
100	364	IP	81	10	0.13	81	11	0.14	81	11	0.15
		PATH		84	1.12		84	1.11		84	0.26
		BP		1	0.01		1	0.01		1	0.01
		MINOS		373	2.96		361	2.96		362	2.85
400	1524	IP	285	9	1.40	341	11	1.61	357	9	1.28
		PATH		320	57.09		376	60.72		392	61.29
		BP		3	0.50		3	0.47		2	0.30
		MINOS		1884	35.44		1750	35.46		1522	35.34

Table 3: Plate of Lead and Obstacle ψ_2

Table 4 shows the performance of the interior-point and block pivoting algorithms for a particular choice of the material, force and obstacle, when the number of elements increases. The importance of the increase on the number of elements is the accuracy of the approximate solution as illustrated in figure 3.

Ele	$\|J\|$	$\|J_1\|$	Alg.	force $f_1=-1$			force $f_2=-5$			force $f_3=-10$		
				cn	it	CPU	cn	it	CPU	cn	it	CPU
16	52	9	IP	9	20	$1.66E-2$	9	21	$1.66E-2$	9	22	$1.85E-2$
			BP		1	$0.01E-2$		1	$0.01E-2$		1	$0.01E-2$
100	364	81	IP	38	12	0.16	61	13	0.17	75	13	0.17
			BP		4	0.07		3	0.05		2	0.03
400	1524	361	IP	89	13	1.77	204	12	1.62	230	11	1.47
			BP		9	2.19		5	0.97		5	0.92
900	3484	841	IP	135	16	11.38	406	12	8.31	518	14	9.69
			BP		14	18.23		8	7.72		9	7.89
2500	9804	2401	IP	362	19	104.92	1014	16	86.43	1248	17	91.38
			BP		22	238.49		15	133.63		13	107.22
4900	19324	4761	IP	666	21	336.10	1920	18	285.74	2370	17	268.23
			BP		31	1236.72		22	711.62		18	540.48
8100	32044	7921	IP	1037	24	1127.97	3039	20	933.32	3850	20	933.21
			BP		43	2679.86		35	2180.55		29	1815.73

Table 4: Plate of Steel and Obstacle ψ_3

So we recommend the use of the interior-point algorithm when the number of elements used to construct the discrete problem is quite large. It is important to add that memory machine limitations unable the solution of a discrete problem with even larger number of elements. As stated in section 3, an implementation based on a preconditioned conjugate-gradient method should be employed to process the discrete model in these cases.

In the next experience we have tested the influence of the intensity of the force on the contact with the obstacle. To do this, we have fixed a particular material (bronze), the number of finite elements (ele = 400) and the obstacle ψ_1. Then for these particular instances, we have processed three convex quadratic programs that differ on the intensity of the force, namely $f_1=-1$, $f_2=-5$ and $f_3=-10$.

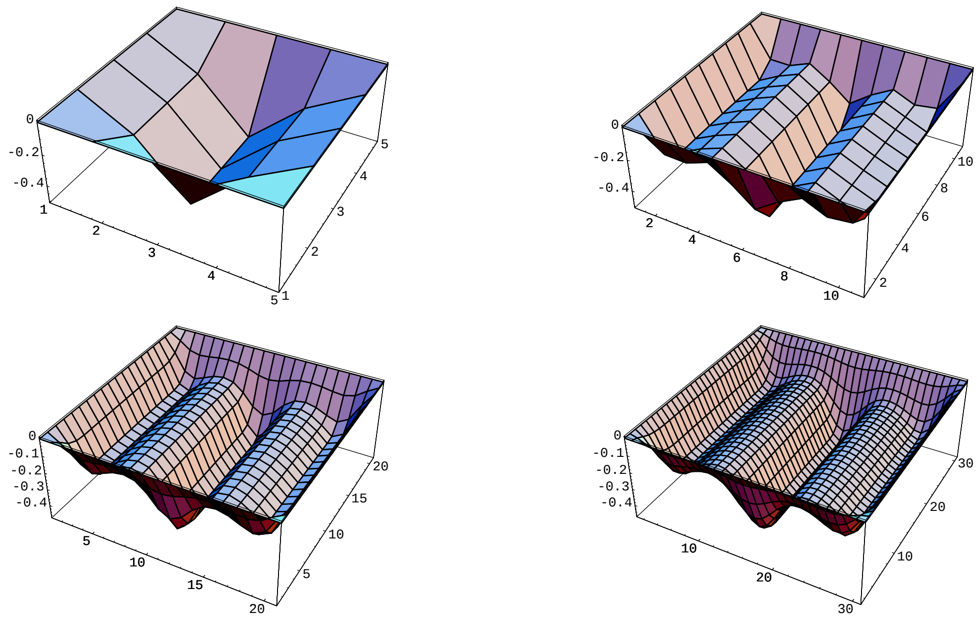

Figure 3: Solution for plate of steel, with obstacle ψ_3, and 16, 100, 400 and 900 finite elements.

The results are displayed in figures 4-6 and table 5, and confirm the physical belief that an increase of the intensity of the force augments the contact.

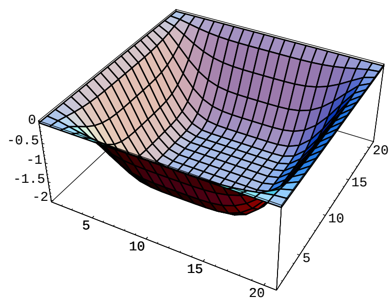

Figure 4: Solution for force f_1

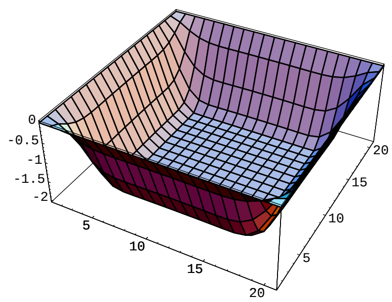

Figure 5: Solution for force f_2

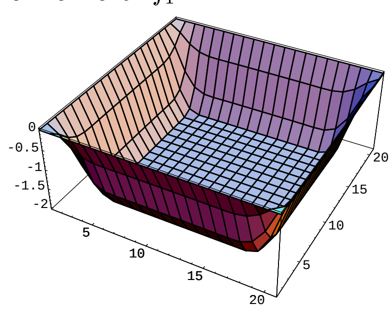

Figure 6: Solution for force f_3

Algorithms	force $f_1 = -1$			force $f_2 = -5$			force $f_3 = -10$		
	cn	it	CPU	cn	it	CPU	cn	it	CPU
IP	77	12	1.68	165	12	1.78	221	12	1.78
BP		9	2.15		6	1.20		5	0.94

Table 5: Plate of Bronze, Obstacle ψ_1, $|J| = 1524$

In the last experience, we have tested the influence of the material on the contact with the obstacle. To do this, we have fixed this time the intensity of the force $f_1 = -1$, the number of elements (ele=400) and the obstacle ψ_2. We have considered three instances that differ from each other on the choice of the material (steel, bronze and lead). The results are illustrated in figures 7-9 and table 6, and confirm that the contact increases with an augment of the Young's modulus.

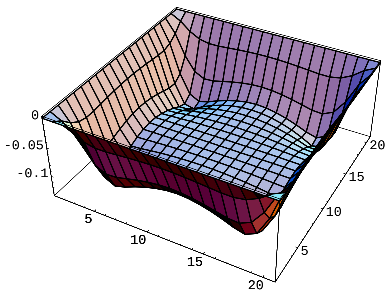

Figure 7: plate of steel

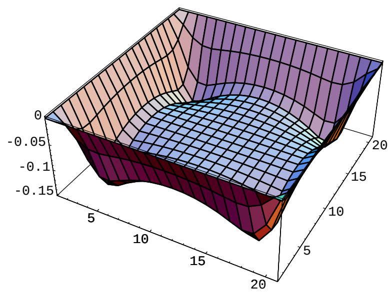

Figure 8: plate of bronze

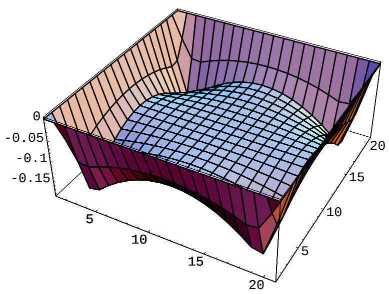

Figure 9: plate of lead

Material	cn	Interior-Point		Block Pivoting	
		it	CPU	it	CPU
Steel	181	11	1.56	7	1.40
Bronze	221	10	1.37	5	0.92
Lead	285	9	1.40	3	0.50

Table 6: Force $f_1 = -1$, Obstacle ψ_2, $|J| = 1524$

5 Conclusion

In this paper we have introduced an obstacle plate model. The linear version of this problem has been studied. The discrete problem that arises by using an appropriate finite element discretization reduces to a strictly convex quadratic program that has a unique global solution. An interior-point and a block principal pivoting primal-dual algorithms have been shown to be quite efficient to process this quadratic program and perform much better than two general purpose codes MINOS and PATH of the GAMS collection. Unfortunately the two primal-dual techniques discussed in this paper can no longer be useful for processing the nonlinear version of this model (corresponding to a geometrically nonlinear plate), as the resulting discrete problem can be shown to be a nonconvex nonlinear optimization problem [10], that is quite difficult to tackle. Some experiments with the PATH algorithm have shown that this latter method has been able to process some special instances of this model, but it fails in general. The study of this discrete nonlinear problem is a subject of our current investigation.

References

[1] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Massachusetts (1995).

[2] A. Brook, D. Kendrich, A. Meeraus, *GAMS, User's Guide, Release 2.25*, Scientific Press, Massachusetts(1992).

[3] P.G. Ciarlet, *Basic Error Estimates for Elliptic Problems*, Handbook of Numerical Analysis, P.G. Ciarlet and J.L. Lions (eds), Vol.II, North-Holland (1991).

[4] S.P. Dirkse, M.C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optimization Methods and Software, **5**, 123-156 (1995).

[5] I. Duff, A. Erisman and J. Reid, *Direct Methods for Sparse Matrices*, Claredon Press, Oxford (1986).

[6] L. Fernandes, J. Júdice and J. Patrício, *An investigation of interior-point and block pivoting algorithms for large scale symmetric monotone linear complementarity problems*, Computational Optimization and Applications **5**, 49-77 (1996).

[7] J.A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, (1981).

[8] J. Haslinger, I. Hlaváček and J. Nečas, *Numerical Methods for Unilateral Problems in Solid Mechanics*, *Handbook of Numerical Analysis, Vol. IV*, Edited by P.G. Ciarlet and J.L. Lions, Elsevier Science B.V., Amsterdam (1996).

[9] J. Júdice and F.M. Pires, *A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems*, Computers and Operations Research **21**, 587-596 (1994).

[10] N. Kikuchi and J.T. Oden, *Contact Problems in Elasticity: a Study of Variational Inequalities and Finite Elements*, SIAM, Philadelphia (1988).

[11] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, (1999).

[12] K. Ohtake, J.T. Oden and N. Kikuchi, *Analysis of certain unilateral problems in von Karman plate theory by a penalty method, Part 2*, Computer Methods in Applied Mechanics Engineering **24**, 317-337 (1980).

[13] J. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.

[14] L. Portugal, M. Resende, G. Veiga and J. Júdice, *A truncated primal-infeasible dual-feasible interior-point network flow method*, Networks **35**, 91-108 (2000).

[15] E. Simantiraki and D. Shanno, *An infeasible interior-point method for linear complementarity problems*, Technical Report RR7-95, New Jersey, (1995).

[16] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia (1997).