Structural Engineering and Mechanics, Vol. 6, No. 8 (1998) 857-870
DOI: http://dx.doi.org/10.12989/sem.1998.6.8.857

Nonlinear programming approach for a class of inverse problems in elastoplasticity

M.C. Ferris †

Computer Sciences Department, University of Wisconsin, Madison, Wisconsin 53706, U.S.A.

F. Tin-Loi ‡

School of Civil and Environmental Engineering,
University of New South Wales, Sydney 2052, Australia

Abstract. This paper deals with a special class of inverse problems in discrete structural plasticity involving the identification of elastic limits and hardening moduli on the basis of information on displacements. The governing equations lead naturally to a special and challenging optimization problem known as a Mathematical Program with Equilibrium Constraints (MPEC), a key feature of which is the orthogonality of two sign-constrained vectors or so-called "complementarity" condition. We investigate numerically the application of two simple algorithms, both based on the use of the general purpose nonlinear programming code CONOPT accessed via the GAMS modeling language, for solving the suitably reformulated problem. Application is illustrated by means of two numerical examples.

Key words: complementarity; elastoplasticity; inverse problems; mathematical programming; structural identification.

1. Introduction

The area of nonsmooth mechanics has become, as a result of vigorous research over the last decade or so (see e.g., Duvaut and Lions 1976, Panagiotopoulos 1985, Moreau *et al.* 1988, Moreau and Panagiotopoulos 1988, and the numerous references contained therein), an important and well-established branch of mechanics in its own right. A distinguishing feature of problems in the area is the fact that the state as a function of the design or control variable is not everywhere differentiable. In most instances, this is caused by the presence of variational inequalities or, equivalently, complementarity conditions.

The work presented in this paper focuses on one such nonsmooth mechanics problem. It concerns a particular class of structural identification problems for which, in its simplest form, it is required to identify the material yield limits and/or hardening parameters presuming the availability of some measured information on the displacement response to a known loading condition. The primary motivation for studying this problem is that the underlying

† Professor
‡ Professor

methodology appears to be an elegant and computationally viable approach for identifying key fracture parameters for quasibrittle materials - a subject of current, intense research interest. For instance, recent work by Bolzon *et al.* (1997) and Bolzon and Maier (1998) describe the formulation related to the identification of the fracture parameters associated with the cohesive crack model.

The generic problem in point was in fact introduced in the early 80s by Maier (Maier 1981, Maier *et al.* 1982, Nappi 1982) with reference to simple discretized bar structures that, in fact, characterize all essential features of the fracture identification problem. We similarly consider this formally simpler, albeit conceptually almost identical, class of conventional discretized elastoplastic structures.

The organization of this paper is as follows. In the next section, we review the state or analysis problem assuming, without undue loss of generality, piecewise linearized holonomic (reversible or nonlinear elastic) plasticity. In essence, the displacement response is sought for a structure of known material properties. As is well-known (Maier 1970), the governing relations lead to a Linear Complementarity Problem (LCP) which involves, as a key mathematical structure, the orthogonality of two sign-constrained vectors. The inverse problem is considered next in Section 3. Briefly, we are now given the displacement response and we wish to obtain some important material properties, in particular both yield limits and hardening parameters in the present case. It is shown that this identification problem can be formulated naturally as a special optimization problem involving the minimization of some error function subject to complementarity and other constraints. This problem falls under the general class of problems commonly known as Mathematical Programs with Equilibrium Constraints, MPECs for short (Luo *et al.* 1996). In Section 4, we present two simple algorithms, both based on the use of a standard nonlinear programming code, for its numerical solution. The first algorithm adopts the classical penalty approach in which the complementarity term, treated as part of the objective function, is driven to zero by means of an associated penalty parameter. The second replaces the complementarity conditions by a nonsmooth equation that is parametrically modified to generate differentiable constraints suitable for the nonlinear code. The parameter is iteratively updated until a complementary solution is found. Both algorithms are implemented via the modeling language GAMS (Brooke *et al.* 1992). Application of these algorithms is illustrated in Section 5 using two examples, the first concerning an 11-bar truss (Nappi 1982) and the second a hypothetical structural identification problem concerning an elastoplastic beam on elastoplastic foundation (Maier *et al.* 1982). We conclude with some brief remarks and recommendations for future work in Section 6.

2. The state or analysis problem

Consider a suitably space-discretized structural system. Under a holonomy assumption, which is reasonable in view of the assumed proportionally applied loads, we formulate the single step analysis problem simply by collecting and manipulating the relations describing the three key ingredients of the structural behavior: statics, kinematics and constitutive laws. We further assume a small deformation theory, inviscid behavior, and adopt piecewise linearized yield surfaces. The well-known (Maier 1970) governing relations for the whole structure are

$$F = C^T Q \tag{1}$$

$$q = Cu \tag{2}$$

$$q = \varepsilon + p \tag{3}$$

$$Q = S\varepsilon \tag{4}$$

$$p = N\lambda \tag{5}$$

$$\phi = N^T Q - H\lambda - R \leq 0,\ \lambda \geq 0,\ \phi^T \lambda = 0 \tag{6}$$

As is typical, vector and matrix quantities represent the unassembled contributions of corresponding elemental entities as concatenated vectors and block diagonal matrices, respectively. For a structure with d degrees of freedom, m member generalized quantities and y yield functions, equilibrium between the nodal loads $F \in \Re^d$ and the natural generalized stresses $Q \in \Re^m$ is expressed by (1) through the compatibility matrix $C \in \Re^{m \times d}$. Eq. (2) represents linear compatibility of strains $q \in \Re^m$ with the nodal displacements $u \in \Re^d$. Relations (3)-(6) embody the holonomic constitutive laws: additivity of elastic $\varepsilon \in \Re^m$ and plastic $p \in \Re^m$ strains in (3); linear elasticity in (4), where $S \in \Re^{m \times m}$ is an elastic matrix of unassembled positive definite element stiffnesses; plastic strains p defined in (5) by an associated flow rule and expressed as functions of the plastic multipliers $\lambda \in \Re^y$ through the constant matrix of outward normals $N \in \Re^{m \times y}$ to the yield surface; piecewise linear yield functions $\phi(Q, \lambda)$: $\Re^{m+y} \rightarrow \Re^y$ in (6) which accommodate, through $H \in \Re^{y \times y}$ a class of hardening models with known yield limits $R \in \Re^y$ and finally, a complementarity relationship in (6) between the sign-constrained quantities ϕ and λ. The latter condition implies that plastic flow can only occur ($\lambda > 0$) when the yield surface is activated (ϕ=0), and that there are no plastic strains (λ=0) when the state is elastic ($\phi < 0$). We also restrict ourselves to the practically significant case when no hardening interaction is present, namely H is a diagonal matrix with all off-diagonal terms equal to zero.

For further details of the above framework and its specific application to finite element formulation in generalized variables, we refer the interested reader to the numerous works of the Milan group, e.g., Corradi (1978, 1983), Comi *et al.* (1992).

A popular approach (Maier 1970) is to simplify relation set (1)-(6) by expressing the problem in λ variables only. This leads to a conventional LCP which, for hardening matrices H, can be solved by Lemke's algorithm (Cottle *et al.* 1992) to give a unique solution.

In our work, the modeling is carried out through the powerful GAMS (an acronym for General Algebraic Modeling System) language (Brooke *et al.* 1992) and we therefore prefer to leave the formulation in Q, u, λ variables. Without entering into details of the well-known advantages provided by modeling languages, it suffices to mention that GAMS is a high-level modeling language especially designed to facilitate construction, solution and maintenance of large and complicated mathematical programming models. It additionally provides simplicity and compactness of model construction, a variety of tested industry standard mathematical programming solvers, and important capabilities such as an internal efficient sparse data representation and automatic differentiation. In our computational testing we adopted the mixed complementarity solver PATH (Dirkse and Ferris 1995) for generating pseudo-experimental data from the solution of a state problem, and the nonlinear programming solver CONOPT (Drud 1994) for the identification problem.

After some obvious substitutions, the state problem then becomes

$$F = C^T Q \tag{7}$$

$$Q = SCu - SN\lambda \tag{8}$$

$$w = -N^T Q + H\lambda + R \geq 0, \ \lambda \geq 0, \ w^T \lambda = 0 \tag{9}$$

where, as is standard, we use nonnegative vectors in the complementarity condition (9).

3. The identification problem

The inverse problem we wish to investigate will now be briefly described. It is assumed that some displacements $u_m \in \Re^k$ are known (measured) deterministic quantities, whereas the vector of yield limits R and the diagonal hardening matrix H are unknown, except possibly for an *a priori* grouping due to knowledge that certain structural members are identical.

If we now denote by $u_c \in \Re^k$ the subset of displacement values corresponding to u_m that would be obtained from the structural model for the same loading, then a natural measure of the discrepancy (or error) ω between the measured and the theoretical displacements is provided by a suitable norm of the difference between u_m and u_c. The identification problem obviously requires the *global* minimum of ω subject to (7)-(9), any prior knowledge that certain members are identical, and any known bounds on R and H. When measurements are perfect, ω=0 so that it is possible to determine if a global optimum (albeit not necessarily leading to a unique set of desired parameters) has been reached.

In their seminal work, Maier *et al.* (1982) considered the case of unknown R only and proposed essentially an enumerative scheme in their attempt to obtain a global minimum. Nappi (1982) later extended that work to unknown H as well, however, under the assumption of perfect measurements. This latter assumption enabled a simple solution scheme to be adopted, namely minimization, using a nonlinear programming solver, of the sum of error and complementarity terms. Recent work by Jiang *et al.* (1997) demonstrated the viability of using a piecewise sequential quadratic programming approach for the unknown R only case.

For simplicity, assume that only one test to measure displacements is carried out and only one load level is applied; extension to the case of multiple tests and several load levels is straightforward (see Maier *et al.* 1982 for details). The identification problem can then be formally stated as the following constrained optimization problem:

$$\min \ \omega = \| u_m - B_1 u \| \tag{10}$$

subject to:

$$F = C^T Q \tag{11}$$

$$Q = SCu - SN\lambda \tag{12}$$

$$w = -N^T Q + diag(B_2 h)\lambda + B_3 r \geq 0, \ \lambda \geq 0, \ w^T \lambda = 0 \tag{13}$$

$$h_l \leq h \leq h_u, \ \ r_l \leq r \leq r_u \tag{14}$$

where $u_c = B_1 u$, $H = diag(B_2 h)$, $R = B_3 r$, with the (n_1+n_2) unknown parameters to be identified collected in the vectors $h \in \Re^{n_1}$ and $r \in \Re^{n_2}$; $B_1 \in \Re^{k \times d}$, $B_2 \in \Re^{y \times n_1}$, $B_3 \in \Re^{y \times n_2}$ are obvious selective Boolean matrices. The lower (subscript l) and upper (subscript u) bounds on h and r are given in (14); these are typically estimable from the particular application.

Optimization problem (10)-(14) is a special case of the so-called Mathematical Program with Equilibrium Constraints or MPEC (Luo *et al.* 1996) for which the variables to be identified do not appear in the objective function, and the equilibrium system takes the form of a complementarity condition. We reformulate problem (10)-(14) in the next section and describe two simple algorithms for solving it.

4. Solution algorithms

The most prominent feature of an MPEC, and one that distinguishes it from a standard nonlinear program, is the presence of complementarity constraints such as (13). These constraints classify this class of mathematical programs as a nonlinear disjunctive (or piecewise) program and therefore carries with it a "combinatorial curse". This makes it very difficult to solve; the various methods proposed to date (see Luo *et al.* 1996) are categorized by the way the complementarity condition is handled.

In the following, we describe two intuitive reformulations of (10)-(14) as standard nonlinear programs. A primary motivation for this treatment is to exploit the availability of sophisticated nonlinear programming solvers such as CONOPT (Drud 1994), especially via the GAMS modeling language. This is in spite of the fact that traditional constraint qualifications are never satisfied, with the implication that the usual numerical methods for solving nonlinear programming problems may be expected to have some difficulties in their solution. We must note, however, that most practical problems in nonsmoooth mechanics (and we believe the present identification problem as well) do not have a large number of points which are nondifferentiable. It is therefore likely that standard algorithms, possibly with modest modifications, may work. Further, whilst there is no guarantee that the solution obtained represents a local minimum to MPEC (10)-(14), we wish to investigate numerically if our simple algorithms can provide reasonable solutions in practice.

We now briefly describe the two algorithms. At variance with Maier *et al.* (1982) and Nappi (1982), we adopt an l_1 norm instead of the square of the Euclidean norm for the error. Theoretical justification for a slight modification of this approach can be found in Mangasarian and Pang (1997).

4.1. Algorithm 1: penalty approach

The basic idea underlying the penalty approach for solving MPEC (10)-(14) consists in choosing a penalty parameter ρ, and converting the MPEC into the following nonlinear programming problem:

$$\min\ e^T u_{err} + \rho w^T \lambda \tag{15}$$

subject to:

$$F = C^T Q \tag{16}$$

$$Q = SCu - SN\lambda \tag{17}$$

$$w = -N^T Q + diag(B_2 h)\lambda + B_3 r \geq 0, \quad \lambda \geq 0 \tag{18}$$

$$h_l \leq h \leq h_u, \quad r_l \leq r \leq r_u \tag{19}$$

$$-u_{err} \leq u_m - B_1 u \leq u_{err}, \quad u_{err} \geq 0 \tag{20}$$

where e is an appropriate size vector of ones and $u_{err} \in \Re^k$ measures the absolute difference between recorded and calculated measurements.

The simple algorithm solves the penalized problem for successively higher values of ρ to force the complementarity term, which is nonnegative at feasible points, to zero. The attraction of this method is that each penalty subproblem is a standard nonlinear program and general purpose codes such as CONOPT can be used. The following pseudo-code further clarifies the algorithm:

Set: initial ρ (e.g., 10^{-4}), maximum number of iterations (*maxiter*), and $w^T\lambda$=100.
for i=1 to *maxiter*
  if $w^T\lambda \leq 10^{-6}$ exit
  solve nonlinear program (15)-(20)
  ρ=10ρ
end

We also provide in the Appendix a short GAMS model, based on the penalty approach, that implements Dempe's MPEC example (refer Luo *et al.* 1996, page 354). The GAMS modeling language used should be clear, even to someone who as never seen such models before. Paucity of space precludes us from listing more sophisticated, albeit realistic, models.

4.2. Algorithm 2: parametric approach

This second algorithm reformulates the MPEC via a smoothing approach on the complementarity constraints. This reformulation is similar to that described in Facchinei *et al.* (1996). The conditions (13) are replaced by the n equations

$$0 = \Phi_\mu(w, \lambda)_i = \phi_\mu(w_i, \lambda_i) \tag{21}$$

The function ϕ_μ has the property that $\phi_\mu(a, b)$=0 if and only if $a \geq 0$, $b \geq 0$, ab=μ, and hence can be used to replace the complementarity constraints by an equation. There are many such functions in the literature; in this paper we use the following smoothing of the Fischer function originally proposed in Kanzow (1996):

$$\phi_\mu(a, b) = a + b - \sqrt{a^2 + b^2 + 2\mu} \tag{22}$$

A key advantage of using μ>0 is that the resulting Eq. (21) is differentiable and hence a standard nonlinear programming solver can be used. However, to satisfy the conditions (13) we must parametrically decrease $\mu(i)$ to zero, each decrement making the equations more ill conditioned. Thus our reformulation is

$$\min e^T u_{err} \tag{23}$$

subject to:

$$F = C^T Q \tag{24}$$

$$Q = SCu - SN\lambda \tag{25}$$

$$w = -N^T Q + diag(B_2 h)\lambda + B_3 r \geq 0, \quad \lambda \geq 0, \quad \Phi_{\mu(i)}(w, \lambda) = 0 \tag{26}$$

$$h_l \leq h \leq h_u, \quad r_l \leq r \leq r_u \tag{27}$$

$$-u_{err} \leq u_m - B_1 u \leq u_{err}, \quad u_{err} \geq 0 \tag{28}$$

In essence, we now try to satisfy complementarity by decreasing the parameter $\mu(i)$ in (26) in a controlled manner through a series of major iterations, each of which corresponds to solving the nonlinear program given above. Although the nonnegativity constraints are not required in (26) we found them to be advantageous in practice. The key fact in the procedure is that the previous solution can be used to start the next iteration. The following pseudo-code elucidates this procedure:

Set: initial μ (e.g., 1), maximum number of iterations (*maxiter*), and $w^T\lambda$=100.
for i=1 to *maxiter*
  if $w^T\lambda \leq 10^{-6}$ exit
  solve nonlinear program (23)-(28)
  $\mu(i+1)=0.1\ \mu(i)$
end

The initial value of μ and its decrease for each major iteration is of course problem dependent. Obviously, too small a decrease will lead to slow convergence but possibly to a better optimum. Furthermore, there are difficulties associated with choosing an initial value either too large (in which case there may be no solution of the nonlinear program) or too small (which leads to a much more ill conditioned initial nonlinear program).

5. Numerical examples

We present two examples in this section to illustrate application of the algorithms. Both examples have been used in the literature (Maier *et al.* 1982, Nappi 1982) to illustrate the parameter identification problem. The models are too small for us to report in detail on computational times; all runs were carried out on a Pentium Pro 200. In any case, we were primarily interested in the robustness and predictive accuracy of the algorithms, rather than on their efficiencies. A criterion of $\leq 10^{-6}$ was adopted for complementarity.

5.1. Example 1

This example concerns the 11-bar truss (Nappi 1982) shown in Fig. 1. Assumed properties (kN, cm units) for generating the pseudo-experimental data with GAMS/PATH are: vertical and horizontal bars $-S$=900, r=309, h=90; diagonals $-S$=400, r=180, h=40. Further, each bar has the same properties in tension and compression, and hardening is of the noninteractive type (leading to a diagonal hardening matrix).

As in Nappi (1982), four load cases were used in the identification process as follows:
Case A: [500, −600, 0, −500, 0, 0, 0, 0]
Case B: [500, −600, 0, −500, 0, −550, 0, 0]
Case C: [500, −500, 0, −500, 0, 0, 0, 0]
Case D: [600, −600, 0, −600, 0, 0, 0, 0]

The same starting point was adopted for all runs, viz. r=10, h=10, w=1, λ=1, Q=1, $u_c=u_m$, u_{err}=0.1.

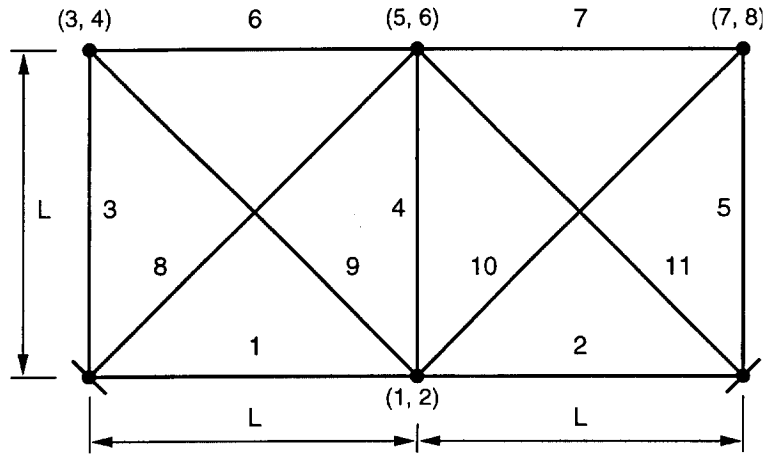

Fig. 1 Example 1: eleven bar truss

Table 1 summarizes the results obtained with both Algorithms 1 and 2; the times indicated are total GAMS/CONOPT times. These results indicate the comparable performance of both algorithms for this specific example. In particular, both algorithms find global solutions of the MPEC and identify the same parameters correctly. Algorithm 1 is somewhat faster than Algorithm 2. The accuracy of the identified parameters are to be expected in view of their "identifiability", which is only possible if the appropriate modes are "active". In particular, we expect that: (a) for load case A, with members {3, 4}, {8, 10, 11} activated, it is likely that both vertical/horizontal and diagonal parameters will be identified; (b) for load case B, with members {3}, {8, 9, 10, 11} activated, it is likely that only diagonal parameters will be identified; (c) for load case C, with members {3}, {8, 11} activated, diagonal parameters may be identified; and (d) for load case D, with members {1, 2, 3, 4}, {8, 10, 11} activated, it is likely that both vertical/horizontal and diagonal parameters can be identified. The results obtained confirm these predictions.

Table 1 Computational results for Example 1

Load Case	Algorithm	Vert. / Horiz.		Diagonal		Iter i	Time (secs)	ω
		r	h	r	h			
Exact	-	309	90	180	40	-	-	-
A	1	309.0	90.0	180.0	40.0	1	0.14	0
A	2	309.0	90.0	180.0	40.0	8	1.34	0
B	1	279.9	96.1	180.0	40.0	1	0.17	0
B	2	500.0	49.9	180.0	40.0	8	1.06	0
C	1	307.7	90.5	180.0	40.0	1	0.11	0
C	2	500.0	24.9	180.0	40.0	8	1.31	0
D	1	309.0	90.0	180.0	40.0	1	0.38	0
D	2	309.0	90.0	180.0	40.0	8	1.12	0

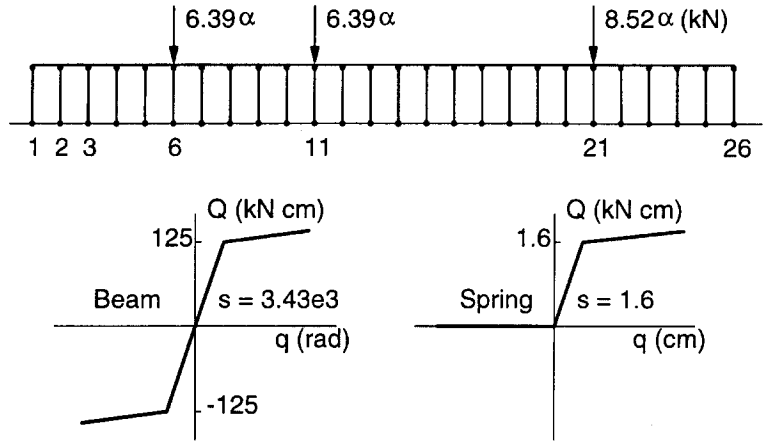

Fig. 2 Example 2: elastoplastic beam on elastoplastic foundation

5.2. Example 2

This example (Maier *et al.* 1982, Nappi 1982) concerns an elastoplastic beam on elastoplastic foundations (Fig. 2). The beam model consists of 24 identical lumped elastoplastic hinges at springs 2-25; the foundations are represented by 26 identical elastoplastic springs spaced at 32cm. Material properties (sagging and compression positive) are as indicated in Fig. 2; note that whilst a tensionless spring (i.e., r=0, h=0) is illustrated, some of our runs did not assume this. In order to simulate measured displacements, we again first ran a GAMS/PATH analysis model of the structure for the following yield limits and hardening parameters: r=[125, 125, 1.6, 0], h=[171.5, 171.5, 0.08, 0], with the first two elements of these vectors referring to beams and the last two to springs. Two load cases were used: α=1 and α=3.

A series of runs were made with GAMS/CONOPT models of Algorithms 1 and 2. Each run can be described by a 4 character code WXYZ with the following meanings:

W (measured freedoms): $a \equiv$ all, $b \equiv 1, 3, ..., 25$, $c \equiv 17, 19, ..., 25$
X (load level): $1 \equiv \alpha = 1$, $3 \equiv \alpha = 3$
Y (accuracy of measurement): $0 \equiv$ exact measurements, $5 \equiv$ increase u by 5%
Z (bounds on r and h): $t \equiv$ tight, $s \equiv$ semi-tight, $l \equiv$ loose

$t \equiv r_l$=[100, 100, 1.3, 0], r_u=[150, 150, 2, 0], h_l=[150, 150, 0, 0], h_u=[200, 200, 0.1, 0]
$s \equiv r_l$=[30, 30, 0, 0], r_u=[300, 300, 5, 0], h_l=[0, 0, 0, 0], h_u=[300, 300, 5, 0]
$l \equiv r_l$=[0, 0, 0, 0], r_u=[1e3, 1e3, 1e3, 1e3], h_l=[0, 0, 0, 0], h_u=[1e3, 1e3, 1e3, 1e3]

Table 2 and Table 3 display the computational results obtained with Algorithm 1 and Algorithm 2, respectively. We will not discuss these results in detail but will only make the following general comments.

(a) Both algorithms performed well. CONOPT managed to solve all the subproblems. In all cases, the same starting points (r=1, h=1, w=1, λ=1, Q=1, $u_c=u_m$, u_{err}=0.1) were adopted. Other nonlinear programming solvers that were tried had significant difficulties solving some of the subproblems.
(b) Obviously, the identification of any parameter is only possible if the corresponding yield

Table 2 Algorithm 1: computational results for Example 2

Run	$r(1)$	$r(2)$	$r(3)$	$r(4)$	$h(1)$	$h(2)$	$h(3)$	$h(4)$	Iter i	ω
Exact	125	125	1.6	0	171.5	171.5	0.08	0	-	-
*a*10*t*	126.2	100.0	1.60	-	150.0	150.0	0.08	-	1	0
*a*10*s*	117.6	53.0	1.60	-	300.0	0	0.08	-	1	0
*a*10*l*	84.6	53.0	1.60	0	874.1	0	0.08	0	1	0
*b*10*t*	126.0	100.0	1.60	-	155.2	150.0	0.08	-	1	0
*b*10*s*	117.6	53.0	1.60	-	300.0	0	0.08	-	1	0
*c*10*t*	126.2	100.0	1.60	-	150.0	150.0	0.08	-	1	0
*c*10*s*	135.1	30.0	1.42	-	4.7	300.0	0.19	-	7	0.034
*a*15*t*	123.3	100.0	1.53	-	200.0	150.0	0.10	-	7	0.358
*a*15*s*	117.1	55.0	1.53	-	300.0	0	0.10	-	7	0.358
*a*30*s*	125.0	125.0	1.60	-	171.5	171.5	0.08	-	1	0
*a*30*l*	125.0	125.0	1.60	0	171.5	171.5	0.08	0	1	0
*b*30*t*	125.0	125.0	1.60	-	171.5	171.5	0.08	-	1	0
*b*30*s*	110.9	165.8	1.31	-	175.3	0	0.09	-	6	3.16
*b*30*l*	19.5	0	3.83	0	1e3	0	0	0	6	24.18
*c*30*t*	100.5	123.2	1.60	-	189.0	182.8	0.08	-	1	0
*c*30*s*	78.4	71.8	0	-	183.1	0	0.13	-	1	0
*a*35*s*	124.6	124.8	1.60	-	162.8	163.5	0.08	-	7	0.322
*a*35*l*	123.2	126.3	1.60	0	163.1	160.4	0.08	0	7	0.268

mode is activated. For the α=1 case, the loading level activates one beam (in sagging) and 17 springs (7 in compression, 10 in tension). The *x*1*xx* results confirm that we can only identify, even for the tight bounds case, the spring parameters. On the other hand, for α=3, 10 beams (3 sagging, 7 hogging) and 25 springs (16 compression, 9 tension) are activated. The identification process for the *x*3*xx* series, as observed, is correspondingly much more successful.

(c) Proper bounds are of great importance in the identification process. Fortunately, in practice, guidance is provided by an *a priori* knowledge of the engineering structure. For instance, the loose bounds applied in the *xxxl* series are not physically very meaningful and are meant to test the behavior of the algorithms only; clearly we need to take advantage of the tensionless nature of the foundation.

(d) The number and position of measurements affect the accuracy of the identification process. For instance, very accurate identification is still possible, as for *b*30*s*, even when not all measurements are made. Obviously, an insufficient number of measurements, as in *c*30*t*, even with tight bounds, can lead to partial success only. Fortunately, in practice, the location of the loads will provide guidance as to where to best measure the displacements.

(e) Both algorithms appear to be able to provide reasonable values for the parameters even in the presence of imprecise measurements (*a*35*s* and *a*35*l*).

(f) It is interesting to note (compare the *c*30*t* results for Algorithms 1 and 2) that different sets of parameters may be obtained for the same global optimum ω=0.

(g) While it is premature to compare the two algorithms, we must mention that, from the

Table 3 Algorithm 2: computational results for Example 2

Run	$r(1)$	$r(2)$	$r(3)$	$r(4)$	$h(1)$	$H(2)$	$h(3)$	$h(4)$	Iter i	ω
Exact	125	125	1.6	0	171.5	171.5	0.08	0	-	-
*a*10*t*	126.2	100.0	1.60	-	150.0	200.0	0.08	-	8	0
*a*10*s*	134.9	53.7	1.60	-	0	0	0.08	-	8	0
*a*10*l*	134.9	1e3	1.6	0	0	564.8	0.08	0	8	0
*b*10*t*	126.2	100.0	1.60	-	150.0	200.0	0.08	-	8	0
*b*10*s*	134.9	53.6	1.60	-	0	0	0.08	-	8	0
*c*10*t*	126.2	100.0	1.60	-	150.0	200.0	0.08	-	13	0
*c*10*s*	134.9	53.5	1.60	-	0	300.0	0.08	-	8	0
*a*15*t*	123.3	100.0	1.53	-	200.0	200.0	0.10	-	8	0.358
*a*15*s*	117.1	59.3	1.53	-	300.0	300.0	0.10	-	8	0.358
*a*30*s*	125.0	125.0	1.60	-	171.5	171.5	0.08	-	9	0
*a*30*l*	125.0	125.0	1.60	0	171.5	171.5	0.08	0	8	0
*b*30*t*	125.0	125.0	1.60	-	171.5	171.5	0.08	-	9	0
*b*30*s*	125.0	125.0	1.60	-	171.5	171.5	0.08	-	8	0
*b*30*l*	125.0	125.0	1.60	0	171.5	171.5	0.08	0	9	0
*c*30*t*	101.3	133.0	1.61	-	188.2	156.0	0.08	-	8	0
*c*30*s*	100.5	203.3	1.67	-	186.8	0	0.08	-	9	0
*a*35*s*	124.6	124.8	1.60	-	162.8	163.5	0.08	-	8	0.322
*a*35*l*	123.1	126.3	1.59	0	163.1	160.4	0.08	0	8	0.268

results displayed in Tables 1 to 3, Algorithm 1 is significantly faster than Algorithm 2. The latter, however, while requiring more iterations, seems to give consistently better (smaller ω) optimum values. We also note that the correct choices of parameter settings in a particular problem instance may significantly affect the quality of the objective value found.

6. Conclusions

This paper briefly introduces a practically useful inverse problem of identifying both yield limits and hardening parameters for an important class of structural problems. On the basis of information on displacement response to given loads, the identification problem can be formulated as a special mathematical program known as an MPEC, involving complementarity constraints.

Two simple and promising algorithms based on a reformulation of the MPEC as standard nonlinear programs are proposed to take advantage of the availability of general purpose nonlinear optimization solvers. Our numerical experiments tend to validate the suitability of this approach. We point out that the nonlinear solvers we used were sophisticated implementations and furthermore that the GAMS environment allowed us to exploit the restarting capabilities of these codes.

However, in spite of these encouraging results, further work and improvements are still

needed. In particular, extensive application to actual realistic examples, especially in the area of quasibrittle fracture, are required; the robustness of both schemes should be improved; and there is also a need to obtain results which could, for instance, indicate how close the solution obtained is to a local minimum of the original MPEC. Of course, efficiently finding a global minimum to the inverse problem will remain a challenging research goal.

Ongoing research is aimed at providing tools to attain this goal. In particular, MPEC modeling formats within GAMS (Dirkse and Ferris 1997a) and an interface to the MATLAB programming environment (Dirkse and Ferris 1997b) are tools that will aid research in this frontier between algorithmic design and application to realistic structural problems.

Acknowledgements

This work was supported by the Australian Research Council and the National Science Foundation. We also thank Professor G. Maier for motivating this work.

References

Bolzon, G., Ghilotti, D. and Maier, G. (1997), "Parameter identification of the cohesive crack model", in *Material Identification Using Mixed Numerical Experimental Methods*, eds. H. Sol and C.W.J. Oomens, Kluwer Academic Publishers, Dordrecht, 213-222.

Bolzon, G. and Maier, G. (1998), "Identification of cohesive crack models for concrete on the basis of three-point bending", in *Computational Modelling of Concrete Structures*, eds. R. de Borst, N. Bicanic, H. Mang, G. Meschke, Balkema, Rotterdam, 301-310.

Brooke, A., Kendrick, D. and Meeraus, A. (1992), *GAMS: A User's Guide, Release 2.25*, Boyd & Fraser Publishing Company, Massachusetts.

Comi, C., Maier, G. and Perego, U. (1992), "Generalized variables finite element modelling and extremum theorems in stepwise holonomic elastoplasticity with internal variables", *Computer Methods in Applied Mechanics and Engineering,* **96**, 213-237.

Corradi, L. (1978), "On compatible finite element models for elastic plastic analysis", *Meccanica,* **13**, 133-150.

Corradi, L. (1983), "A displacement formulation for the finite element elastic-plastic problem", *Meccanica,* **18**, 77-91.

Cottle, R.W., Pang, J.S. and Stone, R.E. (1992), *The Linear Complementarity Problem*, Academic Press.

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

Dirkse, S.P. and Ferris, M.C. (1997a), "Traffic modeling and variational inequalities using GAMS", in *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, eds. P. Toint, M. Labbe, K. Tanczos and G. Laporte, NATO ASI Series F, Springer-Verlag, Berlin.

Dirkse, S.P. and Ferris, M.C. (1997b), "Modeling and solution environments for MPEC: GAMS & MATLAB", *Mathematical Programming Technical Report* 97-09, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin.

Drud, A. (1994), "CONOPT - a large-scale GRG code", *ORSA Journal on Computing,* **6**, 207-216.

Duvaut, D. and Lions, J.L. (1976), *Inequalities in Mechanics and Physics*, Springer, New York.

Facchinei, F., Jiang, H., and Qi, L. (1996), "A smoothing method for mathematical programs with equilibrium constraints", Research Report, Universita di Roma "La Sapienza", Roma, Italy.

Jiang, H., Ralph, D. and Tin-Loi, F. (1997), "Identification of yield limits as a mathematical program

with equilibrium constraints", *Proceedings 15th Australasian Conference on the Mechanics of Structures and Materials*, Melbourne, Australia, December 8-10, 1997, Eds R.H. Grzebieta, R. Al-Mahaidi and J.L. Wilson, Balkema, 399-404.
Kanzow, C. (1996), "Some noninterior continuation methods for linear complementarity problems", *SIAM Journal on Matrix Analysis and Applications,* **17**, 851-868.
Luo, Z.Q., Pang, J.S. and Ralph, D. (1996), *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press.
Maier, G. (1970), "A matrix theory of piecewise linear elastoplasticity with interacting yield planes", *Meccanica,* **5**, 54-66.
Maier, G. (1981), "Inverse problem in engineering plasticity: a quadratic programming approach", *Academia Nazionale dei Lincei*, Serie VIII, vol LXX, 203-209.
Maier, G., Giannessi, F. and Nappi, A. (1982), "Indirect identification of yield limits by mathematical programming", *Engineering Structures,* **4**, 86-98.
Mangasarian, O. and Pang, J.S. (1997), "Exact penalty functions for mathematical programs with equilibrium constraints", *Optimization,* **42**, 1-8.
Moreau, J.J. and Panagiotopoulos, P.D. (eds) (1988), *Nonsmooth Mechanics and Applications*, Springer, Vienna.
Moreau, J.J., Panagiotopoulos, P.D. and Strang, G. (eds) (1988), *Topics in Nonsmooth Mechanics*, Birkhäuser, Basel.
Nappi, A. (1982), "System identification for yield limits and hardening moduli in discrete elastic-plastic structures by nonlinear programming", *Applied Mathematical Modelling,* **6**, 441-448.
Panagiotopoulos, P.D. (1985), *Inequality Problems in Mechanics and Applications*, Birkhäuser, Basel.

Appendix: Example GAMS model

```
* Illustrate Penalty Method for Solving an MPEC
* Dempe's Example (Luo, Pang and Ralph 1996, 354)
* Solution of this model in 5 iterations
* f=31.250, x=1.002, y=0, w=0.998, z=1.001

variables f, x, y, z, w;

* Set lower bounds on y and w
y.lo=0; w.lo=0;

scalar ro 'Penalty parameter' / 1e-3 /;
equations obj, c1, c2;

* Define objective and constraints
obj .. f =e= sqr(x-3.5)+sqr(z+4)+ro*w*y;
c1 .. z-3+2*z*w =e= 0;
c2 .. z*z-x+y =e= 0;

model dempe / all /;
option nlp=conopt;

* Set starting values
x.l=1; y.l=1; w.l=1; z.l=1;

scalar comp / 1e4 /
```

```
* Iterate until complementarity satisfied
set iter / iter1*iter50 /;
loop(iter$(comp gt 1e-8),
   solve dempe using nlp minimizing f;
   comp=w.l*y.l;
   ro=10*ro;

display f.l, comp, x.l, y.l, w.l, z.l;
```